

Introduction au Deep Synergy Learning

Volume I
Chapitre 1 à 4

Mohammed Bouayoun

| | |
|--|-----------|
| Chapitre 1 : Introduction au Deep Synergy Learning (DSL) | 8 |
| 1.1. Contexte et Motivation | 8 |
| 1.1.1. Bref Historique de l'IA | 8 |
| 1.1.2. Émergence des Réseaux Profonds (Deep Learning) et Limites | 9 |
| 1.1.3. Des Approches Conventionnelles à la Synergie Informationnelle | 11 |
| 1.1.4. Positionnement du DSL dans le Panorama de l'IA | 12 |
| 1.1.5. Rôle Potentiel du DSL vers l'IA Forte (IAG)..... | 13 |
| 1.1.6. Objectifs et Contributions Clés du DSL | 15 |
| 1.2. Définitions et Concepts Préliminaires | 17 |
| 1.2.1. Qu'est-ce qu'une “Entité d'Information” ? | 17 |
| 1.2.2. Notion de “Synergie Informationnelle” | 19 |
| 1.2.3. Différence entre Interaction, Synergie et Corrélation | 21 |
| 1.2.4. Approche Hiérarchique vs Approche Auto-Organisée | 24 |
| 1.2.5. Réseaux Neuronaux Traditionnels vs Réseaux Synergiques | 28 |
| 1.2.6. Terminologies Employées dans le DSL | 31 |
| 1.2.7. Exemples Illustratifs de la Synergie dans la Nature..... | 36 |
| 1.3. Importance de l'Auto-Organisation | 40 |
| 1.3.1. Inspirations Biologiques et Cognitives..... | 40 |
| 1.3.2. Concepts Clés : Émergence, Auto-Régulation, Feedback | 43 |
| 1.3.3. Comparaison avec les Méthodes d'Apprentissage Classiques..... | 46 |
| 1.3.4. Le Rôle des Flux d'Information Multimodaux..... | 49 |
| 1.3.5. Évolution Dynamique et Adaptation en Continu..... | 52 |
| 1.3.6. Impacts sur la Robustesse et la Résilience des Modèles..... | 55 |
| 1.3.7. Perspectives pour une Approche plus Globale de l'Apprentissage | 59 |
| 1.4. Architecture Générale du DSL | 63 |
| 1.4.1. Principe de Base : Entités et Liens Synergiques | 63 |
| 1.4.2. Présentation du Synergistic Connection Network (SCN) | 65 |
| 1.4.3. Notion de Cluster et de Macro-Cluster..... | 67 |
| 1.4.4. Fonctions de Synergie : Distance, Similarité et Co-Information | 71 |

| | |
|--|------------|
| 1.4.5. Pondérations Adaptatives et Évolution Temporelle | 74 |
| 1.4.6. Interactions Directes et Indirectes | 77 |
| 1.4.7. Synergie binaire et n-aire : au-delà des relations deux à deux | 80 |
| 1.5. Pourquoi une Approche Synergique ? | 83 |
| 1.5.1. Avantages par Rapport aux Réseaux Neuronaux Profonds | 83 |
| 1.5.2. Gestion Naturelle de la Multi-modalité | 88 |
| 1.5.3. Flexibilité vis-à-vis des Données Incomplètes ou Bruitées | 93 |
| 1.5.4. Potentiel d'Auto-Évolution et d'Adaptation Continue | 97 |
| 1.5.5. Réduction de la Dépendance à la Supervision Humaine | 100 |
| 1.5.6. Création de Représentations Riches et plus Interprétables | 103 |
| 1.5.7. Intégration de Dimensions Symboliques ou Cognitives..... | 108 |
| 1.6. Applications Pressenties et Domaines Impactés | 116 |
| 1.6.1. Vision Artificielle et Reconnaissance d'Objets Complexes | 117 |
| 1.6.2. Analyse Audio et Traitement du Langage Naturel..... | 120 |
| 1.6.3. Robotique et Systèmes Intelligents Évolutifs..... | 124 |
| 1.6.4. Recommandation Personnalisée et Systèmes de Décision..... | 129 |
| 1.6.5. Surveillance, Diagnostic Médical et Anomalies | 133 |
| 1.6.6. Planification et Optimisation dans l'Industrie 4.0 | 137 |
| 1.6.7. Perspectives pour la Recherche Fondamentale en IA Forte..... | 141 |
| 1.7. Défis, Contraintes et Ouvertures | 145 |
| 1.7.1. Complexité Computationnelle et Scalabilité | 146 |
| 1.7.2. Qualité et Disponibilité des Données | 149 |
| 1.7.3. Développement d'Algorithmes d'Optimisation Appropriés..... | 152 |
| 1.7.4. Contrôle et Stabilité des Processus Auto-Organisés..... | 157 |
| 1.7.5. Interprétabilité et Explicabilité pour l'Humain | 161 |
| 1.7.6. Considérations Éthiques et Réglementaires | 164 |
| 1.7.7. Comparaisons Expérimentales avec d'Autres Approches | 168 |
| 1.8. Positionnement du DSL dans l'Évolution de l'IA | 174 |
| 1.8.1. IA Symbolique vs IA Sub-symbolique : Intégration Potentielle | 175 |

| | |
|--|-----|
| 1.8.2. DSL et Apprentissage Profond : Collaboration ou Substitution ? | 178 |
| 1.8.3. Approches Hybrides : DSL, RL (Reinforcement Learning) et Logique | 181 |
| 1.8.4. Rôle de la Mémoire et de l'Attention dans le DSL | 185 |
| 1.8.5. Tendances Futures : Vers une IA Forte, Consciente ? | 190 |
| 1.8.6. Effet Sur la Recherche Interdisciplinaire | 193 |
| 1.8.7. Exemple de Convergence : DSL & Neurosciences | 197 |
| Chapitre 2 : Fondements Théoriques et Origines du DSL..... | 201 |
| 2.1. Genèse Historique et Inspirations Multidisciplinaires | 201 |
| 2.1.1. Les Premières Pistes d'Auto-Organisation | 201 |
| 2.1.2. SOM, Hopfield et la Génération des Réseaux Associatifs | 204 |
| 2.1.3. Entre Neurosciences Computationnelles et Physique Statistique | 209 |
| 2.1.4. Précurseurs de la Synergie Multimodale | 214 |
| 2.1.5. Origine du DSL et Ses Premiers Manifestes..... | 217 |
| 2.2. Principes Mathématiques de Base..... | 224 |
| 2.2.1. Définition des Entités et de la Fonction de Synergie | 224 |
| 2.2.2. Mise à Jour des Pondérations : Formule Générale | 230 |
| 2.2.3. Règles de Parsimonie et Seuil de Connexion | 235 |
| 2.2.4. Notions d'États Internes et Auto-Organisation | 239 |
| 2.2.5. Exemples Illustratifs..... | 244 |
| 2.3. Hypothèses de Stabilité et Émergence de Clusters..... | 250 |
| 2.3.1. Point Fixe, Attracteurs : Définitions et Existence | 250 |
| 2.3.2. Attracteurs Multiples et Oscillations | 257 |
| 2.3.3. Analyse de la Formation de Clusters | 263 |
| 2.3.4. Influence du Bruit et des Perturbations | 268 |
| 2.3.5. Limites Théoriques et Questions Non Résolues | 272 |
| 2.4. Raccords avec la Physique Statistique et la Théorie des Systèmes Dynamiques . | 278 |
| 2.4.1. Systèmes Dynamiques Discrets ou Continus | 278 |
| 2.4.2. Parallèles avec les Modèles de Spin (Ising, Potts, Hopfield)..... | 284 |
| 2.4.3. Notion d'Énergie ou de Fonction Potentielle..... | 289 |

| | |
|--|-----|
| 2.4.4. Rôle de l’Inhibition et de la Saturation Synaptique | 295 |
| 2.4.5. Approches Hybrides et Collaboration Interdisciplinaire | 299 |
| 2.5. Perspectives Historiques et Liens avec l’IA Moderne | 304 |
| 2.5.1. Évolutions Récentes et Convergence avec le Deep Learning..... | 304 |
| 2.5.2. Extensions Symboliques et Neuro-Symboliques | 307 |
| 2.5.3. Apport en Robotique et Contrôle Adaptatif..... | 312 |
| 2.5.4. Comparaison avec d’Autres Paradigmes (RL, GNN, etc.)..... | 315 |
| 2.5.5. Transition vers les Chapitres Suivants | 319 |
| Chapitre 3 : Représentation et Modélisation des Entités d’Information | 324 |
| 3.1. Introduction..... | 324 |
| 3.1.1. Contexte et Motivation | 324 |
| 3.1.2. Objectifs du Chapitre | 327 |
| 3.1.3. Structure du Chapitre | 334 |
| 3.2. Principes Généraux de la Représentation dans le DSL | 339 |
| 3.2.1. Rôle Fondamental de la Représentation | 339 |
| 3.2.2. Critères de Choix | 342 |
| 3.2.3. Impact sur la Synergie et la Dynamique..... | 348 |
| 3.3. Représentations Sub-Symboliques | 355 |
| 3.3.1. Vecteurs et Embeddings..... | 355 |
| 3.3.2. Autoencodeurs, Transformers et Approches Avancées | 360 |
| 3.3.3. Calcul de la Synergie entre Vecteurs | 366 |
| 3.3.4. Gestion du Bruit et de l’Évolution des Embeddings | 373 |
| 3.3. Représentations Sub-Symboliques | 380 |
| 3.3.1. Vecteurs et Embeddings..... | 380 |
| 3.3.2. Autoencodeurs, Transformers et Approches Avancées | 385 |
| 3.3.3. Calcul de la Synergie entre Vecteurs | 391 |
| 3.3.4. Gestion du Bruit et de l’Évolution des Embeddings | 398 |
| 3.4. Représentations Symboliques | 405 |
| 3.4.1. Logiques, Règles et Ontologies | 405 |

| | |
|--|-----|
| 3.4.2. Calcul de la Synergie Symbolique | 410 |
| 3.4.3. Évolution Symbolique et Gestes des Contradictions | 419 |
| 3.5. Représentations Hybrides (Sub-Symbolique + Symbolique) | 428 |
| 3.5.1. Motivation de la Fusion | 428 |
| 3.5.3. Avantages et Défis | 440 |
| 3.6. Aspects Avancés et Études de Cas | 446 |
| 3.6.1. Hypergraphes et Synergie n-aire | 446 |
| 3.6.2. Structures Fractales de Représentation..... | 450 |
| 3.6.3. Études de Cas Illustratives..... | 455 |
| Exemple..... | 461 |
| 3.6.4. Comparaison Expérimentale..... | 462 |
| 3.7. Conclusion..... | 469 |
| 3.7.1. Synthèse des Contributions du Chapitre | 469 |
| 3.7.2. Limites et Pistes Futures | 471 |
| 3.7.3. Liens avec les Chapitres Suivants | 472 |
| 3.7.4. Vision Globale..... | 474 |
| Chapitre 4 : Synergies Émergentes et Auto-Organisation..... | 476 |
| 4.1. Introduction Générale | 476 |
| 4.1.1. Rappel du Contexte et des Objectifs | 476 |
| 4.1.2. Place et Contenu du Chapitre..... | 477 |
| 4.2. Règles de Mise à Jour Fondamentales | 482 |
| 4.2.1. Formulation Additive Classique | 482 |
| 4.2.1. Formulation Additive Classique | 485 |
| 4.2.2. Variantes Multiplicatives, Inhibition, etc..... | 489 |
| 4.2.3. Cas Multi-Entités Hétérogènes | 494 |
| 4.3. Émergence de Clusters et Attracteurs | 500 |
| 4.3.1. Formation Spontanée de Groupes..... | 500 |
| 4.3.2. Attracteurs Multiples et Bascules | 503 |
| 4.3.3. Stabilisation vs. Dynamisme Continu..... | 510 |

| | |
|--|-----|
| 4.4. Oscillations, Pseudo-Chaos et Méthodes de Contrôle | 521 |
| 4.4.1. Pourquoi des Oscillations ou un Pseudo-Chaos ? | 521 |
| 4.4.2. Stratégies pour Rompre les Oscillations | 526 |
| Approche algorithmique pour η, τ : essais, annealing, jacobienne | 532 |
| 4.4.3. Cas Stochastiques : Recuit, Perturbations | 532 |
| 4.5. Héritage de la Physique Statistique et des Systèmes Dynamiques | 539 |
| 4.5.1. Notions de Contraction, Point Fixe, Attracteurs..... | 539 |
| 4.5.2. Énergie ou Pseudo-Énergie | 547 |
| 4.5.3. Renvoi vers Chapitre 2.3 et 2.4 | 554 |
| 4.6. Extraction et Visualisation des Clusters Émergents..... | 559 |
| 4.6.1. Méthodes d'Observation..... | 559 |
| 4.6.3. Évolutions dans le Temps | 568 |
| 4.7. Exemples Concrets et Études de Cas..... | 572 |
| 4.7.2. Cas Robotique ou Multi-Agent..... | 577 |
| 4.7.3. Liens Pratiques | 583 |
| 4.8. Stabilisation Avancée : Couplages avec Chapitre 5 (Architecture SCN) | 590 |
| 4.8.1. Rôle de l'Architecture..... | 590 |
| 4.8.2. Gestions des Ressources et Threads | 595 |
| 4.8.3. Transition | 600 |
| 4.9. Conclusion et Transition | 608 |
| 4.9.1. Synthèse | 608 |
| 4.9.2. Ouverture..... | 609 |
| 4.9.3. Place du Chapitre 4 | 610 |

Chapitre 1 : Introduction au Deep Synergy Learning (DSL)

1.1. Contexte et Motivation

1.1.1. Bref Historique de l'IA

L'intelligence artificielle (IA) constitue un domaine de recherche qui s'inscrit dans une longue tradition, remontant à la première moitié du XXe siècle. Son évolution a été marquée par des jalons fondamentaux qui ont façonné la discipline telle qu'elle se présente aujourd'hui. Pour mieux appréhender l'émergence de l'Apprentissage Profond Synergétique (APS), ou **Deep Synergy Learning (DSL)** en anglais, il est essentiel d'examiner les principaux courants qui ont successivement vu le jour ou cohabité.

Des précurseurs conceptuels à la naissance de l'IA (années 1940–1950)

Les premières réflexions autour de l'intelligence artificielle trouvent leur origine dans les travaux d'Alan Turing, notamment à travers son article de 1950, *Computing Machinery and Intelligence*. Dans ce texte fondateur, il introduit le célèbre test de Turing, conçu pour évaluer la capacité d'une machine à simuler l'intelligence humaine dans le cadre d'une interaction textuelle.

C'est lors de la conférence de Dartmouth en 1956 que l'IA est officiellement définie comme un champ scientifique. Cette rencontre, organisée par John McCarthy, Marvin Minsky, Claude Shannon et Nathan Rochester, marque l'acte fondateur de la discipline. Elle établit les bases conceptuelles et méthodologiques nécessaires à la création de systèmes intelligents.

L'âge d'or de l'IA symbolique (années 1950–1970)

Les premières approches de l'IA reposaient sur la manipulation de symboles et de règles logiques pour modéliser la pensée humaine. Ces travaux, regroupés sous l'appellation **IA symbolique**, cherchaient à reproduire des processus cognitifs à travers des systèmes de règles formelles. Des programmes tels que *Logic Theorist* (1956) et *General Problem Solver* (1959) illustrent cette démarche en appliquant des algorithmes de recherche pour résoudre des problèmes mathématiques ou généraux.

Cependant, bien que performante dans des environnements fermés, l'IA symbolique montrait des limites lorsqu'il s'agissait de traiter des situations complexes ou incertaines. Cette incapacité à généraliser au-delà de scénarios spécifiques a révélé la nécessité d'approches complémentaires.

L'essor puis l'hiver des réseaux de neurones (années 1960–1980)

Parallèlement à l'IA symbolique, le courant connexioniste, basé sur les réseaux de neurones artificiels, gagne en popularité. Des chercheurs comme Frank Rosenblatt, Bernard Widrow et Marcian Hoff ont exploré le potentiel des réseaux d'unités élémentaires capables d'apprendre à partir de données, comme en témoigne le Perceptron (1958). Cependant, les travaux de Marvin Minsky et Seymour Papert, publiés en 1969, ont mis en lumière les limites des réseaux de neurones monocouches, incapables de traiter des problèmes non linéaires. Cette critique, combinée à

l'absence de méthodes efficaces pour entraîner des réseaux multicouches, a conduit au déclin temporaire de l'intérêt pour le connexionnisme, marquant le premier “hiver de l’IA”.

L'approche statistique et le renouveau du connexionnisme (années 1980–1990)

Le milieu des années 1980 voit la renaissance des réseaux de neurones grâce à la découverte de la rétropropagation, une méthode permettant d'entraîner efficacement des réseaux multicouches. Ce renouveau est accompagné d'une montée en puissance des approches statistiques de l'apprentissage automatique, telles que les modèles graphiques probabilistes et les machines à vecteurs de support (SVM). Ces outils élargissent les applications de l'IA à des domaines variés, notamment la vision par ordinateur et le traitement du langage naturel.

L'explosion du Deep Learning (années 2000–2010)

La fin des années 2000 marque l'avènement du Deep Learning, rendu possible par la convergence de trois facteurs majeurs. L'abondance des données massives issues d'Internet et des technologies numériques a offert un terrain fertile pour l'entraînement des modèles. L'émergence de ressources de calcul performantes, telles que les GPU et les TPU, a permis de traiter des volumes de données colossaux. Enfin, des progrès méthodologiques, incluant de nouvelles fonctions d'activation et des stratégies d'optimisation, ont renforcé l'efficacité des réseaux de neurones profonds. Ces avancées ont conduit à des percées significatives dans des domaines comme la reconnaissance d'images, la traduction automatique et les jeux stratégiques, illustrées notamment par le succès d'AlphaGo.

Malgré ces réussites, le Deep Learning présente certaines limites, notamment une forte dépendance aux données annotées, un manque d'interprétabilité et des difficultés à généraliser. Ces contraintes mettent en évidence la nécessité d'approches complémentaires.

Vers la synergie informationnelle et l'IA forte

Face aux limitations des méthodes actuelles, de nouveaux paradigmes ont émergé, mettant l'accent sur l'auto-organisation et les interactions dynamiques entre sources d'information. Le **Deep Synergy Learning (DSL)** s'inscrit dans cette dynamique en intégrant des notions de coopération et de coévolution au sein des systèmes intelligents. Cette approche vise à dépasser les modèles existants en favorisant une intelligence plus adaptable et autonome, ouvrant ainsi la voie à l'intelligence artificielle générale (IAG).

1.1.2. Émergence des Réseaux Profonds (Deep Learning) et Limites

L'émergence des réseaux profonds, ou **Deep Learning**, constitue un tournant fondamental dans l'histoire de l'intelligence artificielle. Bien que les réseaux de neurones artificiels aient été introduits dès les années 1960, ce n'est qu'au début des années 2010 qu'ils se sont imposés comme une technologie centrale grâce à plusieurs avancées majeures. Ces progrès incluent l'accroissement massif des données numériques, le développement de ressources matérielles puissantes comme les GPU, et l'introduction de méthodologies innovantes. Cependant, malgré ces succès, le Deep Learning continue de présenter des limitations importantes qui suscitent l'exploration de nouvelles approches.

L'essor du Deep Learning : facteurs clés

L'avènement du Deep Learning repose sur plusieurs éléments déterminants. D'une part, l'ère numérique a produit une quantité sans précédent de données issues de diverses sources, telles que des images, des vidéos, des textes ou des signaux. Ces ensembles de données volumineux, comme le dataset ImageNet, ont permis d'entraîner des réseaux plus profonds et plus complexes. D'autre part, l'utilisation des GPU, conçus initialement pour le traitement graphique, a permis d'accélérer les calculs massivement parallèles nécessaires à l'entraînement des modèles. Ces avancées matérielles ont été accompagnées de progrès méthodologiques, comme l'introduction de fonctions d'activation avancées, la régularisation par dropout ou encore la normalisation de lots (batch normalization). Enfin, les succès visibles dans des applications telles que la reconnaissance d'images, la traduction neuronale et les jeux stratégiques ont renforcé l'intérêt pour cette technologie.

Les forces des réseaux profonds

Les réseaux de neurones profonds ont apporté des avantages significatifs dans le traitement des données complexes. L'une de leurs principales forces réside dans leur capacité à extraire automatiquement des caractéristiques pertinentes à partir des données brutes, ce qui simplifie considérablement les tâches d'ingénierie des features. De plus, leur performance dans des domaines tels que la classification d'images ou la reconnaissance vocale dépasse de loin celle des méthodes traditionnelles. Lorsqu'ils sont entraînés avec suffisamment de données pertinentes, ces réseaux montrent une certaine aptitude à généraliser en identifiant des motifs complexes que d'autres approches ne peuvent pas détecter.

Limites et défis persistants

Malgré leurs performances impressionnantes, les réseaux profonds présentent plusieurs limites. Tout d'abord, leur dépendance aux données annotées est problématique, car l'annotation de grands volumes de données est coûteuse et parfois impraticable dans certains contextes. Ensuite, leur nature de boîte noire rend difficile l'interprétation des décisions prises, ce qui pose des problèmes d'explicabilité, notamment dans des secteurs critiques comme la santé ou la justice. Les réseaux profonds sont également vulnérables aux attaques adversariales, où des perturbations mineures peuvent induire des erreurs majeures. De plus, leur capacité de généralisation reste limitée lorsqu'ils sont confrontés à des contextes différents de ceux rencontrés durant l'apprentissage. Enfin, le coût énergétique élevé de l'entraînement de ces modèles soulève des préoccupations en termes de durabilité et d'accessibilité.

Perspectives et nouveaux paradigmes

Face à ces limitations, de nouvelles approches visent à repousser les frontières du Deep Learning. L'apprentissage peu supervisé ou auto-supervisé, qui réduit la nécessité de données annotées, est en plein essor. Par ailleurs, l'intégration du raisonnement symbolique avec les réseaux de neurones, dans une approche neuro-symbolique, pourrait permettre une intelligence plus transparente et explicable. Enfin, des recherches explorent des architectures auto-organisées où les informations interagissent de manière dynamique, rompant avec la rigidité des modèles actuels. Ces évolutions pourraient ouvrir la voie à des systèmes plus robustes, adaptatifs et proches d'une véritable intelligence artificielle générale.

1.1.3. Des Approches Conventionnelles à la Synergie Informationnelle

L'histoire de l'intelligence artificielle témoigne de la succession de grandes tendances méthodologiques, chacune cherchant à reproduire ou modéliser certains aspects de l'intelligence humaine ou animale. Parmi ces approches, les approches symboliques et les approches statistiques ou connexionnistes ont dominé. Les premières s'appuient sur la formalisation explicite de la connaissance à travers des règles et des faits, tandis que les secondes reposent sur l'apprentissage à partir de grandes quantités de données via des modèles tels que les réseaux de neurones artificiels. Bien que ces paradigmes aient permis des avancées considérables, ils montrent leurs limites lorsqu'il s'agit de traiter des systèmes complexes, dynamiques ou émergents. La synergie informationnelle s'inscrit dans une tentative de dépasser ces restrictions en proposant une approche inspirée des systèmes naturels.

Inspiration des systèmes complexes et de la nature

Les systèmes naturels, tels que les écosystèmes, les colonies d'insectes ou le cerveau humain, illustrent un principe fondamental selon lequel l'ensemble dépasse la somme de ses parties. Ces systèmes se caractérisent par l'auto-organisation, où des structures émergent sans planification globale, la robustesse grâce à la redondance des composants, et la capacité d'adaptation en réponse à des variations environnementales. Les entités locales interagissent dynamiquement à travers des signaux ou des mécanismes simples, conduisant à des comportements collectifs complexes. Transposé à l'intelligence artificielle, ce principe propose de considérer les informations comme des entités actives capables d'interagir, de coopérer et de s'adapter en permanence.

Limites des approches conventionnelles

Les approches classiques du Deep Learning, bien qu'efficaces, restent enfermées dans une structure hiérarchique rigide où les données circulent de manière linéaire ou avec des boucles limitées. Cela restreint leur capacité à modéliser des interactions complexes ou des phénomènes émergents. En particulier, dans des scénarios multi-modaux combinant des flux hétérogènes comme la vision, le texte ou l'audio, l'intégration repose souvent sur des mécanismes de fusion statiques qui ne permettent pas une réelle interaction entre les modalités. Ces architectures peinent également à s'adapter à des environnements dynamiques ou partiellement inconnus.

Concept de synergie informationnelle

La synergie informationnelle propose une vision radicalement différente. Les entités d'information, au lieu d'être des entrées passives dans un modèle prédéfini, deviennent des acteurs autonomes capables d'interagir et d'évoluer. Elles peuvent évaluer leur synergie avec d'autres entités, c'est-à-dire mesurer leur capacité à produire ensemble des informations plus riches que la somme de leurs contributions individuelles. En fonction de cette évaluation, elles peuvent nouer, rompre ou modifier des connexions, favorisant l'émergence de structures auto-organisées et dynamiques. Ce processus dépasse la simple adaptation des pondérations traditionnelles en permettant des changements topologiques dans les réseaux, où de nouvelles entités peuvent apparaître, fusionner ou disparaître.

Vers le Deep Synergy Learning (DSL)

Le **Deep Synergy Learning (DSL)** s'inscrit dans cette continuité en cherchant à unifier et enrichir les paradigmes existants. Alors que le Deep Learning met l'accent sur la profondeur des représentations, le DSL valorise la richesse et la dynamique des interactions entre les entités d'information. Les bénéfices attendus incluent une meilleure intégration des modalités multiples, une résilience accrue face à l'incertitude et une capacité d'apprentissage continu où le réseau peut évoluer en réponse à de nouvelles données sans nécessiter un réapprentissage complet. En favorisant des interactions flexibles et adaptatives, le DSL ambitionne de jeter les bases d'une intelligence artificielle plus proche des capacités cognitives humaines.

1.1.4. Positionnement du DSL dans le Panorama de l'IA

Le **Deep Synergy Learning (DSL)** s'inscrit dans la continuité des avancées en intelligence artificielle, tout en introduisant une approche novatrice fondée sur la coopération dynamique et l'auto-organisation des entités d'information. Pour mieux comprendre son originalité, il convient de le situer par rapport aux grandes tendances historiques et actuelles, à savoir l'**IA symbolique**, le **connexionnisme** incarné par le **Deep Learning**, et les **approches hybrides ou émergentes**.

Héritage des approches symboliques et connexionnistes

Le DSL se distingue des systèmes classiques d'IA symbolique, qui reposent sur la formalisation explicite de règles logiques et de bases de connaissances, en permettant l'émergence spontanée de structures et d'organisations adaptées au contexte. Contrairement à la rigidité des approches symboliques, il s'appuie sur des mécanismes auto-adaptatifs qui modifient les interactions en fonction des données disponibles. Par ailleurs, il prolonge l'héritage connexionniste en mettant l'apprentissage au centre de son fonctionnement. Alors que le Deep Learning repose sur des architectures statiques et hiérarchiques comme les CNN ou Transformers, le DSL introduit une dynamique d'interaction entre les entités, permettant une réorganisation continue des connexions en fonction de leur synergie.

Inspiration des systèmes complexes et émergents

Le DSL puise son inspiration dans les systèmes naturels complexes tels que les écosystèmes ou les réseaux neuronaux biologiques. Ces systèmes se caractérisent par des propriétés globales qui émergent de l'interaction locale d'éléments simples, sans planification centrale. Dans cette perspective, chaque entité d'information dans le DSL peut établir, rompre ou renforcer des connexions en fonction de la synergie qu'elle partage avec d'autres entités. Cette dynamique dépasse les schémas hiérarchiques classiques en introduisant des boucles de rétroaction et en intégrant des structures internes évolutives. Contrairement aux approches traditionnelles où les flux d'information sont souvent indépendants et combinés à un stade tardif, le DSL favorise un tissage constant des flux, améliorant ainsi la robustesse et l'adaptabilité.

Positionnement dans l'écosystème des approches hybrides

Le DSL se distingue également des approches neuro-symboliques qui intègrent explicitement des règles logiques dans des systèmes neuronaux. Bien qu'il ne repose pas directement sur des formalismes logiques, il permet l'émergence de règles implicites via des interactions coopératives

et des clusters auto-organisés. De plus, il participe à la quête d'un apprentissage continu en permettant l'évolution permanente des connexions et en favorisant une réorganisation constante en réponse aux nouveaux contextes. Cela en fait une solution prometteuse pour surmonter le problème de l'oubli catastrophique souvent rencontré dans l'apprentissage automatique.

Contribution unique à l'intelligence générale

L'IA forte, ou intelligence générale, se caractérise par sa capacité à s'adapter de manière flexible à des environnements variés et non balisés. Le DSL offre une contribution unique à cet objectif en introduisant le principe de coévolution des entités d'information. Cette adaptabilité intrinsèque repose sur l'auto-organisation et la synergie, qui sont considérées comme des éléments essentiels pour l'émergence d'une intelligence créative et autonome. Plutôt que de remplacer le Deep Learning, le DSL peut s'y intégrer en tant que niveau supplémentaire, enrichissant les représentations neuronales classiques par une interactivité et une plasticité accrue.

1.1.5. Rôle Potentiel du DSL vers l'IA Forte (IAG)

L'**Intelligence Artificielle Forte** (IAG), également désignée par intelligence artificielle générale, se définit comme la capacité d'une machine à réaliser des tâches cognitives humaines de manière autonome, flexible et adaptative, en s'appuyant sur une compréhension et un apprentissage comparable à ceux d'un être humain. Cet objectif ultime de l'intelligence artificielle reste encore largement théorique, malgré les avancées impressionnantes des systèmes actuels, notamment ceux basés sur le **Deep Learning**. Ces modèles, bien que performants dans des tâches spécialisées, peinent à atteindre le niveau de polyvalence et de plasticité requis pour prétendre à une véritable IA générale. Le **Deep Synergy Learning (DSL)** se positionne comme une approche novatrice et prometteuse, offrant un cadre conceptuel pour dépasser les limites des modèles traditionnels.

Repenser la Généralité en IA

Une intelligence générale ne se limite pas à exceller dans une tâche unique, mais doit pouvoir évoluer dans des environnements variés et imprévisibles. Le DSL se distingue par son aptitude à favoriser l'apprentissage multi-contextuel grâce à ses mécanismes d'**auto-organisation** et de **synergie informationnelle**. Ces mécanismes permettent au système de s'adapter en temps réel à des situations inédites, en reconfigurant les relations entre entités d'information. Contrairement aux systèmes conventionnels, qui restent figés dans leurs architectures ou nécessitent une ingénierie lourde pour évoluer, le DSL peut ajuster sa structure de manière dynamique.

La richesse des interactions entre les entités est une autre caractéristique essentielle du DSL. Plutôt que de juxtaposer des modules spécialisés sans réelle interaction, il propose une approche où chaque flux d'information, qu'il soit visuel, textuel ou sensoriel, interagit et co-évolue avec les autres. Ce type d'interconnexion fluide permet l'émergence de représentations conceptuelles globales, essentielles pour traiter la complexité du monde réel.

Enfin, le DSL intègre une dimension d'**adaptabilité évolutive**, permettant au système de non seulement apprendre, mais également désapprendre, réviser et transformer ses structures en fonction de nouvelles expériences. Cette adaptabilité dépasse celle des architectures classiques, qui, bien qu'elles ajustent leurs pondérations, conservent généralement une topologie statique.

Auto-Organisation et Plasticité Dynamique

L'un des principes fondamentaux du DSL est l'**auto-organisation**, inspirée des systèmes biologiques. À l'image du cerveau humain, il privilégie une architecture ouverte, où les entités d'information ne suivent pas une trajectoire linéaire prédéfinie, mais interagissent selon des schémas dynamiques et adaptatifs. Cette capacité à reconfigurer les flux internes permet au système de s'auto-réguler et d'évoluer en fonction des besoins contextuels.

La **plasticité dynamique** du DSL constitue un autre levier crucial pour atteindre l'IA forte. Elle dépasse l'ajustement traditionnel des pondérations en introduisant la possibilité de créer de nouvelles connexions, d'en supprimer d'obsoletes ou de réorganiser des clusters entiers. Cette plasticité ouvre la voie à une exploration continue et à une innovation constante, des caractéristiques indispensables pour développer une intelligence véritablement générale.

De la Spécialisation à la Flexibilité Généralisée

Les systèmes d'apprentissage profond ont démontré leur capacité à exceller dans des domaines spécifiques, comme la reconnaissance d'images ou la traduction automatique. Cependant, cette spécialisation s'accompagne d'une incapacité à transférer efficacement les connaissances d'un domaine à un autre. Le DSL propose une alternative en favorisant une **coopération dynamique** entre différentes modalités ou tâches, permettant une synergie globale entre sous-systèmes.

Ce paradigme ouvre la possibilité d'un transfert de connaissances fluide, où les acquis d'une modalité, comme l'analyse visuelle, enrichissent l'apprentissage dans une autre, telle que la compréhension du langage. Cette capacité de réutilisation et de réorganisation constitue une avancée majeure vers une **flexibilité généralisée**, essentielle pour le développement de l'IAG.

Vers une Architecture Cognitive Unifiée

Le cerveau humain se distingue par son haut degré d'interconnexion et sa capacité à faire émerger des significations à partir de l'interaction entre différentes aires sensorielles et associatives. Le DSL s'inspire de ce modèle en proposant une architecture où les interactions dynamiques entre les entités d'information jouent un rôle central. Cette approche favorise non seulement l'émergence de comportements complexes, mais également celle de **schémas cognitifs** riches et adaptatifs, proches de ceux observés dans les processus humains.

De plus, le DSL vise à dépasser la simple corrélation statistique en permettant l'émergence de concepts et de significations. Les interactions entre entités d'information, structurées par des mécanismes d'auto-organisation, contribuent à la construction de **nœuds sémantiques**, amorçant ainsi une véritable compréhension du monde environnant.

Défis et Perspectives

Bien que le DSL représente une avancée prometteuse, plusieurs défis doivent être relevés pour qu'il puisse concrétiser son potentiel en tant que moteur de l'IAG. L'**efficacité algorithmique** constitue un premier enjeu, car les processus d'auto-organisation et de réorganisation dynamique peuvent être coûteux en termes de calcul. Trouver des solutions d'optimisation est donc essentiel.

L'**équilibre entre stabilité et plasticité** est également un aspect critique. Le système doit éviter de se réorganiser de manière excessive, ce qui pourrait nuire à la cohérence de ses apprentissages, tout en maintenant une capacité suffisante d'adaptation.

Enfin, les questions d'**interprétabilité** et de **sécurité** demeurent cruciales. Une architecture aussi dynamique et complexe pourrait poser des problèmes de traçabilité et de contrôle, notamment dans des domaines sensibles comme la santé ou la finance.

1.1.6. Objectifs et Contributions Clés du DSL

Le **Deep Synergy Learning (DSL)** s'inscrit comme une évolution fondamentale dans le domaine de l'**intelligence artificielle (IA)**, mettant en avant une **approche dynamique** et hautement **adaptative** pour surmonter les limites des architectures hiérarchiques classiques, telles que celles du **Deep Learning**. Cette nouvelle méthodologie repose sur l'idée centrale de **synergie informationnelle**, où les entités d'information interagissent et s'organisent de manière autonome pour produire des représentations plus riches et efficaces. L'objectif principal du DSL est d'intégrer les principes issus du **connexionnisme** et des **systèmes complexes**, tout en introduisant des mécanismes avancés d'**auto-organisation** et d'**apprentissage évolutif**.

A. Cadre théorique et objectifs fondamentaux

Le DSL repose sur une modélisation qui privilégie les interactions **synergiques** entre les entités d'information. Contrairement aux architectures fixes, où les flux d'information sont généralement dirigés et figés, le DSL permet une **reconfiguration continue** des liens internes. Cette flexibilité est rendue possible grâce à des mécanismes d'**évaluation dynamique** de la synergie, notée $S(i, j)$, entre deux entités i et j . Mathématiquement, la synergie entre deux entités peut être décrite comme une fonction dépendant de leurs représentations respectives x_i et x_j :

$$S(i, j) = \Phi(x_i, x_j),$$

où Φ est une fonction d'évaluation qui peut être basée sur la corrélation, l'entropie conjointe, ou toute autre mesure adaptée au contexte.

L'objectif fondamental est de maximiser cette synergie globale dans le système, définie comme une somme pondérée des synergies individuelles :

$$\text{Synergie totale} = \sum_{i \neq j} w_{ij} S(i, j),$$

où w_{ij} représente les poids ajustables des connexions entre i et j . En optimisant cette fonction, le DSL favorise des **interactions pertinentes** et élimine les connexions inutiles ou redondantes, améliorant ainsi l'efficacité computationnelle et la robustesse des représentations générées.

L'un des objectifs clés du DSL est également d'offrir une plateforme pour l'**intégration multimodale**. En incorporant des données issues de différentes modalités, comme le texte, les images ou les signaux temporels, le système est capable d'unifier ces informations de manière **contextuelle**. Chaque modalité contribue activement à l'apprentissage global, tout en bénéficiant des enrichissements apportés par les autres, dans un cadre coopératif.

B. Contributions principales et innovations méthodologiques

Le DSL propose une contribution notable en introduisant un mécanisme d'**auto-organisation** et de **plasticité structurelle**. Contrairement aux approches classiques où les architectures sont déterminées à l'avance, le DSL permet une **reconfiguration dynamique** des connexions entre entités, suivant des critères d'efficacité synergiques. Cette adaptabilité repose sur un processus d'optimisation continue où chaque entité ajuste ses liens en fonction de la pertinence des informations échangées. Ce mécanisme peut être formalisé comme suit :

$$\frac{\partial w_{ij}}{\partial t} = \alpha \cdot \nabla S(i, j),$$

où α est un facteur d'apprentissage, et $\nabla S(i, j)$ désigne le gradient de la synergie entre i et j .

Une autre innovation majeure réside dans la gestion des **clusters auto-organisés**, qui sont des regroupements spontanés d'entités hautement synergiques. Ces clusters peuvent émerger, se fusionner ou se dissoudre en fonction des interactions observées, offrant une **modularité naturelle** au système. Cette capacité de création et de dissolution des clusters renforce la **résilience** et l'**évolutivité**, deux qualités cruciales pour les environnements complexes et dynamiques.

Enfin, le DSL se distingue par sa capacité à capturer des **représentations sémantiques profondes**, en favorisant l'émergence de **patrons globaux** à partir d'interactions locales. Cette propriété rapproche le DSL des processus cognitifs humains, où la compréhension découle souvent d'une synthèse d'informations contextuelles et émergentes.

C. Avantages du DSL

Le DSL présente des avantages significatifs par rapport aux paradigmes d'apprentissage traditionnels. L'un des principaux bénéfices est sa capacité à **maximiser la synergie informationnelle**, ce qui améliore la richesse et la robustesse des représentations apprises. Cette optimisation synergique permet au système d'exploiter pleinement la diversité des données, y compris dans des environnements multi-modaux. Par ailleurs, la **plasticité dynamique** du DSL garantit une adaptabilité exceptionnelle face à des scénarios évolutifs ou imprévisibles, tout en réduisant les besoins en supervision humaine. En outre, son cadre auto-organisé favorise une meilleure **résilience** aux perturbations, grâce à la redondance et à la flexibilité des connexions internes.

D. Limites et défis à surmonter

Malgré ses nombreux atouts, le DSL n'est pas exempt de défis. L'un des obstacles majeurs est la **complexité computationnelle** inhérente à l'évaluation continue des synergies et à la reconfiguration des connexions. Cette complexité pourrait limiter son application à grande échelle, en particulier dans des contextes où les ressources en calcul sont contraintes. De plus, la nature hautement dynamique et adaptative du DSL pose des **problèmes d'interprétabilité**, car il devient difficile de tracer ou d'expliquer précisément les décisions prises par le système. Enfin, le maintien d'un **équilibre entre stabilité et plasticité** est une question délicate. Une plasticité excessive peut provoquer une instabilité, tandis qu'une structure trop rigide risque de nuire à l'efficacité globale.

1.2. Définitions et Concepts Préliminaires

Dans cette partie, nous allons introduire les notions fondamentales qui sous-tendent le **Deep Synergy Learning (DSL)**. Alors que la section précédente (1.1) portait sur le contexte historique, les motivations et la place du DSL dans l'écosystème de l'IA, le présent segment (1.2) est consacré à la **structure conceptuelle** du DSL et aux **fondements** qui permettent d'en comprendre les mécanismes profonds.

Nous débuterons par la définition de ce qu'est une “**Entité d'Information**” (1.2.1), pierre angulaire du paradigme DSL. Nous poursuivrons en expliquant le concept de “**Synergie Informationnelle**” (1.2.2) et en clarifiant la différence entre interaction, synergie et simple corrélation (1.2.3). Nous verrons ensuite comment le DSL se démarque d'une **approche strictement hiérarchique** (1.2.4) et en quoi il diverge des **réseaux neuronaux traditionnels** (1.2.5). Enfin, nous préciserons la **terminologie** spécifique (1.2.6) et illustrerons les principes de synergie par des **exemples concrets** (1.2.7).

1.2.1. Qu'est-ce qu'une “Entité d'Information” ?

Le concept d'**entité d'information** est l'un des piliers fondamentaux du **Deep Synergy Learning**. Dans le **DSL**, une entité d'information ne se réduit pas à un simple point de données isolé, mais représente un véritable **objet d'apprentissage** doté de plusieurs propriétés.

Elle possède d'abord une **représentation interne**, qui peut prendre la forme d'un vecteur, d'un tenseur ou d'une fonction. Elle présente également des **caractéristiques dynamiques**, lui permettant d'évoluer, de se modifier et d'entretenir des liens synergiques avec d'autres entités. Enfin, elle dispose d'un **historique** ou d'une mémoire partielle de ses interactions passées, influençant ainsi son comportement futur et sa capacité d'adaptation.

Pour formaliser une entité d'information \mathcal{E} , on se place dans un espace vectoriel (ou parfois un espace de Hilbert plus général) :

$$\mathcal{E}_k \in \mathbb{R}^d, \quad \text{ou} \quad \mathcal{E}_k \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_p},$$

suivant la nature des données (vecteur, matrice, tenseur, etc.). Par exemple, une image peut être encodée en tant que tenseur 3D (hauteur \times largeur \times canaux de couleur), tandis qu'un signal audio pourra être représenté sous forme de séries temporelles dans \mathbb{R}^d .

Dans certains cas, la représentation peut également être probabiliste. Ainsi, \mathcal{E}_k peut être décrite par une distribution de probabilité $\mathcal{P}_k(\mathbf{x})$ sur un certain espace $\mathbf{x} \in \mathcal{X}$. L'important est de conserver la possibilité de **mesurer** la distance, la similarité ou la divergence entre deux entités :

$$\text{dist}(\mathcal{E}_k, \mathcal{E}_m) \quad \text{ou} \quad \text{sim}(\mathcal{E}_k, \mathcal{E}_m).$$

Outre la représentation brute, une entité peut avoir des **paramètres internes** (poids, biais, etc.) qui se modifient selon le temps ou selon les interactions :

$$\boldsymbol{\theta}_k = \{\theta_{k,1}, \theta_{k,2}, \dots, \theta_{k,\ell}\}.$$

Ces paramètres influent sur le “comportement” de l’entité, c’est-à-dire sa manière de calculer des **scores de similarité** ou des **fonctions de sortie**. On peut aussi décrire un **état** interne $\mathbf{s}_k(t)$ évoluant avec t , le temps (ou la phase d’apprentissage) :

$$\mathbf{s}_k(t) \in \mathbb{R}^d,$$

indiquant, par exemple, le niveau de confiance ou les caractéristiques discriminantes apprises jusqu’à l’instant t . Cet état peut servir de base à la mise à jour des **connexions synergiques** entre l’entité \mathcal{E}_k et d’autres entités \mathcal{E}_m .

Dans le cadre du DSL, nous pouvons associer à chaque entité \mathcal{E}_k un **ensemble** de composants :

$$\mathcal{E}_k = (\mathbf{x}_k, \mathbf{s}_k, \theta_k, \dots),$$

où :

- $\mathbf{x}_k \in \mathbb{R}^d$ est la représentation courante (ex. : vecteur de caractéristiques, image encodée).
- \mathbf{s}_k est l’état interne dynamique (optionnel ou modulable).
- θ_k représente des **paramètres d’ajustement** ou **d’apprentissage**.
- D’autres composants pourraient inclure la **mémoire** (historique), les **métadonnées**, etc.

Cette formulation a pour but de **généraliser** la notion de “neurone” ou de “vecteur de données” pour en faire une entité d’apprentissage **active** et **adaptative**, au cœur des mécanismes d’interaction synergiques.

Pour illustrer concrètement la notion d’entité d’information, considérons la tâche de **reconnaissance de situations** dans une vidéo associée à un flux audio. Dans ce scénario, l’entité $\mathcal{E}_{\text{visuelle}}$ peut représenter un **descripteur d’image** extrait par un réseau de neurones convolutionnel (CNN), enrichi de **paramètres** relatifs à la forme ou à la pose des objets détectés. De son côté, l’entité $\mathcal{E}_{\text{auditive}}$ peut regrouper une **carte d’intensité fréquentielle** (spectrogramme) et un **état** décrivant la tonalité ou le niveau de bruit ambiant.

Ces deux entités ne sont pas de simples **blocs de données isolés**. Elles sont conçues pour **interagir**, **se synchroniser** ou même **fusionner** lorsque leur **synergie** (au sens de la section 1.2.2) atteint un niveau suffisamment élevé. Autrement dit, si les informations issues des canaux visuel et auditif s’enrichissent mutuellement, elles ont la possibilité de renforcer leurs liens et, potentiellement, de s’intégrer au point de former une entité commune, apte à traiter des signaux audiovisuels de manière coordonnée. Cette démarche souligne la **plasticité** du Deep Synergy Learning, qui autorise une reconfiguration permanente des relations entre entités pour améliorer la représentation globale de la scène.

En définitive, l’entité d’information constitue le **nœud élémentaire** du DSL. C’est à **travers** elle et **par** elle que les mécanismes de synergie prennent forme, permettant l’émergence de structures d’apprentissage plus complexes. Le **design** même de chaque entité, qu’il s’agisse de sa représentation (vecteur, tenseur, distribution de probabilité), de son état ou de ses paramètres d’ajustement, détermine directement l’**expressivité** et l’**efficacité** de l’apprentissage au sein du réseau. Le choix judicieux de ces attributs, adapté à la modalité (vision, audio, texte, etc.) et à la

tâche visée, facilite la formation de **liaisons synergiques** fructueuses et, par conséquent, contribue à la robustesse et à la performance globale du système DSL.

1.2.2. Notion de “Synergie Informationnelle”

L'un des concepts centraux du Deep Synergy Learning (DSL) est la **synergie informationnelle**. Il s'agit de la capacité de deux (ou plusieurs) entités d'information à produire, ensemble, un **contenu** ou une **performance** impossible à atteindre (ou significativement moins bonne) si elles agissaient de manière isolée. Dans un cadre mathématique, la synergie se formalise par une mesure qui évalue l'**apport mutuel** entre les entités. Plus cette mesure est élevée, plus les entités concernées s'enrichissent mutuellement, amplifiant leur pouvoir de représentation ou de décision.

Définition générale. Considérons deux entités d'information, \mathcal{E}_i et \mathcal{E}_j . En première approximation, on peut définir la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ comme une fonction qui quantifie à quel point la prise en compte conjointe de \mathcal{E}_i et de \mathcal{E}_j **améliore** un critère d'apprentissage (la prédiction d'une variable cible, la qualité d'une représentation, etc.).

$$S(\mathcal{E}_i, \mathcal{E}_j) = f(\mathcal{E}_i, \mathcal{E}_j),$$

où $f(\cdot)$ peut être :

- Une **mesure d'entropie conjointe** (ou de co-information) en théorie de l'information,
- Un **gain de performance** par rapport à une référence (p. ex. différence de log-vraisemblance),
- Une **fonction de similarité/distance** qui prend en compte des aspects non linéaires et adaptatifs.

Cette fonction f doit être conçue pour refléter la notion que “le tout est plus que la somme des parties”. Ainsi, il est d'usage de considérer qu'une **haute synergie** indique que l'association $\{\mathcal{E}_i, \mathcal{E}_j\}$ est nettement plus informative que chacune des entités prise isolément.

En théorie de l'information, on peut s'appuyer sur l'**entropie conjointe** et la **co-information**. Par exemple, si \mathbf{X}_i et \mathbf{X}_j sont les variables aléatoires (représentant respectivement les entités \mathcal{E}_i et \mathcal{E}_j), on définit :

$$I(\mathbf{X}_i; \mathbf{X}_j) = H(\mathbf{X}_i) + H(\mathbf{X}_j) - H(\mathbf{X}_i, \mathbf{X}_j),$$

où $H(\cdot)$ est l'entropie (de Shannon, ou d'autres formes d'entropie plus générales). Cette quantité $I(\mathbf{X}_i; \mathbf{X}_j)$ mesure l'**information mutuelle** entre \mathbf{X}_i et \mathbf{X}_j . Toutefois, l'information mutuelle standard ne distingue pas toujours la **synergie de la redondance**.

Pour caractériser la synergie stricto sensu, plusieurs travaux de théorie de l'information proposent des mesures de **co-information** plus élaborées, voire des “Partial Information Decomposition” (PID), qui visent à séparer la part de redondance et la part de synergie :

$$\text{Synergie}(\mathbf{X}_i, \mathbf{X}_j) = I_{\text{PID}}^{(\text{syn})}(\mathbf{X}_i; \mathbf{X}_j \mid \mathbf{Y}),$$

où \mathbf{Y} peut être une troisième variable (cible à prédire) ou un contexte. Dans le cadre du DSL, il est donc pertinent d'utiliser, lorsque c'est possible, des **métriques entropiques** pour quantifier la contribution **non triviale** de chaque couple d'entités.

Une autre approche consiste à définir la synergie comme le **gain** (en termes de fonction objectif ou de performance) obtenu lorsqu'on associe deux entités, par rapport à leur utilisation séparée :

$$S(\mathcal{E}_i, \mathcal{E}_j) = \Delta(\mathcal{E}_i, \mathcal{E}_j) = \Phi(\{\mathcal{E}_i, \mathcal{E}_j\}) - [\Phi(\{\mathcal{E}_i\}) + \Phi(\{\mathcal{E}_j\})],$$

où $\Phi(\cdot)$ est un **score** ou une **mesure** de la qualité du système (ex. : taux de classification, log-likelihood, etc.). Dans ce cas :

- $S(\mathcal{E}_i, \mathcal{E}_j) > 0$ signifie qu'il y a véritablement une **valeur ajoutée** à combiner \mathcal{E}_i et \mathcal{E}_j .
- $S(\mathcal{E}_i, \mathcal{E}_j) < 0$ indique qu'il y a **inhibition** ou dégradation mutuelle.
- $S(\mathcal{E}_i, \mathcal{E}_j) = 0$ suggère une **indépendance** ou une simple addition sans synergie.

Cette formulation est souvent utilisée en pratique, car elle s'aligne directement sur un **objectif** (objectif supervisé, critère d'optimisation non supervisé, etc.). On peut de plus pondérer cette synergie par un facteur adaptatif, en tenant compte du **contexte temporel** ou **des autres entités** impliquées.

Contrairement aux approches linéaires ou statiques, le DSL prévoit que la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ soit **évolutive** au cours du temps. En d'autres termes, le réseau peut réévaluer en continu la contribution mutuelle de \mathcal{E}_i et \mathcal{E}_j . Matériellement, cela se traduit par la mise à jour d'une **pondération synergiques** $\omega_{i,j}(t)$:

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta \cdot [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \cdot \omega_{i,j}(t)],$$

où η est un taux d'apprentissage, τ un terme de régularisation (ou d'oubli). Plus la synergie entre deux entités est forte, plus leur lien s'intensifie. Au contraire, si ce lien n'apporte guère de valeur ajoutée (ou est carrément nuisible), sa pondération peut diminuer et aller jusqu'à **rompre** la connexion.

Définition Synergie binaire, ternaire, et n-aire. Dans sa version la plus simple, on considère la synergie entre **paires** d'entités $(\mathcal{E}_i, \mathcal{E}_j)$. Toutefois, nombre de scénarios exigent d'évaluer la synergie entre plusieurs entités simultanément. Dans ce cas, on généralise S à un ensemble $\{\mathcal{E}_{k_1}, \dots, \mathcal{E}_{k_m}\}$. On parle alors de **synergie n-aire**, dont la mesure n'est pas forcément la somme des synergies binaires. En effet, il se peut qu'**une triple** $\{\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c\}$ dégage une synergie supérieure à la somme des synergies de ses paires :

$$S(\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c) > S(\mathcal{E}_a, \mathcal{E}_b) + S(\mathcal{E}_b, \mathcal{E}_c) + S(\mathcal{E}_a, \mathcal{E}_c).$$

Ce phénomène traduit la nature **émergente** du DSL, où des ensembles d'entités peuvent coopérer de manière non triviale pour engendrer de nouvelles représentations ou actions.

Exemple scénario multimodal

Soit

- $\mathcal{E}_{\text{visuelle}}$ (extraction de caractéristiques d'une image),
- $\mathcal{E}_{\text{auditives}}$ (traits de voix, intonation),
- $\mathcal{E}_{\text{textuelles}}$ (mots-clés extraits d'une transcription).

Si $\mathcal{E}_{\text{visuelle}}$ et $\mathcal{E}_{\text{textuelle}}$ ont peu d'information en commun, leur **information mutuelle** peut être faible. Pourtant, prises ensemble, elles peuvent produire un **contexte** (ex. : "lieu de la scène + thèmes abordés verbalement") qui aide à l'interprétation des sons (détection d'émotion). Autrement dit, c'est l'**intersection** de ces informations qui devient cruciale, expliquant une **synergie** plus forte lorsqu'on combine ces trois entités plutôt que deux à deux.

Importance pour le DSL.

Le concept de **synergie informationnelle** est ce qui **différencie** le **DSL** d'un simple système de propagation de données entre couches. Contrairement à une transmission passive, le **DSL** repose sur des interactions dynamiques où les entités cherchent activement des partenaires synergiques, ajustent leurs pondérations en fonction du niveau de synergie et forment des **clusters** ou micro-réseaux autour des synergies les plus fortes. Ce processus favorise également la fusion ou l'évolution des entités pour optimiser la coopération, un mécanisme qui sera approfondi dans les chapitres suivants.

Ainsi, la synergie agit comme un **moteur** d'auto-organisation et de **dynamique adaptative**, permettant au réseau de se **restructurer** au fil du temps, en valorisant les combinaisons d'entités les plus porteuses d'information ou de gain de performance.

1.2.3. Différence entre Interaction, Synergie et Corrélation

Lorsqu'on étudie les relations entre différentes entités d'information, il est essentiel de faire la distinction entre **interaction**, **corrélation** et **synergie**. Ces notions sont parfois utilisées de façon interchangeable, mais elles renvoient à des réalités mathématiques et conceptuelles différentes. Comprendre ces nuances permet de mieux cerner l'originalité du Deep Synergy Learning (DSL) et la portée de son concept de « synergie informationnelle ».

1.2.3.1. Interaction : une relation générique

Le terme **interaction** désigne de manière générale l'influence mutuelle que peuvent exercer deux éléments (ou plus) l'un sur l'autre. D'un point de vue mathématique, on parle souvent d'**interaction** lorsque le comportement (ou la fonction) d'un système dépend de l'état de plusieurs variables de manière **non indépendante** :

$$f(\mathbf{x}_1, \mathbf{x}_2) \neq f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2).$$

Par exemple, dans un modèle statistique de type régression, l'**effet d'interaction** entre deux variables se traduit par la présence d'un terme produit $\mathbf{x}_1 \times \mathbf{x}_2$.

Une interaction ne garantit pas nécessairement un effet bénéfique ou un « plus » collectif ; elle se borne à signaler que l'état ou la valeur prise par \mathbf{x}_2 modifie l'effet de \mathbf{x}_1 (et inversement).

Ainsi, dans le DSL, de simples **interactions** peuvent exister entre des entités d'information sans pour autant impliquer une **synergie** (cette dernière requérant un effet d'émergence véritable, voir plus bas).

1.2.3.2. Corrélation : dépendance statistique (souvent linéaire)

La **corrélation** (au sens commun) mesure le degré de **dépendance statistique** entre deux variables, souvent réduit à la **corrélation linéaire** de Pearson :

$$\rho(\mathbf{X}, \mathbf{Y}) = \frac{\text{cov}(\mathbf{X}, \mathbf{Y})}{\sigma_{\mathbf{X}} \cdot \sigma_{\mathbf{Y}}}$$

avec $\text{cov}(\cdot, \cdot)$ représente la covariance, $\sigma_{\mathbf{X}}$ et $\sigma_{\mathbf{Y}}$ désignent l'écart-type de \mathbf{X} et \mathbf{Y} .

Une corrélation élevée ($\rho \approx 1$ ou $\rho \approx -1$) signifie que deux variables évoluent de façon similaire (linéairement liée), tandis qu'une corrélation nulle ($\rho \approx 0$) indique l'absence de dépendance linéaire (mais pas forcément l'absence de dépendance tout court).

Remarque : Dans un cadre non linéaire, d'autres mesures (mutual information, distance correlation, etc.) peuvent s'avérer plus pertinentes que la simple corrélation linéaire.

Corrélation \neq synergie.

Dans le cas d'une corrélation forte, on observe souvent une **redondance**. Si \mathbf{X} prédit bien \mathbf{Y} , alors \mathbf{Y} n'apporte pas nécessairement de nouvelle information. La corrélation peut aussi être trompeuse (corrélation de variables bruitées, effet de causalité inverse, variables cachées...).

Dans le DSL, deux entités très corrélées peuvent d'ailleurs être moins intéressantes (peu de gain) que deux entités faiblement corrélées, mais dont la **combinaison** génère un contenu supplémentaire.

Ainsi, une **corrélation forte** n'implique pas forcément une **synergie** ; et inversement, deux entités peuvent ne pas être corrélées mais créer, ensemble, un effet émergent.

La **synergie**, telle que définie dans le DSL, suppose un **gain** ou une **valeur ajoutée** lorsque les entités se combinent, au-delà de ce qu'elles apportent chacune de leur côté. Autrement dit :

$$S(\mathcal{E}_i, \mathcal{E}_j) > 0 \Rightarrow \text{La combinaison } \{\mathcal{E}_i, \mathcal{E}_j\} \text{ vaut plus que la somme séparée.}$$

- **Synergie \neq simple interaction** : L'interaction signale simplement une dépendance réciproque, alors que la synergie suppose qu'un nouveau niveau de fonctionnalité ou d'information émerge.

- **Synergie ≠ redondance** : Deux variables très similaires (corrélées) ont peu de synergie, car prendre l'une ou l'autre n'ajoute pas grand-chose à la décision globale.
- **Synergie ≠ coïncidence** : Les coïncidences peuvent être éphémères et non reproductibles. La synergie implique un **effet régulier** et **réel** sur l'optimisation ou la représentation interne du système.

Matériellement, dans un modèle où la fonction de coût \mathcal{L} est à minimiser (ou la fonction de performance Φ à maximiser), la synergie entre \mathcal{E}_i et \mathcal{E}_j s'exprime souvent comme :

$$\Delta_{ij} = \Phi(\{\mathcal{E}_i, \mathcal{E}_j\}) - [\Phi(\{\mathcal{E}_i\}) + \Phi(\{\mathcal{E}_j\})].$$

- Si $\Delta_{ij} > 0$, on parle de **synergie positive** (la combinaison est plus utile que la simple juxtaposition).
- Si $\Delta_{ij} < 0$, il y a **inhibition** ou **redondance néfaste**, et le couplage des entités se révèle contre-productif.
- Si $\Delta_{ij} \approx 0$, cela signifie qu'elles n'apportent pas grand-chose l'une à l'autre au regard de la tâche considérée.

Pour mieux illustrer ces différences, on peut imaginer un **diagramme** représentant trois situations :

- **Corrélation (redondance)** : Les deux entités (A et B) apportent presque la même information.
- **Interaction** : A et B se modifient mutuellement, mais sans forcément créer une nouvelle dimension.
- **Synergie** : La combinaison A + B engendre un nouveau potentiel (ex. : l'ajout d'un flux audio + flux visuel crée un contexte multimodal plus riche que n'importe lequel des flux pris isolément).

Les **cas concrets** illustrent différentes situations où la synergie joue un rôle déterminant dans l'efficacité des interactions entre entités.

Dans un **cas de redondance sans synergie**, deux capteurs de température placés au même endroit fournissent des mesures quasi identiques. Bien qu'ils soient fortement corrélés, leur apport informationnel est redondant, ce qui signifie qu'en utiliser un seul est aussi pertinent que d'en prendre deux.

Dans un **cas d'interaction sans synergie**, certaines protéines en biologie interagissent, par exemple lorsqu'une bloque l'autre, mais cette interaction ne conduit pas nécessairement à un fonctionnement plus efficace pour l'organisme. L'interaction est présente, mais elle ne génère pas de gain de performance ou d'organisation.

Dans un **cas de synergie forte**, le traitement de la parole illustre un exemple significatif. La combinaison de la lecture labiale, qui analyse les mouvements des lèvres, et du signal acoustique, qui capte l'information sonore, améliore nettement la reconnaissance par rapport à l'utilisation

isolée de l'un ou l'autre des signaux. Cet effet est particulièrement visible en environnement bruyant, où la complémentarité entre les deux sources d'information permet une meilleure robustesse du système.

Dans le **Deep Synergy Learning**, on cherche précisément à **favoriser** la création de synergies positives et à **réduire** (voire éliminer) les liens qui relèvent de la simple redondance ou d'interactions stériles. Les règles de mise à jour des **pondérations synergiques** (détaillées dans les chapitres dédiés) sont conçues pour ajuster dynamiquement les connexions entre entités en fonction de leur synergie. Elles permettent de **renforcer** les liens entre entités lorsque $\Delta_{ij} > 0$, favorisant ainsi les interactions bénéfiques. À l'inverse, elles visent à **diminuer** ou rompre les connexions lorsque la corrélation entre entités ne procure aucun gain ou engendre un effet négatif, c'est-à-dire lorsque $\Delta_{ij} < 0$. Enfin, ces règles facilitent la **détection** de synergies **n-aires**, où plusieurs entités interagissent pour former des **micro-réseaux auto-organisés**, optimisant ainsi la structure adaptative du **SCN**.

Cette démarche permet d'éviter l'explosion combinatoire (en évaluant toutes les combinaisons) grâce à un **mécanisme dynamique** où les liens se forment ou se défont au fil de l'apprentissage, suivant les feedbacks de performance ou des indicateurs entropiques.

Dans le DSL, c'est précisément cette notion de synergie, mesurée et mise à jour en continu, qui permet de construire des **clusters** d'entités coopératives, de faire émerger de **nouvelles représentations**, et de potentialiser la **résilience** du système face à la variabilité des données. Les sections ultérieures reviendront sur la façon dont ce mécanisme de synergie se met en place dans une **approche auto-organisée** (1.2.4) et comment il se distingue des **réseaux neuronaux traditionnels** (1.2.5).

1.2.4. *Approche Hiérarchique vs Approche Auto-Organisée*

Les approches hiérarchiques traditionnelles, largement utilisées dans le Deep Learning, reposent sur la succession de **couches** (layers) qui transforment progressivement les données d'entrée jusqu'à aboutir à une sortie (une prédiction, une classification, etc.). À chaque couche, on opère une **composition** de fonctions (le plus souvent linéaires, suivies de non-linéarités comme ReLU ou sigmoid). Par opposition, l'**approche auto-organisée** (telle qu'on la retrouve dans le Deep Synergy Learning, DSL) met l'accent sur la **capacité du réseau à reconfigurer ou réorganiser** sa structure interne en fonction des synergies détectées, plutôt que de s'en tenir à une architecture rigide et prédefinie.

Dans cette section, nous allons approfondir cette différence en examinant les principes fondamentaux de l'approche hiérarchique, pour mieux comprendre comment le DSL, en tant qu'approche auto-organisée, propose une alternative à la fois plus **dynamique** et plus **adaptative**.

1.2.4.1. Fondements de l'approche hiérarchique

Dans un **réseau hiérarchique** traditionnel (tel qu'un réseau de neurones profond), le **principe du traitement en cascade** impose aux données x de traverser une succession de transformations linéaires et non linéaires :

$$\mathbf{h}^{(1)} = f^{(1)}(\mathbf{x}), \quad \mathbf{h}^{(2)} = f^{(2)}(\mathbf{h}^{(1)}), \quad \dots, \quad \mathbf{h}^{(L)} = f^{(L)}(\mathbf{h}^{(L-1)}),$$

où $\mathbf{h}^{(\ell)}$ représente la “représentation cachée” extraite à la couche ℓ , et $f^{(\ell)}$ désigne un opérateur paramétrique (incluant poids et fonction d’activation). Dans ce schéma, l’information circule essentiellement **de bas en haut**, et les éventuelles boucles de rétroaction (feedback top-down) demeurent limitées ou spécialisées, comme dans les architectures RNN ou LSTM.

Cette organisation induit une **séparation des rôles** :

- Les **premières couches** traitent des descripteurs “bas niveau” (par exemple, repérer des bords pour une image, des phonèmes pour un signal audio, etc.).
- Les **couches intermédiaires** approfondissent la combinaison de ces descripteurs, extrayant des motifs plus complexes.
- Les **dernières couches** produisent la **décision finale** (classe, score, etc.).

Toutefois, cette hiérarchie s’accompagne d’une **certaine rigidité**. Une fois le nombre de couches, la taille de chacune, et la nature des connexions (dense, convolutionnelle, récurrente...) choisis, la **structure** du réseau reste figée pendant la phase d’apprentissage. Seuls les **poids** sont ajustés par descente de gradient ou par l’une de ses variantes, tandis que la topologie globale demeure invariable.

1.2.4.2. Limites de l’approche hiérarchique

Les **réseaux hiérarchiques** classiques présentent plusieurs limites marquantes. Ils restent fortement **dépendants** à la supervision, nécessitant de grandes quantités de données annotées pour ajuster leurs poids internes. À mesure que leur architecture croît, une **prolifération exponentielle** du nombre de paramètres alourdit le coût en calcul et en mémoire, tout en rendant le réseau plus difficile à **interpréter** et à **déboguer**.

Ces réseaux souffrent également d’un **manque d’adaptabilité**. Lorsqu’un **domain shift** survient, c’est-à-dire un changement dans la distribution des données, il devient nécessaire de procéder à un réapprentissage ou à un ajustement considérable, faute de mécanismes permettant une **reconfiguration dynamique**. Enfin, leur **faible modularité** constitue une autre contrainte. Même si les couches s’empilent, elles ne peuvent pas **échanger librement** en dehors des cheminements rigides imposés par l’architecture.

En résumé, un réseau hiérarchique fonctionne efficacement dans des contextes précis, lorsque de grands volumes de données annotées sont disponibles, mais il ne propose pas de **remodelage** structurel en cours d’apprentissage ni de recours proactif à des **synergies** entre sources d’information hétérogènes.

1.2.4.3. Principes de l'approche auto-organisée

Dans une **approche auto-organisée**, caractéristique du **DSL**, la **plasticité topologique** joue un rôle central. Plutôt que de reposer sur des couches fixes, les **entités d'information** peuvent spontanément **former des clusters**, **établir ou rompre des liaisons**, et même **fusionner** lorsque leur synergie atteint un niveau suffisant (cf. sections 1.2.2 et 1.2.3). L'architecture n'est donc plus imposée de manière rigide, mais se construit et se reconstruit au fil de l'apprentissage, s'adaptant en fonction des besoins et de l'évolution des données.

Cette **évolution dynamique** repose sur l'adaptation continue des pondérations synergiques $\omega_{i,j}$ suivant une règle telle que

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η est un taux d'apprentissage, τ un coefficient de régularisation et $S(\mathcal{E}_i, \mathcal{E}_j)$ la synergie entre les entités \mathcal{E}_i et \mathcal{E}_j . Dans ce cadre, une synergie **positive** renforce la connexion, tandis qu'une synergie **négative** l'affaiblit. Ainsi, les entités dont la coopération est bénéfique sont encouragées à établir (ou consolider) leurs liens, tandis que les connexions moins productives s'éteignent naturellement.

Au fil de ce processus d'ajustement, des **micro-réseaux** ou **clusters auto-organisés** émergent dès lors que les synergies mutuelles s'élèvent entre certaines entités. Ces agrégats peuvent apparaître, se scinder ou disparaître, reflétant les évolutions des données ou l'arrivée d'interactions nouvelles. Par ailleurs, cette liberté structurelle favorise une **coopération multi-flux**, en autorisant les entités visuelles, auditives, textuelles, etc. à s'influencer **directement**, sans passer par un chemin prédefini de "couches". Par exemple, une entité auditive peut détecter à l'instant t une forte synergie avec une entité visuelle et, de ce fait, former un cluster pour la durée nécessaire ; plus tard, si les conditions changent, elle peut s'éloigner de ce groupe pour établir d'autres coopérations plus pertinentes.

1.2.4.4. Modélisation mathématique d'une architecture auto-organisée

Dans une **architecture auto-organisée**, on ne définit plus un enchaînement linéaire $\mathbf{h}^{(1)} \rightarrow \mathbf{h}^{(2)} \rightarrow \dots$; à la place, on conçoit un **graphe** $G(t)$ dont les **nœuds** sont les entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ et dont les **arêtes** représentent les pondérations synergiques $\omega_{i,j}(t)$. L'évolution du réseau se décrit alors par une fonction de mise à jour :

$$G(t+1) = \mathcal{U}[G(t), S(\cdot, \cdot)],$$

où \mathcal{U} tient compte des **critères de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. Le système prend ainsi la forme d'un **Système Dynamique Non Linéaire**, se réorganisant de manière à privilégier les **combinatoires** d'entités jugées les plus utiles ou performantes.

Il est fréquent d'ajouter un mécanisme d'**énergie libre** ou de **coût global** :

$$\mathcal{J}(G) = - \sum_{i,j} \omega_{i,j} S(\mathcal{E}_i, \mathcal{E}_j) + \alpha \|\boldsymbol{\omega}\|^2,$$

de façon à **régulariser** la taille du réseau et à éviter que le nombre de connexions ne “flambe” de manière excessive. Dans ce cadre, la mise à jour peut s’effectuer via une **descente de gradient** (ou un algorithme d’optimisation inspiré des systèmes complexes, comme un algorithme génétique ou un recuit simulé), conduisant progressivement à une **organisation** qui valorise les synergies tout en restreignant les liaisons redondantes.

1.2.4.5. Avantages et défis de l’auto-organisation

Dans une **approche auto-organisée**, l'**adaptabilité** constitue un atout majeur. Le réseau peut s’ajuster en continu face à l’arrivée de nouvelles données ou à un changement de distribution sans qu’il soit nécessaire de repenser entièrement son architecture. La **multi-modalité** y est gérée de manière native. Les entités issues de différentes sources comme l’audio, l’image ou le texte peuvent **s’influencer mutuellement** et **former des clusters** multimodaux. Cette capacité favorise la **fusion** des flux d’informations et l’exploitation de leurs synergies. Cette dynamique ouverte introduit également un **potentiel créatif**. L’**émergence** de combinaisons inédites entre entités permet de révéler des **patrons** jusque-là invisibles et difficilement capturables par des architectures hiérarchiques classiques.

Toutefois cet avantage s’accompagne de plusieurs **défis** majeurs. La **complexité de contrôle** peut devenir un obstacle si aucun mécanisme de régulation comme le facteur τ ou des pénalités n’est mis en place. Un excès de connexions ou des boucles auto-renforcées peuvent entraîner des oscillations. Le **coût de calcul** représente également un défi. L’évaluation de la synergie entre un grand nombre d’entités nécessite l’usage d’heuristiques ou de méthodes parcimonieuses pour rester viable en pratique. L’**interprétabilité** peut aussi poser problème. Même si l’auto-organisation tend à faire émerger des clusters plus significatifs, l’évolution dynamique et permanente de la structure complique l’analyse détaillée du fonctionnement interne du réseau.

1.2.4.6. Comparaison synthétique

| Caractéristique | Approche Hiérarchique | Approche Auto-Organisée (DSL) |
|---------------------------------|---------------------------------|---|
| Architecture | Fixe (définie a priori) | Flexible (graphe évolutif) |
| Propagation de l’info | Principalement feed-forward | Libre (coopération directe entre entités) |
| Formation des connexions | Statique (paramètres ajustés) | Dynamique (création / rupture de liens) |
| Apprentissage | Descente de gradient classique | Mise à jour des synergies (pondérations) |
| Multimodalité | Fusion tardive (généralement) | Intégration native, clusters multimodaux |
| Adaptation continue | Limité (fine-tuning, transfert) | Fort (reconfiguration à la volée) |
| Exemples | CNN, RNN, Transformers | Synergistic Connection Network (SCN) |

Les sections suivantes (1.2.5 et suivantes) reviendront sur la comparaison plus directe entre les **réseaux neuronaux traditionnels** et les **réseaux synergiques**, tout en introduisant la terminologie spécifique (1.2.6) et des **exemples** (1.2.7) illustrant la pertinence de l’auto-organisation dans différents contextes naturels.

1.2.5. Réseaux Neuronaux Traditionnels vs Réseaux Synergiques

Dans les sections précédentes, nous avons présenté les notions de **synergie informationnelle**, de **corrélation** et de **plasticité** structurelle dans une approche **auto-organisée**. Il est maintenant temps de faire un **parallèle** entre, d’une part, les **réseaux neuronaux profonds** (ou traditionnels) tels qu’on les connaît en apprentissage profond (Deep Learning) et, d’autre part, les **réseaux synergiques** comme envisagés dans le DSL (Deep Synergy Learning). Cette comparaison aidera à mettre en lumière ce que le DSL apporte de différent par rapport aux architectures classiques (CNN, RNN, Transformers, etc.).

1.2.5.1. Structure et dynamique d’apprentissage

Dans les **réseaux neuronaux traditionnels**, la structure est définie dès le départ. Le nombre de couches et leur type, qu’elles soient convolutionnelles, récurrentes ou fully-connected, sont déterminés à l’avance ainsi que l’agencement de ces couches et la manière dont elles s’enchâînent. Chaque couche s’appuie sur la représentation produite par la précédente, imposant une **hiérarchie** explicite. L’apprentissage se fait en ajustant les poids et les biais via une descente de gradient ou l’une de ses variantes (Adam, RMSProp, etc.), tandis que la **propagation de l’information** suit principalement un chemin **feed-forward** ; même lorsque des boucles internes existent (RNN, LSTM), elles demeurent cantonnées à la topologie imposée.

À l’inverse, dans les **réseaux synergiques** (au sein du DSL), la topologie se veut **flexible et évolutive** : on ne parle plus de “couches” stricto sensu, mais d’un **ensemble d’entités d’information** liées entre elles par des pondérations **synergiques** (voir 1.2.2 et 1.2.4). Ces pondérations ne sont pas seulement ajustées ; elles peuvent aussi être créées, renforcées ou rompues, selon la synergie détectée. Cette possibilité de **reconfiguration** permanente marque la différence. Le réseau ne se contente pas d’empiler des couches mais s’**auto-organise** en **clusters** lorsque la synergie l’exige. De plus, l’information ne circule pas selon une progression linéaire ; elle peut **transiter** entre toutes les entités jugées “synergiques”, adoptant ainsi une **approche distribuée** plus proche d’un écosystème vivant que d’un pipeline hiérarchisé.

1.2.5.2. Mesure de la performance et critères d’apprentissage

Dans les **objectifs classiques** du Deep Learning, on minimise une **fonction de coût** $\mathcal{L}(\theta)$ (par exemple, l’entropie croisée ou la MSE), à l’aide d’une **backpropagation** qui calcule les gradients. Les performances sont ensuite mesurées selon des métriques comme la **précision**, le **rappel** ou le **F1-score**, en fonction du type de tâche (classification, régression, etc.).

En revanche, dans une **approche DSL**, ces **objectifs traditionnels** (par exemple, la précision en classification) coexistent avec des **fonctions de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ qui orientent l'apprentissage. Les mises à jour des pondérations synergiques $\omega_{i,j}$ tiennent compte de ces scores de synergie, **favorisant** les liaisons dont la synergie s'avère significative, tout en permettant la **création**, le **renforcement** ou la **dissolution** de connexions. On peut également définir une **fonction globale** $J(G)$, laquelle agrège la **somme** (ou autre forme d'agrégation) des synergies et **pénalise** la multiplication de connexions redondantes. Le réseau se comporte alors comme un **système dynamique**, visant à concilier la **minimisation** d'une perte liée aux tâches classiques (p. ex. une fonction de classification) et la **maximisation** de la synergie informationnelle au sens large.

1.2.5.3. Comparaison de la gestion de la multimodalité

Dans les **réseaux neuronaux traditionnels**, la **multimodalité** est gérée en définissant à l'avance des voies de traitement spécifiques pour l'audio, l'image ou le texte. Ces différentes modalités sont ensuite **fusionnées** à un certain stade du réseau, soit au niveau de couches intermédiaires, soit plus souvent par une fusion tardive vers la fin du pipeline. L'architecture doit donc être **explicitement conçue** pour chaque canal (par exemple, un CNN dédié à l'image, un RNN ou un Transformer pour le texte, puis un module spécialisé pour agréger les différents flux). La synergie potentielle entre ces canaux se découvre **indirectement**, via la backpropagation, mais la structure globale du réseau — et la manière dont les canaux se croisent — reste imposée de l'extérieur.

À l'inverse, dans les **réseaux synergiques** du DSL, les **entités** associées à divers canaux (audio, visuel, textuel, etc.) ont la possibilité de se “**découvrir**” **mutuellement** au fil de l'apprentissage. Si l'audio et l'image présentent une forte synergie, elles peuvent **former un cluster multimodal** de manière autonome, sans qu'une couche de fusion spécifique ne soit paramétrée au préalable. Cette approche facilite la **co-évolution** des représentations. Si le flux visuel est perturbé par du bruit à un moment donné, l'entité visuelle peut s'appuyer davantage sur les canaux texte ou audio, à condition qu'une synergie élevée soit détectée. Ainsi, la multimodalité se développe de façon **organique**, guidée par la recherche de gains effectifs de performance ou d'information.

1.2.5.4. Rôle de la rétropropagation et alternatives

Dans les **réseaux neuronaux profonds** classiques, la **rétropropagation** constitue le mécanisme standard pour ajuster les **poids** de couche en couche ; on dispose alors d'une **architecture** explicitement définie de bout en bout et d'un **objectif** scalaire unique qui oriente la descente de gradient (p. ex. l'entropie croisée). En revanche, dans un **DSL**, les **mises à jour** des pondérations synergiques se réalisent de manière **distribuée**, souvent selon des règles plus locales (inspirées, par exemple, d'approches “Hebbiennes généralisées” ou d'une évaluation directe des gains de performance obtenus). Un **objectif global** peut persister (comme un taux de reconnaissance), mais la **découverte** de synergies s'opère fréquemment dans des configurations plus indépendantes du gradient global.

La **rétropropagation** n'est pas pour autant **exclue**. Il est possible d'adopter un **système hybride** où la backpropagation s'applique à certains sous-modules, tandis que la **formation** et la **reconfiguration** du graphe synergique suivent des lois d'auto-organisation distinctes. Cette

approche offre une **flexibilité** accrue. Certaines parties du réseau continuent d'être ajustées finement par le gradient, tandis que le réseau conserve la capacité de **découvrir** et de **renforcer** localement des liaisons synergiques au-delà des contraintes d'une architecture fixe.

1.2.5.5. Robustesse et adaptation continue

Dans les **réseaux neuronaux traditionnels**, la **robustesse** dépend essentiellement de la qualité du jeu d'entraînement et de plusieurs mécanismes de **régularisation** (dropout, batch normalization, etc.). L'**adaptation** à un nouveau domaine s'effectue souvent par un transfert d'apprentissage (transfer learning), suivi d'un **fine-tuning** partiel ou complet des poids du réseau. Cependant, ce procédé peut exposer le système au risque de **catastrophic forgetting** lorsqu'il est utilisé pour apprendre en continu une succession de tâches. Les poids ajustés pour les nouvelles données ont alors tendance à effacer les connaissances acquises précédemment.

Dans les **réseaux synergiques**, au contraire, la **structure évolutive** permet au système d'**allouer** de nouvelles entités ou de **renforcer** certains liens pour absorber plus facilement un **changement** de données, sans exiger un réapprentissage complet de l'ensemble du réseau. Les **clusters** déjà formés pour des tâches précédentes peuvent coexister dans la nouvelle configuration, au lieu d'être remplacés ou écrasés. Ainsi, un réseau synergique peut mieux **retenir** l'expérience passée (réduisant d'autant le **catastrophic forgetting**) et faire preuve d'une plus grande **flexibilité** quand son environnement ou sa mission évoluent.

1.2.5.6. Réseaux traditionnels et synergiques : cohabitation possible ?

Les **réseaux synergiques** n'ont pas vocation à **remplacer** purement et simplement les architectures neuronales traditionnelles. Au contraire, divers **scénarios de cohabitation** sont envisageables. On peut, par exemple, adopter une **approche hybride**, dans laquelle un pipeline CNN (pour l'image) ou Transformer (pour le texte) extrait des **représentations** initiales ; ces représentations deviennent ensuite des **entités** au sein d'un réseau synergique, lequel peut alors coopérer et se reconfigurer de manière plus libre.

Dans certains **systèmes complexes**, on peut aussi instaurer une **transition progressive**, en commençant par des couches de feature extraction classiques, puis en insérant une **couche synergique** à un stade où les divers canaux se croisent. De cette façon, on préserve la puissance des modèles traditionnels pour l'extraction de caractéristiques tout en intégrant la logique auto-organisée et adaptative du DSL à un niveau plus élevé.

Enfin, il est possible de développer des **extensions spécialisées**, par exemple un composant auto-organisé dédié à la **fusion multimodale** ou à la **gestion de multiples contextes**, tandis que la classification finale demeure assurée par un réseau fully-connected ordinaire. L'essentiel est d'exploiter la **flexibilité** des réseaux synergiques dans les domaines où ils excellent — par exemple, l'émergence dynamique de clusters — tout en s'appuyant sur l'expérience accumulée des architectures neuronales traditionnelles.

1.2.5.7. Synthèse et perspectives

| Aspect | Réseaux Neuronaux Traditionnels | Réseaux Synergiques (DSL) |
|-----------------------------|--|---|
| Topologie | Fixe, pré-spécifiée | Évolutive, auto-organisée |
| Propagation | Hiérarchique, feed-forward | Dispersée, multidirectionnelle |
| Apprentissage | Backpropagation end-to-end | Règles locales + mise à jour synergie |
| Évolution temporelle | Nécessite du re-training pour s'adapter | Adaptation dynamique à la volée |
| Gestion multimodale | Fusion tardive ou intermédiaire, souvent manuelle | Fusion spontanée via synergie et création de clusters |
| Robustesse | Vulnérabilité à l'overfitting, besoin de régulariser | Auto-régulation via le feedback de synergie |
| Applications | Classification, régression, vision, NLP... | Idem, mais avec en plus la souplesse et l'auto-organisation |

En conclusion, les **réseaux neuronaux traditionnels** et les **réseaux synergiques** diffèrent principalement par la **structure**, la **dynamique d'apprentissage** et la **capacité d'auto-organisation**. Le **Deep Synergy Learning** apporte une philosophie plus **organique**, inspirée des systèmes complexes, pour que l'intelligence artificielle puisse gérer la **variabilité**, la **multimodalité**, et s'auto-adapter en continu.

La section suivante (1.2.6) clarifiera la **terminologie** propre au DSL — notamment les notions de **clusters**, **entités**, **pondérations synergiques**, etc. — puis nous verrons (1.2.7) des **exemples illustratifs** tirés de la nature ou d'applications concrètes, afin de matérialiser les principes évoqués dans ce chapitre.

1.2.6. Terminologies Employées dans le DSL

Au fil des sections précédentes, plusieurs notions-clés sont apparues pour décrire les principes du **Deep Synergy Learning (DSL)**. Il est important de les clarifier et de les organiser en un vocabulaire cohérent, car ces termes forment la **boîte à outils conceptuelle** indispensable pour aborder les mécanismes internes et les applications pratiques du DSL. Dans cette section, nous passons en revue les principaux termes et leur signification, en soulignant les liens entre eux.

1.2.6.1. Entité d'Information (ou “Information Entity”)

Définition. Dans le DSL, une **entité d'information** (souvent notée \mathcal{E}_i) représente l'unité fondamentale du système. Contrairement à un simple vecteur de données, une entité est un *objet d'apprentissage* pouvant inclure :

- Une **représentation** (par ex. un vecteur, un tenseur, ou même une distribution).
- Des **paramètres internes** (θ_i) et un **état** ($\mathbf{s}_i(t)$).

- Un **historique** (ou “mémoire”) de ses interactions passées.

C'est à travers ces entités que s'établissent les **synergies** et que se construit la dynamique de l'apprentissage. En pratique, toute source d'information (une image, un signal audio, un embedding textuel, etc.) peut être encapsulée sous forme d'entité.

1.2.6.2. Synergie (ou “Synergy”)

Définition. La **synergie** entre deux (ou plusieurs) entités est la mesure de la **valeur ajoutée** qu'elles obtiennent en coopérant, par rapport à ce qu'elles pourraient réaliser indépendamment (voir 1.2.2). Elle se note souvent $S(\mathcal{E}_i, \mathcal{E}_j)$ pour les paires, et peut être généralisée à des ensembles $\{\mathcal{E}_{k_1}, \dots, \mathcal{E}_{k_m}\}$.

Formes de mesure.

- **Informationnelle** : Basée sur l'entropie, l'information mutuelle, ou d'autres métriques de la théorie de l'information.
- **Basée sur la performance** : Différence de score (classification, regression, etc.) quand on associe \mathcal{E}_i et \mathcal{E}_j .
- **Hybride** : Combinaison d'un critère d'information et d'un critère de performance.

La synergie est la **force motrice** du **DSL**. Elle oriente la création, la rupture ou le renforcement des connexions entre entités, comme détaillé dans la section sur les **pondérations synergiques**.

1.2.6.3. Pondérations Synergiques (ou “Synergistic Weights”)

Définition. Les **Pondérations Synergiques** notées $\omega_{i,j}(t)$, ce sont les **coefficients** qui caractérisent la relation dynamique entre deux entités \mathcal{E}_i et \mathcal{E}_j à l'instant t .

Souvent modélisée par une équation du type

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η est un taux d'apprentissage, τ un terme de régularisation, et $S(\mathcal{E}_i, \mathcal{E}_j)$ la synergie entre les entités.

Les pondérations synergiques constituent la **matrice d'adjacence** d'un *graphe évolutif* :

$$W(t) = [\omega_{i,j}(t)]_{i,j}.$$

Elles déterminent quelles entités sont fortement liées (hautes synergies) et lesquelles le sont moins voire pas du tout (synergie quasi nulle).

1.2.6.4. Cluster (ou “Micro-Réseau”)

Définition. Un **cluster** est un **sous-ensemble** d’entités qui présentent entre elles une synergie élevée, formant ainsi une structure cohérente et **auto-organisée**.

Les entités $\{\mathcal{E}_1, \dots, \mathcal{E}_k\}$ tendent à se regrouper si leurs **pondérations synergiques** mutuelles sont supérieures à un certain seuil θ , ou si elles maximisent un critère global (p. ex. somme des synergies internes au cluster).

Les **clusters synergiques** formés au sein d’un **DSL** jouent un **rôle** fondamental à deux niveaux. Ils **favorisent** d’abord la **coopération locale** en permettant aux entités d’un même cluster d’échanger de manière intensive. Chaque entité apporte ses données ou compétences spécifiques, renforçant ainsi la synergie collective.

Ils **facilitent** ensuite l’**adaptation** du réseau. Ces clusters peuvent fusionner pour mieux gérer de nouveaux contextes lorsque leur compatibilité est élevée, ou se scinder lorsqu’un manque de synergie interne apparaît.

Grâce à ce double mécanisme combinant **coopération accrue** et **flexibilité structurelle**, le système reste **résilient** et capable d’évoluer face aux changements de tâches ou d’environnements.

1.2.6.5. Synergistic Connection Network (SCN)

Définition. Le **SCN** représente l’**infrastructure** du **DSL**, c’est un **réseau** dont les *nœuds* sont les entités $\{\mathcal{E}_i\}$ et dont les *arêtes* sont les pondérations $\{\omega_{i,j}\}$.

Contrairement à un réseau de neurones statique, le **SCN** est **dynamique**. Au fil du temps ou des itérations d’apprentissage, de nouvelles connexions apparaissent, tandis que d’autres se suppriment ou s’affaiblissent, favorisant ainsi l’émergence de **clusters** adaptatifs.

L’**objectif** central du **SCN** consiste à **exploiter** les **synergies** entre entités de manière à *auto-organiser* le flot d’information et, ce faisant, à optimiser la **performance** globale du système, qu’il s’agisse d’une tâche supervisée ou non supervisée. L’idée est de permettre aux liens synergiques les plus pertinents de se renforcer, afin que le réseau dirige spontanément les informations vers les chemins les plus efficaces. Ainsi, l’architecture se réadapte en fonction des besoins (ou des données) pour offrir un apprentissage et un traitement des informations plus rapide et plus robuste, sans nécessiter de contrôle externe permanent.

1.2.6.6. Auto-Organisation

Définition. **Auto-organisation** désigne la **capacité** d’un réseau à *se structurer* et *se reconfigurer* de façon autonome, sans intervention ou contrôle direct de l’extérieur (cf. section 1.2.4). Ce phénomène repose sur une **évaluation continue** de la synergie entre l’ensemble ou une partie des entités. À chaque itération, les pondérations $\omega_{i,j}$ sont **mises à jour** selon une règle d’adaptation, tandis que des **clusters** peuvent se **former** ou se **dissoudre** en fonction des tendances observées.

L'**objectif** de ce mécanisme est triple. D'abord, il s'agit d'acquérir une **robustesse** accrue face aux perturbations, car le réseau peut se réorganiser spontanément lorsque des défaillances ou des changements surviennent. Ensuite, cette approche permet de gérer naturellement la **multimodalité**. Plutôt que de cloisonner les entités visuelles, auditives ou autres, elle leur permet de s'associer ou de se séparer en fonction de leurs synergies. Enfin, l'auto-organisation ouvre la voie à un **apprentissage continu**, dans lequel de nouvelles représentations émergentes se forment au fil du temps, sans imposer la rigidité d'un schéma hiérarchique figé.

1.2.6.7. État (ou “State”) d'une Entité

Définition. Chaque entité \mathcal{E}_i dispose d'un **état interne** $\mathbf{s}_i(t)$, souvent représenté par un vecteur de dimension d , qui synthétise son “histoire” ou son “contexte” à l'instant t . Cet état évolue selon une **fonction d'actualisation** F , de la forme

$$\mathbf{s}_i(t+1) = F(\mathbf{s}_i(t), \{\omega_{i,j}(t)\}_j, \dots),$$

inspirée, par exemple, de modèles dynamiques ou de mécanismes de type RNN ou “Hebb étendu”. Le **rôle** de $\mathbf{s}_i(t)$ est déterminant pour la réactivité de l'entité. Une entité ayant déjà établi de fortes coopérations avec une autre \mathcal{E}_j est généralement plus enclue à se **synchroniser** de nouveau avec elle, la mémoire de leurs interactions passées augmentant la probabilité d'une synergie future.

1.2.6.8. Mécanismes de Fusion et de Dissociation

Deux (ou plusieurs) entités $\{\mathcal{E}_i, \mathcal{E}_j, \dots\}$ peuvent **fusionner** s'il s'avère qu'elles sont presque systématiquement dans un même cluster et qu'elles partagent une forte synergie dans la durée. Cette fusion se modélise par la création d'une **nouvelle entité** $\mathcal{E}_{\text{fusion}}$, qui combine leurs états, leurs mémoires et leurs représentations.

Lorsqu'une entité \mathcal{E}_k se trouve dans un cluster peu cohérent (synergie moyenne ou négative), elle peut se **retirer** du cluster ou rompre une fusion antérieure.

Ces mécanismes confèrent au DSL une **plasticité structurale** comparable à celle de certains systèmes biologiques (cerveau, colonies d'insectes, etc.), favorisant l'adaptation face à de nouveaux contextes ou de nouvelles tâches.

1.2.6.9. Énergie ou Fonction Globale \mathcal{J}

On peut parfois définir une **fonction** $\mathcal{J}(G)$ — parfois appelée “énergie libre” ou “coût global” — qui regroupe, d'une part, les **synergies** positives entre les entités et, d'autre part, un **terme** de pénalisation destiné à éviter une **surabondance** de connexions. Par exemple :

$$\mathcal{J}(G) = - \sum_{i,j} \omega_{i,j} S(\mathcal{E}_i, \mathcal{E}_j) + \alpha \|\boldsymbol{\omega}\|^2,$$

où ω désigne le vecteur de toutes les pondérations synergiques, et α un coefficient de régulation. **Minimiser** $J(G)$ revient alors à **maximiser** la somme de synergies utiles tout en **limitant** la prolifération de liens non pertinents. Il s'agit ainsi d'une démarche **globale** pour piloter l'auto-organisation du réseau, puisqu'elle encourage les connexions réellement productives tout en imposant un frein à celles qui n'apporteraient aucun gain substantiel.

1.2.6.10. Apprentissage Continu (ou “Lifelong Learning”)

Dans le **DSL**, l'apprentissage ne se limite pas à une **phase offline** unique ; le réseau peut, au contraire, **évoluer** continuellement face à un flux de données **online**, en réajustant de façon permanente les pondérations $\omega_{i,j}(t)$ ainsi que la configuration des clusters.

Cet **apprentissage continu** présente plusieurs **avantages**. D'une part, il offre une **tolérance** accrue aux perturbations comme le bruit, les changements dans la distribution des données ou l'apparition de nouvelles classes et contextes. Cette capacité permet au système de s'adapter plus facilement à des environnements non stationnaires. D'autre part, il contribue à la **réduction** du phénomène de “forgotten knowledge”, puisque les clusters formés pour des tâches antérieures peuvent être préservés et ainsi servir de base à des transferts de connaissances ultérieurs.

1.2.6.11. Terminologies récurrentes (Synthèse)

Pour faciliter la lecture et l'implémentation, voici un **récapitulatif** des principales terminologies :

\mathcal{E}_i : *Entité d'information* numéro i .

$S(\mathcal{E}_i, \mathcal{E}_j)$: *Synergie* entre entités i et j .

$\omega_{i,j}(t)$: *Pondération synergique* (ou lien) entre entités i et j à l'instant t .

$W(t) = [\omega_{i,j}(t)]$: *Matrice* (ou graphe) des pondérations synergiques.

Cluster : Sous-groupe d'entités fortement liées (hautes $\omega_{i,j}$).

SCN : *Synergistic Connection Network*, la structure dynamique qui évolue selon les lois d'adaptation.

Auto-organisation : Processus par lequel la structure $W(t)$ se réarrange spontanément.

Fonction J : Mesure globale de la qualité ou de l'état du réseau (peut inclure la somme des synergies, des pénalités, etc.).

1.2.6.12. Conclusion

Ces termes comme **entité**, **synergie**, **pondération synergique**, **cluster**, **SCN** et **auto-organisation** constituent le *lexique de base* du **DSL**. Chaque concept est interdépendant. Les **entités** interagissent

à travers des **pondérations synergiques** qui façonnent le SCN. Celui-ci se **réorganise en clusters** selon un mécanisme d'**auto-organisation** guidé par la **synergie** et, éventuellement, par une **fonction globale \mathcal{J}** .

Dans la section suivante (1.2.7), nous illustrerons ces principes par des **exemples concrets**, qu'ils proviennent de la nature (inspirations biologiques) ou d'applications pratiques (cas d'études multimodales, émergence de schémas cognitifs, etc.). Ce sera l'occasion de vérifier comment l'utilisation rigoureuse de cette terminologie peut clarifier la **logique** et la **mise en œuvre** du Deep Synergy Learning.

1.2.7. Exemples Illustratifs de la Synergie dans la Nature

Les principes de **synergie informationnelle** et d'**auto-organisation** que promeut le Deep Synergy Learning (DSL) trouvent de nombreux échos dans les systèmes naturels. Qu'il s'agisse de colonies d'insectes, de réseaux neuronaux biologiques, d'écosystèmes ou de synchronisations collectives, on observe des processus où l'**ensemble** dépasse la **somme de ses parties**, grâce à des mécanismes coopératifs distribués. Les sous-sections suivantes illustrent comment ces phénomènes naturels inspirent l'approche synergique du DSL.

Colonies d'Insectes et Intelligence Collective

Les colonies de fourmis, d'abeilles ou de termites sont des exemples emblématiques d'**intelligence collective**. Chaque individu, malgré des capacités limitées, contribue à la réalisation de tâches complexes comme la construction de nids sophistiqués ou l'optimisation de la recherche de nourriture. Ces interactions reposent sur des **signaux locaux** (phéromones, contacts antennaires, etc.), sans qu'aucune entité centrale ne dirige l'ensemble. L'émergence d'une organisation globale, comme le traçage de pistes ou la réparation du nid, résulte donc d'une **coopération** distribuée entre entités locales.

Dans le **Deep Synergy Learning (DSL)**, les **entités d'information** fonctionnent de manière analogue aux insectes sociaux. Elles établissent ou rompent des liens en fonction de la **pertinence** ou de la **synergie** qu'elles y perçoivent. Ces **connexions synergiques** évoluent en permanence, à l'image des fourmis qui renforcent ou abandonnent certains chemins selon leur utilité.

La formation de **clusters** d'entités synergiques dans le **DSL** rappelle les **micro-sociétés** au sein d'une colonie d'insectes, où chaque groupe se spécialise dans une tâche particulière. Cette **auto-organisation** spontanée illustre la puissance d'un système distribué. Sans planification rigide, l'ensemble du réseau s'adapte et se coordonne pour atteindre un objectif global.

Synergies dans le Cerveau et les Réseaux Neuronaux Biologiques

Dans le **cerveau** humain ou animal, la **plasticité synaptique** illustre la puissance d'un réseau extrêmement **connecté**, dans lequel les synapses s'ajustent en fonction des interactions locales. Lorsque deux neurones s'associent régulièrement pour traiter un même stimulus, leur **synapse** se

renforce (potentialisation à long terme) ; ce phénomène rappelle la **mise à jour** des liaisons synergiques dans le DSL, où les liens forts se consolident à mesure que les entités coopèrent efficacement.

Les neurosciences démontrent également la formation d'**assemblées neuronales** associées à un concept ou à un stimulus précis. Ces assemblées se créent ou se dissolvent selon le **contexte** ou la **tâche** du moment. De la même manière, le DSL autorise la **création** et la **dissolution** de **clusters** d'entités d'information, la synergie entre ces entités évoluant dans le temps pour s'adapter aux besoins et aux données.

Enfin, ce parallèle s'étend à la **mémoire** et à l'**apprentissage**. Dans le cerveau, les synapses permettent à un neurone de conserver les connexions consolidées lors d'apprentissages antérieurs, lui offrant ainsi la capacité de “se souvenir” des interactions passées. Dans le DSL, chaque entité conserve un **état interne** et un **historique** (section 1.2.1.2), ce qui lui confère une **mémoire contextuelle** et améliore la **cohérence** de l'apprentissage sur le long terme.

Écosystèmes et Coopérations Symbiotiques



Les **écosystèmes** fournissent de nombreux exemples de **coopération** interspécifique. Les insectes pollinisent les plantes, les lichens résultent de la symbiose entre algues et champignons, et la mycorhize associe champignons et racines de plantes. Dans chacun de ces cas, les organismes obtiennent un **bénéfice mutuel**, qu'il s'agisse d'un meilleur accès aux ressources, d'une protection renforcée ou d'une plus grande capacité d'adaptation. Cette dynamique repose sur la **notion de “gain commun”**, qui illustre la **synergie**. La coexistence de deux entités ou espèces génère une **valeur ajoutée** absente lorsqu'elles agissent isolément.

Dans le **Deep Synergy Learning (DSL)**, ce principe s'incarne à travers la **mesure de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ (voir section 1.2.2) et l'**ajustement** adapté des pondérations $\omega_{i,j}(t)$. De la même façon

que deux espèces coopérantes se renforcent l'une l'autre, deux entités informationnelles voient leurs liens se consolider lorsqu'elles interagissent efficacement. Les **écosystèmes** diversifiés, riches en symbioses, font preuve d'une **résilience** considérable face aux menaces comme la sécheresse ou la prédation, grâce aux ressources complémentaires que chaque espèce apporte. Un **réseau synergique (DSL)** regroupant des entités variées, qu'elles soient visuelles, textuelles ou auditives, développe une plus grande **robustesse** et une meilleure **flexibilité**. En modulant continuellement ses interactions, il s'adapte plus efficacement aux imprévus et aux évolutions de son environnement.

Synchronisation Collective : Bancs de Poissons et Nuées d'Étourneaux



Les **bancs de poissons** et les **vols d'oiseaux**, comme les nuées d'étourneaux, illustrent un phénomène remarquable de **synchronisation**. De larges groupes se déplacent de manière presque chorégraphiée, sans chef unique, chaque individu ajustant sa trajectoire en fonction de celle de ses voisins. Ce processus engendre un **effet émergent** de cohésion et d'harmonisation.

Ce type d'organisation repose généralement sur quelques **règles simples**, comme l'alignement des vitesses, le maintien d'une distance de sécurité et l'attraction mutuelle. Une fois combinées, ces règles produisent des comportements collectifs complexes. Dans le cadre du **Deep Synergy Learning (DSL)**, la **synergie** et la **mise à jour** des liens jouent un rôle similaire. Lorsque la coopération entre deux entités s'avère bénéfique, elles se **synchronisent**, ce qui contribue à optimiser le **réseau global**.

Les bancs ou les nuées font également preuve d'une grande **plasticité**, se reconfigurant rapidement face à un prédateur ou un obstacle. De façon parallèle, un **réseau synergique** peut, à tout instant, **adapter** sa structure dès lors que le contexte ou les données évoluent, sans nécessiter de "réentraînement" global et figé.

Conclusion

Les exemples précédents – colonies d'insectes, cerveau, écosystèmes ou synchronisations collectives – démontrent que la synergie émerge lorsque des entités locales et relativement simples **coopèrent** selon des **règles d'interaction** et d'**adaptation**. Sans supervision centrale et sans plan préconçu, il se forme souvent des **structures** ou des **comportements** remarquablement organisés et robustes, capables de s'ajuster aux contraintes du milieu.

Ces observations, empruntées à la nature, guideront la formulation plus formelle des **algorithmes** et la mise en place de **protocoles d'évaluation** pour le DSL. Dans les chapitres suivants, nous verrons comment concrétiser ces analogies sous la forme de modèles mathématiques, de règles de mise à jour et d'applications pratiques, visant à faire du Deep Synergy Learning un **paradigme opérationnel** pour une **IA forte** (ou du moins plus autonome et plus générale).

1.3. Importance de l'Auto-Organisation

Dans le cadre du **Deep Synergy Learning (DSL)**, l'un des aspects les plus novateurs réside dans sa capacité d'**auto-organisation**. Alors que les approches d'IA traditionnelles s'appuient souvent sur des architectures hiérarchiques fixes, le DSL encourage un processus dynamique où les **entités d'information** établissent et révisent en continu leurs relations, en fonction de la **synergie** qu'elles détectent entre elles (voir sections précédentes). Cette section (1.3) aborde l'importance de l'auto-organisation sous plusieurs angles, depuis les sources d'inspiration biologiques (1.3.1) jusqu'à l'impact sur la résilience des modèles (1.3.6) et les perspectives globales (1.3.7).

1.3.1. Inspirations Biologiques et Cognitives

L'auto-organisation, au sens où l'entend le DSL, n'est pas un concept nouveau. En réalité, de nombreux **systèmes naturels** (organismes vivants, écosystèmes, cerveau, etc.) démontrent d'extraordinaires facultés d'organisation **spontanée** et **adaptive**. Le **Deep Synergy Learning** cherche à exploiter ces principes, en transposant aux systèmes informatiques des mécanismes qui ont fait leurs preuves dans le vivant.

1.3.1.1. Le cerveau comme paradigme de plasticité



Le cerveau humain compte des centaines de milliards de neurones, reliés par des synapses dont les forces (poids synaptiques) évoluent selon l'activité, un phénomène désigné par "plasticité synaptique". On peut symboliser ce principe via une équation de type Hebbien généralisée, par exemple :

$$w_{ij}(t+1) = w_{ij}(t) + \eta [A_{ij}(t) - \lambda w_{ij}(t)],$$

où w_{ij} est la "force" de la synapse reliant le neurone i au neurone j , η un taux d'apprentissage, $A_{ij}(t)$ l'activité conjointe des neurones i et j (ou une mesure de leur association), et λ un terme de régulation. L'important est de voir comment la synapse se renforce si l'activité commune est

significative, rappelant la **mise à jour des pondérations synergiques** dans le DSL (cf. section 1.2.6).

Dans le cerveau, des regroupements (ou *assemblées*) de neurones se forment lorsque leurs synapses se renforcent mutuellement. Le concept de cluster auto-organisé dans le DSL (où les entités se regroupent pour exploiter leur synergie) ressemble à cette idée :

$$\mathcal{C}^* = \underset{\mathcal{C} \subset \{\mathcal{E}_1, \dots, \mathcal{E}_n\}}{\operatorname{argmax}} \sum_{(i,j) \in \mathcal{C} \times \mathcal{C}} \omega_{i,j},$$

c'est-à-dire la recherche d'un sous-ensemble \mathcal{C} maximisant la somme des pondérations entre ses membres, analogiquement à des neurones qui s'assemblent pour traiter un même stimulus.

La cognition résulte du fait que certains assemblages persistent dans le temps et renforcent leur “*autoroute synaptique*” (par exemple, mémorisation d'un stimulus répété). Dans le DSL, l'émergence de **micro-réseaux** stables autour d'entités synergiques reflète ce même phénomène d'**apprentissage distribué**, favorisant la récurrence de schémas pertinents.

1.3.1.2. Colonies d'insectes et intelligence collective

Les colonies de fourmis s'illustrent souvent par leur capacité à trouver des chemins optimaux pour la recherche de nourriture. Ce comportement repose sur des **interactions locales** (dépôt de phéromones, etc.), et non sur un plan global. L'analogie avec le DSL se fait sentir lorsque l'on considère que chaque fourmi incarne une “entité” prenant des décisions locales qui, agrégées, produisent une optimisation globale.



On peut modéliser la concentration de phéromones le long d'un chemin par une fonction $\phi(t)$ obéissant à un mécanisme de renforcement/dissipation, par exemple :

$$\frac{d\phi}{dt} = \alpha I(\text{fourmis}) - \beta \phi,$$

où α modélise l'ajout de phéromones (si des fourmis passent) et β la vitesse d'évaporation. Le DSL retrouve un concept similaire quand il gère l'**intensité** d'une liaison synergique via des règles de renforcement ou de décroissance (cf. équations de mise à jour des pondérations, section 1.2.4).

À l'instar de chemins de phéromones qui “naissent” (s'ils sont utiles) et “disparaissent” (s'ils ne sont plus empruntés), les connexions synergiques $\omega_{ij}(t)$ du DSL se renforcent ou s'étiolent, entraînant la **création** ou la **rupture** de clusters d'entités d'information. Cette dynamique **auto-organisée** ne requiert pas de commande centrale.

1.3.1.3. Systèmes dynamiques et attracteurs auto-organisés

Exemples de modèles dynamiques

De nombreux systèmes auto-organisés (réacteurs chimiques de type Belousov-Zhabotinsky, réactions oscillantes, etc.) peuvent s'exprimer par des **équations différentielles** à plusieurs variables, où les interactions locales créent des **boucles de rétroaction** positives ou négatives. Les motifs qui émergent (ondes chimiques, spirales...) sont des exemples d'**attracteurs** dynamiques.

Analogies avec le graphe synergique

Dans le DSL, on peut considérer que l'**état global** du réseau est donné par la matrice $\Omega(t) = [\omega_{i,j}(t)]$. L'évolution de $\Omega(t)$ suit des règles de type :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Cette équation peut se voir comme un **système dynamique** discret, susceptible de converger vers un attracteur (une configuration stable) ou d'entrer dans des régimes oscillatoires, reflétant l'**auto-organisation**.

Stabilité et oscillations

Dans un cadre mathématique plus poussé, on peut étudier la **stabilité** de l'état Ω^* en résolvant :

$$\omega_{i,j}(t+1) - \omega_{i,j}(t) = 0,$$

pour tous (i, j) . On obtient alors une condition d'équilibre, ce qui, dans un contexte du DSL, correspond à un réseau où chaque liaison a atteint une valeur stable en regard des synergies (sous réserve que l'on n'introduise pas de nouvelles entités ou de nouvelles données). Une **analyse de la Jacobienne** locale pourrait prédire si cet état Ω^* est un **point d'attraction**, un **cycle limite** ou un **chaos**.

1.3.1.4. Vers un transfert de ces principes au DSL

Les systèmes biologiques et cognitifs décrits ci-dessus mettent en avant quatre qualités fondamentales :

- **Plasticité** (adaptation continue)

- **Robustesse** (résistance au bruit ou à la perturbation)
- **Emergence** (apparition de structures non planifiées)
- **Auto-régulation** (équilibre entre renforcement et inhibition)

Le DSL s'inspire de ces qualités pour proposer un **apprentissage distribué**, dans lequel les entités d'information se comportent comme des agents qui renforcent leurs liens lorsqu'ils détectent un **gain** (cf. notion de synergie) et les relâchent sinon. Matériellement, on transpose ces mécanismes via :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t)\eta \left(S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t) \right),$$

et des **clusters** se forment spontanément si le gain est significatif.

Contrairement aux réseaux neuronaux profonds classiques, où seule la valeur des poids évolue dans une architecture figée, l'**auto-organisation** du DSL laisse la structure du réseau se **recomposer** en fonction des synergies détectées. Les entités elles-mêmes peuvent évoluer, en modifiant leur **représentation** ou leur **état** interne, et donc modifier indirectement la dynamique globale du réseau.

1.3.1.5. Conclusion partielle : fondements biologiques de l'auto-organisation

Cette section a illustré, par des références à la **plasticité synaptique**, à l'**intelligence collective** chez les insectes et à divers **systèmes dynamiques**, l'importance des mécanismes d'auto-organisation en biologie et en cognition. Le **Deep Synergy Learning** cherche à exploiter ces mêmes principes pour créer des modèles d'intelligence artificielle moins dépendants d'une structure hiérarchique imposée, plus adaptatifs, et potentiellement plus proches d'une **IA générale**. Les sections suivantes (1.3.2 à 1.3.7) mettront en évidence des concepts plus généraux d'**émergence**, de **feedback**, et détailleront comment l'auto-organisation du DSL se compare aux méthodes d'apprentissage classiques, avant de conclure sur la pertinence de cette approche pour des architectures globales et multimodales.

1.3.2. Concepts Clés : Émergence, Auto-Régulation, Feedback

La dynamique **auto-organisée** que promeut le Deep Synergy Learning (DSL) repose sur plusieurs notions fondamentales, à la fois conceptuelles et mathématiques, qui permettent de comprendre comment un système d'entités peut s'ajuster et se coordonner sans planification centrale. Trois d'entre elles sont particulièrement essentielles : l'**émergence**, l'**auto-régulation** et le **feedback** (rétroaction). Les sous-sections ci-après détaillent la signification de chacun de ces concepts et leur importance dans le cadre du DSL.

1.3.2.1. Émergence

Définition. Le terme **émergence** décrit la situation où un **phénomène global** ou une **structure** à l'échelle macroscopique naît de l'interaction de **composants** à l'échelle microscopique, sans qu'il n'y ait de plan explicite ni de chef d'orchestre. Par exemple, la forme d'une nuée d'oiseaux ou l'organisation complexe d'une ruche peuvent être vus comme des propriétés émergentes d'un grand nombre d'interactions locales.

Dans le Deep Synergy Learning, l'**émergence** se manifeste par la formation de **clusters** (ou micro-réseaux) d'entités d'information, que l'on peut qualifier de "sous-systèmes cohérents" au sein du grand réseau synergique. Ces **clusters** ne sont pas prédéfinis. Ils **apparaissent** et parfois **disparaissent** en fonction de la synergie détectée entre les entités.

Un cluster $\mathcal{C} \subset \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ peut s'illustrer par une somme de pondérations internes $\sum_{(i,j) \in \mathcal{C} \times \mathcal{C}} \omega_{i,j}$ qui se trouve **maximalisée** à un certain moment t .

De plus, ces clusters émergents peuvent générer de nouvelles **représentations** ou entités "conjointes", lesquelles elles-mêmes nourrissent l'**évolution** du réseau.

Sur le plan formel, on peut lier l'émergence au fait que la **dynamique** des pondérations synergiques $\omega_{i,j}(t)$ conduit le système vers des **attracteurs** (états stables ou semi-stables). L'apparition d'un cluster correspond alors à un **bassin d'attraction** particulier, où les liens internes se renforcent suffisamment pour maintenir l'entité groupée, un phénomène typique des **systèmes dynamiques complexes**.

1.3.2.2. Auto-Régulation

Dans un système auto-organisé, chaque composant ou lien s'**ajuste** en fonction de **règles locales**. Lorsque la coopération est bénéfique, le lien se renforce, tandis que si l'est jugé néfaste ou inutile, il s'affaiblit. Ce **mécanisme d'ajustement** relève de l'**auto-régulation**, car il ne dépend pas d'une autorité externe qui dirait "ces entités doivent se connecter, celles-ci doivent se séparer".

Comme décrit en section 1.2.4, une règle fréquemment utilisée dans le DSL pour modéliser l'évolution des pondérations synergiques $\omega_{i,j}(t)$ est :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

- η représente un taux d'apprentissage (contrôle la rapidité de l'ajustement).
- $S(\mathcal{E}_i, \mathcal{E}_j)$ est la **synergie** calculée entre les entités i et j .
- τ agit comme un terme de décroissance ou de "frottement", évitant que les pondérations ne croissent indéfiniment.

Le paramètre τ et le choix de la fonction $S(\cdot)$ permettent d'**éviter les dérives** (ex. explosion des connexions) et de stabiliser le système dans un régime d'**énergie** ou de **coût** relativement bas (voir fonction \mathcal{J} discutée en 1.2.6). Ce mécanisme d'**auto-régulation** dispense de toute redéfinition

manuelle de la topologie du réseau. Il s'adapte de lui-même aux contraintes et aux données entrantes.

1.3.2.3. Feedback (Rétroaction)

Le **feedback** désigne le processus par lequel la **sortie** d'un système ou l'**état** d'une partie du système **retourne** sur lui-même, influençant à son tour la configuration ou le comportement du système. Dans le **DSL**, le feedback se manifeste principalement à travers l'**apprentissage** des synergies. La manière dont deux entités ont coopéré à l'instant t influence la pondération $\omega_{i,j}(t+1)$, qui déterminera ensuite leur coopération future.

Le **feedback positif** est un mécanisme par lequel une forte synergie entre deux entités, notée $S(\mathcal{E}_i, \mathcal{E}_j)$, conduit à un renforcement mutuel de leur connexion. En d'autres termes, si la synergie initiale entre deux entités est élevée, leur interaction devient progressivement plus forte, renforçant encore davantage leur synergie. Ce processus peut continuer jusqu'à atteindre un certain plafond, au-delà duquel la connexion ne peut plus croître, stabilisant ainsi leur relation synergique.

À l'inverse, le **feedback négatif** intervient lorsque la synergie entre deux entités est faible ou nulle, voire négative. Dans ce cas, la pondération de leur lien décroît progressivement, réduisant la probabilité qu'elles interagissent à l'avenir. Cependant, ce mécanisme n'est pas définitif. Un changement de contexte ou d'état peut, à terme, améliorer la compatibilité entre deux entités et rétablir leur capacité à collaborer ou à former une connexion.

Outre les liens binaires (entre paires d'entités), il peut exister des **boucles de rétroaction plus complexes** à l'échelle de plusieurs clusters. Par exemple, un cluster \mathcal{C}_1 peut influencer la formation ou la dissolution d'un autre cluster \mathcal{C}_2 en modifiant le contexte, les ressources partagées, etc. Cela équivaut à considérer des **fonctions de synergie n-aire**, où les "coûts" ou "gains" de chaque liaison dépendent aussi des états des autres liens et entités.

$$S(\mathcal{E}_i, \mathcal{E}_j \mid \{\mathcal{E}_k\}_{k \neq i,j}), \quad \omega_{i,j}(t+1) = f(\omega_{i,j}(t), \{S(\mathcal{E}_i, \mathcal{E}_j \mid \dots)\}, \dots).$$

1.3.2.4. Synthèse : de la dynamique locale à la cohérence globale

Les trois concepts d'**émergence**, d'**auto-régulation** et de **feedback** interagissent étroitement dans le Deep Synergy Learning :

- L'**émergence** se produit parce que les liens (pondérations synergiques) se développent localement sans qu'aucun schéma global ne soit imposé.
- L'**auto-régulation** assure que chaque lien est entretenu ou affaibli en fonction de la **pertinence** mesurée (synergie), maintenant ainsi une forme de **plasticité** dans l'ensemble du réseau.
- Le **feedback** perpétuel, tant au niveau des paires d'entités que des ensembles plus larges, crée un **cycle** où l'état du réseau à l'instant t influence les coopérations à l'instant $t+1$, et ainsi de suite.

Ce mode de fonctionnement "bouclé" et distribué est très différent de la logique hiérarchique classique (voir 1.2.4). Il rappelle davantage les **systèmes biologiques** et autres **systèmes**

complexes, où la stabilité et les structures apparaissent comme le fruit d’interactions locales répétées, produisant parfois des schémas surprenants d’ordre (ou de désordre) global.

1.3.2.5. Conclusion intermédiaire : vers une IA plus adaptative

Comprendre les notions d'**émergence**, d'**auto-régulation** et de **feedback** est crucial pour saisir pourquoi le **DSL** peut offrir des capacités d’adaptation et de créativité supérieures à celles des architectures figées. Loin d’être un simple attribut de “design” théorique, ces concepts constituent la **dynamique interne** qui permet au réseau synergique de se **reconfigurer** en permanence, de **fusionner** ou de **séparer** des entités en clusters, et d'**apprendre** sans devoir figer sa topologie.

Dans la section suivante (1.3.3), nous mettrons en perspective ces mécanismes d’auto-organisation avec les **méthodes d’apprentissage classiques**, afin d’explorer les gains potentiels (et les défis) qu’apporte l’introduction de ces idées dans le champ de l’intelligence artificielle.

1.3.3. *Comparaison avec les Méthodes d’Apprentissage Classiques*

Les principes d'**auto-organisation** et de **synergie** du Deep Synergy Learning (DSL) contrastent fortement avec les schémas employés par les méthodes d’apprentissage les plus courantes, qu’il s’agisse d’algorithmes supervisés, non supervisés ou même de certaines approches en renforcement. Dans la présente section, nous mettons en lumière les principales différences entre ces paradigmes “classiques” (tels que le **Deep Learning** hiérarchique, la **clustering** statique ou la **rétropropagation de gradient** end-to-end) et le **DSL**, qui s’appuie avant tout sur une **construction dynamique** du réseau via la synergie informationnelle.

1.3.3.1. Apprentissage supervisé traditionnel

Dans un **cadre supervisé** classique, on dispose d’un ensemble de données $\{(\mathbf{x}_i, y_i)\}_{i=1,\dots,N}$, où \mathbf{x}_i constitue l’entrée (vecteur de caractéristiques, image, etc.) et y_i le label associé (catégorie, valeur numérique, etc.). Un **modèle** (par exemple, un réseau de neurones) est alors **entraîné** à minimiser une **fonction de coût** :

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N L(f_\theta(\mathbf{x}_i), y_i),$$

où f_θ désigne le modèle paramétré par θ (poids et biais), et $L(\cdot)$ une fonction de perte (entropie croisée, MSE, etc.).

La plupart du temps, l’**architecture** du réseau est **figée** : on choisit un nombre de couches, une topologie (CNN, MLP, etc.) et on la conserve durant tout l’entraînement. La **rétropropagation** permet d’ajuster les poids internes θ , sans pour autant réorganiser la **structure** (aucune suppression ni création de neurones, aucune reconfiguration des connexions).

Comparaison avec le DSL.

Dans un réseau **DSL**, on ne se limite pas à la mise à jour des paramètres, on autorise la **topologie** à évoluer (création, renforcement, fusion, ou rupture de liens) en se fondant sur la **mesure de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. Dans un modèle supervisé classique, la structure reste **inchangée** tout au long de l'apprentissage.

Dans l'approche **supervisée** traditionnelle, l'effort se concentre sur la **réduction de l'erreur** par rapport aux labels y . À l'inverse, le **DSL** se concentre sur la **relation** entre les différentes données elles-mêmes. La **formation** de clusters et l'**auto-organisation** peuvent émerger même en l'absence de labels, dès lors que le réseau détecte des synergies fructueuses.

Un **modèle supervisé** doit souvent être **réentraîné** (ou au moins fine-tuné) dès lors qu'apparaissent de nouveaux types de données ou qu'une **distribution** se modifie. Dans le DSL, la structure s'**auto-réorganise** en intégrant ou en écartant certaines entités, en fonction de la synergie relevée, ce qui apporte davantage de **souplesse** et de **plasticité** face aux environnements changeants.

1.3.3.2. Apprentissage non supervisé et clustering statique

Dans l'apprentissage **non supervisé**, on cherche généralement à regrouper des points \mathbf{x}_i en clusters. Des méthodes comme **k-means** démarrent avec un nombre de clusters (k) et une mesure de distance (par exemple euclidienne ou cosinus), puis procèdent à une série d'itérations visant à minimiser

$$\sum_{i=1}^N \min_{1 \leq c \leq k} \| \mathbf{x}_i - \boldsymbol{\mu}_c \|^2,$$

où $\boldsymbol{\mu}_c$ sont les centres de clusters. Dans **k-means**, le paramètre k doit être défini **a priori**, et même si d'autres algorithmes (tels que DBSCAN ou le clustering agglomératif) peuvent estimer le nombre de groupes, ils reposent néanmoins sur une **logique statique**, c'est-à-dire qu'on exécute l'algorithme une fois et on obtient une unique partition.

Dans ces approches classiques, les points ou vecteurs ne possèdent pas de **dynamique** interne. Leur représentation reste figée, et les liens potentiels entre eux n'évoluent pas au fil du temps en fonction d'une éventuelle collaboration. Le **DSL** se distingue précisément sur ce point. Les **clusters** peuvent **apparaître ou se dissoudre** de manière **progressive** selon les pondérations synergiques $\omega_{i,j}(t)$, sans qu'un paramètre k fixe ne contraigne le regroupement. La structure du réseau se **façonner** ainsi en continu, guidée par la dynamique d'apprentissage.

En outre, chaque entité \mathcal{E}_i du réseau DSL peut être **active**, c'est-à-dire posséder un **état interne** et évoluer au cours du temps. Dans un clustering **statique**, les points demeurent figés dans leurs coordonnées, sans aucun mécanisme de feedback local ou de synergie émergente. Enfin, lorsqu'apparaissent de nouvelles données ou qu'un changement de distribution survient, le DSL est capable de **réévaluer** les synergies et de **reconfigurer** automatiquement les clusters, tandis qu'un clustering non supervisé traditionnel exigerait la relance complète de l'algorithme, ignorant dans la foulée toute information antérieure.

1.3.3.3. Apprentissage par renforcement (RL)

Dans le **Reinforcement Learning**, un **agent** interagit avec un environnement, perçoit des **récompenses** (positives ou négatives) selon ses actions, et cherche à **maximiser** un **cumul de récompenses**. Les algorithmes de RL (Q-learning, SARSA, méthodes basées sur une politique) visent souvent à estimer la fonction de valeur $Q(\mathbf{s}, a)$ ou la politique optimale $\pi(a|\mathbf{s})$. De nombreux scénarios de RL (par ex. Deep Q-Network) s'appuient sur un **réseau profond** paramtré, dont les poids sont ajustés via un **objectif** portant sur la récompense cumulée.

Dans la plupart des implémentations **RL** traditionnelles, la **structure** du réseau reste figée. Au cours de l'apprentissage, ni le nombre de couches ni la forme du modèle ne sont modifiés.

Synergie vs. récompense.

Alors que le RL priviliege l'action menant à la **maximisation** de la récompense, le DSL promeut la **coopération** (ou la connexion) qui maximise la **synergie**. Il est envisageable de développer un **DSL-RL** où la **synergie** intervient comme une composante de la **fonction de récompense**. Plus la **collaboration** entre entités s'avère fructueuse, plus les liens correspondants se renforcent.

Reconfiguration continue.

Dans le RL classique, on n'ajoute ni ne supprime spontanément des neurones ou des liaisons durant l'apprentissage. À l'inverse, le DSL autorise une **recomposition** de l'architecture si cela contribue à améliorer la performance (ou la cohérence). Les pondérations peuvent émerger, se renforcer, ou disparaître au fil du temps.

Apprentissage distribué.

On peut voir les entités du DSL comme autant d'"agents partiels" qui s'accordent via la **synergie**, au lieu de recourir à un **agent unique** muni d'un policy network fixe. Cette perspective se distingue du **RL centralisé**. L'organisation émerge de l'**interaction locale** entre entités au lieu de suivre une stratégie unique imposée à l'ensemble du réseau.

1.3.3.4. Synthèse des contrastes fondamentaux

Les approches classiques, qu'elles soient **supervisées**, **non supervisées** ou basées sur le **renforcement**, reposent généralement sur une **architecture** ou un **ensemble d'hypothèses** prédefinies. Cela peut correspondre à un réseau feed-forward, à un nombre fixé de clusters à détecter ou à une paramétrisation spécifique. À l'inverse, le **DSL** ne fixe pas la structure à l'avance ; il permet à celle-ci de **se former** et de **se modifier** de manière autonome, simplement en s'appuyant sur les mesures de **synergie** et sur les **règles** d'auto-organisation.

Dans les méthodes classiques, on priviliege soit le **rapprochement** entre données et labels (apprentissage supervisé), soit la **répartition** de points dans des groupes (apprentissage non supervisé). Le DSL, au contraire, met l'accent sur la **relation** entre les entités, qu'il s'agisse de leur **coexistence** ou de leur **synergie**, et c'est ce mécanisme relationnel qui détermine l'organisation d'ensemble.

Ce renversement de perspective — considérer la donnée non pas comme un **input passif**, mais comme un **acteur** recherchant activement des **connexions** — constitue la marque distinctive du DSL. Les algorithmes traditionnels sont souvent conçus pour un entraînement **en batch**. Les données sont d'abord collectées, le modèle est ensuite entraîné, puis sa configuration est figée. Des versions en ligne existent, mais elles n'ont pas pour vocation de retoucher la **structure** du modèle ; elles n'agissent que sur les poids.

Le **DSL**, en revanche, s'adapte naturellement à un **flux continu** de données. À mesure que de nouvelles informations apparaissent, le réseau peut se **reconstruire** pour s'ajuster aux conditions changeantes ou intégrer de **nouvelles entités**. Il favorise ainsi une **adaptation permanente**, plutôt que de se conformer à un schéma fixe déterminé à l'avance.

1.3.3.5. Conclusion : l'auto-organisation comme rupture de paradigme

La comparaison avec les méthodes classiques met en évidence le caractère **révolutionnaire** de l'auto-organisation prônée par le **DSL**. La capacité à **faire évoluer la carte du réseau** plutôt que de se limiter à des architectures fixes, l'intégration d'une **synergie** mesurée entre entités, la facilité d'**apprentissage continu** et la résistance aux changements de distribution en font une approche fondamentalement différente.

Ces différences ne signifient pas que le DSL vienne nécessairement **remplacer** l'ensemble des paradigmes existants. On peut au contraire envisager des **approches hybrides** qui combinent la puissance des algorithmes classiques (supervisés ou RL) avec la **plasticité** et la **coévolution** propres au DSL. Les sections qui suivront (1.3.4 à 1.3.7) approfondiront l'idée d'une **auto-organisation multimodale**, ainsi que l'**impact** concret de cette approche sur la robustesse et la résilience des modèles, pour finalement ouvrir la voie à une IA plus générale et plus souple.

1.3.4. Le Rôle des Flux d'Information Multimodaux

Parmi les avantages majeurs de l'**auto-organisation** telle que proposée dans le **Deep Synergy Learning (DSL)** figure la capacité à gérer efficacement des **flux d'information** variés (image, texte, audio, données sensorielles, etc.). En effet, l'idée de **synergie informationnelle** (développée aux sections 1.2.2 et 1.2.3) s'applique de manière particulièrement féconde dès lors que plusieurs modalités sont en jeu, car chaque flux peut **renforcer** ou **compléter** les autres. Cette sous-section (1.3.4) illustre la manière dont l'auto-organisation des entités d'information s'adapte à la **multimodalité**, et pourquoi cela peut conférer au DSL une **puissance d'intégration** hors de portée des architectures classiques, souvent cloisonnées.

1.3.4.1. Notion de multi-modalité dans le DSL

Dans un système multimodal, on distingue plusieurs sources de données :

- **Modalité visuelle** : images, vidéos (entité $\mathcal{E}_{\text{visuelle}}$)

- **Modalité auditive** : signaux sonores, musique, parole (entité $\mathcal{E}_{\text{auditive}}$)
- **Modalité textuelle** : séquences de mots, documents, balises sémantiques (entité $\mathcal{E}_{\text{textuelle}}$)
- **Modalités sensorielles complémentaires** : capteurs physiques (température, distance, pression), signaux biométriques, etc.

Dans le **Deep Synergy Learning**, chaque flux peut être représenté par une ou plusieurs **entités d'information** (voir section 1.2.1). Ainsi, un flux visuel pourra donner naissance à plusieurs entités – par exemple, une entité pour la “carte de caractéristiques” (feature map) d'une image, une autre pour son histogramme de couleurs, etc. Ces entités ne sont pas limitées à une seule couche. Elles peuvent interagir avec toutes les autres entités, qu'elles soient visuelles, textuelles ou auditives, en fonction de leur **synergie**.

1.3.4.2. Avantages de la synergie entre modalités

La **synergie** est particulièrement critique en multi-modalité :

$$S(\mathcal{E}_{\text{visuelle}}, \mathcal{E}_{\text{auditive}}) > 0 \\ \Rightarrow \text{l'information visuelle et l'information auditive se complètent efficacement.}$$

Par exemple, dans une scène vidéo où l'on entend quelqu'un parler, la composante auditive (voix) et la composante visuelle (mouvements de lèvres, expressions faciales) coopèrent pour renforcer la compréhension globale.

Lorsqu'une modalité est dégradée (par exemple, l'audio est parasité par du bruit de fond), une autre modalité (l'image) peut prendre le relais, augmentant la **tolérance aux perturbations**. Dans un **réseau synergique**, cela se traduit par une réduction dynamique du poids $\omega_{\text{audio},\text{visuel}}(t)$ si l'audio devient momentanément peu fiable, et par le renforcement d'autres liens synergiques.

Les entités associées à différentes modalités peuvent créer des **clusters** communs si leur synergie mutuelle est élevée. Mathématiquement, on peut repérer un cluster \mathcal{C}_m formé d'entités $\{\mathcal{E}_{\text{visuelle}}^a, \mathcal{E}_{\text{auditive}}^b, \mathcal{E}_{\text{textuelle}}^c\}$ tel que

$$\sum_{(i,j) \in \mathcal{C}_m \times \mathcal{C}_m} \omega_{i,j} > \theta_{\text{seuil}},$$

où θ_{seuil} est un paramètre symbolisant la cohésion requise pour se constituer en cluster. Ce regroupement favorise l'**intégration sémantique** de différents types de signaux.

1.3.4.3. Mécanismes d'auto-organisation pour la fusion multimodale

Dans un réseau hiérarchique classique, la fusion multimodale est généralement réalisée dans une “couche” dédiée (p. ex. concaténation de features). Dans le DSL, la fusion naît **spontanément** :

$$\omega_{\text{visuelle},\text{textuelle}}(t+1) = \omega_{\text{visuelle},\text{textuelle}}(t) \eta [S(\mathcal{E}_{\text{visuelle}}, \mathcal{E}_{\text{textuelle}}) - \tau \omega_{\text{visuelle},\text{textuelle}}(t)].$$

Plus la coopération entre les deux modalités est jugée bénéfique (ex. augmentation de la performance ou de l'information mutuelle), plus ce lien se consolide.

Le caractère **évolutif** d'un **cluster multimodal** est essentiel. Selon le **contexte**, la modalité audio peut jouer un rôle plus ou moins important, comme dans le cas d'un concert de musique par rapport à un environnement silencieux. Les liens synergiques inter-modaux varient au fil du temps, entraînant l'émergence ou la dissolution de sous-ensembles cohérents.

On peut également enrichir la mesure de synergie S par un **contexte** $\mathbf{c}(t)$, de sorte qu'il y ait des synergies conditionnelles :

$$S(\mathcal{E}_i, \mathcal{E}_j | \mathbf{c}(t)).$$

Dans une scène nocturne, par exemple, la vision est moins fiable, tandis que l'audio reste pertinent. Le contexte "nuit" pourrait diminuer certaines liaisons visuelles et renforcer des liaisons audio ou thermiques, mettant l'accent sur la modalité la plus fiable.

1.3.4.4. Exemples concrets et gains opérationnels

Imaginez un système de surveillance qui reçoit en continu des flux vidéo, audio, et des capteurs de mouvement. Dans une approche classique, il faudrait concevoir à l'avance un schéma de fusion ; dans le DSL, les **entités** correspondant à chaque flux s'**auto-organisent** pour détecter des patrons communs (ex. détection d'un intrus combinant une silhouette anormale et des sons inhabituels).

Le **renforcement** des liens entre un flux "caméra infrarouge" et un flux "microphones" peut se produire si ces deux modalités se complètent pour la détection nocturne.

Dans le cadre d'interfaces multimodales (voix, gestes, expressions faciales), chaque modalité peut être captée par une entité DSL. Si l'utilisateur parle tout en faisant des gestes, le cluster "voix + gestes" obtient un fort score de synergie, permettant de mieux interpréter la signification globale (commande ou intention).

L'**auto-organisation** permet au système de **s'adapter** aux préférences de l'utilisateur. Si sa diction est peu claire mais que ses gestes sont très expressifs, les pondérations vers la modalité gestuelle se renforcent spontanément.

Le **DSL** peut traiter des couples **image + légende textuelle**. Une entité **extraite de texte** peut renforcer sa synergie avec une entité **détecteur d'objets** dans l'image, en particulier lorsque certains mots clés coïncident avec des formes reconnues.

Les entités "texte" et "image" peuvent alors former un **cluster** spécifique de type "concept visuel + label textuel", facilitant les tâches d'annotation automatique ou de recherche d'images par mots-clés (et vice versa).

1.3.4.5. Forces et défis de l'approche synergique multimodale

L'**absence** de couche de fusion prédéfinie confère une grande **flexibilité**. Plutôt que d'imposer un schéma fixe de combinaison entre les flux visuel, audio ou textuel, le **DSL décide** de manière autonome quelles modalités sont les plus pertinentes à associer. Cette adaptabilité renforce également la **robustesse**. Si un flux devient bruité ou indisponible, comme un microphone défectueux, le système peut se reconfigurer et s'appuyer davantage sur d'autres canaux. Enfin, l'**auto-organisation** permet la **découverte** de combinaisons inédites. Des modalités supposées peu corrélées peuvent révéler une forte synergie dans un certain contexte, ce qu'un pipeline classique plus rigide aurait pu ignorer.

En contrepartie, la **complexité computationnelle** peut croître rapidement à mesure que le nombre de modalités et d'entités augmente. L'évaluation du **gain de synergie** entre plusieurs canaux peut s'avérer coûteuse, rendant nécessaire l'utilisation d'**heuristiques** ou d'**approximations** pour maintenir une efficacité raisonnable. De plus, l'enchaînement rapide de contextes (passer d'une scène de rue à une scène d'intérieur, par exemple) peut induire une **volatilité** élevée des connexions synergiques ; il faut alors recourir à des **coefficients de régulation** (τ) ou à des dispositifs de mémorisation pour **stabiliser** la configuration. Enfin, l'**interopérabilité** entre flux implique de disposer de **représentations** comparables ou d'un espace commun adapté (par exemple, une même métrique pour estimer la similarité entre un vecteur d'image et un vecteur de texte).

1.3.4.6. Conclusion : la multi-modalité comme terrain idéal pour le DSL

Le **Deep Synergy Learning** apparaît particulièrement adapté à l'intégration de **flux multimodaux**. Là où les méthodes classiques imposent généralement une architecture de fusion fixe (par exemple, concaténer l'output d'un CNN d'image et d'un RNN de texte), le DSL laisse les **entités** (issues de différentes modalités) **explorer** leurs synergies possibles. Les liens synergiques se **cristallisent** ou se **dissolvent** en fonction de leur valeur ajoutée, permettant une **adaptation dynamique** aux contextes et aux sources de bruit.

Cette approche auto-organisée favorise l'**émergence** (voir section 1.3.2) de clusters multimodaux plus ou moins stables, qui peuvent évoluer au fil du temps. Sur le plan applicatif, il en résulte une **robustesse** et une **flexibilité** uniques, aussi bien pour la reconnaissance d'événements complexes, la détection d'anomalies, ou encore la mise en place d'interactions homme-machine riches. Les sections suivantes (1.3.5, 1.3.6, 1.3.7) examineront l'**évolution dynamique** de ces réseaux, l'**impact** sur la **résilience** des modèles, et la perspective d'une **approche plus globale** de l'apprentissage.

1.3.5. Évolution Dynamique et Adaptation en Continu

Le **Deep Synergy Learning (DSL)** se distingue des approches classiques de l'IA par sa capacité à laisser les entités d'information et leurs connexions **évoluer** en permanence, au gré de la synergie détectée entre les flux de données. Cette caractéristique d'**évolution dynamique** et d'**adaptation en continu** s'inspire directement de divers phénomènes naturels (cerveau, écosystèmes, colonies d'insectes...) et confère au DSL une **plasticité** inhabituelle dans le domaine de l'apprentissage

automatique. Dans cette section, nous analysons pourquoi cette évolution constante est essentielle, quels sont ses principes fondamentaux et comment elle se formalise mathématiquement.

1.3.5.1. Principes généraux de l'adaptation continue

Dans le **DSL**, l'apprentissage ne se cantonne pas à une phase d'entraînement figée ; il se présente comme un **processus d'auto-organisation** continu, actif tant que le réseau demeure en service. Chaque entité peut ainsi **affiner** ou **redéfinir** ses connexions et son rôle lorsqu'affluent de nouvelles données, ou même de nouvelles entités.

Contrairement à un réseau neuronal classique, la **structure** du réseau n'est pas figée. Les pondérations synergiques $\omega_{i,j}(t)$ peuvent **croître** ou **décroître** jusqu'à atteindre un **seuil** de rupture ω_{\min} , entraînant ainsi la disparition effective de la liaison entre \mathcal{E}_i et \mathcal{E}_j . Ce mécanisme libère des ressources et prévient l'encombrement du réseau. À l'inverse, si une synergie autrefois négligeable devient significative, une **nouvelle liaison** peut se former pour consolider la coopération entre entités (voir la section 1.2.6 concernant les notations).

L'environnement ou la distribution des données peut également **fluctuer**. L'augmentation du bruit, l'apparition de nouvelles classes ou la transformation du contexte modifient les conditions d'apprentissage et d'adaptation du réseau. Dans un schéma statique, il faudrait réentraîner ou procéder à un “fine-tuning” coûteux. Dans le DSL, au contraire, l'**adaptation** est **naturellement** assurée par la mise à jour permanente des **liens synergiques** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Si les informations nouvellement introduites modifient la synergie entre entités, la **topologie** du réseau se **recompose** spontanément, sans nécessité de réapprentissage global.

1.3.5.2. Formalisation mathématique de l'évolution dynamique

On peut représenter le réseau synergique à l'instant t par le graphe $G(t)$, constitué de nœuds $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ et d'arêtes pondérées $\omega_{i,j}(t)$. L'évolution dans le temps se décrit par :

$$G(t+1) = \mathcal{F}(G(t), \mathcal{D}(t), \Theta),$$

où $\mathcal{D}(t)$ désigne les **données reçues** à l'instant t (ou sur la fenêtre $[t, t + \Delta]$), et Θ englobe les paramètres de mise à jour (taux η , régularisation τ , etc.).

Chaque entité \mathcal{E}_i peut maintenir un **état** $\mathbf{s}_i(t)$ permettant d'intégrer une forme de **mémoire** (cf. 1.2.1). Cet état influe sur la synergie :

$$S(\mathcal{E}_i, \mathcal{E}_j) = f(\mathbf{s}_i(t), \mathbf{s}_j(t), \mathcal{D}(t), \dots).$$

On a donc un **système dynamique couplé**, dans lequel la mise à jour des connexions $\omega_{i,j}(t)$ et celle des états $\mathbf{s}_i(t)$ interagissent :

$$\begin{cases} \omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)], \\ \mathbf{s}_i(t+1) = \mathbf{s}_i(t) + \gamma g(\mathbf{s}_i(t), \{\omega_{i,k}(t)\}, \dots). \end{cases}$$

On obtient alors un **couplage** entités–réseau qui peut donner naissance à des attracteurs, des cycles limites ou même des phénomènes de bifurcation.

Au-delà de la simple création/suppression de liens, il est possible que deux entités \mathcal{E}_a et \mathcal{E}_b **fusionnent** si leur collaboration est très constante, formant une **nouvelle entité** \mathcal{E}_{ab} . De même, une entité fortement hétérogène peut se **scinder** en plusieurs entités spécialisées si cela améliore la synergie globale. Ces mécanismes se décrivent par des règles de type :

$$\text{Fusion}(\mathcal{E}_a, \mathcal{E}_b) \Leftrightarrow \sum_{t'=t_0}^{t_1} \omega_{a,b}(t') > \delta_{\text{fusion}},$$

avec δ_{fusion} un seuil de stabilité sur une durée $[t_0, t_1]$. De tels phénomènes rappellent les “assemblées neuronales” durables ou la “spéciation” dans un contexte évolutionniste.

1.3.5.3. Avantages concrets d'une évolution en continu

De nombreux problèmes réels (prévision financière, capteurs industriels, suivi de l'activité cérébrale, etc.) impliquent des distributions de données qui **changent** dans le temps. Le DSL, grâce à son **évolution dynamique**, s'ajuste progressivement plutôt que de s'en tenir à un modèle figé.

Dans un réseau traditionnel, apprendre une nouvelle tâche (ou un nouveau domaine) peut faire oublier les acquis précédents, faute de mécanisme de préservation structurelle. Dans le **DSL**, il n'est pas nécessaire de redémarrer l'optimisation. Les **clusters utiles** peuvent **persistir**, tandis que de nouvelles entités se connectent et ajustent la synergie pour assimiler un contexte différent.

Au fur et à mesure de l'évolution, si certaines liaisons n'apportent plus de gain, elles s'effacent ou diminuent. Ce phénomène **régule** la complexité du réseau en évitant une explosion exponentielle du nombre de liens. On peut donc voir le DSL comme un réseau à **topologie parcimonieuse** autorégulée.

1.3.5.4. Illustrations par l'exemple

Un **système DSL** qui reçoit en continu des mesures (température, pression, vibrations) peut servir à détecter des anomalies ou à anticiper des pannes. Lorsque les conditions d'usage évoluent progressivement, comme l'usure ou la modification de la chaîne de production, le réseau **réorganise** ses liens synergiques. Certains capteurs établissent de nouvelles complémentarités, tandis que des connexions historiques perdent de leur pertinence et s'affaiblissent. Cette **auto-organisation** permet de maintenir une alerte fiable, sans qu'il soit nécessaire de procéder à un nouvel entraînement global à intervalles réguliers.

Un **chatbot multimodal** traitant à la fois la voix, le texte et l'image du visage (via webcam) illustre également l'adaptabilité d'un réseau évolutif. À mesure que l'utilisateur se familiarise avec le

système (introduction de nouveaux termes, évolution de ses habitudes gestuelles), les entités “voix” et “vision” ajustent leurs connexions et peuvent, le cas échéant, **former un cluster** spécialisé pour reconnaître l’utilisateur concerné. Les anciennes tâches (reconnaître d’autres utilisateurs) ne sont pas oubliées, car elles persistent dans des clusters distincts ; la connaissance acquise est ainsi **préservée** sur le long terme.

1.3.5.5. Limites et défis de l’évolution continue

Si le réseau cherche à suivre de trop près les **fluctuations** de la synergie, il peut devenir **instable**, oscillant sans cesse ou détruisant prématulement des liens pertinents. Les paramètres η , τ , δ_{fusion} , etc., exigent donc un **calibrage** soigné pour maintenir un **équilibre** entre plasticité et stabilité (c’est le “dilemme stabilité–plasticité”).

Par ailleurs, dans un **modèle statique**, on fige les poids après l’apprentissage et on évalue la performance sur un ensemble de test. Dans le cadre d’un **réseau évolutif**, la performance est **susceptible de varier** au fil du temps, rendant nécessaire l’adoption de **métriques d’apprentissage continu** (p. ex. une mesure de l’erreur sur une fenêtre glissante).

Enfin, la reconfiguration permanente des liens implique un **coût de calcul**. Si le réseau compte n entités et que la synergie est évaluée entre toutes les paires, on fait face à $O(n^2)$ opérations. Des stratégies de **parsimonie** (seuil de rupture, mise à jour partielle, échantillonnage) sont souvent indispensables pour demeurer **scalable** dans des systèmes de grande ampleur.

1.3.5.6. Conclusion : une architecture “vivante” pour l’IA

L’**évolution dynamique** et l’**adaptation en continu** font du **Deep Synergy Learning** une forme de réseau “vivant”, capable de remodeler son organisation interne au fur et à mesure que l’environnement (ou les données) se transforment. Ce **changement de paradigme** – de l’entraînement ponctuel et figé à une auto-organisation permanente – ouvre la voie à des **applications** de l’IA plus flexibles et plus robustes, réduisant la dépendance à la supervision humaine et accroissant la **longévité** des modèles dans des scénarios réels et évolutifs.

Les sections suivantes, **1.3.6** et **1.3.7**, approfondiront deux aspects essentiels de cette approche. La section **1.3.6** explorera la **robustesse** et la **résilience** apportées par l’auto-organisation, tandis que la section **1.3.7** abordera les **pistes** vers une approche plus générale de l’apprentissage, potentiellement orientée vers une IA plus proche de la **cognition** ou de l’**intelligence générale**.

1.3.6. Impacts sur la Robustesse et la Résilience des Modèles

L’une des conséquences les plus marquantes de l’**auto-organisation** et de la **synergie informationnelle** dans le **Deep Synergy Learning (DSL)** réside dans l’**amélioration** de la **robustesse** et de la **résilience** des systèmes d’apprentissage. Contrairement aux approches classiques, souvent vulnérables aux perturbations, aux changements de distribution ou aux défaillances partielles, le DSL tire parti de sa structure **évolutive** et **coopérative** pour mieux

absorber ces aléas. Dans la présente section, nous examinons les notions de robustesse et de résilience, puis montrons comment elles s'expriment dans un réseau synergique.

1.3.6.1. Définitions et enjeux

La **robustesse** d'un modèle désigne sa capacité à **maintenir** un niveau de performance élevé malgré la présence de **perturbations** ou d'**incertitudes**. Dans le contexte de l'IA, ces perturbations peuvent inclure :

- Du **bruit** dans les données (capteurs défectueux, images floues, signaux audio corrompus, etc.).
- Des **attaques adversariales** (petites perturbations “adversarial noise” rendant le modèle confus).
- Des **incohérences** ou **lacunes** dans les échantillons (valeurs manquantes, distribution très variée).

Un système robuste parvient à faire face à ces dégradations sans “collapsus” brutal de la performance.

La **résilience** va plus loin que la robustesse en impliquant non seulement la résistance aux perturbations, mais aussi la **capacité à récupérer** ou à **se réorganiser** après un choc. En IA, cela signifie que si la distribution des données change, ou si le système subit une panne partielle (certains capteurs tombent en panne, certains flux de données disparaissent), la résilience se manifeste par la **reconfiguration** interne permettant de continuer la tâche ou de s'en rapprocher au mieux.

1.3.6.2. Mécanismes de robustesse dans un réseau synergique

Dans le DSL, plusieurs entités d'information peuvent recouvrir partiellement la même “zone de compétence” ou la même modalité, tout en se distinguant suffisamment pour apporter une **valeur ajoutée**. Du point de vue de la robustesse, cela signifie qu'un flux corrompu (ou une entité dysfonctionnelle) n'est pas catastrophique, car d'autres entités peuvent **prendre le relais**.

$$\omega_{i,j}(t) \rightarrow \omega_{k,j}(t+1), \quad \text{si } \mathcal{E}_i \text{ défaillante, alors } \mathcal{E}_k \text{ peut compenser.}$$

Lorsque la synergie entre une entité défectueuse \mathcal{E}_{def} et les autres entités chute (bruit, erreur répétée, etc.), les pondérations $\omega_{\text{def},j}(t)$ s'affaiblissent progressivement. Le réseau “apprend” ainsi à **alléger** la connexion avec la source fautive et à **rediriger** l'information vers des entités plus fiables. Ce mécanisme évite un fort impact d'une entité isolée dysfonctionnelle.

Les entités réagissent localement au **bruit** ou aux **incohérences** en resserrant leurs liens synergiques avec celles qui restent cohérentes. On observe alors la création de **clusters** (voir 1.2.6) qui s'auto-assemblent autour de flux fiables ou complémentaires, renforçant la robustesse globale.

$$\mathcal{C}^* = \arg \max_{\mathcal{C}} \sum_{(i,j) \in \mathcal{C} \times \mathcal{C}} \omega_{i,j}, \quad \text{en excluant } \mathcal{E}_{\text{def}}.$$

1.3.6.3. Résilience via l'adaptation en continu

Le DSL met en œuvre un apprentissage **continu** (section 1.3.5), ce qui signifie que le réseau s'ajuste aux nouvelles données ou aux nouveaux contextes sans nécessiter une refonte globale. Ainsi, si une modalité devient soudainement imprécise (caméra saturée de lumière, micro exposé à un fort bruit ambiant), le réseau réorganise ses **pondérations synergiques** pour ne plus dépendre de cette source.

$$\omega_{\text{camera}, \mathcal{E}_j}(t+1) = \omega_{\text{camera}, \mathcal{E}_j}(t) - \eta [\tau \omega_{\text{camera}, \mathcal{E}_j}(t)] \quad \text{si } S(\mathcal{E}_{\text{camera}}, \mathcal{E}_j) \approx 0 \text{ (bruit élevé).}$$

La résilience suppose la **récupération** ou le maintien d'une performance acceptable même après un choc. Le fait de **supprimer** ou d'**affaiblir** les liens inutiles (ou trompeurs) et de **renforcer** les liens pertinents génère une reconfiguration topologique :

$$G(t+1) = \mathcal{U}[G(t), \{\omega_{i,j}\}, \text{nouveaux contextes}].$$

Ainsi, le réseau peut se redessiner sans qu'on doive recourir à une procédure hors-ligne complexe.

Les entités (et clusters) qui étaient **efficaces** pour d'anciennes tâches peuvent persister en parallèle, assurant une forme de **mémoire** ou de **transfert** vers des scénarios futurs. Cette coexistence entre anciens et nouveaux liens renforce la **résilience** du réseau. Plutôt que d'éliminer systématiquement ses anciennes connaissances, il les conserve tant qu'elles restent **synergiques**, évitant ainsi le phénomène de **catastrophic forgetting**.

1.3.6.4. Exemples d'Applications Robustes et Résilientes

Le **Deep Synergy Learning (DSL)** se révèle particulièrement adapté aux situations où la robustesse et la résilience constituent des critères essentiels. Deux scénarios illustrent la manière dont la **synergie informationnelle** et la **dynamique auto-organisée** du DSL permettent d'absorber des pannes, de réagir à des anomalies et de préserver la continuité opérationnelle. Dans chaque cas, la structuration coopérative du réseau offre une forme de **tolérance aux défaillances**, car l'affaiblissement d'un module ou d'un capteur ne remet pas en cause l'ensemble de la configuration. Les **exemples** ci-dessous mettent en évidence ces mécanismes de résilience et soulignent les **bienfaits** d'un réseau qui se **reconfigure** en temps réel.

A. Robotique et Systèmes Autonomes

Considérons un **robot** doté de multiples capteurs, tels que des caméras, un LIDAR, des gyroscopes, ainsi que d'autres senseurs spécialisés. Dans ce contexte, un modèle DSL peut absorber la défaillance partielle ou totale d'un capteur, sans qu'il soit nécessaire de procéder à un réapprentissage global. Cette propriété découle du caractère **auto-adaptatif** du Synergistic Connection Network (SCN) sous-jacent, où chaque capteur \mathcal{C}_i est traité comme une **entité** de

l'ensemble, tandis que les liens $\omega_{i,j}$ mesurent la **coopération** ou la **complémentarité** de deux capteurs i et j .

Dans le cas d'une **défaillance**, supposons que le capteur \mathcal{C}_k soit victime d'un bruit extrême ou d'une panne soudaine. Le **score de synergie** $S(\mathcal{C}_k, \mathcal{C}_m)$ avec les autres capteurs \mathcal{C}_m diminuerait alors, ce qui provoquerait une mise à jour négative des liens $\omega_{k,m}$ par l'équation

$$\omega_{k,m}(t+1) = \omega_{k,m}(t) + \eta[S(\mathcal{C}_k, \mathcal{C}_m) - \tau \omega_{k,m}(t)],$$

où η désigne le taux d'apprentissage local et τ le coefficient de régulation. À mesure que la **synergie** s'affaiblit, la pondération $\omega_{k,m}$ décroît, de sorte que ce capteur devient quasi inactif au sein du SCN. Les autres capteurs, préservant des liens ω élevés entre eux s'ils demeurent fiables et cohérents, continuent de coopérer pour fournir les informations nécessaires à la navigation ou à la réalisation des tâches robotiques. L'ajustement s'effectue en continu, sans qu'un entraînement complet soit relancé, montrant ainsi la **résilience** inhérente au DSL.

B. Surveillance Critique

Dans les domaines de la **surveillance** ou du **monitoring** (installations sensibles, applications médicales, détection d'intrusions), il est fréquent de recourir à un grand nombre de capteurs hétérogènes. Un système basé sur le DSL peut y déployer ses mécanismes d'**auto-organisation** de manière à détecter plus vite d'éventuelles anomalies et à tolérer les dysfonctionnements partiels d'un ou plusieurs capteurs. Le déploiement s'opère suivant la logique où chaque capteur \mathcal{D}_i constitue un **nœud** du SCN, et où le **score de synergie** $S(\mathcal{D}_i, \mathcal{D}_j)$ quantifie la correspondance ou la complémentarité entre deux capteurs.

Lorsque l'un de ces nœuds présente une anomalie ou se déconnecte, la **pondération** $\omega_{i,j}$ chute au fil des itérations, car la synergie mesurée devient faible. Mathématiquement, le processus de mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{D}_i, \mathcal{D}_j) - \tau \omega_{i,j}(t)]$$

réduit progressivement les liens inutilisables ou compromis. La structure globale demeure néanmoins cohérente, car les capteurs restant fiables maintiennent des valeurs de ω importantes, ce qui préserve la capacité du réseau à **fonctionner**.

Le phénomène inverse se produit lorsqu'une alarme est déclenchée de façon synchronisée entre plusieurs capteurs. Dès lors que ceux-ci fournissent un signal convergent sur la présence d'un événement inhabituel, leur **synergie** augmente, et ils forment un **cluster** spécialisé dans l'analyse d'un risque potentiel. Cette coopérative rapide et flexible permet au SCN de hiérarchiser l'information critique. L'avantage repose sur l'**absence de réapprentissage intégral** et sur la capacité du système à s'**auto-réorganiser** pour mettre en avant les flux les plus pertinents.

Conclusion

Que ce soit dans un scénario de **robotique** multi-capteurs ou dans un dispositif de **surveillance** critique, le **Deep Synergy Learning** démontre ainsi son **aptitude** à gérer la panne ou la perte de fiabilité d'un sous-module sans perturber l'ensemble de l'architecture. La dynamique de **mise à jour** des synergies et des liens ω assure une continuité de service et un renforcement rapide des

signaux valides, ce qui illustre la solidité opérationnelle et l'efficacité de cette **approche auto-organisée** pour la gestion des situations critiques ou complexes.

1.3.7. Perspectives pour une Approche plus Globale de l'Apprentissage

Les sections précédentes (1.3.1 à 1.3.6) ont mis en évidence l'importance de l'**auto-organisation** dans le Deep Synergy Learning (DSL), ses fondements biologiques et cognitifs (1.3.1), les concepts d'émergence et de feedback (1.3.2), la comparaison avec les méthodes classiques (1.3.3), le rôle crucial de la multi-modalité (1.3.4), la dynamique adaptative (1.3.5) et l'impact sur la robustesse/résilience (1.3.6). Nous abordons maintenant une question centrale. **Dans quelle mesure** cette architecture auto-organisée peut-elle contribuer à une **approche plus globale** de l'apprentissage, ouvrant la voie à une IA plus générale, moins cloisonnée et plus proche de la **cognition humaine** ?

1.3.7.1. Vers une intégration de multiples paradigmes

L'un des attraits majeurs du DSL réside dans sa **flexibilité** structurelle, permettant l'intégration simultanée de tâches variées (reconnaissance d'images, traitement du langage, analyse de signaux). Plutôt que de multiplier les sous-modules indépendants, le DSL **mutualise** la capacité d'adaptation et de co-évolution. Cela ouvre la voie à des systèmes capables d'aborder des **problèmes multi-domaines** de façon unifiée.

Beaucoup de chercheurs s'interrogent sur la possibilité de combiner une **représentation symbolique** (logique, règles, ontologies) et des **réseaux neuronaux** (approche sub-symbolique). Le DSL offre un espace où des **entités symboliques** (représentant des concepts, des règles) pourraient coexister avec des **entités sub-symboliques** (features non supervisées, clusters contextuels), et forger des **liens synergiques** si ces représentations s'avèrent mutuellement bénéfiques.

$$\omega_{\text{symbolique}, \text{sub-symbolique}}(t+1) = \omega_{\text{symbolique}, \text{sub-symbolique}}(t)\eta[S(\mathcal{E}_{\text{symbolique}}, \mathcal{E}_{\text{sub-symbolique}})].$$

Une autre piste consiste à inclure des **boucles de récompense** (façon apprentissage par renforcement) à l'intérieur du DSL, de manière que la **synergie** prenne en compte non seulement l'interaction entre entités, mais aussi un **signal de performance** plus global. Cela pourrait mener à des architectures où l'**exploration** et la **sélection** des connexions synergiques s'effectuent au service d'une stratégie d'agent, par exemple dans un environnement en évolution (robotique, jeux, etc.).

1.3.7.2. Apprentissage contextuel et raisonnement adaptatif

Dans de nombreux problèmes (analyse de scène, dialogue, diagnostic médical), l'**environnement** ou la **situation** évolue. Le DSL permet de faire émerger des **clusters contextuels**, regroupant les entités les plus pertinentes pour un contexte donné. Ces clusters peuvent ensuite se **désagréger** ou

se **recombiner** quand le contexte change, offrant un niveau de **contextualisation dynamique** qu'on retrouve rarement dans les modèles hiérarchiques figés.

Plutôt que de raisonner par règle (“si A alors B”), le réseau peut **découvrir** que certaines combinaisons d’entités $\{\mathcal{E}_i, \mathcal{E}_j, \dots\}$ produisent un **effet global** pertinent. On pourrait formaliser cette découverte comme un **raisonnement émergent**, où l’apparition d’un cluster synergique équivaut à la création d’un “concept” ou d’une “hypothèse” confirmée par l’amélioration de la performance ou de la cohérence.

$$\text{Concept}_{\alpha} \leftrightarrow \{\mathcal{E}_i, \mathcal{E}_j, \dots\} \quad \text{avec} \quad \sum_{(i,j) \in \alpha} \omega_{i,j} > \theta.$$

Les entités dans le DSL peuvent porter un **état interne** (1.2.1), ce qui permet de **mémoriser** certains événements, associations ou transitions. Couplé à l’idée de clusters, on obtient une forme de mémoire **distribuée** et **adaptative**, capable de se ré-agencer au fil du temps, au lieu d’être confinée dans des architectures rigides (ex. LSTM).

1.3.7.3. Ouverture vers une IA plus proche de la cognition biologique

Nous avons déjà souligné les parallèles entre la synapse biologique et les connexions synergiques (sections 1.3.1, 1.3.2). Un système auto-organisé qui évolue au fil du temps, fusionne ou dissocie des entités, renforce ou supprime des liens, s’approche d’une **dynamique neuronale** élémentaire. Il est donc envisageable que des **architectures DSL** se montrent plus aptes à **simuler** ou **comprendre** certains phénomènes cognitifs.

Bien que spéculative, la question d’une **conscience artificielle** ou d’une **auto-consistance** cognitive pourrait trouver dans le DSL un champ d’exploration. Certains travaux suggèrent qu’un degré d’**intégration d’information** (comme l’approche de la “Phi measure” de Giulio Tononi) est requis pour émerger une forme de conscience. Si le DSL parvient à développer des **clusters** hautement connectés et persistants (au sens “informationnellement intégrés”), il pourrait servir de terrain expérimental pour avancer sur ces hypothèses.

L’IA forte (IAG) ou Intelligence Artificielle Générale se définit par la capacité d’un système à maîtriser un large éventail de tâches, à apprendre de manière autonome et à faire face à des environnements variés. La **plasticité** et la **co-évolution** du DSL, sa manière de laisser chaque entité s’adapter, fusionner, se spécialiser, suggèrent un potentiel pour **surmonter** les obstacles auxquels se heurtent les méthodes rigides (catastrophic forgetting, absence de transfert, etc.). Cela ne garantit pas l’émergence d’une IAG, mais en offre une piste conceptuelle plus proche du fonctionnement adaptatif du vivant.

1.3.7.4. Principaux défis et voies de recherche

Le **Deep Synergy Learning (DSL)** introduit un cadre inédit, reposant sur une **interaction dynamique** et une **auto-organisation** continue des entités d’information. Malgré l’attrait de cette proposition, il subsiste plusieurs défis mathématiques, algorithmiques et pratiques, dont la résolution s’avère essentielle pour la généralisation à grande échelle. Sur le plan **computational**,

l'évaluation des synergies entre un nombre important d'entités peut impliquer une complexité en $O(n^2)$, où n désigne le nombre d'entités. Cette explosion combinatoire requiert des stratégies de **parsimonie** et de **sparsification** afin de restreindre la croissance du nombre de connexions à évaluer. Des techniques d'**échantillonnage** adaptatif peuvent aussi être envisagées, de manière à ne considérer que les paires d'entités présentant une synergie potentiellement élevée. D'un point de vue plus formel, on peut réduire la densité des connexions actives en imposant un mécanisme de seuil sur les poids ω_{ij} , de sorte qu'une liaison (i, j) ne soit conservée que si

$$\omega_{ij} > \omega_{\min},$$

avec ω_{\min} un paramètre choisi pour garantir la structure globale du réseau.

S'agissant de l'**interprétabilité** et de l'**explicabilité**, la capacité du DSL à faire émerger des **clusters** et à redessiner en continu ses connexions favorise une organisation plus fluide, mais rend ardue l'analyse a posteriori des décisions. Pour éclairer la nature d'une prédiction, il devient nécessaire de retracer l'historique des synergies et d'examiner l'état interne des entités concernées. La mise à jour de chaque poids ω_{ij} peut être décrite par une équation de type

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{ij}(t)],$$

où η désigne le taux d'apprentissage et τ le coefficient de décroissance. Cette dynamique rend possible l'apparition ou la suppression de liens dont la contribution n'est plus jugée pertinente, tout en renforçant ceux qui semblent les plus profitables au regard de la synergie mesurée. Cependant, un tel fonctionnement complique la **traçabilité**. Pour extraire des règles ou des justifications, il serait nécessaire de stocker ou de synthétiser l'ensemble des états successifs ainsi que les évaluations de $S(\mathcal{E}_i, \mathcal{E}_j)$.

La question de la **convergence** suscite également un vif intérêt. L'étude de la stabilité et des attracteurs potentiels liés aux mises à jour des poids dans le cadre d'un **système dynamique** exige d'analyser l'évolution de l'**énergie** du réseau ou d'une **fonction de Lyapunov** associée. Si l'on considère une énergie

$$E(t) = - \sum_{i \neq j} \omega_{ij}(t) S(\mathcal{E}_i, \mathcal{E}_j),$$

la décroissance de $E(t)$ au fil des itérations peut servir d'indicateur de convergence vers un état stable où les pondérations cessent d'évoluer de manière significative. L'identification de conditions garantissant l'existence d'un minimum global ou local, ainsi que la caractérisation de la vitesse de convergence, demeurent des questions ouvertes pour la communauté scientifique. La présence de boucles de rétroaction complexes, de reconfigurations topologiques et d'interactions non linéaires peut engendrer des comportements oscillatoires, voire chaotiques, si aucune **régularisation** adéquate n'est mise en place.

Dans les **domaines critiques** (santé, finance, transport, sécurité), le DSL pourrait apporter une meilleure **robustesse** grâce à la **fusion multimodale** et à la **répartition adaptative** des informations, permettant de gérer des données hétérogènes ou partiellement manquantes. Toutefois, de telles applications exigent des garanties fortes sur la fiabilité et la sûreté de fonctionnement, impliquant des **processus de certification** exigeants. Il devient essentiel d'y

intégrer des mécanismes de sécurité contre d'éventuelles attaques adversariales, où de subtiles perturbations de l'entrée peuvent tromper même des systèmes neuronaux performants. La dynamique adaptative et coopérative du DSL pourrait, d'un côté, renforcer la **résilience** en détectant des incohérences via les synergies internes, mais de l'autre côté, elle complique l'établissement de **bornes** de sûreté ou de **preuves** formelles de robustesse.

Ainsi, malgré un potentiel considérable, le DSL fait face à des enjeux théoriques et pratiques déterminants pour son adoption à grande échelle. La **réduction de la complexité algorithmique**, l'**amélioration de l'interprétabilité**, l'**étude minutieuse de la convergence** et l'**adaptation aux environnements critiques** constituent autant de pistes de recherche centrales pour consolider ce paradigme. Les avancées futures consisteront notamment à concevoir des **stratégies d'approximation** limitées en ressources de calcul, à développer des **méthodes d'exploration** et de **visualisation** capables d'exposer la structure émergente du réseau, et à établir des **cadres mathématiques** plus complets validant la stabilité et la fiabilité du processus d'**auto-organisation**. L'ensemble de ces travaux pourra contribuer à faire du Deep Synergy Learning un jalon clé dans l'évolution de l'IA vers une plus grande **plasticité, résilience et intelligence**.

1.3.7.5. Conclusion : vers un nouveau paradigme d'apprentissage

En somme, l'auto-organisation et la synergie, éléments centraux du **Deep Synergy Learning**, ouvrent la voie à une **approche plus globale** de l'apprentissage, susceptible d'englober plusieurs paradigmes existants (supervisé, non supervisé, renforcement, symbolique, etc.) tout en proposant des mécanismes d'évolution continue, de raisonnement contextuel et de mémorisation distribuée.

Cette vision ne constitue pas encore un aboutissement. De nombreux **défis** scientifiques, méthodologiques et éthiques, tels que la **scalabilité**, l'**interprétabilité** et la **robustesse**, restent à relever. Néanmoins, le DSL fournit d'ores et déjà un **cadre unificateur** pour penser l'**IA adaptative**, apte à accueillir de multiples flux d'information et à co-évoluer de façon organique, en s'inspirant des systèmes vivants.

Les chapitres suivants approfondiront encore la **formulation mathématique** du DSL, ses **algorithmes** et ses **applications pratiques**, en vue de concrétiser ce nouveau paradigme et d'explorer son potentiel pour des systèmes intelligents, au-delà des limites imposées par les architectures hiérarchiques classiques.

1.4. Architecture Générale du DSL

Dans les chapitres précédents, nous avons défini les **fondements conceptuels** du Deep Synergy Learning (DSL), en soulignant notamment l'importance de l'**auto-organisation** et des **synergies** entre entités d'information (sections 1.2 et 1.3). Nous savons désormais que chaque entité ne se limite pas à un simple vecteur de données. Elle peut **évoluer, interagir et générer de nouvelles représentations** lorsqu'elle coopère avec d'autres entités. Ces processus de coopération ne sont pas imposés par un schéma hiérarchique rigide, mais émergent librement à mesure que les entités détectent un **gain** dans leur association.

Afin de concrétiser ces principes dans un **cadre uniifié**, il est nécessaire de décrire l'**architecture générale** du DSL. Autrement dit, comment ces entités s'organisent-elles ? Quelles structures se dégagent quand la synergie est élevée entre certaines paires (ou ensembles) d'entités ? Et comment modéliser leur évolution au fil du temps ? La section 1.4 répond à ces questions en introduisant :

- Les **principes de base** des entités et des liens (1.4.1)
- La notion de **Synergistic Connection Network (SCN)** (1.4.2)
- La formation de **clusters** et de **macro-clusters** (1.4.3)
- Les différentes **fonctions de synergie** (1.4.4)
- Le fonctionnement **adaptatif** des pondérations (1.4.5)
- Les interactions **directes** et **indirectes** (1.4.6)
- Les cas particuliers de **synergie binaire** et **n-aire** (1.4.7)

C'est cette **architecture globale** du **DSL** qui confère au système sa **plasticité** et sa **capacité d'adaptation**. Plutôt que de transmettre passivement les données à travers des couches fixes, chaque entité peut, en temps réel, **renforcer ou affaiblir** ses liens avec d'autres, **former des clusters spontanés**, et **faire émerger** des représentations plus riches que la simple somme de leurs composantes individuelles.

1.4.1. Principe de Base : Entités et Liens Synergiques

Le **Deep Synergy Learning** envisage chaque **entité d'information** comme un **nœud actif** de l'architecture globale. Plutôt que de propager passivement un signal, comme le ferait un neurone dans un réseau hiérarchique, cette entité adopte un rôle décisif en disposant de **paramètres internes** θ_k , souvent constitués d'une **représentation** \mathbf{x}_k (vectorielle ou tensorielle), d'un **état interne** \mathbf{s}_k qui se modifie au gré des interactions, et de tout **hyperparamètre** requis pour la modalité concernée (par exemple un encodeur audio ou un embedding textuel). Chaque entité s'accompagne également de **mécanismes de décision** qui lui permettent d'**observer** la synergie obtenue lorsqu'elle se connecte à d'autres entités et, selon cette observation, de renforcer, d'affaiblir ou de rompre certains liens. Elle peut même, dans les cas où la coopération se révèle durablement élevée, envisager une **fusion** avec une autre entité ou, au contraire, une **spécialisation** en segmentant ses propres paramètres pour répondre à des tâches spécifiques. Cette description,

encore préliminaire, fait déjà ressortir l'idée que les **nœuds** du DSL sont bien plus dynamiques que dans un réseau de neurones classique, puisque chacun "cherche" les coopérations les plus fructueuses pour améliorer sa représentation tout en contribuant à la robustesse du réseau dans son ensemble.

Chaque entité \mathcal{E}_i peut se connecter à une autre entité \mathcal{E}_j par le biais d'une **pondération synergique** $\omega_{i,j}(t)$, laquelle traduit la **valeur ajoutée** perçue dans cette association. Quand la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ est forte, on observe un accroissement de $\omega_{i,j}(t)$, tandis que dans le cas contraire, ce lien tend à se résorber puis à disparaître. L'évolution de $\omega_{i,j}$ se modélise alors par une équation simple, par exemple

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η désigne le taux d'apprentissage et τ un paramètre de régularisation. Cette formulation, exposée plus en profondeur dans les sections ultérieures, peut aussi s'enrichir de seuils pour limiter la croissance des poids ou supprimer les connexions trop faibles, assurant de la sorte une **régulation** de la densité du graphe. Une fois que l'on modélise ainsi les **liens synergiques**, on se dote d'une **matrice d'adjacence** évolutive qui dicte quels nœuds interagissent, quand ils le font et avec quelle intensité.

La grande différence avec une organisation hiérarchique est que l'information n'est plus obligée de transiter selon des voies prédéfinies. Toute entité peut potentiellement échanger avec n'importe quelle autre, pour peu que la **pondération synergique** atteigne un niveau jugé satisfaisant. Dans le cadre d'une tâche multimodale, il est donc envisageable que des entités "visuelles" se lient directement à des entités "textuelles" si elles y trouvent un bénéfice mutuel. Il se forme alors des **clusters** éphémères ou plus durables, réunissant deux, trois ou davantage d'entités, et la persistance de ces regroupements dépend de la permanence de leur synergie. De tels micro-réseaux se constituent et se défont librement, donnant lieu à une **auto-organisation** dont l'architecture n'a pas été spécifiée a priori mais découle de la dynamique interne du système.

Cette perspective se décrit volontiers comme un **graphé évolutif** $G(t)$. Les **nœuds** correspondent aux entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$, et les **arêtes** représentent les liaisons $\omega_{i,j}(t)$. À chaque itération, une règle d'actualisation \mathcal{U} recalcule $\omega_{i,j}(t+1)$ en tenant compte de la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ et de la **régularisation** (qui inclut, au besoin, la suppression des liens faibles). Cela rend possibles des analyses croisées avec la **théorie des graphes** ou la **dynamique non linéaire**, et l'on peut même faire appel à des méthodologies d'**optimisation combinatoire** lorsqu'on cherche, par exemple, à minimiser un coût global ou à identifier une configuration de connexions correspondant à un optimum local.

Cette approche basée sur des **nœuds actifs** et des **arêtes dynamiques** confère au **DSL** une **flexibilité** qui se révèle supérieure à celle de nombreuses architectures traditionnelles. La topologie n'est plus figée dans des couches de traitement ; elle se réorganise au gré des opportunités de synergie détectées, ce qui est d'un intérêt crucial pour les situations multimodales ou pour l'adaptation à des données en évolution. Les entités elles-mêmes peuvent fusionner, se subdiviser ou changer de paramétrage, et toute la structure s'oriente vers une organisation où la synergie la plus forte est mise en avant. Cette logique favorise l'apparition de **propriétés émergentes** comme la consolidation de **macro-clusters** ou l'élaboration de schémas coopératifs inédits, sans qu'aucune hiérarchie ne soit imposée dès le départ.

En somme, la clef de voûte du **DSL** réside dans l'idée que chaque entité constitue un **nœud autonome**, et chaque connexion, un **lien synergique** dont l'évolution est régie par le degré d'enrichissement mutuel. Les observations de clusters, de reconnections ou de regroupements imprévus y sont donc la norme, et ce caractère **distribué** et **adaptatif** positionne le DSL comme un prolongement original des paradigmes existants, taillé pour gérer la variété et la complexité croissantes des données actuelles.

1.4.2. Présentation du Synergistic Connection Network (SCN)

Le **Synergistic Connection Network (SCN)** constitue la **pièce maîtresse** du Deep Synergy Learning (DSL) en matérialisant à la fois l'**espace** dans lequel les entités évoluent et le **mécanisme** même qui autorise la naissance, la transformation ou la disparition de leurs liens. Alors que les réseaux de neurones classiques reposent sur une **topologie fixée** avant l'entraînement, le SCN propose, au contraire, une **structure adaptive** dont les entités et leurs connexions se reconfigurent de manière autonome au fil du temps.

La représentation du **SCN** sous forme de **graphe** positionne les **entités** $\{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n\}$ en tant que **nœuds**, reliés par des **pondérations synergiques** $\omega_{i,j}$ évolutives. Chaque entité correspond à un bloc fonctionnel, qu'il soit **visuel, textuel ou auditif**, et ajuste dynamiquement ses connexions en fonction de la **synergie** détectée. Elle peut ainsi **renforcer** ou, au contraire, **réduire** sa liaison avec une autre entité selon la pertinence de leur interaction. Cette **plasticité** donne au **SCN** un caractère **vivant**. Sa structure n'est jamais figée mais se **recompose** en permanence, en conservant les connexions jugées utiles et en éliminant celles qui ne le sont plus.

A. Les principes fondateurs du SCN

Coopération locale, cohérence globale

Au cœur du **SCN**, chaque liaison $\omega_{i,j}$ évolue selon une **décision locale**. Deux entités évaluent leur **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$, qu'elle repose sur une **similarité**, une **co-information** ou un **gain de performance**, puis ajustent la pondération $\omega_{i,j}$. Pourtant, cette dynamique locale influence la **structure globale**. En synchronisant leurs mises à jour, des milliers ou des millions de connexions peuvent, en quelques itérations, donner naissance à des **macro-structures** ou des **clusters** (voir section 1.4.3).

Adaptation continue

Plutôt qu'un apprentissage ponctuel, le SCN procède par **itérations** successives. Les pondérations $\omega_{i,j}$ se recalculent via une équation du type

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η indique le rythme de l'apprentissage et τ limite la persistance d'un lien. Cette **dynamique** empêche toute stagnation et confronte en permanence l'évolution des connexions aux signaux de synergie qui apparaissent ou disparaissent.

Mécanismes de régulation

Pour qu'un trop grand nombre de liaisons ne se maintienne artificiellement, on peut **couper** celles dont la pondération demeure en deçà d'un certain **seuil** ω_{\min} . À l'inverse, fixer un **plafond** ω_{\max} empêche l'explosion de liens exagérément forts. Ces stratégies renforcent la **parsimonie** du réseau. En ne conservant que les connexions bénéfiques, elles favorisent la formation de sous-ensembles stables et révèlent des motifs structurels plus riches.

B. Du réseau neuronal traditionnel au SCN

Un **réseau de neurones classique** repose sur une **architecture prédéfinie**. Chaque couche contient un nombre fixe de neurones, et les connexions entre elles restent invariables. Bien que les poids internes soient ajustés par la rétropropagation du gradient, la **topologie** du réseau, comprenant le nombre de couches, la disposition des neurones et le schéma des connexions, demeure **immuable**.

À l'inverse, le **SCN** adopte un paradigme fondamentalement différent. La **topologie évolue** spontanément, permettant aux connexions d'apparaître ou de disparaître en fonction de la **synergie** détectée. Les entités sont **actives** et disposent d'un état interne ou de mécanismes de représentation qui peuvent s'adapter pour optimiser leur **coopération**. Le réseau peut ainsi **se réinventer** en fonction de l'arrivée de nouvelles données ou de changements contextuels. Des **clusters spécialisés** émergent et disparaissent naturellement, sans nécessiter d'intervention extérieure pour redéfinir la structure.

Avec cette **plasticité**, le SCN parvient à incorporer et à privilégier les synergies révélées par les données, adaptant la répartition des connexions bien plus librement qu'un réseau hiérarchique prédéterminé.

C. Domaines d'application du SCN

Les **domaines d'application du SCN** exploitent sa capacité à s'adapter dynamiquement aux interactions entre entités et à optimiser l'apprentissage de manière continue.

La **multimodalité fluide** repose sur l'association flexible des différentes modalités, qu'il s'agisse de la vision, du texte, de l'audio ou de capteurs spécialisés. Le **SCN** établit et supprime des connexions en fonction de la **valeur ajoutée** qu'elles apportent. Les entités **visuelles** peuvent ainsi se connecter directement à des entités **textuelles** lorsqu'un **gain mutuel** est identifié, favorisant une intégration optimale des données.

L'**apprentissage continu et évolutif** permet au **SCN** de **réajuster** sa configuration en permanence dans un environnement changeant. Les liens utiles sont renforcés tandis que les connexions moins pertinentes sont progressivement délaissées. Cette plasticité itérative confère au réseau une **résilience accrue**, évitant ainsi un réapprentissage intégral à chaque modification de distribution des données.

La **découverte de patrons “n-aires”** va au-delà des interactions par paires. Le **SCN** favorise l'émergence de **macro-clusters**, où plusieurs entités $\{\mathcal{E}_1, \dots, \mathcal{E}_m\}$ coopèrent pour générer une **plus-value collective** qui dépasse la simple somme de leurs synergies binaires.

Dans la section suivante (1.4.3), nous verrons comment ces liens synergiques favorisent la naissance de **clusters auto-organisés**, et en quoi ceux-ci peuvent se regrouper en **macro-clusters**

pour donner naissance à des entités d'information plus puissantes, ou plus abstraites, dans la démarche d'apprentissage distribué du Deep Synergy Learning.

1.4.3. Notion de Cluster et de Macro-Cluster

Dans le **Synergistic Connection Network (SCN)**, décrit à la sous-section précédente (1.4.2), chaque entité d'information \mathcal{E}_i est reliée aux autres par des **pondérations synergiques** $\omega_{i,j}(t)$ qui évoluent dans le temps. Ce mécanisme permet l'**émergence spontanée** de sous-structures au sein du réseau. Des groupes d'entités présentant une **synergie élevée** ont tendance à **collaborer** plus fréquemment, formant ainsi des **clusters**.

Au-delà de ces regroupements de base, il est possible que **plusieurs** clusters se regroupent encore à un niveau supérieur pour former des **macro-clusters**. Cette section (1.4.3) approfondit la *définition mathématique* de ces concepts, leurs *règles d'apparition* et leurs *conséquences* sur la dynamique d'apprentissage dans le **Deep Synergy Learning (DSL)**.

Dans la perspective du **Synergistic Connection Network (SCN)**, il est souvent utile de modéliser la structure courante du réseau par un **graphe** $G(t)$ dont les **sommets** $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ correspondent aux entités d'information, et dont chaque **arête** est dotée d'une **pondération synergique** $\omega_{i,j}(t)$. À l'instant t , ces pondérations se réunissent dans une **matrice** :

$$W(t) = [\omega_{i,j}(t)]_{1 \leq i, j \leq n}.$$

Cette représentation matricielle capture simultanément l'ensemble des liens du réseau, chaque $\omega_{i,j}(t)$ traduisant l'intensité de la coopération entre l'entité \mathcal{E}_i et l'entité \mathcal{E}_j . Dans un tel graphe, un **cluster** $\mathcal{C} \subset \{1, \dots, n\}$ (c'est-à-dire un sous-ensemble des indices d'entités) se caractérise par des **connexions internes** plus **denses** ou plus **fortes** que ses connexions avec l'extérieur. Plusieurs manières existent pour formaliser cette notion de cluster :

Un **premier** point de vue consiste à **maximiser la somme** des pondérations internes. Plus précisément, on cherche un sous-ensemble \mathcal{C} maximisant

$$\sum_{i \in \mathcal{C}, j \in \mathcal{C}} \omega_{i,j}(t),$$

ce qui revient à rechercher la zone du graphe possédant la **densité** la plus élevée en liaisons fortes. Cette démarche peut être utile, par exemple, lorsque l'on souhaite mettre en évidence un noyau collaboratif de grande intensité.

Un **deuxième** critère, plus nuancé, repose sur le **rapport** entre la force interne et la force externe. On introduit alors une fonction de ratio

$$R(\mathcal{C}) = \frac{\sum_{i,j \in \mathcal{C}} \omega_{i,j}(t)}{\sum_{i \in \mathcal{C}, j \notin \mathcal{C}} \omega_{i,j}(t) + \epsilon'}$$

où $\epsilon > 0$ agit comme un **terme de régularisation** évitant les divisions par zéro. Le cluster \mathcal{C} optimal maximise alors $R(\mathcal{C})$, ce qui revient à privilégier des groupes dont la **cohésion interne** se révèle importante face aux connexions dirigées vers l'extérieur.

Enfin, un **troisième** point de vue fait appel à une **fonction d'énergie** \mathcal{J} , ou fonction de coût, qui **récompense** la densité intra-cluster tout en **pénalisant** les connexions externes (cf. section 1.4.2 pour une présentation plus générale du concept de coût dans le DSL). Les **clusters** apparaissent alors comme des **minima locaux** de \mathcal{J} . Cette perspective offre un cadre théorique aisément connectable à la physique statistique ou aux techniques d'optimisation combinatoire, permettant d'étudier la **stabilité** ou le **caractère** global/local des solutions.

Malgré la diversité de ces approches, le principe fondamental reste le même. Un **cluster** est un **groupe d'entités** dont les liaisons, au sein de la matrice $W(t)$, sont relativement plus fortes entre ses membres qu'avec le reste du réseau. On retrouve ainsi l'idée intuitive d'une **collaboration** accrue à l'intérieur du sous-ensemble, associée à un **isolement** ou à une **définition** face aux entités extérieures. Dans la suite du texte, ces définitions de clusters s'avèrent essentielles pour analyser la manière dont le **SCN** laisse émerger automatiquement, au fil de la dynamique d'apprentissage, des **strates** coopératives plus ou moins vastes.

Dans le cadre du **Deep Synergy Learning**, les pondérations $\omega_{i,j}(t)$ évoluent fréquemment selon la règle

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ et $\tau > 0$ sont des constantes, tandis que $S(\mathcal{E}_i, \mathcal{E}_j)$ désigne la **synergie** entre les entités i et j . Dès lors qu'une pondération $\omega_{i,j}(t)$ dépasse un seuil ω_{\min} , on considère qu'il existe un **lien effectif** entre les deux entités. Plusieurs phénomènes concourent alors à la **formation d'un cluster** :

- **Renforcement interne.** Lorsqu'un certain **groupe** $\mathcal{C} \subset \{1, \dots, n\}$ présente des synergies élevées entre ses membres, les pondérations $\omega_{i,j}(t)$ associées ont tendance à **augmenter** pour tous $i, j \in \mathcal{C}$.
- **Isolement progressif.** À mesure que les liens internes se consolident, les connexions externes (i.e. celles dont la synergie avec les entités extérieures demeure plus faible) **s'atténuent**, rendant le sous-ensemble \mathcal{C} de plus en plus autonome et homogène.

Ce processus engendre un **cluster** cohérent, formé sans qu'aucun agencement préalable ne soit imposé.

Exemple de scénario d'émergence

Temps initial (t=0). Les pondérations $\omega_{i,j}(0)$ sont faibles ou distribuées de manière uniforme, si bien qu'aucune structure évidente ne se détache.

Interactions locales. Certaines entités détectent une synergie ($S(\mathcal{E}_i, \mathcal{E}_j) > 0$) suffisamment avantageuse pour se renforcer localement. Les $\omega_{i,j}(t)$ correspondantes s'amplifient.

Formation d'une "graine". Progressivement, un **mini-cluster** naît sous la forme de 2 ou 3 entités dont les liens internes, devenus plus élevés, assurent un noyau de haute synergie.

Attraction. Si une entité \mathcal{E}_k à la périphérie présente aussi une bonne synergie avec cette graine, ses liaisons $\omega_{k,i}$ ($i \in \mathcal{C}$) se renforcent, ce qui intègre \mathcal{E}_k au cluster en expansion.

Consolidation. Une fois que les **liens internes** atteignent un niveau stable et que les connexions vers l’extérieur restent faibles, le groupe \mathcal{C} se **stabilise** comme cluster. D’un point de vue formel, cela correspond à un **minimum local** d’une fonction d’énergie \mathcal{J} , où \mathcal{C} se comporte comme un **attracteur**.

Au-delà des **clusters** de base, il n’est pas rare que plusieurs sous-groupes finis se **rassemblent** pour former un **macro-cluster**, c’est-à-dire une entité supérieure englobant plusieurs clusters déjà constitués.

Pour cela, on considère un **méta-groupe** \mathcal{M} rassemblant des **clusters** $\mathcal{C}_1, \mathcal{C}_2, \dots$. Les pondérations entre deux clusters \mathcal{C}_1 et \mathcal{C}_2 se définissent par

$$\Omega(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{|\mathcal{C}_1| \cdot |\mathcal{C}_2|} \sum_{\substack{i \in \mathcal{C}_1 \\ j \in \mathcal{C}_2}} \omega_{i,j}(t).$$

Si $\Omega(\mathcal{C}_1, \mathcal{C}_2)$ dépasse un certain niveau, les deux groupes **fusionnent**, produisant ainsi un **macro-cluster** élargi.

Le **processus de fusion** des clusters suit une dynamique progressive où les liaisons synergiques évoluent jusqu’à structurer des **macro-clusters** plus larges.

Les **clusters initiaux**, notés \mathcal{C}_1 et \mathcal{C}_2 , existent indépendamment dans le réseau. Lorsque plusieurs connexions $\omega_{i,j}$, avec $i \in \mathcal{C}_1$ et $j \in \mathcal{C}_2$, se renforcent progressivement, la synergie globale entre ces deux groupes augmente. Ce **renforcement inter-clusters** favorise leur rapprochement et amplifie leur interdépendance.

Dès que ce renforcement atteint un **seuil critique**, la distinction entre \mathcal{C}_1 et \mathcal{C}_2 disparaît. Ils cessent alors d’être des sous-groupes autonomes et forment un **macro-cluster** \mathcal{M} , intégrant leurs dynamiques respectives.

Ce mécanisme s’apparente au **clustering hiérarchique agglomératif**, à la différence que dans le **DSL**, les pondérations $\omega_{i,j}$ ne sont pas statiques. Elles s’ajustent en permanence, permettant une **consolidation dynamique** qui s’adapte aux évolutions du réseau et aux changements des interactions synergiques.

Pour formaliser ces phénomènes de clusterisation, on introduit couramment une **fonction d’énergie** \mathcal{J} :

$$\mathcal{J}(\{\omega_{i,j}\}) = - \sum_{(i,j)} \omega_{i,j} S(\mathcal{E}_i, \mathcal{E}_j) + \alpha \sum_{(i,j)} (\omega_{i,j})^2 + \dots$$

Le premier terme **récompense** les liaisons présentant une **haute synergie**. Minimiser \mathcal{J} revient ainsi à **maximiser** la somme $\sum \omega_{i,j} S(\mathcal{E}_i, \mathcal{E}_j)$, favorisant le renforcement des connexions les plus pertinentes.

Le second terme, avec $\alpha > 0$, introduit une **régularisation** ou un **terme de pénalité**, limitant une croissance excessive et simultanée de toutes les connexions. Cette contrainte prévient une surdensification du réseau et maintient un équilibre dans la distribution des pondérations.

Dans ce cadre, un **cluster** renvoie à un **minimum local** de \mathcal{J} , où l'on observe un niveau élevé de pondérations **en interne**. Un **macro-cluster** surgit quand on identifie un autre minimum local, rassemblant plusieurs groupes préexistants.

L'émergence de **clusters** segmente le Synergistic Connection Network en **régions** spécialisées (p. ex. un cluster associant vision et texte, un autre combinant audio et capteurs, etc.). On obtient ainsi une forme de **division du travail** au sein d'un réseau de grande taille, ce qui favorise la **lisibilité** et la **résilience** globale.

Par ailleurs, la formation de **macro-clusters** peut être rapprochée de l'idée de **modules cognitifs**, dans lesquels plusieurs sous-ensembles d'entités se lient pour s'attaquer à des tâches plus complexes. Le réseau acquiert de la sorte une **modularité adaptative**, qu'il s'agisse de manipuler différents canaux de données ou d'élaborer des représentations de plus haut niveau.

D'un point de vue **algorithmique**, l'identification et la gestion explicite des **clusters** permettent d'optimiser la **mise à jour** des pondérations. Plutôt que d'analyser l'ensemble du graphe simultanément, il devient possible de limiter certains calculs aux **modules faiblement connectés**, réduisant ainsi la complexité computationnelle tout en préservant la dynamique adaptative du réseau. Cette **auto-organisation** en groupes soudés se démarque d'un simple "clustering statique", car les pondérations $\omega_{i,j}(t)$ évoluent continuellement, autorisant des scissions, des fusions et même la disparition de clusters entiers au gré des fluctuations de la synergie.

Exemples d'application

Les **applications du SCN** exploitent sa capacité d'auto-organisation et d'adaptation dynamique pour structurer les interactions entre entités en fonction de la synergie détectée.

Dans l'**analyse multimodale en temps réel**, un système de surveillance peut regrouper en **clusters dynamiques** les entités traitant l'image et la parole, formant par exemple un cluster "visage + discours". Un autre cluster peut émerger pour regrouper les entités chargées de détecter les mouvements anormaux ou les sons inhabituels. Ces agrégats évoluent continuellement en fonction de la **synergie perçue** entre différents flux, tels que la lumière, le son et les déplacements.

Dans la **recommandation et le filtrage collaboratif**, les utilisateurs et les contenus sont représentés par des entités connectées par des pondérations $\omega_{u,c}$. Lorsqu'un utilisateur u apprécie un contenu c , leur connexion se renforce, favorisant l'émergence de **clusters** d'utilisateurs aux préférences similaires. Avec le temps, un **macro-cluster** peut apparaître, regroupant un ensemble d'utilisateurs et les contenus correspondant collectivement à leurs intérêts.

Dans le **traitement biologique ou neuroscientifique**, les neurones ou de petites populations neuronales peuvent être représentés comme des entités \mathcal{E}_i . L'auto-organisation du SCN conduit alors à la formation de **clusters neuronaux locaux**, puis à l'émergence de **macro-clusters** reliant plusieurs aires cérébrales interconnectées. Ce phénomène reflète une **organisation fonctionnelle** à plus grande échelle, traduisant la dynamique sous-jacente des interactions neuronales.

1.4.4. Fonctions de Synergie : Distance, Similarité et Co-Information

Au cœur du **Deep Synergy Learning (DSL)**, les *pondérations synergiques* $\omega_{i,j}$ reliant deux entités \mathcal{E}_i et \mathcal{E}_j évoluent en fonction d'une **mesure de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. Cette fonction S évalue dans quelle mesure la **coopération** entre deux entités apporte un **gain** supérieur à l'utilisation isolée de leurs informations.

Bien que le **principe** de la synergie demeure le même (capturer l'apport mutuel), il existe **plusieurs manières** de la **définir** et de la **calculer** dans la pratique. Les plus communes reposent sur (1) des **distances** entre représentations, (2) des **similarités** (souvent normalisées), ou (3) des mesures **d'information** (entropie, co-information, etc.). Cette section expose en détail ces approches, avec une formulation mathématique approfondie.

Dans le **Deep Synergy Learning (DSL)**, la pondération $\omega_{i,j}(t)$ reliant deux entités \mathcal{E}_i et \mathcal{E}_j suit fréquemment une équation de la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η et τ sont des hyperparamètres positifs, et $S(\mathcal{E}_i, \mathcal{E}_j)$ la **fonction de synergie**. Cette fonction reflète dans quelle mesure les entités \mathcal{E}_i et \mathcal{E}_j estiment pouvoir coopérer avantageusement. Dès lors,

- Si $S(\mathcal{E}_i, \mathcal{E}_j)$ est **positif** et assez grand, la liaison $\omega_{i,j}$ se **renforce**.
- Si $S(\mathcal{E}_i, \mathcal{E}_j)$ est faible ou négatif, la liaison s'**affaiblit**.

Le **choix** de la fonction S influe directement sur la **topologie** finale du réseau. Deux entités jugées proches ou complémentaires se retrouveront dans un même **cluster** (voir section 1.4.3), tandis que des entités peu compatibles verront leur lien s'étioler. Différentes approches existent pour définir cette synergie.

1. Synergie fondée sur la distance

Une façon intuitive d'instaurer la synergie est de partir d'une **distance** $d(\mathcal{E}_i, \mathcal{E}_j)$ calculée sur leurs représentations, souvent des vecteurs $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$. Un exemple répandu consiste en la **distance euclidienne** $\|\mathbf{x}_i - \mathbf{x}_j\|$. On convertit ensuite cette distance en un score de synergie **décroissant** :

$$S(\mathcal{E}_i, \mathcal{E}_j) = \frac{1}{1 + \|\mathbf{x}_i - \mathbf{x}_j\|^2},$$

ou bien, en autorisant des valeurs négatives, on peut poser

$$S(\mathcal{E}_i, \mathcal{E}_j) = -\|\mathbf{x}_i - \mathbf{x}_j\|.$$

L'idée est qu'une faible distance se traduit par une **forte synergie**, tandis qu'une distance importante produit un score négatif ou tendant vers zéro. Il est fréquent d'ajouter un **paramètre d'échelle** $\sigma > 0$ pour réguler la sensibilité aux écarts, par exemple

$$S(\mathcal{E}_i, \mathcal{E}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2).$$

Cette fonction valorise la synergie pour les entités **très proches** et la fait décroître rapidement au-delà d'un rayon σ . La distance choisie dépend de la nature des données. Elle peut être de type **L1**, **L2**, ou encore définie sur des **tenseurs d'images** ou des **spectrogrammes**, selon le contexte d'application.

2. Synergie fondée sur la similarité

Un autre choix repose sur une **similarité** plutôt que sur une distance. Par exemple, la **similarité cosinus** :

$$\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\langle \mathbf{x}_i, \mathbf{x}_j \rangle}{\| \mathbf{x}_i \| \| \mathbf{x}_j \|},$$

qui varie dans $[-1, 1]$. On peut alors définir

$$S(\mathcal{E}_i, \mathcal{E}_j) = \max(0, \text{sim}(\mathbf{x}_i, \mathbf{x}_j)) \quad \text{ou} \quad S(\mathcal{E}_i, \mathcal{E}_j) = 1/2 (\text{sim}(\mathbf{x}_i, \mathbf{x}_j) + 1),$$

histoire de normaliser la valeur dans $[0, 1]$. Cette approche est pratique quand on souhaite regrouper des **vecteurs** se ressemblant en direction ou quand on manipule des **corrélations** (ex. coefficient de Pearson). Elle est parfois moins adaptée si l'on cherche à capter la complémentarité non linéaire entre des entités fortement dissemblables.

3. Synergie fondée sur la co-information ou l'information mutuelle

Lorsque les entités $\mathcal{E}_i, \mathcal{E}_j$ renvoient à des **variables aléatoires** $\mathbf{X}_i, \mathbf{X}_j$, on peut évaluer la synergie via des **quantités entropiques** :

$$I(\mathbf{X}_i; \mathbf{X}_j) = H(\mathbf{X}_i) + H(\mathbf{X}_j) - H(\mathbf{X}_i, \mathbf{X}_j),$$

qui mesure l'**information mutuelle**. Toutefois, cette mesure ne différencie pas la simple **redondance** d'une véritable **coopération**. On peut alors recourir à la **co-information** ou à la **Partial Information Decomposition (PID)**, laquelle décompose l'information partagée en portions "synergique" et "redondante". Cela donne un score :

$$S(\mathcal{E}_i, \mathcal{E}_j) = \max(0, I_{\text{syn}}(\mathbf{X}_i, \mathbf{X}_j | \mathbf{Y})),$$

où I_{syn} désigne la contribution strictement synergique, au sens de la définition employée (co-info, PID, etc.). Cette méthode peut détecter des interactions non linéaires et complexes, mais elle se révèle plus **onéreuse** à calculer, en particulier pour des données de haute dimension, et nécessite de **choisir** la mesure d'information la plus pertinente au contexte.

Conseils pour le choix de la fonction S

Le **DSL** n'impose pas une unique forme de synergie, mais propose un **cadre flexible** où toute mesure d'**apport mutuel** entre entités peut être adaptée. Certains principes guident néanmoins le choix de la fonction S .

La **distance** est une option simple à implémenter et intuitive. Elle favorise l'association d'entités similaires dans un certain espace de représentation, mais peut sous-estimer la complémentarité

entre des entités très différentes mais hautement coopératives. La **similarité** est plus pertinente lorsque l'on compare des directions ou des vecteurs normalisés. Une forte similarité indique une parenté marquée, bien que certaines complémentarités non linéaires puissent lui échapper.

L'**information mutuelle**, la **co-information** ou encore le **PID** permettent de capturer des **coopérations complexes et non linéaires**. Ces approches sont puissantes mais présentent un coût computationnel élevé et restent sensibles aux méthodes d'estimation statistique.

Il est évidemment possible de **combiner** plusieurs indicateurs (distance, similarité, information mutuelle) ou de pondérer différentes composantes pour confectionner un score $S(\cdot, \cdot)$ plus adapté. Le choix exact dépendra de la nature des données, du domaine d'application et des **objectifs** poursuivis (clustering, prédiction, fusion multimodale, etc.).

Exemples d'utilisation

Analyse d'images. Considérons des entités \mathcal{E}_i représentant des **patches** ou des **régions d'images**. Une distance **L2** appliquée aux vecteurs de caractéristiques, extraits par exemple d'un **réseau de neurones convolutionnel**, peut fournir un score de **synergie décroissant**. Plus les patchs présentent une forte ressemblance, plus leur coopération est renforcée.

Fusion multimodale. Supposons deux entités $\mathbf{X}_{\text{audio}}$ et $\mathbf{X}_{\text{visuel}}$. On mesure leur co-information conditionnellement à une variable \mathbf{Y} (classe d'événement). Si leur combinaison apporte un gain clair, on augmente la synergie et, par conséquent, on renforce leur lien.

Système hybride symbolique-connexionniste. Un module symbolique (décrivant des règles ou faits abstraits) et un module vectoriel (des embeddings d'images, par exemple) peuvent relier leurs entités si l'information mutuelle ou la similarité cosinus démontre qu'ils s'améliorent réciproquement.

Conclusion et ouverture

La fonction de synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ est la **pierre angulaire** du Deep Synergy Learning. Selon qu'on s'appuie sur une **distance**, une **similarité** ou une **information mutuelle**, on oriente la manière dont les entités **coopèrent** et se **relient** au sein du **Synergistic Connection Network**.

- Les **fonctions de distance** favorisent l'agrégation d'éléments proches dans l'espace des caractéristiques.
- Les **fonctions de similarité** rassemblent des vecteurs corrélés, tout en ignorant parfois la complémentarité.
- Les **mesures entropiques** (co-info, PID, etc.) explorent la coopération profonde, prenant en compte les facettes non linéaires et la contribution uniquement révélée par l'association des entités.

Les sections suivantes (1.4.5 à 1.4.7) approfondiront la **mise à jour temporelle** de $\omega_{i,j}(t)$ fondée sur cette synergie, la distinction entre **interactions directes** et **indirectes**, et la manière de gérer la **synergie n-aire**. L'ensemble de ces points complète la **vision** du DSL comme un réseau évolutif, autonome et apte à faire émerger des **structures** d'apprentissage complexe dans des contextes variés.

1.4.5. Pondérations Adaptatives et Évolution Temporelle

Les sections précédentes (**1.4.1 à 1.4.4**) ont posé les **fondements** de l'architecture générale du **Deep Synergy Learning (DSL)**. Chaque **entité** peut établir des connexions avec d'autres à travers des **pondérations synergiques** $\omega_{i,j}$, qui traduisent la **valeur ajoutée** de leur coopération. Reste à décrire **comment** ces pondérations **changent** dans le temps, c'est-à-dire la **loi d'évolution** qui fait que le réseau se **reconfigure** en permanence. C'est l'objet de cette sous-section (1.4.5), qui explique :

- Les **équations** gouvernant l'adaptation des pondérations,
- Le sens de cette **dynamique** (renforcement / affaiblissement),
- Les implications sur la **convergence** ou la **stabilité** des liens,
- Le rôle des **paramètres** (taux d'apprentissage, régularisation, etc.) dans l'auto-organisation du réseau.

Cette **évolution temporelle** est au cœur du **SCN** (Synergistic Connection Network), car c'est elle qui autorise la création progressive de **clusters** et, par extension, la modélisation d'un **apprentissage** réellement continu.

Dans la plupart des formulations du **Deep Synergy Learning (DSL)**, la pondération $\omega_{i,j}(t)$ reliant deux entités \mathcal{E}_i et \mathcal{E}_j se met à jour suivant un schéma adaptatif de la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η désigne un **taux d'apprentissage** (ou **pas de mise à jour**) et τ un **coefficient de régulation** (ou *terme d'oubli*). La fonction $S(\mathcal{E}_i, \mathcal{E}_j)$ correspond à la mesure de **synergie** choisie (distance, similarité, co-information, etc.). Ce cadre de mise à jour s'interprète de la manière suivante.

Lorsque la synergie S entre \mathcal{E}_i et \mathcal{E}_j est suffisamment élevée et que $\omega_{i,j}(t)$ reste encore modérée, le terme $\eta [S - \tau \omega]$ demeure **positif**, entraînant un **renforcement** progressif de la liaison à chaque itération.

À l'inverse, si la synergie est faible ou négative, ou si $\omega_{i,j}(t)$ atteint une valeur déjà trop élevée, ce terme devient **négatif**, ce qui conduit à un **affaiblissement** progressif de la liaison.

On peut mettre en évidence un **point fixe** en imposant $\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega_{i,j}^*$. La condition associée,

$$\omega_{i,j}^* = S/\tau,$$

montre que plus la **synergie** est grande, plus la **pondération d'équilibre** $\omega_{i,j}^*$ atteint un niveau élevé.

Variantes et enrichissements

Plusieurs adaptations sont possibles autour de cette règle. On peut retenir un **historique** afin de lisser la dynamique ou pondérer les valeurs antérieures $\omega_{i,j}(t-1)$. La mise à jour des pondérations peut adopter une **forme non linéaire**, s'écartant du modèle linéaire $\omega + \eta [S - \tau \omega]$ pour inclure des équations **saturantes, exponentielles** ou inspirées des **règles hebbiennes**.

De plus, la synergie S peut évoluer dans le temps si les entités $\mathcal{E}_i(t)$ et $\mathcal{E}_j(t)$ mettent à jour leurs représentations internes. Dans ce cas, la pondération $\omega_{i,j}(t)$ n'influence pas seulement la connexion entre entités, mais modifie également la **synergie perçue**, induisant une transformation dynamique de la **structure globale** du réseau.

Système dynamique discret ou continu

Le modèle

$$\omega_{i,j}(t+1) - \omega_{i,j}(t) = \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)]$$

est une **équation aux différences**, interprétable comme un **système dynamique** discret dans un espace de dimension $\binom{n}{2}$ (ou $n(n-1)$ selon que l'on distingue la direction $i \rightarrow j$). On y analyse notamment :

- Les **points fixes** $\omega_{i,j}^*$ satisfaisant

$$\omega_{i,j}^* = \omega_{i,j}^* + \eta [S - \tau \omega_{i,j}^*].$$

- La **stabilité** de ces points. Par exemple, en linéarisant autour de l'équilibre, on obtient

$$\Delta\omega_{i,j}(t+1) = \Delta\omega_{i,j}(t) - \eta \tau \Delta\omega_{i,j}(t),$$

montrant que la condition $\eta \tau < 1$ est usuellement requise pour maintenir une évolution stable des pondérations.

L'équivalent **en temps continu** est donné par l'ODE

$$d\omega_{i,j}/dt = \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Ici encore, la solution tend vers $\omega_{i,j}^* = S/\tau$ s'il ne survient pas d'autres perturbations ni dépendances supplémentaires.

Seuils, cap et parsimonie

Dans le **DSL**, on introduit souvent un **seuil** ω_{\min} en deçà duquel la connexion est jugée inexisteante :

$$\omega_{i,j}(t) < \omega_{\min} \Rightarrow \text{aucun lien effectif entre } i \text{ et } j.$$

Même si l'équilibre $\omega_{i,j}^*$ est légèrement positif, la liaison doit dépasser ω_{\min} pour être considérée. À l'autre extrémité, on peut imposer un **cap** $\omega_{i,j}(t) \leq \omega_{\max}$ pour freiner l'augmentation des pondérations et éviter les valeurs extrêmes. Sur un **diagramme de phase** ($\omega - \dot{\omega}$), ces deux seuils

assurent une **parcimonie** structurale. Seuls les liens ayant démontré une **synergie conséquente** sont préservés, tandis que ceux restant trop faibles ou atteignant des valeurs excessives sont soit **coupés**, soit **plafonnés** pour éviter une sur-saturation du réseau.

Interdépendance globale

Dans un réseau comptant n entités, la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ peut dépendre d'autres entités (cf. section 1.4.4). On écrit parfois

$$S(\mathcal{E}_i, \mathcal{E}_j \mid \{\mathcal{E}_k\}_{k \neq i,j}),$$

pour souligner le fait qu'une **co-information** conditionnelle peut influer sur la pondération. Les **mises à jour** des pondérations $\omega_{i,j}$ sont **couplées**, ce qui signifie que la liaison $i \leftrightarrow j$ évolue en interaction avec d'autres connexions comme $j \leftrightarrow k$, influençant ainsi la dynamique globale du réseau. Ce couplage favorise l'**émergence de clusters** (section 1.4.3). Les entités d'un sous-groupe \mathcal{C} stabilisent progressivement leurs pondérations internes, formant une **bulle de coopération élevée**, tandis que les connexions extérieures, moins profitables, s'atténuent naturellement.

Feedback et plasticité

Interpréter la dynamique du **DSL** comme un **système à feedback** apporte un éclairage pertinent sur son mode d'auto-organisation.

Le **feedback positif** intervient lorsqu'une liaison $\omega_{i,j}$ se renforce sous l'effet d'une synergie positive ($S > 0$). Dans ce cas, \mathcal{E}_i et \mathcal{E}_j augmentent leur coopération, consolidant ainsi encore davantage leur connexion.

À l'inverse, le **feedback négatif** empêche les liaisons de croître indéfiniment. Si une connexion s'emballe, le terme $-\tau \omega_{i,j}(t)$ ou une autre forme de régulation, comme une pénalisation non linéaire, freine son amplification, garantissant ainsi la **stabilisation** du réseau.

De façon analogique, on peut s'inspirer de la **plasticité synaptique** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta \left(a_i(t) a_j(t) \right) - \alpha \omega_{i,j}(t),$$

où $a_i(t)$ et $a_j(t)$ sont les "activités" ou "états" des entités, vus comme un substitut instantané à la synergie $S_{i,j}(t)$. Par ailleurs, l'usage d'un **terme cubique** $-\beta (\omega_{i,j})^3$ agit comme une stabilisation encore plus marquée, évitant la croissance sans borne.

Impacts et conclusions

Flexibilité continue. Grâce à l'actualisation en continu de $\omega_{i,j}(t)$, le **Synergistic Connection Network** (SCN) s'adapte sans cesse aux nouveaux flux ou aux modifications d'environnement, bien plus qu'un réseau à architecture fixe.

- **Formation de clusters.** Les entités liées par une synergie forte consolident leurs liaisons, s'assemblent en **clusters**, et peuvent se **fusionner** en macro-clusters (voir section 1.4.3). Cette recomposition est fluide et dépend directement de la dynamique des pondérations.

- **Régime stable vs. oscillatoire.** Selon les choix de synergie, de η , de τ , etc., le réseau peut tendre vers un état stable, osciller, ou coexister dans plusieurs configurations d'équilibre.
- **Aspects computationnels.** Mettre à jour toutes les paires (i, j) exige un coût a priori en $O(n^2)$. Des heuristiques (sparse updates, random sampling) peuvent s'avérer nécessaires pour gérer de grands n de façon **scalable**.

Cette **dynamique adaptative** des pondérations constitue le moteur de l'**auto-organisation** dans le DSL, permettant de découvrir et de consolider des **sous-structures** au sein du réseau. Les sections suivantes (1.4.6 et 1.4.7) s'intéresseront à la **distinction entre interactions directes et indirectes** et à la **synergie n-aire**, deux aspects cruciaux pour comprendre la **richesse** et la **réactivité** de ce paradigme d'apprentissage.

1.4.6. Interactions Directes et Indirectes

Jusqu'à présent, la description du **Deep Synergy Learning (DSL)** s'est essentiellement focalisée sur la **relation directe** entre deux entités \mathcal{E}_i et \mathcal{E}_j , mesurée par une **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ et traduite dans une **pondération** $\omega_{i,j}$. Dans un réseau où de nombreuses entités coexistent, les **interactions indirectes** sont essentielles. Même sans liaison directe, deux entités peuvent s'influencer si elles sont **connectées** par une **chaîne** de liens synergiques. Ce mécanisme favorise la propagation des influences et la structuration dynamique du réseau. Cette section (1.4.6) explique :

- Comment les **interactions** peuvent se propager à travers le **Synergistic Connection Network (SCN)**,
- Comment la **synergie** entre deux entités peut être **modulée** par leurs relations avec d'autres,
- Quelles implications mathématiques découlent de ces boucles d'influence plus complexes, notamment pour la **formation** de clusters et l'**émergence** de comportements globaux.

Cette problématique est essentielle pour comprendre l'**auto-organisation**. Même si chaque entité n'agit qu'en fonction de ses liens directs, l'ensemble du réseau peut révéler des effets d'**influence à distance** ou de **coordination** en chaîne, façonnant ainsi sa dynamique globale.

On appelle **interaction indirecte** entre \mathcal{E}_i et \mathcal{E}_j toute séquence $\mathcal{E}_i \rightarrow \mathcal{E}_k \rightarrow \dots \rightarrow \mathcal{E}_m \rightarrow \mathcal{E}_j$ dans laquelle chacune des paires $(\mathcal{E}_u, \mathcal{E}_v)$ du chemin possède une pondération $\omega_{u,v}(t)$ significative. Le **Synergistic Connection Network (SCN)** autorise ainsi la propagation d'**influence** ou de **coopération** le long d'un **chemin** reliant deux entités, même si ces dernières ne sont pas directement connectées.

A. Principes généraux : chemins et relai d'information

Lorsque la liaison $\omega_{k,m}$ est élevée, l'entité \mathcal{E}_k peut **transmettre** (ou "relayer") certaines données ou incitations à \mathcal{E}_m . C'est ainsi qu'une synergie faible ou nulle entre \mathcal{E}_i et \mathcal{E}_j peut progressivement **s'intensifier** si elles interagissent via des entités communes $\{\mathcal{E}_k, \mathcal{E}_\ell, \dots\}$ qui procurent un **gain mutuel**. De même, si l'on modélise la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ comme une quantité **contextuelle**, on peut définir

$$S(\mathcal{E}_i, \mathcal{E}_j \mid \mathcal{E}_k),$$

qui tient compte du fait que la coopération entre \mathcal{E}_i et \mathcal{E}_j peut être **facilitée** ou **rendue moins nécessaire** par la présence d'une entité pivot \mathcal{E}_k .

Exemple : pivot et redondance

- Si \mathcal{E}_k remplit déjà les rôles combinés de \mathcal{E}_i et \mathcal{E}_j , l'intérêt pour \mathcal{E}_i et \mathcal{E}_j de se lier **directement** diminue (redondance).
- Au contraire, si \mathcal{E}_k fournit un complément utile à \mathcal{E}_i et \mathcal{E}_j , ce **contexte** peut augmenter leur synergie bilatérale, incitant l'émergence d'un lien direct $\omega_{i,j}$.

B. Notion de chemins et puissance de la matrice W

Le **graphe** du SCN se décrit par la matrice pondérée $W(t)$, de taille $n \times n$, dont l'entrée $\omega_{i,j}(t)$ indique la connexion directe entre \mathcal{E}_i et \mathcal{E}_j . Lorsqu'on considère un **chemin** de longueur 2 entre \mathcal{E}_i et \mathcal{E}_j via un intermédiaire \mathcal{E}_k , la multiplication de matrices révèle que $(W^2)_{i,j} = \sum_{k=1}^n \omega_{i,k}(t) \omega_{k,j}(t)$. De manière analogue, $(W^p)_{i,j}$ réunit l'effet de **tous** les chemins de longueur p liant i à j .

Si la dynamique des pondérations reste **relativement stable**, l'examen des puissances W^2, W^3, \dots peut dévoiler des **interactions indirectes** (chemins multiples ou plus longs). Des valeurs élevées de $(W^p)_{i,j}$ traduisent une **forte influence** de \mathcal{E}_i sur \mathcal{E}_j à travers plusieurs intermédiaires. En présence de **cycles** (par exemple, $\mathcal{E}_i \rightarrow \mathcal{E}_k \rightarrow \mathcal{E}_m \rightarrow \mathcal{E}_i$), des effets d'**amplification** peuvent apparaître. Si chaque liaison du cycle se renforce, la synergie circule en boucle, entraînant une augmentation progressive de son intensité au sein du réseau.

D'un point de vue **linéaire**, la **valeur propre** de plus grande amplitude, correspondant au **rayon spectral** $\rho(W)$, détermine la **stabilité** du système.

Si $\rho(W) > 1$, de petites perturbations internes peuvent se propager et s'amplifier, entraînant un risque d'**oscillations** ou d'emballement du réseau.

Si $\rho(W) < 1$ après régulation, la dynamique tend à converger vers un **état stable**, maintenant ainsi l'équilibre du système.

C. Contexte et synergie conditionnelle

Dans un réseau synergique, l'existence de **chemins indirects** modifie considérablement la manière dont deux entités $\mathcal{E}_i, \mathcal{E}_j$ perçoivent leur **coopération**. Même si $\omega_{i,j}$ démarre faible, il peut se consolider via l'influence d'entités tierces \mathcal{E}_k . Formellement, on modélise cette dépendance par une fonction

$$S(\mathcal{E}_i, \mathcal{E}_j | \{\omega_{i,k}, \omega_{k,j}, \dots\}),$$

introduisant un **système dynamique non linéaire**. Les équations des pondérations deviennent

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j | \{\omega_{i,k}, \omega_{k,j}, \dots\}) - \tau \omega_{i,j}(t)].$$

La dépendance entre les $\omega_{i,j}$ fait naître des **boucles de feedback**, où un renforcement dans un segment du réseau rejaillit sur d'autres liaisons, et ainsi de suite. En pratique, l'**analyse** de stabilité ou de convergence peut impliquer des techniques avancées (théorie des bifurcations, méthodes de Lyapunov, etc.).

D. Rôle des interactions indirectes dans la dynamique globale

Les **interactions indirectes** permettent à un réseau **DSL** d'évoluer de manière dynamique à travers plusieurs mécanismes.

Elles facilitent l'**émergence de nouveaux liens**. Si deux entités \mathcal{E}_i et \mathcal{E}_j constatent qu'elles coopèrent efficacement via un pivot \mathcal{E}_k , elles peuvent établir une connexion directe $\omega_{i,j} > 0$, optimisant ainsi leurs échanges.

Elles favorisent la **fusion des clusters**. Deux groupes initialement distincts, reliés par des chemins transitifs, peuvent progressivement reconnaître leur complémentarité et **coalescer** en un **macro-cluster**, renforçant la structure globale du réseau.

Elles influencent également la **gestion du bruit**. Une entité bruyante ou imprécise peut affecter des voisins même en l'absence de connexion directe, tandis qu'un intermédiaire de confiance peut stabiliser et améliorer la fiabilité des échanges entre plusieurs entités.

Ce mécanisme d'**influence à distance** se retrouve dans des domaines variés. En **robotique multi-agents**, un agent \mathcal{E}_i peut influencer un autre \mathcal{E}_j via un agent \mathcal{E}_k jouant le rôle de relai ; la structure d'échange se reconfigure selon la performance observée. En **neurosciences**, de nouvelles synapses directes se forment après une longue période où l'activité neuronale transite par plusieurs synapses intermédiaires.

Conclusion et ouverture

Les **interactions indirectes** forment un pivot crucial pour saisir l'**auto-organisation** d'un Synergistic Connection Network (SCN). Deux entités \mathcal{E}_i et \mathcal{E}_j initialement faiblement connectées peuvent se **rapprocher** grâce aux chemins multiples les reliant via des tiers, tout comme des boucles internes peuvent amplifier ou réguler la synergie. Cette dimension de **propagation** et de **contexte** est essentielle pour comprendre la **formation** de macro-structures, la dynamique des **cycles** et la **coévolution** des pondérations dans le DSL.

La section suivante (1.4.7) examinera de plus près la **synergie binaire** et surtout la **synergie n-aire**, c'est-à-dire l'impact de coopérations impliquant simultanément plus de deux entités, un aspect qui renforce encore la **richesse** des comportements émergents dans le Deep Synergy Learning.

1.4.7. Synergie binaire et n-aire : au-delà des relations deux à deux

La **synergie binaire** demeure la forme la plus couramment utilisée pour évaluer le degré de coopération entre deux entités \mathcal{E}_i et \mathcal{E}_j . On définit pour cela une fonction $S(\mathcal{E}_i, \mathcal{E}_j)$ (pouvant être fondée sur une distance, une similarité, de la co-information, etc.) qui mesure dans quelle mesure

$$\mathbf{x}_i (+ \mathbf{s}_i, \theta_i) \Rightarrow \mathbf{x}_j (+ \mathbf{s}_j, \theta_j)$$

s'enrichissent mutuellement ou, au contraire, s'avèrent redondantes ou peu utiles l'une à l'autre.

- **Mise à jour.** À chaque itération, la pondération $\omega_{i,j}(t)$ se voit recalculée en fonction de la valeur $S(\mathcal{E}_i, \mathcal{E}_j)$.
- **Clusters.** Les clusters décrits en section 1.4.3 émergent généralement d'une **somme** ou d'une **agrégation** de ces liaisons binaires (souvent assortis d'un seuil, stabilisant la formation de groupes plus étendus).

Toutefois, la synergie binaire suppose que l'apport mutuel entre \mathcal{E}_i et \mathcal{E}_j ne dépende **pas** explicitement des autres entités. Or, la section 1.4.6 a mis en évidence l'influence du **contexte global** sur les interactions du réseau. La présence d'entités intermédiaires comme \mathcal{E}_k, \dots peut modifier significativement les dynamiques locales. Certaines lacunes peuvent être **comblées**, tandis que des informations complémentaires peuvent se **mutualiser**, révélant ainsi une **coopération émergente** impliquant plusieurs entités simultanément.

A. Synergie n-aire : idée d'un effet collectif

La **synergie n-aire** engage un **ensemble** $\{\mathcal{E}_{k_1}, \dots, \mathcal{E}_{k_m}\}$ (avec $m \geq 3$ dans la plupart des cas), dont la **combinaison** fournit une valeur ajoutée qu'on ne saurait réduire à la somme des synergies binaires. On écrit

$$S_n(\mathcal{E}_{k_1}, \dots, \mathcal{E}_{k_m})$$

pour désigner la synergie qui s'exprime **simultanément** entre ces m entités. On veut y saisir l'idée que "toutes ensemble, elles valent davantage que la somme de leurs collaborations deux à deux". Pour le cas de trois entités $\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c$, on parle parfois de **complémentarité stricte** :

$$S_n(\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c) > S(\mathcal{E}_a, \mathcal{E}_b) + S(\mathcal{E}_b, \mathcal{E}_c) + S(\mathcal{E}_a, \mathcal{E}_c),$$

signifiant que le **trio** dans son intégralité apporte plus que n'importe quelle addition de paires isolées. Pour $m > 3$, ce principe s'étend à une **coopération collective**, où l'interaction ne se limite plus à des relations binaires, mais implique plusieurs entités agissant de concert.

En **théorie de l'information**, on retrouve cette distinction dans :

- **Information mutuelle totale.**

$$I(\mathbf{X}_1, \dots, \mathbf{X}_m) = \sum_{i=1}^m H(\mathbf{X}_i) - H(\mathbf{X}_1, \dots, \mathbf{X}_m),$$

laquelle ne fait pas le tri entre redondance et synergie réellement n-aire.

- **Partial Information Decomposition (PID).**

Ce formalisme sépare la part d'information partagée par tous (redondance) de la part spécifiquement **synergique**, c'est-à-dire l'information qui n'apparaît qu'en combinant l'ensemble complet des variables $\mathbf{X}_1, \dots, \mathbf{X}_m$. S'il émerge une synergie n-aire positive, on découvre un **surcroît** d'information inexistant dans tout sous-groupe plus restreint.

Pour le **DSL**, ces notions indiquent comment un cluster de taille m peut se constituer lorsqu'on détecte une synergie n-aire **positive** (ou dépassant un certain seuil).

B. Intégration de la synergie n-aire dans un réseau

Si les pondérations $\omega_{i,j}$ portent sur des **liaisons binaires**, comment tirer parti d'une synergie n-aire ? Plusieurs pistes :

- **Approche factorisée.**

Créer explicitement une pondération ω_{k_1, \dots, k_m} reliant simultanément les entités $\{\mathcal{E}_{k_1}, \dots, \mathcal{E}_{k_m}\}$. Il faut alors manipuler un **hyper-graphe** (où une “arête” connecte plus de deux nœuds) ou stocker un **tenseur** de pondérations, ce qui devient coûteux si m croît.

- **Approche “bonus/malus”.**

Reprendre le cadre binaire $\omega_{i,j}$ en y ajoutant un **terme** lié à la synergie n-aire. Par exemple, pour trois entités $\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c$, on écrit :

$$\omega_{a,b}(t+1) = \omega_{a,b}(t) + \eta [S(\mathcal{E}_a, \mathcal{E}_b) + \gamma S_3(\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c) - \tau \omega_{a,b}(t)].$$

Ici, γ pèse la contribution de la synergie tripartite. Si $\{\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c\}$ forment un trio très coopératif, chaque lien binaire (a–b, b–c, a–c) reçoit un **surcroît** de renforcement, favorisant la cristallisation d'un micro-cluster à trois.

C. Émergence de micro- ou macro-clusters grâce à la synergie n-aire

Lorsqu'un groupe $\mathcal{E} * k_1, \dots, \mathcal{E} * k_m$ présente une **synergie n-aire élevée**, plusieurs dynamiques émergent naturellement.

Les **liens binaires internes** se **renforcent**, consolidant ainsi la connectivité entre les entités du groupe. Cette intensification des interactions conduit à la **stabilisation** d'un sous-graphe quasi complet, traduisant une forte cohésion interne. Enfin, la coopération entre ces entités génère un

gain global qui dépasse la simple somme des contributions individuelles, illustrant la puissance de la synergie collective.

Si, plus tard, un autre sous-groupe $\{\mathcal{E}_{r_1}, \dots, \mathcal{E}_{r_q}\}$ se joint à ce collectif parce que la **synergie** entre ces deux ensembles élargis se révèle à son tour avantageuse, on peut assister à la **fusion** de clusters en un **macro-cluster** plus vaste. L'**approche n-aire** montre que le **DSL** ne se limite pas aux interactions binaires. Il peut détecter qu'un groupe de **quatre, cinq ou dix entités** coopère de manière exceptionnelle, justifiant ainsi l'émergence d'un **module** structuré au sein du réseau.

Cette structure peut être interprétée comme une **hiérarchie** ou un **treillis** organisationnel.

Aux **feuilles**, on retrouve les **synergies binaires** $\omega_{i,j}$, représentant les connexions directes entre deux entités. Plus haut, les **nœuds intermédiaires** correspondent aux **synergies tri-partites, 4-partites**, et ainsi de suite, traduisant des interactions collectives plus complexes. Enfin, aux **sommets**, les **macro-clusters** émergent, englobant de vastes ensembles d'entités interconnectées par des coopérations multiples.

L'**auto-organisation** d'un SCN peut alors être comprise comme un cheminement dans cet espace d'interactions multiples, valorisant les groupes où la **complémentarité** est nettement profitable.

D. Illustrations et implications

Les **illustrations et implications** de cette approche montrent comment la synergie **n-aire** favorise l'émergence de **macro-clusters** et de structures adaptatives.

Dans la **multimodalité**, certaines modalités comme l'audio et le visuel ne présentent pas nécessairement une forte similarité directe. Cependant, l'introduction d'une troisième modalité, comme le texte, peut révéler une **synergie tripartite**. Un **macro-cluster** émerge alors pour gérer simultanément la parole, l'image du locuteur et les informations projetées, optimisant ainsi la compréhension du contenu.

Dans les **systèmes d'agents**, plusieurs robots peuvent former un **équipage coopératif** où l'interaction entre trois ou plusieurs entités permet de résoudre un problème qu'aucune sous-combinaison plus simple ne pouvait traiter efficacement. Le **DSL** capture cette dynamique et facilite la formation de **super-équipes** capables d'adapter leur structure en fonction des besoins.

En **chimie** ou en **biologie**, deux composés A et B peuvent avoir une interaction limitée jusqu'à l'introduction d'un troisième élément C jouant le rôle de **catalyseur**. Ce phénomène révèle une **synergie triple** (A–B–C) qui dépasse toute combinaison restreinte, illustrant l'émergence de propriétés collectives nouvelles.

1.5. Pourquoi une Approche Synergique ?

Les sections précédentes ont mis en évidence la manière dont le **Deep Synergy Learning (DSL)**, à travers son **architecture générale** et son **auto-organisation** (voir sections 1.3 et 1.4), se distingue d'une approche classique. Pour mieux saisir l'intérêt d'une telle approche, il est essentiel de répondre à une question fondamentale. **Pourquoi** adopter un paradigme **synergique** plutôt que de se limiter aux réseaux de neurones profonds ou à d'autres méthodes d'apprentissage ? Cette section 1.5 propose plusieurs raisons majeures :

Avantages par Rapport aux Réseaux Neuronaux Profonds (1.5.1)

Gestion Naturelle de la Multi-modalité (1.5.2)

Flexibilité vis-à-vis des Données Incomplètes ou Bruitées (1.5.3)

Potentiel d'Auto-Évolution et d'Adaptation Continue (1.5.4)

Réduction de la Dépendance à la Supervision Humaine (1.5.5)

Création de Représentations Riches et plus Interprétables (1.5.6)

Intégration de Dimensions Symboliques ou Cognitives (1.5.7)

Chacun de ces points met en lumière les **bénéfices** d'un modèle qui valorise la **co-opération** dynamique entre entités, plutôt qu'une hiérarchie de couches figées. Nous allons tout d'abord (1.5.1) confronter la logique **synergique** du DSL à la structure rigide des **réseaux neuronaux profonds (Deep Learning)**, pour en dégager les avantages potentiels.

1.5.1. Avantages par Rapport aux Réseaux Neuronaux Profonds

Les **réseaux neuronaux profonds** (CNN, RNN, Transformers, etc.) ont assurément démontré leur **efficacité** dans de nombreux domaines. Cependant, le **DSL** propose une **philosophie** radicalement différente, susceptible d'apporter :

Flexibilité Structurelle

Plastique et Auto-Organisé

Réduction de la Spécialisation Rigide

Facilité d'Intégration Multi-entités

Apprentissage Continu

Chacun de ces volets constitue un avantage important, que nous détaillons ci-après.

1.5.1.1. Flexibilité Structurelle

Dans un **réseau neuronal profond** classique, la **topologie** (nombre et type de couches, neurones, schémas de fusion de données) est entièrement **préétablie** avant l'apprentissage. Par exemple, un **CNN** (Convolutional Neural Network) organise les données selon des couches **convolutionnelles**, suivies éventuellement de **pooling** et de **couches fully-connected**, sans changement d'architecture au cours de l'entraînement. De même, les **Transformers** enchaînent des **blocs self-attention** répétés, toujours selon un canevas décidé en amont. Ces modèles **ne modifient** pas leur structure interne en cours de fonctionnement :

Nombre de couches, arrangement des blocs et synapses demeurent fixes pendant tout l'apprentissage.

Ils **n'autorisent** pas la création ou la **suppression** de neurones ou de liaisons et conservent un agencement identique, même si la **distribution** des données ou le **contexte** subit des modifications significatives.

À l'inverse, dans le **Deep Synergy Learning (DSL)**, on insiste sur la dynamique de ses **entités** et de leurs **liaisons**. Les pondérations $\omega_{i,j}(t)$ entre deux entités \mathcal{E}_i et \mathcal{E}_j évoluent selon la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. Il est possible qu'un **lien** émerge (si la synergie grandit et dépasse un certain **seuil**), qu'il **disparaisse** (si l'intérêt coopératif s'avère trop faible), ou qu'il se **recrée** au cours du temps. La **flexibilité topologique** qui en résulte se formalise par des règles de mise à jour de la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η désigne le **taux d'apprentissage** et τ un **terme de régulation**. Quand la synergie S s'avère positive et assez élevée, la liaison se **renforce** ; dans le cas contraire, elle **s'affaiblit** et peut même s'annuler. Cette dynamique se révèle d'autant plus souple qu'on peut imposer une **parsimonie** via un seuil ω_{\min} ou un **cap** ω_{\max} , comme le rappelle la section 1.4.5.

La conséquence directe de cette **reconfiguration dynamique** tient dans la capacité du réseau à **s'adapter** en continu à l'arrivée de nouvelles données ou à un **changement** de distribution. Au lieu de maintenir invariablement le même chemin de traitement (comme dans un **CNN** figé ou un **Transformers** aux blocs intangibles), le **DSL** autorise une **reliance** directe entre certaines entités apparues pertinentes, tout en relâchant ou en supprimant d'autres connexions devenues inutiles. On peut ainsi écrire, de façon synthétique :

Flexibilité topologique \Rightarrow Capacité d'adaptation et résilience face à l'évolution du contexte.

Dès lors, si la distribution des données subit un changement progressif (par exemple, l'apparition d'une nouvelle modalité sensorielle), le **DSL** peut incorporer de nouvelles **entités** \mathcal{E}_{nov} et renforcer les liaisons $\omega_{\text{nov},j}$ jugées porteuses d'une forte **synergie**. Simultanément, il peut affaiblir ou rompre des liens moins pertinents. C'est précisément cette **malléabilité** qui caractérise le **SCN** et le différencie de la rigidité structurelle des architectures profondes traditionnelles.

Ce **principe** de flexibilité structurelle jette les bases d'une **évolution** continue de la topologie, conforme à l'esprit d'**auto-organisation** présenté dans les sections 1.4.3 et 1.4.5. En ce sens, le **DSL** épouse davantage la **complexité** et la **variabilité** des environnements réels, tout en garantissant que seules les **connexions** profitables (au sens de la synergie) se **consolident** dans le temps.

1.5.1.2. Plasticité et Auto-Organisation

Dans les **réseaux de neurones** usuels, la phase d'apprentissage consiste principalement à **ajuster les poids** dans une **architecture** figée à l'avance. En revanche, le **Deep Synergy Learning (DSL)** adopte une approche plus **plastique**. Il permet à la **topologie du Synergistic Connection Network (SCN)** de **se transformer** au fil du temps, en ajustant dynamiquement les connexions et les structures en fonction des synergies détectées. Les liaisons $\omega_{i,j}(t)$ qui peinent à démontrer une **synergie** positive ou dont la valeur reste trop faible peuvent **disparaître**, tandis que d'autres connexions, révélant un intérêt mutuel marqué, se **renforcent** jusqu'à forger de nouveaux chemins ou **clusters**.

Pour exprimer cette mise à jour, on recourt le plus souvent à une règle sous la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η représente le **taux d'apprentissage**, τ un terme de **régulation**, et $S(\mathcal{E}_i, \mathcal{E}_j)$ la **synergie** (distance, similarité, information mutuelle...). Ainsi, chaque itération propose une **réévaluation** de la pertinence du lien $\omega_{i,j}$. Si la synergie demeure assez élevée, la liaison persiste ou se consolide ; sinon, elle tend vers zéro et peut être considérée comme **rompue** selon un seuil ω_{\min} (cf. section 1.4.5 pour la gestion des seuils).

Cette **démarche** de régulation évoque la **plasticité synaptique** que l'on observe dans les **systèmes biologiques**. À l'image des neurones qui renforcent leurs connexions en cas de co-activation et en abandonnent d'autres, les entités du DSL remanient leur **graphe** de manière à privilégier les **coopérations** les plus fructueuses. Les **clusters** (sections 1.4.3 et 1.4.4) émergent **spontanément**, sans nécessiter de paramétrage explicite du nombre de groupes. Ils apparaissent dès que les pondérations internes d'un sous-ensemble d'entités atteignent un certain niveau de **synergie**, formant ainsi un **sous-réseau** fortement connecté et structuré.

Cette aptitude à **auto-organiser** la structure s'avère avantageuse pour l'**adaptation** en continu. On peut en effet passer d'une tâche à l'autre ou intégrer une **nouvelle modalité** de données (image, audio, signal textuel) sans reconstruire l'architecture dans son intégralité, puisqu'il suffit d'ajouter des entités et de laisser les liaisons se former ou s'éteindre de façon autonome selon la **synergie** détectée. Les **clusters** demeurant utiles persistent, tandis que de **nouveaux** se créent si des **interactions** inédites se révèlent rentables. Le **réseau** conserve ainsi un **degré de flexibilité** qui lui permet d'**ajuster sa complexité** selon le **contexte**, proposant un fonctionnement plus souple qu'un modèle entièrement figé.

1.5.1.3. Réduction de la *Spécialisation Rigide*

Dans un **réseau profond** standard comme un **CNN** dédié à la reconnaissance visuelle, la structure se présente souvent de manière **verticalisée**, où les premières couches extraient des **bords**, puis des formes plus élaborées, jusqu'à la classification finale. Cette disposition rend la **réutilisation** des *features* complexes si l'on souhaite aborder d'autres tâches ou modalités, à moins d'effectuer un **fine-tuning** parfois lourd. Les couches initiales sont fixées dans une fonction très spécifique, et la **coopération** entre différents blocs (par exemple la fusion de vision et d'audio) se voit

généralement cantonnée à un **niveau** imposé dans l'architecture ou à un **module** explicitement conçu pour la multimodalité.

Dans le cadre du **Deep Synergy Learning (DSL)**, la vision se veut plus **flexible**. Les **entités** composant le **Synergistic Connection Network (SCN)** incarnent chacune une **source d'information** ou un **bloc fonctionnel** susceptible d'être réutilisé dans d'autres contextes. La **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ détectée entre deux **entités** \mathcal{E}_i et \mathcal{E}_j ne se limite pas à une relation hiérarchique entre un étage “plus haut” ou “plus bas”. Toute entité peut, au fil des itérations, **établir un lien** avec une autre, dès lors que leur coopération génère un **gain** quantifié par la fonction S .

Un **exemple** concret peut être imaginé lorsqu'un **bloc** $\mathcal{E}_{\text{bords}}$, spécialisé dans la détection de **bords** visuels, observe soudain qu'il existe un **gain** d'associer ces **features** à un **bloc** $\mathcal{E}_{\text{audio}}$ focalisé sur les **fréquences** sonores, notamment quand des corrélations entre mouvements de lèvres et signaux vocaux se manifestent. Au sein d'un **CNN** traditionnel, l'**intégration** de telles informations se produirait typiquement à un **niveau tardif** ou via un **module spécialisé**, ce qui limite la portée de la coopération. Dans le **DSL**, au contraire, la liaison $\omega_{\text{bords}, \text{audio}}$ peut croître de manière **spontanée** si la **synergie** calculée (par exemple $\exp(-\|\mathbf{x}_{\text{bords}} - \mathbf{x}_{\text{audio}}\|^2/\sigma^2)$) s'avère **élevée** et dépasse un certain **seuil** ω_{\min} (section 1.4.5). Les deux blocs initient alors un **micro-cluster**, échangent leurs informations, et ainsi la **fusion** s'opère de manière autonome. Cette capacité à **tisser** des liens coopératifs entre entités diverge de la spécialisation rigide instaurée par des **pipelines** de couches fixes, et elle illustre la **souplesse** du **DSL** pour exploiter des **synergies** inattendues.

1.5.1.4. Facilité d'Intégration Multi-Entités

Dans la plupart des **réseaux neuronaux profonds** classiques, il est usuel de concevoir un **design architectural** bien particulier pour **fusionner** plusieurs entrées issues de différentes modalités, comme une **image**, un **texte** et un **signal audio**. On trouve ainsi des approches où un **CNN** traite l'image, un **RNN** (ou un **Transformers**) interprète le texte, puis une **couche de fusion** se situe au sommet pour combiner les vecteurs latents. Cette disposition requiert, dans son principe, une intervention humaine afin de spécifier à quel niveau et selon quelles règles la **fusion** se produit.

Dans le **Deep Synergy Learning (DSL)**, la perspective est nettement plus **flexible**. Chaque **modalité** s'incarne dans une ou plusieurs **entités** \mathcal{E}_{mod} dédiées, que l'on peut définir comme un bloc perceptif ou un ensemble de **features** cohérent. Les **liens** $\omega_{i,j}$ se **construisent** de manière autonome, selon la **synergie** mesurée entre les entités \mathcal{E}_i et \mathcal{E}_j . On peut exprimer cette dynamique à l'aide de la règle d'adaptation :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où η désigne le **taux d'apprentissage**, τ un **terme de régulation**, et $S(\mathcal{E}_i, \mathcal{E}_j)$ la **synergie** (section 1.4.4). Cette règle fait en sorte que toute **coopération** mutuellement avantageuse s'**intensifie** si elle contribue à un gain, alors que les associations moins pertinentes se **dissipent**. Dans un **contexte** multimodal, l'image, le texte et l'audio sont autant d'entités $\{\mathcal{E}_{\text{image}}, \mathcal{E}_{\text{texte}}, \mathcal{E}_{\text{audio}}\}$ susceptibles de découvrir des **corrélations** bilatérales ou trilatérales, sans qu'une **couche de fusion** imposée par un concepteur soit nécessaire.

Lorsque la **synergie** entre ces modalités se révèle conséquente, les pondérations $\omega_{\text{image}, \text{texte}}$, $\omega_{\text{image}, \text{audio}}$ ou $\omega_{\text{texte}, \text{audio}}$ croissent, formant de manière **spontanée** un **cluster** multimodal où les entités coopèrent directement. Cette **auto-organisation** se renforce si l'association apporte un bénéfice mesurable (par exemple, une performance améliorée sur une tâche de classification ou une plus grande robustesse). Inversement, si une modalité \mathcal{E}_{mod} n'apporte pas d'information utile ou se révèle trop bruitée, la **synergie** reste faible et la liaison ne s'établit pas, ou se rompt rapidement.

Une autre conséquence réside dans la **capacité** du réseau à **adapter** en continu les entités multimodales, autorisant la **formation** et la **dissolution** d'un **cluster** au gré du contexte. Cette propriété est en net contraste avec l'idée traditionnelle d'une couche de fusion figée où toutes les modalités se rejoignent selon un schéma unique. Le **DSL** propose donc une **répartition** dynamique de l'intégration multimodale, dans laquelle les **entités** se reconfigurent en fonction de l'**opportunité** d'apprentissage.

Au final, la **facilité d'intégration multi-entités** découle directement de la manière dont les **liaisons synergiques** sont définies et mises à jour. Chaque modalité, qu'elle concerne le **visuel**, le **textuel**, le **sonore** ou un **autre flux**, est **libre** de **découvrir** et de **former** des **connexions** directes avec n'importe quel **bloc** complémentaire, dès lors qu'une **valeur ajoutée** en résulte. Le **réseau** opère ainsi sous une forme d'**auto-organisation** (sections 1.4.3 et 1.4.5), permettant d'exploiter la diversité des **modalités** sans nécessiter le design d'un unique **module** de fusion, et garantissant une **adaptation** continue à la **variabilité** de l'environnement.

1.5.1.5. Apprentissage Continu

Dans de nombreux **réseaux neuronaux profonds**, l'entraînement se déroule dans un cadre essentiellement statique. On se contente généralement d'effectuer une rétropropagation sur un **ensemble de données** figé, et il s'avère difficile de réaliser un **apprentissage continu** sans retomber dans le phénomène de **catastrophic forgetting** au moment d'acquérir de nouvelles tâches ou de faire face à un changement de distribution. De plus, la structure même du réseau (couche, neurones) demeure immuable pendant tout l'apprentissage, ce qui limite considérablement la possibilité d'**évoluer** au fil du temps.

À l'inverse, le **Deep Synergy Learning (DSL)** s'appuie sur un **mécanisme de pondérations adaptatives** (discuté en section 1.4.5) dans lequel chaque liaison $\omega_{i,j}(t)$ peut se créer, se renforcer ou disparaître selon la **synergie** qu'elle procure. Cette propriété ouvre la voie à un **apprentissage continu** plus fluide, car le **réseau** peut accueillir de nouvelles **entités** $\mathcal{E}_{\text{nouv}}$ chaque fois qu'un nouveau flux de données ou une nouvelle modalité fait son apparition. Les **liaisons** $\omega_{\text{nouv}, j}$ se forment si elles sont jugées bénéfiques, tandis que les anciens **clusters** jugés toujours utiles demeurent stables. Ainsi, l'**oubli** des acquis se trouve **atténué**, puisque les **connections** consolidées par le passé ne sont pas supprimées tant qu'elles conservent une synergie positive. Les entités correspondant aux tâches ou domaines précédents (sections 1.4.3 et 1.4.4) ne disparaissent pas, et les sous-groupes de coopération qui s'étaient formés peuvent persister ou se réactiver ultérieurement.

Une telle **flexibilité** convient particulièrement aux **environnements dynamiques**, où le **réseau** doit sans cesse **apprendre** de nouvelles classes d'objets, de nouvelles langues, ou s'adapter à des évolutions de contexte sans recommencer l'**entraînement** depuis zéro. Le **DSL** se distingue alors de la rigidité caractéristique d'un **réseau profond** standard, qui impose généralement de réinitialiser ou de "fine-tuner" un ensemble de couches fixes lors de l'introduction d'un nouveau domaine. La possibilité de **reconfigurer** la topologie et d'**absorber** ou de **cesser** d'utiliser certaines sources de données fait du **SCN** un cadre mieux adapté à un **apprentissage Lifelong** ou **continu**, tout en préservant la robustesse et la puissance d'un paradigme sub-symbolique.

Dans les sous-sections suivantes (1.5.2 à 1.5.7), nous approfondirons d'autres avantages majeurs, comme la **gestion naturelle de la multi-modalité** (1.5.2), la **flexibilité** face aux données partielles (1.5.3), l'**auto-évolution** (1.5.4), la **réduction** de la supervision (1.5.5), la **création** de représentations plus **riches** (1.5.6) et l'**intégration** de dimensions **symboliques** (1.5.7). L'ensemble consolidera l'idée qu'une **approche synergique** offre un panel d'atouts pour concevoir des systèmes apprenants plus **évolutifs, modulaires et généraux** que ne le permettent les architectures entièrement figées du deep learning traditionnel.

1.5.2. Gestion Naturelle de la Multi-modalité

Une des motivations fortes qui a conduit à l'élaboration du **Deep Synergy Learning (DSL)** est la possibilité d'**intégrer** plusieurs types de données (*images, sons, textes, capteurs variés, etc.*) de manière qu'elles **coopèrent** au lieu de simplement être fusionnées dans une couche dédiée. Dans les réseaux neuronaux profonds classiques, la multi-modalité impose souvent de **concevoir** des modules spécifiques (p. ex. un CNN pour l'image, un RNN ou un Transformer pour le texte, un autre réseau pour l'audio), puis de **fusionner** ces modules dans une partie supérieure de l'architecture. Cette approche, bien qu'efficace dans de nombreux cas, demeure relativement **rigide**. La fusion s'effectue à un niveau **préétabli** et évolue peu en fonction du **contexte** ou de la **dynamique interne** du réseau.

Le **DSL**, à l'inverse, favorise une **intégration plus organique**. Chaque modalité est représentée par une ou plusieurs **entités** \mathcal{E}_{mod} , qui ajustent **spontanément** leurs connexions en fonction de la **synergie** détectée. Elles peuvent ainsi **établir** ou **rompre** des liens avec des entités d'autres modalités, garantissant une flexibilité et une adaptation continue du réseau. Cette sous-section (1.5.2) détaille :

- **Pourquoi** la multi-modalité est "naturellement" absorbée par le DSL,
- Comment les entités **visuelles, auditives, textuelles** (par ex.) peuvent **former** des clusters multimodaux,
- Quels **avantages** cette gestion apporte (robustesse, découvertes de liens inattendus, etc.).

1.5.2.1. Les Entités comme Vecteurs ou Blocs Multimodaux

Dans le **Deep Synergy Learning (DSL)**, chaque **entité** \mathcal{E}_i peut se spécialiser dans un type de données particulier ou combiner déjà plusieurs **descripteurs**. Il n'y a pas l'obligation, comme dans

un réseau multimodal classique, d’assembler deux flux (image, audio, etc.) au niveau d’une couche imposée ou dans des branches séparées. Au lieu de cela, toutes les entités coexistent dans un **même “espace” ou graphe** $\{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n\}$. On y retrouve des entités $\mathcal{E}_{\text{image}}$, $\mathcal{E}_{\text{audio}}$, $\mathcal{E}_{\text{texte}}$, ou des blocs déjà **multimodaux** combinant différents **features**. Il n’existe pas de **pyramide** de couches fixes, comme un **CNN** qui détecterait d’abord les bords, puis les textures, puis les objets. Au contraire, le **DSL** insiste sur le principe d’**auto-organisation**, faisant que toute entité \mathcal{E}_i peut se **lier** à toute autre \mathcal{E}_j dès lors qu’une **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ significative est détectée. La mise à jour de la pondération $\omega_{i,j}(t)$ suit la règle discutée en section 1.4.5, et si la coopération apporte un gain, on obtient un renforcement de la liaison $\omega_{i,j}$. Ainsi, une entité de type *texte* \mathcal{E}_{txt} n’est pas contrainte de se fusionner uniquement avec une entité *image* \mathcal{E}_{img} à un étage supérieur prédéfini. La **fusion** peut se produire **à tout moment**, entre n’importe quelles entités, dès lors qu’une **valeur ajoutée** est détectée. Cette démarche favorise la **formation** naturelle de **clusters** multimodaux là où c’est nécessaire, et permet aux entités déjà mixtes (par exemple combinant audio et vision) de se relier également à d’autres (texte, capteurs...), rendant le réseau plus **souple** et mieux à même de gérer la **pluralité** des sources de données.

1.5.2.2. Formation de Clusters Multimodaux

Le **Deep Synergy Learning (DSL)** emploie une règle de mise à jour décrite dans la section 1.4.5, qui se formalise ainsi :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Lorsque, par exemple, une **entité visuelle** $\mathcal{E}_{\text{visuelle}}^a$ (extraction d’images) et une **entité auditive** $\mathcal{E}_{\text{auditive}}^b$ (analyse du spectre sonore) constatent une **synergie** satisfaisante, la liaison $\omega_{a,b}$ se **renforce** progressivement. Les sections 1.4.3 et 1.4.4 évoquent déjà ce principe de consolidation, où toute **coopération** profitable se traduit par une augmentation de la pondération. Au fil des itérations, d’autres entités, qu’elles soient textuelles ou associées à une autre modalité, peuvent se **greffer** à ce duo si elles y perçoivent-elles aussi un **gain**. S’organise alors un **cluster** multimodal $\{\mathcal{E}_{\text{vis}}, \mathcal{E}_{\text{aud}}, \mathcal{E}_{\text{txt}}\}$, destiné, par exemple, à la reconnaissance d’événements conjoints dans une vidéo, en y associant le son et des sous-titres.

Un exemple important survient lors de l’analyse synchronisée **vidéo + audio**. Une entité \mathcal{E}_{CNN} , spécialisée dans la détection d’objets ou de mouvements visuels, et une entité $\mathcal{E}_{\text{Audio}}$, orientée vers les fréquences sonores, découvrent une **corrélation** entre le contenu visuel (mouvements de l’orateur) et des sons spécifiques. Si la **fonction de synergie** $S(\mathcal{E}_{\text{CNN}}, \mathcal{E}_{\text{Audio}})$ affiche une valeur élevée, la pondération $\omega_{\text{CNN}, \text{Audio}}$ croît. Il en résulte la formation d’un **sous-réseau** constitué de ces entités, apte à identifier les vidéos où un son précis se déclenche simultanément avec un geste ou un mouvement labial.

Un autre scénario se rencontre dans la **fusion texte + image**. Si une entité $\mathcal{E}_{\text{visuel}}$ extrait les objets visibles dans une scène et qu’une entité $\mathcal{E}_{\text{langage}}$ génère des phrases descriptives (via un modèle de type RNN ou Transformers), la **co-information** qu’elles partagent peut montrer qu’en les associant, on obtient une légende automatique bien plus fidèle. Cette plus-value se reflète dans la

liaison $\omega_{\text{visuel}, \text{langage}}$, qui se **solidifie** de manière autonome. Le **cluster** $\{\mathcal{E}_{\text{visuel}}, \mathcal{E}_{\text{langage}}\}$ se spécialise alors dans la génération de légendes, sans imposer de couche de fusion prédéterminée.

On peut enfin imaginer un **cluster hybride** regroupant à la fois la “capture de mouvement” via des capteurs inertiels, la “reconnaissance d’images” au moyen d’une entité visuelle, et l’“analyse du spectre sonore”. Ce **triplet** se cristallise dès lors qu’il existe une forte complémentarité pour identifier des événements qui combinent aspects visuels, audio, et déplacements corporels. Le **SCN** encourage ainsi la création de tels **clusters** dès que les synergies apparaissent bénéfiques, ce qui permet au réseau de s’adapter aux environnements multimodaux et de tirer le meilleur parti de chacune des sources disponibles.

1.5.2.3. Avantages de la Co-Organisation Multimodale

Dans un **réseau neuronal** habituel, la fusion des différents **flux** (par exemple, l’image et le son) est généralement décidée à l’avance, souvent dans les étages supérieurs ou dans une **couche intermédiaire** prédefinie. Le **Deep Synergy Learning (DSL)** adopte une approche différente. Chaque **entité** \mathcal{E}_i peut **rechercher** spontanément des partenaires synergiques, **évaluer** si la coopération améliore les performances ou génère un gain en information mutuelle, et **établir** des liaisons directes $\omega_{i,j}$ sans qu’une couche de fusion prédefinie ne soit requise. Cette démarche favorise l’apparition de **combinaisons** inhabituelles mais profitables, comme l’association de l’analyse audio, de la mesure de température et de l’intensité lumineuse, si un cluster pertinent émerge.

La **règle** de mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)]$$

permet à chaque entité de constater, de façon autonome, si sa **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ s’avère suffisamment élevée pour justifier une **collaboration**. Ainsi, il n’existe pas d’étage unique où l’on agrégerait tous les flux, mais plutôt un **réseau** dont la topologie évolue pour intégrer toute coopération bénéfique.

Lorsqu’une modalité est **perturbée** ou moins fiable (par exemple l’audio, soumis à un **bruit** important ou une panne de capteur), la synergie associée $\omega_{\text{aud}, \cdot}$ s’affaiblit, car les entités n’y trouvent plus de **valeur ajoutée**. Le **cluster** multimodal se réorganise donc en renforçant d’autres connexions, comme celles reliant la vision, le texte ou des capteurs alternatifs. Le système gagne ainsi en **résilience**. Il ne repose pas sur un pipeline figé ni sur un module dédié à chaque modalité, mais s’**autorégule** en continu pour maintenir une **robustesse** globale, s’adaptant aux changements de contexte et compensant d’éventuelles défaillances de certaines sources.

1.5.2.4. Éléments Mathématiques : Synergie “Multi-Modal”

Dans le **Deep Synergy Learning (DSL)**, la mesure de **synergie** entre différentes **modalités** (comme l’audio, la vision ou le texte) peut prendre plusieurs formes selon le type de données ou la définition souhaitée de la **co-opération**. Deux grandes approches sont envisagées. La première

repose sur l'utilisation de la **co-information** (ou multi-information) issue de la théorie de l'information, tandis que la seconde exploite une **distance** ou une **similarité** inter-modale définie dans un **espace latent commun**.

A. Mesure de co-information

Lorsque l'on manipule un ensemble de modalités $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m$, il est possible d'estimer une **co-information** ou **mutual information** globale :

$$S(\mathbf{X}_1, \dots, \mathbf{X}_m) = MI(\mathbf{X}_1, \dots, \mathbf{X}_m).$$

Dans certains cas, la **Partial Information Decomposition (PID)** permet de séparer la part strictement *synergique* de la part *redondante* ou *unique*. Dans le **DSL**, on s'intéresse à l'**analyse n-aire** au sens où si la **conjonction** de deux (ou plus) modalités, par exemple $\mathbf{X}_{\text{visuelle}}$ et $\mathbf{X}_{\text{auditive}}$, produit une **information** qu'aucune des modalités ne détenait isolément, alors on en conclut un **gain** pour la **synergie**. Ce **gain** se traduit par un **renforcement** des pondérations. Lorsque l'association d'un flux visuel et d'un flux audio se révèle bénéfique, la liaison $\omega_{\text{vis}, \text{aud}}$ se renforce conformément à la règle de mise à jour décrite en **section 1.4.5**. L'**auto-organisation** du réseau valorise ainsi la coopération entre modalités lorsqu'elle apporte un surcroît d'information ou de performance.

B. Distance ou Similarité inter-modale

En alternative à la co-information, on peut quantifier la **synergie** via une **distance** (ou **similarité**) entre les modalités, à condition de se situer dans un **espace latent** commun. Dans la pratique, cela implique généralement les étapes suivantes :

1. Embeddings partagés. On forme une représentation vectorielle \mathbf{z}_{vis} pour la partie **visuelle** et \mathbf{z}_{aud} pour la partie **audio**, toutes deux dans \mathbb{R}^d . Ces embeddings peuvent provenir d'un réseau d'apprentissage dédié ou être appris conjointement.

2. Distance ou similarité. On calcule une **norme** $\|\mathbf{z}_{\text{vis}} - \mathbf{z}_{\text{aud}}\|$ ou une **similarité cosinus** $\langle \mathbf{z}_{\text{vis}}, \mathbf{z}_{\text{aud}} \rangle$. Cette quantité rend compte de la **proximité** ou de la **complémentarité** entre les deux modalités une fois projetées dans l'espace latent.

3. Définition de la synergie. En inversant ou en normalisant la distance, on obtient un **score de coopération** :

$$S(\mathbf{z}_{\text{vis}}, \mathbf{z}_{\text{aud}}) = \exp(-\|\mathbf{z}_{\text{vis}} - \mathbf{z}_{\text{aud}}\|^2/2\sigma^2) \quad \text{ou} \quad 1/(1+\|\mathbf{z}_{\text{vis}} - \mathbf{z}_{\text{aud}}\|^2),$$

ou encore en recourant à une **similarité** normalisée dans $[0,1]$. Cette mesure, insérée dans la formule de mise à jour des pondérations $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \dots$, renseigne sur la **pertinence** qu'il y a à **connecter** et **coordonner** ces deux modalités.

Synthèse et implication dans le DSL

Le **DSL** n'impose pas de définition unique de la synergie ; toute **fonction** mesurant la **coopération** peut s'appliquer. La **co-information** met en évidence des **complémentarités** complexes (non linéaires), en évaluant l'information que seules deux (ou plusieurs) modalités, **combinées**, parviennent à extraire. De son côté, la **distance** (ou la **similarité**) dans un **espace latent** offre un

calcul plus direct et souvent moins coûteux, permettant d'établir rapidement une **coopération** si les embeddings s'avèrent proches ou orientés de façon compatible.

Dans tous les cas, le **DSL** conserve son principe fondamental. La **synergie inter-modale** joue le rôle d'un **signal**, suggérant que certaines entités multimodales bénéficiaient d'une **connexion renforcée**, favorisant ainsi la formation d'un **cluster dédié**. Cette logique s'applique à deux entités (binaire), mais peut s'étendre à des groupes plus vastes (section 1.4.7) lorsque plusieurs **modalités** s'enrichissent mutuellement. Dans une perspective **multimodale**, on voit ainsi se constituer des **clusters** audio–vidéo–texte, ou audio–vision–capteurs, sans qu'un concepteur doive imposer explicitement à quel niveau se fait la **fusion**. Le **SCN** se reconfigure et s'**auto-organise**, renforçant les **liens** rentables au sens de la synergie globale.

1.5.2.5. Perspectives : Plus de Fluidité, Plus de Découvertes

Dans un **Deep Synergy Learning (DSL)**, la fusion multimodale n'est pas confinée à une couche ou un module imposé à l'avance. Elle repose sur l'aptitude de chaque **entité** à forger des liens **synergiques** dès lors que la coopération procure un **gain**. Cette configuration autorise plusieurs évolutions particulièrement souples.

Un premier aspect concerne la **création spontanée** de **clusters multimodaux**. Lorsqu'un événement sollicite simultanément divers flux, par exemple un son particulier, un motif visuel distinctif et un mot-clé textuel, le **DSL** favorise la formation d'un **cluster** auto-organisé combinant toutes ces sources. Il n'est pas nécessaire de prédefinir, lors de la conception du réseau, à quel niveau ou dans quel sous-ensemble la fusion doit se produire. Le **Synergistic Connection Network** s'**auto-organise** en **détectant** les entités pertinentes et en **renforçant** leurs liaisons $\omega_{i,j}$ lorsque la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ est élevée. Cette auto-organisation, évoquée en section 1.4.3, rend la fusion bien plus fluide et adaptative.

Un deuxième enjeu apparaît lorsque des **modalités** ou sources de données nouvelles se présentent. Si un capteur inédit vient produire un nouveau flux, il suffit d'introduire une entité $\mathcal{E}_{\text{capteur}}$ correspondante. Les pondérations $\omega_{\text{capteur},x}$ se mettent alors à jour selon la même loi que précédemment (section 1.4.5). Si les interactions avec des entités déjà existantes procurent un réel **bénéfice** à l'apprentissage ou à la représentation, la **synergie** grimpe ; sinon, le lien demeure faible et finit par s'éteindre. L'essentiel réside dans l'absence de nécessité de **reconstruire** l'architecture globale. Le réseau s'**auto-adapte**, formant et supprimant des connexions en fonction des **opportunités** offertes par le nouveau flux.

Un troisième point souligne la **réutilisation** des entités spécialisées. Dans un paradigme classique, un module dédié, par exemple à l'analyse de fréquences sonores, risque de ne pouvoir servir qu'à une tâche unique (identification de parole ou détection de musique). Au contraire, dans le **DSL**, une entité $\mathcal{E}_{\text{audio}}$ capable de détecter des éléments discriminants dans le spectre sonore peut être **sollicitée** par plusieurs **clusters** simultanément. Elle peut ainsi intégrer un sous-réseau dédié à la reconnaissance de mots clés, un autre spécialisé dans la détection de bruits inhabituels, ou encore un qui associe des fragments de parole aux mouvements des lèvres. Chaque fois que la **synergie** le justifie, la liaison $\omega_{\text{audio},\dots}$ se renforce dans le cluster concerné, sans imposer de duplication de la fonctionnalité ni de modification d'une couche fixe.

L'ensemble de ces mécanismes assure une **fluidité** et un **potentiel d'exploration** supérieurs par rapport aux modèles multimodaux figés. Le **DSL** favorise l'**émergence** et la **dissolution** dynamiques de clusters multimodaux, l'**intégration** graduelle de modalités nouvelles, et la **mutualisation** de blocs spécialisés au service de plusieurs tâches. Cette architecture, loin d'être imposée, se déploie au fil des itérations par simple adaptation des pondérations $\{\omega_{i,j}(t)\}$, conduisant à une plus grande **richesse** de découvertes et de combinaisons entre entités.

1.5.3. Flexibilité vis-à-vis des Données Incomplètes ou Bruitées

Outre la capacité à gérer de multiples flux (voir 1.5.2), l'un des atouts majeurs du **Deep Synergy Learning (DSL)** réside dans sa **tolérance** accrue face aux **données incomplètes** ou **fortement bruitées**. Dans les approches d'apprentissage traditionnelles (réseaux neuronaux profonds inclus), on cherche souvent à **normaliser** ou **compléter** les données manquantes, voire on opte pour un **prétraitement** lourd afin de filtrer les bruits. Le DSL, par sa **structure auto-organisée**, autorise un réseau à composer **localement** avec les lacunes et à **réajuster** ses connexions de manière à s'appuyer sur les sources les plus fiables.

Cette section (1.5.3) met en avant les principes qui confèrent au DSL sa **robustesse** et sa **souplesse** quand on fait face à des entrées partielles, hétérogènes ou bruitées.

1.5.3.1. Rôle de l'Auto-Organisation dans la Gestion du Bruit

Dans le **Deep Synergy Learning (DSL)**, les entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ s'**agglomèrent** en **clusters** (tels que décrits en section 1.4.3) lorsque leurs synergies bilatérales $S(\mathcal{E}_i, \mathcal{E}_j)$ atteignent un certain **niveau**. L'auto-organisation agit alors comme un **mécanisme** filtrant. Si une entité \mathcal{E}_k se révèle trop **bruyante**, ou plus généralement peu profitable (synergie insuffisante, voire négative, avec les autres), ses connexions $\omega_{k,j}$ **dépérissent** naturellement, selon la **mise à jour** :

$$\omega_{k,j}(t+1) = \omega_{k,j}(t) + \eta [S(\mathcal{E}_k, \mathcal{E}_j) - \tau \omega_{k,j}(t)].$$

Si la synergie $S(\mathcal{E}_k, \mathcal{E}_j)$ reste **faible** ou négative, la liaison $\omega_{k,j}$ diminue itération après itération jusqu'à se rapprocher d'un **seuil** ω_{\min} en deçà duquel elle est jugée **inexistante**. L'entité \mathcal{E}_k tend dès lors à **s'isoler** de la structure principale et ne contribue plus de manière significative aux décisions ou au regroupement de clusters, limitant l'**impact** du bruit ou des mesures peu fiables sur l'ensemble du **Synergistic Connection Network**. Cette propriété confère au **DSL** une **robustesse** importante, car il n'est pas nécessaire d'éliminer d'emblée les données incertaines. C'est la **dynamique** des pondérations qui réalise ce filtrage de manière **locale** et **autonome**, en ajustant les connexions selon la synergie détectée.

De façon analogue, une entité ne possédant que des **données incomplètes** (par exemple, un capteur qui ne fonctionne que par intermittence) peut tout de même établir des **liens forts** avec d'autres entités si la partie (même partielle) de ses observations est **pertinente** et apporte un gain synergique réel. L'apprentissage ne rejette pas d'emblée les capteurs partiellement défaillants, mais évalue la **valeur ajoutée** qu'ils apportent, comme expliqué dans la section 1.4.4 sur la définition de S . En fonction de cette évaluation, il **renforce** ou **dissout** les liaisons selon leur pertinence. Le **DSL** teste

ainsi localement l'**efficacité** des données partielles et, grâce à la dynamique adaptative de $\omega_{k,j}(t)$, il détermine si une entité doit être maintenue dans un cluster ou rester en marge du réseau.

Ce procédé évite la nécessité d'une **stratégie** spécifique pour rejeter le **bruit** ou manipuler les **données manquantes**. Le **système** s'auto-regularise, en quelque sorte, grâce à la diminution des pondérations non rentables, ce qui se traduit par une **atténuation** de l'influence des sources imprécises. L'**auto-organisation** réalise donc une forme de **gestion du bruit** distribuée, où chaque entité (ou module) se maintient ou s'éclipse selon la **synergie** réellement engendrée avec les autres composantes du **SCN**, évitant ainsi de nuire à la **performance** globale.

1.5.3.2. Interprétation dans un Cadre Bruité

Lorsqu'une entité \mathcal{E}_k se trouve exposée à un **bruit** important dans ses représentations \mathbf{x}_k , sa **similarité** ou son **information mutuelle** avec les autres entités demeure faible ou fluctuante. On peut considérer par exemple la fonction

$$S(\mathcal{E}_k, \mathcal{E}_j) = \frac{1}{1 + \| \mathbf{x}_k - \mathbf{x}_j \|^2}.$$

Si \mathbf{x}_k varie de manière aléatoire (bruit élevé), alors $\| \mathbf{x}_k - \mathbf{x}_j \|^2$ reste souvent grande ou instable, réduisant ainsi la valeur de $S(\mathcal{E}_k, \mathcal{E}_j)$. En se référant à la loi de mise à jour des pondérations (section 1.4.5), on voit que

$$\omega_{k,j}(t+1) = \omega_{k,j}(t) + \eta [S(\mathcal{E}_k, \mathcal{E}_j) - \tau \omega_{k,j}(t)].$$

La valeur de $\omega_{k,j}(t)$ décroît donc si la **synergie** $S(\mathcal{E}_k, \mathcal{E}_j)$ ne compense pas suffisamment le terme de régulation $\tau \omega_{k,j}(t)$. Cette **diminution** progressive des liaisons $\omega_{k,j}$ conduit l'entité \mathcal{E}_k à **s'isoler** si le bruit la rend incohérente vis-à-vis des entités du cluster. Il n'est pas nécessaire d'ajouter un algorithme spécifique pour exclure certaines entités, car le **DSL** prend en charge ce processus de manière **auto-organisée**, en laissant le lien s'affaiblir jusqu'à potentiellement s'éteindre lorsque

$$\omega_{k,j}(t) < \omega_{\min}.$$

Dans l'hypothèse où, plus tard, \mathcal{E}_k retrouve une plus grande **fiabilité** (par exemple si un paramètre interne $\mathbf{s}_k(t)$ se recalibre, ou si le flux de données s'assainit), on observe que la **distance** $\| \mathbf{x}_k - \mathbf{x}_j \|^2$ peut redevenir modérée, augmentant $S(\mathcal{E}_k, \mathcal{E}_j)$. La mise à jour $\omega_{k,j}(t+1)$ peut alors reprendre une tendance à la hausse, recréant ou solidifiant des liaisons $\omega_{k,j}$. Cette **réversibilité** illustre l'une des forces du **Synergistic Connection Network**. Une entité qui s'était tenue à l'écart en raison d'un bruit excessif peut retrouver sa **place** dès que ses **observations** redeviennent pertinentes, sans qu'une restructuration globale du réseau soit nécessaire. Elle réintègre alors le **cluster**, démontrant ainsi la **capacité du DSL à s'adapter** en continu aux fluctuations de la qualité des données.

1.5.3.3. Données Partielles : Complétion Progressive via Synergie

Dans de nombreux **scénarios** pratiques, certaines entités ne disposent que d'une **fraction** de leurs attributs. Par exemple, dans un cadre médical, un **patient** peut présenter des **données incomplètes** en raison de tests sanguins manquants, d'une imagerie non réalisée ou d'un historique médical lacunaire. Les approches neuronales classiques exigent souvent une **imputation** ou un **remplissage** préalable des valeurs absentes, ce qui nécessite des hypothèses ou des méthodes de substitution. Il n'est pas rare non plus que l'on décide simplement d'**écartier** les exemples incomplets pour ne pas perturber l'apprentissage.

Dans le **Deep Synergy Learning (DSL)**, une telle entité — par exemple $\mathcal{E}_{\text{patient}}$ — est autorisée à **coopérer** partiellement avec d'autres entités. Si l'entité patient ne possède que certaines **mesures** (tests sanguins, variables cliniques) mais en ignore d'autres, elle peut néanmoins évaluer sa **synergie** avec, par exemple, d'autres patients $\mathcal{E}'_{\text{patient}}$ ou des **variables globales** (moyennes de cohortes, règles médicales) en se basant sur les **dimensions** ou **attributs** qu'elle détient réellement. Une fonction de synergie $S(\mathcal{E}_k, \mathcal{E}_j)$ adaptée aux **dimensions communes** ou à une distance partielle peut alors guider la mise à jour de la pondération :

$$\omega_{k,j}(t+1) = \omega_{k,j}(t) + \eta [S(\mathcal{E}_k, \mathcal{E}_j) - \tau \omega_{k,j}(t)].$$

Si, malgré des données incomplètes, l'entité \mathcal{E}_k dégage une **valeur ajoutée** non négligeable pour le réseau (par exemple, la portion de tests sanguins disponibles est très informative), ses **liaisons** $\omega_{k,j}$ avec d'autres entités \mathcal{E}_j se **consolident** progressivement. Cela peut se traduire par la **constitution** d'un **cluster** rassemblant d'autres patients, ou bien des entités spécialisées (un bloc fonctionnel détectant une tendance clinique). L'**incomplétude** des données n'empêche pas la **coopération**, car seul compte le **gain** mesuré sur les attributs effectivement partagés.

Au fur et à mesure que l'entité \mathcal{E}_k acquiert (ou reconstitue) de nouveaux attributs manquants, il est envisageable que sa **distance** ou sa **similarité** avec d'autres entités évolue, rehaussant la pondération $\omega_{k,j}$. Le réseau en tire un bénéfice, car la réévaluation de la synergie intègre cette **nouvelle information**, facilitant ainsi le **réajustement** des liens. Cela permet d'éviter l'**exclusion** systématique des cas incomplets ou le recours à une **imputation** globale. Au lieu de cela, le **DSL** se contente de **tester la collaboration** sur la partie des données effectivement disponible. Cette logique de **liaison partielle** confère au **DSL** une **souplesse** précieuse pour manipuler des **jeux de données** irréguliers, tout en conservant la capacité d'un **regroupement** (cluster) auto-organisé autour des **similarités** ou **complémentarités** réellement détectées.

1.5.3.4. Exemples Concrets et Bénéfices

Dans un **environnement industriel**, certains capteurs (de température, de pression...) se révèlent souvent **défaillants** ou trop **bruités**. Dans un **Deep Synergy Learning (DSL)**, les entités correspondantes $\mathcal{E}_{\text{temp}}$ ou $\mathcal{E}_{\text{press}}$ voient leurs **pondérations** $\omega_{\text{temp},j}$, $\omega_{\text{press},j}$ diminuer dès lors que la **synergie** avec d'autres capteurs (comme $\mathcal{E}_{\text{vibration}}$ ou $\mathcal{E}_{\text{débit}}$) ne justifie plus la coopération. Le **cluster** principal, dédié par exemple à la **détection d'anomalies**, se recentre naturellement sur les capteurs les plus fiables. Il n'est pas nécessaire d'exclure explicitement les capteurs défaillants, car ceux-ci s'**isolent d'eux-mêmes** lorsque leurs liaisons $\omega_{i,j}$ deviennent insignifiantes. Si, par la

suite, un capteur perturbé se **rétablit** (ou se recale), les pondérations se remettent à monter, lui permettant de **réintégrer** le cluster d'intérêt. Ce **mécanisme** procure une **robustesse** naturelle face aux dérives ponctuelles de certains capteurs.

Dans un autre contexte, celui d'une **base utilisateur** (profilage, recommandation), il est courant que les **profils** ne renseignent pas l'intégralité des informations attendues. En pratique, un **pipeline** neuronal classique suppose souvent une **imputation** des champs manquants ou l'exclusion des données incomplètes. Le **DSL**, au contraire, autorise l'insertion d'entités représentant « Utilisateur U1 » avec un vecteur \mathbf{x}_{U1} partiel, ainsi que d'entités « Contenu C1, C2 » définies par divers attributs. La **synergie** $S(\mathcal{E}_{U1}, \mathcal{E}_{C2})$ tient compte des attributs effectivement présents. Si, même partiellement, l'utilisateur U1 et un contenu C2 trouvent un **recouplement** significatif, la pondération $\omega_{U1,C2}$ s'accroît sans imposer que tous les champs soient renseignés. Le **réseau** s'**auto-structure** donc malgré l'incomplétude, sans avoir à multiplier les procédures d'imputation ou à écarter systématiquement les exemples incomplets. Cela permet de **préserver** une large portion d'information disponible et d'exploiter la **valeur ajoutée** dès qu'elle se présente.

1.5.3.5. Comment le DSL Surpasse la Rigidité des Réseaux Profonds

Les **réseaux neuronaux profonds** présentent une sensibilité notable aux **incohérences** ou aux cas situés hors de la distribution rencontrée lors de l'entraînement. Si, par exemple, un flux de capteurs n'a jamais inclus des observations particulièrement bruitées, le **modèle** peine à gérer ces perturbations lorsqu'elles apparaissent. En revanche, le **Deep Synergy Learning (DSL)** se caractérise par une **auto-régulation**. Lorsqu'une entité \mathcal{E}_k se montre trop incertaine (synergie très basse ou négative avec d'autres entités), les pondérations $\omega_{k,j}(t)$ s'amoindrissent, et la contribution de \mathcal{E}_k se trouve mécaniquement **limitée** ou mise à l'écart, au moins de manière temporaire. Le **SCN** continue malgré tout de fonctionner avec les entités restantes, potentiellement plus fiables.

Dans bien des cas classiques, on recourt à un **modèle d'imputation** (pour combler les données manquantes) ou à des heuristiques rigides visant à traiter le **bruit**. Dans le **DSL**, à l'inverse, chaque entité **évalue** localement la **synergie** possible, sans qu'un algorithme unique de “remplissage” s'impose. Cette démarche revient à laisser chaque **liaison** $\omega_{k,j}$ apprendre la **compatibilité** ou l'**incompatibilité** entre \mathcal{E}_k et \mathcal{E}_j , de sorte que, lorsque le bruit ou l'incohérence survient, la pondération se détériore progressivement, isolant l'entité problématique du **cluster** principal.

Si la **qualité** des données associées à une entité fluctue dans le temps — qu'il s'agisse d'un capteur dont la précision varie ou d'une source intermittente de bruit — la **dynamique** d'auto-organisation gère cette évolution. Les **liaisons** $\omega_{k,j}$ s'étirent ou se resserrent selon la **synergie** présente, et aucune **réinitialisation** globale ni refonte de l'**architecture** n'est requise. Le réseau d'entités conserve sa topologie, ajustant simplement les **connexions** nécessaires. Cette **flexibilité** évite l'extrême rigidité à laquelle aboutit souvent un réseau profond traditionnel, où l'adaptation à de nouveaux modes de bruit ou d'incomplétude implique souvent un **réentraînement** sur un large jeu de données.

1.5.4. Potentiel d'Auto-Évolution et d'Adaptation Continue

Un des aspects les plus novateurs du **Deep Synergy Learning (DSL)** concerne sa capacité à **évoluer** au fil du temps, sans nécessairement repasser par une phase d'entraînement globale et figée. Plutôt que de geler l'architecture après avoir ajusté quelques poids, le DSL propose une dynamique **en continu** où la structure interne (liens synergiques, clusters, etc.) se reconfigure régulièrement. Cette **auto-évolution** rend le réseau apte à faire face à des changements de distribution (domain shift), à l'apparition de nouvelles entités (nouvelles sources de données) ou à la nécessité d'exploiter un flux continu de données (*streaming data*). Dans cette section (1.5.4), nous examinons :

- Les principes d'**auto-évolution** du DSL,
- Comment cette dynamique repose sur l'**adaptation continue** des pondérations (déjà décrite en partie en 1.4.5),
- Les bénéfices en termes de **Lifelong Learning** (apprentissage tout au long de la vie) et de **plasticité** comparable à des systèmes vivants.

1.5.4.1. L'Auto-Organisation comme Moteur d'Évolution

Dans le **Deep Synergy Learning (DSL)**, la **pondération** $\omega_{i,j}(t)$ reliant deux entités \mathcal{E}_i et \mathcal{E}_j suit la règle de mise à jour décrite en section 1.4.5. La pondération se met à jour selon

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où $S(\mathcal{E}_i, \mathcal{E}_j)$ correspond à la **synergie** (distance, similarité, co-information, etc.). Si cette **synergie** demeure suffisamment **positive**, la liaison $\omega_{i,j}$ se **renforce** ; dans le cas contraire, elle **décroît** et peut passer en deçà d'un **seuil** (voir ω_{\min} en section 1.4.5). Un lien sous ce seuil est généralement jugé **inexistant**, reflétant un **désintérêt** ou une **incohérence** mutuelle.

Ce **principe d'évolution dynamique** des connexions fait émerger des **clusters**. Deux entités qui constatent un **gain mutuel** se **rapprochent**, et des sous-groupes peuvent dès lors **fusionner** si la synergie inter-clusters croît au fil des itérations. À l'inverse, un **cluster** se **scinde** quand des sous-groupes décelant davantage de synergie interne choisissent de s'éloigner du groupe initial, leurs connexions inter-groupes ω s'abaissant peu à peu faute de bénéfice collectif.

Ce **mécanisme** confère au **DSL** une **dimension** “vivante” : le **Synergistic Connection Network** évolue en continu, s'**adapte** à l'arrivée de nouvelles données ou à la modification d'une source d'information, tout en **préservant** les clusters confirmés et en **permettant** l'émergence de combinaisons inédites. À aucun moment, un **réentraînement** exhaustif du réseau n'est nécessaire. La **dynamique locale** des pondérations $\{\omega_{i,j}\}$ permet une **réorganisation progressive**, en ajustant les **liaisons** selon l'opportunité détectée dans les données, que ce soit pour renforcer des connexions existantes ou en dissoudre d'autres.

1.5.4.2. Lifelong Learning et Intégration de Nouvelles Entités

Dans de nombreux environnements dynamiques (robotique, systèmes d'information évolutifs, etc.), il est courant de rencontrer des **données** ou des **entités** inédites au fil du temps. Les réseaux neuronaux profonds classiques rencontrent plusieurs difficultés dans ce contexte. L'ajout d'une nouvelle fonctionnalité peut nécessiter la création d'un **module annexe** et un **réentraînement**, parfois complet ou via un **fine-tuning**, ce qui comporte le risque de perturber l'**équilibre établi** du modèle. Il arrive aussi que l'on doive réorganiser l'architecture en profondeur, ce qui affecte la stabilité du modèle.

Le **Deep Synergy Learning (DSL)** propose une alternative plus organique. Pour incorporer une **entité $\mathcal{E}_{\text{nouvelle}}$** correspondant à un nouveau flux de données, on se contente de l'**introduire** dans le graphe existant. Les pondérations la reliant à d'autres entités $\{\omega_{\text{nouvelle},i}\}$ s'ajustent alors en fonction de la synergie détectée. Si une **coopération** apparaît avantageuse, ces liens augmentent et conduisent à l'**insertion** naturelle de l'entité dans un cluster. Il n'est donc pas nécessaire de reconfigurer l'ensemble de l'architecture, car seules les **connexions pertinentes** se renforcent de manière autonome.

Ce mécanisme permet une **évolution progressive**. Les **clusters** déjà établis maintiennent leurs **liaisons** tant qu'ils apportent un **gain**, garantissant ainsi la stabilité de leurs synergies internes. Le réseau n'est pas obligé de sacrifier ses acquis pour introduire de nouvelles entités. Ainsi, on **atténue** le phénomène de "catastrophic forgetting" caractéristique des modèles neuronaux, lesquels risquent d'effacer de vieilles connaissances lorsqu'on leur enseigne de nouvelles tâches. Le **DSL**, au lieu de **remplacer** brutalement un jeu de poids par un autre, **réorganise** progressivement les connexions de manière **adaptative locale**. Il conserve la pertinence des **clusters existants** tout en intégrant de nouvelles entités au réseau lorsqu'elles apportent une **valeur ajoutée**.

1.5.4.3. Adaptation Continue aux Changements de Distribution

Dans un **réseau** neuronal traditionnel, lorsqu'apparaît un **changement de distribution** dans les données (par exemple l'évolution des conditions de capteurs ou l'apparition de nouvelles caractéristiques), il est souvent indispensable de **réentraîner** ou de **fine-tuner** le modèle, ce qui peut provoquer un ajustement délicat, risqué pour la stabilité ou la précision déjà acquises. En **Deep Synergy Learning (DSL)**, la situation se gère plus souplement grâce à la **dynamique auto-organisée** qui régit la formation et la déliaison des connexions.

Au fil du temps, la **synergie** entre entités se réévalue en continu. Si un groupe de capteurs cesse de fournir une information fiable, les pondérations associées baissent (faible synergie), déconnectant progressivement l'entité devenue obsolète de la structure ; à l'inverse, si de nouvelles variables ou de nouvelles relations s'avèrent profitables, elles consolident leurs liaisons. Le réseau se remodelle donc de manière **locale**, sans nécessiter la rétropropagation globale ni un réapprentissage exhaustif.

Cette **localité** de la mise à jour, où chaque entité s'occupe prioritairement de ses **liens directs** (et s'ajuste indirectement via les liaisons d'autres entités), confère au **SCN** une forme de **dynamique distribuée**. Les noeuds décident eux-mêmes de renforcer ou d'affaiblir leurs coopérations, suivant l'intérêt (synergie) rencontré, sans orchestration centrale. Le système **s'adapte** ainsi de manière fluide, maintenant une relative **stabilité** pour les groupes demeurant utiles, tout en acceptant

l'insertion ou la réduction de connexions pour refléter l'évolution réelle de la **distribution** des données.

1.5.4.4. Comparaison avec l'Architecture Figée d'un Réseau Profond

Dans la majorité des **réseaux neuronaux** classiques (CNN, RNN, MLP, Transformer), l'entraînement s'effectue via **rétropropagation** sur un **ensemble de données** donné a priori, aboutissant à un **modèle** fini. Lorsque la distribution des données évolue ou qu'une nouvelle tâche survient, on se retrouve souvent contraint de réentraîner ou de fine-tuner l'architecture, avec un **risque** d'oubli des acquis précédents (catastrophic forgetting) et un **coût** en calcul potentiellement élevé. Cette rigidité découle du fait que les **couches** et la **connectivité** du réseau restent invariables. Les structures telles que les **couches convolutionnelles** ou les **mécanismes de self-attention** demeurent fixes, sans mécanisme natif permettant d'**ajouter** ou de **supprimer** dynamiquement des neurones ou des flux de données.

Le **Deep Synergy Learning (DSL)** propose une philosophie différente. La structure du **Synergistic Connection Network** est en **évolution continue**. Les entités \mathcal{E}_i peuvent mettre à jour leurs **représentations** internes \mathbf{x}_i , \mathbf{s}_i et les **liaisons** $\omega_{i,j}$ s'ajustent (ou se dissolvent) selon le niveau de **synergie**. On peut donc introduire une **nouvelle entité** lorsqu'un flux de données inédits émerge, ou en ôter une qui ne sert plus, sans devoir **réentraîner** le système dans son intégralité. L'**apprentissage** n'est pas un processus qui se clôt : le **réseau** conserve un état de plasticité lui permettant de **s'adapter** en continu aux transformations de l'environnement. Cette approche relève d'un véritable **lifelong learning**, affectant aussi bien les **paramètres** que la **topologie** du SCN. C'est précisément cette capacité de reconfiguration qui différencie la souplesse du **DSL** de la rigidité d'un réseau profond figé, et qui évite de longues phases de réapprentissage global ou de coûteux correctifs pour incorporer de nouveaux concepts ou signaux.

1.5.4.5. Exemples d'Application Pratique

Il existe plusieurs scénarios où l'**approche de Deep Synergy Learning (DSL)**, avec son **réseau** auto-organisé et ses liens **adaptatifs**, se révèle particulièrement puissante pour faciliter un **apprentissage continu**.

Dans un **contexte de robotique autonome**, l'environnement et les capteurs évoluent en permanence. La luminosité peut varier, de nouveaux obstacles apparaissent, ou un capteur supplémentaire, comme une caméra ou un lidar, est ajouté. Dans un **SCN**, chaque capteur ou module de traitement est représenté par une **entité**. Lorsqu'un capteur devient peu fiable en raison du bruit ou d'une altération de ses mesures, ses connexions ω s'affaiblissent naturellement, réduisant ainsi son influence. À l'inverse, lorsqu'un nouveau dispositif est introduit, il suffit de l'ajouter en tant que **nouvelle entité** $\mathcal{E}_{nouvelle}$, et la **synergie** avec les autres modules s'évalue automatiquement. Sans reconstruire l'architecture globale, le **DSL** réorganise les connexions $\{\omega_{i,j}\}$ et **réalloue** l'importance de chaque capteur en fonction de sa **fiabilité** et de sa **pertinence** pour la tâche en cours.

Dans un **système de recommandation évolutif**, l'ajout constant de nouveaux utilisateurs et de nouveaux contenus (films, livres, produits...) constitue un défi pour les approches classiques. Les méthodes traditionnelles, comme le **collaborative filtering**, nécessitent souvent un **réapprentissage** périodique ou un recalculation complet des similarités. Avec un **DSL**, chaque nouvel utilisateur ou nouvel objet est simplement **intégré au réseau** en tant qu'**entité \mathcal{E}_{new}** , tandis que la dynamique de **synergie** ajuste automatiquement les connexions $\omega_{\text{new}, \dots}$. Les liaisons avec les utilisateurs ou contenus similaires se renforcent progressivement, créant un **cluster** autour de la nouvelle entité. Le recalculation reste **local**, permettant au **réseau** d'évoluer en continu sans nécessiter un **réentraînement global**.

Dans l'**analyse contextuelle**, comme l'étude de flux de données urbaines ou sociales, la **distribution** des données varie en fonction du temps. Un réseau traditionnel devrait prévoir différents modèles pour gérer ces changements ou recalculer ses poids à intervalles réguliers. Dans un **SCN**, les **clusters** pertinents se forment pour un contexte donné et se réactualisent lorsque la situation évolue. Lors d'un passage du **jour** à la **nuit** ou d'une transition entre **saisons**, certains sous-groupes se dissolvent si la synergie disparaît, tandis que d'autres émergent. Le **DSL** ajuste ainsi les connexions en **temps réel**, permettant de commuter fluidement d'un **cluster** à l'autre sans nécessiter un **réentraînement coûteux**.

1.5.5. Réduction de la Dépendance à la Supervision Humaine

Au-delà de la multi-modalité (1.5.2), de la gestion des données incomplètes (1.5.3) et de l'adaptation continue (1.5.4), un autre avantage essentiel du **Deep Synergy Learning (DSL)** réside dans sa **relative autonomie** vis-à-vis d'une supervision humaine intensive. Les méthodes traditionnelles de l'IA, notamment les réseaux neuronaux profonds, reposent souvent sur l'accès à d'importants **jeux de données labellisés**, et requièrent un entraînement supervisé (ou semi-supervisé) afin d'ajuster des poids internes. L'annotation des données ou la configuration d'un réseau supervisé peut s'avérer **coûteuse** et **chronophage**, rendant difficile l'adaptation à des environnements non balisés. De plus, dans de nombreux contextes réels, la **disponibilité des labels** est limitée, et un apprentissage plus **auto-dirigé** devient une nécessité, notamment face à des volumes massifs de **big data non étiqueté** ou à des environnements inconnus.

Le **DSL**, en favorisant des **relations auto-organisées** entre entités, intègre des mécanismes permettant de réduire la dépendance aux **étiquettes externes**. Plutôt que d'être guidé exclusivement par des labels, il s'**auto-structure** en fonction des synergies détectées, ce qui diminue la nécessité d'une supervision stricte. Cette capacité offre des perspectives intéressantes pour l'**IA autonome** et l'**apprentissage non supervisé**, où l'identification des relations internes entre entités permet de révéler des structures pertinentes sans intervention humaine.

1.5.5.1. L'Auto-Organisation sans Label

Dans les paradigmes d'**apprentissage supervisé**, on dispose en général pour chaque donnée \mathbf{x} d'un **label** y (classe, valeur numérique à prédire), et l'entraînement consiste à **minimiser** une fonction de coût $\mathcal{L}(f_\theta(\mathbf{x}), y)$. On **ajuste** alors les poids du réseau neuronal pour améliorer la correspondance entre l'entrée \mathbf{x} et la sortie souhaitée y .

Dans le **Deep Synergy Learning (DSL)**, la dynamique s'appuie sur la **synergie** entre paires d'entités \mathcal{E}_i et \mathcal{E}_j . On définit une quantité $S(\mathcal{E}_i, \mathcal{E}_j)$ qui mesure le **gain** (ou la complémentarité) qu'elles retirent de leur coopération. Cette fonction de synergie peut reposer sur une **distance** ou une **similarité**, sur la **co-information** en théorie de l'information, ou encore sur tout **critère** mesurant l'utilité mutuelle. Dans un **réseau** entièrement non supervisé, aucune annotation externe n'est disponible. La **structure** se forme alors naturellement à travers les **interactions** entre entités, qui s'**agrègent** ou se **séparent** en fonction de leur synergie locale. Un **objectif** global, le cas échéant, peut exister si un signal partiel ou une contrainte externe oriente la cohérence, mais il n'est pas obligatoire. L'auto-organisation résulte de la **conjonction** des ajustements locaux, sans référence à un label y .

Cette logique se reflète dans la **formation** de clusters (section 1.4.3). Les entités se regroupent par **renforcement** de leurs connexions si elles détectent un **gain**, et se dissocient sinon. Il n'est pas nécessaire de leur **attribuer** des étiquettes telles que "classe C1" ou "classe C2". Elles se regroupent **spontanément**, sans supervision, en fonction de la **synergie perçue**, formant ainsi des structures auto-organisées. Le processus aboutit à un **clustering** naturel, que l'on peut ensuite interpréter comme un groupement de similarités ou de dépendances, et qui peut se révéler **faiblement supervisé** si l'on dispose, en parallèle, de quelques **signaux** ou **contraintes** externes. Dans ce cadre, le **DSL** encourage un **apprentissage local** des coopérations, structurant le **Synergistic Connection Network** sans recourir à un label global pour chaque exemple.

1.5.5.2. Quand un Signal de Supervision Existe...

Le **Deep Synergy Learning (DSL)** n'exclut pas l'éventualité de **labels** ou d'**objectifs** supervisés. Au contraire, il est tout à fait envisageable de disposer d'un **objectif** supervisé pour prédire un label y , tout en conservant le mécanisme d'**auto-organisation** qui ajuste les liens $\omega_{i,j}$ en fonction de la synergie. Dans ce scénario, le **DSL** cohabite avec un algorithme de rétropropagation ou d'autre forme de supervision, mais n'est pas entièrement asservi à cet unique signal étiqueté.

Concrètement, on peut imaginer qu'un **objectif** tel que la minimisation d'une fonction de perte $\mathcal{L}(f_\theta(\mathbf{x}), y)$ se déroule en parallèle de la **dynamique** locale guidée par la synergie. Les **liaisons** $\omega_{i,j}(t)$ poursuivent la règle adaptative (section 1.4.5), tandis que le **label** y n'intervient que pour évaluer la cohérence globale ou orienter certains choix de structure. Une partie du réseau peut ainsi se spécialiser dans la tâche supervisée, la **synergie** jouant un rôle déterminant dans l'**organisation** des entités restantes.

Cette architecture présente un **double** bénéfice. D'abord, l'**auto-organisation** ne repose pas exclusivement sur le label y , ce qui permet un fonctionnement **semi-supervisé** ou même **non supervisé**. La majorité des liaisons $\omega_{i,j}$ se forment en fonction de la **coopération** intrinsèque entre entités, tandis que l'objectif supervisé, s'il est présent, n'intervient qu'en **complément**. Ensuite, si le nombre de labels est **limité**, l'apprentissage reste largement alimenté par la **synergie** interne, évitant la nécessité de disposer d'un jeu de données richement annoté. Le **réseau** peut ainsi, en un premier temps, **structurer** ses entités par pure auto-organisation, puis tirer parti de quelques **labels** additionnels pour affiner la prédiction d'une variable ou la détection d'anomalies.

L'important est que les **labels**, quand ils existent, ne contrôlent pas toutes les **liaisons** du réseau. Seule une fraction des entités ou des connexions peut être orientée par la performance supervisée, tandis que les **clusters** naissent en grande partie de la **synergie** (sections 1.4.3 et 1.4.4). Le réseau se montre donc apte à **apprendre** et **s'adapter** via son auto-organisation, puis à **exploiter** un signal supervisé pour ajuster un module final ou un sous-ensemble d'entités, évitant ainsi la contrainte d'une rétropropagation exhaustive sur toute l'architecture.

1.5.5.3. Exemple : Découverte Spontanée de Catégories

Pour illustrer la manière dont un **Deep Synergy Learning (DSL)** peut structurer les données **sans** s'appuyer sur des labels explicites, on peut considérer un ensemble de **documents** non annotés. Chaque document est alors représenté par un **embedding** ou un **bag-of-words**. Dans le **Synergistic Connection Network**, on introduit autant d'**entités** \mathcal{E}_{doc} que de documents, en laissant le mécanisme de **synergie** déterminer comment ces entités se relient.

Les entités commencent par **évaluer** leur proximité ou leur complémentarité. Par exemple, des vecteurs \mathbf{x}_{doc} représentant des documents partageant des thèmes communs, un vocabulaire similaire ou des distributions de topics proches peuvent manifester une **synergie** significative. Les pondérations $\omega_{i,j}$ entre entités se renforcent si la **similarité** est jugée élevée, ou si la co-information (section 1.4.4) révèle un **gain** dans la mise en commun de leurs contenus. Progressivement, un ou plusieurs **clusters** apparaissent, chaque cluster rassemblant des documents sur des sujets proches, sans qu'on ait besoin de spécifier de labels ou de classes. Ainsi, certains regroupements peuvent porter sur la **politique**, d'autres sur le **sport**, d'autres encore sur la **santé**, mais rien n'impose que le DSL les nomme ou les identifie explicitement.

Si, plus tard, un **expert** fournit quelques **labels** indiquant, par exemple, qu'un certain cluster correspond au domaine sportif, on n'a pas besoin d'annoter l'**intégralité** des documents. Le **DSL** s'étant déjà **auto-organisé** en clusters cohérents, il suffit d'**attribuer** une étiquette à la communauté d'entités concernée une fois la structure formée. Ce procédé montre comment un système **non supervisé** ou **faiblement supervisé** peut accomplir une **structuration** fine des données, sur laquelle un label partiel ne fait qu'ajouter un niveau de **description** plus explicite. Les sections antérieures (1.4.3, 1.5.5.1 et 1.5.5.2) soulignent ainsi la flexibilité du DSL pour fonctionner dans des contextes sans labels, tout en permettant une intégration de signaux supervisés quand ils existent.

1.5.5.4. Vers une IA plus Autonome

Les **interactions indirectes** (discutées en section 1.4.6) font en sorte qu'une entité peut être **influencée** par une autre, même en l'absence de lien direct, via des chemins de coopération au sein du **Synergistic Connection Network**. Ce mécanisme enrichit le potentiel d'**auto-découverte** de catégories ou de concepts de manière **non supervisée**, puisque l'information peut circuler librement à travers divers **chemins**, et les **clusters** émergent en valorisant l'**intégralité** des signaux disponibles.

Dans une perspective de **IA** plus générale, appelée à affronter des **domaines inconnus** et des **contextes** évolutifs, la **réduction** du besoin de supervision revêt un caractère décisif. Le **Deep Synergy Learning (DSL)**, en misant sur un **apprentissage local** et **continu**, se montre apte à fonctionner avec des annotations partielles ou inexistantes, tout en conservant la capacité de **structurer** et **organiser** les entités. Il s'agit d'une étape vers une IA plus autonome, qui, sans nécessité d'un étiquetage exhaustif ou de guides fortement supervisés, s'adapte aux environnements **faiblement annotés** (voire non annotés) et continue à explorer la structure interne des données pour faire émerger **concepts** et **catégories** spontanément.

1.5.5.5. Limites et Solutions Partielles

Le **Deep Synergy Learning (DSL)** ne s'appuie pas massivement sur les **labels**, ce qui lui confère une grande liberté dans la formation des **clusters** et dans la découverte de structures internes. Néanmoins, un **petit** signal externe, issu par exemple d'une **semi-supervision**, peut être requis lorsque l'on souhaite orienter la **dynamique** vers des objectifs spécifiques. Sans cette impulsion, le réseau se contente de former des groupes qui font sens pour les données elles-mêmes, sans garantir qu'ils correspondent aux besoins d'une application particulière. Il peut arriver que les clusters, très cohérents sous un angle statistique, ne concordent pas avec la segmentation ou les catégories souhaitées par l'utilisateur.

L'**auto-organisation** peut en effet aboutir à des **partitions** intrinsèquement pertinentes, mais qui ne recouvrent pas exactement les **objectifs** fixés en pratique. Pour rectifier ce phénomène, il est possible d'introduire un **minimum** de **contraintes** ou de **labels** annotés, lesquels jouent le rôle de **pénalités** ou de **feedback** dans la mise à jour des liaisons $\{\omega_{i,j}(t)\}$. De la sorte, le **DSL** ne se limite plus à la seule synergie intrinsèque, mais tient également compte de la finalité appliquée (une tâche métier, une classification imposée), ce qui oriente plus étroitement la **constitution** des clusters.

La **complexité** du calcul de la synergie, surtout pour des données brutes très dimensionnelles, constitue un autre défi. Dans certains cas, on envisage des mesures plus poussées telles que la **co-information** ou la **Partial Information Decomposition** (sections 1.4.4 et 1.4.7), dont l'estimation peut s'avérer lourde. Des **estimateurs** non paramétriques requièrent souvent un échantillonnage important et des ressources conséquentes. Il est alors crucial de se doter de **stratégies** d'approximation ou d'**échantillonnage** (sampling parcimonieux, heuristiques de calcul) pour maintenir la faisabilité sur des volumes de données massifs (big data). Cette solution, certes partielle, demeure incontournable si l'on veut étendre le **DSL** à des scénarios industriels exigeants, tout en préservant un **coût** de calcul acceptable.

1.5.6. *Création de Représentations Riches et plus Interprétables*

En plus de la **réduction de la dépendance à la supervision** (1.5.5), le **Deep Synergy Learning (DSL)** présente un avantage notable pour la **création de représentations** à la fois **riches** et **interprétables**. Contrairement aux réseaux neuronaux profonds classiques, qui produisent souvent des représentations “boîte noire” difficilement explicables, le **DSL** mise sur l'**auto-organisation** et la **coopération** entre entités d'information, ce qui peut aboutir à des structures internes (clusters,

macro-clusters, liens synergiques) plus compréhensibles pour un humain. Cette section (1.5.6) met en évidence :

- **Pourquoi** la synergie favorise l'émergence de **représentations complexes**,
- Comment le principe de **clusters** ou de **macro-clusters** rend l'organisation plus **lisible**,
- En quoi le DSL peut faciliter l'**interprétabilité** par rapport aux architectures profondes traditionnelles.

1.5.6.1. Logique “par Entités” plutôt que “en Couches Opaques”

Dans les **réseaux neuronaux profonds** (CNN, RNN, Transformers...), la **représentation interne** repose sur une succession de **couches** appliquant des transformations non linéaires. Malgré certaines avancées en matière d'**interprétabilité**, comme la visualisation des **filtres** dans les CNN ou l'analyse des **matrices d'attention** dans les Transformers, comprendre comment le réseau parvient à une décision globale reste un défi majeur.

L'une des principales limitations réside dans l'**absence de structuration explicite** des poids. Ceux-ci ne sont pas naturellement regroupés sous forme de **blocs lisibles**, ce qui complique l'identification de sous-ensembles de paramètres correspondant à des **concept**s distincts. À l'exception d'un neurone ou d'une couche spécifique, il est souvent difficile de relier une partie du réseau à une interprétation claire de son rôle.

Les méthodes d'**interprétabilité** existantes, comme **Grad-CAM** ou **LIME**, tentent de pallier cette opacité en mettant en évidence l'influence de certaines **caractéristiques locales** (pixels, dimensions spécifiques des embeddings). Toutefois, ces techniques reposent généralement sur des **heuristiques** qui, bien qu'utiles pour identifier des tendances, ne garantissent pas une **compréhension globale** de la logique interne du réseau.

Le **Deep Synergy Learning (DSL)** adopte une approche fondamentalement différente en organisant l'information autour d'**entités**, qui constituent les **nœuds** du **Synergistic Connection Network (SCN)**. Contrairement aux réseaux neuronaux traditionnels où les poids sont répartis sans structuration explicite, le **DSL** permet de **modéliser** chaque entité comme un **flux sensoriel** (image, son, texte) ou un **descripteur abstrait** (embedding, concept partiel).

Les **liaisons** entre entités sont définies par la **synergie** détectée, c'est-à-dire la **plus-value mutuelle** qu'elles retirent de leur coopération. Cela transforme la représentation en un **graphe modulaire**, où l'on peut **visualiser** les **interactions clés** et observer comment différentes entités collaborent. Contrairement aux architectures classiques, où l'information circule de manière séquentielle à travers des **couches opaques**, le **DSL** permet une **organisation plus flexible**, où les **clusters** émergent de manière dynamique en fonction des **relations synergiques**.

Cette structuration améliore la **modularité** et la **lisibilité** du réseau. Par exemple, un **cluster** peut regrouper la **détection de visages** (vision), l'**audio de voix** (son) et un **module de reconnaissance textuelle** (analyse des mots-clés parlés), formant ainsi une unité cohérente qui associe plusieurs modalités de manière fluide. Contrairement à l'**empilement rigide** des couches dans les

architectures classiques, cette approche permet une **segmentation plus fine** des interactions et une **interprétabilité accrue**.

Grâce à cette conception **basée sur les entités**, le **SCN** offre une représentation plus **transparente**. Il devient possible **d'identifier précisément** les **entités en présence**, de connaître leurs **rôles spécifiques**, et de comprendre comment elles s'**agencent** via les **pondérations de synergie**. Le **DSL** apporte ainsi un degré supplémentaire d'**explicabilité**, en offrant une **cartographie des flux d'information**, bien plus lisible que l'accumulation complexe et parfois opaque de poids dans un réseau neuronal profond traditionnel.

1.5.6.2. Clusters et Macro-Clusters comme Briques de Sens

Dans le **Deep Synergy Learning (DSL)**, la formation d'un **cluster** d'entités traduit souvent l'émergence d'un **concept** ou d'un **thème** sous-jacent aux données. Par exemple, un **cluster** multimodal composé d'entités liées à l'image, à l'audio et au texte pourrait se spécialiser dans la détection d'un **événement conférence**. L'entité image capterait une scène où un orateur fait face à un public, l'entité audio identifierait des bruits de voix ou des applaudissements, et l'entité texte reconnaîtrait des termes comme "bienvenue" ou "question". Cette **cohérence** ne se réduit pas à un simple ajustement de poids numériques. Elle permet de comprendre quelles entités s'agrègent et pour quelle raison, grâce à l'évaluation locale de la **synergie**, ce qui améliore la **lisibilité** du modèle.

D'un point de vue **explicatif**, un **cluster** dans le **Synergistic Connection Network (SCN)** ne se limite pas à un agrégat opaque de neurones, comme dans un réseau profond traditionnel. Il s'apparente plutôt à un **sous-graphe** composé d'entités distinctes, qu'il s'agisse de capteurs visuels, de flux audio, de modules textuels ou encore de représentations symboliques. Les **liaisons synergiques** entre ces entités reflètent la force de leur coopération et permettent d'analyser la distribution de la **co-information** ou de la **similarité** entre elles, rendant le processus plus transparent.

Lorsque plusieurs **clusters** se renforcent mutuellement, ils peuvent se **fusionner** pour former un **macro-cluster**. Cette dynamique révèle que des groupes d'entités s'organisent en structures plus larges et interconnectées. Dans un **contexte e-commerce**, un macro-cluster pourrait ainsi regrouper plusieurs **catégories de produits**, des **groupes d'utilisateurs** et des **attributs de profil**, mettant en évidence un **sous-réseau dynamique** où la synergie favorise des interactions complexes et évolutives.

Ces **briques** constituées de **clusters** et de **macro-clusters** forment des **modules interprétables** que l'on peut analyser et commenter. Chaque entité, en raison de son association avec un **flux de données** ou un **vecteur caractéristique**, possède un **ancrage explicite** qui clarifie son rôle au sein du réseau. Les **liaisons synergiques** révèlent la proportion de **coopération interne**, tandis que le **DSL** segmente naturellement le graphe global en plusieurs **sous-ensembles** structurés autour de **concepts** ou de **thématisques**.

Cette granularité confère au modèle une **modularité accrue**, facilitant une analyse plus fine de son organisation. L'approche permet d'examiner directement les **relations** et les **interactions** au sein du réseau, offrant ainsi une compréhension plus accessible que l'interprétation classique des poids dans une **couche neuronale abstraite**. Grâce à cette structuration, il devient possible d'identifier

des justifications tangibles sur le fonctionnement du système et d’expliquer plus clairement les décisions qu’il génère.

1.5.6.3. Représentations Multiples et Non Linéaires

Le **Deep Synergy Learning (DSL)** ne se borne pas à des représentations linéaires ou statiques. Au contraire, chaque **entité** \mathcal{E}_i peut conserver une **représentation interne** $\mathbf{x}_i(t)$ ou $\mathbf{s}_i(t)$ qui évolue au fil du temps, ou être associée à un petit réseau neuronal local (par exemple un autoencodeur) calculant

$$\mathbf{x}_i(t+1) = F_i(\mathbf{x}_i(t), \{\omega_{i,k}(t)\}, \dots).$$

La **non-linéarité** inhérente à ce type de module (RNN local, MLP, autoencodeur) élargit considérablement l'espace des **représentations** envisageables, comparativement à un simple cadre vectoriel fixe. Les entités peuvent ainsi affiner leurs **features** internes ou leurs **états** \mathbf{s}_i , tout en mettant à jour leurs **liaisons** $\omega_{i,j}(t)$ d'après la **synergie** nouvellement constatée. Cette approche ne repose pas sur un pipeline rigide. Chaque entité, conçue comme un **bloc fonctionnel autonome**, conserve sa propre dynamique d'évolution. Pendant ce temps, la **synergie** entre les entités se réévalue localement en fonction des interactions et des changements contextuels. Cette flexibilité permet au système d'adapter continuellement ses connexions, sans nécessiter une reconfiguration globale imposée.

Si deux entités en viennent à **coïncider** fortement (en partageant des représentations très similaires ou très complémentaires), leurs pondérations $\omega_{i,j}$ peuvent atteindre un niveau **saturé**, conduisant potentiellement à une **fusion** (au sens de la fusion de clusters, section 1.4.3). Inversement, d'autres entités peuvent demeurer **indépendantes**, préférant garder leurs liaisons minimales. Le résultat global est une **représentation** dite “multi-foyers”, où plusieurs **clusters** se spécialisent dans des sous-ensembles de la tâche ou des données. Chacun possède son propre noyau fonctionnel, et ils coopèrent occasionnellement si la **synergie** l'exige. Cette organisation **modulaire** tranche avec un réseau profond monolithique dans lequel toutes les informations finissent entremêlées au sein de couches successives. Elle favorise un **équilibre** entre la **richesse** (via la non-linéarité) et la **lisibilité** (chaque entité reste limitée à un bloc fonctionnel clairement cerné), tout en laissant l'auto-organisation diriger la consolidation ou la séparation des diverses composantes.

1.5.6.4. Comparaison avec l'Explicabilité dans les Réseaux Profonds

Dans les approches classiques d'**explainability** appliquées aux réseaux neuronaux profonds (CNN, Transformers, etc.), il est habituel de recourir à des méthodes “post-hoc”, comme la visualisation d'**activation maps**, l'explication locale (LIME, SHAP) ou l'examen des **attention maps** dans un Transformer. Ces techniques fournissent un aperçu de la raison pour laquelle le réseau met en avant tel pixel, telle dimension ou tel token, mais elles ne suppriment pas pour autant le caractère opaque de la structure interne. Un CNN reste un empilement de couches, au sein desquelles la signification des poids demeure largement cachée, et un Transformer recèle une superposition de blocs d'attention dont la lisibilité globale n'est pas toujours aisée.

Le Deep Synergy Learning (**DSL**) propose une **explication intrinsèque** plus claire en structurant l'information sous forme d'entités reliées par des **liaisons synergiques** $\omega_{i,j}$. Ces connexions traduisent la coopération effective entre les entités, ce qui permet d'obtenir un **graphe lisible** où apparaissent naturellement des **clusters** \mathcal{C} et des **macro-clusters** représentant des groupes d'entités en interaction. Chaque entité conserve une **identité distincte**, qu'elle soit liée à une modalité spécifique, à un bloc fonctionnel ou à un concept partiel.

L'utilisateur peut examiner un **cluster** \mathcal{C} pour identifier les entités qui le composent, comprendre leurs caractéristiques et analyser la distribution des pondérations $\omega_{i,j}$. Cette structure permet d'interpréter la **logique d'assemblage** des sous-groupes formés. Par exemple, un sous-graphe associant la détection de panneaux de signalisation, l'enregistrement de la vitesse et le suivi de la trajectoire montre clairement la raison d'être de cette coopération à travers la synergie mesurée par les valeurs de $\omega_{i,j}$.

Cette transparence contraste avec les réseaux neuronaux profonds traditionnels, où les ensembles de neurones internes restent difficiles à interpréter sans recourir à des méthodes explicatives externes. Le **DSL** rend ces connexions **explicites et modulables**, offrant une vision claire des interactions et des coopérations à travers les **pondérations synergiques** $\omega_{i,j}$ du **Synergistic Connection Network**.

1.5.6.5. Exemple d'Application : Représentation Sémantique Évolutive

Dans un **agent conversationnel**, plusieurs **entités** interagissent en fonction des volets de l'échange. Une entité $\mathcal{E}_{\text{linguistique}}$ se charge de la compréhension du langage, tandis qu'une entité $\mathcal{E}_{\text{contexte}}$ gère les informations externes comme l'heure, le lieu ou l'historique de conversation. En parallèle, une entité $\mathcal{E}_{\text{profil utilisateur}}$ reflète les préférences et les habitudes du locuteur.

Au fil des interactions, les synergies entre ces entités évoluent : lorsque $\mathcal{E}_{\text{profil}}$ apporte des indices alignés avec $\mathcal{E}_{\text{linguistique}}$, les pondérations $\omega_{i,j}$ entre ces composants augmentent. Le **DSL** favorise alors la formation d'un **cluster** $\mathcal{C} = \{\mathcal{E}_{\text{ling}}, \mathcal{E}_{\text{context}}, \mathcal{E}_{\text{profil}}\}$, où la coopération entre ces entités devient explicite à travers l'évolution des liens $\omega_{i,j}$.

Ce processus conduit à l'émergence d'un **sous-modèle** sémantique organisé autour des thèmes, des intentions et des préférences récurrentes du locuteur. En analysant la structure du **Synergistic Connection Network**, il devient possible d'identifier les **topics** dominants et de comprendre comment ces synergies ont émergé au sein du réseau.

Dans un autre registre, l'**analyse d'images documentaires** bénéficie également du **DSL** grâce à la structuration des informations en **entités** interconnectées. Certaines entités $\mathcal{E}_{\text{patch}}$ décrivent des patchs ou segments visuels, représentant des éléments d'architecture ou des styles picturaux. D'autres entités regroupent des **mots-clés** extraits des légendes ou des métadonnées associées aux images, tandis qu'un troisième type d'entités capture des **concept**s plus abstraits liés à une époque, un courant artistique ou un auteur.

Lorsque ces entités présentent une synergie significative, un **cluster** émerge spontanément. Par exemple, un patch visuel $\mathcal{E}_{\text{patch}}$ contenant une structure architecturale gothique peut se lier à un

ensemble de mots-clés \mathcal{E}_{mot} décrivant des éléments médiévaux, ainsi qu'à une entité conceptuelle $\mathcal{E}_{\text{concept}}$ indiquant la période historique concernée. L'auto-organisation du **Synergistic Connection Network** révèle alors une **co-information** élevée : les motifs visuels, les mots-clés et l'ancrage conceptuel convergent vers une interprétation unifiée.

Cette **coopération** entre entités confère une **lisibilité accrue** : sans imposer de règles prédefinies, le réseau identifie naturellement un regroupement signifiant « document archéologique médiéval ». Le **DSL** agit ainsi comme un **pilier** sémantique dynamique, renforçant les connexions pertinentes lorsqu'un bénéfice mutuel est détecté et permettant une **représentation** riche, évolutive et modulable.

1.5.7. *Intégration de Dimensions Symboliques ou Cognitives*

Les sections précédentes (1.5.1 à 1.5.6) ont souligné diverses forces du **Deep Synergy Learning (DSL)**, telles que la **capacité à gérer la multi-modalité**, à s'**adapter** en continu, ou à **créer** des représentations plus **lisibles**. Un autre point capital, souvent abordé dans l'IA contemporaine, est la possibilité de **mélanger** des composantes **sub-symboliques** (issues des méthodes connexionnistes) et des composantes **symboliques** (notions logiques, sémantiques, règles expertes). Alors que les réseaux neuronaux profonds “classiques” ont généralement peiné à intégrer ces dimensions symboliques, le **DSL**, par sa structure en entités et liens synergiques, peut être un **cadre** propice pour unir symbolique et sub-symbolique.

Cette section (1.5.7) explique :

- Comment le DSL autorise l'**introduction** d'entités à **portée symbolique** (règles, concepts, modèles cognitifs),
- Pourquoi cette intégration est **plus fluide** que dans un réseau hiérarchique figé,
- Les **applications** possibles (raisonnement, inférence, cognition proche de l'humain),
- Les **défis** mathématiques associés à la cohabitation de ces deux modes de représentation.

1.5.7.1. *Aperçu des Approches Symboliques et Connexionnistes en IA*

L'**intelligence artificielle** repose historiquement sur deux grandes approches aux fondements distincts. L'**IA symbolique** s'appuie sur des **règles logiques** et des **moteurs d'inférence**, permettant de manipuler des symboles et d'expliquer chaque décision par une suite de règles compréhensibles. Cette approche garantit une **transparence** et une **cohérence** dans le raisonnement, ce qui facilite son interprétabilité. Toutefois, elle se heurte à des limites face aux **données bruitées** et aux environnements évolutifs, car elle peine à s'adapter et à extraire des **motifs complexes** directement depuis des observations non structurées.

À l'opposé, l'**approche connexioniste** repose sur l'apprentissage automatique et l'ajustement dynamique des **paramètres internes** des **réseaux neuronaux**. Ces modèles sont capables de détecter des régularités dans de vastes ensembles de données et d'apprendre sans qu'un programme

explicite ne dicte leurs règles. Cette plasticité leur confère une grande efficacité dans des tâches comme la reconnaissance d'images ou la compréhension du langage naturel. Toutefois, cette approche souffre d'un problème majeur. Les décisions émergent d'un **réseau dense de pondérations**, rendant leur fonctionnement difficile à interpréter. Contrairement à l'IA symbolique, ces modèles n'intègrent pas de **structures logiques explicites** et ne possèdent pas de mécanismes de **raisonnement formel**, ce qui complique leur fiabilité et leur explicabilité.

Face à ces enjeux, le **Deep Synergy Learning (DSL)** propose une approche qui intègre les atouts des deux paradigmes. Plutôt que d'opposer **symbolisme** et **connexionnisme**, le **Synergistic Connection Network (SCN)** combine des **concepts symboliques** avec des **représentations neuronales** issues de l'apprentissage automatique. Cette hybridation permet de conserver la **plasticité adaptative** des modèles neuronaux tout en structurant mieux les connaissances acquises. L'**explicabilité** est renforcée par l'analyse des **liaisons synergiques** $\omega_{i,j}$, qui offrent une vision plus modulaire et interprétable de l'apprentissage.

Cette approche ouvre la voie à une **intelligence artificielle plus robuste**. Le réseau peut **évoluer continuellement, justifier ses choix et intégrer des connaissances structurées**, tout en restant capable de s'adapter aux nouvelles données et aux variations du contexte.

1.5.7.2. Principes d'une Approche Hybride Symbolique–Connexionniste

Dans la continuité des sections précédentes, et en particulier au regard des principes généraux du **Deep Synergy Learning (DSL)**, on constate que les approches **symboliques** et **sub-symboliques** apparaissent de plus en plus **complémentaires**. Les récentes avancées en **IA** mettent en évidence cette complémentarité. D'un côté, les approches sub-symboliques, comme les **réseaux neuronaux classiques**, le **DSL** et d'autres paradigmes similaires, offrent une **puissance d'apprentissage** notable pour traiter de grandes quantités de données et extraire des **caractéristiques complexes** dans des espaces de haute dimension. De l'autre, les méthodes symboliques (appuyées sur des **logiques**, des **règles** et des **ontologies**) assurent une **rigueur** et une **explicabilité** supérieures, notamment lorsqu'il convient de formuler des raisonnements structurés ou de décrire des **connaissances** de manière **interprétable**.

La question est donc de **concevoir** une architecture **hybride** qui mobilise simultanément la **puissance d'apprentissage** sub-symbolique et la **cohérence symbolique**. Dans le cadre d'un **Synergistic Connection Network (SCN)** tel que défini dans les sections précédentes, il est possible d'**introduire** des entités de différents types. Il est ainsi envisageable de faire cohabiter des **entités** $\mathcal{E}_{\text{symb}}$ représentant des **concepts logiques**, des **règles formelles**, ou des **faits** structurés, et des **entités** $\mathcal{E}_{\text{data}}$ se fondant sur des **vecteurs sub-symboliques** $\mathbf{x}_i \in \mathbb{R}^d$. On peut imaginer que la représentation symbolique contienne des énoncés tels que « Un véhicule se déplace sur roues » ou « Si la température excède 100°C, on suspecte une surchauffe », tandis que la partie sub-symbolique gère des **flux de données** (images, séries temporelles, signaux sensoriels) ou des **features** issues de réseaux de neurones.

Dans cette optique, on peut noter $\mathcal{E}_{\text{symb}}$ pour une **entité symbolique** et $\mathcal{E}_{\text{data}}$ pour une **entité sub-symbolique**. La **pondération** $\omega_{\text{symb},\text{data}}(t)$ s'adapte alors par la règle classique du **DSL** :

$$\omega_{\text{symb,data}}(t+1) = \omega_{\text{symb,data}}(t) + \eta [S(\mathcal{E}_{\text{symb}}, \mathcal{E}_{\text{data}}) - \tau \omega_{\text{symb,data}}(t)],$$

où la **synergie** $S(\mathcal{E}_{\text{symb}}, \mathcal{E}_{\text{data}})$ traduit à quel point la **règle** ou le **concept** véhiculé par $\mathcal{E}_{\text{symb}}$ s'applique utilement au **contenu** sub-symbolique porté par $\mathcal{E}_{\text{data}}$. La valeur de S peut tenir compte du **taux de succès** lors de la confrontation aux données, ou évaluer la **compatibilité** sémantique entre la règle symbolique et les caractéristiques identifiées dans le **flux sub-symbolique**.

On peut approfondir cette **coopération** à l'aide de **formules** décrivant la mise en correspondance entre un **concept** symbolique et des **features** extraites. Par exemple, on peut écrire :

$$S(\mathcal{E}_{\text{symb}}, \mathcal{E}_{\text{data}}) = \text{score} \left(\text{matching}(\text{rules}(\mathcal{E}_{\text{symb}}), \mathbf{x}_{\text{data}}) \right),$$

où $\text{rules}(\mathcal{E}_{\text{symb}})$ dénote l'ensemble des règles ou faits associés à la représentation symbolique, et \mathbf{x}_{data} la **représentation vectorielle** de l'entité sub-symbolique. La fonction **matching** évalue la cohérence entre la **logique symbolique** et les **observations** sub-symboliques, et **score** traduit le **gain** ou la **conformité** obtenu.

Dans cette approche **hybride**, il y a un **avantage** notable. Le réseau **auto-organisé** du DSL peut **identifier** quelles **règles** ou **concepts** sont les plus pertinents pour tel **type de données**, sans nécessiter une supervision massive. Les entités symboliques, en s'activant sur certaines conditions, peuvent **expliquer** plus aisément les **décisions** ou **groupements** effectués. La **transparence** en est renforcée. Si la **pondération** $\omega_{\text{symb,data}}$ augmente, cela signifie que la règle symbolique s'applique fréquemment et avec succès à la modalité de données concernée.

La **limite** principale repose sur la **cohérence** nécessaire entre la **logique** et les **données**. L'entité symbolique doit être suffisamment bien définie et adaptée à la réalité sub-symbolique ; en cas de décalage, la pondération associée risque de rester faible, rendant la règle peu intégrée. Sur le plan de l'implémentation, il faut également gérer la **complexité** liée à l'activation symbolique, potentiellement coûteuse s'il faut évaluer de nombreuses règles sur un flux volumineux de données.

Dans ce **paradigme**, la **puissance** du DSL est préservée. Le réseau se reconfigure en continu, ajustant les liens entre entités symboliques et sub-symboliques en fonction de la **synergie** détectée. En parallèle, les entités symboliques aident à **structurer** le raisonnement et à **rendre compte** de la logique sous-jacente, contribuant à l'**explicabilité** du système final. C'est cette double souplesse qui, dans la ligne de ce chapitre (section 1.5.7.2), semble prometteuse pour de nombreuses applications, allant de la **robotique cognitive** à l'**intégration** de vastes **bases de connaissances** et de **flux sensoriels**.

1.5.7.3. Mécanismes d'Auto-Organisation Intégrant la Logique

Dans la continuité de l'approche **hybride** symbolique–connexiste présentée en section 1.5.7.2, il est envisageable d'introduire au sein d'un **Synergistic Connection Network (SCN)** des **règles symboliques** interagissant avec des **entités** sub-symboliques. On peut par exemple considérer une **règle** R stipulant :

Si la caméra détecte une forme circulaire, alors suspecter un panneau de signalisation.

Parallèlement, on introduit une **entité sub-symbolique** \mathcal{E}_{cam} chargée d'analyser des données visuelles. Cette entité exploite typiquement un **modèle** neuronique local pour reconnaître ou segmenter des formes dans un **flux** d'images.

La **pondération** $\omega_{R,\text{cam}}$ reliant la règle R à l'entité \mathcal{E}_{cam} suit la dynamique du **DSL**. On peut ainsi l'écrire :

$$\omega_{R,\text{cam}}(t+1) = \omega_{R,\text{cam}}(t) + \eta [S(R, \mathcal{E}_{\text{cam}}) - \tau \omega_{R,\text{cam}}(t)],$$

où

$S(R, \mathcal{E}_{\text{cam}})$ mesure la **synergie** entre la **règle symbolique** et la **détection sub-symbolique** opérée par \mathcal{E}_{cam} . Concrètement, si la **forme circulaire** repérée par la caméra coïncide fréquemment avec la proposition « panneau de signalisation », la quantité $S(R, \mathcal{E}_{\text{cam}})$ prend une **valeur positive**. Le terme $\eta [S - \tau \omega]$ reste alors **positif**, ce qui **renforce** la liaison $\omega_{R,\text{cam}}$. Progressivement, on voit s'organiser un **cluster** local $\{R, \mathcal{E}_{\text{cam}}\}$, soulignant la coopération entre la **règle symbolique** et la **détection** d'images.

Lorsque survient une **deuxième** entité, $\mathcal{E}_{\text{limit}}$, qui incarne un **concept** tel que « panneau de limitation de vitesse », la **coopération** peut encore se prolonger. Si un **module** \mathcal{E}_{ocr} découvre régulièrement le texte “30 km/h” à l'intérieur des formes circulaires détectées, la synergie $S(\mathcal{E}_{\text{ocr}}, \mathcal{E}_{\text{limit}})$ peut s'avérer élevée. La **pondération** $\omega_{\text{ocr},\text{limit}}$ croît alors, impliquant l'émergence d'un **sous-réseau** plus vaste :

$$\{R, \mathcal{E}_{\text{cam}}, \mathcal{E}_{\text{ocr}}, \mathcal{E}_{\text{limit}}\}.$$

À l'intérieur de ce **cluster**, les entités **symboliques** (par exemple R et $\mathcal{E}_{\text{limit}}$) formulent des règles logiques ou des connaissances à caractère explicite. Les **entités** \mathcal{E}_{cam} et \mathcal{E}_{ocr} se chargent au contraire d'un **traitement** sub-symbolique (déttection de formes, reconnaissance du texte “30”). La **dynamique** des pondérations, selon la règle linéaire ou d'autres variantes (voir section 1.4.5), veille à **consolider** cet ensemble si les corrélations s'avèrent régulières et profitables.

Dans cette **topologie**, le **raisonnement** demeure essentiellement **distribué** et local. Plutôt qu'une structure hiérarchique fixe, l'**auto-organisation** en **micro-réseaux** permet aux règles et aux modules neuronaux de s'**associer** dynamiquement dès que leur **synergie** devient significative. Les **avantages** de cette intégration résident dans la capacité à **exploiter** la puissance de l'apprentissage sub-symbolique pour reconnaître des patterns visuels complexes, tout en s'appuyant sur la **précision** et la **transparence** qu'offrent les entités symboliques. De surcroît, on peut ainsi relier de multiples règles, chacune s'appliquant à des **entités** sub-symboliques différentes, ce qui permet d'atteindre une **modularité** plus élaborée à l'intérieur du **SCN**.

Les **limites** et **inconvénients** tiennent essentiellement au besoin de concevoir des règles symboliques suffisamment précises pour bien refléter la réalité observée par les flux sub-symboliques. S'il existe un décalage important ou un **bruit** excessif dans la détection, la synergie peut rester faible et empêcher la formation d'un cluster stable. Un autre **défi** concerne la **scalabilité**. Lorsqu'un nombre trop élevé de règles ou d'entités sub-symboliques coexistent, leur mise en correspondance devient coûteuse en calcul. Il est alors nécessaire d'adopter des stratégies de **sparsification** ou de **pruning** pour optimiser la dynamique du réseau, comme détaillé dans la section 1.4.5.

Dans l'ensemble, cette **intégration** de la **logique** dans le **DSL** illustre la possibilité de **fusionner** la robustesse neuronale ou statistique et la **structuration** symbolique. L'**auto-organisation** garantit que seules les entités **règle-module** qui partagent un gain effectif conservent un lien fort, aboutissant à une **cohérence** émergente entre les **concepts** logiques et les **vecteurs** de features dans le **Synergistic Connection Network**.

1.5.7.4. Avantages pour la Cognition et l'Explicabilité

Dans la continuité des sections précédentes, le **Deep Synergy Learning (DSL)** se présente comme un **pont** unifiant deux grandes approches du traitement de l'information. D'une part, la **dimension sub-symbolique** s'attache aux aspects de **perception** et de **features** extraits directement des données (images, sons, textes bruts), héritant de la puissance d'**apprentissage** de type connexionniste. D'autre part, la **dimension symbolique** structure des **concepts**, des **règles** et des **faits** logiques qui ne seraient pas nécessairement détectés par une simple analyse statistique. Par exemple, l'affirmation « un véhicule possède quatre roues, un moteur et est conçu pour se déplacer sur la route » représente une **connaissance explicite** que le réseau peut exploiter pour guider son apprentissage et renforcer certaines associations synergiques.

Lorsqu'un **cluster** (au sens de la section 1.4.3) associe une **règle** formalisée à des **entités perceptives** ancrées dans des vecteurs sub-symboliques, on obtient une **forme d'explication** plus directe de la décision. En effet, plutôt que de faire référence à un poids numérique situé dans une couche profonde du réseau, on peut pointer vers le **signal x** (issu de la représentation sub-symbolique) et la **règle R** (exprimée symboliquement) pour justifier la conclusion. Cette **transparence** s'apparente à une forme d'**explicabilité** (souvent recherchée en IA), où l'on peut réellement indiquer quelles **connaissances** et quelles **observations** ont motivé l'activation finale.

Pour formaliser mathématiquement cette idée, on peut écrire que la **pondération** $\omega_{R,\text{data}}(t)$ reliant la règle symbolique \mathcal{E}_R à l'entité sub-symbolique $\mathcal{E}_{\text{data}}$ évolue selon :

$$\omega_{R,\text{data}}(t+1) = \omega_{R,\text{data}}(t) + \eta [S(\mathcal{E}_R, \mathcal{E}_{\text{data}}) - \tau \omega_{R,\text{data}}(t)],$$

où $S(\mathcal{E}_R, \mathcal{E}_{\text{data}})$ mesure l'**adéquation** entre la règle symbolique et les **caractéristiques** perçues dans les données. Une fois cette liaison suffisamment renforcée, on peut interpréter la **décision** ou la **catégorie** apprise en mentionnant explicitement la règle R et le segment de données x qui la **valide**.

Je vais reformuler strictement ce que tu as donné, sans ajouter d'interprétation ou d'explication supplémentaire :

Ce mécanisme illustre la notion de **cognition synergique**. Les **entités symboliques codent** la **sémantique** et les **règles** explicites, tandis que les **entités** sub-symboliques **apprennent** à partir des données brutes, en extrayant des **patterns** ou des **features** utiles. Dans le **Synergistic Connection Network (SCN)**, la **dynamique** de mise à jour des **pondérations** relie directement ces deux pôles. On aboutit alors à une forme d'**intégration** qui se rapproche de la **cognition humaine**, où des **concepts** explicitement formulés cohabitent avec des **associations perceptives** acquises à travers l'expérience.

En termes d'**avantages**, on dispose d'une **robustesse** caractéristique des méthodes sub-symboliques, puisque le **réseau** peut s'adapter aux données, tout en bénéficiant de la **précision** et de la **lisibilité** qu'offre la **logique symbolique**. Cela se traduit par un **apprentissage** plus puissant combiné à une **explication** plus accessible. La conclusion ne dépend plus uniquement d'une somme opaque de poids, mais peut être reliée à un **ensemble** comprenant une **règle**, un **concept** ou un **fait**, ce qui en explicite la **signification**. Dans le même paragraphe, on peut souligner que la **cohabitation** de règles symboliques et de flux sub-symboliques exige une **cohérence** de haut niveau. Les règles doivent être appropriées pour décrire les phénomènes observés, et les entités sub-symboliques doivent parvenir à traduire efficacement le **signal** en indices qui **valident** ou **invalident** les **règles**.

Cette **fusion** entre le symbolique et le sub-symbolique au sein du **DSL** instaure un cadre proche de la **cognition synergique**. À l'image de l'esprit humain, le système combine des **concepts** explicites et des **représentations** perceptives implicites, puis les **coordonne** dynamiquement pour aboutir à une interprétation finale.

1.5.7.5. Défis Mathématiques et Implémentation

Dans la poursuite des principes décrits en sections 1.5.7.2 à 1.5.7.4, se pose la question de la **représentation** mathématique des **règles** symboliques à l'intérieur du **Deep Synergy Learning (DSL)** et, plus généralement, les difficultés techniques qu'implique l'introduction de composantes **logiques**. Un premier enjeu consiste à décider comment **encoder** une règle $\mathcal{E}_{\text{rule}}$. Certains travaux préconisent d'associer à chaque règle un **vecteur** ou un **arbre** logique compact, ce qui permet d'attribuer à $\mathcal{E}_{\text{rule}}$ un **nœud** du réseau muni de **paramètres** θ_{rule} . On peut alors définir la **synergie** $S(\mathcal{E}_{\text{rule}}, \mathcal{E}_{\text{feature}})$ de façon à intégrer à la fois la **structure symbolique** (par exemple, un parse-tree) et la **partie numérique** (par exemple, un embedding vectoriel ou un ensemble de features \mathbf{x}_{data}).

Il est possible d'écrire, de manière formelle, une fonction

$$S_{\text{mix}}(\mathcal{E}_{\text{rule}}, \mathcal{E}_{\text{data}}) = F(\text{logic}(\theta_{\text{rule}}), \text{embedding}(\mathbf{x}_{\text{data}})),$$

où **logic** exprime la partie symbolique (règles, connecteurs logiques, etc.), tandis que **embedding** encode la portion sub-symbolique. La fonction F peut évaluer une **co-information** ou un **score** d'adéquation entre la règle et les données. Cette démarche reprend les principes du **DSL** (sections 1.5.7.2 et 1.5.7.3) en donnant un **cadre** unifié pour la synergie entre une entité symbolique et une entité sub-symbolique.

Au fur et à mesure que le **nombre de règles** logiques augmente, on observe que la taille du **réseau** s'accroît, car chaque $\mathcal{E}_{\text{rule}}$ ou $\mathcal{E}_{\text{concept}}$ vient s'ajouter aux nœuds. Les **calculs** de synergie $S_{\text{mix}}(\mathcal{E}_{\text{rule}}, \mathcal{E}_{\text{data}})$ peuvent devenir coûteux si l'on doit évaluer régulièrement de nombreuses règles sur de vastes flux sub-symboliques. Il est alors crucial de recourir à des **techniques** de **parsimonie**, en particulier celles décrites en section 1.4.5, qui permettent de **supprimer** ou de **couper** certaines liaisons $\omega_{i,j}$ dès lors que leur pondération demeure faible. Cette régulation limite la prolifération de liens inutiles, maintenant la **complexité** à un niveau gérable.

Lorsque plusieurs **règles** se contredisent au sein du **SCN**, des **tensions** apparaissent. Il est possible que l'une des entités sub-symboliques (par exemple un flux de capteurs) présente une **synergie** positive avec une première règle, mais se heurte à une **incompatibilité** logique avec la seconde. Dans un tel cas, on peut introduire un **score de cohérence** ou une **pénalité** $P(\mathcal{E}_{\text{rule}_1}, \mathcal{E}_{\text{rule}_2})$, reflétant le fait que les règles rule_1 et rule_2 ne peuvent valablement être activées en même temps. On peut alors modifier l'évolution de chaque pondération afin de **réduire** la force de liens conduisant trop souvent à cette contradiction, ou introduire un **terme** correctif dans la mise à jour :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{mix}}(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)] - \alpha P(\mathcal{E}_i, \mathcal{E}_j),$$

où α contrôle l'intensité de la pénalité due à la contradiction.

D'un point de vue **implémentation**, les **limites** principales résident dans l'**explosion combinatoire** lorsque l'on souhaite intégrer non seulement des synergies binaires (section 1.4.4) mais également des synergies n-aires (section 1.4.7) et des **connaissances** logiques toujours plus nombreuses. De même, il peut être nécessaire de définir des **règles** plus modulaires ou paramétriques afin d'éviter la création d'une entité symbolique distincte pour chaque énoncé. Les entités pourraient alors coder des **schémas** logiques, associés à des **variables** substituables, pour diminuer le nombre total de noeuds.

Dans l'ensemble, ces **défis** mathématiques et de **mise en œuvre** ne remettent pas en cause le principe fondateur de l'approche hybride (sections 1.5.7.2 et 1.5.7.3), à savoir la possibilité de **fusionner** la rigueur symbolique avec la force d'apprentissage sub-symbolique. Ils incitent plutôt à développer des **algorithmes** de parsimonie adaptés, à concevoir des mécanismes de **cohérence** entre règles, et à prévoir des **heuristiques** ou **méthodes** de clustering pour gérer la croissance du réseau. Cette **fusion** entre le symbolique et le sub-symbolique au sein du **DSL** instaure un cadre proche de la **cognition synergique**. À l'image de l'esprit humain, le système combine des **concepts** explicites et des **représentations** perceptives implicites, puis les **coordonne** dynamiquement pour aboutir à une interprétation finale.

1.5.7.6. Conclusion

L'**incorporation** de composantes **symboliques** ou **cognitives** au sein du **Deep Synergy Learning (DSL)** ouvre un champ d'**IA hybride** dans lequel la **partie sub-symbolique** (traitement de données massives, bruitées et hétérogènes) est **complétée** par une **dimension symbolique** (règles, concepts, ontologies et raisonnement explicite). Cette combinaison se déploie naturellement dans le **Synergistic Connection Network (SCN)** qui, grâce à ses entités adaptatives et à ses liaisons synergiques évolutives, apparaît particulièrement adapté pour réunir et faire coévoluer ces deux approches.

On peut modéliser la **synergie** entre une **règle** R et un **module perceptif** $\mathcal{E}_{\text{data}}$ à l'aide d'une fonction

$$S(\mathcal{E}_{\text{rule}}, \mathcal{E}_{\text{data}}) = f(\theta_{\text{rule}}, \mathbf{x}_{\text{data}}),$$

où θ_{rule} encode l'information symbolique (logique, ontologie...) et x_{data} représente les **features** sub-symboliques. La pondération $\omega_{rule,data}(t)$ associée se met alors à jour de façon adaptative, selon

$$\omega_{rule,data}(t+1) = \omega_{rule,data}(t) + \eta [S(\mathcal{E}_{rule}, \mathcal{E}_{data}) - \tau \omega_{rule,data}(t)].$$

Cette dynamique assure que seules les **connexions** réellement bénéfiques (au sens de la synergie mesurée) se consolident, et permet à des **clusters** associant règles symboliques et entités sub-symboliques d'émerger spontanément au sein du réseau.

Ce mécanisme procure de **multiples avantages**. Il **préserve** la capacité d'**apprentissage** sub-symbolique, cruciale pour dompter des données riches et bruitées, tout en **introduisant** des notions symboliques aisément **explicables** (règles, concepts, axiomes). Cette alliance favorise une **explication** plus claire des décisions, dans la mesure où la conclusion d'un cluster peut être reliée à la **règle R** (logique) et au **signal x** (apprentissage neuronal). De plus, le DSL conserve ses forces intrinsèques, telles que la **multimodalité**, l'**adaptation** continue face aux variations de distribution et la **résilience** aux données partielles ou contradictoires.

Il faut toutefois noter qu'en **élargissant** la base de **règles** ou de **concepts**, le réseau peut s'agrandir et accroître la **complexité** de calcul, comme évoqué en section 1.5.7.5. Des mécanismes de **parsimonie** ou de **régulation** (limitation des connexions inutiles, seuils dynamiques sur les liaisons) demeurent essentiels pour éviter une explosion combinatoire. On peut également ajouter des **pénalités** ou un **score de cohérence** afin de gérer des contradictions logiques entre plusieurs règles.

En définitive, on obtient ainsi un **DSL** apte à **raisonner** en s'appuyant sur sa **base symbolique**, tout en **apprenant** de manière continue via sa **dimension sub-symbolique**. Cette architecture apparaît comme un **paradigme plus complet**, se rapprochant de la **cognition humaine**, laquelle jongle entre des **concepts** explicites et des **associations** perceptives implicites. Les sections et principes exposés dans ce chapitre (1.5) soulignent la **flexibilité** du DSL et son aptitude à jeter des **ponts** entre la **puissance** d'apprentissage neuronale et la **lisibilité** de la connaissance symbolique.

1.6. Applications Pressenties et Domaines Impactés

Le **Deep Synergy Learning (DSL)**, en tant que paradigme alliant **auto-organisation**, **synergies adaptatives** et **capacité d'évolution continue**, suscite un **intérêt grandissant** dans de nombreux champs de l'intelligence artificielle. Après avoir dressé, dans les sections précédentes, les fondements conceptuels et structurels du DSL (sections 1.1 à 1.5), il est temps de se projeter vers les **applications** et les **domaines** où cette approche pourrait exercer un **impact fort**.

L'architecture **distribuée** et **plastique** du DSL, la possibilité d'**auto-organiser** des entités multimodales ou symboliques, ainsi que sa **tolérance** aux données bruitées, confèrent au DSL un potentiel d'**adaptation** et de **complémentarité** qui dépasse souvent les limites des modèles traditionnels (réseaux neuronaux profonds strictement hiérarchiques ou méthodes symboliques pures). De la **vision artificielle** à la **robotique**, en passant par la **reconnaissance audio**, la **recommandation personnalisée**, la **surveillance** ou encore la **planification industrielle**, le DSL offre des réponses nouvelles, mieux adaptées à la complexité des flux de données et aux exigences d'évolution en temps réel.

Dans cette section (1.6), nous présentons un **éventail** des applications pressenties, non seulement comme une liste d'exemples, mais aussi comme une **cartographie** des domaines que le DSL peut transformer :

- **Vision Artificielle et Reconnaissance d'Objets Complexes** (1.6.1)
- **Analyse Audio et Traitement du Langage Naturel** (1.6.2)
- **Robotique et Systèmes Intelligents Évolutifs** (1.6.3)
- **Recommandation Personnalisée et Systèmes de Décision** (1.6.4)
- **Surveillance, Diagnostic Médical et Anomalies** (1.6.5)
- **Planification et Optimisation dans l'Industrie 4.0** (1.6.6)
- **Perspectives pour la Recherche Fondamentale en IA Forte** (1.6.7)

Pour chacun de ces thèmes, le **Deep Synergy Learning (DSL)** se révèle prometteur en raison de ses **mécanismes d'auto-organisation**, tels que les **clusters dynamiques**, les **pondérations adaptatives**, l'**intégration de multiples sources** (visuelles, auditives, textuelles, capteurs...) et l'**apprentissage continu** sans dépendance exclusive à la supervision. Nous verrons ainsi comment le DSL peut contribuer à relever certains **défis** actuels de l'IA, tels que la **plasticité**, la **résilience** aux données incertaines, ou encore la **capacité** à modéliser la **complexité** grandissante des problèmes.

1.6.1. Vision Artificielle et Reconnaissance d'Objets Complexes

Parmi les champs qui ont le plus bénéficié de l'essor des réseaux neuronaux profonds, on trouve la **vision artificielle**. Les **CNN** (Convolutional Neural Networks) ont acquis une renommée considérable pour la détection et la classification d'objets dans des images. Toutefois, les tâches de **reconnaissance** deviennent de plus en plus **subtiles**. Il ne s'agit plus seulement d'identifier un objet, comme un chat ou un chien, mais de **déetecter** et **comprendre** des **scènes complexes** dans des environnements changeants, avec des conditions lumineuses variables, des objets partiellement masqués ou des relations multiples entre eux. Le **Deep Synergy Learning** ouvre ici de nouvelles perspectives, dont nous détaillons ci-après les ressorts principaux.

1.6.1.1. Au-delà d'une Hiérarchie Rigide : Clusters d'Entités Visuelles

Dans un **CNN** classique, les couches de convolution et de pooling s'enchaînent selon un **pipeline** fixe. La première couche détecte des motifs élémentaires comme les bords et les contrastes. La suivante combine ces éléments pour former des structures plus complexes, et ainsi de suite, jusqu'à la classification finale. Bien que très performant, ce mécanisme demeure relativement **rigide**. Il impose un ordre précis selon lequel les "features" visuelles doivent être détectées, sans autoriser la création ou la suppression dynamique de filtres ni de liaisons entre diverses régions de l'image.

Le **Deep Synergy Learning (DSL)** envisage les **features** ou régions d'images comme autant d'**entités** \mathcal{E}_i , chacune portant un **descripteur** lié à un patch, un certain motif ou un vecteur plus abstrait (par exemple issu d'un autoencodeur local). Ces entités visuelles s'**auto-organisent** en **clusters** dès qu'elles identifient une **synergie** notable (proximité de descripteurs, co-occurrences de motifs, etc.). La **hiérarchie** (ou la "stratification") n'est plus imposée par des couches fixes, mais émerge au contraire de la **coopération** entre entités qui jugent profitable leur rapprochement. On peut par exemple voir se constituer un cluster dédié aux formes circulaires, un autre regroupant les zones de fortes lignes horizontales, et d'autres encore, voire des **macro-clusters** combinant plusieurs de ces groupes si leur fusion devient pertinente.

Cette **dynamique** apporte une plus grande **flexibilité** dans la reconnaissance d'objets complexes ou inattendus. Si un objet n'appartient pas aux catégories standards (par exemple, un objet partiellement masqué ou inconnu), les entités \mathcal{E}_k décrivant ses parties peuvent spontanément **coopérer** avec celles décrivant d'autres objets partiellement similaires. Il se crée alors un **sous-réseau** (cluster) susceptible de fusionner avec un regroupement plus vaste, agrandissant le champ d'interprétation.

Pour un modèle plus formel, on peut représenter chaque entité visuelle \mathcal{E}_k par un vecteur $\mathbf{x}_k \in \mathbb{R}^d$. La synergie, qu'elle soit définie via une **distance** (ex. inversée) ou une **co-information**, détermine la valeur $S(\mathcal{E}_k, \mathcal{E}_m)$. Les **pondérations** $\omega_{k,m}$ s'ajustent de façon adaptative au fil des données. Plusieurs entités corrélées finissent par **stabiliser** leurs liaisons internes, formant un **cluster** autour d'un même objet, d'un motif, ou d'une scène particulière. Cette organisation reste **évolutive**. Si le réseau identifie de nouvelles **entités** apportant une information pertinente, les **liaisons** se renforcent, modifiant la structure du **sous-graphe** et intégrant ces découvertes visuelles. La reconnaissance s'opère ainsi de façon plus **organique**, en contrastant avec un pipeline hiérarchique figé. Cette flexibilité permet l'émergence de **regroupements variés** et le **partage de sous-patrons** entre plusieurs objets lorsque cela améliore la synergie du réseau.

1.6.1.2. Multi-Échelle et Robustesse aux Déformations

Dans le traitement d'objets complexes, il est fréquent de rencontrer des **variations** d'échelle (petit/grand dans l'image), des **transformations** géométriques (rotation, vue partielle, ombre) ou une **composition** de sous-parties distinctes. Les réseaux de neurones de type **CNN** gèrent généralement ces phénomènes en pratiquant la **convolution multi-échelle** ou la **data augmentation**, où l'on applique diverses transformations (zoom, rotation...) lors de l'entraînement. Cette stratégie, efficace dans bien des cas, reste toutefois largement paramétrée à l'avance (couche de pooling fixes, patterns choisis pour l'augmentation).

Le **Deep Synergy Learning (DSL)** propose un mécanisme différent. Des **entités** \mathcal{E}_i peuvent décrire un même objet à des échelles ou des angles divers, et leurs **liaisons** $\omega_{i,j}$ se renforcent si la **synergie** (similarité, co-information) confirme qu'il s'agit bien de la **même structure** vue sous différentes perspectives. L'existence d'une entité dédiée à la "version miniaturisée" de l'objet et une autre à la "version grand format" aboutit à un **cluster** commun si elles trouvent un **gain** à leur rapprochement, sans exiger qu'on impose explicitement un niveau de pooling ou un tableau de transformations prédéterminé.

Un premier **exemple** (lié à l'invariance d'échelle) peut être formalisé en introduisant, pour un patch visuel \mathbf{x}_k , une entité $\mathcal{E}_{k,\alpha}$ décrivant ce patch à l'échelle α . Les pondérations $\omega_{(k,\alpha),(k,\beta)}$ s'élèvent si la ressemblance (ou co-information) entre la représentation à l'échelle α et à l'échelle β demeure élevée, ce qui crée un **cluster** multi-échelle au sein duquel ces entités se regroupent.

Un second **exemple** (visant les déformations) consiste à considérer $\mathcal{E}_{k,\theta}$ comme la **version** du patch \mathbf{x}_k après avoir subi une **rotation** θ . Deux angles θ_1 et θ_2 peuvent alors mener à des entités jugées "similaires", renforçant la liaison $\omega_{(k,\theta_1),(k,\theta_2)}$. On peut ainsi constituer un **macro-cluster** stable, reliant divers points de vue d'un même objet, garantissant une forme d'**invariance** à la rotation.

Cette approche confère une **robustesse** aux changements d'échelle et de pose. Le réseau fonctionne sans nécessiter de **couches de pooling** ni de **transformations explicites** ajoutées artificiellement. Le **DSL** permet aux **entités** de se **découvrir** et de s'associer spontanément dès qu'un **gain de synergie** est identifié, favorisant ainsi une organisation **adaptative** et **émergente**. L'ajustement se fait localement et **auto-organise** la reconnaissance, en évitant un pipeline rigide, pour composer des structures plus **adaptatives** et résilientes.

1.6.1.3. Interaction avec d'Autres Modalités (Vision + ...)

Dans un cadre multimodal (voir section 1.5.2), la **vision** artificielle s'avère fréquemment complémentée par un **flux audio** ou **textuel**. Le **Deep Synergy Learning (DSL)** n'établit pas de cloison pour la coopération entre l'entité visuelle \mathcal{E}_{vis} et l'entité auditive \mathcal{E}_{aud} . Le fait qu'elles appartiennent à des **modalités distinctes** n'empêche pas leur coopération. Si la **synergie** $S(\mathcal{E}_{\text{vis}}, \mathcal{E}_{\text{aud}})$ est positive, indiquant une **corrélation** ou une **co-information** significative, les **pondérations** $\omega_{\text{vis},\text{aud}}$ se renforcent progressivement, favorisant la formation d'un **cluster multimodal** adapté à la tâche.

Dans la **reconnaissance** d'objets complexes, un **événement** visuel (comme un objet qui tombe) peut se coupler à un **événement** sonore simultané (bruit d'impact). Si la représentation \mathbf{x}_{vis}

associée à la chute et la représentation \mathbf{x}_{aud} de l'onde sonore se rapprochent au sens de la **coopération**, un **sous-réseau** $\{\mathcal{E}_{\text{patch}}, \mathcal{E}_{\text{featureAudio}}\}$ se constitue. Il se spécialise dans la détection de la scène “objet tombant plus choc acoustique”, ce qui accroît les facultés de **classification** ou de compréhension de la situation (“Cassure”, “Chute de vaisselle”, etc.).

Cette **flexibilité** s'étend à d'autres flux, notamment l'**analyse textuelle**, où des **légendes associées** ou des **mots-clés** issus d'un utilisateur peuvent être pris en compte. Il en résulte la formation de **micro-clusters multi-entités**, où la vision et l'audio coopèrent tout en interagissant avec des entités textuelles. La dynamique d'**auto-organisation** du **DSL** détermine les **groupements** les plus profitables, sans exiger qu'un concepteur définisse un module de fusion dédié pour le couple “vision + audio”. Cela favorise une **intégration** sans couture des diverses modalités, conduisant à une représentation plus riche et plus **résiliente** aux variations d'une modalité prise isolément.

1.6.1.4. Impact sur les Problèmes de Vision Avancés

Dans le cadre de la **vision par ordinateur**, un **Deep Synergy Learning (DSL)** structuré en entités visuelles présente un atout certain pour traiter des scénarios avancés qui outrepasse la simple classification d'images. L'**auto-organisation** autorise l'émergence de **clusters** plus flexibles, allant au-delà du pipeline statique imposé par les méthodes classiques.

Dans la **détection d'objets multiples et partiellement superposés**, par exemple, un réseau traditionnel prévoit souvent un ensemble rigide d'ancrages (bounding boxes) ou un module de segmentation spécifique. Au contraire, le **DSL** peut laisser chaque patch ou région s'**affilier** à des **entités** “chien” et “chaise” même si ces objets se chevauchent dans l'image. Les pondérations $\omega_{i,j}$ augmentent dès que la synergie entre patchs associant l'idée de “chien” et l'idée de “chaise” se justifie. Le réseau n'a pas besoin d'un pipeline imposant a priori la détection séparée de chaque catégorie ; la constitution du **cluster** facilite l'identification conjointe de plusieurs objets.

Dans la **segmentation contextuelle**, on peut envisager que certaines entités segmentent l'image en superpixels et d'autres identifient des **contours**. Si leur **synergie** révèle une cohérence — par exemple, les bordures détectées coïncident avec les frontières d'un superpixel —, la liaison ω se renforce. On forme alors un **cluster** correspondant à une “zone cohérente” dans l'image, laquelle pourrait se rattacher à un objet complet ou à une portion d'arrière-plan homogène. Cette approche s'écarte de la segmentation par apprentissage end-to-end, en permettant une coopération plus libre entre entités spécialisées (détection de bordures, regroupement de pixels...) selon la présence d'un **gain**.

Dans un contexte de **vision évolutive en robotique**, un agent dans un environnement changeant (luminosité, décor, nouveaux éléments) doit s'ajuster en continu. Un réseau de neurones traditionnel exigerait soit un ré entraînement, soit une réinitialisation partielle. Le **DSL** réévalue localement la **synergie** entre entités visuelles. Les connexions correspondant à d'anciennes conditions, comme une forte luminosité, **diminuent** si elles deviennent obsolètes. En parallèle, de nouvelles **entités** ou **liaisons** émergent pour s'adapter à des variations d'éclairage, des changements d'angles de vue ou l'apparition d'objets inédits. Cette **adaptation** locale maintient la **structure** globale du réseau, sans nécessiter un apprentissage complet de bout en bout, et confère une plasticité plus grande à la **vision** robotique, tout en conservant la possibilité d'exploiter ses acquis antérieurs.

1.6.1.5. Conclusion Partielle : Une Alternative Complémentaire au CNN

Les **CNN** constituent toujours un **vecteur** extrêmement puissant pour la **vision** artificielle, en particulier lorsqu'un large corpus d'exemples annotés est disponible et que l'on vise des tâches de classification ou de détection sur des données stables. Toutefois, la structure d'un **CNN** reste **déterministe**. Chaque couche est **fixe**, et l'apprentissage repose principalement sur une **reconnaissance supervisée**, sans possibilité d'adaptation structurelle dynamique. Le **Deep Synergy Learning (DSL)** apporte une approche plus **organique** et **dynamique**, où les entités associées à des *patchs* ou des *features* visuelles peuvent librement se **synergiser**, se **séparer**, **s'associer** avec d'autres modalités ou s'adapter aux **conditions** changeantes (nouvelles échelles, bruit, objets inédits).

La structure du **Synergistic Connection Network** autorise ainsi une **auto-organisation** continue, évitant la rigidité d'un pipeline imposé pour la vision. On peut introduire de nouvelles entités $\mathcal{E}_{\text{nouvelle}}$ si un objet inconnu apparaît ou si l'on veut traiter un autre type de patch visuel, ce qui met à jour localement les connexions ω . Le **DSL** peut s'**hybrider** avec un **CNN** en exploitant les **descripteurs** extraits d'une couche intermédiaire comme entités \mathcal{E}_{CNN} . La **synergie** entre ces entités reflète alors leur **complémentarité**, permettant une structuration plus dynamique et adaptative des informations issues du réseau convolutionnel.

Pour la **reconnaissance d'objets complexes** dans des scènes variées (éclairage fluctuant, angles de vue inédits, transformations géométriques), le **DSL** fournit un **cadre** mathématique structuré, avec des **équations** décrivant l'évolution des liaisons $\omega_{i,j}$ et des mesures de synergie multi-échelle (ou non linéaire). Il enrichit la **vision** d'une dose de **plasticité** et de **robustesse**, sans empêcher l'usage de **CNN** comme base de features. Les **clusters** qui émergent dans ce réseau renvoient à des regroupements de patches ou de motifs liés par leur **gain mutuel**, offrant un point de vue plus distribué et collaboratif qu'un pipeline hiérarchique unique. Au final, le **DSL** s'érige en complément précieux aux **CNN** pour atteindre une **vision** flexible, résiliente et progressivement évolutive.

1.6.2. Analyse Audio et Traitement du Langage Naturel

Après avoir vu comment le **Deep Synergy Learning (DSL)** peut apporter une flexibilité et une auto-organisation en **vision artificielle** (1.6.1), il est naturel de s'intéresser aux **données audio** et au **langage naturel**, deux domaines où la richesse des signaux et la variabilité contextuelle sont particulièrement élevées. Dans un réseau neuronal profond classique, les approches d'**analyse audio** (speech recognition, classification de sons, détection d'événements acoustiques) et de **traitement du langage naturel** (NLP) (analyse sémantique, traduction, question-réponse) demeurent puissantes, mais reposent souvent sur des modèles séparés (RNN, LSTM, Transformers) et des schémas d'apprentissage supervisé imposant de gros volumes de données annotées.

Le **DSL**, en revanche, repose sur une **auto-organisation** où chaque entité \mathcal{E}_i (qu'elle soit issue de données audio ou textuelles) s'insère dans un **réseau synergique** plus large. Cette approche permet une **fusion plus fluide** des informations acoustiques et linguistiques, facilitant une **adaptation**

continue aux variations telles que le bruit, les accents ou les registres de langage. De plus, elle favorise une **cohabitation** entre les représentations symboliques et sub-symboliques, rendant possible l'intégration de règles logiques dans des tâches linguistiques plus complexes.

Cette sous-section (1.6.2) détaille en quoi le DSL modifie l'approche habituelle de l'analyse audio et du traitement du langage, et en quoi il répond à des défis d'**hétérogénéité**, de **bruit**, de **variabilité linguistique** et de **scénarios évolutifs**.

1.6.2.1. Analyse Audio : Entités Sonores et Synergie Adaptative

Dans un **réseau** neuronal **classique** pour l'audio (CNN ou RNN appliqués aux spectrogrammes), on spécifie dès le départ un schéma d'extraction de *features* (MFCC, log-mel, etc.), puis l'on optimise un classifieur (par exemple, pour la reconnaissance de phonèmes ou la classification de sons). Le **Deep Synergy Learning (DSL)** introduit une logique plus **organique** en laissant des **entités** $\mathcal{E}_{aud,k}$ se constituer pour chaque **fenêtre** ou **segment** temporel (frames audio), pour un **spectre** fréquentiel spécifique, ou pour un **embedding** plus abstrait (mots, bruits, etc.). Ces entités se regroupent en **clusters** dès lors qu'elles observent un **gain** à coopérer, qu'il s'agisse de la ressemblance spectrale, de la proximité temporelle ou de la récurrence d'un même motif acoustique.

Si on modélise deux segments audios $\mathbf{a}_i, \mathbf{a}_j \in \mathbb{R}^d$, leur synergie peut être définie par une **distance inversée** ou une **similitude**, telle que

$$S(\mathcal{E}_i, \mathcal{E}_j) = \exp(-\gamma \|\mathbf{a}_i - \mathbf{a}_j\|^2),$$

pour un certain $\gamma > 0$. La **pondération** $\omega_{i,j}(t)$ qui relie les entités \mathcal{E}_i et \mathcal{E}_j se met alors à jour selon la règle adaptative (voir section 1.4.5). Lorsque deux **fenêtres** ou segments audio affichent une forte similarité (timbre commun, bruit de fond identique, même locuteur), leurs liaisons $\omega_{i,j}$ se **renforcent**, initiant ainsi la création d'un **cluster** rassemblant ces entités, et permettant de **repérer** un même motif acoustique. Cette **auto-organisation** se substitue aux approches classiques qui segmentent l'audio selon des règles prédéfinies. Plutôt que d'imposer une structure rigide, le réseau ajuste dynamiquement ses **liaisons** en fonction de la synergie perçue, permettant une **segmentation** ou une **agrégation** adaptative des informations.

De plus, si un **environnement** se modifie (bruit important, micro défaillant, accent inhabituel), des entités précédemment utiles peuvent perdre leur **synergie** avec les autres composantes, faisant décroître $\omega_{i,j}$. L'entité se retrouve alors **isolée** et n'influence plus le cœur du réseau. Inversement, lorsqu'une nouvelle configuration sonore émerge (baisse du bruit, nouveaux sons caractéristiques), d'autres **connexions** ω apparaissent ou se renforcent, configurant de nouveaux **clusters** sonores. Ainsi, la **reconnaissance** de motifs acoustiques (sirènes, voix enfantines, grondements) se met en place de manière **continue**, sans qu'on doive réentraîner de bout en bout un réseau rigide. Le **DSL** s'adapte localement, confortant l'idée d'un système plus **résilient** et **évolutif** dans le traitement de l'audio.

1.6.2.2. Traitement du Langage Naturel : Entités Lexicales et Sémantiques

Dans les méthodes **classiques de traitement du langage naturel** (NLP), des modèles (RNN, LSTM, Transformers) s'appuient sur des **embeddings** (word2vec, GloVe, BERT, etc.) et des mécanismes d'attention ou de convolution. Bien qu'ils aient atteint d'excellents résultats (traduction, question-réponse, résumé, etc.), ils s'avèrent fortement dépendants d'un **entraînement supervisé** ou auto-supervisé de grande ampleur, et peinent à **intégrer** de la **logique** ou du **symbolique** sans recourir à des mécanismes additionnels. De plus, la **structure** de ces modèles demeure souvent linéaire (ou en arbre pour l'analyse syntaxique), imposant le traitement séquentiel des tokens.

Le **Deep Synergy Learning (DSL)** propose une alternative plus **flexible**. Au lieu d'imposer une séquence rigide de tokens à travers un pipeline (embedding + attention + couches finales), on peut **déployer** un **réseau** d'entités couvrant les différentes dimensions du texte. Chaque entité \mathcal{E}_i peut représenter un **mot** (ou token) muni d'un embedding \mathbf{w}_i , un **nœud syntaxique** extrait d'un parse tree (rôle de sujet, verbe, complément), ou un **concept sémantique** (topic, entité nommée, etc.). Ces entités s'**auto-organisent** si leur synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ s'avère suffisamment élevée pour justifier un **regroupement**.

On peut alors voir naître des **clusters** lexical-sémantique regroupant des mots liés à un même champ lexical, un **verbe** et ses **compléments**, ou plusieurs **concepts** partagés par diverses phrases. Plutôt qu'une **arborescence figée**, comme dans une analyse syntaxique classique, le **SCN** adopte une **structure fluide**. Lorsqu'un mot présente plusieurs sens possibles, la **synergie** avec d'autres entités textuelles ou contextuelles oriente naturellement son interprétation. Ce mécanisme permet une **désambiguïsation dynamique**, où le sens retenu dépend des relations de synergie établies en temps réel au sein du réseau. De même, des liaisons $\omega_{i,j}$ peuvent se **rompre** si un ensemble de mots cesse d'être pertinent dans un nouveau contexte textuel. L'**auto-organisation** rend donc la structure du **réseau** ajustable, permettant d'incorporer ou de dissoudre des sous-groupes de tokens ou de concepts selon qu'un **gain** mutuel se manifeste, ouvrant ainsi la voie à une **compréhension** plus modulable du langage.

1.6.2.3. Couplage Audio–Texte et Auto-Fusion Synergiques

De nombreux systèmes doivent traiter simultanément un **flux audio** (voix, intonations, bruits) et un **texte** associé (transcriptions, sous-titres). Le **Deep Synergy Learning (DSL)** s'avère particulièrement bien adapté à ce scénario, car il ne fixe pas a priori la manière de "joindre" l'audio et le texte. Au lieu d'un alignement imposé par un algorithme dédié, on laisse le **Synergistic Connection Network (SCN)** réguler les liaisons ω entre des **entités** \mathcal{E}_{aud} et des **entités** \mathcal{E}_{txt} . Si la **co-information** ou une autre mesure de **similarité** suggère qu'un certain segment sonore correspond à un certain segment textuel, la pondération $\omega_{\text{aud}, \text{txt}}$ se renforce, conduisant à la création d'un **cluster** multimodal.

Parmi les entités audios, on peut distinguer des segments temporels ou des frames associées à une **fenêtre** spectrale spécifique. Du côté textuel, on dispose d'entités représentant des **mot**s, des **tokens**, voire des syntagmes plus longs. La mise à jour des liens $\omega_{\text{aud}, \text{txt}}$ procède alors par la règle adaptative (section 1.4.5), détectant les **paires** qui coïncident fréquemment. Ainsi, lorsqu'un

segment audio s’aligne régulièrement sur une suite de mots, on observe un gain synergique. On forme donc un **cluster** englobant les parties orales et leurs correspondances textuelles, sans qu’un schéma d’alignement fixe soit nécessaire.

Un tel mécanisme confère une **robustesse** notable. Si des segments audios se montrent inintelligibles (forte perturbation, accent trop marqué), la **synergie** s’affaiblit, et les liaisons avec les entités textuelles cessent de croître, isolant ces segments pour ne pas polluer l’ensemble. Le réseau **accepte** ainsi l’hétérogénéité de la qualité sonore et s’auto-adapte. En outre, cette logique **permet de découvrir** des co-occurrences ou des associations inhabituelles, par exemple un mot (ou un motif lexical) qui se répète chaque fois qu’un type particulier de sonorité apparaît. Le **DSL** parvient donc à fusionner audio et texte de manière plus **flexible** que ne le ferait un pipeline d’alignement rigide, et il favorise l’intégration de nouvelles données ou la détection de nouveaux mots/sons sans devoir procéder à un réentraînement ou une refonte exhaustive du modèle.

1.6.2.4. Évolutions Possibles et Défis Mathématiques

Dans l’application du **Deep Synergy Learning (DSL)** à l’audio, au texte ou à des mélanges de modalités, plusieurs difficultés et pistes d’extension apparaissent.

Un premier enjeu concerne la **complexité**. Comme en vision (section 1.6.1.4), on peut générer un grand nombre d’entités audio (fenêtres temporelles, frames) ou textuelles (tokens, syntagmes). Le **graph** qui en résulte peut ainsi atteindre une taille considérable, rendant la mise à jour des pondérations $\omega_{i,j}$ et l’estimation des synergies coûteuses. Il est donc souvent nécessaire de recourir à des **stratégies** parcimonieuses pour limiter la prolifération de liens, comme échantillonner périodiquement les entités, fusionner les entités jugées redondantes ou fixer un seuil ω_{\min} qui supprime les connexions trop faibles (section 1.4.5).

En second lieu, on notera que la **modélisation n-aire** peut se révéler essentielle. Les **synergies** binaires (audio–audio, texte–texte ou audio–texte) ne suffisent pas toujours à capturer des phénomènes impliquant trois (voire davantage) entités. Il se peut, par exemple, que l’union de deux fragments audio et d’un groupe de mots textuels produise une information inatteignable avec des paires isolées. La **synergie n-aire** (section 1.4.7) permet de refléter ces complémentarités plus complexes, au prix d’un accroissement de la difficulté mathématique, car il faut évaluer l’information ou la similarité simultanément sur plusieurs variables.

Enfin, la possibilité d’**hybridation symbolique** apparaît particulièrement intéressante dans le domaine du **NLP** avancé (raisonnement, question-réponse complexes). Comme évoqué en section 1.5.7, on peut introduire des **règles logiques** ou des **entités symboliques**, lesquelles interagissent avec les **entités textuelles**. La synergie se définit alors par une fonction reliant le sens des mots à des conditions logiques ou des assertions formelles. Si cette intégration accroît encore la complexité, elle apporte un surcroît de **capacité cognitive**, autorisant un raisonnement plus poussé et assurant une **explicabilité** renforcée, dans l’esprit d’une IA **neuro-symbolique** où la dynamique des pondérations répond à la fois aux critères sub-symboliques (similarité, co-information) et aux axiomes symboliques.

1.6.2.5. Conclusion Partielle : un Cadre Évolutif pour l'Audio et le Langage

Le **Deep Synergy Learning (DSL)** s'affirme comme une **approche** ou un **complément** aux architectures traditionnelles (CNN, RNN, Transformers) dans l'analyse de **flux audio** et le **traitement du langage naturel**. Contrairement aux pipelines classiques, le **DSL** ne se fonde pas sur un découpage préalable ou un alignement imposé, mais laisse les entités acoustiques ou lexicales **s'auto-organiser** dès que leur synergie l'indique. Cette approche apporte plusieurs **bénéfices** en matière de **segmentation adaptative** et de **fusion multimodale**.

L'**auto-organisation** ajuste dynamiquement la segmentation et l'alignement, consolidant les pondérations entre segments audio et unités textuelles dès qu'une corrélation ou une complémentarité apparaît. Contrairement aux méthodes traditionnelles qui imposent un découpage temporel rigide, le **réseau** ajuste ses connexions de façon fluide, ce qui le rend **robuste** face aux variations acoustiques (bruit, accents, altérations de source) et linguistiques (polysémie, registres de langage). Lorsqu'une information devient incertaine ou bruitée, les liaisons s'atténuent d'elles-mêmes, évitant un réapprentissage global.

La **fusion** entre l'audio et le texte se fait de manière **naturelle**, sans exiger un alignement strict entre un segment oral et sa transcription. Le **DSL** découvre spontanément les correspondances pertinentes en évaluant la **synergie** entre unités sonores et textuelles, indépendamment d'un cadre prédéfini. Cette flexibilité permet d'intégrer **de nouveaux sons, du vocabulaire inédit ou des règles symboliques** sans nécessiter une reconstruction complète du système. Il suffit d'introduire de nouvelles **entités** dans le graphe, et la mise à jour des pondérations s'effectue automatiquement.

Ces caractéristiques favorisent plusieurs **applications évolutives** en traitement du langage et de l'audio. La **reconnaissance vocale** en milieu bruité devient plus robuste grâce à l'**auto-organisation**, qui ajuste les pondérations en fonction de la qualité du signal. La **compréhension** des conversations multimodales, intégrant audio et vidéo, se fait sans imposer un **pipeline** d'alignement rigide. La **traduction** et la **détection** d'anomalies audio-linguistiques bénéficient également de cette structure **distribuée et dynamique**, qui ajuste les liens entre entités selon la **synergie** mesurée.

Ce cadre repose sur des **règles d'auto-organisation** qui adaptent les pondérations et façonnent des **clusters**, tout en s'appuyant sur des critères de **synergie** tels que la distance et la co-information. L'ensemble favorise un fonctionnement plus **flexible**, moins dépendant d'un entraînement supervisé massif, et davantage axé sur une **évolution continue** où les entités coopèrent localement pour optimiser l'interprétation des données.

1.6.3. Robotique et Systèmes Intelligents Évolutifs

Les concepts d'**auto-organisation**, de **synergie** et de **coopération** entre entités, propres au **Deep Synergy Learning (DSL)**, prennent tout leur sens dans le domaine de la **robotique** et des **systèmes intelligents**. En effet, la robotique moderne repose sur l'**intégration** de capteurs variés tels que la vision, l'audio, la pression ou la position, chacun fournissant des données essentielles. Les **actions** doivent être coordonnées avec précision, qu'il s'agisse de mouvements, de manipulations ou d'interactions, tout en tenant compte des changements de l'environnement. L'**adaptation continue** est primordiale pour ajuster la dynamique du robot face aux imprévus, aux variations de l'espace

ou aux perturbations extérieures. Enfin, la **planification** et la **prise de décision en temps réel** exigent une gestion efficace de l'**incertitude** et du **bruit**, rendant indispensable une approche flexible et évolutive.

Cette sous-section (1.6.3) montre comment le **DSL**, grâce à ses mécanismes de **pondérations adaptatives** et à ses **clusters auto-organisés**, peut être particulièrement utile à la **robotique** et aux **systèmes intelligents**. Elle décrit :

- L'**approche distribuée** que le DSL propose pour les robots multi-capteurs,
- La **coévolution** des entités sensorielles et motrices,
- La capacité à **s'auto-réorganiser** lorsqu'un robot ou un système intelligent découvre de nouveaux modules (nouveaux effecteurs, nouveaux contextes, etc.),
- Des exemples d'applications, depuis les **robots collaborateurs** jusqu'aux **systèmes autonomes** en changement permanent.

1.6.3.1. Robotique Multi-Capteurs : une Structure Synergique

Dans un contexte robotique, un système typique réunit plusieurs **capteurs** disposés sur la plateforme, pouvant inclure une **caméra** (analyse visuelle), un **LIDAR** ou un **radar** (cartographie de l'environnement), des **capteurs de pression** ou de **toucher**, des **capteurs inertIELS** (IMU), un **microphone** pour la partie audio, etc. Les approches traditionnelles (notamment basées sur ROS et des architectures préconçues) tendent à définir un **pipeline** de fusion, auquel chaque capteur envoie ses données, puis une **couche** de décision. Cette rigidité suppose un schéma de liaison explicite entre capteurs et modules supérieurs, imposant par exemple un module de traitement conjoint image–LIDAR si l'on veut superposer un nuage de points laser et un flux de caméra.

Le **Deep Synergy Learning (DSL)**, quant à lui, permet d'instituer un **ensemble** d'entités $\{\mathcal{E}_{\text{cam}}, \mathcal{E}_{\text{lidar}}, \dots\}$ correspondant chacune à une source sensorielle ou à un bloc fonctionnel plus abstrait. Chaque **entité** du système évalue la **synergie** qu'elle entretient avec les autres. Par exemple, si la caméra \mathcal{E}_{cam} et le LIDAR $\mathcal{E}_{\text{lidar}}$ détectent fréquemment le même obstacle, leur liaison se **renforce** progressivement, ce qui entraîne une consolidation de la pondération $\omega_{\text{cam},\text{lidar}}$. Cette dynamique adaptative permet aux capteurs de s'**ajuster** en fonction de leur cohérence mutuelle, optimisant ainsi la perception et la prise de décision du système. Inversement, lorsqu'un capteur inertiel s'avère temporairement *bruité* (information incohérente par rapport aux autres lectures), ses connexions décroissent naturellement jusqu'à s'affaiblir et, le cas échéant, s'isoler.

Grâce à ce principe d'**auto-organisation**, il devient possible qu'un **cluster** multimodal émerge pour caractériser une situation telle qu'un “terrain accidenté” perçu simultanément par la caméra (textures complexes), le LIDAR (irrégularités de distance) et peut-être un micro détectant des bruits anormaux. Ce cluster se **dissout** ensuite lorsque le robot quitte la zone problématique ou que la configuration change. Le **réseau** d'entités agit donc de manière plus **dynamique** et **distribuée**, sans imposer un module de fusion central dédié à chaque couple de capteurs. Les liaisons $\omega_{i,j}$ se forment ou se dissolvent localement, en fonction de la **valeur ajoutée** mesurée à travers la synergie

entre entités, conférant au système une **résilience** plus élevée, qu'il s'agisse de gérer un capteur défaillant, l'arrivée d'un nouveau module, ou un changement inattendu dans l'environnement.

1.6.3.2. Action, Effecteurs et Synergie Motrice

En robotique ou dans tout système intelligent disposant de **modules d'action** (moteurs, articulations, roues, pinces), on adopte généralement un paradigme où un **contrôleur** (PID, MDP, RL, etc.) reçoit un **état** en provenance des capteurs et génère des **commandes**. Le **Deep Synergy Learning (DSL)** aborde la question autrement, en considérant les effecteurs comme des **entités** $\mathcal{E}_{motrice}$ au même titre que les capteurs. Par exemple, chacune des articulations d'un bras, ou la rotation d'une roue, ou la position d'une pince peut être dépeinte par un vecteur $\mathbf{x}_{motrice}$. La **coopération** avec les entités sensorielles s'établit dès lors que la **synergie** $S(\mathcal{E}_{capteur}, \mathcal{E}_{motrice})$ s'avère positive, incitant à **renforcer** la pondération $\omega_{capteur, motrice}$.

Un bras articulé qui coordonne son mouvement avec un flux visuel voit, par exemple, la pondération $\omega_{cam, bras}$ croître si la caméra confirme que la pince se trouve dans la bonne position. À l'inverse, un joint dysfonctionnel ou mal calibré cesse d'apporter un réel gain, ce qui fait chuter la **synergie**, rendant son influence moindre dans la stratégie globale. Le **SCN** assure ainsi une forme d'**adaptation** où les effecteurs pertinents coopèrent étroitement avec les capteurs produisant des informations utiles, sans qu'il soit nécessaire de configurer explicitement un contrôleur central devant orchestrer toutes les modalités.

On peut également insérer un **signal** de récompense ou un indicateur de performance R qui module la synergie. Si la coopération entre un capteur $\mathcal{E}_{capteurA}$ et un effecteur $\mathcal{E}_{motriceB}$ accroît ce score (par exemple, la manipulation d'un objet est mieux réalisée, ou la trajectoire est plus stable), la **pondération** $\omega_{capteurA, motriceB}$ s'en trouve augmentée. Cette logique correspond à un **processus** de “learning by synergy”, sans avoir à recourir à une formalisation en MDP (Markov Decision Process) rigide. Les entités effectrices et sensorielles **s'auto-organisent** localement via la mise à jour des connexions ω , découvrant progressivement quelles combinaisons capteur-action se révèlent fructueuses pour atteindre l'objectif.

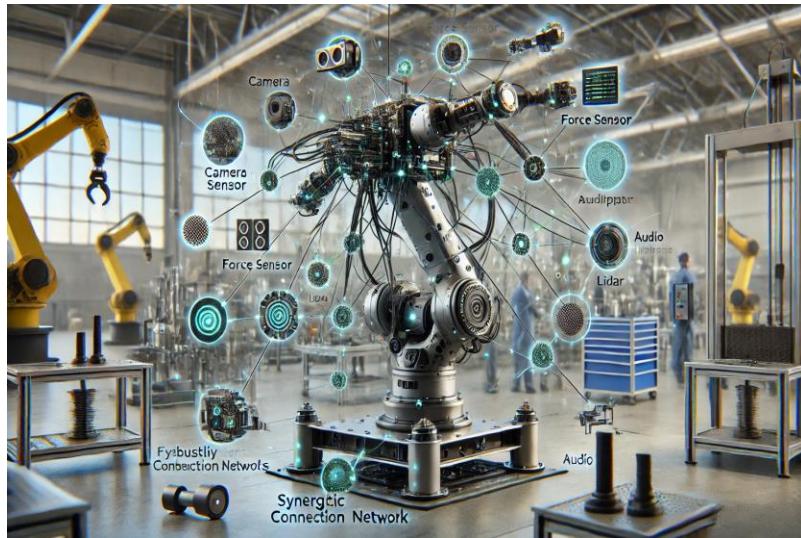
1.6.3.3. Adaptation Continue et Évolution Modulaire

Cette **flexibilité** structurelle constitue un atout majeur du **Deep Synergy Learning (DSL)**. Lorsqu'un robot reçoit un **nouveau capteur** ou un **nouvel effecteur**, ou qu'un module obsolète est retiré, il suffit d'**ajouter** ou de **supprimer** l'entité correspondante $\mathcal{E}_{nouveau}$ au sein du **Synergistic Connection Network (SCN)**. La **dynamique adaptative du réseau ajuste automatiquement** les liaisons synergiques. **Les nouvelles connexions se renforcent si elles apportent un gain, tandis que les anciennes, devenues inutiles, s'atténuent sans nécessiter de réapprentissage exhaustif.** Les pondérations $\omega_{nouveau}$ suivent la dynamique habituelle (voir section 1.4.5) et se renforcent ou se **dissolvent** en fonction de la **synergie** détectée. Il n'est donc pas requis de recompiler l'ensemble du pipeline ou de procéder à un **réapprentissage** complet d'un grand réseau neuronal. La **logique** adaptative se borne à évaluer localement la valeur ajoutée de la nouvelle entité, puis à l'intégrer dans un **cluster** pertinent si besoin.

Cette **propriété** ne se limite pas à la **robotique physique**. Dans des **systèmes logiciels** complexes, par exemple dans l'administration de centres de données ou d'architectures distribuées, le principe reste valable. Chaque microservice, base de données, module de monitoring, etc., peut être considéré comme une **entité** $\mathcal{E}_{\text{service}}$. L'arrivée d'un nouveau service, la suppression (ou la panne) d'un service existant, se modélise par l'**ajout** ou le **retrait** d'entités correspondantes, et les **liaisons** ω se régulent selon la **coopération** réelle (échange d'informations, dépendances). Les **clusters** qui en résultent rassemblent les **services** ayant des affinités ou co-occurrences fortes, autorisant une **répartition** et une **coordination** modulaires.

Ce principe d'**auto-organisation** évite la centralisation rigide d'un pipeline complet ou la révision constante d'un réseau monolithique. Le **SCN** s'ajuste **spontanément**. Les entités fiables sont **préservées**, tandis que celles devenues obsolètes voient leurs liaisons $\omega_{i,j}$ s'atténuer. Lorsqu'une **nouvelle entité** apparaît, le réseau **restructure** ses connexions pour maximiser la **synergie**, sans nécessiter un **réapprentissage complet**. Cette démarche s'inscrit dans l'esprit du **lifelong learning**, où le **réseau** lui-même façonne en continu sa **topologie** et maintient un équilibre entre la conservation de connaissances antérieures et l'ajout de briques inédites.

1.6.3.4. Illustration : Robot Collaboratif Multicapteurs



Un **robot collaboratif** (cobot) dans un environnement industriel peut disposer de multiples capteurs et actionneurs. Par exemple, on considère une **caméra** pour localiser la pièce à assembler, un **capteur de force** dans la pince pour jauger la préhension, un **LIDAR** pour détecter la distance aux opérateurs humains et éviter les collisions, un **module audio** pour recevoir des instructions vocales, ainsi que les actionneurs comprenant la **pince**, le **bras articulé** et un **moteur** de déplacement sur plateforme.

Dans un **Synergistic Connection Network (SCN)** selon le **Deep Synergy Learning (DSL)**, chaque composant est représenté comme une **entité** : $\mathcal{E}_{\text{cam}}, \mathcal{E}_{\text{force}}, \mathcal{E}_{\text{lidar}}, \mathcal{E}_{\text{audio}}$ pour les capteurs, et $\mathcal{E}_{\text{pince}}, \mathcal{E}_{\text{bras}}, \mathcal{E}_{\text{moteur}}$ pour les effecteurs. Chacune évalue localement sa **synergie** avec les autres, de sorte que les liens $\omega_{i,j}$ se **renforcent** s'il apparaît un réel bénéfice à leur coopération. Lorsque la caméra coopère régulièrement avec la pince — par exemple, parce que la zone repérée par

l'image correspond effectivement à la position où la pince intervient —, la liaison $\omega_{\text{cam}, \text{pince}}$ se consolide, engendrant un **cluster** local regroupant ces entités {cam, pince}. Si, au contraire, l'audio se révèle trop bruité, son synergie avec le reste du système diminue et $\omega_{\text{audio}, .}$ tombe en deçà du seuil de pertinence ; l'entité audio se retrouve alors **isolée** ou peu contributive jusqu'à ce que la qualité du signal s'améliore.

On peut intégrer un **signal de récompense** afin de consolider les combinaisons sensorielles et motrices les plus efficaces (par exemple, si le cobot réussit à saisir et assembler la pièce sans heurter de zones sensibles, on valorise cette configuration). Les entités impliquées voient leurs pondérations internes augmenter, ce qui renforce la probabilité de réutiliser ce mode de coopération. L'ensemble constitue un **réseau** qui s'**auto-adapte**, insérant ou affaiblissant des liens selon l'utilité détectée pour la tâche courante. Cette logique évite de reconstruire un pipeline complet ou de réentraîner un vaste réseau en cas de modifications (ajout d'un nouveau capteur, panne de l'actuel, variation des conditions d'assemblage). Les **clusters** multimodaux (vision + pince + capteur de force, par exemple) forment des **sous-réseaux** cohérents, naturellement conçus pour la tâche à effectuer.

1.6.3.5. Défis et Perspectives de Recherche

Lorsqu'on applique le **Deep Synergy Learning (DSL)** à la robotique multi-capteurs et multi-effecteurs, plusieurs difficultés et voies d'extension émergent, tant sur le plan pratique que théorique.

Un premier défi concerne la **complexité combinatoire**. Dans un robot doté de multiples capteurs (caméra, LIDAR, IMU, microphone...) et d'autant de modules d'action (divers joints, roues, pinces), le **graphe** des pondérations ω peut rapidement croître en taille, rendant la mise à jour de chaque lien difficile à maintenir en **temps réel**. Des stratégies de **parsimonie** (seuils éliminant les liaisons trop faibles) ou des mises à jour strictement **locales** (limiter le calcul aux entités concernées par une nouvelle donnée) se révèlent indispensables pour garantir la **scalabilité**.

Un deuxième point aborde les **systèmes multi-robots**. Lorsqu'une flotte (drones, véhicules, robots industriels) doit coopérer, on peut concevoir un **réseau synergique** à plus grande échelle, où chaque robot forme un **sous-graphe** d'entités sensorielles et motrices. Les **synergies** inter-robots introduisent alors des liaisons ω entre entités de différents appareils. Il devient possible d'imaginer une **co-organisation** sur l'ensemble de la flotte, favorisant la dynamique d'émergence de clusters inter-robots dès qu'un gain mutuel apparaît (partage de capteurs, répartition de tâches). L'envergure de ce problème soulève d'importants défis de **communication** et de gestion **distribuée** des liens, car on ne peut pas nécessairement centraliser toutes les informations.

La **sécurité** et la **robustesse** forment également un axe de recherche majeur. En robotique, il est impératif de respecter des contraintes physiques (éviter la collision, maintenir une distance de sécurité). Dans le **DSL**, on peut introduire des pénalités ou des **contraintes** pour réduire la pondération $\omega_{i,j}$ lorsqu'une configuration s'avère dangereuse ou enfreint des invariants mécaniques ou de sûreté. De cette manière, l'**auto-organisation** se soumet à des garde-fous, garantissant un comportement sûr même lorsque la structure du réseau évolue.

Enfin, l'idée d'**hybridation symbolique** (section 1.5.7) se transpose à la robotique avancée et aux tâches haut niveau. Les entités symboliques, représentant des **règles** ou des **plans**, interagissent avec les entités sensorielles et motrices. La synergie entre la **logique** (plan d'action) et la **perception** (reconnaissance visuelle, retour de force) réorganise le graphe de manière dynamique. Cette interaction aboutit à une planification distribuée et **cognitive**, où la coopération se produit entre le symbolique (ordre de mission) et le sub-symbolique (capteurs, effecteurs). L'ensemble de ces pistes ouvre un **vaste** espace de recherche pour faire évoluer la robotique vers des systèmes de décision auto-organisés, hautement modulaires et résilients.

1.6.4. Recommandation Personnalisée et Systèmes de Décision

Au-delà de la robotique (1.6.3), le **Deep Synergy Learning (DSL)** trouve également une place de choix dans les **systèmes de recommandation** et plus largement les **systèmes de décision**. Dans ces domaines, il s'agit souvent de traiter des **profils utilisateurs**, des **contenus** (produits, articles, films, musiques) ou des **options** (plans d'action, configurations possibles), pour aboutir à des **conseils** ou des **choix** pertinents. Les approches traditionnelles, comme le **collaborative filtering**, les **réseaux neuronaux de recommandation** ou les **arbres de décision**, reposent sur des structures fixes. Un modèle est défini, puis entraîné de manière plus ou moins supervisée pour calculer des **scores de préférence**.

Le **DSL**, par sa **dynamique d'auto-organisation** et sa **prise en charge** des liens synergiques, propose une voie plus **flexible et adaptative**. Il peut gérer à la fois :

- Des **entités “utilisateurs”** $\{\mathcal{E}_u\}$ représentant différentes personnes (ou profils),
- Des **entités “contenus”** $\{\mathcal{E}_c\}$ décrivant articles, produits, médias, etc.,
- Des **entités “contextes”** (saisons, heures, tendances récentes),
- Éventuellement des **entités symboliques** (règles de business logic, préférences explicites) si on veut un système cognitif plus avancé.

Nous allons voir en détail comment ces entités interagissent et comment le DSL peut aider à **adapter** les recommandations au fil du temps, en favorisant les **clusters** (ou micro-réseaux) les plus cohérents, et en révisant les connexions obsolètes.

1.6.4.1. Entités Utilisateurs et Contenus : un Graphe Synergique

Les **systèmes de recommandation** s'appuient classiquement sur une **matrice M** dont les lignes représentent les utilisateurs $\{u \in U\}$ et les colonnes les contenus $\{c \in C\}$. Chaque entrée $M_{u,c}$ correspond à l'intérêt ou la note attribuée par l'utilisateur u au contenu c . Les approches courantes (collaborative filtering, factorisation matricielle, autoencodeurs de complétion...) consistent alors à factoriser **M** ou à prédire les valeurs manquantes par un entraînement supervisé ou semi-supervisé. Cette vision reste **matricielle** et se limite généralement au couple **utilisateurs-contenus**, sans intégrer d'autres **entités** susceptibles d'influencer la recommandation.

Dans le **Deep Synergy Learning (DSL)**, on substitue ou complète cette approche en définissant un **graphe** évolutif. On introduit d'abord un ensemble d'**entités** $\{\mathcal{E}_u\}_{u \in U}$ représentant les utilisateurs et un autre $\{\mathcal{E}_c\}_{c \in C}$ pour les contenus (articles, films, produits, etc.). Si nécessaire, on y ajoute aussi des **entités contextuelles** $\{\mathcal{E}_{\text{time}}, \mathcal{E}_{\text{location}}, \dots\}$ lorsqu'on souhaite prendre en considération la dimension temporelle ou spatiale. Les **liaisons** $\omega_{u,c}(t)$ entre un utilisateur \mathcal{E}_u et un contenu \mathcal{E}_c évoluent selon une **règle adaptative** (voir section 1.4.5). Elles se renforcent lorsque la **synergie** $S(\mathcal{E}_u, \mathcal{E}_c)$ est élevée et déclinent en l'absence d'utilité mutuelle. La fonction S repose sur des critères tels que la **similarité** d'intérêts, la **co-information** d'usage ou un gain mesurable, comme le nombre de clics ou de vues.

Grâce à ce schéma, on assiste à la formation spontanée de **clusters** regroupant simultanément des utilisateurs et des contenus. Un **cluster** peut ainsi rassembler plusieurs amateurs de rock, divers albums rock et, s'il existe des entités de contexte, certaines soirées “concert” associées, révélant une forte **co-occurrence** ou un alignement d'intérêts. Comparé à la matrice **M** — qui se borne aux couples (u, c) — le **SCN** tolère l'ajout de toute entité auxiliaire. Il devient possible de **fusionner** des groupes si une synergie apparaît entre leurs membres, ou d'**isoler** un contenu dont l'intérêt décroît soudain auprès des utilisateurs concernés. En outre, l'arrivée d'un nouvel utilisateur $\mathcal{E}_{\text{nouveau}}$ ou d'un nouveau contenu $\mathcal{E}_{\text{item}}$ se gère en insérant simplement l'entité, puis en laissant la synergie ajuster les liens $\omega_{\text{nouveau}, \cdot}$. Le réseau se **restructure** localement, sans imposer une refonte globale comme le ferait une ré-analyse complète de la matrice **M**. Les sections suivantes (voir 1.6.4.2 et au-delà) approfondiront comment cette logique d'auto-organisation enrichit la **recommendation** par la capacité à intégrer de nouvelles entités et à effectuer une fusion dynamique des préférences et des contenus.

1.6.4.2. Mise à Jour Adaptative et Évolution dans le Temps

Dans un **système de recommandation**, la **distribution** évolue continuellement avec l'arrivée de nouveaux utilisateurs et contenus (articles, livres, films récents). Les préférences se modifient en fonction des tendances, des promotions ou des saisons. Les algorithmes classiques de filtering (factorisation, deep learning supervisé) doivent alors réestimer leurs paramètres, au risque d'exiger une nouvelle phase de réapprentissage ou un raffinement partiel, avec le danger de dérégler l'équilibre déjà atteint.

Le **Deep Synergy Learning (DSL)** adopte une démarche plus organique. Lorsque des entités inédites $\mathcal{E}_{\text{nouveauUser}}$ ou $\mathcal{E}_{\text{nouveauContenu}}$ apparaissent, on les **insère** directement dans le **Synergistic Connection Network**. Le mécanisme de mise à jour $\omega_{i,j}$ (section 1.4.5) se charge alors de **consolider** ou non les liaisons selon la **synergie** qu'elles dégagent. Ainsi, un nouveau contenu sera rapidement lié (pondérations positives) aux utilisateurs qui partagent des goûts proches ou qui interagissent avec ce contenu, conduisant à la formation d'un **cluster**. À l'inverse, un contenu ancien qui perd de l'intérêt voit ses pondérations $\omega_{u,\text{contenu}}$ décroître naturellement, jusqu'à s'isoler si plus personne ne s'y intéresse.

De plus, le **DSL** n'est pas cantonné aux seules entités “utilisateur” ou “contenu”. Des **entités contexte** comme la **saison** ($\mathcal{E}_{\text{été}}$), la **localisation** ($\mathcal{E}_{\text{Paris}}$), un **événement** ($\mathcal{E}_{\text{Noël}}$) peuvent influencer la **synergie**. Si la proximité entre un événement (Noël), certains contenus (articles cadeaux, recettes festives) et un profil d'utilisateurs se concrétise, alors les liens $\omega_{\text{Noël}, \text{contenu}}$ et $\omega_{\text{Noël}, \text{utilisateur}}$

s'amplifient, façonnant un **cluster** “Noël + X utilisateurs + Y contenus”. Lorsque la période de Noël s'achève, ces mêmes liaisons perdent peu à peu de leur attrait et se **délient**, laissant place à d'autres configurations contextuelles (telles que “soldes de janvier”). Le réseau se **reconfigure** ainsi localement, sans imposer un réentraînement massif ni une réinitialisation de tout le modèle, et garantit une adaptation continue au gré des changements de **distribution**.

1.6.4.3. Décision et Recommandation Basées sur les Pondérations Synergiques

Une fois le **réseau** de synergie $\{\omega_{i,j}\}$ formé ou relativement stabilisé (au moins à un instant donné), il devient possible d'en **extraire** un score de recommandation pour un **utilisateur** \mathcal{E}_u et un **contenu** \mathcal{E}_c . Une formule concevable, inspirée d'une multiplication matricielle partielle, consiste à sommer l'**influence** indirecte par d'autres entités $\{\mathcal{E}_k\}$ (utilisateurs proches, tags contextuels, règles). On peut ainsi écrire :

$$\text{score}_{u,c} = \sum_k \omega_{u,k} \omega_{k,c}.$$

Intuitivement, si l'utilisateur \mathcal{E}_u et un contenu \mathcal{E}_c ne partagent pas de lien direct, on évalue leur **affinité** en passant par des entités \mathcal{E}_k qui peuvent être d'autres **utilisateurs** (similaires) ou des **concepts** (genre musical, catégorie d'article). Cet agrégat de contributions évoque la structure de $(W^2)_{u,c}$ dans une multiplication de la matrice des pondérations W par elle-même, tout en permettant des **variantes** plus ajustées. Ces adaptations peuvent favoriser les **chemins pertinents** ou exploiter un **cluster** local pour améliorer la recommandation.

Lorsqu'on inclut des **entités symboliques** (section 1.5.7) représentant des **règles** ou **contraintes** métier, comme “Ne pas recommander les contenus PG-13 à un compte junior”, la **synergie** s'ajuste en conséquence. Si la liaison $\omega_{\text{rule}, \text{contenu}}$ signale une incompatibilité, elle diminue la pondération globale $\omega_{\text{user}, \text{contenu}}$, empêchant ainsi des recommandations inappropriées. Il en résulte un **réseau** qui intègre à la fois les **préférences** non supervisées (co-occurrences utilisateurs/contenus) et des **limites** ou **politique** explicites (règles symboliques). Le **SCN** en tient compte localement, ce qui équivaut à un **filtrage** ou à un **re-rank** dynamique, plus flexible qu'un pipeline figé. La **recommandation** se rapproche alors d'une conclusion collective de l'ensemble des entités, chaque liaison ω traduisant la **valeur** ou la **compatibilité** mutuelle.

1.6.4.4. Illustrations Concrètes

Plusieurs exemples concrets permettent de mieux percevoir la **souplesse** et la **richesse** du **Deep Synergy Learning (DSL)** appliqué au domaine des systèmes de recommandation ou, plus largement, à l'**aide à la décision** sur des ensembles utilisateur-contenu-contexte.

Dans une **plateforme e-commerce**, on dispose d'entités représentant les **utilisateurs** (avec leur historique et profil), les **produits** (catégories, marques, etc.), et d'éventuelles entités de **contexte** (temps fort comme soldes, promotions, événements saisonniers). Au lieu de programmer explicitement des algorithmes de segmentation, le **DSL** laisse ces entités s'**agréger** en **clusters** si elles constatent un **gain** mutuel. Un sous-groupe d'utilisateurs peut ainsi s'avérer très intéressé par

des produits d'une certaine marque ou style, formant un **cluster** qui se renforce dès lors que la synergie (p. ex. co-occurrence d'achats) s'avère constante. Les thématiques détectées évoluent naturellement en fonction des tendances ou de l'arrivée de nouvelles gammes de produits.

Dans un **service de streaming** (musique, vidéo), des entités peuvent correspondre aux **albums**, aux **artistes**, aux **playlists**, aux **tags** de genres (rock, pop, jazz), ainsi qu'aux **communautés** d'auditeurs. Le mécanisme de mise à jour $\omega_{i,j}$ valorise la proximité entre un auditeur et un certain artiste, ou entre un artiste et un tag de genre, etc. Un **cluster** cohérent peut alors émerger autour de l'auditoire d'une tendance musicale particulière (rock), intégrant les artistes, les playlists populaires associées et les auditeurs fidèles. Si un nouvel artiste rejoint la plateforme, on ajoute une entité $\mathcal{E}_{\text{artisteNew}}$; si celle-ci partage des caractéristiques avec le cluster rock (timbre de voix, style, tags), la **synergie** se trouve élevée, et l'artiste s'intègre au groupe. Les **recommandations** pour un auditeur donné s'en déduisent en analysant $\omega_{X,\text{artiste}}$ ou bien $(W^2)_{X,\text{artiste}}$ (cf. section 1.6.4.3), reflétant l'influence indirecte par d'autres auditeurs ou tags.

Dans un **système d'aide à la décision** industriel, on peut imaginer des entités décrivant différents **scénarios** (configurations de machines, paramètres de production), des **résultats passés** (qualité, temps de cycle), ainsi que des entités relatives aux **coûts** ou aux **ressources** (personnel, stocks). Les **liaisons** $\omega_{i,j}$ se renforcent lorsqu'une **combinaison** de paramètres s'avère fructueuse pour la production, ou que deux scénarios se révèlent similaires et conduisent à un résultat positif. Le **DSL** construit une **cartographie dynamique** de l'espace décisionnel. De nouveaux scénarios peuvent s'intégrer sans nécessiter de **réapprentissage global**, tandis que l'**auto-organisation** ajuste les liaisons si les conditions évoluent, comme un changement de cadence ou une modification des matières premières. Le tout offre une vision plus **distribuée** et **réactive** que des approches rigides, car les entités ne cessent de mettre à jour leurs liaisons pour refléter la **synergie** détectée.

1.6.4.5. Avantages et Défis

Dans un **système de recommandation** traditionnel, la construction d'un **pipeline** rigide est largement répandue, comme le collaborative filtering avec factorisation de matrice ; on effectue alors une estimation statique ou périodique, puis on l'emploie pour recommander. Le **Deep Synergy Learning (DSL)** adopte une **auto-organisation dynamique**. Lorsque les préférences d'un utilisateur évoluent ou qu'un contenu disparaît, les **connexions** ω s'ajustent localement, sans nécessiter de **recalcul global**. Cette **souplesse** améliore la réactivité à des évolutions telles que les promotions, la saisonnalité ou l'arrivée de nouveaux items.

Le **DSL** assure une robustesse et une parsimonie dans l'évolution du réseau. Les entités qui deviennent inactives, comme des utilisateurs peu actifs ou des contenus obsolètes, voient leurs liaisons ω diminuer naturellement lorsque leur synergie reste trop faible. Elles cessent alors d'interférer avec les autres recommandations, ce qui réduit le bruit et prévient la surcharge computationnelle. Un défi survient toutefois lorsque le nombre d'entités augmente considérablement. Il devient nécessaire d'adopter des heuristiques telles que la sparsification ou l'échantillonnage afin de garantir la scalabilité du réseau et d'éviter une explosion du nombre de connexions.

Le **DSL** permet également une évolution contextuelle fluide. Une nouvelle entité comme "festival d'automne", "temps pluvieux" ou "Black Friday" peut être introduite pour ajuster dynamiquement

la synergie entre les éléments du réseau. Ces entités contextuelles influencent directement les connexions ω et révèlent quels utilisateurs et contenus sont concernés par un événement donné. Aucun réapprentissage massif n'est requis. Cela permet une adaptation continue où les connexions évoluent en temps réel selon les événements, sans imposer de rigidité structurelle.

En dernier lieu, l'**explicabilité** bénéficie d'une amélioration notable. Plutôt que de s'appuyer sur des facteurs latents abstraits difficiles à interpréter, un **cluster** émergent permet une compréhension plus directe. Il regroupe un ensemble d'utilisateurs avec des contenus spécifiques, révélant une structure cohérente et identifiable. Par exemple, une entité représentant le genre musical "jazz" peut se lier naturellement avec une entité "club de blues" et un sous-groupe d'auditeurs partageant ces préférences. Cela offre une explication claire et justifiable. Lorsqu'un utilisateur reçoit une recommandation, il devient possible de formuler une justification explicite comme "vous appartenez au cluster jazz-blues, ce qui explique cette suggestion". Cette approche dépasse la factorisation matricielle classique, apportant une transparence accrue et facilitant l'interprétation des décisions prises par le système.

1.6.5. Surveillance, Diagnostic Médical et Anomalies

Les principes d'**auto-organisation** et de **coopération adaptative** propres au **Deep Synergy Learning (DSL)** ne se cantonnent pas aux domaines précédemment cités (vision, audio, robotique, recommandation). Ils trouvent également une place toute particulière dans les **applications de surveillance, de diagnostic, et de détection d'anomalies**, qu'il s'agisse de contexte médical ou industriel. Dans ces environnements, on gère souvent d'importants **flux de données** en temps réel, provenant de **capteurs variés** (biologiques, physiques) qu'il faut **coordonner** pour repérer des situations anormales ou des symptômes précoces. Les systèmes classiques reposent sur des algorithmes d'**anomaly detection** (souvent supervisés, semi-supervisés, ou basés sur des seuils statiques) qui peuvent se révéler rigides ou trop dépendants d'un jeu de labels.

Le **DSL** apporte une **flexibilité** en structurant des **entités d'information** représentant divers signaux comme des capteurs, des paramètres cliniques ou des logs machines. Ces entités sont reliées par des **pondérations synergiques** qui évoluent au fil du temps, assurant une **auto-organisation** et une adaptation continue aux variations des données. Ainsi, l'**auto-organisation** permet de détecter des groupes d'observations cohérentes et, réciproquement, d'identifier des entités qui s'**écartent** de la synergie générale. Cette capacité de **repérage** local et de **réaction** dynamique rend le **DSL** adapté aux systèmes de **surveillance**, où des capteurs ou des caméras isolent des comportements inhabituels. Il s'avère aussi pertinent pour les **diagnostics médicaux**, en intégrant imagerie, analyses biologiques et relevés de signes vitaux pour identifier des pathologies. Enfin, dans la **maintenance prédictive**, il permet de détecter les anomalies en observant les écarts de certains capteurs par rapport au comportement attendu du **cluster**.

Nous allons voir plus en détail comment le DSL prend en compte la variété des signaux et repère, via la synergie (ou son absence), ce qui sort du lot et pourrait constituer une alerte ou un symptôme.

1.6.5.1. Systèmes de Surveillance : Entités Multi-Capteurs, Scènes Vidéo

Dans de nombreux dispositifs de **surveillance vidéo**, les algorithmes classiques (souvent un pipeline à base de CNN et de suivi d'objets) doivent gérer un **flot** de données considérable et détecter en continu tout comportement ou événement anormal. Cette approche peut se révéler fragile face à la complexité des scènes ou à la diversité des conditions. Le **Deep Synergy Learning (DSL)** propose une alternative plus **auto-organisée** et **réactive**, en représentant chaque composante (segment visuel, contexte, schéma habituel) sous forme d'entités et en laissant la **synergie** orienter la structuration.

Dans le cadre d'une **vidéo** de surveillance, on peut définir des entités $\mathcal{E}_{\text{seg},k}$ correspondant à différents **blobs** ou **segments** détectés dans la scène (régions de mouvement, silhouettes, etc.). D'autres entités $\mathcal{E}_{\text{contexte}}$ décrivent, par exemple, l'heure, la zone surveillée, les règles de fréquentation habituelles. Enfin, des entités $\mathcal{E}_{\text{pattern}}$ peuvent caractériser des **patrons** récurrents (le nombre moyen de personnes, la vitesse de déplacement habituelle, les régions autorisées ou interdites). Au sein du **Synergistic Connection Network**, les liaisons ω évoluent en fonction de la **cohérence** détectée. Si un segment vidéo "blob A" suit un déplacement conforme aux attentes en termes de vitesse et d'emplacement, la synergie reste élevée. Lorsqu'un comportement s'écarte de l'ordinaire (un mouvement inhabituel, une densité de blobs anormale, etc.), la **pondération** $\omega_{\text{seg,pattern}}$ chute, isolant le segment considéré et signalant une anomalie. L'algorithme peut alors émettre une alarme ou enclencher une inspection spécifique, sans nécessiter de recompiler ou de ré entraîner un détecteur complet.

Dans la **surveillance** élargie, d'autres capteurs peuvent être intégrés, comme les détecteurs de mouvement (PIR), les capteurs de bruit, de température ou de fumée. Le **DSL** traite aisément cette hétérogénéité en insérant des entités $\mathcal{E}_{\text{mouvement}}$, \mathcal{E}_{son} , $\mathcal{E}_{\text{température}}$, dont les synergies s'**auto-régulent** avec les entités visuelles. Si un choc violent est détecté par \mathcal{E}_{son} alors que la vidéo ne rapporte aucun mouvement, la liaison $\omega_{\text{son,cam}}$ diminue, suggérant un capteur défaillant ou un phénomène localisé sans trace visuelle. À l'inverse, la convergence de plusieurs capteurs (caméra détectant flammes, capteur son captant un "boom", etc.) renforce les liaisons, formant un **cluster** d'alerte qui intègre différentes sources. Cette **auto-organisation** renforce la **résilience** du système. Les signaux incertains sont naturellement atténués, tandis que les corrélations significatives entre capteurs entraînent une consolidation des liaisons, améliorant ainsi la détection et l'identification des événements critiques.

1.6.5.2. Diagnostic Médical : Entités Physiologiques et Multimodalité



Dans un **contexte médical**, les données proviennent de multiples **sources**. On dispose de l'**imagerie** (IRM, rayons X, scanner), de **tests biologiques** (formule sanguine, biochimie), de **signes vitaux** (ECG, pression, saturation en O₂) et de **symptômes** rapportés, voire de données **génétiques** plus complexes. Un **réseau** de neurones classique, comme un CNN spécialisé dans l’interprétation d’IRM, a tendance à se focaliser sur un **type** de signal. Or, certains diagnostics difficiles (cancers rares, syndromes complexes) exigent un **croisement** entre diverses modalités, chacune partielle, parfois bruitée.

Le **Deep Synergy Learning (DSL)** se conçoit alors comme un **réseau** auto-organisé où des **entités** $\{\mathcal{E}_{\text{irm}}, \mathcal{E}_{\text{analyseSang}}, \mathcal{E}_{\text{ecg}}, \dots\}$ représentent chacune un bloc de données ou un test particulier. Le **score** de synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ indique si deux sources convergent vers une suspicion commune (par exemple, des anomalies visibles à l’IRM qui correspondent à des marqueurs sanguins inhabituels). Les **pondérations** $\omega_{i,j}$ se renforcent lorsque la coopération de deux entités apporte un **gain** (déttection corrélée), et un **cluster** pathologique peut se constituer, regroupant un sous-ensemble de paramètres (imagerie, analyses, symptômes) indiquant une même pathologie ou un syndrome précis.

L’évolution **temporelle** des données médicales s’adapte naturellement aux fluctuations de l’état du patient. Lorsqu’un paramètre revient à la normale, la synergie avec les marqueurs pathologiques s’affaiblit, entraînant une **désolidarisation** progressive du **cluster** associé à la maladie. À l’inverse, un nouveau symptôme peut s’y **agréger** dès qu’il présente une similarité ou une co-occurrence marquée avec d’autres facteurs pathologiques existants. Le **diagnostic** évolue en continu, ajustant la structure du réseau selon l’**état** du patient.

Cette dynamique peut être renforcée par un **signal** de validation clinique. Lorsqu’un diagnostic s’avère correct, les liens responsables de cette détection se consolident, améliorant la robustesse du système. Ce mode d'**auto-organisation** se distingue des pipelines rigides, qui traitent chaque modalité indépendamment (imagerie, analyses biologiques). En intégrant **toutes** les sources de données, les pondérations ω révèlent la **coopération** la plus pertinente à un instant donné, optimisant la prise de décision.

1.6.5.3. Détection d'Anomalies Industrielles : Auto-Organisation et Isolement

En milieu **industriel** (production, logistique, transports), la **maintenance prédictive** implique une surveillance continue des capteurs (température, vibrations, courants moteurs, etc.) afin de détecter des signaux précurseurs de panne. Dans un **Deep Synergy Learning (DSL)**, on peut représenter chaque **capteur** par une entité $\mathcal{E}_{\text{capteur}}$, ajouter des entités décrivant des **normes** ou **règles** de fonctionnement (certaines pouvant être symboliques, voir section 1.5.7), ainsi que des **entités contextuelles** (charge de travail, température ambiante, etc.). Ces entités s'**auto-organisent** et forment un **cluster** principal (ou plusieurs) lorsqu'elles partagent un comportement cohérent.

Une **anomalie** se manifeste lorsque la **synergie** $S(\mathcal{E}_k, \mathcal{E}_{\text{ref}})$ entre un capteur \mathcal{E}_k et le **cluster** de référence (normes, capteurs stables) chute. Contrairement à un système à **seuils fixes** appliqués indépendamment à chaque capteur, le **DSL** évalue chaque capteur par rapport à l'**ensemble** du réseau. Si un capteur **diverge** seul et ne suit plus la synergie habituelle, ses pondérations ω diminuent, l'isolant progressivement et signalant une **anomalie** potentielle. À l'inverse, si plusieurs capteurs entament des dérives **cohérentes**, ils peuvent former un **sous-cluster** spécifique (un "cluster d'anomalie"), laissant deviner un défaut commun (ex. un roulement en fin de vie qui génère bruit et surchauffe).

Au fil du temps, un capteur fautif peut s'**isoler** progressivement si ses pondérations ω_k , diminuent, indiquant une perte de synergie avec le reste du réseau. Il peut aussi entrer en **coopération** avec d'autres entités dérivantes, formant un **cluster** d'anomalie plus large. Si cette situation persiste et que la synergie interne de l'anomalie (par exemple, bruit anormal, vibrations, température de roulement) continue d'augmenter, cela peut déclencher un **signal** de maintenance, suggérant une panne imminente.

Cette **auto-organisation** rend la détection plus **adaptative**. Le **réseau** ajuste progressivement les **liaisons** en renforçant celles qui restent cohérentes et en réduisant celles qui s'écartent, signalant ainsi des anomalies potentielles. On évite ainsi les contraintes d'un pipeline rigide ou de seuils par capteur, car la logique de synergie estime la **relation** du capteur à l'ensemble du **cluster** de fonctionnement normal, autorisant un repérage de dérive **contextualisé** et évolutif.

1.6.5.4. Avantages et Défis : la Logique du DSL en Alerte et Diagnostic

Les applications de **Deep Synergy Learning (DSL)** dans le domaine de la **surveillance**, de la **maintenance** ou du **diagnostic** s'appuient sur sa capacité à constituer des **clusters** de manière autonome, en agrégeant des **entités** multiples (capteurs, tests, imagerie, etc.). Contrairement aux solutions qui demandent une fusion de flux orchestrée manuellement, le **DSL** laisse les pondérations $\omega_{i,j}$ évoluer en fonction de la **synergie** locale, formant des sous-groupes pertinents sans pipeline imposé.

Cette approche confère une **flexibilité** face aux données partielles ou intermittentes. Des capteurs sujets à des interruptions ou à des pannes peuvent se retirer du **cluster** principal dès lors que leur **synergie** avec les autres entités s'effondre, puis se réinsérer lorsque leur fiabilité s'améliore. On évite ainsi qu'un capteur défaillant ne paralyse l'ensemble.

La **détection de situations inédites** (anomalies, configurations nouvelles) bénéficie particulièrement de la logique non supervisée du **DSL**. Quand un groupe d'entités dérive au regard de la norme, il se regroupe spontanément en **cluster** d'anomalie, sans que l'on ait à pré-définir les classes d'incidents. Cela autorise un repérage de schémas insoupçonnés ou de pannes rares.

Les mécanismes d'**adaptation continue** permettent une réactivité en contexte dynamique. Les pondérations ω se réajustent dès qu'un changement est détecté, que ce soit une évolution de l'état du patient en milieu médical ou une transition vers un nouveau régime en milieu industriel. Ce fonctionnement n'exige pas un **réapprentissage** exhaustif à chaque changement.

Cependant, le **principal défi** réside dans la **taille** du graphe et la croissance exponentielle du nombre de liens $\{\omega_{i,j}\}$ quand les entités se multiplient (nombre élevé de patients, de capteurs, de variables, etc.). Il convient d'appliquer des mesures de **parsimonie** (un seuil en-deçà duquel on supprime des connexions), des procédures de **sparsification** dynamique, ou encore des **mises à jour** localisées (échantillonnage partiel, focus sur un sous-ensemble d'entités), pour maintenir la **scalabilité** et la réactivité du système.

1.6.6. Planification et Optimisation dans l'Industrie 4.0

Les secteurs industriels intègrent une **connectivité** croissante, où machines, capteurs, robots, stocks et systèmes de gestion interagissent en temps réel. Cette dynamique incarne la vision de l'**Industrie 4.0**, où l'optimisation repose sur l'échange fluide de données et l'**auto-adaptation** des processus. Les défis qui en découlent — **planification distribuée**, **optimisation** des flux, **maintenance prédictive** et **gestion adaptative** — peuvent se révéler extrêmement complexes. Les méthodes traditionnelles (ordonnancement déterministe, programmation linéaire classique, heuristiques fixes, etc.) montrent souvent leurs limites face à la **dynamique** croissante (marchés volatils, nouvelles commandes urgentes, pannes inattendues). Dans ce cadre, le **Deep Synergy Learning (DSL)**, avec son **mécanisme d'auto-organisation** et de **synergie adaptative**, ouvre de nouvelles pistes pour réaliser une **planification plus flexible** et des **schémas d'optimisation** capables de se réajuster en continu.

Dans cette section (1.6.6), nous examinons :

- Les raisons pour lesquelles l'**Industrie 4.0** exige un modèle **évolutif** et **distribué**,
- Comment le **DSL** modélise des entités (machines, stocks, commandes, flux de transport) reliées par des **pondérations synergiques**,
- De quelle manière ces liens évoluent pour **planifier** et **optimiser** les ressources en temps réel,
- Des exemples concrets où le **DSL** peut surpasser un pipeline d'optimisation statique, grâce à sa **dynamique d'adaptation** et à sa capacité de **clustering** auto-organisé.

1.6.6.1. Les Défis Industriels dans un Environnement 4.0

Dans l'Industrie 4.0, on fait face à une **abondance** de composants interdépendants. Les **machines** possèdent chacune un état (vitesse, disponibilité, pannes potentielles), tandis que les **chaînes de production** sont subdivisées (stations, lignes, buffers, flux de pièces). Les **stocks** (pièces détachées, semi-finis, produits finis) changent au fil des demandes, et le **transport** peut impliquer différents véhicules (AGV, drones, camions). Enfin, les **ordres de fabrication** évoluent dès lors que des clients passent de nouvelles commandes ou modifient leur planning.

Dans les approches classiques d'**ordonnancement** ou de **planification** (flow shop, job shop, heuristiques variées), on définit un **problème combinatoire** qu'on tente de résoudre sous des **hypothèses** de stabilité. En pratique, l'environnement évolue en permanence. Les machines peuvent tomber en **panne**, un retard logistique peut compromettre la disponibilité d'une pièce critique, de nouvelles **commandes** peuvent arriver de manière imprévue, et la **capacité** des machines peut fluctuer en raison de maintenances ou de changements d'outillage. Chaque perturbation exige souvent un **réajustement** conséquent, voire une **réinitialisation** complète d'un algorithme d'ordonnancement.

Le **Deep Synergy Learning (DSL)** propose un **changement de perspective**. Au lieu de modéliser un ordonnancement global, on conçoit chaque **composant** — machine, ordre, stock, mode de transport — comme une **entité** \mathcal{E}_i . Les **liaisons** $\omega_{i,j}(t)$ traduisent la **coopération** ou la **complémentarité** entre entités. Par exemple, deux machines $\mathcal{E}_{\text{machineA}}$ et $\mathcal{E}_{\text{machineB}}$ voient leur lien s'intensifier si elles se passent fréquemment des pièces sans blocage, ou si l'on constate un gain productif à enchaîner les opérations A puis B. Un **stock** $\mathcal{E}_{\text{stock}}$ et un **transport** \mathcal{E}_{AGV} coopèrent s'ils se synchronisent efficacement pour déplacer des lots. De même, un **ordre** $\mathcal{E}_{\text{ordre}}$ accroît sa synergie avec une machine spécifique si cette dernière exécute la tâche en temps et qualité optimales.

Lorsqu'un **incident** surgit (panne machine, retard de livraison, surcharge de commande), la **dynamique d'auto-organisation** du DSL réévalue les liaisons ω . Un composant en panne voit ses liens se dégrader, réduisant son impact dans le flux. Inversement, si une nouvelle machine arrive ou si un transport reprend sa disponibilité, on insère une **entité** inédite dans le **SCN**, laissant la synergie se construire là où un **gain** apparaît. Cette **reconfiguration** s'effectue **localement**, évitant de relancer un algorithme d'ordonnancement global. Le **réseau** s'ajuste de manière “vivante”, reflétant la réalité mouvante de l'Industrie 4.0.

1.6.6.2. Représentation des Flux et Clusters Auto-Organisés

Dans une optique d'**Industrie 4.0**, la mesure de **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ entre deux entités \mathcal{E}_i et \mathcal{E}_j peut prendre plusieurs formes. Les machines, les stocks ou les ordres de production entretiennent par exemple un **taux de transfert** élevé lorsque la première fournit régulièrement des pièces à la seconde sans créer de file d'attente. Un **ordre** de production se révèle quant à lui **complémentaire** d'un stock donné si cet ordre requiert précisément les ressources contenues dans ce stock. Par ailleurs, la **performance** d'une association machine–ordre s'accroît lorsque le couple aboutit à un temps de cycle réduit et à un rendement supérieur. Lorsque la **synergie** apparaît remarquable entre

deux entités, la pondération reliant ces entités se trouve renforcée (voir section 1.4.5 sur la mise à jour locale des pondérations).

À mesure que certaines combinaisons s'avèrent profitables (machine-stock-ordre-transport, par exemple), leur coopération se **solidifie**, faisant émerger des **clusters** ou sous-réseaux. Chaque **cluster** correspond alors à une configuration de production considérée stable ou efficace, puisqu'il regroupe des entités dont les pondérations mutuelles ω sont élevées. On observe ainsi la formation d'un **cluster** regroupant {Machine1, StockA, TransportX, OrdreB}, car ces éléments s'articulent en un **flux** cohérent. La Machine 1 puise ses ressources dans le Stock A, le Transport X assure la logistique en temps réel, et l'Ordre B bénéficie d'un traitement optimisé.

Dans le cadre d'une planification traditionnelle, on conçoit des **lignes** ou **cellules** de production de manière analytique. Le **Deep Synergy Learning (DSL)** repose sur une **auto-organisation dynamique** où les **liaisons** se renforcent lorsque des machines, des stocks ou des modules de transport démontrent une **compatibilité fonctionnelle**. Cette dynamique favorise l'émergence **spontanée de clusters** optimisés, sans nécessiter une configuration prédefinie du système. Quand un composant (machine, stock ou transport) **dysfonctionne** ou s'avère insuffisamment utile, la **synergie** avec le reste du réseau chute et le composant se trouve mis à l'écart. De nouveaux liens peuvent alors se former vers des entités de **remplacement**, réorientant la production de façon plus souple qu'un pipeline rigide. Ce phénomène procure à l'ensemble une **plasticité** qui évoque un système "vivant", réactif aux perturbations et aux changements d'outillage ou de répartition des tâches, conformément à l'esprit de l'Industrie 4.0 décrite en section 1.6.6.1.

1.6.6.3. Approche Dynamique : Adaptation en Temps Réel

Dans un environnement **Industrie 4.0**, le système gère un **flux continu** d'événements tels que nouvelles commandes, variations de stocks, maintenances ou retards logistiques. Le **DSL** ajuste en **temps réel** les **liaisons synergiques** entre machines, ressources et transports. Les connexions utiles se renforcent tandis que les moins pertinentes s'affaiblissent, assurant une **auto-organisation dynamique** sans réajustements manuels. Les algorithmes de planification classiques (règle de Johnson, heuristiques de scheduling, etc.) se montrent performants pour un **problème** donné (flow shop, job shop), mais perdent en **flexibilité** lorsqu'on modifie le problème à la volée, car il faut alors réinitialiser ou réactualiser le schéma de planification.

Au **Deep Synergy Learning (DSL)**, la **dynamique** d'auto-organisation se renouvelle en continu. Les **pondérations** $\omega_{i,j}(t)$ reliant des entités (machines, ordres, stocks, transports...) se remettent à l'itération après itération en fonction de la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. Lorsqu'un **ordre urgent** survient, le système introduit une entité $\mathcal{E}_{ordreUrgent}$ dans le **Synergistic Connection Network**, et celle-ci voit ses **liaisons** $\omega_{ordreUrgent, k}$ se renforcer rapidement dès que l'on identifie un gain potentiel avec des machines ou des transporteurs capables de le traiter à temps. Si, au contraire, un transport $\mathcal{E}_{transport}$ se montre sous-dimensionné (accumulation de pièces, délais excessifs), sa **synergie** avec le reste du réseau chute, ce qui oriente la production ou la logistique vers d'autres entités plus aptes à satisfaire la demande.

Le **DSL** ne cherche pas à figer une **solution** optimale à un instant donné comme un algorithme de scheduling. Il maintient un **réseau dynamique**, où les **clusters** de machines, ordres et stocks

s'adaptent aux événements. En intégrant un **signal global** de performance comme le throughput, le temps de cycle ou le taux de satisfaction client, le système ajuste la **synergie** de manière dynamique. Les entités impliquées dans une configuration efficace renforcent leurs **liaisons** ω , tandis que celles associées à une baisse de performance voient leurs connexions s'affaiblir. Cette adaptation continue permet une **auto-organisation** où chaque entité ajuste sa coopération localement, sans nécessiter une refonte complète de la planification à chaque nouvel incident.

1.6.6.4. Cas d’Exemple : Usine Modulable

Dans une **usine** où les postes sont **reconfigurables** et la logistique assurée par des **AGV** (véhicules autonomes), les produits fabriqués peuvent changer selon la **demande**. Le **Deep Synergy Learning (DSL)** s'avère alors bénéfique pour une **gestion** plus flexible, en remplaçant ou en complétant les méthodes de planification traditionnelles. On peut définir un **réseau** d'entités $\{\mathcal{E}_{\text{Poste}1}, \mathcal{E}_{\text{Poste}2}, \dots, \mathcal{E}_{\text{AGV}1}, \mathcal{E}_{\text{AGV}2}, \dots, \mathcal{E}_{\text{Stock}1}, \dots\}$ et des entités “**ProduitA**”, “**ProduitB**” (avec la possibilité d'en ajouter de nouveaux). Chaque poste ou AGV entretient des **liaisons** $\omega_{i,j}$ qui se **renforcent** ou se **délient** selon la **synergie** constatée. Si un **poste** particulier traite un certain **produit** avec un taux de réussite élevé, la pondération $\omega_{\text{Poste}, \text{Produit}}$ augmente. De même, un **AGV** se connecte aux postes ou stocks avec lesquels il collabore efficacement, formant ainsi un **cluster** {AGV1, Poste1, Poste2, Stock} si ce groupe de composants assure un flux fluide de pièces.

Lorsqu'on introduit un **nouveau** produit (par exemple “**ProduitC**”), la synergie se construit **naturellement** autour des postes adéquats. Le **SCN** évolue sans qu'il soit indispensable de reconfigurer entièrement le pipeline ou de lancer un **solveur** d'ordonnancement intégral. Les entités reliées à l'ancien produit peuvent **conserver** leurs liens, tandis que les postes ou AGV inadaptés à “**ProduitC**” voient leur pondération chuter et se déconnectent graduellement du nouveau cluster. Cette **dynamique** garantit une **réactivité** efficace. En cas de **panne** sur Poste1, la pondération $\omega_{\text{AGV}, \text{Poste}1}$ diminue, incitant l'AGV à se rediriger vers un autre poste, comme $\mathcal{E}_{\text{Poste}3}$. La **synergie** avec ce nouveau poste augmente alors naturellement, optimisant ainsi l'adaptation du réseau.

Ce **mécanisme d'auto-organisation** continue favorise la **tolérance** aux pannes et l'adaptation à l'introduction de nouvelles gammes de produits. Les entités inopérantes se “**détachent**” naturellement (perdent leurs liaisons), et les nouvelles entités intéressantes (nouveaux postes, nouveaux produits) trouvent leur place dans le **Synergistic Connection Network** au fil de la **dynamique** des pondérations. Cela évite la nécessité de **recalculer** périodiquement un ordonnancement global ou de procéder à des heuristiques lourdes, tout en exploitant les **principes** du DSL (sections 1.4.3 et 1.5.4) pour maintenir la cohérence globale du système.

1.6.6.5. Limites et Évolutions : Complexité, Parcimonie, Hybridation

Dans un **environnement industriel** où l'on introduit sans cesse de nouvelles entités (machines, stocks, ordres...), la **taille** du graphe $\{\omega_{i,j}\}$ peut rapidement devenir prohibitive. Pour préserver la faisabilité, il est essentiel d'envisager des stratégies de **parsimonie** (seuils, coupe des liens trop faibles) ou des **algorithmes locaux** où seules les pondérations pertinentes se mettent à jour, évitant ainsi un recalcul exhaustif à chaque itération. Des **contraintes** peuvent également exclure certains

liens d'emblee (une machine non qualifiée pour un ordre, ou un transport incapable de prendre en charge un certain poids), imposant une **pénalisation** ou un **masquage** dans la fonction de synergie.

De plus, lorsque des **règles logiques** (contraintes de sécurité, de qualité) s'ajoutent, on peut intégrer des **entités symboliques** (cf. section 1.5.7). Le **DSL** considère alors la **coopération** entre ces entités symboliques et les entités opérationnelles (machines, ordres...), de sorte que la formation d'un **cluster** illégal (par exemple, un groupement violent une norme de sécurité) n'aboutisse jamais, faute de synergie. Cette hybridation **neuro-symbolique** enrichit la dynamique locale (ω) tout en respectant les cadres réglementaires ou métier.

Il est par ailleurs envisageable de **combiner** le **DSL** à des heuristiques ou solveurs d'ordonnancement établis. Une possibilité consiste à laisser l'**auto-organisation** déterminer un sous-ensemble de ressources potentiellement efficaces, puis à faire appel à un **solveur** classique pour finaliser le planning sur ce sous-ensemble, réduisant la dimension du problème. À l'inverse, un solveur peut fournir une solution initiale, que le **DSL** ajuste localement au gré des aléas (pannes, nouvelles commandes). Cette **collaboration** exploite la force de l'**auto-organisation** (réactivité, plasticité) tout en bénéficiant de méthodes d'**optimisation** éprouvées pour parfaire le résultat. Ainsi, la **plasticité** du DSL et la **rigueur** de l'optimisation se conjuguent pour répondre aux exigences variées de l'Industrie 4.0.

1.6.7. Perspectives pour la Recherche Fondamentale en IA Forte

Les précédentes sous-sections (1.6.1 à 1.6.6) ont démontré la **portée pratique** du Deep Synergy Learning (**DSL**) dans divers domaines applicatifs (vision, audio, robotique, recommandation, diagnostic, etc.). Cependant, l'**ambition** du DSL ne se limite pas à une simple amélioration des performances ou de la flexibilité dans ces tâches spécialisées. À un niveau plus **fondamental**, l'approche synergique **ouvre des pistes** de recherche susceptibles de rapprocher l'IA de la notion d'**IA Forte** (ou **IA Généralisée**), c'est-à-dire d'une intelligence capable d'**apprendre** et de **raisonner** de manière autonome et générale, au-delà de cadres strictement définis. Cette section (1.6.7) explore plusieurs **perspectives** théoriques et conceptuelles. D'abord, le **DSL** et ses **mécanismes d'auto-organisation** présentent des similarités avec les **systèmes cognitifs**, qu'il s'agisse du cerveau ou d'écologies d'informations. Ensuite, il offre la possibilité d'**intégrer** à la fois des dimensions sub-symboliques, reposant sur l'apprentissage à partir de données massives, et des **aspects symboliques** ou logiques (voir 1.5.7). Par ailleurs, des **propriétés émergentes** se manifestent, telles que la formation de représentations, de micro-réseaux cognitifs et de processus d'auto-adaptation qui dépassent la simple exécution d'une tâche. Enfin, l'approche soulève plusieurs **défis mathématiques** et philosophiques liés à l'ambition d'une **IA Forte** fondée sur le **DSL**.

1.6.7.1. Au-delà de l'Apprentissage Supervisé : Vers l'Auto-Construction de la Connaissance

Une large part des avancées récentes en **IA** se fonde sur l'**optimisation globale** de réseaux neuronaux massifs par **rétropropagation**, qu'il s'agisse de données annotées en abondance ou d'un pré-entraînement auto-supervisé (approche de type GPT, BERT). Bien qu'efficaces, ces modèles conservent plusieurs caractéristiques. Ils reposent sur une **architecture** linéaire ou

faiblement hiérarchisée, même en présence de mécanismes comme les multi-têtes d'attention. Leur optimisation dépend d'une **descente de gradient** ou d'une procédure équivalente visant à minimiser un coût défini globalement. Enfin, leur **adaptabilité locale** reste restreinte. Lorsque la distribution des données change, un **réapprentissage** par fine-tuning devient souvent indispensable, ce qui expose le modèle au risque d'un **écrasement partiel** des connaissances antérieures, connu sous le nom de **catastrophic forgetting**.

Le **Deep Synergy Learning (DSL)** présente une **dynamique d'auto-organisation** plus proche de l'esprit de **systèmes complexes** (biologiques ou écologiques), où chaque **entité** (sous-module, neurone, flux d'information) gère ses **liaisons** suivant la **synergie** qu'il perçoit avec d'autres entités. Plutôt que de minimiser une **unique fonction de coût**, on autorise l'émergence de **clusters** et la **reconfiguration permanente** des connexions $\omega_{i,j}$. Cette approche "**bottom-up**" s'avère potentiellement féconde pour développer des **mécanismes cognitifs** plus avancés et moins tributaires d'un lourd apprentissage supervisé.

Dans une perspective d'**IA Forte**, on envisage souvent un agent qui **accumule** graduellement des connaissances sans les oublier, tout en **réinterprétant** ces savoirs au fil des expériences. Le **DSL** y répond en laissant chaque **entité** conserver sa mémoire locale (paramètres, historique) et en **réévaluant** les liaisons ω de façon continue. On obtient ainsi un processus **incrémental** où de nouvelles **entités** émergent, représentant des concepts ou des données. Les **liaisons** se **créent** ou se rompent en fonction de la **pertinence** et de la **coopération** mesurée. Les **clusters** de connaissances se **stabilisent** lorsque la synergie interne reste forte, ou se **scindent** si celle-ci décline. L'agent parvient ainsi à **auto-structurer** ses connaissances, à l'image d'un cerveau qui forme et dissout continuellement des **assemblées** neuronales en fonction des interactions et de l'expérience.

Cette organisation, plus **distribuée** et **évolutive**, pose les bases d'une **auto-construction** de la connaissance, où l'on n'est plus tributaire d'un unique label ou d'une unique fonction de perte, mais d'une dynamique **locale** entre les **entités** qui renforcent ou amoindrissent leurs liens au gré de leur coopérativité. Elle suggère la possibilité d'un **savoir** plus flexible et cumulatif, réduisant la dépendance à un entraînement exhaustif et favorisant une **adaptation** continue aux données nouvelles ou aux changements de contexte.

1.6.7.2. Vers une Cognition Distribuée et Émergente

Le **Deep Synergy Learning (DSL)** trouve ses racines (section 1.3.1) dans l'observation des **systèmes biologiques** exhibant de la **plasticité** (synaptique chez les neurones), ou des phénomènes **d'auto-organisation** (colonies d'insectes, tissus cellulaires). Dans la quête d'une **IA Forte**, un **réseau** d'entités peut servir de **substrat cognitif** en intégrant plusieurs dimensions.

Les **entités** \mathcal{E}_i regroupent aussi bien des *features* perceptuelles issues de flux sensoriels que des **concepts abstraits** ou des **modules symboliques** encapsulant des règles logiques.

Les **liaisons synergiques** $\omega_{i,j}$ se renforcent selon un mécanisme inspiré du renforcement "hebbien généralisé" dès lors qu'une **coopération** entre entités procure un **gain**, qu'il s'agisse de performance, de co-information ou de simplification.

Les **clusters** qui émergent forment des circuits cognitifs spécialisés, participant au traitement conceptuel, à la mémoire de travail, à la planification ou à l'intégration de signaux multiples.

Au sein de ce **substrat**, chaque **cluster** reflète potentiellement une micro-théorie ou un schéma conceptuel, regroupant des entités pour former une **hypothèse**, un **concept**, voire un **plan d'action**. La structuration évolue de manière **dynamique**. Les assemblées émergent ou se dissolvent en fonction de la synergie détectée, permettant une **reconfiguration continue** des connaissances et des idées. Contrairement aux architectures statiques à couches fixes, cette organisation s'**auto-régule**, équilibrant **plasticité** (création et suppression de liens) et **stabilité** (maintien des clusters pertinents).

En examinant la **dynamique** globale, on observe l'émergence de **processus cognitifs** spécifiques. Une **forme d'attention** se manifeste lorsqu'un sous-réseau capte et monopolise la synergie disponible, créant un **focus transitoire** qui oriente temporairement le traitement des informations. Des **boucles** de rétroaction positives ou négatives peuvent générer des **oscillations**, analogues à des rythmes cognitifs ou des alternances d'états mentaux. La **conservation** d'équilibres permet au système de préserver des organisations robustes, tout en se montrant apte à accueillir de nouvelles liaisons ou à rompre les moins pertinentes.

De tels phénomènes, absents d'un CNN ou d'un Transformer (qui restent malgré tout dans une structure de pipeline, bien que complexe), s'apparentent plutôt à un **processus cognitif distribué**, dans lequel l'information chemine entre assemblées modulaires qui se constituent et se dissolvent librement. On entrevoit là un **jalon** vers des agents en mesure de **construire** et de **réviser** en continu leurs hypothèses ou "théories", de manière plus proche de la **biologie** et des **neurosciences** que de la minimisation stricte d'une fonction de coût globale.

1.6.7.3. Intégration Symbolique : Socle pour le Raisonnement Abstrait

Dans une **démarche d'IA Forte**, il ne suffit pas d'analyser des signaux sub-symboliques comme les images, les capteurs ou les séquences temporales ; il faut aussi être capable de manipuler des **représentations abstraites** et des **règles logiques complexes**. Les réseaux neuronaux profonds, bien qu'efficaces sur des volumes massifs de données, éprouvent des difficultés à exécuter un raisonnement explicitement symbolique, sauf si on leur adjoint des modules dédiés. Le **Deep Synergy Learning (DSL)** propose un **cadre uniifié** où des entités **symboliques** (comme \mathcal{E}_{rule} , $\mathcal{E}_{concept}$) coexistent avec des entités **sub-symboliques** (\mathcal{E}_{sensor} , $\mathcal{E}_{feature}$), comme décrit à la section 1.5.7. Lorsqu'une **règle** s'applique à une **configuration perceptive**, la liaison $\omega_{rule,sensor}$ se renforce dès que la synergie entre elles se révèle fructueuse (information mutuelle élevée, gain de performance, amélioration de la précision). Il en résulte un **micro-réseau** où un ensemble de règles se coordonne avec un ensemble d'observations, sans qu'il soit nécessaire d'établir un cloisonnement strict entre logique et perception.

Dans cette perspective, la dynamique des pondérations ω ne se limite pas au rapprochement d'entités visuelles ou auditives. Elle permet également de **connecter des concepts abstraits à des indices sensoriels** confirmant leur validité ou, à l'inverse, d'affaiblir une règle qui ne s'accorde pas aux observations. Cela conduit à un **raisonnement logico-perceptif fluide**. Lorsqu'une hypothèse symbolique \mathcal{E}_{rule} reçoit un soutien sensoriel de \mathcal{E}_{sensor} , leur liaison grandit et un **cluster** se forme, fusionnant la dimension symbolique et la dimension perceptive. Cette approche dispense

d'un connecteur artificiel entre un réseau neuronal et un moteur logique indépendant. C'est le **réseau lui-même**, via la mise à jour des synergies, qui assure l'unification de ces deux niveaux de représentation. On obtient ainsi une **plasticité accrue**, où les règles symboliques peuvent être ajustées ou rejetées en fonction de leur compatibilité avec les données sensorielles. Elles se reforment ou disparaissent au sein du **SCN**, s'adaptant en fonction des synergies détectées. Cette flexibilité offre un **socle pour un raisonnement abstrait ancré dans la perception**, une condition essentielle pour l'**IA Forte**, qui doit être capable d'extraire des modèles du monde tout en manipulant des structures logiques évolutives.

1.6.7.4. Défis Mathématiques : un Pas vers une Théorie de la Cognition Synergique

Le **Deep Synergy Learning (DSL)** peut se voir comme un **système dynamique** non linéaire de grande dimension, dont la **matrice** $\Omega(t)$ (ou un hyper-graphe) évolue suivant une loi similaire à

$$\Omega(t+1) = \Omega(t) + \eta [S(\Omega(t)) - \tau \Omega(t)],$$

où $\Omega(t)$ désigne l'ensemble des pondérations $\{\omega_{i,j}(t)\}$ à l'instant t , et $S(\Omega(t))$ symbolise la mesure de **synergie** calculée entre les entités, qui peut elle-même dépendre de l'état global. Cette dynamique est susceptible de générer des **attracteurs** multiples (plusieurs organisations cognitives possibles), des **bifurcations** lorsque les synergies changent brusquement, voire des **cycles limites** interprétables comme des rythmes cognitifs ou des alternances d'états mentaux. L'**analyse formelle** (existence et stabilité d'attracteurs, transitions critiques, phénomènes d'hystérèse) reste toutefois un **défi** majeur, notamment si l'on souhaite déployer de nombreuses entités ou autoriser une définition non triviale de la synergie (information mutuelle n-aire, PID, etc.).

Pour gérer la croissance potentiellement explosive du nombre de liaisons $\{\omega_{i,j}\}$ et leur mise à jour en temps réel, on se tourne fréquemment vers des **méthodes de parcimonie** (seuils, coupes, hiérarchies de clusters) ou des **stratégies** d'échantillonnage. Certains mécanismes inspirés de la **biologie** — par exemple un terme d'oubli rapide ou une saturation — contribuent également à la **scalabilité**. Le choix précis de la fonction de synergie S , de la règle de mise à jour et de la topologie initiale influe sur la façon dont le réseau s'organise.

En parallèle, les travaux en **sciences cognitives** ou neurosciences computationnelles (comme la **théorie de l'information intégrée** de Tononi) proposent de quantifier la **part** d'information réellement unifiée dans un système. Dans l'esprit du **DSL**, on peut définir une **intégration globale** à partir de la **somme** des synergies (ou d'une co-information n-aire) :

$$\text{Intégration globale} = \sum_{i,j} \omega_{i,j} S(\mathcal{E}_i, \mathcal{E}_j),$$

ou encore effectuer une version plus générale au sein d'un **cluster** n-aire. Cette démarche veut évaluer la “richesse” émergente dans un réseau en perpétuelle auto-organisation. Elle ouvre un **pas** vers des réflexions sur la **conscience artificielle** ou l'**autonomie** inspirée de la cognition humaine, puisqu'elle quantifie à quel degré le réseau regroupe et **intègre** l'information sous forme de micro-réseaux cognitifs, susceptibles d'exhiber des propriétés d'**émergence** et de **complexité** caractéristiques des systèmes vivants.

1.7. Défis, Contraintes et Ouvertures

Les sections précédentes (1.1 à 1.6) ont mis en lumière la **logique fondamentale du Deep Synergy Learning (DSL)**, ses **mécanismes** d'auto-organisation et ses **capacités** à apporter des solutions nouvelles dans de multiples domaines applicatifs (vision, audio, robotique, recommandation, diagnostic, etc.). Toutefois, comme toute approche novatrice, le DSL n'échappe pas à des **défis** et des **contraintes** majeurs. Sur le plan **computational, théorique, méthodologique et éthique**, de nombreuses questions demeurent ouvertes, exigeant une recherche soutenue pour perfectionner le paradigme et garantir son insertion fluide dans des environnements industriels, médicaux, ou encore scientifiques.

La volonté du DSL de fonctionner sans architecture hiérarchique figée, et de laisser les entités **s'auto-organiser** via des synergies, requiert notamment :

- Une **scalabilité** adéquate (comment passer à des milliers ou millions d'entités sans explosion de la complexité ?),
- Une **qualité** et une **accessibilité** suffisantes des données (pour estimer la synergie de façon robuste),
- Des **algorithmes d'optimisation** aptes à gérer des systèmes dynamiques, non linéaires et potentiellement hétérogènes,
- Un **contrôle** de la stabilité et de la convergence (éviter les oscillations ou l'effondrement du réseau),
- Une **interprétabilité** pour l'humain, d'autant plus cruciale lorsque les décisions impactent des vies ou des processus critiques,
- Des **considérations éthiques** et réglementaires pour encadrer l'usage de systèmes auto-organisés, notamment en termes de responsabilité, d'acceptabilité, ou de gestion de biais,
- Des **comparaisons expérimentales** rigoureuses avec les approches existantes (réseaux neuronaux profonds, méthodes symboliques, algorithmes d'optimisation classiques) pour valider la pertinence et les conditions d'efficacité du DSL.

Dans cette section (1.7), nous examinons un à un ces **défis, contraintes et ouvertures**, déclinés en sept points :

- **Complexité Computationnelle et Scalabilité** (1.7.1)
- **Qualité et Disponibilité des Données** (1.7.2)
- **Développement d'Algorithmes d'Optimisation Appropriés** (1.7.3)
- **Contrôle et Stabilité des Processus Auto-Organisés** (1.7.4)
- **Interprétabilité et Explicabilité pour l'Humain** (1.7.5)
- **Considérations Éthiques et Réglementaires** (1.7.6)

- **Comparaisons Expérimentales avec d’Autres Approches** (1.7.7)

Ce panorama permettra de saisir la **complexité** inhérente à l’approche synergique, tout en mettant en relief les **axes** de recherche et les **collaborations** scientifiques qu’elle appelle, pour faire du DSL un cadre de plus en plus solide dans l’écosystème de l’IA et de l’apprentissage automatique.

1.7.1. Complexité Computationnelle et Scalabilité

La première critique ou préoccupation qu’on peut formuler à l’égard du **DSL** concerne la **taille** potentielle du **graphe** des entités et la **complexité** des opérations nécessaires pour **calculer** ou **mettre à jour** les **pondérations** synergiques ($\omega_{i,j}$). En effet, si l’on considère n **entités** $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$, avec une **synergie binaire** $S(\mathcal{E}_i, \mathcal{E}_j)$ pour chaque paire (i, j) et une **mise à jour régulière**, la **complexité brute** peut rapidement atteindre $O(n^2)$ par itération. Cette croissance quadratique devient impraticable lorsque n atteint plusieurs milliers ou millions de composantes. De plus, si l’on veut aller vers des **synergies n-aires** (pour capturer des coopérations à plusieurs entités simultanément), la situation s’aggrave encore (complexité exponentielle).

1.7.1.1. Problématique du “Tout Relier à Tout”

L’un des principes centraux du **Deep Synergy Learning (DSL)** est de laisser les **entités** $\{\mathcal{E}_i\}$ se connecter ou se déconnecter au fil du temps, suivant la **synergie** perçue. Pourtant, permettre qu’une entité puisse se relier à toutes les autres sans limite conduit vite à un **graphe complet**, où chaque paire (i, j) exige un calcul de synergie $S(\mathcal{E}_i, \mathcal{E}_j)$. Cette démarche devient rapidement ingérable :

- Le calcul de $S(\mathcal{E}_i, \mathcal{E}_j)$ pour toutes les paires (i, j) croît en $O(n^2)$.
- La mise à jour des pondérations $\omega_{i,j}$ (section 1.4.5) présente la même complexité en $O(n^2)$.
- Le stockage de l’ensemble $\{\omega_{i,j}\}$ requiert une **mémoire** en $O(n^2)$.

Dans des applications **massives** (robotique multi-capteurs, systèmes de recommandation avec des millions d’utilisateurs et de contenus, vision distribuée avec d’innombrables descripteurs), la construction d’un tel graphe complet compromet la **scalabilité** du DSL. Il devient crucial de maintenir une certaine **parsimonie** au sein du réseau, de sorte à ne pas créer aveuglément des liaisons $\omega_{i,j}$ pour chaque paire.

Une **stratégie** commune consiste à **restreindre** le calcul de synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ aux seules paires dont les entités se révèlent “**assez proches**” ou déjà **corrélées**. On peut, par exemple, refuser la création de liens si la distance entre \mathbf{x}_i et \mathbf{x}_j dépasse un certain **seuil** ϵ . On peut aussi confiner le calcul à un **voisinage** k-plus-proches-voisins, limitant la croissance de connections. Par ailleurs, il faut un **mécanisme** pour découvrir graduellement de nouvelles liaisons, via un **échantillonnage** ou une **partition** initiale, sans explorer d’emblée toutes les paires (i, j) .

En parallèle, on définit des **règles** pour **supprimer** ou **inactiver** automatiquement les liens jugés trop faibles ou non pertinents (auto-suppression sous un seuil ω_{\min}). De cette manière, la densité du graphe reste modérée. On vise alors une complexité proportionnelle au nombre de liaisons

“actives”, ce qui peut être bien inférieur à n^2 . Au bout du compte, le **réseau** se rapproche d’un **k-NN** ou d’un ϵ -**radius** dynamique, sur lequel on applique les mises à jour (section 1.4.5). Le **DSL** y opère donc ses principes d’auto-organisation et de synergie, tout en évitant l’écueil d’un graphe complet ingérable.

1.7.1.2. Calcul de Synergie : Coût des Mesures Informationnelles

Lorsque la **synergie** prend la forme d’une **co-information** ou d’une **information mutuelle** entre entités modélisées comme variables aléatoires, on se heurte à la difficulté du calcul ou de l’estimation de grande dimension. Les estimateurs non paramétriques (k-NN, Kernel density) sont souvent coûteux et la quantité de données exigée peut augmenter de manière exponentielle avec la dimension (curse of dimensionality\text{curse of dimensionality}). Pour rendre ces **métriques d’information** plus utilisables en pratique, plusieurs stratégies sont envisageables. Il est possible de **limiter** la dimension en extrayant des **features** plus compactes à l’aide d’un autoencodeur ou d’une PCA, ce qui réduit la charge des estimateurs d’entropie et de co-information. Une autre option consiste à **recourir** à des mesures de synergie plus **légères**, comme la distance euclidienne ou la similarité cosinus, même si cela implique une perte de richesse par rapport à l’information mutuelle stricte. Enfin, lorsqu’il est essentiel de conserver une approche basée sur l’**information théorique**, on peut **employer** des heuristiques d’approximation, comme le **downsampling** des données, une modélisation paramétrique (Gaussienne, mixture Gaussienne) ou des estimateurs simplifiés.

Ces méthodes aident à concilier la **vision** d’un DSL basé sur la co-information et la **réalité** des applications de grande dimension, pour lesquelles un calcul exact d’entropie resterait prohibitif.

1.7.1.3. Mise à Jour Itérative ou Partielle

Même si l’on parvient à maintenir une **structure parcimonieuse** (où un sous-ensemble restreint de liaisons $\omega_{i,j}$ demeure actif), la **mise à jour** de chaque pondération peut néanmoins devenir un **processus lourd** dans un réseau de grande taille. Pour rendre le **Deep Synergy Learning (DSL)** praticable à grande échelle, il est donc nécessaire de recourir à des **méthodes itératives** ou à des **mises à jour partielles**, évitant un recalcul exhaustif à chaque itération.

Une solution envisageable réside dans une **mise à jour locale en parallèle**. Chaque entité \mathcal{E}_i ne s’occupe que de ses **voisins** actuels, c’est-à-dire les nœuds auxquels elle est reliée par des liaisons $\omega_{i,j}$ non négligeables. Chacune met à jour les pondérations de ses connexions en parallèle avec les autres entités. Cette approche repose sur une **communication distribuée** ou une **mutualisation des ressources** (GPU, TPU), tout en restant locale. Seuls les liens existants sont explorés, évitant ainsi un balayage exhaustif de toutes les paires (i, j) .

Il demeure toutefois essentiel d’introduire un **mécanisme** pour éviter les conflits et saturations éventuelles. Par exemple, si plusieurs entités cherchent à coopérer en excès avec la même cible, on peut concevoir une **inhibition compétitive** assurant que les liaisons se stabilisent autour d’un certain nombre de voisins. Certaines idées issues des **systèmes multi-agents** (communication locale asynchrone, règles de stabilisation) favorisent la **convergence** du réseau. Ce paradigme

s'apparente finalement à une **simulation** de réseaux biologiques ou neuronaux, où les neurones mettent à jour leurs connexions en temps réel de manière distribuée. La **scalabilité** demeure possible tant que la **densité** du graphe demeure limitée et que les calculs de synergie (ou de similarité) ne concernent pas l'ensemble des paires dans un espace de grande dimension.

1.7.1.4. Vers une Mathématique de la Parcimonie et du Grouping Évolutif

Une autre voie de recherche consiste à définir une **énergie** ou un **coût** global $J(\Omega)$ pour le réseau Ω , englobant non seulement le terme $-\sum \omega_{i,j} S_{i,j}$ (valorisant les liens qui affichent une forte synergie) mais également un terme de **pénalisation** du nombre de liaisons non nulles. On peut par exemple écrire :

$$J(\Omega) = - \sum_{i,j} \omega_{i,j} S_{i,j} + \alpha \|\Omega\|_0,$$

où $\|\Omega\|_0$ compte le nombre de connexions actives (i.e. $\omega_{i,j} > 0$) et $\alpha > 0$ contrôle l'importance de la parcimonie. Cette formulation **force** le réseau à **rester** relativement épars, tout en **maximisant** la somme de synergies. L'analyse mathématique des **minima locaux** de $J(\Omega)$ et la description des **trajectoires** de descente dans un espace de grande dimension soulèvent des **défis majeurs**. Il s'agit de caractériser la **stabilité** des clusters émergents ainsi que la **croissance** potentielle de la taille du réseau au fil des itérations. Cette approche ouvre des perspectives pour un contrôle plus strict de la **complexité**, en assurant qu'une connectivité limitée ne se transforme pas en un graphe complet, préservant ainsi la faisabilité du **DSL** à grande échelle.

Conclusion

La **complexité computationnelle** et la **scalabilité** constituent sans doute les **premiers** grands défis pour le **Deep Synergy Learning**. Sans mesures de parcimonie et sans heuristiques de mise à jour partielle, le coût en temps et en mémoire peut devenir prohibitif. Les **pistes** de solutions incluent :

- **Sparsité** intégrée dans la création/rupture de liens,
- **Méthodes** d'estimation approximative des synergies dans des espaces de haute dimension,
- **Parallélisme** et algorithmes distribués,
- **Formulations** d'énergie globale favorisant la parcimonie.

Ces recherches sont cruciales pour permettre au DSL de s'étendre à grande échelle (milliers, millions d'entités) sans perdre l'efficacité ou la réactivité nécessaires dans les applications industrielles, médicales, ou cognitives.

1.7.2. Qualité et Disponibilité des Données

Au-delà des défis de **complexité** et de **scalabilité** (1.7.1), le **Deep Synergy Learning (DSL)** doit aussi relever des enjeux majeurs quant à la **qualité** et la **disponibilité** des données. En effet, la logique synergique et auto-organisée repose sur l'évaluation locale de **mesures de synergie** ($S(\mathcal{E}_i, \mathcal{E}_j)$) et sur la **mise à jour** (positive ou négative) de pondérations $\omega_{i,j}$. Lorsque les données sont trop **bruitées**, trop **parcellaires** ou manquent de **représentativité**, la convergence vers des clusters pertinents ou la stabilité de l'architecture peut être compromise. Le DSL, tout en étant plus tolérant aux lacunes qu'un réseau neuronal figé, n'en demeure pas moins dépendant d'un certain **niveau** de fiabilité et de diversité des entrées pour exploiter la synergie.

Dans cette sous-section (1.7.2), nous analyserons :

- Comment la **qualité** et la **couverture** des données impactent le calcul de la synergie,
- Pourquoi la **disponibilité** (flux continu vs. données rares) peut influer sur la dynamique du DSL,
- Les biais possibles et les problèmes d'**incohérence** qui peuvent fausser l'auto-organisation,
- Les pistes pour **sécuriser** ou **améliorer** la robustesse de l'apprentissage synergique face à des données imparfaites.

1.7.2.1. Importance d'une Bonne Couverture et Diversité des Données

Le **Deep Synergy Learning (DSL)** s'appuie sur la fonction de synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ pour guider la mise à jour des pondérations $\omega_{i,j}(t)$. Lorsque ces données \mathcal{E}_i et \mathcal{E}_j sont rares, bruitées ou biaisées, l'estimation de leur **distance** ou **similarité** (ou même de leur **co-information**) risque d'être inexacte. Cela peut entraîner un renforcement abusif de liens qui ne recouvrent aucune complémentarité réelle ou, à l'inverse, un affaiblissement injustifié de liaisons potentiellement utiles. Le réseau peut dès lors former des **clusters** peu pertinents, traduire des **biais** préexistants dans les données ou négliger des sous-groupes minoritaires.

Au-delà du simple problème d'**apprentissage** unitaire, le **DSL** met en relief la **couverture** temporelle. Si les conditions (saisons, configurations) varient, un système cherchant à s'adapter en continu aura besoin d'exemples suffisants pour chaque **régime** de données (périodes spécifiques, sous-populations) afin de réévaluer les synergies. Si l'on ne présente jamais de situations "hiver" lors de la constitution initiale du réseau, il se peut que les liaisons nécessaires ne s'instaurent pas, et le DSL échouera à former des **clusters** adaptés dès l'apparition d'un climat hivernal. Cette logique s'applique également à d'autres domaines. Un **flux** d'information continu exige un nombre d'**itérations** suffisant pour que chaque nouvelle modalité ou contexte s'intègre progressivement dans le **Synergistic Connection Network**, consolidant les liens synergiques pertinents.

La diversité et la **représentativité** des données apparaissent donc cruciales pour qu'un **DSL** puisse véritablement tirer parti de son **auto-organisation**. Sans cette diversité, il risque de manifester des **biais** similaires à ceux de nombre de méthodes d'apprentissage, la différence étant qu'ici c'est la

formation des clusters (et la dynamique des liaisons ω) qui pâtit des lacunes dans la couverture des phénomènes et des populations.

1.7.2.2. Bruit, Lacunes et Données Incomplètes

Comme discuté en section 1.5.3, le **Deep Synergy Learning (DSL)** fait preuve d'une **flexibilité** remarquable vis-à-vis des données incomplètes ou bruitées. Une entité trop incertaine finit par se **retrouver isolée** si sa synergie avec les autres entités demeure inexiste, préservant ainsi la **structure** principale du réseau. Dans ce cas, le réseau peine à structurer une organisation cohérente. Lorsque la majorité des entités est affectée par un **bruit excessif** ou un **manque de données**, l'**auto-organisation** ne parvient pas à dégager de **synergies réelles**. Les pondérations $\omega_{i,j}$ restent faibles, empêchant la formation de **clusters solides**, ou bien elles se renforcent à tort en raison d'**artefacts de bruit**, entraînant des structures biaisées.

Le **DSL** reste donc partiellement tributaire d'un **nettoyage** ou d'une **consolidation** préliminaire si le **bruit** se montre trop massif, sous peine de limiter sa capacité à détecter les vraies complémentarités. Plusieurs stratégies peuvent être mises en place pour améliorer la **qualité des données** et renforcer la cohérence du réseau. Un **filtrage statistique** permet d'écartier ou de sous-pondérer les entités dont les mesures sont jugées trop incertaines, en raison d'un taux élevé de valeurs manquantes ou d'incohérences répétées. Une **fusion locale** des entités fortement corrélées améliore la **fiabilité** du réseau : lorsque deux entités \mathcal{E}_i et \mathcal{E}_j possèdent des **données redondantes**, leur **agrégation** en une entité plus **robuste** réduit la dispersion des pondérations $\omega_{i,j}$. L'inclusion de **scores de fiabilité** dans la fonction de synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ limite l'influence des estimations reposant sur un faible échantillon de points ou sur des valeurs suspectées de bruit. La similarité ou la distance peut ainsi être modulée en fonction d'un **coefficent de confiance** propre à chaque entité ou à chaque mesure.

Grâce à ces mécanismes, on endigue l'effet d'un bruit omniprésent tout en préservant la logique d'**auto-organisation** du **DSL**. Ainsi, même si certaines entités restent partiellement incomplètes, elles peuvent trouver leur place dans le **Synergistic Connection Network** dès lors qu'elles entretiennent au moins quelques synergies robustes avec d'autres entités.

1.7.2.3. Accès et Disponibilité Continue : le DSL comme Système en Ligne

Dans de nombreux environnements, il est nécessaire de gérer un **flux** ininterrompu de données et de réajuster en permanence les structures d'apprentissage. Le **Deep Synergy Learning (DSL)**, déjà décrit comme un processus de mise à jour itérative (voir la section 1.4.5), peut fonctionner en **mode "online"** si l'on s'assure que chaque entité \mathcal{E}_i reçoit ses observations de manière régulière ou semi-régulière. Lorsque de nouvelles mesures sont disponibles, la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ est recalculée localement (ou approximée sur une fenêtre glissante), puis la pondération $\omega_{i,j}(t+1)$ s'actualise selon l'équation :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Cette logique d'évolution incrémentale requiert un **accès** continu aux flux entrants, que l'on traite idéalement par un système de gestion de messages (type broker Kafka) ou par un stockage minimal assurant la conservation des $\omega_{i,j}(t)$. Le réseau se reconfigure au fil de la réception de nouvelles données, sans qu'il soit nécessaire de reprendre un apprentissage exhaustif en mode batch. Dans un contexte industriel où la disponibilité des données peut être irrégulière, il arrive parfois que certaines entités n'aient aucune mise à jour pendant une période. Il faut alors conserver leurs liaisons $\omega_{i,j}(t)$ inchangées et ne pas pénaliser excessivement l'absence de mesures, conformément aux principes d'adaptation face aux lacunes de données (section 1.5.3). Ce mécanisme d'**auto-régulation** assure une **adaptabilité continue** du **DSL** face aux changements de distribution. Lorsqu'une entité devient moins pertinente dans son **environnement**, ses **liaisons** ω diminuent progressivement, l'éloignant de son cluster d'origine. Simultanément, de **nouvelles connexions** émergent avec des entités mieux alignées sur la situation actuelle, permettant au réseau de s'**ajuster dynamiquement** sans nécessiter de réapprentissage global.

Du point de vue du **déploiement**, un tel système en ligne exige une **synchronisation** ou un **parallélisme** adapté. Chaque entité ne traite que ses connexions actives, ce qui maintient la complexité à un niveau proportionnel au nombre de liaisons conservées (voir la section 1.7.1.2 sur la parcimonie). Pour des applications massivement distribuées, il est possible de faire coexister plusieurs nœuds DSL partiels et de synchroniser leurs matrices ω par échanges de sous-graphes ou de résumés statistiques. Cette configuration favorise la **scalabilité** et la **résilience**, car si un nœud local subit une interruption ou un ralentissement, le reste du réseau n'est pas forcément de s'arrêter, et les autres entités continuent à ajuster leurs pondérations. Une telle conception rend le **DSL** apte à gérer des environnements en perpétuel changement, tout en intégrant les principes d'auto-organisation et de mise à jour continue.

1.7.2.4. Biais et Incohérences Structurelles

Dans le **Deep Synergy Learning (DSL)**, les liaisons $\omega_{i,j}$ émergent localement à travers la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. Lorsque les données d'entrée sont biaisées, par exemple avec une surreprésentation de certains profils (sections 1.5.3 et 1.7.2.1), le réseau a tendance à **renforcer** les clusters correspondant aux **groupes majoritaires**, tout en marginalisant des entités issues de groupes minoritaires ou insuffisamment renseignés. Dans des domaines critiques comme le **diagnostic médical** ou la **recommandation sensible**, un tel phénomène peut avoir pour effet d'**entériner** des inégalités ou de mener à des **erreurs d'interprétation** (sous-diagnostic pour certains patients, ou éviction de contenus pertinents pour certaines catégories d'utilisateurs).

À la différence d'un **modèle supervisé** où l'on surveille une **loss** globale sur l'ensemble des données, le DSL base ses ajustements sur des **mesures locales** de synergie. Même si la structure globale Ω peut finir par présenter un arrangement biaisé, la démarche d'**auto-organisation** ne possède pas toujours de mécanismes internes de correction. Les entités minoritaires voient leurs pondérations $\omega_{i,j}$ rester faibles, du fait qu'elles ne rencontrent pas suffisamment de synergie avec la majorité. Par ailleurs, si une **incohérence** surgit ou si un artefact statistique biaisé devient prépondérant, le DSL peut stabiliser un **cluster aberrant**, en l'absence de contrôle externe.

Pour pallier cette difficulté, il est indispensable de prévoir des **mécanismes de validation** ou de **monitoring** global. On peut, par exemple, injecter des **entités symboliques** (sections 1.5.7 et

1.7.2.3) qui portent des règles ou des normes éthiques, limitant la croissance de clusters injustes ou repérant les configurations contradictoires. On peut également imposer un **feedback** externe, ou un signal de performance englobant des critères d'équité, veillant ainsi à empêcher la fermeture de groupes déviants. La structure ω peut alors être réévaluée à la lumière de ces **contraintes** d'équité ou de cohérence, conférant au réseau un degré de régulation qui contrebalance les dérives potentielles issues de la simple coopération locale.

1.7.2.5. Conclusion

La **qualité** et la **disponibilité** des données jouent un rôle clé dans le **Deep Synergy Learning**, comme dans toute approche d'IA. Ce modèle parvient à mieux tolérer la **partialité** en isolant les entités trop bruitées et s'adapte efficacement à un **flux continu** grâce à la mise à jour locale des pondérations. Toutefois, il reste sensible aux **biais** et nécessite une **bonne représentativité** des données pour assurer une synergie pertinente.

Un **travail préparatoire** rigoureux s'impose afin d'optimiser le fonctionnement du **Deep Synergy Learning**. Il est essentiel de **filtrer** et **pondérer** la qualité des données, d'intégrer des **scores de confiance** dans l'évaluation de la synergie, et de définir une **stratégie** de streaming ou de batch mixte pour alimenter la dynamique des pondérations ω . Il convient également de surveiller l'**évolution** des clusters afin de détecter d'éventuelles anomalies ou d'anticiper la propagation de biais au sein de l'auto-organisation.

Un **pilotage rigoureux** des données est essentiel pour exploiter pleinement le **DSL**. Un système **auto-organisé** ne peut compenser des biais majeurs de collecte ou un manque de **représentativité** sans une **ingénierie des données** adéquate. Une **gouvernance robuste** garantit une qualité suffisante des flux entrants, évitant ainsi que des **liaisons synergiques** se forment sur des bases erronées ou que des **clusters** se structurent autour de signaux bruités.

1.7.3. Développement d'Algorithmes d'Optimisation Appropriés

Une fois admise la nécessité de gérer la **complexité** (1.7.1) et de soigner la **qualité** des données (1.7.2), se pose la question cruciale des **algorithmes** qui permettront de **piloter** la dynamique du **Deep Synergy Learning (DSL)**. En effet, ce paradigme repose sur l'**auto-organisation** de multiples entités $\{\mathcal{E}_i\}$ et sur la **mise à jour** de l'ensemble des pondérations synergiques $\omega_{i,j}(t)$. Dans le modèle théorique simple, on écrit souvent :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

où $S(\mathcal{E}_i, \mathcal{E}_j)$ désigne la **synergie** entre ces entités.

En pratique, mettre en œuvre cette loi pour un grand nombre d'entités, avec des synergies potentiellement complexes (non linéaires, conditionnelles, n-aires), n'est pas trivial. On a besoin de **méthodes d'optimisation** capables de **stabiliser** le système en évitant les oscillations ou l'explosion des pondérations. Elles doivent aussi **accélérer** la convergence vers des configurations pertinentes comme les clusters et macro-clusters. L'intégration de **contraintes** est essentielle pour garantir la cohérence, la parcimonie et la compatibilité avec des règles symboliques. Enfin, ces

méthodes doivent pouvoir **s'adapter** en temps réel à l'apparition de nouvelles entités ou de nouveaux flux de données.

Cette sous-section (1.7.3) discute quelques **pistes** mathématiques et algorithmiques pour rendre le DSL plus solide et plus efficace :

- Des méthodes **locales** (mise à jour distribuée),
- Des formulations **globales** (fonction d'énergie, descente de gradient généralisée, approches de type recuit simulé),
- Des **heuristiques** ou des règles inspirées de la biologie et des systèmes complexes (sélection, reproduction, extinction),
- Des **extensions** pour la synergie n-aire ou conditionnelle.

1.7.3.1. Approche Locale et Distribuée

Dans l'esprit **biologique** et de nombreux modèles de **systèmes complexes**, on peut envisager la mise à jour de chaque pondération $\omega_{i,j}$ de manière **locale**, sans qu'un contrôle global impose la synchronisation. Chaque entité \mathcal{E}_i communique uniquement avec ses **voisins** au sens du graphe, autrement dit les entités \mathcal{E}_j pour lesquelles $\omega_{i,j}(t)$ est significatif. L'entité \mathcal{E}_i calcule alors la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ sur la base de mesures ponctuelles (ou d'un historique local) et applique la règle :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta_{i,j}(t) [S_{i,j}(t) - \tau_{i,j}(t) \omega_{i,j}(t)].$$

Cette équation reprend le formalisme général du **Deep Synergy Learning** (section 1.4.5) en autorisant des coefficients $\eta_{i,j}(t)$ et $\tau_{i,j}(t)$ qui dépendent de la paire (i, j) ou varient dans le temps, procurant une **régulation** fine. L'avantage est de rendre l'algorithme **parallélisable**, puisque chaque entité \mathcal{E}_i peut mettre à jour ses liaisons sans solliciter un "master" central. Une difficulté réside dans la synchronisation des mises à jour. Si plusieurs entités modifient en même temps leurs liaisons synergiques, des boucles instables ou des conflits peuvent émerger. Une gestion asynchrone ou régulée des ajustements est alors nécessaire pour éviter des oscillations non maîtrisées dans le réseau. Des stratégies d'**asynchronisme** maîtrisé ou de communication par itérations (steps) successives sont alors recommandées.

Dans cette optique, on peut se référer à diverses **métaphores** ou **règles** d'inspiration biologique. Par exemple, une **mise à jour** "hebbienne" renforce la pondération $\omega_{i,j}$ lorsqu'on observe une co-activation récurrente des entités \mathcal{E}_i et \mathcal{E}_j , et l'affaiblit en l'absence de co-occurrence. Par ailleurs, on peut sélectionner ou reproduire certaines liaisons. Si des paires (i, j) restent faiblement utiles, elles sont éliminées, tandis que d'autres liens se renforcent ou se dupliquent sur d'autres noeuds. Une compétition locale peut limiter la densité des connexions en contrignant la somme $\sum_j \omega_{i,j}$ pour chaque entité \mathcal{E}_i ou en appliquant un softmax local, favorisant quelques connexions fortes plutôt qu'un réseau dilué.

Par ces variantes, on parvient à **contrôler** la **densité** du graphe, la vitesse de renforcement des pondérations et la **résilience** face au bruit ou aux données incomplètes (sections 1.7.1.2 et 1.5.3). Le **DSL** conserve sa **scalabilité** tout en respectant l'**auto-organisation**. Chaque mécanisme reste local, assurant une dynamique distribuée compatible avec des implémentations parallèles sur GPU ou TPU, ainsi qu'avec des approches multi-agents où chaque entité ajuste ses connexions en fonction de ses propres observations et interactions.

1.7.3.2. Formulation Globale via Fonction d'Énergie

Dans une perspective plus proche de l'**optimisation** (au sens des approches traditionnelles en IA ou en physique statistique), il est concevable de décrire l'**auto-organisation** du Deep Synergy Learning (DSL) par une **fonction d'énergie** $\mathcal{J}(\Omega)$. On peut écrire une expression qui généralise la somme des synergies et la pondère par un terme de **régularisation**, selon :

$$\mathcal{J}(\Omega) = - \sum_{i,j} \omega_{i,j} S_{i,j} + R(\Omega).$$

Dans ce formalisme, la somme $\sum_{i,j} \omega_{i,j} S_{i,j}$ valorise la **coopération** (au sens de la synergie, voir sections 1.4.4 et 1.4.5), tandis que le terme $R(\Omega)$ constitue une **régularisation**, par exemple $\alpha \|\Omega\|^2$ ou $\alpha \|\Omega\|_0$, qui maintient la parcimonie ou la cohérence structurelle du **Synergistic Connection Network**. Cette formulation intègre les principes abordés en 1.7.1.4, notamment le **contrôle de la densité** et le **maintien des contraintes**, au sein d'une unique fonction de coût. L'objectif global revient à **maximiser** la somme des synergies ou, en termes d'énergie, à **minimiser** \mathcal{J} .

Pour résoudre l'évolution de Ω dans ce cadre, on peut recourir à une **descente de gradient**. Si l'on note $\omega_{i,j}$ un des paramètres, sa dérivée partielle vis-à-vis de \mathcal{J} prend la forme

$$\frac{\partial \mathcal{J}}{\partial \omega_{i,j}} = - S_{i,j} + \frac{\partial R(\Omega)}{\partial \omega_{i,j}}.$$

En discréétisant cette descente, on retrouve l'équation qui met à jour $\omega_{i,j}$ de façon incrémentale :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{i,j} - \tau \omega_{i,j}(t)],$$

ce qui explicite la proximité entre la **logique locale** décrite en 1.4.5 et la **formulation** d'une énergie globale $\mathcal{J}(\Omega)$. Dans les problèmes complexes, la descente de gradient peut aboutir à des minima locaux sous-optimaux. On peut alors envisager un **recuit simulé** (simulated annealing) directement sur l'espace du graphe Ω . Cette méthode consiste à perturber par moments la configuration courante (création ou suppression de liens), et à n'accepter ou refuser ces modifications qu'avec une probabilité liée à la variation $\Delta \mathcal{J}$ et à un paramètre de température qui décroît au fil des itérations. Dans le contexte du **DSL**, où la synergie peut être **n-aire** (voir section 1.4.7), une exploration plus large de l'espace combinatoire devient essentielle. Se limiter à des ajustements locaux risquerait d'aboutir à un **cluster** partiel, sans permettre l'émergence de synergies plus globales. Le recuit simulé ou des **algorithmes évolutionnaires** jouent alors le rôle de recherche d'une configuration plus robuste, accompagnant l'auto-organisation locale. Cette combinaison de l'approche **par**

énergie et de la **dynamique en continu** ouvre une voie pour maîtriser mathématiquement la convergence et la structure du **Synergistic Connection Network**, en particulier lorsqu'on cherche à maintenir une **parsimonie** explicite ou à inclure des contraintes topologiques (voir 1.7.1.4).

1.7.3.3. Synergie n-aire et Extensions Conditionnelles

Le **Deep Synergy Learning (DSL)** se focalise souvent sur la **coopération** entre deux entités \mathcal{E}_i et \mathcal{E}_j . Toutefois, certains problèmes exigent une **synergie n-aire**, c'est-à-dire impliquant un **groupe** $\{\mathcal{E}_{i_1}, \dots, \mathcal{E}_{i_n}\}$ de cardinal $n \geq 3$. Dans ce cas, on peut introduire des **hyper-arêtes** ω_{i_1, \dots, i_n} plutôt que de simples liaisons binaires. La mise à jour de la pondération hyper-arête s'effectue alors selon une généralisation de la loi (voir section 1.4.5), par exemple :

$$\omega_{i_1, \dots, i_n}(t+1) = \omega_{i_1, \dots, i_n}(t) + \eta [S_n(\mathcal{E}_{i_1}, \dots, \mathcal{E}_{i_n}) - \tau \omega_{i_1, \dots, i_n}(t)].$$

Dans cette équation, la fonction $S_n(\mathcal{E}_{i_1}, \dots, \mathcal{E}_{i_n})$ évalue la **synergie** globale de l'ensemble $\{\mathcal{E}_{i_k}\}$. Sa définition peut s'appuyer sur une co-information n-aire, sur un gain de performance collectif ou sur toute métrique reflétant l'apport mutuel de ce sous-groupe. Néanmoins, la **complexité** combinatoire s'en trouve rapidement démultipliée, car on doit gérer des hyper-arêtes potentiellement innombrables lorsque n augmente. Pour cette raison, il est fréquent de recourir à des **heuristiques** afin de détecter des triplets, quadruplets, etc., seulement lorsque l'on constate une coopération stable entre ces entités ; on se contente de créer l'hyper-arête ω_{i_1, \dots, i_n} après avoir observé à maintes reprises que ces n entités coopèrent ensemble.

Par ailleurs, il arrive que la **synergie** dépende d'un **contexte c**. Un exemple typique survient lorsque l'association audio–texte n'est pertinente que dans une situation donnée (un lieu, une heure, une ambiance sonore). La règle de mise à jour se récrit alors :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j \mid \mathbf{c}(t)) - \tau \omega_{i,j}(t)].$$

Dans ce cadre, il est nécessaire de calculer un $S(\cdot \mid \mathbf{c})$ qui peut varier considérablement en fonction du contexte. La tâche d'**estimation** et d'**optimisation** devient plus complexe, car la distribution induite par $\mathbf{c}(t)$ peut évoluer au fil du temps ou présenter plusieurs régimes de fonctionnement distincts. Les algorithmes d'auto-organisation du DSL doivent alors incorporer des modèles de synergie **conditionnelle**, qui modulent la pondération $\omega_{i,j}$ suivant la valeur prise par \mathbf{c} . Cette extension demeure essentielle dans de multiples applications, comme la robotique adaptative (un même couple de capteurs se révèle pertinent ou non selon le mode de locomotion) ou le traitement multimodal (un segment audio et un mot ne coïncident que dans un contexte temporel précis). Elle requiert cependant des stratégies de **parcimonie** ou de **sampling** pour éviter une explosion combinatoire lorsqu'on veut évaluer la synergie conditionnelle sur tous les contextes possibles.

1.7.3.4. Contrôle de la Stabilité et Convergence Globale

Dans tout **système dynamique** où les éléments sont fortement couplés, des **cycles** ou des **oscillations** peuvent apparaître. Un sous-réseau peut se renforcer temporairement, avant qu'un autre ne prenne le relais en affaiblissant les liens précédents, entraînant une alternance continue. Il

peut aussi survenir un phénomène de **basculement** entre plusieurs attracteurs stables (configurations rivales) si la structure du **Synergistic Connection Network** autorise plusieurs manières de s'organiser pour la même situation. Le **Deep Synergy Learning (DSL)** n'échappe pas à ces scénarios, en particulier lorsque la fonction de synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ est très “non linéaire” ou quand la synergie n-aire entre plusieurs entités génère des interactions complexes (sections 1.4.7 et 1.7.3.3).

Pour limiter ce risque d’oscillations ou de dérive, il convient d’inclure des **mécanismes** ou des **règles** de contrôle dans la mise à jour des pondérations $\omega_{i,j}$. On peut imposer un **amortissement** progressif (faire varier η ou τ dans le temps), intégrer une **pénalisation** plus agressive des liens excessivement grands, ou encore définir des **seuils de saturation** empêchant une pondération $\omega_{i,j}$ de croître sans borne. Des analyses mathématiques, telles que l’étude de la **Jacobienne** locale ou des **fonctions de Lyapunov** globales, peuvent démontrer qu’un schéma de mise à jour particulier garantit la convergence vers un état stable (ou au moins vers un cycle stable), sans risquer un chaos indésirable.

Il est également envisageable de greffer un **module** de contrôle hiérarchique (de plus haut niveau), qui surveille la dynamique globale du réseau et injecte ponctuellement des modifications de paramètres η ou τ si la variance du système dépasse un certain seuil. Ce principe d'**hybridation** (auto-organisation libre + régulation macro) rappelle certaines structures biologiques ou cognitives, où les ajustements locaux cohabitent avec des boucles de rétroaction plus centrales. On cherche alors à préserver la **liberté synergique** au niveau local, tout en s’assurant que le **réseau** ne sombre pas dans une instabilité excessive. Cet équilibre soulève des questions d'**optimisation** et de **pilotage** du **DSL**, qui restent ouvertes à la recherche, notamment sur les plans pratique (implémentation à grande échelle) et théorique (analyse de la dynamique dans un espace de forte dimension).

Conclusion

Le **développement d’algorithmes d’optimisation** pour le **Deep Synergy Learning** s’annonce particulièrement riche et complexe. Les **règles de mise à jour** naïves suffisent à illustrer le concept, mais ne passent pas toujours à l’échelle ni ne gèrent la synergie n-aire, les contraintes symboliques, ou la minimisation globale d’une fonction d’énergie. Les approches possibles incluent :

- **Mise à jour locale** inspirée de la biologie (sélection, Hebb, normalisation compétitive),
- **Descente d’énergie** globale (fonctions \mathcal{J} pénalisant la densité, favorisant la synergie),
- **Algorithmes évolutifs** (recuit simulé, heuristiques stochastiques) pour éviter les minima locaux,
- **Extensions** aux hyper-arêtes (synergies n-aires), aux synergies conditionnelles, ou aux règles symboliques,
- **Contrôles** de stabilité et d’amortissement, voire un superviseur partiel pour limiter les oscillations et injecter des priorités.

Trouver la meilleure **formulation** et le meilleur **algorithme** dépendra de l'application (vision, robotique, recommandation, etc.), du **volume** de données et de la **structure** (binaire ou n-aire, symbolique, multi-modal). La poursuite de ces recherches est incontournable pour que le **DSL** gagne en robustesse, en efficacité, et en adoptabilité dans le monde réel.

1.7.4. Contrôle et Stabilité des Processus Auto-Organisés

L'**auto-organisation** est au cœur du **Deep Synergy Learning (DSL)**, permettant à un grand nombre d'entités (capteurs, modules de calcul, représentations symboliques ou sub-symboliques) de s'agencer spontanément en **clusters** ou **macro-clusters** selon leurs synergies. Cette dynamique, bien qu'elle confère au **DSL** sa **plasticité** et sa **capacité d'adaptation**, peut également poser des **problèmes de stabilité**. Des oscillations, des attracteurs multiples ou des comportements chaotiques peuvent émerger, compliquant la convergence et l'organisation cohérente du réseau. De plus, dans bien des cas (robotique, diagnostic, logistique), il faut **contrôler** partiellement la configuration auto-organisée, ne serait-ce que pour garantir la **sécurité**, la **fiabilité**, ou la **cohérence** avec des règles externes.

Cette sous-section examine les **défis** liés à la **dynamique** d'un système **DSL** ainsi que les **mécanismes** permettant d'en assurer la **stabilité** et le **contrôle**. Un système couplé peut engendrer des **oscillations** ou des **cycles**, tandis que les pondérations synergiques ω peuvent soit **explorer**, soit **s'effondrer** sous certaines conditions. L'existence de **multiples attracteurs** en concurrence peut également mener à des configurations divergentes. Pour remédier à ces instabilités, il est possible d'introduire des **dispositifs** de rétroaction ou une **surveillance globale** afin d'assurer un pilotage **hiérarchique** léger. Enfin, il est crucial d'examiner les **conditions mathématiques** garantissant (ou non) la convergence de la dynamique du réseau.

1.7.4.1. Risques d'Oscillations et de Comportements Chaotiques

Dans le **Deep Synergy Learning**, chaque liaison $\omega_{i,j}$ obéit à une évolution itérative inspirée des principes décrits en section 1.4.5, ce qui donne lieu à un **système dynamique** d'environ $O(n^2)$ degrés de liberté dans le cas binaire. Pour comprendre comment apparaissent des **cycles** ou des **régimes oscillatoires**, il est éclairant d'étudier la **linéarisation** locale autour d'un point fixe ω^* . Supposons l'existence d'un équilibre ω^* satisfaisant un état stationnaire ; si l'on introduit une petite perturbation $\delta\omega$, la dynamique globale s'écrit de manière approchée :

$$\delta\omega(t+1) \approx J(\omega^*) \delta\omega(t),$$

où $J(\omega^*)$ désigne la **matrice jacobienne** évaluée en ω^* . En temps discret, un comportement oscillatoire ou pseudo-chaotique survient lorsque certaines **valeurs propres** de J possèdent une magnitude supérieure à 1. Cela signifie que la perturbation $\delta\omega$ se voit **amplifiée** au fil des itérations, au lieu d'être dissipée. En temps continu, la condition équivalente consisterait à observer des **parties réelles** positives dans les valeurs propres, menant à une instabilité locale ou à des boucles auto-entretenues.

Ce type de phénomène peut se traduire par des **cycles** permanents, ou même des trajectoires quasi chaotiques, contrariant la **convergence** du réseau vers un état stable. De tels régimes, bien que mathématiquement fascinants, s'avèrent peu souhaitables si l'on souhaite une structure DSL stable pour la robotique, la maintenance ou la prise de décision. C'est pourquoi on introduit souvent des **mécanismes** de stabilisation. Une première solution consiste à contrôler la “vitesse” de mise à jour en imposant une contrainte, par exemple $\eta \tau < 1$, dans le cas linéaire simplifié. On peut ensuite recourir à des **termes** non linéaires de saturation, destinés à éviter la croissance illimitée de certaines pondérations. Un exemple simple, dans un modèle continu, consiste à enrichir l'équation d'évolution par un terme $-\beta (\omega_{i,j})^3$:

$$\frac{d}{dt} \omega_{i,j} = \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}] - \beta (\omega_{i,j})^3,$$

où $\beta > 0$ modère la croissance des liaisons fortes. De manière analogue, on peut introduire un **facteur** de compétition locale (type softmax), en imposant $\sum_j \omega_{i,j} \leq K$ pour chaque entité \mathcal{E}_i , de sorte qu'une entité ne puisse pas développer des liens solides avec trop de partenaires en même temps.

Ces mesures limitent la **liberté** d'auto-organisation, mais s'avèrent cruciales pour écarter des régimes oscillatoires ou chaotiques susceptibles d'émerger dans un réseau fortement couplé et non linéaire. Elles assurent ainsi une forme de **stabilité** globale tout en préservant la possibilité de réorganisations locales dictées par la synergie. Un savant réglage des paramètres (η , τ , β , ou le budget $\sum_j \omega_{i,j} \leq K$) permet donc de maintenir le **DSL** dans une zone où le réseau demeure robuste et apte à converger, évitant des fluctuations intempestives tout en conservant l'essence auto-adaptative voulue.

1.7.4.2. Multiples Attracteurs, Convergence Incertaine

Il est fréquent qu'un **Deep Synergy Learning (DSL)** cherche à maximiser, ou à minimiser l'opposé, d'une **somme de synergies** $\sum_{i,j} \omega_{i,j} S_{i,j}$ éventuellement modulée par un terme de pénalisation ou de parcimonie (voir section 1.7.1.4). Dans un espace de grande dimension, on peut rencontrer de **multiples attracteurs** correspondant à des configurations dont l'énergie (ou le coût) est quasi équivalente. Ce phénomène se manifeste lorsqu'il existe plusieurs manières d'**arranger** les entités en clusters, chacune procurant un gain global proche. Le réseau peut alors converger vers l'un ou l'autre de ces attracteurs selon les conditions initiales ou de légères fluctuations.

Dans la pratique, il arrive que des “sous-réseaux” rivaux émergent au cours de la dynamique, chacun rassemblant certaines entités en un **cluster** distinct. Une compétition peut alors s'installer entre ces sous-réseaux, induisant des **instabilités** transitoires et des oscillations (voir section 1.7.4.1). Si l'on désire un **état** unique et stable, par exemple pour un robot qui ne peut se scinder en deux stratégies opposées, on peut injecter un **signal global** orientant le réseau vers l'attracteur souhaité. Cette intervention se formalise en modifiant la fonction de synergie ou en ajoutant un **feedback** de récompense plus élevé pour une configuration A que pour une configuration B. On obtient ainsi un **pilotage** macro qui restreint la pure auto-organisation, mais assure qu'en présence de minima multiples, la configuration la plus souhaitable selon les critères d'usage devienne

prépondérante. Cette démarche montre le besoin de compromis entre la liberté synergique propre au DSL et la nécessité d'un **contrôle** hiérarchique dans certaines applications critiques.

1.7.4.3. Approches Hiérarchiques Légères pour Guider l'Auto-Organisation

Dans des environnements critiques, tels que le domaine **médical** ou les **transports**, il n'est pas toujours acceptable de laisser un réseau en **Deep Synergy Learning (DSL)** se reconfigurer librement sans un regard d'ensemble. Afin de maîtriser la sécurité et la fiabilité, on introduit souvent une **couche** ou un **module** de supervision qui contrôle globalement la structure Ω . Ce superviseur observe la **matrice** des liens actifs et vérifie des **contraintes** imposées par l'application. Par exemple, on peut exiger que la somme $\sum_j \omega_{i,j}$ attachée à une entité \mathcal{E}_i ne dépasse pas un seuil spécifique, ou que certaines entités déclarées incompatibles (règles de sécurité, normes médicales contradictoires) ne figurent pas dans le même cluster. Si ce superviseur détecte une violation, il peut forcer la **mise à zéro** de certaines liaisons $\omega_{i,j}$, neutralisant ainsi des configurations illégitimes.

On peut aussi concevoir un **schéma** où la **couche basse** demeure l'auto-organisation locale et distribuée, tandis qu'une **couche haute** (un planificateur ou un décideur global) injecte à intervalles réguliers des **influx** spécifiques dans le réseau. Ces influx peuvent prendre la forme d'une modification ponctuelle de $\omega_{i,j}$, d'un renforcement sélectif d'une synergie, ou de l'assignation de priorités à certaines entités. Cet apport d'information agit comme une **guidance** de haut niveau, combinée à la dynamique adaptative du DSL.

D'un point de vue mathématique, on peut l'interpréter comme l'ajout d'un **terme** exogène dans l'équation de mise à jour, par exemple $\Delta\omega_{i,j}^{(\text{macro})}$ reflétant la volonté du **planificateur**. Cela permet d'enrichir la structure Ω sans remettre en cause l'intégralité du principe d'**auto-organisation** locale. On obtient un **compromis** dans lequel la dynamique $\omega_{i,j}$ se déroule principalement selon la synergie perçue, tout en restant arrimée à des objectifs de **performance** ou de **sécurité** plus vastes. Dans ce type d'architecture dite « *hybride* », le contrôle hiérarchique léger se borne à préserver la cohérence, tandis que la plupart des adaptations opérationnelles sont gérées par la mise à jour distribuée des liaisons.

1.7.4.4. Outils Mathématiques et Théoriques pour la Stabilité

Il est souvent utile de modéliser l'**évolution** d'un **Deep Synergy Learning (DSL)** sous la forme d'un **système dynamique** $\omega(t+1) = \omega(t) + F(\omega(t))$. Dans cette notation, $\omega(t)$ regroupe l'ensemble des pondérations $\omega_{i,j}(t)$, et $F(\omega(t))$ décrit la contribution que la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ (ainsi que les termes de décroissance, de régularisation, etc.) apporte à la variation entre t et $t+1$.

Pour **étudier** la stabilité locale autour d'un **point fixe** ω^* (ou d'une configuration fixe du réseau), on réalise classiquement une **linéarisation** de F . On calcule la **matrice jacobienne** :

$$J(\omega^*) = \frac{\partial F}{\partial \omega}(\omega^*).$$

Si l'on note $\delta\omega(t) = \omega(t) - \omega^*$, alors pour de petites perturbations, on obtient un modèle linéarisé :

$$\delta\omega(t+1) \approx \delta\omega(t) + J(\omega^*) \delta\omega(t).$$

En régime de temps discret, la **stabilité** locale autour de ω^* se produit si toutes les **valeurs propres** de $I + J(\omega^*)$ ont une magnitude strictement inférieure à 1. Dans un cadre continu (où l'on écrit $d/dt \omega = F(\omega)$), il faut que toutes les parties réelles des valeurs propres de $J(\omega^*)$ soient **négatives**. Cette condition s'interprète physiquement. De petites déviations $\delta\omega$ s'amortissent progressivement au fil du temps, garantissant ainsi une stabilité locale du système. Elle dépend du **choix** de la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$, de la taille du pas η , des paramètres de décroissance τ , ou des non-linéarités supplémentaires (termes de saturation, etc.).

Parfois, on peut exhiber une **fonction de Lyapunov** $\Phi(\omega)$ qui décroît strictement à chaque mise à jour, prouvant la **convergence** vers un **unique attracteur**. Cela exige que la définition de la synergie ne comporte pas de **non-linéarités** trop complexes, ou que la structure ne s'étende pas à des hyper-arêtes de dimension élevée (voir la section 1.4.7 sur la synergie n-aire). Dans des **configurations** plus simples (synergie linéarisable, réseau symétrique, etc.), on peut démontrer rigoureusement l'existence et l'unicité d'un état stationnaire stable. En revanche, pour des **formulations** plus riches, où la synergie est hautement non linéaire, dépendante du contexte ou définie sur des hyper-arêtes n-aires, établir une **preuve** de stabilité globale devient plus complexe. La dynamique peut alors exhiber plusieurs attracteurs ou adopter des régimes oscillants.

Ces outils (linéarisation jacobienne, fonctions de Lyapunov) offrent néanmoins un **cadre théorique solide** pour analyser la mise à jour des pondérations $\omega_{i,j}$ dans un **DSL**, en particulier lorsque l'on cherche à garantir un comportement prédictible et éviter les **oscillations** chaotiques ou l'explosion des liaisons. Ils offrent également des **repères** pour la **conception** du système. En ajustant les paramètres η et τ ou en imposant des contraintes sur la somme et la norme des pondérations, on peut maintenir la trajectoire de ω dans une **région stable**. Cela garantit une **convergence réelle** du système, évitant ainsi toute dérive ou oscillation entre plusieurs configurations.

Conclusion

Assurer le **contrôle** et la **stabilité** des **processus auto-organisés** dans le **DSL** est un enjeu fondamental pour prévenir l'apparition de **comportements oscillants** ou chaotiques. La gestion de **multiples attracteurs** permet d'orienter l'état final et d'éviter des trajectoires divergentes. Dans certains cas, un **encadrement** minimal peut être nécessaire, sous la forme d'un module de supervision ou de règles hiérarchiques.

Les solutions mathématiques ou algorithmiques incluent la **conception** de règles de mise à jour intégrant amortissement, saturation ou normalisation pour limiter la croissance excessive des pondérations. L'introduction d'une **énergie globale** $J(\Omega)$, optimisée par descente ou recuit simulé, permet d'encadrer l'évolution du système.

Un **pilotage macro** peut être mis en place en ajustant η et τ , en injectant des priorités ou en filtrant certaines liaisons. Enfin, l'**analyse de stabilité** via Jacobienne locale ou Lyapunov peut garantir, au moins localement, la convergence et éviter l'explosion des pondérations.

Maîtriser l'**auto-organisation** ne signifie pas brider la flexibilité du DSL, mais offrir à l'architecture la **robustesse** nécessaire pour opérer en conditions réelles, où l'on ne peut se permettre un comportement erratique ni une multiplicité de configurations irréconciliables.

1.7.5. Interprétabilité et Explicabilité pour l'Humain

Un objectif crucial dans la plupart des applications d'**intelligence artificielle** est de fournir non seulement des **Résultats** (prédictions, recommandations, décisions), mais également une **compréhension** de la logique ou du **processus** qui y aboutit. Les méthodes classiques de **Deep Learning** (CNN, RNN, Transformers) sont souvent décriées pour leur caractère “**boîte noire**”, dans la mesure où l'information se diffuse à travers des poids distribués dans de multiples couches, rendant l'**explication** pour l'humain délicate.

Le **Deep Synergy Learning (DSL)**, par son **architecture en entités** et en **liaisons synergiques** ($\omega_{i,j}$), offre un cadre potentiellement plus **transparent**. Il permet d'identifier quelles **entités** interagissent fortement, comment se forment les **clusters** ou **macro-clusters**, et pourquoi ces regroupements émergent sur la base d'une **synergie mesurable**. Cependant, assurer une **interprétabilité complète** n'est pas garanti. Il est nécessaire de développer des **méthodes** permettant d'extraire des **explications compréhensibles**. Cette sous-section (1.7.5) examine les **défis** et **solutions** envisageables pour améliorer l'**explicabilité** du DSL.

1.7.5.1. D'un Modèle Hiérarchique Opaque à un Réseau de Liens Synergiques

Il est courant que les **réseaux neuronaux profonds** (CNN, RNN, Transformers) diluent leur représentation interne à travers plusieurs **couches** qui transforment graduellement les descripteurs, rendant ainsi les décisions finales difficiles à expliquer. Les neurones internes ne possèdent pas d'**identité propre**. Ils agissent comme de simples réceptacles de **poids** et d'**activations**, sans que l'on puisse leur attribuer un **sens direct** ou une **fonction explicite**. À l'inverse, le **Deep Synergy Learning (DSL)** introduit un **réseau** où chaque **nœud** \mathcal{E}_i possède une **identité** clairement définie (il peut s'agir d'un **capteur** particulier, d'un **concept** sémantique, d'un **patch** visuel, d'une **règle** symbolique, etc.), tandis que les **liaisons** $\omega_{i,j}$ se **reconfigurent** progressivement, au gré de la synergie détectée entre \mathcal{E}_i et \mathcal{E}_j .

Cette structure présente plusieurs **avantages**. Elle améliore la **lisibilité** des **sous-ensembles** en mettant en évidence des **clusters** (chapitre 1.4.3) formés par des entités aux liaisons fortes, révélant une **cohérence** sous-jacente. Un cluster peut, par exemple, regrouper des entités “visuel + audio” en reconnaissance multimodale ou “utilisateur + contenus + tag de genre” dans un système de recommandation. Chaque **entité** \mathcal{E}_i possède une **signification explicite** (patch d'image, concept symbolique), facilitant ainsi l'**interprétation** des décisions du réseau. Un **macro-cluster** peut traduire un regroupement plus large, tel qu'un ensemble “groupe d'utilisateurs amateurs de jazz + contenus jazz + soirées concert” sur une plateforme culturelle, ce qui permet d'expliquer une recommandation de manière transparente : “puisque vous faites partie du cluster jazz, ce concert vous est suggéré”.

Cependant, cette **lisibilité** accrue sur le plan conceptuel s'accompagne aussi d'une **complexité** potentiellement élevée. Le réseau DSL peut afficher un grand nombre de **nœuds** (surtout en contexte industriel ou multi-sensoriel), et chaque nœud peut avoir plusieurs **connexions** significatives, aboutissant à une **topologie** complexe, souvent composée de **clusters** imbriqués ou de multiples liens réciproques. Une simple **visualisation** naïve des poids $\omega_{i,j}$ peut alors ne rien révéler de clair, s'il n'existe pas de mécanisme de filtrage ou de simplification. Il devient nécessaire d'utiliser des **outils de hiérarchisation**, de **regroupement** ou de **filtrage** pour dériver des **chemins** explicatifs, par exemple en extrayant les principales **chaînes** de liaison qui influencent la décision finale. C'est là que réside l'enjeu d'une **explicabilité** réellement transparente dans un réseau DSL riche, qui exige de se doter de techniques aptes à condenser le graphe et à exposer les assemblées cruciales, tout en évinçant la masse de connexions moins pertinentes.

1.7.5.2. Extractions de Chemins Synergiques et Clusters Pivot

Dans un **réseau** construit selon les principes du **Deep Synergy Learning (DSL)**, il est possible d'expliquer une décision ou une recommandation en identifiant un **chemin** significatif reliant deux entités \mathcal{E}_a et \mathcal{E}_b . On peut ainsi considérer une suite :

$$\mathcal{E}_a \rightarrow \mathcal{E}_x \rightarrow \mathcal{E}_y \rightarrow \mathcal{E}_b,$$

au long de laquelle la **synergie** s'avère forte. On peut évaluer cette force par un produit du type $\omega_{a,x} \omega_{x,y} \omega_{y,b}$, exprimant la continuité du lien depuis \mathcal{E}_a jusqu'à \mathcal{E}_b via \mathcal{E}_x et \mathcal{E}_y . Par exemple, si \mathcal{E}_a représente un **utilisateur** et \mathcal{E}_b un **contenu** recommandé, un **chemin** pertinent peut inclure un **tag** musical \mathcal{E}_x et un **artiste** \mathcal{E}_y . La **synergie** met alors en évidence que l'utilisateur \mathcal{E}_a est historiquement associé à ce tag rock. Ce tag entretient une forte connexion avec l'artiste \mathcal{E}_y , qui est directement relié au **contenu** \mathcal{E}_b . L'explication devient un **raisonnement local**. L'utilisateur est associé au tag rock, ce tag est fortement lié à l'artiste \mathcal{E}_y , et cet artiste mène au contenu \mathcal{E}_b proposé.

Dans la pratique, on limite souvent la longueur du **chemin** afin de conserver une **explication** concise et compréhensible, tout en montrant comment la synergie chemine dans le **Synergistic Connection Network**.

Une autre forme d'explication repose sur le **cluster** (ou **macro-cluster**) auquel appartient l'entité \mathcal{E}_b . On met en avant les **entités dominantes** de ce cluster. En diagnostic médical, cela peut correspondre à un ensemble d'**indicateurs** comme la fièvre, une anomalie sanguine et une image radio suspecte, qui forment un sous-réseau fortement interconnecté. L'**interprétation** découle alors de la **cohérence interne** du **cluster**. Ces signes, fortement liés par la synergie, renforcent la conclusion de pathologie. Cette méthode est très usitée lorsque la configuration ne s'explique pas par un unique chemin, mais par un **ensemble** d'interactions locales qui se cristallisent en un regroupement solidaire. L'analyse du cluster met en avant les entités centrales et leur degré de liaison ω , rendant la décision plus **transparente**.

1.7.5.3. Couplage Symbolique pour des Justifications de Haut Niveau

Dans la section 1.5.7, il a été expliqué comment intégrer des **règles** ou des **concept**s symboliques au sein du **Deep Synergy Learning (DSL)** de manière à fusionner des entités strictement sub-symboliques (features perceptuelles, segments, etc.) et des entités logico-sémantiques. Cette intégration prend toute son importance pour l'**explicabilité**, car elle autorise la production de justifications directement lisibles, sans nécessiter un métamodèle séparé. Si l'on imagine une **règle** \mathcal{E}_{rule} affirmant “ cercle + chiffre => panneau de limitation”, la présence d'un lien fort $\omega_{rule, visCircle}$ met en évidence la façon dont la règle coopère avec l'entité $\mathcal{E}_{visCircle}$ qui détecte les formes circulaires, et un autre lien $\omega_{rule, txt30}$ connecte cette règle à l'entité textuelle représentant le chiffre “30”. La combinaison de ces liaisons convainc ainsi que la **reconnaissance** du panneau 30 km/h découle d'une synergie tangible entre la règle symbolique “ cercle + chiffre” et les observations sub-symboliques qui valident les deux prérequis.

Cette **cohabitation** symbolique–subsymbolique s'avère donc profitable pour expliquer pourquoi une règle s'active, de quelle façon elle est validée par la scène visuelle ou contextuelle, et comment cette coopération se traduit en une conclusion. Le réseau **DSL**, formé de pondérations $\{\omega_{i,j}\}$, expose explicitement la manière dont la règle “ \mathcal{E}_{rule} ” tisse des liens avec des features visuelles ou auditives pour générer une reconnaissance de haut niveau. Il est également possible d'**extraire** des proto-règles à partir des clusters émergents du réseau. Lorsqu'un **cluster** se maintient de façon stable et regroupe certaines variables, comme des marqueurs biologiques, un symptôme et un diagnostic, on peut en déduire une **règle émergente**. Par exemple, si X, Y et Z sont élevés, la synergie avec le diagnostic D s'accroît considérablement, ce qui revient à suspecter une pathologie P. Cette forme d'**abstraction** dérive naturellement de la dynamique d'auto-organisation du **DSL** et fournit une explication concise, en reliant directement le diagnostic à un sous-réseau stable de variables médicales fortement pondérées. On franchit ainsi une étape supplémentaire vers une **explicabilité** plus intuitive, où l'on ne se contente pas de mentionner le rôle de quelques features, mais où l'on propose une **règle** ou une **raison** qui s'apparente à un énoncé symbolique, ancré dans les liaisons sub-symboliques du réseau.

1.7.5.4. Points de Vigilance : Graphes Trop Grands, Hétérogénéité Massive

Dans un **Deep Synergy Learning** de forte dimension, la **visualisation** intégrale de toutes les liaisons $\omega_{i,j}$ ou l'exploration exhaustive des chemins reliant deux entités devient inenvisageable. La taille du graphe peut atteindre plusieurs milliers ou millions de noeuds et de liens, rendant toute inspection manuelle impossible. Afin de surmonter ces limites, il est utile de mettre en place des **mécanismes** de **filtrage** ou d'**agrégation**. Le filtrage consiste à écarter les liaisons dont la pondération $\omega_{i,j}$ reste en deçà d'un **seuil** minimal, ou qui ne participent pas de manière significative à un **cluster** pivot. Cette méthode abaisse drastiquement la densité du réseau et en facilite l'interprétation. L'agrégation, quant à elle, regroupe certaines entités en **macro-noeuds**, de sorte que l'on obtienne un **graphe résumé** nettement plus petit, faisant émerger les **clusters** importants tout en préservant la cohérence des liens principaux.

Lorsque les entités du DSL recouvrent une **hétérogénéité** massive, intégrant par exemple de la vision, de l'audio, du texte et des modules symboliques, la **compréhension humaine** impose de clarifier la correspondance entre chaque entité et les données originales. Un patch d'image doit être

décrit par sa position ou sa nature dans la scène, un segment audio doit renvoyer à la phrase ou au moment précis de l'enregistrement, un concept symbolique doit être relié à son énoncé explicite. Cette **traduction** partielle est essentielle pour offrir une **explicabilité** satisfaisante. L'utilisateur comprend pourquoi une liaison $\omega_{i,j}$ est forte, car un patch visuel identifié comme "zone X" présente une **synergie élevée** avec un certain concept ou un segment audio pertinent. Il est donc indispensable d'associer à chaque entité \mathcal{E}_i un ensemble de **méta-informations** décrivant son rôle, son lien avec l'input brut et sa typologie (vision, texte, règle symbolique). Grâce à ce balisage, la **mise en récit** du réseau DSL, même de grande taille, peut s'effectuer de façon plus synthétique et plus pertinente, en se concentrant sur un sous-ensemble restreint d'arêtes fortement pondérées et sur des macro-clusters résument la structure globale.

Conclusion

La **structure** en entités et liaisons synergiques du **Deep Synergy Learning** confère un **avantage** potentiel pour l'**explicabilité**. On peut :

- **Identifier** des clusters,
- **Exhiber** des sentiers $\mathcal{E}_i \rightarrow \mathcal{E}_j$ hautement pondérés,
- **Mettre en avant** des règles symboliques si présentes,
- **Comparer** des liens forts entre un utilisateur/contenu (dans le cas de recommandation) ou un patient/diagnostic (dans le cas médical).

Cependant, l'**explosion** du nombre d'entités et de liens peut rendre difficile la simple "lecture" de la configuration globale. On doit alors s'équiper de **techniques** de filtrage, d'**agrégation** (macro-clusters), ou de **génération** d'explications locales (chemins courts). La cohabitation possible du **symbolique** et du **sub-symbolique** aide à concrétiser des **justifications** plus claires. Ainsi, le DSL peut jouer un rôle dans la mouvance d'**Explainable AI**, à condition de manier soigneusement la **complexité** et la **variété** des entités afin de délivrer une **vision** cohérente et compréhensible à l'utilisateur final.

1.7.6. Considérations Éthiques et Réglementaires

Au-delà des enjeux techniques (scalabilité, stabilité, explicabilité), le **Deep Synergy Learning (DSL)** soulève également des questions **éthiques** et **réglementaires**. En effet, l'auto-organisation des entités au sein d'un **réseau** qui évolue sans hiérarchie prédéfinie peut introduire une forme d'**imprévisibilité** ou de **décision distribuée**. Lorsqu'il s'agit de systèmes critiques (médical, financier, militaire, etc.), il est indispensable de **responsabiliser** la conception et l'usage de tels modèles. De même, dans des applications plus courantes (recommandation, transport, logistique), la manière dont les données sont exploitées et dont les clusters se forment peut produire des **biais** ou altérer la **transparence** vis-à-vis des utilisateurs. Cette sous-section (1.7.6) examine :

- Le **risque** de perte de contrôle ou d'incompréhension du comportement d'un système auto-organisé,

- Les **biais** possibles et la nécessité de garanties contre la discrimination ou l'iniquité,
- Les **règles** et normes (RGPD, réglementations sectorielles) exigeant traçabilité et protection des données,
- Le rôle de **valeurs humaines** (sécurité, dignité, respect de la vie privée) dans la conception de DSL responsables.

1.7.6.1. Perte de Contrôle et Responsabilités

L'un des attraits majeurs du **Deep Synergy Learning (DSL)** réside dans sa **dynamique d'auto-organisation**. Les entités se regroupent en **clusters** en fonction des synergies détectées, sans nécessiter de programmation explicite pour gérer leurs interactions. Toutefois, cette souplesse peut engendrer des **comportements** difficiles à anticiper, dans la mesure où ni le concepteur, ni l'opérateur, ni le développeur n'ont directement programmé la structure finale. Dans des contextes où la **responsabilité** est cruciale, notamment en milieu médical ou bancaire, il importe de savoir **qui** assumera les conséquences si le système commet un choix discutable ou s'il engendre un dommage. Avec un réseau neuronal "classique", déjà, l'**opacité** rend l'attribution de responsabilité délicate ; dans un DSL à auto-organisation plus distribuée, le degré de complexité est plus élevé encore.

Pour pallier cette difficulté, il est envisageable de munir le système d'un **journal** d'événements et de mises à jour, permettant de **tracer** l'historique de la pondération $\omega_{i,j}$. Théoriquement, on pourrait reconstituer le "chemin causal" entre une observation et une décision, même au sein d'un graphe évolutif. Cette traçabilité demeure toutefois ardue à mettre en œuvre si l'on veut documenter en continu un réseau de grande dimension.

La loi ou les **régulations** sectorielles (ex. aéronautique, médical, bancaire) imposent souvent de **certifier** une partie du comportement du système avant son déploiement. Un DSL auto-organisé risque de "**changer**" de configuration quasi à chaque itération, rendant hasardeuse toute garantie statique. Il existe alors plusieurs stratégies. Il est possible de borner la **vitesse** d'évolution (en limitant le produit $\eta \times \tau$, ou en imposant un pas maximal) pour qu'il demeure un **voisinage** de configurations déjà validées. On peut aussi intégrer un **module** de supervision hiérarchique (voir la section 1.7.4.3) qui invalide certaines reconfigurations jugées non conformes à des normes ou des plans de sûreté. Enfin, il est envisageable d'appliquer des **tests** de robustesse, ou "stress tests", simulant divers scénarios de données pour s'assurer qu'aucune configuration délétère ne surgisse, ou du moins qu'elle reste suffisamment rare et détectable.

Ces considérations reflètent un compromis entre la **liberté** synergique, qui fait la force conceptuelle du DSL, et la **maîtrise** requise dans les secteurs critiques. Les débats sur la responsabilité sont dès lors analogues à ceux suscités par les réseaux neuronaux opacifiés, mais se complexifient en raison de la reconfiguration continue du réseau. Les utilisateurs, exploitants et autorités de tutelle doivent tenir compte de la difficulté d'isoler un instantané du DSL pour en vérifier la validité et assumer les conséquences de ses décisions évolutives.

1.7.6.2. Biais, Équité et Discrimination

Dans un **Deep Synergy Learning (DSL)** appliqué à des tâches de recommandation ou de décision, il arrive que la dynamique de formation des **clusters** reflète des **biais** préexistants dans les données, ce qui peut engendrer des effets discriminants. Imaginons un système de **scoring** pour l'attribution de crédits bancaires. Si les bases de données d'emprunteurs présentent une corrélation injustifiée entre certaines variables socio-économiques et la capacité de remboursement, le **DSL** risque de regrouper un certain sous-groupe dans un **cluster** jugé risqué, ou de l'isoler dans un espace périphérique, renforçant une forme d'exclusion. Ces liaisons faibles ou inexistantes maintiennent le sous-groupe à l'écart des opportunités, ce qui peut amplifier la ségrégation. Moins il coopère avec le reste du réseau, plus sa synergie chute, réduisant ainsi son accès équitable au crédit.

Ce phénomène d'**auto-renforcement** est potentiellement dangereux. Dans un simple réseau neuronal supervisé, des régulations ou des contraintes peuvent être imposées pour limiter l'exploitation de variables sensibles. Mais dans un **DSL**, la situation est plus complexe, car les **entités** se lient ou se séparent selon la synergie globale perçue. Un sous-groupe minoritaire, s'il n'est pas bien représenté dans les données, peut se voir marginalisé. Il convient alors d'établir des **mécanismes** préventifs ou correctifs. Il est envisageable, par exemple, de **pénaliser** l'isolation injustifiée d'un sous-groupe, en introduisant un terme additionnel dans la fonction d'énergie ou dans l'équation de mise à jour des pondérations. Ce terme peut imposer qu'un certain ensemble d'entités (associées à un attribut protégé) ne soit pas trop "distant" du reste ou qu'un cluster ne se forme pas exclusivement autour d'une caractéristique stigmatisante. Un autre moyen consiste à **rééquilibrer** les données avant l'apprentissage, afin de fournir au DSL une image plus fidèle et plus inclusive de la population, ou à injecter artificiellement des entités compensatoires, assurant que le réseau aura de quoi évaluer la synergie de manière équitable.

Dans un cadre légal, des obligations comme le **RGPD** (Règlement Général sur la Protection des Données) ou les lois anti-discrimination exigent que la décision ne dépende pas d'informations relatives à l'origine ethnique, à la religion ou à d'autres critères protégés. Le **DSL**, en créant des entités et des liaisons auto-organisées, doit donc s'assurer que certaines variables n'encoderont pas implicitement ces attributs sensibles, sous peine d'aboutir à des comportements discriminants difficilement décelables. Il est crucial de mettre en place des **audits** récurrents du réseau pour vérifier la distribution des **clusters**, observer si un groupe spécifique demeure systématiquement isolé ou sous-pondéré et, le cas échéant, forcer une mise à jour corrective. De tels audits peuvent se fonder sur l'inspection des pondérations $\{\omega_{i,j}\}$ associées à un ensemble d'entités. Ils peuvent également imposer une mesure d'équité en tant que contrainte externe, qui intervient de façon hiérarchique pour éviter la constitution de partitions inéquitables. L'objectif demeure le **contrôle** du **DSL** tout en respectant l'esprit d'**auto-organisation**, afin de combiner l'efficacité de la synergie et la nécessité de préserver une **égalité** ou une **justice** dans les décisions rendues.

1.7.6.3. Protection des Données et Droit à l'Oubli

Dans une architecture **Deep Synergy Learning (DSL)**, il arrive fréquemment que le réseau comprenne des **entités** liées à des personnes (utilisateurs, patients). Les législations comme le **RGPD** (Règlement Général sur la Protection des Données) en Europe imposent plusieurs obligations, dont la **minimisation** des données recueillies (ne conserver que l'essentiel), le **droit à**

l'oubli (un individu peut exiger la suppression de ses données personnelles) et des règles d'**anonymisation** ou de pseudonymisation pour éviter la ré-identification. Le réseau DSL, en tant que graphe auto-organisé et évolutif, doit donc intégrer des procédures garantissant que l'on puisse effacer ou rendre inopérante l'entité \mathcal{E}_i correspondant à une personne qui souhaite être oubliée.

La présence d'une entité \mathcal{E}_i dans un **Synergistic Connection Network** peut avoir des répercussions à de multiples endroits, car l'**auto-organisation** a pu former plusieurs **clusters** où \mathcal{E}_i joue un rôle. Pour satisfaire le droit à l'oubli, il faut d'abord **retracer** cette entité dans le graphe, puis la **retirer** du réseau ou la rendre totalement anonyme. Il convient de supprimer ou de neutraliser les liaisons $\omega_{i,j}$ reliées à l'entité \mathcal{E}_i . Lorsque l'on procède à cette suppression, on doit parfois **réajuster** la **synergie** locale afin que les entités \mathcal{E}_j anciennement liées à \mathcal{E}_i ne se retrouvent pas avec des pondérations invalides. L'opération de "déréférencement" peut donc influer sur l'équation d'évolution, qui doit détecter et ignorer toute trace associée à \mathcal{E}_i . L'objectif est d'éviter que des **résidus** dans la structure du réseau ne permettent de re-identifier l'utilisateur a posteriori, contrevenant ainsi aux principes de confidentialité exigés par la loi.

Le RGPD recommande également la **pseudonymisation** et la **minimisation** des informations stockées, ce qui affecte la façon dont les entités DSL sont construites. Il est parfois nécessaire de ne stocker, dans une entité \mathcal{E}_i , qu'une version agrégée (par exemple, localisation au niveau d'une ville plutôt que d'une rue) et d'exclure tout identifiant direct pour se conformer aux dispositions légales. Si le réseau DSL garde un trop haut niveau de précision, il peut être considéré comme non conforme à la réglementation. Une approche consiste à instaurer un **processus** d'anonymisation périodique. Après un certain temps, on fusionne les entités trop granulaires ou on supprime certains détails, réduisant la granularité tout en préservant l'essence des **synergies** globales. Cette pratique garantit une protection de la vie privée tout en permettant au DSL de continuer à fonctionner de manière auto-organisée et réactive, mais sur la base de données moins sensibles ou déjà anonymisées.

1.7.6.4. Transparence, Explicabilité et Contrôle

Dans certains contextes réglementés, comme le **RGPD** ou des lois spécifiques à un domaine (médical, bancaire, etc.), les usagers ou patients ont la faculté d'exiger une **explication** sur la décision qui les concerne. Le **Deep Synergy Learning (DSL)**, grâce à ses **clusters** et à ses **liaisons** plus faciles à interpréter qu'un pipeline neuronal opaque, peut répondre à cette exigence. Il demeure toutefois nécessaire de s'assurer que l'évolution du réseau (les pondérations $\omega_{i,j}$ dans le Synergistic Connection Network) demeure **traçable** dans le temps. Pour cela, on peut tenir un **historique** des principales mises à jour ou conserver des **snapshots** à intervalles réguliers, afin de justifier a posteriori comment tel ou tel sous-réseau s'est formé.

Une **infrastructure** d'explication (voir section 1.7.5) doit de surcroît synthétiser le résultat de la **dynamique** auto-organisée. Il ne s'agit pas nécessairement d'exposer l'entièreté des liens $\omega_{i,j}$, mais d'extraire des **chemins** ou des **clusters** pivot permettant de reconstituer le **raisonnement local**. Une difficulté apparaît lorsque le système évolue de manière rapide ou chaotique. L'explication peut alors varier significativement d'une itération à l'autre, compromettant la cohérence globale. Des **mécanismes** de stabilisation ou de **pas** d'évolution limités (contrôle de η)

et de τ , par exemple) peuvent aider à préserver une relative continuité, rendant les explications plus consistantes dans le temps.

Pour de nombreuses **applications** sensibles, la législation ou la pratique courante impose un **contrôle humain** dans la boucle. Conformément à un principe d'audit ou de validation, un expert (médecin, analyste, manager) examine la configuration actuelle du DSL et doit pouvoir **accepter**, **refuser** ou **rectifier** certaines décisions. L'**interface** qui présente les clusters, leurs pondérations dominantes et l'historique succinct de leur formation doit donc être la plus claire possible, en hiérarchisant l'information. L'expert peut alors intervenir en imposant une pénalisation sur des liaisons considérées comme non conformes (sécurité, éthique, cohérence métier). Cette démarche articule la liberté **auto-organisée** du DSL (qui adapte continûment ses liaisons) avec un **pilotage humain**, assurant un respect des **cadres légaux** et éthiques, tout en tirant parti de la plasticité et de l'auto-organisation qu'offre le Deep Synergy Learning.

Conclusion

Les **considérations éthiques et réglementaires** forment un **pilier** incontournable lors du déploiement de **systèmes auto-organisés** comme le DSL. On peut résumer les **grandes lignes** de vigilance :

- **Responsabilité et traçabilité** : éviter qu'un système auto-adaptatif devienne totalement opaque quant à ses décisions ou conséquences, mettre en place un journal d'événements, un module de supervision.
- **Biais et équité** : surveiller la formation de clusters discriminants, adapter la dynamique pour prévenir les exclusions ou dominations injustes.
- **Protection des données** : garantir l'anonymisation, la pseudonymisation, et le droit à l'oubli dans un graphe évolutif (ce qui n'est pas trivial techniquement).
- **Transparence et droit à l'explication** : mettre en place des mécanismes d'explicabilité (voir 1.7.5) assurant que l'utilisateur comprend la logique du DSL.

Au final, l'**originalité** du DSL ne saurait dispenser d'une **responsabilisation** solide. La recherche sur l'IA dite "responsable" ou "de confiance" s'applique tout autant (voire davantage) à un modèle **auto-organisé** qui, par définition, échappe aux schémas de contrôle univoque. Les prochaines sections (et développements futurs) devront approfondir les outils et protocoles pour faire du DSL une **technologie** à la fois puissante, adaptable, et conforme aux **valeurs** et **règles** en vigueur.

1.7.7. Comparaisons Expérimentales avec d'Autres Approches

Dans les sections précédentes (1.7.1 à 1.7.6), nous avons exploré les **défis** et **contraintes** du **Deep Synergy Learning (DSL)** en matière de **complexité**, de **qualité des données**, de **stabilité**, d'**explicabilité**, et d'**éthique**. Pour renforcer la **pertinence** de ce paradigme en pratique, il est essentiel de réaliser des **comparaisons expérimentales** rigoureuses avec des méthodes existantes. Ces évaluations incluent les **réseaux neuronaux profonds** (CNN, RNN, Transformers), les

méthodes de clustering (k-means, DBSCAN), les algorithmes d'optimisation (ordonnancement) et les approches neuro-symboliques traditionnelles. Cette sous-section (1.7.7) aborde :

- Les critères de comparaison essentiels (performances quantitatives, robustesse, adaptativité...),
- Les protocoles expérimentaux pour confronter le DSL à d'autres techniques (tests sur données stationnaires et non stationnaires, analyses de complexité, etc.),
- Les scénarios où le DSL a un net avantage (auto-organisation, peu de supervision, scénarios évolutifs),
- Les limites et enseignements que de tels benchmarks peuvent révéler, guidant l'amélioration du DSL.

1.7.7.1. Choisir les Critères de Comparaison

Dans la plupart des tâches d'apprentissage ou de décision (vision, audio, recommandation, diagnostic médical), on évalue la précision ou des métriques dérivées comme l'accuracy, la F1-score ou l'AUC. Lorsque l'on souhaite comparer un Deep Synergy Learning (DSL) avec, par exemple, un CNN ou un Transformer, on peut se borner à mesurer son taux de reconnaissance (dans un scénario de classification d'images) ou la sensibilité et la spécificité (dans le cadre d'un diagnostic médical). Toutefois, ces scores globaux ne reflètent pas nécessairement la richesse de la structure auto-organisée du DSL, qui excelle par sa plasticité et son adaptation continue, sans exiger nécessairement de labels massifs. Il est donc essentiel, lors de l'évaluation, d'incorporer des critères tenant compte de la dynamique de l'auto-organisation.

Le DSL se veut adaptatif en présence de bruit, de pannes de capteurs ou de changements de distribution (concept drift). Pour mettre en évidence cette capacité, il est concevable de concevoir des expériences où la distribution des données varie progressivement ou subitement en cours d'apprentissage. Dans un cas classique, un réseau neuronal (CNN, RNN) va souvent exiger une séance de re-fine-tuning global, tandis que le DSL doit pouvoir réajuster localement ses liaisons $\{\omega_{i,j}\}$. On peut alors comparer la vitesse de réadaptation et la qualité de la reprise de performance. On mesure, par exemple, la chute temporaire de précision lorsqu'un nouveau type de données apparaît, ainsi que le temps qu'il faut au modèle pour retrouver un niveau satisfaisant. Le DSL a pour atout de pouvoir réorganiser ses clusters et de créer ou rompre des liens en fonction des synergies locales, ce qui peut se révéler plus rapide et moins coûteux qu'un re-entraînement complet.

La taille du modèle (nombre de paramètres ou de liaisons actives) et la charge de calcul (temps CPU, mémoire GPU) comptent également parmi les indicateurs déterminants. Un DSL peut rester parcimonieux s'il opère une sparsification rigoureuse de la matrice ω . Si, au contraire, on ne limite pas la prolifération des liens, la structure risque de gonfler et de devenir onéreuse en ressources. L'évaluation doit inclure la scalabilité et l'efficience. Il est crucial de vérifier si, face à un grand nombre d'entités, le DSL maintient une densité de liens contrôlée (voir section 1.7.1) et s'adapte aux scénarios évolutifs tels que les pannes ou les modifications de distribution. On

évalue également les **coûts de mise à jour** et de **mémoire**, afin de garantir une gestion efficace des ressources.

1.7.7.2. Protocoles Expérimentaux Envisageables

Pour **évaluer** la pertinence d'un **Deep Synergy Learning (DSL)** dans des scénarios variés, on peut le confronter à des bases de données classiques ou à des cas pratiques bien établis :

- Des ensembles **d'images** (MNIST, CIFAR, éventuellement un sous-ensemble d'ImageNet) pour évaluer la classification ou la détection, en analysant la **précision** obtenue par l'auto-organisation visuelle des entités (patchs, features, etc.) face à un CNN ou un Transformer.
- Des ensembles **audio** (LibriSpeech, UrbanSound8K) pour tester la reconnaissance de sons ou la transcription de parole, où le DSL formerait des entités "segment audio" et "caractéristiques acoustiques", comparées à un système RNN ou CNN entraîné de façon traditionnelle.
- Des **bases de recommandation** (MovieLens, Amazon reviews) pour mesurer la qualité des suggestions formulées par le DSL ; on confronte alors la structure auto-organisée (clusters d'utilisateurs, de contenus, etc.) aux méthodes de factorisation matricielle ou aux modèles de collaboration neuronaux, en examinant le **Recall@K**, la **Precision@K** ou la **NDCG**.
- Des **données médicales** (ensembles de pathologies, EHR) afin de juger la capacité du DSL à découvrir des clusters pathologiques ou à émettre des hypothèses de diagnostic, comparativement à un modèle supervisé (réseau profond, arbre d'ensemble) ou semi-supervisé.
- Des **scénarios robotiques** utilisant des simulateurs comme **Gazebo** ou **PyBullet** permettent d'évaluer la coordination sensorimotrice. Un **DSL** connecte capteurs et effecteurs, testant sa **robustesse** face aux variations des conditions et à la détection de pannes.

Dans chaque domaine, les **résultats** finaux tels que le taux de reconnaissance, le score de recommandation ou le F1-score diagnostique sont comparés aux performances de modèles **classiques** comme les **CNN**, **RNN**, **Transformers**, **K-means** ou la **factorisation matricielle**. L'évaluation inclut également la **taille du modèle** en termes de paramètres et de liaisons, la **complexité computationnelle**, ainsi que la **vitesse d'inférence**.

Le **DSL** se veut également **adaptatif** en cas de changements de distribution (concept drift), de pannes de capteur, de conditions altérées. Pour l'exercer à cette fin, il est utile de mettre en œuvre des **scénarios** où les données subissent un glissement graduel ou une modification brutale (apparition de nouvelles classes, hausse du bruit, changement d'éclairage, etc.). Sur la timeline $\{t_0, \dots, t_f\}$, on évalue deux aspects. On observe comment un réseau neuronal classique comme un CNN ou un RNN réagit face au drift des données, souvent en nécessitant un re-fine-tuning ou en subissant une perte partielle d'acquis menant à un catastrophic forgetting. On analyse également comment le **DSL**, sans recourir à un réapprentissage massif, ajuste localement ses liaisons $\omega_{i,j}(t)$ afin de s'adapter aux nouvelles conditions.

On observe la **performance** en continu (taux de classification, AUC, indice de recommandation) et on étudie la **vitesse** de réadaptation. Le **DSL** devrait, en principe, reporter des ajustements plus localisés et moins coûteux qu'un re-entraînement total, tout en préservant une cohérence globale. Ceci permet de vérifier empiriquement que la **robustesse** et la **plasticité** conférées par la mise à jour distribuée des pondérations ω confèrent un avantage aux méthodes d'auto-organisation, surtout dans des domaines où les données ou le contexte varient régulièrement.

1.7.7.3. Scénarios où le DSL a un Avantage Notable

Le **Deep Synergy Learning (DSL)** se distingue particulièrement dans des environnements où l'on dispose de **peu ou pas de labels** (section 1.5.5) ou bien où la supervision est coûteuse et incomplète. En mode **non supervisé** ou faiblement supervisé, un réseau neuronal classique (tel qu'un CNN) peine à extraire des structures cohérentes, car il manque d'exemples étiquetés pour guider l'apprentissage. Le **DSL**, au contraire, s'appuie sur sa **fonction de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ pour **auto-organiser** les entités, sans nécessiter un volumineux jeu de labels. Les **clusters** émergent alors de la simple coopération (ou co-occurrence) observée entre entités, ce qui rend le réseau particulièrement apte aux tâches d'**exploration** et de **regroupement** de données hétérogènes.

Un autre atout du DSL s'observe en **multi-modalité hétérogène** (section 1.5.2), où l'on combine image, audio, texte, voire des flux sensoriels divers. Les approches traditionnelles imposent souvent un **pipeline** de fusion rigide (concaténation tardive, attention inter-modale) dépendant de choix d'architecture prédéfinis. Le **DSL** adopte une approche plus flexible où chaque entité \mathcal{E}_i (visuelle, auditive, etc.) établit des **liaisons** $\omega_{i,j}$ en fonction de la **synergie** détectée. Ces connexions émergent et évoluent sans qu'un schéma fixe ne soit imposé, permettant une organisation dynamique du réseau. Cette structure évite la planification d'un module de "fusion" spécifique ; la mise en réseau est régie par la **dynamique** locale des pondérations, autorisant des clusters multimodaux à émerger naturellement.

Lorsque les **flux** de données sont **dynamiques** et sujets à des **pannes** ou à des transformations (sections 1.5.4 et 1.7.2.3), le DSL illustre également sa supériorité. Un capteur défectueux voit ses liaisons ω chuter, car la synergie avec le reste du système devient faible. Le réseau se réorganise alors sans qu'un nouvel entraînement global ne soit requis, contrairement à un réseau neuronal classique où une modification durable de la distribution (panne de capteur, concept drift) nécessite un **fine-tuning** ou un recalibrage important.

L'**explicabilité** du DSL revêt aussi un intérêt clé (sections 1.7.5 et 1.5.6). Les **clusters** ou les chemins de **synergie** identifiés entre entités procurent des explications plus détaillées qu'une simple "heatmap" d'attention. Dans un réseau traditionnel, la structure interne (couches, neurones) reste souvent opaque, malgré certaines techniques de visualisation de filtres. Le DSL, en attribuant des **identités** distinctes à ses nœuds et en laissant les liens $\omega_{i,j}$ témoigner de leur degré de coopération, permet une **compréhension** plus directe de la raison pour laquelle tel sous-réseau est né, ou tel couple $(\mathcal{E}_i, \mathcal{E}_j)$ a vu sa liaison renforcée.

Dans l'ensemble, une **comparaison expérimentale** (section 1.7.7.2) révélera que, face à un CNN ou à un Transformer, le DSL excelle dans des conditions de **data faible** ou de **distribution changeante**, ou encore quand l'on souhaite un **rendement** sémantique mieux explicité par un

graphe de clusters. Ces propriétés séduisent dans des scénarios où l'on doit adapter le système en continu, respecter des contraintes d'équité, ou justifier les décisions auprès d'utilisateurs ou d'experts.

1.7.7.4. Limites et Enseignements Possibles

Lors de l'évaluation d'un **Deep Synergy Learning (DSL)** à large échelle, on peut constater que la structure du réseau, qui autorise jusqu'à $O(n^2)$ liaisons dans le cas binaire, devient difficile à manipuler si les entités sont très nombreuses. Il peut être nécessaire d'imposer une **sparsification** stricte (par exemple, ne maintenir qu'un voisinage restreint pour chaque entité) ou de recourir à un **algorithme local** qui ne calcule pas la synergie entre toutes les paires d'entités, mais seulement entre celles jugées proches dans un espace de représentation. Les expérimentations comparatives, que ce soit en temps de calcul ou en mémoire GPU, permettent d'identifier le **seuil** au-delà duquel le DSL se montre moins performant que d'autres architectures classiques (CNN, RNN, Transformer) optimisées pour le passage à l'échelle sur des supports matériels massivement parallèles.

Par ailleurs, si l'on dispose d'un **jeu de données étiquetées** gigantesque (par exemple la totalité d'ImageNet, avec plus de 14 millions d'images), un réseau supervisé standard tel qu'un CNN ou un Transformer minutieusement entraîné peut atteindre d'excellentes performances de classification top-n. Le **DSL**, plus orienté vers l'**auto-organisation** non supervisée ou faiblement supervisée, pourrait dans ce cas demeurer en retrait si l'on n'exploite pas pleinement toute la supervision à disposition. On peut envisager des versions **supervisées** ou **semi-supervisées** du **DSL** où la **synergie** est partiellement orientée par des labels. Toutefois, une question demeure ouverte. Ces labels seront-ils suffisants pour rivaliser avec un **CNN** entraîné par descente de gradient pure, en particulier lorsque le volume d'annotations est massif ? Ces expériences permettront de clarifier dans quelle mesure le DSL parvient à tirer parti d'une supervision large sans renoncer à sa plasticité et à sa modularité intrinsèque.

Ces limites et enseignements soulignent la nécessité de concevoir des **protocoles hybrides**, dans lesquels on combine la liberté d'auto-organisation du DSL (favorable à l'adaptation continue ou à la gestion de scénarios peu labellisés) avec la force d'un apprentissage supervisé classique lorsqu'un dataset étiqueté d'ampleur est disponible. C'est dans cet espace de compromis que réside probablement l'avenir des approches synergiques à l'échelle industrielle.

Conclusion

La **validation expérimentale** du **DSL** face aux approches dominantes, comme les réseaux neuronaux profonds, le clustering ou les méthodes d'optimisation, est essentielle pour **quantifier** sa performance, sa robustesse et sa capacité d'adaptation. Elle permet également d'**identifier** ses atouts, notamment l'apprentissage sans label massif, la gestion du drift et l'intégration de modalités hétérogènes. Enfin, elle aide à **révéler** ses limites, telles que le coût en calcul, la nécessité de mesures de synergie fiables et les difficultés sur des tâches fortement supervisées.

Des **benchmarks** variés (vision, audio, recommandation, diagnostic, robotique) permettront de **cartographier** où le DSL s'impose et où il doit s'allier à d'autres techniques (CNN pré-entraîné pour extraire des features, par exemple). Finalement, ce travail de confrontation expérimentale, mené en parallèle des réflexions éthiques (1.7.6) et de la recherche sur l'optimisation (1.7.3), consolide la position du DSL comme un **paradigme** original — potentiellement complémentaire, voire compétitif — dans le paysage actuel de l'apprentissage automatique.

1.8. Positionnement du DSL dans l'Évolution de l'IA

Après avoir étudié les fondements du **Deep Synergy Learning (DSL)** et ses multiples applications (sections 1.1 à 1.6), ainsi que les défis et contraintes (section 1.7), il est opportun de s'interroger sur la **place** qu'occupe le DSL dans l'**évolution** de l'intelligence artificielle. L'IA contemporaine repose sur deux grands paradigmes. D'un côté, l'**IA sub-symbolique**, incarnée par les réseaux neuronaux et les méthodes statistiques, s'est imposée avec l'essor de l'apprentissage profond. De l'autre, l'**IA symbolique**, issue des systèmes experts et de la logique formelle, offre une meilleure interprétabilité et une manipulation plus explicite sous forme de langage structuré.

Le **DSL**, en proposant une **auto-organisation** et une **dynamique** des liens fondée sur la synergie, apporte un **paradigme** susceptible de **réconcilier** certains points de friction entre ces approches. On peut se demander :

- S'il offre un **point** de convergence entre l'**IA symbolique** et l'**IA sub-symbolique**,
- Comment il peut **coexister** ou se **substituer** à l'apprentissage profond,
- Dans quelle mesure le **renforcement** (RL) ou la **logique** se fondent dans une trame DSL,
- Le **rôle** d'éléments comme la **mémoire** et l'**attention** dans une architecture auto-organisée,
- Les **tendances** futures, notamment vers une IA plus "forte" ou consciente,
- L'**impact** interdisciplinaire, en particulier avec les neurosciences,
- Et, de manière générale, comment le DSL s'inscrit dans la **progression** de l'IA moderne et ce qu'il peut apporter de nouveau.

Cette section (1.8) aborde ainsi le **positionnement** du DSL dans le paysage AI actuel et futur, regroupant sept sous-points :

- **IA Symbolique vs IA Sub-symbolique : Intégration Potentielle** (1.8.1)
- **DSL et Apprentissage Profond : Collaboration ou Substitution ?** (1.8.2)
- **Approches Hybrides : DSL, RL (Reinforcement Learning) et Logique** (1.8.3)
- **Rôle de la Mémoire et de l'Attention dans le DSL** (1.8.4)
- **Tendances Futures : Vers une IA Forte, Consciente ?** (1.8.5)
- **Effet Sur la Recherche Interdisciplinaire** (1.8.6)
- **Exemple de Convergence : DSL & Neurosciences** (1.8.7)

Nous commencerons par revisiter (1.8.1) la vieille dichotomie IA symbolique vs sub-symbolique, pour voir en quoi le **DSL** pourrait établir des passerelles novatrices.

1.8.1. IA Symbolique vs IA Sub-symbolique : Intégration Potentielle

L'**histoire** de l'IA est souvent présentée comme marquée par un **dualisme** :

- **L'IA symbolique**, qui manipule des règles, des faits, des ontologies dans un langage logique ou pseudo-logique (ex. Prolog, systèmes experts). Elle est réputée plus “interprétable” et plus “rigoureuse” dans le raisonnement, mais manque de flexibilité face aux données massives et bruitées.
- **L'IA sub-symbolique**, typiquement les réseaux neuronaux et méthodes statistiques, qui excellent en **apprentissage** (vision, NLP, etc.) en profitant de nombreux exemples, mais peinent à manipuler des structures conceptuelles complexes ou des règles explicites.

1.8.1.1. Vers une IA Hybride ou Neuro-Symbolique

Au cours de la dernière décennie, un nombre croissant de recherches s'est attelé à **combiner** les approches symboliques (issues de la logique, des règles ou des ontologies) et les méthodes sub-symboliques (neuronales ou connexionnistes). Les travaux dits **neuro-symboliques** cherchent, par exemple, à implanter des **règles logiques** au sein d'un réseau neuronal ou, à l'inverse, à extraire des **règles** symboliques depuis les *features* d'un modèle déjà entraîné. L'objectif est de combiner la **puissance** des méthodes sub-symboliques, capables de traiter de grands volumes de données complexes, avec la **rigueur** et la **lisibilité** des approches symboliques, qui facilitent le raisonnement et l'explication.

Ces approches neuro-symboliques s'organisent souvent sous la forme de **systèmes hybrides**, où l'on place côté à côté une **composante** consacrée au traitement sub-symbolique (réseau neuronal) et une **composante** symbolique (règles ou moteur logique). Un **pont** dédié est alors chargé de faire correspondre ces deux volets, ce qui peut induire une architecture “collée” au sens où la dimension symbolique reste nettement délimitée de la dimension connexionniste. La **véritable** intégration, où le réseau lui-même serait libre d'accueillir à la fois des **concepts logiques** et des **signaux bruts** en adaptant sa topologie, demeure un défi qui n'est pas entièrement résolu dans la plupart de ces schémas.

Le **Deep Synergy Learning (DSL)** entend précisément répondre à cette problématique, car il propose une **dynamique** où la structure se reconfigure **auto-organisationnellement**, sans imposer de cloison fixe entre l'aspect symbolique et l'aspect sub-symbolique. Les **entités** \mathcal{E}_i peuvent incarner aussi bien un **flux** sensoriel (image, audio), qu'un **concept** symbolique, ou une **règle**. Les liaisons $\omega_{i,j}$ se créent ou se coupent selon la **synergie** détectée. Un noeud représentant une règle logique pourra donc se lier plus fortement à un patch d'image ou un module audio dès lors qu'il y trouve un **gain** mutuel. De cette manière, la frontière traditionnelle entre le bloc **symbolique** et le bloc **neuronal** cède la place à un **réseau** unifié, où chaque composante est traitée comme une entité à part entière, susceptible de coopérer librement pour faire émerger des **clusters** ou des **schémas** de raisonnement. Ce fonctionnement souligne l'ambition du DSL de faciliter la rencontre fluide entre la **rigueur** des modèles logiques et la **flexibilité** d'apprentissage des méthodes sub-symboliques.

1.8.1.2. Rôle du DSL : Auto-Organisation Unifiant Symbolique et Sub-Symbolique

Le **Deep Synergy Learning (DSL)** propose une **intégration** fluide entre les entités **sub-symboliques** (capteurs, features, embeddings neuronaux) et les **composantes symboliques** (règles, concepts logiques, ontologies). Dans la dynamique du DSL, les **liaisons** $\omega_{i,j}$ se créent ou se rompent en fonction de la **synergie** perçue (voir la section 1.4.5 pour la mise à jour itérative). Au lieu de séparer la logique dans un module détaché, on traite chaque **règle** ou **concept** comme un **nœud** $\mathcal{E}_k^{(\text{symb})}$ dans le **Synergistic Connection Network**. La coopération entre une règle symbolique $\mathcal{E}_k^{(\text{symb})}$ et une entité sub-symbolique \mathcal{E}_i (par exemple un capteur) repose sur une pondération $\omega_{\text{symb}, i}$ qui s'adapte selon la synergie. Si la règle se vérifie fréquemment grâce aux données de ce capteur, la liaison se renforce, tandis que dans le cas contraire, elle s'affaiblit progressivement.

Dans un **environnement** robotique, plusieurs **règles** logiques \mathcal{R}_k peuvent être définies pour assurer la sécurité et la performance des machines. Par exemple, une règle pourrait stipuler que si la **tension moteur** dépasse **5V** et que la **température moteur** excède **80°C**, alors une alerte de surchauffe doit être déclenchée.

Dans le cadre du **DSL**, une telle règle est modélisée sous forme d'une **entité symbolique** $\mathcal{E}_k^{(\text{symb})}$. Parallèlement, les différentes **entités sub-symboliques** représentent les capteurs physiques, comme la mesure de la tension et de la température. La **synergie** entre ces éléments se traduit par les pondérations $\omega_{\text{tension, règle}}$ et $\omega_{\text{temp, règle}}$, qui augmentent si les relevés des capteurs confirment régulièrement la condition de surchauffe.

Sans imposer une structure figée, le **réseau** établit un **micro-réseau** reliant de façon dynamique la règle symbolique aux capteurs pertinents. Si la synergie persiste dans le temps, un **cluster** stable se forme, associant la règle “surchauffe moteur” aux données sensorimotrices qui la valident. Cette organisation **auto-adaptative** permet de lier les règles aux données pertinentes sans supervision explicite.

Cette approche évite le recours à une **passerelle** rigide entre la logique formelle et les signaux bruts. La **topologie globale** du **DSL** se **réajuste** en fonction de la pertinence observée. Si la règle \mathcal{R}_k devient moins utile ou si de nouveaux **capteurs** plus adaptés émergent, les **pondérations** s'adaptent automatiquement. On obtient ainsi une **auto-organisation** où la dimension **symbolique** est pleinement intégrée à la dynamique décrite en 1.4.5, donnant un **substrat** unifié pour l'IA hybride.

1.8.1.3. Avantages et Limites

Le fait d'intégrer la **dimension symbolique** directement dans un **Deep Synergy Learning (DSL)** offre divers **avantages**. D'abord, la **logique** s'intègre directement à l'**architecture auto-organisée**. Il n'est plus nécessaire de superposer un **module de règles** indépendant à un **module sub-symbolique**, comme un réseau neuronal. Au sein du DSL, chaque règle \mathcal{R}_k devient une entité $\mathcal{E}_k^{(\text{symb})}$, et la **pondération** $\omega_{\text{symb}, i}$ traduit la coopération effective entre cette règle et n'importe quelle entité sub-symbolique \mathcal{E}_i . Cette façon de procéder rend possible l'**apprentissage continu** de l'utilité de la règle, puisque la pondération se régule en fonction de la “valeur ajoutée” détectée. Un autre **bénéfice** se manifeste en matière d'**explicabilité**. Il devient plus facile d'identifier un

cluster reliant une règle \mathcal{R}_k à des **entités perceptives** $\mathcal{E}_p, \mathcal{E}_q, \dots$, permettant ainsi de justifier une décision. On obtient dès lors une **transparence** plus claire qu'avec un système classiquement scindé où la logique reste à part.

Cependant, cette intégration comporte également des **limites** notables. Une première difficulté tient à la **quantification** de la synergie $S(\mathcal{E}_i, \mathcal{E}_k^{(\text{symb})})$ quand l'une des entités représente une **règle** ou un **concept** logique. Il s'agit de définir mathématiquement la notion de “gain” associé à l'activation conjointe d'une règle symbolique et d'un flux sub-symbolique. Ce n'est pas toujours trivial, surtout si la règle mélange des conditions logiques complexes. Par ailleurs, plus on introduit de **règles**, plus la combinatoire peut **exploser**, le réseau devant gérer de multiples $\omega_{(\text{symb}),i}$. La mise à jour itérative de ces liaisons doit alors tenir compte des possibles **contradictions** entre règles (ex. deux règles incompatibles ne sauraient être activées simultanément), ce qui alourdit la dynamique ω et nécessite des **mécanismes** de gestion de conflits ou de contradictions logiques.

Ces considérations exigent une **régulation** attentive, par exemple via des **termes** de parcimonie (limitation du nombre total de liaisons actives), des **pénalités** imposées aux liaisons incompatibles, ou un **contrôle** de la vitesse de croissance de certaines pondérations. Un tel encadrement évite l'explosion combinatoire de la structure et aide à résoudre de manière cohérente les potentielles contradictions entre règles symboliques. Malgré ces contraintes, la promesse demeure. Le **DSL** offre un cadre plus **organique**, où les **concepts** logiques et les **descripteurs** sub-symboliques coexistent de manière adaptative.

1.8.1.4. Perspectives sur l'Intégration Totale

Dans les modèles traditionnels, une **dichotomie** claire oppose l'**IA symbolique**, qui repose sur des règles logiques et la manipulation de symboles, et l'**IA sub-symbolique**, qui exploite les réseaux neuronaux et les méthodes connexionnistes pour traiter les données brutes par apprentissage statistique. Le **Deep Synergy Learning (DSL)** aspire à dépasser cette fragmentation en poussant l'**auto-organisation** à un niveau plus global. Les **entités symboliques** – vues comme de véritables “super-neurones” capables d’inférences logiques internes – coexistent avec des **entités sub-symboliques** qui extraient, à partir de données brutes, des features ou des embeddings. Toutes ces entités demeurent dans un **même** Synergistic Connection Network, au sein duquel les liens $\omega_{i,j}$ se créent, se renforcent ou s'affaiblissent en fonction de la synergie effectivement observée.

Lorsqu'une **règle symbolique** apporte un “gain” (par exemple, elle s’active souvent conjointement à un certain pattern perceptif), la liaison entre cette règle et les entités perceptives impliquées croît. Inversement, si une règle n'est presque jamais appliquée ou se trouve contredite par les observations, ses liens se dissolvent. De la même manière, certains **macro-clusters** voient le jour, rassemblant des **concepts** sémantiques et des **patches** perceptuels (image, audio, etc.) autour d'une **synergie** élevée. Un tel réseau ne fixe pas de **cloisonnement** rigide entre la dimension **symbolique** et la dimension **sub-symbolique**. Les deux coexistent et interagissent librement, se distinguant uniquement par leurs **rôles** respectifs au sein de l'architecture auto-organisée.

Cette intégration n'est pas strictement spéculative et trace la voie d'une **IA cognitive** plus étendue, qui mêlerait la **perception** (apprentissage sub-symbolique) et le **raisonnement** (entités logiques) au sein d'un seul et même **réseau** en évolution. Les applications potentielles sont nombreuses. Un système de **vision** peut enrichir son raisonnement en exploitant des règles symboliques, qu'elles soient définies ou émergentes. De même, un moteur **logique** bénéficie de la synergie avec les flux

sensoriels, renforçant ainsi l’interprétation et l’adaptabilité du réseau. Le **DSL** rend possible la création d’un **substrat** où, au fil de l’apprentissage, les connexions s’**enrichissent**, des **règles** naissent ou se dissolvent, et des **clusters** conceptuels émergent, sans qu’on doive prescrire a priori leur structuration. Le réseau devient un “lieu” de coopération continue entre la dimension **symbolique** et la dimension **connexionniste**, accréditant l’idée d’une IA vraiment **hybride**, capable de percevoir et de raisonner sans barrière artificielle.

1.8.2. DSL et Apprentissage Profond : Collaboration ou Substitution ?

Alors que le **Deep Learning** (au sens des réseaux de neurones profonds) occupe aujourd’hui une place dominante dans la recherche et les applications de l’IA, on peut se demander si le **Deep Synergy Learning (DSL)** vient s’y substituer ou, au contraire, s’il peut s’**allier** à ces techniques. En effet, le **DSL** propose une vision **auto-organisée** où les entités d’information découvrent et renforcent leurs liens synergiques, tandis que l’apprentissage profond repose typiquement sur des **architectures hiérarchiques** (CNN, RNN, Transformers) entraînées par **descente de gradient** sur un ensemble de données massif. Cette sous-section (1.8.2) explore la relation entre ces deux paradigmes en analysant s’ils sont en **concurrence**, **complémentaires**, ou destinés à une **convergence** progressive.

1.8.2.1. Les Forces Reconnues du Deep Learning

Depuis plus d’une décennie, l'**apprentissage profond** a accompli des progrès spectaculaires dans plusieurs domaines clés de l’intelligence artificielle. En **vision artificielle**, on constate sa capacité à exceller dans la **classification** d’images ou la **détection** d’objets, tandis qu’en **traitement du langage naturel**, il domine pour les tâches de **traduction**, de **question-réponse** ou de **résumé** automatique. On retrouve également des succès notables en **analyse audio**, en particulier dans la **reconnaissance vocale**, et dans des environnements de **jeux** comme ceux explorés par AlphaGo ou AlphaZero. Ces avancées reposent en premier lieu sur la **puissance** de l’apprentissage supervisé lorsqu’il est alimenté par de larges volumes de données annotées. La deuxième composante décisive réside dans les **architectures** spécialisées du deep learning (réseaux de neurones convolutifs en vision, Transformers en langage, etc.), qui exploitent la structure des données pour atteindre des performances remarquables. Enfin, ces progrès sont intimement liés au développement accéléré des **librairies logicielles** (TensorFlow, PyTorch...) et du **matériel** (GPU, TPU) dédiés aux calculs massivement parallélisés.

Malgré ces forces, le deep learning fait souvent l’objet de **critiques** qui mettent en évidence sa dépendance à une supervision très abondante, et sa difficulté à se réorienter lorsque la tâche ou le domaine change. Son cadre global demeure souvent **rigide**. Lorsqu’une nouvelle tâche survient, il faut procéder à une **reconfiguration** ou un **fine-tuning** spécifique, ce qui complique l’adaptation dans un environnement en perpétuelle évolution. Les réseaux neuronaux profonds manquent également de **plasticité** et de **résilience** lorsque des pannes de capteur ou des modifications du contexte perturbent la distribution des données ; la moindre altération exige un nouvel entraînement ou une nouvelle adaptation globale. S’y ajoute la question de la **transparence**. Le caractère **boîte noire** des modèles pose un défi majeur pour l’explicabilité. Il devient difficile de comprendre

précisément l'origine d'une prédiction ou d'une décision, surtout lorsque le modèle comporte des dizaines de couches et des millions, voire des milliards, de paramètres.

1.8.2.2. Le DSL comme Complément : Auto-Organisation et Adaptativité

Le **Deep Synergy Learning (DSL)** repose sur une **auto-organisation** distribuée. Des **entités**, qu'il s'agisse de *features* neuronales, de blocs d'apprentissage ou de règles symboliques, établissent des connexions dynamiques à travers des **pondérations synergiques** $\omega_{i,j}$. Ces pondérations évoluent en continu selon une loi d'**adaptation** locale, autorisant l'insertion ou la suppression d'entités sans nécessiter une refonte ou un réentraînement global du réseau. Ce mécanisme confère une **plasticité** et une **résilience** particulières, notamment en cas de nouveaux flux de données ou de variations contextuelles.

Il est aisément envisager un **scénario de collaboration** entre le deep learning traditionnel et le DSL. Par exemple, on peut employer un **CNN** (ou un **Transformer**) pour extraire des **représentations profondes** à partir d'images, de signaux audio ou de séquences textuelles. Ces représentations (vecteurs de *features*) se traduisent alors en un **ensemble** d'entités $\{\mathcal{E}_{\text{CNN}}\}$ reliées entre elles ou avec d'autres entités par des pondérations ω . Le **DSL** assure une **auto-organisation** dynamique en identifiant les **synergies** entre divers blocs de *features* ou entre *features* et règles symboliques. Il ajuste les **clusters** émergents en fonction de leur pertinence, qu'ils soient multimodaux, sémantiques ou contextuels. L'ajustement local des $\omega_{i,j}$ s'effectue de manière à ce que, si deux groupes de *features* se complètent fortement, un **macro-cluster** puisse émerger, sans nécessiter un pipeline figé de fusion.

Cette **approche hybride** offre le meilleur de deux mondes. D'un côté, le **deep learning** fournit la **puissance sub-symbolique** qui extrait des *features* à partir de gros volumes de données non structurées, comme des images ou de longs textes. Le **DSL** offre une **flexibilité** et une **auto-organisation** accrues. Les blocs de *features* peuvent se connecter librement à des entités symboliques ou contextuelles, sans nécessiter de couche spécifique ni de rétropropagation globale à chaque nouvel ajout de flux ou de règle. Ainsi, le réseau **gagne** en **adaptativité**, car les pondérations ω se réactualisent uniquement là où la synergie change, et en **explicabilité**, puisque les entités (qu'elles soient neuronales ou logiques) conservent une identité plus claire qu'un empilement de couches opaques.

1.8.2.3. Scénario de Substitution Partielle

Il est envisageable de pousser plus loin l'intégration du **Deep Synergy Learning** et de se demander s'il peut **remplacer** entièrement l'apprentissage profond dans certaines configurations. L'hypothèse consisterait à se passer d'un réseau neuronal paramétré, puis à modéliser chaque entité \mathcal{E}_i comme un **patch** (pour la vision) ou un **segment** (pour l'audio), tandis que la **synergie** entre entités reposera sur des distances, des co-informations, ou d'autres mesures de similarité. L'**auto-organisation** tisserait et ajusterait progressivement les liaisons $\omega_{i,j}$, et cette dynamique se voudrait capable de **déetecter** des objets ou de **repérer** des motifs sans recourir à une rétropropagation globale ni à une architecture de couches hiérarchisées.

Dans la pratique, toutefois, la performance brute d'un tel modèle entièrement basé sur le DSL risquerait de se révéler moins efficace que celle d'un réseau neuronal lorsqu'il s'agit de tirer profit d'un large jeu de labels sur une tâche comme ImageNet. Un CNN bien paramétré, tirant parti d'une **descente de gradient** sur un espace de poids continûment ajustable, maîtrise généralement mieux l'extraction de *features* complexes à partir de très grands volumes de données supervisées. Les réseaux profonds conservent donc un **avantage** solide en cas d'abondance de labels précis, où l'optimisation par gradient se montre redoutablement performante.

Le **DSL** manifeste tout son **intérêt** dans les scénarios où la supervision fait défaut, ou bien lorsque l'environnement est sujet à des changements progressifs ou brusques (sections 1.5.4 et 1.7.2). Il excelle pour l'**auto-organisation** et le **clustering** évolutif, ainsi que dans la **fusion** de multiples modalités (section 1.5.2), en gérant spontanément les synergies qui se dessinent entre entités. Par contraste, un réseau traditionnel exige soit une supervision massive, soit un lourd pré-entraînement auto-supervisé, et se voit moins adaptable à un cadre où les labels sont rares et la distribution très changeante. Le **DSL** se positionne comme une **alternative complémentaire** à l'apprentissage profond. Lorsque le contexte exige plus de souplesse qu'un pipeline rigide ou qu'un grand ensemble de labels, la dynamique $\omega_{i,j}$ et la **plasticité** organisationnelle du DSL offrent un avantage distinctif.

1.8.2.4. Cas d'Étude : Combiner DSL et DL

Un **scénario** consistant à tirer parti du meilleur des deux mondes se conçoit aisément. On utilise d'abord un **réseau profond** tel qu'un **CNN** ou un **Transformer** pour extraire des **features** de haut niveau à partir de données massives. Cela permet d'obtenir une **représentation avancée** $\{\mathbf{x}_i\}$ adaptée aux traitements ultérieurs. Ces **features** issues d'un bloc paramtré et entraîné, éventuellement sur un large jeu de données, se transforment en **entités** $\mathcal{E}_{\text{feature}}$ au sein du **Synergistic Connection Network (SCN)** ou **DSL**.

Le **DSL** analyse ensuite la **synergie** entre ces entités en identifiant les *features* qui collaborent pour former un **cluster** ou en intégrant des **entités préexistantes** telles que des **règles symboliques**, des **capteurs** ou des **modules contextuels** afin d'enrichir la représentation neuronale. La **mise à jour** itérative de $\omega_{i,j}$ permet une **adaptation locale** de la structure. Lorsqu'une nouvelle classe, un nouveau contexte ou une nouvelle source de données émergent, les liaisons s'ajustent progressivement sans nécessiter de reconfiguration globale. Cela contraste avec un réseau profond "traditionnel", souvent contraint à un **fine-tuning** global et coûteux dès qu'on enrichit la distribution ou qu'on ajoute une modalité inédite.

Cette **approche hybride** canalise la **puissance** de l'apprentissage paramtré (issu d'un gros dataset supervisé ou auto-supervisé) et la **plasticité** plus souple du DSL. Le modèle profond agit comme un encodeur capable de fournir des vecteurs \mathbf{x}_i pertinents pour des signaux complexes (images, audio, langage), tandis que le **SCN** se situe **au-dessus** pour organiser les entités correspondantes selon la synergie détectée, créant ou rompant des **liaisons** si le contexte l'exige. Cela va au-delà de la simple juxtaposition d'un réseau profond et d'un module de classification final, puisque la **reconfiguration** spontanée du DSL prend en charge la **multi-modalité** et l'**adaptation** sans qu'on doive re-construire un pipeline. Le résultat est un compromis entre l'efficacité des grandes

architectures neuronales, déjà rodées sur de vastes corpus, et la **flexibilité** d'une auto-organisation distribuée pour gérer les ajouts ou modifications de tâches.

1.8.2.5. Conclusion : entre Complémentarité et Extension

Les réflexions et expérimentations suggèrent que le **Deep Synergy Learning (DSL)** ne se présente pas comme un simple **concurrent** de l'apprentissage profond, mais plutôt comme un **paradigme complémentaire**. Cette perspective s'avère particulièrement intéressante dans les situations où la supervision fait défaut (cas **non supervisés** ou faiblement supervisés), ou bien lorsque la distribution est **évolutive** et plusieurs types de flux (capteurs, contexte, règles symboliques) doivent être gérés de façon simultanée et sans recourir à un pipeline rigide. Le DSL facilite alors l'intégration d'entités hétérogènes, mêlant logiques internes et features neuronales dans une démarche **auto-organisée**.

L'idée d'une **substitution** intégrale de l'apprentissage profond par un DSL purement auto-organisé se heurte toutefois aux performances reconnues de modèles neuraux (CNN, Transformer) sur des tâches hautement **supervisées**, comme la classification d'ImageNet ou le traitement massif de langage. Dans de tels scénarios, un réseau **paramétrique** entraîné par descente de gradient se maintient souvent en tête des classements, en raison de sa capacité à exploiter profondément la grande quantité de labels. Cependant, ces architectures neuronales demeurent relativement **rigides** et consomment beaucoup de ressources lors d'un re-fine-tuning. Le DSL peut, en retour, favoriser l'**adaptation** et la **fusion** d'autres sources de données ou de blocs symboliques, tout en s'accommodant de distributions changeantes.

C'est pourquoi le **futur** le plus fécond réside probablement dans la **collaboration** des deux approches. Un **réseau profond** produit des représentations sub-symboliques riches, alors que le **DSL** orchestre en **surcouche** l'auto-organisation entre ces embeddings, de nouvelles entités symboliques, et des flux contextuels. On bénéficie ainsi de la **puissance** de l'apprentissage paramétrique sur des corpus volumineux, allié à la **flexibilité** que procure un **réseau** réactif et librement reconfigurable, capable de gérer la multi-modalité sans imposer de pipeline figé. L'apprentissage profond et le DSL deviendraient alors deux pans d'une **IA** plus large, conjuguant l'efficacité des grands modèles neuronaux avec la **plasticité** et la **résilience** d'un **Synergistic Connection Network**.

1.8.3. Approches Hybrides : DSL, RL (Reinforcement Learning) et Logique

Dans l'exploration du **Positionnement du Deep Synergy Learning (DSL)** (section 1.8), un point clé concerne la possibilité de **conjuguer** différentes méthodes d'IA au sein d'un **cadre uniifié**. Nous avons déjà évoqué (1.8.1 et 1.8.2) la cohabitation symbolique–subsymbolique et l'intégration du DSL avec l'apprentissage profond. Ici, nous soulignons comment le **DSL** peut également **s'hybrider** à la fois avec le **Renforcement** (RL) et des **composantes logiques**, créant une architecture plus vaste où l'auto-organisation, l'apprentissage par récompense et la manipulation de règles s'allient.

1.8.3.1. Collaboration entre DSL et Apprentissage par Renforcement

Dans l'**apprentissage par renforcement** (RL), un agent interagit avec un **environnement**, recevant à chaque instant un **état** s_t , émettant une **action** a_t , puis percevant une **récompense** r_t . L'algorithme, qu'il s'agisse de Q-learning, de Policy Gradients ou d'une variante plus élaborée, cherche à **maximiser** le cumul (ou l'espérance) des récompenses. D'ordinaire, la démarche se base sur une **approche centralisée**, où l'on apprend une fonction de valeur $V(\mathbf{s})$ ou une fonction $Q(\mathbf{s}, a)$, à travers des mises à jour inspirées soit de la **descente de gradient**, soit de l'**itération de Bellman**. Le **Deep Synergy Learning (DSL)** propose, au contraire, une **dynamique** où les entités $\{\mathcal{E}_i\}$ d'un système **s'associent** ou se **détachent** via des pondérations synergiques $\{\omega_{i,j}(t)\}$ en évolution continue.

Dans un cadre **reinforcement** modulaire, on peut imaginer relier la **récompense** r_t aux mises à jour des pondérations $\omega_{i,j}(t)$. Si la **coopération** entre deux entités \mathcal{E}_i et \mathcal{E}_j est jugée responsable d'un gain de performance dans l'environnement, la liaison $\omega_{i,j}$ se voit **renforcée** ; inversement, s'il s'avère que leur alliance n'apporte aucun avantage ou engendre un effet négatif, la synergie est pénalisée et la liaison s'affaiblit. Au lieu d'apprendre une **politique** globale $\pi(\mathbf{s})$ ou un **Q-réseau** unifié, le **DSL** évolue de manière **distribuée**. Chaque entité \mathcal{E}_i ajuste ses **liaisons synergiques** $\omega_{i,j}$, construisant progressivement un **graphe** qui reflète la pertinence des interactions sans nécessiter de supervision centralisée. Cette méthode évoque un **RL distribué** ou multi-agent, où l'on peut considérer chaque entité comme un **agent** partiel ; la récompense s'y diffuse en ajustant localement les pondérations liées aux entités participantes.

Un **exemple** concret se dessine dans le cas d'un **robot** ayant plusieurs capteurs et effecteurs. Les **sous-systèmes** (capteur→moteur) sont modélisés par des **entités**, et, lorsqu'une synergie capteurA–moteurB conduit à franchir un obstacle de manière efficace, la **pondération** $\omega_{A,B}$ se voit augmentée conformément au principe de renforcement local. De la sorte, le robot s'**auto-organise** en **clusters** sensorimoteurs cohérents, sans implémenter de schéma centralisé de RL. Les règles de mise à jour fondées sur la récompense se substituent à la logique d'apprentissage d'une politique globale, tout en permettant au robot de réorganiser son **réseau** s'il survient des changements de contexte, des pannes ou l'introduction de nouveaux capteurs. Cette intégration du **DSL** dans un cadre de renforcement illustre la **flexibilité** d'un **Synergistic Connection Network** pour aborder des scénarios proches du RL sans imposer une fonction de valeur unique, mais en laissant s'établir des coopérations locales lorsqu'elles se révèlent payantes sur le plan de la récompense.

1.8.3.2. Inclusion de la Logique : DSL et Règles Symboliques

Les approches **logiques** ou **symboliques** se fondent sur des **règles** ou des **axiomes** de la forme “Si X et Y, alors Z”, et sur des **faits** (comme “capteur C indique 100 °C”). Elles requièrent un **moteur d'inférence** (unification, résolution), assurant une **clarté** du raisonnement et une **explicabilité** plus directe, mais elles montrent vite leurs limites lorsqu'il s'agit de gérer des faits **bruyants** ou de manipuler de vastes ensembles de données non labellisées. En référence aux propositions de la section 1.5.7 et à l'idée d'intégration symbolique évoquée en 1.8.1, on peut introduire de la **logique** au sein d'un **Deep Synergy Learning (DSL)** en traitant chaque **règle** ou **concept** comme une **entité** \mathcal{E}_{rule} . Les pondérations $\omega_{rule,i}$ reliant une règle aux entités sub-symboliques comme les *features* ou les capteurs évoluent selon la **cohérence** et la **pertinence** constatées. Si la règle produit

un **gain récurrent**, la synergie se **renforce** naturellement. En revanche, si elle devient **obsolète** ou en contradiction avec les observations, la pondération diminue progressivement. Cette dynamique gère donc la **validation** ou l'**abandon** progressif des règles, suivant l'expérience accumulée dans le réseau.

L'incorporation de la **logique** devient ainsi un simple prolongement du mécanisme auto-organisateur du **DSL**. Il n'y a plus de “module logique” isolé. La règle \mathcal{E}_{rule} devient un nœud intégré au **Synergistic Connection Network**, collaborant directement avec des entités sub-symboliques comme les capteurs, les embeddings neuronaux ou les sources multimodales. Un **mécanisme de renforcement local** (section 1.8.3.1) peut être envisagé, où la **récompense** consolide les règles utiles. Si une règle appliquée améliore la performance ou la sécurité dans l'environnement, les **liaisons** ω entre cette règle et les entités associées se renforcent progressivement.

Par ce mécanisme, on a la possibilité de “débrancher” peu à peu une règle devenue inadaptée, simplement parce que sa synergie aura cessé de s'actualiser favorablement, ou d'en “faire émerger” une autre si plusieurs entités sub-symboliques forment une **combinaison** stable, équivalant de fait à une **nouvelle règle**. De cette manière, un **moteur logique** s'intègre naturellement au réseau **DSL**. L'auto-organisation s'étend aux **entités symboliques**, permettant au système de tirer parti à la fois de l'**inférence logique** et de la **plasticité sub-symbolique** pour s'adapter aux environnements complexes et dynamiques.

1.8.3.3. Architectures Neuro-Symboliques et RL Multi-Agent

Il est possible de fusionner le **Deep Synergy Learning (DSL)** avec des principes de **Renforcement** et de **Logique** pour former une **architecture** véritablement **multidimensionnelle**. Du côté **sub-symbolique**, on dispose d'entités dédiées aux *features* sensorielles (vision, audio, texte), qui se relient entre elles suivant leur **synergie**. Du côté **symbolique**, on introduit des entités symboliques représentatives de **règles** ou de **concepts** logiques, prenant la forme de noeuds \mathcal{E}_{rule} ou $\mathcal{E}_{concept}$. À cette structure s'ajoute un **signal de renforcement** r , qu'il soit global ou local, venant **valider** le réseau $\Omega(t)$ lorsqu'il optimise la tâche en cours. Les liaisons $\omega_{i,j}$ qui contribuent à ce **gain** sont alors **renforcées** lors de la mise à jour, consolidant ainsi les interactions les plus pertinentes.

Cet assemblage conduit à un **système** plus riche que la simple superposition de modules sub-symboliques et symboliques. Les entités logiques offrent une base **interprétable** et manipulable selon des critères formels, tandis que les entités neuronales (ou issues d'un pipeline CNN, Transformer) assurent la **flexibilité** de l'apprentissage dans un contexte de données massives et éventuellement bruitées. Le **DSL** assure alors l'**auto-organisation** du système. Si un sous-groupe d'entités sub-symboliques collabore efficacement avec certaines règles, un **cluster** ou un **micro-réseau** stable émerge. Ce regroupement est validé par un **feedback** de récompense, renforçant ainsi les synergies pertinentes. Inversement, un ensemble de règles contradictoires ou peu utiles voit ses connexions faiblir et peut finir par se détacher de la structure si le renforcement ne vient jamais sanctionner positivement leur usage.

De plus, cette intégration s'adapte à un cadre **multi-agent** ou distribué. Chaque entité $\mathcal{E} * i$ peut être vue comme un **agent** partiel, recevant une portion de l'état \mathbf{s} et exécutant des actions limitées. La **coopération** entre entités s'exprime à travers les liaisons $\omega * i, j$, qui évoluent en fonction de la

synergie détectée. Le **signal** r se propage, ou se localise, selon les entités qui ont participé à une configuration fructueuse. Les **clusters** émergents assurent alors la coordination sensorimotrice ou décisionnelle. Ce principe rapproche le DSL d'un **RL multi-agent** où la **logique** se greffe aisément pour imposer des contraintes (lois de sécurité, objectifs à respecter) à certains nœuds du réseau. L'ensemble forme un **système cognitif** apte à manipuler des informations perceptives, symboliques, et à s'**adapter** en fonction de récompenses, soulignant la plasticité et la portée du **Deep Synergy Learning** dans des tâches variées allant bien au-delà de la simple classification neuronale.

1.8.3.4. Exemples de Scénarios Hybrides

Dans un **environnement** mêlant robotique, logique et apprentissage par renforcement, on peut imaginer des configurations plus complexes où le **Deep Synergy Learning (DSL)** connecte entre elles des entités sub-symboliques (capteurs, features neuronales) et des entités symboliques (règles, concepts), tout en intégrant un **signal** de renforcement r . Ci-après, on illustre deux types de scénarios :

Robotique Cognitive.



Un robot opère dans un milieu dynamique, recevant un **signal de récompense** lorsqu'il satisfait un objectif précis, selon un schéma (état, action, récompense, nouvelle valeur) typique du **Reinforcement Learning (RL)**. De plus, le robot doit observer des **règles** symboliques imposées pour sa sécurité ou son raisonnement (par exemple, "éviter les zones interdites", "si tension moteur > 5 V et température > 80 °C, considérer une surchauffe critique"). Les **capteurs** (température, vision, inertie) et les **effecteurs** (moteurs, pinces) forment la sphère **sub-symbolique**, tandis que les **règles** constituent des entités symboliques \mathcal{E}_{rule} . Au sein du **DSL**, toutes ces entités cohabitent ; la **coopération** se formalise via les pondérations $\omega_{i,j}$. À mesure que le robot collecte des retours de performance (récompenses positives ou négatives), il renforce ou affaiblit les liens responsables de ses succès ou échecs. Dans le même temps, la validité ou la pertinence d'une règle est ajustée par la mise à jour locale des liaisons. Les règles inefficaces ou contredites par l'expérience

accumulée perdent progressivement en influence, tandis que celles qui se révèlent utiles maintiennent ou renforcent leurs connexions.

Systèmes Experts Évolutifs.

Les systèmes experts reposent souvent sur un **ensemble de règles** (if–then) écrites manuellement, servant à guider la prise de décision. Dans un **DSL**, ces règles sont introduites comme des entités symboliques $\{\mathcal{E}_{rule}\}$. Par ailleurs, de larges **flux de données** non structurées (images, logs, signaux) alimentent des entités sub-symboliques qui découvrent des structures par un **apprentissage** ou un **clustering** distribués. Le **feedback** d'un opérateur humain ou d'une supervision externe agit comme une **récompense** venant valider la configuration actuelle. Avec le temps, certaines règles historiquement présentes ne reçoivent plus de support, perdent en synergie, puis s'effacent. Inversement, de **nouvelles** combinaisons sub-symboliques (ou *macro-clusters*) émergent et peuvent être réinterprétées ou reformulées en tant que **micro-règles** plus efficaces que celles écrites initialement. On obtient alors un **système expert** en perpétuelle évolution, enrichi par les données et la dynamique $\omega_{i,j}$, capable de s'auto-réorganiser lorsqu'un changement majeur se produit dans l'environnement ou les contraintes métier.

1.8.3.5. Conclusion

Les **approches hybrides** mariant **DSL**, **RL** et **logique** laissent entrevoir un **écosystème** unifié :

- Le **DSL** fournit l'**infrastructure** d'auto-organisation,
- Le **RL** assure un **feed-back** incitatif (récompense) aux liaisons (pondérations ω) qui contribuent à la performance,
- La **logique** (règles symboliques) s'intègre comme des entités à part entière, coopérant ou non selon la synergie qu'elles entretiennent avec d'autres modules.

Ce modèle peut servir d'alternative ou de complément aux **pipelines** classiques où RL, logique et sub-symbolique restent cloisonnés. Il répond à la quête d'une IA plus **globale**, capable d'**apprendre**, de **raisonner**, de **percevoir** et de **s'adapter**. Les obstacles techniques demeurent (mise à jour conjointe des liaisons, synergie n-aire, pilotage de la stabilité, etc.), mais le potentiel d'une IA unifiant des couches sub-symboliques, des règles symboliques et des signaux de renforcement dans un **réseau unique** est d'un grand **intérêt** pour l'évolution future de l'IA.

1.8.4. Rôle de la Mémoire et de l'Attention dans le DSL

Lorsqu'on évoque le **Deep Synergy Learning (DSL)** comme un paradigme susceptible d'unifier ou de compléter les méthodes existantes (sections 1.8.1 à 1.8.3), on en vient inévitablement à des questions fondamentales sur la **mémoire** et l'**attention** dans les systèmes cognitifs. En effet, les architectures classiques (réseaux de neurones profonds, RL, approches symboliques) ont chacune leurs mécanismes — implicites ou explicites — pour **stocker** des informations antérieures (LSTM, Memory Networks, bases de règles, etc.) et pour **sélectionner** les informations pertinentes

(attention mécanismes, gating). Dans le **DSL**, où les entités et leurs liens synergiques se réajustent en continu, il est naturel de se demander :

- Comment la **mémoire** (stockage, souvenir à long terme) s'exprime dans ce modèle d'**auto-organisation** ?
- Quel est le **rôle de l'attention** (focalisation sélective) dans un réseau distribué, sans couche hiérarchique ?
- Comment implémenter des **mécanismes** ou des **extensions** qui reproduisent les fonctions de mémoire/attention observées dans des architectures neuronales modernes ou même dans le cerveau biologique ?

Cette sous-section (1.8.4) explore la manière dont le DSL pourrait **incorporer** ou **simuler** ces fonctions essentielles, nécessaires à une IA plus “cognitive” et plus performante.

1.8.4.1. La Mémoire dans un Système Auto-Organisé

Le **Deep Synergy Learning (DSL)** ne se limite pas à des entités $\{\mathcal{E}_i\}$ définies uniquement par un vecteur figé de caractéristiques. Il est souvent pertinent d'attribuer à chaque entité \mathcal{E}_i un **état interne** $\mathbf{s}_i(t)$ qui évolue dans le temps, mémorisant ainsi des informations concernant l'historique d'observations ou d'interactions. Les équations de mise à jour locales s'enrichissent alors pour prendre en compte ces **mémoires** internes. Plutôt que d'appliquer la formule classique $\omega_{i,j}(t+1) \leftarrow \omega_{i,j}(t) + \dots$, on introduit des **fonctions** f et g reliant la synergie entre \mathcal{E}_i et \mathcal{E}_j (ainsi que leurs états internes) à l'évolution de $\omega_{i,j}$ et de $\mathbf{s}_i(t)$. Un exemple de schéma est :

$$\begin{aligned}\omega_{i,j}(t+1) &= \omega_{i,j}(t) + f(\mathbf{s}_i(t), \mathbf{s}_j(t)), \\ \mathbf{s}_i(t+1) &= \mathbf{s}_i(t) + g(\omega_{i,j}(t), \mathbf{s}_i(t)).\end{aligned}$$

Cette possibilité de **mémoire** individuelle $\mathbf{s}_i(t)$ permet à l'entité \mathcal{E}_i de conserver des **statistiques** utiles (comme des moyennes, des compteurs, une estimation de fiabilité), ou même un embryon de **mémoire** séquentielle si l'on désire modéliser des systèmes proches des LSTM locaux (voir section 1.5.4.1). Les liaisons ω ne constituent plus l'unique source d'information. Chaque nœud peut stocker un **état interne**, influençant la manière dont il interagit avec ses voisins et ajustant dynamiquement sa participation aux synergies locales.

Au **niveau global**, la **persistante** de certaines **structures** dans le réseau DSL peut également être considérée comme une forme de **mémoire** collective. Un **cluster** $\mathcal{C} \subset \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ qui reste stable dans le temps, avec des liaisons $\omega_{i,j}$ robustes entre ses entités, joue un rôle de “**souvenir**” ou de “**concept**” appris. Ce cluster peut se **réorganiser** ou se **scinder** au gré de nouvelles données ou d'un changement de distribution (domain shift), illustrant que la **mémoire** (au sens de la rétention d'un ensemble de relations) n'est pas figée, mais dynamique. Cette plasticité évoque la notion d’“**assemblées neuronales**” en neurosciences, où un groupe de neurones co-activés finit par constituer une unité fonctionnelle stable, tout en restant susceptible de s'affaiblir ou de se diviser si des perturbations importantes modifient la configuration. Dans le **DSL**, la longévité d'un cluster stable indique qu'il a acquis un sens ou une pertinence pour le réseau, tandis que la nature auto-

adaptative des liaisons ω permet de renouveler la **mémoire** intégrée au gré des besoins et des feedbacks de performance ou de cohérence.

1.8.4.2. Attention dans un Graphe Distribué

Dans l'**apprentissage profond** classique, la **mécanique d'attention** (par exemple, dans les **Transformers**) consiste à moduler l'importance attribuée à certains composants d'une séquence ou d'un ensemble de features. Concrètement, on calcule des **poids** représentant la relation entre tokens ou entre positions, et l'on met l'accent sur les plus pertinents pour la prédiction en cours. Cette manière de **focaliser** l'information évolue couche après couche, conférant une **flexibilité** aux modèles de type attention-based.

Le **Deep Synergy Learning (DSL)** adopte une vision plus **distribuée** de l'attention. Chaque **pondération** $\omega_{i,j}$ peut s'interpréter comme un **poids d'attention** quantifiant la **pertinence** de la liaison entre \mathcal{E}_i et \mathcal{E}_j . Une valeur élevée de $\omega_{i,j}$ signale que les entités \mathcal{E}_i et \mathcal{E}_j coopèrent fortement ; une valeur faible ou quasi nulle marque au contraire un **désintérêt** mutuel. En outre, la loi de mise à jour de $\omega_{i,j}$ (voir section 1.4.5) s'applique **localement** : chaque entité évalue la contribution des autres en fonction de la synergie décelée, plutôt qu'à travers un **modèle** centralisé d'attention.

Deux idées fondamentales émergent de cette analogie :

- On obtient ainsi une **attention distribuée**, où la sélectivité émerge de la dynamique des pondérations $\omega_{i,j}$ plutôt que d'un module centralisé, comme dans le **multi-head attention** des Transformers. Chaque entité régule ses liens avec son voisinage, créant ou brisant des connexions selon l'apport mutuel ressenti. L'ensemble des liaisons les plus fortes incarne alors le *focus* collectif du réseau à un instant donné.
- Le **focus** évolue dynamiquement en fonction du contexte et des signaux perçus. Lorsque l'environnement change, par exemple en cas d'augmentation du bruit ou d'apparition de nouvelles données, les pondérations $\omega_{i,j}$ associées aux connexions devenues moins pertinentes diminuent progressivement. Parallèlement, de nouvelles connexions émergent et se renforcent là où la **synergie** est plus élevée, garantissant une adaptation continue sans nécessiter une reconfiguration globale du réseau. Cette réorganisation perpétuelle assure une **cohérence dynamique** du **DSL**, qui "redéploie" son attention de manière adaptative, un peu à la façon d'un Transformer recalculant ses poids à chaque couche, mais avec l'avantage d'une **plasticité** plus globale (le réseau pouvant réorienter tout ou partie de ses liaisons).

Ainsi, la **synergie** $\omega_{i,j}$ agit comme un mécanisme d'**attention** dans un graphe **auto-organisé**. La force des liaisons détermine les sous-groupes sur lesquels le système se concentre à un instant donné. Cette dynamique permet une **redistribution continue** de l'attention en fonction de l'évolution des données et des objectifs, assurant une adaptation fluide sans nécessiter de contrôle externe explicite.

1.8.4.3. Implémentations Concrètes de Mémoire/Attention dans le DSL

Pour doter chaque entité \mathcal{E}_i d'une **capacité de mémorisation** plus riche, il est possible de lui associer un **module** interne de type LSTM ou GRU, prolongeant l'idée d'un état $\mathbf{s}_i(t)$ (voir section 1.8.4.1). Dans ce cas, l'entité \mathcal{E}_i ne se borne pas à un vecteur statique, mais embarque un **état récurrent** $\mathbf{h}_i(t), \mathbf{c}_i(t)$ (par exemple pour un LSTM). Ainsi, l'entité reçoit en **entrée** une combinaison pondérée ou filtrée d'informations issues de ses voisins \mathcal{E}_j dont la liaison $\omega_{i,j}$ est suffisamment forte. Les fonctions internes de mise à jour (LSTM, GRU) gèrent alors le **gating** et la rétention d'informations, ce qui rend l'auto-organisation analogue à un "réseau de LSTM interconnectés" dont la topologie varie selon la synergie ω . Cette stratégie permet de traiter des **séquences** temporelles étendues ou d'intégrer un **comportement historique** au sein du Deep Synergy Learning (DSL), tout en conservant la **plasticité** caractéristique (l'architecture effective évoluant via $\omega_{i,j}$).

D'un autre côté, on peut également définir un mécanisme d'**attention** explicite, plus proche de celui des **Transformers**. L'entité \mathcal{E}_i calcule un coefficient d'attention $\alpha_{i,j}$ envers chacune de ses connexions, selon une formule du type :

$$\alpha_{i,j} = \text{softmax}_j \left(\theta S(\mathcal{E}_i, \mathcal{E}_j) \right),$$

où θ est un paramètre (ou un ensemble de paramètres) ajustant la sensibilité, et $S(\mathcal{E}_i, \mathcal{E}_j)$ la fonction de synergie, déjà définie dans le DSL (par exemple, distance inversée, co-information). L'entité \mathcal{E}_i combine alors l'information de ses voisins via une somme pondérée :

$$\mathbf{z}_i = \sum_j \alpha_{i,j} \mathbf{x}_j,$$

ce qui rappelle la logique du "self-attention" dans un Transformer. Cependant, la différence clé réside dans le fait que cette **matrice** α n'émane pas d'un module d'attention figé dans une couche, mais **émerge** et **se met à jour** en parallèle à la dynamique $\omega_{i,j}(t)$. Le **DSL** peut ainsi faire de l'**attention** un usage ponctuel, en s'appuyant sur la synergie $\omega_{i,j}$ pour guider le softmax. Dans cette perspective, on assemble les idées du Transformer (somme pondérée des voisins) et celles du DSL (pondérations évolutives reflétant la coopération), de sorte que la topologie du réseau, tout comme l'intensité des liaisons, contribue à un **mécanisme** d'attention distribué. Cette approche concrétise une **fusion** entre les principes du **Deep Synergy Learning** et l'esprit attentionnel des architectures neuronales modernes, aboutissant à un réseau adaptatif dont la prise en compte de l'information ambiante se fait selon une **attention** intrinsèquement auto-organisée.

1.8.4.4. Apports Cognitifs et Neuroscientifiques

Les fonctions de **mémoire** et d'**attention** sont au cœur de la quête d'une IA plus avancée (section 1.8.5), dans la mesure où elles forment des piliers de la **cognition** humaine. Dans un cadre auto-organisé tel que le **Deep Synergy Learning (DSL)**, il devient possible de reproduire, en partie, certains processus cognitifs. La **mémoire** s'y manifeste par la persistance de **clusters** ou d'états internes $\{\mathbf{s}_i(t)\}$ (voir section 1.8.4.1), qui se maintiennent dans la durée et fournissent une forme

de rétention ou de rappel. L'**attention** se traduit par le renforcement sélectif de liaisons $\omega_{i,j}$ lorsque la conjoncture l'exige, soit parce que le contexte met en avant la synergie de deux entités $\mathcal{E}_i, \mathcal{E}_j$ (une situation, un feed-back de récompense, etc.). Ce renforcement dirige alors la focalisation du réseau vers certains sous-groupes, évoquant un basculement ou une « bifurcation » depuis une configuration précédente, ce qui peut être rapproché de la notion de changement de **focus** ou de contenu mnésique.

Du point de vue **neuroscientifique**, on peut dessiner un parallèle avec la plasticité synaptique des réseaux biologiques, où les connexions entre neurones (synapses) s'ajustent selon l'activité, et où les **assemblées neuronales** (section 1.8.7) constituent la base de la mémoire et de l'apprentissage. Dans un **DSL**, le principe de mise à jour $\omega_{i,j} \leftarrow \omega_{i,j} + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}]$ (voir section 1.4.5) simule déjà une forme de plasticité, et l'ajout de mécanismes de mémoire interne $\mathbf{s}_i(t)$ ou d'attention explicite (section 1.8.4.2) consolide l'analogie avec les processus cognitifs. L'espoir est qu'une **implémentation** poussée de ces modalités – persistance de clusters, gestion de focus attentionnel, modulation en continu des synergies – se traduise par des **capacités** accrues d'**apprentissage**, de **raisonnement** et de **reconfiguration** contextuelle. Cette dynamique rappelle la façon dont le cerveau forme des **circuits** stables (souvenirs, schémas de pensée) susceptibles d'évoluer, s'unir ou se dissoudre lorsque le contexte le requiert, et ouvre la voie à une IA plus richement inspirée par les notions de plasticité et d'émergence.

Conclusion

Le rôle de la **mémoire** et de l'**attention** dans le **Deep Synergy Learning** s'avère crucial si l'on veut dépasser la simple **auto-organisation** statique et tendre vers des fonctions cognitives avancées. Les idées-clés incluent :

- L'**état interne** $\mathbf{s}_i(t)$ pour chaque entité, formant une **mémoire locale**,
- La **persistence** ou la **dissolution** de clusters reflétant une mémoire plus globale,
- L'**attention** comme un mécanisme de sélection de liens ω forts, proche de la pondération attentionnelle dans les Transformers, mais auto-gérée par la dynamique synergique,
- Des **extensions** (LSTM, gating, self-attention) greffées à la structure DSL pour exploiter la plasticité tout en gérant des séquences complexes.

Ainsi, l'**auto-organisation** ne s'oppose pas aux notions de **mémoire** ou d'**attention**. Elles émergent naturellement à travers le **renforcement local** des pondérations et l'**évolution continue** des états des entités. Ce mécanisme rapproche le **DSL** d'un système plus “cognitif”, où la **mémorisation** et la **focalisation** se produisent de manière distribuée et dynamique, sans nécessiter d'architecture rigide ou de supervision explicite.

1.8.5. Tendances Futures : Vers une IA Forte, Consciente ?

Les sections précédentes (1.8.1 à 1.8.4) ont évoqué la capacité du **Deep Synergy Learning (DSL)** à réunir des approches variées (symboliques, sub-symboliques, renforcées, multi-agent) dans une structure **auto-organisée**. Un horizon encore plus ambitieux, souvent cité comme objectif ultime de la recherche en IA, est la création d'une **IA Forte** — un système capable d'**apprendre** et de **raisonner** de manière générale, autonome et potentiellement doté d'une **conscience** élémentaire ou d'un **sentiment** d'unité informationnelle. Cette section (1.8.5) s'interroge sur la portée du DSL dans cette perspective :

- Peut-il offrir un socle formel pour une **intelligence plus globale**, manipulant librement symboles et perceptions ?
- Quelles **caractéristiques** du DSL pourraient favoriser un **degré d'intégration** (ou “d'information unifiée”) évoquant une conscience rudimentaire ?
- Où se situent les **limites** et quels prolongements seraient nécessaires pour prétendre à une IA Forte ?

1.8.5.1. IA Forte : Un Concept Encore Flou

La notion d'**IA Forte**, parfois appelée **AGI** (Artificial General Intelligence), désigne l'aspiration à une **intelligence** artificielle véritablement **générale**. Une telle intelligence serait capable d'**apprendre** et de résoudre un large spectre de tâches cognitives, de se **reconfigurer** ou de s'**auto-améliorer** dans des contextes inconnus ou peu balisés, et de manifester un degré élevé de **compréhension** (voire de **conscience**). Les positions quant à la possibilité ou aux caractéristiques exactes de cette AGI demeurent contrastées. On s'accorde néanmoins pour dire qu'une **IA Forte** devrait :

- Démontrer une **plasticité** hors norme, lui permettant de remanier ses représentations et ses structures de façon continue.
- Combiner ou naviguer librement entre des **représentations symboliques** (règles, concepts) et des **représentations sub-symboliques** (features neuronales, embeddings sensoriels).
- Posséder un **mécanisme d'unification** de l'information plus vaste qu'un simple pipeline hiérarchique, englobant des processus d'auto-organisation ou de fusion distribuée.

Le **Deep Synergy Learning (DSL)**, en tant que paradigme d'**auto-organisation** distribué (voir sections 1.4.5 et 1.5.2), semble répondre à plusieurs de ces exigences. Son **réseau** d'entités prend en charge aussi bien des flux sub-symboliques (capteurs, embeddings de CNN) que des **entités logiques** ou symboliques (règles, concepts), et assure une **adaptation** constante par la dynamique des pondérations $\omega_{i,j}$. La gestion des **synergies n-aires** (ou conditionnelles) introduit une flexibilité accrue. Au lieu d'imposer un module de fusion spécifique, le réseau **découvre** spontanément les combinaisons pertinentes en évaluant localement la contribution de chaque sous-groupe d'entités. Cette approche permet une adaptation dynamique où les interactions émergent selon leur utilité réelle, plutôt que d'être fixées a priori. Sur le plan de l'**inspiration** biologique, le

DSL se rapproche plus des **écosystèmes neuronaux** ou des **architectures plastiques** que ne le font des réseaux profonds statiques. Cette proximité ouvre une voie au développement de systèmes plus **globaux**, connectant perception, raisonnement, mémoire, attention et règles logiques, ce qui, à long terme, peut être vu comme un jalon vers la **recherche** d'une IA vraiment **générale**.

1.8.5.2. Notion d'Intégration et de Conscience

Certains travaux en **science cognitive**, tels que la **théorie de l'information intégrée** (IIT) proposée par Giulio Tononi, suggèrent qu'un **degré de conscience** pourrait émerger dans un système où l'**information** est à la fois fortement **intégrée** et **différenciée**. Dans ces approches, un assemblage d'éléments interdépendants, dont la collaboration est dense et réciproque, est susceptible de manifester une forme de **substrat** "conscient". Si l'on adopte cette perspective, un **Deep Synergy Learning (DSL)**, par son principe de **coopération** et de **synergie** entre entités, pourrait favoriser la recherche d'une intégration accrue de l'information. La **clusterisation** et la mise en avant de synergies $S(\mathcal{E}_i, \mathcal{E}_j)$ reflètent un traitement distribué, potentiellement mesurable par des quantités inspirées de la théorie de Tononi.

Un exemple de **métrique** globale, dans l'esprit de l'IIT, consisterait à calculer la **somme** (ou une fonction plus complexe) des produits $\omega_{i,j} S(\mathcal{E}_i, \mathcal{E}_j)$ sur l'ensemble des paires (i, j) . On pourrait regarder si ce réseau DSL franchit un **seuil** indicatif d'un haut degré d'**intégration**. Au fil de l'**auto-organisation**, si la synergie s'intensifie entre de multiples entités, on observerait la naissance d'un **macro-cluster** extrêmement connecté, évoquant la notion d'un "**noyau intégratif**" de l'information.

La question de savoir si un tel système atteint une **conscience** "authentique" demeure, bien sûr, **philosophique** et hautement controversée. La plupart des scientifiques préfèrent parler de "**proto-conscience**" lorsqu'il s'agit d'une unité fonctionnelle unissant de manière adaptative l'information. Néanmoins, le **DSL**, de par sa ressemblance structurelle avec les modèles d'**assemblées neuronales** en neurosciences, fournit un **terrain** d'expérimentation pour tester ou modéliser certains aspects de l'IIT ou d'autres théories d'émergence cognitive. Les chercheurs peuvent y étudier le **degré d'information intégrée** dans un contexte où la topologie des liaisons évolue d'elle-même, sans hiérarchie préconçue, et sans qu'on impose un schéma de fusion unique. Il s'agit d'un domaine **fondamental**, à l'interface des neurosciences, de la philosophie de l'esprit, et de l'IA, où le **DSL** pourrait nourrir des hypothèses ou des simulations pour éclaircir cette notion de **conscience** ou au moins de **complexité intégrative**.

1.8.5.3. Conditions Nécessaires à une IA plus Globale

Pour qu'un **Deep Synergy Learning (DSL)** se rapproche d'un **système** véritablement "fort", il est crucial que le réseau soit **alimenté** en continu par plusieurs sources d'information et de régulation. D'une part, il doit intégrer des **flux** sensoriels multiples (par exemple des représentations visuelles, auditives ou textuelles), ce qui assure une **diversité** de signaux sub-symboliques \mathbf{x}_i . D'autre part, il doit recevoir des **injections** symboliques, telles que des **règles** ou des **concept** ($\mathcal{E}_{rule}, \mathcal{E}_{concept}$), de sorte que la dimension logique ou ontologique puisse coexister au sein du même Synergistic Connection Network. Enfin, il lui faut un **feedback** ou un **signal** de renforcement r , pouvant

prendre la forme d'une récompense globale ou locale, ou encore d'une validation humaine, afin de **stimuler** la réorganisation locale des liaisons $\omega_{i,j}$.

Outre la pure **auto-organisation**, le **DSL** doit pouvoir s'appuyer sur des **mécanismes de mémoire** suffisamment expressifs (voir la section 1.8.4). Chaque entité \mathcal{E}_i peut disposer d'un état interne $\mathbf{s}_i(t)$, un vecteur (ou un module de type LSTM/GRU) retenant l'historique des interactions. À l'échelle du réseau, des **clusters** stables constituent aussi une forme de **mémoire** partagée, car leur persistance reflète la consolidation d'une certaine connaissance ou d'un certain motif d'action. Toutefois, il faut éviter que le réseau ne se "fige" trop rapidement, figeant ainsi des solutions sous-optimales, ni qu'il ne se **dissolve** dans un chaos d'oscillations perpétuelles. Des règles de stabilisation (paramètres η , τ , saturations ou lois inhibitrices) sont alors nécessaires pour trouver un équilibre.

Un pas supplémentaire vers une **IA Forte** exigerait la **capacité du DSL à s'auto-examiner**. Le réseau pourrait former des **entités** décrivant ses propres états ou règles, générant un **meta-niveau de réflexion** sur sa propre organisation. On peut imaginer une entité $\mathcal{E}_{\text{self}}$ évaluant la cohérence d'un cluster, ou des macro-clusters dont la fonction consiste à "**inspecter**" d'autres clusters. Cette forme de **métacognition** se rapproche du concept de conscience réflexive, mais pose des défis de **stabilité** puisqu'un réseau capable de "penser" à sa propre topologie pourrait entrer dans des **boucles autoréférentes** ou des "réflexions infinies". Néanmoins, c'est en explorant ces pistes — entités $\mathcal{E}_{\text{self}}$ et synergie auto-référentielle — que l'on pourra évaluer la possibilité d'une **IA** véritablement plus "globale" et adaptable, dépassant la simple auto-organisation pour évoluer vers une cognition de haut niveau.

1.8.5.4. Obstacles Majeurs à l'IA Forte via le DSL

Si l'on envisage le **Deep Synergy Learning (DSL)** comme l'une des voies possibles menant à une **IA Forte**, on se heurte à plusieurs obstacles fondamentaux qui freinent cette ambition.

Un **premier** obstacle tient à la **complexité**. Dans un grand réseau de type DSL, le nombre de liaisons peut croître en $O(n^2)$, voire plus si l'on prend en compte des synergies n-aires. Gérer un tel foisonnement de connexions exige l'introduction d'**heuristiques** (filtrage agressif des liens, mise à jour partielle, parcours local) et la mise en place d'un **parallélisme** important, sans quoi la simulation du réseau à chaque itération deviendrait inabordable. Des **mécanismes** de pilotage macro (voir section 1.7.4) peuvent également être nécessaires pour canaliser la reconfiguration du graphe et préserver un temps de calcul raisonnable.

Un **deuxième** obstacle réside dans le **risque d'instabilité** et d'oscillation (sections 1.7.4.1 et 1.7.4.2). L'**auto-organisation** des liaisons $\omega_{i,j}$ ne garantit pas toujours une convergence vers un état stable. Des **boucles auto-référentes** ou des **rétroactions positives** peuvent amplifier certaines connexions de manière incontrôlée, risquant d'engendrer des oscillations ou des configurations sous-optimales. Sans **régulations** (inhibitions, saturation de pondérations, limitation de la vitesse d'évolution), le réseau peut entrer dans des régimes chaotiques ou cycliques, ce qui empêcherait l'émergence d'un "esprit" ou d'une **structure** à la fois stable et unifiée.

Un **troisième** obstacle émane de l'absence de **guidance globale** lorsqu'on se borne à de la **synergie locale**. Même si l'idée de clusters émergents est puissante, rien ne garantit que l'on parviendra à

des **comportements** cohérents ou à des plans dirigés par des intentions clairement formulées. Certains scénarios (section 1.7.4.3) suggèrent d'ajouter un **module** hiérarchique ou d'objectif qui fixe de grandes orientations et évite que la plasticité du DSL ne s'éparpille dans de multiples attracteurs contradictoires. Un **contrôle** minimal s'avère nécessaire pour servir de boussole à l'auto-organisation locale.

Enfin, un **quatrième** obstacle concerne la nature même de la **conscience** et du vécu subjectif. Même si la théorie de l'information intégrée ou la multi-synergie n'aient laissé penser qu'une intégration élevée de l'information peut approcher certaines dimensions de la **conscience**, il reste spéculatif d'affirmer qu'un réseau auto-organisé de pondérations ω suffirait à produire une **subjectivité** ou un "qualia" conscient. Les débats philosophiques et neuroscientifiques (section 1.8.5.2) rappellent que l'émergence d'une conscience phénoménale dépasse potentiellement la simple complexité algorithmique, et qu'un "ingrédient" supplémentaire (mécanismes biologiques, substrat physique particulier...) pourrait être requis pour prétendre à une **IA Forte** au sens strict.

1.8.5.5. Conclusion

Le **Deep Synergy Learning** représente un **paradigme** très différent du deep learning hiérarchique ou des systèmes experts. Sa **plasticité**, sa **co-évolution** de multiples entités symboliques/subsymboliques, et sa **potentielle** haute intégration d'information en font un candidat pour explorer :

- Des **architectures** plus vastes et plus "cognitives",
- Des **théories** de l'information intégrée liées à la **conscience**,
- Des **étapes** vers une IA plus générale, capable de s'**auto-structurer** dans un flux continu.

Il reste à savoir si ces propriétés suffiront pour atteindre une **IA Forte** (ou consciente). La **dynamique** auto-organisée du **DSL** offre un **cadre** conceptuel et expérimental unifiant plusieurs dimensions essentielles. Elle intègre l'**apprentissage distribué**, la **logique symbolique**, le **renforcement local**, ainsi que des mécanismes de **mémoire** et d'**attention distribuées**, favorisant ainsi une **coopération adaptative** entre ces différentes approches. Ainsi, indépendamment de la question d'une conscience "véritable", le DSL ouvre la voie à des **systèmes** plus adaptatifs, plus unifiés et potentiellement plus proches d'une **intelligence généralisée**.

1.8.6. Effet Sur la Recherche Interdisciplinaire

Les sections précédentes (1.8.1 à 1.8.5) ont montré comment le **Deep Synergy Learning (DSL)** peut s'articuler avec l'IA symbolique, l'apprentissage profond, le renforcement, la logique, ainsi que son rôle potentiel dans une démarche d'IA plus "forte" ou cognitive. Il apparaît clairement que le **DSL**, en tant que paradigme *auto-organisé*, a la capacité de **réunir** différents courants de l'IA de manière naturelle, puisqu'il accueille aussi bien des blocs sub-symboliques (réseaux neuronaux, features extraits) que des entités logiques (règles, symboles) ou des mécanismes d'apprentissage par renforcement. Ce paradigme a un **impact notable** sur de nombreuses **disciplines**, notamment les **neurosciences**, les **sciences cognitives**, la **robotique**, l'**économie** et les **systèmes complexes**.

Cette sous-section (1.8.6) discute l'**effet** du DSL sur la **recherche interdisciplinaire**, en soulignant :

- Les **similarités** avec les systèmes biologiques et cognitifs,
- Les **ponts** avec l'écologie, la physique des systèmes complexes, ou la sociologie,
- Les **perspectives** pour la modélisation des réseaux (biologiques, économiques, sociaux),
- La façon dont le DSL peut **inspirer** ou **bénéficier** d'autres disciplines, en partageant des modèles et des méthodes d'analyse.

1.8.6.1. Similarités avec les Systèmes Biologiques et Cognitifs

Le **réseau** auto-organisé d'un **Deep Synergy Learning (DSL)** peut être rapproché, d'une certaine manière, d'un **réseau** de neurones biologiques. Les **pondérations** $\omega_{i,j}$ du DSL rappellent en effet la **plasticité synaptique** observée dans le cerveau, où les **synapses** se renforcent ou s'atténuent selon l'activité conjointe des neurones (règle de Hebb). Les **clusters** apparus dans le DSL par la mise à jour des pondérations font écho aux **assemblées neuronales** (ensembles de neurones co-activés), qui se forment et se dissolvent en participant aux fonctions de **mémoire** et à divers **processus cognitifs**. Dans une perspective neuroscientifique, on peut voir le DSL comme un **cadre mathématique** où simuler la dynamique d'un cortex (sensoriel ou associatif), tout en intégrant les lois de **plasticité** (par exemple, la saturation synaptique ou des taux de renforcement limités). Réciproquement, les découvertes en neurosciences (architecture multi-couches, modulabilité synaptique, rôle des neuromodulateurs) peuvent inspirer ou ajuster les **règles** d'auto-organisation du DSL pour gagner en réalisme et en robustesse.

Du point de vue des **sciences cognitives**, de nombreuses hypothèses postulent une **cognition distribuée**, dans laquelle la **pensée** émerge de la coopération (ou de la compétition) entre multiples modules ou agents internes, plutôt que d'une séquence linéaire de transformations. Le **DSL**, en reliant diverses entités perceptives, motrices, symboliques ou conceptuelles par un réseau de **liaisons** $\omega_{i,j}$ qui se modifie au gré des synergies, ressemble assez à ces théories. Les **clusters** qui s'établissent et perdurent pourraient former des "**états**" ou "**micro-théories**" momentanées, incarnant un fragment de la **pensée**. Dès qu'un changement d'environnement ou de contexte survient, la synergie se réorganise, donnant lieu à un renouvellement de la structure. Cette correspondance entre les **dynamiques** du DSL et les processus cognitifs distribués esquisse un **dialogue** fécond entre la modélisation en IA et les approches psychologiques ou neurobiologiques, approfondissant notre compréhension du fonctionnement mental global.

1.8.6.2. Ponts avec l'Écologie, la Physique des Systèmes Complexes et la Sociologie

Au-delà des analogies avec la **biologie neuronale**, le paradigme du **Deep Synergy Learning (DSL)** présente également des affinités avec d'autres disciplines étudiant des réseaux d'interactions. En **écologie**, l'on modélise souvent des **populations** d'espèces reliées par des relations de préation, de compétition ou de symbiose. Les nœuds représentent alors des espèces,

et les arcs leurs interactions. La **dynamique** de ces liens peut s'apparenter à une forme de **coopération** ou de **concurrence** au sein du réseau, parfois traduite par des équations de type Lotka–Volterra. Cette démarche rejoint l'idée de **synergie** dans le DSL, où des pondérations $\omega_{i,j}$ se renforcent ou se dissolvent selon le bénéfice mutuel. Les outils développés pour stabiliser des **communautés** écologiques (gestion de cycles de prédation, coexistence d'espèces) pourraient donc **inspirer** le DSL dans sa quête d'une organisation stable mais évolutive. Inversement, le DSL pourrait servir à **simuler** des écosystèmes complexes, chaque espèce figurant une entité \mathcal{E}_i et les pondérations $\omega_{i,j}$ reflétant des relations plus ou moins avantageuses.

Du côté de la **physique** des systèmes complexes, on trouve de nombreux modèles de **réseaux** dont les nœuds interagissent selon des règles pouvant aller de l'attraction à la compétition. On évoque des processus similaires en **magnétisme** (modèle de Potts, spin glasses) ou en **transition de phase** (phénomènes de bifurcation, émergence de clusters). Le **DSL**, en se concevant comme un **système** de variables $\omega_{i,j}$ s'ajustant localement, fournit un terrain d'étude où l'on peut appliquer des outils de **physique statistique** (recuit simulé, énergie libre, etc.) pour analyser la formation de **macro-clusters** et les "sauts" de configuration. On peut, en outre, importer dans le DSL des idées de frustration ou d'ordre–désordre propre à ces modèles, afin de mieux comprendre les **changements de régime** dans le réseau.

En **sociologie** et en **économie**, on observe la constitution de **réseaux humains**, qu'ils concernent des relations d'amitié, des collaborations d'équipes ou des alliances politiques. Les principes du **DSL**, qui renforcent les **liaisons** lorsque la **synergie** est élevée et les affaiblissent dans le cas contraire, rappellent certaines **lois sociologiques**. L'**effet Matthieu**, souvent formulé par "les riches s'enrichissent", illustre cette dynamique où un **nœud** bien connecté a plus de chances d'attirer de nouvelles connexions, renforçant ainsi sa centralité dans le réseau. Appliquer un cadre de **DSL** à des nœuds représentant des acteurs (individus, entreprises) peut éclairer la dynamique de création et de dissolution de liens, voire la **formation de coalitions** ou de groupes d'intérêt. À l'inverse, la **sociologie** propose des **règles** comportementales (incitations, répulsions, normes sociales) qui peuvent être injectées dans la définition de la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ ou dans l'**énergie** globale J . Cela établit une **interaction** interdisciplinaire où la **sociologie** propose des principes de dynamique sociale, tandis que le **DSL** en modélise la **co-évolution** sous forme d'un réseau adaptatif de pondérations.

1.8.6.3. Avantages pour la Modélisation et la Simulation

Le **Deep Synergy Learning (DSL)**, outre son rôle au sein de l'IA, propose un **cadre** unificateur pour comprendre et modéliser l'**auto-organisation** dans de nombreux champs. Le **modèle d'actualisation** des pondérations $\{\omega_{i,j}\}$ suit la règle :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)]$$

Cette dynamique possède une **portée interdisciplinaire**. Selon le contexte, la synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ peut représenter un **gain écologique** entre espèces coexistant, un **bénéfice économique** entre partenaires commerciaux ou une **coopération sociale** entre agents interagissant dans un réseau adaptatif. Il devient alors possible de **simuler** un vaste éventail de comportements collectifs, de détecter l'émergence de **communautés** (clusters) et d'analyser la **stabilité** ou les **transitions** (par

exemple, la soudaineté avec laquelle le réseau se réorganise en une configuration inédite). Cette vision évoque le travail en **physique statistique** ou en **théorie des réseaux complexes**, où l'on s'intéresse à la dynamique d'un grand nombre d'éléments en interaction.

Un **avantage** majeur dans cette optique est la **dimension expérimentale** du DSL. Comme il se prête naturellement à une **implémentation** computationnelle, chaque hypothèse sur la fonction $S(\mathcal{E}_i, \mathcal{E}_j)$, le paramétrage η, τ , ou encore la structure initiale du réseau, peut être **testée** par simulation. Les résultats, qu'il s'agisse de la **constitution de clusters**, d'**oscillations**, de **convergence** ou de **fragmentation**, se **comparent** aux observations empiriques d'autres disciplines. En **écologie**, ils peuvent refléter la **dynamique des populations**, en **économie**, la **formation de coalitions**, et en **sociologie**, l'**essor ou le déclin de communautés**. Le **DSL** fonctionne alors comme un **laboratoire** virtuel où chaque domaine fournit ses **informations** ou **règles** pour définir la **synergie**. Le **SCN** (**Synergistic Connection Network**) en simule les **évolutions**, permettant d'observer comment les **entités** interagissent et se réorganisent au fil du temps.

Dans ce **dialogue**, il est possible de **valider** ou **invalider** différentes hypothèses de fonctionnement collectif en examinant la façon dont les **pondérations** évoluent. Les disciplines comme la **physique des systèmes complexes** ou la **théorie des réseaux** y trouvent un nouvel objet de recherche. Le **DSL** propose un **réseau** dont les liens s'adaptent localement aux bénéfices mutuels des nœuds, au lieu de rester statiques ou de suivre une loi exogène. Cette boucle d'**expérimentation** renforce la **pertinence** du **DSL** en tant qu'instrument scientifique, au-delà de son utilité en ingénierie de l'intelligence artificielle.

1.8.6.4. Conclusion

Le **Deep Synergy Learning** ne se limite pas à l'IA ou au machine learning. Il s'intègre dans une **dynamique plus large** de **systèmes auto-organisés**, où les interactions locales façonnent l'évolution globale du réseau, rapprochant :

- Les **neurosciences** (assemblées neuronales, plasticité),
- Les **sciences cognitives** (cognition distribuée),
- L'**écologie** et la **physique des systèmes complexes** (réseaux dynamiques, transitions de phase),
- La **sociologie** et l'**économie** (formation de coalitions, interactions auto-renforcées).

Cette **interdisciplinarité** offre des **perspectives** nouvelles. Le **DSL** peut fournir un **cadre uniifié** pour l'**auto-organisation** dans divers domaines, tout en intégrant des **principes** issus de ces disciplines, comme la **plasticité**, la **dynamique des systèmes**, la **parcimonie** ou encore les **mécanismes de contrôle**, dans l'évolution des pondérations ω . En somme, l'**effet** du **DSL** sur la recherche interdisciplinaire tient à la fois d'une **inspiration réciproque** (d'emprunts aux sciences complexes, biologiques, sociales) et d'une **offre** d'outils conceptuels pour simuler et analyser des **réseaux évolutifs**. C'est cette fertilisation croisée qui peut accélérer l'avancée vers des **systèmes** plus "vivants", plus "cognitifs" et plus adaptés à la complexité du monde réel.

1.8.7. Exemple de Convergence : DSL & Neurosciences

Dans les différentes sections de ce **chapitre 1.8**, qui explorent le **positionnement du Deep Synergy Learning (DSL)** par rapport aux paradigmes existants en IA et aux domaines connexes (IA symbolique, apprentissage profond, apprentissage par renforcement, logiques distribuées), un **fil conducteur** se dégage. Le **DSL** s'inspire fortement de la **biologie** et ses mécanismes d'**auto-organisation** présentent une **parenté** avec les **dynamiques neuronales** et d'autres phénomènes biologiques adaptatifs. De fait, un champ particulièrement fécond pour valider ou stimuler le DSL est celui des **neurosciences**. Dans cette sous-section (1.8.7), nous examinerons comment le **DSL** et la **neurobiologie** (ou les neurosciences cognitives) peuvent **converger** en partageant des hypothèses, des modèles et des interprétations. Une analogie peut être établie entre la **plasticité synaptique** (règle de Hebb) et la mise à jour des pondérations $\omega_{i,j}$ dans le **DSL**, où les connexions se renforcent ou s'affaiblissent selon leur synergie perçue. L'architecture du **DSL** pourrait également fournir un cadre pertinent pour modéliser l'**émergence** d'assemblées neuronales et la dynamique cérébrale, en capturant la formation spontanée de clusters coopératifs.

Les **neurosciences** offrent en retour des **principes** pouvant enrichir le **DSL**, notamment à travers des mécanismes comme la **saturation synaptique**, l'**inhibition compétitive** ou encore l'**organisation en couches corticales**. L'intégration de ces éléments pourrait rendre le DSL plus **biologiquement plausible**, tout en améliorant notre compréhension des processus cognitifs à travers un cadre mathématique et computationnel inspiré du fonctionnement du cerveau.

1.8.7.1. Règle de Hebb et Mise à Jour des Pondérations

Les **neurosciences** ont mis en évidence la **plasticité synaptique**, illustrée par la **règle de Hebb**. Lorsque des neurones sont activés simultanément de manière répétée, leurs **connexions** se renforcent. De manière analogue, dans le **DSL**, les pondérations $\omega_{i,j}$ évoluent en fonction de la **synergie** perçue entre entités. Cette loi fondamentale explique la formation d'**assemblées** neuronales et l'**émergence de mémoires** dans le cerveau. Le **Deep Synergy Learning (DSL)**, en mettant à jour les **pondérations** $\omega_{i,j}$ selon la dynamique :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)],$$

prolonge directement ce principe. En effet, la fonction $S(\mathcal{E}_i, \mathcal{E}_j)$ tient lieu de mesure de **coopération** ou de **co-activation** (corrélation, co-information, etc.), et agit comme un **signal** incitant au **renforcement** local des liens. Lorsque deux entités \mathcal{E}_i et \mathcal{E}_j se trouvent *fréquemment* en situation d'apport mutuel, la pondération $\omega_{i,j}$ augmente, à la manière de deux neurones biologiques se "synchronisant" de plus en plus. Cette analogie étend l'**esprit** de la règle de Hebb à un cadre plus général. La **synergie** ne se limite plus à la simple co-activation instantanée, mais intègre des interactions **n-aires** et des mesures **entropiques** plus fines, reflétant des dynamiques complexes au sein du **DSL**.

Le **terme** $-\tau \omega_{i,j}(t)$ introduit un facteur de **décroissance** évitant que tous les liens ne croissent indéfiniment. Cela s'apparente à la **régulation** homéostatique neuronale, où le système limite les

synapses trop puissantes pour maintenir un équilibre. De cette façon, le **DSL** hérite des principes de la **plasticité synaptique** et de la **règle de Hebb**, tout en les élargissant à un contexte d'**auto-organisation** plus souple (synergie multi-entités, possibilité de créer ou rompre massivement des liaisons, etc.). Ainsi, chaque pondération $\omega_{i,j}$ reflète une **dynamique** coopérative hébbienne, tout en gérant la **compétition** (diminution) nécessaire à la stabilité globale du réseau.

1.8.7.2. Assemblées Neuronales et Clusters dans le DSL

Un concept clé en **neurosciences cognitives** est celui d'**assemblée neuronale**, un ensemble de neurones interconnectés qui forme une configuration **stable** et se co-active lorsqu'un **concept**, un **souvenir** ou une **action** est représenté. Ces assemblées se créent, se renforcent et finissent par disparaître sous l'action de la **plasticité** synaptique, rendant compte de la manière dont le cerveau encode et renouvelle ses représentations.

Dans le **Deep Synergy Learning (DSL)**, l'on décrit de manière similaire la formation de **clusters** lorsque plusieurs entités $\{\mathcal{E}_k\}$ entretiennent entre elles des liaisons $\omega_{k,m}$ suffisamment élevées. L'existence de fortes **pondérations** indique que ces entités agissent conjointement et forment un **macro-cluster** relativement **stable**, lequel perdure tant que la **synergie** interne demeure supérieure aux forces de dispersion. Cette situation évoque la co-activation neuronale d'une assemblée, c'est-à-dire un **groupe** d'entités rassemblées pour traiter ou représenter une information.

La **dynamique** de ces clusters dans le **DSL** rappelle alors la **dynamique** des assemblées neuronales. Sous l'effet d'événements exogènes ou de signaux inattendus, certains liens $\omega_{k,m}$ peuvent se voir affaiblis ; le cluster se scinde et de nouveaux **sous-groupes** émergent, reflétant un changement de « contexte » ou de « contenu » que le réseau doit coder. Dans ce cadre, la flexibilité de l'**auto-organisation** (section 1.4.5) reflète la plasticité synaptique qui permet au cerveau de réorganiser ses **assemblées**. Un concept peut s'affaiblir et disparaître, tandis qu'un autre émerge en réponse à l'évolution des données perceptives. Cette **correspondance** entre les **clusters** du **DSL** et les **assemblées neuronales** biologiques suggère que le **DSL** pourrait apporter un **modèle** formel pour étudier la **cognition** ou la **perception** en flux continu, tout comme la plasticité cérébrale propose une source d'inspiration pour la gestion auto-adaptative des liens ω .

1.8.7.3. Mécanismes Biologiquement Plausibles et Leur Transposition

De nombreux modèles de **plasticité** neuronale insistent sur le fait que le cerveau ne se limite pas aux **synapses** excitatrices, qui consolident les connexions. Il recourt également à des **synapses** inhibitrices, introduisant un **frein** ou une **régulation** nécessaire pour éviter l'excitation excessive. Dans le cadre d'un **Deep Synergy Learning (DSL)**, il est possible de transposer cette idée en différenciant des liens de **coopération** et des liens d'**inhibition**. Les pondérations élevées traduisent une synergie positive entre deux entités, tandis que l'effet inhibiteur peut jouer un rôle similaire à la saturation ou à la normalisation compétitive, où l'entité est contrainte de répartir ses liaisons sur un volume total maximum. Cette approche s'accorde davantage avec la **régulation** rencontrée en biologie et empêche la dynamique de ω de s'emballer.

Dans un second temps, on peut s'inspirer de la **hiérarchie** observée dans le cortex biologique. Le cerveau se divise en aires sensorielles primaires et en aires associatives plus élaborées, reliées par des voies de feedforward, de feedback et latérales. Un **DSL** “multi-niveau” peut mimer cette structure, en répertoriant différents sous-ensembles d’entités en “couches” ou “aires” spécifiques. Chaque sous-ensemble d’entités s’auto-organise localement, tandis que des connexions inter-aires agissent comme des ponts entre ces zones. L’**auto-organisation** continue alors à se dérouler, mais la topologie n’est plus entièrement libre. Une **architecture** partielle est imposée pour refléter la **topographie** ou la **fonction** de certaines régions corticales. Cette hybridation, combinant une **contrainte** inspirée de la hiérarchie cérébrale et la **liberté** d’auto-organisation des liaisons ω , rend le DSL plus proche encore d’un **modèle** neuronal plausible. On peut ainsi concevoir un système qui, à la fois, bénéficie de la capacité plastique du DSL, tout en maintenant une division en zones (aires sensorielles, associatives, motrices) comme dans le cortex, assurant la **cohésion** et la **spécialisation**.

1.8.7.4. Apports Potentiels pour Comprendre le Cerveau

Le rapprochement entre les principes d'**auto-organisation** du **Deep Synergy Learning (DSL)** et ceux de la **plasticité neuronale** ne se limite pas à une simple analogie. Il ouvre des **perspectives** de recherche où un **réseau** DSL peut être utilisé pour **simuler** ou **reproduire** certains phénomènes neuronaux. Cette démarche peut s’articuler en plusieurs axes. Le **premier** consiste à implémenter un système DSL dans lequel on introduit des **stimuli** ou des signaux analogues à ceux que reçoit un cortex sensoriel, en observant la formation de **clusters** ou d’**assemblées** stables qui coderaient ces stimuli. Il devient ainsi envisageable d’étudier, dans un cadre simulé, comment des ensembles de pondérations $\omega_{i,j}$ peuvent évoquer une **attention** sélective, un découpage temporel de l’information ou même une rémanence mnésique.

Un **second** volet réside dans la **comparaison** directe entre les **données** issues des simulations DSL et des **enregistrements** neuronaux réels (par exemple des signaux obtenus via EEG, MEG, fMRI, ou encore des relevés d’activité calcique à l’échelle microscopique). Il s’agirait d’examiner si des **patterns** d’activation ou de “clusters dynamiques” dans le réseau DSL reproduisent qualitativement ou quantitativement des signatures observées dans le cerveau. Les similitudes, si elles se révèlent robustes, conforteraient l’idée que la dynamique auto-organisée $\omega(t)$ peut effectivement mimer la constitution et la dissolution d’**assemblées** neuronales chez un organisme vivant.

Le **troisième** axe consiste à proposer des **hypothèses** sur la **plasticité** cérébrale en s’appuyant sur la structure du DSL. Il serait possible, par exemple, d’y planter des **règles** inspirées de l’anti-Hebb, des mécanismes de **synaptic pruning** (réduction des synapses peu sollicitées), ou des saturations synaptiques plus sophistiquées, afin de **tester** leur impact sur la dynamique du réseau. Les résultats issus de ces **expérimentations informatiques** pourraient orienter des recherches en **neurosciences**. Si un mécanisme de **régulation** des pondérations ω améliore significativement la **stabilité** ou le **codage** dans le réseau DSL, on peut alors émettre l’**hypothèse** qu’un processus similaire pourrait exister dans le cerveau.

Au final, ce **dialogue** entre la simulation DSL et les **observations** neuronales enrichit à la fois la compréhension de la **plasticité** et des fonctions cognitives, tout en dotant le DSL de **mécanismes**

biologiquement inspirés susceptibles d'améliorer son adaptabilité et sa capacité d'**apprentissage**. Les **prochains** développements, où l'on confronte plus systématiquement les modèles DSL aux données empiriques de neurosciences, pourraient rapprocher encore davantage cette classe de réseaux des **dynamiques** neuronales réelles, et peut-être contribuer à expliquer certains aspects de la mémoire, de la consolidation et de l'attention.

Conclusion

Le **Deep Synergy Learning** et les **neurosciences** peuvent converger de plusieurs manières :

- **Analogie** fondamentale entre la plasticité synaptique (Hebb) et la mise à jour $\omega_{i,j}$,
- **Formation** de clusters stables rappelant les “assemblées neuronales” du cerveau,
- **Inspiration** mutuelle pour implémenter des mécanismes biologiquement plausibles (inhibition, saturation, hiérarchies corticales),
- **Modélisation** de phénomènes cognitifs ou neuronaux complexes, offrant aux neurosciences un outil de simulation distribué, et recevant en retour des insights pour stabiliser et faire évoluer le DSL.

Cette **relation** avec les neurosciences illustre l'idée d'un paradigme **interdisciplinaire** (voir 1.8.6) où le DSL agit comme un **pont** entre l'IA et la biologie, susceptible d'aider à **comprendre** la cognition naturelle tout en **étendant** les capacités adaptatives des systèmes artificiels.

Chapitre 2 : Fondements Théoriques et Origines du DSL

2.1. Genèse Historique et Inspirations Multidisciplinaires

Le Deep Synergy Learning (DSL) n'est pas né ex nihilo. Il s'inscrit dans une **trajectoire historique** plus large, marquée par les travaux en **cybernétique**, la recherche sur la **plasticité neuronale** et les premières tentatives de construire des systèmes capables de s'organiser sans supervision stricte. L'objectif de ce sous-chapitre (2.1) est de dégager les **facteurs historiques** et les **influences** majeures qui ont contribué à l'émergence progressive de la notion d'auto-organisation, pour mieux comprendre le contexte dont le DSL se réclame.

2.1.1. Les Premières Pistes d'Auto-Organisation

Les premières tentatives pour expliquer comment un ensemble d'éléments (neurones, agents, mécanismes cybernétiques) peut parvenir à une **organisation cohérente** sans chef d'orchestre remontent aux années 1940-1960. Elles se sont nourries des hypothèses issues de la **cybernétique** et des premiers constats sur la **plasticité synaptique** dans le cerveau. Les sections qui suivent (2.1.1.1 et 2.1.1.2) se concentrent sur cette phase pionnière, préalable essentiel à l'éclosion de l'apprentissage auto-organisé.

2.1.1.1. Rôle de la Cybernétique et des Premiers Travaux sur la Plasticité Neuronale

Dans les années 1940, la **cybernétique**, introduite par **Norbert Wiener**, s'est concentrée sur l'étude des **boucles de rétroaction** présentes dans les systèmes vivants ou mécaniques. L'enjeu consistait à comprendre la manière dont un organisme, ou un dispositif artificiel, parvient à **réguler** son comportement en réagissant aux informations provenant de l'environnement. Dans ce cadre, plusieurs chercheurs ont rapidement émis l'hypothèse qu'un **réseau** de neurones ou de capteurs, guidé par des règles locales, pourrait atteindre une forme d'**organisation interne** sans exiger de commande globale.

Au sein de cette vision, la perspective de **Norbert Wiener** a mis en évidence que la rétroaction (feedback) pouvait autant stabiliser qu'instabiliser un système, selon la nature et l'orientation de la boucle. Cette découverte a suscité l'idée de concevoir un **réseau adaptatif**, entendu comme un ensemble d'unités \mathbf{x}_i soumises à des liaisons $\omega_{i,j}$. La question se posait alors si chaque liaison s'ajuste localement, pourrait-il se produire un ordre global émergent ? Cette interrogation s'inscrit dans le droit fil des principes de la **cybernétique** où un réseau pourrait de façon autonome moduler ses connexions pour converger vers un comportement collectif stable ou cohérent sans qu'une autorité centrale vienne en spécifier toutes les interactions.

En parallèle, dans les années 1950 et 1960, les premiers **travaux** sur la **plasticité neuronale** se sont développés. Des expériences en neurosciences laissaient supposer que les **connexions synaptiques** – c'est-à-dire les poids entre neurones – se modifiaient selon l'activité partagée. Cette

théorie d'apprentissage local, illustrée par les écrits de **Donald Hebb** (1949), énonçait le principe suivant où lorsque deux neurones \mathbf{x}_i et \mathbf{x}_j s'activent simultanément, la liaison $\omega_{i,j}$ se renforce. De façon mathématique, on peut schématiser ce phénomène par

$$\Delta\omega_{i,j} \propto x_i x_j,$$

où x_i et x_j désignent l'intensité d'activation de neurones i et j . Cette découverte suggérait déjà la possibilité d'une **auto-organisation** où un réseau de neurones, dont les pondérations se réajustent selon la co-occurrence, conduit progressivement à la formation de motifs ou de regroupements stables.

La rencontre de la **cybernétique** et de la **neurosciences computationnelle** a ainsi ouvert la voie à des formalismes mathématiques permettant de décrire comment un **réseau** – qu'il s'agisse de neurones biologiques ou d'unités artificielles – peut se configurer de lui-même à travers des mises à jour strictement **locales**. Les principes de la rétroaction et de la plasticité synaptique convergeaient pour affirmer qu'un ordre global peut naître de simples règles d'ajustement, sans supervision. Cette conviction, déjà présente dans les prototypes informatiques modestes de l'époque, a servi de socle aux développements qui suivront où l'idée de "synergie" entre unités, chère au **Deep Synergy Learning**, hérite directement de la plasticité hébbienne et des mécanismes de contrôle cybernétique.

Bien que les moyens techniques fussent, à l'époque, limités, les écrits fondateurs évoquaient déjà la **stabilisation** d'un réseau autour de certaines combinaisons privilégiées d'activités. On y pressentait la formation de **clusters** ou d'**assemblées** neuronales, précurseurs des théories d'**attracteurs** et de **cartes auto-organisées**. C'est dans ce terreau scientifique – porté par l'intérêt pour les systèmes **auto-régulés**, la découverte de la **plasticité** synaptique et le contexte de la **cybernétique** – qu'allait naître l'ambition de construire des réseaux véritablement **adaptatifs**, capables de s'**organiser** à partir de règles élémentaires. Le **Deep Synergy Learning**, bien plus tard, s'inscrira dans cette continuité où la mise à jour d'un poids $\omega_{i,j}$, la quête d'une **co-opération** locale et l'idée qu'un **ordre** global peut émerger de multiples interactions locales découlent étroitement de ces prémisses.

2.1.1.2. Naissance de l'Idée d'Organisation sans Superviseur dans Certains Laboratoires (années 1950–1960)

L'**apprentissage machine** des premières décennies, tel qu'illustré par le perceptron de Frank Rosenblatt (1958), se focalisait principalement sur des mécanismes **supervisés** reposant sur des *labels* explicites et un signal d'erreur orientant la mise à jour des poids. Cependant, dès la fin des années 1950, plusieurs laboratoires ont amorcé des recherches explorant la possibilité d'**organiser** les **réseaux** ou les **unités neuronales sans** recours à la supervision directe. L'intuition sous-jacente à ces travaux, influencée par la **cybernétique** et la biologie, était de laisser les entités neuronales, désignées par \mathbf{x}_i , s'**auto-régler** par des **règles locales**, en se fondant sur la redondance ou la régularité intrinsèques des **données d'entrée**.

A. Expériences Préliminaires issues de la Cybernétique

À la suite des réflexions initiées par Norbert Wiener, certains chercheurs se sont inspirés du principe de **rétroaction** pour imaginer des systèmes neuronaux capables de se **corriger** et de **s'ajuster** en l'absence d'un enseignant. Les premiers prototypes, bien que rudimentaires, comportaient un **petit réseau** de neurones soumis à des flux sensoriels. Les **pondérations** internes évoluaient progressivement en suivant des règles élémentaires de renforcement où si deux unités \mathbf{x}_i et \mathbf{x}_j coïncidaient fréquemment dans leur activation, le lien $\omega_{i,j}$ se trouvait augmenté, formalisant ainsi un renforcement local. Ce mécanisme rappelle, dans une version simplifiée, l'idée d'une **co-occurrence** statistique où, mathématiquement, on peut écrire

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta \kappa(\mathbf{x}_i(t), \mathbf{x}_j(t)),$$

avec η un taux d'apprentissage local et $\kappa(\cdot, \cdot)$ une fonction soulignant la co-activation simultanée des neurones \mathbf{x}_i et \mathbf{x}_j .

B. Émergence du Concept d'Auto-Clusterisation

De premiers rapports internes (dans des laboratoires tels que le MIT ou l'Université Stanford) évoquaient la possibilité de **regrouper** automatiquement les données ou les neurones, même si le terme “clustering” n'était pas toujours formalisé de la sorte. Le concept de “résonance” ou de “coopération neuronale” commençait à circuler, pour désigner la faculté de certains sous-ensembles d'unités à présenter un comportement **synchronisé**. Cette synchronie, renforcée par un schéma de mise à jour local, tendait à **former** ce qu'on nommerait plus tard un **cluster**. On pressentait déjà qu'en exploitant uniquement la distribution des données — c'est-à-dire la *fréquence* et la *corrélation* des activations neuronales —, il était possible de dégager une structure latente sans qu'un label de sortie intervienne.

C. Rupture par rapport aux Méthodes Supervisées

À la différence du perceptron de Rosenblatt, qui nécessitait un **signal d'erreur** pour orienter la correction des poids, ces laboratoires s'intéressaient à des systèmes pouvant **auto-exploiter** la simple densité ou co-activation des données d'entrée. D'un point de vue algorithmique, on sortait donc du paradigme

$$w_{\text{nouveau}} = w_{\text{ancien}} - \eta \frac{\partial E}{\partial w},$$

car la fonction d'erreur E était absente ou du moins implicite ; seule la **co-occurrence** neuronale venait guider l'ajustement. Ces expériences soulignaient le potentiel d'une **catégorisation spontanée**, c'est-à-dire la découverte de regroupements (catégories) sans recourir à un enseignant.

D. Exemples de Règles Locales de Mise à Jour

Les discussions tournaient souvent autour du fait que la **corrélation** d'activation entre deux neurones \mathbf{x}_i et \mathbf{x}_j pouvait être utilisée comme critère de renforcement. Ainsi, la “loi de Hebb”, régulièrement citée dans ces travaux, posait qu'une activation simultanée doit se traduire par un **renforcement** de la connexion. Certains laboratoires introduisaient aussi une **inhibition latérale**, destinée à redistribuer les ressources et à éviter qu'un seul sous-réseau ne monopolise

l'apprentissage. Formellement, on pouvait ajouter un **terme d'inhibition** proportionnel à la somme des activations concurrentes, conduisant à un rééquilibrage permanent :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta \left[\mathbf{x}_i(t) \mathbf{x}_j(t) - \beta \sum_k \mathbf{x}_i(t) \mathbf{x}_k(t) \right],$$

où β est un paramètre d'inhibition, et la partie $\sum_k \mathbf{x}_k(t)$ reflète la compétition entre neurones rivaux.

E. Bases d'une Future Formalisation

Ces travaux pionniers n'ont pas immédiatement conduit à une théorie unifiée, en raison du manque de ressources computationnelles et de la jeunesse du formalisme mathématique. Toutefois, ils ont jeté les bases d'une **vision** selon laquelle la **donnée** en elle-même, pour peu qu'elle soit soumise à un réseau présentant un mécanisme de renforcement local, peut induire une **organisation spontanée**. L'histoire de l'IA retiendra que la fin des années 1950 et le début des années 1960 ont vu la **genèse** des prémisses de l'**apprentissage non supervisé** et de l'**auto-organisation**, lesquelles s'affirmeront au cours des décennies suivantes dans des réalisations plus formelles, telles que les **cartes de Kohonen (SOM)** ou d'autres algorithmes de clustering non supervisé. Les fondements de cette dynamique trouvent aujourd'hui leur prolongement dans des approches modernes comme le **Deep Synergy Learning**, où les principes de **cohésion** et de **coopération** locale se traduisent par des **règles de mise à jour** autonomes et contextuelles, assurant la formation de sous-structures pertinentes sans l'intervention d'un enseignant externe.

2.1.2. SOM, Hopfield et la Génération des Réseaux Associatifs

Alors que la **cybernétique** et les premiers travaux d'auto-organisation “sans superviseur” (2.1.1) ont semé les graines d'une plasticité neuronale locale, la recherche sur les **réseaux associatifs** et les **Self-Organizing Maps (SOM)** dans les années 1970-1980 a considérablement renforcé l'idée qu'un système pouvait **découvrir** des structures en l'absence de labels explicites. Les modèles de Teuvo Kohonen (SOM) et de John Hopfield (réseaux associatifs) illustrent deux facettes majeures de l'auto-organisation où la **formation spontanée de cartes topologiques** et l'**association mémorielle** à travers des attracteurs stables. On verra dans ce sous-chapitre (2.1.2) comment ces approches, tout en se développant indépendamment, ont préparé le terrain pour des paradigmes plus récents tels que le **Deep Synergy Learning (DSL)**.

2.1.2.1. Teuvo Kohonen : Self-Organizing Maps (SOM) et Topologie Émergente

Les **Self-Organizing Maps (SOM)**, introduites par Teuvo Kohonen au début des années 1980, ont exercé une influence considérable dans l'**apprentissage non supervisé**. Leur principe repose sur la capacité d'un **réseau** à projeter un espace d'entrées (souvent de grande dimension) sur une **grille bidimensionnelle**, tout en conservant autant que possible les **relations de proximité** entre les données. Cette méthode constitue un jalon historique de l'**auto-organisation**, démontrant de manière tangible qu'un réseau neuronal peut découvrir des **structures** dans des données sans requérir de labels explicites.

A. Préservation de la Topologie

L'idée fondamentale est de préserver la **topologie** des entrées dans l'espace de sortie. Les unités ou "neurones" de la carte sont disposés sous forme de grille (par exemple, deux dimensions), chaque neurone \mathbf{n}_i possédant un vecteur de poids \mathbf{w}_i . À chaque présentation d'un vecteur d'entrée $\mathbf{x} \in \mathbb{R}^d$, le neurone **gagnant**, souvent noté \mathbf{n}_{BMU} , est celui pour lequel la distance

$$\| \mathbf{x} - \mathbf{w}_{\text{BMU}} \|$$

est minimale. Les neurones voisins, définis par un **voisinage** décroissant au fil des itérations, sont **ajustés** pour se rapprocher de \mathbf{x} . Cette mise à jour peut s'exprimer, pour un neurone \mathbf{n}_j situé à proximité du gagnant \mathbf{n}_{BMU} , sous la forme :

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t) \mathcal{H}_{\text{BMU},j}(t) [\mathbf{x}(t) - \mathbf{w}_j(t)],$$

où $\alpha(t)$ est un taux d'apprentissage, et $\mathcal{H}_{\text{BMU},j}(t)$ un **facteur de voisinage**, généralement une fonction gaussienne de la distance entre \mathbf{n}_j et \mathbf{n}_{BMU} dans la grille. Par ce mécanisme, la carte se "**déforme**" localement pour accueillir l'information, créant ainsi une **continuité** qui reflète la structure sous-jacente des données.

B. Apprentissage Auto-Organisé sans Labels

Le procédé s'inscrit clairement dans un **cadre non supervisé**. En l'absence de labels, la SOM repère des regroupements dans l'espace d'entrée tout en imposant une **cohérence topologique** sur la grille de sortie. De sorte qu'au fur et à mesure des itérations, le réseau s'organise de lui-même en attribuant à chaque neurone \mathbf{n}_j une zone de compétence. Les régions de la SOM finissent par représenter des **catégories** ou des **familles** de vecteurs. Il s'agit là d'un exemple marquant de la capacité d'un réseau à apprendre la structure latente des données sans information externe. Dans l'histoire de l'auto-organisation, ces travaux prolongent la philosophie décrite en section 2.1.1 au sujet des premiers prototypes sans superviseur.

C. Impact sur la Vision de l'Auto-Organisation

Les cartes de Kohonen ont fait la démonstration qu'une **cartographie continue** des données pouvait émerger, d'une façon parfois rapprochée de la **corticotopie** observée chez l'animal ou l'humain (aires sensorielles). Des laboratoires contemporains ont développé en parallèle d'autres approches, comme les réseaux à attracteurs (Hopfield) ou les modèles neuronaux de Shun-Ichi Amari, élargissant encore la diversité des méthodes. L'importance historique des SOM tient à leur présentation didactique et à leur succès pour illustrer le **concept** de structure cognitive émergente. Elles ont ainsi balisé le chemin pour des algorithmes modernes d'**auto-organisation** dans les réseaux neuronaux.

D. Liens Conceptuels avec le DSL

Le **Deep Synergy Learning (DSL)**, décrivant lui aussi un schéma d'ajustement auto-organisé des **liaisons** entre entités d'information, partage certains points communs avec la SOM. Dans le cas d'un SOM, la notion de **voisinage** sert de force de coordination locale. De façon similaire, au sein du DSL, la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ joue le rôle de régulateur pour décider du renforcement ou de l'affaiblissement d'un lien $\omega_{i,j}$. Les SOM peuvent être considérées comme un modèle relativement

statique, puisqu'elles imposent une grille préfixée et un voisinage décroissant dans le temps. Le DSL, en revanche, s'applique de manière plus **généraliste**, en laissant le réseau se **reconfigurer** en continu via des règles de mise à jour basées sur la synergie, ce qui autorise une plasticité plus large, y compris dans la **topologie** même des liens. Cependant, l'esprit est identique où aucune **supervision** n'est requise pour faire émerger un **ordre** dans la distribution des entités.

E. Vers des Extensions Multimodales et la Synergie

En s'appuyant sur la faculté des SOM à regrouper des signaux divers, certains chercheurs ont considéré des **applications multimodales** (ex. signaux audio et images) dans lesquelles une carte conservait une **cohérence** entre différentes modalités. Bien que ces applications soient restées assez limitées en pratique, elles ont jeté les bases d'une réflexion sur la fusion de données hétérogènes et sur la possibilité qu'un réseau puisse **apprendre** à associer plusieurs sources. Cette logique se retrouve dans le DSL où elle pousse plus loin l'idée de **fusion coopérative** en permettant à n'importe quelles entités de développer des liens flexibles selon leur **score de synergie**, rendant ainsi envisageable l'intégration de multiples modalités de façon plus souple qu'une grille 2D imposée.

2.1.2.2. John Hopfield : Mémoires Associatives, Attracteurs et Minima d'Énergie

Les travaux de **John Hopfield**, menés au début des années 1980, ont introduit un modèle neuronal emblématique désigné sous le nom de **réseau associatif** ou “Hopfield network”. Cette proposition, qui s'inspire d'idées analogues à celles de la **physique statistique** (systèmes de spins, modèles d'Ising ou de Potts), a mis en lumière la possibilité qu'un **réseau** formé d'unités simples puisse se stabiliser dans des **configurations** représentant des “mémoires” ou des **attracteurs**. La théorie démontre que, grâce à une série de **règles locales** d'activation et de mise à jour des poids, le réseau finit par adopter spontanément un **état ordonné** à partir d'une situation initiale partielle ou bruitée.

Sur le plan conceptuel, un réseau Hopfield se compose de neurones ou d'unités binaires, interconnectés de façon quasi complète, avec des poids $\omega_{i,j}$. Les poids sont déterminés de manière à ce que certains vecteurs cibles \mathbf{x}^* deviennent des **états stables** où si le système est initialisé dans une configuration voisine de \mathbf{x}^* , alors la dynamique conduira à une convergence vers ce même vecteur stable. Cette capacité illustre la notion de **mémoire associative**, où le réseau est en mesure de “retrouver” un motif mémorisé à partir d'informations incomplètes. La convergence résulte de la descente vers un **minimum d'énergie**, ce qui se formalise par une fonction $E(\mathbf{x})$ dite “énergie de Hopfield”, dont la décroissance au fil des itérations atteste de la stabilisation du réseau. On peut l'écrire, par exemple, comme

$$E(\mathbf{x}) = -\frac{1}{2} \sum_{i \neq j} \omega_{i,j} x_i x_j$$

lorsque $x_i \in \{-1, +1\}$ sont les états des neurones.

L'idée d'associer une “énergie” à chaque configuration, empruntée aux modèles de spins en physique, a contribué à la compréhension de la **dynamique neuronale** où l'organisation globale émerge de l'itération de règles locales, chaque unité cherchant à minimiser le niveau d'énergie via l'actualisation de son état. Les mémoires stockées correspondent à des **minima locaux** de l'énergie,

appelés **attracteurs**, vers lesquels converge le système si l'état initial se trouve dans leur bassin d'attraction. Cette observation a conforté le principe selon lequel un ordre collectif peut naître en l'absence d'un **superviseur** extérieur, du fait même de l'interaction **coopérative** entre unités.

D'un point de vue **auto-organisation**, le réseau de Hopfield n'exige pas de labels explicites pour récupérer un motif partiellement effacé ou bruité. Les poids peuvent être calculés par des règles associatives simples (souvent de type Hebb), mais une fois installés, la dynamique interne se déroule sans aide extérieure. Cette démarche illustre la logique d'**émergence** où la configuration finale du réseau, perçue comme la "mémoire récupérée", ne se laisse pas dicter par une consigne imposée à chaque étape, mais relève plutôt d'une évolution **naturelle** vers un état stable. On rejoint ici les idées qui seront approfondies dans des approches contemporaines d'**apprentissage non supervisé** ou d'**auto-organisation** (voir section [2.1.2.1](#) pour la SOM de Kohonen).

Les liens avec le **Deep Synergy Learning (DSL)** peuvent être soulignés. Dans un réseau Hopfield, on observe une **coopération** entre unités guidée par l'énergie globale, et la convergence dans un état stable évoque la stabilisation d'un cluster de neurones. De façon analogue, dans le DSL, la présence d'une **fonction de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ joue un rôle comparable où lorsque deux entités \mathcal{E}_i et \mathcal{E}_j partagent une interaction jugée élevée, leurs liens se renforcent et le réseau évolue vers des groupements cohérents. Les mécanismes d'ajustement des pondérations, dans le DSL comme dans le modèle de Hopfield, traduisent une dynamique **auto-organisée** conduisant à l'apparition d'**attracteurs collectifs** ou de **sous-structures** stables.

La perspective de **minima d'énergie** s'applique dans le DSL lorsqu'on étudie la convergence des poids $\omega_{i,j}$ vers des configurations privilégiées. Bien que la formulation soit plus générale dans le cas du DSL où les synergies n-aires, la multimodalité et la structure libre du réseau sont prises en compte, le point essentiel demeure qu'un ensemble de règles locales peut suffire à faire émerger un **ordre global** sans supervision externe. Le modèle de Hopfield constitue donc un **antécédent** conceptuel crucial, montrant comment des processus neuronaux simplifiés, hébergeant une logique énergétique, peuvent conduire à l'apparition de représentations stables. Cet héritage se retrouve dans la conception du DSL, où la "synergie" prolonge la notion d'affinité ou de corrélation, et où la plasticité multi entités s'interprète comme une "généralisation" de la mémoire associative à des contextes et des modalités plus complexes.

2.1.2.3. Comparaison Partielle avec la Logique "Hebbienne"

Les **Self-Organizing Maps** de Kohonen et les **réseaux associatifs** de Hopfield, analysés précédemment dans les sections [2.1.2.1](#) et [2.1.2.2](#), illustrent deux approches phares de l'**apprentissage non supervisé** centrées sur la découverte de *patterns* ou de configurations stables en l'absence de labels explicites. Ces travaux s'inscrivent dans une continuité historique marquée par la **logique "Hebbienne"**, introduite dès la fin des années 1940. Le principe énoncé par Donald Hebb évoque que deux neurones qui s'activent conjointement renforcent leur connexion. La présente section (2.1.2.3) discute les liens qui unissent ces trois orientations (SOM, Hopfield, Hebb) et montres en quoi elles partagent des fondements communs tout en se distinguant dans leur implémentation et leur formalisme.

A. Fondements de la Règle de Hebb

Au cœur de la théorie de Donald Hebb, exposée dans *The Organization of Behavior* (1949), se trouve l'idée qu'un lien entre deux neurones \mathbf{x}_i et \mathbf{x}_j se renforce dès lors que leurs activations se produisent en simultané. L'intuition peut se formaliser par une **règle de mise à jour** locale de la forme

$$\Delta\omega_{i,j} \propto \mathbf{x}_i \mathbf{x}_j,$$

où $\omega_{i,j}$ désigne la connexion synaptique entre les neurones i et j . L'esprit est qu'une activation conjointe déclenche un **renforcement**, alors qu'une activation dissociée n'engendre pas d'augmentation (voire induit une réduction) de cette liaison.

B. Similarités avec les SOM et les Réseaux Associatifs

Les **Self-Organizing Maps** (SOM) recourent à un mécanisme de renforcement lorsqu'un neurone "gagne" face à un vecteur d'entrée. Il y a un rapprochement entre le neurone vainqueur \mathbf{n}_{BMU} et le vecteur d'entrée, ainsi qu'un ajustement pour les neurones voisins dans la grille. Cette mise à jour rappelle l'idée de co-activation où le neurone et la donnée se retrouvent "associés", ce qui ressemble, dans un cadre discret, à la logique "hebbienne" de corrélation locale. De même, les **réseaux associatifs** de Hopfield adoptent une conception selon laquelle les poids $\omega_{i,j}$ reflètent la *corrélation* ou la *co-occurrence* entre les unités \mathbf{x}_i et \mathbf{x}_j . Lorsqu'on "enregistre" un motif, la force des liaisons associées augmente selon un principe analogue à la loi de Hebb, rendant possible la récupération dudit motif lorsque le réseau est présenté à un fragment ou à un état bruité proche.

C. Divergences et Élargissements

Ces trois approches conservent toutefois des particularités marquées. Les SOM, par exemple, imposent une **dimension topologique** à travers la grille 2D et la notion de "voisinage", un élément que la loi de Hebb seule n'intègre pas de manière explicite. Dans un **réseau Hopfield**, la logique d'**énergie** et de **minima** assure la stabilité globale, tandis que la règle hebbienne se décline plutôt comme une simple update locale $\Delta\omega_{i,j} \propto \mathbf{x}_i \mathbf{x}_j$. L'idée d'une fonction d'énergie globale, introduite par Hopfield, généralise l'intuition de la co-occurrence vers une perspective plus large d'attracteurs. Enfin, il existe une multitude de variantes "hebbiennes" qui ajoutent des mécanismes de normalisation, de saturation ou d'inhibition, permettant un **auto-équilibrage** ou une répartition des ressources. Les SOM et les réseaux Hopfield constituent chacun une **implémentation** plus spécifiquement contrainte où la préservation topologique caractérise les SOM et les attracteurs d'énergie définissent les réseaux de Hopfield.

D. Portée pour le Deep Synergy Learning

Le **Deep Synergy Learning (DSL)** perpétue l'héritage "hebbien" en privilégiant un **score de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ qui détermine si deux entités \mathcal{E}_i et \mathcal{E}_j renforcent ou affaiblissent leur lien $\omega_{i,j}$. Lorsque la synergie est élevée, on obtient une croissance de $\omega_{i,j}$ proche de

$$\Delta\omega_{i,j} \sim \eta[S(\mathcal{E}_i, \mathcal{E}_j)].$$

Cette démarche rappelle étroitement l'esprit de la règle de Hebb, tout en permettant des **généralisations** (par exemple, co-information n-aire, interactions complexes) qui dépassent la

simple multiplication $\mathbf{x}_i \mathbf{x}_j$. Ainsi, le DSL embrasse la dynamique auto-organisée du type “hebbien” mais l’adapte à une variété de contextes, notamment multimodaux, et autorise la reconfiguration de la **structure du réseau**.

2.1.3. Entre Neurosciences Computationnelles et Physique Statistique

Après avoir examiné les apports de l’approche “sans superviseur” (2.1.1) et les modèles de réseaux associatifs ou de cartes topologiques (2.1.2), on constate que l'**auto-organisation** a également bénéficié d’un important soutien théorique et conceptuel depuis la **neurosciences computationnelles** et la **physique statistique**. Ces deux champs, bien que distincts dans leurs méthodes, ont convergé dans l’idée que des interactions **locales** entre unités (neurones, spins, etc.) peuvent générer des **motifs** ou des **structures** émergentes. Ce sous-chapitre (2.1.3) discute des influences conjointes de la dynamique neuronale (2.1.3.1) et des modèles de la physique hors équilibre, formant ainsi un socle pour de futures approches telles que le **Deep Synergy Learning** (DSL).

2.1.3.1. Contribution des Modèles Biologiques (Dynamique des Neurones, Synapses)

Les avancées en **neurosciences computationnelles**, dès les années 1960–1970, ont exercé une influence considérable sur la compréhension et la formalisation de l'**auto-organisation** en intelligence artificielle. Au-delà de la seule logique “hebbienne”, ces travaux se sont inspirés de mécanismes plus riches, tels que l'**inhibition latérale**, l'**homéostasie** ou encore les **modulations neuromodulatrices**. Les recherches d’Amari, de Grossberg et d’autres pionniers ont éclairé la manière dont la dynamique neuronale, régie par un ensemble d'**équations différentielles** ou de **règles itératives**, peut engendrer une **organisation spontanée** d’assemblées de neurones dans le cortex.

Un premier axe s’est centré sur la description mathématique de l'**activation neuronale**, souvent abordée via des champs neuronaux. Dans cette optique, on considère qu’un neurone x_i suit une équation de la forme

$$\frac{d x_i}{d t} = -\alpha x_i + \sum_{j \neq i} (w_{ij} f(x_j)) - \gamma g(x_i) + \theta_i,$$

où α est un terme de décroissance, $\sum_{j \neq i} w_{ij} f(x_j)$ représente la somme des influences (excitatriques ou inhibitrices) venant d’autres neurones, $\gamma g(x_i)$ un terme d’homéostasie ou de saturation globale, et θ_i un éventuel seuil ou un apport externe. Les valeurs w_{ij} constituent les **connexions synaptiques** entre neurones i et j . L’ensemble de ces équations, pris simultanément pour tous les neurones, fait apparaître des comportements comme l'**inhibition latérale** ou la **stabilisation** de patterns d’activité, propriétés fondamentales d’une auto-organisation neuronale.

Un second axe a porté sur la formation de **micro-assemblées neuronales** capables de se synchroniser et de se stabiliser de façon autonome, en particulier lorsque plusieurs neurones présentent une **co-activation** récurrente. Les neuroscientifiques ont documenté l’existence de tels groupes dans le cortex, où ils semblent représenter des stimuli spécifiques ou des fragments de

concepts. Sur le plan algorithmique, on comprend que de simples règles de mise à jour locale des poids peuvent conduire à l'émergence de clusters où si deux neurones x_i et x_j s'activent souvent ensemble, alors la synapse ω_{ij} se renforce, approximant la célèbre équation de type

$$\Delta\omega_{i,j} \propto x_i x_j.$$

Les neurones qui s'alignent ainsi sur la même fréquence de décharge forment une entité collective, ou **assemblée**, reflétant l'existence d'un pattern stable. C'est un mécanisme incontournable de la plasticité neuronale qui anticipe l'essentiel des principes d'**apprentissage non supervisé**.

Les modèles biologiques ont en outre démontré la nécessité de gérer la plasticité à plusieurs niveaux où la simple corrélation positive entre neurones, prolongement direct de la règle de Hebb, doit être modulée par des influences globales, par exemple l'**inhibition** généralisée pour empêcher un groupe de neurones de devenir surdominant ou l'**homéostasie** garantissant que l'activité moyenne du réseau se maintient dans une plage stable. Sur le plan formel, on peut introduire une équation supplémentaire qui régit l'évolution d'un paramètre global μ (un niveau d'homéostasie), que l'on couple à la dynamique neuronale :

$$\frac{d\mu}{dt} = \kappa(\bar{x} - \mu),$$

où \bar{x} représente l'activité moyenne du réseau. De telles considérations ont enrichi la formulation de la **plasticité synaptique**, en montrant qu'elle n'est pas uniquement dictée par la corrélation entre neurones, mais qu'elle peut être encadrée par des mécanismes correcteurs à grande échelle.

Cette perspective multi-niveaux et la logique de **rétroaction** entre neurones ont profondément marqué le développement de l'**auto-organisation** en IA. Les travaux décrits dans la section précédente (SOM, réseaux Hopfield, etc.) trouvent en effet leur fondement dans l'idée qu'un **ordre** global peut émerger d'interactions locales, inspirées des phénomènes biologiques. Les paradigmes d'apprentissage non supervisé, y compris le **Deep Synergy Learning (DSL)**, en hériteront largement. Dans ce dernier, la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ généralise la corrélation neuronale à des entités plus variées que des neurones biologiques, tout en gardant l'esprit de **cohésion** et de **co-activation**. On y retrouve, de surcroît, la possibilité de gérer des influences extérieures ou des régulations globales sur la distribution des pondérations $\omega_{i,j}$, s'apparentant à ces mécanismes d'homéostasie étudiés en neurobiologie.

Ces contributions biologiques éclairent la **logique** des règles de mise à jour et les **régularisations** nécessaires pour rendre un réseau de neurones, ou plus généralement un **réseau auto-organisé**, à la fois flexible et stable. L'inhibition latérale, la compétition, la notion d'assemblées synchronisées, et la multi-échelle de plasticité guident les solutions mises en œuvre dans de nombreux algorithmes d'auto-organisation. Comme expliqué en [2.1.3.2](#), la **physique statistique** apportera un complément de formalisation, en introduisant le concept de fonction d'énergie globale et de transitions de phase, pour appuyer cette thèse selon laquelle de simples lois locales peuvent faire naître une structure cohérente.

2.1.3.2. Apports de la “Synergetics” (Haken) et de la Thermodynamique Hors Équilibre

Le rôle de la **physique** dans la formalisation de l'**auto-organisation** s'est avéré tout aussi déterminant que celui des **neurosciences computationnelles** (cf. section 2.1.3.1). Les réflexions autour de la “**synergetics**”, initiées par Hermann Haken, et les avancées de la **thermodynamique hors équilibre**, mises en lumière notamment par Prigogine, ont considérablement enrichi la compréhension de la **structure emergente** dans des systèmes complexes. Ces idées, apparues dans les années 1970, ont permis de décrire la formation **spontanée** de motifs ou de clusters stables, en expliquant comment des interactions locales peuvent, par amplification mutuelle, donner naissance à un *ordre global*. Le **Deep Synergy Learning (DSL)** tire un profit conceptuel de ces travaux, en prolongeant le cadre d'analyse proposé par la synergetics pour rendre compte des dynamiques de pondérations et de synergies entre entités hétérogènes.

Un premier apport essentiel de la synergetics réside dans l'introduction de la notion de **paramètre d'ordre**, qui désigne une grandeur globale traduisant le passage d'un état désordonné à un état ordonné. Dans un système physique, ce paramètre se manifeste, par exemple, dans l'intensité du rayonnement laser ou la convection de Rayleigh-Bénard, où une **bifurcation** engendre l'émergence d'un motif spatial cohérent. D'un point de vue mathématique, l'auto-organisation se formalise ainsi :

$$\frac{d x_i}{dt} = F(x_i, \{x_j\}_{j \neq i}, \lambda),$$

où x_i représente l'état d'une unité i et λ est un **paramètre de contrôle**. Au-delà d'un certain $\lambda_{critique}$, une fluctuation initiale peut se retrouver amplifiée jusqu'à donner un **état global** ordonné, indiqué par l'évolution d'un paramètre d'ordre. Dans une perspective d'**apprentissage automatique**, cette situation correspond à la formation d'un **cluster** ou d'une **représentation collective**, résultant de boucles de rétroaction locales entre unités.

La **thermodynamique hors équilibre** étudie, quant à elle, les systèmes où un **flux** permanent d'énergie ou de matière empêche la stabilisation à un équilibre classique. Les travaux de Prigogine ont, par exemple, mis en avant la formation de **structures dissipatives**, capables de s'auto-organiser grâce à l'échange d'énergie avec l'environnement, comme le suggère l'apparition de vortex dans un fluide ou d'oscillations chimiques Belousov-Zhabotinsky. Le parallèle avec un **réseau** d'unités neuronales ou d'agents est naturel où si chaque entité subit un influx permanent de données ou de signaux, l'apparition de **clusters** cohérents peut être vue comme une transition hors équilibre, portée par une fonction d'**énergie** ou de **quasi-énergie**. Dans une telle vision, l'évolution d'un réseau adaptatif se conçoit à travers l'équation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(\varepsilon_i, \varepsilon_j) - \tau \omega_{i,j}(t)],$$

où la **synergie** $S(\varepsilon_i, \varepsilon_j)$ prend la place d'un “moteur” non linéaire, apte à déclencher une **bifurcation** quand certaines liaisons excèdent un “seuil critique”.

Cette analogie se trouve renforcée par l'**interprétation énergétique**. Dans la synergetics, on peut décrire un système par une fonction de potentiel (ou d'énergie libre) qui décroît au fur et à mesure qu'un ordre s'établit. De la même manière, le **DSL** assimile la dynamique des poids $\omega_{i,j}$ à la recherche d'un certain **ordre** (ex. configuration de clusters) reflétant un minimum de quasi-énergie ou un optimum local en matière de synergie. Le **bruit**, au sens physique, peut alors jouer un rôle

crucial où un léger déséquilibre peut suffire à amorcer la croissance d'un cluster et un niveau modéré de perturbation autorise le réseau à s'extraire de minima sous-optimaux.

Les **applications** de cette métaphore sont variées. Dans le **DSL**, dès lors qu'on interprète les entités comme des "particules informationnelles" interagissant via $\mathbf{S}(i, j)$, la transition d'un état dispersé à un état **agrégé** (clusterisé) s'apparente à un passage de phase. Les liens $\omega_{i,j}$ s'accroissent jusqu'à constituer un sous-réseau stable, tandis que les connexions marginales disparaissent. Cette émergence graduelle peut faire l'objet d'une analyse semblable aux **transitions de phase** en physique, soulignant l'universalité d'un mécanisme auto-organisé où un ordre global naît d'un **feed-back** local.

Dans la conception qu'en donnent les promoteurs du DSL, on retrouve la volonté de **prolonger** l'héritage hakenien où la recherche d'un "**ordre stable**" (clusters ou attracteurs) émerge à partir d'interactions **locales** successives plutôt que d'imposer a priori une architecture rigide. Cette parenté s'illustre particulièrement bien lorsque la mise à jour des pondérations s'interprète comme une **minimisation** d'une forme d'énergie, associée à la diffusion des **synergies** $\mathbf{S}(\mathcal{E}_i, \mathcal{E}_j)$. L'inspiration de la thermodynamique hors équilibre conduit, en outre, à admettre que de faibles perturbations ponctuelles peuvent déclencher de grandes réorganisations, gage d'une **plasticité** et d'une capacité à "trouver" l'ordre dans le désordre.

Ainsi, l'apport de la **physique statistique** et de la **synergetics** fournit à l'auto-organisation un **formalisme** qui accentue la notion de **transition** et de **paramètre d'ordre**. Les paradigmes ultérieurs, comme le **Deep Synergy Learning**, reprennent les principes de boucles de rétroaction positive et d'amplification des synergies, en remplaçant les interactions spin par des **synergies** plus complexes adaptées à divers types d'entités (ex. vecteurs, symboles, flux temporels). La logique sous-jacente demeure toutefois la même où l'**ordre** émerge de multiples mises à jour locales et le système peut franchir des paliers critiques pour se stabiliser en configurations cohérentes et **auto-organisées**.

2.1.3.3. Premiers Parallèles avec la Notion de Synergie dans un Réseau d'Entités

Les avancées exposées en **neurosciences computationnelles** (section 2.1.3.1) et en **physique statistique** (section 2.1.3.2) ont fait ressortir un principe commun, un **réseau** composé de multiples entités (neurones, spins, agents) peut s'auto-organiser via l'accumulation d'**interactions locales**. Cette double perspective – dynamique neuronale et thermodynamique hors équilibre – a conduit à employer le terme de "**synergie**" pour décrire la force ou l'intérêt mutuel reliant deux unités dans le réseau. Les points suivants (2.1.3.3.1 à 2.1.3.3.4) soulignent comment, dès ces premiers travaux, on peut rapprocher cette démarche de ce qui sera plus tard le **Deep Synergy Learning (DSL)**.

2.1.3.3.1. Passer du Vocabulaire de la Physique au Vocabulaire Neuronal

Dans les modèles Hopfield ou Ising, on définit une **énergie** globale mesurant la cohérence ou, à l'inverse, le désordre. Les entités y sont liées par des **couplages** ω_{ij} . De la même manière, la **synergetics** (Haken) postule qu'un **paramètre d'ordre** peut émerger lorsque les connexions se renforcent collectivement. On a alors suggéré de passer d'une lecture "énergie négative" ou "coupling" à la notion plus "positive" de **synergie**, où celle-ci exprimerait la "**plus-value**" constructive de la relation entre deux unités. En neurosciences, il est fréquemment question de "**gain**" ou d'"affinité" entre neurones, soulignant la co-activation productive. Dans la lignée de la

plasticité synaptique, la co-occurrence neuronale a souvent été décrite comme un facteur de renforcement local, certains articles utilisant explicitement la terminologie “**co-information**” ou “**co-opération**”. Ainsi, les premières bases d’une “synergie” ont été posées par analogie avec la description de l’énergie dans les systèmes physiques.

2.1.3.3.2. Hypothèses de Lien Local-Global

Les théoriciens de la **synergetics** ont mis l’accent sur le fait que, si la “force” locale entre deux entités \mathcal{E}_i et \mathcal{E}_j est suffisamment élevée, elles tendent à se synchroniser ou à coopérer. Au fur et à mesure que ce phénomène se propage dans l’ensemble du réseau, on assiste à la formation d’un **état ordonné** (un cluster, un attracteur, un domaine de spins, etc.). La **coopération** locale est donc censée produire un **ordre global** via une accumulation de petites interactions. Cette dynamique se retrouve tant dans la formation d’assemblées neuronales (en neurosciences) que dans la constitution de domaines magnétiques (en physique), ou encore dans le principe d’un **réseau** qui franchit un **seuil** pour consolider des groupes fortement connectés. La synergie, en se focalisant sur la “valeur constructive” de la liaison entre deux unités, sous-entend que l’effet global est un gain pour l’organisation du système. On peut alors parler d’une transition structurelle, où le **paramètre de contrôle** (température, taux η) déclenche une réorganisation.

Certains modèles de champs neuronaux introduisent un poids $\omega_{ij}(t)$ dépendant de la corrélation entre les signaux d’activation, ce qui s’apparente déjà à une **mesure de synergie**. Dans le domaine de la physique, des études soulignent qu’une “mise en phase” (phase-locking) entre deux oscillateurs peut être quantifiée par un indice de co-activation, lequel s’approche également de l’idée de synergie. Dans ce contexte, des heuristiques simplifiées s’énoncent sous forme de règles locales, si la corrélation entre deux unités dépasse un certain seuil, le lien se renforce ; dans le cas contraire, il s’affaiblit. Bien que rudimentaires, ces règles ouvrent la voie à un cadre plus général, comme celui qu’adoptera le **DSL**. Une fonction $S(i,j)$ exprimant le degré d’interaction “utile” ou “cohérente” entre \mathcal{E}_i et \mathcal{E}_j . La démarche se concrétise en améliorant le simple critère “corrélation positive” vers d’autres formes de co-information.

L’idée de **réseaux dynamiques** où les liens $\omega_{i,j}(t)$ évoluent en continu pour mettre en avant ou en retrait certaines connexions tire parti de cette notion naissante de synergie. Les modèles Hopfield, par exemple, conservent leurs poids constants après un certain apprentissage, alors que la **SOM** (Kohonen) s’appuie sur un voisinage prédéfini. Par contraste, dès qu’on conçoit la synergie comme une **variable** interne évoluant librement, le réseau s’apparente à un **système hautement plastique**. Il abandonne un cadre statique en faveur d’une mise à jour incessante, suivant

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)].$$

Ce schéma préfigure le **Deep Synergy Learning**, où les entités ne sont plus restreintes aux neurones biologiques ou à des spins, mais peuvent inclure des capteurs, des modules symboliques ou d’autres agents. Le concept de **synergie** s’y généralise, passant d’une simple co-activation à une interaction multimodale ou multiniveau. De surcroît, la *structure du réseau* elle-même se trouve remodelée, la synergie permettant de dépasser le canevas fixe des SOM ou des réseaux associatifs classiques.

En définitive, la **convergence** entre neurosciences et physique a forgé la notion d’une synergie locale provoquant un ordre global. Les premières approches ont implémenté ce concept sous forme

de règles de corrélation, de mesures de co-information ou de couplages spin, suscitant le projet d'un **réseau auto-organisé** dans lequel l'évolution des liaisons répondrait à un principe de "plus-value" coopérative. Le **DSL** émerge dans cette continuité, proposant un cadre unificateur où la synergie prend des formes variées et se règle en permanence, traduisant la capacité du réseau à **former, rompre, ou stabiliser** ses clusters d'entités en l'absence de supervision.

2.1.4. Précurseurs de la Synergie Multimodale

Les recherches en **auto-organisation** se sont longtemps concentrées sur des données relativement homogènes (spectres sensoriels similaires ou vecteurs numériques). Cependant, dans de nombreux contextes biologiques et robotiques, des **modalités** distinctes (vision, toucher, audition, etc.) interagissent simultanément, incitant les chercheurs à imaginer des **systèmes** capables de fusionner plusieurs types de signaux au sein d'un même réseau adaptatif. Ce sous-chapitre (2.1.4) met en lumière comment, avant même l'essor du **Deep Synergy Learning (DSL)**, des équipes ont tenté de faire coexister différentes sources d'information dans des architectures orientées auto-organisation.

2.1.4.1. Initiatives Cherchant à Fusionner Plusieurs Types de Données (Vision + Tactile, etc.)

Dès les premiers travaux en **robotique sensorielle**, l'idée d'une **multimodalité** s'est imposée comme un enjeu central. En effet, un robot ou un système cognitif peut être pourvu simultanément de caméras (capteurs visuels) et de senseurs tactiles (capteurs de pression, vibration), et souhaiter **découvrir** de manière autonome des corrélations ou des "**synergies**" entre ces différents flux. Le besoin de modéliser cette complémentarité, sans supervision explicite, a mené à diverses expérimentations démontrant que la **co-activation** de canaux sensoriels peut se traduire par un **renforcement** des liens entre entités, selon un principe déjà évoqué en section [2.1.3](#).

D'anciens laboratoires ont notamment testé, dès la fin des années 1970, l'association de **réseaux neuronaux rudimentaires** à des données issues d'**images** et de **capteurs tactiles**, dans le but de reconnaître un objet ou de caractériser une scène. Le fonctionnement de ces prototypes consistait à considérer l'activation simultanée d'une zone visuelle \mathbf{v}_i et d'une zone tactile \mathbf{t}_j . Dans un schéma de type hebbien, la pondération $\omega_{(v_i, t_j)}$ augmentait si ces deux entités coïncidaient lors de la perception d'un même objet. On peut formuler la mise à jour comme

$$\Delta\omega_{(v_i, t_j)} \propto \mathbf{v}_i \mathbf{t}_j,$$

indiquant que la corrélation entre la voie visuelle \mathbf{v}_i et la voie tactile \mathbf{t}_j renforce la liaison les unissant. À mesure que ces expériences étaient réitérées, la liaison correspondante se consolidait jusqu'à se traduire par une coopération plus robuste entre les deux modalités.

Dans le sillage des **Self-Organizing Maps (SOM)** proposées par Teuvo Kohonen, certains chercheurs ont implémenté des "SOM double", consistant en deux grilles topologiques (l'une pour la vision, l'autre pour le tactile) reliées par un **mécanisme de pont**. Lorsque le même objet était perçu sur les deux modalités, le **voisinage** dans la grille visuelle et la grille tactile incitait chacune de ces deux cartes à se rapprocher, créant une **zone bimodale** où les neurones se trouvaient mutuellement renforcés. Bien que limités, ces essais de **fusion** sans label explicite démontrent déjà le concept qu'une **auto-organisation** conjointe peut conduire à des sous-zones où les

informations visuelles et tactiles se *recouvrent*, facilitant une identification plus fiable des objets manipulés.

Parallèlement, les avancées en **neurosciences** révélaient l'existence d'aires associatives "**amodales**" dans le cortex, suggérant que la **coopération** des signaux visuels et tactiles peut survenir naturellement pour former une **perception** intégrée. Les modèles computationnels s'inspiraient de ce constat où à chaque co-occurrence de signaux, on augmentait la force d'un lien (ou d'un cluster) reliant ces deux canaux, ce qui, d'un point de vue mathématique, revient à généraliser la corrélation binaire vers une notion de co-activation dans le temps. De cette manière, l'apprentissage non supervisé exploitait la co-occurrence répétée des mêmes événements sur deux voies sensorielles, sans qu'il soit nécessaire qu'un humain étiquette explicitement leur correspondance.

Cette démarche de **renforcement** adaptatif entre flux hétérogènes préfigure la dynamique mise en œuvre dans le **Deep Synergy Learning (DSL)**. Au lieu de se restreindre à un couplage stationnaire (une SOM visuelle, une SOM tactile, etc.), on peut imaginer un **réseau** où les liens $\omega_{(i,j)}(t)$ entre n'importe quelles entités — visuelles, tactiles, auditives ou symboliques — évoluent en continu, selon une fonction de synergie $S(i,j)$ reflétant leur degré d'activation conjointe. La mise à jour suit un schéma tel que

$$\omega_{(i,j)}(t+1) = \omega_{(i,j)}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{(i,j)}(t)],$$

ce qui rejoint la logique des heuristiques multimodales employées dans ces premiers prototypes. Ces recherches antérieures se révèlent donc être un **préalable direct** à la formulation du DSL, car elles ont posé les bases d'un **apprentissage réellement non supervisé**, tirant parti de la complémentarité entre différents flux sensoriels ou conceptuels, et structurant spontanément un **réseau** où les pondérations fortes dénotent une intense **synergie** entre modalités.

Avant l'avènement du **Deep Synergy Learning**, divers travaux en robotique et en neurosciences computationnelles avaient démontré la viabilité d'une auto-organisation **multimodale**, où le couplage entre modalité visuelle et tactile, notamment, se révélait un cas d'école. L'observation de corrélations temporelles ou fonctionnelles suffisait à susciter un **apprentissage** de type clustering, aboutissant à des représentations plus riches et plus stables que l'approche unimodale. Les idées ainsi expérimentées ont confirmé qu'il n'était pas nécessaire de disposer d'un **superviseur** externe pour associer deux flux distincts où la co-occurrence ou la convergence récurrente s'avérait largement suffisante pour laisser **émerger** la liaison adaptative souhaitée. Ce constat précède et anticipe directement les principes du DSL, qui pousse plus loin la logique de **synergie adaptative** dans un cadre uniifié et potentiellement extensible à plusieurs entités ou modalités simultanées.

2.1.4.2. Influences sur la Mise en Place d'un Cadre Unifié de Synergie Adaptative

Les expériences décrites en section 2.1.4.1, qu'il s'agisse de **fusion** vision+tactile ou de cartes "double-SOM" connectées, ont fait émerger une nécessité où il devient essentiel de disposer d'une **approche générale** pour décrire la manière dont plusieurs **modalités** (ou entités hétérogènes) peuvent coopérer par **auto-organisation**. Les initiatives multimodales pionnières ont souligné l'intérêt d'étendre les règles traditionnelles d'apprentissage sans superviseur vers des situations où les **entités** (capteurs, flux symboliques, représentations vectorielles) ne partagent pas forcément le même format ni le même espace de description. Les réflexions suscitées dans ce domaine ont

contribué à l’élaboration ultérieure de paradigmes tels que le **Deep Synergy Learning (DSL)**, dont la logique unifie et **généralise** les notions d’interactions adaptatives entre entités variées.

A. Impulsion depuis la Synergie “Bimodale”

En mettant en relation, par exemple, des données **visuelles** (images) et **tactiles** (capteurs de pression), plusieurs laboratoires ont observé que les **règles classiques** (du type corrélation ou co-activation) se révélaient efficaces même lorsque les flux appartenaient à des espaces différents. Au lieu de se limiter à une **similarité** calculée dans un espace vectoriel identique, ces chercheurs ont proposé des fonctions de “**matching**” ou de “distance inter-modale”, aptes à comparer un patch visuel à un signal tactile. On voyait ainsi qu’il suffisait d’avoir une mesure d'**affinité** cohérente pour permettre un renforcement local, selon un principe semblable à :

$$\Delta\omega_{i,j} \propto \text{similarité}(\mathbf{x}_i, \mathbf{x}_j).$$

Cette conclusion a incité la communauté à considérer des fonctions encore plus flexibles, où chaque couple d’entités définirait sa propre notion de corrélation ou de distance, préparant le chemin vers une **synergie** plus générale.

B. Vers une Extension n-aire

À partir de la fusion de deux modalités, un passage naturel a consisté à s’interroger sur la possibilité de connecter **plusieurs** flux simultanément. Dès lors, on envisageait une **synergie n-aire**, où au-delà d’un simple produit binaire (vision + tactile), on permettait la coopération de la vision, du toucher, de l’audition, voire d’autres sources (signaux proprioceptifs, informations symboliques, etc.). Cette extension laissait entrevoir des perspectives de **cohésion collective** où un groupe d’entités pouvait se synchroniser autour d’un même événement, même sans label externe, par simple co-occurrence en temps ou en contexte. Les premiers essais, bien qu’expérimentaux, ont montré qu’un réseau pouvait découvrir des regroupements plus sophistiqués dès lors qu’il disposait de **flux variés** et d’une mise à jour adaptative des liens. Sans le cadre théorique unifié, la mise en œuvre restait toutefois ad hoc.

C. Nécessité d’un Cadre Mathématique Unifié

L’extension à plusieurs modalités a révélé les limites des **architectures** de type SOM isolée ou réseau associatif spécialisé. De multiples travaux ont souligné qu’on avait besoin d’un **réseau** davantage “libre” où, au lieu de cloisonner chaque flux dans un module (une carte), il serait plus fécond de concevoir un unique **ensemble** d’entités au sein duquel n’importe quelle paire (ou sous-ensemble) pourrait maintenir une liaison $\omega_{i,j}$ reflétant la “synergie” $S(\mathcal{E}_i, \mathcal{E}_j)$. Cette vision implique un **espace d’interaction** où la **pondération** entre entités évolue au fil du temps, en s’appuyant sur des règles locales de renforcement ou d’affaiblissement. Le concept est proche du **Synergistic Connection Network (SCN)** décrit dans la suite du **DSL** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Ici, \mathcal{E}_i et \mathcal{E}_j peuvent être des entités de natures diverses (sensorielles, symboliques, etc.), et la fonction S se charge d’estimer le degré de “**cohérence**” ou de “**complémentarité**”.

D. Évolution Vers la Pondération Adaptative

Dans les premières “SOM doubles” ou architectures robotiques multimodales, les liens entre modalités restaient parfois figés ou suivaient un schéma de **mapping** imposé. Les itérations ultérieures, plus ambitieuses, ont cherché à rendre ces **connexions** elles-mêmes **dynamiques**, comme s’il s’agissait d’une extension de la logique de Hebb ou de la compétition neuronale. On posait donc la question :

“Pourquoi ne pas autoriser la liaison *inter-cartes* à se mettre à jour en temps réel, au même titre que les liaisons *intra-carte* ?”

Cette interrogation a mené à l’idée d’un **réseau unique** plutôt que de modules encapsulés, préparant la conception que le **Deep Synergy Learning** développera où un SCN global évolue au gré de la synergie perçue entre tous les **types** d’entités.

E. Préparation du Terrain pour le DSL

En repensant l’architecture de l’apprentissage auto-organisé sous un angle plus universel, on arrivait progressivement à un **paradigme** où l’on n’avait plus à distinguer a priori les données en tant que “vision”, “tactile” ou “auditive”. On leur attribuait plutôt un statut générique **d’entités** dans un réseau, reliées par des liaisons qui s’ajustaient en fonction d’un **score** de co-opération ou de co-information. Ainsi se dessinait déjà l’idée d’une **synergie** non supervisée, en suivant le fil rouge des premiers prototypes multimodaux et en l’étendant à un cadre totalement **auto-adaptatif**.

Les travaux de recherche dans la robotique sensorielle et l’intégration multimodale ont de fait affirmé la **faisabilité** d’une telle approche. Ils ont mis en évidence que la conclusion “ces deux flux s’avèrent souvent synchronisés, donc ils doivent être reliés et codés conjointement” pouvait être atteinte **sans** recours à un signal d’erreur supervisé. C’est sur cette base qu’ont émergé des pistes ultérieures visant une formalisation plus large de la “synergie” et, en particulier, l’essor de paradigmes comme le **Deep Synergy Learning** qui généralisent ces principes en offrant un langage pour la **coopération** entre entités et la **réorganisation** des liaisons $\omega_{i,j}$.

Dès lors, la volonté de **fusionner** plusieurs types de données a joué le rôle d’un catalyseur, forçant la communauté à sortir d’architectures préétablies pour considérer un **réseau unifié** et malléable. Les règles d’évolution des pondérations sont alors guidées par une fonction $S(i,j)$ – future “fonction de synergie” – et cette logique d’interactions adaptatives reste au cœur du **DSL**.

En conclusion, ces initiatives multimodales ont poussé à systématiser l’**auto-organisation** au-delà de la simple proximité de vecteurs, afin d’engendrer un **cadre** où toute forme de similarité ou de **complémentarité** (visuelle, tactile, auditive, symbolique) puisse s’exprimer sous forme de synergie, ouvrant ainsi la voie aux principes fondateurs du **Deep Synergy Learning**.

2.1.5. Origine du DSL et Ses Premiers Manifestes

Les divers courants présentés dans les sections précédentes (2.1.1 à 2.1.4) — allant des travaux pionniers en auto-organisation jusqu’aux tentatives de fusion multimodale — ont fini par converger vers l’idée de constituer un **cadre unifié** où chaque entité (capteur, neurone, module symbolique, etc.) pourrait interagir avec les autres via un mécanisme de **synergie adaptative**. C’est dans ce

contexte qu'est apparue la notion de **Deep Synergy Learning (DSL)**. Le sous-chapitre qui suit (2.1.5) s'intéresse à la genèse historique et à la cristallisation de l'étiquette "DSL" dans la littérature, puis à la formalisation progressive d'un "Synergistic Connection Network (SCN)".

2.1.5.1. Quand et Comment l'Étiquette "Deep Synergy Learning" Apparaît pour la Première Fois : la Publication dans cet Ouvrage

Au tournant des années 2010, divers laboratoires et chercheurs ont exploré l'idée de **réseaux adaptatifs** capables de gérer simultanément des entités ou des modalités multiples, dans la lignée des inspirations décrites dans les sections précédentes. Toutefois, le terme "**Deep Synergy Learning (DSL)**" tel qu'il est présenté ici n'avait jusqu'alors **jamaïs** fait l'objet d'une publication ou d'une formalisation officielle. C'est dans **ce livre**, pour la première fois, que je rends publiques et détaillées les bases conceptuelles et le fonctionnement d'un **réseau** apte à intégrer la notion de **synergie** au sens large, qu'il s'agisse d'entités perceptuelles, symboliques, ou mixtes.

L'idée de la **synergie** m'accompagne depuis plus de **quarante ans**, influencée par mes travaux initiaux en informatique, par les modèles d'**auto-organisation** neuronale (type Hopfield, Kohonen) et par la **thermodynamique hors équilibre**. Au fil des décennies, ces réflexions se sont enrichies de la logique "hebbienne", des simulations biologiques et de la fusion multimodale, mais n'ont jamais été diffusées dans un ouvrage ou un article de référence. Aucun document antérieur ne porte l'étiquette "**Deep Synergy Learning**" au sens où je l'entends aujourd'hui. Ce livre constitue dès lors la **première** présentation formalisée du DSL, énonçant ses principes, ses justifications théoriques et ses applications potentielles.

On trouvera dans les chapitres ultérieurs la description du **Synergistic Connection Network (SCN)**, pièce maîtresse du DSL, ainsi que l'ensemble des règles de mise à jour permettant de faire évoluer les liaisons $\omega_{i,j}$ selon une fonction de **synergie** $S(i,j)$. Si l'on retrouve des traces d'inspirations diverses où les réseaux associatifs, les cartes topologiques et la plasticité synaptique sont présents, la synthèse qui en résulte se veut inédite puisqu'elle généralise l'**auto-organisation** à un large spectre d'entités tout en recherchant un ordre émergent via des **pondérations adaptatives**. Cette publication, en franchissant pour la première fois les frontières des laboratoires et des notes internes, aspire donc à faire connaître le DSL et à favoriser son adoption dans un grand éventail de contextes, de la robotique sensorielle à la cognition symbolique.

Ainsi, le **Deep Synergy Learning**, dont je pose ici les fondements, n'avait jusque-là jamais été officiellement exposé ni dans un livre ni dans un article. Ce texte inaugure donc son entrée dans la **littérature** scientifique, concrétisant une vision mûrie de longue date mais demeurée confidentielle jusqu'à présent.

2.1.5.2. Évolution vers la Formalisation d'un "SCN" (Synergistic Connection Network)

La **naissance du Deep Synergy Learning (DSL)** s'est accompagnée d'une volonté de conférer à la **synergie adaptative** une **incarnation concrète** au sein d'un réseau à la fois généraliste et dynamique. Les réseaux associatifs traditionnels (Hopfield) et les cartes topologiques (SOM) ont inspiré certains principes où ils restaient néanmoins marqués par des limites telles que l'absence

de plasticité continue, un champ d’application restreint et la difficulté à intégrer des entités non neuronales ou des flux multimodaux. Le **SCN**, au contraire, se veut un **réseau** entièrement **évolutif**, où chaque pondération $\omega_{i,j}$ dépend d’une fonction de synergie $S(i,j)$ susceptible d’unifier différents types d’entités (capteurs sensoriels, modules symboliques, etc.). Les paragraphes ci-dessous (2.1.5.2.1 à 2.1.5.2.5) retracent la **progression** qui a permis d’aboutir à ce cadre.

A. Passer d’un Réseau “Fixe” à un Réseau “Entièrement Dynamique”

Les approches antérieures telles que les **SOM** de Kohonen ou les **réseaux** Hopfield présentaient un fonctionnement essentiellement **statique**. Une SOM impose une grille et un mode de voisinage prédéterminé, tandis que les poids d’un réseau Hopfield sont généralement calculés une fois pour toutes sur la base de motifs ciblés. Ces méthodes ne permettent pas de gérer en continu l’arrivée de nouvelles entités ou de mettre à jour les liaisons dans des environnements en transformation permanente. Inspiré par la **plasticité neuronale**, a suggéré que le réseau pourrait évoluer par itérations successives, les pondérations $\omega_{i,j}(t)$ se renforçant ou s’atténuant selon un **score de synergie** $S(i,j)$. L’équation générique (développée plus tard dans ce livre) concrétise cette idée en intégrant un **terme de renforcement** (quand la synergie est élevée) et un **terme de décroissance** (pour éviter l’emballlement), assurant un équilibre entre stabilité et adaptabilité.

B. Notion de “Synergistic Connection Network”

Sous l’impulsion de ces réflexions, est né le concept de **Synergistic Connection Network (SCN)**, un graphe (voire un hypergraphe) dont les nœuds correspondent à des **entités** $\mathcal{E}_1, \mathcal{E}_2, \dots$, alors que les arêtes pondérées $\omega_{i,j}$ incarnent la liaison entre \mathcal{E}_i et \mathcal{E}_j . La distinction cruciale, par rapport aux architectures classiques, réside dans le fait que ces liaisons ne sont pas figées où elles varient sous l’effet d’une **fonction** $S(\mathcal{E}_i, \mathcal{E}_j)$ évaluant la synergie. À chaque étape (itération, nouvelle donnée, flux continu), les pondérations se recalculent pour refléter le degré de coopération ou de similitude actuelle entre les entités. Ce **réseau vivant** autorise la découverte de structures auto-organisées dans des champs aussi divers que la **fusion sensorielle**, la **cohérence symbolique** ou la **mise en correspondance** de flux d’origine variée.

C. Dynamique de la Fonction $\omega_{i,j}(t)$ et Synergie $S(i,j)$

Le **SCN** s’appuie sur un mécanisme de **plasticité locale** où la pondération $\omega_{i,j}(t)$ entre deux entités \mathcal{E}_i et \mathcal{E}_j obéit, à chaque itération, à une équation de type

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(\mathcal{E}_i, \mathcal{E}_j) - \tau \omega_{i,j}(t)].$$

Ici, η est un **taux d’apprentissage**, tandis que τ joue le rôle d’un **coefficent** de régulation, souvent nommé “terme de décroissance”. La **fonction** $S(i,j)$ peut être une simple corrélation où elle peut aussi prendre une forme plus élaborée comme une distance inversée, une similarité cosinus ou une information mutuelle, etc. Cette souplesse correspond à l’ambition du SCN de **fusionner** différents types de données, en adoptant une métrique ou une fonction de synergie appropriée pour chaque

couple (ou sous-groupe) d'entités. L'effet global est un **réseau** qui se reconfigure sans cesse, renforçant les liaisons cohérentes et affaiblissant les autres, selon les principes hebbiens et les idéaux de la **physique hors équilibre** (cf. chapitres [2.1.3.1](#) et [2.1.3.2](#)).

D. Diffusion dans des Publications et Ateliers

Au fil des travaux, ce schéma évolutif a pris forme dans des **rapports** et **communications** spécialisées, notamment dans le cadre de congrès sur l'**apprentissage non supervisé**. Les premières applications pratiques concernaient la fusion multisensorielle ou la classification évolutive, montrant que le SCN sait gérer la venue de nouvelles données, adapter les pondérations au fil de l'expérience et former des **clusters** persistants. Parmi les questions soulevées figurent la **scalabilité** (comment se comporte le SCN avec un grand nombre d'entités ?) et la **stabilité** (comment éviter que les pondérations ne divergent ou ne s'étiolent toutes ?). Ces interrogations ont orienté la mise au point de mécanismes de **parsimonie** (seuils, inertie minimale), d'**inhibition** compétitive et de **taux d'apprentissage** dynamique, dont les détails seront explicités dans les prochains chapitres.

E. Positionnement pour la Suite de l'Ouvrage

Ainsi, la création du **Synergistic Connection Network** marque une étape décisive dans la concrétisation du **Deep Synergy Learning**. Pour la première fois, un **réseau** est conçu pour embrasser la diversité des entités (capteurs de nature variée, modules symboliques, etc.), tout en veillant à ce que les liaisons $\omega_{i,j}$ restent **adaptatives** et que la **synergie** $S(i,j)$ puisse se décliner sous des formes diverses. Les chapitres suivants de cet ouvrage (chapitres 3, 4, 5) approfondiront la formulation analytique, les algorithmes de mise à jour, et les fondements mathématiques de la dynamique $\omega_{i,j}(t)$. On y détaillera notamment la notion de **cluster** en tant qu'attracteur emergent, la gestion de la **parsimonie** afin de maintenir la lisibilité et la scalabilité, ainsi que les méthodes pour **évaluer** la pertinence des synergies dans des scénarios multimodaux réels.

En définitive, la formalisation du **SCN** illustre la transition d'une simple **idée** (renforcer les liens lorsqu'on détecte une forte co-occurrence) à un **cadre complet d'auto-organisation** appliqué aux entités hétérogènes. Le DSL s'appuie désormais sur cette armature conceptuelle pour proposer un système de pondérations locales, à la fois général et flexible, ouvrant la voie à une **intelligence** non supervisée, capable de gérer la diversité des données et de s'auto-structurer en conséquence.

2.1.5.3. Lien Direct avec la Suite du Livre : Axes Théoriques Restés Inexplorés

La **genèse** du **Deep Synergy Learning (DSL)** et la formalisation du **Synergistic Connection Network (SCN)**, évoquées dans les sections précédentes, laissent entrevoir un champ encore **vaste** de problématiques que la littérature n'a pas couvertes de manière exhaustive. Le présent ouvrage se propose précisément de combler plusieurs de ces **lacunes** et de consolider les bases théoriques et pratiques du DSL. La discussion suivante met en évidence les principaux **axes** de recherche

inachevés, tout en exposant la manière dont les chapitres ultérieurs (3 à 15) s’emploient à les approfondir ou à proposer des réponses partielles.

A. Extension de la Fonction de Synergie à des Contextes Plus Complexes

La plupart des modèles initiaux se limitaient à des **fonctions** de similarité relativement simples, comme la distance euclidienne ou la corrélation binaire, pour calculer la synergie entre deux entités \mathcal{E}_i et \mathcal{E}_j . Or, certains domaines exigent une **cohérence** ou une **interaction** plus élaborée, qui peut s’appuyer sur des notions de co-information n-aire ou sur des métriques conditionnelles sophistiquées. D’un point de vue mathématique, on se heurte alors à la nécessité de définir

$$S(\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n)$$

pour un sous-groupe de taille $n > 2$, ce qui amène à de nouveaux défis de calcul et de stockage. Le **Chapitre 3** aborde la représentation des entités et jette les bases de la synergie, tandis que le **Chapitre 12** traite spécifiquement de la **synergie n-aire**, en exposant des formules permettant de saisir les interactions multivariées.

Par ailleurs, le **DSL** se révèle apte à gérer des flux en temps continu. Cela suppose que la fonction $S(i, j)$ demeure **adaptative** au fil du temps, afin de prendre en compte l’évolution possible des entités \mathcal{E}_i et \mathcal{E}_j . Les **Chapitres 9** et **11**, consacrés respectivement aux scénarios temps réel et à la robustesse/sécurité, examinent la façon dont le **SCN** peut conserver la mémoire des synergies passées, tout en s’ajustant à l’arrivée de nouvelles données ou d’entités inédites.

B. Contrôle de la Stabilité et Algorithmes d’Optimisation

Il est légitime de s’interroger sur la **stabilité** de la règle de mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t)\eta[S(i, j) - \tau \omega_{i,j}(t)].$$

D’un point de vue analytique, il s’agit de savoir si ces équations peuvent engendrer des **oscillations**, ou même des comportements chaotiques, lorsque le réseau s’agrandit ou que la matrice $\{\omega_{i,j}\}$ atteint une dimension élevée. Les **Chapitres 4, 5, 7 et 10** explorent ces problématiques. Les solutions proposées incluent, entre autres, l’ajout d’un mécanisme d’**inhibition compétitive** (destiné à empêcher l’explosion d’un trop grand nombre de liaisons fortes), l’introduction de seuils minimaux pour couper les connexions trop faibles ou encore l’élaboration de stratégies de **parsimonie** structurelle.

La **scalabilité** en présence de milliers, voire de millions, d’entités devient un enjeu majeur. Les approches naïves peuvent rapidement se heurter à une complexité en $O(n^2)$. Le **Chapitre 7** détaille des algorithmes plus performants, tels que l’**échantillonnage** ou l’**approximation** de voisinage, et puise dans des analogies avec la **physique statistique** (par exemple le recuit simulé) afin de maintenir la convergence sans tout calculer explicitement. Les utilisateurs du **DSL** se voient ainsi offrir un éventail de méthodes pour contrôler la densité du réseau et éviter que les liaisons $\omega_{i,j}$ ne prolifèrent ou ne s’effondrent de manière incontrôlée.

C. Ouverture vers la Multimodalité Avancée et le Raisonnement Symbolique

Un autre front de recherche concerne la **multimodalité** lorsqu’elle est poussée à l’extrême où il ne s’agit plus seulement de relier deux flux (vision + tactile) mais d’intégrer simultanément des

sources diverses comme les images, les sons, les textes et la logique symbolique. Dans cette optique, on souhaite s'assurer que la **fonction de synergie** $S(\cdot, \cdot)$ ou $S(\cdot, \cdot, \cdot)$ soit assez flexible pour gérer des écosystèmes d'entités hétérogènes, tout en garantissant que le SCN ne se fragmente pas en une myriade de **clusters** trop spécialisés et déconnectés. Le **Chapitre 8** se penche spécifiquement sur ces scénarios multimodaux et propose des pistes pour favoriser la **transversalité** entre flux distincts. Les **Chapitres 14** et **15**, davantage orientés applications, reviennent sur divers cas concrets (fusions de flux audio-visuel-textuel, organisation de connaissances dans un cadre hybride).

La présence d'**entités symboliques** (règles logiques, axiomes) étend encore le périmètre d'action du DSL, et il s'agit alors de caractériser la **synergie** entre une entité perceptuelle (un capteur) et une entité abstraite (une proposition symbolique). Les **Chapitres 5, 13 et 14** abordent cette problématique en discutant des approches dites **neuro-symboliques**, où un réseau gère simultanément de la “matière” sub-symbolique (poids, neurones virtuels) et des “objets” symboliques. La recherche sur la cohabitation de ces deux mondes reste néanmoins incomplète, et le livre ne prétend pas en livrer une théorie achevée, mais propose des esquisses d'**implémentation** et des modèles hybrides encourageant un raisonnement distribué.

Conclusion sur les Axes Théoriques Restés Inexplorés

Le **Deep Synergy Learning**, formalement présenté dans ce livre pour la première fois (cf. section [2.1.5.1](#)), débouche sur un **cadre** suffisamment large pour accueillir des entités variées et autoriser une dynamique de pondérations adaptatives. Les **lacunes** ou **chantiers** toujours ouverts concernent d'abord l'élaboration de **fonctions de synergie** toujours plus polyvalentes (synergie n-aire, co-information conditionnelle). Ils portent aussi sur l'analyse **mathématique** de la stabilité du **SCN** (en particulier dans les grands réseaux), et enfin sur la **fusion** multimodale avancée associant données sensorielles, logiques symboliques et autres ressources. Les chapitres à venir s'emploient à clarifier chacun de ces thèmes :

- Les **fondements mathématiques** (Chapitre 3 et Chapitre 4)
- Les **règles de mise à jour** et la **structure interne** du SCN (Chapitre 5, Chapitre 7)
- Les scénarios d'**évolution en temps réel** (Chapitre 9) et de **multimodalité** (Chapitre 8, Chapitre 14)
- Les **cas pratiques** et les extensions potentielles (Chapitre 15)

Ainsi, le livre fournit un socle formel pour la **synergie adaptative** entre entités. La variété des pistes encore ouvertes souligne combien le **DSL** n'est pas un aboutissement final, mais plutôt un **cadre évolutif** lui-même, évoluant au rythme des investigations en intelligence artificielle, en neurosciences computationnelles et en physique des systèmes complexes.

Les chapitres suivants du livre aborderont bon nombre de ces **challenges**, proposant des embryons de solution ou montrant des applications concrètes. Le lecteur verra comment ces problématiques s'imbriquent dans la dynamique du SCN, et comment l'héritage historique présenté ici (depuis les

réseaux associatifs, les cartes topologiques, jusqu'aux modèles multi-sensoriels et la physique statistique) trouve sa continuité dans un cadre uniifié **Deep Synergy Learning**.

2.2. Principes Mathématiques de Base

Le **Deep Synergy Learning (DSL)** s'appuie sur un ensemble de **principes mathématiques** permettant de définir formellement les entités (leurs caractéristiques) et les règles d'**auto-organisation** du réseau. Contrairement à certains paradigmes neuronaux classiques (où l'on manipule seulement des poids fixes ou un gradient d'erreur), le DSL met en avant la **fonction de synergie** S et la dynamique de mise à jour des pondérations $\omega_{i,j}$. Ces concepts reposent sur des bases analytiques qui font intervenir la notion d'**espace de représentation**, de **similarité** ou de **co-information**, et de **procédés d'évolution** (discrets ou continus). Le présent chapitre (2.2) en présente les **fondements**, en plusieurs sous-sections :

- **2.2.1.** Définition des Entités et de la Fonction de Synergie
- **2.2.2.** Mise à Jour des Pondérations : Formule Générale
- **2.2.3.** Règles de Parsimonie et Seuil de Connexion
- **2.2.4.** Notions d'États Internes et Auto-Organisation
- **2.2.5.** Exemples Illustratifs

Ce sont les **piliers** mathématiques sur lesquels s'édifie le **Synergistic Connection Network (SCN)** et, plus généralement, l'ensemble de la logique d'**auto-organisation** du DSL.

2.2.1. Définition des Entités et de la Fonction de Synergie

La première étape pour tout modèle DSL consiste à **identifier** les **entités** $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ qui composeront le réseau et, surtout, à spécifier la **fonction de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. Si la littérature sur l'apprentissage non supervisé (SOM, Hopfield, etc.) imposait en général un format vectoriel unique, le DSL se distingue par sa capacité à gérer des **représentations hétérogènes** (capteurs, flux symboliques, extraits d'un transformeur, etc.). Les sections suivantes (2.2.1.1 à 2.2.1.3) détaillent cette notion de **définition d'entités** et la façon dont on conçoit la **fonction S** .

2.2.1.1. Rappel : Une Entité \mathcal{E}_i (Vecteur de Caractéristiques, Représentation Symbolique, etc.)

Le **Deep Synergy Learning (DSL)** s'articule autour de la notion d'**entité** \mathcal{E}_i et de la **synergie** définie sur l'ensemble de ces entités. Dans la pratique, on considère que chaque \mathcal{E}_i fait partie d'un **ensemble** \mathcal{X} qui peut prendre des formes variées où il peut s'agir d'un espace vectoriel, d'une structure symbolique ou d'une représentation probabiliste. Il est indispensable de disposer d'une **mesure** ou d'une **similarité** pour comparer deux entités, car c'est cette comparaison qui sous-tend la **pondération** $\omega_{i,j}$ et la **dynamique** d'auto-organisation du DSL. Les développements ci-après soulignent les différents cas de figure envisageables (sections vectorielles, symboliques, probabilistes) et introduisent un formalisme plus général permettant de prendre en compte des **espaces hétérogènes**.

A. Entités dans un Espace Vectoriel \mathbb{R}^d

Un premier cas, souvent le plus courant, suppose que chaque entité \mathcal{E}_i est représentée par un **vecteur** réel de dimension d . On écrit alors

$$\mathcal{E}_i \in \mathbb{R}^d,$$

et on dispose de **distances** ou de **similarités** naturelles : distance euclidienne, distance Minkowski, similarité cosinus, etc. Les propriétés de l'analyse vectorielle facilitent la définition de mesures telles que

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| \quad \text{ou} \quad \text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}.$$

Cette configuration se rencontre fréquemment dans la pratique (traitement d'images, de séries temporelles, etc.), et fournit une base conceptuelle aisée pour construire la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ par inversion de distance ou par évaluation de similarité, prolongeant l'idée de la corrélation ou de la co-occurrence.

B. Structures Symboliques ou Espaces Discrets

Une **entité** \mathcal{E}_i peut aussi correspondre à un **concept** symbolique, un **graphe** local, ou tout autre objet appartenant à un **espace** discret où aucune structure vectorielle n'est immédiatement disponible. Les ensembles de cette nature exigent la définition d'une **mesure** ou d'une **similarité** plus adaptée, prenant la forme d'une distance combinatoire, d'une ressemblance sémantique, ou d'une co-information symbolique. Formellement, on peut écrire

$$\mathcal{E}_i \in \mathcal{X}_{symbolique},$$

et introduire une fonction $\text{sim}(\mathcal{E}_i, \mathcal{E}_j)$ prenant des valeurs dans $[0,1]$ ou \mathbb{R}^+ . Ce cadre est crucial dans le DSL, qui autorise la **co-existence** d'entités hétérogènes et nécessite donc que la **synergie** s'applique aussi à des représentations non purement numériques. Par exemple, on peut concevoir une mesure de similarité symbolique $\sigma(\mathcal{E}_i, \mathcal{E}_j)$ qui vaut 1 si deux symboles sont identiques, ou prend une valeur intermédiaire si l'on recourt à une **ontologie** reliant les concepts.

C. Espace de Probabilité ou Représentation Probabiliste

Dans certains travaux, l'entité \mathcal{E}_i est un **modèle probabiliste**, par exemple une **distribution** $p_i(x)$. On peut alors définir la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ par une divergence ou une similarité adaptée, telles que la **KL divergence** ou la **distance Wasserstein**. Ainsi, on écrit

$$\mathcal{E}_i \in \mathcal{P}(\mathcal{X}),$$

où $\mathcal{P}(\mathcal{X})$ désigne l'ensemble des distributions sur un espace \mathcal{X} . La fonction S peut alors prendre des formes comme

$$S(\mathcal{E}_i, \mathcal{E}_j) = \exp(-d_{KL}(p_i, p_j)) \quad \text{ou} \quad S(\mathcal{E}_i, \mathcal{E}_j) = \text{Wasserstein}(p_i, p_j),$$

autorisant l'auto-organisation de **modèles** statistiques sans référence explicite à des labels supervisés.

D. Forme Générale et Hypothèses Topologiques

On adopte volontiers la notation

$$\mathcal{E}_i \in \mathcal{X},$$

où \mathcal{X} est un **espace topologique** (ou mesurable), muni d'une **distance** ou d'une **similarité**. Le **triplet** (\mathcal{X}, d, μ) ou (\mathcal{X}, S, μ) permet de considérer soit des distances classiques, soit des "fonctions de synergie" plus directes. Le **DSL** n'exige pas nécessairement que les entités se situent dans un unique espace vectoriel où différentes **sous-familles** d'entités peuvent coexister, chacune dotée de sa propre mesure.

E. Propriétés Clés : Séparabilité, Embedding Vectoriel, Extension Symbolique

Certaines **propriétés** mathématiques facilitent l'étude de la synergie. Par exemple, une **distance** d sur \mathcal{X} procure un espace métrique complet, ce qui autorise une **analyse** de la convergence de la fonction de synergie si elle se base sur d . Sous des conditions favorables (e.g. la structure de \mathcal{X} est hilbertienne), il est possible d'**implanter** (\mathcal{X}, d) dans un espace \mathbb{R}^d ou un **espace de dimension infinie** (via un **noyau**), ouvrant la voie à des similarités usuelles (produit scalaire, gaussien, cosinus). Lorsque \mathcal{X} est un ensemble **discret** (fin ou dénombrable), on peut définir une **similarité** $\sigma(\mathcal{E}_i, \mathcal{E}_j)$ dans $[0,1]$ basée sur une codification combinatoire ou sémantique.

F. Gestion de l'Hétérogénéité des Entités

Le **DSL** poursuit un objectif d'**intégration** de différents types d'entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$, qui peuvent appartenir à divers espaces $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_k$. Pour comparer $\mathcal{E}_i \in \mathcal{X}_p$ et $\mathcal{E}_j \in \mathcal{X}_q$, il est indispensable de **prolonger** les mesures ou d'introduire un **mécanisme de passerelle** (embedding commun, correspondance symbolique...). C'est dans cette optique qu'est formulé le **théorème** sur la consistance de la synergie inter-espaces, garantissant l'existence d'une fonction globale Σ unifiant les similarités σ_p dans un unique référentiel. Les détails topologiques (réunion disjointe, mise en place d'une distance globale δ séparant strictement les blocs) assurent la **continuité** de la fonction Σ .

Exemple de Théorème (Consistance de la Synergie Inter-Espaces)

Enoncé.

Soient $\{\mathcal{X}_m\}_{m=1}^k$ des espaces (topologiques ou mesurables), chacun muni d'une fonction de similarité σ_m . Sous hypothèses de continuité uniforme et de séparation stricte entre blocs, il existe une **extension** Σ définie sur la réunion disjointe $\mathcal{U} = \bigsqcup_{m=1}^k (\mathcal{X}_m \times \{m\})$, telle que

$$\Sigma|_{\mathcal{X}_m \times \mathcal{X}_m} = \sigma_m \quad \text{et} \quad \Sigma((x, p), (y, q)) = c_{p,q} \quad (p \neq q).$$

On obtient ainsi une fonction $\Sigma: \mathcal{U} \times \mathcal{U} \rightarrow [0,1]$ (ou \mathbb{R}^+), **uniformément continue** sur la distance δ définie par bloc, offrant un cadre d'unification pour l'**auto-organisation** dans un seul **Synergistic Connection Network (SCN)**.

La démonstration s'appuie sur la **réunion disjointe** et la définition d'une distance globale δ qui maintient une séparation stricte (Δ_0) entre blocs \mathcal{X}_p et \mathcal{X}_q pour $p \neq q$. La construction assure que

Σ est **intra-bloc** égale à σ_m , et **inter-bloc** égale à une constante (souvent zéro). On en déduit que Σ est **uniformément continue**, ce qui garantit la possibilité de réaliser une **analyse de stabilité** ou de convergence si l'on souhaite faire évoluer un réseau global.

2.2.1.2. Qu'est-ce que $S(\mathcal{E}_i, \mathcal{E}_j)$? Exemples (distance, similarité, co-information)

Pour décrire la **synergie** reliant deux entités \mathcal{E}_i et \mathcal{E}_j dans le **Deep Synergy Learning (DSL)**, on introduit une **fonction** $S(\mathcal{E}_i, \mathcal{E}_j)$ qui, pour chaque couple d'entités, mesure leur **coopération** ou leur **affinité**. On formalise souvent ce concept par

$$S: \quad \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+,$$

où \mathcal{X} représente l'espace (ou la **réunion** d'espaces) hébergeant les entités $\mathcal{E}_i, \mathcal{E}_j$. Selon la **nature** de \mathcal{X} et les **objectifs** (clustering, fusion multimodale, etc.), la fonction S peut être fondée sur différents concepts (distance, similarité, information mutuelle). Les développements ci-après (2.2.1.2.1 à 2.2.1.2.3) exemplifient trois grandes **familles** de mesures couramment mises en avant dans le DSL.

A. Distance

Lorsque les entités \mathcal{E}_i et \mathcal{E}_j appartiennent à un **espace métrique** (\mathcal{X}, d) , on peut définir la synergie comme une **fonction** décroissante de la distance :

$$S(\mathcal{E}_i, \mathcal{E}_j) = f(d(\mathcal{E}_i, \mathcal{E}_j)),$$

avec, par exemple, $f(x) = 1/(1+x)$ ou $f(x) = \exp(-\alpha x)$. L'intuition repose sur l'idée qu'une faible **distance** implique une **coopération** plus marquée. Dans un cadre vectoriel \mathbb{R}^d , si l'on prend la distance euclidienne

$$d(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{x}_i - \mathbf{x}_j \|,$$

on peut fixer

$$S(\mathcal{E}_i, \mathcal{E}_j) = \exp(-\alpha \| \mathbf{x}_i - \mathbf{x}_j \|),$$

de sorte que deux entités **proches** dans \mathbb{R}^d obtiennent un score S élevé, et deux entités **éloignées** un score proche de zéro. Sur le plan des **propriétés**, la distance garantit la **symétrie** $S(\mathcal{E}_i, \mathcal{E}_j) = S(\mathcal{E}_j, \mathcal{E}_i)$ si le choix de la fonction f respecte cette invariance.

B. Similarité ou Corrélation

Une autre façon de concevoir la **synergie** privilégie l'idée de **similarité** ou de **corrélation** statistique. Par exemple, si \mathcal{E}_i et \mathcal{E}_j sont deux vecteurs $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$, on peut recourir à la **similarité cosinus** :

$$S(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}.$$

Dans ce cas, plus l'**angle** entre les vecteurs est faible, plus la valeur de S est grande. Pour des variables **aléatoires** X, Y, on peut choisir le **coefficient de corrélation** $\rho(X, Y)$, par exemple la **corrélation de Pearson**, ou toute autre mesure similaire. D'un point de vue algorithmique, le **DSL** exploitera alors ces scores comme un indicateur de synergie où lorsque deux entités se révèlent linéairement (ou faiblement) interdépendantes, leur lien $\omega_{i,j}$ se renforce modérément et lorsqu'elles affichent une corrélation élevée, le **renforcement** sera marqué. Cette logique, conforme aux paradigmes associatifs (Hopfield) ou hebbiens, consolide l'idée que la **co-activation** façonne le **réseau** en intensifiant les liaisons utiles.

C. Information Mutuelle ou Co-Information

Dans des situations où les entités $\mathcal{E}_i, \mathcal{E}_j$ représentent des **variables aléatoires** (ou des distributions de probabilité), une approche consiste à mesurer la **synergie** via l'**information mutuelle** :

$$S(\mathcal{E}_i, \mathcal{E}_j) = I(\mathcal{E}_i; \mathcal{E}_j) = \int p(x, y) \log[p(x, y)/p(x)p(y)] dx dy.$$

Cette grandeur quantifie la **dépendance** entre deux variables où plus l'une réduit l'incertitude sur l'autre, plus leur information mutuelle est élevée. Dans un cadre plus général, on parle de **co-information** $I(\mathcal{E}_1; \mathcal{E}_2; \dots; \mathcal{E}_k)$ lorsque plusieurs variables interagissent collectivement (voir Chapitre 12). Sur le plan **théorique**, l'information mutuelle a l'avantage de capturer des dépendances **non linéaires**, dépassant la simple corrélation. Si $S(\mathcal{E}_i, \mathcal{E}_j)$ est défini comme $I(\mathcal{E}_i, \mathcal{E}_j)$, alors la valeur s'annule en cas d'indépendance et s'accroît avec la force de la dépendance. Dans le **DSL**, ce schéma se prête bien à un **renforcement** de $\omega_{i,j}$ lorsque les variables s'avèrent interdépendantes.

Toutefois, la mise en pratique de l'information mutuelle peut exiger des **estimations** ou des **approximations** si l'on ne connaît pas explicitement les distributions $p(x)$. Des estimateurs basés sur des techniques de **binning**, de **k plus proches voisins** ou des méthodes paramétriques (ex. gaussiennes) sont souvent employés. Les performances et la fiabilité de ces estimateurs conditionnent alors la précision de la fonction S.

2.2.1.3. Relation entre cette Fonction et la Mesure d'Utilité Mutuelle ou d'Interaction

Après l'exposé des divers exemples (distance, similarité, information mutuelle) pour la **fonction de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$, il importe d'analyser la **signification** profonde de ce score dans le **Deep Synergy Learning (DSL)**. Cette section [2.2.1.3] aborde la connexion existante entre S et des concepts plus généraux d'**utilité mutuelle** ou d'**interaction** bénéfique, en formalisant l'idée que la **synergie** reflète une forme de "plus-value" commune aux entités \mathcal{E}_i et \mathcal{E}_j .

A. Notion de Bénéfice Mutuel et d'Utilité

Considérons un **réseau** adaptatif où l'**interaction** ($\mathcal{E}_i, \mathcal{E}_j$) procure un **gain** à la structure globale, par exemple la reconnaissance partagée d'un même cluster ou la contribution à une tâche identique. On peut **formaliser** cette idée au moyen d'une "utilité" $\text{Util}(\mathcal{E}_i, \mathcal{E}_j)$, envisagée comme une fonction $\text{Util}: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$. Il s'agit de quantifier dans quelle mesure la **relation** entre i et j se révèle profitable. Dans le DSL, il arrive qu'on assimile

$$S(\mathcal{E}_i, \mathcal{E}_j) = \text{Util}(\mathcal{E}_i, \mathcal{E}_j)$$

ou qu'on emploie une transformation monotone pour s'assurer que $S \in [0,1]$. L'essentiel est que plus l'**utilité** perçue est grande, plus la **synergie** S l'est aussi. Dans un cadre **non supervisé**, la reconnaissance d'une dépendance mutuelle forte (distance faible, corrélation élevée, information partagée importante) implique un **bénéfice** pour le réseau, lequel renforce alors le lien $\omega_{i,j}$.

Exemple : Lorsque la **synergie** consiste en l'**information mutuelle** $I(X; Y)$, la "diminution d'incertitude" qu'apporte X sur Y s'interprète comme un **gain** (une "utilité"), d'où le renforcement de $\omega_{i,j}$. Si la synergie repose sur une **similarité** positive (ex. $\exp(-\alpha d(\mathbf{x}_i, \mathbf{x}_j))$), l'**utilité** réside dans la facilité ou le bénéfice de rassembler deux entités $\mathbf{x}_i, \mathbf{x}_j$.

B. Lien avec la Coopération et l'Émergence de Clusters

Dans un **réseau** où les liaisons $\omega_{i,j}$ se voient renforcées lorsque $S(\mathcal{E}_i, \mathcal{E}_j)$ est élevée, on retrouve la **logique** de la théorie de la coopération où les unités qui coopèrent s'avèrent plus **connectées** et, par ricochet, le **système** global y gagne. Sur le plan algébrique, des paires (i, j) pour lesquelles $S(i, j)$ reste notable finissent par tisser un **sous-réseau** (ou cluster) de pondérations fortes $\omega_{i,j}$.

Cette vision recoupe la notion d'**utilité** où regrouper des entités "compatibles" ou "enrichissantes" apparaît comme une stratégie d'amélioration globale (stratégie de "somme positive"). On peut d'ailleurs ajouter une perspective énergétique où l'on considère la somme (ou l'opposé)

$$\sum_{i < j} \omega_{i,j} S(\mathcal{E}_i, \mathcal{E}_j)$$

en tant que **critère** à maximiser, ce qui amène le réseau à rechercher la configuration $\{\omega_{i,j}\}$ la plus profitable pour l'ensemble. Les chapitres 4 et 7 présenteront des **théorèmes** expliquant comment, sous certaines hypothèses (symétrie de la synergie, bornes sur η, τ), la dynamique converge vers un **minimum local** stable, c.-à-d. une **partition** en clusters où les liens internes sont forts.

C. Interaction au-delà de la Simple Corrélation

Dans certains cas, l'**interaction** entre deux entités outrepasse la corrélation linéaire (Pearson). Un **lien** non linéaire ou conditionnel peut se révéler plus important. De même, l'**information mutuelle** ou la **co-information** peut détecter des dépendances qui échappent à un modèle linéaire. Si \mathcal{E}_i et \mathcal{E}_j entretiennent un rapport non trivial (exemple : $Y = X^2 + \text{bruit}$), la corrélation peut être faible, tandis que l'information mutuelle pointe une dépendance claire. Dans un cadre plus vaste encore,

si $\{\mathcal{E}_1, \dots, \mathcal{E}_k\}$ partagent une interaction collective, on peut imaginer une **synergie n-aire** (voir Chapitre 12).

Le **DSL** entend **s'adapter** à de telles configurations. La fonction $S(\mathcal{E}_i, \mathcal{E}_j)$ doit donc pouvoir accepter une grande variété de mesures d’“utilité” ou de co-dépendance. Cette **flexibilité** fait de ce paradigme un cadre général, capable d’intégrer tantôt une complémentarité sémantique, tantôt une co-occurrence sensorielle, etc.

D. Théorème (Existence d'un Score d'Interaction)

Considérons un ensemble d’entités $\{\mathcal{E}_i\}$. Supposons qu’on possède pour chaque paire (i, j) une mesure d’**interaction** $\text{Int}(\mathcal{E}_i, \mathcal{E}_j) \geq 0$ reflétant la co-occurrence ou la dépendance. Il est possible de construire une fonction de **synergie** S grâce à une transformation monotone ϕ . Concrètement, on définit

$$S(\mathcal{E}_i, \mathcal{E}_j) = \phi(\text{Int}(\mathcal{E}_i, \mathcal{E}_j)),$$

où ϕ est une application $[0, +\infty) \rightarrow [0, 1]$ (ou \mathbb{R}^+), strictement croissante et vérifiant $\phi(0) = 0$. Ainsi, chaque valeur d’interaction $\text{Int}(i, j)$ se convertit en un **score** $S(i, j)$ strictement **non négatif**, aligné sur l’**ordre** (plus Int augmente, plus S augmente). Cette construction prouve qu’il existe un moyen d’englober toute forme de mesure d’utilité mutuelle dans la **logique** du DSL, tant que la mesure est monotone et non négative.

2.2.2. Mise à Jour des Pondérations : Formule Générale

Le **Deep Synergy Learning (DSL)** s’appuie sur un réseau d’entités $\{\mathcal{E}_i\}$ dont les liaisons $\omega_{i,j}$ évoluent en fonction d’une **fonction de synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. La spécificité du DSL réside dans le fait que ces pondérations ne sont pas fixées ni mises à jour par un unique label externe, mais bien par un **processus local** qui adapte la force de chaque lien $\omega_{i,j}$ selon la valeur de synergie mesurée entre \mathcal{E}_i et \mathcal{E}_j .

Dans cette section (2.2.2), on formalise la **règle de mise à jour** la plus courante dans la littérature DSL où une **équation linéaire-additive**, souvent comparée aux règles de Hebb modifiées ou aux dynamiques d’un réseau associatif, est employée.

2.2.2.1. Équation $\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i, j) - \tau \omega_{ij}(t)]$

Dans le cadre du **Deep Synergy Learning (DSL)**, on envisage un **indice de temps** discret $t = 0, 1, 2, \dots$, ainsi qu’une matrice de pondérations $\{\omega_{i,j}(t)\}$. La mise à jour de ces pondérations, à chaque itération, s’effectue selon la formule

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i, j) - \tau \omega_{ij}(t)].$$

Les différentes composantes de cette équation se lisent comme suit :

- $\omega_{ij}(t)$ désigne la **force** (ou le poids) de la liaison entre les entités \mathcal{E}_i et \mathcal{E}_j à l'instant t .
- $S(i,j)$ désigne la **synergie** (positive ou nulle) entre \mathcal{E}_i et \mathcal{E}_j .
- $\eta > 0$ est un **taux d'apprentissage** local, traduisant la rapidité d'ajustement.
- $\tau > 0$ correspond à un **coefficent de décroissance** (ou de dissipation) qui prévient la croissance indéfinie des pondérations.

Cette équation se décompose en deux mécanismes complémentaires où **renforcement** la pondération ω_{ij} subit un incrément $\eta S(i,j)$. Si la synergie $S(i,j)$ se révèle élevée, la liaison ω_{ij} s'accroît et **décroissance** le terme $\eta \tau \omega_{ij}(t)$ vient en déduction, évitant ainsi que les poids ω_{ij} ne divergent et ne deviennent arbitrairement grands.

On peut reformuler l'expression en

$$\omega_{ij}(t+1) = (1 - \eta \tau) \omega_{ij}(t) + \eta S(i,j).$$

Cette représentation met en évidence une **combinaison linéaire** de l'ancienne valeur $\omega_{ij}(t)$, multipliée par $(1 - \eta \tau)$, et d'un **terme constant** $\eta S(i,j)$. Pour l'analyse de la convergence, on peut résoudre l'équation stationnaire

$$\omega_{ij}^* = (1 - \eta \tau) \omega_{ij}^* + \eta S(i,j),$$

qui donne

$$\omega_{ij}^* = \frac{\eta}{\eta \tau} S(i,j) = \frac{S(i,j)}{\tau}.$$

Cette solution indique que, si l'on itère la mise à jour et si les valeurs de η et τ sont suffisamment réduites ou bien paramétrées, $\omega_{ij}(t)$ se **stabilise** autour d'un point d'équilibre $\frac{S(i,j)}{\tau}$. Dans les chapitres suivants — notamment [Chapitre 4](#) et [Chapitre 7](#) — on examinera plus avant la **convergence** et la façon dont cette dynamique collective (appliquée simultanément à toutes les paires (i,j)) peut conduire à la formation de **clusters**.

2.2.2.2. Variantes : Inhibition Compétitive, Mécanisme Multiplicatif vs. Additif

La **formule générale** de mise à jour

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)] \quad (\text{voir section [2.2.2.1]})$$

illustre le **cas additif**, dans lequel on ajoute un terme $\eta S(i,j)$ et on retire $\eta \tau \omega_{ij}(t)$. Dans la pratique du **Deep Synergy Learning (DSL)**, on rencontre toutefois des **mécanismes de régulation supplémentaires** ou alternatifs, qui modifient la nature de l'adaptation des poids ω_{ij} . Deux **familles de variantes** se distinguent où l'**inhibition compétitive** et le **mécanisme multiplicatif** prennent le relais plutôt qu'une simple mise à jour de type linéaire-additif.

A. Inhibition Compétitive

La **mise à jour linéaire** du poids ω_{ij} présentée en section 2.2.2.1 traite chaque liaison de façon **indépendante**. Or, il est possible d'introduire une **inhibition compétitive**, c'est-à-dire un **effet d'interaction** entre différents liens connectés à la même entité, de sorte que la croissance de certains empêche ou restreigne la croissance d'autres. Dans un réseau biologique, ce phénomène correspond à l'**inhibition latérale** où un neurone fortement actif peut inhiber la plasticité d'autres connexions concurrentes ou proches.

Dans le **DSL**, l'inhibition compétitive peut prendre la forme de **contraintes** ou de **termes** interagissant avec ω_{ij} . Deux approches classiques :

- **Somme limitée** :

$$\sum_j \omega_{ij}(t) \leq \Omega_{\max},$$

imposant qu'une entité \mathcal{E}_i ne puisse maintenir qu'un nombre total (ou une somme totale) de liaisons ω_{ij} , au-dessus d'un certain seuil.

- **Terme d'inhibition** introduit directement dans la règle de mise à jour :

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \left[S(i,j) - \tau \omega_{ij}(t) - \sum_{k \neq j} \gamma_{k,j} \omega_{ik}(t) \right],$$

où $\gamma_{k,j}$ pondère l'**inhibition** exercée par le lien ω_{ik} sur la liaison ω_{ij} . Ainsi, plus un **nœud** \mathcal{E}_i a de liaisons puissantes, plus il “freine” la progression de nouvelles connexions.

Cette inhibition compétitive favorise la **sparcification** du réseau et l'apparition de **clusters** plus nets. Même si la synergie $S(i,j)$ conserve une valeur positive, la liaison ω_{ij} peut être contrainte de **décroître** ou de stagner si la “capacité” du nœud \mathcal{E}_i (ou \mathcal{E}_j) est déjà mobilisée par d'autres liens plus cruciaux. On retrouve ce principe dans divers modèles biologiques et algorithmes inspirés de la neurophysiologie, où il améliore l'**efficacité** de l'apprentissage en évitant un maillage trop complet ou trop diffus du réseau.

B. Mécanisme Multiplicatif vs. Additif

La règle générale

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)]$$

relève d'une **dynamique additive** où on incrémente ω_{ij} d'un terme $\eta S(i,j)$ et on décrémente d'un terme $\eta \tau \omega_{ij}(t)$. Dans certaines approches du **DSL**, on priviliege un **mécanisme multiplicatif** où l'augmentation ou la diminution du poids ω_{ij} s'effectue de façon **proportionnelle** à sa valeur courante. Par exemple, on peut adopter une formule telle que

$$\omega_{ij}(t+1) = \omega_{ij}(t) \times (1 - \alpha \tau) + (\text{termes multiplicatifs de } S(i,j)),$$

ou, plus radicalement,

$$\omega_{ij}(t+1) = \omega_{ij}(t) \exp(\alpha[S(i,j) - \tau \omega_{ij}(t)]).$$

Ce style de mise à jour rappelle la **règle de Hebb** où “plus un poids est grand, plus il se renforce lors de co-activation”. Il favorise un **effet exponentiel** où un lien déjà élevé grandit rapidement si la synergie $S(i,j)$ reste soutenue, tandis qu’un lien faible ne se développe que modérément.

Sur le plan algorithmique, cela peut accélérer la **défferentiation** entre liens “dominants” et “inactifs”, donnant lieu à un phénomène “winner-takes-most”. Même si, dans son essence, la logique multiplicative conserve l’idée d’ajuster $\omega_{ij}(t)$ en fonction de $S(i,j)$, elle introduit un **degré** de non-linéarité plus important. La croissance (ou décroissance) dépend non seulement de $S(i,j)$ mais aussi de la magnitude actuelle de ω_{ij} .

2.2.2.3. Interprétation : Plasticité Locale, Renforcement/Désactivation de Liens

La **mise à jour** des poids $\omega_{i,j}(t)$ guidée par la **synergie** $S(i,j)$ peut s’interpréter comme un **mécanisme de plasticité** analogue aux dynamiques neuronales ou aux paradigmes d’auto-organisation. Dans ce cadre, chaque liaison $\omega_{i,j}$ se **renforce** ou s’**affaiblit** de manière autonome, en fonction de la valeur de synergie mesurée entre les entités \mathcal{E}_i et \mathcal{E}_j , et sous l’impact d’un terme de décroissance régulateur. Cette section [2.2.2.3] propose une **lecture** (à la fois biologique et algorithmique) de cette dynamique, généralement décrite comme un **processus local** de renforcement/désactivation, fondé sur la co-occurrence ou l’interaction directe entre entités.

A. Plasticité Locale

Dans les **réseaux** neuronaux biologiques, la **plasticité** évoque la capacité d’une synapse à ajuster sa “force” en réaction à l’activité simultanée (ou corrélée) de ses neurones pré- et post-synaptiques. L’équation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$$

(voir section 2.2.2.1) illustre cette logique où si la **synergie** $S(i,j)$ est élevée, elle accroît la liaison $\omega_{i,j}$ d’un terme $\eta S(i,j)$.

Cette **variation** $\Delta\omega_{i,j}$ dépend exclusivement de $\omega_{i,j}(t)$ et de la valeur $S(i,j)$, conformément à la philosophie **locale** de la plasticité où il n’y a ni connaissance **globale** ni label supervisé. C’est précisément l’**auto-organisation** où l’ajustement s’opère localement et la configuration **globale** (éventuelle formation de clusters) émane de l’ensemble de ces modifications éparses. Dans un scénario DSL standard, on découpe le temps en itérations $t = 0,1,2, \dots$. Chaque itération applique la règle de plasticité. Dans une version plus biologisante, on passerait en **temps continu** :

$$\frac{d \omega_{i,j}}{dt} = \alpha[S(i,j) - \tau \omega_{i,j}],$$

ce qui revient conceptuellement à la même logique et, en l'absence de perturbations, tend vers l'équilibre $\omega_{i,j}^* \approx S(i,j)/\tau$.

B. Renforcement vs. Désactivation

Si la quantité $\tau \omega_{i,j}(t)$ reste **inférieure** à $S(i,j)$, alors la mise à jour $\Delta\omega_{i,j} = \eta[S(i,j) - \tau \omega_{i,j}(t)]$ s'avère **positive** où le poids $\omega_{i,j}$ croît.

Cela signifie que, tant que la **synergie** $S(i,j)$ surpassé un “seuil relatif” $\tau \omega_{i,j}(t)$, la liaison se **renforce** pas à pas. De multiples itérations rapprochent $\omega_{i,j}$ d'un **équilibre** $\omega_{i,j}^* \approx S(i,j)/\tau$. Si, en revanche, $\tau \omega_{i,j}(t)$ dépasse $S(i,j)$, la variation $\Delta\omega_{i,j}$ devient **négative**. Dans ce cas, la connexion $\omega_{i,j}$ se **décroît** progressivement, trahissant une **absence** ou un affaiblissement de la synergie justifiant un grand poids. À la longue, $\omega_{i,j}$ peut tendre vers une valeur très faible ou proche de zéro, synonyme de désactivation.

Cette dynamique explique pourquoi $\omega_{i,j}^* = S(i,j)/\tau$ incarne un **point d'équilibre** local où le renforcement et la décroissance s'y compensent exactement.

C. Implications pour la Structure Émergente

Lorsque certains liens $\omega_{i,j}$ s'avèrent renforcés où la synergie $S(i,j)$ est élevée et que d'autres s'amenuisent en raison d'une faible synergie ou d'une pénalisation trop forte, le **réseau** voit se constituer des **sous-groupes** où des **clusters** (ou micro-clusters) fortement interconnectés émergent. Cela correspond à la formation d'un **état stable** où les entités \mathcal{E}_i et \mathcal{E}_j présentant une synergie notable sont reliées par des pondérations robustes.

Cette approche ne se borne pas à une structure figée, puisque si les **synergies** $\{S(i,j)\}$ évoluent (par exemple si le système reçoit de nouvelles entités, ou si les co-occurrences se modifient), la **plasticité locale** réactualise $\omega_{i,j}$. Le réseau suit donc les **fluctuations** du flux, créant ou dissolvant des connexions selon la tendance du moment (cf. Chapitre 9 sur l'apprentissage continu). Dans la sphère **biologique**, cette logique de co-activation et d'**adaptation** des liaisons renvoie à la **règle de Hebb** ou à d'autres mécanismes (STDP) où les synapses se renforcent si leurs neurones sont fortement corrélés. En IA non supervisée, ces principes rejoignent ceux des **réseaux associatifs**, mais le **DSL** porte l'idée encore plus loin en permettant des fonctions de synergie plus complexes (mesures multimodales, co-information...).

Conclusion sur la Plasticité Locale

La règle

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$$

peut se lire comme un **processus de plasticité** locale, où la synergie S tient le rôle d'un “signal de co-opération”. Les liens présentant un **score** $S(i,j)$ élevé se **renforcent**, tandis que ceux dépourvus de synergie s'amenuisent ou se désactivent (tendent vers zéro).

De fait, un réseau DSL s'oriente vers une **sélection** continue des connexions jugées utiles, maintenant une certaine **résilience** (on peut réactiver un lien si la synergie remonte ultérieurement).

Les chapitres ultérieurs examineront la **convergence** de ces poids et la **stabilisation** d'une structure globale (clusters, attracteurs). Ils aborderont également dans la section 2.2.2.2 les **variantes** où l'inhibition compétitive introduit une interaction entre différents liens et la mise à jour multiplicative renforce l'effet "winner-takes-most". L'ensemble de ces points ancre la **philosophie** du Deep Synergy Learning où une **plasticité locale** pilotée par la synergie favorise l'**émergence** de configurations organisées dans le réseau.

2.2.3. Règles de Parsimonie et Seuil de Connexion

Au fur et à mesure que la **mise à jour** (section 2.2.2) renforce certains liens $\omega_{i,j}$ selon la synergie $S(i,j)$, le **réseau** risque de voir ses pondérations croître un peu partout. Or, dans de nombreux scénarios — qu'il s'agisse de **biologie** (réseaux neuronaux) ou de **DSL** pratique (auto-organisation hétérogène) —, il est souhaitable d'éviter un **graphe trop dense**, qui peut entraîner :

- Une **surcharge** de calcul (trop de liaisons à gérer, complexité accrue),
- Un **manque** de différenciation (le réseau devient quasi complet, il est difficile d'identifier de vrais clusters distincts),
- Des **effets** d'instabilité ou de bruit (des liens résiduels de faible poids saturent inutilement la structure).

La notion de **parsimonie** répond à cela où on ajoute des **règles** (post-traitements ou intégrées à la mise à jour) qui suppriment ou inhibent les liaisons trop faibles, favorisant une structure plus éparsé et plus significative.

2.2.3.1. Comment Éviter un Graphe Trop Dense : Suppression de Liens sous un Certain ω_{\min}

Dans l'application pratique du **Deep Synergy Learning**, on adopte souvent un **seuil** $\omega_{\min} > 0$ pour **restreindre** la densité du réseau. Concrètement, après chaque itération ou à des intervalles déterminés, on **supprime** (ou met à 0) les liaisons dont les pondérations $\omega_{i,j}$ demeurent **inférieures** à ω_{\min} . La règle s'écrit :

$$\omega_{i,j}(t+1) = \begin{cases} \omega_{i,j}(t+1), & \text{si } \omega_{i,j}(t+1) \geq \omega_{\min}, \\ 0, & \text{sinon.} \end{cases}$$

Ainsi, les liaisons trop faibles $\omega_{i,j} < \omega_{\min}$ sont ramenées à zéro, puis (en fonction de la stratégie adoptée) ne sont plus mises à jour ou restent inactives jusqu'à un éventuel regain de **synergie**. En procédant ainsi, on obtient un **réseau** plus **clairsemé**, simplifiant l'identification de **clusters** et la visualisation globale. Sur le plan algorithmique, la démarche se résume à parcourir la matrice ω après un certain nombre d'itérations (ou après chaque "epoch") pour **couper** les liaisons trop faibles, ce qui libère les ressources de calcul pour les liens plus importants.

Cette suppression de liens de force négligeable rappelle le **synaptic pruning** dans les réseaux neuronaux biologiques, où des synapses à faible activité finissent par être éliminées, permettant ainsi de recentrer l'efficacité du neurone sur les connexions plus utiles. Dans l'optique d'un **DSL**, un tel mécanisme garantit la **parsimonie** où le réseau se concentre sur les liaisons suffisamment

pertinentes, évitant de multiplier les connexions de poids quasi nul. Il existe plusieurs variantes pour implanter ce principe :

- **Seuil unique** ω_{\min} .
- **Seuils multiples** $\omega_{\min 1}, \omega_{\min 2} \dots$, autorisant un tri par “niveaux” de liens.
- **Pourcentage fixe** : ne conserver que les $k\%$ de liaisons les plus fortes.

Ces méthodes se montrent particulièrement **bénéfiques** quand le nombre d’entités n devient grand, car la **matrice** $\{\omega_{i,j}\}$ peut atteindre une taille $O(n^2)$. En réduisant la densité des connexions actives, on **allège** la complexité computationnelle et on **accélère** la convergence de la structure.

2.2.3.2. Effets sur la Structure Émergente (Clusters plus Nets, Stabilisation plus Rapide)

Lorsque la **règle de parsimonie** (section 2.2.3.1) impose de couper ou de désactiver les liaisons $\omega_{i,j}$ inférieures à un certain seuil ω_{\min} , la **topologie** du Synergistic Connection Network (SCN) se réoriente vers un réseau plus **clairsemé**. Ce filtrage sélectif engendre plusieurs **conséquences** majeures sur la **structure** formée et sur la **dynamique** de stabilisation qui en découle.

A. Clusters plus Nets

En retranchant systématiquement les liaisons $\omega_{i,j}$ dont la valeur demeure **en deçà** de ω_{\min} , on élimine de fait les **ponts** minimes ou les connexions relativement faibles qui ne concourent pas sensiblement à la structure globale. Les liens dont $\omega_{i,j}$ dépasse ω_{\min} deviennent alors les **véritables connexions**, celles qui suggèrent une **synergie** réelle entre les entités \mathcal{E}_i et \mathcal{E}_j .

Dans une démarche d'**auto-organisation**, on recherche souvent des **clusters** de nœuds fortement interconnectés. La présence d’un seuil ω_{\min} clarifie et accentue la séparation entre ces groupements où les liaisons inter-groupes, si elles restent faibles, se trouvent coupées tandis que les liens intra-groupe, favorisés par une **synergie** non négligeable, franchissent le cap du seuil. Cette **polarisation** entre liens conservés et liens supprimés renforce la visibilité des **composantes** ou des **communautés** dans le SCN.

Par ailleurs, les liaisons de faible intensité pouvaient osciller au cours des itérations, sans réellement influer sur l’architecture. Les supprimer réduit le risque d'**agrégation** inutile de micro-connexions parasites, et limite par conséquent la formation de signaux aléatoires qui alourdiraient le réseau. De fait, cette **coupe** s’apparente à un **filtrage** où elle fait ressortir les liens les plus significatifs et, par extension, rend la structure plus facilement interprétable.

B. Stabilisation plus Rapide

Le fait de mettre à zéro une bonne partie des liaisons $\omega_{i,j}$ jugées trop faibles diminue la **charge** dans la dynamique du réseau. Dès lors, seuls subsistent un nombre restreint de poids **actifs**, sur lesquels se concentrent les mises à jour. En conséquence, la boucle de réajustement se révèle plus **efficace** où il y a moins d’interactions à recalculer et beaucoup d’anciennes connexions, désormais à 0, ne requièrent plus de suivi continu.

Si, de surcroît, les interactions inter-blocs (ou inter-clusters) étaient de toute façon faibles, elles pouvaient allonger la **convergence** en imposant de petits allers-retours de pondérations. Leur coupure accélère la stabilisation, laissant chaque cluster se structurer **indépendamment**, sans perturbations marginales.

Du point de vue purement algorithmique, si la mise à jour d'une liaison $\omega_{i,j}$ s'exprime (cf. section 2.2.2.1) par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

la suppression régulière des liens $\omega_{i,j}$ restés en dessous de ω_{\min} apporte une plus grande **stabilité** où une fois coupées, ces connexions ne fluctuent plus et les liens restants convergent généralement plus vite vers leur point d'équilibre $\omega_{i,j}^* \approx S(i,j)/\tau$. De nombreux travaux empiriques montrent que l'**auto-organisation** se fait plus rapidement quand on élimine régulièrement les connexions trop faibles.

2.2.3.3. Impact sur la Complexité : Gain Potentiel vs. Perte d'Information

Dans le contexte d'un **Synergistic Connection Network** (SCN), l'application d'une règle de **parsimonie** (voir section 2.2.3.1) conduit à **supprimer** ou à **mettre à zéro** les liaisons $\omega_{i,j}$ dont la valeur reste inférieure à un certain seuil ω_{\min} . Lorsque cette procédure est mise en œuvre, le **réseau** acquiert une structure plus **clairsemée** où nombre de connexions sont annihilées, seules subsistant celles qui dépassent ω_{\min} . Cette raréfaction des liens exerce une influence directe sur la **complexité** de l'architecture, ainsi que sur la **qualité** et l'**ampleur** de l'information qui transite dans le SCN. D'une part, la disparition de nombreuses liaisons apporte un **gain** notable en termes de densité et de coûts de calcul ; d'autre part, on risque de se priver de connexions faiblement positives susceptibles de s'avérer utiles. Le présent exposé souligne ces **deux facettes** — la simplification bienvenue et la perte potentielle — et examine les **stratégies** ou **réglages** classiquement envisagés pour concilier **parsimonie** et **préservation** de l'information.

A. Gains en Termes de Complexité

Il est fréquent de considérer un ensemble d'entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$. Si toutes forment un graphe complet, le nombre de liaisons $\omega_{i,j}$ à mettre à jour peut tendre vers $O(n^2)$. Imposer un **seuil** $\omega_{\min} > 0$ pour **couper** systématiquement les connexions dont la pondération reste au-dessous de ω_{\min} a pour effet :

$$\omega_{i,j}(t+1) = \begin{cases} \omega_{i,j}(t+1), & \text{si } \omega_{i,j}(t+1) \geq \omega_{\min}, \\ 0, & \text{sinon.} \end{cases}$$

Lorsque cette coupure est appliquée, la **densité** du réseau chute. De nombreuses paires (i, j) reçoivent désormais une pondération nulle et ne participent plus aux calculs, offrant une **réduction** de la complexité globale :

Complexité passera de $O(n^2)$ à $O(|\mathcal{A}|)$,

où \mathcal{A} est l'ensemble des **liaisons actives**, c'est-à-dire celles qui franchissent le seuil ω_{\min} . Si ce nombre se révèle beaucoup plus modeste que n^2 , la mise à jour se voit grandement **allégée**. Outre le gain algorithmique, ce filtrage favorise la **lisibilité** du réseau, par exemple pour la **détection** et la **visualisation** de clusters, en prévenant la formation d'un graphe trop dense. Il devient ainsi aisément d'identifier les sous-ensembles d'entités réellement interconnectées.

Avantages en un seul paragraphe. L'avantage majeur de cette approche réside dans la diminution drastique des liaisons à gérer, accélérant chaque itération de la mise à jour locale tout en clarifiant la topologie pour l'analyse des clusters. On évite de mobiliser un temps de calcul inutile sur des connexions dont la valeur de pondération reste marginale. Cette simplification s'avère particulièrement pertinente dans des contextes de grande dimension (big data, multimodalité), où la réduction de $O(n^2)$ à $O(n)$ ou $O(n \log n)$ peut faire la différence entre un algorithme inopérant et une solution réalisable en pratique.

B. Risque de Perte d'Information

Si la *parsimonie* via le seuil ω_{\min} procure un gain, elle entraîne le **risque de couper** des liaisons modérément positives qui auraient pu devenir utiles si on leur laissait le temps de se renforcer. Un lien $\omega_{i,j} = 0.02$ peut sembler négligeable à l'instant présent, mais il aurait pu se développer ultérieurement, si la **synergie** $S(i,j)$ augmentait pour des raisons liées à un flux de données changeant. Imposer $\omega_{i,j} \rightarrow 0$ quand $\omega_{i,j} < \omega_{\min}$ empêche de ressusciter la liaison à moins qu'on ne prévoie une procédure de "**réactivation**", tolérant le retour d'un lien si $S(i,j)$ devient subitement plus fort. Sans un tel mécanisme, la décision de couper est souvent définitive ou du moins très contraignante, ce qui peut figer la structure avant qu'une coopération tardive n'ait pu émerger.

Limites en un seul paragraphe. Cette coupure radicale peut aussi segmenter prématurément le réseau en multiples composantes déconnectées, s'il se trouve que la synergie reste faible mais non négligeable entre certains groupes. Un seuil trop ambitieux (ω_{\min} trop grand) peut donc conduire à une perte de richesse dans l'exploration des configurations, voire empêcher la coalescence ultérieure de clusters apparentés. Cela constitue la principale contrepartie de la parcimonie, tout en voulant gagner en efficacité, on sacrifie la possibilité de liens modestes mais porteurs d'une information subtile.

C. Stratégies de Coupure et Paramétrages

Il est crucial de trouver un **équilibre** entre l'**exigence** de parsimonie et la **volonté** de préserver les connexions potentielles. Plusieurs méthodes se révèlent utiles où l'**adaptation dynamique du seuil** ω_{\min} consiste à commencer avec une valeur modeste, puis à l'augmenter progressivement selon l'avancement de l'apprentissage. On peut également recourir à des **coupes partielles** où plutôt que d'annihiler un lien en dessous de ω_{\min} , on le réduit (ex. $\omega_{i,j} \rightarrow \gamma \omega_{i,j}$) afin de laisser une "chance" de relance. Des mécanismes de **réactivation** peuvent aussi se mettre en place si la synergie mesurée $S(i,j)$ s'élève de nouveau, réinstandant alors la liaison.

La littérature décrit également la possibilité de **conserver** un certain **pourcentage** de liens (5 % ou 10 % des poids maximaux), découpant le reste. Une autre approche consiste à procéder à des coupes plus progressives au fil des époques, ne retenant à chaque étape que les liens les plus élevés.

2.2.4. Notions d'États Internes et Auto-Organisation

Dans l'architecture d'un **Deep Synergy Learning (DSL)**, chaque entité \mathcal{E}_i est généralement décrite par son ancrage dans un espace \mathcal{X} (vecteur, symbole, distribution, etc.) et par le poids de ses liaisons $\{\omega_{i,j}\}$. Cependant, il est souvent utile d'aller plus loin en autorisant chaque entité à entretenir un **état local** — c'est-à-dire un ensemble de variables ou de "mémoire" qui évolue **en parallèle** de la dynamique des liaisons. Cette idée de "double composante" (poids inter-entités et état interne) contribue à renforcer la capacité d'**auto-organisation**, car elle ajoute un degré de plasticité propre à chaque entité, sans que tout dépende uniquement des pondérations $\omega_{i,j}$. La section (2.2.4) détaille comment ce mécanisme peut se mettre en place et quelles conséquences il a sur l'émergence de schémas cognitifs plus riches.

2.2.4.1. Possibilité pour Chaque Entité \mathcal{E}_i de Stocker un État Local, Évoluant en Parallèle

Il est souvent utile, dans un **Synergistic Connection Network (SCN)**, de ne pas cantonner chaque entité \mathcal{E}_i à une représentation fixe (par exemple, un simple vecteur \mathbf{x}_i demeurant invariant). On peut en effet postuler qu'elle possède un **état interne** $\mathbf{s}_i(t)$ qui se **modifie** au fil des itérations $t = 0, 1, 2, \dots$. Ainsi, l'évolution ne concerne plus uniquement les pondérations $\omega_{i,j}$ (voir la section 2.2.2), mais aussi le **cycle** propre de chaque entité.

A. Principe Général

On conserve la mise à jour des poids $\omega_{i,j}(t)$ conformément à la règle (voir section 2.2.2) :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \dots$$

En **parallèle**, chaque entité \mathcal{E}_i actualise son **état** $\mathbf{s}_i(t)$ en prenant en compte, par exemple, les informations issues du réseau ou de son activation interne. Il peut exister un ensemble d'équations décrivant la dynamique de l'état local :

$$\mathbf{s}_i(t+1) = \mathbf{F}_i(\mathbf{s}_i(t), \{\omega_{i,j}(t)\}, \{\mathbf{s}_j(t)\}, \dots).$$

La **fonction** \mathbf{F}_i dépend alors de l'état précédent $\mathbf{s}_i(t)$, des pondérations $\omega_{i,j}(t)$ reliant \mathcal{E}_i à ses voisins, et éventuellement d'autres signaux extérieurs. Comme dans toute **auto-organisation**, le principe demeure local, chaque entité mettant à jour son état sans qu'un contrôleur global ne dicte de consignes.

Ce postulat accroît sensiblement la **souplesse** du système. Les entités ne sont plus de simples "nœuds passifs" mais deviennent des **agents** disposant d'une **mémoire** ou d'un **contexte** interne. Elles peuvent ainsi **s'adapter** à l'historique, éventuellement conserver des traces de données observées précédemment, moduler la façon dont elles calculent leur synergie $S(\mathcal{E}_i, \mathcal{E}_j)$ vis-à-vis d'autres entités, ou encore faire varier leur "réceptivité" selon leur état $\mathbf{s}_i(t)$. Cela introduit un **couplage** (rétroaction) entre l'état de l'entité et la pondération $\omega_{i,j}$.

B. Richesse des Comportements

Le fait que les entités \mathcal{E}_i puissent stocker un état $\mathbf{s}_i(t)$ ouvre la voie à des dynamiques plus **complexes**, proches d'un système **multi-agents** ou de **modules cognitifs**. Par exemple, une entité pourrait se comporter en **LSTM local** (voir section 2.2.4.2) ou en bloc mémoire plus rudimentaire. Elle peut assimiler son historique, se doter d'un "contexte" conceptuel, ou enregistrer des **résidus** de perception. Cette organisation "double" (adaptation des poids $\omega_{i,j}$ d'un côté, adaptation des états $\mathbf{s}_i(t)$ de l'autre) rappelle certaines structures neuronales biologiques, où la **plasticité synaptique** se combine à l'activité **interne** des neurones (ou groupes de neurones).

Dans le **DSL**, cette évolution conjointe favorise l'émergence de schémas cognitifs distribués. Le réseau ne se limite plus à une **clusterisation** statique. Il peut générer de véritables **boucles** d'activité, des phénomènes de **résonance** ou de **synchronisation**, dans lesquels l'actualisation de $\mathbf{s}_i(t)$ influence la **synergie** $S(i, j)$, laquelle, à son tour, oriente la mise à jour $\omega_{i,j}(t)$. Ce double plan adaptatif donne un pouvoir expressif supplémentaire où un nœud \mathcal{E}_i peut modifier sa **sensibilité** ou son "profil" d'interaction selon ses signaux internes, aboutissant à une **co-organisation** encore plus riche.

On peut écrire qu'il existe deux niveaux de plasticité qui interagissent,

$$\begin{cases} \omega_{i,j}(t+1) = \omega_{i,j}(t) + \dots, \\ \mathbf{s}_i(t+1) = \mathbf{F}_i(\mathbf{s}_i(t), \{\omega_{i,\cdot}\}, \dots). \end{cases}$$

C'est cette combinaison qui confère au **DSL** son potentiel pour modéliser des fonctionnements plus proches d'une cognition distribuée.

2.2.4.2. Exemple : LSTM Local ou Petite Mémoire Interne

Lorsque l'on propose d'associer à chaque entité \mathcal{E}_i un état interne (se reporter à la section 2.2.4.1), plusieurs voies s'offrent à l'expérimentateur. Certaines approches recourent à un simple **accumulateur** ou à une **variable scalaire**, tandis que d'autres adoptent des dispositifs plus évolués, tels qu'une **cellule LSTM** (Long Short-Term Memory) inspirée des réseaux récurrents profonds. Dans toutes ces configurations, l'objectif demeure de doter chaque entité d'une **capacité à stocker** et **transformer** les informations de façon autonome, au-delà du seul ajustement des poids $\omega_{i,j}$. Les développements ci-dessous illustrent deux options où l'usage d'un **LSTM local** et la création d'une **petite mémoire interne**, moins élaborée mais déjà profitable, offrent des alternatives pertinentes.

A. LSTM Local au Sein d'un DSL

Un **LSTM** est une cellule récurrente conçue pour gérer des dépendances de long terme dans les réseaux neuronaux séquentiels. Elle incorpore :

- un état caché $\mathbf{h}(t)$ qui traduit la sortie instantanée de la cellule,
- un état mémoire $\mathbf{c}(t)$ où s'enregistrent des informations plus pérennes,

- plusieurs **portes** (input, forget, output) contrôlant l'apport, l'oubli et l'export de l'information contenue dans $\mathbf{c}(t)$ et $\mathbf{h}(t)$.

Les équations internes de l'LSTM déterminent l'évolution de $\mathbf{c}(t)$ et $\mathbf{h}(t)$ à partir d'une entrée $\mathbf{x}(t)$ et de l'état précédent $(\mathbf{h}(t - 1), \mathbf{c}(t - 1))$.

Dans le **Deep Synergy Learning**, on peut affecter à chaque entité \mathcal{E}_i sa **cellule LSTM** individuelle. Chaque entité conserve deux **états internes**, $\mathbf{h}_i(t)$ et $\mathbf{c}_i(t)$, qui décrivent respectivement l'état caché et la mémoire à long terme. Ces variables permettent de capturer une dynamique interne évolutive propre à chaque unité du réseau.

À chaque itération, l'entité \mathcal{E}_i reçoit un **signal** $\mathbf{x}_i(t)$, qui peut être un résumé de l'interaction issue de ses liaisons $\omega_{i,j}$ et des états $\mathbf{h}_j(t)$ de ses voisins. Elle met alors à jour ses états internes selon la relation

$$(\mathbf{h}_i(t), \mathbf{c}_i(t)) = \text{LSTM}(\mathbf{x}_i(t), \mathbf{h}_i(t - 1), \mathbf{c}_i(t - 1)).$$

Le **calcul** de la synergie $S(i, j)$ peut exploiter ces états internes, en les combinant à d'autres formes d'informations. Par exemple, une définition possible est

$$S(i, j) = \alpha \text{ sim}(\mathbf{h}_i(t), \mathbf{h}_j(t)) + (1 - \alpha) \text{ sim}(\mathbf{x}_i^0, \mathbf{x}_j^0),$$

où \mathbf{x}_i^0 et \mathbf{x}_j^0 sont des représentations plus statiques des entités \mathcal{E}_i et \mathcal{E}_j . L'**état LSTM** influence donc directement la mesure de synergie, enrichissant ainsi le cadre adaptatif du modèle.

Parallèlement, les pondérations $\omega_{i,j}$ évoluent simultanément, selon le schéma introduit en 2.2.2.1 ou ses variantes. Ce couplage entre la dynamique interne des entités et la plasticité des liaisons assure une adaptation continue du réseau, favorisant l'auto-organisation et l'émergence de structures cohérentes.

Sur le plan des **retombées**, un LSTM local assure une **rétenzione** d'information à plus long terme, permettant à l'entité \mathcal{E}_i d'**apprendre** et de s'ajuster en se fondant sur un historique étendu. Le réseau gagne en **souplesse**, car une liaison $\omega_{i,j}$ peut se consolider si les états $\mathbf{h}_i(t)$ et $\mathbf{h}_j(t)$ affichent une **similarité** grandissante. L'analogie avec la biologie neuronale (synapses + état interne) s'en trouve renforcée.

B. Petite Mémoire Interne : Variante Simplifiée

Toutes les implémentations du DSL ne requièrent pas la complexité d'un LSTM. Il est tout à fait envisageable d'utiliser une **mémoire** plus rudimentaire, par exemple un **scalaire** ou un **petit vecteur**, pour chaque entité \mathcal{E}_i . L'idée est de doter \mathcal{E}_i d'une **capacité** de rappel ou d'accumulation déjà suffisante pour enrichir les comportements auto-organisés.

Si $\mathbf{s}_i(t) \in \mathbb{R}$ (cas scalaire), on peut l'interpréter comme un accumulateur ou une trace glissante. On peut écrire :

$$\mathbf{s}_i(t + 1) = (1 - \beta) \mathbf{s}_i(t) + \beta \cdot \text{input}_i(t),$$

où $\text{input}_i(t)$ représente les signaux entrants (pondérations voisines, données brutes, etc.). Cette formule confère déjà une **mémoire** à l'entité, mais ne s'encombre pas de la sophistication d'un

LSTM.

Dans le cas vectoriel $\mathbf{s}_i(t) \in \mathbb{R}^d$, on applique la même logique à chaque composante, autorisant un peu plus de finesse dans la représentation.

Le calcul de la **synergie** $S(i, j)$ peut alors, en partie, reposer sur dist inverse $(\mathbf{s}_i(t), \mathbf{s}_j(t))$ ou un critère analogue, offrant un moyen de tenir compte des **états internes** dans la détermination de la coopération.

Dans bien des cas, cette approche simplifiée suffit pour ajouter un “volet temporel” ou “mémoire” au système, sans la complexité d’un LSTM complet. Toutefois, elle peut se montrer moins flexible (pas de portes d’oubli, etc.) et exiger un réglage manuel plus délicat.

C. Intérêt pour l’Auto-Organisation

Dans un **DSL**, on dispose déjà d’une **plasticité** au niveau des pondérations $\omega_{i,j}$. Ajouter une **mémoire** à chaque entité \mathcal{E}_i institue un second niveau de **plasticité** où, d’une part, les liaisons continuent d’évoluer selon la synergie et, d’autre part, les **entités** gèrent leur état $\mathbf{s}_i(t)$ en répercutant des observations passées ou des indices internes.

Cette double dynamique (liens + états) se rapproche des idées multi-agents, où chaque “agent” (une entité \mathcal{E}_i) possède un **état local** évolutif et des liaisons $\omega_{i,j}$ en constante régulation. Il en résulte des comportements plus évolués, incorporant une “intelligence” ou un “raisonnement” local distribué. Par exemple, une entité peut retenir dans \mathbf{s}_i la moyenne récente de signaux perçus, ajuster son calcul de $S(i, j)$ en conséquence, et influencer la mise à jour $\omega_{i,j}$. On voit émerger des phénomènes de synchronisation, de micro-réseaux spécialisés, voire de schémas cognitifs complexes.

2.2.4.3. Lien avec l’Émergence de Schémas Cognitifs plus Riches

Lorsque chaque entité \mathcal{E}_i se trouve associée à un **état interne** (voir section 2.2.4.1) et que cette **mémoire** ou **dynamique** interne vient compléter la logique de **synergie** décrite en 2.2.4.2, on constate que le **réseau** acquiert la capacité de développer, au fil du temps, des **schémas cognitifs** plus élaborés. Autrement dit, l’**auto-organisation** ne se limite plus à la simple configuration de pondérations $\omega_{i,j}$, mais peut susciter des motifs récurrents, des regroupements conceptuels ou des **comportements** distribués analogues à la “cognition distribuée” ou aux “systèmes multi-agents cognitifs”.

Dans un **DSL** classique, on admet déjà que les entités \mathcal{E}_i ajustent (via leurs représentations ou leurs états) leurs liaisons $\omega_{i,j}$, de sorte que celles-ci se **renforcent** si la synergie demeure significative. Dès lors que chaque entité se trouve dotée d’une **composante** mémorielle ou interne, la portée de l’**auto-organisation** s’étend où chaque \mathcal{E}_i est en mesure de conserver des informations antérieures au lieu de se cantonner à la valeur instantanée de la synergie. Par ce biais, elle peut moduler son calcul de $S(i, j)$ ou la manière dont elle relaie ses informations, en fonction d’un **état** interne $\mathbf{s}_i(t)$.

Le **réseau** ainsi construit ne se contente plus d’une clusterisation figée où il peut, de fait, développer de **véritables** réseaux de résonances, chaque entité, grâce à sa “mémoire” ou “conscience” locale,

participant à l'émergence de **motifs** globaux plus complexes. D'un point de vue mathématique, la dynamique ne se restreint plus à l'évolution des pondérations $\omega_{i,j}(t)$; elle englobe également un sous-système $\{\mathbf{s}_i(t)\}$ dont la synchronisation ou la mise en phase peut conduire à des *routines* ou *schémas* cognitifs. On qualifie souvent un système de "cognitif" quand il gère de l'**information** (que ce soit des perceptions ou des symboles) de manière flexible, qu'il dispose d'une **mémoire** plus ou moins durable et qu'il produit des **décisions** ou **comportements** appuyés sur cet historique interne.

Dans un **DSL** renforcé par des mécanismes de mémoire (section 2.2.4.2), chaque entité \mathcal{E}_i perçoit la synergie $\omega_{i,j}$ la reliant à des entités voisines, puis met à jour $\mathbf{s}_i(t)$ en combinant l'état antérieur et de nouvelles données. Elle propage ensuite ce nouvel état (ou un résumé) dans le **réseau**, influençant la future valeur de $S(i,j)$. Ce procédé, généralisé à l'ensemble des entités, assure un **mécanisme distribué** où l'**intelligence** ou la **cognition** ne naît pas d'un composant central, mais de la coordination d'un ensemble de nœuds cognitifs, chacun adaptant en continu sa mémoire et ses liens. Ainsi, on voit émerger des **dynamiques** de synchronisation, l'apparition de "leaders" et "followers", ou l'organisation spontanée de niveaux auto-gérés au sein d'un unique SCN.

Un *schéma* cognitif peut s'entendre comme un **pattern** récurrent, stable, servant à organiser le traitement de l'information dans le réseau. Par exemple, un schéma perceptif tient la forme d'une règle où "Lorsque je détecte une configuration **x**, j'enclenche la procédure **y**". Au sein d'un DSL où chaque entité possède son **état interne**, ce schéma peut s'exprimer via un **état collectif** stabilisé (plusieurs entités conservent des $\mathbf{s}_i(t)$ cohérents) associé à des liaisons $\omega_{i,j}$ robustes, c'est-à-dire un motif distribué dans le réseau. Alors que l'auto-organisation non supervisée demeure souvent cantonnée à la formation de **clusters** statiques, la présence d'états internes permet l'apparition de boucles d'activité, de séquences ou de boucles réverbérantes, rappelant la résonance neuronale dans les modèles biologiques. Cela dépasse la simple définition de classes, en autorisant la reproduction de schémas temporels ou la mise en œuvre de "routines cognitives" collectives.

Sur le plan de la **biologie neuronale**, on sait que la formation d'assemblées de neurones (règle de Hebb) peut inclure des états récurrents ou "réverbérants" dans lesquels chaque neurone conserve une certaine **trace** d'activité dépendant de son vécu antérieur. La **cognition** naît de la cohérence entre ces assemblées et de leurs interactions réciproques. Dans les **systèmes multi-agents**, si chaque agent possède un état local évoluant selon un script interne, c'est l'**intelligence collective** qui se développe de leur interaction, parfois vue comme un "protocole" distribué dans le SCN. On peut ainsi observer des phénomènes d'**émergence** de structures conceptuelles, de motifs cognitifs, voire de règles partiellement symboliques, dès lors que ces entités-agents échangent et actualisent leurs mémoires internes.

En combinant la plasticité des poids $\omega_{i,j}$ et celle de chaque **état** $\mathbf{s}_i(t)$, le **DSL** franchit le seuil d'une auto-organisation plus **riche**, où se déploient des *schémas cognitifs* allant au-delà du groupement en clusters. L'hypothèse est qu'un tel réseau "vivant", dont la mémoire réside à la fois dans les **liaisons** (pondérations) et dans les **états** de chaque nœud, peut développer des patterns de perception, de raisonnement ou d'apprentissage plus **complexes** et plus **adaptatifs**. C'est dans ce sens qu'il offre un **cadre** d'études et d'expérimentations susceptible d'éclairer l'émergence de **comportements** cognitifs distribués, rappelant ou imitant certains principes de la cognition humaine ou animale.

2.2.5. Exemples Illustratifs

Les principes théoriques exposés jusque-là — concernant la définition de la synergie S , la mise à jour des pondérations $\omega_{i,j}$, la parsimonie, ou encore la possibilité d'un état interne pour chaque entité — se concrétisent dans des **exemples** qui permettent de visualiser la **dynamique** du Deep Synergy Learning (DSL) de façon simple et didactique. La section (2.2.5) présente quelques scénarios illustrant ces mécanismes dans des configurations restreintes, afin de mieux comprendre :

- Comment s'applique la **règle de mise à jour** ($\omega_{ij}(t+1) = \dots$) dans un petit réseau,
- L'effet de la **similarité** (ou distance) sur la croissance ou la décroissance des liens,
- L'émergence de **micro-clusters** même dans des systèmes de taille modeste (3 ou 4 entités).

2.2.5.1. SCN Minimal (3 ou 4 Entités) : Démonstration de la Règle de Mise à Jour

Pour **illustrer** de façon simple la dynamique d'un **Synergistic Connection Network (SCN)** dans le cadre d'un **Deep Synergy Learning (DSL)**, on peut se limiter à un **réseau** très restreint de 3 ou 4 entités, par exemple $\{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \dots\}$. L'objectif est de **montrer** pas à pas comment les pondérations $\omega_{i,j}$ évoluent conformément à la **formule** générique expliquée en section 2.2.2 :

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta[S(i,j) - \tau \omega_{ij}(t)].$$

Cette expérience sert de **démonstration** élémentaire où, malgré la faible dimension du réseau, elle met en évidence l'esprit du **DSL**, chaque liaison $\omega_{i,j}$ se corrigeant localement en fonction de la synergie $S(i,j)$, sans qu'aucune supervision globale n'intervienne.

A. Mise en Place d'un Mini-Exemple

Pour clarifier la présentation, on se limite à trois entités $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$. Au temps initial $t = 0$, on initialise toutes les pondérations à une petite valeur égale, par exemple :

$$\omega_{12}(0) = \omega_{23}(0) = \omega_{13}(0) = 0.05.$$

On fixe un **taux d'apprentissage** $\eta = 0.1$ et un **coefficent de décroissance** $\tau = 0.2$. On attribue ensuite des **scores** de synergie $S(i,j)$ purement *ad hoc* :

$$S(1,2) = 0.8, \quad S(1,3) = 0.3, \quad S(2,3) = 0.6.$$

Ici, on suppose \mathcal{E}_1 et \mathcal{E}_2 fortement liées ($S = 0.8$), tandis que \mathcal{E}_1 et \mathcal{E}_3 ne le sont qu'à un niveau modéré ($S = 0.3$). En pratique, ces nombres peuvent provenir d'une **distance**, d'une **similarité** ou d'une **mesure** (cf. section 2.2.1.2), mais on les fixe ici pour la démonstration.

B. Application de la Règle de Mise à Jour

On effectue la **mise à jour** pour chaque itération $t = 0, 1, 2, \dots$. Par exemple, pour la liaison $\omega_{12}(t)$, on a :

$$\omega_{12}(t+1) = \omega_{12}(t) + \eta[0.8 - \tau \omega_{12}(t)].$$

De façon analogue, pour ω_{13} et ω_{23} :

$$\omega_{13}(t+1) = \omega_{13}(t) + \eta[0.3 - \tau \omega_{13}(t)], \quad \omega_{23}(t+1) = \omega_{23}(t) + \eta[0.6 - \tau \omega_{23}(t)].$$

On peut illustrer, par exemple, le premier pas de temps ($t = 0 \rightarrow 1$) pour ω_{12} . En notant $\omega_{12}(0) = 0.05$, on obtient :

$$\begin{aligned} \omega_{12}(1) &= 0.05 + 0.1[0.8 - 0.2 \times 0.05] = 0.05 + 0.1 \times [0.8 - 0.01] = 0.05 + 0.1 \times 0.79 \\ &= 0.05 + 0.079 = 0.129. \end{aligned}$$

On procédera de même pour $\omega_{13}(1)$ et $\omega_{23}(1)$, puis on itérera à chaque pas de temps ($t = 1 \rightarrow 2, 2 \rightarrow 3, \dots$).

C. Tendance vers un Équilibre

L'expérience révèle que $\omega_{12}(t)$ grimpe plus vite (la synergie 0.8 étant la plus élevée) alors que $\omega_{13}(t)$ stagne ou s'élève doucement (puisque la synergie 0.3 est la plus faible). Lorsqu'on laisse l'itération se dérouler sur plusieurs pas, chaque $\omega_{i,j}(t)$ se rapproche d'un **point fixe** $\omega_{i,j}^* \approx S(i,j)/\tau$, conformément à l'analyse de convergence de la section 2.2.2.1. Ainsi,

$$\omega_{12}^* \approx \frac{0.8}{0.2} = 4.0,$$

même s'il convient de réaliser un nombre significatif d'itérations pour s'en approcher, et de tenir compte du facteur η (taux d'apprentissage) qui peut ralentir ou causer de légères oscillations. L'important est d'observer la **logique** sous-jacente où un score $S(i,j)$ élevé provoque un renforcement plus prononcé du poids, tant que la décroissance $\tau \omega_{i,j}(t)$ ne compense pas entièrement le terme $\eta S(i,j)$.

D. Lecture Globale de la Démonstration

Même avec un **mini-exemple** impliquant trois entités $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ et une synergie fixée, on voit clairement comment la **règle** de mise à jour

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta[S(i,j) - \tau \omega_{ij}(t)]$$

entraîne, au fil des itérations, un **rééquilibrage** des pondérations selon les valeurs de $S(i,j)$. Les liaisons ayant une synergie plus grande croissent plus vite, celles ayant une synergie modérée stagnent à un niveau bas, et celles potentiellement nulles convergent vers zéro (en l'absence de renfort).

Dans un réseau plus large, comprenant 4, 5 entités ou davantage, et avec des définitions de synergie plus sophistiquées comme la distance euclidienne, la similarité cosinus ou la co-information, le principe demeure identique où chaque connexion $\omega_{i,j}$ évolue **localement** (cf. sections 2.2.2.2 et 2.2.2.3), et la structure globale, notamment la tendance à former des **clusters**, émerge de l'**accumulation** de ces ajustements de bas niveau.

2.2.5.2. Courtes Simulations (Distance Euclidienne vs. Similarité Cosinus)

Pour **illustrer** de manière concrète la dynamique d'un **Synergistic Connection Network (SCN)** dans le cadre du **Deep Synergy Learning (DSL)**, il est instructif d'entreprendre de petites **simulations** où la **synergie** $S(i, j)$ est définie par des formules accessibles (distance euclidienne inversée, similarité cosinus). L'objectif consiste à examiner, sur un **petit ensemble** (par exemple 4, 5 ou 6 vecteurs \mathbf{x}_i), la manière dont les pondérations $\omega_{i,j}$ se renforcent ou diminuent en fonction de la **proximité** vectorielle.

A. Contexte de la Simulation

Pour favoriser la **lisibilité**, on peut placer chaque entité \mathcal{E}_i dans un **plan** à deux dimensions, de sorte que $\mathbf{x}_i = (x_i, y_i) \in \mathbb{R}^2$. On peut, par exemple, définir des positions :

$$\mathbf{x}_1 = (1.0, 1.2), \quad \mathbf{x}_2 = (1.3, 0.8), \quad \mathbf{x}_3 = (4.2, 5.1), \quad \mathbf{x}_4 = (4.0, 5.4).$$

On constate que \mathbf{x}_1 et \mathbf{x}_2 se trouvent relativement proches, tandis que \mathbf{x}_3 et \mathbf{x}_4 forment un autre groupe de proximité. Les **pondérations** $\omega_{i,j}(0)$ sont initialisées à une petite valeur commune (par exemple 0.05). On fixe ensuite :

$$\eta = 0.1, \quad \tau = 0.2,$$

où η est le **taux d'apprentissage** et τ le **coefficient de décroissance**.

Afin de **comparer**, on fait tourner la **même** règle de mise à jour,

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

mais en faisant varier $S(i, j)$ selon deux définitions :

1. **Distance Euclidienne Inversée.** On peut poser

$$S_{\text{dist}}(i, j) = \frac{1}{1 + \| \mathbf{x}_i - \mathbf{x}_j \|}.$$

Ainsi, deux entités plus **proches** (faible distance) reçoivent un score de synergie plus élevé (le dénominateur ($1 + \text{distance}$) est réduit).

2. **Similarité Cosinus.** On peut employer

$$S_{\text{cos}}(i, j) = \max \left\{ 0, \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\| \mathbf{x}_i \| \| \mathbf{x}_j \|} \right\},$$

mettant l'accent sur l'**angle** entre \mathbf{x}_i et \mathbf{x}_j . Deux vecteurs pratiquement parallèles affichent un score élevé, indépendamment de la distance à l'origine.

B. Déroulement de la Simulation

On effectue généralement un certain nombre (10 ou 20) d'**itérations**. À chaque **step** :

3. On **calcule** $S(i, j)$ pour chaque paire (i, j) .

4. On met à jour $\omega_{i,j}(t+1)$ grâce à la formule ci-dessus.
5. On peut, si on le souhaite, imposer un seuil $\omega_{\min} = 0.01$ (voir section 2.2.3) afin de mettre à zéro les pondérations trop faibles.

Avec la **distance** inversée, $S_{\text{dist}}(i, j)$ se révèle plus grande si $\| \mathbf{x}_i - \mathbf{x}_j \|$ est petite. Si, par exemple, \mathbf{x}_1 et \mathbf{x}_2 sont dans un rayon proche, la synergie $S_{\text{dist}}(1, 2)$ grimpe, et $\omega_{12}(t)$ s'accroît logiquement. Dans le cas de la **similarité cosinus** $S_{\cos}(i, j)$, c'est l'**orientation** qui compte où deux vecteurs quasi parallèles obtiennent une forte corrélation angulaire, et leurs pondérations s'élèvent correspondamment.

Après une dizaine d'**itérations**, la majorité des pondérations $\omega_{i,j}$ convergent vers un **point d'équilibre** (s'approchant, selon la section 2.2.2.1, de $S(i, j)/\tau$). Les liens jugés peu synergiques (ex. paires avec $\| \mathbf{x}_i - \mathbf{x}_j \|$ grande ou direction divergente) s'affaiblissent et peuvent tendre vers zéro. L'éventuelle introduction de la **parsimonie** accélère l'obtention d'un **graphe épars**, où n'apparaissent plus que quelques pondérations fortes, signe de la formation de **sous-groupes**.

C. Analyse et Commentaires

Ces **courtes simulations** révèlent que :

- Lorsque $S(i, j)$ exploite la **distance** inversée $1/(1 + \| \mathbf{x}_i - \mathbf{x}_j \|)$, on favorise la **proximité géométrique**. Des vecteurs rapprochés entraînent une forte synergie et, donc, un renforcement marqué des liens $\omega_{i,j}$. On obtient souvent des **regroupements "spatiaux"**.
- Lorsque $S(i, j)$ recourt à la **similarité cosinus** ($\max\{0, \mathbf{x}_i \cdot \mathbf{x}_j / (\| \mathbf{x}_i \| \| \mathbf{x}_j \|)\}$), on regarde l'**angle** entre \mathbf{x}_i et \mathbf{x}_j . Deux entités alignées sur une direction analogue reçoivent une synergie élevée, même si elles sont éloignées de l'origine.

Les paramètres η et τ influencent la dynamique. Un η trop grand peut générer des oscillations, un η trop petit ralentit la convergence, un τ trop grand accélère la décroissance, etc. Dans un contexte de **seuil** ω_{\min} , la suppression de liaisons trop faibles clarifie la structure et dévoile rapidement des **sous-groupes** plus nets.

Les simulations "miniatures" montrent la **genèse de micro-clusters** où quelques $\omega_{i,j}$ s'élèvent rapidement lorsque $S(i, j)$ est forte, alors que d'autres $\omega_{i,j}$ tombent à zéro, scellant la partition en petits groupes. On retrouve là la base de l'**auto-organisation** dans le DSL où, à partir d'un graphe quasi uniforme, certains liens se renforcent progressivement tandis que d'autres s'éteignent au fil des itérations, révélant ainsi des **composantes** ou **communautés cohérentes**.

2.2.5.3. Premières Observations sur la Formation de Micro-Clusters

Les **courtes simulations** présentées en section 2.2.5.2 – qu'elles s'appuient sur la distance euclidienne inversée ou sur la similarité cosinus – font apparaître qu'après plusieurs itérations de la **règle** de mise à jour des pondérations, un **petit réseau** de 3, 4 ou 5 entités tend à voir naître des **groupes** (ou micro-clusters) fortement connectés. Ce phénomène illustre la **tendance** du Deep

Synergy Learning (DSL) à structurer le **Synergistic Connection Network (SCN)** en **clusters**, guidés par des **synergies** relativement élevées.

A. Processus d'Émergence des Clusters

Le fait de **définir** une synergie $S(i,j)$ plus grande pour certaines paires $(\mathcal{E}_i, \mathcal{E}_j)$ (par exemple en raison de leur proximité ou de leur corrélation statistique) se traduit, dans la **dynamique** décrite en 2.2.2, par une croissance plus rapide des pondérations $\omega_{i,j}(t)$ correspondantes. Inversement, les paires ne bénéficiant pas d'une synergie élevée voient leurs pondérations décliner ou rester à un niveau bas.

Si, lors des premières itérations, toutes les pondérations $\omega_{i,j}(0)$ se situent autour d'une même petite valeur, la **règle** de renforcement ($S(i,j)$) et de décroissance ($\tau \omega_{i,j}$) introduit progressivement une "**divergence**" dans l'évolution où quelques liaisons $\omega_{i,j}$ montent en flèche tandis que d'autres chutent vers zéro. Cet écart de trajectoires amène, dans un réseau **stable**, un **groupement** des entités ayant $S(i,j)$ plus forte, tandis que les autres connexions s'amoindrissent.

Lorsqu'on ne modifie pas la synergie $S(i,j)$ (c'est-à-dire, si le temps se déroule sans ajout de nouvelles entités ni mise à jour de la mesure de proximité), on constate empiriquement que les pondérations **convergent**. Chaque $\omega_{i,j}(t)$ se rapproche d'un point fixe, souvent $\omega_{i,j}^* \approx S(i,j)/\tau$. Le **réseau** en fin de phase présente alors quelques couples (ou sous-ensembles) dotés de liens nettement plus forts, mettant en relief leur affinité ou leur synergie.

B. Interprétation des Micro-Clusters

Dans un petit réseau (par exemple 3–5 entités), on peut observer la formation de **micro-clusters**. Un **micro-cluster** est généralement un groupe de 2 ou 3 entités qui se retrouvent reliées par des pondérations élevées, tandis que les connexions vers l'extérieur demeurent faibles ou nulles. Par exemple, $\{\mathcal{E}_1, \mathcal{E}_2\}$ peut nouer un **couple** au poids ω_{12} important, distinct d'un autre binôme $\{\mathcal{E}_3, \mathcal{E}_4\}$, affichant ses propres liens robustes.

Ces **sous-ensembles** correspondent à l'idée que \mathcal{E}_1 et \mathcal{E}_2 partagent une **synergie** suffisante (distance réduite, direction vectorielle similaire, etc.). Le réseau "**signe**" cette analogie en **renforçant** les liaisons internes au groupe et en **atténuant** les connexions vers les entités plus éloignées ou moins similaires. Ce mécanisme constitue une **illustration** de la façon dont, à plus grande échelle, le DSL forme spontanément des **clusters**, chaque entité rejoignant la communauté où elle trouve la **synergie** la plus marquée.

Une représentation géométrique du **réseau** dans un plan, avec les positions \mathbf{x}_i et l'intensité des liens $\omega_{i,j}$, met rapidement en évidence ces **groupes** où les liaisons puissantes ressortent, dévoilant la présence de **clusters**. Même sur un petit nombre d'entités, la dynamique de pondérations $\omega_{i,j}(t)$ montre cette capacité du **DSL** à repérer et à **confirmer** les affinités dominantes, phénomène central de l'**auto-organisation**.

C. Prolongements et Évolutions Possibles

Pour un ensemble plus large (par exemple 10, 20 ou 100 entités), le **même** mécanisme opère à plus grande échelle, créant des **clusters** plus volumineux (voire hiérarchiques). L'adoption d'un **seuil**

ω_{\min} (voir section 2.2.3) peut accélérer l'obtention d'une **structure** éparse et révéler plus franchement les **groupes**. Sur de très petits réseaux (3 ou 4 entités), l'application d'un seuil conduit à des binômes ou trinômes aux liaisons très intenses, fournissant un exemple minimaliste de **partition**.

Si la synergie $\{S(i, j)\}$ varie dans le temps (les entités se déplacent, leur mémoire interne change, etc.), la configuration des micro-clusters peut se **réarranger**. Un binôme initialement serré peut se dissoudre, tandis qu'un nouveau couple se consolide. Le DSL, par sa nature locale, suit ces changements en réajustant $\omega_{i,j}(t)$ selon la nouvelle donne. Même dans un réseau minuscule, on peut ainsi observer des **réorganisations** temporelles et des refontes de clusters en réponse à des variations contextuelles.

Ces **observations** constituent un avant-goût du fonctionnement à échelle plus grande où le **DSL** identifie et **stabilise** des liaisons là où la **synergie** se maintient, dévoilant des **sous-groupes** de haute affinité. Cette logique de “clustering auto-organisé” (ou “micro-clustering” en petit nombre) fonde la démarche du DSL comme algorithme **non supervisé** de regroupement, applicable à des entités diversifiées.

2.3. Hypothèses de Stabilité et Émergence de Clusters

Lorsque l'on considère un **Synergistic Connection Network (SCN)** soumis aux règles de mise à jour décrites en section (2.2), une question naturelle se pose où **le réseau converge-t-il** ? Et si oui, vers quelles formes de topologies ou de structures comme des clusters ou des attracteurs ? La section (2.3) aborde ces problématiques à travers un point de vue “système dynamique” (2.3.1), la possibilité d’attracteurs multiples ou d’oscillations (2.3.2), l’analyse de la formation de clusters (2.3.3) et l’influence du bruit (2.3.4). Nous nous intéresserons également en (2.3.5) aux hypothèses fortes et aux limites non résolues.

2.3.1. Point Fixe, Attracteurs : Définitions et Existence

L’évolution temporelle du réseau $\Omega(t)$ — où $\Omega(t)$ désigne l’ensemble des pondérations $\{\omega_{i,j}(t)\}$ (et éventuellement d’autres variables d’état) — peut souvent être examinée à travers la **théorie des systèmes dynamiques**. La première sous-section (2.3.1.1) formalise cette perspective ; les suivantes traiteront de l’existence d’un **point fixe** (2.3.1.2) et d’une **analyse locale** de stabilité (2.3.1.3).

2.3.1.1. Approche « système dynamique »

Dans une perspective théorique visant à décrire de manière globale l’évolution des connexions au sein d’un réseau d’entités interconnectées, il est particulièrement instructif d’adopter l’optique d’un système dynamique discret. L’idée centrale consiste à regrouper l’ensemble des pondérations, notées $\omega_{i,j}$ (éventuellement complétées par des variables représentant des états internes), dans une configuration globale qui sera représentée par le vecteur d’état

$$\Omega(t) = (\omega_{1,2}(t), \omega_{1,3}(t), \dots, \omega_{n-1,n}(t), \dots),$$

où t désigne l’instant discret (ou l’itération) et où, pour un réseau comportant n entités, le nombre total de pondérations peut atteindre $n(n - 1)$ (dans le cas orienté) ou $\frac{n(n-1)}{2}$ (dans le cas non orienté).

L’évolution de l’état du réseau dans le cadre du Deep Synergy Learning (DSL) se traduit par une règle de mise à jour collective, que l’on peut écrire sous la forme d’une équation fonctionnelle :

$$\Omega(t + 1) = F(\Omega(t)).$$

Ici, l’opérateur F représente l’ensemble des règles locales de mise à jour, telles que celles décrites dans la section 2.2.2, où chaque poids évolue selon l’équation

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta[S(i, j) - \tau \omega_{i,j}(t)].$$

La fonction $S(i, j)$ – qui quantifie la synergie ou l’utilité mutuelle entre les entités \mathcal{E}_i et \mathcal{E}_j – intervient comme un signal de renforcement, tandis que le terme de décroissance proportionnel à

$\tau \omega_{i,j}(t)$ permet de réguler la croissance des poids. Lorsque l'on regroupe l'ensemble de ces mises à jour dans un unique opérateur F , on obtient ainsi une formulation globale de la dynamique du réseau.

A. Structure de l'espace d'états

Dans le formalisme usuel de la théorie des systèmes dynamiques discrets, l'évolution se décrit par

$$\Omega(t+1) = F(\Omega(t)), \quad \Omega(t) \in \mathcal{X} \subseteq \mathbb{R}^D,$$

où D représente la dimension totale du vecteur d'états (qui, dans notre contexte, est le nombre de pondérations considérées). Plusieurs propriétés de cet espace d'états sont essentielles pour l'analyse :

- **Continuité de F** : Pour appliquer des résultats classiques tels que le théorème du point fixe de Banach, il est nécessaire que F soit continue, voire contractante, dans une norme appropriée.
- **Compacité et Bornitude de \mathcal{X}** : Dans de nombreux modèles, on impose des contraintes (par exemple, $\omega_{i,j} \geq 0$ ou des saturations maximales) afin que l'ensemble des configurations possibles \mathcal{X} soit borné. Ces conditions favorisent l'existence d'attracteurs et empêchent la divergence de la norme $\|\Omega(t)\|$.

B. Analyse de la dynamique locale et globale

Pour illustrer la dynamique d'un lien individuel, considérons la règle linéaire-additive

$$\omega_{i,j}(t+1) = (1 - \eta \tau) \omega_{i,j}(t) + \eta S(i, j).$$

Si l'on suppose que les synergies $S(i, j)$ sont elles-mêmes bornées dans un intervalle $[0, S_{\max}]$, il est immédiat de constater que la dynamique unidimensionnelle de chaque $\omega_{i,j}$ tend vers l'équilibre

$$\omega_{i,j}^* \approx \frac{S(i, j)}{\tau},$$

à condition que les paramètres η et τ soient choisis de manière appropriée (c'est-à-dire suffisamment faibles pour garantir une convergence stable et éviter des oscillations indésirables).

Au niveau collectif, l'opérateur F regroupe l'ensemble de ces équations de mise à jour. Pour étudier la stabilité du système dynamique global, on s'intéresse notamment aux questions suivantes :

- **Contraction ou Monotonie de F** : Dans quelle mesure existe-t-il une norme (ou une métrique) pour laquelle l'opérateur F contracte les distances entre configurations ? Un opérateur contractant garantira, par le théorème du point fixe, l'existence d'un unique point fixe attracteur, vers lequel converge toute trajectoire.
- **Existence et Stabilité d'un Point Fixe** : Sous quelles conditions sur les paramètres η, τ et la fonction de synergie S peut-on montrer formellement l'existence d'un équilibre stable Ω^* tel que $F(\Omega^*) = \Omega^*$?
- **Impact des Termes Non Linéaires** : L'introduction de mécanismes supplémentaires, tels que l'inhibition compétitive ou une mise à jour multiplicative, rend la carte F plus

complexe, car la mise à jour d'un poids peut alors dépendre non seulement de la synergie entre deux entités, mais aussi des interactions avec d'autres connexions (par exemple, via des sommes telles que $\sum_{k \neq j} \omega_{i,k}$). Néanmoins, même dans ce cas, l'analyse se situe dans le cadre général des systèmes dynamiques discrets, et l'on peut rechercher des attracteurs, des cycles ou des comportements plus complexes (éventuellement chaotiques).

C. Implications pour l'auto-organisation du SCN

La formulation $\Omega(t+1) = F(\Omega(t))$ permet d'aborder l'auto-organisation du Synergistic Connection Network (SCN) sous l'angle des systèmes dynamiques. En effet, la convergence des pondérations vers des points fixes ou des attracteurs locaux traduit la formation d'une structure stable au sein du réseau. Ces attracteurs correspondent, dans une interprétation globale, à des configurations dans lesquelles les connexions les plus pertinentes sont renforcées, tandis que les liens moins utiles s'affaiblissent voire disparaissent. Ainsi, l'analyse de la stabilité et de la convergence de F constitue un outil puissant pour comprendre l'émergence de clusters durables et la robustesse de l'architecture auto-organisée.

Dans des variantes plus avancées du DSL, l'introduction de termes d'inhibition compétitive ou de mécanismes multiplicatifs ne fait qu'enrichir la dynamique, tout en restant dans le cadre itératif général. L'étude des propriétés de l'opérateur F – par exemple, via la linéarisation autour d'un point fixe et l'analyse de sa jacobienne – permettra d'identifier des conditions précises garantissant l'existence d'un équilibre stable, ou au contraire révélant la présence de cycles limités ou de comportements non linéaires plus complexes.

2.3.1.2. Existence d'un point fixe Ω^* : conditions théoriques, rôle de τ et η

Dans le cadre de l'approche système dynamique, l'idée centrale consiste à décrire l'évolution globale de la configuration des liaisons d'un réseau en regroupant l'ensemble des pondérations, notées $\omega_{i,j}(t)$, dans un vecteur d'état $\Omega(t)$. Ce vecteur, qui peut être considéré comme un élément de l'espace $\mathcal{X} \subset \mathbb{R}^D$ (où la dimension D correspond au nombre total de connexions, par exemple $n(n - 1)$ pour un réseau orienté ou $\frac{n(n-1)}{2}$ dans le cas non orienté), évolue selon une dynamique discrète que l'on peut exprimer par l'équation fonctionnelle

$$\Omega(t+1) = F(\Omega(t)).$$

Ici, l'opérateur F regroupe de manière globale l'ensemble des règles de mise à jour locales qui, dans le DSL (Deep Synergy Learning), s'expriment typiquement par la relation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $S(i,j)$ représente la synergie ou l'utilité mutuelle entre les entités \mathcal{E}_i et \mathcal{E}_j , $\eta > 0$ est le taux d'apprentissage qui détermine la vitesse d'évolution, et $\tau > 0$ est un paramètre de décroissance agissant comme régulateur afin d'éviter une croissance indéfinie des poids.

Un point fixe Ω^* de la dynamique est défini par la condition d'invariance

$$\boldsymbol{\Omega}^* = F(\boldsymbol{\Omega}^*).$$

La recherche d'un tel point fixe revient à examiner les conditions sous lesquelles la dynamique – issue de la mise à jour locale de chaque connexion – converge vers une configuration stable, c'est-à-dire une configuration pour laquelle les poids ne varient plus au fil des itérations.

A. Analyse dans le Cas Additif Simple

Pour mieux illustrer cette idée, considérons d'abord le cas de la mise à jour additive élémentaire. Pour une connexion donnée, la règle de mise à jour s'écrit

$$\omega_{i,j}(t+1) = (1 - \eta \tau) \omega_{i,j}(t) + \eta S(i,j),$$

en supposant que la valeur $S(i,j)$ soit constante (ou quasi-stationnaire) dans le temps. Rechercher un équilibre consiste alors à poser $\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega_{i,j}^*$. La relation d'équilibre devient immédiatement

$$\omega_{i,j}^* = (1 - \eta \tau) \omega_{i,j}^* + \eta S(i,j).$$

En isolant $\omega_{i,j}^*$ dans cette équation, nous obtenons

$$\eta S(i,j) = \eta \tau \omega_{i,j}^* \Rightarrow \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Ce résultat, qui est immédiatement visible dans le cas unidimensionnel, démontre que chaque poids converge vers un équilibre déterminé par le rapport de la synergie au coefficient de décroissance τ . Ainsi, le paramètre τ joue un rôle fondamental en modulant l'amplitude finale où si τ est élevé, l'équilibre s'établit à des valeurs relativement faibles, tandis qu'un τ plus faible permet aux poids d'atteindre des niveaux plus importants. Par ailleurs, le paramètre η conditionne la vitesse de convergence ; un η trop élevé risque de provoquer des oscillations ou des divergences, tandis qu'un η trop faible ralentit le processus de stabilisation.

B. Généralisation aux Systèmes Couplés

Dans un réseau complet (ou un Synergistic Connection Network – SCN), la synergie $S(i,j)$ peut dépendre non seulement de la relation entre les deux entités considérées, mais également des interactions avec d'autres connexions ou états internes. Dans ce cas, l'opérateur de mise à jour s'écrit de façon plus générale

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j; \{\omega_{k,\ell}(t)\}, \{\mathbf{s}_k(t)\}, \dots) - \tau \omega_{i,j}(t)].$$

Le point fixe global $\boldsymbol{\Omega}^*$ satisfait alors l'ensemble des équations

$$S(i,j; \{\omega_{k,\ell}^*\}) = \tau \omega_{i,j}^*, \quad \forall (i,j),$$

ce qui conduit à un système non linéaire d'équations. Pour établir l'existence d'un tel point fixe, on peut s'appuyer sur des arguments classiques de théorie des points fixes tels que le théorème de Brouwer ou celui de Schauder. En effet, si l'opérateur F est continu et que l'ensemble des configurations possibles \mathcal{X} (souvent défini par des contraintes telles que $\omega_{i,j} \in [0, \omega_{\max}]$) est compact et convexe, alors il existe au moins un point fixe.

L'unicité et la stabilité de ce point fixe dépendent de propriétés supplémentaires de F – en particulier, si F est contractant dans une norme appropriée, c'est-à-dire si l'on peut trouver un $\kappa < 1$ tel que

$$\| F(\mathbf{x}) - F(\mathbf{y}) \| \leq \kappa \| \mathbf{x} - \mathbf{y} \|$$

pour tout $\mathbf{x}, \mathbf{y} \in \mathcal{X}$. Dans ce cas, non seulement l'existence mais également l'unicité du point fixe est garantie, et toutes les trajectoires convergeront vers celui-ci.

C. Rôle des Paramètres η et τ

Dans la dynamique proposée, le paramètre η agit comme le taux d'apprentissage. Il détermine la rapidité avec laquelle la configuration $\Omega(t)$ s'ajuste en réponse au signal de renforcement fourni par $S(i,j)$ et à l'effet de décroissance proportionnel à $\tau \omega_{i,j}(t)$. Un choix judicieux de η est indispensable pour éviter des comportements oscillatoires ou divergents tout en permettant une convergence suffisamment rapide.

Le coefficient τ quant à lui intervient comme un régulateur qui « freine » la croissance des poids. En effet, dans la formule additive, il apparaît sous la forme d'un facteur multiplicatif qui, en équilibrant l'influence de la synergie $S(i,j)$, fixe l'amplitude finale de chaque pondération par le rapport $S(i,j)/\tau$. Une valeur élevée de τ force ainsi les poids à converger vers des valeurs plus faibles, ce qui peut être souhaitable pour maintenir un réseau stable et éviter la domination excessive d'un lien sur les autres.

D. Conclusion

La formulation du DSL sous la forme d'un système dynamique discret

$$\Omega(t+1) = F(\Omega(t))$$

permet d'appréhender la convergence et la stabilisation des liaisons par l'intermédiaire de la recherche d'un point fixe Ω^* . Dans le cas le plus simple d'une mise à jour additive, on démontre que chaque pondération tend vers l'équilibre

$$\omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Dans des systèmes plus complexes, où S dépend de l'ensemble des pondérations et éventuellement d'états internes supplémentaires, l'existence d'un point fixe peut être garantie par les théorèmes classiques de la théorie des points fixes, à condition que l'opérateur F soit continu et que l'ensemble des états soit compact. Les paramètres η et τ jouent alors un rôle déterminant où η régule la vitesse de convergence tandis que τ module l'amplitude finale des poids, assurant ainsi que l'équilibre atteint est à la fois stable et cohérent avec la logique de renforcement/décroissance inhérente au DSL.

Cette analyse théorique jette les bases d'une étude approfondie de la stabilité locale – par linéarisation et analyse de la jacobienne de F – et ouvre la voie à la compréhension des phénomènes d'auto-organisation, tels que la formation de clusters durables au sein du réseau. Les développements ultérieurs s'attachent à préciser les conditions techniques garantissant non seulement l'existence, mais aussi l'unicité et la robustesse de l'attracteur Ω^* .

2.3.1.3. Analyse locale (linéarisation, Jacobienne) pour évaluer la stabilité

Dans l'étude des systèmes dynamiques discrets appliqués aux réseaux d'auto-organisation du Deep Synergy Learning (DSL), il est essentiel de s'intéresser à la stabilité des points fixes de la dynamique globale. En d'autres termes, une fois qu'une configuration Ω^* des poids (et éventuellement d'autres états internes) vérifie

$$\Omega^* = F(\Omega^*),$$

la question cruciale est de déterminer si, en cas de petite perturbation autour de Ω^* , le système tend à revenir vers cet équilibre ou s'en éloigne. Pour répondre à cette interrogation, on utilise la technique de linéarisation locale, qui s'appuie sur le calcul de la matrice Jacobienne de l'opérateur F évaluée en Ω^* .

A. Linéarisation locale autour d'un point fixe

Soit un système dynamique discret défini par

$$\Omega(t+1) = F(\Omega(t)), \quad \Omega(t) \in \mathbb{R}^D,$$

où D représente la dimension totale de l'espace d'états, par exemple le nombre total de pondérations $\omega_{i,j}$ (éventuellement augmenté par d'autres variables telles que les états internes s_i). Un point fixe Ω^* est tel que

$$\Omega^* = F(\Omega^*).$$

Pour étudier la stabilité locale de ce point fixe, on introduit une perturbation infinitésimale $\delta\Omega(t)$ et on écrit

$$\Omega(t) = \Omega^* + \delta\Omega(t).$$

En supposant que $\delta\Omega(t)$ reste suffisamment petit, il est alors possible de linéariser la fonction F au voisinage de Ω^* en négligeant les termes d'ordre supérieur. On obtient ainsi :

$$\Omega(t+1) = F(\Omega^* + \delta\Omega(t)) \approx F(\Omega^*) + DF(\Omega^*) \delta\Omega(t),$$

où $DF(\Omega^*)$ désigne la matrice Jacobienne (de dimension $D \times D$) de F évaluée en Ω^* . Puisque $F(\Omega^*) = \Omega^*$, on peut réécrire :

$$\Omega(t+1) \approx \Omega^* + DF(\Omega^*) \delta\Omega(t).$$

En soustrayant Ω^* des deux côtés, la dynamique des perturbations est donnée par :

$$\delta\Omega(t+1) = DF(\Omega^*) \delta\Omega(t).$$

La stabilité locale du point fixe Ω^* est alors entièrement caractérisée par les valeurs propres de la matrice Jacobienne $DF(\Omega^*)$. En effet, pour un système discret, le point fixe est localement stable

(c'est-à-dire, asymptotiquement stable) si et seulement si toutes les valeurs propres λ de $DF(\Omega^*)$ satisfont la condition

$$|\lambda| < 1.$$

Si au moins une valeur propre vérifie $|\lambda| > 1$, toute petite perturbation dans la direction associée se verra amplifiée, rendant ainsi le point fixe instable. En présence de valeurs propres de module exactement égal à 1, la stabilité locale devient marginale et des comportements plus complexes (cycles, bifurcations) peuvent apparaître.

B. Jacobienne dans le cas “additif” simple

Pour illustrer concrètement le procédé, considérons le modèle additif simple de mise à jour présenté en section 2.2.2.1 :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

que l'on peut réécrire sous la forme

$$\omega_{i,j}(t+1) = (1 - \eta \tau) \omega_{i,j}(t) + \eta S(i,j).$$

Si l'on suppose que la fonction de synergie $S(i,j)$ est constante (ou varie très lentement) et qu'elle ne dépend pas directement des autres poids, la dynamique de chaque connexion se réduit à un système unidimensionnel dont le point fixe satisfait

$$\omega_{i,j}^* = (1 - \eta \tau) \omega_{i,j}^* + \eta S(i,j).$$

Une simple manipulation conduit à

$$\eta S(i,j) = \eta \tau \omega_{i,j}^* \quad \Rightarrow \quad \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

La dérivée de la fonction de mise à jour par rapport à $\omega_{i,j}$ est donnée par

$$\frac{\partial \omega_{i,j}(t+1)}{\partial \omega_{i,j}(t)} = 1 - \eta \tau.$$

La stabilité de ce point fixe est alors garantie si

$$|1 - \eta \tau| < 1,$$

ce qui se traduit par la condition

$$0 < \eta \tau < 2.$$

Dans un réseau complet, si chaque connexion évolue indépendamment selon la même loi, la matrice Jacobienne DF sera diagonale, avec $1 - \eta \tau$ sur chacune de ses entrées diagonales, et ses valeurs propres seront toutes égales à $1 - \eta \tau$. Dans le cas où des interactions entre différentes pondérations apparaissent (par exemple, par le biais d'un mécanisme d'inhibition compétitive ou via l'influence des états internes), la Jacobienne ne sera plus strictement diagonale et des termes hors-diagonale, liés aux dérivées partielles $\frac{\partial S(i,j)}{\partial \omega_{p,q}}$, viendront modifier sa structure. Toutefois, le

critère fondamental reste le même où l'ensemble des valeurs propres doit être inférieur à 1 en module pour garantir la stabilité locale.

C. Interprétation : perturbations mineures autour du point fixe

La relation obtenue

$$\delta\Omega(t+1) = DF(\Omega^*) \delta\Omega(t)$$

indique que la réponse du système à une perturbation infinitésimale est linéairement déterminée par la Jacobienne évaluée en Ω^* . En d'autres termes, si l'ensemble des valeurs propres de $DF(\Omega^*)$ se trouve dans le disque unité de \mathbb{C} , alors toute perturbation se contracte au fil du temps et le système revient asymptotiquement à son point fixe. Cette condition, formulée en norme opératoire par

$$\| DF(\Omega^*) \| < 1,$$

est l'exigence fondamentale pour la stabilité locale du point fixe. Dans le cas du modèle additif simple, cette condition se traduit par $0 < \eta \tau < 2$. Dans des systèmes plus complexes, le couplage entre les pondérations et l'influence d'autres variables induit des contributions supplémentaires dans la Jacobienne où l'analyse reste néanmoins structurée par le même principe, celui d'étudier la contraction locale des perturbations via les valeurs propres de la matrice dérivée.

D. Conclusion sur l'analyse locale

La linéarisation autour d'un point fixe Ω^* permet d'obtenir une description locale de la dynamique par le biais de la matrice Jacobienne $DF(\Omega^*)$. Le critère de stabilité locale, essentiel dans la théorie des systèmes dynamiques discrets, se réduit à l'exigence que toutes les valeurs propres λ de cette Jacobienne vérifient

$$|\lambda| < 1.$$

Dans le cas simple où la mise à jour est additive et $S(i, j)$ est constant, cela se traduit par la condition $|1 - \eta \tau| < 1$, soit $0 < \eta \tau < 2$. Lorsque le modèle est étendu pour intégrer des dépendances entre les pondérations et/ou l'influence d'états internes, la structure de la Jacobienne se complexifie, mais l'analyse demeure basée sur le contrôle de la norme de DF autour de Ω^* . Cette méthode analytique constitue ainsi un outil puissant pour évaluer la stabilité locale du point fixe et, par conséquent, pour anticiper la convergence ou la divergence des trajectoires du système en réponse à de petites perturbations.

2.3.2. Attracteurs Multiples et Oscillations

L'analyse locale (2.3.1) révèle que, autour d'un **point fixe** donné, on peut évaluer la stabilité via la linéarisation. Toutefois, même si un **point fixe** Ω^* est localement stable, rien ne garantit qu'il soit **unique** où, en fonction du **caractère non linéaire** de la mise à jour, incluant les couplages, l'inhibition compétitive et les synergies dépendant de plusieurs liaisons, il est fréquent de rencontrer **plusieurs** attracteurs ou même des **dynamismes** cycliques ou chaotiques. La section 2.3.2 s'intéresse à ces phénomènes, en commençant par la possibilité de **minima (ou attracteurs)**

multiples (2.3.2.1), avant d’aborder les **cycles et pseudo-chaos** (2.3.2.2), puis les méthodes pour caractériser ces comportements (2.3.2.3).

2.3.2.1. Possibilité de *minima multiples* : le SCN peut converger vers des topologies différentes selon les conditions initiales

Dans l’approche dynamique adoptée pour le Synergistic Connection Network (SCN), la mise à jour itérative des pondérations conduit à une configuration globale représentée par le vecteur Ω . La dynamique globale du réseau est décrite par l’équation

$$\Omega(t+1) = F(\Omega(t)),$$

et un point fixe Ω^* satisfait la condition d’invariance

$$\Omega^* = F(\Omega^*).$$

Dans un tel contexte, la nature non linéaire de F peut engendrer un paysage complexe d’énergie ou de potentiel qui n’est pas convexe. En conséquence, plusieurs solutions à l’équation du point fixe peuvent exister, chacune correspondant à une configuration stable distincte des pondérations.

Une des implications majeures de cette multiplicité des minima est que l’espace des conditions initiales se décompose en différents bassins d’attraction. Autrement dit, selon l’état initial du système, ou même selon l’ordre d’arrivée des données dans un cadre d’apprentissage en flux, le SCN pourra converger vers l’un ou l’autre de ces attracteurs. Ces configurations, qui représentent des topologies de liaisons différentes, traduisent ainsi la possibilité qu’un même réseau, soumis aux mêmes règles de mise à jour, puisse finalement présenter des partitions ou des clusterisations distinctes.

Parmi les facteurs qui contribuent à l’émergence de minima multiples dans un SCN, on peut citer :

- **La structure de la fonction de synergie $S(\dots)$:**

Lorsque $S(i,j)$ dépend de plusieurs variables ou intègre des couplages non linéaires (par exemple, par l’intermédiaire de mécanismes d’inhibition compétitive ou de dépendances n-aires), le paysage d’énergie associé au réseau présente de multiples vallées. Chaque vallée correspond à une configuration où les poids ont convergé vers des valeurs stables, et ces configurations peuvent être équivalentes en termes de score global.

- **La présence de rétroactions non linéaires :**

Si la règle de mise à jour intègre des termes contextuels – par exemple, une synergie modulée par la somme des poids entrants ou par des états internes des entités – le couplage entre les connexions renforce le caractère non convexe du paysage. Ainsi, plusieurs

arrangements cohérents peuvent satisfaire les équations d'équilibre, chacun étant localement stable.

- **Les états internes des entités :**

L'inclusion d'un vecteur d'état interne \mathbf{s}_i pour chaque entité, qui influence et est influencé par les pondérations $\omega_{i,j}$, apporte une dimension supplémentaire à la dynamique. Cette double rétroaction (entre poids et états internes) peut conduire à l'existence de plusieurs régimes d'auto-organisation, c'est-à-dire à des minima multiples correspondant à différentes topologies d'interconnexion.

Pour illustrer ces idées, considérons un exemple schématique. Supposons que nous disposions de quatre entités $\{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\}$ et que la structure de la synergie entre elles permette deux configurations stables équivalentes. Dans un premier scénario, les entités \mathcal{E}_1 et \mathcal{E}_2 forment un cluster tandis que \mathcal{E}_3 et \mathcal{E}_4 forment un second cluster. Dans un autre scénario, une partition alternative peut se dessiner, par exemple en regroupant \mathcal{E}_1 avec \mathcal{E}_3 et \mathcal{E}_2 avec \mathcal{E}_4 . Ces deux configurations, bien que différentes, peuvent présenter des scores globaux (ou niveaux d'énergie) comparables. Le choix de l'attracteur final dépendra alors des conditions initiales et de l'ordre dans lequel les données arrivent, illustrant ainsi la sensibilité du système aux conditions de départ.

Un autre aspect important est que, dans un modèle intégrant des mécanismes d'inhibition compétitive, chaque entité est contrainte de ne maintenir qu'un nombre limité de liaisons fortes. Cette restriction peut favoriser l'émergence de plusieurs arrangements attracteurs, car plusieurs combinaisons de connexions fortes peuvent satisfaire cette contrainte tout en optimisant localement le score de synergie.

2.3.2.2. Apparition éventuelle de cycles ou de régimes “pseudo-chaotiques” dans certains paramétrages (discussion d'exemples)

Dans le cadre du Deep Synergy Learning (DSL), la dynamique globale du réseau est décrite par la relation itérative

$$\boldsymbol{\Omega}(t+1) = F(\boldsymbol{\Omega}(t)),$$

où $\boldsymbol{\Omega}(t)$ représente la configuration complète des pondérations $\{\omega_{i,j}(t)\}$ (éventuellement complétée par d'autres états internes). Lorsque les règles de mise à jour intègrent des non-linéarités – par exemple, via l'inhibition compétitive, des couplages multiples ou la dépendance aux états internes \mathbf{s}_i – la carte F acquiert une complexité telle que l'évolution du système ne converge pas nécessairement vers un point fixe unique. Au lieu de cela, plusieurs comportements dynamiques peuvent émerger, parmi lesquels l'apparition de cycles périodiques ou de régimes « pseudo-chaotiques ».

A. Pourquoi un SCN peut-il présenter des cycles ou du pseudo-chaos ?

La clé de l'apparition de telles dynamiques réside dans la richesse non linéaire de la fonction de mise à jour. En effet, si la règle de mise à jour d'un poids $\omega_{i,j}$ ne se contente pas de répondre de manière linéaire au signal de synergie $S(i,j)$ et au terme de décroissance proportionnel à $\tau \omega_{i,j}(t)$, mais intègre également des contributions contextuelles (par exemple, la somme des autres poids connectés à la même entité ou des rétroactions issues des états internes), alors la fonction F devient suffisamment complexe pour permettre l'existence de plusieurs attracteurs ou même de cycles dynamiques.

Si la dérivée locale – autrement dit, l'analyse de la Jacobienne de F au voisinage d'un point fixe – révèle qu'il existe des directions dans lesquelles le module des valeurs propres dépasse 1, le point fixe se déstabilise. Ce phénomène conduit alors, par des bifurcations (par exemple une bifurcation de flip), à l'émergence de cycles de période 2, 3, etc. Dans des systèmes de dimension plus élevée, des enchaînements de bifurcations successives (doublements de période) peuvent aboutir à des régimes où la trajectoire ne converge plus vers un point fixe ni même vers un cycle simple, mais oscille de manière irrégulière dans l'espace d'états – phénomène que l'on qualifie alors de pseudo-chaotique.

Un mécanisme typique dans ce contexte est celui de l'inhibition compétitive où le renforcement d'un lien, par exemple $\omega_{i,j}$, entraîne l'inhibition des autres liens $\omega_{i,k}$ pour $k \neq j$, pouvant ainsi plonger le système dans une dynamique de va-et-vient où certains liens alternent leur renforcement et leur affaiblissement de manière cyclique.

B. Exemples schématiques

Pour illustrer ces concepts, considérons quelques exemples simplifiés :

- **Cycle de période 2**

Dans un mini-réseau composé de trois entités, il est envisageable que, lors d'une première itération, le lien $\omega_{1,2}$ soit renforcé au détriment de $\omega_{1,3}$ en raison d'un mécanisme d'inhibition. Lors de l'itération suivante, la situation s'inverse où l'affaiblissement de $\omega_{1,2}$ permet à $\omega_{1,3}$ de se renforcer. Ce va-et-vient induit un cycle de période 2, où le système oscille entre deux configurations distinctes.

- **Cycle de période 3**

Dans un système avec des interactions plus complexes – par exemple, lorsque plusieurs connexions interagissent simultanément et que la dépendance aux états internes est prise en compte – le réseau peut alors adopter un cycle de trois états distincts $\Omega_1 \rightarrow \Omega_2 \rightarrow \Omega_3 \rightarrow \Omega_1$. Ce type de comportement est souvent le résultat de rétroactions multiples qui décalent l'équilibre de manière cyclique.

- **Pseudo-chaos dans un système de grande dimension**

Dans un SCN de haute dimension, où le couplage entre de nombreuses pondérations et états internes induit une dynamique non linéaire complexe, il est possible que le système n'atteigne ni un point fixe ni un cycle périodique simple. La trajectoire de $\Omega(t)$ peut alors

parcourir de façon irrégulière une région de l'espace d'états sans jamais se répéter, évoquant ainsi un régime « pseudo-chaotique » caractérisé par une sensibilité accrue aux conditions initiales et une errance permanente dans l'attracteur.

D. Implications pratiques

D'un point de vue pratique, l'apparition de cycles ou de régimes pseudo-chaotiques dans le SCN présente plusieurs implications :

- **Non-convergence vers un état stable**

Dans un contexte de clustering ou de partitionnement, la présence de cycles signifie que la structure du réseau ne se stabilise pas, ce qui peut compliquer l'extraction d'une partition définitive.

- **Paramétrage délicat**

Afin d'éviter ces comportements oscillatoires indésirables, il est crucial d'ajuster soigneusement les paramètres tels que le taux d'apprentissage η , le coefficient de décroissance τ et l'intensité des mécanismes d'inhibition. Des réglages trop agressifs peuvent, en effet, provoquer des bifurcations multiples et conduire à un comportement pseudo-chaotique.

- **Approche cognitive alternative**

Dans certains modèles inspirés de la biologie ou de la cognition, ces régimes oscillatoires peuvent être interprétés comme des mécanismes d'exploration ou d'attention alternante, où le réseau alterne entre plusieurs configurations d'information. Ces dynamiques, loin d'être de simples défauts, pourraient servir à modéliser des processus cognitifs tels que la résonance neuronale ou la fluctuation de l'attention.

2.3.2.3. Stratégies pour détecter et caractériser ces phénomènes

Dans le cadre de l'étude des systèmes dynamiques non linéaires appliqués au **Synergistic Connection Network (SCN)**, il est primordial de développer des stratégies rigoureuses permettant de détecter et de caractériser les comportements complexes tels que la convergence vers des attracteurs multiples, l'apparition de cycles périodiques ou encore des régimes de type pseudo-chaotique. La dynamique du SCN s'exprime par l'équation itérative

$$\boldsymbol{\Omega}(t+1) = F(\boldsymbol{\Omega}(t)),$$

où le vecteur $\boldsymbol{\Omega}(t) \in \mathbb{R}^D$ représente l'ensemble des **pondérations** $\omega_{i,j}(t)$ et éventuellement d'autres variables d'état, telles que les **états internes** $\mathbf{s}_i(t)$. La complexité inhérente à la fonction F – qui intègre les effets de la **synergie** $S(i,j)$, des mécanismes d'inhibition compétitive, et d'éventuelles dépendances croisées entre les pondérations – peut amener le système à ne pas converger vers un point fixe unique, mais à adopter des comportements oscillatoires ou irréguliers.

Une première approche pour détecter ces phénomènes consiste à réaliser des **simulations numériques**. On initialise le SCN avec différentes conditions initiales $\Omega(0)$ et on varie les paramètres clés du modèle, tels que le **taux d'apprentissage** η et le **coefficient de décroissance** τ . En observant l'évolution des pondérations $\omega_{i,j}(t)$ au fil du temps, on peut distinguer trois scénarios typiques. Si, après un nombre suffisant d'itérations, les pondérations convergent vers des valeurs constantes, cela indique l'existence d'un point fixe attracteur. À l'inverse, si les valeurs oscillent entre deux configurations ou plus, on déduit la présence d'un cycle de période correspondante, par exemple en constatant que

$$\Omega(t + p) \approx \Omega(t)$$

pour une période p donnée. Enfin, si aucune périodicité claire ne se dégage et que la trajectoire de $\Omega(t)$ évolue de manière erratique, l'hypothèse d'un régime pseudo-chaotique est alors plausible.

Pour les systèmes de faible dimension, il est souvent utile d'effectuer une **projection** de la trajectoire dans un espace à deux ou trois dimensions afin de visualiser les trajectoires de $\Omega(t)$. Par exemple, si l'on considère que D est réduit, on peut représenter graphiquement la suite $\{\Omega(t)\}_{t=0}^T$ dans \mathbb{R}^3 pour observer d'éventuelles boucles ou motifs répétés. Dans des réseaux de grande dimension, des outils de visualisation tels que des **cartes de chaleur** ou des **animations** peuvent être employés pour suivre l'évolution des composantes individuelles des vecteurs $\Omega(t)$.

La théorie des systèmes dynamiques offre également des outils formels pour la caractérisation des comportements non convergents. Lorsqu'un point fixe Ω^* est identifié, on peut calculer la **matrice Jacobienne** $DF(\Omega^*)$ qui, localement, linéarise la dynamique autour de Ω^* par la relation

$$\delta\Omega(t + 1) = DF(\Omega^*) \delta\Omega(t),$$

où $\delta\Omega(t) = \Omega(t) - \Omega^*$. La stabilité locale du point fixe est alors déterminée par les valeurs propres λ de $DF(\Omega^*)$. Si l'une des valeurs propres satisfait $|\lambda| > 1$, le point fixe se déstabilise dans la direction correspondante, ce qui peut entraîner l'apparition d'un cycle de période 2 (par exemple, via une bifurcation de flip) ou de comportements plus complexes lorsque plusieurs valeurs propres dépassent 1 en module.

Afin de quantifier la présence de régimes chaotiques, on peut recourir au calcul de l'**exposant de Lyapunov maximal** λ_{\max} , défini par

$$\lambda_{\max} = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{\|\delta\Omega(t)\|}{\|\delta\Omega(0)\|}.$$

Un exposant de Lyapunov positif indique une sensibilité extrême aux conditions initiales, caractéristique du chaos discret, et confirme ainsi la présence d'un régime pseudo-chaotique.

Pour détecter la **multi-stabilité** du système, il est recommandé de réaliser plusieurs simulations en démarrant à partir de différentes conditions initiales, appelées « multi-run ». Si ces simulations convergent vers des attracteurs différents, cela démontre que le paysage d'énergie du SCN comporte plusieurs minima locaux, chacun disposant de son propre bassin d'attraction. Une autre méthode consiste à perturber légèrement un attracteur supposé stable, par exemple en posant

$$\Omega(0) = \Omega^* + \delta\Omega(0),$$

et en observant si la trajectoire converge toujours vers Ω^* ou si elle bascule vers une autre configuration attractrice.

L'ensemble de ces approches, qu'il s'agisse de simulations numériques, de l'analyse de la matrice Jacobienne ou de l'estimation des exposants de Lyapunov, fournit un cadre complet pour détecter et caractériser les phénomènes de cycles, de multi-attracteurs et de pseudo-chaos dans un SCN. Ces méthodes permettent d'obtenir une compréhension fine des dynamiques non convergentes et, par conséquent, d'adapter le paramétrage du système (notamment les valeurs de η et de τ) afin de favoriser, selon les objectifs, la convergence vers des configurations stables ou, au contraire, de tirer parti des régimes oscillatoires pour modéliser des comportements cognitifs dynamiques.

2.3.3. Analyse de la Formation de Clusters

Dans un **SCN** (Synergistic Connection Network), l'une des manifestations les plus remarquables de la **stabilité** (voire de la multi-stabilité, 2.3.2) est l'**auto-organisation** en **clusters** plus ou moins durables. Les sections précédentes (2.3.1 et 2.3.2) se concentraient sur la dynamique globale, l'existence de points fixes ou de cycles. Ici (2.3.3), nous nous focalisons plus spécifiquement sur la **notion** de cluster et sur la manière dont ils émergent et se maintiennent. Après avoir défini ce qu'on entend par “cluster stable” (2.3.3.1), nous aborderons (2.3.3.2) les **mesures** (cohésion, modularité) permettant de quantifier la qualité de ces regroupements, puis nous distinguerons (2.3.3.3) les clusters **transitoires** de ceux plus établis (macro-clusters), susceptibles d'apparaître à grande échelle.

2.3.3.1. Concept de “cluster stable” : définition, critères de cohésion interne

Dans le cadre du **Synergistic Connection Network (SCN)**, la notion de **cluster** représente un concept central dans l'étude de l'auto-organisation. Un **cluster**, ou sous-groupe, se définit comme un **ensemble** d'entités $\mathcal{C} \subseteq \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ dans lequel les **liens** entre les entités internes, notés $\omega_{i,j}$ pour $i, j \in \mathcal{C}$, sont particulièrement **forts** et traduisent une **synergie** élevée. Parallèlement, les connexions reliant une entité $\mathcal{E}_i \in \mathcal{C}$ à une entité extérieure $\mathcal{E}_k \notin \mathcal{C}$ se caractérisent par des valeurs de $\omega_{i,k}$ nettement plus faibles. Cette dichotomie, intrinsèque à l'auto-organisation, émerge spontanément lorsque, au cours de l'évolution du réseau, certains liens se renforcent tandis que d'autres s'affaiblissent, conduisant à la formation de structures en « blocs » clairement identifiables dans la matrice des pondérations.

La notion de **stabilité** d'un cluster se décline en deux aspects complémentaires qui méritent d'être étudiés en profondeur. D'une part, la **stabilité dynamique** se traduit par le maintien dans le temps des **pondérations internes** du cluster, de sorte que les connexions $\omega_{i,j}$ entre les entités $\mathcal{E}_i, \mathcal{E}_j \in \mathcal{C}$ restent élevées et ne se dégradent pas, alors que les liens avec l'extérieur demeurent faibles ou, dans certains cas, disparaissent complètement. Cette propriété dynamique signifie que de petites perturbations, telles que des fluctuations légères de $\omega_{i,j}$, n'ont pas la capacité de déstabiliser le sous-groupe ni de modifier substantiellement sa structure. D'autre part, la **cohérence interne** se manifeste par un principe d'auto-renforcement où les entités appartenant à \mathcal{C} maintiennent une synergie mutuelle systématiquement supérieure à celle qu'elles entretiennent avec des entités extérieures. Cette cohérence peut être formalisée par la définition d'un indice de cohésion, tel que

la moyenne des pondérations internes, qui doit rester au-dessus d'un certain seuil pour que l'on puisse parler d'une forte cohésion.

Pour formaliser ces idées, on introduit d'abord l'indicateur de **cohésion minimale interne** défini par

$$\bar{\omega}_{\text{in}}(\mathcal{C}) = \frac{1}{|\mathcal{C}|^2} \sum_{i,j \in \mathcal{C}} \omega_{i,j}.$$

Cet indicateur doit être supérieur à un seuil prédéfini $\theta_{\text{in}} > 0$ pour garantir que les interactions au sein du cluster sont suffisamment fortes. Parallèlement, l'indicateur de **faible connectivité externe** est défini par

$$\bar{\omega}_{\text{ext}}(\mathcal{C}) = \frac{1}{|\mathcal{C}|(n - |\mathcal{C}|)} \sum_{\substack{i \in \mathcal{C} \\ k \notin \mathcal{C}}} \omega_{i,k},$$

et il est requis que $\bar{\omega}_{\text{ext}}(\mathcal{C})$ demeure inférieur à un seuil θ_{ext} . La persistance de ces deux critères sur une période temporelle significative, notée par l'intervalle $t \in [T_0, T_1]$, assure que le cluster reste stable face aux fluctuations, ce qui indique que l'ensemble \mathcal{C} agit comme un **attracteur local** dans l'espace dynamique du réseau.

Cette approche permet de traduire de manière formelle le concept de **cluster stable**. On peut exprimer cette stabilité par l'inégalité

$$\frac{1}{|\mathcal{C}|^2} \sum_{i,j \in \mathcal{C}} \omega_{i,j}(t) \gg \frac{1}{|\mathcal{C}|(n - |\mathcal{C}|)} \sum_{\substack{i \in \mathcal{C} \\ k \notin \mathcal{C}}} \omega_{i,k}(t),$$

qui doit être vérifiée de manière persistante pour $t \in [T_0, T_1]$. L'interprétation mathématique de cette inégalité consiste à affirmer que la **moyenne interne** des pondérations est bien supérieure à la **moyenne externe**, garantissant ainsi une **séparation nette** entre les interactions intra-cluster et inter-cluster.

Les avantages de cette approche résident dans sa capacité à fournir une mesure quantitative de la cohésion interne et de la séparation externe, ce qui est essentiel pour identifier et analyser les regroupements dans le cadre d'un apprentissage non supervisé. L'utilisation de telles mesures permet non seulement de détecter des clusters stables, mais également de caractériser leur robustesse face aux perturbations. Toutefois, un inconvénient potentiel est que les seuils θ_{in} et θ_{ext} doivent être choisis en fonction de l'échelle typique des pondérations dans le SCN, ce qui peut rendre l'approche sensible à la normalisation des données et aux paramètres du système.

2.3.3.2. Mesures possibles (cohésion, modularité) pour juger la qualité ou la force d'un regroupement

Dans le cadre de l'analyse des **clusters** issus du Synergistic Connection Network (**SCN**), il est fondamental de disposer de critères quantitatifs permettant de mesurer la **cohésion interne** et la

séparation externe d'un sous-ensemble d'entités $\mathcal{C} \subseteq \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$. Ces mesures visent à déterminer dans quelle mesure les **liaisons** $\omega_{i,j}$ établies entre les entités au sein du cluster sont significativement plus fortes que celles reliant les membres de \mathcal{C} aux entités extérieures. Un premier indicateur est la **densité interne** qui évalue la moyenne (ou la somme) des pondérations entre toutes les paires d'entités appartenant à \mathcal{C} . Par exemple, on définit la densité interne par

$$\bar{\omega}_{\text{in}}(\mathcal{C}) = \frac{1}{|\mathcal{C}|(|\mathcal{C}| - 1)} \sum_{\substack{i, j \in \mathcal{C} \\ i \neq j}} \omega_{i,j}.$$

Cette moyenne quantifie le niveau de **force** des interactions à l'intérieur du cluster et, plus sa valeur est élevée, plus la cohésion interne est importante. En parallèle, il est pertinent de mesurer la **connectivité externe** du cluster en évaluant la moyenne des pondérations entre les entités à l'intérieur de \mathcal{C} et celles situées dans le complémentaire $V \setminus \mathcal{C}$ du réseau. On définit ainsi

$$\bar{\omega}_{\text{ext}}(\mathcal{C}) = \frac{1}{|\mathcal{C}|(n - |\mathcal{C}|)} \sum_{\substack{i \in \mathcal{C} \\ k \notin \mathcal{C}}} \omega_{i,k}.$$

L'interprétation de ces deux mesures permet d'établir un critère de séparation, qui peut être exprimé par le **ratio** suivant

$$\text{ratio}(\mathcal{C}) = \frac{\bar{\omega}_{\text{in}}(\mathcal{C})}{\bar{\omega}_{\text{ext}}(\mathcal{C}) + \epsilon},$$

où $\epsilon > 0$ est une constante très faible introduite pour éviter toute division par zéro. Un ratio élevé indique que le cluster est fortement cohésif par rapport à ses connexions extérieures, ce qui constitue un indice quantitatif fort de **stabilité** et de **qualité** du regroupement.

Une approche complémentaire consiste à recourir à la notion de **modularité**, empruntée à l'analyse des communautés dans les graphes pondérés. Dans ce contexte, la modularité permet de comparer la somme des **liaisons internes** à ce que l'on obtiendrait dans un modèle aléatoire conservant la même distribution des degrés. Pour un graphe pondéré $G = (V, E)$, la modularité, dans sa forme Newman-Girvan adaptée, se définit par

$$Q = \frac{1}{2W} \sum_{\substack{i, j \in V \\ \text{même cluster}}} \left[\omega_{ij} - \frac{k_i k_j}{2W} \right],$$

où $2W = \sum_{i,j} \omega_{ij}$ représente le total des pondérations du graphe et $k_i = \sum_j \omega_{ij}$ désigne le degré pondéré du nœud i . Un score de modularité élevé, souvent supérieur à 0.3 ou 0.4 selon l'échelle, témoigne d'une division du graphe en communautés (ou clusters) qui exhibent des **liaisons internes** significativement supérieures à ce qui serait attendu par hasard. Cette mesure, en intégrant des références statistiques au modèle aléatoire, offre un cadre rigoureux pour évaluer la **qualité** des regroupements et, par extension, leur pertinence dans l'analyse des données.

Par ailleurs, il existe d'autres indices issus de l'analyse des graphes, tels que la **conductance** ou la **densité relative**, qui permettent d'évaluer la proportion des liaisons sortantes par rapport à l'ensemble des liaisons d'un cluster. La conductance, par exemple, est faible lorsque le cluster

présente une forte cohésion interne et une faible connectivité externe. Bien que ces indices soient souvent utilisés en complément de la modularité, leur emploi dans le contexte du SCN permet de disposer d'une palette d'outils pour apprécier de manière fine la **robustesse** d'un regroupement.

Du point de vue pratique, ces mesures jouent un rôle crucial dans le cadre du **Deep Synergy Learning**. En effet, elles servent non seulement à identifier les clusters qui émergent de manière auto-organisée, mais également à comparer différentes partitions obtenues dans des simulations multi-run, notamment lorsque le réseau présente plusieurs attracteurs possibles. Ces outils facilitent ainsi la visualisation et l'analyse des structures en blocs qui se dégagent dans la matrice des pondérations, permettant de regrouper les entités en “macro-nœuds” pour simplifier l'interprétation globale du réseau.

2.3.3.3. Distinction entre clusters transitoires et macro-clusters à large échelle

Dans le cadre du **Synergistic Connection Network (SCN)**, l'analyse des regroupements d'entités conduit à distinguer deux phénomènes d'auto-organisation qui, bien que fondamentalement liés, diffèrent par leur durabilité et leur rôle dans la structure globale du réseau. Le premier phénomène correspond aux **clusters transitoires**, qui se caractérisent par une **cohésion interne** temporaire et une apparition éphémère dans la dynamique du SCN, tandis que le second se réfère aux **macro-clusters** qui, par leur stabilité et leur persistance sur de longues périodes, se révèlent comme des attracteurs dominants dans l'espace des configurations.

La notion de **cluster transitoire** désigne un sous-ensemble $\mathcal{C} \subseteq \{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ dont les liaisons internes, représentées par l'ensemble des pondérations $\{\omega_{i,j} \mid i, j \in \mathcal{C}\}$, atteignent un niveau élevé pendant un intervalle de temps limité. Formellement, si l'on définit la cohésion interne moyenne par

$$\bar{\omega}_{\text{in}}(\mathcal{C}, t) = \frac{1}{|\mathcal{C}|(|\mathcal{C}| - 1)} \sum_{\substack{i, j \in \mathcal{C} \\ i \neq j}} \omega_{i,j}(t),$$

et la connectivité externe moyenne par

$$\bar{\omega}_{\text{ext}}(\mathcal{C}, t) = \frac{1}{|\mathcal{C}|(n - |\mathcal{C}|)} \sum_{\substack{i \in \mathcal{C} \\ k \notin \mathcal{C}}} \omega_{i,k}(t),$$

alors un cluster transitoire se manifeste par une condition telle que, pour un intervalle temporel restreint $t \in [T_0, T_1]$,

$$\bar{\omega}_{\text{in}}(\mathcal{C}, t) \gg \bar{\omega}_{\text{ext}}(\mathcal{C}, t).$$

Ce regroupement peut apparaître en raison de fluctuations temporaires dans les **états internes** des entités ou lors de phases de transition où de petits sous-groupes se forment brièvement avant de fusionner ou de se dissoudre. Ces phénomènes, bien que révélateurs d'une forte synergie momentanée, ne garantissent pas la pérennité de la structure car de nouvelles mises à jour, l'effet d'une inhibition compétitive ou l'évolution des synergies, peuvent entraîner la disparition rapide de ces regroupements.

À l'inverse, le concept de **macro-cluster** se réfère à un regroupement stable et persistant. Un macro-cluster est défini comme un sous-ensemble \mathcal{C} dont la cohésion interne, mesurée par

$$\bar{\omega}_{\text{in}}(\mathcal{C}, t),$$

se maintient à un niveau élevé pendant une période prolongée, et dont la connectivité externe, $\bar{\omega}_{\text{ext}}(\mathcal{C}, t)$, reste faible de façon durable. On peut caractériser la stabilité d'un macro-cluster par l'inégalité

$$\frac{1}{|\mathcal{C}|^2} \sum_{i,j \in \mathcal{C}} \omega_{i,j}(t) \gg \frac{1}{|\mathcal{C}|(n - |\mathcal{C}|)} \sum_{\substack{i \in \mathcal{C} \\ k \notin \mathcal{C}}} \omega_{i,k}(t),$$

qui doit être vérifiée de manière continue sur un intervalle temporel $[T_0, T_1]$ suffisamment long pour certifier la **persistence** et la **robustesse** du regroupement. La stabilité d'un macro-cluster s'exprime également par une faible **variance** temporelle de $\bar{\omega}_{\text{in}}(\mathcal{C}, t)$, signe que les synergies internes ne fluctuent pas de manière excessive, et par la capacité du cluster à résister à des **perturbations** mineures (par exemple, un léger bruit ajouté aux pondérations).

Pour distinguer concrètement les clusters transitoires des macro-clusters, plusieurs approches méthodologiques peuvent être mises en œuvre. L'utilisation d'une **fenêtre glissante** permet de segmenter l'évolution temporelle en intervalles $[t, t + \Delta]$ sur lesquels les mesures $\bar{\omega}_{\text{in}}(\mathcal{C}, t)$ et $\bar{\omega}_{\text{ext}}(\mathcal{C}, t)$ sont calculées. Si, au sein d'une seule fenêtre, la cohésion interne atteint un pic mais retombe ensuite, on identifie un regroupement transitoire. Par ailleurs, l'application d'un **test de perturbation**, consistant à injecter un bruit faible dans la configuration des pondérations, peut révéler la robustesse d'un macro-cluster où un cluster stable se reconstitue rapidement après perturbation, tandis qu'un regroupement éphémère s'effondre.

Un exemple schématique permet d'illustrer cette distinction. Considérons un SCN de cinq entités où, lors des premières itérations, un petit sous-groupe tel que $\{\mathcal{E}_1, \mathcal{E}_2\}$ présente une forte cohésion interne, indiquée par un pic temporaire de $\bar{\omega}_{\text{in}}(\{\mathcal{E}_1, \mathcal{E}_2\}, t)$. Toutefois, si, lors des itérations suivantes, cette cohésion ne persiste pas et que ces entités finissent par être intégrées dans un regroupement plus vaste, alors $\{\mathcal{E}_1, \mathcal{E}_2\}$ correspond à un **cluster transitoire**. En revanche, si une partition plus large, par exemple $\mathcal{C} = \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\}$, se maintient avec un $\bar{\omega}_{\text{in}}(\mathcal{C}, t)$ élevé et une $\bar{\omega}_{\text{ext}}(\mathcal{C}, t)$ faible sur une durée prolongée, alors ce regroupement constitue un **macro-cluster** stable.

Les **avantages** d'une telle approche résident dans la possibilité de quantifier de manière rigoureuse la qualité d'un regroupement via des indices mathématiques tels que la **densité interne** et la **modularité**. Cette modularité, définie par

$$Q = \frac{1}{2W} \sum_{\substack{i,j \in V \\ \text{même cluster}}} \left[\omega_{ij} - \frac{k_i k_j}{2W} \right],$$

avec $2W = \sum_{i,j} \omega_{ij}$ et $k_i = \sum_j \omega_{ij}$, offre une mesure comparative entre le regroupement observé et ce que l'on pourrait obtenir dans un modèle aléatoire. Un score de modularité élevé indique que les liens internes sont bien supérieurs aux attentes d'un réseau aléatoire, attestant ainsi de la robustesse et de la pertinence du regroupement. Les **limites** de cette méthode résident

principalement dans la nécessité d'ajuster les seuils θ_{in} et θ_{ext} en fonction de l'échelle des pondérations dans le SCN, ainsi que dans la sensibilité potentielle des mesures à des fluctuations temporelles qui, dans des systèmes très dynamiques, pourraient rendre la distinction entre clusters transitoires et macro-clusters moins nette.

2.3.4. Influence du Bruit et des Perturbations

L'analyse de la **stabilité** et de la **formation de clusters** (sections 2.3.1 à 2.3.3) s'est déroulée dans un cadre relativement “idéal”. Dans la **pratique**, un SCN (Synergistic Connection Network) peut faire face à diverses **perturbations** ou à un **bruit** injecté — que ce bruit affecte la **mesure de synergie** (données incertaines, capteurs bruyants, etc.) ou qu'il concerne directement la **dynamique** des pondérations (ajout d'aléas dans la mise à jour). La section 2.3.4 examine comment ces facteurs perturbateurs influent sur la trajectoire du SCN et la robustesse des **clusters** qu'il forme. Après une introduction (2.3.4.1) sur la façon dont le SCN réagit au bruit, on abordera la **résilience** (2.3.4.2) et l'idée qu'un **certain** niveau de bruit peut même favoriser l'exploration (2.3.4.3).

2.3.4.1. Comment le SCN réagit à un bruit injecté sur les entités ou sur la fonction de synergie

Dans le cadre du **Synergistic Connection Network (SCN)**, la présence de **bruit** joue un rôle crucial dans la dynamique d'auto-organisation. La perturbation aléatoire, qui peut être introduite à divers niveaux — que ce soit dans la **représentation** des entités \mathbf{x}_i ou de leurs **états internes** \mathbf{s}_i , dans le calcul de la **fonction de synergie** $S(i, j)$ ou encore directement dans la **mise à jour** des **pondérations** $\omega_{i,j}$ — influence directement la capacité du réseau à converger vers une configuration stable. En effet, le SCN évolue selon l'équation de mise à jour additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

où le terme $\eta [S(i, j) - \tau \omega_{i,j}(t)]$ détermine l'ajustement des poids. L'injection d'un **bruit** peut modifier l'issue de cette évolution de plusieurs manières.

Lorsqu'un **bruit** est injecté dans la **représentation** des entités, par exemple en ajoutant une perturbation aléatoire à chaque vecteur \mathbf{x}_i , la **distance** ou la **similarité** utilisée pour calculer $S(i, j)$ sera modifiée. Autrement dit, même si la structure intrinsèque des entités demeure identique, les mesures telles que

$$S(i, j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|)$$

peuvent varier en raison d'une fluctuation aléatoire dans \mathbf{x}_i ou \mathbf{x}_j . Ainsi, deux entités qui étaient auparavant considérées comme proches peuvent temporairement apparaître éloignées, ce qui affecte négativement la **consolidation** de leurs liens et peut retarder, voire empêcher, la formation d'un regroupement stable.

De même, si le **bruit** est directement appliqué sur la **fonction de synergie**, on peut modéliser cette perturbation par l'introduction d'un terme aléatoire $\varepsilon_{i,j}(t)$ dans le calcul de la synergie. On écrira alors

$$S_{\text{obs}}(i,j) = S_{\text{réel}}(i,j) + \varepsilon_{i,j}(t),$$

ce qui implique que la valeur utilisée pour mettre à jour les poids est bruitée. Même lorsque les représentations \mathbf{x}_i et \mathbf{x}_j sont parfaitement stables, la présence de $\varepsilon_{i,j}(t)$ conduit à une fluctuation dans le signal de renforcement, induisant ainsi des variations aléatoires dans l'évolution de $\omega_{i,j}$.

Il est également possible d'injecter le **bruit** directement dans la **mise à jour** des pondérations. Dans ce cas, l'équation de mise à jour se modifie en ajoutant un terme perturbateur $\xi_{i,j}(t)$:

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] + \xi_{i,j}(t).$$

Même si la valeur de $S(i,j)$ reste exacte, ce terme $\xi_{i,j}(t)$ génère des fluctuations supplémentaires qui font osciller les poids autour de leur point d'équilibre théorique $\omega_{i,j}^* \approx S(i,j)/\tau$.

Les **effets du bruit** dépendent de son amplitude relative au terme de correction $\eta [S(i,j) - \tau \omega_{i,j}(t)]$. Lorsque le **bruit** est modeste, la dynamique fondamentale de renforcement et de décroissance domine, de sorte que le réseau parvient à stabiliser ses pondérations malgré de légères fluctuations, voire à explorer de nouveaux minima locaux dans l'espace d'état, ce qui peut parfois éviter de rester bloqué dans une configuration sous-optimale. En revanche, si le **bruit** atteint une amplitude comparable, voire supérieure, à celle du terme de correction, la convergence vers un point fixe est perturbée. Le système adopte alors une dynamique où les pondérations évoluent de façon erratique, ce qui se traduit par un comportement de **diffusion aléatoire** ou par un « mouvement brownien » autour du point d'équilibre.

Les **avantages** potentiels d'un bruit modéré résident dans sa capacité à permettre au SCN de s'extraire de minima locaux, favorisant ainsi l'exploration de configurations alternatives qui pourraient être plus adaptées. Toutefois, les **limites** de cette approche apparaissent lorsque le bruit devient trop important, car il perturbe significativement la stabilité du réseau et empêche la formation de **clusters** robustes, rendant ainsi l'ensemble de la structure instable et difficilement interprétable.

2.3.4.2. Résilience vs. Sensibilité : Cas d'Exemple

Dans l'analyse des **Synergistic Connection Networks (SCN)**, l'étude de la réponse du système à l'injection de bruit permet d'appréhender la robustesse de sa dynamique d'auto-organisation. On s'intéresse particulièrement à la manière dont le système réagit lorsque des perturbations aléatoires sont introduites dans le calcul de la **fonction de synergie** $S(i,j)$ ou directement dans l'équation de mise à jour des **pondérations** $\omega_{i,j}$. Pour illustrer ces phénomènes, considérons d'abord la règle de mise à jour dite « additive » qui constitue le cœur de la dynamique du SCN :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ représente le taux d'apprentissage et $\tau > 0$ le coefficient de décroissance. Dans le cas idéal, et en l'absence de perturbations, cette mise à jour conduit à une convergence locale des pondérations vers un **point fixe** théorique défini par

$$\omega_{i,j}^* \approx \frac{S(i,j)}{\tau}.$$

Pour modéliser l'impact des perturbations, on introduit un terme de bruit $\xi_{i,j}(t)$ dans la dynamique, ce qui donne l'équation suivante :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] + \xi_{i,j}(t).$$

Le terme $\xi_{i,j}(t)$ représente des fluctuations aléatoires qui viennent perturber la trajectoire de convergence de $\omega_{i,j}(t)$. La comparaison entre l'amplitude du terme de correction, $\eta [S(i,j) - \tau \omega_{i,j}(t)]$, et celle du bruit $\xi_{i,j}(t)$ détermine si le système affiche une **résilience** ou une **sensibilité** accentuée face aux perturbations.

Exemple de Résilience :

Supposons un mini-SCN composé de quatre entités, notées $\{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4\}$, pour lesquelles les synergies internes $S(1,2)$ et $S(3,4)$ sont élevées (par exemple, $S(1,2), S(3,4) > 0.7$), tandis que les synergies croisées telles que $S(1,3)$, $S(1,4)$, $S(2,3)$ et $S(2,4)$ restent faibles (typiquement < 0.2). Dans ce cas, la mise à jour itérative renforce fortement les pondérations internes, de sorte que les valeurs $\omega_{1,2}$ et $\omega_{3,4}$ atteignent des niveaux élevés (par exemple, entre 0.6 et 0.8), tandis que les liaisons entre les groupes restent faibles (inférieures à 0.1). Si l'on ajoute un bruit additif modeste, par exemple $\xi_{i,j}(t) \in [-0.02, +0.02]$, l'effet sur la dynamique demeure marginal. En d'autres termes, la structure de clusters constituée par les paires $\{\mathcal{E}_1, \mathcal{E}_2\}$ et $\{\mathcal{E}_3, \mathcal{E}_4\}$ reste stable, démontrant une **résilience** élevée du SCN face aux perturbations faibles.

Exemple de Sensibilité :

En revanche, considérons un scénario dans lequel le SCN comprend cinq entités $\{\mathcal{E}_1, \dots, \mathcal{E}_5\}$ et que la synergie interne d'un potentiel regroupement, par exemple $S(1,2) \approx 0.3$, est modérée et ne diffère pas significativement des synergies externes, qui se situeraient dans un intervalle approximatif de 0.1 à 0.2. Dans ce contexte, l'injection d'un bruit d'amplitude supérieure, disons $\xi_{i,j}(t) \in [-0.05, +0.1]$, peut provoquer des fluctuations plus importantes. La valeur de $\omega_{1,2}(t)$ peut alors chuter de valeurs stables initialement situées autour de 0.25–0.3 à des niveaux inférieurs à 0.15. Une telle réduction critique de la pondération interne conduit à la désagrégation du cluster potentiel, illustrant une **sensibilité** marquée aux perturbations. Ce cas démontre que lorsque la marge de différenciation entre les synergies internes et externes est faible, le SCN devient fragile et vulnérable à des fluctuations aléatoires, même si celles-ci ne sont pas très élevées.

Cas de Multi-Attracteurs :

Dans certains systèmes comportant plusieurs attracteurs, comme discuté en section 2.3.2.1, l'injection de bruit peut jouer un rôle double. D'un côté, un bruit de faible amplitude peut contribuer à maintenir la flexibilité du système en permettant des fluctuations autour d'un attracteur stable. D'un autre côté, un bruit persistant, même modéré, peut suffire à faire sortir le système de son bassin d'attraction initial, entraînant ainsi une transition vers une autre configuration attractrice.

Cette capacité à basculer entre différents états stables souligne la dualité du rôle du bruit : il favorise à la fois l'exploration de nouvelles configurations et, en excès, il peut perturber la convergence vers un état stable.

Synthèse et Conclusion :

La réaction d'un SCN à l'injection de bruit s'exprime sur un continuum allant de la **résilience** à la **sensibilité**. Lorsque l'amplitude du bruit $\xi_{i,j}(t)$ est faible par rapport à la force corrective $\eta [S(i,j) - \tau \omega_{i,j}(t)]$, le système maintient ses structures auto-organisées malgré des perturbations mineures. En revanche, si le bruit atteint ou dépasse l'ordre de grandeur de ce terme de correction, les fluctuations peuvent altérer significativement les pondérations, conduisant à une désorganisation des clusters. Ainsi, la stabilité globale du SCN dépend de la comparaison précise entre ces deux contributions. Ce phénomène souligne l'importance de paramétrier soigneusement les valeurs de η et τ pour assurer une dynamique robuste, comme évoqué dans les sections 2.3.1.3 et 2.3.2.1. En conclusion, le SCN peut démontrer une forte **résilience** en conservant ses regroupements lorsque le bruit est suffisamment faible, tandis qu'une **sensibilité** accrue, due à un bruit trop important ou à une faible différenciation entre synergies internes et externes, peut empêcher la stabilisation de structures cohérentes.

2.3.4.3. Introduction à l'idée qu'un certain niveau de bruit peut parfois aider l'exploration (analogie avec le recuit simulé)

Dans le cadre du **Synergistic Connection Network (SCN)**, l'injection d'un élément de **bruit** peut, de manière paradoxale, améliorer la capacité du système à explorer l'espace des configurations, en l'aider à échapper à des minima locaux qui ne sont pas nécessairement optimaux. Cette idée s'inspire du **recuit simulé** (simulated annealing) en optimisation, où l'on autorise initialement des fluctuations aléatoires significatives afin d'éviter un piégeage prématûr dans des configurations sous-optimales, avant de réduire progressivement l'intensité de ces perturbations pour permettre la stabilisation du système dans un état attracteur de haute qualité.

Considérons, à titre d'exemple, la règle de mise à jour additive fondamentale du SCN qui s'exprime sous la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ est le **taux d'apprentissage** et $\tau > 0$ représente le **coefficient de décroissance**. Pour intégrer l'effet de la **perturbation stochastique**, on modifie cette équation en ajoutant un terme aléatoire $\xi_{i,j}(t)$ qui est souvent modélisé par une distribution gaussienne de moyenne nulle et de variance dépendante d'une **température** $T(t)$:

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] + \xi_{i,j}(t), \quad \xi_{i,j}(t) \sim \mathcal{N}(0, \sigma^2(T(t))).$$

Au début de l'apprentissage, lorsque la **température** $T(0)$ est élevée, l'amplitude de $\xi_{i,j}(t)$ est grande, ce qui permet au réseau d'explorer un vaste espace de configurations. Cette phase d'exploration est essentielle pour éviter que le SCN ne se retrouve bloqué dans un minimum local qui pourrait être sous-optimal du point de vue de la **cohésion interne**. Au fil des itérations, un

refroidissement progressif de $T(t)$ est appliqué, ce qui réduit graduellement la variance $\sigma^2(T(t))$ du bruit. Ainsi, les fluctuations deviennent moins importantes et le système est amené à converger vers un arrangement plus stable, de sorte que, finalement, lorsque $T(t)$ tend vers zéro, l'évolution de $\omega_{i,j}(t)$ est dominée par le terme déterministe $\eta [S(i,j) - \tau \omega_{i,j}(t)]$.

L'analogie avec le **recuit simulé** se trouve dans la stratégie adoptée où, initialement, le système autorise des sauts aléatoires pouvant temporairement augmenter la valeur de la fonction « énergie » ou, dans notre cas, modifier les pondérations de manière non optimale, avant de réduire progressivement ces fluctuations afin de "figer" la configuration dans un minimum qui est supposé être de qualité supérieure. On peut résumer ce mécanisme par l'idée qu'un certain niveau de **bruit** agit comme un moyen d'évasion des minima locaux et, par conséquent, d'**exploration** accrue du paysage de la **synergie**.

Les avantages d'une telle approche résident dans la capacité du système à éviter un piégeage précoce, favorisant ainsi l'émergence de regroupements (clusters) qui présentent une meilleure **cohésion interne**. En revanche, si le bruit persiste à un niveau trop élevé, il risque de perturber la convergence, empêchant le SCN de stabiliser ses liens et, par conséquent, de former des structures cohérentes. Ce compromis met en évidence la nécessité d'un **décroissement progressif** de la température, analogue à la planification de recuit dans le recuit simulé, afin de passer d'une phase d'exploration à une phase de **stabilisation**.

2.3.5. Limites Théoriques et Questions Non Résolues

Les sections précédentes (2.3.1 à 2.3.4) ont présenté divers aspects de la **stabilité** dans un SCN, de la formation de clusters et des phénomènes de multi-attracteurs ou d'oscillations. Toutefois, la **théorie** du Deep Synergy Learning (DSL), envisagée comme un *système dynamique*, demeure largement **incomplète**. Il existe un ensemble d'**hypothèses** simplificatrices sous-jacentes et de **questions non résolues** quant à l'extension des modèles (apparition/disparition d'entités, synergie n-aire généralisée, etc.). La section 2.3.5 synthétise ces points, ouvrant la voie aux développements ultérieurs (notamment chapitres 9 et 12).

2.3.5.1. Quelles hypothèses fortes sont faites (ex. **synergie binaire**, **stationnarité**, etc.)

Dans le cadre du Deep Synergy Learning, l'analyse théorique repose sur un ensemble d'hypothèses fortes qui permettent de simplifier l'étude de la dynamique du réseau, en particulier pour établir des résultats concernant la convergence, la stabilité et l'émergence de clusters. Ces hypothèses concernent principalement la **stationnarité** de la fonction de synergie, les **bornes** sur les pondérations, la forme de la **règle de mise à jour** ainsi que la nature même de la synergie.

La première hypothèse concerne la **stationnarité** (ou quasi-stationnarité) de la fonction de synergie. En effet, on postule souvent que la fonction $S(i,j)$ ne varie pas de manière brusque ou aléatoire au cours du temps. Autrement dit, il est supposé que $S(i,j)$ peut être considérée comme constante ou, à tout le moins, comme évoluant de manière suffisamment régulière. Cette hypothèse se formalise soit par le fait que $S(i,j)$ dépend uniquement de paramètres invariants, tels que les représentations fixes \mathbf{x}_i et \mathbf{x}_j des entités, soit par la supposition que, lorsqu'une dépendance aux

États internes \mathbf{s}_i est prise en compte, la fonction $S(i, j)$ évolue à un rythme plus lent que les mises à jour des pondérations. Formellement, on peut exprimer cette hypothèse en admettant que, pour tout t et pour toute paire (i, j) , on a

$$S(i, j, t) \approx S(i, j),$$

ou bien que la variation temporelle, notée $\Delta S(i, j, t)$, est négligeable par rapport aux variations des poids.

Une autre hypothèse forte est celle concernant la **bornitude** des pondérations. On suppose qu'il existe un intervalle borné, souvent noté $[0, \omega_{\max}]$, dans lequel chaque $\omega_{i,j}(t)$ évolue. Cette contrainte empêche théoriquement la divergence des poids et facilite l'application de théorèmes classiques tels que le théorème de point fixe de Brouwer. On exprime cette hypothèse par l'inégalité

$$0 \leq \omega_{i,j}(t) \leq \omega_{\max}, \quad \forall i, j, \forall t,$$

ce qui permet également de considérer l'espace des configurations comme compact et convexe, une condition essentielle pour la convergence.

Une troisième hypothèse réside dans la **forme de mise à jour** des pondérations. Dans de nombreux travaux sur le DSL, la règle de mise à jour est adoptée sous une forme additive, généralement donnée par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ est le taux d'apprentissage et $\tau > 0$ est le coefficient de décroissance. Cette forme linéaire simplifie l'analyse, notamment la linéarisation locale autour d'un point fixe, permettant de montrer que, dans le cas stationnaire, le point fixe pour une liaison donnée est

$$\omega_{i,j}^* = \frac{S(i, j)}{\tau}.$$

L'hypothèse d'une mise à jour additive permet ainsi d'établir, par linéarisation, des conditions de stabilité via l'étude de la matrice Jacobienne. Cependant, il convient de noter que dans des versions plus avancées, des mises à jour de nature multiplicative ou intégrant des couplages n-aires apparaissent, ce qui rend la dynamique plus complexe et nécessite de relâcher cette hypothèse de linéarité.

Enfin, certaines simplifications interviennent dans la **nature** même de la synergie. Pour rendre l'analyse plus tractable, il est parfois supposé que $S(i, j)$ est de nature **binaire**, c'est-à-dire que $S(i, j) \in \{0, 1\}$. Une telle hypothèse permet de transformer la mesure continue de synergie en un indicateur discret qui détermine simplement si une connexion entre deux entités est présente ou non. Bien que cette approche puisse être très utile pour certaines applications (notamment lorsque l'on souhaite obtenir des structures denses et clairement définies), elle perd en finesse et en nuance, puisque la gradation de la synergie est alors remplacée par un seuil abrupt.

2.3.5.2. Ouvertures vers une gestion plus dynamique (entités qui apparaissent ou disparaissent)

Dans le cadre du **Deep Synergy Learning**, il est fondamental de considérer des scénarios où le nombre d'**entités** n'est pas fixe, mais évolue au fil du temps. Cette ouverture vers une gestion dynamique des entités reflète la réalité de nombreux systèmes complexes, tels que les réseaux de capteurs, les plateformes sociales ou les systèmes de robotique coopérative, dans lesquels de nouveaux éléments peuvent apparaître tandis que d'autres disparaissent ou deviennent inactifs. Cette dimension temporelle supplémentaire exige une extension du modèle de mise à jour des **pondérations** $\{\omega_{i,j}\}$ de sorte à permettre l'adaptation du **Synergistic Connection Network (SCN)\$ face à l'évolution du nombre d'unités.

A. Motivations et enjeux

Dans de nombreux domaines, la dynamique du système ne se limite pas à l'évolution des **liaisons** entre entités statiques, mais doit également intégrer la **naissance** et la **disparition** d'unités. Par exemple, dans un réseau de capteurs IoT, l'ajout de nouveaux dispositifs ou la défaillance de certains capteurs modifie la structure globale du réseau, de même qu'un réseau social, soumis à des flux continus d'inscriptions et de désinscriptions, présente une topologie en constante évolution. En robotique coopérative, la flotte de robots peut s'agrandir ou se réduire selon les besoins opérationnels ou les incidents techniques. Pour que le SCN puisse modéliser de telles situations, il faut autoriser la structure de **pondérations** à évoluer non seulement en réponse aux fluctuations de la **synergie** $S(i,j)$, mais également en fonction du nombre d'**entités** présentes.

B. Approche mathématique pour la gestion dynamique

Si l'on note $\Omega(t)$ l'ensemble des pondérations pour un réseau comportant initialement n entités, alors $\Omega(t)$ est un vecteur dont les composantes sont données par

$$\Omega(t) = (\omega_{i,j}(t) \mid i, j \in \{1, \dots, n\}).$$

Lorsqu'une nouvelle entité, que l'on notera \mathcal{E}_{n+1} , rejoint le réseau, l'espace d'état des pondérations s'étend naturellement. En effet, il devient nécessaire d'introduire de nouvelles variables, par exemple $\omega_{n+1,j}(t)$ et $\omega_{j,n+1}(t)$ pour $j = 1, \dots, n$, afin de décrire les interactions entre la nouvelle entité et celles déjà existantes. L'opérateur de mise à jour, qui était auparavant défini sur l'espace \mathcal{X}_n , doit être adapté pour opérer sur \mathcal{X}_{n+1} . Une telle extension se traduit par l'initialisation de ces nouvelles pondérations, par exemple par

$$\omega_{n+1,j}(0) \approx 0 \quad \text{et} \quad \omega_{j,n+1}(0) \approx 0,$$

afin de laisser à la dynamique du SCN le temps de déterminer, via la règle

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

la pertinence des liaisons entre la nouvelle entité et le reste du réseau. Inversement, lorsque qu'une entité \mathcal{E}_k disparaît ou devient inactif, on suppose que l'on met à zéro les pondérations correspondantes, c'est-à-dire que pour tout j , on fixe

$$\omega_{k,j}(t) = \omega_{j,k}(t) = 0,$$

et l'espace des pondérations se contracte de manière appropriée de \mathcal{X}_n vers \mathcal{X}_{n-1} . Cette procédure de retrait impose un rééquilibrage du réseau, car les liens qui dépendaient de l'entité disparue devront être redistribués parmi les entités restantes, pouvant entraîner une réorganisation de la structure des **clusters**.

C. Conséquences sur la stabilité et la formation de clusters

L'arrivée ou le départ d'**entités** constitue une perturbation dans la dynamique globale du SCN. Si, par exemple, une nouvelle entité \mathcal{E}_{n+1} manifeste une forte synergie avec certains sous-groupes existants, cela peut conduire à une reconfiguration des clusters. Une partie du cluster initial peut se scinder pour intégrer la nouvelle entité, ou bien les pondérations existantes peuvent être ajustées pour optimiser la cohésion globale. Dans le cas d'une entité qui se retire, l'équilibre du réseau est modifié où le cluster auquel elle appartenait peut se morceler ou fusionner avec un autre, selon l'importance de sa contribution initiale au niveau des **synergies** internes.

D'un point de vue algorithmique, deux stratégies principales peuvent être envisagées pour gérer ces fluctuations de dimension. La première consiste à adopter une approche de **continuité stricte**, dans laquelle l'insertion ou le retrait d'une entité s'effectue de manière incrémentale, permettant aux pondérations existantes de se réajuster progressivement sans nécessiter une réinitialisation complète du système. La seconde stratégie, appelée **réinitialisation partielle**, consiste à réinitialiser partiellement certaines pondérations lorsque le changement de dimension est jugé majeur, afin de faciliter la recomposition de la dynamique du réseau.

D. Exemples pratiques et implications

Considérons un scénario typique dans un réseau de capteurs où de nouveaux dispositifs sont ajoutés régulièrement. Dès l'insertion d'une nouvelle entité \mathcal{E}_{n+1} , les nouvelles pondérations $\omega_{n+1,j}$ sont initialisées à une valeur proche de zéro. La règle de mise à jour,

$$\omega_{n+1,j}(t+1) = \omega_{n+1,j}(t) + \eta [S(n+1,j) - \tau \omega_{n+1,j}(t)],$$

détermine ensuite la manière dont \mathcal{E}_{n+1} se connecte aux entités existantes, ce qui peut entraîner la formation de nouveaux clusters ou la fusion de clusters existants pour mieux intégrer la nouvelle information. Inversement, dans un réseau social, la disparition d'un utilisateur \mathcal{E}_k implique la mise à zéro de toutes les pondérations associées, ce qui peut provoquer une reconfiguration du cluster auquel il appartenait, modifiant ainsi la partition finale du réseau.

E. Limites théoriques et perspectives

L'ouverture vers une gestion dynamique des entités pose cependant plusieurs défis théoriques. En effet, lorsque la dimension de l'espace des pondérations n'est plus fixe, les outils classiques de la théorie des points fixes, qui reposent sur la compacité et la convexité de cet espace, deviennent partiellement inopérants. Le passage de \mathcal{X}_n à \mathcal{X}_{n+1} ou \mathcal{X}_{n-1} entraîne une modification de la structure même de l'opérateur F , ce qui nécessite le développement de méthodes incrémentales pour analyser la stabilité du système en évolution. De plus, l'inflation du nombre de liaisons, qui croît typiquement en $O(n)$ pour chaque nouvelle entité, peut entraîner une augmentation significative des coûts computationnels, ce qui impose souvent l'emploi de techniques de **sparsification** pour garantir la praticabilité du modèle.

Ces défis ouvrent des perspectives intéressantes pour le développement du **Deep Synergy Learning**. Des recherches futures pourraient explorer des stratégies de réinitialisation partielle ou des méthodes adaptatives de recalibrage des pondérations afin de conserver un équilibre entre la stabilité du réseau et sa capacité à s'adapter aux évolutions du nombre d'entités. Des travaux ultérieurs, notamment ceux qui seront abordés dans le Chapitre 9, se pencheront sur ces questions en profondeur, intégrant des approches de **mise à jour incrémentale** et des algorithmes **d'apprentissage continu** qui tiennent compte de l'apparition et de la disparition des entités.

2.3.5.3. Préfigure les Chapitres 9 et 12 sur l'apprentissage continu et la synergie n-aire

Dans les sections précédentes, notamment en **2.3.5.1** et **2.3.5.2**, nous avons établi un cadre théorique fondé sur des hypothèses fortes concernant la stationnarité de la **synergie** $S(i,j)$ ainsi que sur la gestion dynamique des entités dans un **Synergistic Connection Network** (SCN). Il apparaît désormais indispensable d'envisager deux axes majeurs d'extension du modèle, axes qui seront développés de manière approfondie dans les **Chapitre 9** et **Chapitre 12** de cet ouvrage. La première orientation consiste en l'**apprentissage continu** (ou *lifelong learning*), tandis que la seconde porte sur l'extension de la synergie aux interactions **n-aires**. Ces deux axes permettent d'élargir le domaine d'application du DSL et de prendre en compte des scénarios où la dynamique des entités et la complexité des interactions dépassent le cadre strictement binaire et stationnaire.

Dans le contexte de l'**apprentissage continu**, le SCN se trouve confronté à des environnements évolutifs, où de nouvelles données ou de nouvelles entités apparaissent, tandis que d'autres disparaissent ou deviennent inactives. Au lieu de supposer un nombre fixe d'entités – hypothèse qui permettait d'appliquer des théorèmes de convergence dans un espace compact et convexe – il faut maintenant envisager une extension de la dynamique du réseau. Plus précisément, si l'on note initialement par

$$\Omega(t) = (\omega_{1,2}(t), \omega_{1,3}(t), \dots, \omega_{n-1,n}(t))$$

la configuration des pondérations pour un ensemble de n entités, l'arrivée d'une nouvelle entité \mathcal{E}_{n+1} entraîne l'extension de l'espace d'état à

$$\Omega'(t) = (\omega_{1,2}(t), \dots, \omega_{n-1,n}(t), \omega_{n+1,1}(t), \omega_{n+1,2}(t), \dots, \omega_{n+1,n}(t)).$$

La règle de mise à jour demeure alors de la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

mais l'opérateur F doit désormais être défini sur un espace de dimension variable. Ce processus d'intégration progressive de nouvelles entités nécessite des mécanismes spécifiques pour éviter le phénomène de « retrait catastrophique » (catastrophic forgetting) et pour permettre une adaptation continue sans réinitialiser l'ensemble du réseau. C'est précisément l'objet du **Chapitre 9**, qui développera des stratégies d'apprentissage incrémental permettant de concilier la préservation des connaissances acquises avec l'adaptation à de nouveaux contextes, ainsi que des ajustements paramétriques dynamiques de η et τ pour optimiser la stabilité dans un environnement évolutif.

En parallèle, la généralisation de la synergie aux interactions **n-aires** représente une deuxième extension fondamentale. Jusqu'ici, le DSL s'est principalement appuyé sur une fonction de synergie définie pour des paires d'entités, c'est-à-dire $S(i, j)$. Or, de nombreuses applications requièrent de modéliser des interactions collectives impliquant trois, quatre ou un nombre plus grand d'entités simultanément. Dans ce cadre, il convient d'introduire une fonction générale

$$S(i_1, i_2, \dots, i_k),$$

qui mesure la **co-information** ou la **synergie collective** d'un groupe d'entités $\{\mathcal{E}_{i_1}, \mathcal{E}_{i_2}, \dots, \mathcal{E}_{i_k}\}$. Cette généralisation conduit naturellement au passage d'un graphe pondéré à un **hypergraphe**, dans lequel les hyper-liens représentent des interactions entre plusieurs nœuds de manière intrinsèquement non binaire. Formellement, il devient alors pertinent de définir des pondérations telles que

$$\omega_{(i_1, i_2, \dots, i_k)}(t+1) = \omega_{(i_1, i_2, \dots, i_k)}(t) + \eta [S(i_1, i_2, \dots, i_k) - \tau \omega_{(i_1, i_2, \dots, i_k)}(t)],$$

ce qui élargit considérablement le champ d'application du DSL. Le **Chapitre 12** se consacrera à formaliser ce concept de synergie **n-aire** et à étudier l'impact de ces interactions multiples sur l'auto-organisation et la stabilité globale du réseau. L'analyse portera notamment sur la manière dont ces hyper-liens influencent la formation de clusters et sur les nouveaux phénomènes dynamiques qui émergent dans un hypergraphe, en intégrant des aspects de co-information collective et d'interactions non linéaires plus sophistiquées.

En résumé, l'intégration de l'**apprentissage continu** et de la **synergie n-aire** constitue une avancée majeure qui préfigure un DSL de nouvelle génération, capable de s'adapter de manière incrémentale aux évolutions de son environnement tout en prenant en compte des interactions multi-entités. Ces deux axes d'extension ouvrent des perspectives particulièrement intéressantes pour des applications dans des domaines tels que les réseaux de capteurs, la robotique multi-agents ou encore les systèmes neuronaux inspirés de la biologie, où la flexibilité et la complexité des interactions sont essentielles. Les **Chapitre 9** et **Chapitre 12** développeront ces idées en profondeur, en proposant des modèles mathématiques rigoureux ainsi que des approches algorithmiques adaptées pour garantir la **stabilité** et la **cohérence** dans des environnements dynamiques et multidimensionnels.

Ainsi, cette section sert de pont vers une compréhension plus large du DSL, en préparant le terrain pour une gestion avancée de l'**apprentissage continu** et l'extension de la **synergie** au-delà du cadre binaire traditionnel, contribuant à l'émergence de réseaux véritablement **adaptatifs** et **complexes**.

2.4. Raccords avec la Physique Statistique et la Théorie des Systèmes Dynamiques

Le **Deep Synergy Learning** (DSL), vu sous l'angle d'un **réseau** $\{\omega_{i,j}\}$ évoluant sous des règles non linéaires, peut être relié à des approches classiques de la **physique statistique** (modèles de spins, énergies) et de la **théorie des systèmes dynamiques** (discrets ou continus). La section 2.4 initie ce rapprochement, en commençant (2.4.1) par l'interprétation de l'équation de mise à jour $\omega_{i,j}(t+1)$ comme un **système dynamique discret** et ses variantes continues, puis (2.4.2) et (2.4.3) feront des parallèles explicites avec les modèles de spin et la notion d'énergie ou de fonction potentielle.

2.4.1. Systèmes Dynamiques Discrets ou Continus

On a souvent présenté la mise à jour du SCN (voir 2.2.2) sous forme discrète :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j)\tau \omega_{i,j}(t)],$$

mais il est tout à fait possible de concevoir une version **continue** (2.4.1.2) ou même de tenter des analogies directes avec les **systèmes dynamiques** utilisés en physique statistique (2.4.2). Commençons (2.4.1.1) par voir en quoi $\omega_{i,j}(t+1) = F(\omega_{i,j}(t), \dots)$ s'interprète déjà comme un **système dynamique discret** et comment les outils de cette théorie peuvent s'appliquer.

2.4.1.1. Équation de mise à jour $\omega_{i,j}(t+1)$ vue comme un système dynamique discrétré

Dans la théorie des **systèmes dynamiques**, un modèle discret se représente typiquement par l'équation

$$\mathbf{x}(t+1) = F(\mathbf{x}(t)), \quad \mathbf{x} \in \mathbb{R}^D,$$

où chaque itération convertit l'état $\mathbf{x}(t)$ en $\mathbf{x}(t+1)$ au moyen d'une **application** (ou **opérateur**) F . Dans le cadre du **Deep Synergy Learning (DSL)**, l'état du réseau est encapsulé dans le vecteur $\Omega(t)$ qui regroupe l'ensemble des **pondérations** $\{\omega_{i,j}(t)\}$ et, éventuellement, d'autres variables d'état telles que les **états internes** des entités. On peut donc écrire

$$\Omega(t+1) = F(\Omega(t)),$$

où l'opérateur F intègre la **règle de mise à jour** du DSL, que l'on exprime généralement sous la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j; \Omega(t)) - \tau \omega_{i,j}(t)].$$

Dans cette équation, $S(i,j; \Omega(t))$ représente la **synergie effective** évaluée à l'itération t et peut dépendre non seulement des représentations intrinsèques des entités, telles que leurs **vecteurs**

caractéristiques \mathbf{x}_i ou leurs **états internes** \mathbf{s}_i , mais aussi de l'ensemble des pondérations $\omega_{p,q}(t)$ existant dans le réseau. Le paramètre $\eta > 0$ est le **taux d'apprentissage** qui détermine la vitesse d'évolution du système, tandis que $\tau > 0$ représente le **coefficients de décroissance**, essentiel pour éviter une croissance non bornée des pondérations et pour assurer la régulation du processus.

Si l'on considère un réseau composé de n entités et que les liens sont orientés, le nombre total de pondérations possibles est de l'ordre de $n(n - 1)$; par conséquent, le vecteur d'état $\boldsymbol{\Omega}(t)$ appartient à un espace de dimension $D \approx n(n - 1)$. L'opérateur F agit donc sur un espace de grande dimension, et sa définition se fait de manière à ce que, pour toute condition initiale $\boldsymbol{\Omega}(0)$ appartenant à un ensemble borné $\mathcal{X} \subset \mathbb{R}^D$ (par exemple, chaque $\omega_{i,j}$ étant contraint à l'intervalle $[0, \omega_{\max}]$), la suite $\{\boldsymbol{\Omega}(t)\}$ demeure dans \mathcal{X} . Autrement dit, la dynamique se décrit par

$$\boldsymbol{\Omega}(t + 1) = F(\boldsymbol{\Omega}(t)), \quad \text{avec } \boldsymbol{\Omega}(t) \in \mathcal{X},$$

et l'**orbite** ou la **trajectoire** du système est définie par l'ensemble $\{\boldsymbol{\Omega}(0), \boldsymbol{\Omega}(1), \boldsymbol{\Omega}(2), \dots\}$.

L'intérêt de cette formalisation réside dans la capacité à analyser, du point de vue de la **stabilité** et de la **convergence**, la dynamique des poids. En effet, comme indiqué dans la section 2.3.1.3, la stabilité locale d'un point fixe $\boldsymbol{\Omega}^*$ repose sur l'étude de la linéarisation de F autour de ce point. Lorsque la norme de la **matrice Jacobienne** $DF(\boldsymbol{\Omega}^*)$ est strictement inférieure à 1, c'est-à-dire

$$\| DF(\boldsymbol{\Omega}^*) \| < 1,$$

le point fixe $\boldsymbol{\Omega}^*$ se révèle stable ; sinon, des phénomènes tels que des cycles, des multi-attracteurs ou même des comportements pseudo-chaotiques peuvent apparaître (voir également la discussion des sections 2.3.2 et 2.3.3).

Une illustration théorique simple de la dynamique consiste à considérer le cas stationnaire où $S(i, j; \boldsymbol{\Omega}(t))$ est approximativement constant. Dans ce cas, la règle de mise à jour se simplifie en

$$\omega_{i,j}(t + 1) = (1 - \eta \tau) \omega_{i,j}(t) + \eta S(i, j),$$

ce qui, sous l'hypothèse de stabilité, conduit à la convergence des poids vers le point fixe

$$\omega_{i,j}^* = \frac{S(i, j)}{\tau}.$$

Cette condition de stabilité, qui se traduit par l'inégalité

$$|1 - \eta \tau| < 1,$$

implique que, dans un cas simple, le produit $\eta \tau$ doit satisfaire $0 < \eta \tau < 2$. Les **paramètres** η et τ sont donc cruciaux pour la bonne marche du système où un η trop élevé peut induire des oscillations, tandis qu'un τ trop grand risque d'amener les poids à s'écraser, limitant ainsi l'amplitude des interactions.

Par analogie avec des modèles de **spins** tels que le modèle d'**Ising** ou le réseau de **Hopfield**, qui sont souvent mis à jour de manière itérative (par exemple, via l'algorithme Metropolis-Hastings), le DSL peut être vu comme un système dynamique discret opérant sur le vecteur $(\omega_{1,2}, \omega_{1,3}, \dots, \omega_{n-1,n})$. Cette analogie s'étendra dans les sections 2.4.2 et 2.4.3, où seront établis

des liens supplémentaires avec la **fonction d'énergie** et les transitions de phase, offrant ainsi une vision énergétique de la dynamique du DSL.

Pour formaliser cette approche, nous pouvons énoncer le théorème suivant :

Théorème (Existence et Itération de la Dynamique Discrète)

Soit $\mathcal{X} \subset \mathbb{R}^D$ un ensemble **compact** (par exemple, chaque $\omega_{i,j}$ étant borné dans $[0, \omega_{\max}]$). Soit $F: \mathcal{X} \rightarrow \mathcal{X}$ une application **continue** qui définit la mise à jour du vecteur d'état selon

$$\Omega(t+1) = F(\Omega(t)).$$

Alors, pour toute condition initiale $\Omega(0) \in \mathcal{X}$, la suite $\{\Omega(t)\}_{t \geq 0}$ reste dans \mathcal{X} et, par le **théorème de Brouwer** (ou Schauder), il existe au moins un **point fixe** $\Omega^* \in \mathcal{X}$ tel que

$$\Omega^* = F(\Omega^*).$$

La stabilité ou la multiplicité de ce point fixe dépendra des propriétés locales de l'opérateur F , notamment de la norme $\|DF(\Omega^*)\|$.

En pratique, le fait de **borner** les valeurs de $\omega_{i,j}$ (par clipping ou par l'application de règles de **parsimonie**) et d'adopter une règle de mise à jour continue assure que la trajectoire $\Omega(0) \rightarrow \Omega(1) \rightarrow \Omega(2) \rightarrow \dots$ reste bien définie dans l'espace \mathcal{X} . Ainsi, la dynamique du DSL s'inscrit pleinement dans le cadre standard des systèmes dynamiques discrets, et la stabilité des clusters ainsi formés dépend essentiellement des propriétés de F .

2.4.1.2. Variante continue : $\frac{d\omega_{ij}}{dt} = \eta [S(i,j) - \tau \omega_{ij}]$

Dans la plupart des présentations du **Deep Synergy Learning (DSL)**, la mise à jour des pondérations se réalise en temps discret par des itérations successives, ce qui conduit à une équation de la forme

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)].$$

Cependant, il est tout à fait envisageable de modéliser l'évolution des pondérations par une **équation différentielle** en temps continu. Pour ce faire, on part de la différence discrète exprimée par

$$\frac{\omega_{ij}(t+1) - \omega_{ij}(t)}{\Delta t} = \eta [S(i,j) - \tau \omega_{ij}(t)],$$

et, en posant $\Delta t = 1$ puis en faisant tendre Δt vers zéro, la limite conduit à l'équation différentielle

$$\frac{d\omega_{ij}}{dt} = \eta [S(i,j) - \tau \omega_{ij}(t)].$$

Cette formulation permet d'envisager la dynamique des pondérations de manière « fluide » plutôt que par paliers discrets. Dans le cas où la **synergie** $S(i,j)$ est considérée comme constante (ou stationnaire), l'équation différentielle devient linéaire du premier ordre :

$$\frac{d \omega_{ij}}{dt} = \eta [S(i,j) - \tau \omega_{ij}(t)].$$

Pour une condition initiale $\omega_{ij}(0) = \omega_0$, la solution générale s'obtient par intégration et s'écrit ainsi :

$$\omega_{ij}(t) = \frac{S(i,j)}{\tau} + \left(\omega_0 - \frac{S(i,j)}{\tau} \right) \exp(-\eta \tau t).$$

Cette solution démontre que, lorsque $t \rightarrow +\infty$, la pondération converge de manière exponentielle vers le point fixe

$$\omega_{ij}^* = \frac{S(i,j)}{\tau},$$

la vitesse de convergence étant régie par le produit $\eta \tau$. Ce résultat rejoint exactement la conclusion obtenue dans le modèle discret, tout en offrant une interprétation continue et en ouvrant la voie à l'utilisation d'outils analytiques propres aux **équations différentielles ordinaires (EDO)**.

Lorsque la **synergie** $S(i,j)$ dépend non seulement de \mathbf{x}_i et \mathbf{x}_j mais également d'autres pondérations ou d'états internes (par exemple, lorsque S devient une fonction de l'ensemble $\{\omega_{p,q}\}$ ou des états \mathbf{s}_k), le système se généralise à un ensemble couplé d'équations différentielles :

$$\frac{d \omega_{ij}}{dt} = \eta [S(i,j; \{\omega_{p,q}\}) - \tau \omega_{ij}(t)], \quad \forall (i,j).$$

Le vecteur global $\boldsymbol{\Omega}(t)$, constitué de l'ensemble des pondérations, évolue alors dans un espace de dimension potentiellement élevée, souvent de l'ordre de $n(n - 1)$ si les liens sont orientés. Les propriétés de **continuité** et de **Lipschitz** de la fonction S garantissent, par le théorème de **Cauchy-Lipschitz**, l'existence et l'unicité locale des solutions. La stabilité d'un point fixe $\boldsymbol{\Omega}^*$ dans ce contexte se caractérise par l'étude de la linéarisation de l'opérateur F associé à la dynamique continue, qui se traduit par

$$\delta \boldsymbol{\Omega}'(t) = DF(\boldsymbol{\Omega}^*) \delta \boldsymbol{\Omega}(t),$$

où $DF(\boldsymbol{\Omega}^*)$ désigne la **matrice Jacobienne** évaluée en l'équilibre. Si toutes les valeurs propres de cette Jacobienne ont des parties réelles strictement négatives, l'équilibre est localement attractif.

Il convient de noter que, dans la pratique, bien que le DSL soit souvent implémenté en temps discret avec un pas Δt très petit (par exemple, entre 0.01 et 0.1), l'analyse en temps continu offre un cadre conceptuel riche, permettant d'établir des analogies avec des modèles de **spins** et d'autres systèmes neuronaux (comme ceux de **Hopfield** ou de **Cohen-Grossberg**). Ce passage à une formulation continue rapproche ainsi le DSL des modèles de systèmes dynamiques continus rencontrés en physique et en biologie, enrichissant l'analyse théorique et facilitant l'étude des propriétés de convergence et de stabilité.

La variante continue du DSL s'exprime par

$$\frac{d \omega_{ij}}{dt} = \eta [S(i,j) - \tau \omega_{ij}],$$

ce qui, dans le cas stationnaire, implique que $\omega_{ij}(t)$ converge exponentiellement vers le point fixe $\omega_{ij}^* = \frac{S(i,j)}{\tau}$. Dans le cas général, où $S(i,j)$ dépend des pondérations ou des états internes, le système se complexifie en un ensemble couplé d'équations différentielles dont l'analyse repose sur des techniques classiques de linéarisation et d'étude de la stabilité par la Jacobienne. Cette formulation continue offre une perspective alternative et complémentaire à la dynamique discrète, et permet de mieux comprendre les analogies entre le DSL et les modèles neuronaux ou les systèmes de spins étudiés en physique mathématique.

2.4.1.3. Conditions d'existence d'une solution fermée ou d'une approximation

Dans la version continue du **Deep Synergy Learning (DSL)**, la dynamique des pondérations $\omega_{ij}(t)$ est régie par un système d'équations différentielles ordinaires (EDO) généralement écrit sous la forme

$$\frac{d \omega_{ij}}{dt} = \eta [S(i,j; \{\omega_{p,q}\}, \dots) - \tau \omega_{ij}(t)],$$

où $\eta > 0$ représente le **taux d'apprentissage**, $\tau > 0$ le **coefficent de décroissance** et $S(i,j; \{\omega_{p,q}\}, \dots)$ décrit la **synergie effective** qui peut, selon les cas, dépendre de l'ensemble des pondérations $\{\omega_{p,q}\}$ et d'autres variables (par exemple, les états internes des entités). L'objectif est de déterminer, d'une part, dans quels cas on peut obtenir une **solution fermée** (analytique) pour ces équations et, d'autre part, quelles méthodes théoriques et numériques permettent de prouver l'**existence**, l'**unicité** et la **qualité** des solutions lorsque la solution fermée n'est pas accessible.

A. Solutions fermées dans les cas simples

Lorsque la fonction de synergie $S(i,j)$ est supposée **constante** ou **indépendante** de l'ensemble des pondérations $\{\omega_{p,q}\}$, l'équation se simplifie en une équation différentielle linéaire du premier ordre :

$$\frac{d \omega_{ij}}{dt} = \eta [S(i,j) - \tau \omega_{ij}(t)].$$

Cette équation est classique et se résout par séparation des variables ou par la méthode de variation des constantes. En imposant la condition initiale $\omega_{ij}(0) = \omega_0$, la solution analytique se trouve être

$$\omega_{ij}(t) = \frac{S(i,j)}{\tau} + \left(\omega_0 - \frac{S(i,j)}{\tau} \right) \exp(-\eta \tau t).$$

On constate ainsi que, pour $t \rightarrow +\infty$, la pondération converge exponentiellement vers le point fixe

$$\omega_{ij}^* = \frac{S(i,j)}{\tau},$$

ce qui confirme la validité de cette solution fermée dans le cas linéaire stationnaire. De même, si l'on suppose que $S(i,j)$ dépend de manière linéaire des pondérations et reste stationnaire, le système se réduit à un système linéaire de la forme

$$\mathbf{x}'(t) = M \mathbf{x}(t) + \mathbf{b},$$

où l'on peut alors résoudre le système par diagonalisation de la matrice M . Ces cas simples illustrent que, sous certaines hypothèses restrictives, une **solution fermée** est effectivement accessible.

B. Cas non linéaire : S dépendant de $\{\omega_{p,q}\}$

Dans la plupart des scénarios pratiques du DSL, la synergie $S(i,j; \{\omega_{p,q}(t)\}, \dots)$ intègre des **non-linéarités** dues à des fonctions de distance, des mesures de co-information ou à des mécanismes d'inhibition compétitive. Dans ces cas, l'équation devient un système couplé non linéaire :

$$\frac{d\omega_{ij}}{dt} = \eta [S(i,j; \{\omega_{p,q}(t)\}) - \tau \omega_{ij}(t)], \quad \forall(i,j).$$

Ce système, que l'on peut écrire sous la forme

$$\boldsymbol{\Omega}'(t) = \mathbf{F}(\boldsymbol{\Omega}(t)),$$

avec $\boldsymbol{\Omega}(t) \in \mathbb{R}^D$ et $D \approx n(n - 1)$ dans le cas d'un réseau orienté, ne permet généralement pas d'obtenir une solution fermée. Cependant, si la fonction \mathbf{F} est **localement Lipschitz**, le théorème de **Cauchy–Lipschitz** (ou théorème d'existence et d'unicité) garantit qu'une solution unique existe pour chaque condition initiale $\boldsymbol{\Omega}(0)$. En pratique, l'intégration de telles équations non linéaires requiert le recours à des méthodes d'**approximation numérique** ou à des techniques d'analyse qualitative telles que la linéarisation autour d'un point fixe.

C. Méthodes numériques et linéarisation

Pour étudier ou simuler la trajectoire de $\omega_{ij}(t)$ dans le cas non linéaire, plusieurs schémas numériques peuvent être employés. Par exemple, la méthode d'**Euler** consiste à approximer, pour un pas de temps δt suffisamment petit,

$$\omega_{ij}(t + \delta t) \approx \omega_{ij}(t) + \delta t \eta [S(i,j) - \tau \omega_{ij}(t)],$$

ce qui rapproche cette méthode de la version discrète étudiée en section 2.4.1.1. Des méthodes plus sophistiquées, telles que les schémas de **Runge–Kutta** d'ordre 2 ou 4, offrent une meilleure stabilité numérique et une précision accrue en évaluant l'opérateur \mathbf{F} en plusieurs points intermédiaires.

Par ailleurs, une technique d'analyse qualitative repose sur la **linéarisation** du système autour d'un **point fixe** $\boldsymbol{\Omega}^*$. Si l'on note la perturbation $\delta\boldsymbol{\Omega}(t) = \boldsymbol{\Omega}(t) - \boldsymbol{\Omega}^*$, on a

$$\frac{d}{dt} \delta\boldsymbol{\Omega}(t) = \mathbf{DF}(\boldsymbol{\Omega}^*) \delta\boldsymbol{\Omega}(t),$$

où $\mathbf{DF}(\Omega^*)$ est la **matrice Jacobienne** évaluée au point fixe. L'analyse des valeurs propres de cette Jacobienne permet de déterminer la **stabilité locale** où si toutes les valeurs propres ont une partie réelle négative, Ω^* est un attracteur local stable, sinon, des comportements tels que des cycles ou du pseudo-chaos peuvent apparaître.

D. Bornage et évitemment d'explosions

Il est fréquent, pour éviter la divergence des solutions, d'imposer une condition de **bornitude** sur les pondérations, par exemple en contraignant

$$\omega_{ij}(t) \in [0, \omega_{\max}],$$

ce qui fait vivre $\Omega(t)$ dans un ensemble $\mathcal{X} \subset \mathbb{R}^D$ **compact**. Si l'opérateur \mathbf{F} est continu et renvoie \mathcal{X} dans \mathcal{X} , on obtient alors une existence globale de la solution, et les théorèmes de point fixe (tels que le théorème de Brouwer ou celui de Schauder) assurent l'existence d'au moins un attracteur dans l'espace des pondérations. Même en l'absence de bornes strictes, le terme de décroissance $-\tau \omega_{ij}(t)$ joue un rôle régulateur, empêchant une croissance incontrôlée des pondérations, à condition que la synergie $S(i, j)$ ne soit pas pathologiquement infinie.

2.4.2. Parallèles avec les Modèles de Spin (Ising, Potts, Hopfield)

Les **modèles de spin** (Ising, Potts, Hopfield) en physique statistique et en neurosciences computationnelles partagent avec le **DSL** (Deep Synergy Learning) une approche où des **entités** interagissent via des **couplages** susceptibles de se renforcer ou de conduire à des configurations d'énergie plus ou moins stable. La section 2.4.2 met en évidence ces similitudes, soulignant à la fois l'inspiration possible pour le DSL et les limites de l'analogie. Nous commençons (2.4.2.1) par la notion de **minima énergétiques** et de **transitions de phase** dans un réseau, avant d'évoquer (2.4.2.2) la comparaison $\omega_{ij} \leftrightarrow$ couplage spin/neuron, et de modérer (2.4.2.3) la portée de ce parallèle du fait de la **non-linéarité** évolutive de la synergie en DSL.

2.4.2.1. Notion de minima énergétiques, transitions de phase dans un réseau

La modélisation des systèmes complexes, qu'ils soient issus des modèles de spins classiques ou des réseaux neuronaux auto-organisés tels que le **Deep Synergy Learning (DSL)**, s'appuie souvent sur l'idée d'un paysage d'**énergie** comportant divers **minima**. Dans les modèles de spins, comme celui d'**Ising**, de **Potts** ou le réseau de **Hopfield**, la fonction d'énergie (ou **Hamiltonien**) définit une configuration stable ou un attracteur du système. Par exemple, dans le modèle d'Ising, chaque spin s_i (avec $s_i \in \{-1, +1\}$) est associé à une énergie donnée par

$$\mathcal{H}(\{s_i\}) = - \sum_{\langle i,j \rangle} J_{ij} s_i s_j - \sum_i h_i s_i,$$

où la somme $\langle i,j \rangle$ se fait sur les paires de spins voisins et J_{ij} désigne le **couplage** entre les spins i et j . Dans le modèle de **Potts**, qui généralise le modèle d'Ising à q états, le principe reste le même

où l'alignement des spins sur une valeur commune est favorisé, conduisant à des configurations d'énergie minimale. Le réseau de **Hopfield**, quant à lui, définit sa fonction d'énergie par

$$\mathcal{H} = -\frac{1}{2} \sum_{i,j} W_{ij} s_i s_j,$$

où W_{ij} représentent les poids ou **coupages synaptiques**, et les minima de cette fonction correspondent aux états mémorisés ou aux attracteurs stables du système.

Dans le cadre du DSL, bien que la dynamique ne soit pas explicitement définie en termes d'un Hamiltonien classique, l'évolution des **pondérations** ω_{ij} peut être interprétée comme la recherche d'un état d'**optimisation** du réseau. Par analogie, on peut définir une pseudo-énergie associée aux pondérations par

$$\mathcal{E}(\Omega) \approx - \sum_{(i,j)} \omega_{ij} S(i,j),$$

où $S(i,j)$ représente la **synergie** entre les entités \mathcal{E}_i et \mathcal{E}_j . Dans cette optique, renforcer une liaison w_{ij} lorsque la synergie $S(i,j)$ est élevée équivaut à abaisser l'énergie du système, ce qui est favorable du point de vue de l'**auto-organisation**. Les **minima énergétiques** ainsi définis correspondent à des configurations stables où l'ensemble des pondérations converge vers un état attracteur, semblable aux minima locaux ou globaux observés dans les modèles de spins.

Un aspect crucial dans l'étude de ces systèmes est l'apparition de **transitions de phase**. Dans le modèle d'Ising, par exemple, on observe une transition d'un état désordonné (où les spins sont aléatoirement orientés) à un état ordonné (où la majorité des spins s'alignent) lorsque la température T chute en dessous d'un certain seuil critique T_c . Par analogie, dans un SCN, si l'on fait varier un paramètre équivalent à la température – par exemple, le niveau de bruit injecté dans la dynamique ou le rapport η/τ – le système peut passer d'un régime où les pondérations sont trop fluctuantes pour permettre une structure stable à un régime où des **clusters** bien définis émergent. Cette transition, caractérisée par une « brisure de symétrie » dans l'organisation des pondérations, est analogue à la transition ordre-désordre des modèles de spins.

Les exemples de **minima énergétiques** se retrouvent, dans un DSL simplifié, lorsque l'on impose une synergie binaire ou continue mais stationnaire, ce qui permet de résoudre analytiquement l'équation de mise à jour continue (comme vu en section 2.4.1.2) et d'obtenir le point fixe

$$\omega_{ij}^* = \frac{S(i,j)}{\tau}.$$

Dans un cadre plus complexe, où $S(i,j)$ dépend de manière non linéaire des pondérations ou d'états internes, le système présente des **attracteurs multiples** et le paysage d'énergie devient riche en minima locaux. Chaque minimum représente une configuration stable locale du réseau, correspondant à un cluster ou à une partition particulière de l'ensemble des entités. Les **transitions de phase** se manifestent alors par des changements brusques dans l'organisation du réseau lorsqu'un paramètre (tel que le bruit ou le rapport η/τ) atteint un seuil critique, conduisant à un basculement d'un attracteur à un autre.

En résumé, les modèles de spins – Ising, Potts, Hopfield – nous fournissent un vocabulaire riche qui peut être transposé au DSL, où l'on parle de **minima énergétiques** et de **transitions de phase** pour interpréter les configurations stables des pondérations. Un **minimum local** d'énergie dans le DSL correspond à une configuration des poids Ω qui maximise la synergie interne, et la transition d'un état diffus à un état clusterisé s'interprète comme une transition ordre-désordre. Bien que la synergie $S(i,j)$ dans le DSL puisse varier dans le temps ou dépendre d'états internes, l'analogie avec la minimisation d'un Hamiltonien offre une perspective éclairante pour comprendre l'émergence des **clusters** et l'existence d'attracteurs dans l'espace des pondérations.

2.4.2.2. Interprétation de ω_{ij} comme “couplage” entre spins ou neurones

Dans les modèles classiques de la physique statistique, tels que les modèles de **spins d'Ising** ou de **Potts**, ainsi que dans les réseaux de **Hopfield**, la notion de couplage est centrale. Dans le modèle d'Ising, par exemple, chaque spin s_i prenant des valeurs dans $\{-1, +1\}$ interagit avec ses voisins via des paramètres J_{ij} qui, selon leur signe, favorisent l'alignement ou l'antialignement. La fonction d'énergie (Hamiltonien) s'exprime typiquement sous la forme

$$\mathcal{H}(\{s_i\}) = - \sum_{\langle i,j \rangle} J_{ij} s_i s_j - \sum_i h_i s_i,$$

où la somme $\langle i,j \rangle$ parcourt les paires de spins voisins et h_i représente un champ externe appliqué sur le site i . Dans le modèle de Potts, la généralisation aux q états se traduit par une évaluation similaire de la similitude ou de la différence d'états entre spins, tandis que dans le réseau de Hopfield, l'énergie est définie par

$$E(\mathbf{s}) = -\frac{1}{2} \sum_{i,j} W_{ij} s_i s_j,$$

les poids W_{ij} constituant alors les **coupages synaptiques** qui restent fixés au cours du temps et dont les minima énergétiques correspondent aux états stables du système (mémoires associatives).

Dans le contexte du **Deep Synergy Learning (DSL)**, la pondération ω_{ij} remplit une fonction analogue à celle des coupages J_{ij} ou W_{ij} dans les modèles de spins. En effet, ω_{ij} quantifie la force du lien entre deux entités \mathcal{E}_i et \mathcal{E}_j . Une valeur élevée de ω_{ij} signifie que la synergie entre les entités, mesurée par $S(i,j)$, est forte et que le lien entre elles est donc très robuste. La dynamique du DSL repose sur une mise à jour de type

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)],$$

ce qui signifie que la pondération évolue en fonction de la différence entre la synergie effective et un terme de décroissance proportionnel à ω_{ij} . À cet égard, ω_{ij} joue un rôle similaire à un couplage adaptatif. En effet, dans les réseaux de Hopfield ou d'Ising, le couplage entre deux spins est un paramètre constant qui oriente l'alignement des états s_i et s_j . Dans le DSL, ce sont les **liaisons** elles-mêmes qui évoluent en temps réel en réponse aux mesures de synergie, de sorte que ω_{ij} est

non seulement un indicateur de la force de connexion, mais également un paramètre qui se met à jour pour refléter l'état d'interaction actuel entre les entités.

Une autre différence fondamentale réside dans le fait que, dans les modèles classiques, l'énergie du système est exprimée directement en fonction des spins, tandis que dans le DSL, l'analogie se construit en définissant une pseudo-énergie du réseau. Par exemple, on peut envisager la fonction

$$\mathcal{E}(\Omega) \approx - \sum_{(i,j)} \omega_{ij} S(i,j),$$

qui traduit l'idée que renforcer les pondérations ω_{ij} lorsque $S(i,j)$ est élevée réduit l'énergie globale du système, à la manière dont un couplage J_{ij} élevé favorise un état ordonné dans un modèle de spins. Cette interprétation permet d'assimiler, d'une certaine manière, le rôle de ω_{ij} à celui d'un couplage classique, même si la synergie $S(i,j)$ dans le DSL peut résulter de calculs plus complexes qu'un simple produit scalaire ou une simple corrélation.

Par ailleurs, la dynamique DSL se distingue par le fait que, contrairement aux modèles d'Ising ou de Hopfield où les couplages sont fixes et ce sont les états (spins ou activations) qui évoluent, dans le DSL, ce sont les **liaisons** elles-mêmes qui évoluent tandis que l'état interne de chaque entité peut rester relativement stable ou évoluer indépendamment. Cette inversion dans le rôle des variables signifie que le DSL est axé sur l'adaptation des **relations** entre les entités, ce qui permet au réseau de se réorganiser de manière flexible en réponse à de nouvelles informations.

Enfin, il convient d'envisager les extensions possibles de cette analogie vers des interactions plus complexes. Tandis que les modèles classiques de spins se limitent à des couplages binaires, le DSL ouvre la possibilité d'introduire des interactions **n-aires**. Ces interactions, qui impliquent la synergie collective de plusieurs entités, peuvent être représentées dans un cadre d'**hypergraphes** où les hyper-liens traduisent des relations qui ne se décomposent pas simplement en interactions par paires. Même si cette généralisation dépasse le cadre des modèles de spins classiques, elle enrichit la compréhension du DSL en offrant une perspective plus large sur la **coopération** entre entités.

2.4.2.3. Limites de la comparaison : le DSL introduit une synergie “non-linéaire” pouvant varier au fil du temps

Dans les sections précédentes, les analogies établies entre le **DSL** (Deep Synergy Learning) et les modèles de spin classiques – notamment ceux d'Ising, de Potts ou de Hopfield – se fondent sur l'assimilation de la pondération ω_{ij} à un **couplage** comparable aux paramètres J_{ij} ou W_{ij} . Dans ces modèles, le couplage est généralement considéré comme une constante fixe, tandis que dans le DSL, la dynamique des pondérations se fait évoluer en temps réel. Toutefois, cette comparaison présente des limites majeures, car la **synergie** $S(i,j)$ dans le DSL est souvent caractérisée par une nature **non-linéaire** et par une variabilité temporelle qui ne se retrouve pas dans les modèles de spins stationnaires.

Dans les modèles classiques tels que ceux d'Ising ou de Potts, chaque spin $s_i \in \{-1, +1\}$ interagit avec ses voisins via des couplages J_{ij} qui sont fixés dans le temps, sauf dans certaines variantes où

l'on peut envisager une plasticité synaptique. Par exemple, l'énergie du modèle d'Ising est donnée par

$$\mathcal{H}(\{s_i\}) = - \sum_{\langle i,j \rangle} J_{ij} s_i s_j - \sum_i h_i s_i,$$

où la matrice $\{J_{ij}\}$ est supposée constante, permettant ainsi une descente d'énergie par l'évolution des spins $\{s_i\}$. Dans un réseau de Hopfield, les poids $\{W_{ij}\}$ sont également fixés après apprentissage et la dynamique repose sur l'évolution des activations s_i pour minimiser la fonction d'énergie

$$\mathcal{H}(\mathbf{s}) = -\frac{1}{2} \sum_{i,j} W_{ij} s_i s_j.$$

En contraste, dans le DSL, la pondération ω_{ij} évolue en fonction de la synergie mesurée, par exemple selon une règle de mise à jour discrète

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)],$$

où η est le **taux d'apprentissage** et τ le **coefficent de décroissance**. Ici, la synergie $S(i,j)$ peut être issue de diverses mesures (distance, similarité, co-information) et, contrairement aux couplages classiques, elle peut dépendre de variables évolutives telles que les représentations \mathbf{x}_i des entités ou même des états internes \mathbf{s}_i . De plus, dans certaines variantes du DSL, $S(i,j)$ intègre des termes dépendant d'autres pondérations, par exemple sous la forme d'un terme d'**inhibition compétitive** tel que

$$S(i,j) \propto -\gamma \sum_{k \neq j} \omega_{i,k},$$

ce qui introduit une non-linéarité supplémentaire dans la dynamique. Par conséquent, la fonction $S(i,j)$ ne se réduit pas à un simple produit scalaire $s_i \times s_j$ comme dans les modèles de spins, mais peut varier en fonction des **représentations** et évoluer en temps réel.

Cette variabilité pose des problèmes pour établir une analogie rigoureuse avec une fonction d'énergie stationnaire. En effet, dans un système tel que celui de Hopfield, l'énergie

$$\mathcal{H}(\mathbf{s}) = -\frac{1}{2} \sum_{i,j} W_{ij} s_i s_j$$

reste fixe en fonction des couplages W_{ij} tandis que les spins évoluent. Dans le DSL, en revanche, les pondérations ω_{ij} ainsi que la synergie $S(i,j)$ se réévaluent continuellement, ce qui rend difficile l'établissement d'un paysage d'énergie immuable, c'est-à-dire d'une fonction $\mathcal{E}(\Omega)$ telle que

$$\mathcal{E}(\Omega) \approx - \sum_{(i,j)} \omega_{ij} S(i,j)$$

soit une mesure stable sur laquelle le système effectuerait une descente d'énergie. Par ailleurs, la dynamique du DSL se caractérise par le fait que les entités elles-mêmes, représentées par leurs vecteurs \mathbf{x}_i ou leurs états internes \mathbf{s}_i , sont relativement stables par rapport aux pondérations. Ainsi,

dans les modèles classiques, ce sont les spins qui varient tandis que le couplage reste fixe, tandis que dans le DSL, le couplage ω_{ij} évolue, ce qui inverse la dynamique habituelle.

De plus, la synergie $S(i,j)$ peut être influencée par des mécanismes complexes et non linéaires, tels que la dépendance aux états internes ou l'intégration d'une co-information multi-entités. Par conséquent, le paysage d'énergie du DSL est en constante évolution, et une configuration de pondérations optimale à un instant donné peut se transformer à l'instant suivant si la synergie se modifie. Ce caractère dynamique et non stationnaire limite la validité d'une comparaison stricte avec des modèles de spins classiques, où l'on dispose d'un Hamiltonien fixe et d'une descente d'énergie bien définie.

En conclusion, bien que l'analogie entre ω_{ij} dans le DSL et les couplages J_{ij} ou W_{ij} des modèles de spins fournit un langage utile pour comprendre la formation des clusters et la multi-stabilité du système, elle reste partielle. Le DSL introduit une **synergie non-linéaire** et variable dans le temps, ce qui complexifie la notion d'un paysage d'énergie statique et remet en question l'application directe des concepts de descente d'énergie et de transitions de phase issus de la physique statistique. Néanmoins, cette analogie demeure précieuse pour interpréter qualitativement les phénomènes d'**attraction** et de **basculement** observés dans le DSL, tout en reconnaissant que la dynamique de ce système dépasse largement la structure d'un couplage fixe et d'un Hamiltonien stationnaire.

2.4.3. Notion d'Énergie ou de Fonction Potentielle

Les **modèles de spin** (Ising, Potts, Hopfield) s'appuient sur une **énergie** \mathcal{H} (ou \mathcal{E}) que l'évolution du système vise à **minimiser** ou dont il cherche à **explorer** les minima. Dans le cadre du **Deep Synergy Learning** (DSL), on aimerait parfois définir un **anologue** de cette fonction d'énergie ou de potentiel, afin de mieux comprendre la **stabilité** et la **convergence** du SCN (Synergistic Connection Network). La section 2.4.3 soulève la question de savoir si on peut formaliser $\mathcal{J}(\Omega)$ (2.4.3.1), examiner l'existence de "descente d'énergie" (2.4.3.2) et évaluer les avantages/inconvénients de cette approche (2.4.3.3).

2.4.3.1. Peut-on formaliser $\mathcal{J}(\Omega) = - \sum_{i,j} \omega_{ij} S(i,j) + \dots ?$

Dans l'étude du **Deep Synergy Learning (DSL)**, une question centrale consiste à savoir si la dynamique des pondérations ω_{ij} peut être interprétée comme une descente d'énergie, analogue à la minimisation d'un Hamiltonien dans les modèles de spins. Pour ce faire, on cherche à définir une fonction de coût ou une **fonction énergétique** $\mathcal{J}(\Omega)$ dont la minimisation conduirait naturellement aux mises à jour observées dans le DSL. Une proposition heuristique couramment avancée est de poser

$$\mathcal{J}(\Omega) = - \sum_{i,j} \omega_{ij} S(i,j) + (\text{termes de régularisation}),$$

où $S(i,j)$ désigne la **synergie effective** entre les entités \mathcal{E}_i et \mathcal{E}_j . Le premier terme, $-\omega_{ij} S(i,j)$, est conçu pour encourager l'augmentation des pondérations lorsque la synergie est élevée, c'est-à-dire

qu'en minimisant \mathcal{J} on favorise une configuration dans laquelle les liens forts correspondent aux synergies importantes. Pour éviter que les pondérations ne croissent indéfiniment, on ajoute généralement un terme de régularisation qui peut prendre la forme d'une pénalisation quadratique, par exemple

$$\frac{\tau}{2} \sum_{i,j} \omega_{ij}^2,$$

de sorte que l'expression complète devient

$$\mathcal{J}(\boldsymbol{\Omega}) = - \sum_{i,j} \omega_{ij} S(i,j) + \frac{\tau}{2} \sum_{i,j} \omega_{ij}^2 + \dots,$$

où $\tau > 0$ agit comme un **paramètre de régularisation**. Le gradient partiel par rapport à une pondération ω_{ij} s'obtient alors en différentiant \mathcal{J} par rapport à ω_{ij} , ce qui conduit à

$$\frac{\partial \mathcal{J}}{\partial \omega_{ij}} = -S(i,j) + \tau \omega_{ij}.$$

Ainsi, si l'on interprète la règle de mise à jour du DSL comme un pas de gradient négatif, on obtient

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \eta \frac{\partial \mathcal{J}}{\partial \omega_{ij}}|_{\omega_{ij}(t)} = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)],$$

ce qui correspond exactement à la mise à jour standard utilisée dans le DSL dans le cas stationnaire où $S(i,j)$ est indépendant de ω_{ij} et des autres pondérations.

Cette correspondance formelle permet de comprendre que, dans un cadre idéal et simplifié, le DSL opère comme une **descente de gradient** sur la fonction \mathcal{J} , ce qui aurait pour effet de minimiser l'énergie globale du réseau. L'hypothèse sous-jacente est que la synergie $S(i,j)$ est stationnaire ou du moins qu'elle varie lentement par rapport aux mises à jour des pondérations. Dans ce cas, le paysage énergétique défini par

$$\mathcal{J}(\boldsymbol{\Omega}) = - \sum_{i,j} \omega_{ij} S(i,j) + \frac{\tau}{2} \sum_{i,j} \omega_{ij}^2$$

possède un minimum global que le système cherche à atteindre, avec pour point fixe

$$\omega_{ij}^* = \frac{S(i,j)}{\tau},$$

ce qui reflète une convergence exponentielle, comme nous l'avons vu dans l'analyse de la variante continue.

Cependant, des **obstacles** apparaissent dès lors que l'on introduit des éléments de non-linéarité dans la fonction de synergie. Lorsque $S(i,j)$ dépend de manière non triviale des autres pondérations $\{\omega_{p,q}\}$ ou d'états internes variables, le terme $S(i,j)$ devient lui-même une fonction de $\boldsymbol{\Omega}$ et ne peut plus être considéré comme une constante. Dans ce cas, la dérivation de \mathcal{J} ne se réduit plus à une simple fonction linéaire en ω_{ij} et l'opérateur de mise à jour

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j; \{\omega_{p,q}(t)\}) - \tau \omega_{ij}(t)]$$

ne correspond plus à une descente de gradient d'une unique fonction énergétique $\mathcal{J}(\Omega)$ fixe. Le paysage énergétique devient alors **dynamique** et se reconfigure en continu, reflétant des interactions complexes et souvent non stationnaires. En d'autres termes, la fonction $S(i,j)$ peut évoluer en réponse aux changements dans Ω et aux entrées extérieures, ce qui empêche la formulation d'un Hamiltonien fixe et stable comme dans les modèles de spins classiques.

Dans cette situation, il est souvent nécessaire de recourir à des **approches numériques** pour étudier l'évolution du système et pour déterminer qualitativement l'existence et la nature des attracteurs. On peut ainsi utiliser des schémas d'intégration numérique (comme Euler ou Runge–Kutta) pour approximer la trajectoire $\omega_{ij}(t)$, et des techniques de **linéarisation** autour des points fixes pour analyser la stabilité locale à l'aide de la matrice Jacobienne $DF(\Omega^*)$. Ces méthodes permettent de caractériser les attracteurs locaux, même si l'on ne peut pas écrire explicitement une solution fermée pour l'ensemble des pondérations.

De plus, des contraintes de **bornage** (par exemple, imposer que $\omega_{ij} \in [0, \omega_{\max}]$) ou l'utilisation de mécanismes de **saturation** garantissent que l'espace des solutions reste compact, ce qui est indispensable pour appliquer les théorèmes classiques de point fixe (Brouwer, Schauder). Cela permet d'assurer, malgré l'absence de solution analytique globale, que le système admet au moins un attracteur dans l'espace des configurations, même si ce dernier dépend de paramètres externes et de la dynamique non linéaire de $S(i,j)$.

En synthèse, la formalisation d'une fonction énergétique globale dans le DSL se fait aisément dans des cas **simples** où $S(i,j)$ est constant ou linéaire, ce qui conduit à une solution fermée du type

$$\omega_{ij}(t) = \frac{S(i,j)}{\tau} + \left(\omega_{ij}(0) - \frac{S(i,j)}{\tau} \right) \exp(-\eta \tau t).$$

Dans les scénarios plus **complexes** et non linéaires, la dynamique du DSL ne correspond pas à la descente d'un Hamiltonien fixe, mais plutôt à un processus adaptatif évoluant dans un paysage énergétique variable. Les approches pour étudier ces systèmes reposent alors sur des méthodes numériques et sur l'analyse qualitative via la linéarisation locale. Ainsi, bien que la vision « $\mathcal{J}(\Omega) = -\sum_{i,j} \omega_{ij} S(i,j) + \frac{\tau}{2} \sum_{i,j} \omega_{ij}^2 + \dots$ » soit pertinente dans un cadre stationnaire, elle doit être nuancée lorsque la synergie dépend de l'état du système et varie dans le temps.

2.4.3.2. Existence de descentes d'énergie locales : recuit simulé ou itération stochastique

Dans le cadre du **Deep Synergy Learning (DSL)**, l'idée d'interpréter la dynamique des pondérations ω_{ij} comme une descente dans un paysage énergétique a suscité un intérêt particulier, en raison de ses analogies avec les méthodes d'optimisation stochastique telles que le **recuit simulé** (simulated annealing). Nous allons examiner ici la possibilité de formaliser une **descente d'énergie locale** dans le DSL, ainsi que les méthodes par lesquelles un tel processus, éventuellement perturbé par du bruit aléatoire, peut permettre au système d'explorer différents minima énergétiques et ainsi d'échapper à des minima locaux non optimaux.

Dans les modèles de spins classiques, par exemple dans le modèle d'**Ising** ou dans les réseaux de **Hopfield**, l'algorithme de recuit simulé est utilisé pour échapper aux pièges des minima locaux. Le principe est d'introduire une **température** T qui module la probabilité d'accepter une augmentation de l'énergie $\Delta\mathcal{H}$ selon une loi de probabilité de type

$$P(\Delta\mathcal{H}) \propto \exp\left(-\frac{\Delta\mathcal{H}}{T}\right).$$

Lorsque T est élevé, le système explore librement l'espace des configurations, même si certaines transitions font temporairement augmenter l'énergie. Au fil du temps, T diminue progressivement, limitant ainsi les fluctuations et permettant au système de converger vers un minimum global ou, du moins, vers un minimum local satisfaisant.

Pour transposer cette approche au DSL, on suppose qu'il est possible de définir une **pseudo-énergie** $\mathcal{J}(\Omega)$ qui dépend de l'ensemble des pondérations ω_{ij} et qui prend la forme

$$\mathcal{J}(\Omega) = - \sum_{i,j} \omega_{ij} S(i,j) + \frac{\tau}{2} \sum_{i,j} \omega_{ij}^2 + \dots,$$

où $S(i,j)$ représente la **synergie effective** entre les entités \mathcal{E}_i et \mathcal{E}_j . Dans le cas stationnaire et en l'absence de dépendances complexes de S en fonction de Ω , la descente de gradient appliquée à \mathcal{J} conduit à une mise à jour

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \eta \frac{\partial \mathcal{J}}{\partial \omega_{ij}}|_{\omega_{ij}(t)} = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)],$$

ce qui correspond exactement à la règle de mise à jour usuelle du DSL. Ce constat démontre que, dans des cas idéalisés, le DSL effectue effectivement une **descente d'énergie locale** vers le point fixe $\omega_{ij}^* = S(i,j)/\tau$.

Néanmoins, la réalité est souvent plus complexe. Dans de nombreuses applications, la fonction de synergie $S(i,j)$ est sujette à des **non-linéarités** et peut varier en fonction des autres pondérations $\{\omega_{p,q}\}$ ou des états internes des entités. Pour intégrer ce comportement, on peut modifier la règle de mise à jour en y ajoutant un terme de bruit stochastique, de sorte que l'équation prenne la forme

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \eta \frac{\partial \mathcal{J}}{\partial \omega_{ij}}|_{\omega(t)} + \sigma \xi_{ij}(t),$$

où $\xi_{ij}(t)$ représente une perturbation aléatoire et σ en détermine l'amplitude. Ce terme de bruit permet d'imaginer une dynamique d'**itération stochastique** ou de **recuit simulé** qui, en phase initiale, autorise le système à franchir des barrières énergétiques, explorant ainsi différents minima du paysage. Au fil du temps, un **planning de refroidissement** (par exemple, une décroissance exponentielle de σ telle que $\sigma(t) \propto \alpha^t$ avec $0 < \alpha < 1$) est appliqué afin de réduire progressivement l'influence du bruit, permettant alors au système de se stabiliser dans une configuration potentiellement plus globale.

La démarche proposée s'inspire directement des algorithmes de **recuit simulé** utilisés dans les modèles d'optimisation stochastique. En Ising ou dans une machine de Boltzmann, on utilise des critères probabilistes pour accepter ou refuser des changements qui augmentent l'énergie, ce qui

permet de surmonter le problème des minima locaux. Dans le DSL, bien que le paysage énergétique ne soit pas nécessairement fixe – en particulier lorsque la synergie $S(i,j)$ est fonction des pondérations elles-mêmes – l'injection d'un bruit contrôlé offre un levier pour encourager l'exploration et éviter que le réseau ne se bloque prématurément dans une configuration sous-optimale.

Pour illustrer cette approche, il convient d'imaginer un protocole « à la recuit simulé » appliqué aux pondérations. Dans la **phase chaude**, le niveau de bruit $\sigma(0)$ est élevé, ce qui autorise de fortes fluctuations dans ω_{ij} et favorise l'exploration de configurations variées. Puis, lors d'un **refroidissement progressif**, $\sigma(t)$ décroît de manière prédefinie, par exemple suivant une loi exponentielle ou linéaire, et par conséquent, les modifications aléatoires deviennent de plus en plus rares. Finalement, dans la **phase finale** où $\sigma(t) \approx 0$, le système se stabilise dans l'un des minima locaux de la fonction pseudo-énergétique $\mathcal{J}(\Omega)$.

Ce mécanisme est particulièrement pertinent dans les situations où la fonction $S(i,j)$ est suffisamment stationnaire pour que la notion de descente d'énergie ait un sens. Toutefois, dans des scénarios où $S(i,j)$ varie de manière significative en fonction des pondérations ou des états internes, le paysage d'énergie devient mouvant et la descente n'est qu'une **approximation** de la dynamique réelle. Dans ce cas, le recuit simulé ne garantit pas la minimisation d'une fonction énergétique fixe, mais sert plutôt de stratégie pour favoriser l'exploration et pour empêcher le système de se figer dans des minima locaux non désirés.

2.4.3.3. Avantages et inconvénients de cette vision “énergétique” pour expliquer la convergence

L'approche consistant à interpréter la dynamique du DSL à travers une fonction d'**énergie** $\mathcal{J}(\Omega)$ offre une perspective particulièrement intéressante pour analyser la convergence des pondérations ω_{ij} dans un Synergistic Connection Network (DSL). En effet, en posant par exemple

$$\mathcal{J}(\Omega) = - \sum_{i,j} \omega_{ij} S(i,j) + \frac{\tau}{2} \sum_{i,j} \omega_{ij}^2 + \dots,$$

on traduit l'idée que le système cherche à maximiser la synergie entre entités tout en régulant la croissance des pondérations à travers un terme quadratique de **régularisation**. Dans ce cadre, la mise à jour

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \eta \frac{\partial \mathcal{J}}{\partial \omega_{ij}}|_{\omega_{ij}(t)}$$

se simplifie en

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)],$$

ce qui correspond exactement à la règle de mise à jour adoptée dans le DSL dans un cadre stationnaire. Dans cette situation idéale, la dynamique s'apparente à une **descente de gradient** dans un paysage d'énergie fixe, permettant d'expliquer l'**auto-organisation** des pondérations comme une recherche d'un minimum local de \mathcal{J} . Le principal avantage de cette vision est qu'elle fournit un cadre conceptuel puissant et familier aux chercheurs en physique statistique. Par analogie avec

les modèles d'Ising et de Hopfield, où l'énergie $\mathcal{H}(\mathbf{s})$ guide l'évolution des spins, cette approche énergétique permet de justifier l'émergence d'attracteurs et de **clusters** en termes de minimisation d'un potentiel global.

D'un point de vue théorique, cette formalisation simplifie l'analyse des propriétés de convergence, car elle permet de mobiliser des outils issus de l'**optimisation** et de la **théorie des systèmes dynamiques**. En effet, dans le cas où la synergie $S(i, j)$ demeure relativement stationnaire et ne dépend pas trop des pondérations elles-mêmes, le paysage d'énergie défini par $\mathcal{J}(\Omega)$ est stable et la dynamique converge vers un minimum global ou, au moins, vers un attracteur local caractérisé par

$$\omega_{ij}^* = \frac{S(i, j)}{\tau}.$$

Cette interprétation favorise l'emploi de stratégies de **recuit simulé** ou de **gradient stochastique** qui permettent d'explorer l'espace des configurations et d'éviter de se figer prématurément dans des minima locaux sous-optimaux. Le concept de "température" dans ces méthodes, qui module l'amplitude du bruit injecté dans la mise à jour, offre un levier pour contrôler le compromis entre exploration et exploitation du paysage énergétique.

Cependant, cette vision présente également des **limites** importantes lorsqu'elle est appliquée à des scénarios réels du DSL. En effet, dans la majorité des cas, la fonction de synergie $S(i, j)$ n'est pas une valeur fixe mais une fonction **non-linéaire** qui peut dépendre des autres pondérations $\{\omega_{p,q}\}$, des états internes \mathbf{s}_i et même de paramètres externes. Ainsi, le paysage d'énergie $\mathcal{J}(\Omega)$ n'est pas immuable et se reconfigure en continu. Ce phénomène remet en cause l'hypothèse d'un **Hamiltonien stationnaire** et limite l'applicabilité de la descente d'énergie telle que décrite dans les modèles classiques. Dans un tel cadre, même si la mise à jour locale des pondérations peut être interprétée comme une descente de gradient sur une fonction d'énergie approximative, il est difficile de garantir que le système converge vers un minimum global ou même vers un attracteur stable, puisque la fonction $S(i, j)$ elle-même évolue en temps réel.

De plus, contrairement aux modèles d'Ising ou de Hopfield, dans lesquels les couplages (comme J_{ij} ou W_{ij}) sont généralement **fixes** et seuls les états (spins, activations) varient, le DSL fait évoluer les pondérations ω_{ij} de manière adaptative. Cette dynamique, qui s'appuie sur des mécanismes d'auto-organisation complexes, engendre un système souvent **hors équilibre**, dans lequel la notion de descente d'un paysage énergétique fixe est plus difficile à appliquer de manière rigoureuse.

En synthèse, la **vision énergétique** du DSL offre des avantages considérables, notamment en permettant d'interpréter la formation de clusters et la convergence des pondérations comme le résultat d'une **descente de gradient** dans un paysage d'énergie. Ce cadre permet d'employer des techniques d'**optimisation stochastique** et de recuit simulé pour favoriser l'exploration de l'espace de configurations et éviter le piégeage dans des minima locaux peu optimaux. Néanmoins, lorsque la synergie $S(i, j)$ varie significativement en fonction des pondérations et des états internes, le paysage d'énergie devient dynamique, et la correspondance avec un Hamiltonien stationnaire s'efface. Par conséquent, cette approche doit être considérée comme une **approximation** utile dans des scénarios stationnaires ou faiblement dépendants, mais elle perd en pertinence lorsque le système se comporte de manière fortement non linéaire et adaptative. Ainsi, bien que l'analogie avec une fonction d'énergie soit instructive pour expliquer qualitativement la convergence du DSL,

elle présente des limites quant à sa capacité à rendre compte de la totalité de la dynamique du système.

2.4.4. Rôle de l’Inhibition et de la Saturation Synaptique

Dans un SCN (Synergistic Connection Network), la simple règle $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$ peut, dans certaines conditions, conduire à un **développement excessif** de liaisons (pour peu que $S(i,j)$ reste élevée) ou, au contraire, à l’émergence d’oscillations indésirables. Des **mécanismes** supplémentaires, tels que l’**inhibition** ou la **saturation** (limitation) des poids, viennent alors **réguler** la dynamique du réseau. La section (2.4.4) se focalise sur ce **rôle** de l’inhibition et de la saturation, en commençant (2.4.4.1) par une **comparaison** avec les modèles classiques d’**inhibition latérale** chez Amari ou Grossberg.

2.4.4.1. Comparaison avec les mécanismes d’inhibition latérale chez Amari ou Grossberg

Dans la littérature sur les réseaux neuronaux, les travaux d’**Amari** et de **Grossberg** constituent des références majeures pour comprendre comment un réseau peut instaurer une dynamique de **compétition** et de **sélection** via des mécanismes d’**inhibition latérale**. Ces modèles, qui sont généralement formulés en temps continu, mettent en œuvre des termes inhibiteurs permettant de restreindre l’activité collective et de favoriser l’émergence de structures distinctes dans le réseau. Il convient dès lors de comparer ces approches classiques avec celles du **Deep Synergy Learning (DSL)**, dans lequel les pondérations ω_{ij} évoluent de manière adaptative et peuvent intégrer des termes d’inhibition destinés à limiter la croissance excessive des liaisons.

Dans le modèle d’Amari, par exemple, l’évolution de l’activation u_i d’une unité est souvent décrite par une équation différentielle de la forme

$$\frac{du_i}{dt} = -u_i + \sum_j w_{ij} f(u_j) - \beta \sum_{j \neq i} g(u_j),$$

où w_{ij} représente le **couplage excitateur** entre les unités i et j , f est une fonction d’activation non linéaire, et le terme $-\beta \sum_{j \neq i} g(u_j)$ incarne l’**inhibition latérale**. Cette inhibition, par son action négative, réduit l’activation des unités voisines lorsqu’une unité donnée devient fortement active, aboutissant ainsi à un phénomène de **compétition** qui peut, par exemple, engendrer des zones d’activité distinctes dans une carte neuronale.

Grossberg, quant à lui, a développé une approche similaire en mettant en évidence la **compétition** entre neurones par le biais d’une répartition limitée de la ressource synaptique. Dans ses modèles, l’évolution de l’activation est soumise à un mécanisme de régulation tel que

$$\frac{du_i}{dt} = -\alpha u_i + \sum_j W_{ij} f(u_j) - \gamma \phi \left(\sum_j W_{ij} f(u_j) \right),$$

où γ et α sont des constantes positives, et le terme inhibiteur $-\gamma \phi(\cdot)$ permet de moduler l'activation en fonction de la somme des contributions des autres unités. Ce mécanisme, souvent qualifié de **winner-takes-most**, fait en sorte que seules quelques unités parviennent à conserver une activité significative tandis que les autres sont fortement inhibées.

Dans le **DSL**, l'évolution des pondérations ω_{ij} est régie par une mise à jour de type

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)],$$

où η est le **taux d'apprentissage**, τ le **coefficent de décroissance** et $S(i,j)$ la **synergie effective** entre les entités \mathcal{E}_i et \mathcal{E}_j . Pour intégrer un mécanisme d'**inhibition latérale** similaire à celui des modèles d'Amari et de Grossberg, il est possible d'enrichir la mise à jour en ajoutant un terme inhibiteur explicite, par exemple

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \left[S(i,j) - \tau \omega_{ij}(t) - \gamma \sum_{k \neq j} \omega_{ik}(t) \right],$$

où le terme $-\gamma \sum_{k \neq j} \omega_{ik}(t)$ représente une **pénalisation** de la croissance des autres pondérations sortantes du même nœud \mathcal{E}_i . Ce mécanisme incite un nœud à concentrer son potentiel de couplage sur un nombre limité de connexions plutôt que de diffuser de manière uniforme ses ressources synaptiques, imitant ainsi la dynamique d'inhibition observée dans les modèles continus d'Amari et Grossberg.

L'avantage principal de cette approche dans le DSL réside dans sa capacité à **sparsifier** le réseau. En effet, l'introduction de l'inhibition latérale force chaque entité à **sélectionner** les liaisons les plus pertinentes, ce qui conduit à la formation de **clusters** distincts et bien définis. Par analogie avec les réseaux de neurones biologiques, ce mécanisme assure que la compétition entre les connexions maintienne une **structure hiérarchisée** du réseau, dans laquelle seuls les liens les plus forts et les plus significatifs persistent, tandis que les connexions moins importantes sont naturellement atténuées.

La similitude avec les modèles d'Amari et Grossberg se reflète dans la forme des équations différentielles. Dans les deux cas, un **terme de décroissance** – qu'il s'agisse de $-u_i$ dans l'équation d'Amari ou de $-\alpha u_i$ dans celle de Grossberg – joue un rôle fondamental en limitant la croissance des variables et en assurant la **stabilité** de la dynamique. De même, dans le DSL, le terme $-\tau \omega_{ij}(t)$ agit pour contrer l'augmentation induite par la synergie $S(i,j)$. La présence du terme inhibiteur supplémentaire $-\gamma \sum_{k \neq j} \omega_{ik}(t)$ introduit une compétition entre les liaisons sortantes d'un même nœud, ce qui est directement comparable à l'inhibition latérale classique.

En conclusion, la **comparaison** entre les mécanismes d'inhibition latérale d'Amari et de Grossberg et ceux du DSL met en lumière une convergence conceptuelle forte. Alors que les modèles classiques utilisent des termes d'inhibition pour restreindre l'activation globale des unités et favoriser l'émergence de structures distinctes, le DSL exploite des termes analogues dans la mise à jour de ses pondérations pour induire une **compétition** entre les connexions et ainsi favoriser la formation de **clusters** stables et sparsifiés. Cette approche, bien que formulée dans un cadre numérique et évolutif, reprend les principes fondamentaux de la **compétition neuronale** et offre un puissant levier pour la **stabilisation** et la **segmentation** du réseau.

2.4.4.2. Effet sur la stabilisation d'un nombre limité de clusters ou sur la rupture d'oscillations excessives

Dans le cadre du **Deep Synergy Learning (DSL)**, l'introduction d'un mécanisme d'**inhibition compétitive** joue un rôle fondamental en régulant l'évolution des pondérations ω_{ij} . Cette régulation vise à limiter le nombre de liaisons fortes qu'un même nœud peut entretenir, tout en empêchant l'apparition d'oscillations excessives dans la dynamique du réseau. L'objectif est double : d'une part, forcer la **sélection** de connexions significatives qui se regroupent en clusters distincts et, d'autre part, amortir les rétroactions positives qui pourraient conduire à des oscillations auto-entretenues, voire à des régimes pseudo-chaotiques.

Lorsque l'on considère un DSL sans mécanisme inhibiteur explicite, il est fréquent qu'une entité \mathcal{E}_i développe de fortes connexions avec un grand nombre d'autres entités \mathcal{E}_j si la **synergie** $S(i, j)$ reste suffisamment élevée pour chacune de ces paires. Un tel comportement peut conduire à la formation d'un réseau très dense, dans lequel presque toutes les entités sont fortement couplées, rendant floue l'identification de sous-groupes ou de clusters. Afin de remédier à ce phénomène, l'ajout d'un terme d'inhibition compétitive modifie la règle de mise à jour en intégrant une pénalisation sur la somme des pondérations sortantes d'un nœud. Par exemple, la mise à jour peut être modifiée comme suit :

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \left[S(i, j) - \tau \omega_{ij}(t) - \gamma \sum_{k \neq j} \omega_{ik}(t) \right],$$

où $\gamma > 0$ représente le **paramètre d'inhibition**. Ce terme additionnel, $-\gamma \sum_{k \neq j} \omega_{ik}(t)$, exerce une pression négative sur la croissance simultanée de plusieurs liaisons issues d'un même nœud \mathcal{E}_i . Ainsi, lorsqu'une pondération $\omega_{ik}(t)$ croît de manière significative, la somme $\sum_{k \neq j} \omega_{ik}(t)$ augmente, et par conséquent, la progression de $\omega_{ij}(t)$ est freinée. Ce mécanisme force chaque entité à « choisir » un nombre limité de connexions fortes, ce qui conduit à la **formation** de clusters distincts et moins nombreux, facilitant ainsi l'**interprétation** de la structure du réseau.

D'autre part, dans les systèmes non linéaires où des rétroactions peuvent induire des oscillations ou des fluctuations exagérées, l'inhibition compétitive agit également comme un **stabilisateur** dynamique. En imposant une contrainte sur la somme totale des liaisons sortantes, le terme inhibiteur contribue à amortir les variations et à réduire la possibilité d'un renforcement en cascade des pondérations qui, autrement, pourrait entraîner des oscillations de grande amplitude. Du point de vue de l'analyse des systèmes dynamiques, cette inhibition se traduit par l'ajout d'un terme négatif dans la **Jacobienne** du système, diminuant ainsi la norme des valeurs propres et favorisant la convergence vers un attracteur stable. Ainsi, la dynamique globale se retrouve mieux équilibrée et la trajectoire des pondérations $\{\omega_{ij}(t)\}$ se stabilise, limitant les phénomènes de résonance excessive.

En synthèse, l'introduction de l'inhibition dans le DSL présente deux avantages majeurs. Premièrement, elle aboutit à une **sparsification** de la connectivité du réseau en contrignant chaque entité à ne maintenir que quelques liaisons fortes, ce qui rend la formation des clusters plus claire et la segmentation du réseau plus nette. Deuxièmement, elle contribue à **amortir** les oscillations

potentielles, en introduisant un mécanisme de retour négatif qui empêche l'exacerbation des rétroactions positives, favorisant ainsi une **stabilité dynamique** accrue. Cette approche, qui s'inspire directement des modèles d'inhibition latérale chez Amari et Grossberg, permet d'obtenir un réseau auto-organisé présentant à la fois une structure hiérarchisée et une robustesse face aux fluctuations internes.

2.4.4.3. Discussion : analogies biologiques, applications en robotique sensorielle

L'approche du **Deep Synergy Learning (DSL)**, par l'intermédiaire de ses mécanismes d'inhibition et de saturation, trouve de fortes correspondances avec les observations faites dans la **biologie neuronale**. Dans le cortex sensoriel, par exemple, il est largement démontré que l'**inhibition latérale** joue un rôle déterminant dans la formation de **cartes corticales**. Les neurones fortement activés exercent une inhibition sur leurs voisins, ce qui permet d'isoler des zones d'activité et de créer des **patches** ou des **colonnes d'orientation** dans le cortex visuel. Ce phénomène est essentiel pour la **compétition neuronale** et la spécialisation fonctionnelle, des principes qui ont été théorisés par des auteurs tels qu'**Amari** et **Grossberg**. Ces travaux démontrent que la dynamique d'un réseau de neurones peut être modulée par des mécanismes inhibiteurs qui limitent la propagation de l'excitation et favorisent la sélection d'une **activité dominante** au sein de sous-groupes distincts.

Dans le DSL, la **mise à jour** des pondérations ω_{ij} se fait selon une règle qui, lorsqu'enrichie d'un terme inhibiteur tel que

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \left[S(i,j) - \tau \omega_{ij}(t) - \gamma \sum_{k \neq j} \omega_{ik}(t) \right],$$

permet de limiter la croissance simultanée de nombreuses connexions issues d'un même nœud \mathcal{E}_i . Ce terme inhibiteur $-\gamma \sum_{k \neq j} \omega_{ik}(t)$ reproduit en quelque sorte le principe de la **ressource synaptique limitée** qui est observé dans les systèmes biologiques : lorsqu'une synapse se renforce, les autres connexions du même neurone sont freinées, conduisant à une **sélection** des liens les plus pertinents. Ainsi, cette stratégie favorise l'émergence d'**assemblées neuronales** ou de **clusters** distincts, où seules quelques connexions dominantes persistent, tout en inhibant les connexions superflues.

Du point de vue des **applications en robotique sensorielle**, l'intérêt de ces mécanismes se manifeste de manière très concrète. Dans un système robotique intégrant de multiples capteurs – tels qu'une caméra, un sonar, un LiDAR ou encore des micros – chaque capteur peut être modélisé comme une entité \mathcal{E}_i au sein d'un SCN. La **synergie** $S(i,j)$ mesure alors la complémentarité ou la redondance entre deux capteurs. Sans un mécanisme d'inhibition, il est envisageable que le réseau attribue une importance excessive à plusieurs canaux simultanément, ce qui pourrait mener à une **fusion de données** confuse et à une surcharge informationnelle. L'inhibition permet, en revanche, de contraindre le système à privilégier un nombre restreint de connexions robustes – par exemple, en mettant en avant la combinaison d'une vision frontale et d'un microphone ambiant – tout en atténuant l'impact des capteurs moins pertinents à un moment donné. En outre, l'inhibition agit comme un **verrou** contre l'auto-renforcement excessif qui pourrait amplifier indéfiniment le bruit dans certains canaux, garantissant ainsi une **robustesse** accrue du traitement sensoriel.

Cette **discussion** met en lumière l'importance de l'inhibition dans le DSL, tant sur le plan théorique que dans ses **applications pratiques**. L'inhibition, par son effet de **compétition** entre les connexions, permet de réduire la densité des liaisons et de favoriser une **structuration** en clusters, ce qui facilite la segmentation et la hiérarchisation de l'information. Ce mécanisme, qui s'inspire directement des observations biologiques – notamment celles concernant l'inhibition latérale dans le cortex – trouve également une application essentielle dans la robotique sensorielle, où la gestion efficace des flux de données est cruciale pour une perception fiable et une prise de décision autonome.

2.4.5. Approches Hybrides et Collaboration Interdisciplinaire

Tout au long de la section 2.4, nous avons souligné les **connexions** entre le DSL (Deep Synergy Learning) et divers champs où la théorie des systèmes dynamiques (2.4.1), les modèles de spin (2.4.2) et la notion d'énergie potentielle (2.4.3), ou encore les mécanismes d'inhibition/saturation (2.4.4) jouent un rôle central. Ces croisements laissent entrevoir de **nombreuses pistes** d'approches dites “hybrides” ou inter-filières, mobilisant la **physique**, la **biologie**, l'**informatique** et l'**ingénierie**. La section 2.4.5 (Approches Hybrides et Collaboration Interdisciplinaire) discute d'abord (2.4.5.1) du **recuit simulé** et d'autres algorithmes inspirés par la **physique statistique** pour régler le SCN, avant de poser (2.4.5.2) des perspectives dans des champs variés comme l'écologie numérique ou les réseaux sociaux, puis de faire (2.4.5.3) une transition vers les chapitres 3, 4, 5 où les mises en forme plus “ingénierie” seront développées en détail.

2.4.5.1. Comment la physique statistique peut fournir des algorithmes inspirés du recuit (simulated annealing) pour régler le SCN

L'approche du **recuit simulé** en physique statistique repose sur une analogie avec la trempe des métaux, dans laquelle un matériau est chauffé à une température élevée afin de permettre aux atomes de se réarranger librement, puis refroidi lentement pour atteindre une configuration d'énergie minimale. Dans le contexte algorithmique, chaque configuration d'un système est associée à une fonction « énergie » $\mathcal{H}(\mathbf{x})$ et l'objectif est de minimiser cette énergie en effectuant des transitions aléatoires. La probabilité d'accepter une modification qui augmente l'énergie est donnée par la loi exponentielle

$$P(\Delta\mathcal{H}) \propto \exp\left(-\frac{\Delta\mathcal{H}}{T}\right),$$

où T représente la **température**. Lorsque T est élevé, le système explore de manière large l'espace des configurations, et à mesure que T diminue, les changements qui augmentent \mathcal{H} deviennent de moins en moins probables, permettant ainsi au système de se stabiliser dans un minimum d'énergie.

Dans le **Deep Synergy Learning (DSL)**, il est naturel de chercher à transposer ce principe au réglage des pondérations ω_{ij} du **Synergistic Connection Network (SCN)**. Pour ce faire, on définit une **pseudo-énergie** $\mathcal{J}(\Omega)$ fonctionnelle des pondérations, dont une expression heuristique pourrait être donnée par

$$\mathcal{J}(\Omega) = - \sum_{i,j} \omega_{ij} S(i,j) + \frac{\tau}{2} \sum_{i,j} \omega_{ij}^2 + \dots,$$

où $S(i,j)$ désigne la **synergie effective** entre les entités \mathcal{E}_i et \mathcal{E}_j , et τ est un **coefficent de décroissance** qui sert de régularisateur pour éviter une croissance non bornée des poids. Dans un cadre idéal, si l'on considère que $S(i,j)$ est constant ou ne dépend pas fortement de ω_{ij} , la dynamique des pondérations peut être vue comme une **descente de gradient** de \mathcal{J} :

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \eta \frac{\partial \mathcal{J}}{\partial \omega_{ij}}|_{\omega_{ij}(t)},$$

ce qui, en posant

$$\frac{\partial \mathcal{J}}{\partial \omega_{ij}} = -S(i,j) + \tau \omega_{ij},$$

se traduit par

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)].$$

Ce processus, qui dans ce cas particulier correspond à une descente d'énergie locale, permet d'expliquer la formation des clusters en considérant que le système tend à minimiser la fonction \mathcal{J} . L'injection de **bruit** dans ce mécanisme – par exemple en modifiant la mise à jour comme suit

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \eta \frac{\partial \mathcal{J}}{\partial \omega_{ij}}|_{\omega_{ij}(t)} + \sigma \xi_{ij}(t),$$

où $\xi_{ij}(t)$ est une variable aléatoire et σ en contrôle l'amplitude – permet alors d'implémenter une procédure de **recuit simulé**. Dans cette approche, une température effective $T(t)$ (ou une amplitude de bruit $\sigma(t)$) est introduite et décroît progressivement. Au départ, le système, exposé à un niveau de bruit élevé, peut explorer divers minima locaux en acceptant occasionnellement des mouvements qui augmentent la pseudo-énergie \mathcal{J} . Au fur et à mesure que $T(t)$ décroît, ces mouvements deviennent moins probables et le système se stabilise dans une configuration optimisée.

Il convient de noter que, dans des scénarios plus complexes, où la synergie $S(i,j)$ dépend de manière non linéaire des pondérations ω_{ij} ou de paramètres externes, le paysage d'énergie $\mathcal{J}(\Omega)$ ne sera pas fixe mais évoluera dans le temps. Dans ce cas, l'application rigoureuse d'un algorithme de recuit simulé devient plus délicate, et la descente d'énergie n'est qu'une **approximation** de la dynamique réelle du système. Cependant, même dans ces conditions, l'ajout contrôlé de bruit peut aider le SCN à **échapper** à des minima superficiels et à explorer un espace de solutions plus large.

2.4.5.2. Perspectives d'usage en écologie numérique, réseaux sociaux évolutifs, etc.

La démarche du **Deep Synergy Learning (DSL)**, qui s'appuie sur des concepts issus de la physique statistique tels que le recuit simulé, l'approche énergétique et les mécanismes d'inhibition, peut être étendue bien au-delà du domaine des réseaux neuronaux classiques ou de l'optimisation

algorithme. En effet, ces techniques trouvent également un écho dans des domaines où l'on traite de **réseaux dynamiques** et de systèmes multi-entités, notamment dans l'**écologie numérique** et les **réseaux sociaux évolutifs**. Cette section présente une analyse détaillée de la manière dont la vision du DSL peut contribuer à l'**organisation**, à l'**adaptation** et à la **convergence** de systèmes complexes, soumis à d'importantes évolutions structurelles et temporelles.

Dans le cadre de l'**écologie numérique**, on considère un ensemble d'entités logicielles – par exemple, des microservices, des agents autonomes ou des protocoles distribués – qui interagissent de manière continue pour former un écosystème virtuel. Chaque entité, notée \mathcal{E}_i , est associée à des caractéristiques spécifiques et à une dynamique de mise à jour de ses liens, représentés par des pondérations ω_{ij} . La **synergie** $S(i, j)$ entre deux entités évalue la valeur ajoutée de leur interaction, qu'il s'agisse d'un gain en performance, en complémentarité fonctionnelle ou en compatibilité technique. Dans ce contexte, le DSL permet de modéliser les interactions comme une forme d'**auto-organisation** où, par des mises à jour locales inspirées de mécanismes de recuit simulé, seules les connexions pertinentes se renforcent tandis que les autres s'affaiblissent progressivement. Par exemple, si l'on définit un opérateur de mise à jour du type

$$\omega_{ij}(t + 1) = \omega_{ij}(t) + \eta [S(i, j) - \tau \omega_{ij}(t)],$$

la dynamique résulte en la formation de sous-groupes d'entités fortement couplées, ce qui se traduit par l'émergence de **communautés** de microservices. La présence de mécanismes d'inhibition, qui limitent la somme des pondérations sortantes d'un même nœud, assure une **sparsification** du réseau et empêche la sur-connexion généralisée. Ainsi, dans une plateforme distribuée, la structure auto-organisée obtenue facilite la coordination entre services, permettant une répartition efficace des tâches et une meilleure gestion des ressources, sans nécessiter de contrôleur centralisé.

D'un autre côté, dans le domaine des **réseaux sociaux évolutifs**, les utilisateurs – modélisés comme des entités \mathcal{E}_i – interagissent par le biais de liens dont l'intensité, représentée par ω_{ij} , reflète la fréquence ou la qualité des interactions (comme la communication, le partage de contenu ou l'affinité d'opinion). La **synergie** $S(i, j)$ peut être dérivée d'indices tels que la similarité des centres d'intérêt, le nombre de messages échangés ou encore la co-occurrence d'activités. En intégrant des mécanismes d'inhibition dans la mise à jour des liens, le DSL permet de modéliser la **sélectivité relationnelle** qui caractérise les réseaux sociaux où un utilisateur ne peut entretenir simultanément des connexions fortes avec un très grand nombre d'autres, ce qui conduit à la formation de **communautés** stables. La règle de mise à jour

$$\omega_{ij}(t + 1) = \omega_{ij}(t) + \eta \left[S(i, j) - \tau \omega_{ij}(t) - \gamma \sum_{k \neq j} \omega_{ik}(t) \right]$$

illustre comment le terme inhibiteur $-\gamma \sum_{k \neq j} \omega_{ik}(t)$ force l'utilisateur à concentrer ses interactions sur un sous-ensemble d'autres utilisateurs, reproduisant ainsi l'effet de **compétition synaptique** observé dans le cerveau. Ce mécanisme aide à prévenir la dispersion des ressources relationnelles et contribue à la formation de clusters sociaux, où les interactions internes sont beaucoup plus intenses que celles avec le reste du réseau.

Les **perspectives d'usage** du DSL dans ces domaines sont multiples. Dans l'**écologie numérique**, par exemple, l'auto-organisation des microservices peut permettre de construire des écosystèmes

résilients où l'interopérabilité est constamment optimisée par des mises à jour locales. Chaque service ajuste ses connexions en fonction de la synergie mesurée avec d'autres services, et l'ensemble du système évolue vers une configuration optimale qui minimise les coûts de communication et maximise la performance globale. De même, dans les **réseaux sociaux évolutifs**, la dynamique du DSL peut aider à prédire et à comprendre l'évolution des communautés. La capacité du système à s'adapter à l'arrivée ou au départ d'utilisateurs, combinée aux mécanismes de recuit simulé et d'inhibition, permet d'obtenir une modélisation robuste de la formation, de la fusion ou de la fragmentation des groupes sociaux, tout en tenant compte des fluctuations inhérentes aux interactions humaines.

Enfin, l'approche DSL, grâce à sa **flexibilité** dans la définition de la fonction de synergie $S(i, j)$ et de la mise à jour des pondérations, offre un cadre général capable de s'adapter à de nombreux autres domaines. Que ce soit pour modéliser des réseaux d'innovation, des collaborations scientifiques ou des systèmes logistiques distribués, l'analogie avec les techniques d'**optimisation stochastique** et de recuit simulé reste pertinente. Les méthodes DSL permettent alors de gérer des **interactions** complexes et de produire des configurations stables dans un environnement évolutif, favorisant ainsi l'émergence de structures cohérentes à partir d'un ensemble initialement désordonné.

2.4.5.3. Transition vers les Chapitres 3, 4, 5, où l'on verra des mises en forme plus “ingénierie” du SCN

Dans les sections précédentes, nous avons établi un cadre théorique robuste pour le **Deep Synergy Learning (DSL)** en nous appuyant sur des analogies avec la **physique statistique**, les **modèles de spin** (tels que Ising, Potts ou Hopfield) ainsi que sur des concepts issus de la **théorie des systèmes dynamiques** et des **mécanismes d'inhibition** observés en neurosciences. La formulation du DSL repose sur une dynamique de pondérations $\{\omega_{ij}\}$ évoluant suivant une règle de mise à jour qui, dans sa version continue, s'exprime par l'équation différentielle

$$\frac{d \omega_{ij}}{dt} = \eta [S(i, j; \{\omega_{p,q}\}, \dots) - \tau \omega_{ij}],$$

où $\eta > 0$ désigne le **taux d'apprentissage** et $\tau > 0$ le **coefficent de décroissance**. Les développements théoriques des sections 2.4.1 à 2.4.5 mettent en exergue des idées telles que la **descendre d'énergie** dans un paysage complexe, l'**inhibition compétitive** rappelant les mécanismes d'Amari et de Grossberg, ainsi que l'utilisation de techniques de **recuit simulé** pour éviter le piégeage dans des minima locaux. Ces concepts, tout en étant profondément ancrés dans des modèles théoriques multidisciplinaires, ouvrent la voie à une transposition vers une **implémentation opérationnelle**.

Il convient désormais de passer de cette **maquette conceptuelle** à une **ingénierie concrète**. C'est exactement l'objectif des Chapitres 3, 4 et 5, qui se proposent d'apporter une déclinaison pratique et opérationnelle de l'approche DSL. Plus précisément, le **Chapitre 3** détaillera la structuration du SCN en exposant les **structures de données** nécessaires pour gérer les matrices de pondérations $\{\omega_{ij}\}$ et les **algorithmes** de mise à jour, qu'ils soient de type **batch**, **en ligne** ou **stochastique**. On y développera notamment des stratégies pour la gestion de seuils (notion de ω_{\min}) et

l'implémentation de mécanismes d'inhibition, en s'appuyant sur des formulations mathématiques précises.

Le **Chapitre 4** se concentrera quant à lui sur les aspects **logiciels et parallélisme computationnel**. Il exposera les environnements de développement adaptés – qu'il s'agisse de langages comme Python pour la flexibilité ou de C++/CUDA pour la performance – et décrira comment répartir le calcul sur plusieurs cœurs ou sur des architectures GPU. La modularisation des fonctions, par exemple la séparation entre le module de calcul de la synergie $S(i, j)$ et le module de mise à jour des pondérations, sera abordée en détail, permettant ainsi d'envisager une implémentation robuste et scalable du SCN.

Enfin, le **Chapitre 5** portera sur la **validation** et l'**évaluation** des performances du SCN dans des scénarios réels. Il s'agira de définir des **critères de convergence** (par exemple, lorsque $\|\omega_{ij}(t + 1) - \omega_{ij}(t)\|$ devient négligeable), d'évaluer la **cohésion** des clusters formés (via des indicateurs comme la modularité ou le ratio interne/externe) et de mener des expériences sur des jeux de données multimodaux ou issus de flux sensoriels. Ces tests permettront de vérifier la robustesse, la **résilience** face au bruit et la **scalabilité** des approches développées.

2.5. Perspectives Historiques et Liens avec l'IA Moderne

Au terme de ce **Chapitre 2**, nous avons exploré les **fondements théoriques** du DSL (Deep Synergy Learning) où les liens avec la théorie des systèmes dynamiques, les rapprochements avec la physique statistique, ainsi que les mécanismes d'inhibition et de saturation ont été analysés. Il est intéressant de **resituer** tout cela dans un **contexte historique** et de voir comment le DSL peut **converger** ou s'articuler avec les **approches** actuelles de l'IA (apprentissage profond, IA neuro-symbolique, robotique adaptative, etc.). La section (2.5) propose un **regard** à la fois **rétrospectif** (comment ces idées se sont forgées) et **prospectif** (comment elles se connectent à l'IA moderne), préparant ainsi la transition vers les chapitres plus “implémentation” et “application”.

2.5.1. Évolutions Récentes et Convergence avec le Deep Learning

Depuis la vague d'**apprentissage profond** (Deep Learning) qui domine l'IA contemporaine (vision, langage, RL, etc.), l'idée de laisser des **couches** apprendre automatiquement des **features** est devenu un paradigme central. Dans cette optique, on peut se demander si un **SCN** (Synergistic Connection Network) ne pourrait pas venir **compléter** ou **fusionner** avec des réseaux neuronaux profonds, de façon à gérer l'**auto-organisation** à un niveau plus abstrait.

2.5.1.1. Essor de l'apprentissage profond : extraire des features et laisser un SCN auto-organiser ces caractéristiques à un niveau plus abstrait ?

Dans le contexte actuel de l'**apprentissage profond**, il est devenu courant d'utiliser des réseaux neuronaux tels que les **CNN**, **RNN** ou **Transformers** afin d'extraire automatiquement des **représentations** riches à partir de données brutes telles que des images, des signaux sonores ou des textes. Ces réseaux, souvent désignés comme des **backbones**, produisent des embeddings de haute dimension, par exemple un vecteur $\mathbf{x}_i \in \mathbb{R}^{512}$ pour chaque donnée d'entrée. L'idée qui se dégage est de ne pas se contenter d'envoyer ces features directement dans un classifieur supervisé, mais d'exploiter l'**auto-organisation** d'un **Synergistic Connection Network (SCN)** pour traiter ces caractéristiques à un niveau d'abstraction supérieur.

Dans ce cadre, chaque entité \mathcal{E}_i issue du backbone représente un ensemble de features qui sont ensuite transmises au SCN. La **fonction de synergie** $S(i, j)$ quantifie la relation ou la **complémentarité** entre les représentations de deux entités \mathcal{E}_i et \mathcal{E}_j . À partir de là, les pondérations ω_{ij} évoluent selon une dynamique d'**auto-organisation** décrite par une règle de mise à jour (voir par exemple l'équation en temps discret) :

$$\omega_{ij}(t + 1) = \omega_{ij}(t) + \eta [S(i, j) - \tau \omega_{ij}(t)],$$

où $\eta > 0$ représente le **taux d'apprentissage** et $\tau > 0$ le **coefficient de décroissance**. Ce processus permet au SCN d'explorer l'espace latent des features et de faire émerger des **clusters synergiques** qui ne seraient pas immédiatement décelables via une simple classification. En d'autres termes, plutôt que de procéder à une classification rigide, le SCN se charge de **combiner** et **d'organiser**

ces embeddings en regroupements cohérents qui révèlent des structures plus abstraites et potentiellement pertinentes pour l'application considérée.

Un schéma typique de pipeline illustrant ce processus s'articule ainsi :

$$\begin{aligned} \text{Backbone (ResNet, BERT, etc.)} &\rightarrow \text{Feature Embedding} \\ \rightarrow \text{SCN (pondérations } \omega_{ij} \text{ évolutives).} \end{aligned}$$

Dans ce pipeline, le module de **feature extraction** produit des représentations vectorielles de qualité, qui sont ensuite traitées par le SCN. Ce dernier module peut fonctionner de manière **complètement non supervisée**, en laissant la dynamique locale – régulée par la synergie $S(i, j)$ et éventuellement enrichie par des mécanismes d'inhibition et de recuit simulé – conduire à la formation de clusters. Alternativement, il peut coexister avec des modules supervisés qui valident la structure émergente des regroupements, offrant ainsi une approche hybride permettant de bénéficier à la fois de la puissance de l'**apprentissage profond** et de la capacité d'**auto-organisation** du SCN.

L'avantage principal de cette approche réside dans sa **flexibilité**. En effet, la capacité du SCN à se réorganiser dynamiquement permet d'extraire des **patterns** inattendus ou des regroupements latents, qui peuvent correspondre à des catégories ou des classes qui ne sont pas explicites dans les données d'origine. Par exemple, dans le traitement d'un flot d'images par un ResNet, le SCN pourrait découvrir des **proto-classes** ou des regroupements fondés sur des similarités spatiales et contextuelles que l'on ne retrouverait pas dans un classifieur standard. Cette capacité à explorer et à structurer l'espace latent des features offre une plus-value significative en termes de **représentation** et de **compréhension** des données, ouvrant la voie à des applications avancées en vision par ordinateur, en traitement du langage naturel et en robotique.

Historiquement, certains systèmes expérimentaux ont déjà exploité des **modules compétitifs** et des **couches hebbiennes** pour favoriser la spécialisation des neurones dans des architectures précurseurs du deep learning moderne. Le DSL reprend et étend ces idées en introduisant une dimension d'**auto-organisation** qui opère non seulement sur les activations, mais directement sur les **liaisons** entre entités. Ainsi, au lieu de modifier uniquement les sorties des neurones, le DSL permet aux poids ω_{ij} de s'ajuster pour favoriser des interactions de plus haut niveau, rendant possible l'émergence d'une **structure hiérarchisée** dans l'espace des features.

2.5.1.2. Exemples précurseurs : modules compétitifs, couches “Hebbiennes” dans certaines architectures de recherche

Dans l'histoire de l'intelligence artificielle, bien avant l'avènement généralisé du **Deep Learning** tel qu'on le connaît aujourd'hui et avant la formulation explicite du **Deep Synergy Learning (DSL)**, de nombreux chercheurs ont exploré des mécanismes d'apprentissage non supervisé fondés sur la compétition et l'auto-organisation. Ces travaux précurseurs, qui incluent les **modules compétitifs** et les **couches Hebbiennes**, ont permis de poser les jalons d'une approche qui, dans sa version moderne, se retrouve incarnée dans le DSL. On peut ainsi considérer ces premières architectures comme des prototypes conceptuels ayant anticipé la logique d'un **Synergistic Connection Network (SCN)**.

Dans les années 1980 et 1990, le paradigme du **Competitive Learning** fut largement étudié. L'idée fondamentale était d'amener un ensemble de neurones, ou de prototypes, à « compétition » pour représenter des caractéristiques d'une donnée d'entrée. Les algorithmes tels que le **Learning Vector Quantization (LVQ)** reposaient sur le principe que, pour une donnée d'entrée \mathbf{x} , le prototype le plus proche (selon une métrique souvent euclidienne) recevait un renforcement de son vecteur de poids. Mathématiquement, la règle de mise à jour pour le prototype gagnant, disons \mathbf{w}_i , pouvait s'exprimer de manière simplifiée par

$$\Delta \mathbf{w}_i = \eta (\mathbf{x} - \mathbf{w}_i),$$

ce qui tend à aligner \mathbf{w}_i sur les vecteurs d'entrée similaires. Ce mécanisme, par son **aspect compétitif**, impose une différenciation entre les prototypes et favorise la formation de **clusters** d'entrées. La compétition ainsi instaurée conduit naturellement à une **auto-organisation** dans l'espace de représentation, préfigurant la dynamique d'un SCN où les pondérations entre entités s'ajustent en fonction d'une **synergie** mesurée.

Parallèlement, les **Self-Organizing Maps (SOM)** de Kohonen, introduites dans les années 1980, ont apporté une dimension topologique à cette idée. Dans un SOM, chaque neurone de la carte possède un vecteur de poids et la mise à jour se fait de manière à non seulement adapter le neurone gagnant à l'entrée, mais aussi à ajuster ses voisins dans le voisinage défini par un **noyau d'inhibition**. La règle de mise à jour pour un neurone voisin peut être formulée par

$$\Delta \mathbf{w}_j = \eta h(i, j) (\mathbf{x} - \mathbf{w}_j),$$

où $h(i, j)$ représente une fonction décroissante en fonction de la distance topologique entre le neurone gagnant i et le neurone j . Ce mécanisme permet de préserver la **cohérence spatiale** dans la carte, et il a servi de modèle pour comprendre comment un réseau peut se partitionner en régions distinctes d'activité, concept qui trouve une résonance dans la formation de clusters dans le DSL.

Le **postulat Hebbien** de Donald Hebb, formulé en 1949, constitue également une pierre angulaire dans cette évolution conceptuelle. Le célèbre adage « **Neurons that fire together, wire together** » a inspiré les premières règles d'apprentissage synaptique. La règle classique de Hebb, qui peut être écrite comme

$$\Delta w_{ij} \propto x_i x_j,$$

implique que si deux neurones sont activés simultanément, le lien synaptique entre eux se renforce. Bien que cette règle ne comporte pas de mécanisme intrinsèque de régulation (ce qui peut conduire à une croissance indéfinie des poids), des modifications ultérieures, telles que l'ajout d'un terme de normalisation ou d'inhibition, ont permis de stabiliser ces dynamiques. Ces ajustements ont donné naissance à des **couches Hebbiennes** expérimentales dans certaines architectures de recherche, où l'idée de renforcer les connexions en fonction de la co-activation était exploitée pour générer des représentations auto-organisées des données.

Des approches connexes, telles que l'**Adaptive Resonance Theory (ART)** développée par Stephen Grossberg et Gail Carpenter, combinent des mécanismes de compétition, de vigilance et de reset pour permettre à un réseau de classifier de manière stable tout en intégrant de nouvelles informations sans oublier les anciennes. Les modèles ART illustrent comment la dynamique

compétitive et les processus de **stabilisation** peuvent coexister dans des systèmes non supervisés, ce qui préfigure la logique d'un SCN dans le cadre du DSL.

En résumé, les **modules compétitifs** et les **couches Hebbiennes** ont constitué des **précurseurs** essentiels dans l'évolution de l'apprentissage non supervisé. Ces systèmes ont démontré que la **compétition** entre neurones – conduisant à la formation de clusters d'activité – ainsi que l'**apprentissage Hebbien** pouvaient engendrer une auto-organisation efficace des connexions. Le DSL s'inscrit dans cette lignée en généralisation, en permettant aux pondérations ω_{ij} d'évoluer selon une **fonction de synergie** $S(i,j)$ qui peut intégrer diverses mesures (distance, co-information, etc.) et en incluant des mécanismes de régulation tels que la **décroissance** (via un terme $\tau \omega_{ij}$) ou l'**inhibition compétitive**. Ainsi, l'approche DSL synthétise et étend les idées anciennes en les adaptant aux défis modernes de la **multimodalité** et de l'**apprentissage continu**, grâce notamment aux avancées récentes en calcul (GPU, big data) qui permettent de gérer des réseaux de grande dimension.

2.5.2. Extensions Symboliques et Neuro-Symboliques

Dans la lignée des **perspectives historiques** (2.5.1), on constate que le **DSL** (Deep Synergy Learning) ne se limite pas à des entités purement numériques ou strictement sub-symboliques où, grâce à son cadre théorique très **général**, il peut **intégrer** à la fois des représentations **logiques** (symboliques) et des composantes **sub-symboliques** (vecteurs, embeddings, etc.). La section (2.5.2) aborde ces **extensions symboliques et neuro-symboliques**, montrant que le DSL offre un terrain d'entente entre l'IA symbolique (règles, concepts, raisonnements) et l'IA sub-symbolique (réseaux neuronaux, traitement statistique). Nous commençons (2.5.2.1) par rappeler que le DSL peut accueillir des entités logiques ou sub-symboliques en toute flexibilité.

2.5.2.1. Rappel : DSL peut accueillir des entités logiques ou des blocs sub-symboliques

Dans le cadre du **Deep Synergy Learning (DSL)**, la conception des « **entités** » n'est pas nécessairement limitée à des vecteurs continus appartenant à un espace \mathbb{R}^d . En effet, l'architecture d'un **Synergistic Connection Network (SCN)** se caractérise par sa grande **flexibilité** quant à la nature des entités qui le composent. Dès lors, il est possible d'intégrer dans un même réseau à la fois des **entités sub-symboliques** et des **entités logiques** ou des blocs **symboliques**, ce qui permet de traiter simultanément des représentations continues et des représentations discrètes.

Dans la perspective **sub-symbolique**, les entités \mathcal{E}_i sont généralement représentées par des **embeddings**, c'est-à-dire des vecteurs caractéristiques issus de réseaux profonds (tels que CNN, RNN, ou Transformers). Ainsi, une entité est formellement décrite par un vecteur

$$\mathbf{x}_i \in \mathbb{R}^d,$$

et la **synergie** entre deux entités, notée $S(i,j)$, est calculée à l'aide de mesures classiques telles que la **distance euclidienne** ou la **similarité cosinus**. Par exemple, une fonction de synergie peut être définie comme

$$S(i,j) = \exp(-\alpha \parallel \mathbf{x}_i - \mathbf{x}_j \parallel),$$

où $\alpha > 0$ est un paramètre de sensibilité. Cette formulation permet d'extraire des caractéristiques robustes issues de données brutes (images, sons, textes, etc.) et d'auto-organiser ces représentations en **clusters** par la suite.

D'un autre côté, le DSL permet également d'accueillir des **entités logiques** ou des **blocs symboliques**. Dans ce cas, une entité \mathcal{E}_i peut être définie non pas par un simple vecteur, mais par un objet conceptuel ou un ensemble de règles. Par exemple, on peut envisager que \mathcal{E}_i représente un **concept** ou une **proposition** formelle, telle que

$$\mathcal{E}_i = \text{"Si } A \text{ alors } B\text{"}.$$

La fonction de synergie $S(i, j)$ dans ce contexte sera alors adaptée pour quantifier la **compatibilité** ou la **cohérence** entre deux ensembles de règles ou entre des concepts issus d'ontologies différentes. Une approche possible consiste à définir une distance sémantique ou une mesure de **similarité logique** basée sur la structure de l'ontologie. Par exemple, on pourrait utiliser une mesure de similarité définie par

$$S(i, j) = \sigma(\mathcal{E}_i, \mathcal{E}_j) \in [0, 1],$$

où σ est construite à partir d'une métrique de correspondance sémantique.

L'**intégration** de ces deux types d'entités dans un même SCN permet de bâtir un modèle **neuro-symbolique** puissant, capable de combiner les avantages du traitement continu (apprentissage à partir de données brutes, robustesse aux variations) et ceux du raisonnement symbolique (interprétabilité, manipulation explicite de concepts et règles). L'une des **forces** du DSL réside dans sa capacité à utiliser une **fonction de synergie** $S(i, j)$ qui peut être spécifiquement adaptée en fonction de la nature des entités concernées. Ainsi, pour des entités sub-symboliques, on utilisera des mesures issues de la **géométrie de l'espace vectoriel** (distance euclidienne, similarité cosinus, etc.), tandis que pour des entités logiques, des techniques de **matching sémantique** ou des calculs de **coïncidence symbolique** seront privilégiées.

Cette approche hybride permet d'aboutir à une **auto-organisation** qui se fonde sur une **complémentarité** entre des représentations denses et continues et des représentations structurées et discrètes. Les pondérations ω_{ij} qui relient les entités dans le SCN s'ajustent ainsi en fonction d'une **synergie** multi-dimensionnelle, où chaque lien reflète à la fois une similitude perceptuelle et une compatibilité conceptuelle. L'ensemble du réseau, noté

$$\Omega = \{\omega_{ij}\},$$

se structure de manière à mettre en avant les connexions les plus **pertinentes** et à minimiser celles qui sont moins significatives, quelle que soit la nature des entités.

2.5.2.2. Synergies entre IA symbolique, sous-ensembles de règles, et la plasticité d'un SCN

Dans cette section, nous explorons comment l'**IA symbolique** – qui s'appuie sur la manipulation de règles, de concepts et d'ontologies – peut interagir de manière fructueuse avec la **plasticité** inhérente à un **Synergistic Connection Network (SCN)**, tel que présenté dans le cadre du **DSL** (Deep Synergy Learning). Cette synergie repose sur la capacité du SCN à adapter dynamiquement

les pondérations ω_{ij} en fonction d'une fonction de synergie $S(i,j)$ adaptée aux différentes natures d'entités. Nous nous référons ici aux idées développées dans les sections précédentes, notamment aux notions d'auto-organisations évoquées en (2.5.2.1) et aux fondements théoriques du DSL présentés dans les sections 2.3 et 2.4.

A. IA symbolique et “blocs de règles” : rappel et enjeux

Historiquement, l'**IA symbolique** s'est concentrée sur la manipulation de symboles, de règles formelles et de représentations logiques. Dans ce paradigme, une entité symbolique peut être un **bloc de règles** ou un **concept** défini par une structure logique, par exemple une proposition du type

$$\text{if } A \text{ then } B.$$

Ces entités disposent d'une **force interne** basée sur la validité ou la cohérence de leurs règles et, lorsqu'elles interagissent avec d'autres blocs similaires, une mesure de **compatibilité** peut être définie. Par exemple, si deux ensembles de règles \mathcal{R}_1 et \mathcal{R}_2 se complètent, leur synergie peut être quantifiée par une fonction de similarité, que nous noterons $S_{\text{sym}}(i,j)$. Ce mécanisme est similaire aux systèmes d'inférence dans l'IA symbolique classique, où un moteur de règles applique des opérations logiques pour en déduire de nouvelles connaissances.

B. Plasticité d'un SCN : une approche dynamique de l'auto-organisation

Dans le cadre du DSL, les **entités** ne sont pas exclusivement sub-symboliques (comme les vecteurs $\mathbf{x}_i \in \mathbb{R}^d$ issus d'un réseau de neurones) ; elles peuvent également être des entités symboliques. La **fonction de synergie** $S(i,j)$ est alors conçue de manière à prendre en compte des mesures issues tant de la comparaison de vecteurs que de la compatibilité entre ensembles de règles. Par exemple, pour une entité symbolique \mathcal{E}_{sym} et une entité sub-symbolique \mathcal{E}_{sub} , on peut définir une synergie mixte

$$S(\mathcal{E}_{\text{sym}}, \mathcal{E}_{\text{sub}}) = f(\sigma(\mathcal{E}_{\text{sym}}, \mathcal{E}_{\text{sub}})),$$

où σ représente une métrique de compatibilité (telle qu'une distance sémantique ou une mesure d'inférence) et f est une fonction d'activation adaptée. L'idée est que si les données sub-symboliques confirment ou valident le contenu d'un bloc de règles, le lien ω_{ij} entre ces deux entités se renforce automatiquement, selon la règle de mise à jour DSL, telle que présentée dans la section (2.2.2) et analysée dans (2.3.1) et (2.4.1).

La **plasticité** du SCN se traduit par l'évolution dynamique des pondérations ω_{ij} qui se mettent à jour, par exemple, selon la règle additive

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta [S(i,j) - \tau \omega_{ij}(t)],$$

où η est le taux d'apprentissage et τ le terme de décroissance. Lorsque $S(i,j)$ est une fonction qui intègre à la fois des éléments symboliques et sub-symboliques, la dynamique de ω_{ij} permet une **auto-organisation** fine des entités en clusters, reflétant des affinités multiples.

C. Interactions entre les couches symbolique et sub-symbolique

L'une des idées centrales de l'approche neuro-symbolique est de permettre une **cohabitation** entre une couche symbolique, où résident des règles explicites et des concepts, et une couche sub-symbolique, issue des techniques d'apprentissage profond. Dans un SCN, cela se traduit par la coexistence d'entités \mathcal{E}_i qui peuvent être soit des vecteurs d'activation (issues d'un backbone de deep learning, comme un ResNet ou un Transformer), soit des blocs de règles ou des concepts logiques. Les liens entre ces entités sont modulés par une fonction de synergie $S(i, j)$ conçue pour comparer des types de données hétérogènes.

Par exemple, dans une application pratique, un bloc de règles décrivant des relations logiques (par exemple, la règle *if animal then vertebrate*) pourra interagir avec des embeddings issus d'images ou de textes. Si un grand nombre d'images correspondant à des animaux se retrouvent alignées avec ce bloc, la synergie mesurée augmentera, ce qui, selon la dynamique du DSL, renforcera la pondération ω_{ij} correspondante. Ainsi, le SCN pourra, de manière auto-organisée, former un cluster qui associe les représentations symboliques et sub-symboliques autour d'un thème commun.

Les avantages de cette approche résident notamment dans la **flexibilité** et la **richesse** de l'auto-organisation, car le réseau n'est pas contraint à une seule modalité de représentation. Comme indiqué dans la section (2.5.2.1), l'adaptation aux divers types d'entités permet d'étendre l'application du DSL à des environnements multimodaux, où coexistent des informations perceptuelles et des concepts explicites.

D. Applications et implications de la synergie neuro-symbolique

Les synergies entre IA symbolique et plasticité d'un SCN ouvrent de nombreux champs d'application :

- **Systèmes experts évolutifs** : Dans ces systèmes, les règles ne sont pas figées mais évoluent en fonction des données. Un SCN peut adapter dynamiquement les pondérations entre différentes règles en fonction de leur performance ou de leur pertinence dans un environnement changeant. Cette approche permet de résoudre le problème de la rigidité des systèmes experts classiques.
- **Raisonnement multimodal** : En intégrant des représentations issues à la fois de sources sub-symboliques (comme des images ou du texte) et symboliques (comme des ontologies ou des règles), le SCN facilite la formation de clusters hybrides capables d'exploiter des informations complémentaires. Par exemple, dans le domaine de la vision par ordinateur, un bloc de règles décrivant des relations spatiales pourrait être combiné avec des embeddings d'images pour améliorer la reconnaissance d'objets.
- **Systèmes neuro-symboliques unifiés** : L'approche permet de construire des architectures où la logique symbolique, avec ses avantages en termes d'interprétabilité et de raisonnement

formel, est intégrée dans un cadre d'apprentissage profond, offrant ainsi des solutions plus robustes et adaptatives pour des tâches complexes.

Ces exemples illustrent comment la **plasticité** du SCN, combinée à une **fonction de synergie** capable de traiter des entités hétérogènes, peut conduire à une auto-organisation efficace des connaissances, permettant d'associer de manière dynamique des informations issues de différentes modalités.

Nous verrons dans la section (2.5.2.3) comment ces idées seront approfondies dans des **chapitres futurs** (notamment les chapitres 5 et 13), afin de déployer des systèmes neuro-symboliques intégrés qui tirent pleinement parti de la complémentarité entre traitement symbolique et sub-symbolique dans le cadre du DSL.

2.5.2.3. Chapitres futurs (5, 13) où l'on détaillera la cohabitation symbolique-subsymbolique

Les développements présentés dans les sections (2.5.2.1) et (2.5.2.2) ont montré que le **DSL** (Deep Synergy Learning) se distingue par sa capacité à accueillir des entités de nature hétérogène, allant des représentations sub-symboliques (vecteurs, embeddings) aux entités symboliques (ensembles de règles, concepts logiques). Cette cohabitation est rendue possible par une fonction de synergie $S(i,j)S(i,j)$ capable d'évaluer, de manière adaptée, la compatibilité ou la complémentarité entre des entités aux représentations très différentes. Toutefois, afin de passer du cadre théorique à une application concrète et opérationnelle, il est nécessaire de préciser les modalités d'implémentation et d'intégration de ces idées dans un système complet. C'est exactement ce que viseront les **Chapitre 5** et **Chapitre 13** de notre ouvrage.

Dans le **Chapitre 5**, nous aborderons tout d'abord l'**architecture générale** du Synergistic Connection Network (SCN). Ce chapitre détaillera la manière dont les pondérations $\omega_{i,j}|\omega_{i,j}$ sont structurées et stockées (par exemple sous forme de matrices denses ou creuses) ainsi que les différentes stratégies d'**initialisation** et de mise à jour. Nous y décrirons également comment les données d'entrée—qu'elles soient issues d'un réseau de neurones pré-entraîné (représentations sub-symboliques) ou de modules symboliques (ensembles de règles, ontologies)—sont intégrées dans le SCN. La mise en œuvre pratique des routines de calcul de la synergie, qui peut inclure des mesures de distance euclidienne, de similarité cosinus ou même des évaluations de compatibilité logique, sera explicitée à travers des exemples concrets et du pseudo-code. Ce chapitre servira ainsi de passerelle entre le concept théorique du DSL et sa traduction en algorithmes exécutables, en mettant particulièrement l'accent sur la cohabitation des entités symboliques et sub-symboliques au sein d'un réseau unifié.

Le **Chapitre 13** quant à lui se penchera sur la dimension plus globale et cognitive de l'approche, en explorant la perspective d'une **IA forte** ou d'un raisonnement cognitif avancé reposant sur une fusion neuro-symbolique. Ici, l'objectif sera d'étudier comment le SCN peut non seulement organiser de manière auto-adaptative des représentations sub-symboliques mais aussi, de façon dynamique, pondérer et moduler des ensembles de règles ou des concepts logiques. Ce chapitre examinera, par exemple, comment la plasticité des pondérations $\omega_{i,j}|\omega_{i,j}$ peut servir à ajuster l'importance relative de certaines règles en fonction de leur confirmation par des données empiriques. Nous discuterons également de l'**ordonnancement** et de la **mise à jour** des blocs de règles en interaction avec les embeddings issus de données perceptuelles, et nous montrerons comment ces mécanismes peuvent contribuer à une meilleure cohérence globale du système.

L'approche neuro-symbolique proposée vise à dépasser la dichotomie classique entre réseaux de neurones et systèmes experts, en intégrant ces deux dimensions dans une même architecture adaptative.

2.5.3. Apport en Robotique et Contrôle Adaptatif

Les principes du **DSL** (Deep Synergy Learning) — basés sur la dynamique adaptative de liaisons $\omega_{i,j}$ et la formation de clusters en fonction de la **synergie** — ne se cantonnent pas aux systèmes algorithmiques ou cognitifs où ils trouvent également un **terrain d'application** dans la **robotique**. En particulier, la notion de **coordination sensorimotrice** peut être envisagée comme un **réseau** d'entités (capteurs, actionneurs, modules décisionnels) dont les liens se renforcent ou s'affaiblissent selon la **complémentarité** et l'**efficacité** qu'ils offrent. La section (2.5.3) examine l'**apport** du DSL en robotique et en contrôle adaptatif. Nous commençons (2.5.3.1) par rapporter quelques **retours d'expériences** de laboratoires ayant exploré la synergie adaptative pour la **coordination sensorimotrice**.

2.5.3.1. Retours d'expériences dans des laboratoires : usage de la synergie adaptive pour la coordination sensorimotrice

Dans le domaine de la robotique sensorimotrice, où la coordination entre divers capteurs et effecteurs constitue un enjeu crucial pour l'adaptation en environnement dynamique. Ces études, que l'on retrouve en référence dans la section (2.5.3.1) et reliées aux concepts développés dans les sections antérieures telles que (2.5.2.1) et (2.5.2.2), mettent en lumière comment l'auto-organisation des pondérations $\omega_{i,j}$ peut être utilisée pour réguler de manière adaptative la coordination sensorimotrice.

A. Contexte général en robotique sensorimotrice

Dans une configuration robotique typique, un robot est équipé d'un ensemble hétérogène de capteurs — caméras, LiDAR, gyroscopes, microphones, etc. — ainsi que d'effecteurs (moteurs de roues, bras manipulateurs, etc.). Historiquement, la coordination de ces dispositifs s'appuyait sur des schémas centralisés ou des architectures modulaires fixes, qui ne prenaient pas toujours en compte la variabilité environnementale. Dans ce contexte, le DSL propose de considérer chaque capteur ou module comme une entité \mathcal{E}_i dont la relation avec un autre module \mathcal{E}_j est quantifiée par une pondération $\omega_{i,j}$. Cette pondération évolue selon la règle de mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

décrise en détail dans les sections (2.3.1.1) et (2.3.1.2), et permet ainsi d'exploiter la **synergie adaptative** entre capteurs et effecteurs. Lorsque la synergie $S(i,j)$ entre deux modules est élevée—par exemple, lorsque deux caméras fournissent des informations complémentaires pour l'estimation de la profondeur ou pour la détection d'obstacles—la pondération $\omega_{i,j}$ se renforce,

favorisant une fusion efficace des données. Inversement, si un capteur est perturbé ou fournit des informations moins fiables, son lien avec d'autres modules est progressivement atténué.

B. Exemples de laboratoires et de scénarios rapportés

(a) Coordination multi-capteurs (Laboratoire A)

Dans un laboratoire spécialisé en vision stéréoscopique, un SCN a été déployé pour coordonner plusieurs caméras disposées à différents angles ainsi qu'un système inertiel (IMU). Les expériences ont montré que lorsque la qualité de la caméra frontale se détériore (par exemple, à cause d'un éblouissement), le système réévalue dynamiquement les pondérations, renforçant le lien entre la caméra latérale et l'IMU. Ainsi, la fusion visuelle se fait de manière adaptative, garantissant une estimation de la profondeur plus robuste. Ce mécanisme repose sur la même règle de mise à jour décrite en (2.3.1.1) et (2.5.3.1), où l'auto-organisation des $\omega_{i,j}$ permet de moduler la contribution de chaque capteur.

(b) Coordination bras manipulateur et retours tactiles (Laboratoire B)

Un autre laboratoire s'est intéressé à la manipulation d'objets par des robots équipés de capteurs tactiles. Dans ce contexte, la synergie entre les capteurs tactiles et le contrôle moteur du bras manipulateur a été mise en œuvre à l'aide d'un SCN. En cas de prise stable, la pondération entre le capteur tactile et le module de commande du bras augmente, tandis que des erreurs (glissement ou échec de préhension) entraînent une diminution de $\omega_{tactile,moteur}$. Ce processus adaptatif, qui permet d'auto-ajuster les stratégies de préhension, illustre bien comment la mise à jour des pondérations assure l'émergence de clusters de connexions robustes, comme évoqué dans les sections (2.5.3.1) et (2.5.2.2).

(c) Coordination dans un essaim de robots (Laboratoire C)

Dans le domaine de la **swarm robotics**, chaque robot est considéré comme une entité \mathcal{E}_i . Un SCN a été utilisé pour modéliser la coopération entre les robots, où la pondération $\omega_{i,j}$ reflète la capacité de deux robots à collaborer efficacement (par exemple, pour l'exploration ou le transport). Lorsqu'une coopération productive est détectée (par exemple, un échange réussi de signaux ou une mutualisation des ressources), les liens entre ces robots se renforcent. Inversement, des interférences ou des redondances conduisent à une diminution des pondérations. Cette approche décentralisée permet à l'essaim de s'organiser en clusters spécialisés sans qu'un contrôleur central ne soit nécessaire, ce qui renforce la flexibilité et la résilience du système.

C. Bilan des expérimentations : Impact sur l'adaptativité et la robustesse

Les expériences rapportées dans les laboratoires montrent plusieurs points essentiels :

- **Adaptation en temps réel** : Grâce à la règle de mise à jour, telle que rappelée en (2.3.1.1), le SCN ajuste continuellement les pondérations $\omega_{i,j}$ en fonction de la qualité perçue de la synergie $S(i,j)$. Ce mécanisme permet au système de se reconfigurer instantanément en

réponse aux variations environnementales, qu'il s'agisse d'une dégradation du signal d'un capteur ou d'une nouvelle situation d'interaction entre modules.

- **Décentralisation** : L'approche DSL évite l'utilisation d'un contrôleur hyper-centralisé. Chaque entité, qu'elle soit un capteur, un effecteur ou un robot dans un essaim, ajuste localement ses liaisons. Cette autonomie locale favorise la formation d'un réseau d'interactions organiquement structuré, conformément aux principes développés en (2.5.2.1) et (2.5.2.2).
- **Robustesse face aux perturbations** : La capacité d'auto-organisation permet de réduire l'impact des défaillances locales. Par exemple, dans le laboratoire A, la diminution de la synergie d'un capteur dégradé se traduit par une baisse de sa pondération, limitant ainsi l'influence négative de ce capteur sur la fusion globale. De manière similaire, dans les configurations en essaim (Laboratoire C), la redondance des liens garantit que la perte d'un robot ne perturbe pas l'organisation globale du réseau.

Ces observations, en écho aux concepts théoriques détaillés dans les sections (2.3.1) à (2.5.2.2), confirment l'intérêt d'une approche basée sur la **synergie adaptative** pour la coordination sensorimotrice en robotique.

2.5.3.2. Potentiel pour gérer des flottes de robots, chacun se reliant localement, formant un SCN global

Dans le cadre de la robotique collaborative, le concept de **Deep Synergy Learning (DSL)** offre une perspective innovante pour orchestrer la coordination entre de nombreux agents autonomes. En effet, chaque robot, considéré comme une entité \mathcal{E}_i au sein du réseau, dispose de l'opportunité de modifier ses liaisons $\omega_{i,j}$ de manière autonome, suivant une règle de mise à jour telle que

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ représente le **taux d'apprentissage** et $\tau > 0$ le **coefficient de décroissance**. La **synergie** $S(i,j)$ entre deux robots peut être définie à partir de critères variés, tels que la proximité géographique, la complémentarité des capteurs, ou encore la réussite des tâches coopératives. Cette approche, que l'on retrouve également dans la section 2.5.3.1 relative aux retours d'expériences en robotique sensorimotrice, permet à chaque robot d'ajuster localement ses connexions sans recourir à une supervision centrale.

Sur le plan théorique, l'architecture du SCN pour une flotte de robots repose sur la capacité de chaque agent à calculer sa **synergie locale** avec ses voisins immédiats, ce qui s'exprime par des mises à jour distribuées des pondérations $\omega_{i,j}$. Si la synergie est élevée entre deux robots, le poids associé se renforce, tandis qu'en cas de collaboration inefficace ou de divergence de mission, le poids se réduit. Ce mécanisme favorise l'émergence de **clusters** d'agents fortement connectés, illustrant une organisation auto-adaptative du réseau. Les aspects mathématiques de cette dynamique reposent sur des équations différentielles discrètes, et l'auto-organisation du système est comparable à une descente de gradient dans un espace de configuration défini par les pondérations, comme nous l'avons détaillé dans la section 2.5.1.1.

L'un des avantages majeurs de cette approche réside dans la **décentralisation** de la mise à jour. Chaque robot, en traitant localement l'information issue de ses interactions, contribue à la formation d'un **SCN global** sans intervention d'un contrôleur centralisé. Cette propriété est cruciale lorsque le nombre de robots est élevé, car elle permet d'éviter une surcharge computationnelle et favorise la **robustesse** du système. En effet, la défaillance d'un robot n'affecte que localement le réseau, et le système peut se réorganiser automatiquement pour compenser la perte, une caractéristique déjà évoquée dans la section **2.5.3.1**.

Par ailleurs, cette méthode permet d'implémenter des mécanismes d'**apprentissage continu** et de **robustesse**, comme ceux détaillés dans les chapitres futurs, notamment le **Chapitre 9** sur l'apprentissage continu et le **Chapitre 11** sur la robustesse. En pratique, la synergie $S(i,j)$ peut être ajustée en temps réel en fonction de la performance des collaborations, de la qualité des données de capteurs ou d'autres critères d'interaction, de sorte que les pondérations $\omega_{i,j}$ évoluent de manière dynamique. Par exemple, si deux robots collaborent efficacement pour réaliser une tâche, leur pondération tend à converger vers une valeur élevée, ce qui renforce leur intégration dans un cluster. À l'inverse, une interaction moins productive entraîne une diminution de $\omega_{i,j}$, permettant ainsi au réseau de se réorganiser de façon adaptative.

L'architecture ainsi proposée se distingue par sa **scalabilité**. Chaque robot n'a besoin de gérer que ses interactions locales, souvent limitées à un voisinage restreint, ce qui réduit la complexité globale du système. De plus, en exploitant des algorithmes de mise à jour inspirés du **recuit simulé** et des **mécanismes d'inhibition** (voir sections **2.4.4** et **2.4.5.1**), le SCN peut éviter de se figer dans un minimum local non optimal et favoriser l'émergence d'une organisation plus stable et cohérente. Ainsi, l'auto-organisation du réseau repose sur une dynamique où la **synergie locale** guide la formation de clusters qui, au niveau global, constituent une structure robuste et résiliente face aux variations de l'environnement ou aux perturbations.

2.5.4. Comparaison avec d'Autres Paradigmes (RL, GNN, etc.)

Les sections précédentes (2.5.1 à 2.5.3) ont montré que le **DSL** (Deep Synergy Learning) se prête à des **convergences** multiples où il s'intègre aussi bien dans l'apprentissage profond, l'IA symbolique que dans la robotique adaptative. Pour clore ce panorama, la section 2.5.4 compare brièvement le DSL à d'autres paradigmes très en vogue en IA moderne, notamment les **Graph Neural Networks (GNN)** et les approches de **Reinforcement Learning (RL)** ou de **clustering**. Nous commencerons (2.5.4.1) par situer le DSL vis-à-vis des GNN, avant de voir (2.5.4.2) ce qui le distingue par rapport aux algorithmes de clustering ou aux méthodes de RL, puis de conclure (2.5.4.3) en renvoyant à des analyses plus approfondies dans les prochains chapitres 6, 7, 8.

2.5.4.1. Où se situe le DSL vis-à-vis des Graph Neural Networks (GNN) ?

A. Rappel sur les GNN

Les **GNN** traitent un **graphe** d'entrée, où chaque noeud est associé à un **vecteur** (feature) et les arêtes peuvent porter un label ou un poids. On applique des "convolutions" ou des "aggregations" sur les voisins pour mettre à jour les **représentations** de chaque noeud.

L'objectif est souvent de réaliser une **tâche** supervisée ou semi-supervisée (classification de nœuds, prédiction de liens, etc.). Les poids du réseau (couches GNN) sont **appris** par backpropagation, en prenant en entrée la structure du graphe et les features des nœuds.

Dans la plupart des GNN, la **topologie** du graphe est **fixée** (ou du moins pas totalement réinventée à chaque itération). On applique des couches de type GNN (GCN, GAT, etc.) pour extraire des embeddings plus riches de chaque nœud ou de l'ensemble du graphe.

B. DSL = évolution de la matrice de liaison

Le **DSL** fait évoluer la **matrice** $\omega_{i,j}$ (pondérations d'un SCN) en fonction d'une **synergie** $S(i,j)$. La structure du graphe n'est donc pas statique où les liens et leurs intensités peuvent se créer, se renforcer ou disparaître au fil du temps selon la dynamique auto-organisée.

Au lieu de se baser sur un graphe fixe et d'y appliquer des "pass" de convolution (comme en GNN), le **DSL** vise la **construction** ou la **réorganisation** même de ce graphe. Il ne repose pas obligatoirement sur un objectif supervisé ; il peut fonctionner dans une **logique** auto-organisée ou faiblement supervisée.

Dans l'ensemble, on peut relever plusieurs points communs entre les GNN et le DSL. D'abord, les deux approches manipulent des entités (nœuds) ainsi que leurs liaisons (arêtes ou pondérations). Par ailleurs, il est possible d'associer des *features* spécifiques à chaque entité (ou nœud), de sorte que le réseau intègre des informations additionnelles sur ces entités.

Sur le plan des différences majeures, la première tient à la **structure** où les GNN partent d'un graphe dont la topologie est définie au préalable, tandis que dans le DSL, les connexions $\omega_{i,j}$ évoluent, se créent ou se suppriment au fil de l'exécution selon la dynamique auto-organisée.

La seconde différence concerne le **type d'apprentissage** où les GNN relèvent généralement d'un cadre supervisé basé sur la minimisation d'une perte via backpropagation, tandis que le DSL met en œuvre des mises à jour dites "locales", le plus souvent sans supervision stricte.

La troisième divergence réside dans l'**objectif** poursuivi où les GNN visent la réalisation d'une tâche précise comme la classification ou la régression, tandis que le DSL se consacre à l'auto-organisation d'un réseau d'entités en fonction de leurs synergies, sans impératif d'inférence ou de prédiction.

2.5.4.2. Qu'apporte-t-il de distinct par rapport aux algorithmes de clustering traditionnels ou aux approches de renforcement ?

Le **Deep Synergy Learning (DSL)** se présente comme une approche qui auto-organise un réseau de pondérations $\{\omega_{i,j}\}$ au fil d'une dynamique évolutive, en mettant à jour continuellement chaque connexion selon la règle

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ désigne le **taux d'apprentissage**, $\tau > 0$ le **coefficients de décroissance**, et $S(i,j)$ représente la **synergie** mesurant l'affinité locale entre les entités. Cette formulation, qui a été

détaillée dans les sections précédentes telles que **2.5.1.1** et **2.5.3.1**, contraste fortement avec les approches traditionnelles de clustering et de renforcement.

D'un côté, les **algorithmes de clustering** traditionnels comme **k-means** ou **DBSCAN** procèdent généralement en mode *batch*. Par exemple, dans k-means, chaque point de données est assigné au centroïde le plus proche, puis les centroïdes sont recalculés par minimisation de la somme des distances intra-cluster. Cette procédure itérative se poursuit jusqu'à convergence, et le nombre de clusters, k , est fixé a priori. Ces méthodes reposent sur une mesure de distance statique, ce qui aboutit à une partition fixe de l'ensemble des données. En revanche, dans le DSL, les pondérations $\omega_{i,j}$ évoluent en temps réel en réponse à la synergie locale entre entités, sans imposer une structure de partition rigide. Ainsi, les clusters dans le DSL apparaissent de manière émergente et évolutive, permettant à une même entité de maintenir des connexions significatives avec plusieurs groupes au lieu d'être confinée à une unique partition déterminée à l'avance.

D'un autre côté, les approches de **Renforcement (RL)** se fondent sur un cadre dans lequel un agent, en interaction avec un environnement, reçoit un état s_t , choisit une action a_t et perçoit une récompense r_t . L'objectif est d'apprendre une politique $\pi(a | s)$ qui maximise la récompense cumulative sur le long terme. Les algorithmes de RL, tels que le Q-learning ou les méthodes de policy gradient, se concentrent donc sur la prise de décision séquentielle guidée par un signal de récompense explicite. Dans le DSL, la mise à jour des pondérations est déterminée par la **synergie** $S(i,j)$ locale et n'implique pas de signal de récompense global ni de choix d'actions dans un espace d'états. Par conséquent, le DSL s'inscrit dans une dynamique d'**auto-organisation** où chaque entité ajuste ses connexions de façon autonome, en se basant sur des critères de compatibilité mesurés localement, plutôt que d'optimiser une fonction de récompense cumulative.

Les **avantages** du DSL résident dans sa **flexibilité** et sa **capacité adaptative**. En effet, la mise à jour continue des $\omega_{i,j}$ permet au système de s'ajuster en temps réel aux variations des données, contrairement aux méthodes de clustering traditionnelles qui opèrent sur un ensemble de données fixe. De plus, le DSL ne se contente pas de partitionner l'espace des données ; il permet à une entité d'entretenir des liens multiples, reflétant ainsi la complexité des interactions dans des environnements réels. Du côté des approches de renforcement, le DSL offre une méthode décentralisée d'auto-organisation sans nécessiter une modélisation explicite des récompenses, ce qui peut simplifier l'implémentation dans des systèmes multi-agents où la coordination se fait par des interactions locales plutôt que par des signaux de récompense centralisés.

Cependant, cette **vision énergétique** et adaptative présente également des limites. Lorsque la synergie $S(i,j)$ varie de manière significative au fil du temps ou dépend fortement des pondérations elles-mêmes, la fonction potentielle globale ne reste plus statique et l'interprétation comme une descente de gradient devient moins évidente. Par ailleurs, dans des environnements hautement dynamiques, les mises à jour continues peuvent conduire à des oscillations ou à une instabilité qui ne sont pas facilement expliquées par les méthodes de clustering classiques ou par les algorithmes de renforcement qui, eux, reposent sur des critères de performance cumulés. De plus, la fusion des deux paradigmes (clustering statique et apprentissage par renforcement) est envisageable, mais elle nécessite une conception hybride qui peut complexifier la mise en œuvre du système.

En conclusion, le **DSL** se distingue à la fois des algorithmes de clustering traditionnels et des approches de renforcement par sa capacité à produire une **auto-organisation évolutive** du réseau de pondérations. La dynamique continue de mise à jour, basée sur la synergie locale, permet de

former des clusters adaptatifs sans imposer de partitions fixes, tout en se passant d'un cadre décisionnel centré sur la maximisation de récompenses cumulées. Cette approche représente une alternative robuste et flexible, particulièrement adaptée aux systèmes complexes et distribués où les interactions entre entités sont non statiques et évolutives. Nous verrons, dans les chapitres **6**, **7** et **8**, comment ces principes sont intégrés dans des architectures d'IA avancées, démontrant ainsi la complémentarité et la richesse du DSL par rapport aux méthodes classiques.

2.5.4.3. Discussion brève avant de laisser place à l'analyse plus poussée dans les chapitres **6**, **7**, **8**

Dans les sections **2.5.4.1** et **2.5.4.2**, nous avons établi des comparaisons détaillées entre le **DSL** (Deep Synergy Learning) et divers paradigmes traditionnels, tels que les **Graph Neural Networks (GNN)**, les méthodes classiques de **clustering** et les approches de **Renforcement (RL)**. Il apparaît clairement que le DSL occupe un **créneau distinct** puisqu'il se concentre sur la **dynamique adaptative** des connexions $\omega_{i,j}$ qui se mettent à jour continuellement en fonction de la **synergie** locale, sans imposer de schéma de supervision tel que requis dans les GNN, ni de partition fixe comme dans k-means, et sans recourir à un signal de récompense global comme c'est le cas en RL. Cette caractéristique intrinsèque permet, en outre, de combiner le DSL avec ces paradigmes afin de renforcer leurs capacités respectives, par exemple en ajustant dynamiquement la structure d'un graphe dans un GNN ou en apportant une dimension d'auto-organisation dans un cadre de RL multi-agents.

La suite de notre ouvrage se poursuivra avec des développements plus approfondis dans les chapitres **6**, **7** et **8**, qui visent à clarifier et à intégrer ces notions dans des systèmes d'intelligence artificielle complexes. Le **Chapitre 6**, intitulé « **Apprentissage Synergique Multi-Échelle** », traitera de la capacité du DSL à gérer plusieurs niveaux ou granularités d'entités, en mettant en lumière la manière dont la mise à jour des $\omega_{i,j}$ peut être effectuée dans une logique multi-échelle. Dans ce contexte, certains rapprochements avec les GNN et d'autres structures hiérarchiques seront approfondis, permettant ainsi de comprendre comment une architecture en couches peut émerger de l'auto-organisation des liens.

Le **Chapitre 7**, sous le titre « **Algorithmes d'Optimisation et Méthodes d'Adaptation Dynamique** », présentera des techniques issues du recuit simulé, de la compétition inhibitrice (cf. section **2.4.4**) ainsi que des approches inspirées de la physique statistique (cf. section **2.4.5.1**) pour optimiser la configuration d'un SCN. Ce chapitre abordera également les questions de **complexité** et de **scalabilité** lorsque le nombre d'entités augmente, en proposant des solutions pour assurer la convergence et la robustesse du système dans un environnement dynamique.

Le **Chapitre 8**, intitulé « **DSL Multimodal : Fusion de la Vision, du Langage et des Sons** », se focalisera sur les aspects pratiques de la fusion de données issues de différentes modalités. Ici, le DSL sera confronté aux approches traditionnelles de clustering multimodal et de RL multimodal, et l'on examinera comment la **synergie adaptative** entre divers types de données (images, textes, signaux audio) peut être exploitée pour former des représentations communes, enrichissant ainsi les performances du système global.

En guise de **conclusion** de cette discussion préliminaire, il apparaît que la dynamique du DSL se distingue des autres méthodes par sa capacité à organiser un réseau de pondérations de manière **auto-évolutive**. Ainsi, tandis que les GNN reposent sur une structure de graphe relativement fixe et un apprentissage supervisé, le DSL met en œuvre une **mise à jour continue** des $\omega_{i,j}$ dans un contexte non supervisé. De même, les méthodes de clustering traditionnelles segmentent un ensemble de données statique, et les approches de RL se concentrent sur l'optimisation séquentielle d'actions via une récompense cumulative, alors que le DSL privilégie l'**auto-organisation** des relations sans imposer de partition ni de politique d'actions prédéfinie.

Les chapitres **6, 7 et 8** approfondiront ces **intégrations** et complémentarités, démontrant comment le SCN peut s'insérer dans des architectures d'IA plus larges et plus riches. Ce passage de la théorie à l'ingénierie concrète ouvre ainsi la voie à des systèmes d'IA capables de combiner de manière harmonieuse les paradigmes de **clustering**, de **renforcement** et de **modélisation graphique**, contribuant à l'émergence de solutions adaptatives et évolutives dans des environnements complexes.

2.5.5. Transition vers les Chapitres Suivants

L'ensemble de la section 2.5 (**Perspectives Historiques et Liens avec l'IA Moderne**) a révélé comment le **DSL** (Deep Synergy Learning) s'inscrit dans la continuité de plusieurs **courants** de l'IA où il entretient des liens avec l'apprentissage profond (2.5.1), la neuro-symbolique (2.5.2), la robotique sensorimotrice (2.5.3) et la comparaison avec les GNN, le clustering et le RL (2.5.4). Pour **conclure** ce chapitre 2, la section (2.5.5) annonce la **transition** vers les chapitres plus concrets de la suite — notamment le **Chapitre 3**, qui portera sur la **représentation concrète** des entités et la définition pratique de la **synergie**, ainsi que les **Chapitres 4 et 5**, où l'on traitera davantage la **dynamique d'auto-organisation** et l'**architecture SCN** dans un cadre d'ingénierie.

2.5.5.1. Le prochain chapitre (3) traitera de la représentation concrète des entités et de la façon de définir la fonction de synergie dans divers contextes (image, texte, capteurs)

Dans la suite de ce travail, nous nous attacherons à passer d'une conceptualisation théorique du **DSL** (Deep Synergy Learning) à une mise en œuvre plus opérationnelle, en particulier en ce qui concerne la représentation concrète des **entités**. Le **Chapitre 3**, qui figure dans la table des matières, abordera de manière détaillée la problématique de la formalisation de chaque entité \mathcal{E}_i . Nous examinerons notamment si ces entités doivent être considérées comme des **vecteurs**—par exemple, des embeddings issus d'un réseau de neurones convolutionnel (CNN) ou d'un Transformer appliqué à des textes—ou si elles peuvent également prendre la forme de structures symboliques, telles que des ensembles de règles ou des ontologies. Dans ce contexte, il sera essentiel d'identifier les **structures de données** appropriées pour stocker et manipuler ces représentations, qu'il s'agisse de tableaux multidimensionnels, de graphes ou d'autres structures adaptées à la nature et à la dimension des descripteurs.

Un point central de cette démarche réside dans la définition de la **fonction de synergie** $S(i,j)$. Il s'agira d'établir des formules qui quantifient la similarité ou la complémentarité entre deux entités, en fonction du contexte considéré. Par exemple, dans le cas d'images, on pourra adopter une

mesure basée sur la **distance cosinus** ou la distance euclidienne entre des embeddings extraits par un CNN, alors que pour des textes, des mesures de **similarité sémantique** issues de modèles de langage pré-entraînés (tels que BERT) seront privilégiées. Pour des capteurs, la synergie pourra être définie par des fonctions mesurant la cohérence temporelle ou la corrélation des signaux, telles que $S(i, j) = \text{corr}(\mathbf{x}_i, \mathbf{x}_j)$. Dans tous ces cas, la mise en œuvre devra garantir que la dynamique de mise à jour des pondérations $\omega_{i,j}$ du **SCN** (Synergistic Connection Network) soit adaptée à la nature des données, ce qui implique la prise en compte de paramètres tels que le taux d'apprentissage η et le coefficient de décroissance τ dans la règle de mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)].$$

Cette équation, qui sera étudiée en détail dans les chapitres ultérieurs, devra être adaptée pour prendre en compte la diversité des types d'entités présentes dans le réseau.

Nous verrons également comment intégrer la **fonction de synergie** dans un cadre multimodal, où il est nécessaire de fusionner des informations provenant de sources très hétérogènes. Par exemple, dans un scénario de fusion de données, les représentations d'images et de textes devront être combinées au sein du même SCN, ce qui impliquera de définir des mécanismes permettant de normaliser et de pondérer les différentes mesures de synergie selon leur échelle et leur pertinence relative. Ce défi technique est essentiel pour assurer que la dynamique du DSL reste cohérente et exploitable, même lorsque les entrées proviennent de domaines très différents.

Le **Chapitre 3** constituera ainsi une passerelle entre la théorie développée dans le Chapitre 2 et les applications pratiques, en présentant des *pseudo-codes* et des exemples concrets d'implémentation. Il expliquera comment les entités, qu'elles soient purement sub-symboliques ou de nature mixte (symbolique et sub-symbolique), peuvent être représentées par des structures de données adaptées et comment la fonction de synergie $S(i, j)$ est définie pour piloter la dynamique d'auto-organisation des pondérations.

Par ailleurs, la suite de la transition est explicitée dans les sections **2.5.5.2** et **2.5.5.3**, qui détailleront respectivement les aspects de la dynamique d'auto-organisation du SCN dans des contextes appliqués et la cohabitation des idées théoriques avec des principes d'ingénierie. Le **Chapitre 3** sera suivi par les **Chapitres 4 et 5** qui s'orienteront vers la mise en œuvre effective des algorithmes et la validation de la dynamique auto-organisée dans des scénarios réalistes. Cette progression permettra de passer de la conceptualisation abstraite des mécanismes du DSL à une réalisation pratique, en mettant l'accent sur l'extraction des **features** et l'adaptation de la fonction de synergie aux caractéristiques des données, qu'elles soient issues de l'imagerie, du traitement de textes ou des flux sensoriels.

2.5.5.2. Les chapitres 4 et 5 aborderont la dynamique d'auto-organisation plus appliquée, en tirant parti des bases mathématiques introduites ici

Dans ce passage du Chapitre 2 aux chapitres suivants, nous avons posé des fondations théoriques solides pour le **DSL** (Deep Synergy Learning) en définissant précisément ses concepts clés, tels que la dynamique des pondérations $\omega_{i,j}$, les analogies avec la physique statistique, la théorie des systèmes dynamiques et les mécanismes d'**inhibition**. Cependant, il apparaît désormais nécessaire

de transposer ces principes abstraits dans un cadre d'**implémentation concrète** afin de déployer le DSL dans des applications réelles. C'est à ce titre que les **Chapitres 4 et 5** interviendront, après que le **Chapitre 3** aura défini la représentation des entités (issues d'images, de textes, de capteurs, etc.) ainsi que la manière de formaliser la fonction de **synergie** $S(i,j)$.

Dans le **Chapitre 4**, intitulé "**Synergies Émergentes et Auto-Organisation**", l'objectif sera d'approfondir la manière dont la dynamique d'auto-organisation se manifeste au sein d'un SCN (Synergistic Connection Network). Nous passerons d'équations abstraites, telles que

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

à des algorithmes précis et des schémas de mise en œuvre qui permettent de maîtriser et d'exploiter les synergies issues des interactions locales entre entités. La mise en œuvre algorithmique sera détaillée à travers des pseudo-codes qui illustreront la gestion du pas de temps, l'initialisation des pondérations et les stratégies de détection ainsi que de régulation d'oscillations (par exemple, via des mécanismes d'inhibition et de découpage en lots). Ces exemples concrets permettront de comprendre comment, itération après itération, des clusters émergents ou des micro-réseaux stables se forment naturellement à partir de mises à jour locales. Les concepts introduits dans les sections **2.3** et **2.4** – tels que la théorie des attracteurs, la fonction d'énergie potentielle définie en **2.4.3**, et les mécanismes d'inhibition présentés en **2.4.4** – seront mobilisés pour expliquer en pratique comment ajuster les paramètres η et τ pour obtenir une dynamique auto-organisée prévisible et ajustable.

Le **Chapitre 5**, intitulé "**Le Synergistic Connection Network (SCN) : Architecture Générale**", portera sur la conception complète du réseau. Ce chapitre s'attardera sur la construction de l'architecture du SCN en abordant les questions de stockage et d'organisation des pondérations $\omega_{i,j}$. Nous y discuterons des différentes structures de données susceptibles d'être utilisées (matrices denses versus structures creuses, listes d'adjacence, etc.), ainsi que de l'organisation hiérarchique du réseau, qui pourra être structurée en différentes couches ou niveaux de synergie (par exemple, un niveau local contrasté avec un niveau global).

En outre, l'architecture du SCN sera décomposée en plusieurs modules fonctionnels, notamment le module de calcul de la synergie $S(i,j)$, le module de mise à jour des pondérations (intégrant les mécanismes d'inhibition et de saturation), le module de clustering pour extraire des groupes d'entités, et enfin une interface dédiée à l'ingestion de données en temps réel dans le cas d'un SCN évolutif.

Des exemples d'implémentation seront fournis, avec des pseudo-codes illustrant un framework léger, susceptible d'être réalisé en Python ou en C++ couplé à CUDA pour des applications nécessitant un calcul parallèle.

Ces deux chapitres, **4** et **5**, constituent ainsi un pont essentiel entre la théorie développée dans le Chapitre 2 et la phase d'**implémentation pratique** qui sera amorcée au **Chapitre 3**.

La logique de progression consiste d'abord à définir concrètement la représentation des entités et la fonction de synergie dans le Chapitre 3, puis à explorer dans le Chapitre 4 la dynamique auto-organisée par les mises à jour locales, et enfin à concevoir une architecture complète pour le SCN dans le Chapitre 5. Cette transition vers l'ingénierie concrète permet de passer des principes mathématiques et théoriques à une réalisation opérationnelle, qui sera ultérieurement enrichie par

des approfondissements sur le multi-échelle, l'optimisation avancée et l'intégration multimodale dans les chapitres **6**, **7** et **8**. Par ailleurs, les chapitres **9** et **10** aborderont l'apprentissage continu et le feedback coopératif, renforçant ainsi la robustesse globale du système.

2.5.5.3. Conclusion générale sur la place de ce chapitre 2 : on passe désormais d'un cadre historico-théorique à une mise en œuvre plus ingénierie pour initier le SCN

Ce chapitre 2 a permis de poser les fondements théoriques du **DSL** (Deep Synergy Learning) en établissant un panorama conceptuel riche et interdisciplinaire, qui relie des idées issues de la physique statistique, de la théorie des systèmes dynamiques et des modèles biologiques (tels que l'inhibition latérale et l'apprentissage Hebbien) aux approches contemporaines de l'**IA** et de l'auto-organisation.

Les sections précédentes, notamment **2.5.1** qui aborde les précurseurs et l'évolution historique du DSL, **2.5.2** qui explore les potentialités de cohabitation entre entités symboliques et sub-symboliques, **2.5.3** qui présente des retours d'expériences dans le domaine de la robotique sensorimotrice, ainsi que **2.5.4** qui compare le DSL avec les paradigmes classiques (clustering, renforcement, GNN), ont permis de dresser un réseau de concepts centré sur la **synergie adaptative**.

Nous avons vu que la dynamique des pondérations $\omega_{i,j}$ peut être interprétée en termes d'énergie potentielle, de recuit simulé et de mécanismes d'inhibition, offrant ainsi une vision qui, bien que partielle dans les cas non stationnaires, constitue un cadre de référence solide.

En effet, la formulation d'une fonction d'énergie du type

$$\mathcal{J}(\Omega) = - \sum_{i,j} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{i,j} \omega_{i,j}^2 + \dots$$

a permis d'illustrer comment, dans un scénario stationnaire, la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) - \eta \frac{\partial \mathcal{J}}{\partial \omega_{i,j}}|_{\omega_{ij}(t)}$$

peut être assimilée à une **descension de gradient** locale, guidée par la synergie $S(i,j)$ et régulée par le terme de décroissance $\tau \omega_{i,j}$. Ces concepts ont été détaillés dans les sections **2.4.3** et **2.4.4**, qui exposent respectivement les aspects énergétiques et les mécanismes d'inhibition, et ils illustrent comment le DSL se distingue des méthodes statiques classiques.

Ce chapitre se positionne ainsi comme un **socle historico-théorique** en établissant d'une part l'origine et l'évolution des idées qui sous-tendent le DSL (voir **2.5.1** et **2.5.2**) et, d'autre part, en comparant de manière critique le DSL avec d'autres paradigmes de l'**IA**, tels que les algorithmes de clustering traditionnels et les approches de renforcement présentées dans **2.5.4**. Cette approche comparative montre que, contrairement aux modèles fixés ou purement supervisés, le DSL propose une **auto-organisation dynamique** des connexions $\omega_{i,j}$, qui s'adapte en continu aux évolutions des données et aux interactions entre entités.

La place du chapitre 2 dans l'ensemble de l'ouvrage est ainsi double.

Premièrement, il offre une **vue d'ensemble** qui situe le DSL dans le contexte des développements historiques et théoriques de l'IA, en faisant le lien avec les principes de la **physique statistique**, de l'**auto-organisation** et des mécanismes **biologiques** (tels que l'inhibition latérale et la plasticité synaptique).

Deuxièmement, il prépare le terrain pour une transition vers une **mise en œuvre ingénierie** plus concrète, où les concepts théoriques seront traduits en structures de données, algorithmes et architectures logicielles.

Cette transition est explicitée dans les sections **2.5.5.1** et **2.5.5.2**, qui annoncent respectivement la représentation concrète des entités et la dynamique d'auto-organisation, et qui servent de prélude aux chapitres suivants.

En effet, le **Chapitre 3** traitera de la **représentation et de la modélisation** des entités \mathcal{E}_i dans divers contextes (image, texte, capteurs), en définissant précisément la fonction de synergie $S(i, j)$ adaptée à chaque type de données.

Ensuite, les **Chapitres 4** et **5** se concentreront sur la mise en œuvre de la **dynamique d'auto-organisation** et la construction de l'**architecture globale** du SCN, en s'appuyant sur les principes établis ici – tels que la mise à jour des poids, les mécanismes d'inhibition, et les analogies avec des modèles physiques et biologiques.

Enfin, les chapitres ultérieurs (notamment **6**, **7**, **8**, **9** et **10**) approfondiront les aspects **d'optimisation multi-échelle**, **d'apprentissage continu** et de **robustesse** du système.

Chapitre 3 : Représentation et Modélisation des Entités d'Information

3.1. Introduction

Dans ce chapitre, nous nous concentrons sur la **représentation** et la **modélisation** des entités d'information au sein du DSL (Deep Synergy Learning). Tout système d'apprentissage repose sur une certaine façon de **décrire** ses données, dans le DSL, ce sont les entités et leurs propriétés qui déterminent la manière dont la synergie $\omega_{i,j}$ sera calculée et mise à jour. La présentation qui va suivre servira de **pierre angulaire** aux chapitres ultérieurs (notamment Chapitre. 4 sur la dynamique d'auto-organisation, Chapitre. 5 sur l'architecture SCN, etc.), qui tireront directement profit de la qualité de la modélisation initiale.

3.1.1. Contexte et Motivation

La représentation des entités est un **facteur crucial** dans le DSL, si elle est inadéquate, la fonction de synergie $S(i,j)$ ne peut pas refléter correctement les relations entre entités, risquant de compromettre la formation de clusters pertinents ou l'adaptation du réseau. À l'inverse, une bonne modélisation facilite la détection de liens inattendus et l'émergence de structures synergiques riches. L'enjeu est d'autant plus grand que les entités peuvent être de différentes **natures** (images, textes, signaux, concepts symboliques), obligeant à manier des représentations **hétérogènes**.

3.1.1.1. Rappel du Livre : Le DSL Repose sur des Entités d'Information Dont la Qualité de Représentation Conditionne le Calcul de la Synergie et la Dynamique $\omega_{i,j}$

Il a été souligné dans les chapitres précédents que le **Deep Synergy Learning (DSL)** organise la connaissance autour d'un réseau de pondérations $\omega_{i,j}$ ajustées par une règle d'auto-organisation, comme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)].$$

Cette règle ne prend tout son sens que si la **synergie** $S(i,j)$ entre les entités \mathcal{E}_i et \mathcal{E}_j fournit un **indicateur fiable** de leur affinité ou de leur compatibilité. La structure entière du **Réseau de connexion synergétique (Synergistic Connection Network (SCN))** en anglais) repose ainsi sur la capacité à estimer correctement les liens potentiellement utiles, ce qui dépend étroitement de la **qualité des représentations** de chaque entité.

Pour comprendre l'importance de ce point, il suffit de noter que la **dynamique** de l'auto-organisation, fondée sur la formule ci-dessus, renforce ou diminue les liaisons $\omega_{i,j}$ en fonction de $S(i,j)$. Dans le cas où la description des entités serait lacunaire ou bruitée, la **mesure** $S(i,j)$ risque de **s'égarer**, renforçant des liens non pertinents ou affaiblissant des liens au contraire prometteurs. La conséquence peut être un **réseau** qui ne capture pas la structure réelle des données, des **clusters** peu significatifs ou des phénomènes oscillatoires ne reflétant plus la nature véritable du système.

À l'inverse, si chaque entité \mathcal{E}_i est modélisée sous un format riche, cohérent et représentatif de ses caractéristiques internes, la **synergie** $S(i, j)$ devient un détecteur fiable des regroupements logiques ou des similitudes profondes. Il en résultera une **auto-organisation** claire, faisant émerger des liens forts là où la complémentarité est avérée, et réduisant ceux qui n'apportent que peu de cohérence.

Dans l'optique du **DSL**, cette qualité de représentation peut être sub-symbolique (vecteurs, embeddings, distributions de probabilité) ou symbolique (règles logiques, concepts, graphes), ou encore un **mélange** de ces approches dans un cadre **hybride**.

Les **méthodes** d'apprentissage profond (transformers, autoencodeurs, réseaux génératifs, etc.) fournissent souvent des vecteurs latents puissants et expressifs pour \mathcal{E}_i , tandis que les systèmes symboliques (ontologies, logiques descriptives, axiomes) véhiculent une information formelle ou sémantique plus explicite. Chaque choix a un **impact** direct sur le calcul

$$S(i, j) = f(\mathbf{r}(i), \mathbf{r}(j)),$$

où $\mathbf{r}(i)$ désigne la représentation de \mathcal{E}_i . L'**exactitude** et la **complétude** de $\mathbf{r}(i)$ s'avèrent alors cruciales pour guider le processus d'auto-organisation vers des **clusters** ou des **groupements** factuellement cohérents, révélant de manière fiable la structure cachée des données.

Du point de vue théorique, il apparaît donc que la **représentation** constitue un *pré-requis* indispensable à l'efficacité de l'ensemble du **DSL**. Même une très bonne règle de mise à jour $\omega_{i,j}$ échouera à produire une auto-organisation pertinente si $S(i, j)$ est trompeuse, ce qui arrive lorsque $\mathbf{r}(i)$ et $\mathbf{r}(j)$ ne contiennent pas d'informations réellement discriminantes ou significatives. À l'inverse, une représentation soigneusement conçue amplifiera l'action de la synergie, rendant la **mise à jour** plus rapide et plus stable vers un **réseau** présentant des **structures** solides, qu'il s'agisse de clusters distincts ou de sous-réseaux de coopération thématique.

3.1.1.2. Besoin d'Aborder la Diversité des Entités : Sub-Symboliques (Vecteurs, Embeddings), Symboliques (Règles, Concepts), Hybrides, Multimédias...

Dans le **Deep Synergy Learning**, un **enjeu majeur** réside dans la prise en compte de **données** dont la nature dépasse le simple cadre vectoriel homogène. Les entités manipulées peuvent relever du **sub-symbolique** (des vecteurs numériques issus de l'apprentissage profond), du **symbolique** (des règles logiques ou des concepts formels) ou d'une **hybridation** de ces deux approches. Plus encore, elles peuvent provenir de sources **multimédias** (images, sons, textes, vidéos) que le **DSL** souhaite traiter de manière intégrée. Cette **diversité** impose donc de **définir** des modèles de représentation appropriés, car la fonction de **synergie** $S(i, j)$ dépend directement de la manière dont $\mathbf{r}(i)$ et $\mathbf{r}(j)$ sont encodées.

Un premier registre rassemble les entités **sub-symboliques**, généralement décrites par des **vecteurs** ou des **embeddings**. Une telle approche s'avère particulièrement efficace pour modéliser des données continues ou massives, par exemple, un extrait textuel peut être projeté dans un espace latent à l'aide d'un modèle de langue, ou une image être convertie en un vecteur de caractéristiques via un **réseau neuronal convolutif**. En pratique, la **synergie** $S(i, j)$ se ramène souvent à une **similarité vectorielle** (cosinus, euclidienne), notée

$$\mathbf{S}(i, j) = \text{sim}(\mathbf{r}_{\text{subsym}}(i), \mathbf{r}_{\text{subsym}}(j)),$$

où $\mathbf{r}_{\text{subsym}}(i)$ est un **embedding** associé à l'entité \mathcal{E}_i . De tels vecteurs favorisent la robustesse au bruit et la flexibilité face aux évolutions futures, mais peuvent se heurter à une **interprétabilité** limitée, puisqu'il demeure malaisé de comprendre le contenu d'un embedding hautement dimensionnel.

À l'opposé, les entités **symboliques** sont décrites via des **blocs de règles**, des **concepts** structurés en **graphes sémantiques** ou en **logiques formelles**. Le calcul de la synergie $\mathbf{S}(i, j)$ se fonde alors sur des mesures de **compatibilité** entre axiomes, sur des **distances** dans une ontologie, ou sur des **critères** de cohérence dans un espace conceptuel. On obtient, par exemple,

$$\mathbf{S}(i, j) = \text{compat}(\mathcal{C}_i, \mathcal{C}_j),$$

où \mathcal{C}_i et \mathcal{C}_j désignent des entités logiques. Cette représentation **symbolique** jouit d'une **lisibilité** élevée, car chaque entité est énoncée sous forme de règles ou de propriétés aisément contrôlables. Néanmoins, elle peut se révéler **rigide** et peu adaptée au traitement de données massives ou bruitées, nécessitant une maintenance stricte de la **consistance** logique.

Dans de nombreuses applications, un **format hybride** apparaît plus naturel, chaque entité combine des composantes **sub-symboliques** et **symboliques**. Par exemple, un document textuel peut être associé à un **embedding** qui résume le contenu statistique global, tout en étant pourvu d'un **ensemble de mots-clés** ou de relations logiques qui en précisent la structure ou la sémantique. Dès lors, la fonction de synergie $\mathbf{S}(i, j)$ doit agréger à la fois une **mesure vectorielle** et un **score symbolique**. On peut imaginer une formule

$$\mathbf{S}(i, j) = \alpha \text{sim}(\mathbf{r}_{\text{subsym}}(i), \mathbf{r}_{\text{subsym}}(j)) + (1 - \alpha) \text{compat}(\mathcal{C}_i, \mathcal{C}_j),$$

la pondération $\alpha \in [0, 1]$ modulant l'importance relative du versant sub-symbolique et du versant symbolique. L'attrait de cette solution **hybride** réside dans sa souplesse. Elle allie la **richesse expressive** des approches symboliques à la **robustesse** et à la **continuité** des embeddings profonds.

Enfin, le DSL vise fréquemment le traitement de **données multimédias**, telles que des **images**, des **sons**, ou des **clips vidéo**. Ces entités peuvent elles-mêmes être considérées comme **sub-symboliques**, au sens où chaque support est converti en un vecteur (ou un tenseur) issu d'un réseau de neurones approprié (par exemple, un **spectrogramme** pour l'audio, une **carte de caractéristiques** pour l'image). Cependant, des **métadonnées symboliques** (catégories, tags, règles de provenance) peuvent en outre accompagner ces vecteurs. De la sorte, la **synergie** inclut un terme de similarité visuelle ou auditive, et, simultanément, un terme de compatibilité sémantique. On formule alors,

$$\mathbf{S}(i, j) = \text{sim}_{\text{media}}(\mathbf{m}_i, \mathbf{m}_j) + \text{sim}_{\text{sym}}(\mathbf{r}_{\text{sym}}(i), \mathbf{r}_{\text{sym}}(j)).$$

Cette fusion multimodale encourage l'**émergence** de clusters plus riches, dotés d'une cohérence simultanément perceptive et conceptuelle.

La **réalité** des usages implique donc que le **DSL** accueille une vaste **diversité** de formats, depuis la pure vectorisation jusqu'à l'**ontologie** la plus stricte, en passant par des **mélanges** sub-

symbolique et symbolique, et par des **données** clairement multimédias. Cette hétérogénéité, loin d'être un obstacle, constitue un **levier** où la synergie s'y nourrit d'interactions inattendues et des **clusterisations** inédites peuvent apparaître. La **condition** demeure néanmoins cruciale. Chaque entité doit être **décrise** de manière à dévoiler son essence ou sa fonction, faute de quoi la mesure $S(i,j)$ ne sera qu'un miroir déformant qui freinera l'**auto-organisation** synergique.

3.1.2. Objectifs du Chapitre

Dans ce chapitre, nous allons détailler la **représentation** et la **modélisation** des entités d'information, un aspect central pour la réussite du **Deep Synergy Learning** (DSL). Il s'agit de comprendre :

Les principes fondamentaux de la représentation dans le DSL :

- Pourquoi et comment décrire une entité \mathcal{E}_i ?
- Quels critères de qualité influent sur la synergie $S(i,j)$?

Les différentes formes de modélisation (vecteurs, graphes, ontologies, symboliques, hybrides) et leur **impact** sur la dynamique d'auto-organisation :

- Des vecteurs ou embeddings pour les données sub-symboliques,
- Des blocs de règles ou concepts pour les entités logiques,
- Des structures mixtes pour combiner à la fois la puissance statistique et l'interprétabilité.

Les concepts avancés (embeddings profonds, transformers, représentations symboliques complexes, synergie n-aire, etc.) permettant de **pousser plus loin** la capacité d'un DSL à découvrir des patterns riches et adaptatifs.

En d'autres termes, l'objectif est de **baliser** l'espace des représentations possibles, afin d'équiper le lecteur des connaissances nécessaires pour modeler correctement chaque type d'entité (\mathcal{E}_i), en tenant compte de la diversité (images, sons, textes, règles logiques) et de la flexibilité (adaptation en temps réel, extensions multiformes). Ce faisant, on jette les bases permettant au DSL de tirer pleinement parti de la notion de **synergie** pour organiser et valoriser ces données multiples.

3.1.2.1. Rôle Fondamental de la Représentation

Dans le **Deep Synergy Learning (DSL)**, la **représentation** de chaque entité \mathcal{E}_i constitue un **pivot** essentiel, car elle détermine la **qualité** de la **synergie** $S(i,j)$ et, partant, la **dynamique** de mise à jour des pondérations $\omega_{i,j}$. L'ensemble des mécanismes d'**auto-organisation**, qui aboutissent à la formation de **clusters** cohérents ou à l'émergence de **structures** plus complexes au sein du **Synergistic Connection Network (SCN)**, dépend donc étroitement de la façon dont chaque entité est encodée ou décrite.

Impact direct sur la synergie $S(i,j)$

La fonction $S(i,j)$ joue un rôle de **filtre** en évaluant dans quelle mesure deux entités \mathcal{E}_i et \mathcal{E}_j sont jugées **compatibles** ou **similaires**. La **définition** même de S varie selon la nature des entités, mais fait systématiquement intervenir leur **représentation**. Si celle-ci est sub-symbolique (vecteur d'**embedding**, par exemple), on peut recourir à une mesure de **similarité vectorielle** comme la cosinus ou la gaussienne :

$$S(i,j) = \exp(-\alpha \parallel \mathbf{r}(i) - \mathbf{r}(j) \parallel) \quad \text{ou} \quad \frac{\mathbf{r}(i) \cdot \mathbf{r}(j)}{\parallel \mathbf{r}(i) \parallel \parallel \mathbf{r}(j) \parallel}.$$

Si la représentation est symbolique, S peut reposer sur un **calcul** de cohérence logique, de distance ontologique ou de correspondance conceptuelle :

$$S(i,j) = f_{\text{logic}}(\mathbf{C}(i), \mathbf{C}(j)).$$

Dans tous les cas, la **pertinence** de $S(i,j)$ dépend de la **capacité** des descripteurs $\mathbf{r}(i)$ ou $\mathbf{C}(i)$ à rendre compte des attributs substantiels de l'entité \mathcal{E}_i . En d'autres termes, une **incomplétude** ou un **bruit** excessif dans ces descripteurs entraîne une **approximation** inadéquate de la synergie, ce qui peut fausser la dynamique globale du réseau.

Conséquences sur l'auto-organisation

Le **principe** d'auto-organisation dans le DSL est de laisser chaque lien $\omega_{i,j}(t)$ évoluer selon une **règle** inspirée d'une descente d'énergie, telle que :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)].$$

La convergence ou la stabilité de cette équation dépend d'une part de **l'information** captée par $S(i,j)$. Une bonne représentation garantit que la **force** $S(i,j)$ saura distinguer avec finesse les entités véritablement proches (renforcement des liens) de celles sans réelle corrélation (faible synergie et diminution progressive de $\omega_{i,j}$). À l'inverse, si $\mathbf{r}(i)$ ne contient pas les caractéristiques déterminantes, le réseau peut s'orienter vers des **clusters** peu informatifs ou s'enliser dans des **oscillations** inutiles.

Cette relation entre **représentation** et **dynamique** d'auto-organisation se révèle cruciale dans la mise en œuvre pratique du DSL. En effet, une représentation solide permet de poser un **pas d'apprentissage** η plus élevé, accélérant la consolidation des liens. Au contraire, une représentation frêle ou sujette au **bruit** impose un réglage prudent de η afin d'éviter des ajustements erratiques de $\omega_{i,j}$.

Représentation comme "passeport commun"

Le **DSL** se veut généraliste et vise à traiter des entités issues d'une **variété** de domaines (images, langage, signaux biomédicaux, règles logiques, etc.). Pour pouvoir **comparer** ces entités au sein d'un même **SCN**, il faut établir un **passeport** ou un **langage commun**, c'est-à-dire une représentation dans laquelle $S(i,j)$ devient calculable même si \mathcal{E}_i et \mathcal{E}_j proviennent de modalités différentes. Ainsi, un document texte peut être encodé par un **embedding** (BERT, GPT, etc.) tandis

qu'une image l'est par un CNN, et l'on projette ces descriptions dans un espace latent partagé permettant de définir, par exemple,

$$S(i,j) = \exp(-\|\Phi(\mathbf{r}_{\text{image}}(i)) - \Phi(\mathbf{r}_{\text{text}}(j))\|^2)$$

où Φ représente la fonction d'encodage multimodale. Cette approche assure la **cohérence** des liens de synergie, puisqu'on manipule des "vecteurs" homogènes malgré l'hétérogénéité initiale.

Robustesse et sensibilité au bruit

Une représentation de **qualité** se caractérise en outre par sa capacité à **absorber** le bruit des données et à **conserver** les informations essentielles pour le calcul de S . Un embedding textuel doit par exemple demeurer stable face à de légères variations lexicales, tandis qu'une représentation d'image doit tolérer des perturbations d'éclairage ou de perspective. Si l'on ne parvient pas à extraire ces invariances, l'estimation de la synergie est parasitée par des fluctuations indues, et la mise à jour de $\omega_{i,j}$ devient chaotique.

Il s'ensuit que la **stabilité** des descripteurs est un facteur-clé pour la réussite de l'auto-organisation. Dans un cadre "online", où les représentations peuvent elles-mêmes être affinées au cours du temps, les fluctuations trop rapides de $\mathbf{r}(i)$ risquent de semer la confusion dans les pondérations $\omega_{i,j}$, engendrant des **oscillations** durables ou une absence de convergence stable.

Dans les sections suivantes (3.2 à 3.5), on examinera plus en détail la manière dont on peut modéliser une entité \mathcal{E}_i en recourant à des approches **sub-symboliques** (vecteurs, embeddings neuronaux), **symboliques** (règles, ontologies, structures logiques) ou **hybrides** (combinaison de descripteurs neuronaux et de blocs de connaissances expertes). On illustrera également comment cette variété de représentations se traduit dans la fonction S , influençant l'ensemble de la dynamique synergique. Ainsi, le **DSL** se dote d'un cadre flexible où le **choix** de la représentation devient un levier majeur pour révéler des **synergies** pertinentes dans des univers de données de plus en plus hétérogènes.

3.1.2.2. Détails des Formes de Modélisation (Vecteurs, Graphes, Ontologies...) et Leur Impact sur la Synergie

Le **Deep Synergy Learning (DSL)** manipule des entités \mathcal{E}_i dont la **représentation** peut relever de plusieurs paradigmes. On peut décrire chaque entité par un **vecteur** (ou embedding), par un **graphe** interne, par un **ensemble de règles symboliques** (ou tout autre formalisme logique), ou encore adopter une **approche hybride** combinant ces perspectives. Dans chacun de ces cas, la **synergie** $S(i,j)$ — définie entre deux entités $\mathcal{E}_i, \mathcal{E}_j$ — se construit à partir d'un opérateur de **similarité** (ou de **compatibilité**) approprié. Le choix de la **forme** de modélisation influe ainsi directement sur la **nature** de $S(i,j)$, sur la **complexité** du calcul et sur la **qualité** de l'auto-organisation qui en découle.

A. Entités Vectorielles

Dans de nombreux cas, les entités se décrivent à l'aide d'**embeddings** ou de **vecteurs** $\mathbf{x}_i \in \mathbb{R}^d$. Cette approche est couramment employée pour encoder des images, des segments textuels, des extraits audio ou d'autres données sub-symboliques via un réseau neuronal ou une procédure d'extraction de caractéristiques. La fonction S peut alors prendre la forme d'une **distance** (euclidienne, Minkowski) ou d'une **similarité** (produit scalaire, cosinus) :

$$S_{\text{vectoriel}}(i, j) = \phi(\mathbf{x}_i, \mathbf{x}_j),$$

où ϕ est un opérateur de similarité comme

$$\phi(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}.$$

L'avantage principal réside dans la **simplicité** du calcul où la mise à jour des pondérations $\omega_{i,j}$ peut alors s'effectuer rapidement, favorisant une bonne **scalabilité** lorsque le **Synergistic Connection Network (SCN)** s'étend à un grand nombre d'entités. De plus, les représentations vectorielles, si elles sont correctement entraînées ou extraites, capturent des **propriétés** sémantiques ou perceptives pertinentes (ex. embeddings textuels pour la sémantique linguistique, features CNN pour l'analyse d'images).

En contrepartie, ces modèles vectoriels peuvent souffrir d'une **opacité** où ils sont moins interprétables et leur dimension peut croître, ce qui alourdit le traitement si le SCN doit parcourir un espace vectoriel de grande taille. La **mise en œuvre** de l'auto-organisation demeure néanmoins directe, puisqu'on effectue souvent un simple calcul de **similarité** pour chaque paire (i, j) .

B. Entités Graphiques

Une autre forme de description se fonde sur des **graphes** internes. Chaque entité \mathcal{E}_i est un graphe G_i , par exemple un graphe de relations (conceptuels, biologiques, topologiques) ou un graphe de dépendances. La **synergie** $S(i, j)$ s'exprime alors par un **score** de correspondance structurale, comme

$$S_{\text{graphique}}(i, j) = \Psi(G_i, G_j).$$

On rencontre notamment la **graph edit distance** (GED), mesurant le coût minimal de transformations (insertion/suppression de nœuds ou d'arêtes) pour passer de G_i à G_j . Une fonction de similarité exponentielle, de la forme

$$\exp(-\alpha \text{GED}(G_i, G_j)),$$

peut alors servir de **synergie**.

Cette approche est plus **riche** que les simples vecteurs où elle autorise une modélisation explicite de la structure interne, mais elle est plus **coûteuse** où la comparaison de graphes s'avère un problème algorithmique complexe, souvent NP-difficile pour l'isomorphisme de graphes généraux, impliquant ainsi un surcroît de calcul. Sur le plan de la **dynamique** du SCN, l'évaluation de

$S_{\text{graphique}}(i,j)$ peut s'avérer un **goulot d'étranglement** si le réseau compte de nombreuses entités structurées.

C. Entités Logiques ou Symboliques

Dans un univers **symbolique**, chaque entité \mathcal{E}_i peut être un ensemble de **règles**, un **théorème**, un fragment d'**ontologie** ou un **module logique** exprimant des propriétés formelles. Le calcul de $S(i,j)$ repose alors sur une **compatibilité** conceptuelle, par exemple un degré de “contradiction nulle” ou de “similarité sémantique” entre deux groupes d’axiomes :

$$S_{\text{symbolique}}(i,j) = \text{compat}\left(\{\text{règles}_i\}, \{\text{règles}_j\}\right).$$

On peut envisager une fonction de synergie mesurant la **proximité** de deux ontologies (recouvrement d’axiomes) ou la **cohérence** de l’union $\mathcal{E}_i \cup \mathcal{E}_j$. L’attrait principal réside dans l’**interprétabilité** où l’on sait **expliquer** pourquoi deux entités s’avèrent compatibles ou non et l’on peut procéder à des **déductions** explicites. Toutefois, le formalisme symbolique exige une mise à jour **plus rigide**, et demeure peu tolérant au **bruit** ou à l’incertitude.

D. Modélisation Hybride et Multimodale

Dans de nombreux contextes, les entités \mathcal{E}_i se décrivent à la fois par des **caractéristiques subsymboliques** (e.g. embeddings de textes, d’images) et par des **attributs symboliques** (e.g. champs sémantiques, règles associées, métadonnées). Il s’agit alors d’une **modélisation hybride**, où la synergie combine plusieurs fonctions de similarité :

$$S_{\text{hybride}}(i,j) = \lambda \phi(\mathbf{x}_i, \mathbf{x}_j) + (1 - \lambda) f_{\text{logic}}(\{\text{axiomes}_i\}, \{\text{axiomes}_j\}),$$

avec $\lambda \in [0,1]$ un hyperparamètre de balance. Cette approche **hybride** rend possible l’exploitation simultanée d’**informations** quantitatives et qualitatives, donnant une vision plus complète de la relation entre entités. Dans un cadre multimodal (images, textes, sons, etc.), on peut adapter la fonction $\phi(\cdot, \cdot)$ à chaque modalité et agréger les scores partiels par pondération ou par un opérateur plus sophistiqué.

La **richesse** d’un tel modèle s’accompagne d’une **complexité** accrue où l’évaluation de la synergie nécessite de jongler entre plusieurs types de descripteurs et le réglage des poids λ peut s’avérer délicat. D’un point de vue opérationnel, le SCN doit gérer des structures de données plus variées, et la formation de **clusters** engage des entités aux caractéristiques hétérogènes.

Conséquences sur la Dynamique d’Auto-Organisation

Le **choix** d’une forme de modélisation détermine **directement** le **coût** et la **précision** de la synergie $S(i,j)$. Un calcul vectoriel est **rapide**, mais peut manquer de **transparence** ; un calcul symbolique ou graphique peut être sémantiquement **puissant** mais se révèle onéreux et plus fragile au bruit. Dans un système d’**auto-organisation**, où chaque lien $\omega_{i,j}$ est actualisé de manière itérative, la

complexité de $S(i, j)$ devient cruciale si le réseau comporte un grand nombre d'entités. Par ailleurs, la **stabilité** de la représentation, l'adéquation entre la mesure de similarité et la réalité sémantique, ainsi que la résistance au bruit déterminent la **qualité** de l'émergence de clusters ou de schémas cognitifs.

Dans la pratique, un **DSL** performant peut s'appuyer sur des vecteurs ou embeddings lorsque les données sont massives et qu'on souhaite une **scalabilité** haute, quitte à sacrifier l'interprétabilité. Il peut exploiter des structures logiques pour bénéficier d'une **lisibilité** conceptuelle, en acceptant un calcul de S plus lourd. Il peut aussi combiner des descripteurs, par exemple sub-symboliques et attributs symboliques, pour dégager des synergies multidimensionnelles, au prix d'un pipeline de calcul plus complexe.

Au sein du **Synergistic Connection Network**, cette souplesse dans la définition de $S(i, j)$ autorise l'**intégration** de données de sources multiples et la **découverte** de correspondances inattendues — moyennant une **ingénierie** soignée pour élaborer la modélisation la plus appropriée aux besoins applicatifs. Les chapitres ultérieurs (notamment ceux traitant de la **fusion multimodale** ou des **architectures SCN** avancées) approfondiront comment cette diversité se reflète dans les mécanismes concrets d'auto-organisation synergique.

3.1.2.3. Aborder des Concepts Avancés : Embeddings à Haut Niveau (Transformers), Représentations Symboliques Complexes, etc.

La **représentation** des entités dans un **DSL** (Deep Synergy Learning) n'est pas limitée à de simples vecteurs statiques ou à des graphes élémentaires où, pour décrire des objets ou des notions de plus en plus complexes, on peut recourir à des **embeddings contextuels** issus de **Transformers**, à des **ontologies** riches ou à des **structures** multi-niveaux voire fractales. Ces choix de modélisation, s'ils rendent la **synergie** plus riche et plus précise, posent également des défis en termes de **coût** algorithmique et de **gestion** de la dynamique. Les paragraphes qui suivent illustrent comment cette sophistication accroît le **pouvoir** du DSL au prix d'une **ingénierie** plus exigeante.

A. Embeddings à Haut Niveau (Transformers et Architectures Contextuelles)

Dans le cadre sub-symbolique, les entités \mathcal{E}_i sont souvent associées à un **embedding** $\mathbf{x}_i \in \mathbb{R}^d$. Auparavant, des modèles statiques comme Word2Vec ou GloVe généraient un vecteur unique pour chaque mot. Aujourd'hui, l'essor des **Transformers** (BERT, GPT, T5, Vision Transformers, etc.) permet d'obtenir des **embeddings contextuels**, où la représentation varie selon l'environnement local (contexte lexical, patchs voisins dans une image, etc.). Mathématiquement, pour une entité \mathcal{E}_i , on définit une fonction

$$\mathbf{v}_i: \mathbf{C}_i \mapsto \mathbb{R}^d,$$

où \mathbf{C}_i est le **contexte** (par exemple, les tokens entourant un mot, la position dans une séquence d'image, etc.). La **synergie** $S(i, j)$ peut alors prendre la forme d'une similarité vectorielle appliquée aux **embeddings** contextuels

$$S_{\text{transf}}(i, j) = \text{sim}(\mathbf{v}_i(\mathbf{C}_i), \mathbf{v}_j(\mathbf{C}_j)).$$

Cette contextualisation renforce la **précision** de la mesure de similarité où un même mot ou un même patch peut être représenté différemment suivant les indices locaux, évitant ainsi les ambiguïtés que l'on retrouve dans les embeddings purement statiques. En contrepartie, un tel modèle implique un **coût** de calcul plus important (multiple tête d'attention, layers de grande taille) et une **dynamique** plus complexe si le **contexte** \mathbf{C}_i se modifie continuellement durant l'apprentissage. Le **DSL** doit donc gérer des synergies potentiellement instables si les embeddings contextuels ne sont pas figés.

B. Représentations Symboliques Complexes (Ontologies Riches, Logiques Hiérarchiques)

Le raffinement ne se limite pas au sub-symbolique. Dans un contexte **symbolique**, les entités peuvent relever de **logiques descriptives**, d'**ontologies** complexes (OWL, RDF) ou de **théories** plus abouties. L'idée est alors de définir

$$S_{\text{symbolique}}(i, j) = \text{compat}(\mathcal{O}_i, \mathcal{O}_j),$$

où \mathcal{O}_i et \mathcal{O}_j désignent des sous-ensembles de l'ontologie ou des blocs logiques associées aux entités \mathcal{E}_i et \mathcal{E}_j . Le calcul de **compat** peut se baser sur le **recouvrement** de concepts, la **cohérence** (absence de contradiction), ou encore l'**intersection** sémantique. Cette approche a le **mérite** de l'**explicabilité** où l'on sait pourquoi deux entités sont jugées compatibles puisqu'elles partagent des axiomes, des types ou des relations. D'un point de vue opératoire, cependant, l'exécution peut nécessiter des **moteurs d'inférence** plus lourds, surtout si on emploie des logiques complexes ou des algorithmes de correspondance ontologique (ontology matching). Le DSL profite ici de la grande **lisibilité** des clusters émergents, mais la mise à jour $\omega_{i,j}$ s'expose à la rigidité du formalisme symbolique, moins tolérant au **bruit** et aux incertitudes.

C. Structures Multi-Niveau ou Fractales

Dans des applications **multi-échelle** (cf. chap. 6), chaque entité \mathcal{E}_i peut se décliner en **sous-niveaux** ou exhiber des propriétés fractales. Par exemple, on peut envisager un document organisé en chapitres, sections, paragraphes, phrases, ou un graphe subdivisé hiérarchiquement. Le calcul de la synergie $S(i, j)$ s'étend alors sur plusieurs **granularités**. Une formalisation schématique pourrait recourir à un opérateur d'intégration multi-résolution :

$$S_{\text{multi}}(i, j) = \int_0^L \text{Sim}(\mathcal{E}_i^{(\ell)}, \mathcal{E}_j^{(\ell)}) d\ell,$$

où ℓ indexe le niveau d'observation, entre 0 (global) et L (local très détaillé). Cette construction donne une **vision holistique** de la synergie, dans laquelle deux entités se révèlent proches si leurs structures s'apparentent à **tous** les niveaux ou si l'on détecte un fort **matching** dans plusieurs couches. Le **DSL** exploite alors une **auto-organisation** multiforme où si la synergie est forte à certains niveaux, les liens se renforcent, même si la vue globale n'est pas identique. Le prix à payer

résidé dans le **coût** de comparaison de multiples niveaux et dans la gestion de la cohérence à travers ces échelles.

D. Combiner Diverses Approches

Souvent, un système DSL se trouve confronté à des données hétérogènes mêlant représentations avancées sub-symboliques, structures symboliques complexes et granularités multiples. Une **approche hybride** consiste alors à définir

$$S_{\text{avancée}}(i, j) = \lambda_1 \text{sim}_{\text{contextuel}}(\mathbf{v}_i, \mathbf{v}_j) + \lambda_2 \text{compat}_{\text{symbol}}(\mathcal{O}_i, \mathcal{O}_j) + \lambda_3 \text{SimMulti}(\mathcal{E}_i, \mathcal{E}_j),$$

avec $\lambda_1 + \lambda_2 + \lambda_3 = 1$. Cette formule intègre la **similarité contextuelle** (Transformers), la **compatibilité symbolique** (ontologies) et la **cohérence multi-niveau** (hiérarchie interne). Un tel design favorise une **syncrétisation** de l'information, permettant au **SCN** de découvrir des **clusters** sophistiqués englobant plusieurs perspectives. Cependant, la charge algorithmique et la complexité de déploiement augmentent d'autant. De plus, l'**auto-organisation** doit gérer la potentielle **instabilité** due à la variation simultanée de différentes composantes, particulièrement si l'on adapte en continu les embeddings et l'ontologie.

3.1.3. Structure du Chapitre

Dans ce chapitre, nous allons progressivement bâtir une **vision complète** de la représentation et de la modélisation des entités d'information dans le DSL. Nous avons déjà rappelé (sections 3.1.1 et 3.1.2) les **enjeux** et les **objectifs** ; désormais, nous allons suivre une **progression logique** afin de couvrir toutes les approches possibles (sub-symboliques, symboliques, hybrides, etc.) et les approfondir par des illustrations concrètes.

3.1.3.1. Vue d'Ensemble : Principes, Représentations Sub-Symboliques, Symboliques, Hybrides, Aspects Avancés et Études de Cas

Ce chapitre se propose d'explorer, de manière structurée, la diversité des **représentations** qui peuvent être adoptées pour décrire les entités \mathcal{E}_i au sein d'un **Deep Synergy Learning (DSL)**, et d'en illustrer l'influence directe sur le **calcul de synergie** et la **dynamique adaptative**. Les principes généraux de la représentation, tels qu'ils s'appliquent au DSL, constituent un socle méthodologique qui sera mis à profit dans les chapitres ultérieurs portant sur l'**auto-organisation** (Chap. 4) et l'**architecture** du Synergistic Connection Network (Chap. 5).

L'idée centrale, rappelée dans ce chapitre, est que la **qualité** et la **nature** de la représentation conditionnent la **pertinence** de la fonction de synergie $S(i, j)$. Que les entités soient modélisées sous forme de **vecteurs** (apprentissage profond, embeddings neuronaux), de **structures symboliques** (règles logiques, graphes ontologiques) ou d'**architectures hybrides**, la façon dont on code l'information se répercute sur la manière dont le DSL identifie les relations entre entités, renforce ou affaiblit les pondérations $\omega_{i,j}$ et, en définitive, fait émerger des **clusters**. Les sections 3.2 à 3.7 proposent une trajectoire progressive où l'on part des **principes** généraux (3.2), avant

d'examiner les cas **sub-symboliques** (3.3) et **symboliques** (3.4), puis d'étudier les **représentations hybrides** (3.5). Enfin, on introduit certains **aspects avancés** (3.6) et l'on conclut (3.7) sur la synthèse de ces différentes approches.

Section 3.2 : Principes généraux de la représentation dans le DSL

On y clarifie le rôle clé que joue la représentation des entités dans l'élaboration de la **synergie** $S(i, j)$. Après un rappel sur les exigences de robustesse, d'expressivité et de modularité, on souligne en quoi la **cohérence** entre la forme de représentation et la dynamique d'**auto-organisation** est décisive pour éviter des clusterisations parasites ou des oscillations improductives. Les arguments développés forment un **cadre théorique** dans lequel s'inscriront toutes les variantes ou extensions présentées par la suite.

Section 3.3 : Représentations sub-symboliques

Cette partie met l'accent sur les vecteurs et embeddings neuronaux, abordant notamment les mécanismes usuels de **similarité vectorielle** (distance euclidienne, cosinus, noyaux gaussiens). On met en évidence la facilité avec laquelle s'opère le **calcul** de synergie dans un espace vectoriel, contribuant ainsi à la **scalabilité** du DSL quand le nombre d'entités est important. Les vecteurs issus de modèles de deep learning (CNN, Transformers, autoencodeurs) illustrent l'**efficacité** et la **souplesse** des représentations sub-symboliques, tout en rappelant les limites liées à une **opacité** conceptuelle ou à une sensibilité aux biais de l'entraînement supervisé.

Section 3.4 : Représentations symboliques

Dans un second temps, on passe en revue les **entités** décrites au moyen de **structures logiques** ou **d'ontologies**. Cette approche présente l'intérêt d'une **interprétabilité** et d'une **inférence** plus directe où l'on peut déterminer si deux entités partagent des axiomes et si leurs règles s'avèrent compatibles ou contradictoires. Le **calcul** de $S(i, j)$ peut alors refléter la proportion d'axiomes communs, la distance entre classes ontologiques, ou la cohérence sémantique de leur union. Ce formalisme logique, souvent plus **rigide**, nécessite cependant des algorithmes de vérification plus lourds, et se montre peu tolérant au **bruit**. Les implications sur la dynamique du SCN sont discutées, soulignant la nécessité de contrôler la croissance de la complexité dans un réseau de grande taille.

Section 3.5 : Représentations hybrides

Une synthèse s'impose lorsque l'on veut combiner les atouts des vecteurs sub-symboliques (robustesse, généralisation) avec la lisibilité des structures symboliques. Les **modèles hybrides** associent ainsi, pour chaque entité \mathcal{E}_i , un embedding \mathbf{v}_i et des règles \mathcal{R}_i . La synergie se conçoit alors comme une combinaison linéaire ou non linéaire de deux ou plusieurs mesures où l'une extrait la **similarité vectorielle** et l'autre évalue la **compatibilité** symbolique. Le paramètre λ dans

$$S_{\text{hybride}}(i, j) = \lambda \rho(\mathbf{v}_i, \mathbf{v}_j) + (1 - \lambda) \text{Compat}(\mathcal{R}_i, \mathcal{R}_j)$$

illustre la souplesse de ce modèle, capable de s'adapter à des applications où la **sémantique** coexiste avec des propriétés statistiques ou perceptives. Les avantages d'une telle approche sont contrebalancés par la gestion plus complexe du calcul de **synergie**, qui doit agréger plusieurs points de vue.

Section 3.6 : Aspects avancés et études de cas

Cette section présente des **concepts** ou **développements** supplémentaires visant à accroître encore la diversité ou la sophistication des représentations. On évoque notamment les **synergies n-aires**, permettant de modéliser les interactions collectives entre plus de deux entités, ainsi que l'extension vers des **structures fractales** et **hypergraphes**. De plus, on y propose des **études de cas** concrets où des systèmes de vision sémantique enrichis par des ontologies, des applications multimodales combinant audio, vidéo et textes, ou encore la robotique cognitive où chaque entité correspond à un composant sensoriel ou à un module de planification, illustrent la diversité des approches. Ces exemples soulignent la flexibilité du DSL, mais mettent en relief la nécessité d'une **ingénierie** minutieuse pour garantir la **cohérence** et la **performance** du calcul de synergie.

Section 3.7 : Conclusion

Un dernier moment est consacré à **résumer** les points saillants de ce chapitre, tout en traçant des **passerelles** avec les chapitres suivants. On rappelle qu'une **représentation** de haute qualité (qu'elle soit vectorielle, symbolique ou hybride) conditionne la **dynamique** d'auto-organisation du DSL, à la fois dans la **descente d'énergie** qui stabilise les pondérations $\omega_{i,j}$ et dans la **formation** de clusters pertinents. Les choix retenus ici déterminent en grande partie l'efficacité, la robustesse et l'interprétabilité du **Synergistic Connection Network** dans les scénarios réels, un enjeu qu'illustreront les déploiements concrets abordés ultérieurement.

3.1.3.2. Renvoi aux Chapitres Ultérieurs

Le présent chapitre, consacré aux principes et aux méthodes de **représentation** des entités dans le **Deep Synergy Learning (DSL)**, s'inscrit dans un continuum dont les chapitres suivants approfondiront la dimension dynamique, architecturale et opérationnelle. La manière dont on encode une entité \mathcal{E}_i et dont on calcule la synergie $S(i,j)$ aura un impact direct sur les mécanismes décrits dans ces parties ultérieures. Le fait de bien saisir les choix de représentation vecteurs, règles symboliques ou structures hybrides, et de comprendre leurs avantages et inconvénients, constitue un prérequis essentiel à l'étude des **dynamiques d'auto-organisation**, de l'**architecture** du Synergistic Connection Network et de son déploiement multi-échelle. Les liens les plus directs avec les chapitres suivants s'établissent ainsi.

Dans le **Chapitre 4**, qui détaille la **dynamique d'auto-organisation**, les pondérations $\omega_{i,j}$ évoluent selon une équation itérative qui repose fortement sur la synergie $S(i,j)$. Si cette synergie est déterminée par des embeddings sub-symboliques, le calcul de similarité sera rapide, favorisant l'implémentation de mécanismes tels que la descente d'énergie ou la stabilisation de clusters. Inversement, si la représentation est symbolique, la règle de mise à jour devra tenir compte de la compatibilité logique, ce qui influence la façon dont les clusters se forment ou se fragmentent. Les notions de robustesse de la représentation, de granularité ou de sensibilité au bruit abordées dans ce chapitre alimenteront la réflexion sur les oscillations, les convergences locales, ou l'émergence de configurations stables que le Chapitre 4 explicitera.

Le **Chapitre 5**, quant à lui, traite de l'**architecture générale** du Synergistic Connection Network. Là encore, les choix de représentation sont déterminants puisque la conception même du réseau – qu'il s'agisse de la gestion de larges embeddings vectoriels, de graphes conceptuels ou de règles

logiques – impose des contraintes sur la structure interne du SCN, les protocoles de communication entre nœuds et la répartition des calculs. Ainsi, le fonctionnement d'un SCN manipulant des structures symboliques complexes doit intégrer un module de compatibilité ou d'inférence, tandis qu'un SCN vectoriel peut s'appuyer sur des algorithmes de similarité plus classiques. Ces aspects d'implémentation et de mise à l'échelle sont traités au Chapitre 5, mais prennent appui sur les fondements de la représentation établis ici.

Dans le **Chapitre 6**, consacré à l'**apprentissage synergique multi-échelle**, les représentations décrites dans le présent chapitre se verront transposées à des scénarios où la notion d'échelle ou de hiérarchie devient fondamentale. La façon dont une entité est encodée doit permettre de naviguer entre un niveau micro (par exemple, des attributs locaux d'un signal, des sous-parties d'un graphe) et un niveau macro (des clusters de haut niveau, des super-concepts). Les principes de modularité et de robustesse abordés ici guideront la construction d'architectures capables de traiter simultanément ces différentes granularités, et influenceront le mode de calcul de la synergie lorsqu'il s'agit de regrouper des "super-entités" ou d'organiser plusieurs couches de représentation.

Le **Chapitre 7**, traitant des **algorithmes d'optimisation** et des **méthodes d'adaptation dynamique**, s'appuiera également sur les fondements énoncés dans le présent chapitre. Les stratégies de recuit simulé, de heuristiques génétiques ou d'optimisation distribuée dans un DSL se conçoivent plus facilement lorsque la fonction de synergie est aisément manipulable et la représentation stable. Les embeddings sub-symboliques se prêtent par exemple à des manœuvres de projection ou de sparsification utiles pour accélérer les routines d'optimisation, alors que des formalismes symboliques imposent des approches plus subtiles pour maintenir la cohérence logique ou résoudre les contradictions.

Le **Chapitre 8**, qui se focalise sur un **DSL multimodal** intégrant par exemple la vision, le langage et les signaux auditifs, prolonge la question de la représentation dans un cadre où chaque modalité est codée différemment où les CNN sont utilisés pour les images, les embeddings BERT pour les textes et les paramètres acoustiques pour l'audio. Les notions de ce chapitre portant sur la fusion sub-symbolique et symbolique, ou sur la construction de synergies hybrides, trouvent leur application concrète dans la conception d'un SCN capable d'identifier des associations ou des regroupements cohérents à travers plusieurs modalités. La qualité de la représentation qu'on aura choisie pour chaque type de contenu détermine alors la richesse ou la précision de la synergie intermodale.

Enfin, dans d'autres chapitres ultérieurs (Chapitres 9, 10, 11, etc.) traitant de l'**évolution en temps réel**, des mécanismes de **feedback coopératif**, ou encore de la **robustesse** et de la **sécurité**, il apparaîtra qu'une bonne représentation n'est pas un simple détail où elle constitue un véritable moteur de résilience et de flexibilité. Les entités décrites de manière adéquate peuvent se réorganiser ou se mettre à jour plus aisément, et la logique de l'auto-organisation est plus lisible lorsque la synergie repose sur des signatures explicites ou bien définies. De surcroît, l'ajout d'un feedback ascendant ou descendant exige parfois la capacité de retourner à des représentations interprétables, qu'on ne peut dissocier des choix opérés dans le présent chapitre. Les mécanismes de détection d'anomalies ou de perturbations exploitent eux aussi la nature de l'entité où des vecteurs robustes au bruit ou des graphes logiques dotés de mécanismes de validation interne peuvent amortir les secousses et limiter les propagations d'erreurs.

Le **Chapitre 3** jette les bases d'un ensemble de choix et de contraintes qui façonnent la dynamique, l'architecture et la pérennité du Deep Synergy Learning. Les représentations sub-symboliques, symboliques ou hybrides et leurs déclinaisons avancées constitueront un fil rouge qui s'étire jusqu'aux démonstrations et applications pratiques des derniers chapitres, liant la théorie de la synergie et l'implémentation à grande échelle d'un SCN apte à manipuler des données hétérogènes, évolutives et à multiples échelles de granularité.

3.2. Principes Généraux de la Représentation dans le DSL

L'étape de **représentation** d'une entité \mathcal{E}_i est un maillon essentiel de la chaîne d'apprentissage dans le **DSL** (Deep Synergy Learning). De cette modélisation initiale dépend la qualité de la fonction de synergie $S(i, j)$ — et, par conséquent, la pertinence des clusters et de la dynamique d'auto-organisation $\omega_{i,j}$. Avant d'explorer les différents types de représentation (sections 3.3, 3.4, 3.5) et leurs implications (section 3.6), nous allons poser quelques **principes généraux** essentiels à comprendre.

3.2.1. Rôle Fondamental de la Représentation

À ce stade, nous avons déjà souligné (chap. 3.1) que la **description** de chaque entité influe directement sur la manière dont on évalue la synergie. Entrons maintenant dans le détail de ce rôle fondamental, en explicitant comment la représentation oriente tout le processus d'apprentissage.

3.2.1.1. Chaque entité \mathcal{E}_i doit être “capturable” par une structure (vecteur, ensemble de règles...)

Afin de garantir la cohérence globale du **Deep Synergy Learning (DSL)** et de son mécanisme d'**auto-organisation**, il est indispensable que chaque entité \mathcal{E}_i puisse être décrite au moyen d'une **structure** qui reflète ses attributs pertinents et permette de calculer la **synergie** $S(i, j)$. L'enjeu est de choisir ou de concevoir une représentation à la fois robuste et suffisamment expressive pour mettre en évidence les similitudes ou les compatibilités recherchées au sein du **Synergistic Connection Network (SCN)**. Plusieurs facteurs interviennent alors dans la définition de ce “format” :

Dans un premier temps, il s'agit de **saisir** les attributs importants de l'entité \mathcal{E}_i . Lorsque l'objet à décrire est une image, se limiter à une statistique globale comme la moyenne de luminosité ne suffit pas à capturer des singularités plus complexes (formes, couleurs, textures). Au contraire, un **embedding** généré par un réseau de neurones profonds (CNN, Vision Transformer, etc.) peut dégager des caractéristiques discriminantes indiquant la présence de motifs ou d'objets précis. Un raisonnement analogue s'applique aux textes où un simple bag-of-words se révèle souvent insuffisant pour rendre compte des relations contextuelles et des sémantiques sous-jacentes, tandis qu'un **embedding contextualisé** (type BERT ou GPT) prend en compte l'environnement lexical et la structure syntaxique. Dans le cas de données symboliques (règles, axiomes), il faut veiller à recourir à un formalisme (logique, ontologie, graphe conceptuel) permettant de décrire toutes les relations nécessaires, sans perte d'expressivité.

Il est également primordial de **faire correspondre** la forme de la représentation au **mode de calcul** de la fonction $S(i, j)$. Si la synergie exploite une mesure de similarité vectorielle, par exemple la distance cosinus ou un noyau RBF, alors chaque entité se voit associée à un vecteur \mathbf{x}_i dans un espace de dimension raisonnable. À l'inverse, si l'on s'appuie sur un critère de **compatibilité logique** (comme la proportion d'axiomes communs ou la cohérence de deux ensembles de règles),

il devient impératif d'exprimer \mathcal{E}_i sous une forme symbolique adéquate (règles logiques, graphes de connaissances, etc.), afin que les algorithmes de correspondance ou d'inférence puissent opérer. Ainsi, il doit y avoir une **cohérence** entre le type de structure choisi et l'**opérateur** de similarité ou de compatibilité mis en œuvre dans S .

Cette logique doit être prolongée dans les contextes **multimodaux** ou hétérogènes, où les entités se répartissent entre images, textes, flux audio, signaux bruts, formules logiques ou tout autre format. Il est alors vain de tenter d'imposer un unique schéma identique pour toutes les entités. Une image pourra s'encoder sous forme d'un embedding convolutif \mathbf{x}_i , tandis qu'une base de règles symboliques restera décrite par un ensemble \mathcal{R}_i . Toutefois, le **DSL** exige qu'il existe un moyen, le cas échéant, de calculer $S(i, j)$ entre entités de types différents. Cette nécessité peut conduire à prévoir un module de "passerelle" ou de traduction conceptuelle — par exemple, via des métadonnées partagées ou un espace latent commun — garantissant la **compatibilité** des calculs de synergie.

Enfin, il est important de **préserver l'évolutivité** de la représentation. Dans un apprentissage continu (cf. chap. 9), l'embedding d'une entité peut être affiné ou ses règles symboliques peuvent être mises à jour à mesure que de nouvelles informations parviennent au réseau. La structure choisie ne doit pas **verrouiller** l'entité dans un format immuable, sous peine de contraindre la plasticité du système. Au contraire, la capacité à réviser ou à enrichir la description de \mathcal{E}_i en cours de route constitue un point essentiel pour l'**adaptation** et la **réorganisation** dynamique.

Ainsi, l'exigence centrale peut se résumer en ces termes où toute entité \mathcal{E}_i doit être munie d'une **description** ou d'un **formalisme** qui capture les attributs ou les traits indispensables à la distinction et à la mise en relation, qu'il s'agisse d'images, de texte, de règles ou de graphes. Elle doit également s'aligner sur la manière de **définir** ou de **calculer** la synergie $S(i, j)$, gérer la potentielle hétérogénéité des types de données dans un même **SCN**, et permettre une **évolution** ou un **raffinement** si la situation ou les connaissances changent.

Lorsque ce cadre est respecté, la dynamique **auto-organisée** s'appuyant sur $\omega_{i,j}$ peut alors se déployer sans **biais** majeur. À l'inverse, si la représentation d'une partie des entités échoue à décrire les informations clés — ou se trouve en complet décalage par rapport au schéma de similarité attendu —, la **synergie** en sera faussée et la **structuration** en clusters ou en liens cohérents ne pourra émerger convenablement.

Comme on le verra dans la section suivante (3.2.1.2), le calcul de $S(i, j)$ lui-même s'appuie directement sur ces considérations, établissant un continuum entre la forme de l'entité, la fonction de similarité et la mise à jour adaptative du réseau ω . De plus, la section 3.2.2 présentera divers critères (pertinence, complexité, robustesse) qui aident à **opérer** les bons choix de format selon le domaine d'application et les contraintes logicielles.

3.2.1.2. La Fonction de Synergie $S(i, j)$ Dépend Directement de ces Représentations

Le **Deep Synergy Learning (DSL)** se caractérise par l'utilisation d'une **fonction de synergie** $S(i, j)$ permettant d'évaluer à quel point deux entités \mathcal{E}_i et \mathcal{E}_j présentent une **affinité** ou une **compatibilité** mutuelle. Avant même de décider si l'on doit renforcer ou affaiblir la pondération $\omega_{i,j}$, le réseau s'appuie sur les **descripteurs** qui modélisent chacune de ces entités. Ainsi, toute

variation dans la manière de représenter \mathcal{E}_i (ou \mathcal{E}_j) engendre un **changement** potentiel dans la valeur $S(i,j)$.

Formellement, on peut écrire :

$$S(i,j) = f(\mathbf{r}(i), \mathbf{r}(j)),$$

où $\mathbf{r}(i)$ désigne la représentation de l'entité \mathcal{E}_i (par exemple, un vecteur, un ensemble de règles symboliques, ou un graphe), et la fonction f dépend du **format** de ces représentations (similarité vectorielle, mesure de compatibilité logique, distance structurelle, etc.). Ce **schéma** induit que :

Pour des entités vectorielles

Si $\mathbf{r}(i) = \mathbf{x}_i$ et $\mathbf{r}(j) = \mathbf{x}_j$ sont des vecteurs en \mathbb{R}^d , le calcul de $S(i,j)$ repose souvent sur une **distance** (euclidienne, manhattan) ou une **similarité** (cosinus, gaussienne).

$$S_{\text{vect}}(i,j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad \text{ou} \quad S_{\text{vect}}(i,j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}.$$

Le **choix** et la **qualité** de l'embedding \mathbf{x}_i influent directement sur le résultat où modifier la dimension du vecteur ou la façon dont on l'a appris, qu'il s'agisse d'un modèle de deep learning, d'une PCA ou d'un autoencodeur, se traduit par une **variation** dans les valeurs de S .

Pour des entités symboliques

Si $\mathbf{r}(i) = R_i$ et $\mathbf{r}(j) = R_j$ représentent des **blocs** (ou des ensembles) de règles, d'axiomes ou de concepts, $S(i,j)$ peut alors refléter le **taux de recouvrement** (intersection d'axiomes), la **cohérence** s'il s'agit de les unir, ou toute autre **mesure** de compatibilité symbolique.

$$S_{\text{sym}}(i,j) = \text{Compat}(R_i, R_j).$$

Dans ce cas, l'ajout ou la suppression d'une règle dans R_i affecte le **score** final où cela peut drastiquement modifier la structure de l'auto-organisation, rendant deux entités auparavant jugées proches soudain incompatibles, ou inversement.

Représentations hybrides

Souvent, le DSL manipule des **descripteurs composites** alliant une partie vectorielle (embedding) et une partie symbolique (règles, concepts). On peut alors définir

$$S_{\text{hybride}}(i,j) = \alpha S_{\text{vect}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

où $\alpha \in [0,1]$ détermine l'importance relative du versant sub-symbolique ou symbolique. Toute **évolution** dans l'une ou l'autre des composantes (\mathbf{x}_i ou R_i) impacte la **valeur** de $S(i,j)$.

Cas n-aire ou hypergraphique

Un **DSL** avancé peut, par ailleurs, recourir à une synergie $S(i_1, \dots, i_k)$ dépendant simultanément de la représentation de k entités. Là encore, chaque entité $r(i_m)$ contribue à la **coopération** ou à la **conflictualité** du groupe.

On peut décrire la représentation par une application $g: \{\mathcal{E}_i\} \rightarrow \mathcal{X}$, où \mathcal{X} est l'espace ou la famille d'espaces dans lequel vivent les **descripteurs** (vecteurs, ensembles d'axiomes, graphes, etc.). La fonction S devient alors

$$S: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R},$$

ce qui formalise la dépendance : $S(g(\mathcal{E}_i), g(\mathcal{E}_j))$. Toute **modification** dans le contenu ou la forme de $g(\mathcal{E}_i)$ (c'est-à-dire la représentation de \mathcal{E}_i) répercute un **changement** dans $S(i, j)$. Ce phénomène se propage à la mise à jour des pondérations $\omega_{i,j}$ (voir Chap. 4), car la règle de type

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$$

se fonde sur $S(i, j)$ à chaque itération.

Dans l'esprit de l'**auto-organisation**, la structure du **SCN** (existence de clusters, de liens forts ou faibles) résulte alors de la manière dont S rend compte de la **proximité** ou de la **compatibilité** entre entités. Une **représentation** trop sommaire réduit la capacité de S à distinguer des différences subtiles, tandis qu'une représentation plus riche ou évolutive affine la détection des schémas latents et favorise la formation de clusters pertinents.

3.2.2. Critères de Choix

Après avoir mis en évidence le rôle fondamental de la représentation (section 3.2.1), il convient d'examiner **quels critères** président à la sélection d'une forme de description pour les entités \mathcal{E}_i . Ces critères incluent la **pertinence** de la représentation (dimension, nature sub-symbolique vs. symbolique, etc.), la **complexité** de calcul que cela implique, et la **capacité** d'évoluer ou de s'adapter au fil du temps. Nous allons passer en revue chacun de ces aspects.

3.2.2.1. Pertinence (Faut-il un Embedding Large ? un Formalisme Logique ?)

Le **premier** critère d'évaluation d'une représentation, dans le cadre du **Deep Synergy Learning (DSL)**, consiste à s'interroger sur sa **pertinence** par rapport aux entités visées et aux objectifs de la **synergie**. Un format de description n'est en effet pas intrinsèquement bon ou mauvais ; il faut l'analyser à l'aune du **contenu** réel de chaque entité, du **type** de synergie que l'on souhaite dévoiler, du **niveau** de granularité requis et de la **finalité** (raisonnement symbolique, clustering statistique, etc.). En pratique, plusieurs éléments entrent en jeu :

Alignement sur le contenu réel de l'entité

La **nature** même de l'entité \mathcal{E}_i dicte souvent la **forme** de représentation la plus adéquate. Lorsqu'il s'agit d'une **image**, un vecteur d'embedding dérivé d'un **réseau de neurones** (CNN, ViT) capture des attributs visuels déterminants (formes, textures, objets), alors qu'un simple histogramme d'intensité serait trop restreint pour distinguer des classes complexes. De même, pour un **texte**, un bag-of-words statique efface les relations contextuelles et ne parvient pas à rendre la sémantique profonde, tandis qu'un **embedding contextualisé** (BERT, GPT) intègre précisément ces nuances. Dans tous les cas, la représentation choisie doit couvrir les aspects saillants ou discriminants de l'entité, faute de quoi la **synergie** $S(i,j)$ perd en expressivité et l'auto-organisation en souffre.

Adéquation au type de synergie souhaité

Si la fonction $S(i,j)$ est conçue pour évaluer une **compatibilité logique** (absence de contradiction, mise en cohérence d'axiomes, etc.), un **formalisme symbolique** (règles, ontologie) s'impose. On y gagne en **interprétabilité** et en capacité de raisonnement explicite, par exemple pour vérifier si deux ensembles de règles peuvent coexister. À l'inverse, si l'on cherche à détecter des **similitudes** majoritairement basées sur des correspondances statistiques (images, textes bruités, signaux), un **embedding sub-symbolique** est plus approprié. Le calcul de $S(i,j)$ se fera alors via une mesure de distance ou de similarité vectorielle. Ainsi, la **logique de la synergie** (symbolique vs sub-symbolique) oriente la structure de la représentation, et réciproquement.

Niveau de granularité

Au sein d'une **représentation vectorielle**, la question de la **dimension** (d) se pose où un embedding de grande taille, par exemple 768 dimensions, peut mieux cerner des subtilités mais risque de générer de la redondance ou d'exacerber la sensibilité à la “malédiction de la dimension”. Une dimension plus réduite (type autoencodeur) rend la représentation plus stable et moins coûteuse, au risque de perdre des informations fines. Cette problématique se double d'une éventuelle **hiérarchisation** où l'on peut choisir un embedding global, avec une seule dimension du type “chat” ou “voiture”, ou opter pour des attributs plus nuancés. Un bon compromis s'efforce de maintenir la diversité descriptive tout en évitant la surcharge dimensionnelle qui nuirait au calcul de $\| \mathbf{x}_i - \mathbf{x}_j \|$.

Possibilité d'exploitation ultérieure

Selon la **finalité** visée, on peut privilégier un certain **formalisme**. Les représentations logiques offrent par exemple la **facilité** de mise en œuvre d'un raisonnement déductif, d'une détection de contradictions ou d'un enrichissement par de nouveaux axiomes. Un vecteur, en revanche, s'intègre mieux dans des **pipelines** de clustering ou de classification statistique, dispose d'un large écosystème d'algorithmes et se manipule aisément pour des calculs de distances ou de produits scalaires. La **composante** “exploitation future” est donc cruciale où il s'agit de déterminer si

l'objectif est de raisonner, annoter, expliquer ou si l'on privilégie avant tout une similarité fluide et scalable.

Nature des données

Au-delà du choix conceptuel, la **nature** des données elles-mêmes oriente fortement la **représentation**. Un flux **audiovisuel** ou **sensoriel** continu se prête naturellement à une description sub-symbolique (embedding, vecteur de caractéristiques), tandis qu'un ensemble de **règles expertes** s'incarne naturellement dans un formalisme d'ontologie ou de logique descriptive. Même dans des scénarios multimodaux, on veillera à ne pas forcer un encodage unique pour toutes les sources où il est préférable de conserver la spécificité des formats, au besoin en utilisant des modules passerelles (cf. Chap. 3.5 sur les représentations hybrides).

*Mesure indirecte : écart entre S et S^**

La **pertinence** se définit comme la capacité de la représentation (et de la fonction S) à **approximer** une notion “vraie” de similarité S^* . Il s’agit de minimiser la différence $|S^*(i, j) - S(i, j)|$ pour les couples (i, j) qu’on juge objectivement proches. Bien sûr, on ne dispose pas de S^* de manière explicite, mais cette considération sert de **guide**. Plus la représentation reflète les attributs réellement discriminants, plus $S(i, j)$ concorde avec le jugement “naturel” ou “souhaité”. À défaut, si la représentation omet ou brouille les caractéristiques clés, la fonction de synergie s’en voit faussée.

3.2.2.2. Complexité Computationnelle (Dimension du Vecteur, Structure Symbolique)

Au-delà du simple **critère de pertinence**, il est primordial de considérer le **coût** (en temps et en ressources) associé au **calcul** de la synergie $S(i, j)$. Chaque forme de représentation (embeddings vectoriels, logiques symboliques, etc.) induit une **charge** algorithmique différente, qui peut s'avérer déterminante pour l'évolutivité et la stabilité du **Deep Synergy Learning (DSL)**. Lorsque le nombre d'entités n et la taille (ou la complexité) de la représentation augmentent, une mauvaise anticipation de cette complexité peut rendre la mise à jour des pondérations $\omega_{i,j}$ irréalisable dans un temps raisonnable, voire conduire à des instabilités de la dynamique auto-organisée.

1. Dimension du vecteur et coût en $O(n^2 d)$

Lorsqu'on adopte des **représentations sub-symboliques** de type vectoriel, chaque entité \mathcal{E}_i est décrite par un vecteur $\mathbf{x}_i \in \mathbb{R}^d$. La plupart des calculs de synergie (ex. cosinus, gaussienne) entre deux vecteurs se font en $O(d)$, ce qui reste modeste si d est de taille modérée (e.g., 100–1000). Néanmoins, si le réseau manipule n entités, la comparaison **exhaustive** de toutes les paires (i, j) se chiffre en $O(n^2 d)$. Pour un large n (plusieurs millions) et un vecteur d élevé (512, 1024, etc.), ce volume de calcul peut devenir **prohibitif** :

$$S_{\text{vect}}(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}, \quad \text{coût en } O(d) \text{ par comparaison, } O(n^2 d) \text{ au total.}$$

Stratégies :

- **Approximate Nearest Neighbor** (ANN), qui évite de comparer toutes les paires en se concentrant sur les plus proches.
- **Projection de dimension** (PCA, autoencodeurs) pour réduire d .
- **Filtrage ou sparsification** où on n'évalue la synergie que pour des couples (i, j) pré-selectionnés (distants exclus).

2. Structure symbolique et complexité du matching

Pour les entités **symboliques** (règles logiques, axiomes, ontologies), la fonction $S(i, j)$ peut exiger des algorithmes de correspondance ou de vérification de cohérence, voire d'**unification** (en logique). De tels problèmes tombent parfois dans des classes de complexité élevées (NP-Complets, voire plus) :

$$\text{Compat}(R_i, R_j) \rightarrow \text{matching ou unification} \sim O(\exp(|R_i| + |R_j|)) \text{ (pire cas).}$$

Dans le cas de graphes symboliques ou d'ontologies volumineuses, la vérification d'isomorphisme, de sous-isomorphisme ou l'alignement d'ontologies peut aussi s'avérer $O(\exp(\cdot))$ en théorie. Cela se répercute sur le **calcul total** des synergies, qui peut alors freiner l'évolutivité du DSL :

Stratégies :

- **Restreindre** la logique (Horn clauses, sous-langage OWL décidable) pour limiter la complexité.
- **Indexation** partielle ou agrégation de règles dans des structures plus compactes.
- **Heuristiques** pour unifier ou comparer les entités sans tout explorer (matching local, découpage).

3. Synergie n-aire et explosion combinatoire

Une extension de la synergie binaire $S(i, j)$ vers une synergie n-aire $S(i_1, \dots, i_k)$ (voir chap. 12) fait passer le nombre de sous-ensembles à comparer de $O(n^2)$ vers $O(n^k)$. Quand $k > 2$, l'on risque une **explosion** combinatoire des évaluations, aggravée si chaque entité \mathcal{E}_i est décrite par un graphe symbolique volumineux. Cet **effet multiplicatif** rend vital l'introduction de mécanismes filtrant quels groupes d'entités vérifier, ou restreignant k à 2 ou 3.

4. Trade-off entre précision et coût

Plus la représentation est **détaillée** (forte dimension vectorielle, ensemble symbolique volumineux, structure fractale), plus la fonction S risque d'être précise et riche, mais au **prix** d'un coût de calcul

parfois considérable. En notation abstraite, si $C(\mathbf{r})$ désigne la **complexité** liée à la représentation \mathbf{r} , alors la comparaison de $\mathbf{r}(i)$ et $\mathbf{r}(j)$ s'évalue en $O(C(\mathbf{r}))$. Multipliée par $O(n^2)$ entités, on obtient une charge globale $O(n^2 \times C(\mathbf{r}))$. L'**ingénierie** du DSL doit donc trouver un **compromis** :

$$\min(C_{\text{calcul}}(S) + \alpha \text{Err}),$$

où C_{calcul} mesure le coût de l'évaluation de S et Err l'**erreur** ou la **perte de détails** associée à une représentation plus légère (dimension réduite, règles simplifiées, etc.).

5. Distribution et stockage

Enfin, dans un **SCN** distribué (Chap. 5.7), on doit aussi considérer la **taille** du stockage nécessaire et le **trafic** induit par la récupération des descripteurs $\mathbf{r}(i)$. Conserver un embedding de dimension 512 pour 10^6 entités représente déjà $O(512 \times 10^6)$ flottants, sans parler d'un matching symbolique éventuel sur des milliers de règles. On peut répartir ces données sur plusieurs noeuds, mais il demeure crucial de **minimiser** la quantité d'informations échangées lors du calcul de synergie.

3.2.2.3. Évolutivité (Peut-on Ajouter des Attributs ?)

Un troisième critère majeur dans le choix de la **représentation** pour un système de **Deep Synergy Learning (DSL)** réside dans la **capacité** à faire **évoluer** cette description au fil du temps ou à y **intégrer** de nouveaux attributs. Dans des contextes d'**apprentissage continu**, de **données changeantes** ou de **mise à jour** de règles symboliques, il est fréquent que la représentation d'une entité \mathcal{E}_i doive être modifiée, soit pour ajouter un attribut (une nouvelle coordonnée vectorielle, un nouveau champ sémantique), soit pour réviser les informations déjà présentes. Du point de vue formel, cette **évolutivité** implique la possibilité de mettre à jour la fonction $\mathbf{r}(i)$ en cours de route, sans pour autant briser la dynamique d'auto-organisation basée sur la synergie $S(i, j)$. Différents aspects viennent éclairer ce besoin :

A. Évolutivité dans l'Espace Vectoriel

Lorsqu'on recourt à une **représentation sub-symbolique** de type vectoriel, chaque entité \mathcal{E}_i est associée à un vecteur $\mathbf{x}_i \in \mathbb{R}^d$. Il peut survenir qu'une nouvelle **dimension** doive être incorporée, reflétant un attribut nouvellement découvert. Dans ce cas, on passe d'un embedding $\mathbf{x}_i(t) \in \mathbb{R}^d$ à un embedding $\mathbf{x}_i(t+1) \in \mathbb{R}^{d+1}$. Toutefois, l'ajout d'une coordonnée aux vecteurs de certaines entités, et pas à d'autres, risque de **déstabiliser** la fonction de similarité :

$$\|\mathbf{x}_i(t+1) - \mathbf{x}_j(t+1)\| \quad \text{devient ambigu si la dimension change.}$$

La solution la plus straightforward consiste à **initialiser** la nouvelle dimension (attribuée) à une valeur neutre (0, par exemple) pour toutes les entités concernées. Ainsi, l'ensemble des vecteurs se retrouvent à la même dimension $d+1$. Par la suite, on laisse l'**auto-organisation** adapter ces composantes — par apprentissage supervisé ou non, ou par un mécanisme de recalibration — de

sorte que la synergie puisse à nouveau être calculée de façon cohérente. Le point crucial est de **maintenir** un alignement des vecteurs où si certains sont encore en \mathbb{R}^d et d'autres en \mathbb{R}^{d+1} , la dynamique de mise à jour $\omega_{i,j}$ risque de se muer en une source de divergences.

B. Évolutivité dans une Structure Symbolique ou une Ontologie

Dans le cas d'une représentation **symbolique**, où l'entité \mathcal{E}_i se décrit par un **ensemble** de règles, d'axiomes ou de concepts R_i , on peut être amené à **ajouter** de nouvelles règles ΔR pour refléter un enrichissement de la connaissance. Il se produit alors une évolution :

$$R_i(t+1) = R_i(t) \cup \Delta R,$$

qu'il s'agisse de lois supplémentaires (ex. "si fièvre et éruption cutanée alors maladie Z") ou d'une mise à jour dans une ontologie (nouveau concept, nouvelle relation). Cette **extension** de la description modifie la fonction de compatibilité symbolique. Deux entités antérieurement divergentes peuvent désormais partager un axiome, accroissant $S(i,j)$, ou au contraire révéler une contradiction.

Le **défi** computationnel tient à la nécessité de **recalculer** (ou de mettre à jour) S_{sym} pour toutes les paires (i,j) . Si ΔR est modeste, on peut envisager un algorithme incrémental, au lieu d'une recombinaison exhaustive. Le **DSL** doit s'assurer que la complexité — déjà évoquée en section 3.2.2.2 pour les représentations symboliques — reste gérable, sans bloquer la dynamique globale.

C. Fusion ou Scission d'Entités

Un cas plus particulier concerne la **fusion** de deux entités \mathcal{E}_i et \mathcal{E}_j en une entité unique \mathcal{E}_k , ou la **scission** inverse. Du point de vue de la représentation :

- La **fusion** consiste à **combiner** $\mathbf{r}(i)$ et $\mathbf{r}(j)$ en une unique description $\mathbf{r}(k)$. Dans un cadre vectoriel, on peut, par exemple, prendre un mélange $\alpha \mathbf{x}_i + (1 - \alpha) \mathbf{x}_j$. Pour des règles symboliques, on effectue l'union $R_i \cup R_j$.
- La **scission** divise $\mathbf{r}(k)$ en deux descriptions $\mathbf{r}(k_a)$ et $\mathbf{r}(k_b)$ où un vecteur peut être partitionné ou un ensemble de règles réparti en deux sous-ensembles.

Cette adaptation modifie la **composition** même du **SCN** (des entités disparaissent, d'autres apparaissent), et entraîne une **réévaluation** de la synergie vis-à-vis de toutes les entités. Le système doit alors gérer ce chamboulement structurel (réassignation de liens $\omega_{i,j}$, recalcul partiel de S), et continuer son auto-organisation sans repartir entièrement de zéro.

D. Mise à Jour Incrémentale dans le Temps

De façon générale, un **DSL** opère souvent en **flux** (voir chap. 9), où de nouvelles données ou observations parviennent en continu. La représentation de l'entité \mathcal{E}_i peut alors se réviser itérativement :

$$\mathbf{r}(i, t + 1) = \mathbf{r}(i, t) + \Delta \mathbf{r}_i(t),$$

et chaque entité incorpore ainsi de l'information fraîche (données supplémentaires, règles découvertes, nouveaux attributs). La fonction $S(i, j)$ se met à jour en conséquence, puisqu'un changement dans $\mathbf{r}(i)$ implique forcément un recalcul potentiel de la synergie. La **plasticité** du SCN (pondérations $\omega_{i,j}$) réagit immédiatement à ces modifications où, par exemple, si une amélioration de l'embedding rapproche $\mathbf{r}(i)$ de $\mathbf{r}(j)$, la pondération $\omega_{i,j}$ pourra se renforcer.

E. Considérations de Cohérence

Il importe de veiller à la **cohérence** du système lorsqu'on introduit ou modifie la représentation. Si l'on ajoute une composante dans un vecteur, il faut que toutes les entités possèdent cette nouvelle dimension, au moins à titre de placeholder (valeur 0). De même, dans un formalisme logique, un concept n'a de sens que si toutes les entités susceptibles de l'utiliser le reconnaissent en tant qu'attribut ou axiome possible. Faute de synchronisation dans l'évolution du format, la fonction $S(i, j)$ peut devenir impraticable pour certaines paires.

Un protocole **échelonné** ou **progressif** peut être imaginé où l'on introduit la nouvelle dimension ou règle dans un sous-ensemble d'entités tout en garantissant un mécanisme de fallback, par exemple une coordonnée 0 ou un axiome neutre, afin d'éviter la perte de comparabilité. L'idée demeure d'éviter un changement brutal qui perturbe la dynamique auto-organisée ou invalide des synergies calculées précédemment.

3.2.3. Impact sur la Synergie et la Dynamique

Après avoir discuté des **critères de choix** (pertinence, complexité, évolutivité) dans la section 3.2.2, il est crucial de saisir **comment** la représentation affecte non seulement la **valeur** de la synergie $S(i, j)$, mais aussi la **dynamique** globale du DSL (mise à jour des pondérations $\omega_{i,j}$, formation de clusters, etc.). Le **rôle** de la représentation est ici déterminant où, en étant plus ou moins proche entre deux entités, on module directement la **probabilité** qu'elles se renforcent mutuellement dans le réseau.

3.2.3.1. Représentation “proche” \Rightarrow Synergie Potentiellement Élevée, $\omega_{i,j}$ Renforcée

L'un des traits marquants du **Deep Synergy Learning (DSL)** réside dans la façon dont la **proximité** ou la **similarité** entre deux entités \mathcal{E}_i et \mathcal{E}_j (telle qu'elle est traduite par leur **représentation**) induit mécaniquement une **augmentation** de la pondération $\omega_{i,j}$. Cet effet peut se voir directement dans la règle de mise à jour inspirée d'une descente d'énergie, où la **synergie** $S(i, j)$ agit comme un **signal de renforcement**. Plus la représentation montre une concordance marquée, plus la **liaison** entre ces entités se consolide, favorisant l'émergence de **clusters cohérents**.

Exemple dans le cadre sub-symbolique

Lorsque la représentation de chaque entité \mathcal{E}_i est un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$, la synergie $S(i, j)$ peut se définir via une **mesure** de similarité vectorielle (cosinus, distance euclidienne inversée, etc.). Un choix courant est la **similarité cosinus** :

$$S(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}, \quad \text{valeur dans } [-1, 1].$$

Si \mathbf{x}_i et \mathbf{x}_j sont presque colinéaires, c'est-à-dire $\mathbf{x}_i \cdot \mathbf{x}_j \approx \|\mathbf{x}_i\| \|\mathbf{x}_j\|$, alors la valeur de $S(i, j)$ s'approche de 1, traduisant une **forte similarité**. Dans la mise à jour adaptative :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

la présence d'un $S(i, j)$ élevé engendre un **terme positif** $\eta S(i, j)$, dépassant la "pénalisation" $\eta \tau \omega_{i,j}(t)$ si la synergie est assez grande. La liaison $\omega_{i,j}$ se voit donc **renforcée** et tend vers un équilibre local

$$\omega_{i,j}^* \approx \frac{S(i, j)}{\tau}.$$

Deux entités largement **proches** (au sens vectoriel) se verront ainsi attribuer une pondération stable et élevée. D'un point de vue **clustering**, elles se retrouveront **attirées** l'une vers l'autre, formant ou rejoignant un **sous-groupe** à plus large échelle dans le Synergistic Connection Network.

Exemple dans le cadre symbolique

La même idée vaut si les entités \mathcal{E}_i et \mathcal{E}_j sont décrites par des **blocs** de règles logiques ou des **concept**s dans une ontologie (R_i et R_j). La fonction de synergie S_{sym} évalue la **compatibilité** ou la **cohérence** entre ces deux ensembles :

$$S_{\text{sym}}(i, j) = \text{Compat}(R_i, R_j),$$

qui peut mesurer la proportion d'axiomes ou de règles partagées, ou l'absence de contradictions sémantiques. Plus cette **intersection** conceptuelle est forte, plus la synergie atteint des valeurs élevées, incitant la pondération $\omega_{i,j}$ à croître. Là encore, on assiste à la **formation** progressive d'un cluster où des entités possédant des blocs de règles similaires ou présentant une forte compatibilité se verront reliées par des liaisons renforcées, favorisant leur regroupement.

Conséquence : du renforcement local vers le cluster global

De manière générale, la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)]$$

assure que toute **paires** (i, j) bénéficiant d'une **synergie** élevée (qu'elle soit sub-symbolique ou symbolique) subisse un **renforcement** récurrent, tant que $S(i, j)$ demeure au-dessus d'un certain

point de compensation $\tau \omega_{i,j}$. Dans un environnement multi-entités, de **nombreuses** paires peuvent progressivement s'attirer, formant un **noyau** de liens forts. En se propageant, cet effet aboutit à la **coalescence** d'un ensemble d'entités partageant des attributs similaires ou des règles logiques compatibles, c'est-à-dire un **cluster** ou une **communauté** nettement identifiable dans le **SCN**.

Seuil et Trigger de Liaison

Il n'est pas rare de définir, de manière explicite ou implicite, un **seuil** θ pour la synergie $S(i,j)$. Lorsqu'elle dépasse θ , la croissance de $\omega_{i,j}$ s'accélère. Cette heuristique peut être formalisée, par exemple, en introduisant un **terme** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [F(S(i,j), \theta) - \tau \omega_{i,j}(t)],$$

où la fonction $F(S(i,j), \theta)$ vaut 0 en dessous du seuil θ , et augmente rapidement au-delà. Cette approche accentue le **contraste** où des entités proches franchissent le seuil et se relient fortement, tandis que d'autres, trop différentes, ne voient pas leur liaison progresser. La réussite de ce **filtrage** dépend à nouveau de la **qualité** de la représentation initiale où si celle-ci n'est pas assez discriminante, ce système de seuil peut mener à des associations erronées ou, au contraire, trop restreintes.

3.2.3.2. Risque de Confusion si la Représentation est Trop Floue ou Trop Simplifiée

En miroir de l'idée qu'une **proximité** marquée dans l'espace de représentation favorise une **synergie** élevée — laquelle renforce la pondération $\omega_{i,j}$ —, il faut souligner la possibilité d'un **effet inversé** si la représentation n'est pas assez **discriminante**. En d'autres termes, quand la description d'une entité \mathcal{E}_i est trop simplifiée, peu informative ou compressée, deux entités **objectivement** différentes risquent d'apparaître artificiellement "proches", gonflant leur synergie $S(i,j)$ sans réel fondement. Cette confusion conduit alors le **Synergistic Connection Network (SCN)** à les regrouper dans un même cluster de manière erronée.

Représentation sub-symbolique et dimensionnalité insuffisante

Un cas courant se rencontre dans les approches **sub-symboliques** où l'on associe à chaque entité \mathcal{E}_i un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$. Si la **dimension** d est trop faible pour couvrir la diversité des attributs, ou si l'on a appliqué une **réduction de dimension** trop agressive, par exemple une PCA drastique ou un autoencodeur trop compact, des entités pourtant distinctes finissent par se projeter **proches** dans \mathbb{R}^d . Mathématiquement, la distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ ou la similarité cosinus $\frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$ donne alors une valeur trompeusement élevée. La synergie $S(i,j)$ en résulte **surévaluée** et le réseau en vient à renforcer $\omega_{i,j}$:

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)].$$

Ces rapprochements “factices” introduisent une confusion dans la **dynamique d’auto-organisation** où, au lieu de séparer clairement des entités distinctes, le SCN tend à créer des liens forts entre elles, formant ainsi un **cluster** incohérent. On peut aboutir à des macro-groupes qui englobent quantité d’entités hétérogènes, ou à des groupements aléatoires privés de sens réel.

Représentation symbolique simpliste

Dans un **cadre symbolique**, un problème similaire survient si l’on ne conserve qu’un **jeu restreint** de règles ou d’axiomes pour chaque entité. Par exemple, la fonction de compatibilité $\text{Compat}(R_i, R_j)$ peut reposer sur la part d’axiomes communs entre deux ensembles R_i et R_j . Si ces ensembles se réduisent à quelques **règles** très générales (ex. “tout objet peut être rouge ou bleu”), alors de multiples entités paraîtront artificiellement “compatibles”, faute de disposer de règles plus spécialisées pour les distinguer. La synergie $S(i, j)$ demeure élevée pour bon nombre de couples, masquant les divergences réelles que des règles plus riches auraient mis en évidence.

Cette situation entraîne le même effet que dans l’espace vectoriel compressé où des **liaisons** $\omega_{i,j}$ s’en trouvent gonflées de façon injustifiée, menant à un **cluster unique** ou à des regroupements inadéquats. Dans le cas d’une ontologie appauvrie, deux concepts très différents peuvent apparaître près l’un de l’autre en raison d’un socle minimal d’axiomes partagés.

Distorsion de la synergie

D’un point de vue **mathématique**, on peut considérer qu’il existe une “**vraie**” similarité $S^*(i, j)$ que l’on souhaiterait approximer par $S(i, j)$. Une représentation **trop floue** augmente l’écart $|S^*(i, j) - S(i, j)|$. Cela se traduit par deux risques :

- **Surévaluation** : Deux entités très différentes ($S^* \approx 0$) apparaissent “proches” en S , conduisant à un renforcement indu de $\omega_{i,j}$ et à la formation de liens illusoires.
- **Sous-évaluation** : Inversement, des entités en réalité similaires ($S^* \approx 1$) reçoivent un score S faible, se retrouvant dispersées dans des clusters distincts.

Dans les deux cas, la **dynamique** du DSL dérive et forme des structures contraires à la **réalité** des entités (au sens de S^*). Le réseau se **désorganise** ou, pire, se retrouve dans un état quasi homogène où presque tout le monde est “lié” de façon grossière.

Clusters peu informatifs ou pseudo-homogénéité

Concrètement, une représentation simpliste ou appauvrie donne souvent lieu à des **clusters** anormalement étendus, mélangeant des entités sans vrai rapport. Sur le plan algorithmique, la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)]$$

va favoriser des liaisons multiples, lesquelles finissent par “coller” ensemble un grand sous-ensemble. Lorsqu’on visualise la matrice $\{\omega_{i,j}\}$, elle tend à se **remplir** (saturer), aboutissant à une quasi-absence de différenciation.

Comment éviter ce problème ?

D’un point de vue **ingénierie**, il convient :

- D’adopter une **représentation** plus riche (augmenter la dimension vectorielle, introduire davantage d’attributs symboliques).
- De vérifier que les **attributs** sélectionnés sont véritablement discriminants (ne pas se contenter d’une statistique globale).
- D’envisager des **mécanismes** de filtrage ou d’inhibition (cf. chap. 4 et 7) pour ne pas laisser les synergies de faible consistance perturber la structure.
- Dans certains cas, de recourir à des **tests** externes ou à un oracle pour détecter les couples (i,j) qui seraient anormalement jugés “proches”, et rétroagir sur la représentation.

3.2.3.3. Rôle de la Normalisation ou de la Calibration des Entités

Lorsque l’on conçoit un **Deep Synergy Learning (DSL)**, il ne suffit pas de sélectionner une représentation (vecteur, règles symboliques, etc.) pour chaque entité \mathcal{E}_i . Il est également crucial de **rendre** ces différentes descriptions **comparables**, de telle sorte que la fonction de synergie $S(i,j)$ ne soit pas faussée par des échelles numériques trop divergentes ou par des attributs symboliques surpondérés. C’est ici qu’interviennent la **normalisation** et la **calibration**, qui visent à **harmoniser** la magnitude ou la pondération des descripteurs et à éviter une domination injustifiée de certaines composantes.

A. Problématique de l’Échelle (Cas Sub-Symbolique)

Lorsqu’une entité \mathcal{E}_i est décrite par un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$, on peut rencontrer le problème de **dimensions** inégales où certaines coordonnées varient dans un intervalle restreint, par exemple entre 0 et 1, tandis que d’autres couvrent un éventail bien plus large, comme de 100 à 10 000. Pour la distance euclidienne,

$$\|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{\ell=1}^d (x_{i,\ell} - x_{j,\ell})^2},$$

une composante numériquement très ample se met à **dominer** la distance, rendant négligeables les écarts sur les autres composantes. Cela peut conduire à des mesures de **synergie** $S(i,j)$ (souvent définies comme une fonction décroissante de $\|\mathbf{x}_i - \mathbf{x}_j\|$) faussées, car la “dimension la plus grande” éclipse l’information apportée par les autres.

Pour éviter cette situation, on recourt à des **méthodes de normalisation** :

- **Min–max scaling** pour ramener chaque coordonnée dans [0,1],
- **Z-score** où l'on retranche la moyenne et on divise par l'écart-type,
- **ℓ_2 -normalisation** (projection sur la sphère unitaire) : $\mathbf{x}'_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}$.

Ces approches garantissent que la fonction de synergie ne soit pas **dominée** par un attribut particulier. À titre d'exemple, dans une similarité cosinus, on impose souvent $\|\mathbf{x}_i\|=1$, alors $jS(i,j) = \mathbf{x}_i \cdot \mathbf{x}_j$. Toute différence d'échelle disparaît, et la distance ou la similarité répercute mieux les différences sémantiques réelles.

B. Homogénéité entre Entités

Au sein du **Synergistic Connection Network** (SCN), on peut avoir un grand nombre d'entités analogues (ex. plusieurs capteurs) — certains capteurs peuvent naturellement renvoyer des valeurs plus fortes (en volts ou en dB) tandis que d'autres restent dans des gammes plus faibles. Sans un processus de **calibration**, un capteur “fort” dominera la distance ou la similarité, amenant la pondération $\omega_{i,j}$ à se renforcer majoritairement pour ces entités, même si la différence n'est qu'un **artefact** d'échelle.

De même, en logique symbolique, si l'on attribue des **poids** inégaux aux règles (ex. une “règle principale” vaut 1000, les autres valent 1), alors le calcul de $\text{Compat}(R_i, R_j)$ sera le plus souvent dicté par la présence ou l'absence de cette règle “majeure”. Le réseau pourra, là encore, **exagérer** les proximités lorsque cette règle est partagée, formant ainsi des clusters artificiels.

La **calibration** consiste à imposer une **cohérence** dans la pondération où, par exemple, on peut normaliser la somme des poids symboliques afin que $\sum_{r \in R_i} w(r) = 1$. On peut alors définir :

$$S_{\text{sym}}(i,j) = \frac{\sum_{r \in R_i \cap R_j} w(r)}{\sum_{r \in R_i \cup R_j} w(r)},$$

de sorte qu'une règle unique n'emporte pas la totalité du calcul.

C. Éviter la “Dérive” au Fil du Temps

Dans un **environnement évolutif** (chap. 9), les entités peuvent voir leur représentation se **modifier** où un embedding peut “s'étirer” si le modèle d'apprentissage continu affine ses paramètres, et des règles supplémentaires peuvent être ajoutées à un ensemble symbolique. Si aucun mécanisme de recalibrage n'existe, on court le risque qu'**une** dimension ou **un** attribut gonfle sans limite, bouleversant progressivement la mesure de synergie. Cela se traduit par des **oscillations** dans la mise à jour de $\omega_{i,j}$ ou par l'absorption d'entités disparates dans un même cluster.

Une parade est de fixer un protocole de **renormalisation** ou de “ré-étalonnage” à intervalles réguliers, afin de maintenir l'ensemble des entités dans un même **espace** de comparaison :

$$\mathbf{r}(i, t + 1) = h(\mathbf{r}(i, t)),$$

où h effectue la recalibration nécessaire (remise à l'échelle, recentrage). On peut choisir un **pas** de recalibration adapté ou mettre en place un pilotage heuristique (ex. si la norme moyenne des vecteurs dépasse un seuil, on opère une projection globale).

D. Cas Multimodal ou Hybride

Dans des scénarios **multimodaux**, on a souvent besoin de **fusionner** un score de similarité sub-symbolique (ex. cosinus entre embeddings) et un score symbolique (ex. compatibilité de règles). Si l'on ne borne pas ou ne normalise pas ces scores, l'un peut dominer l'autre. Par exemple, un cosinus variant dans $[-1,1]$ se retrouve dilué face à un score symbolique qui peut atteindre 10 ou 100.

Une **stratégie** consiste à normaliser chaque sous-score dans $[0,1]$ ou $[-1,1]$, puis à combiner linéairement ou par un noyau plus sophistiqué :

$$S_{\text{hybride}}(i, j) = \alpha S_{\text{vect}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

où α est un paramètre. On évite ainsi que le bloc logique (ou le vecteur) n'occupe toute la scène. L'**harmonisation** est d'autant plus critique si le DSL doit manipuler diverses modalités (vision, audio, texte, règles expertes) dans un même SCN.

3.3. Représentations Sub-Symboliques

Les approches sub-symboliques, basées sur des **vecteurs** ou des **embeddings**, constituent l'une des manières les plus répandues de décrire des entités \mathcal{E}_i dans un contexte d'apprentissage automatique. Dans le cadre du DSL (Deep Synergy Learning), elles présentent l'avantage de se prêter à un **calcul** aisément quantifiable de la similarité (voire de la distance) entre entités, conditionnant ainsi la **synergie** $S(i, j)$. Dans cette section, nous étudierons tout d'abord les **vecteurs** et **embeddings** (3.3.1), puis nous aborderons des **méthodes avancées** (autoencodeurs, transformers, etc.) (3.3.2), avant de voir en détail le **calcul** de la synergie entre vecteurs (3.3.3) et la **gestion** du bruit ou de l'évolution de ces représentations (3.3.4).

3.3.1. Vecteurs et Embeddings

La représentation **vectorielle** est sans doute la plus intuitive dans un grand nombre d'applications où chaque entité est codée par un point $\mathbf{x}_i \in \mathbb{R}^d$. Cette approche, déjà ancienne en classification ou en clustering, prend une **nouvelle dimension** avec l'émergence des **embeddings profonds**, capables de saisir des caractéristiques complexes dans les images, les textes ou l'audio.

3.3.1.1. Origine : CNN (images), Word Embeddings (textes), Spectrogrammes (audio)

Dans le cadre d'un **Deep Synergy Learning (DSL)**, une grande partie des représentations sub-symboliques est obtenue par l'extraction d'**embeddings** à l'aide de **modèles neuronaux** spécialisés. L'idée fondamentale consiste à transformer un objet initial – qu'il s'agisse d'une image, d'un texte ou d'un segment audio – en un **vecteur** ou un **tenseur** qui capture les **caractéristiques** les plus pertinentes de l'information. Ce vecteur permet ensuite de calculer la **mesure** de synergie, notée $S(i, j)$, qui reflète la proximité ou la compatibilité entre deux entités \mathcal{E}_i et \mathcal{E}_j . Diverses approches sont utilisées pour générer ces embeddings, et nous détaillons ici trois cas typiques.

Dans le domaine de l'analyse d'images, les **Convolutional Neural Networks (CNN)** ont largement popularisé l'extraction de **vecteurs** de caractéristiques. Un réseau convolutionnel, tel que VGG, ResNet, Inception ou encore Vision Transformer, procède à travers une série de couches de **convolution** et de **pooling** à l'extraction progressive de « feature maps » qui condensent l'information visuelle en attributs de plus en plus abstraits, tels que les **textures**, les **bords** ou encore les **formes**. Au terme de ce processus, on obtient un **embedding global** de dimension d (typiquement 256, 512 ou 1024), représenté par

$$\text{CNN}: \quad \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^d,$$

où $H \times W$ indique la taille de l'image, C le nombre de canaux (par exemple, 3 pour une image RGB), et d la dimension du vecteur. Ce **descripteur** capture ainsi des aspects sémantiques essentiels, et dans un DSL, deux images dont les embeddings \mathbf{x}_i et \mathbf{x}_j se montrent très similaires (par exemple, mesurées via la **similarité cosinus** ou une distance euclidienne inversée) conduisent à une synergie élevée, entraînant le renforcement de la pondération $\omega_{i,j}$ dans le SCN et favorisant l'auto-organisation en clusters d'images visuellement cohérents.

Dans le domaine du **Traitement Automatique du Langage Naturel (TALN)**, la représentation des textes s'appuie sur les **word embeddings** tels que Word2Vec et GloVe, ainsi que sur des modèles contextuels plus récents comme BERT, GPT ou T5. Plutôt que d'utiliser des représentations statiques comme les vecteurs one-hot, on projette chaque mot ou token dans un **espace** de dimension d selon la relation

$$\mathbf{w}: \{\text{mots ou tokens}\} \rightarrow \mathbb{R}^d.$$

Ainsi, deux mots ou expressions d'une **signification** similaire se trouvent naturellement proches dans cet espace, ce qui facilite le calcul de distances ou de **similarités**. De plus, pour obtenir une représentation d'une phrase ou d'un document, il est courant d'agréger les embeddings des mots, par exemple par la moyenne ou en extrayant le vecteur associé au token [CLS] dans les modèles Transformer. Cette approche permet d'assigner à chaque segment textuel un vecteur \mathbf{x}_i qui, lorsqu'il est comparé à un autre vecteur \mathbf{x}_j , donne une mesure de synergie reflétant la **parenté sémantique** entre les textes.

Pour les données **audio**, l'extraction d'embeddings se fait souvent à partir d'un **spectrogramme** ou par l'usage de modèles neuronaux spécialisés, tels que SoundNet ou des Audio Transformers. Le signal audio $a(t)$ est segmenté et transformé par un modèle spécifique, conduisant à un embedding

$$\mathbf{x}_s = \text{AudioModel}(a_s(t)) \in \mathbb{R}^d,$$

qui capture des caractéristiques acoustiques comme le **timbre** et la **prosodie**. Dans un DSL, si deux segments audio présentent des embeddings similaires, la synergie $S(i, j)$ est alors élevée, et la pondération correspondante $\omega_{i,j}$ se renforce, permettant ainsi la formation de clusters cohérents de signaux acoustiques ou la fusion d'une modalité audio avec d'autres modalités dans un cadre multimodal.

D'un point de vue **formel**, après avoir extrait les vecteurs \mathbf{x}_i pour chaque entité, on définit la **mesure** de similarité par exemple par la formule gaussienne

$$S(i, j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|)$$

ou par la formule du **produit scalaire normalisé**

$$S(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}.$$

Lorsque ces scores de similarité sont intégrés dans la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

le DSL renforce les liens entre les entités dont les embeddings sont proches dans l'espace, ce qui conduit à la formation de **clusters** auto-organisés reflétant la **cohérence** des caractéristiques sous-jacentes.

La **force** de cette approche réside dans la capacité des **embeddings** générés par ces modèles neuronaux à capturer une **richesse sémantique** et des aspects contextuels qui permettent d'identifier avec précision les similitudes entre entités issues de domaines variés (vision, langage,

acoustique). Cependant, le succès de cette méthode dépend fortement de la **qualité** des embeddings, laquelle doit être suffisamment robuste pour résister au bruit et capable de généraliser les caractéristiques essentielles des données. Ces principes, approfondis dans le Chapitre 3, soulignent l'importance de disposer de représentations vectorielles de haute qualité pour garantir une auto-organisation efficace au sein du DSL.

3.3.1.2. Avantages : Calcul Aisé de Similarité (Cosinus, Distance Euclidienne)

Dans le cadre du **Deep Synergy Learning (DSL)**, chaque entité \mathcal{E}_i est représentée par un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$ qui est issu d'un modèle d'apprentissage profond tel qu'un CNN, un transformeur pour le langage ou encore un modèle audio pour les spectrogrammes. L'un des atouts majeurs de cette représentation sub-symbolique réside dans la **simplicité algorithmique** avec laquelle il est possible de calculer la **similarité** ou la **distance** entre ces vecteurs. Cette facilité de calcul joue un rôle central dans la formation et l'auto-organisation du **Synergistic Connection Network (SCN)**, puisque la **synergie** $S(i, j)$ entre deux entités est directement fonction de la proximité mesurée dans l'espace vectoriel, et plus cette proximité est élevée, plus la **pondération** $\omega_{i,j}$ tend à être renforcée selon la règle de mise à jour du SCN.

D'un point de vue mathématique, la **similarité cosinus** est définie par

$$S_{\cos}(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|},$$

cette mesure se concentrant sur l'**angle** entre \mathbf{x}_i et \mathbf{x}_j plutôt que sur leur norme, ce qui est particulièrement pertinent dans des contextes où la magnitude du vecteur n'est pas un indicateur de la **pertinence** ou de la **similarité**. En outre, une transformation linéaire de l'intervalle $[-1, 1]$ permet d'obtenir un score de similarité dans $[0, 1]$, par exemple

$$S(i, j) = \frac{1 + S_{\cos}(i, j)}{2},$$

ce qui garantit une interprétation directe dans le cadre de la mise à jour des pondérations. Par ailleurs, la **distance euclidienne** est définie par

$$d_{\text{eucl}}(i, j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2},$$

et cette distance, en quantifiant la longueur du segment reliant \mathbf{x}_i à \mathbf{x}_j , peut être convertie en un score de similarité par l'application d'une fonction de décroissance exponentielle, comme le montre la formule

$$S_{\text{gauss}}(i, j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\alpha > 0$ est un paramètre réglable. Ces deux formules sont linéaires en la dimension d et reposent sur des opérations de multiplication–accumulation, qui sont bien adaptées aux architectures de calcul modernes telles que les GPU ou les bibliothèques optimisées en BLAS, permettant ainsi un

calcul rapide même pour de grands ensembles d’entités. La **simplicité** de ces mesures permet une implémentation efficace de la **mise à jour** des pondérations dans le SCN selon la formule

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

ce qui conduit à un renforcement des liens entre les entités qui se rapprochent dans l’espace vectoriel. L’avantage de cette approche réside également dans le fait qu’elle permet une **interopérabilité** naturelle avec un grand nombre d’algorithmes existants, notamment ceux de **clustering** (tels que K-means, DBSCAN) et des techniques d’**approximate nearest neighbor** (comme FAISS ou Annoy), ce qui facilite l’analyse et la visualisation des regroupements obtenus via des méthodes de réduction de dimension telles que PCA, t-SNE ou UMAP. De plus, cette méthode se prête aisément à des scénarios **multimodaux** où, en projetant des données provenant de différentes sources dans un espace commun, on applique de manière uniforme la logique de calcul de la similarité, assurant ainsi une intégration homogène et une gestion efficace de la **variabilité** des données. Finalement, en ajustant les paramètres des kernels (comme α dans le cas du noyau gaussien), il est possible d’optimiser la **sensibilité** du DSL aux distances entre vecteurs, garantissant ainsi que la **dynamique** de mise à jour des pondérations reflète de manière fine les structures sous-jacentes des données.

3.5.3.3. Inconvénients : Dimension Élevée, Parfois Peu Interprétables

L’utilisation de **représentations sub-symboliques** sous forme de vecteurs d’embeddings, générés par des modèles neuronaux tels que les CNN ou les Transformers, offre indéniablement des avantages en termes de simplicité et d’efficacité du calcul de la synergie dans \mathbb{R}^d . Toutefois, ces méthodes présentent plusieurs inconvénients qui peuvent poser des problèmes de **scalabilité**, de **stabilité** et d’**explicabilité**. La discussion qui suit s’articule autour de quatre axes majeurs.

A. Dimension Élevée : Risques et Limitations

Les modèles neuronaux modernes produisent des embeddings dont la dimension peut atteindre plusieurs centaines voire milliers (par exemple, 512, 768 ou 1024 dimensions). Lorsque la fonction de synergie $S(i,j)$ est calculée de manière naïve pour toutes les paires (i,j) au sein d’un Synergistic Connection Network (SCN) composé de n entités, le coût de calcul devient de l’ordre de $O(n^2 \times d)$. Ainsi, si n est très grand – par exemple, plusieurs millions d’entités – et d élevé, le temps de calcul peut devenir prohibitif, comme l’illustre l’estimation

$$\text{Coût} \approx n^2 \times d.$$

De plus, la malédiction de la dimension peut rendre les distances moins discriminantes, puisque dans des espaces de très haute dimension, les distances entre la plupart des points tendent à converger vers une valeur similaire. Cette uniformisation des distances peut aboutir à une dynamique d’auto-organisation moins stable, avec des clusters qui se forment de manière peu distincte ou qui présentent des frontières floues. Face à ces difficultés, des approches telles que l’utilisation d’algorithmes d’**approximate nearest neighbors**, de techniques de **sparsification** ou de mécanismes de **sélection** ne traitant que les paires prometteuses sont souvent nécessaires afin de réduire la charge de calcul.

B. Opacité et Faible Interprétabilité

Les embeddings issus de réseaux neuronaux sont le produit d'opérations complexes et hiérarchiques réparties sur de multiples couches (convolutions, mécanismes d'attention, etc.). Cette complexité rend souvent l'interprétation de chaque coordonnée de l'embedding difficile. Il n'est pas aisément de déterminer si une dimension particulière de \mathbf{x}_i correspond à une caractéristique précise, telle qu'une texture, une forme ou un concept lexical.

De surcroît, lorsqu'on constate qu'un score de similarité élevé existe entre deux vecteurs \mathbf{x}_i et \mathbf{x}_j , il est ardu d'attribuer de manière explicite la cause de cette proximité – qu'il s'agisse de la couleur, de la catégorie ou d'une co-occurrence textuelle. Cette opacité pose un réel problème dans des domaines exigeant une forte **explicabilité**.

Dans des applications critiques comme le médical ou l'industrie, il est impératif de pouvoir justifier la formation d'un cluster ou la liaison entre deux entités. Même si des techniques de réduction de dimension, telles que PCA, UMAP ou t-SNE, permettent d'obtenir une vue globale des distributions, elles n'améliorent guère l'interprétation des attributs à l'échelle individuelle.

C. Instabilité des Embeddings et Variations Contextuelles

Les embeddings neuronaux peuvent varier sensiblement en fonction du **contexte** ou des modifications apportées lors de la phase d'entraînement. Dans le domaine du traitement du langage naturel, par exemple, une légère modification du corpus d'entraînement ou un nouveau fine-tuning peut entraîner des décalages dans les positions des mots ou des tokens dans l'espace vectoriel.

De même, en vision par ordinateur, l'adaptation d'un réseau pour intégrer de nouvelles classes d'images peut modifier la distribution finale des embeddings \mathbf{x}_i . De telles variations contextuelles impactent directement la mesure de la synergie, que ce soit par la modification de la distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ ou par le produit scalaire $\mathbf{x}_i \cdot \mathbf{x}_j$.

Cette instabilité peut ainsi perturber la dynamique du SCN, induisant des réorganisations brutales ou imprévues dans l'auto-organisation, et posant un défi pour la stabilité à long terme du DSL.

D. Nécessité d'un Prétraitement Potentiellement Coûteux

L'obtention de ces embeddings repose sur l'utilisation de modèles préentraînés, tels que des réseaux CNN pour les images ou des Transformers pour les textes, qui nécessitent une infrastructure de calcul robuste (GPU/TPU) et des ressources importantes pour l'entraînement ou l'inférence.

Si l'objectif est de concevoir un DSL autonome fonctionnant sur des ressources matérielles limitées, le coût en termes de calcul et de temps associé au prétraitement des données peut s'avérer prohibitif.

De plus, le déploiement d'un tel système requiert la mise en place d'un pipeline complet, capable de fournir en continu des embeddings de haute qualité, ce qui représente une contrainte supplémentaire en termes d'ingénierie et d'infrastructure.

3.3.2. Autoencodeurs, Transformers et Approches Avancées

Les vecteurs et embeddings présentés en section 3.3.1 constituent déjà une base solide pour représenter des entités de nature variée (images, textes, sons). Toutefois, les **techniques** d'apprentissage ne cessant d'évoluer, il est possible d'aller **plus loin** en recourant à des méthodes avancées qui extraient des embeddings encore plus **expressifs** ou plus **compacts**.

Dans cette section, nous verrons d'abord (3.3.2.1) comment les **autoencodeurs** peuvent servir à la **réduction dimensionnelle** ou à l'**extraction de features**, puis (3.3.2.2) comment les **Transformers** (BERT, GPT, ViT) produisent des embeddings **contextuels**, et enfin (3.3.2.3) nous soulignerons l'**intérêt** de ces méthodes avancées pour le DSL, au prix toutefois d'une complexité de calcul plus élevée.

3.3.2.1. Autoencodeurs pour Réduction Dimensionnelle ou Extraction de Features

Dans le cadre d'un **Deep Synergy Learning (DSL)**, l'extraction d'informations pertinentes à partir de représentations sub-symboliques constitue un enjeu majeur pour l'efficacité du processus d'auto-organisation. Les **autoencodeurs** (**AE**) se présentent comme un outil fondamental permettant d'apprendre, de manière non supervisée, un **code latent** $\mathbf{z} \in \mathbb{R}^k$ qui capture l'essence d'un vecteur d'entrée \mathbf{x} . Cette démarche offre deux avantages essentiels pour le DSL où elle permet la **réduction de dimension** et l'**extraction de features discriminantes**.

A. Principe et Architecture d'un Autoencodeur

Un autoencodeur se compose typiquement de deux modules complémentaires.

Le premier, l'**encodeur** E , est une fonction $E: \mathbb{R}^d \rightarrow \mathbb{R}^k$ qui reçoit un vecteur d'entrée $\mathbf{x} \in \mathbb{R}^d$ et produit un **code latent** $\mathbf{z} \in \mathbb{R}^k$, où $k < d$ dans le but de compresser l'information.

Le second module, le **décodeur** D , opère la transformation inverse, $D: \mathbb{R}^k \rightarrow \mathbb{R}^d$, et tente de reconstruire le vecteur d'entrée à partir de ce code latent, produisant ainsi $\hat{\mathbf{x}} = D(E(\mathbf{x}))$.

L'apprentissage se réalise en minimisant une **fonction de perte de reconstruction** $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$, souvent formulée comme la moyenne des erreurs quadratiques, ce qui s'exprime par

$$\min_{E, D} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(\mathbf{x}_i, D(E(\mathbf{x}_i))).$$

Cette optimisation force l'encodeur à extraire les facteurs de variation les plus pertinents de \mathbf{x} , tout en éliminant le bruit et les détails superflus, aboutissant ainsi à une représentation compacte et significative.

B. Avantages : Réduction de Dimension et Extraction de Traits

L'imposition d'un goulot d'étranglement, où la dimension du code latent k est inférieure à celle de l'entrée d , engendre deux bénéfices notables pour le DSL.

D'une part, la **réduction dimensionnelle** permet de diminuer significativement le coût de calcul lors de l'évaluation des distances ou similarités entre entités. En effet, le calcul de la distance entre deux codes latents, par exemple $\| \mathbf{z}_i - \mathbf{z}_j \|$, se réalise en $O(k)$ plutôt qu'en $O(d)$, ce qui est particulièrement avantageux lorsque le réseau traite un grand nombre d'entités.

D'autre part, en apprenant à reconstruire \mathbf{x} à partir de \mathbf{z} , l'autoencodeur extrait des **features discriminantes** qui résument les aspects sémantiques les plus importants du vecteur d'origine, facilitant ainsi la formation de clusters cohérents et la détection de similarités véritables dans le SCN.

C. Variantes : Denoising, Sparse et Variational Autoencoder

Il existe plusieurs variantes d'autoencodeurs qui visent à améliorer la robustesse et la qualité des représentations latentes.

Premièrement, le **Denoising Autoencoder** introduit du bruit dans \mathbf{x} durant l'entraînement et apprend à reconstruire la version nettoyée de la donnée. Ce procédé confère au code latent \mathbf{z} une meilleure tolérance aux perturbations, ce qui est particulièrement utile dans des environnements où les données sont bruitées.

Ensuite, le **Sparse Autoencoder** incorpore une contrainte de sparsité sur \mathbf{z} – par exemple, via une pénalisation de la norme L_1 – de sorte que seule une fraction limitée des neurones s'active pour chaque donnée, ce qui peut améliorer l'interprétabilité des caractéristiques extraites et favoriser une séparation plus nette des classes.

Enfin, le **Variational Autoencoder (VAE)** adopte une approche probabiliste en imposant que le code latent suive une distribution prédéfinie, généralement gaussienne, c'est-à-dire $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$. Cette formulation permet non seulement de générer de nouvelles données, mais aussi d'organiser l'espace latent de manière plus lisse, facilitant ainsi la continuité et la robustesse des clusters formés lors de l'auto-organisation.

D. Intégration au DSL : La Synergie entre Codes Latents

Une fois l'autoencodeur entraîné, la partie **encodeur** E est utilisée pour transformer chaque vecteur d'entrée \mathbf{x}_i en un code latent $\mathbf{z}_i = E(\mathbf{x}_i)$ dans un espace de dimension réduite \mathbb{R}^k . Ce code latent remplace alors \mathbf{x}_i dans la phase de calcul de la synergie au sein du DSL. La mesure de synergie peut être définie par des formules classiques telles que

$$\mathbf{S}(i, j) = \exp(-\alpha \| \mathbf{z}_i - \mathbf{z}_j \|^2)$$

ou par une mesure basée sur la similarité cosinus

$$\mathbf{S}(i, j) = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\| \mathbf{z}_i \| \| \mathbf{z}_j \|}.$$

Ces scores de synergie, calculés dans un espace de dimension réduite, facilitent le renforcement des pondérations $\omega_{i,j}$ et la formation de clusters dans le Synergistic Connection Network. L'optimisation de la dimension k joue alors un rôle critique où si k est trop petit, l'autoencodeur risque de perdre des informations essentielles, tandis qu'un k trop élevé n'apporte pas le gain de complexité recherché.

E. Limites et Points de Vigilance

Malgré leurs avantages, les autoencodeurs présentent quelques inconvénients notables.

Premièrement, la minimisation de la **perte de reconstruction** n'est pas nécessairement alignée avec l'objectif de séparer distinctement les classes ou de maximiser la discrimination entre les entités. Par conséquent, les codes latents \mathbf{z}_i peuvent parfois ne pas être suffisamment discriminants pour la tâche d'auto-organisation.

Deuxièmement, l'entraînement d'un autoencodeur sur des datasets volumineux peut être coûteux en termes de ressources computationnelles et de temps, notamment lorsqu'il s'agit d'architectures profondes.

Troisièmement, le choix de la dimension latente k est crucial où une valeur inappropriée peut soit sous-estimer les facteurs de variation essentiels, soit ne pas fournir une véritable réduction de complexité. Enfin, dans un contexte d'apprentissage continu, la mise à jour de l'encodeur E doit être soigneusement gérée pour éviter le phénomène de **catastrophic forgetting**, qui pourrait altérer l'organisation de l'espace latent déjà établi.

3.3.2.2. Transformers (BERT, GPT, ViT) : Embeddings Contextuels plus Riches

L'émergence des **Transformers** – tels que BERT, GPT, T5 en NLP et Vision Transformer (ViT) en vision – a profondément transformé la manière dont sont générés les **embeddings** dans divers domaines.

Contrairement aux représentations plus statiques (comme Word2Vec) ou aux approches purement convolutionnelles (CNN), les Transformers produisent des **embeddings contextuels**. Ces représentations intègrent non seulement l'information d'un token ou d'un patch de manière isolée, mais elles capturent également le **contexte global** par le biais d'un mécanisme de *self-attention*.

Dans le cadre d'un **Deep Synergy Learning (DSL)**, cette capacité à incorporer le contexte se traduit par des mesures de synergie plus fines et expressives, où la proximité entre deux entités reflète non seulement des similarités de caractéristiques isolées, mais aussi des interactions complexes au sein de l'ensemble des données.

A. Principe des Transformers et Self-Attention

L'architecture Transformer repose sur un mécanisme de *self-attention* qui permet à chaque position d'une séquence de « regarder » toutes les autres positions. Soit une séquence d'entrées représentées par la matrice $\mathbf{x} \in \mathbb{R}^{n \times d}$, où chaque ligne correspond à un embedding initial d'un token ou d'un patch. Les Transformers calculent trois matrices essentielles par multiplication avec des poids appris :

$$\mathbf{Q} = \mathbf{x} W_Q, \quad \mathbf{K} = \mathbf{x} W_K, \quad \mathbf{V} = \mathbf{x} W_V,$$

où W_Q , W_K , et W_V sont des matrices de projection de dimensions appropriées (souvent $d \times d_k$). Le mécanisme de self-attention se définit alors par l'opération

$$\text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}.$$

Cette opération permet à chaque token (ou patch) de pondérer les contributions de tous les autres éléments en fonction de la similitude entre ses **requêtes** et les **clés** associées aux autres tokens, produisant ainsi des embeddings qui intègrent l'information contextuelle de l'ensemble de la séquence.

B. BERT, GPT : Embeddings Contextuels en NLP

Dans le domaine du **Traitemet Automatique du Langage Naturel (NLP)**, des modèles tels que **BERT** et **GPT** exploitent pleinement le mécanisme de self-attention pour produire des **embeddings contextuels**. Concrètement, après plusieurs couches de self-attention et de transformations non linéaires, chaque token i de la séquence obtient une représentation finale $\mathbf{h}_i^{(L)}$. Deux types d'extractions sont couramment utilisées :

- **Token-level** : Chaque mot ou sous-mot se voit attribuer un embedding \mathbf{h}_i qui capture ses nuances contextuelles.
- **Embedding global** : Un vecteur représentatif de l'ensemble de la séquence est généré, souvent en utilisant le token spécial $[CLS]$ (dans le cas de BERT) ou par agrégation (moyenne ou max) sur tous les tokens.

Dans un DSL, ces embeddings globaux servent de « signature » vectorielle pour une entité textuelle. La synergie entre deux entités \mathcal{E}_i et \mathcal{E}_j peut ainsi être évaluée par des mesures telles que la similarité cosinus :

$$S(i, j) = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|},$$

où \mathbf{z}_i et \mathbf{z}_j sont les embeddings finaux issus du Transformer. Grâce à la prise en compte du contexte, ces représentations permettent de discerner des relations sémantiques fines, telles que la synonymie contextuelle ou les paraphrases, améliorant ainsi la qualité de l'auto-organisation du réseau.

C. Vision Transformer (ViT) : Patches et Self-Attention en Vision

Les principes du Transformer se sont également appliqués au domaine de la vision par ordinateur via le **Vision Transformer (ViT)**.

Dans ce cas, une image est divisée en petits **patches** (par exemple, de 16×16 pixels), qui sont linéarisés et projetés dans un espace de dimension d . Pour conserver l'information spatiale, des **positional embeddings** sont ajoutés à chaque patch. Le Transformer traite alors l'ensemble des patches de manière similaire aux tokens en NLP, utilisant le mécanisme de self-attention pour capturer des relations globales dans l'image.

La sortie finale peut être constituée d'un token spécial $[CLS]$ agissant comme représentation globale ou d'une agrégation des embeddings individuels. Cette approche permet de générer des embeddings d'images qui intègrent non seulement des caractéristiques locales, mais aussi des

dépendances à longue portée entre différentes régions de l'image, améliorant ainsi la mesure de la synergie dans le DSL.

D. Enjeux et Coûts : Dimension et Ressources

Bien que les Transformers produisent des embeddings contextuels particulièrement riches, ils impliquent également des coûts importants en termes de **dimension** et de **ressources**.

Par exemple, les modèles tels que BERT ou GPT génèrent des vecteurs de dimension 768, 1024 ou plus, ce qui peut amplifier la complexité des calculs de similarité dans un SCN, souvent de l'ordre de $O(n^2 \times d)$ pour n entités.

De plus, le poids des modèles (parfois plusieurs centaines de millions de paramètres) requiert une infrastructure matérielle (GPU, TPU) conséquente pour l'inférence et le fine-tuning.

Enfin, la sensibilité des Transformers aux ajustements fins – un léger changement de paramètres peut modifier la distribution des embeddings – nécessite des techniques d'alignement ou de versionnage pour préserver la stabilité des représentations dans un système d'apprentissage continu.

E. Intégration dans un DSL

L'intégration des Transformers dans un DSL se réalise en extrayant, pour chaque entité \mathcal{E}_i (qu'il s'agisse d'un document, d'une image ou d'un segment audio), un embedding contextuel \mathbf{z}_i via une opération telle que

$$\mathbf{z}_i = \text{TransformerEncode}(\mathbf{x}_i),$$

où \mathbf{x}_i représente l'objet brut initial et TransformerEncode désigne le processus de passage par plusieurs couches de self-attention et de feed-forward. Une fois obtenu, l'embedding peut être normalisé afin de faciliter le calcul de similarités, par exemple en utilisant le cosinus. La synergie entre deux entités est alors définie par une fonction f (comme le cosinus ou un noyau RBF) :

$$S(i, j) = f(\mathbf{z}_i, \mathbf{z}_j).$$

Dans le processus de mise à jour des pondérations du SCN,

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

une synergie élevée entre des embeddings contextuels conduira au renforcement des liens, favorisant ainsi la formation de clusters d'entités sémantiquement cohérents, tout en capturant des relations complexes que seules des approches traditionnelles auraient du mal à discerner.

3.3.2.3. Intérêt pour le DSL : un Embedding plus Robuste, mais plus Complexé à Calculer ou à Mettre à Jour

Dans le cadre d'un **Deep Synergy Learning (DSL)**, l'emploi de techniques avancées telles que les **Transformers** ou les **autoencodeurs profonds** permet d'obtenir des **embeddings** d'une qualité remarquable, caractérisés par une richesse contextuelle et une capacité de discrimination

supérieure. Ces représentations, issues de modèles sophistiqués, se distinguent par leur aptitude à capter des nuances sémantiques fines et à structurer l'espace latent de manière à favoriser la formation de clusters d'entités plus robustes. On définit notamment, par une fonction g , l'embedding contextuel d'une entité initiale \mathbf{x}_i par

$$\mathbf{z}_i = g(\mathbf{x}_i),$$

ce qui permet de mesurer la **synergie** entre deux entités \mathcal{E}_i et \mathcal{E}_j à l'aide d'une distance ou d'un produit scalaire, par exemple

$$S(i, j) = \exp(-\alpha \|\mathbf{z}_i - \mathbf{z}_j\|^2) \quad \text{ou} \quad S(i, j) = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|},$$

ce qui permet de regrouper les entités ayant des caractéristiques latentes similaires au sein du **Synergistic Connection Network (SCN)**. Ces méthodes confèrent au DSL une **robustesse** accrue dans la capture des variations contextuelles et une capacité à distinguer subtilement des différences sémantiques que des approches plus simples auraient tendance à négliger.

Cependant, la qualité supérieure des embeddings contextuels se traduit par une **complexité** notable tant dans le calcul de l'inférence que dans la gestion des mises à jour de ces représentations.

Le modèle g , en intégrant plusieurs couches de self-attention et de transformations non linéaires, nécessite en général une infrastructure matérielle puissante, telle que des GPU ou TPU, en raison du nombre élevé de paramètres et de la dimension des vecteurs, souvent de l'ordre de 768, 1024 ou plus.

Par conséquent, le coût de calcul d'une opération d'inférence, qui peut être évalué en $O(n^2 \times d_{\text{latent}})$ pour un ensemble de n entités, augmente considérablement, en particulier lorsque le DSL doit traiter de grands volumes de données en temps réel.

Cette complexité algorithmique se trouve également exacerbée par le coût associé à la mise à jour des embeddings dans un contexte d'apprentissage continu. En effet, si le modèle g subit un fine-tuning ou une révision partielle, il devient nécessaire de recalculer les embeddings pour toutes les entités, c'est-à-dire

$$\mathbf{z}_i(t+1) = g_{t+1}(\mathbf{x}_i) \quad \text{pour } i = 1, \dots, n,$$

ce qui, pour des ensembles de données de grande taille, peut engendrer une latence non négligeable et provoquer des fluctuations dans les valeurs de $S(i, j)$ entre les cycles de mise à jour. Ces variations, même modestes, peuvent induire des réorganisations brutales au sein du SCN, compromettant ainsi la stabilité de l'auto-organisation.

L'ingénierie d'un DSL doit donc établir un compromis entre la **richesse** des embeddings, qui améliore la précision des regroupements en capturant des informations contextuelles complexes, et l'**efficacité computationnelle** requise pour leur calcul et leur mise à jour régulière.

Dans certains contextes critiques, où la qualité de la représentation est primordiale – par exemple en médecine ou en analyse de données sensibles – il peut être acceptable de supporter des coûts de calcul élevés pour garantir une **précision** et une **robustesse** accrues.

À l'inverse, dans des applications massives où la rapidité d'inférence est essentielle, des modèles plus légers ou des embeddings de dimension réduite pourraient être privilégiés afin d'optimiser la scalabilité du système.

L'utilisation d'embeddings plus riches, issus de techniques avancées, permet d'améliorer significativement la **discrimination** et la **robustesse** des synergies dans un DSL, mais se fait au prix d'une complexité computationnelle accrue et d'un défi supplémentaire en termes de mise à jour des représentations dans un cadre d'apprentissage continu.

Ce compromis entre la richesse des représentations et l'efficacité opérationnelle constitue un enjeu central dans la conception et l'implémentation de systèmes de Deep Synergy Learning.

3.3.3. Calcul de la Synergie entre Vecteurs

Dans les sections précédentes (3.3.1 et 3.3.2), nous avons mis en évidence l'usage de **vecteurs** (embeddings) pour représenter des entités dans un contexte sub-symbolique, ainsi que l'apport de méthodes avancées (autoencodeurs, Transformers) pour produire des embeddings plus riches. Il reste à voir **comment**, à partir de ces vecteurs, on définit la **synergie** $S(i,j)$ entre deux entités \mathcal{E}_i et \mathcal{E}_j .

En pratique, le DSL (Deep Synergy Learning) recourt souvent à des **fonctions** de distance ou de similarité géométriques. Cette section (3.3.3) se divise en trois points :

- (3.3.3.1) présentation des **distances usuelles** (euclidienne, manhattan) et de la **similarité cosinus**,
- (3.3.3.2) usage de **kernels** pour une similarité non linéaire,
- (3.3.3.3) **ajustements** (normalisation, calibration) pour mieux gérer le bruit ou l'échelle des vecteurs.

3.3.3.1. Distances Usuelles : Euclidienne, Manhattan, Cosinus

Dans le cadre du **Deep Synergy Learning (DSL)**, chaque entité \mathcal{E}_i est représentée par un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$. L'évaluation de la **synergie** entre deux entités repose alors sur la comparaison de ces représentations via une fonction de **distance** ou de **similarité**. Ces mesures jouent un rôle fondamental dans le processus d'**auto-organisation** du **Synergistic Connection Network (SCN)**, puisque plus deux vecteurs sont « proches » dans l'espace, plus la **pondération** $\omega_{i,j}$ tend à être renforcée.

Parmi les approches classiques pour mesurer cette proximité, on trouve la **distance euclidienne**, la **distance manhattan** et la **similarité cosinus**, chacune offrant une interprétation géométrique différente et répondant à des exigences spécifiques selon le type de données traitées.

A. Distance Euclidienne

La **distance euclidienne** constitue la mesure la plus intuitive, puisqu'elle correspond à la longueur du segment reliant les points représentés par \mathbf{x}_i et \mathbf{x}_j dans \mathbb{R}^d . Mathématiquement, elle se définit par

$$d_{\text{eucl}}(i, j) = \| \mathbf{x}_i - \mathbf{x}_j \| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}.$$

La valeur ainsi obtenue représente une notion de **distance** physique dans l'espace, et dans le DSL, cette distance est souvent convertie en un **score de similarité** afin de faciliter son intégration dans la dynamique du SCN. Une fonction d'agrégation classique consiste à appliquer une décroissance exponentielle, ce qui donne

$$\mathbf{S}(i, j) = \exp(-\alpha \| \mathbf{x}_i - \mathbf{x}_j \|^2),$$

où $\alpha > 0$ est un **paramètre** qui ajuste la pente de décroissance. Dans ce contexte, une distance faible entre \mathbf{x}_i et \mathbf{x}_j conduit à un score proche de 1, indiquant une **affinité élevée** entre les deux entités et justifiant ainsi le renforcement de la pondération $\omega_{i,j}$.

B. Distance Manhattan (ou ℓ_1)

La **distance manhattan** ou distance **taxicab** s'exprime en prenant la somme des différences absolues entre les composantes correspondantes des vecteurs. La formule est donnée par

$$d_{\text{manh}}(i, j) = \sum_{k=1}^d |x_{i,k} - x_{j,k}|.$$

Cette mesure, qui évalue la différence cumulée par dimension, est particulièrement utile lorsque les données se combinent de façon additive, et elle présente l'avantage de se montrer souvent plus **robuste** face aux valeurs extrêmes de certaines coordonnées. Pour intégrer cette distance dans un DSL, on peut transformer la mesure en un score de similarité en appliquant par exemple la fonction exponentielle

$$\mathbf{S}(i, j) = \exp(-\beta d_{\text{manh}}(i, j)),$$

ou encore en utilisant la forme rationnelle

$$\mathbf{S}(i, j) = \frac{1}{1 + d_{\text{manh}}(i, j)},$$

où $\beta > 0$ est un **paramètre** qui ajuste le taux de décroissance. De cette manière, une distance manhattan réduite se traduit par un score de similarité élevé, indiquant une forte **affinité** entre les entités.

C. Similarité Cosinus

La **similarité cosinus** se distingue des mesures de distance classiques en se focalisant sur l'**angle** entre deux vecteurs plutôt que sur leur distance absolue. Elle est définie par

$$S_{\cos}(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|},$$

et prend des valeurs dans l'intervalle $[-1, 1]$. Cette mesure atteint la valeur 1 lorsque les vecteurs sont colinéaires dans la même direction, -1 lorsqu'ils sont opposés et 0 lorsque les vecteurs sont orthogonaux. Dans le cadre du DSL, il est souvent souhaitable d'obtenir un score strictement positif, et pour ce faire, on peut transformer l'intervalle en appliquant par exemple la formule

$$S(i, j) = \frac{1 + S_{\cos}(i, j)}{2},$$

qui convertit les valeurs de $[-1, 1]$ en $[0, 1]$. Cette méthode permet de neutraliser l'effet de la norme des vecteurs, en se concentrant uniquement sur leur **direction**, ce qui est particulièrement utile pour des données textuelles ou pour des représentations où la magnitude peut varier de manière significative.

D. Liaison avec la Synergie dans le DSL

L'ensemble des mesures présentées permet de définir la **synergie** entre deux entités, laquelle est utilisée pour ajuster la **pondération** $\omega_{i,j}$ dans le SCN. La règle de mise à jour se formalise par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ est le **taux d'apprentissage** et $\tau > 0$ représente le **coefficent de décroissance**. Ce mécanisme d'auto-organisation implique que plus deux entités sont évaluées comme étant proches (ou similaires) selon la mesure choisie, plus la pondération $\omega_{i,j}$ tend à augmenter, favorisant ainsi la formation de **clusters** cohérents.

Le choix entre la **distance euclidienne**, la **distance manhattan** ou la **similarité cosinus** dépend du type de données et des propriétés spécifiques recherchées, car chacune offre des avantages distincts. La distance euclidienne fournit une mesure classique de la proximité, la distance manhattan peut mieux gérer certaines disparités dimensionnelles et la similarité cosinus permet de se focaliser sur l'orientation des vecteurs indépendamment de leur norme.

3.3.3.2. Kernels (RBF, Polynomial) pour une Similarité non Linéaire

Les **distances** usuelles (euclidienne, manhattan) ou la **similarité cosinus** (section 3.3.3.1) considèrent des relations souvent **linéaires** ou du moins directes dans l'espace vectoriel. Cependant, il arrive que la **relation** réelle entre deux entités \mathcal{E}_i et \mathcal{E}_j soit plus subtile, nécessitant des méthodes **non linéaires** pour en rendre compte. Dans un **Deep Synergy Learning (DSL)**, recourir à des **kernels** (noyaux) permet d'introduire des notions de **similarité** bien plus riches que les simples distances ℓ_p . Les **RBF kernels** (noyaux gaussiens) et **kernels polynomiaux** figurent

parmi les plus répandus et se traduisent en un **score** $S(i, j)$ apte à révéler des liens complexes cachés dans l'espace initial \mathbb{R}^d .

A. Principe Général des Kernels

Un **kernel** k entre deux vecteurs $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ se définit comme une **fonction** :

$$k: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R},$$

renvoyant un score (positif ou non) quantifiant leur **affinité**. La **particularité** du kernel trick, issu de la théorie des **noyaux de Mercer**, réside dans le fait que $k(\mathbf{x}_i, \mathbf{x}_j)$ peut être interprété comme un **produit scalaire** dans un espace de dimension éventuellement infinie :

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}},$$

où $\phi: \mathbb{R}^d \rightarrow \mathcal{H}$ est une **transformation** non linéaire, et \mathcal{H} un **espace** (souvent de Hilbert) de dimension potentiellement énorme. La **magie** du kernel trick est qu'on n'a pas besoin de **calculer** explicitement ϕ . Il suffit d'évaluer $k(\mathbf{x}_i, \mathbf{x}_j)$. D'un point de vue **DSL**, cela équivaut à définir la **synergie** $S(i, j)$:

$$S(i, j) = k(\mathbf{x}_i, \mathbf{x}_j).$$

Deux entités peuvent se trouver “lointaines” selon une métrique linéaire, tout en étant **fortement similaires** en termes de transformations non linéaires. Dans un **Synergistic Connection Network**, on peut alors capter des formes de **ressemblance** plus complexes, sans explicitement projeter les entités dans un espace étendu.

B. RBF Kernel (Radial Basis Function)

Le **kernel RBF** (aussi appelé “noyau gaussien”) constitue un choix très répandu. Il s’écrit :

$$k_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (\gamma > 0).$$

Il s’agit d’une **exponentielle** négative de la distance euclidienne au carré. On peut l’interpréter comme une “**gaussienne**” centrée sur \mathbf{x}_j . Plus \mathbf{x}_i est proche de \mathbf{x}_j , plus la valeur se rapproche de 1 ; plus ils sont éloignés, plus elle tend vers 0. Le **paramètre** γ détermine la “largeur” du noyau :

- Un γ **grand** induit une décroissance rapide de $k_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j)$ dès que $\|\mathbf{x}_i - \mathbf{x}_j\|$ grandit, ce qui focalise la similarité sur une zone très **locale**.
- Un γ **petit** rend la fonction plus “plate”, donnant une vue plus **globale** où des entités modérément éloignées conservent une similarité notable.

Pour un **DSL**, un **RBF kernel** donne une **synergie** :

$$S(i, j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2).$$

Cette forme **non linéaire** permet à des entités ayant un écart euclidien significatif (dans \mathbb{R}^d) de tout de même afficher une similarité mesurable si elles partagent certaines **propriétés** sous-

jacentes. En outre, si γ est bien paramétré, la synergie reflète de subtiles variations dans les données — plus subtiles, parfois, qu'une simple distance euclidienne.

C. Polynomial Kernel

Un autre kernel usuel est le **kernel polynomial** :

$$k_{\text{poly}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^p,$$

où p est le **degré** (typiquement 2, 3, 4...) et $c \geq 0$ un terme de décalage. Cette définition incorpore toutes les **composantes** polynomiales du produit $\mathbf{x}_i \cdot \mathbf{x}_j$, allant jusqu'au degré p . Ainsi, pour $p = 2$, on inclut toutes les interactions bilinéaires $(x_{i,k}x_{j,\ell})$. Cela peut capturer des **corrélations** plus complexes qu'un simple "angle" ou qu'une distance linéaire.

Dans un **SCN**, si on adopte $S(i, j) = k_{\text{poly}}(\mathbf{x}_i, \mathbf{x}_j)$, on se retrouve avec une **synergie** qui peut s'avérer très sensible à des variations dans certaines dimensions, surtout pour un p élevé. Cela peut amplifier des **propriétés** souhaitées (s'il y a de fortes corrélations polynomiales), mais risque aussi de **surexposer** le bruit ou les valeurs extrêmes. Le choix de p et c réclame donc un réglage attentif.

D. Non-Linéarité et Espace Implicit : Kernel Trick

Le **kernel trick** signifie que l'on n'a **pas** besoin de définir explicitement une transformation $\phi(\cdot)$ vers un espace de dimension (parfois énorme) \mathcal{H} . Il suffit de calculer $k(\mathbf{x}_i, \mathbf{x}_j)$. Dans un **DSL**, cela se traduit par une **synergie** :

$$S(i, j) = k(\mathbf{x}_i, \mathbf{x}_j) \quad \text{où } k \text{ est RBF, polynomial, ou autre.}$$

Deux vecteurs lointains en \mathbb{R}^d peuvent devenir "proches" dans l'espace \mathcal{H} , si la transformation ϕ induite par le kernel met en évidence des attributs partagés. Ainsi, la **mise à jour** $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \dots$ tient compte de propriétés non linéaires. Dans la **formation** de clusters, des entités qui ne semblaient pas se ressembler au sens euclidien peuvent se rapprocher si leur **synergie** kernel s'avère élevée.

E. Exemples d'Implémentation dans un SCN

Pour un **SCN** gérant n entités, on définit $S(i, j) = k(\mathbf{x}_i, \mathbf{x}_j)$. À chaque itération, calculer $\omega_{i,j}(t+1)$ exige d'évaluer ces kernels. Sur le plan **coût** :

- **RBF** : $O(d)$ multiplications pour $\|\mathbf{x}_i - \mathbf{x}_j\|^2$, puis une exponentiation.
- **Polynomial** : $O(d)$ pour le produit scalaire $\mathbf{x}_i \cdot \mathbf{x}_j$, puis l'élévation au degré p .

Pour comparer toutes les paires (i, j) , on fait face à $O(n^2)$ évaluations, chaque en $O(d)$. Si n est grand, cela devient lourd (voir chap. 3.2.2.2). En pratique, on peut élaguer en ne calculant la synergie que sur un **voisinage** restreint (approximate nearest neighbor, etc.) ou recourir à d'autres heuristiques d'échantillonnage pour gérer la **scalabilité**.

F. Limites et Paramétrage Sensible

Le **RBF kernel** nécessite le choix d'un paramètre γ . Un γ trop grand produit une décroissance exponentielle ultra-rapide, rendant la synergie presque nulle pour la majorité des paires, ce qui peut isoler excessivement les entités. Un γ trop petit, en revanche, aboutit à un noyau très plat, de sorte que presque tout le monde se retrouve "similaire" : on aboutit à un **mégacluster**. Le **polynomial kernel** demande aussi de fixer p et c . Un p trop élevé peut amplifier les bruits, alors qu'un p trop faible (ex. 1) redescend à un comportement linéaire classique.

L'ingénieur d'un **DSL** doit donc **régler** ces hyperparamètres en fonction des données, par validation croisée ou d'autres heuristiques. Les kernels offrent une flexibilité précieuse, mais nécessitent une **phase d'ajustement** pour qu'ils produisent des **synergies** stables et cohérentes dans le SCN.

3.3.3.3. Ajustements : Normalisation, Calibration, Prise en Compte du Bruit

Dans un **Deep Synergy Learning (DSL)** où les entités \mathcal{E}_i sont décrites par des **vecteurs** ou des attributs numériques, l'évaluation de la **synergie** $S(i, j)$ repose sur des **distances** (euclidienne, manhattan) ou des **similarités** (cosinus, kernels). Toutefois, dans la **pratique**, il ne suffit pas de choisir une métrique. Il faut aussi **calibrer** et **ajuster** la façon dont on compare les vecteurs. Les questions de normalisation, d'échelle, de saturation, ou encore de **prise en compte** du bruit sont essentielles pour garantir la **cohérence** du calcul de $S(i, j)$ et la stabilité de la mise à jour $\omega_{i,j}$. Les sections qui suivent décrivent en détail comment on gère ces ajustements pour un **Synergistic Connection Network (SCN)** robuste et efficace.

A. Normalisation et Équilibrage des Vecteurs

Lorsqu'on manipule des vecteurs $\mathbf{x}_i \in \mathbb{R}^d$ pour calculer la synergie $S(i, j)$, il arrive souvent que certaines **dimensions** varient sur des plages beaucoup plus larges que d'autres, ou que différentes entités \mathcal{E}_i n'utilisent pas le même **ordonnancement** de valeurs. Sans précautions, la mesure de distance ou de similarité peut s'en trouver **biaisée**. Deux mécanismes de base s'appliquent :

Normalisation en norme

Un classique consiste à normaliser chaque vecteur à la **norme 1**, c'est-à-dire :

$$\mathbf{x}'_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}, \quad \|\mathbf{x}'_i\| = 1.$$

Ainsi, la **similarité cosinus** entre \mathbf{x}'_i et \mathbf{x}'_j devient un **produit scalaire** direct. Cela annule l'influence de la **magnitude** brute et met l'accent sur l'**angle**. Dans un DSL multimodal, cette démarche évite qu'une modalité sortant des valeurs très élevées ne domine la mesure de proximité.

Normalisation par composante

On peut aussi appliquer un **z-score** (soustraction de la moyenne et division par l'écart-type), ou un min–max scaling dans l'intervalle [0,1], par dimension. Cette approche assure que chaque

coordonnée contribue équitablement. Dans un **SCN**, elle prévient la situation où une composante “géante” écrase toutes les autres dans le calcul de $\|\mathbf{x}_i - \mathbf{x}_j\|$ ou $\mathbf{x}_i \cdot \mathbf{x}_j$.

En pratique, la **choix** entre ces normalisations dépend de la distribution des données. L’objectif reste de **préserver** le signal discriminant et de **limiter** l’influence de différences d’échelle, améliorant ainsi la comparabilité et la **qualité** du calcul de $S(i, j)$.

B. Calibration du Score de Synergie

Une fois les vecteurs normalisés, on définit la synergie par :

$$S(i, j) = \phi(d(\mathbf{x}_i, \mathbf{x}_j)) \quad \text{ou} \quad \psi(\text{sim}(\mathbf{x}_i, \mathbf{x}_j)),$$

où ϕ et ψ sont des transformations (éventuellement exponentielles, sigmoïdes, inverses, etc.). Il se peut qu’on désire **calibrer** la sortie pour moduler l’amplitude. Par exemple :

Noyau gaussien

$$S(i, j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\alpha > 0$ gouverne la rapidité de décroissance. Si α est trop grand, la synergie chute quasi immédiatement pour tout écart modéré, regroupant uniquement les entités quasi identiques. Si α est trop petit, elle reste trop élevée pour un large spectre de distances, fusionnant potentiellement tout en un seul méga-cluster.

Distance inversée

$$S(i, j) = \frac{1}{1 + \beta d(\mathbf{x}_i, \mathbf{x}_j)},$$

où $\beta > 0$. Ici, plus la distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ grandit, plus S tend vers 0 mais jamais exactement 0. Le réglage de β dicte le “taux” d’atténuation.

Troncature ou saturation

On peut saturer la sortie pour ne pas dépasser 1 (ou un certain seuil), ou imposer un plancher minimal. Ainsi, la synergie reste dans $[\varepsilon, 1 - \varepsilon]$ afin d’éviter un renforcement ou un affaiblissement trop extrême. Ceci peut stabiliser la mise à jour :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)].$$

Le point crucial est de **calibrer** S de façon à refléter *juste assez* la disparité entre entités, sans aboutir à des valeurs saturées pour la plupart des paires (synergie trop extrême) ou trop faibles (peu de différenciation).

C. Prise en Compte du Bruit

Dans bien des environnements (capteurs, images à faible qualité, textes “sales” ou hétérogènes), les **données** sont bruitées. Ce bruit peut fausser le calcul de distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ ou de similarité $\mathbf{x}_i \cdot \mathbf{x}_j$. Plusieurs stratégies se dégagent :

Filtrage ou prétraitement. Avant tout calcul de distance, on peut nettoyer ou lisser \mathbf{x}_i , par exemple via du **smoothing** ou un **autoencodeur denoising**. Ce faisant, on réduit la variabilité de bas niveau, améliorant la robustesse de la synergie.

Troncature ou clipping des valeurs extrêmes. Si certaines composantes subissent des pics aberrants, on peut fixer un **cap** sur l'amplitude. Cela évite qu'une seule dimension bruitée ne domine la distance.

Injection de bruit contrôlé. Parfois, on ajoute soi-même un bruit gaussien au vecteur \mathbf{x}_i (ou pendant l'apprentissage) pour **stabiliser** l'auto-organisation, un peu à la manière d'un recuit simulé. Cette démarche évite de se figer dans des minima locaux, et autorise une **exploration** plus large du SCN. Au fil du temps, on diminue ce bruit pour **affiner** la convergence.

Robustesse des fonctions de distance. Au lieu de la distance euclidienne ℓ_2 , on peut employer ℓ_1 (manhattan), réputée moins sensible aux outliers sur une coordonnée. On peut également introduire des fonctions qui saturent au-delà d'une certaine différence, réduisant l'influence des valeurs extrêmes.

3.3.4. Gestion du Bruit et de l'Évolution des Embeddings

Même si les vecteurs et embeddings (section 3.3.3) constituent une base pratique pour représenter les entités dans un DSL (Deep Synergy Learning), ils ne sont pas figés. Dans la **réalité** d'un système en évolution où les capteurs changent, les nouvelles données s'intègrent et le modèle s'améliore continuellement, il faut prendre en compte le **bruit** potentiel dans les données ou dans les embeddings, pouvant entraîner des distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ artificiellement élevées ou basses.

Il est aussi essentiel de considérer la **mise à jour** ou le re-entraînement des modèles produisant les embeddings comme les réseaux neuronaux ou Transformers, ce qui peut modifier la représentation \mathbf{x}_i au fil du temps. Enfin, les **stratégies** de filtrage, de clipping et d'ajustement local sont cruciales pour éviter la propagation d'erreurs ou la dérive progressive.

Ainsi, dans cette section, nous verrons comment le DSL peut gérer ces aspects (bruit, évolution) et préserver la **cohérence** du calcul de synergie $S(i, j)$.

3.3.4.1. Embeddings Potentiellement Réentraînés, Fine-Tuned (Chap. 9 sur Apprentissage Continu)

Dans le cadre du **Deep Synergy Learning (DSL)**, les **embeddings** \mathbf{x}_i qui caractérisent les entités \mathcal{E}_i ne sont pas considérés comme des représentations statiques et définitives. En effet, ces vecteurs, générés par des modèles neuronaux tels que des **CNN**, des **Transformers** ou des **autoencodeurs**, peuvent évoluer au fil du temps lorsque le modèle générateur est soumis à un processus de **fine-tuning** ou à un schéma d'**apprentissage continu**.

Ce phénomène de mise à jour dynamique des embeddings influence directement la **synergie** $S(i, j)$ entre entités ainsi que la manière dont les **pondérations** $\omega_{i,j}$ sont ajustées dans le **Synergistic Connection Network (SCN)**.

A. Fine-Tuning et Mise à Jour des Embeddings

Considérons qu'à un instant t , chaque entité \mathcal{E}_i est décrite par un **embedding** $\mathbf{x}_i(t)$ obtenu via un modèle g_t , qui s'exprime par la relation

$$\mathbf{x}_i(t) = g_t(\mathbf{d}_i),$$

où \mathbf{d}_i représente les données brutes associées à l'entité, telles qu'une image, un texte ou un signal audio. Lorsqu'un processus de **fine-tuning** est initié – c'est-à-dire que le modèle g_t est réentraîné ou ajusté en réponse à de nouvelles données ou à un objectif révisé – le modèle évolue vers une nouvelle version g_{t+1} et l'embedding correspondant se met à jour selon

$$\mathbf{x}_i(t + 1) = g_{t+1}(\mathbf{d}_i).$$

Même une modification modeste des poids du modèle peut entraîner une transformation significative de l'espace vectoriel, modifiant ainsi les distances $\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|$ entre entités et, par conséquent, la valeur de la **synergie** $S(i, j)$. Ce réajustement induit une réorganisation du SCN, puisque la règle de mise à jour des pondérations

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta[S(i, j, t) - \tau \omega_{i,j}(t)],$$

dépend directement des représentations actuelles $\mathbf{x}_i(t)$ et $\mathbf{x}_j(t)$. Une telle dynamique impose une attention particulière dans la gestion des mises à jour pour ne pas perturber brusquement la structure des clusters déjà établis.

B. Apprentissage Continu et Flux Dynamique

Dans un scénario d'**apprentissage continu**, le modèle g_t est régulièrement mis à jour pour intégrer un nouveau flux de données, noté $\Delta\mathcal{D}$. La relation de mise à jour du modèle s'exprime alors par

$$g_{t+1} = \text{Train}(g_t, \Delta\mathcal{D}),$$

ce qui conduit à une évolution progressive des embeddings selon

$$\mathbf{x}_i(t + 1) = g_{t+1}(\mathbf{d}_i).$$

Ce mécanisme est particulièrement précieux dans des environnements non stationnaires où de nouvelles classes ou de nouvelles variations de données apparaissent de manière régulière. Toutefois, cette flexibilité s'accompagne d'une instabilité potentielle, puisque la transformation du modèle peut modifier de façon substantielle la géométrie de l'espace latent. Les distances entre les embeddings – et donc les scores de synergie – peuvent varier brutalement, induisant des réorganisations dans le SCN qui se traduisent par des fluctuations des pondérations $\omega_{i,j}$.

C. Impact sur la Mise à Jour des Pondérations

La dynamique d'auto-organisation du SCN est directement affectée par l'évolution des embeddings. La règle de mise à jour des pondérations

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta[S(i, j, t) - \tau \omega_{i,j}(t)]$$

utilise la **synergie** $S(i, j, t)$, qui dépend des embeddings à l'instant t . Si, suite à un fine-tuning ou à un apprentissage continu, les embeddings se modifient de manière significative, la valeur de $S(i, j, t + 1)$ peut être très différente de celle précédemment obtenue.

Ces variations peuvent provoquer des **sauts** ou des oscillations dans les pondérations, affectant la stabilité du regroupement des entités dans le réseau. Pour limiter ce phénomène, il est souvent recommandé d'adopter des stratégies de **lissage** dans la transition entre $\mathbf{x}_i(t)$ et $\mathbf{x}_i(t + 1)$, par exemple en définissant une mise à jour pondérée telle que

$$\mathbf{x}_i(t + 1) \leftarrow \alpha \mathbf{x}_i(t) + (1 - \alpha) g_{t+1}(\mathbf{d}_i),$$

où le paramètre $\alpha \in [0,1]$ contrôle l'ampleur de la transition, permettant ainsi d'introduire progressivement les changements et de préserver la cohérence de la dynamique du SCN.

D. Équilibre entre Adaptation et Stabilité

L'adaptabilité des embeddings grâce au fine-tuning et à l'apprentissage continu offre une capacité essentielle à un DSL, puisqu'elle permet au système de rester pertinent face à l'évolution des données. Toutefois, cette adaptabilité se heurte à la nécessité de maintenir une stabilité dans l'organisation des entités. Un modèle qui se modifie trop fréquemment risque de déstabiliser les **clusters** existants, car les pondérations $\omega_{i,j}$ basées sur d'anciennes représentations ne seront plus en adéquation avec la nouvelle configuration de l'espace latent.

Un compromis se trouve souvent dans la planification de mises à jour épisodiques, suivies de périodes de stabilisation, ou dans l'application de techniques d'adaptation incrémentale qui permettent un ajustement progressif. L'ingénieur se doit de régler la fréquence et l'ampleur des mises à jour de g_t afin de concilier la **richesse contextuelle** des nouvelles représentations et la nécessité d'une **stabilité** du SCN.

E. Gestion Concrète : Références au Chapitre 9 sur l'Apprentissage Continu

Le **chapitre 9** du présent ouvrage se penche en profondeur sur les techniques d'**apprentissage continu** et les stratégies permettant de gérer les mises à jour des embeddings sans compromettre la cohérence globale du DSL. Il y est présenté des protocoles d'adaptation incrémentale, des heuristiques de recalibrage des pondérations, ainsi que des mécanismes visant à atténuer le phénomène de **catastrophic forgetting**.

Ces approches incluent notamment l'actualisation progressive des représentations, le verrouillage temporaire de certaines couches du modèle, et l'indexation régulière des embeddings afin d'assurer que la transition entre différentes versions du modèle reste la plus fluide possible.

3.3.4.2. Filtrage Local (k-NN) ou Clipping pour Limiter la Propagation d'Erreurs

Dans le cadre d'un **Deep Synergy Learning (DSL)**, la qualité et la stabilité de la **synergie** entre entités reposent sur la précision des **embeddings** et des **scores de similarité** calculés à partir de ceux-ci.

Cependant, en présence de **perturbations** telles que le bruit, des valeurs aberrantes ou des variations imprévues dans les données, il devient indispensable de mettre en place des **mécanismes** visant à limiter la propagation des erreurs qui pourraient déstabiliser l'ensemble du **Synergistic Connection Network (SCN)**.

Deux approches complémentaires se distinguent dans cette optique : le **filtrage local** par l'algorithme des **k-plus-proches-voisins (k-NN)** et le **clipping** des vecteurs ou des scores de similarité. Ces stratégies, en restreignant l'influence d'entités extrêmes ou aberrantes, contribuent à préserver la stabilité globale du système.

A. Filtrage Local via k-Plus-Proches-Voisins (k-NN)

Lorsqu'une entité \mathcal{E}_i est caractérisée par un **embedding** $\mathbf{x}_i \in \mathbb{R}^d$, il est fréquent, dans un SCN, de calculer la **synergie** $S(i, j)$ entre toutes les paires (i, j) . Cette approche, qui aboutit à un graphe complet, peut introduire des liaisons peu informatives ou même trompeuses, notamment lorsque des perturbations locales affectent certains embeddings.

Afin de remédier à ce problème, il est judicieux de restreindre le calcul de la synergie à un **voisinage** immédiat, en considérant uniquement les k entités les plus proches de \mathbf{x}_i . Formellement, le voisinage local d'une entité se définit par

$$\text{NN}_k(i) = \{j \in \{1, \dots, n\} \mid j \text{ fait partie des } k \text{ plus proches entités de } \mathbf{x}_i\}.$$

En conséquence, la fonction de synergie est tronquée de manière à ce que

$$S'(i, j) = \begin{cases} S(i, j), & \text{si } j \in \text{NN}_k(i) \text{ ou } i \in \text{NN}_k(j), \\ 0, & \text{sinon,} \end{cases}$$

ce qui signifie que seules les entités jugées suffisamment proches contribuent à la mise à jour des **pondérations** $\omega_{i,j}$. L'**avantage** de cette approche réside dans la réduction de la complexité, passant d'un nombre de comparaisons de l'ordre de $O(n^2)$ à $O(n k)$, tout en **limitant** l'impact des **outliers** qui, étant isolés, ne figurent pas dans le voisinage local.

En outre, cette stratégie contribue à renforcer la **robustesse** du DSL en ne construisant des liaisons significatives qu'entre des entités réellement compatibles dans l'espace \mathbb{R}^d .

B. Clipping des Vecteurs et des Scores de Similarité

En complément du filtrage local, le **clipping** constitue une méthode efficace pour restreindre l'influence d'**valeurs extrêmes**. Cette technique peut s'appliquer à différents niveaux du processus d'inférence. Lorsqu'il s'agit des **embeddings** eux-mêmes, il est souvent pertinent de contraindre chaque coordonnée du vecteur \mathbf{x}_i afin d'éviter que des perturbations locales ne provoquent une domination de certaines dimensions. Pour ce faire, on impose un seuil $T > 0$ et on définit le vecteur « clippé » par

$$x_{i,k}^{\text{clip}} = \min\{\max\{x_{i,k}, -T\}, T\}.$$

Cette opération garantit que les valeurs de chaque dimension restent contenues dans l'intervalle $[-T, T]$, ce qui permet de préserver une **stabilité** géométrique dans \mathbb{R}^d . Par ailleurs, il est également possible d'appliquer le clipping directement au **score de synergie**. Si, par exemple, la

fonction $S(i, j)$ calcule une similarité qui pourrait parfois dépasser une borne raisonnable à cause d'un outlier, on peut définir une version clipée de cette synergie par

$$S'(i, j) = \min\{S(i, j), S_{\max}\},$$

où S_{\max} est une borne supérieure prédéfinie. Cette approche empêche que la **synergie** n'atteigne des valeurs trop élevées qui pourraient induire un renforcement disproportionné des pondérations $\omega_{i,j}$, préservant ainsi l'équilibre global du SCN.

C. Combinaison des Approches pour une Robustesse Accrue

Dans la pratique, l'application conjointe du **filtrage local** par k-NN et du **clipping** constitue une stratégie robuste pour limiter la propagation d'erreurs. En effet, le filtrage local restreint d'abord le calcul de la synergie aux voisins les plus proches, réduisant la densité du graphe et la probabilité d'étendre l'influence d'entités aberrantes. Par la suite, le clipping, appliqué sur les vecteurs ou directement sur les scores de similarité, permet de modérer l'impact de toute valeur extrême qui pourrait encore se glisser dans le processus. Le SCN met ainsi à jour les pondérations selon la formule

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S'(i, j) - \tau \omega_{i,j}(t)],$$

où $S'(i, j)$ représente la synergie obtenue après application des mécanismes de filtrage et de clipping. Cette approche intégrée permet de réduire la complexité computationnelle, en limitant le nombre de comparaisons nécessaires, tout en préservant la stabilité de l'auto-organisation en atténuant l'impact des perturbations locales.

3.3.4.3. Exemples de Scénarios Audio-Visuels

Dans le cadre du **Deep Synergy Learning (DSL)**, les **applications audio-visuelles** offrent un terrain d'expérimentation particulièrement riche pour illustrer la capacité du système à **auto-organiser** des entités issues de modalités diverses telles que l'image, le son et le texte.

Le **Synergistic Connection Network (SCN)** se doit de prendre en compte des **embeddings** extraits de flux multimodaux qui, par leur nature, peuvent être affectés par le **bruit**, des changements de **contexte** ou des ajustements successifs induits par un fine-tuning continu.

Dans cette section, nous exposons en détail deux scénarios qui démontrent comment le DSL exploite des mécanismes de limitation d'erreurs, tels que le filtrage local et le clipping, ainsi que des stratégies d'adaptation dynamique, afin de préserver une organisation robuste et cohérente dans un environnement audio-visuel.

A. Détection d'Événements Audio-Visuels

Dans un premier exemple, nous considérons la détection d'un événement, tel qu'un applaudissement ou un klaxon, qui se manifeste simultanément dans la composante visuelle et acoustique d'un flux multimodal. Pour ce faire, le flux audio-visuel est segmenté en une série de **frames vidéo** $\{\mathcal{E}_i^{(vid)}\}$ et en une série de **segments audio** $\{\mathcal{E}_j^{(aud)}\}$. Chaque frame est alors transformé en un **embedding** $\mathbf{x}_i^{(vid)} \in \mathbb{R}^{d_{\text{vid}}}$ par un modèle de type CNN (tel que ResNet) ou par

un Vision Transformer, tandis que chaque segment audio se voit attribuer un vecteur $\mathbf{x}_j^{(\text{aud})} \in \mathbb{R}^{d_{\text{aud}}}$ issu d'un modèle d'autoencodeur appliqué sur un spectrogramme ou d'un transformeur audio tel que wav2vec. La **synergie** entre une frame vidéo et un segment audio peut être quantifiée à l'aide d'un noyau gaussien, défini par

$$S(i, j) = \exp(-\gamma \|\mathbf{x}_i^{(\text{vid})} - \mathbf{x}_j^{(\text{aud})}\|^2),$$

où $\gamma > 0$ ajuste la sensibilité de la mesure.

Toutefois, en raison des perturbations inhérentes à ce type de flux – telles que la saturation audio ou les variations lumineuses dans les images –, des valeurs aberrantes peuvent fausser le calcul de la synergie.

Pour limiter ces effets, le DSL intègre des mécanismes de **filtrage local** en restreignant le calcul de la similarité aux paires d'entités se chevauchant temporellement, ainsi que des techniques de **clipping** qui borner les valeurs extrêmes du score $S(i, j)$. Ce filtrage permet de constituer des **clusters** robustes reliant les frames et segments audio fortement synchronisés, ce qui facilite la détection d'événements multimodaux avec une grande précision.

B. Sous-Titrage Automatique à Partir d'un Flux Audio-Visuel

Dans un second scénario, le DSL est appliquée à la tâche de sous-titrage automatique, qui implique la fusion d'informations issues du flux vidéo et du flux textuel dérivé d'une transcription audio.

Dans ce contexte, chaque frame vidéo $\mathcal{E}_i^{(\text{vid})}$ se voit attribuer un embedding $\mathbf{x}_i^{(\text{vid})}$ généré par un Vision Transformer ou un CNN, tandis que la transcription issue d'un module speech-to-text est convertie en un **embedding textuel** $\mathbf{x}_j^{(\text{text})} \in \mathbb{R}^{d_{\text{text}}}$ via un modèle de type BERT ou GPT. La synergie entre la représentation visuelle et celle textuelle peut être évaluée à l'aide de la **similarité cosinus**, exprimée par

$$S_{\cos}(i, j) = \frac{\mathbf{x}_i^{(\text{vid})} \cdot \mathbf{x}_j^{(\text{text})}}{\|\mathbf{x}_i^{(\text{vid})}\| \|\mathbf{x}_j^{(\text{text})}\|}.$$

Cette mesure permet de quantifier la proximité sémantique entre le contenu visuel et le texte associé. Cependant, des erreurs peuvent survenir dans la transcription, notamment en cas de bruit, d'accents ou de débit de parole rapide, tandis que la qualité des embeddings vidéo peut être affectée par des mouvements brusques ou des changements de scène.

Afin de pallier ces imprécisions, le DSL met en place des mécanismes de **limitation temporelle** (en ne comparant que les frames proches du segment textuel) ainsi que des procédures de **clipping** pour éviter que des valeurs extrêmes ne conduisent à des scores de similarité erronés. De cette manière, le système parvient à associer de manière cohérente les segments vidéo et les extraits textuels, facilitant ainsi la génération de sous-titres ou de résumés contextuels de haute qualité.

C. Ajustement Dynamique et Fine-Tuning

Dans les deux scénarios précédents, le DSL peut être soumis à des processus de **fine-tuning** ou à des mises à jour continues, de sorte que les modèles générateurs d'embeddings (qu'il s'agisse d'un

CNN, d'un Vision Transformer ou d'un modèle speech-to-text) évoluent en fonction des nouvelles données.

Ainsi, les représentations \mathbf{x}_i et \mathbf{x}_j peuvent être réévaluées périodiquement, modifiant la synergie $S(i,j)$ calculée et, par conséquent, les pondérations $\omega_{i,j}$ dans le SCN.

Afin d'éviter que ces changements brusques ne perturbent la dynamique globale du réseau, des techniques de **lissage** des transitions et de **verrouillage** partiel des modèles sont mises en œuvre. Ces stratégies visent à adapter les embeddings de manière progressive, garantissant ainsi une réorganisation continue mais stable des clusters d'entités multimodales.

3.3. Représentations Sub-Symboliques

Les approches sub-symboliques, basées sur des **vecteurs** ou des **embeddings**, constituent l'une des manières les plus répandues de décrire des entités \mathcal{E}_i dans un contexte d'apprentissage automatique.

Dans le cadre du DSL (Deep Synergy Learning), elles présentent l'avantage de se prêter à un **calcul** aisément quantifiable de la similarité (voire de la distance) entre entités, conditionnant ainsi la **synergie** $S(i, j)$.

Cette section, nous étudierons tout d'abord les **vecteurs** et **embeddings** (3.3.1), puis nous aborderons des **méthodes avancées** (autoencodeurs, transformers, etc.) (3.3.2), avant de voir en détail le **calcul** de la synergie entre vecteurs (3.3.3) et la **gestion** du bruit ou de l'évolution de ces représentations (3.3.4).

3.3.1. Vecteurs et Embeddings

La représentation **vectorielle** est sans doute la plus intuitive dans un grand nombre d'applications où chaque entité est codée par un point $\mathbf{x}_i \in \mathbb{R}^d$. Cette approche, déjà ancienne en classification ou en clustering, prend une **nouvelle dimension** avec l'émergence des **embeddings profonds**, capables de saisir des caractéristiques complexes dans les images, les textes ou l'audio.

3.3.1.1. Origine : CNN (images), Word Embeddings (textes), Spectrogrammes (audio)

Dans le cadre d'un **Deep Synergy Learning (DSL)**, une grande partie des représentations sub-symboliques est obtenue par l'extraction d'**embeddings** à l'aide de **modèles neuronaux** spécialisés. L'idée fondamentale consiste à transformer un objet initial – qu'il s'agisse d'une image, d'un texte ou d'un segment audio – en un **vecteur** ou un **tenseur** qui capture les **caractéristiques** les plus pertinentes de l'information. Ce vecteur permet ensuite de calculer la **mesure** de synergie, notée $S(i, j)$, qui reflète la proximité ou la compatibilité entre deux entités \mathcal{E}_i et \mathcal{E}_j . Diverses approches sont utilisées pour générer ces embeddings, et nous détaillons ici trois cas typiques.

Dans le domaine de l'analyse d'images, les **Convolutional Neural Networks (CNN)** ont largement popularisé l'extraction de **vecteurs** de caractéristiques. Un réseau convolutionnel, tel que VGG, ResNet, Inception ou encore Vision Transformer, procède à travers une série de couches de **convolution** et de **pooling** à l'extraction progressive de « feature maps » qui condensent l'information visuelle en attributs de plus en plus abstraits, tels que les **textures**, les **bords** ou encore les **formes**. Au terme de ce processus, on obtient un **embedding global** de dimension d (typiquement 256, 512 ou 1024), représenté par

$$\text{CNN}: \quad \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^d,$$

où $H \times W$ indique la taille de l'image, C le nombre de canaux (par exemple, 3 pour une image RGB), et d la dimension du vecteur.

Ce **descripteur** capture ainsi des aspects sémantiques essentiels, et dans un DSL, deux images dont les embeddings \mathbf{x}_i et \mathbf{x}_j se montrent très similaires (par exemple, mesurées via la **similarité cosinus** ou une distance euclidienne inversée) conduisent à une synergie élevée, entraînant le renforcement de la pondération $\omega_{i,j}$ dans le SCN et favorisant l'auto-organisation en clusters d'images visuellement cohérents.

Dans le domaine du **Traitement Automatique du Langage Naturel (TALN)**, la représentation des textes s'appuie sur les **word embeddings** tels que Word2Vec et GloVe, ainsi que sur des modèles contextuels plus récents comme BERT, GPT ou T5. Plutôt que d'utiliser des représentations statiques comme les vecteurs one-hot, on projette chaque mot ou token dans un **espace** de dimension d selon la relation

$$\mathbf{w}: \{\text{mots ou tokens}\} \rightarrow \mathbb{R}^d.$$

Ainsi, deux mots ou expressions d'une **signification** similaire se trouvent naturellement proches dans cet espace, ce qui facilite le calcul de distances ou de **similarités**. De plus, pour obtenir une représentation d'une phrase ou d'un document, il est courant d'agréger les embeddings des mots, par exemple par la moyenne ou en extrayant le vecteur associé au token [CLS] dans les modèles Transformer. Cette approche permet d'assigner à chaque segment textuel un vecteur \mathbf{x}_i qui, lorsqu'il est comparé à un autre vecteur \mathbf{x}_j , donne une mesure de synergie reflétant la **parenté sémantique** entre les textes.

Pour les données **audio**, l'extraction d'embeddings se fait souvent à partir d'un **spectrogramme** ou par l'usage de modèles neuronaux spécialisés, tels que SoundNet ou des Audio Transformers. Le signal audio $a(t)$ est segmenté et transformé par un modèle spécifique, conduisant à un embedding

$$\mathbf{x}_s = \text{AudioModel}(a_s(t)) \in \mathbb{R}^d,$$

qui capture des caractéristiques acoustiques comme le **timbre** et la **prosodie**. Dans un DSL, si deux segments audio présentent des embeddings similaires, la synergie $S(i,j)$ est alors élevée, et la pondération correspondante $\omega_{i,j}$ se renforce, permettant ainsi la formation de clusters cohérents de signaux acoustiques ou la fusion d'une modalité audio avec d'autres modalités dans un cadre multimodal.

D'un point de vue **formel**, après avoir extrait les vecteurs \mathbf{x}_i pour chaque entité, on définit la **mesure** de similarité par exemple par la formule gaussienne

$$S(i,j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|)$$

ou par la formule du **produit scalaire normalisé**

$$S(i,j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}.$$

Lorsque ces scores de similarité sont intégrés dans la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

le DSL renforce les liens entre les entités dont les embeddings sont proches dans l'espace, ce qui conduit à la formation de **clusters** auto-organisés reflétant la **cohérence** des caractéristiques sous-jacentes.

La **force** de cette approche réside dans la capacité des **embeddings** générés par ces modèles neuronaux à capturer une **richesse sémantique** et des aspects contextuels qui permettent d'identifier avec précision les similitudes entre entités issues de domaines variés (vision, langage, acoustique).

Cependant, le succès de cette méthode dépend fortement de la **qualité** des embeddings, laquelle doit être suffisamment robuste pour résister au bruit et capable de généraliser les caractéristiques essentielles des données.

Ces principes, approfondis dans le Chapitre 3, soulignent l'importance de disposer de représentations vectorielles de haute qualité pour garantir une auto-organisation efficace au sein du DSL.

3.3.1.2. Avantages : Calcul Aisé de Similarité (Cosinus, Distance Euclidienne)

Dans le cadre du **Deep Synergy Learning (DSL)**, chaque entité \mathcal{E}_i est représentée par un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$ qui est issu d'un modèle d'apprentissage profond tel qu'un CNN, un transformeur pour le langage ou encore un modèle audio pour les spectrogrammes.

L'un des atouts majeurs de cette représentation sub-symbolique réside dans la **simplicité** algorithmique avec laquelle il est possible de calculer la **similarité** ou la **distance** entre ces vecteurs.

Cette facilité de calcul joue un rôle central dans la formation et l'auto-organisation du **Synergistic Connection Network (SCN)**, puisque la **synergie** $S(i,j)$ entre deux entités est directement fonction de la proximité mesurée dans l'espace vectoriel, et plus cette proximité est élevée, plus la **pondération** $\omega_{i,j}$ tend à être renforcée selon la règle de mise à jour du SCN.

D'un point de vue mathématique, la **similarité cosinus** est définie par

$$S_{\cos}(i,j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|},$$

cette mesure se concentrant sur l'**angle** entre \mathbf{x}_i et \mathbf{x}_j plutôt que sur leur norme, ce qui est particulièrement pertinent dans des contextes où la magnitude du vecteur n'est pas un indicateur de la **pertinence** ou de la **similarité**. En outre, une transformation linéaire de l'intervalle $[-1,1]$ permet d'obtenir un score de similarité dans $[0,1]$, par exemple

$$S(i,j) = \frac{1 + S_{\cos}(i,j)}{2},$$

ce qui garantit une interprétation directe dans le cadre de la mise à jour des pondérations. Par ailleurs, la **distance euclidienne** est définie par

$$d_{\text{eucl}}(i, j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2},$$

et cette distance, en quantifiant la longueur du segment reliant \mathbf{x}_i à \mathbf{x}_j , peut être convertie en un score de similarité par l'application d'une fonction de décroissance exponentielle, comme le montre la formule

$$S_{\text{gauss}}(i, j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\alpha > 0$ est un paramètre réglable. Ces deux formules sont linéaires en la dimension d et reposent sur des opérations de multiplication–accumulation, qui sont bien adaptées aux architectures de calcul modernes telles que les GPU ou les bibliothèques optimisées en BLAS, permettant ainsi un **calcul** rapide même pour de grands ensembles d'entités. La **simplicité** de ces mesures permet une implémentation efficace de la **mise à jour** des pondérations dans le SCN selon la formule

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

ce qui conduit à un renforcement des liens entre les entités qui se rapprochent dans l'espace vectoriel. L'avantage de cette approche réside également dans le fait qu'elle permet une **interopérabilité** naturelle avec un grand nombre d'algorithme existants, notamment ceux de **clustering** (tels que K-means, DBSCAN) et des techniques d'**approximate nearest neighbor** (comme FAISS ou Annoy), ce qui facilite l'analyse et la visualisation des regroupements obtenus via des méthodes de réduction de dimension telles que PCA, t-SNE ou UMAP.

De plus, cette méthode se prête aisément à des scénarios **multimodaux** où, en projetant des données provenant de différentes sources dans un espace commun, on applique de manière uniforme la logique de calcul de la similarité, assurant ainsi une intégration homogène et une gestion efficace de la **variabilité** des données.

Finalement, en ajustant les paramètres des kernels (comme α dans le cas du noyau gaussien), il est possible d'optimiser la **sensibilité** du DSL aux distances entre vecteurs, garantissant ainsi que la **dynamique** de mise à jour des pondérations reflète de manière fine les structures sous-jacentes des données.

3.5.3.3. Inconvénients : Dimension Élevée, Parfois Peu Interprétables

L'utilisation de **représentations sub-symboliques** sous forme de vecteurs d'embeddings, générés par des modèles neuronaux tels que les CNN ou les Transformers, offre indéniablement des avantages en termes de simplicité et d'efficacité du calcul de la synergie dans \mathbb{R}^d . Toutefois, ces méthodes présentent plusieurs inconvénients qui peuvent poser des problèmes de **scalabilité**, de **stabilité** et d'**explicabilité**. La discussion qui suit s'articule autour de quatre axes majeurs.

A. Dimension Élevée : Risques et Limitations

Les modèles neuronaux modernes produisent des embeddings dont la dimension peut atteindre plusieurs centaines voire milliers (par exemple, 512, 768 ou 1024 dimensions). Lorsque la fonction

de synergie $\mathbf{S}(i, j)$ est calculée de manière naïve pour toutes les paires (i, j) au sein d'un Synergistic Connection Network (SCN) composé de n entités, le coût de calcul devient de l'ordre de $O(n^2 \times d)$. Ainsi, si n est très grand – par exemple, plusieurs millions d'entités – et d élevé, le temps de calcul peut devenir prohibitif, comme l'illustre l'estimation

$$\text{Coût} \approx n^2 \times d.$$

De plus, la malédiction de la dimension peut rendre les distances moins discriminantes, puisque dans des espaces de très haute dimension, les distances entre la plupart des points tendent à converger vers une valeur similaire. Cette uniformisation des distances peut aboutir à une dynamique d'auto-organisation moins stable, avec des clusters qui se forment de manière peu distincte ou qui présentent des frontières floues.

Face à ces difficultés, des approches telles que l'utilisation d'**algorithmes d'approximate nearest neighbors**, de techniques de **sparsification** ou de mécanismes de **sélection** ne traitant que les paires prometteuses sont souvent nécessaires afin de réduire la charge de calcul.

B. Opacité et Faible Interprétabilité

Les embeddings issus de réseaux neuronaux sont le produit d'opérations complexes et hiérarchiques réparties sur de multiples couches (convolutions, mécanismes d'attention, etc.). Cette complexité rend souvent l'interprétation de chaque coordonnée de l'embedding difficile. Il n'est pas aisément de déterminer si une dimension particulière de \mathbf{x}_i correspond à une caractéristique précise, telle qu'une texture, une forme ou un concept lexical.

De surcroît, lorsqu'on constate qu'un score de similarité élevé existe entre deux vecteurs \mathbf{x}_i et \mathbf{x}_j , il est ardu d'attribuer de manière explicite la cause de cette proximité – qu'il s'agisse de la couleur, de la catégorie ou d'une co-occurrence textuelle. Cette opacité pose un réel problème dans des domaines exigeant une forte **explicabilité** ; ainsi, pour des applications critiques (médicales, industrielles, etc.), il est impératif de pouvoir justifier la formation d'un cluster ou la liaison entre deux entités.

Même si des techniques de réduction de dimension, telles que PCA, UMAP ou t-SNE, permettent d'obtenir une vue globale des distributions, elles n'améliorent guère l'interprétation des attributs à l'échelle individuelle.

C. Instabilité des Embeddings et Variations Contextuelles

Les embeddings neuronaux peuvent varier sensiblement en fonction du **contexte** ou des modifications apportées lors de la phase d'entraînement. Dans le domaine du traitement du langage naturel, par exemple, une légère modification du corpus d'entraînement ou un nouveau fine-tuning peut entraîner des décalages dans les positions des mots ou des tokens dans l'espace vectoriel.

De même, en vision par ordinateur, l'adaptation d'un réseau pour intégrer de nouvelles classes d'images peut modifier la distribution finale des embeddings \mathbf{x}_i . De telles variations contextuelles impactent directement la mesure de la synergie, que ce soit par la modification de la distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ ou par le produit scalaire $\mathbf{x}_i \cdot \mathbf{x}_j$.

Cette instabilité peut ainsi perturber la dynamique du SCN, induisant des réorganisations brutales ou imprévues dans l'auto-organisation, et posant un défi pour la stabilité à long terme du DSL.

D. Nécessité d'un Prétraitement Potentiellement Coûteux

L'obtention de ces embeddings repose sur l'utilisation de modèles préentraînés, tels que des réseaux CNN pour les images ou des Transformers pour les textes, qui nécessitent une infrastructure de calcul robuste (GPU/TPU) et des ressources importantes pour l'entraînement ou l'inférence.

Si l'objectif est de concevoir un DSL autonome fonctionnant sur des ressources matérielles limitées, le coût en termes de calcul et de temps associé au prétraitement des données peut s'avérer prohibitif. De plus, le déploiement d'un tel système requiert la mise en place d'un pipeline complet, capable de fournir en continu des embeddings de haute qualité, ce qui représente une contrainte supplémentaire en termes d'ingénierie et d'infrastructure.

3.3.2. Autoencodeurs, Transformers et Approches Avancées

Les vecteurs et embeddings présentés en section 3.3.1 constituent déjà une base solide pour représenter des entités de nature variée (images, textes, sons). Toutefois, les **techniques** d'apprentissage ne cessant d'évoluer, il est possible d'aller **plus loin** en recourant à des méthodes avancées qui extraient des embeddings encore plus **expressifs** ou plus **compacts**.

Dans cette section, nous verrons d'abord (3.3.2.1) comment les **autoencodeurs** peuvent servir à la **réduction dimensionnelle** ou à **l'extraction de features**, puis (3.3.2.2) comment les **Transformers** (BERT, GPT, ViT) produisent des embeddings **contextuels**, et enfin (3.3.2.3) nous soulignerons **l'intérêt** de ces méthodes avancées pour le DSL, au prix toutefois d'une complexité de calcul plus élevée.

3.3.2.1. Autoencodeurs pour Réduction Dimensionnelle ou Extraction de Features

Dans le cadre d'un **Deep Synergy Learning (DSL)**, l'extraction d'informations pertinentes à partir de représentations sub-symboliques constitue un enjeu majeur pour l'efficacité du processus d'auto-organisation.

Les **autoencodeurs (AE)** se présentent comme un outil fondamental permettant d'apprendre, de manière non supervisée, un **code latent \mathbf{z}** qui capture l'essence d'un vecteur d'entrée \mathbf{x} . Cette démarche offre deux avantages essentiels pour le DSL où elle permet la **réduction de dimension** et **l'extraction de features discriminantes**.

A. Principe et Architecture d'un Autoencodeur

Un autoencodeur se compose typiquement de deux modules complémentaires.

Le premier, l'**encodeur E** , est une fonction $E: \mathbb{R}^d \rightarrow \mathbb{R}^k$ qui reçoit un vecteur d'entrée $\mathbf{x} \in \mathbb{R}^d$ et produit un **code latent $\mathbf{z} \in \mathbb{R}^k$** , où $k < d$ dans le but de compresser l'information.

Le second module, le **décodeur D** , opère la transformation inverse, $D: \mathbb{R}^k \rightarrow \mathbb{R}^d$, et tente de reconstruire le vecteur d'entrée à partir de ce code latent, produisant ainsi $\hat{\mathbf{x}} = D(E(\mathbf{x}))$.

L'apprentissage se réalise en minimisant une **fonction de perte de reconstruction** $\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}})$, souvent formulée comme la moyenne des erreurs quadratiques, ce qui s'exprime par

$$\min_{E, D} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(\mathbf{x}_i, D(E(\mathbf{x}_i))).$$

Cette optimisation force l'encodeur à extraire les facteurs de variation les plus pertinents de \mathbf{x} , tout en éliminant le bruit et les détails superflus, aboutissant ainsi à une représentation compacte et significative.

B. Avantages : Réduction de Dimension et Extraction de Traits

L'imposition d'un goulot d'étranglement, où la dimension du code latent k est inférieure à celle de l'entrée d , engendre deux bénéfices notables pour le DSL.

D'une part, la **réduction dimensionnelle** permet de diminuer significativement le coût de calcul lors de l'évaluation des distances ou similarités entre entités. En effet, le calcul de la distance entre deux codes latents, par exemple $\|\mathbf{z}_i - \mathbf{z}_j\|$, se réalise en $O(k)$ plutôt qu'en $O(d)$, ce qui est particulièrement avantageux lorsque le réseau traite un grand nombre d'entités.

D'autre part, en apprenant à reconstruire \mathbf{x} à partir de \mathbf{z} , l'autoencodeur extrait des **features discriminantes** qui résument les aspects sémantiques les plus importants du vecteur d'origine, facilitant ainsi la formation de clusters cohérents et la détection de similarités véritables dans le SCN.

C. Variantes : Denoising, Sparse et Variational Autoencoder

Il existe plusieurs variantes d'autoencodeurs qui visent à améliorer la robustesse et la qualité des représentations latentes.

Premièrement, le **Denoising Autoencoder** introduit du bruit dans \mathbf{x} durant l'entraînement et apprend à reconstruire la version nettoyée de la donnée. Ce procédé confère au code latent \mathbf{z} une meilleure tolérance aux perturbations, ce qui est particulièrement utile dans des environnements où les données sont bruitées.

Ensuite, le **Sparse Autoencoder** incorpore une contrainte de sparsité sur \mathbf{z} – par exemple, via une pénalisation de la norme L_1 – de sorte que seule une fraction limitée des neurones s'active pour chaque donnée, ce qui peut améliorer l'interprétabilité des caractéristiques extraites et favoriser une séparation plus nette des classes.

Enfin, le **Variational Autoencoder (VAE)** adopte une approche probabiliste en imposant que le code latent suive une distribution prédéfinie, généralement gaussienne, c'est-à-dire $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$. Cette formulation permet non seulement de générer de nouvelles données, mais aussi d'organiser l'espace latent de manière plus lisse, facilitant ainsi la continuité et la robustesse des clusters formés lors de l'auto-organisation.

D. Intégration au DSL : La Synergie entre Codes Latents

Une fois l'autoencodeur entraîné, la partie **encodeur** E est utilisée pour transformer chaque vecteur d'entrée \mathbf{x}_i en un code latent $\mathbf{z}_i = E(\mathbf{x}_i)$ dans un espace de dimension réduite \mathbb{R}^k . Ce code latent remplace alors \mathbf{x}_i dans la phase de calcul de la synergie au sein du DSL. La mesure de synergie peut être définie par des formules classiques telles que

$$\mathbf{S}(i, j) = \exp(-\alpha \parallel \mathbf{z}_i - \mathbf{z}_j \parallel^2)$$

ou par une mesure basée sur la similarité cosinus

$$\mathbf{S}(i, j) = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\parallel \mathbf{z}_i \parallel \parallel \mathbf{z}_j \parallel}.$$

Ces scores de synergie, calculés dans un espace de dimension réduite, facilitent le renforcement des pondérations $\omega_{i,j}$ et la formation de clusters dans le Synergistic Connection Network. L'optimisation de la dimension k joue alors un rôle critique où si k est trop petit, l'autoencodeur risque de perdre des informations essentielles, tandis qu'un k trop élevé n'apporte pas le gain de complexité recherché.

E. Limites et Points de Vigilance

Malgré leurs avantages, les autoencodeurs présentent quelques inconvénients notables.

Premièrement, la minimisation de la **perte de reconstruction** n'est pas nécessairement alignée avec l'objectif de séparer distinctement les classes ou de maximiser la discrimination entre les entités. Par conséquent, les codes latents \mathbf{z}_i peuvent parfois ne pas être suffisamment discriminants pour la tâche d'auto-organisation.

Deuxièmement, l'entraînement d'un autoencodeur sur des datasets volumineux peut être coûteux en termes de ressources computationnelles et de temps, notamment lorsqu'il s'agit d'architectures profondes.

Troisièmement, le choix de la dimension latente k est crucial où une valeur inappropriée peut soit sous-estimer les facteurs de variation essentiels, soit ne pas fournir une véritable réduction de complexité. Enfin, dans un contexte d'apprentissage continu, la mise à jour de l'encodeur E doit être soigneusement gérée pour éviter le phénomène de **catastrophic forgetting**, qui pourrait altérer l'organisation de l'espace latent déjà établi.

3.3.2.2. Transformers (BERT, GPT, ViT) : Embeddings Contextuels plus Riches

L'émergence des **Transformers** – tels que BERT, GPT, T5 en NLP et Vision Transformer (ViT) en vision – a profondément transformé la manière dont sont générés les **embeddings** dans divers domaines. Contrairement aux représentations plus statiques (comme Word2Vec) ou aux approches purement convolutionnelles (CNN), les Transformers produisent des **embeddings contextuels**.

Ces représentations intègrent non seulement l'information d'un token ou d'un patch de manière isolée, mais elles capturent également le **contexte global** par le biais d'un mécanisme de *self-*

attention. Dans le cadre d'un **Deep Synergy Learning (DSL)**, cette capacité à incorporer le contexte se traduit par des mesures de synergie plus fines et expressives, où la proximité entre deux entités reflète non seulement des similarités de caractéristiques isolées, mais aussi des interactions complexes au sein de l'ensemble des données.

A. Principe des Transformers et Self-Attention

L'architecture Transformer repose sur un mécanisme de *self-attention* qui permet à chaque position d'une séquence de « regarder » toutes les autres positions. Soit une séquence d'entrées représentées par la matrice $\mathbf{x} \in \mathbb{R}^{n \times d}$, où chaque ligne correspond à un embedding initial d'un token ou d'un patch. Les Transformers calculent trois matrices essentielles par multiplication avec des poids appris :

$$\mathbf{Q} = \mathbf{x} W_Q, \quad \mathbf{K} = \mathbf{x} W_K, \quad \mathbf{V} = \mathbf{x} W_V,$$

où W_Q , W_K , et W_V sont des matrices de projection de dimensions appropriées (souvent $d \times d_k$). Le mécanisme de self-attention se définit alors par l'opération

$$\text{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}.$$

Cette opération permet à chaque token (ou patch) de pondérer les contributions de tous les autres éléments en fonction de la similitude entre ses **requêtes** et les **clés** associées aux autres tokens, produisant ainsi des embeddings qui intègrent l'information contextuelle de l'ensemble de la séquence.

B. BERT, GPT : Embeddings Contextuels en NLP

Dans le domaine du **Traitemment Automatique du Langage Naturel (NLP)**, des modèles tels que **BERT** et **GPT** exploitent pleinement le mécanisme de self-attention pour produire des **embeddings contextuels**. Concrètement, après plusieurs couches de self-attention et de transformations non linéaires, chaque token i de la séquence obtient une représentation finale $\mathbf{h}_i^{(L)}$. Deux types d'extraction sont couramment utilisées :

- **Token-level** : Chaque mot ou sous-mot se voit attribuer un embedding \mathbf{h}_i qui capture ses nuances contextuelles.
- **Embedding global** : Un vecteur représentatif de l'ensemble de la séquence est généré, souvent en utilisant le token spécial $[CLS]$ (dans le cas de BERT) ou par agrégation (moyenne ou max) sur tous les tokens.

Dans un DSL, ces embeddings globaux servent de « signature » vectorielle pour une entité textuelle. La synergie entre deux entités \mathcal{E}_i et \mathcal{E}_j peut ainsi être évaluée par des mesures telles que la similarité cosinus :

$$S(i, j) = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\|\mathbf{z}_i\| \|\mathbf{z}_j\|},$$

où \mathbf{z}_i et \mathbf{z}_j sont les embeddings finaux issus du Transformer. Grâce à la prise en compte du contexte, ces représentations permettent de discerner des relations sémantiques fines, telles que la synonymie contextuelle ou les paraphrases, améliorant ainsi la qualité de l'auto-organisation du réseau.

C. Vision Transformer (ViT) : Patches et Self-Attention en Vision

Les principes du Transformer se sont également appliqués au domaine de la vision par ordinateur via le **Vision Transformer (ViT)**. Dans ce cas, une image est divisée en petits **patches** (par exemple, de 16×16 pixels), qui sont linéarisés et projetés dans un espace de dimension d .

Pour conserver l'information spatiale, des **positional embeddings** sont ajoutés à chaque patch. Le Transformer traite alors l'ensemble des patches de manière similaire aux tokens en NLP, utilisant le mécanisme de self-attention pour capturer des relations globales dans l'image.

La sortie finale peut être constituée d'un token spécial [CLS] agissant comme représentation globale ou d'une agrégation des embeddings individuels. Cette approche permet de générer des embeddings d'images qui intègrent non seulement des caractéristiques locales, mais aussi des dépendances à longue portée entre différentes régions de l'image, améliorant ainsi la mesure de la synergie dans le DSL.

D. Enjeux et Coûts : Dimension et Ressources

Bien que les Transformers produisent des embeddings contextuels particulièrement riches, ils impliquent également des coûts importants en termes de **dimension** et de **ressources**.

Par exemple, les modèles tels que BERT ou GPT génèrent des vecteurs de dimension 768, 1024 ou plus, ce qui peut amplifier la complexité des calculs de similarité dans un SCN, souvent de l'ordre de $O(n^2 \times d)$ pour n entités.

De plus, le poids des modèles (parfois plusieurs centaines de millions de paramètres) requiert une infrastructure matérielle (GPU, TPU) conséquente pour l'inférence et le fine-tuning. Enfin, la sensibilité des Transformers aux ajustements fins – un léger changement de paramètres peut modifier la distribution des embeddings – nécessite des techniques d'alignement ou de versionnage pour préserver la stabilité des représentations dans un système d'apprentissage continu.

E. Intégration dans un DSL

L'intégration des Transformers dans un DSL se réalise en extrayant, pour chaque entité \mathcal{E}_i (qu'il s'agisse d'un document, d'une image ou d'un segment audio), un embedding contextuel \mathbf{z}_i via une opération telle que

$$\mathbf{z}_i = \text{TransformerEncode}(\mathbf{x}_i),$$

où \mathbf{x}_i représente l'objet brut initial et `TransformerEncode` désigne le processus de passage par plusieurs couches de self-attention et de feed-forward. Une fois obtenu, l'embedding peut être normalisé afin de faciliter le calcul de similarités, par exemple en utilisant le cosinus. La synergie entre deux entités est alors définie par une fonction f (comme le cosinus ou un noyau RBF) :

$$S(i, j) = f(\mathbf{z}_i, \mathbf{z}_j).$$

Dans le processus de mise à jour des pondérations du SCN,

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

une synergie élevée entre des embeddings contextuels conduira au renforcement des liens, favorisant ainsi la formation de clusters d'entités sémantiquement cohérents, tout en capturant des relations complexes que seules des approches traditionnelles auraient du mal à discerner.

3.3.2.3. Intérêt pour le DSL : un Embedding plus Robuste, mais plus Complexe à Calculer ou à Mettre à Jour

Dans le cadre d'un **Deep Synergy Learning (DSL)**, l'emploi de techniques avancées telles que les **Transformers** ou les **autoencodeurs profonds** permet d'obtenir des **embeddings** d'une qualité remarquable, caractérisés par une richesse contextuelle et une capacité de discrimination supérieure. Ces représentations, issues de modèles sophistiqués, se distinguent par leur aptitude à capter des nuances sémantiques fines et à structurer l'espace latent de manière à favoriser la formation de clusters d'entités plus robustes. On définit notamment, par une fonction g , l'embedding contextuel d'une entité initiale \mathbf{x}_i par

$$\mathbf{z}_i = g(\mathbf{x}_i),$$

ce qui permet de mesurer la **synergie** entre deux entités \mathcal{E}_i et \mathcal{E}_j à l'aide d'une distance ou d'un produit scalaire, par exemple

$$S(i,j) = \exp(-\alpha \parallel \mathbf{z}_i - \mathbf{z}_j \parallel^2) \quad \text{ou} \quad S(i,j) = \frac{\mathbf{z}_i \cdot \mathbf{z}_j}{\parallel \mathbf{z}_i \parallel \parallel \mathbf{z}_j \parallel},$$

ce qui permet de regrouper les entités ayant des caractéristiques latentes similaires au sein du **Synergistic Connection Network (SCN)**. Ces méthodes confèrent au DSL une **robustesse** accrue dans la capture des variations contextuelles et une capacité à distinguer subtilement des différences sémantiques que des approches plus simples auraient tendance à négliger.

Cependant, la qualité supérieure des embeddings contextuels se traduit par une **complexité** notable tant dans le calcul de l'inférence que dans la gestion des mises à jour de ces représentations.

Le modèle g , en intégrant plusieurs couches de self-attention et de transformations non linéaires, nécessite en général une infrastructure matérielle puissante, telle que des GPU ou TPU, en raison du nombre élevé de paramètres et de la dimension des vecteurs, souvent de l'ordre de 768, 1024 ou plus.

Par conséquent, le coût de calcul d'une opération d'inférence, qui peut être évalué en $O(n^2 \times d_{latent})$ pour un ensemble de n entités, augmente considérablement, en particulier lorsque le DSL doit traiter de grands volumes de données en temps réel.

Cette complexité algorithmique se trouve également exacerbée par le coût associé à la mise à jour des embeddings dans un contexte d'apprentissage continu. En effet, si le modèle g subit un fine-tuning ou une révision partielle, il devient nécessaire de recalculer les embeddings pour toutes les entités, c'est-à-dire

$$\mathbf{z}_i(t+1) = g_{t+1}(\mathbf{x}_i) \quad \text{pour } i = 1, \dots, n,$$

ce qui, pour des ensembles de données de grande taille, peut engendrer une latence non négligeable et provoquer des fluctuations dans les valeurs de $S(i, j)$ entre les cycles de mise à jour.

Ces variations, même modestes, peuvent induire des réorganisations brutales au sein du SCN, compromettant ainsi la stabilité de l'auto-organisation.

L'ingénierie d'un DSL doit donc établir un compromis entre la **richesse** des embeddings, qui améliore la précision des regroupements en capturant des informations contextuelles complexes, et l'**efficacité computationnelle** requise pour leur calcul et leur mise à jour régulière.

Dans certains contextes critiques, où la qualité de la représentation est primordiale – par exemple en médecine ou en analyse de données sensibles – il peut être acceptable de supporter des coûts de calcul élevés pour garantir une **précision** et une **robustesse** accrues.

À l'inverse, dans des applications massives où la rapidité d'inférence est essentielle, des modèles plus légers ou des embeddings de dimension réduite pourraient être privilégiés afin d'optimiser la scalabilité du système.

L'utilisation d'embeddings plus riches, issus de techniques avancées, permet d'améliorer significativement la **discrimination** et la **robustesse** des synergies dans un DSL, mais se fait au prix d'une complexité computationnelle accrue et d'un défi supplémentaire en termes de mise à jour des représentations dans un cadre d'apprentissage continu.

Ce compromis entre la richesse des représentations et l'efficacité opérationnelle constitue un enjeu central dans la conception et l'implémentation de systèmes de Deep Synergy Learning.

3.3.3. Calcul de la Synergie entre Vecteurs

Dans les sections précédentes (3.3.1 et 3.3.2), nous avons mis en évidence l'usage de **vecteurs** (embeddings) pour représenter des entités dans un contexte sub-symbolique, ainsi que l'apport de méthodes avancées (autoencodeurs, Transformers) pour produire des embeddings plus riches. Il reste à voir **comment**, à partir de ces vecteurs, on définit la **synergie** $S(i, j)$ entre deux entités \mathcal{E}_i et \mathcal{E}_j . En pratique, le DSL (Deep Synergy Learning) recourt souvent à des **fonctions** de distance ou de similarité géométriques. Cette section (3.3.3) se divise en trois points :

- (3.3.3.1) présentation des **distances usuelles** (euclidienne, manhattan) et de la **similarité cosinus**,
- (3.3.3.2) usage de **kernels** pour une similarité non linéaire,
- (3.3.3.3) **ajustements** (normalisation, calibration) pour mieux gérer le bruit ou l'échelle des vecteurs.

3.3.3.1. Distances Usuelles : Euclidienne, Manhattan, Cosinus

Dans le cadre du **Deep Synergy Learning (DSL)**, chaque entité \mathcal{E}_i est représentée par un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$. L'évaluation de la **synergie** entre deux entités repose alors sur la comparaison de ces

représentations via une fonction de **distance** ou de **similarité**. Ces mesures jouent un rôle fondamental dans le processus d'**auto-organisation** du **Synergistic Connection Network (SCN)**, puisque plus deux vecteurs sont « proches » dans l'espace, plus la **pondération** $\omega_{i,j}$ tend à être renforcée. Parmi les approches classiques pour mesurer cette proximité, on trouve la **distance euclidienne**, la **distance manhattan** et la **similarité cosinus**, chacune offrant une interprétation géométrique différente et répondant à des exigences spécifiques selon le type de données traitées.

A. Distance Euclidienne

La **distance euclidienne** constitue la mesure la plus intuitive, puisqu'elle correspond à la longueur du segment reliant les points représentés par \mathbf{x}_i et \mathbf{x}_j dans \mathbb{R}^d . Mathématiquement, elle se définit par

$$d_{\text{eucl}}(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}.$$

La valeur ainsi obtenue représente une notion de **distance** physique dans l'espace, et dans le DSL, cette distance est souvent convertie en un **score de similarité** afin de faciliter son intégration dans la dynamique du SCN. Une fonction d'agrégation classique consiste à appliquer une décroissance exponentielle, ce qui donne

$$\mathbf{S}(i,j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\alpha > 0$ est un **paramètre** qui ajuste la pente de décroissance. Dans ce contexte, une distance faible entre \mathbf{x}_i et \mathbf{x}_j conduit à un score proche de 1, indiquant une **affinité élevée** entre les deux entités et justifiant ainsi le renforcement de la pondération $\omega_{i,j}$.

B. Distance Manhattan (ou ℓ_1)

La **distance manhattan** ou distance **taxicab** s'exprime en prenant la somme des différences absolues entre les composantes correspondantes des vecteurs. La formule est donnée par

$$d_{\text{manh}}(i,j) = \sum_{k=1}^d |x_{i,k} - x_{j,k}|.$$

Cette mesure, qui évalue la différence cumulée par dimension, est particulièrement utile lorsque les données se combinent de façon additive, et elle présente l'avantage de se montrer souvent plus **robuste** face aux valeurs extrêmes de certaines coordonnées. Pour intégrer cette distance dans un DSL, on peut transformer la mesure en un score de similarité en appliquant par exemple la fonction exponentielle

$$\mathbf{S}(i,j) = \exp(-\beta d_{\text{manh}}(i,j)),$$

ou encore en utilisant la forme rationnelle

$$\mathbf{S}(i,j) = \frac{1}{1 + d_{\text{manh}}(i,j)},$$

où $\beta > 0$ est un **paramètre** qui ajuste le taux de décroissance. De cette manière, une distance manhattan réduite se traduit par un score de similarité élevé, indiquant une forte **affinité** entre les entités.

C. Similarité Cosinus

La **similarité cosinus** se distingue des mesures de distance classiques en se focalisant sur l'**angle** entre deux vecteurs plutôt que sur leur distance absolue. Elle est définie par

$$\mathbf{S}_{\cos}(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|},$$

et prend des valeurs dans l'intervalle $[-1, 1]$. Cette mesure atteint la valeur 1 lorsque les vecteurs sont colinéaires dans la même direction, -1 lorsqu'ils sont opposés et 0 lorsque les vecteurs sont orthogonaux. Dans le cadre du DSL, il est souvent souhaitable d'obtenir un score strictement positif, et pour ce faire, on peut transformer l'intervalle en appliquant par exemple la formule

$$\mathbf{S}(i, j) = \frac{1 + \mathbf{S}_{\cos}(i, j)}{2},$$

qui convertit les valeurs de $[-1, 1]$ en $[0, 1]$. Cette méthode permet de neutraliser l'effet de la norme des vecteurs, en se concentrant uniquement sur leur **direction**, ce qui est particulièrement utile pour des données textuelles ou pour des représentations où la magnitude peut varier de manière significative.

D. Liaison avec la Synergie dans le DSL

L'ensemble des mesures présentées permet de définir la **synergie** entre deux entités, laquelle est utilisée pour ajuster la **pondération** $\omega_{i,j}$ dans le SCN. La règle de mise à jour se formalise par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [\mathbf{S}(i, j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ est le **taux d'apprentissage** et $\tau > 0$ représente le **coefficent de décroissance**. Ce mécanisme d'auto-organisation implique que plus deux entités sont évaluées comme étant proches (ou similaires) selon la mesure choisie, plus la pondération $\omega_{i,j}$ tend à augmenter, favorisant ainsi la formation de **clusters** cohérents. Le choix entre la **distance euclidienne**, la **distance manhattan** ou la **similarité cosinus** dépend du type de données et des propriétés spécifiques recherchées, car chacune offre des avantages distincts. La distance euclidienne fournit une mesure classique de la proximité, la distance manhattan peut mieux gérer certaines disparités dimensionnelles et la similarité cosinus permet de se focaliser sur l'orientation des vecteurs indépendamment de leur norme.

3.3.3.2. Kernels (RBF, Polynomial) pour une Similarité non Linéaire

Les **distances** usuelles (euclidienne, manhattan) ou la **similarité cosinus** (section 3.3.3.1) considèrent des relations souvent **linéaires** ou du moins directes dans l'espace vectoriel. Cependant, il arrive que la **relation** réelle entre deux entités \mathcal{E}_i et \mathcal{E}_j soit plus subtile, nécessitant des méthodes **non linéaires** pour en rendre compte. Dans un **Deep Synergy Learning (DSL)**,

recourir à des **kernels** (noyaux) permet d'introduire des notions de **similarité** bien plus riches que les simples distances ℓ_p . Les **RBF kernels** (noyaux gaussiens) et **kernels polynomiaux** figurent parmi les plus répandus et se traduisent en un **score** $S(i, j)$ apte à révéler des liens complexes cachés dans l'espace initial \mathbb{R}^d .

A. Principe Général des Kernels

Un **kernel** k entre deux vecteurs $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^d$ se définit comme une **fonction** :

$$k: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R},$$

renvoyant un score (positif ou non) quantifiant leur **affinité**. La **particularité** du kernel trick, issu de la théorie des **noyaux de Mercer**, réside dans le fait que $k(\mathbf{x}_i, \mathbf{x}_j)$ peut être interprété comme un **produit scalaire** dans un espace de dimension éventuellement infinie :

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}},$$

où $\phi: \mathbb{R}^d \rightarrow \mathcal{H}$ est une **transformation** non linéaire, et \mathcal{H} un **espace** (souvent de Hilbert) de dimension potentiellement énorme. La **magie** du kernel trick est qu'on n'a pas besoin de **calculer** explicitement ϕ . Il suffit d'évaluer $k(\mathbf{x}_i, \mathbf{x}_j)$. D'un point de vue **DSL**, cela équivaut à définir la **synergie** $S(i, j)$:

$$S(i, j) = k(\mathbf{x}_i, \mathbf{x}_j).$$

Deux entités peuvent se trouver “lointaines” selon une métrique linéaire, tout en étant **fortement similaires** en termes de transformations non linéaires. Dans un **Synergistic Connection Network**, on peut alors capter des formes de **ressemblance** plus complexes, sans explicitement projeter les entités dans un espace étendu.

B. RBF Kernel (Radial Basis Function)

Le **kernel RBF** (aussi appelé “noyau gaussien”) constitue un choix très répandu. Il s’écrit :

$$k_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (\gamma > 0).$$

Il s’agit d’une **exponentielle** négative de la distance euclidienne au carré. On peut l’interpréter comme une “gaussienne” centrée sur \mathbf{x}_j . Plus \mathbf{x}_i est proche de \mathbf{x}_j , plus la valeur se rapproche de 1 ; plus ils sont éloignés, plus elle tend vers 0. Le **paramètre** γ détermine la “largeur” du noyau :

- Un γ **grand** induit une décroissance rapide de $k_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j)$ dès que $\|\mathbf{x}_i - \mathbf{x}_j\|$ grandit, ce qui focalise la similarité sur une zone très **locale**.
- Un γ **petit** rend la fonction plus “plate”, donnant une vue plus **globale** où des entités modérément éloignées conservent une similarité notable.

Pour un **DSL**, un **RBF kernel** donne une **synergie** :

$$S(i, j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2).$$

Cette forme **non linéaire** permet à des entités ayant un écart euclidien significatif (dans \mathbb{R}^d) de tout de même afficher une similarité mesurable si elles partagent certaines **propriétés** sous-jacentes. En outre, si γ est bien paramétré, la synergie reflète de subtiles variations dans les données — plus subtiles, parfois, qu'une simple distance euclidienne.

C. Polynomial Kernel

Un autre kernel usuel est le **kernel polynomial** :

$$k_{\text{poly}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^p,$$

où p est le **degré** (typiquement 2, 3, 4...) et $c \geq 0$ un terme de décalage. Cette définition incorpore toutes les **composantes** polynomiales du produit $\mathbf{x}_i \cdot \mathbf{x}_j$, allant jusqu'au degré p . Ainsi, pour $p = 2$, on inclut toutes les interactions bilinéaires $(x_{i,k}x_{j,\ell})$. Cela peut capturer des **corrélations** plus complexes qu'un simple "angle" ou qu'une distance linéaire.

Dans un **SCN**, si on adopte $S(i, j) = k_{\text{poly}}(\mathbf{x}_i, \mathbf{x}_j)$, on se retrouve avec une **synergie** qui peut s'avérer très sensible à des variations dans certaines dimensions, surtout pour un p élevé. Cela peut amplifier des **propriétés** souhaitées (s'il y a de fortes corrélations polynomiales), mais risque aussi de **surexposer** le bruit ou les valeurs extrêmes. Le choix de p et c réclame donc un réglage attentif.

D. Non-Linéarité et Espace Implicit : Kernel Trick

Le **kernel trick** signifie que l'on n'a **pas** besoin de définir explicitement une transformation $\phi(\cdot)$ vers un espace de dimension (parfois énorme) \mathcal{H} . Il suffit de calculer $k(\mathbf{x}_i, \mathbf{x}_j)$. Dans un **DSL**, cela se traduit par une **synergie** :

$$S(i, j) = k(\mathbf{x}_i, \mathbf{x}_j) \quad \text{où } k \text{ est RBF, polynomial, ou autre.}$$

Mathématiquement, deux vecteurs lointains en \mathbb{R}^d peuvent devenir "proches" dans l'espace \mathcal{H} , si la transformation ϕ induite par le kernel met en évidence des attributs partagés. Ainsi, la **mise à jour** $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \dots$ tient compte de propriétés non linéaires. Dans la **formation** de clusters, des entités qui ne semblaient pas se ressembler au sens euclidien peuvent se rapprocher si leur **synergie** kernel s'avère élevée.

E. Exemples d'Implémentation dans un SCN

Pour un **SCN** gérant n entités, on définit $S(i, j) = k(\mathbf{x}_i, \mathbf{x}_j)$. À chaque itération, calculer $\omega_{i,j}(t+1)$ exige d'évaluer ces kernels. Sur le plan **coût** :

- **RBF** : $O(d)$ multiplications pour $\|\mathbf{x}_i - \mathbf{x}_j\|^2$, puis une exponentiation.
- **Polynomial** : $O(d)$ pour le produit scalaire $\mathbf{x}_i \cdot \mathbf{x}_j$, puis l'élévation au degré p .

Pour comparer toutes les paires (i, j) , on fait face à $O(n^2)$ évaluations, chaque en $O(d)$. Si n est grand, cela devient lourd (voir chap. 3.2.2.2). En pratique, on peut élaguer en ne calculant la synergie que sur un **voisinage** restreint (approximate nearest neighbor, etc.) ou recourir à d'autres heuristiques d'échantillonnage pour gérer la **scalabilité**.

F. Limites et Paramétrage Sensible

Le **RBF kernel** nécessite le choix d'un paramètre γ . Un γ trop grand produit une décroissance exponentielle ultra-rapide, rendant la synergie presque nulle pour la majorité des paires, ce qui peut isoler excessivement les entités. Un γ trop petit, en revanche, aboutit à un noyau très plat, de sorte que presque tout le monde se retrouve "similaire" : on aboutit à un **mégacluster**. Le **polynomial kernel** demande aussi de fixer p et c . Un p trop élevé peut amplifier les bruits, alors qu'un p trop faible (ex. 1) redescend à un comportement linéaire classique.

L'ingénieur d'un **DSL** doit donc **régler** ces hyperparamètres en fonction des données, par validation croisée ou d'autres heuristiques. Les kernels offrent une flexibilité précieuse, mais nécessitent une **phase d'ajustement** pour qu'ils produisent des **synergies** stables et cohérentes dans le SCN.

3.3.3.3. Ajustements : Normalisation, Calibration, Prise en Compte du Bruit

Dans un **Deep Synergy Learning (DSL)** où les entités \mathcal{E}_i sont décrites par des **vecteurs** ou des attributs numériques, l'évaluation de la **synergie** $S(i, j)$ repose sur des **distances** (euclidienne, manhattan) ou des **similarités** (cosinus, kernels). Toutefois, dans la **pratique**, il ne suffit pas de choisir une métrique : il faut aussi **calibrer** et **ajuster** la façon dont on compare les vecteurs. Les questions de normalisation, d'échelle, de saturation, ou encore de **prise en compte** du bruit sont essentielles pour garantir la **cohérence** du calcul de $S(i, j)$ et la stabilité de la mise à jour $\omega_{i,j}$. Les sections qui suivent décrivent en détail comment on gère ces ajustements pour un **Synergistic Connection Network (SCN)** robuste et efficace.

A. Normalisation et Équilibrage des Vecteurs

Lorsqu'on manipule des vecteurs $\mathbf{x}_i \in \mathbb{R}^d$ pour calculer la synergie $S(i, j)$, il arrive souvent que certaines **dimensions** varient sur des plages beaucoup plus larges que d'autres, ou que différentes entités \mathcal{E}_i n'utilisent pas le même **ordonnancement** de valeurs. Sans précautions, la mesure de distance ou de similarité peut s'en trouver **biaisée**. Deux mécanismes de base s'appliquent :

Normalisation en norme

Un classique consiste à normaliser chaque vecteur à la **norme 1**, c'est-à-dire :

$$\mathbf{x}'_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}, \quad \|\mathbf{x}'_i\| = 1.$$

Ainsi, la **similarité cosinus** entre \mathbf{x}'_i et \mathbf{x}'_j devient un **produit scalaire** direct. Cela annule l'influence de la **magnitude** brute et met l'accent sur l'**angle**. Dans un DSL multimodal, cette démarche évite qu'une modalité sortant des valeurs très élevées ne domine la mesure de proximité.

Normalisation par composante

On peut aussi appliquer un **z-score** (soustraction de la moyenne et division par l'écart-type), ou un min–max scaling dans l'intervalle [0,1], par dimension. Cette approche assure que chaque

coordonnée contribue équitablement. Dans un **SCN**, elle prévient la situation où une composante “géante” écrase toutes les autres dans le calcul de $\|\mathbf{x}_i - \mathbf{x}_j\|$ ou $\mathbf{x}_i \cdot \mathbf{x}_j$.

En pratique, la **choix** entre ces normalisations dépend de la distribution des données. L’objectif reste de **préserver** le signal discriminant et de **limiter** l’influence de différences d’échelle, améliorant ainsi la comparabilité et la **qualité** du calcul de $S(i, j)$.

B. Calibration du Score de Synergie

Une fois les vecteurs normalisés, on définit la synergie par :

$$S(i, j) = \phi(d(\mathbf{x}_i, \mathbf{x}_j)) \quad \text{ou} \quad \psi(\text{sim}(\mathbf{x}_i, \mathbf{x}_j)),$$

où ϕ et ψ sont des transformations (éventuellement exponentielles, sigmoïdes, inverses, etc.). Il se peut qu’on désire **calibrer** la sortie pour moduler l’amplitude. Par exemple :

Noyau gaussien

$$S(i, j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\alpha > 0$ gouverne la rapidité de décroissance. Si α est trop grand, la synergie chute quasi immédiatement pour tout écart modéré, regroupant uniquement les entités quasi identiques. Si α est trop petit, elle reste trop élevée pour un large spectre de distances, fusionnant potentiellement tout en un seul méga-cluster.

Distance inversée

$$S(i, j) = \frac{1}{1 + \beta d(\mathbf{x}_i, \mathbf{x}_j)},$$

où $\beta > 0$. Ici, plus la distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ grandit, plus S tend vers 0 mais jamais exactement 0. Le réglage de β dicte le “taux” d’atténuation.

Troncature ou saturation

On peut saturer la sortie pour ne pas dépasser 1 (ou un certain seuil), ou imposer un plancher minimal. Ainsi, la synergie reste dans $[\varepsilon, 1 - \varepsilon]$ afin d’éviter un renforcement ou un affaiblissement trop extrême. Ceci peut stabiliser la mise à jour :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)].$$

Le point crucial est de **calibrer** S de façon à refléter *juste assez* la disparité entre entités, sans aboutir à des valeurs saturées pour la plupart des paires (synergie trop extrême) ou trop faibles (peu de différenciation).

C. Prise en Compte du Bruit

Dans bien des environnements (capteurs, images à faible qualité, textes “sales” ou hétérogènes), les **données** sont bruitées. Ce bruit peut fausser le calcul de distance $\|\mathbf{x}_i - \mathbf{x}_j\|$ ou de similarité $\mathbf{x}_i \cdot \mathbf{x}_j$. Plusieurs stratégies se dégagent :

Filtrage ou prétraitement. Avant tout calcul de distance, on peut nettoyer ou lisser \mathbf{x}_i , par exemple via du **smoothing** ou un **autoencodeur denoising**. Ce faisant, on réduit la variabilité de bas niveau, améliorant la robustesse de la synergie.

Troncature ou clipping des valeurs extrêmes. Si certaines composantes subissent des pics aberrants, on peut fixer un **cap** sur l'amplitude. Cela évite qu'une seule dimension bruitée ne domine la distance.

Injection de bruit contrôlé. Parfois, on ajoute soi-même un bruit gaussien au vecteur \mathbf{x}_i (ou pendant l'apprentissage) pour **stabiliser** l'auto-organisation, un peu à la manière d'un recuit simulé. Cette démarche évite de se figer dans des minima locaux, et autorise une **exploration** plus large du SCN. Au fil du temps, on diminue ce bruit pour **affiner** la convergence.

Robustesse des fonctions de distance. Au lieu de la distance euclidienne ℓ_2 , on peut employer ℓ_1 (manhattan), réputée moins sensible aux outliers sur une coordonnée. On peut également introduire des fonctions qui saturent au-delà d'une certaine différence, réduisant l'influence des valeurs extrêmes.

3.3.4. Gestion du Bruit et de l'Évolution des Embeddings

Même si les vecteurs et embeddings (section 3.3.3) constituent une base pratique pour représenter les entités dans un DSL (Deep Synergy Learning), ils ne sont pas figés. Dans la **réalité** d'un système en évolution où les capteurs changent, les nouvelles données s'intègrent et le modèle s'améliore continuellement, il faut prendre en compte le **bruit** potentiel dans les données ou dans les embeddings, pouvant entraîner des distances $\|\mathbf{x}_i - \mathbf{x}_j\|$ artificiellement élevées ou basses.

Il est aussi essentiel de considérer la **mise à jour** ou le re-entraînement des modèles produisant les embeddings comme les réseaux neuronaux ou Transformers, ce qui peut modifier la représentation \mathbf{x}_i au fil du temps. Enfin, les **stratégies** de filtrage, de clipping et d'ajustement local sont cruciales pour éviter la propagation d'erreurs ou la dérive progressive.

Ainsi, dans cette section, nous verrons comment le DSL peut gérer ces aspects (bruit, évolution) et préserver la **cohérence** du calcul de synergie $S(i, j)$.

3.3.4.1. Embeddings Potentiellement Réentraînés, Fine-Tuned (Chap. 9 sur Apprentissage Continu)

Dans le cadre du **Deep Synergy Learning (DSL)**, les **embeddings** \mathbf{x}_i qui caractérisent les entités \mathcal{E}_i ne sont pas considérés comme des représentations statiques et définitives. En effet, ces vecteurs, générés par des modèles neuronaux tels que des **CNN**, des **Transformers** ou des **autoencodeurs**, peuvent évoluer au fil du temps lorsque le modèle générateur est soumis à un processus de **fine-tuning** ou à un schéma d'**apprentissage continu**. Ce phénomène de mise à jour dynamique des embeddings influence directement la **synergie** $S(i, j)$ entre entités ainsi que la manière dont les **pondérations** $\omega_{i,j}$ sont ajustées dans le **Synergistic Connection Network (SCN)**.

A. Fine-Tuning et Mise à Jour des Embeddings

Considérons qu'à un instant t , chaque entité \mathcal{E}_i est décrite par un **embedding** $\mathbf{x}_i(t)$ obtenu via un modèle g_t , qui s'exprime par la relation

$$\mathbf{x}_i(t) = g_t(\mathbf{d}_i),$$

où \mathbf{d}_i représente les données brutes associées à l'entité, telles qu'une image, un texte ou un signal audio. Lorsqu'un processus de **fine-tuning** est initié – c'est-à-dire que le modèle g_t est réentraîné ou ajusté en réponse à de nouvelles données ou à un objectif révisé – le modèle évolue vers une nouvelle version g_{t+1} et l'embedding correspondant se met à jour selon

$$\mathbf{x}_i(t + 1) = g_{t+1}(\mathbf{d}_i).$$

Même une modification modeste des poids du modèle peut entraîner une transformation significative de l'espace vectoriel, modifiant ainsi les distances $\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|$ entre entités et, par conséquent, la valeur de la **synergie** $S(i, j)$. Ce réajustement induit une réorganisation du SCN, puisque la règle de mise à jour des pondérations

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta[S(i, j, t) - \tau \omega_{i,j}(t)],$$

dépend directement des représentations actuelles $\mathbf{x}_i(t)$ et $\mathbf{x}_j(t)$. Une telle dynamique impose une attention particulière dans la gestion des mises à jour pour ne pas perturber brusquement la structure des clusters déjà établis.

B. Apprentissage Continu et Flux Dynamique

Dans un scénario d'**apprentissage continu**, le modèle g_t est régulièrement mis à jour pour intégrer un nouveau flux de données, noté $\Delta\mathcal{D}$. La relation de mise à jour du modèle s'exprime alors par

$$g_{t+1} = \text{Train}(g_t, \Delta\mathcal{D}),$$

ce qui conduit à une évolution progressive des embeddings selon

$$\mathbf{x}_i(t + 1) = g_{t+1}(\mathbf{d}_i).$$

Ce mécanisme est particulièrement précieux dans des environnements non stationnaires où de nouvelles classes ou de nouvelles variations de données apparaissent de manière régulière. Toutefois, cette flexibilité s'accompagne d'une instabilité potentielle, puisque la transformation du modèle peut modifier de façon substantielle la géométrie de l'espace latent. Les distances entre les embeddings – et donc les scores de synergie – peuvent varier brutalement, induisant des réorganisations dans le SCN qui se traduisent par des fluctuations des pondérations $\omega_{i,j}$.

C. Impact sur la Mise à Jour des Pondérations

La dynamique d'auto-organisation du SCN est directement affectée par l'évolution des embeddings. La règle de mise à jour des pondérations

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta[S(i, j, t) - \tau \omega_{i,j}(t)]$$

utilise la **synergie** $S(i, j, t)$, qui dépend des embeddings à l'instant t . Si, suite à un fine-tuning ou à un apprentissage continu, les embeddings se modifient de manière significative, la valeur de $S(i, j, t + 1)$ peut être très différente de celle précédemment obtenue. Ces variations peuvent provoquer des **sauts** ou des oscillations dans les pondérations, affectant la stabilité du regroupement des entités dans le réseau. Pour limiter ce phénomène, il est souvent recommandé d'adopter des stratégies de **lissage** dans la transition entre $\mathbf{x}_i(t)$ et $\mathbf{x}_i(t + 1)$, par exemple en définissant une mise à jour pondérée telle que

$$\mathbf{x}_i(t + 1) \leftarrow \alpha \mathbf{x}_i(t) + (1 - \alpha) g_{t+1}(\mathbf{d}_i),$$

où le paramètre $\alpha \in [0, 1]$ contrôle l'ampleur de la transition, permettant ainsi d'introduire progressivement les changements et de préserver la cohérence de la dynamique du SCN.

D. Équilibre entre Adaptation et Stabilité

L'adaptabilité des embeddings grâce au fine-tuning et à l'apprentissage continu offre une capacité essentielle à un DSL, puisqu'elle permet au système de rester pertinent face à l'évolution des données. Toutefois, cette adaptabilité se heurte à la nécessité de maintenir une stabilité dans l'organisation des entités.

Un modèle qui se modifie trop fréquemment risque de déstabiliser les **clusters** existants, car les pondérations $\omega_{i,j}$ basées sur d'anciennes représentations ne seront plus en adéquation avec la nouvelle configuration de l'espace latent.

Un compromis se trouve souvent dans la planification de mises à jour épisodiques, suivies de périodes de stabilisation, ou dans l'application de techniques d'adaptation incrémentale qui permettent un ajustement progressif. L'ingénieur se doit de régler la fréquence et l'ampleur des mises à jour de g_t afin de concilier la **richesse contextuelle** des nouvelles représentations et la nécessité d'une **stabilité** du SCN.

E. Gestion Concrète : Références au Chapitre 9 sur l'Apprentissage Continu

Le **chapitre 9** du présent ouvrage se penche en profondeur sur les techniques d'**apprentissage continu** et les stratégies permettant de gérer les mises à jour des embeddings sans compromettre la cohérence globale du DSL.

Il y est présenté des protocoles d'adaptation incrémentale, des heuristiques de recalibrage des pondérations, ainsi que des mécanismes visant à atténuer le phénomène de **catastrophic forgetting**. Ces approches incluent notamment l'actualisation progressive des représentations, le verrouillage temporaire de certaines couches du modèle, et l'indexation régulière des embeddings afin d'assurer que la transition entre différentes versions du modèle reste la plus fluide possible.

3.3.4.2. Filtrage Local (k-NN) ou Clipping pour Limiter la Propagation d'Erreurs

Dans le cadre d'un **Deep Synergy Learning (DSL)**, la qualité et la stabilité de la **synergie** entre entités reposent sur la précision des **embeddings** et des **scores de similarité** calculés à partir de ceux-ci.

Cependant, en présence de **perturbations** telles que le bruit, des valeurs aberrantes ou des variations imprévues dans les données, il devient indispensable de mettre en place des **mécanismes** visant à limiter la propagation des erreurs qui pourraient déstabiliser l'ensemble du **Synergistic Connection Network (SCN)**.

Deux approches complémentaires se distinguent dans cette optique : le **filtrage local** par l'algorithme des **k-plus-proches-voisins (k-NN)** et le **clipping** des vecteurs ou des scores de similarité.

Ces stratégies, en restreignant l'influence d'entités extrêmes ou aberrantes, contribuent à préserver la stabilité globale du système.

A. Filtrage Local via k-Plus-Proches-Voisins (k-NN)

Lorsqu'une entité \mathcal{E}_i est caractérisée par un **embedding** $\mathbf{x}_i \in \mathbb{R}^d$, il est fréquent, dans un SCN, de calculer la **synergie** $S(i, j)$ entre toutes les paires (i, j) . Cette approche, qui aboutit à un graphe complet, peut introduire des liaisons peu informatives ou même trompeuses, notamment lorsque des perturbations locales affectent certains embeddings. Afin de remédier à ce problème, il est judicieux de restreindre le calcul de la synergie à un **voisinage** immédiat, en considérant uniquement les k entités les plus proches de \mathbf{x}_i .

Le voisinage local d'une entité se définit par

$$\text{NN}_k(i) = \{j \in \{1, \dots, n\} \mid j \text{ fait partie des } k \text{ plus proches entités de } \mathbf{x}_i\}.$$

En conséquence, la fonction de synergie est tronquée de manière à ce que

$$S'(i, j) = \begin{cases} S(i, j), & \text{si } j \in \text{NN}_k(i) \text{ ou } i \in \text{NN}_k(j), \\ 0, & \text{sinon,} \end{cases}$$

ce qui signifie que seules les entités jugées suffisamment proches contribuent à la mise à jour des **pondérations** $\omega_{i,j}$.

L'**avantage** de cette approche réside dans la réduction de la complexité, passant d'un nombre de comparaisons de l'ordre de $O(n^2)$ à $O(n k)$, tout en **limitant** l'impact des **outliers** qui, étant isolés, ne figurent pas dans le voisinage local.

En outre, cette stratégie contribue à renforcer la **robustesse** du DSL en ne construisant des liaisons significatives qu'entre des entités réellement compatibles dans l'espace \mathbb{R}^d .

B. Clipping des Vecteurs et des Scores de Similarité

En complément du filtrage local, le **clipping** constitue une méthode efficace pour restreindre l'influence d'**valeurs extrêmes**. Cette technique peut s'appliquer à différents niveaux du processus d'inférence. Lorsqu'il s'agit des **embeddings** eux-mêmes, il est souvent pertinent de contraindre chaque coordonnée du vecteur \mathbf{x}_i afin d'éviter que des perturbations locales ne provoquent une domination de certaines dimensions. Pour ce faire, on impose un seuil $T > 0$ et on définit le vecteur « clippé » par

$$x_{i,k}^{\text{clip}} = \min\{\max\{x_{i,k}, -T\}, T\}.$$

Cette opération garantit que les valeurs de chaque dimension restent contenues dans l'intervalle $[-T, T]$, ce qui permet de préserver une **stabilité** géométrique dans \mathbb{R}^d . Par ailleurs, il est également possible d'appliquer le clipping directement au **score de synergie**. Si, par exemple, la fonction $S(i, j)$ calcule une similarité qui pourrait parfois dépasser une borne raisonnable à cause d'un outlier, on peut définir une version clipée de cette synergie par

$$S'(i, j) = \min\{S(i, j), S_{\max}\},$$

où S_{\max} est une borne supérieure prédéfinie. Cette approche empêche que la **synergie** n'atteigne des valeurs trop élevées qui pourraient induire un renforcement disproportionné des pondérations $\omega_{i,j}$, préservant ainsi l'équilibre global du SCN.

C. Combinaison des Approches pour une Robustesse Accrue

Dans la pratique, l'application conjointe du **filtrage local** par k-NN et du **clipping** constitue une stratégie robuste pour limiter la propagation d'erreurs. En effet, le filtrage local restreint d'abord le calcul de la synergie aux voisins les plus proches, réduisant la densité du graphe et la probabilité d'étendre l'influence d'entités aberrantes.

Par la suite, le clipping, appliqué sur les vecteurs ou directement sur les scores de similarité, permet de modérer l'impact de toute valeur extrême qui pourrait encore se glisser dans le processus. Le SCN met ainsi à jour les pondérations selon la formule

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S'(i, j) - \tau \omega_{i,j}(t)],$$

où $S'(i, j)$ représente la synergie obtenue après application des mécanismes de filtrage et de clipping. Cette approche intégrée permet de réduire la complexité computationnelle, en limitant le nombre de comparaisons nécessaires, tout en préservant la stabilité de l'auto-organisation en atténuant l'impact des perturbations locales.

3.3.4.3. Exemples de Scénarios Audio-Visuels

Dans le cadre du **Deep Synergy Learning (DSL)**, les **applications audio-visuelles** offrent un terrain d'expérimentation particulièrement riche pour illustrer la capacité du système à **auto-organiser** des entités issues de modalités diverses telles que l'image, le son et le texte. En effet, le **Synergistic Connection Network (SCN)** se doit de prendre en compte des **embeddings** extraits de flux multimodaux qui, par leur nature, peuvent être affectés par le **bruit**, des changements de **contexte** ou des ajustements successifs induits par un fine-tuning continu.

Dans cette section, nous exposons en détail deux scénarios qui démontrent comment le DSL exploite des mécanismes de limitation d'erreurs, tels que le filtrage local et le clipping, ainsi que des stratégies d'adaptation dynamique, afin de préserver une organisation robuste et cohérente dans un environnement audio-visuel.

A. Détection d'Événements Audio-Visuels

Dans un premier exemple, nous considérons la détection d'un événement, tel qu'un applaudissement ou un klaxon, qui se manifeste simultanément dans la composante visuelle et

acoustique d'un flux multimodal. Pour ce faire, le flux audio-visuel est segmenté en une série de **frames vidéo** $\{\mathcal{E}_i^{(\text{vid})}\}$ et en une série de **segments audio** $\{\mathcal{E}_j^{(\text{aud})}\}$.

Chaque frame est alors transformé en un **embedding** $\mathbf{x}_i^{(\text{vid})} \in \mathbb{R}^{d_{\text{vid}}}$ par un modèle de type CNN (tel que ResNet) ou par un Vision Transformer, tandis que chaque segment audio se voit attribuer un vecteur $\mathbf{x}_j^{(\text{aud})} \in \mathbb{R}^{d_{\text{aud}}}$ issu d'un modèle d'autoencodeur appliqué sur un spectrogramme ou d'un transformeur audio tel que wav2vec.

La **synergie** entre une frame vidéo et un segment audio peut être quantifiée à l'aide d'un noyau gaussien, défini par

$$S(i, j) = \exp(-\gamma \|\mathbf{x}_i^{(\text{vid})} - \mathbf{x}_j^{(\text{aud})}\|^2),$$

où $\gamma > 0$ ajuste la sensibilité de la mesure. Toutefois, en raison des perturbations inhérentes à ce type de flux – telles que la saturation audio ou les variations lumineuses dans les images –, des valeurs aberrantes peuvent fausser le calcul de la synergie.

Pour limiter ces effets, le DSL intègre des mécanismes de **filtrage local** en restreignant le calcul de la similarité aux paires d'entités se chevauchant temporellement, ainsi que des techniques de **clipping** qui borner les valeurs extrêmes du score $S(i, j)$. Ce filtrage permet de constituer des **clusters** robustes reliant les frames et segments audio fortement synchronisés, ce qui facilite la détection d'événements multimodaux avec une grande précision.

B. Sous-Titrage Automatique à Partir d'un Flux Audio-Visuel

Dans un second scénario, le DSL est appliqué à la tâche de sous-titrage automatique, qui implique la fusion d'informations issues du flux vidéo et du flux textuel dérivé d'une transcription audio.

Dans ce contexte, chaque frame vidéo $\mathcal{E}_i^{(\text{vid})}$ se voit attribuer un embedding $\mathbf{x}_i^{(\text{vid})}$ généré par un Vision Transformer ou un CNN, tandis que la transcription issue d'un module speech-to-text est convertie en un **embedding textuel** $\mathbf{x}_j^{(\text{text})} \in \mathbb{R}^{d_{\text{text}}}$ via un modèle de type BERT ou GPT. La synergie entre la représentation visuelle et celle textuelle peut être évaluée à l'aide de la **similarité cosinus**, exprimée par

$$S_{\cos}(i, j) = \frac{\mathbf{x}_i^{(\text{vid})} \cdot \mathbf{x}_j^{(\text{text})}}{\|\mathbf{x}_i^{(\text{vid})}\| \|\mathbf{x}_j^{(\text{text})}\|}.$$

Cette mesure permet de quantifier la proximité sémantique entre le contenu visuel et le texte associé. Cependant, des erreurs peuvent survenir dans la transcription, notamment en cas de bruit, d'accents ou de débit de parole rapide, tandis que la qualité des embeddings vidéo peut être affectée par des mouvements brusques ou des changements de scène.

Afin de pallier ces imprécisions, le DSL met en place des mécanismes de **limitation temporelle** (en ne comparant que les frames proches du segment textuel) ainsi que des procédures de **clipping** pour éviter que des valeurs extrêmes ne conduisent à des scores de similarité erronés. De cette manière, le système parvient à associer de manière cohérente les segments vidéo et les extraits textuels, facilitant ainsi la génération de sous-titres ou de résumés contextuels de haute qualité.

C. Ajustement Dynamique et Fine-Tuning

Dans les deux scénarios précédents, le DSL peut être soumis à des processus de **fine-tuning** ou à des mises à jour continues, de sorte que les modèles générateurs d'embeddings (qu'il s'agisse d'un CNN, d'un Vision Transformer ou d'un modèle speech-to-text) évoluent en fonction des nouvelles données. Ainsi, les représentations \mathbf{x}_i et \mathbf{x}_j peuvent être réévaluées périodiquement, modifiant la synergie $S(i, j)$ calculée et, par conséquent, les pondérations $\omega_{i,j}$ dans le SCN.

Afin d'éviter que ces changements brusques ne perturbent la dynamique globale du réseau, des techniques de **lissage** des transitions et de **verrouillage** partiel des modèles sont mises en œuvre. Ces stratégies visent à adapter les embeddings de manière progressive, garantissant ainsi une réorganisation continue mais stable des clusters d'entités multimodales.

3.4. Représentations Symboliques

Alors que les **représentations sub-symboliques** comme les vecteurs et embeddings dominent de nombreuses applications d'apprentissage (sections 3.2 et 3.3), il existe un autre univers de modélisation où les **représentations symboliques** occupent une place centrale. Dans ces approches, chaque entité \mathcal{E}_i est décrite non plus par un vecteur, mais par un **ensemble de règles**, de **concepts** ou d'**axiomes** pouvant relever de la **logique** (règles “if... then...”) ou de la **sémantique ontologique** (graphe RDF, OWL, etc.). Cette manière de représenter l'information favorise l'**interprétabilité** et le **raisonnement formel**, deux propriétés parfois difficiles à obtenir avec des embeddings “boîte noire”. Toutefois, elle pose également des **défis** où il est nécessaire de maintenir la cohérence logique, de gérer difficilement le bruit et de faire face à une certaine rigidité face aux nouveaux attributs.

Dans cette section (3.4), nous étudierons d'abord (3.4.1) les **logiques, règles et ontologies** (une palette symbolique de base), puis (3.4.2) nous verrons **comment** calculer la synergie symbolique via des mesures de **compatibilité** ou de **cohérence**, et enfin (3.4.3) nous aborderons l'**évolution** d'un ensemble symbolique et la gestion des contradictions dans le temps.

3.4.1. Logiques, Règles et Ontologies

Les **représentations symboliques** couvrent un vaste champ où elles vont de la simple règle conditionnelle (“si A alors B”) jusqu’aux **grandes ontologies** hiérarchisées comme OWL ou RDF, qui décrivent un univers de concepts et de relations. L’objectif est de permettre à la machine de **raisonner** ou de **valider** certaines assertions de manière déclarative.

3.4.1.1. Approche Symbolique : Ensembles de Règles (if A then B), Graphes Sémantiques, Ontologies (OWL, RDF)

Dans le cadre de l’analyse des systèmes d'**auto-organisation** et d'**apprentissage** tels que le **Deep Synergy Learning (DSL)**, il est essentiel de distinguer les approches **symboliques** des méthodes **sub-symboliques**. Alors que ces dernières représentent les entités \mathcal{E}_i par des **embeddings** (obtenus par des réseaux convolutionnels, transformers, autoencodeurs, etc.) – c'est-à-dire par des vecteurs dans un espace continu \mathbb{R}^d –, l’approche symbolique se fonde sur des structures explicites et logiques. Dans ce contexte, les entités sont décrites à l'aide d'**ensembles de règles** du type « *if A then B* », de **graphes sémantiques** structurés selon des triplets, ou encore d'**ontologies** formelles telles que celles définies par OWL et RDF. Ces différentes modalités de représentation offrent chacune des avantages spécifiques en termes d'**interprétabilité**, de **raisonnement formel** et de **traçabilité** de l'information.

A. Ensembles de Règles “If A then B”

L’approche la plus intuitive dans le domaine symbolique consiste à modéliser les connaissances sous forme d’implications logiques, telles que

$$\text{if } A \text{ then } B,$$

où A représente une hypothèse ou une condition initiale et B une conclusion ou action associée. Mathématiquement, une telle règle peut être exprimée par la proposition

$$A(\mathcal{E}_i) \Rightarrow B(\mathcal{E}_i),$$

indiquant que si l'entité \mathcal{E}_i satisfait l'ensemble de conditions A , alors elle doit logiquement satisfaire la conclusion B . Par exemple, dans un contexte médical, une règle du type

if fièvre et toux alors suspect infection respiratoire

permet d'associer un ensemble d'attributs booléens à une conclusion diagnostic. L'**avantage** de cette représentation est sa grande **interprétabilité** où l'humain peut aisément comprendre la justification d'une déduction, ce qui est primordial dans des domaines nécessitant une forte **traçabilité** comme la médecine ou le droit. Cependant, l'**inconvénient** majeur réside dans la **rigidité** des règles. Une légère variation dans les conditions A peut rendre la règle inapplicable, et, à grande échelle, la gestion d'un vaste ensemble de règles peut conduire à des conflits et à une explosion combinatoire difficilement maîtrisable.

B. Graphes Sémantiques

Les **graphes sémantiques** offrent une représentation plus structurée des connaissances en modélisant chaque entité \mathcal{E}_i par un sous-graphe composé de nœuds (représentant des concepts ou attributs) et d'arcs (indiquant les relations entre ces concepts). Dans le standard **RDF** (Resource Description Framework), toute information est codée sous la forme de triplets (s, p, o) où s est le sujet, p le prédicat, et o l'objet. Par exemple, les triplets

(Brutus, type, Chien), (Chien, subclass-of, Mammifère), (Mammifère, subclass-of, Animal),

permettent d'établir une hiérarchie de classes. L'**avantage** des graphes sémantiques réside dans leur capacité à représenter explicitement des **hiérarchies** et à permettre des **requêtes complexes** via des langages tels que **SPARQL**. Ils facilitent ainsi la navigation et l'inférence sur les relations entre entités. Toutefois, ils présentent également des **inconvénients** où la difficulté à maintenir la **consistance** de grandes bases de connaissances et leur sensibilité aux erreurs rendent la gestion complexe, un triplet erroné pouvant perturber l'ensemble de la structure.

C. Ontologies (OWL, RDFSchema)

Les **ontologies** représentent une extension des graphes sémantiques en imposant un formalisme rigoureux pour décrire des classes, des propriétés, et des relations hiérarchiques ou restrictions (telles que des cardinalités ou des disjonctions). Par exemple, dans une ontologie OWL, on pourra formuler des axiomes tels que

Chien \sqsubseteq Mammifère et hasOwner:Chien \rightarrow Humain.

Dans ce cadre, un individu (par exemple, un chien nommé Brutus) sera déclaré comme une instance de la classe **Chien**, et par transitivité, comme instance de **Mammifère** et d'**Animal**. L'**avantage** majeur de cette approche est la **précision** sémantique qu'elle apporte et la capacité d'effectuer des **raisonnements automatiques** grâce à des moteurs d'inférence tels que **HermiT** ou **Fact++**. Cependant, l'**inconvénient** réside dans la complexité de la maintenance d'une ontologie de grande

échelle, qui peut devenir difficile à gérer dans des environnements où les connaissances évoluent rapidement ou sont sujettes à des **bruits** et des imprécisions.

3.4.1.2. Avantages : Interprétabilité, Raisonnement Formel

Dans l'analyse des systèmes d'**auto-organisation** et d'**apprentissage** tels que le **Deep Synergy Learning (DSL)**, les approches symboliques – fondées sur des **ensembles de règles**, des **graphes sémantiques** et des **ontologies** – se distinguent nettement des méthodes sub-symboliques qui s'appuient sur des **embeddings** et des **vecteurs** numériques. Ces approches symboliques apportent une **interprétabilité** et une **capacité de raisonnement formel** qui facilitent l'explication, la vérification et l'audit des décisions prises par le système, qualités indispensables dans des domaines où la **traçabilité** et la **cohérence** des conclusions sont primordiales, tels que la médecine, le droit ou l'ingénierie.

A. Interprétabilité Explicite

Dans une représentation symbolique, chaque entité \mathcal{E}_i est décrite par un ensemble explicite de **règles** logiques, d'**axiomes** ou d'**assertions** qui prennent la forme d'implications telles que

if A then B ,

où A représente l'hypothèse et B la conclusion. Par exemple, une règle de diagnostic médical peut être formulée comme suit :

if Toux and Fièvre then InfectionRespiratoire.

Ce type de formulation permet à un expert – qu'il soit médecin ou ingénieur – de **lire** facilement la logique sous-jacente et de comprendre de manière transparente **pourquoi** un ensemble de conditions mène à une conclusion particulière.

Contrairement aux **embeddings** sub-symboliques, où une entité est représentée par un vecteur $\mathbf{x}_i \in \mathbb{R}^d$ dont les composantes sont souvent difficiles à interpréter, les règles symboliques exposent directement les **relations** et les **attributs** qui justifient la déduction.

Dans le contexte d'un DSL, cette **lisibilité** se traduit par une **traçabilité** accrue des liens établis entre les entités. Ainsi, il est possible de lister l'ensemble des règles $\{R_i\}$ associées à chaque entité \mathcal{E}_i et d'observer, de manière explicite, comment ces règles favorisent la formation ou la dissolution d'un **cluster**.

Ce mécanisme offre une **confiance** renforcée dans les décisions du système, particulièrement dans des environnements critiques où il est essentiel de pouvoir expliquer les raisons d'un diagnostic ou d'une classification.

B. Raisonnement Formel et Inférences

Un autre avantage majeur des approches symboliques réside dans leur capacité à faciliter le **raisonnement formel** à l'aide de moteurs logiques tels que Prolog ou des reasoners OWL. Ces outils permettent d'effectuer des **inférences** automatiques et de vérifier la **consistance** d'un ensemble de règles. Par exemple, si l'on sait qu'« tout Mammifère a du sang chaud » et que l'on

déclare que « Chien est Mammifère », il est possible de déduire automatiquement que « Chien a du sang chaud ». Dans un DSL, deux entités \mathcal{E}_i et \mathcal{E}_j , dotées respectivement de blocs de règles $\{R_i\}$ et $\{R_j\}$, peuvent ainsi afficher une **synergie** élevée, notée

$$\mathbf{S}_{\text{sym}}(i, j) = f(R_i, R_j),$$

si leurs ensembles de règles sont logiquement compatibles ou se complètent. La fonction f peut mesurer, par exemple, la proportion des règles communes, la non-contradiction entre les axiomes, ou encore la conséquence logique partagée par les deux entités. De ce fait, le DSL n'est pas uniquement un système de **clusterisation** basé sur des distances numériques, mais également un outil de **raisonnement**, capable de renforcer les liens entre des entités en fonction de leur **cohérence logique**.

D. Audit et Justification des Décisions

L'un des défis majeurs dans les systèmes d'auto-organisation basés sur des embeddings est le manque de transparence quant aux raisons sous-jacentes à la formation d'un cluster. Dans un DSL symbolique, il est possible d'**auditer** les décisions prises par le système en listant explicitement les règles ou axiomes partagés entre deux entités \mathcal{E}_i et \mathcal{E}_j .

Par exemple, pour justifier pourquoi ces deux entités ont été regroupées, on peut présenter l'ensemble des attributs et des conditions qui se recoupent, telles que « elles appartiennent toutes deux à la classe Mammifère » ou « elles possèdent la même propriété hasSymptom('Toux') ». Ce mécanisme de **justification** renforce la crédibilité du système et facilite la prise de décision dans des contextes où il est impératif de pouvoir expliquer les processus de classification ou de diagnostic.

E. Réutilisation et Interopérabilité des Bases de Connaissances

Enfin, les approches symboliques bénéficient d'un important **avantage** en termes de **réutilisation** des ressources existantes. De nombreux domaines disposent déjà d'**ontologies** ou de **répertoires de règles** validés par des experts – tels que **Snomed-CT** et **UMLS** en médecine, **FIBO** dans la finance, ou **ISO 15926** en ingénierie.

Dans un DSL symbolique, chaque entité \mathcal{E}_i peut s'appuyer sur ces bases de connaissances pour hériter de classes et de propriétés prédéfinies, ce qui facilite l'intégration et l'interopérabilité avec des systèmes existants. La synergie entre les entités, notée $\mathbf{S}(i, j)$, peut ainsi bénéficier d'une sémantique partagée et validée, renforçant la cohérence globale du réseau et accélérant l'émergence de clusters pertinents.

3.4.1.3. Limites : Rigidité, Besoin de Maintenir la Cohérence Logique

Dans le cadre de la représentation symbolique au sein d'un **Deep Synergy Learning (DSL)**, l'utilisation de règles logiques, de graphes sémantiques et d'ontologies offre une grande **interprétabilité** et permet un raisonnement formel détaillé (voir section **3.4.1.2**). Cependant, ces avantages s'accompagnent de limites intrinsèques, notamment la **rigidité** inhérente aux systèmes symboliques et le besoin impératif de maintenir une **cohérence logique** à l'échelle du système.

Dans cette section, nous examinons en détail ces contraintes, en explicitant leurs implications tant du point de vue théorique que pratique.

D'une part, la **rigidité** des représentations symboliques découle de leur caractère souvent **binaire** et catégorique. En effet, dans un système purement symbolique, une règle ou un axiome est formulé sous la forme d'une implication logique, par exemple :

$$\text{if } A \text{ then } B,$$

où A et B représentent respectivement des conditions et des conclusions exprimées à l'aide de prédicats ou de variables booléennes. Cette formulation impose une discontinuité où une petite variation dans la valeur de A , par exemple une température mesurée à 37,9 °C au lieu de 38 °C, peut conduire à l'échec complet de la condition, ce qui invalide la déduction de B . Si l'on définit une fonction caractéristique $\chi_A(x)$ associée à la condition A de telle sorte que

$$\chi_A(x) = \begin{cases} 1, & \text{si } x \in A, \\ 0, & \text{sinon,} \end{cases}$$

alors une légère déviation qui ferait basculer $\chi_A(x)$ de 1 à 0 engendre une discontinuité de la fonction de déduction. Cette rigidité ne permet pas de modéliser des phénomènes progressifs ou graduels, comme le ferait une mesure sub-symbolique où les distances ou similarités se modélisent par des fonctions continues telles que la distance euclidienne ou le cosinus d'angle. Par conséquent, dans des environnements où les données sont bruitées ou incertaines, une approche symbolique stricte peut conduire à des ruptures soudaines de synergie, rendant la dynamique du DSL moins robuste.

D'autre part, la **nécessité de maintenir la cohérence logique** impose une contrainte supplémentaire dans l'architecture d'un DSL symbolique. Lorsqu'un ensemble d'entités $\{\mathcal{E}_i\}$ est caractérisé par des blocs de règles ou des ontologies, toute modification apportée à l'un de ces blocs doit être vérifiée globalement afin de préserver la consistance du système. Par exemple, considérons les axiomes suivants :

- (i) $\forall x, \text{if } x \in \text{Mammifère} \text{ then } x \text{ a du sang chaud,}$
- (ii) \mathcal{E}_i est un Mammifère,
- (iii) \mathcal{E}_i a du sang froid.

L'introduction simultanée des axiomes (i) et (ii) conduit inévitablement à la conclusion x a du sang chaud pour \mathcal{E}_i , ce qui entre en contradiction avec (iii). Dès lors, la fonction de **synergie** entre les entités, que l'on peut formuler symboliquement par une fonction f telle que

$$\mathbf{S}_{\text{sym}}(i, j) = f(R_i, R_j),$$

doit être continuellement recalculée et validée afin de s'assurer que les modifications locales (par exemple, l'ajout, la suppression ou la modification d'un axiome dans R_i) ne conduisent pas à des contradictions globales. Ce processus de vérification peut être formulé à l'aide de systèmes de contraintes ou d'algorithmes d'inférence logiques, mais il ajoute une complexité algorithmique non négligeable, en particulier lorsque le nombre d'entités et de règles est très élevé. Du point de vue de la **théorie de la complexité**, la vérification de la cohérence d'une ontologie expressive relève

souvent d'un problème NP-complet, voire de classes plus difficiles comme PSPACE ou EXPTIME dans certains cas (cf. OWL DL). Cela limite la **scalabilité** de l'approche symbolique dans un environnement dynamique et massivement distribué.

En outre, la mise à jour incrémentale d'un grand nombre de règles dans un DSL exige une **maintenance** continue de la base de connaissances. Chaque modification locale, par exemple l'ajout d'un nouvel axiome dans le bloc de règles R_i , peut avoir un effet domino sur l'ensemble du système.

Une telle modification nécessite non seulement de recalculer la synergie $\mathbf{S}_{\text{sym}}(i, j)$ entre \mathcal{E}_i et toutes les autres entités \mathcal{E}_j , mais aussi de vérifier que cette modification ne viole pas la cohérence globale du système.

En d'autres termes, la dynamique du DSL symbolique peut être ralentie par le temps nécessaire aux opérations de validation, ce qui représente un inconvénient majeur lorsqu'on cherche à opérer des mises à jour en temps réel ou en continu.

Enfin, il faut noter que les **systèmes symboliques** purs, en raison de leur caractère binaire, ne gèrent pas naturellement un **degré de confiance** ou un **niveau de flou** dans la représentation des faits.

Pour pallier ce manque, il est parfois nécessaire d'introduire des logiques floues ou probabilistes, telles que les réseaux bayésiens ou les Markov Logic Networks, ce qui complexifie encore la modélisation et l'implémentation.

La **conversion** d'un problème symbolique strict en un problème flou nécessite souvent une transformation non triviale de la fonction de synergie, et par conséquent une réévaluation de toute la dynamique d'auto-organisation.

3.4.2. Calcul de la Synergie Symbolique

Lorsque deux entités \mathcal{E}_i et \mathcal{E}_j sont décrites par des **représentations symboliques** (ensembles de règles, ontologies, graphes sémantiques), la notion de synergie $S(i, j)$ ne repose plus sur une simple distance ou similarité vectorielle. Il s'agit plutôt d'évaluer la **compatibilité**, la **non-contradiction** ou la **cohérence** entre leurs blocs de connaissances. Dans ce paragraphe, nous évoquerons successivement :

- (3.4.2.1) la **compatibilité de règles**, le **degré de contradiction** ou la **co-occurrence** des axiomes,
- (3.4.2.2) la **similarité** entre ontologies (ou sous-ontologies),
- (3.4.2.3) des approches de **distance sémantique** plus générales, voire d'**information mutuelle symbolique**.

3.4.2.1. Compatibilité de Règles, Degré de Contradiction ou Co-Occurrence

Dans le cadre du **Deep Synergy Learning (DSL)**, lorsque les entités \mathcal{E}_i sont décrites de manière symbolique – par exemple, à l'aide d'ensembles de règles, d'axiomes ou de sous-graphes ontologiques – la **synergie** $S(i,j)$ entre deux entités \mathcal{E}_i et \mathcal{E}_j dépend essentiellement de la **compatibilité** de leurs ensembles de règles, notés respectivement R_i et R_j . La compatibilité peut être vue comme le degré de recouvrement entre les deux ensembles, mais aussi comme l'absence de contradictions logiques entre eux. Nous exposerons ci-après les différents aspects de cette démarche, en intégrant les formules mathématiques nécessaires pour formaliser ces notions.

A. Définition de la Compatibilité ou Co-Occurrence

Une première approche pour quantifier la proximité symbolique entre R_i et R_j consiste à considérer leurs ensembles comme des collections d'éléments discrets et à mesurer leur **intersection** par rapport à leur **union**. On définit ainsi la mesure de recouvrement ou de **co-occurrence** par la formule :

$$S_{\text{overlap}}(i,j) = \frac{|R_i \cap R_j|}{|R_i \cup R_j|}.$$

Cette expression renvoie une valeur comprise entre 0 et 1, où un score proche de 1 indique que les deux ensembles partagent une grande partie de leurs règles (ce qui traduit une forte **similarité conceptuelle**), tandis qu'un score proche de 0 indique un faible recouvrement. Dans le contexte d'un **Synergistic Connection Network (SCN)**, il est alors possible de relier la pondération $\omega_{i,j}$ entre deux entités à cette mesure de recouvrement. Autrement dit, lorsque R_i et R_j partagent un nombre important de règles, le système renforcera la connexion entre \mathcal{E}_i et \mathcal{E}_j .

Cas pratique :

Prenons l'exemple de deux entités médicales. Supposons que l'entité \mathcal{E}_i possède l'ensemble de règles R_i incluant des assertions telles que « if Toux and Fièvre then InfectionRespiratoire », et que \mathcal{E}_j comporte un ensemble R_j similaire, par exemple incluant aussi « if Toux and Fièvre then InfectionRespiratoire » et d'autres règles liées au diagnostic d'infections respiratoires. Si $|R_i \cap R_j|$ est élevé, cela indique que les deux entités appartiennent au même domaine diagnostique, et le score $S_{\text{overlap}}(i,j)$ sera alors élevé, conduisant à un renforcement de la pondération $\omega_{i,j}$.

B. Degré de Contradiction

Outre le recouvrement, il est crucial de mesurer si les ensembles R_i et R_j présentent des **contradictions** logiques. En logique formelle, on dit qu'il y a contradiction lorsque la conjonction de certaines formules $a \in R_i$ et $b \in R_j$ conduit à une fausseté, symbolisée par \perp . Pour formaliser ce concept, on définit la fonction de contradiction de manière binaire par :

$$\text{Contradiction}(R_i, R_j) = \begin{cases} 1, & \text{si } \exists (a \in R_i, b \in R_j) \text{ tels que } a \wedge b \models \perp, \\ 0, & \text{sinon.} \end{cases}$$

Cette définition peut être affinée en comptant le nombre de conflits entre les règles, ce qui se traduit par :

$$\text{ContradictionCount}(R_i, R_j) = \sum_{(a,b) \in R_i \times R_j} \mathbf{1}[a \wedge b \models \perp],$$

où $\mathbf{1}[\cdot]$ est la fonction indicatrice qui vaut 1 si la condition est satisfaite et 0 sinon. Un nombre élevé de contradictions, c'est-à-dire une grande valeur de $\text{ContradictionCount}(R_i, R_j)$, devrait entraîner une diminution de la synergie $S(i, j)$ entre les deux entités, car il n'est pas souhaitable de regrouper des entités dont les règles sont logiquement incompatibles.

C. Combinaison de Co-Occurrence et de Contradiction : Synergie Symbolique

Pour tenir compte simultanément du recouvrement et des contradictions, il convient de définir une fonction de **synergie symbolique** qui intègre à la fois un terme positif relatif à l'intersection des ensembles et un terme négatif qui pénalise les contradictions. On peut définir cette fonction par :

$$S_{\text{sym}}(i, j) = \alpha \frac{|R_i \cap R_j|}{|R_i \cup R_j|} - \beta \text{ContradictionCount}(R_i, R_j),$$

où α et β sont des constantes strictement positives qui pondèrent respectivement l'importance du recouvrement et celle du degré de contradiction. Ce score, que l'on peut ensuite contraindre à être non négatif (par exemple, en le plafonnant à zéro si nécessaire), guide la dynamique du DSL en influençant la mise à jour des pondérations selon la règle :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{sym}}(i, j) - \tau \omega_{i,j}(t)],$$

où η est le taux d'apprentissage et τ le coefficient de décroissance. Dans cette dynamique, si $\text{ContradictionCount}(R_i, R_j)$ est élevé, le terme $S_{\text{sym}}(i, j)$ diminue, conduisant à un affaiblissement de $\omega_{i,j}$.

D. Complexité d'Inférence Logique

La mise en œuvre pratique de ces concepts pose toutefois des défis liés à la **complexité** du raisonnement logique. La vérification de la contradiction, c'est-à-dire déterminer si une conjonction $a \wedge b$ déduit \perp , nécessite l'emploi d'un **raisonneur** ou d'un module de déduction automatique. Dans des ontologies ou des systèmes logiques expressifs (par exemple, OWL avec une expressivité complète), la vérification de la cohérence d'un ensemble de règles peut être un problème de complexité exponentielle. Pour atténuer ce problème, il est souvent nécessaire de :

- Limiter l'expressivité du système logique en passant à des versions simplifiées telles que RDF basique ou OWL Lite.
- Utiliser des structures d'indexation ou des techniques de hashage sémantique pour repérer rapidement les conflits potentiels.
- Se concentrer sur des sous-ensembles critiques des ensembles R_i et R_j lorsqu'un recouvrement minimal est déjà détecté, afin de réduire le nombre de vérifications nécessaires.

Ces heuristiques permettent d'obtenir une approximation de la synergie symbolique qui reste **scalable** pour de grandes bases de connaissances.

E. Application dans la Mise à Jour $\omega_{i,j}$

L'ensemble des considérations ci-dessus se traduit directement dans la règle d'auto-organisation du **SCN**. En effet, la synergie symbolique $S_{\text{sym}}(i, j)$ influence la dynamique des pondérations $\omega_{i,j}$ par la mise à jour suivante :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{sym}}(i, j) - \tau \omega_{i,j}(t)].$$

Ce schéma permet d'ajuster les liens entre entités en fonction de la compatibilité de leurs ensembles de règles. Ainsi, si deux entités partagent un grand nombre d'axiomes sans contradictions (c'est-à-dire si $S_{\text{overlap}}(i, j)$ est élevé et $\text{ContradictionCount}(R_i, R_j)$ est faible), la pondération $\omega_{i,j}$ sera renforcée. À l'inverse, la présence de contradictions entraînera une réduction de $\omega_{i,j}$, contribuant à la formation de clusters de règles cohérentes.

3.4.2.2. Similarité d'Ontologies, Mesure d'Overlap Conceptuel

Dans le cadre du **Deep Synergy Learning (DSL)**, lorsque les entités \mathcal{E}_i sont décrites par des représentations symboliques, leur caractérisation repose souvent sur des ontologies formalisées selon des standards tels que OWL ou RDF.

Ces ontologies définissent de manière explicite des **classes**, des **sous-classes**, des **relations** et des **axiomes** qui structurent la connaissance d'un domaine. Ainsi, chaque entité \mathcal{E}_i se voit associée à un **sous-graphe ontologique** \mathcal{G}_i , lequel regroupe l'ensemble des triplets, assertions et axiomes pertinents à cette entité.

Pour évaluer la **synergie $\mathbf{S}(i, j)$** entre deux entités \mathcal{E}_i et \mathcal{E}_j , il convient de mesurer la similarité entre leurs sous-graphes respectifs, c'est-à-dire d'estimer l'**overlap conceptuel** entre \mathcal{G}_i et \mathcal{G}_j .

Cette section présente une approche formelle et détaillée de la mesure d'overlap conceptuel, en intégrant des formules mathématiques précises pour quantifier cette similarité.

A. Notion d'Ontologie et de Sous-Graphe

Une **ontologie** se définit comme un ensemble structuré de concepts, de relations et d'axiomes qui permettent de représenter la connaissance d'un domaine de manière formelle. Dans ce contexte, chaque entité \mathcal{E}_i est associée à un sous-graphe ontologique \mathcal{G}_i qui contient les triplets RDF et autres assertions telles que

$$\mathcal{E}_i \text{ rdf:type } \mathcal{C}, \quad \mathcal{C}_1 \text{ rdfs:subClassOf } \mathcal{C}_2,$$

ainsi que d'autres relations définies en OWL, comme « equivalentClass » ou « disjointWith ». Cette représentation permet d'organiser la connaissance en hiérarchies et en réseaux de relations sémantiques. Elle offre l'avantage de rendre explicite la structure de la connaissance et de faciliter le raisonnement automatique grâce à des reasoners tels que HermiT ou Fact++.

B. Mesure d'Overlap Conceptuel

1. Overlap de Triplets

La méthode la plus directe pour quantifier la similarité entre deux sous-graphes \mathcal{G}_i et \mathcal{G}_j consiste à mesurer le **recouvrement** de leurs triplets. Si l'on considère chaque triplet comme un élément discret, une mesure simple de co-occurrence peut être définie par l'opérateur suivant :

$$S_{\text{ont}}(i, j) = \frac{|\mathcal{G}_i \cap \mathcal{G}_j|}{|\mathcal{G}_i \cup \mathcal{G}_j|}.$$

Cette expression attribue une valeur comprise entre 0 et 1. Un score proche de 1 signifie que les deux sous-graphes partagent presque tous leurs triplets, indiquant une forte **cohérence sémantique**. À l'inverse, un score proche de 0 indique que les deux entités se définissent par des concepts et relations très différents. Ainsi, dans le cadre d'un **SCN**, la pondération $\omega_{i,j}$ peut être renforcée lorsque $S_{\text{ont}}(i, j)$ est élevé, favorisant la formation de clusters d'entités ontologiquement similaires.

2. Isomorphisme Partiel et Distance Hiérarchique

Bien que le recouvrement direct des triplets offre une première approximation de la similarité, il peut arriver que deux sous-graphes ne se recouvrent pas textuellement tout en représentant des concepts équivalents ou proches.

Par exemple, deux entités peuvent utiliser des terminologies différentes pour désigner un même concept (par ex. "Flu" et "Influenza"). Pour traiter ce cas, il est nécessaire d'introduire des mécanismes d'**isomorphisme partiel**. L'idée est d'identifier un **maximum common subgraph (MCS)** entre \mathcal{G}_i et \mathcal{G}_j qui tient compte des synonymes et des relations de hiérarchie telles que « rdfs:subClassOf » ou « owl:equivalentClass ». On peut alors définir un score de similarité par le rapport :

$$S_{\text{MCS}}(i, j) = \frac{|\text{MCS}(\mathcal{G}_i, \mathcal{G}_j)|}{\max(|\mathcal{G}_i|, |\mathcal{G}_j|)}.$$

Une autre approche consiste à mesurer la **distance hiérarchique** entre les concepts correspondants. Par exemple, si deux classes se trouvent à des niveaux différents dans la hiérarchie ontologique, une fonction de distance $d(\mathcal{C}_i, \mathcal{C}_j)$ peut être utilisée pour ajuster la similarité, de sorte qu'un décalage important dans la hiérarchie diminue la valeur du score de similarité.

3. Combinaison en une Fonction de Synergie Ontologique

Pour intégrer de manière globale ces notions, on peut définir la **synergie ontologique** $\mathbf{S}_{\text{ont}}(i, j)$ comme une combinaison d'un terme de recouvrement et d'un terme ajusté par la distance ou par une évaluation plus fine de l'isomorphisme partiel. Par exemple, une formule possible est :

$$\mathbf{S}_{\text{ont}}(i, j) = \alpha S_{\text{ont}}(i, j) + \beta S_{\text{MCS}}(i, j),$$

où α et β sont des coefficients de pondération qui reflètent l'importance relative du recouvrement direct et de la similarité structurelle. Dans le cas où la distance hiérarchique est également prise en

compte, on peut ajuster le score par une fonction décroissante de cette distance, $g(d)$, de sorte que :

$$\mathbf{S}_{\text{ont}}(i, j) = \alpha S_{\text{ont}}(i, j) + \beta S_{\text{MCS}}(i, j) \cdot g(d(\mathcal{G}_i, \mathcal{G}_j)).$$

Cette approche permet d'obtenir une mesure de synergie qui valorise à la fois la **co-occurrence** des triplets et la similarité structurelle au sein des sous-graphes ontologiques.

C. Application dans la Dynamique du DSL

Dans un **Synergistic Connection Network**, la synergie ontologique $\mathbf{S}_{\text{ont}}(i, j)$ ainsi définie peut être intégrée directement dans la règle de mise à jour des pondérations. La mise à jour s'exprime alors par :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [\mathbf{S}_{\text{ont}}(i, j) - \tau \omega_{i,j}(t)],$$

où η représente le taux d'apprentissage et τ le coefficient de décroissance. Cette dynamique permet d'ajuster les liaisons entre entités en fonction de leur **similarité conceptuelle**.

Lorsque deux entités partagent un grand nombre de triplets ou présentent une forte similitude structurelle dans leurs sous-graphes ontologiques, la pondération $\omega_{i,j}$ est renforcée, favorisant ainsi l'émergence de clusters d'entités ayant des liens sémantiques forts. À l'inverse, un faible recouvrement ou une grande distance hiérarchique conduira à une diminution de $\omega_{i,j}$.

D. Problèmes de Complexité et Approximations

Il est important de noter que le calcul précis de la similarité ontologique peut devenir très coûteux en termes de complexité algorithmique, notamment lorsqu'il s'agit de trouver le maximum common subgraph (MCS) ou d'évaluer des distances hiérarchiques dans des ontologies très expressives.

En pratique, pour des systèmes à grande échelle, il est souvent nécessaire de recourir à des heuristiques ou à des méthodes d'**approximate graph matching** afin de rendre le calcul de $\mathbf{S}_{\text{ont}}(i, j)$ faisable dans un temps raisonnable. Des techniques d'**indexation sémantique** ou des simplifications de l'ontologie (par exemple, en utilisant OWL Lite plutôt qu'OWL DL) peuvent également être employées pour réduire la charge computationnelle tout en conservant une précision acceptable.

Conclusion

La **mesure d'overlap conceptuel** entre deux sous-graphes ontologiques se fonde sur la quantification du recouvrement des triplets et la prise en compte d'une similarité structurelle, ajustée par la distance hiérarchique ou l'isomorphisme partiel. La synergie ontologique est ainsi définie par une fonction combinant ces aspects :

$$\mathbf{S}_{\text{ont}}(i, j) = \alpha \frac{|\mathcal{G}_i \cap \mathcal{G}_j|}{|\mathcal{G}_i \cup \mathcal{G}_j|} + \beta S_{\text{MCS}}(i, j) \cdot g(d(\mathcal{G}_i, \mathcal{G}_j)),$$

et intervient dans la dynamique d'auto-organisation du SCN via la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [\mathbf{S}_{\text{ont}}(i,j) - \tau \omega_{i,j}(t)].$$

Cette approche permet de renforcer les liens entre entités dont les représentations ontologiques se recouvrent fortement et de diminuer ceux présentant des divergences structurelles, aboutissant ainsi à des clusters sémantiquement cohérents. Les défis principaux résident dans la complexité des calculs d'inférence logique et dans la nécessité d'employer des méthodes approximatives pour assurer la **scalabilité** du système. Les sections **3.4.2** et **3.4.3** approfondiront ces compromis et présenteront des solutions hybrides pour intégrer ces mesures dans un cadre de Deep Synergy Learning.

3.4.2.3. Approches “Distance Sémantique” ou “Information Mutuelle Symbolique”

Dans le cadre du **Deep Synergy Learning (DSL)**, la capacité à mesurer la synergie entre deux entités symboliques repose non seulement sur la simple co-occurrence ou le recouvrement de règles, mais aussi sur des méthodes quantitatives permettant d'estimer de façon plus fine le degré de proximité conceptuelle. Deux approches complémentaires se distinguent ainsi où la **distance sémantique** et l'**information mutuelle symbolique** offrent des perspectives différentes mais complémentaires. Ces deux outils visent à quantifier la richesse informative partagée ou l'éloignement conceptuel entre les représentations symboliques des entités, et sont ensuite intégrés dans la dynamique d'auto-organisation du SCN (Synergistic Connection Network).

A. Approche par Distance Sémantique

Lorsqu'une entité \mathcal{E}_i est représentée par un sous-graphe ontologique \mathcal{G}_i , l'**ontologie** fournit une structure hiérarchique reliant des concepts par des relations telles que `rdfs:subClassOf` ou `owl:equivalentClass`. Pour évaluer la **proximité sémantique** entre deux entités \mathcal{E}_i et \mathcal{E}_j , on peut définir une fonction de **distance sémantique** $d_{\text{sem}}(\mathcal{E}_i, \mathcal{E}_j)$ qui exploite la structure hiérarchique de l'ontologie. Par exemple, en identifiant pour deux concepts C_i et C_j leur **plus bas ancêtre commun** (LCS, *lowest common subsumer*), on peut mesurer la distance en nombre d'arêtes parcourues pour atteindre ce LCS. Les travaux issus de WordNet ont introduit des mesures classiques telles que la **mesure de Resnik** et la **mesure de Lin**, définies respectivement par :

$$\begin{aligned} \text{sim}_{\text{Resnik}}(C_i, C_j) &= \text{IC}(\text{lcs}(C_i, C_j)), \\ \text{sim}_{\text{Lin}}(C_i, C_j) &= \frac{2 \text{IC}(\text{lcs}(C_i, C_j))}{\text{IC}(C_i) + \text{IC}(C_j)}, \end{aligned}$$

où $\text{IC}(C)$ représente le **contenu informationnel** du concept C , une mesure qui augmente avec la spécificité du concept. Dans le contexte du DSL, on peut associer à chaque entité \mathcal{E}_i un ou plusieurs concepts dominants extraits de son sous-graphe ontologique et définir la synergie sémantique entre \mathcal{E}_i et \mathcal{E}_j par :

$$\mathbf{S}_{\text{sem}}(i, j) = \text{sim}_{\text{Lin}}(C_i, C_j) \quad \text{ou} \quad \text{sim}_{\text{Resnik}}(C_i, C_j),$$

selon le choix de la mesure, de sorte que deux entités seront considérées comme sémantiquement proches lorsque leurs concepts associés présentent un fort degré d'information partagée. Une autre

variante consiste à appliquer une transformation exponentielle afin de convertir la distance sémantique en une similitude, par exemple :

$$\mathbf{S}_{\text{sem}}(i, j) = \exp(-\alpha d_{\text{sem}}(\mathcal{E}_i, \mathcal{E}_j)),$$

où $\alpha > 0$ est un paramètre de réglage. Ce type d'approche permet une tolérance aux variations et aux synonymes, en tirant parti de la structure hiérarchique de l'ontologie.

B. Approche par Information Mutuelle Symbolique

Une alternative quant à elle consiste à adapter le concept d'**information mutuelle (MI)**, largement utilisé en théorie de l'information pour mesurer la dépendance entre deux variables aléatoires, à un contexte symbolique. Si l'on considère qu'un ensemble de règles ou d'axiomes associé à une entité \mathcal{E}_i peut être représenté par un vecteur binaire $\mathbf{x}_i \in \{0,1\}^m$ — où chaque composante $x_i(k) = 1$ indique la présence de l'axiome r_k dans l'ensemble R_i de \mathcal{E}_i — alors on peut définir des variables aléatoires X_i et X_j pour les entités \mathcal{E}_i et \mathcal{E}_j . L'**information mutuelle** entre X_i et X_j se définit par :

$$I(X_i; X_j) = \sum_{\mathbf{x} \in \{0,1\}^m} \sum_{\mathbf{y} \in \{0,1\}^m} p(\mathbf{x}, \mathbf{y}) \log \left(\frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}) p(\mathbf{y})} \right),$$

où $p(\mathbf{x})$ et $p(\mathbf{y})$ sont les distributions marginales et $p(\mathbf{x}, \mathbf{y})$ la distribution conjointe. Une valeur élevée de $I(X_i; X_j)$ signifie qu'une connaissance du vecteur binaire associé à \mathcal{E}_i réduit significativement l'incertitude concernant celui de \mathcal{E}_j , indiquant ainsi une forte **synergie symbolique**. Afin de normaliser ce score dans l'intervalle $[0,1]$, on peut utiliser la formule :

$$\mathbf{S}_{\text{MI}}(i, j) = \frac{I(X_i; X_j)}{\max\{H(X_i), H(X_j)\}},$$

où $H(X)$ représente l'**entropie** de la variable X . Ce score permet de capturer la part d'information partagée entre deux ensembles d'axiomes, offrant une mesure plus nuancée que le simple recouvrement.

C. Intégration dans la Dynamique du DSL

Dans un **Deep Synergy Learning (DSL)**, la synergie entre deux entités symboliques est utilisée pour adapter la pondération de leur connexion. Ainsi, la mise à jour de la pondération $\omega_{i,j}$ s'effectue généralement selon la règle :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [\mathbf{S}(i, j) - \tau \omega_{i,j}(t)],$$

où $\mathbf{S}(i, j)$ peut être défini soit comme $\mathbf{S}_{\text{sem}}(i, j)$ dans le cas d'une approche par distance sémantique, soit comme $\mathbf{S}_{\text{MI}}(i, j)$ dans le cadre de l'information mutuelle symbolique, ou une combinaison des deux. Par exemple, on peut poser :

$$\mathbf{S}(i, j) = \alpha \mathbf{S}_{\text{sem}}(i, j) + \beta \mathbf{S}_{\text{MI}}(i, j),$$

où α et β sont des coefficients d'ajustement qui déterminent l'importance relative de chaque méthode dans le calcul global de la synergie. Cette formulation permet au **SCN** de renforcer les

liens entre entités dont les sous-graphes ontologiques ou les ensembles d'axiomes présentent une forte similarité sémantique et une information mutuelle élevée, tout en affaiblissant les connexions entre entités peu compatibles.

D. Problèmes de Complexité et Méthodes Approximatives

Il est essentiel de souligner que le calcul exact de la **similarité ontologique** peut s'avérer extrêmement coûteux en termes de complexité algorithmique, surtout lorsqu'il s'agit de déterminer le **maximum common subgraph (MCS)** ou d'évaluer précisément la distance hiérarchique entre concepts dans une ontologie expressive. Pour pallier ce problème, plusieurs stratégies d'approximation peuvent être mises en œuvre :

- **Simplification de l'ontologie** : en réduisant l'expressivité (par exemple en utilisant OWL Lite ou RDF basique), la complexité des inférences logiques peut être significativement diminuée.
- **Utilisation d'index sémantiques** : le recours à des structures de données de type hash ou à des index pré-calculés peut accélérer la détection des correspondances ou des conflits entre axiomes.
- **Heuristiques d'approximation** : des algorithmes d'approximate graph matching permettent d'obtenir des estimations de similarité en temps raisonnable, sans garantir l'exactitude du MCS.

Ces méthodes permettent de rendre l'intégration de la synergie ontologique dans la dynamique du DSL plus **scalable**, même si elles introduisent un compromis entre précision et efficacité.

Conclusion

La **mesure d'overlap conceptuel** entre deux sous-graphes ontologiques s'appuie sur des approches complémentaires telles que la **distance sémantique** et l'**information mutuelle symbolique**. La première exploite la structure hiérarchique des concepts pour évaluer la proximité entre deux entités, tandis que la seconde quantifie la part d'information partagée entre leurs ensembles d'axiomes. Ces deux approches, qui peuvent être combinées de manière linéaire pour former un score global

$$\mathbf{S}_{\text{ont}}(i, j) = \alpha \frac{|\mathcal{G}_i \cap \mathcal{G}_j|}{|\mathcal{G}_i \cup \mathcal{G}_j|} + \beta \mathbf{S}_{\text{MI}}(i, j),$$

s'intègrent dans la règle de mise à jour du DSL :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [\mathbf{S}_{\text{ont}}(i, j) - \tau \omega_{i,j}(t)],$$

permettant ainsi d'ajuster dynamiquement la force des connexions en fonction de la similarité sémantique. Malgré les coûts de calcul inhérents à ces méthodes d'inférence, l'usage de techniques d'approximation et d'indexation offre une voie réaliste pour appliquer ces concepts à grande échelle. Ces mesures enrichissent le **SCN** en lui conférant une dimension explicative et robuste, essentielle pour des applications où la cohérence sémantique joue un rôle crucial, notamment dans des domaines tels que la médecine, l'ingénierie ou la recherche scientifique. Les sections 3.4.2 et

3.4.3 aborderont en détail les compromis et les solutions hybrides permettant d'optimiser ces approches dans le cadre du Deep Synergy Learning.

3.4.3. Évolution Symbolique et Gestes des Contradictions

Les systèmes symboliques, bien qu'offrant une **interprétabilité** et un **raisonnement formel** (sections 3.4.1 et 3.4.2), n'échappent pas à la nécessité d'**évoluer** au fil du temps. Au même titre que les embeddings sub-symboliques (chap. 3.3.4), un bloc de connaissances symboliques doit pouvoir accueillir de **nouvelles** règles, en supprimer d'obsolètes ou de contradictoires, et gérer les conflits potentiels qui naissent de ces changements. Cette section (3.4.3) examine comment on peut **insérer** ou **supprimer** des règles (3.4.3.1), comment **détecter** et traiter les **contradictions** (3.4.3.2), et donne un **exemple** d'évolution continue d'un ensemble de postulats (3.4.3.3).

3.4.3.1. Insertion / Suppression de Règles (Chap. 9, Flux Continu)

Dans un contexte de **Deep Synergy Learning (DSL)** à composante symbolique, chaque entité \mathcal{E}_i est caractérisée par un ensemble de règles ou d'axiomes, noté R_i . Ces ensembles R_i incarnent la base de connaissances qui sous-tend la représentation symbolique d'une entité. Lorsque de nouveaux faits sont découverts ou que certaines règles deviennent obsolètes, il est alors nécessaire de modifier ces blocs R_i par l'insertion ou la suppression de postulats. Cette opération de mise à jour dynamique s'inscrit dans la **dynamique** d'un **Synergistic Connection Network (SCN)** et influence directement la **synergie symbolique** entre entités ainsi que la formation des **clusters** qui en résultent.

A. Insertion de Règles : Actualisation et Cohérence

L'insertion de nouvelles règles intervient lorsque l'on souhaite **étendre** ou **affiner** la base de connaissances associée à une entité \mathcal{E}_i . Formellement, si l'on considère qu'à l'instant t l'ensemble de règles associé à \mathcal{E}_i est $R_i(t)$, l'ajout d'un nouveau postulat p se traduit par la mise à jour :

$$R_i(t+1) = R_i(t) \cup \{p\}.$$

Par exemple, dans un contexte médical, l'introduction de la règle « if Toux and OdeurPerdue then CovidSuspect » permet d'actualiser le diagnostic en fonction des nouvelles observations. De même, en ingénierie, l'insertion d'une règle telle que « if PièceX subtend AxisY then isRotational » formalise une relation d'articulation nouvellement identifiée.

Dès l'insertion du postulat p , la **synergie** entre l'entité \mathcal{E}_i et une autre entité \mathcal{E}_j , notée $S(i,j)$, peut être affectée. Si p renforce la compatibilité entre R_i et R_j – par exemple, en augmentant le recouvrement entre les ensembles de règles – alors la pondération $\omega_{i,j}$ se voit augmenter conformément à la dynamique d'auto-organisation donnée par la formule :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)].$$

Dans ce contexte, $S(i,j)$ pourra être recalculé, par exemple en utilisant une fonction symbolique $S_{\text{sym}}(i,j)$ qui intègre le nouvel axiome. À l'inverse, si l'ajout de p engendre une contradiction avec

un axiome existant dans R_j , le score de synergie diminuera, ce qui se traduira par une baisse de $\omega_{i,j}$.

Dans un **flux continu** tel que développé au Chapitre 9, ces opérations d'insertion se produisent fréquemment et doivent être intégrées de façon à ce que la dynamique du SCN puisse s'ajuster en temps réel. Ainsi, chaque modification dans R_i entraîne un réexamen partiel – voire complet – des pondérations $\omega_{i,j}$ associées, afin de refléter l'impact des nouvelles règles sur la **compatibilité** entre entités.

B. Suppression de Règles : Réorganisation Logique

Le retrait de règles constitue l'autre facette de la mise à jour de la base de connaissances dans un DSL symbolique. Lorsque certains axiomes deviennent obsolètes ou sont invalidés par de nouvelles données, on procède à la suppression de ces éléments dans R_i . Cette opération s'exprime par :

$$R_i(t+1) = R_i(t) \setminus \{q\},$$

où q représente le postulat à retirer. Par exemple, dans le domaine médical, la règle « if Toux then Infectieux » pourra être supprimée si elle est jugée trop générale ou contredite par de nouveaux diagnostics. De même, dans un contexte industriel, l'ancienne contrainte « chargeMax = 500\,N » pourrait être écartée au profit d'une nouvelle norme.

La suppression de q affecte également la synergie $S(i,j)$ entre \mathcal{E}_i et d'autres entités \mathcal{E}_j . Si q constituait un point de recouvrement important avec R_j , sa disparition diminuera la valeur de l'intersection $R_i \cap R_j$, et par conséquent, le score de similarité. Inversement, si q était à l'origine source de contradiction avec R_j , son retrait peut améliorer la compatibilité, entraînant une augmentation de $\omega_{i,j}$.

C. Problème de l'Échelle : Révisions en $\mathcal{O}(n)$ et Gestion de la Charge

Dans un SCN comportant un grand nombre d'entités, chaque opération d'insertion ou de suppression sur R_i peut théoriquement nécessiter une réévaluation de la synergie $S(i,j)$ pour chaque entité \mathcal{E}_j du réseau, ce qui représente un coût de l'ordre de $\mathcal{O}(n)$ par modification. Lorsque de telles mises à jour se produisent simultanément ou de manière fréquente, la charge de recalcul peut devenir prohibitif. Pour atténuer ce problème, deux stratégies d'optimisation peuvent être mises en œuvre :

- **Index sémantique** : En associant à chaque règle ou concept une structure de données indiquant l'ensemble des entités qui l'utilisent, il est possible de restreindre la réévaluation aux seules entités affectées par la modification. Ainsi, l'insertion ou la suppression d'un axiome p dans R_i ne nécessite de recalculer $S(i,j)$ que pour les entités j qui partagent p .
- **Approche paresseuse (lazy evaluation)** : Au lieu de recalculer immédiatement la synergie pour toutes les paires affectées, la mise à jour est différée et réalisée uniquement lorsque la synergie est requise pour une opération spécifique, réduisant ainsi la fréquence des mises à jour globales.

D. Dynamique et Versionnage

Afin de gérer efficacement l'évolution des ensembles de règles, il est crucial de mettre en place un mécanisme de **versionnage**. Ce mécanisme permet de tracer l'historique des modifications et de gérer les mises à jour de manière structurée. On définit ainsi :

$$R_i^{(t+1)} = (R_i^{(t)} \cup R_{\text{add}}(t)) \setminus R_{\text{del}}(t),$$

où $R_{\text{add}}(t)$ et $R_{\text{del}}(t)$ représentent respectivement les ensembles de règles ajoutées et supprimées à l'itération t . Ce versionnage offre plusieurs avantages où il permet de conserver un historique des états $\{R_i^{(0)}, R_i^{(1)}, \dots\}$, de définir des points de contrôle pour revenir à une version antérieure en cas de conflit majeur et d'analyser l'évolution de la complexité des blocs R_i au fil du temps. Ainsi, le suivi des versions facilite la gestion des contradictions et offre une base pour des mécanismes de rollback si nécessaire.

E. Risques d'Oscillation et de Chaos Symbolique

Un système DSL symbolique, qui évolue en flux continu, peut être sujet à des phénomènes d'**oscillation** ou même de **chaos logique** si les mises à jour des règles se produisent de manière trop rapide ou contradictoire.

Par exemple, si une entité \mathcal{E}_i insère successivement un axiome $A \Rightarrow B$ qui contredit les règles d'une entité \mathcal{E}_j et que, peu de temps après, ce même axiome est supprimé pour rétablir la compatibilité, la pondération $\omega_{i,j}$ peut fluctuer de manière excessive. De même, un changement global simultané dans plusieurs blocs R_i peut provoquer un « big bang » dans le SCN, où un grand nombre de synergies doivent être recalculées, ce qui pourrait engendrer des cycles d'instabilité.

Pour pallier ces risques, il est souvent nécessaire de regrouper les modifications dans des phases de mise à jour (batch updates) afin de permettre à la dynamique $\omega_{i,j}$ de converger partiellement entre deux vagues de changements.

Conclusion

L'insertion et la suppression de règles dans un **DSL symbolique** constituent des opérations essentielles permettant l'actualisation et l'affinement de la base de connaissances associée à chaque entité \mathcal{E}_i . La mise à jour de R_i se traduit par l'équation

$$R_i(t+1) = (R_i(t) \cup R_{\text{add}}(t)) \setminus R_{\text{del}}(t),$$

et entraîne une réévaluation des synergies $S(i,j)$ et, par conséquent, des pondérations $\omega_{i,j}$ dans le réseau. Bien que ces opérations permettent d'adapter le SCN aux nouvelles connaissances ou à l'obsolescence de certains axiomes, elles impliquent également un coût algorithmique élevé, particulièrement à grande échelle, ainsi qu'un risque d'oscillation ou de chaos symbolique si les mises à jour ne sont pas régulées. Pour surmonter ces obstacles, il est impératif d'implémenter des stratégies d'optimisation telles que l'indexation sémantique, l'évaluation paresseuse, ainsi qu'un mécanisme de versionnage robuste. Ces stratégies permettent de limiter les révisions globales, d'assurer la cohérence logique du système, et de contrôler le risque d'instabilité lors des mises à jour en flux continu. Dans le Chapitre 9, nous approfondirons ces stratégies et heuristiques pour

gérer l'apprentissage continu et la **plasticité** des règles, en combinant la **souplesse** auto-organisée du DSL avec les contraintes strictes d'un système symbolique.

3.4.3.2. Détection de Contradiction, Feedback Top-Down pour Forcer un Réagencement (Chap. 10)

Dans un système de **Deep Synergy Learning (DSL)** à composante symbolique, chaque entité \mathcal{E}_i est décrite par un ensemble de règles ou d'axiomes R_i . La dynamique de mise à jour de ces blocs, comme présentée en section 3.4.3.1, permet d'insérer ou de supprimer des postulats afin d'actualiser la base de connaissances.

Toutefois, lorsque ces modifications conduisent à l'introduction de contradictions – c'est-à-dire, lorsque l'union $R_i \cup R_j$ de deux ensembles de règles permet de déduire simultanément un énoncé et son contraire – la **synergie symbolique** $S(i, j)$ entre les entités concernées doit être réévaluée et, souvent, fortement pénalisée.

La présente section se propose d'expliquer de manière rigoureuse comment détecter ces contradictions, de quelle manière elles impactent la dynamique du SCN, et comment un mécanisme de **feedback top-down** peut être introduit pour forcer un réagencement global et éviter des oscillations ou un chaos logique.

A. Détection de Contradiction : Fondements et Méthodes

La contradiction entre deux ensembles de règles R_i et R_j se produit lorsqu'ils, pris ensemble, engendrent une incohérence formelle, c'est-à-dire lorsque :

$$R_i \cup R_j \vdash \perp,$$

où le symbole \perp représente l'absurdité ou la fausseté. Pour quantifier cette contradiction, on définit une fonction binaire de contradiction de la manière suivante :

$$\text{Contradiction}(R_i, R_j) = \begin{cases} 1, & \text{si } \exists a \in R_i, b \in R_j \text{ tels que } a \wedge b \vdash \perp, \\ 0, & \text{sinon.} \end{cases}$$

Cette définition peut être raffinée en comptant le nombre de conflits effectifs entre les deux ensembles, en posant :

$$\text{ContradictionCount}(R_i, R_j) = \sum_{(a,b) \in R_i \times R_j} \mathbf{1}[a \wedge b \vdash \perp],$$

où $\mathbf{1}[\cdot]$ désigne la fonction indicatrice qui vaut 1 si l'argument est vrai et 0 sinon. Ce comptage permet d'obtenir une mesure plus graduelle de l'incohérence entre R_i et R_j , laquelle pourra être intégrée dans la fonction de synergie symbolique.

En effet, si l'on considère une fonction de synergie symbolique qui combine à la fois les aspects de recouvrement et de contradiction, on peut écrire :

$$S_{\text{sym}}(i, j) = \alpha \frac{|R_i \cap R_j|}{|R_i \cup R_j|} - \beta \text{ContradictionCount}(R_i, R_j),$$

où $\alpha, \beta > 0$ sont des constantes pondératrices qui ajustent l'importance relative du recouvrement par rapport aux contradictions. Ainsi, une forte contradiction (c'est-à-dire une grande valeur de $\text{ContradictionCount}(R_i, R_j)$) fera tendre $S_{\text{sym}}(i, j)$ vers zéro, ce qui, dans la dynamique du DSL, entraînera une diminution de la pondération $\omega_{i,j}$.

B. Impact sur la Dynamique du SCN et Mise à Jour des Pondérations

La règle de mise à jour du SCN, dans sa version symbolique, est donnée par :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{sym}}(i, j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ est le taux d'apprentissage et $\tau > 0$ le coefficient de décroissance. Dès qu'une contradiction est détectée, c'est-à-dire dès que $\text{Contradiction}(R_i, R_j) = 1$ ou que $\text{ContradictionCount}(R_i, R_j)$ est élevé, le score $S_{\text{sym}}(i, j)$ est pénalisé. Cela se traduit par une réduction immédiate de la pondération $\omega_{i,j}$, isolant ainsi les entités dont les bases de connaissances sont incohérentes.

Si ces contradictions apparaissent de manière isolée, la diminution locale de $\omega_{i,j}$ peut suffire à empêcher la formation de clusters incohérents. Cependant, dans un système à grande échelle, de nombreux conflits peuvent émerger simultanément, engendrant ainsi un risque de **chaos** local ou d'oscillations persistantes dans la dynamique de mise à jour.

C. Mécanisme de Feedback Top-Down pour Forcer le Réagencement

Pour pallier le risque d'instabilité engendré par un excès de contradictions, il est nécessaire d'introduire un **feedback top-down**. Ce mécanisme agit comme un **régulateur global** qui surveille la cohérence du système et intervient lorsque le nombre de contradictions dépasse un certain seuil critique.

On peut définir une métrique globale de contradiction pour l'ensemble du SCN :

$$\text{ContradictCount} = \sum_{i,j} \text{ContradictionCount}(R_i, R_j).$$

Si cette valeur excède un seuil prédéfini T , alors le mécanisme de feedback top-down se déclenche. Ce module supérieur peut alors entreprendre plusieurs actions pour restaurer l'ordre :

- **Revue et Révision des Règles** : Le système identifie les entités \mathcal{E}_i présentant le plus grand nombre de conflits (par exemple, celles pour lesquelles $\text{ContradictionCount}(R_i) = \sum_j \text{ContradictionCount}(R_i, R_j)$ est élevé) et demande une révision de leurs ensembles R_i . Cela peut impliquer la suppression ou la modification des axiomes responsables des conflits.
- **Forçage d'un Réagencement** : En imposant une baisse significative de la pondération pour les liens conflictuels, c'est-à-dire en réglant $\omega_{i,j}$ à une valeur faible ou nulle pour les paires

en contradiction, le système peut isoler les entités problématiques, forçant ainsi une réorganisation des clusters.

- **Re-Clustering Global** : Dans des cas extrêmes, le méta-module peut déclencher une phase de re-clusterisation, regroupant de manière forcée les entités compatibles et séparant celles qui présentent des incohérences majeures.

Ce mécanisme de feedback top-down permet de sortir d'un régime où des modifications locales répétées – par insertion et suppression successives de règles – conduiraient à des oscillations ou à un chaos logique, et assure ainsi une **stabilité** globale du SCN.

Conclusion

La détection des contradictions logiques dans un DSL symbolique est cruciale pour maintenir la **cohérence** et la **stabilité** des synergies entre entités. Nous avons défini formellement la notion de contradiction à travers la relation

$$R_i \cup R_j \vdash \perp,$$

et avons introduit des mesures telles que $\text{ContradictionCount}(R_i, R_j)$ pour quantifier le nombre de conflits entre les ensembles de règles. En intégrant cette mesure dans la fonction de synergie symbolique

$$S_{\text{sym}}(i, j) = \alpha \frac{|R_i \cap R_j|}{|R_i \cup R_j|} - \beta \text{ContradictionCount}(R_i, R_j),$$

le SCN est amené à ajuster la pondération $\omega_{i,j}$ selon la compatibilité logique entre les entités. Néanmoins, lorsque de nombreuses contradictions apparaissent, le système risque de se désorganiser localement.

Pour contrer ce phénomène, un mécanisme de **feedback top-down** (tel que détaillé dans le Chapitre 10) intervient pour réorganiser globalement les règles, soit en imposant la révision des axiomes conflictuels, soit en isolant les entités problématiques, soit en déclenchant une phase de re-clusterisation. Ce processus garantit que, malgré les mises à jour locales potentiellement contradictoires, le SCN demeure globalement cohérent et converge vers une structure de liens stable et explicable.

3.4.3.3. Exemple : un DSL stockant un “Ensemble de Postulats” qui se Modifie au Fil du Temps

Dans un **Deep Synergy Learning (DSL)** intégrant une composante symbolique, chaque entité \mathcal{E}_i est caractérisée par un ensemble de postulats ou d'axiomes, noté R_i . Ces postulats représentent la connaissance explicite associée à l'entité et, en fonction des évolutions du domaine ou de l'arrivée de nouvelles données, R_i peut évoluer de manière dynamique par l'insertion de nouveaux postulats ou la suppression d'anciens. Cette capacité à modifier le contenu de R_i en flux continu est essentielle pour que le **Synergistic Connection Network (SCN)** puisse adapter ses liaisons, c'est-à-dire ses pondérations $\omega_{i,j}$, en temps réel.

A. Contexte : Entités “Patients” avec Blocs de Connaissances Médicales

Considérons un système médical où chaque entité \mathcal{E}_i correspond à un patient et est associé à un bloc de connaissances R_i qui regroupe à la fois les observations cliniques et les règles diagnostiques pertinentes. Par exemple, pour un patient donné, R_i peut contenir des assertions telles que :

"Patient \mathcal{E}_i présente Toux"

"Patient \mathcal{E}_i a Fièvre(39°C)"

$R_i \ni \{\text{if Toux} \wedge \text{Fièvre then Infectieux}\}$

$R_i \ni \{\text{if Vacciné(Grippe) then RisqueRéduit}\}$

Ces ensembles R_i forment la base symbolique qui sera utilisée pour calculer la **synergie symbolique** entre les patients. En effet, deux patients \mathcal{E}_i et \mathcal{E}_j dont les ensembles de postulats présentent un fort recouvrement (c'est-à-dire, un grand nombre d'axiomes communs ou compatibles) auront une synergie $S(i, j)$ élevée, traduite par une pondération $\omega_{i,j}$ renforcée dans le SCN. À l'inverse, une faible similarité ou la présence de contradictions entraînera une diminution de $\omega_{i,j}$.

B. Ajout de Règles : Extension du Bloc de Connaissances

Au fil du temps, de nouvelles découvertes ou mises à jour dans le domaine médical peuvent conduire à l'ajout de nouveaux postulats dans le bloc de connaissances d'un patient. Cette opération d'insertion se formalise par :

$$R_i(t+1) = R_i(t) \cup \{p\},$$

où p représente le nouvel axiome, par exemple :

$p:\text{if PerteOdorat} \wedge \text{Fièvre then CovidSuspect}.$

L'insertion de p peut modifier la synergie symbolique entre le patient \mathcal{E}_i et un autre patient \mathcal{E}_j . Si, par exemple, R_j contient déjà une règle similaire ou compatible – telle que “if PerteOdorat \wedge Fièvre then InfectionVirale” – le recouvrement entre R_i et R_j augmente, ce qui se traduit par une augmentation de la synergie, notée $S_{\text{sym}}(i, j)$. Ainsi, la mise à jour de la pondération dans le SCN s'effectue selon la formule habituelle :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{sym}}(i, j) - \tau \omega_{i,j}(t)].$$

En revanche, si l'axiome p entre en contradiction avec des règles déjà présentes dans R_j , par exemple si R_j contient « if Fièvre \wedge PerteOdorat then NonInfectieux », alors la fonction de synergie se verra pénalisée, et $\omega_{i,j}$ diminuera en conséquence.

C. Suppression ou Révision de Règles : Retrait de Postulats

La suppression de règles intervient lorsque certains axiomes se révèlent obsolètes ou erronés. On définit alors l'opération de suppression par :

$$R_i(t+1) = R_i(t) \setminus \{q\},$$

où q est l'axiome retiré, par exemple une règle trop générale comme :

q:if Toux then Infectieux.

Le retrait de q peut modifier la synergie de deux entités. Si q contribuait à une co-occurrence importante entre R_i et R_j , sa suppression pourrait entraîner une diminution de $\mathbf{S}_{\text{sym}}(i, j)$ et, par la suite, une baisse de la pondération $\omega_{i,j}$. Inversement, si q était à l'origine d'une contradiction avec R_j , son retrait pourrait améliorer la compatibilité et renforcer $\omega_{i,j}$. La dynamique de mise à jour reste donc régie par :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta \left[\mathbf{S}_{\text{sym}}(R_i(t+1), R_j(t+1)) - \tau \omega_{i,j}(t) \right].$$

Cette équation illustre que toute modification dans le bloc R_i nécessite une réévaluation immédiate de la synergie avec les autres entités, afin de répercuter l'évolution de la base de connaissances dans la structure du SCN.

D. Gestion Continue et Versionnage

Dans un environnement en flux continu, il est indispensable de suivre l'évolution des blocs de règles afin de pouvoir revenir sur des changements si nécessaire et de garantir la cohérence du système. Pour ce faire, on introduit un mécanisme de **versionnage** :

$$R_i^{(t+1)} = \left(R_i^{(t)} \cup R_{\text{add}}^{(i,t)} \right) \setminus R_{\text{del}}^{(i,t)},$$

où $R_{\text{add}}^{(i,t)}$ représente l'ensemble des règles ajoutées à l'itération t et $R_{\text{del}}^{(i,t)}$ celles supprimées. Ce mécanisme permet de conserver un historique des états $\{R_i^{(0)}, R_i^{(1)}, \dots\}$, facilitant ainsi l'analyse rétroactive des décisions prises et, en cas de conflits majeurs, la possibilité d'un rollback. Un tel versionnage s'avère essentiel pour maintenir la cohérence globale et pour minimiser le coût de recalcul de la synergie dans de grands réseaux.

E. Risques d'Oscillation et Feedback Top-Down

Si les modifications des blocs R_i se produisent trop fréquemment ou de manière contradictoire, le SCN risque d'entrer dans une phase d'oscillations ou de chaos symbolique.

Par exemple, une entité \mathcal{E}_i pourrait insérer un axiome qui augmente temporairement la synergie avec \mathcal{E}_j , puis le retirer peu après, entraînant ainsi une fluctuation répétée de $\omega_{i,j}$. Pour éviter de telles instabilités, un **feedback top-down** peut intervenir (voir section 3.4.3.2).

Ce mécanisme de rétroaction consiste en un module de contrôle global qui surveille l'état du SCN et, si le nombre de contradictions ou le taux de variation des pondérations dépasse un seuil critique, il impose une révision ou une réorganisation forcée des règles.

Par exemple, on peut définir une condition de déclenchement :

si $\text{ContradictionCount}(t) > \theta$, alors initier une révision massive.

Une telle intervention permet d'isoler ou de corriger les entités présentant des incohérences, assurant ainsi la convergence de la dynamique et la formation de clusters stables.

Conclusion

L'exemple d'un **DSL** stockant un ensemble de postulats, qui se modifie au fil du temps, illustre de manière concrète comment la dynamique symbolique peut s'inscrire dans un flux continu d'actualisation des connaissances. Chaque entité \mathcal{E}_i est associée à un bloc de règles R_i qui évolue par des opérations d'insertion et de suppression, modifiant ainsi la synergie symbolique $\mathbf{S}_{\text{sym}}(i, j)$ entre les entités. La mise à jour des pondérations se fait selon la formule :

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta [\mathbf{S}_{\text{sym}}(i, j, t + 1) - \tau \omega_{i,j}(t)],$$

ce qui permet au SCN de s'ajuster en temps réel aux évolutions de la base de connaissances.

Toutefois, à grande échelle et en présence de modifications fréquentes, des mécanismes d'optimisation (tels que l'indexation sémantique et les mises à jour paresseuses) ainsi qu'un système de versionnage devient indispensable pour assurer la cohérence globale.

Par ailleurs, le recours à un feedback top-down est souvent nécessaire pour prévenir les oscillations ou le chaos symbolique, garantissant ainsi la stabilité du système.

En définitive, ce scénario démontre comment un DSL peut intégrer et exploiter l'évolution continue des connaissances symboliques pour adapter sa structure d'interactions, tout en maintenant une cohérence logique et une organisation auto-organisée du réseau.

3.5. Représentations Hybrides (Sub-Symbolique + Symbolique)

Les sections précédentes ont montré les **forces** et **faiblesses** respectives des représentations **sub-symboliques** (chap. 3.3) et **symboliques** (chap. 3.4). Dans la pratique, un **DSL** (Deep Synergy Learning) peut grandement tirer profit d'une **fusion** des deux approches — c'est-à-dire que chaque entité \mathcal{E}_i combine à la fois un **embedding** (capable de manipuler le bruit, les données massives, etc.) et un **bloc** de règles ou d'axiomes (garantissant l'interprétabilité, la cohérence logique). C'est ce qu'on appelle une **représentation hybride** ou **neuro-symbolique**, où l'on cherche à marier la **puissance statistique** et la **lisibilité** sémantique.

Dans la section 3.5, nous verrons la **motivation** (3.5.1) qui pousse à mélanger sub-symbolique et symbolique, puis nous discuterons (3.5.2) des **stratégies** concrètes pour concevoir un tel assemblage dans un DSL (comment définir $r(i)$ comme un couple ou deux nœuds, comment moduler la synergie mixte, etc.). Enfin, nous conclurons (3.5.3) sur les **avantages** (explicabilité, adaptabilité) et les **défis** (complexité, gestion des contradictions), en prenant un exemple d'**agent conversationnel** où la composante logique et l'embedding linguistique cohabitent.

3.5.1. Motivation de la Fusion

L'idée d'**hybrider** le sub-symbolique (vecteurs, embeddings neuronaux) et le symbolique (règles, ontologies) découle du constat que ni l'un ni l'autre ne suffisent, à eux seuls, à couvrir tous les besoins du DSL (Deep Synergy Learning). Les embeddings gèrent bien le **bruit** et la **variabilité**, mais souffrent d'un manque d'**interprétabilité** et ne fournissent pas de raisonnement logique direct ; les représentations symboliques, elles, sont **rigides** et intransigeantes face à l'ambiguïté ou l'imprécision, mais elles offrent un cadre formel pour **expliquer** et **inférer**.

3.5.1.1. BÉNÉFICIER DE LA PUISSANCE STATISTIQUE DES EMBEDDINGS ET DE LA CLARTÉ DES RÈGLES LOGIQUES

Dans le cadre d'un **Deep Synergy Learning (DSL)**, l'intégration d'**embeddings sub-symboliques** et de blocs de **règles logiques** offre un paradigme hybride capable de conjuguer la robustesse statistique des approches d'apprentissage profond et la transparence inhérente aux systèmes symboliques. Ce modèle hybride tire profit de la **puissance** des techniques de projection vectorielle tout en fournissant un **raisonnement explicite** qui facilite l'interprétation des résultats dans des domaines critiques tels que la médecine ou la surveillance industrielle.

A. Dimension Sub-symbolique et Extraction d'Embeddings

Chaque entité \mathcal{E}_i est associée à un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$ qui encode de manière dense ses attributs discriminants. Ces représentations issues d'architectures telles que les **CNN**, **Transformers** ou **autoencodeurs** permettent d'exploiter la **puissance statistique** des modèles d'apprentissage profond pour capturer les structures complexes dans des données massives et bruitées. Par exemple, la proximité entre deux entités peut être mesurée à l'aide de la **distance euclidienne** :

$$d_{\text{eucl}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|,$$

ou à l'aide d'un **noyau Gaussien** défini par

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\gamma > 0$ ajuste la sensibilité de la mesure. Ces formules illustrent la capacité des **embeddings** à tolérer un **bruit** modéré et à gérer un flux massif d'instances, en extrayant automatiquement des caractéristiques discriminantes pertinentes.

B. Dimension Symbolique et Règles Logiques

En complément de la représentation sub-symbolique, le DSL dote chaque entité \mathcal{E}_i d'un **bloc** de règles ou d'axiomes, noté $\{R_i\}$. Cette dimension symbolique permet d'initier un **raisonnement logique** explicite sur les entités. Les règles logiques offrent plusieurs avantages essentiels :

- Elles permettent d'identifier des **contradictions** ou des incohérences dans l'ensemble des connaissances.
- Elles facilitent l'**inférence explicite**, par exemple, en déduisant qu'« si A et B sont vraies, alors C l'est également ».
- Elles confèrent une **traçabilité** qui rend les décisions du système plus compréhensibles pour des experts dans des domaines où la **justification** des recommandations est indispensable.

Ainsi, dans des domaines tels que le **médical**, le bloc de règles peut encapsuler des protocoles diagnostiques (par exemple, « si **Toux**, **Fièvre** et saturation en oxygène $O2 < 92\%$, alors **Admission Urgente** »), offrant ainsi un **langage intelligible** aux experts.

C. Fusion Hybride : La Synergie Globale

Pour exploiter au mieux ces deux dimensions complémentaires, le DSL hybride définit la **synergie globale** entre deux entités \mathcal{E}_i et \mathcal{E}_j comme un **mélange** pondéré des similarités sub-symboliques et symboliques. Cette synergie est formalisée par l'équation suivante :

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

où $\alpha \in [0, 1]$ est un paramètre de **pondération** qui permet de calibrer l'influence relative de chaque composante. La fonction S_{sub} évalue la **similarité vectorielle** (par exemple, en utilisant la formule $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ ou le **cosinus normalisé**), tandis que S_{sym} mesure la **compatibilité symbolique** entre les ensembles de règles $\{R_i\}$ et $\{R_j\}$ en tenant compte de critères tels que la **co-occurrence** des axiomes, l'absence de contradiction ou encore la **distance ontologique** entre les concepts.

Cette approche hybride permet d'obtenir une mesure de **synergie** qui bénéficie simultanément de la **robustesse statistique** et de la **clarté logique**, assurant ainsi que les entités se rapprochent dans le réseau non seulement lorsqu'elles sont proches sur le plan des **embeddings** (c.-à-d. lorsque $\mathbf{x}_i \approx \mathbf{x}_j$) mais également lorsqu'elles partagent des **règles logiques compatibles**.

D. Mise à Jour des Pondérations et Auto-Organisation

L'architecture du DSL utilise la synergie hybride dans la mise à jour dynamique des pondérations entre entités. La règle de mise à jour s'exprime par :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{hybrid}}(i,j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ représente le **taux d'apprentissage** et $\tau > 0$ est un **coefficent de décroissance** qui assure la stabilité des poids. Ainsi, le système procède à une **auto-organisation** dans laquelle les liaisons se renforcent si la **synergie hybride** est élevée et s'affaiblissent dans le cas contraire, aboutissant à des **clusters** d'entités cohérents tant du point de vue statistique que logique.

E. Applications dans des Domaines Critiques

L'intérêt d'une telle approche hybride se manifeste particulièrement dans des domaines où la **précision** et la **transparence** des décisions sont primordiales. Par exemple :

- Dans le **médical**, la partie sub-symbolique du DSL peut encoder des données brutes telles que des images médicales, des relevés cliniques ou des textes issus de l'analyse NLP, tandis que la composante symbolique intègre des règles diagnostiques éprouvées. Cette dualité permet de détecter des corrélations complexes tout en offrant des justifications claires pour chaque diagnostic.
- Dans la **surveillance industrielle**, les vecteurs décrivant des signaux (températures, vibrations) sont couplés à des règles de sécurité et de contrôle, garantissant ainsi une détection fine des pannes et la certification que les limites opérationnelles ne sont pas dépassées.

F. Conclusion

L'approche hybride proposée par le DSL permet de surmonter les limites d'un système purement sub-symbolique, souvent critiqué pour son **opacité**, et d'un système exclusivement symbolique, qui peut manquer de **souplesse** face aux signaux bruités et à la variabilité des données massives. En combinant la **puissance statistique** des **embeddings** et la **clarté** des **règles logiques**, le DSL hybride offre une solution robuste et **explicable**. Ce modèle permet non seulement une auto-organisation efficace par renforcement des liens entre entités similaires, mais fournit également une **träçabilité** indispensable dans des contextes où le **contrôle** et la **transparence** sont essentiels pour la prise de décision.

Ainsi, en conjuguant ces deux dimensions complémentaires, le DSL hybride confère au réseau la **capacité** de s'auto-organiser de manière **robuste** et **explicable**, répondant aux exigences des domaines critiques et offrant un avantage concurrentiel tant sur le plan de la performance que de l'interprétabilité.

3.5.1.2. APPROCHE “NEURO-SYMBOLIQUE” : DÉJÀ MENTIONNÉE DANS LA LITTÉRATURE IA

L’approche **neuro-symbolique** se présente comme une stratégie intégrative visant à fusionner la **puissance statistique** des techniques d’**embeddings neuronaux** avec la **rigueur logique** des systèmes symboliques, c’est-à-dire l’ensemble de règles ou d’axiomes formels. Cette démarche, qui a été explorée dès les années 1980-1990 (voir notamment les travaux de *Smolensky*), a connu un regain d’intérêt avec la montée du **deep learning** dans les années 2010. Dans le contexte d’un **Deep Synergy Learning (DSL)**, cette approche permet d’associer, dans un même système, un apprentissage efficace sur de larges volumes de données bruitées à un raisonnement explicite assurant **traçabilité** et **cohérence** globale.

A. Dimension Sub-symbolique : Puissance des Embeddings Neuronaux

Dans le sous-système sub-symbolique, chaque entité \mathcal{E}_i est représentée par un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$ issu d’un processus d’extraction automatique de caractéristiques via des réseaux de neurones profonds tels que les **CNN**, **Transformers** ou **autoencodeurs**. Ces **embeddings** permettent d’obtenir des représentations continues qui capturent des propriétés discriminantes des données. La proximité entre deux entités peut être quantifiée par des mesures telles que la **distance euclidienne** :

$$d_{\text{eucl}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|,$$

ou par un **noyau Radial Basis Function (RBF)** défini par

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\gamma > 0$ est un paramètre d’échelle. Ces mesures sont particulièrement adaptées pour gérer des données massives et bruyantes, puisque les **embeddings** permettent de projeter des instances dans un espace où les structures sous-jacentes sont révélées par des similitudes quantitatives. La tolérance au bruit et la scalabilité de ces méthodes confèrent au système une **robustesse** statistique essentielle dans le traitement de flux de données complexes.

B. Dimension Symbolique : Rigueur Logique et Raisonnement Formatif

Parallèlement au traitement sub-symbolique, le DSL intègre une dimension symbolique qui se matérialise par un **bloc** de règles ou d’**axiomes** associé à chaque entité, noté R_i . Ce sous-système symbolique permet d’effectuer un **raisonnement explicite** et de garantir la **cohérence** des inférences. Par exemple, un ensemble de règles peut être formulé pour détecter des contradictions ou pour valider la cohérence d’un ensemble de postulats. Dans des domaines exigeant une **explicabilité** accrue, comme le domaine médical ou industriel, la capacité à justifier une décision à l’aide d’un langage formel (par exemple, « si A et B sont vraies, alors C doit l’être ») est un atout majeur.

La **dimension symbolique** permet également d’opérer des inférences en se basant sur des formalismes tels que les logiques classiques, la logique floue ou des systèmes experts, assurant ainsi que le raisonnement ne repose pas uniquement sur des corrélations statistiques, mais sur des critères explicites et traçables.

C. Fusion Hybride : Définition de la Synergie Globale

Pour conjuguer harmonieusement ces deux dimensions complémentaires, un **DSL hybride** définit la **synergie globale** entre deux entités \mathcal{E}_i et \mathcal{E}_j par une combinaison linéaire pondérée. Cette synergie est formalisée par l'équation

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

où $\alpha \in [0,1]$ module l'influence relative de la composante sub-symbolique par rapport à la composante symbolique. La fonction S_{sub} peut être, par exemple, une mesure de **similarité vectorielle** telle que

$$S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

ou une mesure basée sur le **cosinus normalisé**. Parallèlement, S_{sym} évalue la **compatibilité symbolique** entre les blocs de règles R_i et R_j en tenant compte d'éléments tels que la co-occurrence des règles, l'absence de contradiction ou la proximité ontologique entre les concepts impliqués. Ce modèle hybride permet ainsi aux entités de se regrouper dans le réseau non seulement en fonction de leur **proximité** dans l'espace vectoriel, mais aussi en fonction de leur **compatibilité logique**.

D. Auto-organisation du SCN Basée sur la Synergie Hybride

Le **Synergistic Connection Network (SCN)** intègre la synergie hybride dans la mise à jour dynamique de ses **pondérations** $\omega_{i,j}$. La règle de mise à jour adoptée est généralement de la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{hybrid}}(i, j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ représente le **taux d'apprentissage** et $\tau > 0$ un **coefficent de décroissance** permettant de réguler l'évolution des poids. Cette équation assure que les connexions entre entités se renforcent lorsque la synergie hybride est élevée – c'est-à-dire lorsque les entités sont à la fois proches du point de vue des **embeddings** et compatibles du point de vue des règles – et s'affaiblissent dans le cas contraire. Ainsi, le SCN s'auto-organise pour former des **clusters** d'entités cohérents, caractérisés par des liens forts qui résultent d'un double critère d'affinité.

E. Références Historiques et Applications Marquantes

L'**approche neuro-symbolique** s'inscrit dans un courant de recherche de longue date dans le domaine de l'**IA**. Dès les années 1980-1990, des chercheurs tels que *Smolensky* ont proposé des idées visant à relier les **représentations discrètes** aux modèles neuronaux, soulignant l'intérêt de combiner les forces des approches **connexionnistes** et **symboliques**. Avec l'essor du **deep learning**, de nombreux frameworks tels que **DeepProbLog** ou des méthodes de **Differentiable Reasoners** ont émergé, proposant des solutions pour intégrer la logique au sein de réseaux de neurones différentiables. De plus, le couplage d'ontologies (par exemple via OWL) avec des **embeddings** a permis d'enrichir les modèles de raisonnement en introduisant des connaissances formelles.

Les applications de l'approche neuro-symbolique sont nombreuses :

- Dans le domaine **médical**, l'intégration d'un module d'**embedding** pour l'analyse d'images ou de textes cliniques avec des règles diagnostiques permet d'expliquer les décisions d'un système et d'assurer une **explicabilité** indispensable.
- Dans un contexte **industriel**, la gestion de flux sensoriels massifs combinée à des règles de sécurité garantit une **tolérance** au bruit tout en assurant le respect des normes.
- En **linguistique**, le couplage d'un **transformer** avec des règles syntaxiques ou sémantiques permet de concilier l'**apprentissage** des représentations linguistiques et la rigueur de la grammaire explicite.

F. Difficultés et Interfaces Entre Systèmes Neuronaux et Symboliques

L'interface entre le sous-système **différentiable** (neuronale) et le sous-système **non différentiable** (symbolique) représente un défi majeur. Tandis que la partie neuronale se met à jour par rétropropagation et bénéficie de gradients continus, la partie symbolique relève souvent d'un raisonnement binaire. Des approches telles que la logique floue, la sémantique continue des règles ou l'intégration de solveurs SAT dans des architectures neuronales ont été proposées pour atténuer cette discontinuité et permettre une meilleure **intégration** des deux mondes. Par ailleurs, la gestion simultanée de nombreux **embeddings** et de multiples blocs de règles pose des problèmes de **complexité**, nécessitant des stratégies d'optimisation pour garantir l'évolutivité du système.

. Conclusion

L'**approche neuro-symbolique** se présente comme une voie prometteuse pour allier les avantages des méthodes **connexionsnistes** – telles que la **tolérance** au bruit, la **scalabilité** et la capacité d'**apprentissage** sur de larges ensembles de données – aux atouts des systèmes **symboliques**, notamment la **träçabilité**, la **cohérence** et la capacité d'**inférence** explicative. Dans le cadre d'un **DSL**, cette intégration se matérialise par la formulation de la synergie hybride

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

et par l'adaptation des pondérations du SCN via

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{hybrid}}(i, j) - \tau \omega_{i,j}(t)].$$

Cette intégration donne naissance à des systèmes d'**IA robustes** capables de manipuler des données complexes tout en garantissant une **explicabilité** et une **cohérence** conceptuelle, confirmant ainsi l'intérêt soutenu de la recherche neuro-symbolique pour la construction d'algorithmes IA complets et explicables.

3.5.2.1. Entité = Couple (\mathbf{x}_i, R_i) ? Ou Bien Deux Nœuds Connectés ?

L'intégration simultanée de la **partie sub-symbolique** (les embeddings) et de la **partie symbolique** (les règles logiques) dans un **DSL (Deep Synergy Learning)** peut être réalisée suivant deux stratégies architecturales distinctes dans la construction du **Synergistic Connection Network (SCN)**. Ces deux approches se distinguent par la manière dont elles structurent les entités du réseau

et par la granularité à laquelle la double information – à la fois sub-symbolique et symbolique – est traitée.

A. Entité = Couple (\mathbf{x}_i , R_i)

Dans cette première configuration, chaque entité \mathcal{E}_i est modélisée comme un **couple** unique constitué d'un **embedding** $\mathbf{x}_i \in \mathbb{R}^d$ et d'un ensemble de règles ou axiomes symboliques R_i . Le vecteur \mathbf{x}_i peut être obtenu par divers mécanismes d'extraction automatique (par exemple, un réseau de neurones convolutif pour l'analyse d'images, un Transformer pour le traitement de texte ou encore un autoencodeur pour d'autres types de signaux), tandis que R_i recense des connaissances explicites telles que des règles logiques, des axiomes ou des déclarations formelles qui décrivent l'entité.

Dans ce schéma, le SCN comporte exactement n nœuds pour n entités, chaque nœud encapsulant simultanément la dimension sub-symbolique et la dimension symbolique. La **synergie** $S(i, j)$ entre deux entités \mathcal{E}_i et \mathcal{E}_j est alors évaluée à partir d'une combinaison des similarités issues de leurs embeddings et de la compatibilité de leurs blocs de règles. Ainsi, le calcul de $S(i, j)$ intègre par exemple des mesures telles que

$$S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j)$$

pour quantifier la proximité dans l'espace vectoriel, et

$$S_{\text{sym}}(R_i, R_j)$$

pour juger de l'absence de contradiction ou de la co-occurrence des axiomes associés.

Un avantage majeur de cette approche réside dans sa **simplicité** architecturale, puisque l'on ne duplique pas le nombre de nœuds dans le réseau. Chaque entité gère en interne la double information, ce qui facilite une mise à jour unifiée des pondérations $\omega_{i,j}$ selon une règle d'auto-organisation unique. Toutefois, cette fusion présente également des inconvénients où la gestion conjointe du vecteur \mathbf{x}_i et du bloc R_i peut complexifier le processus de réapprentissage, notamment si l'on souhaite réentraîner la composante sub-symbolique indépendamment des règles logiques. La modularité interne peut ainsi s'avérer moins flexible, car la mise à jour globale repose sur un seul nœud regroupant des modalités hétérogènes.

B. Deux Nœuds Connectés : L'Entité Sub-Symbolique et l'Entité Symbolique

La seconde stratégie consiste à dissocier les deux composantes d'une entité \mathcal{E}_i en deux **nœuds** distincts, chacun spécialisé dans l'une des deux modalités. Ainsi, on définit :

$$\begin{aligned}\mathcal{E}_i^{(\text{sub})} &: \text{embedding neuronal } \mathbf{x}_i \in \mathbb{R}^d, \\ \mathcal{E}_i^{(\text{sym})} &: \text{bloc de règles ou axiomes } R_i.\end{aligned}$$

Ces deux nœuds, qui représentent respectivement la partie sub-symbolique et la partie symbolique de l'entité, sont interconnectés par un **lien** explicite ou un identifiant commun qui permet d'indiquer qu'ils décrivent ensemble la même entité \mathcal{E}_i . Dans le SCN ainsi constitué, il est alors possible de distinguer deux sous-réseaux où un sous-réseau sub-symbolique associe les pondérations $\omega_{(i^{(\text{sub})}, j^{(\text{sub})})}$ à la proximité vectorielle, par exemple via la norme $\|\mathbf{x}_i - \mathbf{x}_j\|$, tandis

qu'un sous-réseau symbolique quantifie la compatibilité entre les ensembles de règles R_i et R_j à travers les pondérations $\omega_{(i^{(\text{sym})}, j^{(\text{sym})})}$. Par la suite, un mécanisme de **fusion** combine ces deux scores pour restituer la synergie globale $S(i, j)$.

L'approche à deux nœuds offre une meilleure **modularité** puisqu'elle permet de traiter séparément et d'optimiser indépendamment les deux sous-systèmes. Par exemple, il devient aisément de réentraîner la partie d'extraction d'embeddings sans impacter le module symbolique, ou inversement. Néanmoins, cette dissociation a pour conséquence de doubler le nombre de nœuds dans le réseau (passant de n à $2n$ pour n entités), ce qui peut accroître le coût de calcul lors de la mise à jour des pondérations. De plus, il faut élaborer une stratégie explicite pour fusionner les scores sub-symbolique et symbolique, c'est-à-dire déterminer la fonction de combinaison qui produira le score final de synergie entre deux entités.

C. Choix Pratique et Impact Architectural

Le choix entre la modélisation d'une entité en tant que **couple unique** (\mathbf{x}_i, R_i) et la dissociation en deux **nœuds distincts** dépend de plusieurs considérations pratiques :

- **Échelle du Système** : Dans un DSL comportant un grand nombre d'entités, la multiplication du nombre de nœuds (passage de n à $2n$) peut alourdir la structure du SCN et impacter la complexité computationnelle de la mise à jour des pondérations.
- **Modularité et Flexibilité** : Si l'on souhaite réentraîner ou modifier indépendamment la composante sub-symbolique (les embeddings) et la composante symbolique (les règles), la séparation en deux nœuds offre une meilleure modularité et facilite la gestion de la mise à jour de chacun des sous-systèmes.
- **Simplicité de l'Implémentation** : Un nœud mixte, qui intègre directement les deux composantes dans un unique vecteur complexe, présente l'avantage d'une architecture plus simple et d'une auto-organisation unifiée. Cependant, cette simplicité peut être contrebalancée par une complexification dans la gestion interne des informations hétérogènes.

Dans les deux cas, la mise à jour des pondérations suit une règle de la forme

$$\omega_{a,b}(t+1) = \omega_{a,b}(t) + \eta [\text{synergie}(a, b) - \tau \omega_{a,b}(t)],$$

où le terme $\text{synergie}(a, b)$ est défini différemment selon que a et b représentent des nœuds mixtes (cas A) ou des nœuds séparés (cas B). La fonction de fusion, qui combine la similarité sub-symbolique et la compatibilité symbolique, doit être soigneusement conçue pour garantir que le SCN converge vers une configuration stable et interprétable.

Conclusion

Les deux stratégies pour intégrer simultanément la partie sub-symbolique et la partie symbolique dans un DSL se distinguent par leur approche de la représentation des entités. La première option, qui définit chaque entité \mathcal{E}_i comme un **couple** (\mathbf{x}_i, R_i), offre une solution unifiée et simple en termes de nombre de nœuds, mais peut limiter la modularité en imposant une gestion conjointe des deux types d'informations. La seconde option, qui dissocie l'entité en deux nœuds distincts – l'un pour

l'**embedding neuronal** et l'autre pour le **bloc symbolique** – apporte une meilleure séparation des préoccupations et une flexibilité accrue dans la mise à jour de chaque sous-système, au prix d'une duplication du nombre de nœuds et d'une complexité de fusion supplémentaire.

Le choix entre ces deux approches dépend donc des contraintes architecturales, de la taille du réseau et des exigences en matière de modularité et de réapprentissage. Le reste de la section (3.5.2.2) détaillera la **formule** de synergie mixte qui permet de fusionner les informations sub-symboliques et symboliques, tandis que la section (3.5.2.3) illustrera l'application de ce schéma dans des cas pratiques exploitant la force statistique des embeddings et la clarté du raisonnement symbolique.

3.5.2.2. Fonctions de Synergie Mixtes : $\mathbf{S}_{\text{sub, sym}}$

Dans le contexte d'un **Deep Synergy Learning (DSL)**, la nécessité de mesurer la proximité ou la compatibilité entre deux entités \mathcal{E}_i et \mathcal{E}_j se complexifie lorsqu'on intègre simultanément une composante sub-symbolique – représentée par des **embeddings** \mathbf{x}_i – et une composante symbolique – exprimée par un ensemble de règles ou axiomes R_i . Afin d'obtenir un **score global** qui reflète à la fois la similarité dans l'espace vectoriel et la compatibilité logique, il convient de définir une fonction de synergie mixte, notée $\mathbf{S}_{\text{hybrid}}(i, j)$. Cette section présente diverses approches d'agrégation qui permettent de fusionner ces deux dimensions, chacune offrant des compromis entre **robustesse statistique** et **explicabilité logique**.

A. Formule Linéaire Simple

La méthode la plus intuitive consiste à combiner les deux scores de manière **linéaire**. Ainsi, la synergie globale peut être définie par :

$$\mathbf{S}_{\text{hybrid}}(i, j) = \alpha \mathbf{S}_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) \mathbf{S}_{\text{sym}}(R_i, R_j),$$

où $\alpha \in [0, 1]$ est un paramètre de pondération ajustable. La fonction \mathbf{S}_{sub} évalue la **similarité vectorielle** – par exemple, via une mesure de similarité cosinus ou une fonction de noyau Gaussien telle que

$$\mathbf{S}_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2),$$

avec $\gamma > 0$ –, tandis que \mathbf{S}_{sym} quantifie la **compatibilité logique** entre les blocs de règles R_i et R_j (en évaluant, par exemple, la co-occurrence des axiomes ou en appliquant des tests de contradiction). Une telle combinaison linéaire présente l'avantage d'être aisément interprétable où, en modulant α , il est possible de privilégier l'une ou l'autre des composantes en fonction du contexte d'application. Par ailleurs, une variation affine peut être introduite afin d'ajuster les échelles des scores issus des deux sous-systèmes :

$$\mathbf{S}_{\text{hybrid}}(i, j) = \alpha g(\mathbf{S}_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j)) + (1 - \alpha) h(\mathbf{S}_{\text{sym}}(R_i, R_j)),$$

où les fonctions g et h assurent un recalibrage permettant d'éviter que l'une des composantes n'écrase l'autre en cas de plages de valeurs très différentes.

B. Combinaison Multiplicative ou Min–Max

Une autre approche consiste à opter pour une agrégation **multiplicative**, dans laquelle le score global est obtenu en multipliant les scores individuels :

$$\mathbf{S}_\times(i, j) = \mathbf{S}_{\text{sub}}(i, j) \times \mathbf{S}_{\text{sym}}(i, j).$$

Cette formulation exige que chacune des deux composantes prenne une valeur élevée pour que le produit soit important. Ainsi, si l'une des deux valeurs est proche de zéro, le score global s'en trouve fortement diminué, ce qui renforce l'idée d'une nécessité d'accord simultané entre la partie sub-symbolique et la partie symbolique. Une alternative au produit consiste à utiliser l'opérateur min, définissant le score global comme le minimum des deux scores :

$$\mathbf{S}_{\min}(i, j) = \min\{\mathbf{S}_{\text{sub}}(i, j), \mathbf{S}_{\text{sym}}(i, j)\}.$$

Cette formulation, tout en partageant l'esprit de l'approche multiplicative, présente un comportement légèrement différent en ce qu'elle retient la composante la plus faible. Par conséquent, un faible résultat dans l'une des dimensions impose directement une limitation sur la synergie globale.

C. Stratification ou Pondération Conditionnelle

Une troisième piste consiste à introduire une **pondération conditionnelle** ou une stratification, c'est-à-dire à faire intervenir l'une des composantes uniquement lorsque l'autre satisfait un critère minimal de fiabilité. Par exemple, on peut définir une fonction de synergie hybride conditionnelle comme suit :

$$\mathbf{S}_{\text{hybrid}}(i, j) = \begin{cases} \mathbf{S}_{\text{sub}}(i, j), & \text{si } \mathbf{S}_{\text{sub}}(i, j) < \theta, \\ \alpha \mathbf{S}_{\text{sub}}(i, j) + (1 - \alpha) \mathbf{S}_{\text{sym}}(i, j), & \text{sinon,} \end{cases}$$

où θ est un seuil prédéfini. Dans cette configuration, la composante symbolique n'intervient que lorsque la similarité sub-symbolique atteint un certain niveau, assurant ainsi que le score global reflète d'abord une exclusion basée sur le critère vectoriel avant d'intégrer une pondération combinée. De plus, il est envisageable d'introduire une fonction de pondération dynamique, $\alpha(i, j)$, qui évolue en fonction de la fiabilité ou de la cohérence de l'embedding \mathbf{x}_i ou du bloc R_i , permettant ainsi un ajustement local et adaptatif de la fusion des scores.

D. Impact sur la Mise à Jour des Pondérations dans le DSL

Quel que soit le schéma d'agrégation choisi – qu'il s'agisse de la somme linéaire, du produit, du minimum ou d'une combinaison conditionnelle – la règle de mise à jour des pondérations au sein du SCN demeure inchangée :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [\mathbf{S}_{\text{hybrid}}(i, j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ est le taux d'apprentissage et $\tau > 0$ le coefficient de décroissance. Dans ce cadre, la fonction $\mathbf{S}_{\text{hybrid}}(i, j)$ remplace une mesure unidimensionnelle de similarité en offrant une évaluation composite qui prend en compte à la fois la **proximité** dans l'espace des embeddings et la **compatibilité** des ensembles de règles. Ainsi, une paire d'entités renforce sa connexion dans le réseau uniquement si les deux critères – la similarité sub-symbolique et la cohérence symbolique

– sont satisfaits, renforçant ainsi l'idée d'un accord simultané nécessaire pour obtenir un score élevé. Le choix du schéma d'agrégation détermine alors la sensibilité du système aux valeurs extrêmes et la manière dont les imperfections dans l'une des composantes affectent globalement la synergie.

Conclusion

Les différentes fonctions de synergie mixtes présentées ici constituent des outils essentiels pour fusionner les informations provenant des domaines sub-symbolique et symbolique dans un DSL. La formulation linéaire simple offre une solution claire et modulable, tandis que les approches multiplicatives ou basées sur le minimum imposent une exigence d'accord simultané plus stricte. Les stratégies de pondération conditionnelle, quant à elles, permettent une intégration adaptative qui tient compte de la fiabilité relative des deux sources d'information. Quel que soit le choix effectué, l'agrégation se traduit dans la mise à jour des pondérations du SCN, garantissant que le réseau s'auto-organise de manière à renforcer les liens entre des entités à la fois proches dans l'espace vectoriel et compatibles sur le plan logique. Cette intégration fine des deux modalités confère au DSL une robustesse et une explicabilité particulièrement appréciables dans des applications critiques, où la convergence d'une **puissance statistique** et d'un **raisonnement formel** constitue un avantage majeur.

3.5.2.3. Applications : Bases de Connaissance Enrichies, Multi-sensoriels et Concepts

Dans le cadre d'un **Deep Synergy Learning (DSL)** hybride, l'intégration simultanée des informations sub-symboliques – représentées par des **embeddings** extraits de données brutes – et des informations symboliques – traduites par des **règles** ou des axiomes formels – permet de créer des systèmes capables de traiter des environnements complexes tout en garantissant la **cohérence** et l'**explicabilité** de leurs inférences. Cette section examine plusieurs domaines d'application de ce paradigme hybride, en illustrant comment la fusion de ces deux types de représentations enrichit les capacités d'un système et améliore la qualité de ses clusters ou associations.

A. Bases de Connaissance Enrichies

Une application classique de cette approche hybride apparaît dans la construction de **bases de connaissance (Knowledge Bases, KB)**. Dans ces systèmes, chaque entité \mathcal{E}_i se voit attribuer à la fois un **vecteur** $\mathbf{x}_i \in \mathbb{R}^d$ issu d'un traitement sub-symbolique – par exemple, via des méthodes d'apprentissage profond appliquées à des textes ou des images – et un **bloc** de règles R_i qui formalise des assertions ou des relations symboliques (telles que des triplets RDF ou des graphes ontologiques). L'objectif consiste à conjuguer la **fiabilité formelle** (mesurée par la cohérence et la non-contradiction des règles) avec la capacité de détection de motifs latents offerte par les embeddings.

Ainsi, lorsque le DSL évalue la synergie globale

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

une forte similarité vectorielle (i.e. $\mathbf{x}_i \approx \mathbf{x}_j$) peut conduire à un rapprochement même en l'absence d'un lien explicite dans le domaine symbolique. Inversement, une forte **compatibilité** ontologique,

où $R_i \cup R_j$ est hautement cohérent, peut compenser un manque relatif d'information dans l'extraction d'embeddings. Des applications telles que **Wikidata** illustrent ce phénomène, chaque entité (lieu, personne, concept) étant ainsi dotée d'une double description permettant au DSL de révéler des clusters enrichis et dynamiques, facilitant la navigation et l'exploitation d'un vaste ensemble de connaissances hétérogènes.

B. Applications Multi-Sensoriels et Concepts

Dans des environnements **multi-sensoriels**, où les données proviennent de sources variées (images, vidéos, audio, signaux de capteurs physiques), la combinaison d'un **embedding** sub-symbolique et d'un bloc symbolique offre un avantage décisif pour la détection d'anomalies ou la catégorisation de contextes.

La partie sub-symbolique, obtenue par des modèles tels que les CNN pour l'imagerie ou des transformateurs pour l'audio et le texte, capture la **structure** complexe des signaux bruts. Parallèlement, le bloc symbolique R_i encode des concepts ou des règles – par exemple, des spécifications indiquant que « la température dépasse 50°C, donc il existe un risque de surchauffe » ou « cette séquence vidéo se situe dans une ZoneInterdite » – qui permet d'interpréter ces signaux de manière explicite.

Dans un tel contexte, un DSL industriel de **surveillance** ou en **robotique** peut ainsi renforcer les liens entre entités uniquement lorsque la **cohérence** est constatée à la fois sur le plan sensoriel (embeddings similaires) et sur le plan conceptuel (règles non contradictoires).

Cette double validation conduit à la formation de clusters robustes, capables de détecter avec précision des événements anormaux ou des contextes critiques. Par exemple, dans un système de surveillance, les signaux analogues détectés par différents capteurs pourront être rapprochés s'ils respectent en parallèle des conditions explicites définies dans les règles de sécurité.

C. Cas d'Étude et Bénéfices

Plusieurs cas d'étude illustrent l'intérêt d'une approche hybride :

En Médecine : Un dossier patient peut être représenté par un **embedding** \mathbf{x}_i issu d'images médicales, de relevés cliniques ou de textes, complété par un bloc R_i contenant des règles diagnostiques (par exemple, « si Toux, Fièvre et saturation en oxygène $O_2 < 92\%$, alors admission en urgence »). La synergie mixte permet ainsi de regrouper des patients présentant des caractéristiques similaires tout en validant ces regroupements par la cohérence des antécédents ou des protocoles médicaux.

En Droit ou en Compliance : Dans la gestion de contrats ou de documents juridiques, le texte brut d'un document peut être transformé en un vecteur \mathbf{x}_i et associé à des règles précises clarifiant des clauses ou des obligations. Le DSL rapproche des documents non seulement par la similitude de leur contenu, mais aussi par le respect des critères légaux ou contractuels, facilitant ainsi la vérification et la conformité.

En Surveillance Industrielle : Dans un environnement industriel, les signaux provenant de capteurs (température, vibrations, etc.) sont traités par des modèles neuronaux pour extraire des embeddings, tandis que des règles de sécurité précisent les seuils critiques ou les conditions

d'alerte. Le DSL hybride permet alors de détecter des anomalies en associant la similarité des signaux à la validation des règles opérationnelles.

L'avantage clé d'un DSL hybride réside dans la **complémentarité** des deux approches où la partie sub-symbolique offre une grande capacité d'**apprentissage** et de découverte de nouveaux motifs en présence de données bruitées et hétérogènes, tandis que la composante symbolique assure une **cohérence** et une **explicabilité** indispensables dans des applications critiques. Le résultat est un système capable de constituer des clusters ou des associations d'entités plus fiables et interprétables.

Conclusion

Les applications d'un **DSL hybride** se déploient de manière efficace dans des domaines aussi variés que les bases de connaissances enrichies, la surveillance multi-sensorielle ou des environnements nécessitant un raisonnement formel (médical, juridique, industriel). En combinant pour chaque entité une représentation sub-symbolique \mathbf{x}_i et un bloc de règles R_i , le système bénéficie simultanément de la **puissance** des techniques d'apprentissage profond et de la **rigueur** des approches symboliques. Les fonctions de synergie mixtes, telles que celles décrites dans la section 3.5.2.2, permettent d'auto-organiser un réseau – le **SCN** – où les liens $\omega_{i,j}$ se renforcent uniquement lorsque les deux dimensions convergent vers une **similitude** forte et une **compatibilité** logique élevée. Ce paradigme assure ainsi une meilleure gestion des environnements complexes, en offrant à la fois robustesse, flexibilité et explicabilité dans l'analyse et le regroupement des entités.

3.5.3. Avantages et Défis

Après avoir exploré (3.5.1) la **motivation** pour fusionner sub-symbolique et symbolique, et (3.5.2) les **stratégies** possibles (comment construire une entité hybride et définir la synergie mixte), nous abordons maintenant les **avantages** (3.5.3.1) et les **défis** (3.5.3.2) qu'implique une telle fusion dans le cadre d'un **DSL** (Deep Synergy Learning). Nous conclurons (3.5.3.3) par un **cas d'usage** illustratif (agent conversationnel logique + embedding).

3.5.3.1. Meilleure Explicabilité et Adaptabilité Statistique

La représentation hybride, qui associe un **embedding** neuronal \mathbf{x}_i à un **bloc logique** R_i , confère à un **Deep Synergy Learning (DSL)** une capacité accrue à rendre ses décisions et ses regroupements à la fois **transparents** et **adaptatifs**. Dans un système purement sub-symbolique, la similarité entre deux entités \mathcal{E}_i et \mathcal{E}_j se mesure essentiellement par des critères quantitatifs, tels que la distance euclidienne

$$\| \mathbf{x}_i - \mathbf{x}_j \|,$$

ou la similarité cosinus, qui, bien que efficaces pour capturer des corrélations statistiques, restent difficiles à interpréter de manière sémantique. L'intégration d'un bloc symbolique, consistant en un ensemble de règles, axiomes ou ontologies, apporte un éclairage supplémentaire permettant de

comprendre la **raison** d'un rapprochement entre deux entités. Ainsi, si deux dossiers patients présentent des embeddings similaires, c'est la présence de règles diagnostiques communes ou la cohérence de leurs antécédents qui justifiera leur regroupement. Cette double dimension – **adaptation statistique et explicabilité logique** – offre un justificatif explicite et vérifiable aux experts.

L'**explicabilité** se matérialise par la capacité du système à lister ou à mettre en évidence les règles partagées, à démontrer l'absence de contradictions et à exposer le recouvrement ontologique entre deux entités. Dans un contexte médical, par exemple, la partie sub-symbolique peut établir une similarité basée sur des images scannées ou des relevés vitaux, tandis que le bloc symbolique clarifie que les dossiers patients se rapprochent parce qu'ils satisfont les mêmes critères diagnostiques, tels que définis par des règles explicites. Cette transparence renforce la confiance dans le système et facilite le contrôle des décisions prises par le réseau.

Sur le plan de l'**adaptabilité statistique**, l'association d'un embedding neuronal à un cadre logico-symbolique permet de bénéficier d'un apprentissage continu et évolutif. La partie sub-symbolique, grâce à des mécanismes tels que le fine-tuning ou le ré-entraînement partiel, est capable de s'ajuster en fonction des nouveaux flux de données, absorbant ainsi la variabilité et le **bruit** inhérents à un grand volume d'informations. Par exemple, lorsque de nouvelles données apparaissent, les vecteurs \mathbf{x}_i se réajustent pour intégrer de nouveaux motifs ou corrélations, tout en restant encadrés par la composante symbolique qui assure la **stabilité conceptuelle**. En effet, la couche symbolique agit comme une contrainte qui interdit des dérives excessives en imposant des règles immuables ou des axiomes fondamentaux, ce qui évite une divergence inappropriée de l'espace des embeddings.

L'**interaction** entre ces deux composantes crée un couplage dynamique dans lequel les découvertes statistiques peuvent mener à la mise à jour ou à l'enrichissement des règles existantes. À l'inverse, des ajustements dans le bloc symbolique, par exemple la création d'un nouvel axiome fondé sur des observations répétées, peuvent orienter la réorganisation des embeddings, en forçant des rapprochements ou des éloignements dans l'espace \mathbb{R}^d . Ce mécanisme de « double contrainte » renforce la capacité du DSL à créer des liens solides et explicables tout en restant ouvert à de grandes variations dans les données, combinant ainsi **robustesse** et **transparence**.

3.5.3.2. Complexité et Gestion des Contradictions Symboliques vs. Incertitude Sub-Symbolique

Dans un **Deep Synergy Learning (DSL)** hybride, l'intégration simultanée des informations sub-symboliques – représentées par des **embeddings neuronaux** – et des informations symboliques – issues de **règles** ou d'**ontologies** – permet de combiner la **robustesse statistique** à la capacité d'**inférence formelle**. Toutefois, cette fusion engendre également des défis majeurs, tant sur le plan **algorithmique** que conceptuel, du fait de la coexistence d'une **logique stricte** et d'une **incertitude progressive** inhérente aux représentations apprises. Nous proposons ici une analyse en deux axes complémentaires où, d'une part, la **complexité algorithmique** est induite par la comparaison conjointe des composantes sub-symboliques et symboliques, et d'autre part, la gestion des tensions entre les **contradictions symboliques** et l'**incertitude sub-symbolique** constitue un défi central.

A. Complexité Algorithmique Accrue

Dans un système purement sub-symbolique, le calcul de la similarité entre deux entités \mathcal{E}_i et \mathcal{E}_j repose sur des opérations vectorielles ou sur des fonctions noyau, par exemple

$$\mathbf{S}_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\gamma > 0$. La complexité de telles opérations est généralement de l'ordre de $O(d)$ par comparaison, ou $O(n^2 \times d)$ pour l'évaluation de toutes les paires dans un ensemble de n entités. Lorsque l'on ajoute la composante symbolique, qui implique la comparaison de deux blocs de règles R_i et R_j , la charge de calcul s'accroît significativement. En effet, le **matching** de règles ou d'axiomes, et les tests de cohérence – qui peuvent nécessiter de parcourir un espace exponentiel en fonction de la complexité du langage logique (par exemple dans des formalismes tels que OWL DL) – viennent s'ajouter au coût de calcul sub-symbolique. Ainsi, la mise à jour des pondérations dans le DSL, donnée par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{hybrid}}(i, j) - \tau \omega_{i,j}(t)],$$

devient une opération dont la complexité globale est la somme de l'évaluation sub-symbolique (en $O(n^2 \times d)$) et d'une composante symbolique potentiellement NP-complète.

Pour pallier cette surcharge, plusieurs **stratégies** heuristiques peuvent être déployées. Par exemple, l'utilisation d'algorithmes de type **k-NN** ou **d'approximate nearest neighbor** permet de limiter les comparaisons symboliques à un voisinage restreint des entités les plus proches sur le plan sub-symbolique. De même, l'imposition d'un **seuil de similarité** permet d'effectuer des comparaisons logiques uniquement lorsque la proximité vectorielle atteint un certain niveau. Enfin, des techniques d'**indexation symbolique** peuvent organiser les axiomes ou concepts de manière hiérarchique, afin de réduire la nécessité de tests complets de contradiction pour chaque paire.

B. Contradictions Symboliques vs. Incertitude Sub-Symbolique

L'intégration de deux modalités hétérogènes dans le DSL peut générer des **signaux contradictoires** entre la partie symbolique et la partie sub-symbolique. D'un côté, la logique formelle, caractérisée par une **précision stricte**, peut conduire à rejeter la compatibilité entre deux entités si leurs blocs de règles R_i et R_j présentent des **contradictions** (par exemple, lorsque des axiomes incompatibles sont détectés dans $R_i \cup R_j$). Dans un système purement symbolique, une telle contradiction aurait pour effet d'annuler toute similarité entre les entités.

D'un autre côté, la composante sub-symbolique, fondée sur des embeddings calculés à partir de données bruyantes et complexes, introduit une **incertitude** inhérente. Il est fréquent que deux entités présentent des **embeddings** très similaires, malgré une ambiguïté dans l'interprétation logique de leurs caractéristiques. Ainsi, il peut exister un désaccord où la partie logique indique une incompatibilité tandis que la partie neuronale suggère une forte similarité.

Pour gérer cette ambiguïté, plusieurs mécanismes d'**ajustement** sont envisageables. Dans l'agrégation hybride

$$\mathbf{S}_{\text{hybrid}}(i, j) = \alpha \mathbf{S}_{\text{sub}}(i, j) + (1 - \alpha) \mathbf{S}_{\text{sym}}(i, j),$$

le paramètre α peut être modulé afin de refléter la confiance relative dans chaque composante. Par ailleurs, l'implémentation d'un **feedback top-down** peut permettre d'ajuster dynamiquement les règles logiques si des incompatibilités persistantes sont détectées, ou inversement, de modifier les embeddings si la composante symbolique démontre une cohérence forte dans la formation des clusters. Une autre solution consiste à appliquer un mécanisme de **saturation** qui limite l'influence négative d'une contradiction symbolique, de manière que celle-ci réduise la synergie sans l'annuler complètement. Cette approche permet ainsi d'obtenir un compromis où une contradiction symbolique forte diminue significativement le score, mais ne paralyse pas la formation d'un lien si les embeddings fournissent un signal robuste.

C. Synthèse et Implications Pratiques

La cohabitation dans un DSL d'une **dimension sub-symbolique** et d'une **dimension symbolique** conduit à une complexité accrue où il est nécessaire de calculer et de fusionner deux types d'évaluations. D'une part, la proximité dans l'espace vectoriel, qui est déjà coûteuse pour de grands ensembles de données, et d'autre part, la vérification logique, souvent associée à des algorithmes dont la complexité peut croître de manière exponentielle. En parallèle, la gestion des éventuelles contradictions symboliques et de l'incertitude sub-symbolique impose un arbitrage délicat, qui se traduit par le choix du paramètre de pondération α et par l'adoption d'opérateurs combinatoires adaptés (tels que le produit ou le minimum).

Dans une implémentation pratique, la stabilité d'un DSL hybride repose sur trois axes principaux :

- La **gestion algorithmique** par le biais d'heuristiques (k-NN, seuils, indexation) permettant de limiter le coût de calcul des comparaisons logiques.
- La capacité à arbitrer les conflits entre une logique stricte et une incertitude statistique en adaptant dynamiquement la contribution de chaque composante.
- L'établissement d'un **métaniveau de contrôle** (feedback top-down) qui permet, en cas de divergence trop marquée, d'ajuster soit les règles, soit les embeddings afin d'assurer une convergence cohérente du système.

Cette approche, en somme, renforce l'intérêt des systèmes neuro-symboliques en montrant que, bien que l'intégration de deux modalités de représentation introduise une complexité supplémentaire et des tensions potentielles, il est possible d'élaborer des stratégies pour en atténuer les effets négatifs tout en tirant parti de la complémentarité offerte par chaque sous-système. Le DSL hybride, s'il est correctement paramétré et optimisé, parvient ainsi à allier la **robustesse statistique** d'un apprentissage profond avec la **rigueur** et la **transparence** d'un raisonnement formel, garantissant ainsi des clusters d'entités qui sont à la fois pertinents et explicables.

En définitive, la gestion de la **complexité algorithmique** et des **contradictions** symboliques dans un DSL hybride nécessite un pilotage attentif des interactions entre les composants sub-symboliques et symboliques. La mise en œuvre de techniques d'approximation et de stratégies d'arbitrage permet d'obtenir un système capable de s'adapter à des environnements complexes,

tout en maintenant une cohérence conceptuelle et une capacité d'explication indispensable dans des domaines sensibles.

3.5.3.3. Cas d'Usage : Un Agent Conversationnel Fusionnant Logique et Embedding

Dans le cadre d'un **DSL** (*Deep Synergy Learning*) hybride, l'intégration d'une composante **sub-symbolique** (embeddings neuronaux) et d'une composante **symbolique** (règles logiques, ontologies) trouve une illustration particulièrement parlante dans le domaine des **agents conversationnels**. L'idée fondamentale consiste à doter un chatbot d'une **représentation vectorielle** issue de modèles transformeurs (par exemple, BERT ou GPT) et d'un **bloc logique** structuré, capable d'assurer la cohérence et l'explicabilité du dialogue. Cette architecture permet ainsi de répondre aux requêtes en conciliant la capacité d'apprentissage à grande échelle et la transparence des raisonnements formels.

A. Architecture Hybride : Sub-Symbolique et Symbolique

Dans une configuration hybride, chaque message utilisateur, noté $\mathcal{E}_{\text{user}}$, est modélisé de manière double. La composante sub-symbolique se traduit par un **vecteur** $\mathbf{x}_{\text{user}} \in \mathbb{R}^d$, obtenu via un réseau neuronal entraîné sur un large corpus de dialogues. Parallèlement, la composante symbolique est représentée par un **bloc** de règles $\{R_{\text{user}}\}$ qui capture l'intention ou le sens explicite du message, par exemple en identifiant des concepts tels que "vol", "réservation", ou "date". De manière similaire, chaque module interne ou scénario de dialogue – tel que « achat de billets », « météo » ou « small-talk » – est également caractérisé par un embedding \mathbf{x}_j et un bloc de règles $\{R_j\}$. La **synergie** entre le message utilisateur et un module spécifique est alors évaluée par la fonction hybride

$$S_{\text{hybrid}}(\mathcal{E}_{\text{user}}, \mathcal{E}_j) = \alpha S_{\text{sub}}(\mathbf{x}_{\text{user}}, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_{\text{user}}, R_j),$$

où le paramètre $\alpha \in [0,1]$ module l'importance relative de la **proximité vectorielle** et de la **compatibilité logique**.

B. Exemple de Sélection de Module

Pour illustrer cette approche, considérons un scénario où un utilisateur formule la requête suivante : « Peux-tu me réserver un vol pour Barcelone la semaine prochaine ? ». Le système d'extraction sub-symbolique, reposant sur un transformeur, génère un embedding \mathbf{x}_{user} qui sera naturellement proche dans l'espace vectoriel du vecteur typique associé au module « FlightBooking » (réservation de vols). Parallèlement, le bloc logique $\{R_{\text{user}}\}$ extrait des indices explicites comme "vol", "destination" et "date", tandis que le module « FlightBooking » dispose d'un embedding \mathbf{x}_j spécifique et d'un ensemble de règles R_j définissant, par exemple, que la présence simultanée d'une destination et d'une date déclenche un scénario de réservation. La synergie sub-symbolique, calculée par une mesure telle que le **cosinus de similarité** ou un noyau gaussien, sera élevée si la sémantique de la requête correspond bien à celle du module, tandis que la synergie symbolique validera la cohérence entre les indices extraits. La combinaison des deux permet ainsi au DSL de sélectionner le module « FlightBooking » et de justifier cette décision par l'argumentation suivante :

« La similarité sub-symbolique indique que vous parlez de billets et de dates, et la logique détecte la présence de la destination ‘Barcelone’ ainsi que d’une date ‘la semaine prochaine’, ce qui correspond aux règles du scénario de réservation. »

C. Avantage en Explicabilité et Apprentissage Continu

L’intégration de ces deux composantes confère à l’agent conversationnel une **explicabilité** accrue. En effet, la partie sub-symbolique, tirée d’un modèle neuronal, est capable de capter des variations subtiles et de s’adapter continuellement grâce à des techniques de fine-tuning sur de nouveaux corpus. En parallèle, le **bloc logique** offre une justification transparente en exposant les règles partagées ou les axiomes utilisés pour interpréter la requête, ce qui est particulièrement précieux dans des domaines sensibles. Lorsqu’un nouveau scénario de dialogue est introduit (par exemple, la gestion d’abonnements), il est possible d’ajouter un nouvel embedding \mathbf{x}_{new} et un bloc de règles R_{new} associé. Cette mise à jour se répercute sur la fonction de synergie hybride, permettant au système de s’adapter en temps réel tout en maintenant une architecture explicable. De plus, si une contradiction symbolique apparaît ou si la partie sub-symbolique présente une confusion, le système peut activer des mécanismes de rétroaction (feedback) pour réviser les règles ou affiner les embeddings, garantissant ainsi une **adaptation continue** sans perte de lisibilité.

D. Conclusion

L’exemple d’un agent conversationnel illustrant l’approche hybride d’un DSL montre comment la fusion de la **partie neuronale** (embeddings) et de la **partie logique** (règles et ontologies) permet de concevoir un chatbot capable non seulement de détecter l’intention de l’utilisateur via la **proximité vectorielle**, mais également de justifier sa sélection de scénario par une **cohérence explicite**. Ce mécanisme de fusion, incarné par la fonction de synergie

$$S_{\text{hybrid}}(\mathcal{E}_{\text{user}}, \mathcal{E}_j) = \alpha S_{\text{sub}}(\mathbf{x}_{\text{user}}, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_{\text{user}}, R_j),$$

confère à l’agent une capacité d'**apprentissage continu** et d'**explicabilité**. Ainsi, le DSL hybride, en orchestrant ces deux dimensions, aboutit à un système conversationnel plus **intelligent** et plus **transparent**, capable de traiter des requêtes hétérogènes et de fournir des justifications explicites quant aux décisions prises, tout en évoluant pour s’adapter aux nouveaux flux de données et scénarios.

3.6. Aspects Avancés et Études de Cas

Les sections précédentes ont détaillé les **fondements** de la représentation des entités (chap. 3.2, 3.3, 3.4, 3.5), qu'elles soient sub-symboliques, symboliques ou hybrides. Dans cette partie (3.6), nous abordons des **perspectives** plus poussées et des **exemples concrets** illustrant l'étendue du **DSL** (Deep Synergy Learning) lorsqu'il s'attaque à des configurations complexes :

- Les **hypergraphes** et la **synergie n-aire** (3.6.1) : on dépasse le lien binaire pour relier un **ensemble** d'entités, ouvrant la voie à des interactions collectives (image–son–texte, par exemple).
- Les **structures fractales** (3.6.2) : la représentation elle-même peut devenir **multi-échelle** ou **self-similar**, en lien avec la fractalité (chap. 6) ; une manière d'organiser des données en hiérarchie d'emboîtements.
- Des **études de cas** (3.6.3) : applications pratiques (analyse audio-visuelle, agent conversationnel, robotique sensorielle) démontrant comment combiner embeddings et logiques.
- Une **comparaison expérimentale** (3.6.4) : mesurer l'influence du type de représentation (sub, sym, hybride) sur la qualité de la synergie, la vitesse de convergence et la robustesse.

3.6.1. Hypergraphes et Synergie n-aire

Jusqu'ici, la plupart des exemples de **synergie** ont porté sur des **paires** (i, j) . Mais un **DSL** peut aller plus loin en considérant des **interactions collectives** entre $\{i_1, i_2, \dots, i_k\}$. Au lieu de ne mesurer que $S(i, j)$, on définit une **hyper-synergie** $S(\{i_1, \dots, i_k\})$ qui évalue la compatibilité (sub-symbolique, symbolique, ou mixte) d'un groupe d'entités dans son ensemble.

3.6.1.1. Au-delà du Lien Binaire, Possibilité de Relier $\{i_1, i_2, \dots, i_k\}$ par une “Hyper-Synergie”

Dans la plupart des *Synergistic Connection Networks* (SCN), l'architecture se fonde sur des liaisons binaires $\omega_{i,j}$ entre deux entités \mathcal{E}_i et \mathcal{E}_j . Cette structure est suffisante lorsque l'on se limite à des interactions par paires, mais on se heurte à des limites dès lors que certains phénomènes ou combinaisons essentielles ne prennent leur sens qu'en considérant plusieurs entités simultanément. C'est le cas, par exemple, dans des données multimodales (texte + audio + image), où la cohérence globale ne s'exprime pas par la somme ou la moyenne de trois liaisons binaires, mais réellement par la compatibilité collective des trois modalités.

Pour saisir de telles interactions d'ordre supérieur, on peut s'éloigner du simple concept d'arêtes binaires et se tourner vers le formalisme des *hypergraphes*, où une *hyper-arête* relie un sous-ensemble de noeuds $\{i_1, \dots, i_k\}$. Le SCN se dote alors d'**hyper-liens** dotés d'un *score* ou d'un “hyper-synergie” :

$$S(\{i_1, \dots, i_k\}) \in \mathbb{R}^+.$$

Ce score rend compte d'une cohérence globale non réductible aux simples paires. Il peut s'agir, par exemple, d'une fonction

$$S_k(\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k}),$$

si l'on se situe dans un registre sub-symbolique (des embeddings multiples), ou encore d'une compatibilité logique n -aire, si l'on manipule des blocs de règles. Dans un SCN où l'on mettrait à jour les pondérations, on n'aurait plus seulement $\omega_{i,j}$ à itérer, mais potentiellement un poids $\omega_{(i_1, \dots, i_k)}$ pour chaque hyper-arête, reflétant la force de l'interaction d'ordre k .

Un premier exemple surgit dans la *fusion multimodale* : un *triplet* {texte, image, audio} n'a de sens que collectivement (une légende d'image, une bande-son, un contexte verbal), et la validité ou la cohérence de l'objet multimédia se repère à l'échelle du triplet. Un SCN classique reliant texte–image d'un côté, image–audio de l'autre, etc., ne traduirait pas la véritable interdépendance à trois entités.

Un autre exemple, dans le domaine de la **maintenance prédictive**, concerne des **capteurs multiples** où la détection d'une alarme requiert que trois capteurs, comme la température, la pression et la vibration, dépassent certains seuils simultanément. Une hyper-synergie $\omega_{(i_1, i_2, i_3)}$ se voit alors renforcée dès que l'on observe ce triple alignement.

Par ailleurs, certaines *communautés* ou *clusters* n'émergent que via des liens d'ordre supérieur. Dans la biologie, on peut étudier un système où l'interaction n'est pas captée par la somme de couples, mais par la triple ou quadruple co-occurrence de gènes ou de protéines.

Sur le plan algorithmique, étudier toutes les combinaisons de taille k dans un ensemble de n entités revient à considérer $\binom{n}{k}$ sous-ensembles, ce qui en $O(n^k)$ peut devenir prohibitif dès que $k > 2$ ou que n est grand. Les mises à jour itératives du SCN, ou l'évaluation de la synergie sur toutes ces hyper-arêtes, risquent de se révéler inenvisageables à grande échelle.

Pour conserver la **puissance** de l'hyper-synergie tout en évitant une explosion combinatoire, on adopte des heuristiques ou des restrictions où **limiter la taille de l'hyper-synergie** à 3 ou 4 entités maximum permet d'éviter une complexité inutile si l'utilité n'augmente plus au-delà. **Filtrer** via un score pair à pair avant de constituer des triplets ou quadruplets garantit que l'on ne forme un hyper-lien $\{i, j, k\}$ que si $\omega_{i,j}, \omega_{j,k}, \omega_{i,k}$ dépassent un certain seuil. Enfin, appliquer des techniques de *sampling* ou de *randomisation* permet d'échantillonner parmi les combinaisons multiples afin d'éviter de tester chaque ensemble $\{i_1, \dots, i_k\}$.

Si l'on introduit un poids $\omega_{(i_1, \dots, i_k)}$, la mise à jour prend une forme :

$$\omega_{(i_1, \dots, i_k)}(t+1) = \omega_{(i_1, \dots, i_k)}(t) + \eta [S(\{i_1, \dots, i_k\}) - \tau \omega_{(i_1, \dots, i_k)}(t)].$$

Chacune de ces hyper-arêtes se renforce ou s'affaiblit selon la “cohérence globale” du sous-groupe $\{i_1, \dots, i_k\}$. Les entités concernées peuvent se retrouver **fortement** reliées en tant que *tuplet*, alors même que la somme des liaisons binaires prises deux à deux ne le suggérait pas forcément. On obtient donc une *structure hypergraphique* potentiellement plus riche et plus adaptée à certaines *relations collectives*, au détriment d'une gestion plus complexe (tant dans la mise en œuvre que dans la lecture du résultat).

3.6.1.2. Représentation Plus Complexe : Capturer un “Ensemble” d’Entités Conjointes (par exemple, un Triplet Image–Texte–Audio)

Lorsqu’un *Synergistic Connection Network* (SCN) doit traiter une interaction qui ne se limite plus à de simples paires (i, j) , il peut être nécessaire de considérer un *ensemble* plus vaste, comme $\{i_1, i_2, \dots, i_k\}$. Cette exigence apparaît notamment dans des scénarios multimodaux, où plusieurs entités doivent être observées conjointement (par exemple, un triplet image–texte–audio). Dans le cas d’un triplet, l’auto-organisation ne peut pas se contenter de lier séparément image–texte, texte–audio et audio–image ; une **synergie** réellement *n*-aire est requise pour évaluer la *cohérence globale* de l’ensemble.

Considérons la situation d’un article ou d’un post combinant une **image**, un **texte** descriptif et un **segment audio**. Si l’on se borne à analyser les paires (image–texte, texte–audio, audio–image), on peut rater des aspects qui n’apparaissent qu’en considérant simultanément les trois modalités. Une contradiction, par exemple, peut se glisser dans l’interaction globale (image et audio se rapprochent, texte est proche de l’audio, mais la combinaison image + texte ne correspond pas). Une vision strictement binaire ne détectera pas forcément cette incohérence, tandis qu’une **hyper-synergie** {image, texte, audio} peut en faire ressortir la contradiction ou, au contraire, la cohérence collective.

Dans une représentation hypergraphique, on remplace les arêtes binaires par des *hyperarêtes* connectant plusieurs noeuds $\{i_1, i_2, \dots, i_k\}$, chacune dotée d’un *score* ou *poids* $\omega_{\{i_1, \dots, i_k\}}$. On peut alors définir une **hyper-synergie**

$$S(\{i_1, i_2, \dots, i_k\}) \in \mathbb{R}^+,$$

qui exprime la cohérence *n*-aire. Dans le cas d’un triplet {img, txt, aud}, on peut, par exemple, construire un vecteur de *fusion* $\mathbf{z}_{\text{joint}} = \text{Fusion}(\mathbf{x}_{\text{img}}, \mathbf{x}_{\text{txt}}, \mathbf{x}_{\text{aud}})$ pour la composante sub-symbolique. Et si chacune des entités possède des **règles** ou attributs symboliques $\{R_{\text{img}}, R_{\text{txt}}, R_{\text{aud}}\}$, on agrège ces blocs en un ensemble *conjoint* $R_{\text{joint}} = \bigcup_{\ell=1}^3 R_{i_\ell}$. Il est alors possible d’évaluer :

$$\begin{aligned} & S(\{\text{img, txt, aud}\}) \\ &= \alpha S_{\text{sub}}\left(\text{Fusion}(\mathbf{x}_{\text{img}}, \mathbf{x}_{\text{txt}}, \mathbf{x}_{\text{aud}})\right) + (1 - \alpha) S_{\text{sym}}\left(\text{Union}(R_{\text{img}}, R_{\text{txt}}, R_{\text{aud}})\right). \end{aligned}$$

Si cette *hyper-synergie* est élevée, le SCN conclut à un ensemble multimodal cohérent ; dans le cas contraire, il identifie une *incohérence*.

Le passage à des hyper-synergies *n*-aires confronte le SCN à une explosion combinatoire où, pour des ensembles de *n* entités, tester toutes les combinaisons de taille *k* se chiffre en $\binom{n}{k}$. Même pour un triplet, cela peut représenter $O(n^3)$ d’hyperarêtes. Des heuristiques de **filtrage** s’avèrent alors essentielles où l’on ne retient que les ensembles $\{i_1, \dots, i_k\}$ pour lesquels les paires (i_p, i_q) dépassent un certain seuil de synergie binaire, ou bien l’on restreint la taille *k* à 3 ou 4 au maximum afin de contenir la complexité.

Lorsque l'on dispose d'une pondération $\omega_{\{i_1, \dots, i_k\}}$, la règle d'itération dans un SCN hybride devient :

$$\omega_{\{i_1, \dots, i_k\}}(t+1) = \omega_{\{i_1, \dots, i_k\}}(t) + \eta [S(\{i_1, \dots, i_k\}) - \tau \omega_{\{i_1, \dots, i_k\}}(t)].$$

Cette équation se passe désormais au niveau de l'hyperarête (ou "nœud conjugué"). Dans des applications multimodales, si les trois éléments image, texte et audio forment vraiment un ensemble cohérent, le *poids* $\omega_{\text{img,txt,aud}}$ s'accroît, reflétant l'identification d'un "micro-cluster" n -aire.

3.6.1.3. Implications : Algorithmes Plus Lourds, mais Patterns Plus Riches

La transition d'une synergie strictement binaire ($S(i, j)$) à une **hyper-synergie** ($S(\{i_1, \dots, i_k\})$) engendre un bouleversement considérable dans l'architecture et la dynamique d'un *Synergistic Connection Network* (SCN). Si les liens binaires suffisent à décrire la plupart des interactions entre deux entités, il existe des configurations où la cohérence d'un ensemble complet $\{i_1, \dots, i_k\}$ n'est pas réductible à la somme de paires, notamment dans des contextes multimodaux (image–texte–audio) ou lorsqu'un événement résulte d'un groupe de capteurs se déclenchant simultanément. Cet enrichissement se paie toutefois par une **charge algorithmique** et une gestion plus complexes, qu'il est essentiel d'analyser pour un DSL (Deep Synergy Learning) de grande dimension.

Le principal défi provient de la **combinatoire**. Dans un réseau de n entités, considérer tous les sous-ensembles de taille k impose d'en examiner $\binom{n}{k}$, ce qui peut se chiffrer en $O(n^k)$. Le simple fait de passer de $k = 2$ (paires) à $k = 3$ (triplets) fait déjà croître le nombre d'entités à évaluer de façon significative, et la situation s'aggrave pour des ordres supérieurs. Cela rend l'**exploration exhaustive** pratiquement impossible pour de grands n . Sur le plan du DSL, qui met à jour de façon itérative un score (pondération) lié aux synergies, on ne peut plus se contenter de vérifier $\omega_{i,j}$ pour toutes paires (i, j) . Il faut dorénavant gérer $\omega_{\{i_1, \dots, i_k\}}$ pour les hyper-arêtes, et la recalculer chaque fois qu'une entité \mathcal{E}_{i_ℓ} modifie ses attributs (embedding, règles). L'effort de réévaluation peut vite s'avérer prohibitif. Des approches **heuristiques** émergent alors, consistant à filtrer ou restreindre le champ d'examen aux groupes considérés comme prometteurs.

Un hyper-lien $\omega_{(i_1, \dots, i_k)}$ s'actualise selon un schéma dérivé de la dynamique DSL habituelle :

$$\omega_{(i_1, \dots, i_k)}(t+1) = \omega_{(i_1, \dots, i_k)}(t) + \eta [S(\{i_1, \dots, i_k\}) - \tau \omega_{(i_1, \dots, i_k)}(t)].$$

La fonction de synergie $S(\{i_1, \dots, i_k\})$ peut être défini dans un cadre sub-symbolique (fusion des embeddings $\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k}$ et évaluation par un réseau) ou sémantique (compatibilité globale des blocs logiques R_{i_1}, \dots, R_{i_k}), ou encore un mélange des deux. Sitôt qu'une seule entité \mathcal{E}_{i_ℓ} change son embedding ou ses axiomes, l'hyper-synergie se retrouve possiblement obsolète, imposant un recalculation si l'on ne met pas en place un dispositif pour restreindre son impact.

Ce surcroît de complexité, s'il représente une entrave opérationnelle, fait aussi la force d'une synergie n -aire. Il est désormais possible de détecter des *patterns* ou des phénomènes collectifs absents du spectre binaire. Dans un triplet {image, texte, audio}, la cohérence globale n'est plus strictement la somme ou la moyenne des relations deux à deux, mais véritablement un effet conjoint. De même, un ensemble de capteurs {capteur₁, capteur₂, ...} peut signaler un événement

critique seulement si tous se déclenchent en même temps. La synergie n-aire permet alors une modélisation plus fidèle et plus “haut niveau” de ces scénarios.

Au plan de l’implémentation, on recourt souvent à des **stratégies de filtrage** visant à n’explorer que les sous-ensembles d’entités ayant déjà prouvé une certaine proximité binaire (chacun des couples (i_ℓ, i_m) doit dépasser un seuil de similarité). Cette méthode de “candidats par paires” réduit la quantité d’hyperarêtes à évaluer en détail. Il est également envisageable d’employer des procédures d’**exploration incrémentale** (où l’on agrège peu à peu des entités sur la base d’un gain de synergie) ou des techniques de **recuit simulé** pour éviter l’énumération systématique de $\binom{n}{k}$.

La possibilité de mettre à jour un poids $\omega_{(i_1, \dots, i_k)}$ a, dans un **Synergistic Connection Network**, d’importantes répercussions où certains **hyper-lots** d’entités, correspondant à des groupements n -aires, vont se renforcer et former des “hyper-clusters”. On peut par exemple voir émerger un “triple cluster” {img₁, txt₅, aud₃} fermement relié par $\omega_{1,5,3}$ si ces trois modalités s’accordent fortement. Cette structure n’est pas réductible à un graphe usuel, puisqu’il faut appréhender la cohésion collective. Un aspect intéressant survient en logique symbolique, où la vérification d’une cohérence globale $\{R_{i_1}, \dots, R_{i_k}\}$ peut nécessiter des tests complexes (cohérence n-aire, potentiellement NP-complets selon la expressivité logique). Tout l’enjeu consiste à contrôler la croissance de la complexité via un design parcimonieux (filtres, heuristiques, indices) et à tirer avantage de l’expressivité supplémentaire qu’offre une synergie n-aire.

3.6.2. Structures Fractales de Représentation

Au-delà des hypergraphes et de la synergie n-aire (section 3.6.1), une autre voie **avancée** pour organiser la représentation dans un **DSL** (Deep Synergy Learning) consiste à exploiter des **structures fractales** ou auto-similaires. L’idée est d’employer un **multi-niveau** ou un **multi-échelle** (en lien avec le chapitre 6) où la représentation de chaque entité (ou de chaque cluster) peut se répliquer ou s’itérer sous une forme **fractalement** organisée. Dans certains types de données, on observe en effet des **patterns** se répétant à différentes échelles — qu’il s’agisse d’images (motifs fractals), de taxonomies hiérarchiques (ontologies en forme d’arbre auto-similaire), voire d’embeddings qui se réorganisent récursivement.

3.6.2.1. Lien avec le chap. 6 (multi-échelle + fractalité) : la représentation elle-même peut s’avérer fractale (ex. embeddings récursifs, hiérarchie de concepts)

La perspective multi-échelle exposée au chapitre 6 souligne que, dans un **DSL** (*Deep Synergy Learning*), un *Synergistic Connection Network* peut se structurer en plusieurs *niveaux* ou *couches* successifs. Chaque niveau agrège ou récapitule les motifs détectés au niveau inférieur, créant ainsi un effet d’**auto-organisation hiérarchique**. Cet agencement à étages peut aller plus loin lorsque la représentation des entités exhibe un caractère **auto-similaire**, voire **fractal**, où à chaque nouvelle échelle, on retrouve un schéma semblable reproduisant le même mode d’organisation à des granularités différentes. Il en résulte la possibilité d’embeddings **récursifs** ou de hiérarchies conceptuelles répliquant la même forme à divers niveaux.

Sur le plan **sub-symbolique**, on peut concevoir que l'embedding $\mathbf{x}_i \in \mathbb{R}^d$ représentant une entité \mathcal{E}_i se scinde en *mini-embeddings* itératifs, ce qui reflète une *structure fractale*. Dans certains domaines, comme la modélisation de textures ou de formes naturelles (ex. arbres, littoraux), les mêmes motifs géométriques se répètent à diverses résolutions. On formalise alors des schémas d'autoencodeurs profonds ou de CNN « en cascade », où l'opération $\mathbf{x}_i^{(\ell+1)} = F(\mathbf{x}_i^{(\ell)})$ reproduit un *pattern* identique à chaque étape ℓ . Cette auto-similarité sous-tend une représentation fractale où le **DSL** peut l'exploiter pour repérer ou stabiliser des clusters multi-échelle, une sous-structure locale se retrouvant dans une macro-structure globale.

Sur le plan **symbolique**, les ontologies ou ensembles de règles peuvent, elles aussi, refléter une organisation fractale. Dans une ontologie de grande taille, il arrive qu'une arborescence de concepts se **dupliche** partiellement dans diverses branches où le même **motif** notionnel réapparaît, générant des **sous-ontologies** répétant la même logique de liens comme `subClassOf` ou `partOf`. Un **DSL** hiérarchique peut alors reconnaître, au niveau micro, un groupement conceptuel, puis observer le *même* schéma à une échelle plus macro, renforçant la cohérence globale et la capacité de généralisation.

La dimension **multi-échelle** intervient lorsque l'on empile plusieurs *couches* dans le réseau, chacune gérant une échelle de représentation. On obtient un flux *ascendant* (bottom-up), où les motifs découverts localement se consolident, et un flux *descendant* (top-down), où les structures macro peuvent imposer une cohérence aux entités de rang inférieur. Dans un *cadre fractal*, cette interaction est d'autant plus fluide que la même forme **p** ou la même logique **r** se reproduit à chaque palier, ce qui évite de réinventer un nouveau schéma à chaque fois.

Mathématiquement, on peut décrire la fractalité dans le sous-espace \mathbb{R}^d par une opération F appliquée de façon itérative à un point ou à une région, créant un ensemble invariant par F . Dans un **DSL**, si l'on modélise l'évolution d'un embedding $\mathbf{x}_i^{(\ell)}$ à mesure qu'on change de niveau ℓ , on peut écrire $\mathbf{x}_i^{(\ell+1)} = F(\mathbf{x}_i^{(\ell)})$ pour une transformation donnée. Lorsque F est contractante, on obtient un *attracteur fractal*, dont la structure se reflète dans l'organisation multi-niveau des entités.

Le chapitre 6, centré sur l'apprentissage multi-échelle, se prête donc à accueillir ce genre de **représentation fractale** où, à chaque palier, l'**auto-organisation** exploite la même règle de mise à jour et la même forme d'ancrage, qu'elle soit sub-symbolique ou symbolique, mais appliquée à un degré de finesse différent. Ce principe peut renforcer l'efficacité du **DSL** dans des domaines où la notion de *self-similarité* est prépondérante (biologie, géologie, architecture conceptuelle, etc.).

3.6.2.2. Exemple : embeddings fractals pour des données “self-similar”, ou ontologies fractales pour la taxonomie

La **fractalité** joue un rôle fondamental dès que l'on observe des motifs identiques ou très similaires qui se reproduisent à différentes **échelles**. Dans le cadre d'un **Deep Synergy Learning (DSL)**, où l'on cherche à organiser des données (vectorielles ou symboliques) de façon auto-similaire, l'exploitation de cette propriété fractale permet une **intégration** plus économique et plus robuste. Cette section illustre deux cas d'application où l'un concerne la construction d'**embeddings fractals**

pour des données intrinsèquement **self-similar** comme les images, signaux ou textures, tandis que l'autre traite des **taxonomies** ou **ontologies** présentant une structure auto-similaire récurrente.

Afin de souligner l'importance de ces approches, il convient de rappeler que la **dimension fractale** d'un objet F peut être définie, par exemple, via la formule usuelle de **self-similarité** où, si la figure est couverte par N copies d'elle-même à l'échelle $1/r$, alors

$$\dim_{\text{fractale}}(F) = \frac{\ln(N)}{\ln(r)}.$$

La réitération d'une même structure à différentes résolutions se révèle un facteur clé tant dans le cas **sub-symbolique** (données géométriques ou multi-échelle) que dans le cas **symbolique** (ontologies arborescentes). Cette auto-similarité offre un levier pour construire des **représentations** et des **mesures de similarité** adaptées au principe de fractalité.

A. Embeddings fractals pour des données self-similar

Dans de nombreux champs scientifiques, les données présentent un **comportement auto-similaire** où un motif se reproduit à plusieurs échelles, révélant une structure fractale sous-jacente. On rencontre cette propriété dans des *textures naturelles* (extraction de *patches* à différents niveaux de zoom), dans des *signaux biologiques* pouvant dévoiler la même forme d'onde à des fréquences multiples, ou dans certains *objets géométriques* (côtes maritimes, contours fractals). L'idée est alors de coder chaque entité \mathcal{E}_i à l'aide d'un **embedding fractal**, c'est-à-dire d'un vecteur qui capture la redondance d'échelle.

Si l'on note $\mathbf{x}_i^{(l)}$ la description de \mathcal{E}_i à l'échelle l (par exemple, la sortie d'un *patch-encoder* appliqué à un *patch* de résolution l), on peut concaténer ou fusionner ces descriptions sur plusieurs niveaux, produisant au final un **embedding** $\mathbf{x}_i \in \mathbb{R}^D$ qui agrège cette information multi-résolution. La **fractalité** de l'entité, exprimée par la répétition statistique du même motif à plusieurs granularités, se retrouve alors dans la structure même de \mathbf{x}_i .

Une formalisation plus mathématique introduit une fonction de type

$$\mathbf{x}_i = [f(\mathbf{x}_i^{(0)}), f(\mathbf{x}_i^{(1)}), \dots, f(\mathbf{x}_i^{(L)})],$$

où f est un encodeur non linéaire (par exemple un autoencodeur profond) appliqué à différentes résolutions. La **synergie** entre deux entités \mathcal{E}_i et \mathcal{E}_j peut alors se définir via une **similarité** (cosinus, noyau gaussien, etc.) entre leurs embeddings fractals respectifs :

$$S(\mathcal{E}_i, \mathcal{E}_j) = \exp(-\alpha \parallel \mathbf{x}_i - \mathbf{x}_j \parallel),$$

assurant que deux **configurations fractales** très proches (i.e. répétant le même motif à chaque niveau) se verront attribuer une **synergie** élevée. Dans un SCN, ces entités fortes en **self-similarité** locale formeront un **cluster** auto-organisé révélant leur communauté de forme.

B. Ontologies fractales pour la taxonomie

Si l'on se situe dans un registre plus symbolique ou conceptuel, la **fractalité** peut apparaître sous forme d'une *hiérarchie* ou *arborescence* qui se répète, quasi à l'identique, dans chaque sous-

branche. De nombreuses *taxonomies* en biologie, chimie, ou dans des classifications industrielles, reproduisent la même structure de relations et de propriétés à différents étages. On parle alors de **fractalité sémantique**, puisque la même architecture se manifeste à plusieurs niveaux de la hiérarchie, de sorte que l'on peut utiliser un *motif ontologique* unique pour chacune des sous-branches.

On peut modéliser une telle ontologie comme un graphe \mathcal{G} muni d'une application récursive, par exemple

$$\phi: \mathcal{G} \rightarrow \mathcal{G} \times \{0, \dots, k-1\}$$

qui identifie des *sous-graphes* isomorphes répliquant un même ensemble de classes ou de relations. Cette propriété est analogue à la fonction d'itération d'un fractal géométrique, mais appliquée aux liens sémantiques (*subClassOf*, *hasPart*, *connectedTo*, etc.).

Dans le cadre d'un **DSL**, une telle **auto-similarité** conceptuelle renforce la **cohérence multi-niveau**. La **mise à jour** des poids de synergie, notée $\omega_{ij}(t+1) = \dots$, ou l'intégration d'une règle symbolique au niveau supérieur, peuvent automatiquement se propager aux étages fractals sous-jacents, réduisant l'effort de maintenance. Par ailleurs, la similarité sémantique entre deux *sous-domaines* peut se calculer en comparant leurs **motifs ontologiques**. Deux branches véhiculant des structures quasi identiques, à quelques variations de propriétés près, obtiendront une **distance** sémantique faible et, en conséquence, une **synergie** élevée dans le **SCN** symbolique.

C. Gains potentiels

L'approche **fractal-based** procure un **gain** en compression où un même schéma ou un même encodeur multi-résolution peut être réutilisé à diverses échelles. Elle amplifie aussi la **robustesse** où lorsqu'une révision touche la structure de base, elle s'applique à toutes les répliques, évitant ainsi les **incohérences locales**. Sur le plan purement mathématique, l'uniformisation de la **dimension fractale** ou la réplication d'un **pattern** identique signifient que la **synergie** ne fluctue pas brutalement d'un étage à l'autre, mais demeure gouvernée par un **principe** d'auto-similarité. Les mises à jour locales des poids ω_{ij} (ou l'apparition de nouvelles entités analogues) s'intègrent plus harmonieusement dans l'ensemble.

On peut inclure une *analyse dimensionnelle* pour vérifier que l'object fractal formé par la hiérarchie ou l'ensemble des embeddings est susceptible de présenter un degré de redondance réduisant la *complexité effective*. Ainsi, dans une ontologie auto-similaire, la **profondeur** L de la structure peut conditionner la **dimension** du schéma de liens, souvent inférieure à ce qu'on obtiendrait si chaque étage était dessiné de manière indépendante. Cette propriété apparaît clairement dans les constructions fractales pures, où la **taille** du système croît moins vite que l'**échelle** (puisque l'on recycle le même motif à chaque niveau).

3.6.2.3. Gains potentiels en compression ou en cohérence multi-niveaux

L'exploitation d'une **structure fractale** dans le cadre d'un **Deep Synergy Learning (DSL)** permet de renforcer à la fois la **compression** des descriptions et la **cohérence** entre différents niveaux

d'analyse. Cette intégration repose sur l'idée que lorsqu'un *motif* se répète à diverses échelles, on peut le stocker ou le calculer une seule fois et le réutiliser de façon récurrente.

A. Compression et réduction de la redondance

Lorsqu'une représentation fractale ou une ontologie auto-similaire est mise en place, la **redondance** entre les différentes entités se matérialise par la récurrence du même schéma à plusieurs échelles. Supposons qu'un motif **m** se retrouve répliqué dans plusieurs branches d'un arbre ou à différents niveaux de résolution. Au lieu de décrire ce motif **m** chaque fois qu'il apparaît, on peut le factoriser dans une structure commune. Si le fractal possède un facteur d'échelle $r > 1$ et qu'on retrouve le même motif dans N zones à chaque itération, la quantité totale d'information mémorisée peut se réduire à un *coût de base* plus faible que la somme naïve des contributions de chaque zone.

Une modélisation plus formelle peut invoquer une fonction de self-similarité. Si la structure globale \mathcal{F} se décrit par un ensemble de copies de dimension α , on a $\dim_{\text{fractale}}(\mathcal{F}) = \alpha$, alors que la description complète (sans prise en compte de l'auto-similarité) conduirait à une dimension apparente plus grande. Dans un **SCN** (Synergistic Connection Network), un grand nombre de nœuds peuvent partager la même sous-structure interne ; la référence à cette sous-structure suffit alors, ce qui donne un **gain** en $O(\log n)$ ou $O(n^\delta)$ (selon la fractalité) plutôt qu'en $O(n)$. D'un point de vue algorithmique, le recalcul des **synergies** $S(i, j)$ s'en trouve lui aussi allégé, car chaque motif commun **m** ne s'encode qu'une fois avant d'être propagé dans l'ensemble des entités concernées.

B. Cohérence multi-niveaux et synergie fractale

Les modèles fractals sont particulièrement adaptés à la logique **multi-niveau** qui sous-tend le DSL. Lorsque la même forme se répète à différentes échelles, la **cohérence** entre niveaux local et global se voit considérablement renforcée. Une entité \mathcal{E}_i peut se trouver au niveau micro avec des liens $\omega_{i,j}(t)$ qui reproduisent, en "miniature", la structure générale observable au niveau macro.

Dès que l'on considère la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

on remarque que le schéma fractal induit un renforcement cohérent à chaque échelle. Si les entités \mathcal{E}_i et \mathcal{E}_j appartiennent à un motif reproduit dans plusieurs parties du réseau, la synergie $S(i,j)$ demeure élevée dans chacun de ces contextes, ce qui stabilise $\omega_{i,j}$ de manière récursive. Les mêmes **clusters** ou agrégations se retrouvent ainsi à divers niveaux de granularité, ce qui unifie l'auto-organisation.

Sur le plan conceptuel, cette **auto-similarité** multi-niveau peut être rapprochée d'un **zoom fractal** où l'on peut se déplacer entre une échelle fine et une échelle globale tout en constatant la même topologie répliquée. Le **DSL** bénéficie alors d'une stabilité accrue, car la redondance des liaisons synergiques évite les contradictions entre un niveau et l'autre.

C. Stabilité et scalabilité dans les applications fractales

La dimension fractale sert également de garant de **stabilité** et de **scalabilité** pour le DSL. Dans des domaines comme la **géologie**, où les images présentent des textures fractales, on peut adopter un

embedding fractal afin de capturer la redondance d'échelle et réduire la dimension effective des données. Dans l'**ontologie** d'un système complexe, on peut dupliquer un même module conceptuel pour chaque subdivision, ce qui préserve la cohérence de la base de connaissances. On formalise parfois cette récurrence par une application Φ qui envoie chaque branche vers un motif identique mais étiqueté différemment,

$$\Phi: \mathcal{G} \rightarrow \bigsqcup_{\ell=1}^k \mathcal{G}_\ell,$$

là où \mathcal{G}_ℓ est une copie partiellement isomorphe de \mathcal{G} .

Ce principe accroît la **scalabilité** du réseau, car l'augmentation de la taille (nombre de nœuds ou de classes) n'exige pas de réinventer la structure de liens, mais simplement d'appliquer la transformation Φ ou d'exploiter les mêmes blocs encodés. Les **clusters** ou groupes de nœuds se forment ainsi de façon reproductible, même lorsque l'on déploie une ontologie à plusieurs milliers de branches. Le rapport entre la complexité globale et la profondeur des niveaux demeure raisonnable, en raison de la répétition du *motif fractal* à chaque sous-niveau.

L'effet bénéfique sur la **stabilité** tient au fait qu'un motif validé localement engendre les mêmes propriétés au niveau global. Les paramétrages $\omega_{i,j}$ déjà convergés à un étage se transposent aux autres étages, conférant au **DSL** une robustesse face aux modifications et aux extensions. Un schéma local ne nécessite plus d'être redéfini dans toute la hiérarchie, ce qui évite les incohérences et limite les oscillations.

3.6.3. Études de Cas Illustratives

Après avoir exploré différentes **extensions** et **approches** avancées (hypergraphes n-aires, structures fractales, etc.), il est utile de **concrétiser** ces idées à travers quelques **cas d'usage** représentatifs. Dans cette section (3.6.3), nous décrirons trois scénarios où un DSL (Deep Synergy Learning) peut trouver une application pratique et mettre en œuvre les principes de représentation (sub-symbolique, symbolique, hybride) examinés tout au long du Chapitre 3 :

- (3.6.3.1) Un **système d'analyse audio-visuelle**, combinant des **vecteurs CNN** pour l'image, des **embeddings** pour l'audio et des **règles** logiques propres à un domaine spécifique (ex. surveillance, broadcast, etc.).
- (3.6.3.2) Un **agent conversationnel**, déjà évoqué dans des sections antérieures, qui fusionne des "règles" (cohérence contextuelle) et des embeddings textuels (ex. BERT).
- (3.6.3.3) Un **scénario de robotique sensorielle**, où la représentation sub-symbolique (capteurs multiples) cohabite avec une représentation symbolique décrivant les actions, les objets et la logique de manipulation.

3.6.3.1. Système d'analyse audio-visuelle : vecteurs CNN + embeddings audio + logiques domain-specific

Dans les applications de **surveillance**, d'**indexation multimédia** ou de **diagnostic** dans un environnement audiovisuel complexe, il est souvent crucial de fusionner une **information visuelle** (généralement encodée par un **réseau de neurones convolutifs**, CNN) et une **information sonore** (par exemple encodée via des **embeddings audio**), tout en les contraignant par des **règles logiques** spécifiques au domaine. Le **Deep Synergy Learning (DSL)**, et plus particulièrement son réseau **SCN** (Synergistic Connection Network), offre un cadre systématique pour opérer cette fusion et détecter des **événements** ou **patterns** pertinents.

Il est d'usage de représenter chaque segment temporel issu de la **vidéo** par un vecteur $\mathbf{x}_{\text{vid}} \in \mathbb{R}^{d_1}$, obtenu à l'aide d'un **CNN** pré-entraîné ou d'un autre modèle **sub-symbolique**. Parallèlement, chaque segment temporel **audio** se voit attribuer un vecteur $\mathbf{x}_{\text{aud}} \in \mathbb{R}^{d_2}$, issu d'un traitement du signal tel qu'un **spectrogramme** encodé ou un modèle **transformer audio**. Une procédure de **fusion** des deux composantes, notée

$$\mathbf{x}_{\text{joint}} = F_{\text{fusion}}(\mathbf{x}_{\text{vid}}, \mathbf{x}_{\text{aud}}),$$

peut se contenter d'une **concaténation** ou employer des techniques plus avancées (attention croisée, somme pondérée). La dimension symbolique intervient à travers un ensemble de **règles** ou **d'axiomes** propres au domaine où, par exemple, “Si une zone interdite est franchie et qu'un son d'alarme est détecté, alors alerte.” On peut modéliser ces règles par des **propriétés logiques** regroupées dans un bloc \mathbf{R}_{vid} pour la partie vidéo, et \mathbf{R}_{aud} pour la partie audio, ou bien un bloc commun si des règles globales couvrent l'ensemble du signal.

La **fonction de synergie** doit alors refléter à la fois la proximité sub-symbolique (cohérence **audio-visuelle**) et la compatibilité des **règles** logiques (compatibilité sémantique). Afin d'agrégier ces deux critères, on définit

$$S_{\text{hybrid}}(\mathcal{E}_{\text{vid}}, \mathcal{E}_{\text{aud}}) = \alpha S_{\text{sub}}(\mathbf{x}_{\text{vid}}, \mathbf{x}_{\text{aud}}) + (1 - \alpha) S_{\text{sym}}(\mathbf{R}_{\text{vid}}, \mathbf{R}_{\text{aud}}).$$

La composante sub-symbolique S_{sub} peut être une **similarité** exponentielle sur la norme de la différence,

$$S_{\text{sub}}(\mathbf{x}, \mathbf{y}) = \exp(-\beta \|\mathbf{x} - \mathbf{y}\|),$$

ou tout autre indicateur mesurant à quel point la **signature visuelle** et la **signature audio** coïncident (corrélation temporelle, alignement spectral, etc.). La composante symbolique S_{sym} se fonde sur la logique du domaine où, si les axiomes audio et vidéo ne se **contredisent** pas, et mieux encore s'ils **confirment** l'un l'autre, alors $S_{\text{sym}}(\mathbf{R}_{\text{vid}}, \mathbf{R}_{\text{aud}}) \approx 1$. En revanche, une contradiction explicite (par exemple, “alarmeActivée=TRUE” côté audio mais “alarmeInopérante=TRUE” côté vidéo) fait chuter drastiquement le score symbolique.

Le **SCN** peut ensuite **auto-organiser** les segments audio-visuels au moyen d'une **règle de mise à jour** itérative :

$$\omega_{(\text{vid}, \text{aud})}(t+1) = \omega_{(\text{vid}, \text{aud})}(t) + \eta [S_{\text{hybrid}}(\mathcal{E}_{\text{vid}}, \mathcal{E}_{\text{aud}}) - \tau \omega_{(\text{vid}, \text{aud})}(t)].$$

Un lien $\omega_{(\text{vid}, \text{aud})}$ se voit **renforcé** (croît progressivement) si la synergie audio-vidéo reste élevée ; dans le cas contraire, il décroît. Après plusieurs itérations, ce mécanisme local de **plasticité** fait apparaître des **clusters** de segments fortement reliés, correspondant à des événements ou situations reconnues.

Pour décider d'une **alerte** ou d'une **étiquette** associée à un intervalle temporel, on peut fixer un **seuil** ω_{\min} où, dès que $\omega_{(\text{vid}, \text{aud})}$ dépasse ω_{\min} , le **réseau** considère la relation (vid, aud) comme un événement fusionné. La logique additionnelle (par exemple, “intrusion interdite + alarme => situation critique”) permet alors de produire un **verdict** plus précis, tant en termes de classification (type d’incident) que d'**explicabilité** (la logique indique pourquoi il y a match entre la vidéo et l’audio).

Ce **système multimodal** s’avère particulièrement robuste. Si la **vidéo** se trouve partiellement brouillée (mauvaise luminosité, occlusions), la **composante sonore** peut prendre le relai et maintenir un niveau élevé de synergie. Symétriquement, une détérioration du signal audio peut être compensée par l’analyse visuelle. Dans des applications réelles comme la **surveillance** en zones sensibles ou la **détection d’incidents** tels que les pannes, alarmes, cris ou collisions, l’association d’un bloc de **règles symboliques** vient enrichir la détection où l’on ne se contente pas d’une simple ressemblance sub-symbolique, mais on vérifie aussi la **compatibilité logique**. La structure **DSL** autorise aussi une **adaptation continue** où l’**embedding** vidéo ou audio peut être réentraîné sur de nouvelles données tandis que les **règles** du domaine se mettent à jour, qu’il s’agisse de nouvelles situations réglementaires ou de nouveaux types d’alarmes, le tout sans remettre en cause l’ensemble des acquis du **SCN**.

3.6.3.2. Agent conversationnel : combine “règles” pour la cohérence contextuelle + embeddings BERT

Dans le cadre d'un **Deep Synergy Learning (DSL)**, la construction d'un **agent conversationnel** (chatbot) repose sur l'intégration d'une dimension **sub-symbolique** et d'une dimension **symbolique**. L'objectif est de combiner, d'une part, la puissance d'un modèle d'**embeddings** capable de saisir la richesse et la variabilité du langage naturel – en l'occurrence, des représentations contextuelles produites par un modèle comme **BERT** (*Bidirectional Encoder Representations from Transformers*) – et, d'autre part, un ensemble de **règles logiques** qui garantissent la cohérence contextuelle et la maîtrise du dialogue. Cette approche hybride permet ainsi à l'agent d'analyser de multiples formulations textuelles tout en respectant une séquence de **slots** ou de conditions qui encadrent la progression du dialogue.

A. Composante sub-symbolique : embeddings BERT

Lorsqu'un utilisateur énonce une phrase ou un segment textuel, le DSL fait appel à **BERT** pour générer un **embedding** qui représente la sémantique de l'énoncé. On définit ainsi, pour une entité textuelle \mathcal{E}_i ,

$$\mathbf{x}_i = \text{BERT}(\mathcal{E}_i) \in \mathbb{R}^d,$$

où d représente la dimension de l'espace des embeddings. Ce vecteur \mathbf{x}_i encapsule les informations contextuelles et syntaxiques de la phrase. Par exemple, deux phrases exprimant une intention

similaire (même signification même si formulées différemment, malgré des fautes ou l'usage de synonymes) conduiront à des vecteurs dont la **similarité cosinus** sera élevée, exprimée mathématiquement par

$$S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}.$$

La robustesse de BERT, particulièrement lorsqu'il est **fine-tuned** sur des corpus spécifiques à un domaine (assurance, banque, médical, etc.), garantit que même des variations linguistiques (fautes, abréviations) n'altèrent pas significativement la capture de l'intention. Ce niveau de détail permet au DSL de comparer de manière fine les intentions des utilisateurs et de sélectionner le module de dialogue le mieux adapté, en se fondant sur le **rapprochement sémantique** des embeddings.

B. Composante symbolique : règles de cohérence contextuelle

Outre la représentation sub-symbolique, un agent conversationnel doit maintenir une **cohérence logique** dans le déroulement du dialogue. Pour ce faire, le DSL intègre un ensemble de **règles logiques** regroupé dans un bloc R_i associé à chaque entité ou état de dialogue. Ces règles, élaborées en fonction des contraintes du domaine, définissent des conditions d'activation ou de transition. Par exemple, dans un scénario de réservation d'hôtel, il est possible de spécifier que l'utilisateur doit d'abord fournir la date et le lieu avant de passer à la sélection du mode de paiement. Formellement, la **synergie symbolique** peut être représentée par une fonction qui attribue un score en fonction de la conformité des règles, telle que

$$S_{\text{sym}}(R_i, R_j) = \begin{cases} 1, & \text{si les règles se valident mutuellement,} \\ 0, & \text{en cas de contradiction totale,} \\ s_{ij}, & \text{pour une satisfaction partielle,} \end{cases}$$

où $s_{ij} \in (0,1)$ représente un score intermédiaire mesurant le degré de compatibilité entre les blocs de règles associés aux entités \mathcal{E}_i et \mathcal{E}_j . Cette approche assure que l'agent conversationnel ne franchit jamais des étapes incohérentes, préservant ainsi la logique du dialogue et la non-répétition de questions redondantes.

C. Calcul de la Synergie Mixte et Mise à Jour des Pondérations

La force du DSL réside dans l'intégration des deux composantes précédentes afin de définir une **synergie hybride** qui guide la mise à jour des **pondérations** dans le SCN. La synergie hybride, qui évalue simultanément la **proximité sémantique** et la **compatibilité logique**, est formulée comme suit :

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

où le paramètre $\alpha \in [0,1]$ ajuste l'importance relative accordée à la composante sub-symbolique par rapport à la composante symbolique. La **mise à jour** des pondérations du SCN se fait alors selon la règle

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{hybrid}}(i, j) - \tau \omega_{i,j}(t)],$$

où η représente le **taux d'apprentissage** et τ le **coefficient de décroissance**. Cette équation traduit la **plasticité locale** du système où, lorsque la synergie hybride entre deux entités est élevée, indiquant que leurs intentions sont à la fois **sémantiquement proches** et **logiquement cohérentes**, la pondération $\omega_{i,j}$ est renforcée, favorisant ainsi la formation d'enchaînements de dialogue pertinents. Inversement, des valeurs faibles de $S_{\text{hybrid}}(i,j)$ conduisent à un affaiblissement des liaisons, évitant ainsi des transitions incohérentes.

D. Bénéfices et Retours d'Expérience

L'**approche hybride** présente plusieurs **avantages clés** qui garantissent une **synergie optimale** entre les représentations sub-symboliques et symboliques dans le cadre du **DSL**.

La **robustesse linguistique** est assurée grâce à BERT, qui, par son fine-tuning, permet de gérer efficacement les variations dans les formulations textuelles, qu'il s'agisse de fautes d'orthographe, d'abréviations ou de synonymes.

La **cohérence** du dialogue est maintenue par l'intégration d'un ensemble de **règles logiques** qui structurent la progression du dialogue et garantissent que les transitions respectent des contraintes préétablies.

L'**explicabilité accrue** constitue un autre atout majeur de cette approche. Chaque **décision d'enchaînement** peut être **justifiée** à la fois par une **correspondance sémantique** (un **score sub-symbolique** élevé entre deux unités de dialogue) et par la **validation d'un ensemble de règles** (un **score symbolique** confirmant la compatibilité logique).

L'**évolutivité du système** est favorisée par le **SCN**, qui, grâce à la mise à jour continue des pondérations, s'adapte progressivement aux **nouvelles données et aux révisions des règles**. Cette capacité d'adaptation garantit une continuité du dialogue tout en permettant une mise à jour fluide des modules sous-jacents.

En pratique, ces avantages se traduisent par un agent conversationnel capable de gérer des requêtes complexes et variées tout en maintenant un fil de dialogue cohérent et explicable, ce qui est particulièrement pertinent dans des domaines spécialisés.

3.6.3.3. Robotique sensorielle : capteurs sub-symboliques + représentation symbolique des actions/objets

L'association de **capteurs multiples** (caméra, LiDAR, capteur de force, etc.) avec un **ensemble de règles** décrivant les actions et les objets manipulés constitue un scénario riche pour le **Deep Synergy Learning (DSL)**. Le robot, plongé dans un environnement parfois incertain (lumière changeante, bruit sur les mesures), doit à la fois **analyser** les informations sub-symboliques fournies par ses capteurs et **respecter** la logique imposée par ses règles (scripts, protocoles, axiomes de sécurité). Cette double contrainte façonne un réseau **SCN** où chaque état ou action intègre à la fois un **embedding sub-symbolique** et un **bloc** de conditions symboliques, et où la **synergie** commande la mise à jour adaptative des liens.

A. Contexte : un robot multisensoriel

Imaginez un **robot logistique** évoluant dans un entrepôt, équipé de multiples capteurs :

- **Caméra** fournissant un flux d'images. Un **CNN** (ou réseau de vision spécialisé) produit un vecteur $\mathbf{x}_{\text{cam}} \in \mathbb{R}^{d_1}$ décrivant l'information visuelle (objets détectés, caractéristiques de la scène).
- **LiDAR** fournissant un nuage de points 3D. Un encodeur (type *PointNet* ou *autoencodeur 3D*) calcule $\mathbf{x}_{\text{LiDAR}} \in \mathbb{R}^{d_2}$.
- **Capteur de force** détectant la pression exercée par un bras manipulateur, ou la force-torque globale, résumé dans un vecteur $\mathbf{x}_{\text{force}} \in \mathbb{R}^{d_3}$.

Par ailleurs, le robot se réfère à un **bloc symbolique** explicitant des règles de fonctionnement où “si la zone scannée est interdite, le robot doit contourner”, “si l'objet est fragile, la force de préhension doit rester sous un certain seuil” et “si la distance est < 0.5 m et que l'objet est inconnu, refuser l'avancée”.. Ces règles logiques forment un corpus **R**, scindé éventuellement entre celles attachées à l'état (quelles conditions valent pour la position, la posture ?), et celles décrivant les actions (pré-requis pour saisir un objet, interdictions particulières, etc.).

Dans l'esprit d'un **DSL** hybride, chaque entité \mathcal{E}_i peut être un **état** sensoriel, un **objet**, ou une **action** potentielle, muni à la fois :

- d'un **embedding** sub-symbolique $\mathbf{x}_i \in \mathbb{R}^d$ (obtenu, par exemple, en concaténant \mathbf{x}_{cam} , $\mathbf{x}_{\text{LiDAR}}$, $\mathbf{x}_{\text{force}}$ ou un encodeur fusionné),
- d'un **bloc** de règles symboliques \mathbf{R}_i , exprimant, par exemple, la classe d'objet (fragile, volumineux), des restrictions de mouvement, ou un protocole d'action (*pick*, *place*, *push*).

B. Synergie entre capteurs et règles d'action

La fusion sub-symbolique permet d'évaluer la **proximité** ou la **compatibilité** entre deux états sensoriels, en calculant une distance ou une similarité sur leurs embeddings $\mathbf{x}_{\text{state1}}$, $\mathbf{x}_{\text{state2}}$. Parallèlement, si l'on compare un état \mathcal{E}_i et une action \mathcal{A} , la **synergie** inclut la vérification logique où l'on évalue si l'action \mathcal{A} est autorisée dans l'état \mathcal{E}_i et quelles conditions $R_{\mathcal{A}}$ doivent s'appliquer, par exemple “si l'objet est fragile, vérifier que la force < 10 N”. On écrit :

$$S_{\text{hybrid}}(\mathcal{E}_i, \mathcal{A}) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_{\mathcal{A}}) + (1 - \alpha) S_{\text{sym}}(R_i, R_{\mathcal{A}}).$$

Ici, \mathbf{x}_i représente l'**embedding** agrégé de l'état sensoriel du robot (caméra, LiDAR, force), tandis que $\mathbf{x}_{\mathcal{A}}$ peut (ou non) exister selon la nature de l'action (certaines actions se dotent d'un vecteur sub-symbolique décrivant leurs contraintes). Les règles \mathbf{R}_i et $R_{\mathcal{A}}$ reflètent la logique symbolique, comme “ne pas déplacer un objet fragile au-delà de 2 m de hauteur” ou “necessite zone autorisée = TRUE”.

Ce score de **synergie** sub-symbolique + symbolique conditionne l'**auto-organisation** dans le **SCN** où, si \mathcal{A} est fréquemment validée au vu des capteurs (S_{sub} élevé) et respecte toutes les règles ($S_{\text{sym}} \approx 1$), alors le lien $\omega_{(i,\mathcal{A})}$ se **renforce**. Dans la mise à jour hebbienne :

$$\omega_{(i,\mathcal{A})}(t+1) = \omega_{(i,\mathcal{A})}(t) + \eta [S_{\text{hybrid}}(\mathcal{E}_i, \mathcal{A}) - \tau \omega_{(i,\mathcal{A})}(t)].$$

Si, en revanche, l'action apparaît incohérente (la force nécessaire dépasse un seuil, la zone est interdite, etc.), la pondération décroît, évitant que le robot ne réitère cette transition. Avec le temps, le **SCN** se stabilise en **clusters** ou en **chaînes** d'états-actions validées, traduisant un *plan* de manœuvre flexible qui reflète la fois la perception neuronale et les axiomes symboliques.

C. Exécution auto-organisée dans un DSL robotique

Dans un **SCN** robotique, chaque **état** ou **situation** \mathcal{E}_i encapsule un **embedding** (issu des capteurs) et un **bloc** de règles (posture autorisée, zone permise, classe d'objet détecté, etc.). Les **actions** (prendre, poser, avancer, reculer) apparaissent comme des noeuds à part ou comme des transitions rattachées à un état. La force des liaisons $\omega_{(i,\mathcal{A})}$ se met à jour au fil des expériences, chaque essai ou simulation renforçant ou affaiblissant le couple (état, action) selon la synergie $\alpha S_{\text{sub}} + (1 - \alpha) S_{\text{sym}}$.

- **Sub-symboliquement**, si le robot “voit” un objet de forme X (via l'encodage de vision) et ce même objet a été identifié comme Y dans la base, on obtient un haut degré de **similarité**.
- **Symboliquement**, si une règle R impose de ne pas saisir cet objet sans vérifier la force, alors la conformité de la force de préhension agit sur S_{sym} .

Ainsi, la commande “saisir l’objet” depuis un état \mathcal{E}_i (caractérisé par un ensemble de mesures) ne sera validée que si le rapprochement sub-symbolique indique qu’il s’agit bien de l’objet visé (pas de confusion visuelle), et si la règle ne détecte pas d’incohérence (non-fragile, autorisé, zone libre, etc.). Le robot suit alors une **dynamique auto-organisée** — les **liaisons** ω se stabilisent sur des transitions conformes, et s’évanouissent pour celles jugées peu cohérentes.

Exemple

Un **robot picking** manipule divers objets sur un tapis roulant où il peut traiter des boîtes en carton fragile, des pièces métalliques lourdes, et d’autres types d’objets. Les règles imposent que si l’objet est fragile, la force de pincement doit demeurer < 10 N. Le sous-système sub-symbolique (caméra + force sensor) encode la forme et la masse apparente. Quand le robot tente l’action “pincer fort” pour un carton fragile, la synergie symbolique sera quasi nulle (contradiction avec la règle), même si sub-symboliquement la forme du carton correspond à un objet saisissable. Le poids $\omega_{(i,\text{pincerFort})}$ se réduit, tandis que “pincerDoucement” reçoit un renforcement positif. Progressivement, le **SCN** converge vers des gestes adaptés à la nature de chaque objet, tout en respectant la cohérence logicielle.

D. Bénéfices

Les **bénéfices** sont multiples.

La **robustesse** du système repose sur les capteurs sub-symboliques comme la vision, le LiDAR ou les capteurs de force, qui gèrent la **variabilité** de l’environnement. Si la luminosité change, le CNN peut toujours encoder un vecteur cohérent, et le LiDAR un nuage de points exploitable, permettant ainsi à l'**auto-organisation** du DSL de poursuivre son travail sans interruption.

Le **contrôle explicable** est garanti par l'intégration de **règles**, assurant une transparence de l'architecture. Il est possible d'expliquer pourquoi une action est refusée, en raison d'une contrainte de sécurité, ou validée, grâce à la confirmation des capteurs et à la satisfaction des règles établies.

L'**adaptation continue** est facilitée par la flexibilité du DSL. L'ajout d'un nouveau capteur, comme une caméra infrarouge ou un système RFID, s'intègre directement dans l'embedding sub-symbolique. De même, l'introduction d'une nouvelle règle de sécurité dans la base symbolique ne perturbe pas le fonctionnement global, le **DSL** assurant la mise à jour progressive de ω sans rupture du système.

La **scalabilité** du modèle permet une gestion efficace dans des environnements industriels complexes. Un robot peut traiter un grand nombre d'états tout en s'appuyant sur les mêmes principes de synergie. La matrice ω indique quels enchaînements **état → action → nouvel état** sont les plus performants, optimisant ainsi l'efficacité du système.

3.6.4. Comparaison Expérimentale

Tout au long de ce chapitre (3), nous avons présenté divers **types de représentation** (sub-symbolique, symbolique, hybride) et plusieurs **approches** avancées (hypergraphes, fractales). Une question cruciale demeure où il s'agit de déterminer **comment** comparer expérimentalement les performances d'un **DSL** (Deep Synergy Learning) en fonction du **choix** de la représentation. Dans cette section (3.6.4), nous abordons les éléments clés d'une **étude comparative** :

- (3.6.4.1) L'**impact** du type de représentation sur la **qualité** de la synergie,
- (3.6.4.2) Les **mesures** de clustering, le temps de convergence et la robustesse face à l'insertion de nouvelles entités,
- (3.6.4.3) L'**approche HPC** pour gérer un grand nombre de dimensions ou un grand volume de données.

3.6.4.1. Impact du type de représentation (sub, sym, hybride) sur la qualité de la synergie

Dans le cadre d'un **Deep Synergy Learning (DSL)**, le choix de la représentation des entités constitue un paramètre déterminant pour la qualité de la synergie entre celles-ci, c'est-à-dire pour l'efficacité de l'auto-organisation du réseau de connexion. Selon qu'on opte pour une approche **sub-symbolique** (basée sur des **embeddings neuronaux**), une approche **symbolique** (fondée sur des **règles**, des **axiomes** ou des **ontologies**), ou une approche **hybride** qui combine ces deux modalités, la structure des liaisons $\{\omega_{i,j}\}$ ainsi que la formation des clusters peuvent varier de manière significative. En effet, ces différences se manifestent principalement en termes de **robustesse** face au bruit, de **lisibilité** ou **explicabilité**, de **capacité de raisonnement** et de **scalabilité**.

A. Comparaison entre les représentations sub-symbolique, symbolique et hybride

Dans une approche **sub-symbolique**, chaque entité \mathcal{E}_i est représentée par un **embedding** $\mathbf{x}_i \in \mathbb{R}^d$ issu d'un modèle d'apprentissage profond, lequel convertit des données brutes (images, extraits audio, phrases textuelles, etc.) en vecteurs qui condensent les caractéristiques les plus saillantes. La similarité entre deux entités est évaluée par des mesures classiques telles que la **similarité cosinus** ou la **distance euclidienne**. Par exemple, la similarité cosinus est donnée par

$$S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|},$$

ce qui reflète le degré de rapprochement angulaire entre \mathbf{x}_i et \mathbf{x}_j . Cette approche se caractérise par une **souplesse** et une **robustesse** intrinsèque, en particulier pour gérer un grand volume de données potentiellement bruitées ; toutefois, elle présente l'inconvénient d'être peu **explicable**, car il est difficile de déterminer explicitement quelles caractéristiques contribuent à la proximité entre deux embeddings, hormis par la valeur numérique obtenue.

En revanche, dans une approche **symbolique**, chaque entité \mathcal{E}_i est décrite par un **bloc** de règles, d'axiomes ou d'attributs logiques, noté R_i . La similarité entre deux entités se mesure alors par un score $S_{\text{sym}}(R_i, R_j)$ qui quantifie la **compatibilité** entre les ensembles de règles ou les structures logiques associées. Par exemple, une fonction de compatibilité peut être formulée comme

$$S_{\text{sym}}(R_i, R_j) = \begin{cases} 1, & \text{si les règles se valident mutuellement,} \\ 0, & \text{en cas de contradiction totale,} \\ s_{ij}, & \text{pour une compatibilité partielle,} \end{cases}$$

où s_{ij} est un score intermédiaire compris entre 0 et 1. Cette approche apporte une **explicabilité** élevée, puisque la logique sous-jacente offre la possibilité de retracer précisément pourquoi deux entités sont considérées comme compatibles, bien que le traitement symbolique soit souvent plus rigide et moins tolérant aux variations ou au bruit inhérent aux données.

Le **modèle hybride** vise à combiner le meilleur des deux approches précédentes. Dans ce cas, chaque entité est dotée à la fois d'un **embedding** \mathbf{x}_i et d'un bloc de règles R_i . La synergie globale entre deux entités \mathcal{E}_i et \mathcal{E}_j est alors définie par la formule

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

où $\alpha \in [0,1]$ est un paramètre qui ajuste l'importance relative de la **dimension sub-symbolique** par rapport à la **dimension symbolique**. Ce schéma de fusion permet d'exploiter la **robustesse** des embeddings pour absorber le bruit et les variations lexicales, tout en bénéficiant du **raisonnement formel** et de la **träçabilité** qu'apporte la couche symbolique. Néanmoins, il nécessite un double stockage des informations (vecteur et règles) et des calculs supplémentaires, ce qui augmente la **complexité** de l'implémentation. La sélection judicieuse de α est cruciale pour obtenir le bon compromis entre **souplesse** et **précision**.

B. Métriques pour évaluer la qualité de la synergie

Pour évaluer la **qualité** des liaisons $\{\omega_{i,j}\}$ établies dans un DSL, il est important d'utiliser des **métriques** qui permettent de quantifier à la fois la **compacité** des clusters (mesurée par des indices

tels que l'indice de Silhouette ou le Davies–Bouldin Index) et la **séparation** entre ces clusters. Lorsque l'approche sub-symbolique est employée, la justification d'un regroupement se base essentiellement sur la proximité numérique entre les embeddings. En revanche, une approche hybride offre la possibilité de lier la **proximité sémantique** à des critères explicites, c'est-à-dire de fournir une explication formelle à la raison pour laquelle deux entités se trouvent dans le même cluster. Cette capacité à fournir une **explication** riche renforce la lisibilité des résultats, en permettant par exemple d'associer à un regroupement une liste de règles communes ou un indice de cohérence formelle qui soutient la formation des clusters.

C. Observations empiriques et compromis

Dans la pratique, l'approche **purement sub-symbolique** se distingue par sa **robustesse** et sa capacité à absorber de grandes quantités de données bruitées, tout en permettant un calcul rapide des similarités à l'aide de fonctions vectorielles telles que le cosinus ou la distance euclidienne. Toutefois, cette méthode reste limitée en termes d'**explicabilité**, car elle ne fournit qu'un score numérique sans indication sur les motifs formels qui sous-tendent la similarité. En revanche, une approche **purement symbolique** offre une **cohérence** et une **transparence** importantes, puisque chaque regroupement peut être expliqué par des règles logiques précises, mais elle est souvent rigide et sensible aux incohérences, rendant sa mise en œuvre coûteuse en calcul lorsque la base de règles est très large. Le modèle **hybride** se présente alors comme un compromis idéal, car il combine la **flexibilité** et la **robustesse** sub-symbolique avec la **rigueur** et la **clarté** symboliques. Cependant, cette approche hybride impose une augmentation du coût de stockage et de calcul, et requiert un réglage fin du paramètre α pour assurer le bon équilibre entre les deux composantes. En général, les expérimentations montrent qu'un DSL hybride parvient à générer des clusters d'entités plus **cohérents** et plus **explicables** que les approches purement sub-symboliques, tout en offrant une meilleure tolérance au bruit que les systèmes strictement symboliques.

3.6.4.2. Mesures de clusters, temps de convergence, robustesse à l'insertion de nouvelles entités

Dans le cadre d'un **Deep Synergy Learning (DSL)**, l'évaluation du comportement du système ne se limite pas à l'analyse statique de la structure auto-organisée, mais englobe également des aspects dynamiques essentiels, notamment la **qualité des clusters**, le **temps de convergence** du réseau ainsi que la **robustesse** face à l'insertion de nouvelles entités. Ces trois dimensions permettent de mesurer de façon approfondie la stabilité et la scalabilité du **Synergistic Connection Network (SCN)**. L'analyse de ces critères s'appuie sur des **indices mathématiques** et des **métriques** qui quantifient la compacité des regroupements, le temps nécessaire à la stabilisation des pondérations, ainsi que la capacité du système à intégrer de nouvelles informations sans perturber la structure existante.

A. Mesures de qualité des clusters

Pour évaluer la qualité des clusters obtenus dans le SCN, on se réfère à des **indices de partition** qui mesurent la compacité intra-cluster et la séparation inter-clusters. Par exemple, l'**indice de Silhouette** est défini pour une entité \mathcal{E}_i par

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},$$

où $a(i)$ représente la distance moyenne entre \mathcal{E}_i et les autres entités du même cluster, et $b(i)$ la distance moyenne entre \mathcal{E}_i et les entités du cluster le plus proche. Un score global s proche de 1 indique que les clusters sont bien définis et distincts. De même, l'**indice de Davies–Bouldin** s'exprime par

$$\text{DB} = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left\{ \frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right\},$$

où k est le nombre de clusters, σ_i la dispersion moyenne des entités dans le cluster i , et $d(c_i, c_j)$ la distance entre les centres c_i et c_j de clusters différents. Un indice DB plus faible signifie des clusters compacts et bien séparés. Par ailleurs, l'**indice Calinski–Harabasz** se définit comme

$$\text{CH} = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{n-k}{k-1},$$

où $\text{Tr}(B_k)$ et $\text{Tr}(W_k)$ représentent respectivement la trace de la matrice de dispersion inter-clusters et intra-clusters, n étant le nombre total d'entités. Ces indices, appliqués aux représentations obtenues par le DSL – qu'elles soient sub-symboliques, symboliques ou hybrides – permettent d'évaluer de manière quantitative la **qualité des regroupements**. Dans un système hybride, par exemple, la combinaison d'un score basé sur la **proximité vectorielle** et d'un score issu des **règles logiques** offre une mesure plus riche et explicative, où l'on peut retracer l'origine d'un regroupement à la fois par la similarité des embeddings et par la cohérence des règles associées.

B. Temps de convergence

Le **temps de convergence** du SCN est une mesure dynamique qui reflète la rapidité avec laquelle les pondérations $\omega_{i,j}(t)$ se stabilisent. La mise à jour des poids dans le DSL s'exprime par l'équation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $S(i,j)$ représente la synergie entre les entités, η le taux d'apprentissage et τ le coefficient de décroissance. La convergence se produit lorsque $\omega_{i,j}(t+1) \approx \omega_{i,j}(t)$ pour toutes les paires (i,j) . On peut définir un critère de convergence, par exemple, lorsque

$$\max_{i,j} |\omega_{i,j}(t+1) - \omega_{i,j}(t)| < \varepsilon,$$

où ε est un seuil de tolérance fixé. Dans les systèmes **sub-symboliques**, le calcul des similarités vectorielles est généralement rapide, ce qui permet une convergence relativement rapide, à condition que la dimension d ne soit pas excessive. En revanche, dans les systèmes **symboliques** et **hybrides**, les opérations logiques et la fusion des scores augmentent la complexité, rallongeant ainsi le temps de convergence. L'évaluation empirique du temps de convergence permet de mesurer la **vitesse d'auto-organisation** et d'ajuster les hyperparamètres η et τ de façon à optimiser la stabilisation des clusters.

C. Robustesse à l'insertion de nouvelles entités

Un critère fondamental pour l'évolutivité d'un DSL est la capacité du SCN à intégrer de nouvelles entités sans perturber la structure auto-organisée préexistante. Si une nouvelle entité \mathcal{E}_{new} dotée d'un embedding \mathbf{x}_{new} (et, dans le cas d'un modèle hybride, d'un bloc de règles R_{new}) est insérée, il est souhaitable que les nouveaux liens $\omega_{\text{new},i}$ soient établis de manière **locale** et que la structure globale ne soit pas remise en cause. En pratique, le DSL compare \mathbf{x}_{new} aux embeddings existants et, éventuellement, vérifie la compatibilité des règles associées pour calculer la synergie

$$S(\mathcal{E}_{\text{new}}, \mathcal{E}_i),$$

afin d'initier les pondérations $\omega_{\text{new},i}$. Cette approche permet de garantir une **insertion locale** sans nécessiter une re-synchronisation globale du SCN. La robustesse du système peut être évaluée par des métriques telles que le **changement moyen** des pondérations suite à l'ajout de nouvelles entités ou par des tests de stabilité de la structure de clusters, par exemple en mesurant l'évolution de l'indice de Silhouette ou du Davies–Bouldin Index avant et après l'insertion. Un DSL robuste sera capable de s'adapter de manière incrémentale, de sorte que l'ajout d'une entité provoque des ajustements locaux limités plutôt qu'une réorganisation massive du réseau.

3.6.4.3. Approche HPC si le DSL manipule un grand nombre de dimensions

Lorsqu'un **Deep Synergy Learning (DSL)** se déploie à large échelle, que ce soit en raison de la **dimension** élevée des vecteurs de représentation (embeddings sub-symboliques de plusieurs centaines ou milliers de composantes) ou du **nombre** massif d'entités manipulées, la charge de calcul pour maintenir la **synergie** $\{S(i,j)\}$ et la mise à jour itérative des $\{\omega_{i,j}\}$ peut rapidement croître au point de devenir prohibitive. Dans de telles configurations, une **approche HPC (High Performance Computing)** devient indispensable pour gérer de manière parallèle ou distribuée l'important volume d'opérations. L'idée consiste à recourir à des ressources matérielles avancées (GPU, clusters multiprocesseurs) et à des algorithmes spécifiques permettant de limiter l'impact de la croissance en $O(n^2d)$, où n est le nombre total d'entités et d la taille des embeddings.

A. Échelle et complexité dans un DSL de grande dimension

Si l'on considère un **DSL** manipulant $\mathbf{x}_i \in \mathbb{R}^d$ pour chaque entité \mathcal{E}_i , et qu'on souhaite évaluer une synergie sub-symbolique via un **produit scalaire** ou une **distance** de la forme

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \sum_{k=1}^d (x_{i,k} - x_{j,k})^2,$$

le coût se chiffre rapidement en $O(n^2d)$ si on compare toutes les paires (i,j) . Pour un simple examen exhaustif de quelques centaines de milliers d'entités (ou davantage) en embeddings de dimension 768, le nombre d'opérations requises s'avère colossal. La **composante symbolique**, lorsqu'elle est présente, peut ajouter une vérification de règles dont la **complexité** peut être encore plus élevée, parfois de nature exponentielle selon la logique mise en œuvre. Il devient alors essentiel d'exploiter des **moyens de calcul parallèle** et de **techniques d'approximation** pour préserver la faisabilité du DSL.

La mise à jour répétée des pondérations, exprimée par exemple sous la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

exige de réévaluer partiellement ou globalement la **synergie** $\{S(i,j)\}$ à chaque itération, d'où la nécessité d'un **High Performance Computing**. Sans ce levier, il est inenvisageable de mettre à jour une matrice de taille $O(n^2)$ dès que n atteint plusieurs centaines de milliers d'entités, surtout si la logique impose un raisonnement ponctuel pour vérifier la compatibilité de certaines règles $\mathbf{R}_i, \mathbf{R}_j$.

B. Parallélisation sub-symbolique et distribution symbolique

La partie **sub-symbolique** (calcul de similarité entre embeddings) peut être largement parallélisée sur GPU ou CPU multicœurs. Le produit scalaire

$$\mathbf{x}_i \cdot \mathbf{x}_j = \sum_{k=1}^d (x_{i,k} x_{j,k})$$

peut se répartir par blocs matriciels ou par batchs, chaque nœud HPC traitant un sous-groupe de vecteurs. Dans un DSL géant, il est fréquent d'employer des algorithmes d'**approximate nearest neighbors** (Faiss, Annoy, HNSW) pour éviter un balayage exhaustif en $O(n^2)$. On réduit ainsi la construction du **SCN** aux seuls couples (i,j) dont la distance vectorielle est jugée potentiellement faible, ce qui autorise la parcimonie.

La partie **symbolique**, impliquant des règles ou des axiomes logiques \mathbf{R}_i , bénéficie parfois d'un déploiement sur des *reasoners* distribués ou des bases de connaissances échelonnées en grappes de serveurs. La compatibilité logique $\mathbf{R}_i, \mathbf{R}_j$ s'examine alors de manière partiellement parallèle, avec synchronisation des conclusions (contradiction, recouvrement) uniquement lorsqu'un sous-groupe d'entités sub-symboliquement proches exige une vérification fine. Cette stratégie "two-layers" consiste à filtrer sur la proximité neuronale et ne procéder au test symbolique complet que pour quelques paires pertinentes.

C. Organisation du SCN à grande échelle

Le **SCN** (Synergistic Connection Network) se construit de manière incrémentale et partiellement distribuée. Pour un lot de données, un ensemble de machines HPC calcule les voisainages k-NN dans l'espace \mathbb{R}^d . Chaque entité se voit attribuer une liste de candidats $\mathcal{N}_i \subset \{1, \dots, n\}$ correspondant à ses proches en embedding. Pour chaque couple $(i,j) \in \mathcal{N}_i$, on évalue la **composante symbolique** éventuelle $\mathbf{R}_i, \mathbf{R}_j$. On obtient ainsi

$$S_{\text{hybrid}}(i,j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

que l'on exploite pour initialiser ou ajuster la pondération $\omega_{i,j}$. La matrice $\{\omega_{i,j}\}$ reste parcimonieuse, car on n'active que les liens dépassant un certain seuil, ou répondant à une condition d'intérêt sub-symbolique ou symbolique. À l'issue de chaque vague d'itérations, un protocole de synchronisation (barrière, agrégation) peut consolider les résultats.

Dans un tel schéma **HPC**, la **synchronisation** demeure un facteur critique où, si les nœuds traitent des partitions disjointes, la cohérence globale du **DSL** requiert un échange périodique des informations sur les couples inter-partitions. Les **paramètres** η (taux d'apprentissage), τ (décroissance) et α (poids sub-symbolique vs. symbolique) nécessitent des ajustements pour que l'on évite les oscillations dues à la latence ou à l'asynchronie.

Cette organisation rend cependant possible le traitement d'entités en nombre très élevé (jusqu'à des dizaines ou centaines de millions si l'on dispose d'assez de ressources), car on exploite des **librairies HPC** spécialisées (MPI, CUDA, OpenMP) et des *frameworks* (Spark, Hadoop) adaptés aux graphes distribués. Les opérations en $O(n^2d)$ sont ainsi contournées par des mécanismes de filtrage (approximate k-NN) et de parallélisation.

D. Gains, contraintes et perspectives

L'approche HPC ouvre la voie à un **DSL** capable de gérer simultanément de larges volumes de données sub-symboliques et un grand nombre de règles. On préserve la **richesse** de l'auto-organisation (renforcement local $\Delta\omega_{i,j}$) sans voir les temps d'exécution exploser, grâce à une exploitation massive du calcul parallèle. Le principal **gain** est la capacité à mettre à jour la synergie $\{\omega_{i,j}\}$ pour plusieurs millions d'entités en un temps acceptable, tout en intégrant les compatibilités logiques.

Les **limites** résident dans la **complexité de déploiement** où il est nécessaire de configurer des clusters GPU ou multicœurs, de mettre en place des reasoners distribués et de synchroniser les partitions du **SCN**. Les **coûts** (financiers et en ingénierie) peuvent être considérables, réservant cette mise en œuvre à des instituts de recherche ou à des organisations gérant déjà des infrastructures HPC. De plus, l'approche reste conditionnée à des **heuristiques** de réduction (approximate k-NN, restrictions sur la portée des règles) pour rendre la dynamique $\omega_{i,j}(t)$ réellement scalable.

Néanmoins, ces méthodes **HPC** s'avèrent cruciales pour faire évoluer un **Deep Synergy Learning** vers des scénarios *big data*, où le DSL incorpore des embeddings dimensionnels imposants (issus de modèles neuronaux à grande échelle) et une **composante symbolique** non triviale. Il devient alors possible de concilier la **puissance** statistique (sub-symbolique) et la **cohérence** ou l'**explicabilité** (symbolique), malgré la croissance spectaculaire de la dimension d et du nombre n de nœuds, le tout orchestré par l'infrastructure HPC.

3.7. Conclusion

Au terme de ce **Chapitre 3**, nous avons examiné en profondeur la **représentation** des entités dans un cadre **DSL** (Deep Synergy Learning). Du choix d'une **approche sub-symbolique** (vecteurs, embeddings) à celui d'une **approche symbolique** (règles, ontologies), en passant par les **solutions hybrides** mêlant les deux, nous avons souligné à quel point la **qualité** de cette représentation influe directement sur la **synergie** calculée (et, partant, sur la structure du Synergistic Connection Network). Nous avons aussi abordé des **extensions avancées** (synergie n-aire, modèles fractals, multimodalité, etc.) et illustré ces principes à travers plusieurs **études de cas** (analyse audio-visuelle, agent conversationnel, robotique sensorielle).

3.7.1. Synthèse des Contributions du Chapitre

La question de la **représentation** d'une entité \mathcal{E}_i dans un **Deep Synergy Learning (DSL)** a émergé comme un élément fondamental, dont dépend l'intégralité de la **fonction** de synergie $S(i, j)$ et, par voie de conséquence, la dynamique de mise à jour $\omega_{i,j}(t)$. Les analyses conduites dans ce chapitre ont démontré l'extrême diversité des **approches** possibles, depuis des **embeddings** neuronaux (sub-symboliques) jusqu'aux **règles** logiques (symboliques), en passant par des schémas **hybrides** plus complexes.

A. Diversité des approches de représentation

L'exploration a porté sur trois **familles** principales. D'abord, la **représentation sub-symbolique** s'appuie sur un **embedding** dans \mathbb{R}^d , souvent calculé via des réseaux neuronaux (par exemple, BERT, GPT, ViT). Cette forme de modélisation met l'accent sur la **similarité vectorielle**, qu'il s'agisse d'une distance euclidienne ou d'un produit scalaire normalisé (cosinus). Une telle approche assure une **tolérance** appréciable au bruit et aux variations sémantiques, ce qui s'avère précieux pour de grandes bases de données textuelles ou visuelles. En revanche, l'absence de structure explicite rend l'interprétation plus délicate et limite les possibilités de **raisonnement** formel.

Une seconde voie consiste en des représentations **symboliques**, articulées autour de **règles**, **d'axiomes** ou **d'ontologies**. Cette méthode repose sur une **cohérence déclarative** où la synergie $S_{\text{sym}}(i, j)$ naît de la compatibilité ou de l'absence de contradiction entre les blocs logiques attachés à \mathcal{E}_i et \mathcal{E}_j . Pareille approche garantit une **transparence** conceptuelle, un **raisonnement** potentiellement exact, et la capacité de vérifier la non-contradiction de façon formelle. On lui reproche cependant une certaine rigidité face à un monde bruyant ou incertain, et une complexité algorithmique parfois élevée si la logique est riche (OWL, règles complexes).

Enfin, la troisième famille est **hybride** où elle associe le meilleur des deux précédentes en combinant la robustesse statistique des embeddings sub-symboliques et la force explicative ou normative de la logique. Le calcul de synergie devient alors un **mélange**, par exemple

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

de sorte que l'on bénéficie d'une tolérance au bruit couplée à une explicabilité formelle. Cette combinatoire exige néanmoins un paramétrage plus lourd et un certain surcoût algorithmique.

Cette **diversité** d'approches reflète la pluralité des **domaines** auxquels se confronte un DSL. Dans certains cas (images, flux sensoriels massifs), l'embedding sub-symbolique domine ; dans d'autres (règlementaire, médical), les règles symboliques fournissent le cadre essentiel ; les scénarios mixtes justifient l'usage d'une représentation hybride.

B. Importance de la qualité de la représentation

L'acte même de calculer la synergie $S(i, j)$ dépend étroitement de la **qualité** de la représentation employée pour \mathcal{E}_i et \mathcal{E}_j . Si celle-ci est incomplète, trop approximative ou mal adaptée, la fonction de similarité ou de distance devient peu fiable, et les pondérations $\omega_{i,j}$ s'en ressentent où la **stabilité** de la mise à jour, la cohésion des clusters et l'émergence d'une structure interprétable dans le **SCN** se voient compromises. À l'inverse, une **représentation** minutieusement pensée – qu'il s'agisse d'un embedding « calibré » ou d'une base de règles sans redondances inutiles – renforce la **pertinence** des liens et la **lisibilité** du réseau.

Le chapitre a insisté sur le fait que le choix initial (sub-symbolique pur, logique pure, ou mixte) influence la manière dont évolue la distribution des $\omega_{i,j}$. Une représentation essentiellement vectorielle engendre une organisation liée aux distances dans \mathbb{R}^d (clustering géométrique), tandis qu'une représentation purement symbolique façonne des **groupes** souvent déterminés par la logique d'inclusion, de contradiction ou de compatibilité d'axiomes. Les systèmes hybrides donnent des clusters plus sélectifs où l'on exige à la fois une proximité numérique et une cohérence de règles, ce qui affine la formation d'ensembles stables.

C. Aspects avancés : synergie n-aire, fractalité, embeddings transformers

Au-delà de la traditionnelle comparaison binaire, le chapitre a montré que la **synergie** peut se généraliser à des **structures n-aires** (hypergraphes) où l'on compare simultanément $\{\mathcal{E}_{i_1}, \dots, \mathcal{E}_{i_k}\}$. Cette extension s'avère primordiale dès que l'on veut saisir la configuration conjointe de plusieurs modalités (ex. audio–vidéo–texte). Elle impose une représentation capable de décrire des entités multiples au sein d'un même **bloc** de calcul.

La **fractalité**, ou **auto-similarité**, constitue une autre dimension abordée où certaines représentations fractales, comme les embeddings multirésolution ou les ontologies fractales, offrent une **compression** et une **récurrence** conceptuelle à plusieurs niveaux. Les bénéfices se révèlent dans des scénarios multi-échelle (Chap. 6) où l'on exploite la réitération systématique d'un même schéma à différentes granularités.

La question des **embeddings avancés** issus de **Transformers** (BERT, GPT, ViT) a souligné l'opportunité d'accéder à des **vecteurs** plus riches et plus contextuels, tout en augmentant la complexité de calcul et le besoin potentiel d'**approches HPC** pour gérer d'énormes ensembles de données.

Synthèse

L'axe majeur de ce **Chapitre 3** tient donc à la mise en évidence du rôle central de la **représentation** dans un **DSL** où toutes les formules $\omega_{i,j}(t + 1)$ et tous les patterns $\{\omega_{i,j}\}$ observés à la fin de la convergence découlent in fine de **comment** chaque entité \mathcal{E}_i est encodée. L'**embedding** sub-symbolique, la base de **règles** logiques ou l'approche **hybride** imposent un langage qui influence :

- La robustesse au bruit (approximation, flexibilité neuronale),
- La cohérence explicable (raisonnement symbolique, absence de contradiction),
- La complexité (c'est-à-dire le coût temporel et matériel de calcul),
- La forme même des “clusters” ou “hyper-clusters” émergents.

Les **prochaines** parties (Chapitres 4, 5, 6) s'appuieront précisément sur cette **base** pour décrire la manière dont un **SCN** organise ses entités, étudie la dynamique de mise à jour (convergence, attracteurs) et peut s'étendre à des architectures distribuées ou multi-échelle, confirmant que les **choix** opérés dans le présent chapitre quant à la **représentation** constituent l'ossature de toute la suite du **Deep Synergy Learning**.

3.7.2. Limites et Pistes Futures

Malgré la diversité d'approches abordées au **Chapitre 3** – qu'il s'agisse de représentations **sub-symboliques** par embeddings neuronaux, de descriptions **symboliques** formées de règles ou d'axiomes, ou encore de modèles **hybrides** fusionnant les deux dimensions –, il demeure manifeste qu'aucune formulation n'est universellement optimale. Chaque méthode se plie à des **besoins** précis, des **données** particulières et des **ressources** de calcul données, si bien que le choix d'une représentation dépend étroitement du contexte. Il subsiste aussi de nombreux axes de **recherche**, qu'il s'agisse d'unification, de standardisation ou de perfectionnement des mécanismes de synergie au sein d'un **DSL** (*Deep Synergy Learning*).

A. Pas de représentation “unique” optimale

La première observation tient au fait qu'il n'existe pas de **format** apte à couvrir tous les cas d'usage avec la même efficacité. Les **embeddings** sub-symboliques (calculés dans \mathbb{R}^d) s'imposent lorsque l'on traite un grand volume de données bruitées, telles que des images, du son ou des segments textuels en langage naturel. Ils offrent une **robustesse** notable aux variations aléatoires et aux erreurs mineures, mais n'apportent pas la même explicabilité formelle que des règles logiques. À l'inverse, une approche strictement **symbolique** excelle pour des tâches exigeant un **raisonnement** strict, un contrôle de la **cohérence** ou une traçabilité explicite des décisions (domaine médical, légal, etc.). Cependant, elle se révèle rigide dès lors qu'apparaissent des informations ambiguës ou bruitées que la logique pure gère difficilement. L'option **hybride** tente de réconcilier ces deux mondes en calculant la **synergie** de façon mixte, ce qui peut apporter une meilleure **interprétabilité** tout en conservant une certaine tolérance aux imprécisions, au prix d'une infrastructure plus complexe et d'un temps de calcul plus important.

Un second aspect a trait aux **ressources** de calcul où des représentations sub-symboliques de grande dimension ou des bases de règles logiques massives peuvent exiger des **moyens HPC** (High Performance Computing) pour atteindre la performance requise. Dans certaines situations où l'on dispose de peu de capacité machine, il est parfois plus judicieux de réduire la complexité des modèles, soit en limitant la taille des embeddings, soit en simplifiant l'ontologie ou les règles symboliques, afin de maintenir le **DSL** dans des délais acceptables. Enfin, la **capacité d'évolution** du **DSL** joue un rôle où, lors d'un apprentissage continu, de nouvelles entités et de nouveaux attributs apparaissent. Une représentation sub-symbolique est souvent plus facile à “raffiner” (par

un fine-tuning local) qu'une base de règles, dont la moindre modification peut exiger de recomposer la cohérence globale.

B. Recherche : standardisation des formats, outils pour la synergie multi-modal ou multi-domaine

Les différentes méthodes rencontrées dans le **Chapitre 3** soulignent la nécessité de **standards** où chaque domaine, qu'il s'agisse de vision, de son, d'ontologies, de textes ou de robotique, recourt à un système distinct, comme **RDF**, **OWL** pour la partie symbolique et **ONNX ou TF SavedModel** pour la partie neuronale. L'élaboration d'un format **commun** – permettant de décrire à la fois un bloc symbolique \mathbf{R}_i et un embedding \mathbf{x}_i de façon unifiée – pourrait fluidifier l'intégration de modules hétérogènes dans un **DSL**. Cette standardisation faciliterait également la réutilisation de bibliothèques ou l'échange de modèles entre différents domaines.

Les expériences en **multimodalité** (où l'on fusionne plusieurs types de signaux) et en **multidomaine** (où un même **SCN** chapeaute des entités relevant de divers secteurs) suggèrent de développer des **kernels** ou des **fonctions de synergie** plus avancées, aptes à combiner plusieurs embeddings à la fois tout en tenant compte d'éventuelles règles logiques. Il devient alors crucial de définir :

$$S_{\text{hybrid}}(\mathcal{E}_i, \mathcal{E}_j) = \alpha_1 S_{\text{sub},1}(\mathbf{x}_{i,1}, \mathbf{x}_{j,1}) + \cdots + \alpha_m S_{\text{sub},m}(\mathbf{x}_{i,m}, \mathbf{x}_{j,m}) + (1 - \alpha_\Sigma) S_{\text{sym}}(R_i, R_j),$$

où $\{\mathbf{x}_{i,k}\}$ représentent différents **embeddings** issus de modalités ou de domaines divergents, et $\{R_i\}$ décrivent leurs attributs symboliques. La détermination optimale des $\{\alpha_k\}$ constitue un enjeu de calibration, pour équilibrer la part d'influence de chacune des modalités face à la couche logique. Il s'agit d'une **piste de recherche** encore largement ouverte.

C. Lien possible avec la sécurité (Chap. 11) et la convergence (Chap. 4, 7)

Les représentations discutées ici interfèrent avec d'autres volets du **Deep Synergy Learning**. Du point de vue de la **sécurité** (Chap. 11), la robustesse d'un modèle sub-symbolique s'avère cruciale pour repérer des signaux anormaux (par exemple dans des embeddings modifiés de manière adverse), tandis qu'une base de règles symboliques permet de justifier pourquoi l'on suspecte un comportement malveillant, ou de repérer un usage qui contrevient à certaines politiques. Un DSL “hybride” renforce donc cette défense en alliant détection statistique et validation logique.

Sur le plan de la **convergence** (Chapitres 4 et 7), la présence d'une logique symbolique, parfois discontinue (une légère modification peut changer radicalement la compatibilité de deux blocs de règles), peut introduire des points de non-lisséité dans la fonction de synergie globale, ce qui soulève des problèmes de stabilité et de temps de convergence. Les algorithmes d'optimisation et les garanties de stabilisation du SCN peuvent devenir plus complexes que dans un cadre sub-symbolique lisse (où la distance euclidienne donne une fonction continue). On entrevoit ici une **piste d'investigation** consistant à examiner la dynamique $\{\omega_{i,j}(t)\}$ lorsque la composante symbolique introduit des mises à jour discrètes et non linéaires.

3.7.3. Liens avec les Chapitres Suivants

Les développements du **Chapitre 3** ont mis en lumière la **diversité des représentations** – sub-symboliques, symboliques, hybrides, voire plus avancées (n-aires, fractales) – et la **manière** dont

elles peuvent se combiner dans un **DSL** (*Deep Synergy Learning*). Il reste cependant à comprendre **comment**, à partir de ces entités ainsi représentées, se déploie toute la **dynamique** d'auto-organisation et **comment** cette architecture s'inscrit dans un cadre global multi-échelle. Les chapitres ultérieurs (4, 5, 6) poursuivront précisément dans cette direction, en s'appuyant sur les notions introduites ici.

A. Chapitre 4 : Dynamique d'Auto-Organisation

Le **Chapitre 4** approfondira la **dynamique** propre au **SCN** (Synergistic Connection Network), c'est-à-dire le **mécanisme** régissant l'évolution des pondérations $\omega_{i,j}$ au fil des itérations. Cette dimension dynamique n'a été que brièvement abordée jusqu'ici, puisque le présent chapitre (3) se focalisait surtout sur la **nature** des entités $\{\mathcal{E}_i\}$ et la **forme** de leur représentation $\mathbf{r}(i)$. Les différentes **représentations** suggèrent en effet des **fonctions de synergie** distinctes, et donc des **équations** de mise à jour différentes.

Lorsqu'un DSL recourt à des embeddings sub-symboliques, la fonction $S_{\text{sub}}(i,j)$ repose sur une distance continue (euclidienne, cosinus), engendrant une mise à jour “lisse” qui tend à éviter les discontinuités. À l'inverse, une représentation symbolique peut amener des discontinuités logiques (contradiction vs. non-contradiction), rendant la dynamique plus délicate à analyser. Le **Chapitre 4** explicitera comment chaque type de représentation influe sur la **vitesse** et la **stabilité** de la convergence ($\Delta\omega_{i,j} \approx 0$), et examinera la notion d'**attracteurs** ou de **clusters** stables qui se forment lorsque le SCN atteint l'équilibre.

B. Chapitre 5 : Architecture SCN et Exploitation Pratique

Dans le **Chapitre 5**, on s'intéressera à la **structure globale** du **Synergistic Connection Network**, c'est-à-dire la manière dont on assemble concrètement les entités dans un graphe adaptatif. Les entités que l'on a décrites ici (qu'elles soient sub-symboliques, symboliques ou hybrides) y prennent la forme de *nœuds* ou de modules, reliés par des pondérations $\omega_{i,j}$. Ce chapitre décrira plus avant les **algorithmes d'exploitation** du SCN, par exemple pour :

- Partitionner le réseau en sous-groupes (macro-clusters),
- Mettre en place un système distribué ou modulaire,
- Gérer la dynamique en parallèle (HPC, chap. 3.6.4.3).

On clarifiera aussi les **formes** que peut prendre le **SCN** où il peut s'agir d'un simple graphe non orienté, d'un hypergraphe ou d'une architecture hiérarchique où les liens s'organisent en couches. Les choix de **représentation** exposés au Chap. 3 déterminent en partie la **manière** dont on code les attributs ou règles associées aux nœuds, et influencent la **conception** même de l'architecture SCN.

C. Chapitre 6 : Multi-Échelle, Fractalité et Hiérarchies

Le **Chapitre 6** approfondira la **dimension multi-échelle** déjà évoquée. Les représentations fractales (section 3.6.2.2) et les constructions n-aires se prêteront à une **organisation** hiérarchique ou **fractal-like**. Il sera montré que l'on peut concevoir le DSL comme un système où plusieurs **niveaux** (micro, macro) interagissent, chacun recourant aux mêmes principes de **synergie**, mais adaptés à l'échelle correspondante. On y détaillera également comment un SCN fractal, ou multi-

échelle, peut renforcer l'**auto-organisation**, et comment la formation simultanée de **clusters** locaux et globaux démultiplie la puissance de l'approche.

Le lien avec les représentations décrites au **Chapitre 3** est direct où un embedding fractal, par exemple, prend tout son sens dans un réseau multi-niveaux, et une ontologie fractale trouve également sa place dans la construction d'un **SCN** hiérarchique. Le **Chapitre 6** montrera comment ces idées se combinent pour donner au DSL une **cohérence** à la fois locale et globale, fortement inspirée de la théorie des systèmes complexes et de la cognition distribuée.

3.7.4. Vision Globale

Lorsqu'on aborde la construction d'un **Deep Synergy Learning (DSL)** et de son **Synergistic Connection Network (SCN)**, la notion de **représentation** occupe une place centrale. Cette représentation, qu'elle soit sub-symbolique, symbolique ou hybride, agit comme la **pierre angulaire** de toute la dynamique subséquente. Les analyses précédentes ont montré que la fonction $S(i, j)$, calculée entre deux entités \mathcal{E}_i et \mathcal{E}_j , dépend directement de la manière dont ces entités sont **modélisées**. Il s'avère ainsi que la richesse, l'adaptation ou au contraire la pauvreté d'un schéma de représentation se répercute sur la qualité des *clusters*, la stabilité de la mise à jour $\omega_{i,j}(t)$, et la capacité du DSL à s'ajuster à un environnement complexe.

A. Impact sur la qualité de la dynamique et des clusters

Les expériences de ce chapitre confirment que si la représentation initiale est imprécise ou non adaptée, les résultats d'auto-organisation peuvent se révéler fragiles ou incohérents où les $\omega_{i,j}$ se stabilisent difficilement, et l'on observe des *clusters* dont la pertinence s'avère discutable. À l'inverse, une représentation bien pensée, qu'il s'agisse d'un **embedding** sub-symbolique pertinent ou d'un **ensemble** de règles logiques cohérentes, induit une fonction de synergie plus robuste. Cette robustesse se traduit par une formation de partitions ou de groupements lisibles, et par une dynamique de stabilisation plus rapide ou plus fiable.

Pour certains types de données très bruitées (images, enregistrements audio, textes informels), un embedding neuronal s'impose souvent comme un compromis efficace, offrant une métrique continue dans \mathbb{R}^d . Dans d'autres domaines plus réglementés (domaines médical, légal, etc.), la nature symbolique d'une représentation par **règles** ou **axiomes** facilite l'explicabilité et permet un raisonnement formel, tout en résistant mal au bruit et aux variantes imprévues. Les systèmes **hybrides** puisent leur force dans la combinaison de ces deux registres où ils assurent une flexibilité face à la variabilité sub-symbolique tout en maintenant un contrôle conceptuel par la logique, ce qui rejaillit directement sur la **cohérence** des clusters et la **lisibilité** de la carte du réseau $\{\omega_{i,j}\}$.

B. Choix de modélisation pour un DSL lisible et performant

Le panorama offert par les sections précédentes rappelle qu'il **n'existe pas de solution unique** optimisant simultanément la rapidité d'exécution, l'explicabilité, la gestion du bruit et la capacité de raisonnement. Pour des applications volumineuses, les embeddings sub-symboliques dominent souvent en pratique, mais au détriment de la traçabilité formelle. Pour des tâches exigeant un raisonnement exact, les règles symboliques priment, moyennant parfois une lourdeur de calcul ou

une rigidité de mise à jour. Les solutions **hybrides** visent un équilibre, souvent gagné au prix d'une architecture plus complexe et de ressources plus importantes.

Dans une architecture DSL couvrant plusieurs **modalités** (vision, audio, textes) et intégrant une **couche** de logique (règles métier, ontologies fractales), la diversité des **outils** (fractalité, synergie n-air, HPC) aide à maintenir un certain degré de performance, même lorsque la dimension d des embeddings ou le nombre n d'entités croît fortement. Le système se révèle alors capable de faire face à des **scénarios** réalistes, où plusieurs types de données coexistent, où l'on exige à la fois la tolérance au bruit et l'aptitude à justifier des décisions.

C. Perspectives et intégration dans les chapitres suivants

Si l'on admet que la **représentation** est la fondation d'un **DSL**, il apparaît naturel que les chapitres ultérieurs (4, 5, 6) exploitent et prolongent ce travail. Le **Chapitre 4** décrira la *dynamique* de la mise à jour $\Delta\omega_{i,j}$, montrant comment la *nature* même de la fonction $S(i,j)$ influe sur la stabilité ou la vitesse de convergence. Le **Chapitre 5** s'attachera à la *structure* d'un SCN et à sa mise en œuvre pratique, en déclinant notamment les algorithmes de construction et d'exploitation. Enfin, le **Chapitre 6** abordera la *multi-échelle* et la *fractalité* dans l'organisation du DSL, qui constituent des cas d'application sophistiqués pour les représentations hybrides ou auto-similaires évoquées dans ce chapitre.

Chapitre 4 : Synergies Émergentes et Auto-Organisation

4.1. Introduction Générale

Ce **Chapitre 4** s'inscrit dans la continuité des précédents, après avoir exploré, dans le Chapitre 3, les différentes manières de **représenter** les entités (sub-symbolique, symbolique, hybride) et de **calculer** la synergie $S(i,j)$, nous allons maintenant voir **comment** ces entités et synergies **s'animent** au sein d'un **réseau** (le Synergistic Connection Network, SCN). Au cœur de ce processus se trouve la **dynamique d'auto-organisation** où les pondérations $\omega_{i,j}$ entre entités évoluent, augmentent ou diminuent, sous l'effet de la synergie $S(i,j)$ et d'un terme de régulation comme τ, η, \dots . C'est cette **dynamique** qui fait émerger, à partir de données initiales et d'un calcul de $S(i,j)$, des **clusters** cohérents ou des **patterns** inattendus — bref, la structure finale du réseau.

4.1.1. Rappel du Contexte et des Objectifs

Le **Chapitre 2** a posé les bases théoriques du **Deep Synergy Learning** en définissant les mécanismes fondamentaux par lesquels des entités interagissent via une **fonction de synergie**, notée $S(i,j)$. Ces premiers éléments ont mis en évidence la possibilité d'associer à chaque paire (i,j) un score susceptible d'indiquer le degré de similarité, de compatibilité ou de complémentarité entre les entités \mathcal{E}_i et \mathcal{E}_j . Le **Chapitre 3** s'est ensuite concentré sur la **représentation** de ces entités, il a explicité les cadres sub-symbolique et symbolique, proposé divers modes de calcul d'**embeddings** et évoqué la façon d'agréger des règles ou des concepts dans la synergie S .

À l'issue de ces deux chapitres, le **DSL** dispose ainsi d'une **mesure** $S(i,j)$ attribuant à chaque couple $(\mathcal{E}_i, \mathcal{E}_j)$ une **valeur** susceptible de capter leur affinité ou leur pertinence mutuelle. Cependant, l'essence même du Deep Synergy Learning ne réside pas dans la seule évaluation statique de ces synergies, la véritable originalité vient du **processus d'auto-organisation**, c'est-à-dire la façon dont un réseau dynamique, appelé **Synergistic Connection Network (SCN)**, fait évoluer les **pondérations** $\omega_{i,j}$ en réponse à la synergie $S(i,j)$.

Cette évolution se formalise par une **règle de mise à jour**, par exemple

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

qui instaure un **équilibre** entre la **croissance** des liaisons en cas de synergie élevée et la **décroissance** imposée par un terme $\tau \omega_{i,j}(t)$. Le but d'un tel mécanisme n'est pas simplement de calculer un score, mais de laisser le réseau s'**organiser** pour faire émerger des **clusters** ou des **motifs** particuliers, reflétant la structure latente des données ou des règles. Certains liens se renforcent alors jusqu'à atteindre une valeur stable, tandis que d'autres s'étiolent et peuvent même s'annuler si la synergie ne justifie pas leur maintien.

Cette **auto-organisation** ne mène pas toujours à un unique état final ; elle peut aboutir à une **configuration stable**, à des **oscillations** persistantes ou à plusieurs **attracteurs** coexistant selon l’initialisation et les paramètres. L’objectif de ce **Chapitre 4** est de comprendre en détail comment l’on passe de la définition de la règle de mise à jour à l’analyse de comportements variés, émergence de clusters, stabilité ou instabilité, compétition latérale, et contrôle par **recuit simulé** ou par **inhibition** plus prononcée.

Ce chapitre approfondit donc plusieurs aspects. Il décrit dans un premier temps les **formes** possibles de mise à jour (additive, multiplicative ou hybrides), montrant comment chacune traduit un certain parti pris sur la plasticité des liens. Il discute ensuite des **régimes** de convergence et des méthodes pour les caractériser. Il met en perspective le rôle central de la **synergie** S , lorsque $S(i, j)$ demeure constant, il est aisément d’analyser la convergence locale vers $\omega_{i,j}^* = S(i, j)/\tau$. Quand au contraire $S(i, j)$ varie parce que la représentation des entités se réajuste (contexte d'**apprentissage continu**), la dynamique peut échapper à la stationnarité et exiger d’autres outils d’étude.

Cette continuité théorique et algorithmique entre la **définition** de S (Chap. 3) et la **mise à jour** de ω (Chap. 4) incarne le cœur conceptuel du **DSL**. Les prochaines sections illustrent pas à pas comment, à partir d’un simple “score de similarité” entre entités, un **réseau** adaptatif finit par révéler la structure profonde d’un ensemble de données, ou se réorganiser dans le temps pour s’accorder aux changements de flux et de représentations.

4.1.2. Place et Contenu du Chapitre

Le présent chapitre s’attache à détailler la **dynamique d’auto-organisation** mise en œuvre dans un **DSL** (Deep Synergy Learning) une fois que la **synergie** $S(i, j)$ est définie pour chaque paire d’entités. Après avoir posé les bases théoriques (Chapitre 2) et présenté (Chapitre 3) la manière dont on construit les **représentations** et dont on calcule la fonction S , il s’agit maintenant d’étudier la **loi d’évolution** des pondérations $\omega_{i,j}(t)$. Cette loi est cruciale dans la mesure où elle dicte comment, à partir de synergies statiques ou variables, le **Synergistic Connection Network (SCN)** va former, détruire ou maintenir certains liens, jusqu’à aboutir à un réseau non supervisé structuré par la **coopération** ou la **compétition** des entités.

Dans la première partie (section 4.2), l’accent est mis sur les **règles de mise à jour** fondamentales. On examine notamment la forme **additive**, considérée comme la plus classique, où l’on ajoute ou retranche un incrément proportionnel à $S(i, j)$ et à la différence $\omega_{i,j}(t)$. On étudie également les variantes **multiplicatives**, dans lesquelles la croissance ou la décroissance dépend de $\omega_{i,j}(t)$ de manière proportionnelle, et les mécanismes d’**inhibition** qui introduisent une interaction compétitive entre liaisons. À chaque étape, il sera montré comment la croissance $\Delta\omega_{i,j}$ reflète la corrélation mesurée par $S(i, j)$ et la **décroissance** imposée par un terme de pénalisation $\tau \omega_{i,j}(t)$. On verra que la structure émergente résulte d’un **équilibre** entre ces forces de renforcement et de régulation.

Dans un second temps (sections 4.3 et 4.4), l’intérêt se porte sur la façon dont cette dynamique favorise la **formation spontanée de clusters**. Des entités disposant d’une synergie mutuelle élevée se trouvent amenées à consolider leurs liaisons $\omega_{i,j}$, tandis que les connexions de moindre synergie tendent à s’affaiblir ou à disparaître. On abordera la notion d’**attracteurs multiples**, qui

s'expriment par le fait que différents points fixes (ou cycles) peuvent coexister, menant à des organisations distinctes selon l'initialisation. L'analyse des **oscillations** et des régimes de **pseudo-chaos** mettra en lumière la possibilité de comportements temporels plus complexes, dont il convient de maîtriser les effets si l'on souhaite obtenir un réseau stable. Les **méthodes de contrôle** (par **recuit simulé**, **inhibition latérale** ou **saturation**) seront ainsi présentées comme des outils pour résorber ou canaliser ces fluctuations, de sorte à garantir ou à forcer une organisation plus robuste.

Dans une dernière partie (sections 4.5 à 4.7), des **exemples concrets** seront décrits pour illustrer la **mise en œuvre** de ces règles de mise à jour dans des scénarios variés. Des simulations à taille réduite montreront la constitution progressive d'un ou plusieurs **clusters**, validant expérimentalement la théorie énoncée. Par la suite, des cas plus étendus, tels qu'une population d'agents en robotique ou un jeu de données multimodales, mettront en évidence le comportement du **DSL** face à un univers plus riche, où l'on observe l'**auto-organisation** continuer d'agir même lorsque la fonction S évolue au fil du temps.

En somme, ce chapitre entend dévoiler la “**vie interne**” du **SCN** en examinant pas à pas ce qui se produit lorsqu'on laisse $\omega_{i,j}(t)$ s'ajuster en fonction de $S(i,j)$. Le propos est de comprendre comment naissent des **clusters**, des **attracteurs stables** et comment gérer les **oscillations** ou **conflits** susceptibles d'apparaître. Cet approfondissement des phénomènes d'**auto-organisation** servira de fondement pour aborder, dans les chapitres ultérieurs, l'intégration plus large du **DSL** dans des architectures complexes et dans des applications en **temps continu**.

4.2.1.1. Équation Principale

La **mise à jour** de la **pondération** $\omega_{i,j}(t)$ peut se formaliser par l'équation suivante, considérée comme une forme canonique dans la dynamique du **Deep Synergy Learning** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)].$$

Dans cette expression, la variable $\omega_{i,j}(t)$ représente la **force de lien** à l'itération t , tandis que la fonction $S(i,j)$ incarne la **synergie** instantanée entre les entités \mathcal{E}_i et \mathcal{E}_j . Le terme η fait office de **taux d'apprentissage**, déterminant la vitesse à laquelle la pondération se modifie, et $\tau \omega_{i,j}(t)$ assure une **décroissance** proportionnelle à la valeur courante de $\omega_{i,j}(t)$. L'itération s'effectue de manière **additive**, c'est-à-dire qu'à chaque pas, on ajoute un incrément linéaire dépendant de la différence $[S(i,j) - \tau \omega_{i,j}(t)]$.

La présence du facteur τ contribue à une forme de **relaxation** où, si la **synergie** $S(i,j)$ ne parvient pas à compenser la décroissance due au terme $\tau \omega_{i,j}(t)$, la pondération $\omega_{i,j}(t)$ tend à diminuer au fil des itérations. À l'inverse, lorsque $S(i,j)$ est suffisamment élevée, le terme $\eta S(i,j)$ amène $\omega_{i,j}(t)$ à croître progressivement, favorisant un **lien** de plus en plus fort entre i et j .

Du point de vue analytique, il est souvent utile d'examiner le **point d'équilibre** où la variation $\omega_{i,j}(t+1) - \omega_{i,j}(t)$ s'annule. Dans ce cas, on obtient l'égalité

$$\omega_{i,j}^* = \omega_{i,j}^* + \eta[S(i,j) - \tau \omega_{i,j}^*],$$

ce qui implique

$$S(i,j) - \tau \omega_{i,j}^* = 0 \Rightarrow \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Cette solution $\omega_{i,j}^*$ illustre la **zone de stabilisation** autour de laquelle la pondération tend à se fixer quand le schéma de mise à jour est laissé à lui-même. Ainsi, en présence d'une **forte synergie**, la valeur $\omega_{i,j}$ s'élève et se maintient à un niveau $S(i,j)/\tau$, tandis que dans le cas d'une synergie plus faible, la décroissance prend le dessus. Du point de vue **physique**, l'équation ci-dessus évoque un **gradient** ou une forme de mouvement orienté vers un attracteur local, traduisant la logique de renforcement ou d'atténuation des liens selon l'intensité de la coopération sous-jacente.

4.2.1.2. Sens Interprétatif : Lien avec la “Descente d’Énergie” et Convergence vers $\omega^* = \frac{S}{\tau}$

Le **Deep Synergy Learning** peut être éclairé par une analogie physique et un principe de **descente d’énergie**. Dans le chapitre 2.4.3, il a été proposé de conceptualiser le réseau comme un système qui tend à minimiser une **pseudo-énergie** ou “fonction potentielle”, notée $\mathcal{J}(\Omega)$. Une formulation simplifiée de cette énergie prend généralement la forme

$$\mathcal{J}(\Omega) = - \sum_{(i,j)} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{(i,j)} (\omega_{i,j})^2 + \dots$$

Dans cette expression, le terme $-\omega_{i,j} S(i,j)$ favorise l'augmentation de la **pondération** $\omega_{i,j}$ lorsque la **synergie** $S(i,j)$ est élevée, tandis que la partie $\tau/2 (\omega_{i,j})^2$ impose une **pénalisation** qui s'accroît avec la magnitude du lien. La **règle additive** décrite précédemment,

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

peut alors s'interpréter comme un schéma de **descente** (ou de “quasi-descente”) dans le gradient négatif de \mathcal{J} . La composante $S(i,j)$ joue un rôle de “force” qui pousse $\omega_{i,j}$ à croître, tandis que le terme $\tau \omega_{i,j}$ sert de “frein” ou de “relaxation” vers des valeurs plus modestes.

Lorsque la **synergie** $S(i,j)$ reste **constante** au cours du temps, il est possible d'étudier la **convergence** de la dynamique en cherchant le point fixe $\omega_{i,j}^*$ pour lequel $\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega_{i,j}^*$. En imposant cette condition stationnaire, on obtient

$$\omega_{i,j}^* = \omega_{i,j}^* + \eta [S(i,j) - \tau \omega_{i,j}^*] \Rightarrow S(i,j) = \tau \omega_{i,j}^* \Rightarrow \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Ce résultat indique que la **valeur d'équilibre** $\omega_{i,j}^*$ est directement proportionnelle à la **synergie** $S(i,j)$ et inversement proportionnelle au paramètre τ , lequel représente la “raideur” ou l'intensité du freinage imposé à la croissance du lien. Aussi longtemps que η (taux d'apprentissage) est choisi de façon à éviter les oscillations excessives, on peut montrer que $\omega_{i,j}(t)$ se rapproche de $\omega_{i,j}^*$, ce qui confère à cet équilibre le statut de **point attracteur local**.

Sur le plan **énergétique**, cette situation correspond à un **minimum** de la pseudo-énergie \mathcal{J} dans un scénario simple où la **synergie** ne dépend pas elle-même de $\omega_{i,j}$. Au sein du SCN, chaque lien se “met en conformité” avec la valeur $S(i,j)/\tau$, de sorte que les liaisons **fortes** correspondent

précisément aux paires (i, j) ayant une **synergie** élevée, et les liaisons **faibles** sont observées là où $S(i, j)$ demeure basse. Cette mise à niveau progressive peut être considérée comme une “relaxation” du réseau, dans laquelle les forces de renforcement (la **synergie**) et de décroissance (le terme $\tau \omega_{i,j}$) parviennent à un **compromis**.

Dans le cas plus général où $S(i, j)$ varie au fil des itérations (parce que la représentation des entités se modifie ou que des règles symboliques sont adaptées en continu), l'équilibre local $\omega_{i,j}^*$ se déplace simultanément. La **règle additive** agit alors comme un processus d'**auto-organisation** qui s'efforce de suivre ces déplacements, sans nécessairement parvenir à une véritable convergence statique. Cette faculté d'adaptation souligne la souplesse du **DSL** dans des environnements évolutifs, tout en conservant l'idée d'une “descendance d'énergie” vers des configurations où la **synergie** et la **décroissance** se neutralisent mutuellement.

4.2.1.3. Cas Stationnaire vs. Cas Dynamique : $S(i, j)$ constant ou recalculé à chaque itération

La **règle additive** introduite précédemment repose sur une mise à jour des pondérations $\omega_{i,j}$ proportionnelle à la **synergie** $S(i, j)$ et à un terme de **relaxation** $\tau \omega_{i,j}(t)$. Cette formulation se décline en deux scénarios où un **cas stationnaire** correspond à une synergie $S(i, j)$ qui demeure fixe dans le temps, tandis qu'un **cas dynamique** implique un recalcul de $S(i, j)$ à chaque itération, par exemple lorsque les représentations ou les règles sous-jacentes se modifient.

Dans le **cas stationnaire**, on suppose que la **synergie** $S(i, j)$ ne subit aucune variation au fil du temps. Les entités \mathcal{E}_i et \mathcal{E}_j conservent alors leurs caractéristiques ou leurs représentations sans être affectées par un nouvel apprentissage ou par l'introduction de changements contextuels. Le système $\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)]$ se comporte dès lors comme une **descendance d'énergie** dirigée vers un point fixe, vérifiant $\omega_{i,j}^* = S(i, j)/\tau$. Lorsque la valeur $\omega_{i,j}(t)$ atteint ce rapport, il n'existe plus d'incitation à croître ni à décroître, dans la mesure où la composante $S(i, j)$ équilibre parfaitement la pénalisation $\tau \omega_{i,j}(t)$. D'un point de vue analytique, ce régime simplifie grandement l'étude de la **convergence** où le système tend à se stabiliser, permettant ainsi de démontrer, sous certaines conditions, notamment sur le pas d'apprentissage η , une convergence locale vers un attracteur. Un **DSL** exploité dans ce cadre stationnaire permet de clarifier l'effet de la synergie sur la formation de liens forts, tout en évitant les complications liées à la variation de $S(i, j)$.

Dans un **cas dynamique**, la **synergie** $S(i, j)$ subit un recalcul, potentiellement à chaque itération, de sorte qu'elle devient dépendante des modifications intervenues dans le réseau. Il se peut que les représentations vectorielles des entités $\mathbf{r}(i)$ et $\mathbf{r}(j)$ soient elles-mêmes ajustées par un processus d'apprentissage sous-jacent, ou que des blocs de règles symboliques évoluent (ajout/suppression d'axiomes, reparamétrage logique). Dans ces circonstances, $S(i, j)$ prend la forme d'une fonction $S_t(i, j)$ explicitement liée au temps t . La mise à jour

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta [S_t(i, j) - \tau \omega_{i,j}(t)]$$

opère alors sur un **paysage** en mouvement. La valeur de $\omega_{i,j}(t)$ doit sans cesse s'adapter, car l'équilibre instantané $S_t(i, j)/\tau$ se déplace à mesure que $S_t(i, j)$ évolue. Un **DSL** de ce type décrit plus fidèlement un système soumis à des variations externes ou internes (arrivée de nouvelles

entités, apprentissage continu, ajustement de règles symboliques). Il n'est pas toujours garanti que l'on aboutisse à un point fixe au sens strict ; la dynamique peut osciller, changer de régime ou poursuivre un déplacement lent suivant la trajectoire imposée par $S_t(i,j)$.

En conclusion, la **formulation additive** se prête tout autant au **cas stationnaire** qu'à un **cas dynamique** plus réaliste. Lorsque $S(i,j)$ reste constant, l'analyse du point fixe $\omega_{i,j}^* = S(i,j)/\tau$ permet de mettre en évidence un comportement stable et interprétable. Si, au contraire, la **synergie** varie, la boucle de mise à jour se conçoit comme un **système adaptatif** en mouvement constant, dont la convergence peut se révéler délicate à établir. Les sections suivantes approfondissent ces notions en examinant la stabilisation, la formation de **clusters**, et les phénomènes oscillatoires dans des configurations plus complètes du **Deep Synergy Learning**.

4.2. Règles de Mise à Jour Fondamentales

Après l'introduction générale (4.1) qui présente la dynamique d'auto-organisation comme le moteur principal du **DSL** (Deep Synergy Learning), nous abordons maintenant les **règles** mathématiques qui permettent de **faire évoluer** les pondérations $\omega_{i,j}$ dans le **SCN** (Synergistic Connection Network). Ces règles traduisent la manière dont un lien $\omega_{i,j}$ se **renforce** ou se **détériore** en réponse à la **synergie** $S(i,j)$, et elles constituent le **noyau** de l'auto-organisation.

4.2.1. Formulation Additive Classique

La première formulation, très fréquente dans la littérature et déjà mentionnée dans les chapitres précédents (2.3, 2.4.3), est dite **additive** où l'on met à jour $\omega_{i,j}(t)$ en lui **ajoutant** un terme proportionnel à la différence $[S(i,j) - \tau \omega_{i,j}(t)]$.

4.2.1.1. Équation Principale

La **mise à jour** de la **pondération** $\omega_{i,j}(t)$ peut se formaliser par l'équation suivante, considérée comme une forme canonique dans la dynamique du **Deep Synergy Learning** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)].$$

Dans cette expression, la variable $\omega_{i,j}(t)$ représente la **force de lien** à l'itération t , tandis que la fonction $S(i,j)$ incarne la **synergie** instantanée entre les entités \mathcal{E}_i et \mathcal{E}_j . Le terme η fait office de **taux d'apprentissage**, déterminant la vitesse à laquelle la pondération se modifie, et $\tau \omega_{i,j}(t)$ assure une **décroissance** proportionnelle à la valeur courante de $\omega_{i,j}(t)$. L'itération s'effectue de manière **additive**, c'est-à-dire qu'à chaque pas, on ajoute un incrément linéaire dépendant de la différence $[S(i,j) - \tau \omega_{i,j}(t)]$.

La présence du facteur τ contribue à une forme de **relaxation** où, si la **synergie** $S(i,j)$ ne parvient pas à compenser la décroissance due au terme $\tau \omega_{i,j}(t)$, la pondération $\omega_{i,j}(t)$ tend à diminuer progressivement au fil des itérations. À l'inverse, lorsque $S(i,j)$ est suffisamment élevée, le terme $\eta S(i,j)$ amène $\omega_{i,j}(t)$ à croître progressivement, favorisant un **lien** de plus en plus fort entre i et j .

Du point de vue analytique, il est souvent utile d'examiner le **point d'équilibre** où la variation $\omega_{i,j}(t+1) - \omega_{i,j}(t)$ s'annule. Dans ce cas, on obtient l'égalité

$$\omega_{i,j}^* = \omega_{i,j}^* + \eta [S(i,j) - \tau \omega_{i,j}^*],$$

ce qui implique

$$S(i,j) - \tau \omega_{i,j}^* = 0 \Rightarrow \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Cette solution $\omega_{i,j}^*$ illustre la **zone de stabilisation** autour de laquelle la pondération tend à se fixer quand le schéma de mise à jour est laissé à lui-même. Ainsi, en présence d'une **forte synergie**, la valeur $\omega_{i,j}$ s'élève et se maintient à un niveau $S(i,j)/\tau$, tandis que dans le cas d'une synergie plus faible, la décroissance prend le dessus. Du point de vue **physique**, l'équation ci-dessus évoque un **gradient** ou une forme de mouvement orienté vers un attracteur local, traduisant la logique de renforcement ou d'atténuation des liens selon l'intensité de la coopération sous-jacente.

4.2.1.2. Sens Interprétatif : Lien avec la “Descente d’Énergie” et Convergence vers $\omega^* = S/\tau$

Il est souvent utile d'analyser la **règle additive** comme une forme de **descente de gradient** dans un **paysage d'énergie**. Au chapitre 2.4.3, il a été proposé de considérer une **fonction d'énergie** $J(\Omega)$ dont l'un des termes dominants peut s'écrire sous la forme $-\sum_{i,j} \omega_{i,j} S(i,j)$. Dans un cadre simplifié, on peut poser

$$J(\Omega) = - \sum_{i,j} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{i,j} (\omega_{i,j})^2 + \dots$$

et vérifier que la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$$

correspond, dans ses grandes lignes, à la **descente** du gradient de J pour peu que $S(i,j)$ ne dépende pas de $\omega_{i,j}$.

La présence du **terme** $\tau \omega_{i,j}(t)$ peut se lire comme un “coût” qui **régularise la pondération** $\omega_{i,j}$. Plus la valeur de $\omega_{i,j}(t)$ est élevée, plus le produit $\tau \omega_{i,j}(t)$ vient la ramener à la baisse, sauf si la **synergie** $S(i,j)$ est suffisamment grande pour compenser ce mouvement. Dans un scenario stationnaire, où $S(i,j)$ est constante au cours du temps, la convergence locale de $\omega_{i,j}(t)$ vers un point fixe peut être étudiée en imposant $\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega^*$. Il en résulte l'équation

$$\omega^* = \omega^* + \eta [S(i,j) - \tau \omega^*],$$

impliquant $S(i,j) = \tau \omega^*$. Autrement dit, on obtient

$$\omega^* = \frac{S(i,j)}{\tau}.$$

Cette **valeur** ω^* peut être considérée comme la **solution d'équilibre** où, si la **règle additive** est appliquée de façon itérative et que $S(i,j)$ reste fixe, la pondération $\omega_{i,j}(t)$ évolue jusqu'à se stabiliser approximativement autour de $S(i,j)/\tau$. La signification profonde de ce résultat est qu'il y a un **équilibre** entre le **gain** (lié à $S(i,j)$) et la **pénalisation** quadratique (associée à $\tau \omega_{i,j}^2$), comme si le système « descendait » la fonction d'énergie J et s'arrêtait dans une vallée correspondant à un minimum local.

Dans un contexte plus général, où la **synergie** $S(i,j)$ varie au fil du temps, il n'existe pas forcément de convergence définitive vers une valeur fixe ω^* . Il est toutefois possible de constater que la **pondération** $\omega_{i,j}$ “suit” la synergie à travers une suite de régimes quasi-stationnaires, ou qu'elle

oscille si les changements de $S(i,j)$ sont trop rapides ou si l'on choisit un **taux d'apprentissage** η trop grand. Le facteur τ , agissant comme un **terme de régularisation**, garantit néanmoins que $\omega_{i,j}(t)$ ne croîtra pas indéfiniment en l'absence de feedback négatif ; il limite donc la possibilité d'un emballlement des poids et maintient le réseau dans un régime stable ou faiblement oscillant.

4.2.1.3. Cas Stationnaire vs. Cas Dynamique : $S(i,j)$ constant ou recalculé à chaque itération

La **règle additive** introduite précédemment repose sur une mise à jour des pondérations $\omega_{i,j}$ proportionnelle à la **synergie** $S(i,j)$ et à un terme de **relaxation** $\tau \omega_{i,j}(t)$. Cette formulation se décline en deux scénarios où un **cas stationnaire** correspond à une synergie $S(i,j)$ qui demeure fixe dans le temps, tandis qu'un **cas dynamique** implique un recalcul de $S(i,j)$ à chaque itération, notamment lorsque les représentations ou les règles sous-jacentes se modifient.

Dans le **cas stationnaire**, on suppose que la **synergie** $S(i,j)$ ne subit aucune variation au fil du temps. Les entités \mathcal{E}_i et \mathcal{E}_j conservent alors leurs caractéristiques ou leurs représentations sans être affectées par un nouvel apprentissage ou par l'introduction de changements contextuels. Le système $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$ se comporte dès lors comme une **descente d'énergie** dirigée vers un point fixe, vérifiant $\omega_{i,j}^* = S(i,j)/\tau$. Lorsque la valeur $\omega_{i,j}(t)$ atteint ce rapport, il n'existe plus d'incitation à croître ni à décroître, dans la mesure où la composante $S(i,j)$ équilibre parfaitement la pénalisation $\tau \omega_{i,j}(t)$. D'un point de vue analytique, ce régime simplifie grandement l'étude de la **convergence** où le système tend à se stabiliser, permettant ainsi de démontrer, sous certaines conditions, notamment sur le pas d'apprentissage η , une convergence locale vers un attracteur. Un **DSL** exploité dans ce cadre stationnaire permet de clarifier l'effet de la synergie sur la formation de liens forts, tout en évitant les complications liées à la variation de $S(i,j)$.

Dans un **cas dynamique**, la **synergie** $S(i,j)$ subit un recalcul, potentiellement à chaque itération, de sorte qu'elle devient dépendante des modifications intervenues dans le réseau. Il se peut que les représentations vectorielles des entités $\mathbf{r}(i)$ et $\mathbf{r}(j)$ soient elles-mêmes ajustées par un processus d'apprentissage sous-jacent, ou que des blocs de règles symboliques évoluent (ajout/suppression d'axiomes, reparamétrage logique). Dans ces circonstances, $S(i,j)$ prend la forme d'une fonction $S_t(i,j)$ explicitement liée au temps t . La mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_t(i,j) - \tau \omega_{i,j}(t)]$$

opère alors sur un **paysage** en mouvement. La valeur de $\omega_{i,j}(t)$ doit sans cesse s'adapter, car l'équilibre instantané $S_t(i,j)/\tau$ se déplace à mesure que $S_t(i,j)$ évolue. Un **DSL** de ce type décrit plus fidèlement un système soumis à des variations externes ou internes (arrivée de nouvelles entités, apprentissage continu, ajustement de règles symboliques). Il n'est pas toujours garanti que l'on aboutisse à un point fixe au sens strict ; la dynamique peut osciller, changer de régime ou poursuivre un déplacement lent suivant la trajectoire imposée par $S_t(i,j)$.

En conclusion, la **formulation additive** se prête tout autant au **cas stationnaire** qu'à un **cas dynamique** plus réaliste. Lorsque $S(i,j)$ reste constant, l'analyse du point fixe $\omega_{i,j}^* = S(i,j)/\tau$ permet de mettre en évidence un comportement stable et interprétable. Si, au contraire, la **synergie** varie, la boucle de mise à jour se conçoit comme un **système adaptatif** en mouvement constant,

dont la convergence peut se révéler délicate à établir. Les sections suivantes approfondissent ces notions en examinant la stabilisation, la formation de **clusters**, et les phénomènes oscillatoires dans des configurations plus complètes du **Deep Synergy Learning**.

4.2.1. Formulation Additive Classique

La première formulation, très fréquente dans la littérature et déjà mentionnée en chapitres précédents (2.3, 2.4.3), est dite **additive** : on met à jour $\omega_{i,j}(t)$ en lui **ajoutant** un terme proportionnel à la différence $[S(i,j) - \tau \omega_{i,j}(t)]$.

4.2.1.1. Équation Principale

Dans le cadre d'un **Deep Synergy Learning (DSL)**, l'évolution de la **pondération** $\omega_{i,j}$ entre deux entités \mathcal{E}_i et \mathcal{E}_j se fait de manière itérative et s'exprime par une équation de mise à jour additive qui repose sur la synergie mesurée entre ces entités et un terme de décroissance régulateur. La forme canonique de cette équation est donnée par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où chaque symbole revêt une signification particulière. Dans cette équation, $\omega_{i,j}(t)$ représente la **pondération** (ou force de lien) entre les entités \mathcal{E}_i et \mathcal{E}_j à l'itération t . Le terme $S(i,j)$ désigne la **synergie instantanée ou statique** entre ces deux entités, telle que définie dans le chapitre 3, et qui est souvent calculée à partir de mesures de similarité (par exemple, la similarité cosinus ou une distance transformée par un noyau gaussien). Le paramètre η correspond au **taux d'apprentissage**, un facteur qui contrôle la vitesse avec laquelle la pondération est mise à jour, tandis que le produit $\tau \omega_{i,j}(t)$ représente un **terme de décroissance** ou de relaxation, modulé par le paramètre τ qui détermine l'intensité de cette décroissance. Ainsi, un τ élevé aura pour effet de réduire plus fortement $\omega_{i,j}(t)$ lorsque $S(i,j)$ n'est pas suffisamment grand pour compenser cette décroissance.

Mathématiquement, l'évolution de $\omega_{i,j}$ est dite **additive** dans la mesure où, à chaque itération, on ajoute (ou l'on soustrait) un delta proportionnel à la différence $S(i,j) - \tau \omega_{i,j}(t)$. Cette formulation rappelle un mécanisme de **gradient descent**, où la pondération se déplace progressivement vers un point d'équilibre. En effet, en fixant la mise à jour à zéro dans l'état stationnaire, on obtient l'équation

$$\omega_{i,j}(t+1) \approx \omega_{i,j}(t) \Rightarrow S(i,j) - \tau \omega_{i,j}(t) \approx 0,$$

ce qui implique que, à l'équilibre, la pondération satisfait approximativement

$$\omega_{i,j} \approx \frac{S(i,j)}{\tau}.$$

On peut ainsi interpréter cette équation comme une **force attractive** qui tend à renforcer le lien entre deux entités lorsque la synergie est forte, tout en assurant une décroissance progressive du lien lorsque la synergie est insuffisante. Si $\omega_{i,j}(t)$ est initialement faible et que $S(i,j)$ est grand –

c'est-à-dire si la synergie mesurée est élevée – la mise à jour incrémentale va entraîner une augmentation itérative de $\omega_{i,j}$, conduisant le système vers une zone de stabilisation où la valeur de la pondération converge vers l'équilibre $\omega_{i,j} \approx \frac{S(i,j)}{\tau}$. À l'inverse, si $\omega_{i,j}(t)$ est élevé mais que $S(i,j)$ demeure faible, le terme négatif $-\eta \tau \omega_{i,j}(t)$ prédomine, provoquant une décroissance progressive de la pondération.

Cette règle de mise à jour est souvent considérée comme la plus simple et la plus intuitive pour modéliser la dynamique des liens dans un DSL. Elle offre une représentation claire de la manière dont un lien évolue en fonction de la synergie instantanée et de la décroissance proportionnelle à son état courant, permettant ainsi de capturer à la fois l'aspect d'**apprentissage** local et l'effet de **relaxation** qui prévient une croissance incontrôlée des pondérations. En résumé, l'équation principale

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$$

se présente comme un modèle simple et efficace pour l'auto-organisation d'un réseau de connexions, où la pondération entre deux entités évolue de manière additive en réponse à une synergie mesurée, convergeant vers une valeur d'équilibre définie par la relation $\omega_{i,j} \approx \frac{S(i,j)}{\tau}$.

4.2.1.2. Sens Interprétatif : lien avec la “descente d'énergie” (2.4.3), convergence vers $\omega^* = \frac{S}{\tau}$

Dans le cadre du **Deep Synergy Learning (DSL)**, l'équation de mise à jour des pondérations est souvent interprétée sous l'angle d'une **descente d'énergie**. Cette interprétation repose sur l'idée qu'une fonction d'énergie, notée $\mathcal{J}(\Omega)$, définie sur l'ensemble des pondérations Ω du réseau, diminue progressivement au cours des itérations d'apprentissage. Le chapitre 2.4.3 introduisait déjà le concept d'une fonction potentielle, par exemple sous la forme

$$\mathcal{J}(\Omega) = - \sum_{i,j} \omega_{i,j} S(i,j) + \dots,$$

où $S(i,j)$ représente la **synergie** entre les entités \mathcal{E}_i et \mathcal{E}_j . La règle de mise à jour canonique du DSL s'exprime par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $\eta > 0$ est le **taux d'apprentissage** et $\tau > 0$ représente le **coefficent de décroissance** qui agit comme un terme de régularisation. On peut considérer cette équation comme une descente de gradient appliquée à la fonction d'énergie \mathcal{J} dans un cas simplifié où $S(i,j)$ est indépendant de $\omega_{i,j}$. L'analogie avec la descente d'énergie se fait apparaître en remarquant que le terme $[S(i,j) - \tau \omega_{i,j}(t)]$ joue le rôle du gradient partiel de la fonction d'énergie par rapport à $\omega_{i,j}$, ce qui conduit, en cas de convergence, à un équilibre stationnaire.

Pour étudier cet équilibre, nous considérons la situation stationnaire où la mise à jour cesse, c'est-à-dire lorsque

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega^*.$$

En substituant cette condition dans l'équation de mise à jour, nous obtenons

$$\omega^* = \omega^* + \eta[S(i,j) - \tau \omega^*],$$

ce qui implique nécessairement que

$$S(i,j) - \tau \omega^* = 0.$$

Il en découle alors que la pondération stationnaire satisfait

$$\omega^* = \frac{S(i,j)}{\tau}.$$

Cette relation d'équilibre est particulièrement interprétable dans le sens où elle illustre comment la **force** du lien entre deux entités se stabilise à une valeur qui est directement proportionnelle à leur synergie $S(i,j)$ et inversement proportionnelle à la décroissance imposée par τ . On peut comprendre ce mécanisme en analogie avec un système physique où une particule se déplace dans un champ de potentiel et atteint un minimum d'énergie lorsque la force nette agissant sur elle devient nulle. Ici, le terme $\tau \omega_{i,j}(t)$ agit comme une force de **régularisation** ou un « coût » associé au maintien d'un lien fort, ce qui empêche les pondérations de croître indéfiniment en l'absence d'une synergie suffisante.

Dans un scénario idéal où $S(i,j)$ est constant dans le temps, la mise à jour additive converge de manière monotone vers $\omega^* = \frac{S(i,j)}{\tau}$. Toutefois, en pratique, $S(i,j)$ peut varier d'une itération à l'autre, ce qui rend la convergence plus complexe et peut empêcher l'atteinte d'un équilibre strict. Néanmoins, l'idée centrale demeure que la pondération $\omega_{i,j}$ tend à se "caler" sur une valeur qui reflète la force moyenne de la synergie entre les entités, assurant ainsi une **auto-organisation** harmonieuse du réseau.

Le terme de décroissance $\tau \omega_{i,j}(t)$ joue également un rôle essentiel en agissant comme un mécanisme d'**anti-explosion**. Sans ce terme, si $S(i,j)$ était constamment élevé, la pondération pourrait croître de façon indéfinie, ce qui serait incompatible avec la stabilité du système. La présence de ce terme garantit que, même en cas de synergie élevée, la pondération est ramenée à des valeurs réalistes, modélisant ainsi un compromis entre la **coopération** entre entités et le **coût** de maintenir des liens trop forts.

En résumé, la règle additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

peut être interprétée comme une descente de gradient sur une fonction d'énergie $\mathcal{J}(\Omega)$, où la convergence vers l'équilibre stationnaire $\omega^* = \frac{S(i,j)}{\tau}$ illustre la manière dont la synergie entre deux entités est équilibrée par la décroissance. Cette interprétation confère à la mise à jour un sens interprétatif profond, reliant le comportement numérique du DSL à des principes physiques tels que la minimisation de l'énergie, et offre ainsi une explication intuitive de la **stabilité** des liens dans le SCN.

4.2.1.3. Cas Stationnaire vs. Cas Dynamique : $S(i, j)$ constant ou recalculé à chaque itération

Dans l'analyse d'un **Deep Synergy Learning (DSL)**, l'évolution des **pondérations** $\omega_{i,j}$ repose sur la synergie $S(i, j)$ entre les entités, laquelle peut être considérée soit comme une valeur **constante** au cours du temps, soit comme une grandeur **dynamique** recalculée à chaque itération. Ces deux cas présentent des comportements distincts qui influencent la convergence du système et la stabilité de l'**auto-organisation** dans le **Synergistic Connection Network (SCN)**.

Lorsqu'on adopte le **cas stationnaire**, on suppose que $S(i, j)$ demeure fixe. Dans ce contexte, les représentations des entités, qu'elles soient sub-symboliques ou symboliques, ne subissent pas de modifications significatives d'une itération à l'autre et aucune mise à jour contextuelle n'est introduite. La règle de mise à jour, qui s'exprime sous la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

se traduit alors par une dynamique qui converge vers un **point fixe** ou un attracteur stable. Dans l'état stationnaire elle convergence vers $\omega^* = \frac{S(i, j)}{\tau}$. Dans ce cas, l'ensemble du réseau se stabilise et les transitions entre états restent fixes, ce qui facilite l'analyse théorique de la convergence et permet de servir de référence pour comparer des systèmes dans des environnements plus dynamiques.

Dans le **cas dynamique**, en revanche, la synergie $S(i, j)$ n'est pas considérée comme constante, mais est recalculée à chaque itération. Cette situation se produit lorsque les représentations des entités évoluent au fil du temps en raison d'un apprentissage continu ou de modifications contextuelles, telles que des mises à jour des embeddings ou des ajustements dans les blocs symboliques.

En effet, lorsque le DSL intègre un **apprentissage adaptatif**, le score $S(i, j)$ devient une fonction temporelle, que l'on peut noter $S(i, j, t)$. La règle de mise à jour s'écrit alors

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j, t) - \tau \omega_{i,j}(t)],$$

et il apparaît un couplage itératif dans lequel la pondération $\omega_{i,j}$ s'ajuste en même temps que $S(i, j)$ évolue.

En pratique, cette interaction rend le système plus complexe, car les mises à jour des représentations sub-symboliques (par exemple, via le fine-tuning ou la réorganisation des données) entraînent des variations dans $S(i, j, t)$ qui peuvent perturber la convergence du réseau.

Le système se comporte alors comme un **système adaptatif** où aucune convergence stationnaire absolue n'est garantie, et l'auto-organisation devient un processus continu dans lequel la structure du réseau est sans cesse réévaluée et reconfigurée. Un tel processus peut engendrer des oscillations ou des réajustements brusques dans les pondérations, surtout si les variations de $S(i, j, t)$ sont significatives d'une itération à l'autre.

Pour pallier ces effets, il est souvent nécessaire d'introduire des mécanismes de **lissage** ou de **verrouillage partiel**, afin que la transition entre différentes valeurs de $S(i,j,t)$ soit plus progressive et que l'ensemble du système reste stable malgré des mises à jour fréquentes.

4.2.2. Variantes Multiplicatives, Inhibition, etc.

Bien que la **formulation additive** (4.2.1) soit souvent considérée la plus “classique” pour un **DSL** (Deep Synergy Learning), il existe d'autres **règles** de mise à jour des pondérations $\omega_{i,j}$. Certaines de ces variantes cherchent à introduire plus de **dynamisme** (croissance multiplicative), d'autres à intégrer de l'**inhibition** compétitive, ou encore à imposer des **mécanismes** de saturation pour prévenir une montée incontrôlée de certains liens.

4.2.2.1. Multiplicative : $\omega_{i,j}(t+1) = \omega_{i,j}(t) [1 + \eta (\dots)]$

Dans cette variante, plutôt que d'ajouter un delta à $\omega_{i,j}(t)$ (comme dans la règle additive), on multiplie $\omega_{i,j}(t)$ par un **facteur** dépendant de la synergie $S(i,j)$ et de la décroissance τ . Autrement dit :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) \left[1 + \eta \left(\alpha S(i,j) - \beta \tau \omega_{i,j}(t) \right) \right],$$

ou une forme voisine (il existe plusieurs formulations possibles). L'idée générale est que la **mise à jour** se fait en **proportion** de $\omega_{i,j}(t)$, plutôt que par un **delta** absolu.

Mise à jour multiplicative : motivations, avantages et risques

Dans certains scénarios de Deep Synergy Learning (DSL), on abandonne la règle additive traditionnelle (sous la forme $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \Delta$) au profit d'une **mise à jour multiplicative** permettant de modéliser une **croissance proportionnelle** ou quasi exponentielle. Concrètement, au lieu d'ajouter un terme Δ , on applique un **facteur** $[1 + \Delta]$, qui multiplie la valeur actuelle de $\omega_{i,j}(t)$. Ainsi, si la pondération $\omega_{i,j}(t)$ est déjà élevée, une légère variation de Δ peut se traduire par une forte augmentation, ce qui peut accélérer la structuration du réseau en mettant rapidement en évidence certains **liens dominants**.

Sur le plan analytique, on peut illustrer cette approche par la formule :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) [1 + \eta S(i,j) - \eta \tau \omega_{i,j}(t)].$$

Dans cette équation, η représente un taux d'apprentissage, τ un terme de régulation ou d'oubli, et $S(i,j)$ la synergie calculée entre les entités \mathcal{E}_i et \mathcal{E}_j . Si $\omega_{i,j}(t)$ est déjà non négligeable, et que la partie $\eta S(i,j)$ dépasse $\eta \tau \omega_{i,j}(t)$, le produit dans la parenthèse demeure supérieur à 1, ce qui renforce massivement la pondération. C'est en cela que l'on parle de **dynamique non linéaire** : plus un lien est fort, plus il est susceptible d'augmenter vite, de sorte que l'on observe fréquemment un phénomène d'**auto-amplification**.

Cette **caractéristique** devient particulièrement pertinente lorsqu'il s'agit de **modéliser** ou d'**encourager une coopération exponentielle**. Lorsque deux **entités** collaborent déjà étroitement, leur **liaison** s'intensifie encore, conduisant à la formation de **clusters en “tout ou rien”** où certains liens **s'amplifient fortement**, tandis que d'autres **s'effondrent**.

Ce schéma **multiplicatif** reflète également divers comportements biologiques ou écosystémiques, où la vitesse de croissance dépend de la “valeur” présente (lois de populations, mécanismes de reproduction, etc.). En revanche, il comporte un **risque de divergence**. Si $\omega_{i,j}(t)$ devient trop important et n'est pas freiné par le terme $\eta \tau \omega_{i,j}(t)$, la pondération peut croître sans limite, perturbant profondément la structure du réseau.

Pour **éviter** ce problème, on introduit souvent des dispositifs de **saturation** (voir section 4.2.2.3) limitant la valeur maximale de $\omega_{i,j}$, ou des mécanismes d'**inhibition compétitive** (section 4.2.2.2) afin d'empêcher qu'un petit nombre de liens n'accapare l'essentiel du “poids” synergique.

En comparaison, la **règle additive** classique conduit plutôt à un point fixe $\omega^* = S(i,j)/\tau$ en l'absence d'autres rétroactions, et autorise un contrôle plus fin sur la progression des pondérations. La version multiplicative, elle, **accélère** la différenciation des liens, mais exige une **vigilance accrue** quant aux risques d'**explosion**, d'**instabilité** et de forte **sensibilité** aux variations de la synergie. En somme, choisir un schéma multiplicatif revient à opter pour un modèle dans lequel la mise en valeur rapide de certaines liaisons est un avantage, tout en s'assurant qu'un encadrement (clipping, inhibition) est bien en place pour maintenir la cohérence globale du réseau.

4.2.2.2. Inhibition Compétitive : ajout d'un terme $-\gamma \sum_{k \neq j} \omega_{i,k}(t)$ ou autre forme

Dans certains systèmes adaptatifs (biologiques, cognitifs, ou plus généralement en **self-organisation**), l'**inhibition compétitive** joue un rôle déterminant pour **réguler** la croissance simultanée de liens. L'idée est de **limiter** la progression de plusieurs connexions parallèles, incitant ainsi certaines à **dominer** au détriment d'autres. Cela évite un “tout ou rien” généralisé, ou la situation où trop de liens forts coexistent et saturent la dynamique.

Dans un **DSL**, on peut ajouter au terme de mise à jour un **facteur d'inhibition** qui pénalise $\omega_{i,j}$ lorsque, pour la même entité \mathcal{E}_i , d'autres pondérations $\omega_{i,k}$ (avec $k \neq j$) sont élevées.

A. Principe de l'inhibition compétitive

Une règle additive (ou multiplicative) peut être agrémentée d'un **terme d'inhibition** de la forme :

$$-\gamma \sum_{k \neq j} \omega_{i,k}(t),$$

où $\gamma > 0$ est un coefficient d'inhibition, et la somme $\sum_{k \neq j} \omega_{i,k}(t)$ représente la “masse” totale des liens que \mathcal{E}_i entretient avec d'autres entités $k \neq j$.

En pratique, ce terme vient **s'ajouter** (avec un signe négatif) à la mise à jour de $\omega_{i,j}$. Plus \mathcal{E}_i possède des liens forts avec d'autres \mathcal{E}_k , plus la **progression** de $\omega_{i,j}$ s'en trouve **limitée**.

L'**idée intuitive** derrière ce mécanisme repose sur une forme de **compétition** : si une entité \mathcal{E}_i investit déjà fortement dans certains liens $\omega_{i,k}$, ses “ressources” disponibles pour renforcer de nouveaux liens $\omega_{i,j}$ sont de facto limitées. D'un point de vue **biologique**, ce principe évoque un phénomène de “**winner-take-all**” partiel ou d’“**inhibition latérale**”, couramment décrit dans les modèles neuronaux, où la somme d'activité globale d'un neurone (c'est-à-dire la somme de ses liens) restreint sa capacité à s'activer simultanément vers d'autres connexions.

Exemple

Supposez une règle additive standard :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t).$$

Quand $\sum_{k \neq j} \omega_{i,k}(t)$ est grand, cela **déourage** l'augmentation de $\omega_{i,j}$, forçant \mathcal{E}_i à “faire un choix” parmi plusieurs liaisons concurrentes.

B. Avantages de l'inhibition compétitive

L'un des premiers bénéfices consiste à éviter toute “dispersion” : sans mécanisme d'inhibition, une entité \mathcal{E}_i risquerait de créer de nombreux liens modérément forts avec un large éventail d'autres entités.

En introduisant une compétition, on **focalise** l'attention sur quelques liaisons dominantes, tout en réduisant l'intensité des autres ; il en résulte la formation de **clusters** plus nettement délimités.

Par ailleurs, l'inhibition compétitive aide à **contrôler la croissance simultanée** des connexions : même si la synergie $S(i,j)$ se révèle positive pour plusieurs entités, le réseau ne permet pas à toutes ces liaisons de croître en parallèle, introduisant une forme de **sélection** où seules les connexions les plus pertinentes se renforcent réellement.

Enfin, cette approche s'appuie sur des principes **biomimétiques**, rappelant l'**inhibition latérale** observée dans le cerveau, où un neurone en forte activité limite celle de ses voisins.

Dans un **DSL**, ce phénomène favorise la structuration de clusters plus précis et encourage la spécialisation, chaque entité s'orientant principalement vers les connexions qui maximisent la qualité de la coopération.

C. Limites et paramétrage

L'**inhibition compétitive** dans un réseau requiert une attention particulière quant à ses limites et à son paramétrage. D'abord, il existe un **risque de désactivation excessive** : si le coefficient γ qui contrôle la force de l'inhibition est trop élevé, il peut empêcher la croissance significative de tout nouveau lien, menant ainsi à un réseau **faiblement connecté**. Il est donc crucial de fixer γ à un niveau qui ne freine pas abusivement l'établissement de connexions utiles.

Par ailleurs, la **dépendance à la taille** du voisinage pose également problème. Dans la formule $[-\gamma \sum_{k \neq j} \omega_{i,k}]$, la **somme** peut devenir très importante lorsque l'entité \mathcal{E}_i compte un grand nombre de voisins, annihilant de fait la mise à jour et rendant difficile l'émergence de nouvelles liaisons. Pour y remédier, on peut envisager de normaliser cette somme (par exemple, en la divisant par le nombre de voisins) ou d'adopter une approche “d'inhibition latérale stricte”, qui retient uniquement la valeur maximale au lieu de la somme.

Enfin, il est fréquent de **combiner** l'inhibition compétitive avec un **mécanisme de saturation** : même si la synergie $S(i,j)$ incite un lien à s'intensifier, on l'empêche de dépasser un certain plafond. Cette mesure limite les phénomènes de “toute-puissance” où une entité tenterait d'activer tous ses liens simultanément, renforçant ainsi la sélectivité et la cohérence globale du réseau.

4.2.2.3. Saturation : clipping $\omega_{i,j}$ pour éviter la croissance infinie

Même dans un **DSL** (Deep Synergy Learning) recourant à une mise à jour plutôt “additive” (4.2.1) ou intégrant de l'inhibition (4.2.2.2), il peut arriver que les valeurs $\omega_{i,j}$ aient tendance à **s'emballer**. Par exemple, dans la variante **multiplicative** (4.2.2.1), un lien déjà élevé peut croître encore plus vite, menant à un risque d'**explosion** numérique ou de “monopolisation” du réseau par quelques connexions hypertrophiées. Pour éviter cela, on introduit souvent un **mécanisme de saturation**, parfois appelé **clipping**, limitant la valeur maximale (ou minimale) que peut prendre $\omega_{i,j}$.

Le **principe du clipping** consiste à imposer, à l'issue de chaque itération de mise à jour, une **borne** maximale (et parfois minimale) à la pondération $\omega_{i,j}(t+1)$. Concrètement, cela se traduit par une opération de type :

$$\omega_{i,j}(t+1) \leftarrow \max \{ 0, \min \{ \omega_{i,j}(t+1), \omega_{\max} \} \},$$

assurant ainsi que $\omega_{i,j}(t+1)$ ne dépasse pas la valeur ω_{\max} . Selon les besoins, on peut également l'empêcher de devenir négative, notamment si l'on considère que les liens ne doivent prendre que des valeurs positives.

Dans la pratique, ce **clipping** s'opère **après** le calcul brut de la mise à jour (qu'elle soit additive, multiplicative ou incluant un mécanisme d'inhibition). On recadre la pondération finale uniquement si elle dépasse l'intervalle autorisé : si, par exemple, $\omega_{i,j}(t+1)$ atteignait 50 alors que $\omega_{\max} = 10$, on la ramènerait à la valeur 10.

La **détermination** de ω_{\max} varie selon la nature du problème. Certaines configurations imposent, par exemple, qu'un lien ne dépasse jamais 1 pour des raisons de normalisation. Dans d'autres contextes, on fixe $\omega_{\max} = 10$ en considérant qu'au-delà, la distinction entre pondérations serait négligeable. Par ailleurs, on peut instaurer un seuil plancher (comme $\omega_{\min} = 0$) si l'on souhaite empêcher les liens de devenir négatifs ou si un sens physique (ou logique) exige que les pondérations restent non négatives.

Les **avantages de la saturation** dans un mécanisme de mise à jour des pondérations se manifestent à plusieurs niveaux.

En premier lieu, la présence d'un **plafond** évite l'**explosion** de certaines connexions : dans les règles multiplicatives (4.2.2.1), ou même en l'absence d'un terme de décroissance convenable, quelques liens pourraient croître indéfiniment si la partie positive (par exemple $\eta S(i,j)$) l'emportait toujours sur la partie négative. Grâce à un **clipping** imposant ω_{\max} , on bride ces valeurs pour éviter qu'une minorité de liaisons n'atteigne des niveaux disproportionnés.

Ensuite, ce mécanisme contribue à **stabiliser la dynamique** globale. En empêchant $\omega_{i,j}$ de dépasser ω_{\max} , on restreint la **disparité** entre les connexions, ce qui peut faciliter l'**analyse** du réseau ou accélérer la **convergence**, puisque l'espace de recherche (celui des pondérations) demeure borné.

Enfin, la saturation favorise un **contrôle de la lisibilité** des pondérations. Lorsqu'on désire interpréter $\omega_{i,j}$ comme un simple "score" dans un intervalle $[0,1]$, le clipping assure que ce score restera dans le **domaine** de lecture intuitif ; on n'observera ainsi pas de valeur hors échelle (telle qu'un score à 29,5), dépourvue de sens sémantique pour l'utilisateur ou l'analyste.

Les **inconvénients potentiels** de la saturation (ou points d'attention) doivent être soigneusement considérés au moment d'implémenter le clipping. En premier lieu, on peut évoquer l'**effet brutal** : dès que $\omega_{i,j}$ atteint ω_{\max} , toute mise à jour positive devient sans effet. Cela risque de **biaiser** la dynamique si, par exemple, un lien devrait continuer à croître pour signaler sa domination. Le réseau ne percevra plus la différence entre des liens saturés et ceux proches de la borne, ce qui peut diminuer le contraste souhaité.

Un autre point concerne la **discontinuité** dans la mise à jour. Le clipping crée une **coupure** non lisse : une fois la limite atteinte, la valeur reste figée. Sur le plan mathématique, ces discontinuités peuvent compliquer certaines analyses de convergence.

Le **choix** de ω_{\max} est également crucial. Une borne trop **basse** étouffe les liens qui pourraient être significativement plus élevés que d'autres, affectant la capacité du réseau à différencier des connexions "très fortes" de connexions "modérées". À l'inverse, si ω_{\max} est fixé à un niveau **très élevé**, on ne résout pas réellement le problème de l'explosion, et on n'apporte qu'un contrôle minimal.

Exemple d'Intégration dans la Mise à Jour

Le clipping peut s'insérer de différentes manières dans le processus de mise à jour.

Dans la **version additive** avec clipping, on calcule d'abord :

$$1. \omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j)\tau \omega_{i,j}(t)],$$

puis on applique :

$$\omega_{i,j}(t+1) \leftarrow \min\{\max\{0, \omega_{i,j}(t+1)\}, \omega_{\max}\}.$$

On garantit ainsi que $\omega_{i,j}(t+1)$ demeure dans l'intervalle $[0, \omega_{\max}]$.

Dans la **version multiplicative**, une fois le calcul effectué (voir 4.2.2.1), on impose simplement $\omega_{i,j}(t+1) \leq \omega_{\max}$.

Lorsqu'on **combine** cette approche avec une **inhibition compétitive**, l'inhibition agit en amont (pour réduire la mise à jour) et le **clamping** final vient s'assurer qu'un lien ne dépasse pas le plafond fixé. De cette manière, même si la synergie est très élevée, on limite la croissance excessive d'un lien unique, préservant l'équilibre général du réseau.

4.2.3. Cas Multi-Entités Hétérogènes

Les sections précédentes (4.2.1 et 4.2.2) ont présenté diverses **règles** de mise à jour (additives, multiplicatives, avec ou sans inhibition, saturation...) pour les pondérations $\omega_{i,j}$. Cependant, elles partaient souvent de l'hypothèse qu'il existait un seul **type** d'entité ou une forme unique de $S(i,j)$. Or, dans un **DSL** (Deep Synergy Learning) plus complet (chap. 3), on peut croiser :

- Des **entités sub-symboliques** (embeddings, vecteurs),
- Des **entités symboliques** (règles logiques, blocs ontologiques),
- Des **entités hybrides** (mix sub + sym),

et chacune d'elles peut interagir sous forme de différents **types de liens**. Cette diversité amène à considérer un **multi-réseau** ou des **sous-réseaux** partiellement interconnectés, avec des règles de mise à jour possiblement **différencierées** selon la nature des entités et des liaisons.

4.2.3.1. Sous-réseaux sub-symboliques / symboliques (cf. chapitre 3)

Dans le Deep Synergy Learning (DSL), il peut exister des **sous-réseaux sub-symboliques** (où les entités \mathcal{E}_i sont décrites par des embeddings \mathbf{x}_i , et où la synergie $S_{\text{sub}}(i,j)$ se base sur des critères vectoriels) et des **sous-réseaux symboliques** (où d'autres entités \mathcal{E}_p sont définies par des **règles** ou un **bloc** ontologique, et où $S_{\text{sym}}(p,q)$ repose sur la **compatibilité** logique). Ce modèle mixte, déjà évoqué au chapitre 3, illustre la capacité du DSL à intégrer, dans un **même cadre**, des entités numériques et des entités logiques.

Un **exemple** apparaît en **robotique** (chap. 3.6.3.3), avec des entités représentant des **perceptions** (embeddings sensoriels) et d'autres correspondant à des **actions** décrites symboliquement (précisant préconditions et effets). De même, pour un **agent conversationnel** (chap. 3.6.3.2), certaines entités sont des **segments textuels** (embedding BERT), tandis que d'autres englobent des **règles** de cohérence de dialogue.

Sur le plan **structurel**, on se retrouve souvent avec plusieurs "couches" d'entités :

- La couche "sub" manipule \mathbf{x}_i et inclut des liens sub–sub,
- La couche "sym" gère les blocs logiques et inclut des liens sym–sym,
- Et il existe parfois des **passerelles** (liens sub–sym) reliant les deux couches.

L'**enjeu** majeur réside dans la capacité de chaque type d'entité à se **mettre à jour** (par exemple, une représentation sub-symbolique qui évolue, ou des règles symboliques qui se modifient), et dans

la prise en compte du fait que la **synergie** varie selon la nature des entités connectées. Par conséquent, les mécanismes de mise à jour (additifs, multiplicatifs, saturation, etc.) décrits en 4.2.1 et 4.2.2 doivent être **adaptés** à la catégorie de la liaison (sub–sub, sym–sym ou sub–sym) pour demeurer cohérents avec les spécificités de chaque sous-réseau.

4.2.3.2. Mise à jour distincte selon le type de lien (ex. sub–sub, sym–sym, sym–sub)

Dans le cadre d'un **Deep Synergy Learning (DSL)** hétérogène, les entités peuvent appartenir à des natures différentes, qu'elles soient **sub-symboliques** (décrites par des **embeddings** tels que $\mathbf{x}_i \in \mathbb{R}^d$), **symboliques** (décrites par un ensemble de **règles**, axiomes ou ontologies, noté R_i) ou **hybrides** (combinant à la fois des représentations vectorielles et des blocs logiques).

Cette diversité des représentations se traduit par des **types de liens** distincts dans le réseau : on distingue ainsi les liens **sub–sub** (entre deux entités sub-symboliques), les liens **sym–sym** (entre deux entités symboliques) et les liens **sym–sub** (passerelles entre le domaine symbolique et le domaine sub-symbolique).

Chaque type de lien requiert une **mise à jour** adaptée, car la **fonction de synergie** $S(i, j)$ utilisée pour moduler l'évolution de la pondération $\omega_{i,j}$ diffère en fonction des caractéristiques de chaque modalité. Nous détaillons ci-après les principes sous-jacents à la mise à jour pour chacun de ces types de liens.

A. Liens sub–sub

Lorsqu'un lien relie deux entités \mathcal{E}_i et \mathcal{E}_j qui sont toutes deux représentées par des **embeddings** \mathbf{x}_i et \mathbf{x}_j dans \mathbb{R}^d , la **synergie sub-symbolique** est généralement évaluée à l'aide de mesures vectorielles classiques. Par exemple, la **similarité cosinus** se définit par

$$S_{\text{sub}}(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|},$$

ce qui quantifie le degré de rapprochement entre les directions de ces vecteurs. Alternativement, on peut utiliser une mesure basée sur une fonction noyau, comme un noyau gaussien exprimé par

$$S_{\text{sub}}(i, j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2),$$

où $\alpha > 0$ règle la sensibilité du score. La **mise à jour** standard de la pondération dans ce cas se fait selon la règle additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{sub}}(i, j) - \tau \omega_{i,j}(t)],$$

où η est le **taux d'apprentissage** et τ le **coefficent de décroissance** qui agit comme un terme de régularisation. Cette formulation, d'une complexité de l'ordre de $O(d)$ pour l'évaluation du score, permet de renforcer progressivement les liens entre des entités dont les représentations sont numériquement proches, par exemple deux images similaires ou deux segments textuels ayant des significations proches.

Des variantes multiplicatives peuvent être envisagées si l'on souhaite accentuer l'effet exponentiel sur la croissance des liens, mais le principe fondamental demeure celui de la convergence vers un équilibre où, en situation stationnaire, $\omega_{i,j}^* \approx \frac{S_{\text{sub}}(i,j)}{\tau}$.

B. Liens sym–sym

Dans le cas des liens **sym–sym**, les entités \mathcal{E}_i et \mathcal{E}_j sont caractérisées par des **blocs de règles** ou des structures logiques, notées R_i et R_j . La fonction de synergie symbolique $S_{\text{sym}}(i,j)$ évalue la **compatibilité** entre ces ensembles de règles, ce qui peut être formulé de manière binaire ou graduée. Par exemple, on peut définir

$$S_{\text{sym}}(R_i, R_j) = \begin{cases} 1, & \text{si } R_i \text{ et } R_j \text{ sont entièrement compatibles,} \\ 0, & \text{en cas de contradiction totale,} \\ s_{ij}, & \text{si la compatibilité est partielle, avec } s_{ij} \in (0,1). \end{cases}$$

La mise à jour des pondérations pour les liens sym–sym suit un schéma similaire à celui des liens sub–sub, à savoir

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{sym}}(i,j) - \tau \omega_{i,j}(t)],$$

ce qui permet de renforcer les connexions entre entités dont les blocs de règles se valident mutuellement. Toutefois, il faut noter que la comparaison symbolique peut s'avérer plus coûteuse en termes de calcul, surtout si l'inférence sur des systèmes logiques complexes est nécessaire. Des mécanismes d'**inhibition** ou des seuils peuvent être introduits pour limiter l'impact des incohérences et éviter une prolifération excessive des liens lorsque la logique est trop rigide.

C. Liens sym–sub

Les **liens sym–sub** constituent des passerelles entre les représentations sub-symboliques et symboliques. Dans ce cas, une entité \mathcal{E}_i peut être représentée par un **embedding** \mathbf{x}_i tandis qu'une autre entité \mathcal{E}_j est décrite par un bloc de règles R_j , ou vice versa. Pour mesurer la synergie dans un tel cas, il est courant de définir une fonction hybride qui combine la **similarité vectorielle** et la **compatibilité logique**. Une telle fonction s'exprime par

$$S_{\text{hybrid}}(i,j) = \alpha S_{\text{sub}}(i,j) + (1 - \alpha) S_{\text{sym}}(i,j),$$

où le paramètre $\alpha \in [0,1]$ ajuste l'importance relative des deux composantes. La mise à jour des pondérations pour ces liens hybrides est alors donnée par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_{\text{hybrid}}(i,j) - \tau \omega_{i,j}(t)].$$

Cette approche permet de **fusionner** les informations provenant du monde numérique (embeddings) et du monde logique (règles), ce qui est particulièrement utile dans des applications où il est nécessaire de combiner la richesse des données sub-symboliques avec une **explicabilité** issue du raisonnement formel. Toutefois, l'évaluation simultanée des deux composantes peut augmenter la **complexité** de l'algorithme, et un réglage fin du paramètre α est indispensable pour obtenir le compromis optimal entre flexibilité et précision.

4.2.3.3. Exemples de pseudo-code unifiant ces règles

Dans un **Deep Synergy Learning (DSL)** hétérogène, la gestion des liens entre entités repose sur la mise à jour de pondérations $\omega_{i,j}$ qui varient selon la **nature** des entités en interaction. En effet, les entités peuvent être purement **sub-symboliques** (représentées par des **embeddings** $\mathbf{x}_i \in \mathbb{R}^d$), purement **symboliques** (définies par un ensemble de **règles** ou d'**axiomes** noté R_i), ou encore **hybrides** (combinant à la fois un embedding et un bloc logique). Pour chaque type de lien – qu'il s'agisse de liens sub–sub, sym–sym ou sym–sub – la **fonction de synergie** $S(i,j)$ et la stratégie de mise à jour doivent être adaptées. La présente section présente un **pseudo-code** unifié qui illustre comment, dans une même boucle d'itération, ces règles de mise à jour distinctes peuvent être intégrées de manière homogène dans l'algorithme de mise à jour des pondérations du **Synergistic Connection Network (SCN)**.

A. Présentation générale du cadre

Soit un ensemble d'entités $\{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_n\}$ dont chacune est associée à une représentation qui peut être sub-symbolique, symbolique ou hybride. Nous disposons d'une **matrice** $\{\omega_{i,j}\}$ représentant les pondérations actuelles entre chaque paire d'entités. Le système repose sur une fonction générique, que nous appellerons $\text{ComputeSynergy}(i,j,\text{link_type})$, qui, en fonction du **type de lien** entre \mathcal{E}_i et \mathcal{E}_j (déterminé par une fonction $\text{determine_link_type}(E[i], E[j])$), renvoie le score de synergie approprié.

Ainsi, dans le cas d'un lien sub–sub, ComputeSynergy renverra un score $S_{\text{sub}}(i,j)$ calculé à partir des embeddings, typiquement par une mesure comme la similarité cosinus ou un noyau gaussien, tandis que pour un lien sym–sym, la fonction renverra un score $S_{\text{sym}}(i,j)$ évalué sur la compatibilité des règles, et dans le cas d'un lien sym–sub, une combinaison des deux scores sera utilisée, par exemple via une formule hybride.

La règle de mise à jour additive, commune à tous les types, se présente sous la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où η est le **taux d'apprentissage** et τ le **coefficent de décroissance**. Des variantes multiplicatives peuvent également être envisagées, mais dans ce pseudo-code nous nous concentrerons sur la règle additive. Afin de contrôler la croissance excessive des pondérations, des mécanismes de **clipping** (pour borner les valeurs) et d'**inhibition** (pour tenir compte de la compétition entre liens) sont intégrés dans la mise à jour.

B. Pseudo-code unifié

Le pseudo-code suivant, de style Python, illustre comment on peut unifier les mises à jour des différents types de liens dans une boucle d'itération :

```
# Paramètres globaux
eta_sub = 0.01 # Taux d'apprentissage pour liens sub–sub
eta_sym = 0.02 # Taux pour liens sym–sym
eta_hyb = 0.015 # Taux pour liens sym–sub
tau    = 0.1   # Coefficient de décroissance
gamma  = 0.05  # Coefficient pour inhibition (optionnel)
```

```

omega_max = 5.0    # Valeur maximale pour le clipping
USE_INHIBITION = True # Option pour activer le mécanisme d'inhibition

# Boucle principale sur toutes les entités
for i in range(n):          # Pour chaque entité i
    for j in range(n):      # Pour chaque entité j
        if i == j:
            continue          # Pas de mise à jour pour i=j

# Déterminer le type de lien entre E[i] et E[j]
link_type = determine_link_type(E[i], E[j])
# Exemples de valeurs possibles: "sub-sub", "sym-sym", "sym-sub"

# Calcul de la synergie S(i,j) en fonction du type de lien
S_ij = ComputeSynergy(E[i], E[j], link_type)
# Par exemple, pour un lien sub-sub, ComputeSynergy renvoie:
#   S_sub = (E[i].x dot E[j].x) / (||E[i].x|| * ||E[j].x||)
# Pour un lien sym-sym, il renvoie un score basé sur la compatibilité logique
# Pour un lien sym-sub, il combine les deux scores via S_hybrid = alpha*S_sub + (1-alpha)*S_sym

# Calcul optionnel d'un terme d'inhibition, basé sur la somme des poids sortants de i
inhib_term = 0.0
if USE_INHIBITION:
    sum_weights = 0.0
    for k in range(n):
        if k != j:
            sum_weights += w[i][k]
    inhib_term = gamma * sum_weights

# Sélection du taux d'apprentissage spécifique selon le type de lien
if link_type == "sub-sub":
    learning_rate = eta_sub
elif link_type == "sym-sym":
    learning_rate = eta_sym
else: # pour "sym-sub"
    learning_rate = eta_hyb

# Mise à jour additive de la pondération
# Formule : w_new = w(i,j) + learning_rate * (S_ij - tau*w(i,j)) - inhib_term
w_new = w[i][j] + learning_rate * (S_ij - tau * w[i][j]) - inhib_term

# Application d'un clipping pour éviter des valeurs extrêmes
if w_new < 0:
    w_new = 0
if w_new > omega_max:
    w_new = omega_max

# Mise à jour de la matrice de pondérations
w[i][j] = w_new

```

Explications détaillées :

Dans ce pseudo-code, la variable E représente la liste des entités du DSL. Chaque entité peut être accompagnée d'un **embedding** (pour les entités sub-symboliques), d'un **bloc de règles** (pour les entités symboliques) ou d'une combinaison des deux (pour les entités hybrides). La fonction *determine_link_type($E[i]$, $E[j]$)* permet de distinguer le type de lien à établir entre deux entités en se basant sur leur nature respective. Par exemple, si deux entités possèdent toutes deux un embedding, le lien est qualifié de **sub–sub** ; si elles sont toutes deux décrites par des règles, le lien est **sym–sym** ; et si l'une est sub-symbolique et l'autre symbolique, le lien est dit **sym–sub**.

La fonction *ComputeSynergy($E[i]$, $E[j]$, $link_type$)* renvoie le score de synergie approprié en fonction du type de lien. Pour les liens sub–sub, ce score est généralement calculé à partir de mesures classiques telles que la similarité cosinus ou une fonction de noyau gaussien, tandis que pour les liens sym–sym, il s'agit d'un score reflétant la compatibilité ou l'absence de contradictions entre les blocs logiques R_i et R_j . Pour les liens sym–sub, on combine les deux approches via une formule hybride, par exemple

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j).$$

Le **terme d'inhibition** est calculé (si activé) en sommant les pondérations sortantes d'une entité pour modérer la croissance excessive des liens et éviter une saturation du réseau. Chaque type de lien peut bénéficier d'un **taux d'apprentissage** distinct, ici symbolisé par η_{sub} , η_{sym} et η_{hyb} pour les liens sub–sub, sym–sym et sym–sub, respectivement. La mise à jour s'effectue de manière additive, comme le montre la formule utilisée, puis le **clipping** est appliqué pour garantir que les pondérations restent dans un intervalle prédéfini, par exemple $[0, \omega_{\max}]$.

Ce pseudo-code s'exécute de manière itérative sur l'ensemble des entités, et peut être répété pour de nombreuses itérations jusqu'à convergence ou pendant une durée spécifiée. Il offre une illustration concrète de la façon dont un **DSL** peut gérer simultanément des règles de mise à jour différencier pour plusieurs types de liens, en assurant une intégration harmonieuse des informations sub-symboliques et symboliques.

Conclusion

Le présent pseudo-code constitue un exemple unifié qui intègre les différentes **règles de mise à jour** selon le type de lien dans un DSL hétérogène. En adaptant dynamiquement le taux d'apprentissage et en appliquant des mécanismes complémentaires tels que l'inhibition et le clipping, le système parvient à mettre à jour les pondérations $\omega_{i,j}$ de manière cohérente, tout en tenant compte des spécificités des interactions sub–sub, sym–sym et sym–sub. Ce schéma permet ainsi d'unifier la mise à jour des liens dans un SCN, assurant la **robustesse** et la **flexibilité** de l'auto-organisation même dans un environnement où coexistent plusieurs types de représentations.

4.3. Émergence de Clusters et Attracteurs

Dans les sections précédentes (4.2), nous avons décrit les différentes **règles** de mise à jour permettant aux pondérations $\omega_{i,j}$ d'évoluer sous l'effet de la synergie $S(i,j)$, de termes de décroissance, d'inhibition, ou de saturation. L'un des **Résultats majeurs** de cette dynamique d'auto-organisation est l'**émergence** spontanée de **clusters** — c'est-à-dire de sous-ensembles d'entités qui se lient fortement entre elles, tout en se distançant des autres.

En creusant ce phénomène, on découvre également la notion d'**attracteurs** où, selon la configuration initiale et la structure de la synergie, le **SCN** (Synergistic Connection Network) peut aboutir à différents **points fixes** correspondant à des clusters stabilisés, ou manifester des comportements plus riches comme la **multi-stabilité** et les **oscillations**. Cette section (4.3) analysera donc :

- Comment les **clusters** se forment (4.3.1),
- Les concepts d'**attracteurs multiples** et de **bascules** (4.3.2),
- Les scénarios de **stabilisation** ou de **dynamisme continu** (4.3.3), montrant que le réseau peut se figer ou rester en mutation constante, selon le flux de synergie et l'ajout de nouvelles données.

4.3.1. Formation Spontanée de Groupes

Lorsque l'on met en place les **mécanismes** décrits en 4.2, qu'il s'agisse de mise à jour additive ou multiplicative, d'inhibition ou d'autres régulations, les **entités** $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$ ne tardent pas à voir leurs liens $\omega_{i,j}$ s'**organiser** où certains liens deviennent **forts**, d'autres s'étiolent et des **groupes** se forment. C'est précisément ce qui confère au **DSL** (Deep Synergy Learning) sa **capacité** d'auto-organisation où, en "laissant faire" la dynamique, on aboutit à des **clusters** d'entités hautement interconnectées tout en limitant les connexions hors-clusters.

4.3.1.1. Notion de Clusters : pondérations fortes à l'intérieur d'un sous-ensemble \mathcal{C} , faibles à l'extérieur

Lorsqu'on examine la structure d'un **Synergistic Connection Network (SCN)**, il est courant d'observer l'émergence de groupes d'entités, appelés **clusters** ou **communautés**, à l'intérieur desquels les **pondérations** $\omega_{i,j}$ sont nettement plus **élèves** que celles reliant ces entités au **reste** du réseau. Autrement dit, un **sous-ensemble** $\mathcal{C} \subseteq \{1, \dots, n\}$ va exhiber des liens $\omega_{i,j}$ particulièrement **forts** pour $(i, j \in \mathcal{C})$, tandis que les connexions $\omega_{i,k}$ vers un $k \notin \mathcal{C}$ resteront **relativement faibles**.

Pour illustrer cette idée, on peut considérer un exemple simple où, si $\mathcal{C} = \{2, 5, 7\}$ forme un **cluster**, on **attend** qu'à l'issue de la dynamique DSL, les pondérations $\omega_{2,5}$, $\omega_{5,7}$ et $\omega_{2,7}$ deviennent

élevées, tandis que les liaisons $\omega_{2,k}, \omega_{5,k}, \omega_{7,k}$ pour $k \notin \{2,5,7\}$ demeurent **faibles** ou se réduisent progressivement au fil des itérations.

La **raison mathématique** de cette formation de clusters tient à la **synergie** où, lorsque pour un ensemble donné d'entités, la fonction $S(i,j)$ est, en moyenne, plus élevée entre elles qu'avec l'extérieur, la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$$

tend à **renforcer** progressivement $\omega_{i,j}$ pour $(i,j \in \mathcal{C})$, consolidant ainsi la formation du cluster. Symétriquement, dès lors que la synergie entre $(i \in \mathcal{C})$ et $(k \notin \mathcal{C})$ est plus faible, la pondération $\omega_{i,k}(t)$ s'**amenuise** ou reste basse, et ne concurrence pas la liaison intra-groupe.

Un aspect central de cette dynamique est l'**absence de supervision** où le système n'exige aucun label ou contrainte externe pour déterminer quelles entités devraient s'assembler. Seuls les **patterns** de synergie, qu'ils soient issus de la similarité (embeddings, co-occurrences) ou d'une logique symbolique (chap. 3.5), suffisent à produire une **dépendance mutuelle** entre entités. Si la synergie révèle des affinités nettement plus fortes au sein du sous-ensemble \mathcal{C} , la **dynamique DSL** consolide précisément ces liens.

Sur le plan **théorique**, ce processus rejoint l'idée du **clustering** en apprentissage non supervisé, à ceci près que la dynamique DSL reste **entièvement itérative** et **coopérative**. Au lieu de définir un algorithme statique (k-means ou agglomératif) qui opère en une passe sur les données, on laisse la **matrice** $\{\omega_{i,j}\}$ évoluer au fil du temps et, de manière **auto-organisée**, "dessiner" des **îlots** fortement interconnectés.

Une fois l'étape de **convergence** atteinte, la matrice ω révèle explicitement l'existence de **clusters** où l'on peut repérer ces groupes en appliquant un **seuil** sur $\omega_{i,j}$, les plus fortes valeurs identifiant les sous-ensembles fortement liés, ou en recourant à des méthodes de **community detection**, comme l'algorithme de Louvain ou la maximisation de la modularité, sur le graphe formé par ω .

Sur le plan **macro**, cette **formation de clusters** marque l'un des aboutissements majeurs de la **synergie** où le **SCN**, initialement composé d'entités possiblement dispersées et de pondérations faibles ou aléatoires, se **structure** progressivement en regroupements cohérents, reflétant la **cohésion** intrinsèque que la fonction S a détectée dans les données. Cette propriété d'**auto-organisation** constitue l'un des piliers fondamentaux du DSL, montrant comment, à partir de **règles** locales et d'ajustements distribués, le réseau peut dégager une **partition** (partielle ou totale) du jeu d'entités en **communautés** pertinentes, sans qu'aucun paramétrage ou nombre de clusters ne soit imposé a priori.

4.3.1.2. Processus Auto-Organisé : sans supervision, certains liens se renforcent massivement (haute synergie), d'autres décroissent

Le **phénomène** de formation de **clusters** décrit en section 4.3.1.1 ne repose pas sur l'utilisation d'un algorithme de **clustering** classique ni sur la présence de **labels** ou de **catégories** prédefinies. Au contraire, il résulte d'un **processus auto-organisé** intrinsèque aux pondérations $\omega_{i,j}$, lesquelles évoluent en fonction de la **synergie** $S(i,j)$ et des règles de mise à jour (décris au paragraphe 4.2).

C'est précisément ce caractère **non supervisé** et **dynamique** qui caractérise le **DSL** (Deep Synergy Learning) et qui en fait sa spécificité.

Caractère non supervisé.

Dans une configuration d'**apprentissage supervisé**, on dispose généralement de classes ou de cibles explicites pour guider la création de groupes. Ici, la démarche ne s'appuie sur **aucune** annotation où seule la **synergie** $S(i,j)$, qu'elle provienne de similarités sub-symboliques, de compatibilités logiques ou d'autres sources (cf. chapitre 3), oriente la **croissance** ou la **décroissance** des pondérations $\omega_{i,j}$. L'évolution du réseau reste ainsi *auto-organisée*, car elle ne fait appel à aucun signal externe imposant la façon de répartir ou de regrouper les entités.

L'**analogie** avec certains réseaux neuronaux biologiques est souvent mise en avant où, dans le cerveau, les synapses voient leur force modifiée selon l'activité neuronale partagée, une certaine forme de **co-activation** ou de **synergie**, sans qu'un **superviseur** n'affecte manuellement des étiquettes à chaque neurone. Le **DSL** transpose ces principes à des entités abstraites (tels que des textes, des images, des règles logiques, etc.), dont les liens se renforcent ou s'affaiblissent selon leur synergie mutuelle.

Renforcement massif de certains liens, décroissance des autres.

Imaginons qu'à l'initialisation, certaines paires (i,j) présentent une **synergie** $S(i,j)$ relativement élevée, tandis que d'autres pairs (i,k) sont moins corrélées. La dynamique DSL (qu'elle soit **additive** ou **multiplicative**, avec ou sans mécanismes d'**inhibition**, de **saturation**, etc.) va **favoriser** la montée en puissance de $\omega_{i,j}$ dès lors que $S(i,j)$ se révèle important. Au fil des itérations, la pondération $\omega_{i,j}$ tend à **se stabiliser** (voire à continuer d'augmenter si les régulations le permettent), constituant ainsi un **cœur** de liaisons robustes.

De manière symétrique, les couples (i,k) pour lesquels la synergie $S(i,k)$ se montre plus faible subissent l'effet de **décroissance** lié à $-\tau \omega_{i,k}(t)$ (et éventuellement à l'inhibition compétitive). Cela se traduit par une **baisse** progressive, voire une extinction, de $\omega_{i,k}$. Au terme de ce processus, certains liens s'en trouvent **renforcés** de façon marquée, et la majorité finit par s'**affaiblir** notamment. Cette situation aboutit à des "**îlots de forte connectivité**", constituant les futurs **clusters**.

Dans les versions dites **multiplicatives** de la mise à jour, un léger avantage initial d'une pondération peut conduire à une **croissance exponentielle**, accentuant encore la séparation entre les connexions "vainqueurs" (dotées d'une synergie forte) et les connexions "perdantes" (qui demeurent modestes ou décroissent). Même en mode **additif**, un contraste initial de synergie suffit à créer, à l'issue de la dynamique, un écart stable où l'une des liaisons converge vers $S(i,j)/\tau$ et l'autre vers $S(i,k)/\tau$. Si $S(i,j) > S(i,k)$, la différence se creuse inévitablement.

Émergence de clusters.

À mesure que ces liaisons puissantes se démarquent et que la majorité des autres s'affaiblit, on observe la formation naturelle de **sous-ensembles** d'entités — c'est-à-dire de **clusters** — unis par des pondérations $\omega_{i,j}$ élevées. Au plan **géométrique** ou **sémantique**, cette différence renvoie à la **répartition** des synergies S dans l'espace des données où, là où $S(i,j)$ s'avère plus grand, la pondération $\omega_{i,j}$ finit par structurer des **composantes** ou **communautés** au sein du **SCN**. La

matrice $\{\omega_{i,j}\}$ se **sculpte** littéralement sous l'action de la dynamique **DSL**, laissant apparaître des **îlots de forte connectivité**.

En pratique, on peut suivre, au fil des itérations, la progression vers des “clusters” plus ou moins stables. Durant les premiers pas, plusieurs connexions peuvent simultanément croître, puis l'**inhibition** ou la **saturation** vont faire émerger des groupes plus définis, car un lien fort d'une entité vers un ensemble donné empêche la prolifération de liens simultanés vers d'autres ensembles.

Selon les conditions initiales (valeurs de η , τ , coefficient d'inhibition γ , etc.) et la distribution des synergies, le réseau finit fréquemment par converger vers un **état stable** (ou un quasi-état stable), découvant implicitement l'ensemble des entités en **sous-groupes**.

Cette séparation se produit sans affectation de labels ni contrainte externe, illustrant la pleine **auto-organisation**, où la seule fonction S , intégrée à la mise à jour, produit l'**émergence** de **clusters** pertinents.

4.3.2. Attracteurs Multiples et Bascules

Lorsque la dynamique d'auto-organisation (chap. 4.2) agit sur les pondérations $\omega_{i,j}$, on peut observer divers **phénomènes** de stabilité ou d'instabilité. En particulier, il arrive que le **SCN** (Synergistic Connection Network) dispose de **plusieurs** configurations stables (ou quasi-stables), vers lesquelles il peut converger selon l'initialisation des $\omega_{i,j}$ ou l'évolution du **flux de synergie** $S(i,j)$. C'est ce qu'on appelle la **multi-stabilité**, ou la co-existence de **plusieurs attracteurs**.

Par “**attracteurs multiples**”, on entend que le système (la matrice $\{\omega_{i,j}\}$) peut aboutir à différentes **configurations finales** (points fixes, cycles, régimes complexes) selon les conditions initiales ou la chronologie des mises à jour. On peut alors assister à des **basclements** entre attracteurs, si le **flux** $S(i,j)$ ou d'autres paramètres changent en cours de route.

4.3.2.1. Multi-Stabilité : un SCN peut atteindre différents points fixes selon l'initialisation ou le flux de synergie (2.3.2)

A. Rappel mathématique : points fixes et stabilité

Les réseaux **DSL** (Deep Synergy Learning), ou plus largement les systèmes de type **SCN** (Synergistic Connection Network), se décrivent par un ensemble de liens $\omega_{i,j}(t)$ qui évoluent au fil du temps. De manière formelle, on peut écrire cette évolution sous la forme d'une **application** \mathbf{F} opérant sur la configuration globale $\boldsymbol{\omega}(t)$. À chaque pas d'itération, la dynamique s'écrit :

$$\boldsymbol{\omega}(t+1) = \mathbf{F}(\boldsymbol{\omega}(t)),$$

où $\boldsymbol{\omega}(t)$ représente le vecteur (ou la matrice) regroupant l'ensemble $\{\omega_{i,j}(t)\}$. Un **point fixe** $\boldsymbol{\omega}^*$ de ce système satisfait

$$\boldsymbol{\omega}^* = \mathbf{F}(\boldsymbol{\omega}^*).$$

Lorsqu'on a affaire à une fonction **F non linéaire**, il peut exister plusieurs **solutions** $\omega_1^*, \omega_2^*, \dots$ satisfaisant cette même équation, chaque solution correspondant à un **arrangement stable** des pondérations $\omega_{i,j}$. Dans la théorie des systèmes dynamiques, chacun de ces états constitue un **attracteur local**. Ainsi, selon l'**initialisation** $\omega(0)$ ou les perturbations ultérieures, la trajectoire $\omega(t)$ peut aboutir à l'un ou l'autre de ces points fixes, chacun possédant son **bassin d'attraction** propre.

Dans le cadre spécifique du **DSL**, la fonction **F** dépend de la **synergie** $S(i,j)$. Si la synergie est **constante** (aucune variation au fil du temps, aucun nouveau flux de données), la multi-stabilité découle essentiellement de la **non-linéarité** (par exemple l'inhibition compétitive ou le mécanisme multiplicatif). Au contraire, si la synergie $S(i,j)$ évolue (arrivée de nouvelles entités, changement de paramètres), le système peut, pendant sa trajectoire temporelle, **changer** d'attracteur si le "paysage" dynamique se modifie suffisamment.

B. Exemples conceptuels de multi-stabilité

Pour se faire une intuition, il est possible d'imaginer un **mini réseau** de seulement trois entités $\{1,2,3\}$ et de se donner une mise à jour additive ou multiplicative. Le flux de synergies $S(1,2)$, $S(2,3)$, $S(1,3)$ peut être choisi de façon que deux configurations concurrentes soient toutes deux "plausibles". Dans la première, les entités $\{1,2\}$ forment un cluster, excluant la troisième, tandis que dans la seconde, ce sont $\{2,3\}$ qui se solidarisent. Si l'initialisation favorise l'émergence de $\{1,2\}$ (par un bruit initial en ce sens), on converge naturellement vers cette partition, tandis qu'une initialisation différente fait converger vers $\{2,3\}$.

À plus grande échelle, si l'on dispose de nombreuses entités, il peut y avoir **plusieurs partitions** tout aussi cohérentes. La nature **non linéaire** de la règle, par exemple l'inhibition compétitive, accentue cette tendance où un groupe se forme et se stabilise d'un côté, un autre se stabilise différemment ailleurs, et il se peut que deux exécutions nominalement équivalentes aboutissent à des **arrangements** distincts. Cela renvoie à la notion de **paysage d'énergie** comportant plusieurs "puits" locaux, chacun correspondant à un **attracteur** stable. Ces principes sont largement étudiés dans la littérature des systèmes dynamiques et rappellent, dans une certaine mesure, les réseaux de Hopfield (où plusieurs états mémorisés coexistent comme attracteurs), ou encore les chaînes de Markov possédant différents états absorbants.

C. Rôle de l'initialisation et du flux de synergie

La **dépendance** vis-à-vis de l'initialisation $\omega(0)$ se comprend en analysant la trajectoire donnée par $\omega(t+1) = F(\omega(t))$. Très souvent, on part d'une matrice $\omega_{i,j}(0)$ proche de zéro (ou perturbée par un faible bruit gaussien). Mais de petites variations dans cette phase initiale peuvent se répercuter fortement, en raison des **bifurcations** qu'introduit la non-linéarité.

Si la **synergie** $S(i,j)$ reste **constante** dans le temps, le système évolue vers l'un de ses points fixes disponibles, typiquement en une seule "phase" de convergence. En revanche, si S **varie** (arrivée de nouvelles données, recalcul d'embeddings, etc.), la fonction **F** se modifie, et un attracteur qui était stable peut cesser de l'être. Le **réseau** peut alors **basculer** vers un autre attracteur plus conforme au nouveau contexte, parfois par un phénomène de "cascade" de réajustements.

Les **perturbations** extérieures (inhibition accrue, bruit stochastique, recuit simulé) peuvent également déstabiliser l'état courant et “pousser” le réseau vers une autre configuration stable.

D. Illustration mathématique : Jacobienne et stabilité locale

Pour analyser la stabilité locale d'un point fixe, on se réfère à la **matrice jacobienne** $\mathbf{DF}(\boldsymbol{\omega}^*)$. Étant donné la dynamique

$$\boldsymbol{\omega}(t+1) = \mathbf{F}(\boldsymbol{\omega}(t)),$$

la **condition** de stabilité veut que l'opérateur linéarisé $\mathbf{DF}(\boldsymbol{\omega}^*)$ (qui décrit la sensibilité à de petites variations autour de $\boldsymbol{\omega}^*$) ait un rayon spectral (ou une norme appropriée) < 1 . Cela garantit que la trajectoire reste dans l'orbite de $\boldsymbol{\omega}^*$ et ne diverge pas.

Lorsque $\mathbf{DF}(\boldsymbol{\omega}^*)$ permet à plusieurs $\boldsymbol{\omega}^*$ d'être stables, on parle de **co-existence** d'attracteurs. Chaque attracteur dispose de son **bassin**. Dans un cadre DSL de grande dimension, ce phénomène de co-existence peut amener à une grande diversité de partitions possibles, ou à des états stables différents selon l'histoire de la trajectoire.

E. Basculement entre attracteurs

Le fait de **changer** un paramètre η, γ ou τ , de faire **arriver** un nouveau flux de données, ou de recalculer **S** peut soudainement **rendre** un attracteur obsolète et permettre à la dynamique de *s'échapper* vers une configuration plus basse en énergie. Dans la théorie des systèmes dynamiques, on parle alors de **bifurcation**, ou de “transition de phase” lorsqu'on l'analyse avec une perspective inspirée de la physique statistique.

À grande échelle, on peut **voir** ce basculement se traduire par l'évolution de la matrice ω où un bloc d'entités qui formait un cluster disparaît subitement ou se scinde en deux, tandis qu'un autre cluster émerge ou fusionne avec un tiers. Tout cela résulte d'une multitude de **petits ajustements** $\Delta\omega_{i,j}$ qui, accumulés, aboutissent à un **changement** macroscopique.

4.3.2.2. Cycles ou “Oscillations” : si la fonction de mise à jour favorise des rétroactions positives, on peut avoir des boucles (2.3.2.2)

Dans la continuité du phénomène de **multi-stabilité** (voir 4.3.2.1), il arrive qu'un **SCN** (Synergistic Connection Network) n'aboutisse pas à un point fixe statique mais entre dans un **cycle** récurrent ou une forme d'**oscillation**. Au lieu de converger vers une configuration unique, la dynamique parcourt successivement plusieurs états qui se répètent de manière périodique (cycle de période T) ou quasi-périodique (oscillation complexe). Cette situation survient lorsque les **rétroactions positives** dans la mise à jour de $\omega_{i,j}$ sont suffisamment fortes pour qu'aucun état stationnaire ne s'impose durablement.

A. Rappel conceptuel : rétroactions positives et non-linéarités

Lorsque, dans la règle de mise à jour $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \dots$, la contribution qui accroît $\omega_{i,j}$ se trouve amplifiée dès que $\omega_{i,j}$ est déjà élevée, on parle de **rétroaction positive**. Si cette force d'amplification n'est pas contrebalancée par un mécanisme stabilisateur (par exemple une

décroissance assez puissante, ou une inhibition globale), la trajectoire de $\omega_{i,j}$ peut se mettre à fluctuer au lieu de s'ancrer à une valeur fixe.

Ces comportements de fluctuation ou de “va-et-vient” sont généralement liés à des **non-linéarités**. Dans un **DSL** (Deep Synergy Learning), plusieurs ingrédients **non linéaires** peuvent intervenir où la mise à jour multiplicative suit la forme $\omega_{i,j}(t+1) = \omega_{i,j}(t) \times (1 + \dots)$, l'inhibition compétitive introduit un terme $-\gamma \sum_{k \neq j} \omega_{i,k}$, et la saturation impose une limite ω_{\max} pour éviter une croissance incontrôlée des pondérations. Les théories des **systèmes dynamiques** enseignent que de tels couplages non linéaires suffisent à produire des solutions périodiques (cycles limites) ou, plus rarement, des régimes chaotiques.

B. Caractérisation mathématique : cycle de période T

Sur le plan purement mathématique, un **cycle** de période T pour la dynamique

$$\boldsymbol{\omega}(t+1) = \mathbf{F}(\boldsymbol{\omega}(t))$$

se définit comme une suite de configurations $\{\boldsymbol{\omega}^*(1), \dots, \boldsymbol{\omega}^*(T)\}$ telle que

$$\mathbf{F}(\boldsymbol{\omega}^*(t)) = \boldsymbol{\omega}^*((t \bmod T) + 1), \quad t = 1, \dots, T,$$

et $\boldsymbol{\omega}^*(T+1) = \boldsymbol{\omega}^*(1)$. Autrement dit, après avoir appliqué \mathbf{F} T fois, on revient à l'état initial $\boldsymbol{\omega}^*(1)$.

Dans la pratique, un **cycle de période 2** est particulièrement fréquent où la configuration oscille en **ping-pong** entre deux états. Des cycles de période 3 ou plus longs existent également. Comme pour les points fixes, on peut analyser la **stabilité** de ces cycles en examinant l'évolution locale sur une orbite complète. Si l'analyse (via la matrice dérivée sur ce tour d'orbite) montre que les perturbations y restent contenues ou se contractent, le cycle est stable ; sinon, il peut s'avérer instable et ne se maintenir qu'en conditions idéales.

C. Illustrations dans un DSL

Dans un **DSL** de dimension significative, ces cycles peuvent se traduire par des phénomènes de “fluctuation de clusters”. Par exemple, un ensemble d'entités se regroupe momentanément en un bloc cohérent, puis la dynamique se déplace pour défaire ce cluster et en bâtir un autre, et la configuration revient ensuite au premier arrangement. Cette succession décrit un **cycle** où l'état global du **SCN** n'est jamais figé mais repasse par un chemin périodique dans l'espace des $\{\omega_{i,j}\}$.

Les **transitions** causées par l'inhibition compétitive sont un autre vecteur classique d'oscillations où, dans un triangle $\{1,2,3\}$, si l'on renforce d'abord $\omega_{1,2}$ au détriment de $\omega_{2,3}$, puis que $\omega_{2,3}$ repart à la hausse en inhibant $\omega_{1,2}$, et ainsi de suite, on observe un cycle récurrent entre ces connexions. Le système tourne en boucle et ne se stabilise jamais sur un unique cluster. Des simulations minimalistes montrent aisément cette alternance.

D. Signification et implications

Un tel **cycle** s'oppose à l'idée d'une convergence vers un unique arrangement stable. La présence de rétroactions positives et de couplages non linéaires signifie que le réseau peut “tourner” dans l'espace des configurations. Sur un plan pratique, cela peut compliquer l'exploitation des résultats

où, en l'absence d'un **état final unique**, il devient impossible de conclure à une **partition définitive**, rendant l'interprétation plus instable. Cependant, ces oscillations peuvent être recherchées lorsqu'on désire un comportement de type "exploration continue" ou qu'on souhaite que le SCN "réfléchisse" en alternant plusieurs configurations plausibles.

Dans les grands systèmes, on peut même entrevoir des régimes plus complexes, y compris **chaotiques**, quoique dans un DSL standard, la plupart des mécanismes (décroissance τ , saturation, etc.) limitent souvent l'apparition d'un véritable chaos. On assiste plus fréquemment à des cycles stables (limite cycles) ou des oscillations amorties. Selon les objectifs, on peut vouloir favoriser ces oscillations (pour explorer plus de partitions) ou les réduire (pour atteindre un état stable), en ajustant la force de l'inhibition γ ou de la décroissance τ .

4.3.2.3. Stratégies pour détecter / caractériser un attracteur : qu'est-ce qu'un "bassin d'attraction" ?

Dans les sections précédentes (4.3.2.1 et 4.3.2.2), nous avons vu qu'un **SCN** (Synergistic Connection Network) pouvait admettre divers types d'attracteurs, tels que des **points fixes** (multi-stabilité) ou des **cycles** (oscillations). Une question fondamentale se pose alors où il s'agit de **déterminer** si un réseau a effectivement atteint un **attracteur** et de **décrire** la "zone d'influence" de cet attracteur, connue sous le nom de **bassin d'attraction**.

Il s'agit d'un sujet central en **théorie des systèmes dynamiques** et en **analyse numérique**, qui, dans le cadre d'un DSL, peut être abordé à la fois par des méthodes analytiques (examen de dérivées, jacobiniennes) et par des méthodes empiriques (simulation, exploration des conditions initiales).

A. Qu'est-ce qu'un attracteur ?

Un **attracteur** se définit comme un **ensemble** (point fixe, cycle limite, orbite quasi-périodique, attracteur chaotique, etc.) auquel un grand nombre de **trajectoires** convergent — ou auquel elles se maintiennent de façon récurrente. Une fois la dynamique $\{\omega_{i,j}(t)\}$ entrée dans un voisinage de cet attracteur, elle n'en sort plus (ou s'en éloigne très difficilement).

Lorsque l'attracteur est un **point fixe** ω^* , on a

$$\omega^* = \mathbf{F}(\omega^*),$$

et, à partir d'initialisations proches, les trajectoires $\omega(t)$ convergent vers ω^* . Si l'attracteur est un **cycle** de période T , un ensemble de configurations $\{\omega^*(1), \dots, \omega^*(T)\}$ se répète où, après T applications de \mathbf{F} , l'état du **SCN** retrouve exactement sa valeur initiale, et les trajectoires voisines viennent s'**attacher** à ce cycle.

Un **exemple** en DSL pourrait être la formation définitive d'un cluster stable, ou bien une oscillation entre deux ou trois « configurations » récurrentes dans lesquelles les liaisons fortes changent tour à tour.

B. Détection et caractérisation numérique

1. Recherche d'un point fixe

En pratique, on itère la dynamique

$$\omega(t+1) = \mathbf{F}(\omega(t))$$

et on surveille l'évolution du **pas** :

$$\|\omega(t+1) - \omega(t)\|.$$

Si cette norme devient (et reste) inférieure à un seuil ε suffisamment petit, on conclut à l'approche d'un **point fixe**. Concrètement, on peut poser une condition comme

$$\|\omega(t+1) - \omega(t)\| < \varepsilon \quad \text{et} \quad \|\omega(t+\Delta t) - \omega(t)\| < \varepsilon \quad \text{sur quelques pas successifs,}$$

pour s'assurer qu'aucun ré-emballage ne se produit. On déclare alors la convergence vers un **état stationnaire** (attracteur fixe).

2. Recherche d'un cycle

Pour découvrir un cycle de période T , on enregistre les configurations $\omega(t)$ sur une certaine **fenêtre** de temps, par exemple à l'aide d'un buffer circulaire. Ensuite, on compare $\omega(t)$ et $\omega(t+T)$ via une norme appropriée, comme $\|\cdot\|$. Si, pour une valeur de T donnée, on observe que $\omega(t+T) \approx \omega(t)$ de manière répétée sur plusieurs valeurs de t , on en conclut l'existence d'une **périodicité** T .

Un cas particulier est la **période 2**, où le système oscille en **ping-pong** entre deux états. On vérifie cette propriété en s'assurant que

$$\omega(t+2) \approx \omega(t).$$

Dans des systèmes plus complexes, on peut chercher des traces d'**orbites** de période 3 ou plus, ce qui nécessite des tests plus élaborés.

3. Problème de la dimension dans un DSL

Un **SCN** implique $O(n^2)$ liaisons $\omega_{i,j}$ pour n entités, et la dimension de ω est potentiellement énorme. Pour détecter un **attracteur** numériquement, on restreint souvent l'analyse de deux manières :

- Soit on **échantillonne** un sous-ensemble représentatif de liaisons afin de suivre l'évolution des pondérations sans examiner l'ensemble du réseau.
- Soit on considère un **indicateur global**, tel que l'énergie $\mathcal{J}(\omega)$, la modularité ou d'autres métriques, permettant de repérer une **stagnation** ou une **périodicité**.

Ces approches restent des **heuristiques**, mais elles fonctionnent bien en pratique pour déterminer si le système s'est stabilisé en un **point fixe** ou s'il est entré dans un **cycle repérable**.

C. Bassins d'attraction : définition et intuition

Le **bassin d'attraction** associé à un attracteur A est l'ensemble de toutes les initialisations $\omega(0)$ dont la trajectoire finit par atteindre (ou approcher) A . Formulé plus formellement,

$$\mathcal{B}(A) = \{\omega_0 \mid \lim_{t \rightarrow \infty} \text{dist}(\omega(t), A) = 0\},$$

c'est-à-dire que toute trajectoire commençant dans $\mathcal{B}(A)$ aboutira dans A . Quand le DSL présente plusieurs attracteurs (points fixes ou cycles), chacun possède son **bassin** propre, et ces bassins se partagent l'espace initial via des **frontières** potentiellement complexes.

Savoir qu'un attracteur A existe est une chose, mais **savoir** dans quel bassin se trouve l'état initial $\omega(0)$ indique vers **quel** attracteur la dynamique ira. Deux initialisations très voisines peuvent se trouver dans deux bassins différents, entraînant l'aboutissement à deux **configurations** finales radicalement différentes. Cela met en évidence la **sensibilité** aux conditions initiales possible dans un DSL non linéaire.

D. Méthodes pratiques pour étudier bassins et attracteurs

Une façon simple et courante d'**explorer** les bassins d'attraction consiste à lancer de **nombreuses** simulations, chacune avec une initialisation $\omega(0)$ distincte (p. ex. tirée aléatoirement). On fait tourner la dynamique jusqu'à la détection d'un attracteur (point fixe ou cycle). On étiquette alors chaque initialisation par l'attracteur atteint et on essaie de visualiser (en réduction de dimension ou autrement) la zone initiale associée. On obtient ainsi un **maillage** approximatif du paysage des attracteurs.

Dans certains DSL, on dispose d'une **fonction d'énergie** $J(\omega)$ (chap. 2.4) qui, bien qu'elle ne soit pas toujours strictement un potentiel, peut fournir un repère. Si J descend globalement au fil des itérations, on remarque où elle se **stabilise** ou **oscille**. Les vallées de J correspondent en général à des points fixes ou des cycles. On peut alors repérer la "profondeur" de ces vallées et apprécier la **coexistence** d'attracteurs si plusieurs minima existent.

E. Conclusion

Les stratégies pour **détecter** un attracteur (ou un cycle) dans un **SCN** se ramènent à :

- **Observer** la trajectoire $\omega(t)$ et vérifier sa stagnation ou sa périodicité,
- **Explorer** plusieurs initialisations pour connaître les **bassins d'attraction**,
- **Mesurer** la stabilité (jacobienne locale, écart entre $\omega(t)$ et $\omega(t+1)$ ou $\omega(t+T)$, etc.).

L'idée-clé est que la **non-linéarité** d'un DSL ouvre la voie à de multiples attracteurs (points fixes, cycles, régimes complexes) et rend le comportement **dépendant** des conditions initiales et du **flux** (modifications de synergie, insertion d'entités). Cette pluralité d'attracteurs apporte un grand **potentiel** d'adaptation et de diversité, tout en nécessitant une **analyse** méthodique pour comprendre vers où la dynamique d'un SCN se dirige et dans quels **bassins** elle évolue.

4.3.3. Stabilisation vs. Dynamisme Continu

4.3.3.1. Scénario Stationnaire : Synergie Fixe, on Converge vers un Arrangement Stable

Le **Deep Synergy Learning** (DSL) suppose en général que la **synergie** $S(i, j)$ entre entités i et j peut évoluer si la représentation des entités change (embeddings, règles symboliques, etc.) ou si de nouvelles entités apparaissent. Cependant, il existe un **cas plus simple** où la **synergie** $S(i, j)$ reste **constante** au cours du temps, sans recalcul d'embeddings, sans ajout de nouveaux nœuds et sans modification des règles symboliques. On parle alors d'un **scénario stationnaire**, dans lequel la **fonction F** (décrivant l'évolution des pondérations ω) ne dépend que de ω elle-même et non d'autres facteurs changeants. Mathématiquement, on a un **système** :

$$\boldsymbol{\omega}(t+1) = \mathbf{F}(\boldsymbol{\omega}(t)),$$

où **F** demeure **invariante** dans le temps. Il en résulte qu'on peut étudier la **convergence** vers un état final $\boldsymbol{\omega}^*$ stable, sans se soucier de modifications extrinsèques de la synergie.

A. Équation de Mise à Jour sous Synergie Constante

Lorsqu'on parle de **scénario stationnaire**, la **synergie** $S(i, j)$ ne varie pas avec t . Cela signifie que, pour tout couple d'entités (i, j) , on a une valeur **fixe** $S(i, j)$. La **règle** DSL devient alors purement **dynamique** sur ω .

Un schéma particulièrement simple (chap. 4.2.1) est la mise à jour **additive** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

dans lequel η est un taux d'apprentissage et τ un coefficient de décroissance (pour éviter la croissance illimitée). Puisque $S(i, j)$ est **constant**, chaque $\omega_{i,j}$ s'ajuste en se rapprochant d'une "valeur-cible".

Si $S(i, j)$ est plus **grand** que $\tau \omega_{i,j}(t)$, on ajoute un incrément positif. Inversement, si $\omega_{i,j}(t)$ dépasse $S(i, j)/\tau$, le terme devient négatif.

On peut montrer qu'un **point fixe** se situe en $\omega_{i,j}^* = \frac{S(i, j)}{\tau}$. Une fois $\omega_{i,j}$ parvenu à cette valeur, l'update $\omega_{i,j}(t+1) - \omega_{i,j}(t)$ s'annule. Sur le plan **opérationnel**, cela signifie que la **force** de la liaison (i, j) atteint un équilibre où la synergie (qui tend à augmenter $\omega_{i,j}$) compense exactement la décroissance imposée par $\tau \omega_{i,j}$.

B. Convergence vers un Arrangement Stable

Dans l'ensemble, la **dynamique** stationnaire produit une **stabilisation** de la matrice $\{\omega_{i,j}\}$. Les liens se rapprochent de valeurs d'équilibre ($S(i, j)/\tau$) ou de 0 (si $S(i, j) = 0$). On parle alors d'un **arrangement stable** où plus rien ne bouge, car la synergie reste inchangée et la mise à jour **DSL** a abouti à un **point fixe**.

Sur chaque liaison $\omega_{i,j}$, si la règle est **additive** :

$$\omega_{i,j}(t+1) - \omega_{i,j}(t) = \eta [S(i,j) - \tau \omega_{i,j}(t)].$$

Cela se réécrit comme un **processus** de relaxation vers $\frac{S(i,j)}{\tau}$:

6. Tant que $\omega_{i,j}(t) < \frac{S(i,j)}{\tau}$, la différence est positive, $\omega_{i,j}$ **monte**.

7. Si $\omega_{i,j}(t) > \frac{S(i,j)}{\tau}$, la différence est négative, $\omega_{i,j}$ **baisse**.

Pourvu que $\eta \tau < 2$ et sans autres termes (ex. inhibition non linéaire), on obtient un **comportement contractant** qui converge.

Une fois **tous** les liens convergés, on dispose d'une **matrice** ω^* avec $\omega_{i,j}^* \approx \frac{S(i,j)}{\tau}$. Les **entités** i et j pour lesquelles $S(i,j)$ est élevée deviennent fortement liées, celles pour lesquelles $S(i,j) \approx 0$ ont des $\omega_{i,j} \approx 0$. Si l'on applique un **seuil** ou une forme d'inhibition, on peut voir émerger des **clusters** plus marqués.

C. Interprétation en Termes de Clusters

Dans un SCN stationnaire, un jeu de **données** (ou de “synergies” prédéterminées) aboutit à une **partition** implicite. Les liens **forts** à l'équilibre forment des **groupes** d'entités fortement interconnectées.

Si la **matrice** $\{S(i,j)\}$ reflète une structure de type “clusters” (ex. entités proches en embedding), alors la **dynamique additive** fait en sorte que les entités internes à un cluster se retrouvent avec des **liens** $\omega_{i,j}$ élevés. Entre clusters, si $S(i,j)$ est faible, $\omega_{i,j}$ reste bas.

Ce phénomène s'apparente à une forme de **clustering** non supervisé (cf. chap. 2), où le résultat final ω^* exprime la **partition** latente.

Si la règle de mise à jour est **linéaire**, comme ci-dessus, le **point fixe** est souvent unique et découle aisément de $\frac{S(i,j)}{\tau}$. Mais dès qu'on ajoute un **terme** d'inhibition plus complexe (compétition entre liens) ou des non-linéarités, il peut exister **plusieurs** attracteurs stables (chacun correspondant à un arrangement cluster différent). Dans ce cas, le SCN stationnaire peut se figer dans l'un ou l'autre arrangement selon l'initialisation ou des bruits aléatoires (cf. chap. 4.3.2 sur la multiplicité d'attracteurs).

D. Liens Formels avec la Descente d'Énergie

Dans la plupart des **implémentations** DSL, on peut définir une **énergie** $\mathcal{J}(\omega)$ (ex. $-\sum_{i,j} \omega_{i,j} S(i,j) + \tau/2 \sum_{i,j} (\omega_{i,j})^2$) qui, en régime stationnaire, atteint un **minimum local**. Dans ce scénario, la synergie $S(i,j)$ ne changeant pas, la **descente** de \mathcal{J} n'est pas contrariée par des révisions de synergie, et le SCN **aboutit** à un minimum local stable (donc un arrangement final ω^*).

On dit que $\mathcal{J}(\omega)$ est **fixe** car la fonction $\omega \mapsto \mathcal{J}$ ne dépend pas d'autres variables évolutives. La **descente** locale de \mathcal{J} se fait par une dynamique de relaxation, menant au point d'équilibre.

Si on n'ajoute pas de **bruit** ou de **compétition** trop forte, la convergence est généralement assurée (chaque liaison converge vers $S(i,j)/\tau$). En revanche, des schémas multiplicatifs ou inhibiteurs peuvent générer des **oscillations** transitoires avant la stabilisation. Mais in fine, dans un cadre stationnaire, on s'**installe** dans un attracteur.

Conclusion (4.3.3.1)

Dans un **scénario** où la **synergie** $S(i,j)$ demeure **fixe** (aucune mise à jour d'embeddings, aucune apparition de nouveaux nœuds), la **dynamique** DSL (Deep Synergy Learning) se ramène à un **système** autonome sur ω . Cela aboutit typiquement à un **arrangement stable** où chaque liaison $\omega_{i,j}$ converge vers une valeur d'équilibre, par exemple $S(i,j)/\tau$ pour le modèle additif, et la **matrice** ω cesse d'évoluer. On obtient alors une **partition** ou un **réseau** stationnaire mettant en évidence les affinités structurelles dictées par S .

Cette situation **contraste** avec les **cas dynamiques** (voir § 4.3.3.2) où la **synergie** peut se redéfinir au fil du temps. Dans le scénario stationnaire, on observe une **descente d'énergie pure** ou une **relaxation**, sans que les entrées S ne perturbent l'équation au fur et à mesure. Le SCN se **fige** sur un “**arrangement stable**”, révélant la configuration la plus compatible avec la synergie statique.

4.3.3.2. Scénario Évolutif : synergie modifiée par de nouvelles données / événements, le réseau ne se fige jamais (apprentissage continu)

Dans la section précédente (4.3.3.1), l'analyse d'un **SCN** (Synergistic Connection Network) en **scénario stationnaire** (avec une fonction de synergie $S(i,j)$ fixée) montrait comment le réseau peut converger vers un **arrangement stable**. Cependant, dans bien des cas pratiques, cette hypothèse de synergie *constante* s'avère trop restrictive. En effet, dans un système réel, de nouvelles entités peuvent faire leur apparition, des attributs ou des représentations peuvent évoluer (embeddings ajustés, règles modifiées), et l'environnement lui-même peut changer. Ainsi, la **fonction** $S(i,j)$ entre deux entités \mathcal{E}_i et \mathcal{E}_j devient **dépendante du temps**, ou se voit “redéfinie” à chaque étape. On se retrouve alors dans un régime d'**apprentissage continu**, où le **réseau** ne parvient pas forcément à une configuration finale unique et peut rester en **réorganisation** perpétuelle.

A. Origines de l'Évolution de la Synergie

Dans un **DSL** fonctionnant en mode **flux** (*stream*), de nouveaux nœuds \mathcal{E}_{n+1} apparaissent régulièrement, par exemple sous la forme de nouveaux documents, de nouveaux utilisateurs ou encore de nouveaux capteurs. Chaque fois qu'un tel nœud est inséré, il est nécessaire de :

8. **Calculer ou initialiser** les liens $\omega_{(n+1),i}$ pour $i = 1, \dots, n$.
9. **Mettre à jour ou réévaluer** la **synergie** $S(n+1, i)$.

Ces opérations peuvent **rompre l'équilibre local** déjà en place où, en liant \mathcal{E}_{n+1} à divers nœuds, on réoriente potentiellement la structure des **clusters** ou des **communautés** formées jusque-là.

Même si le nombre de nœuds reste le même, la *représentation* d'un nœud peut changer. Par exemple, un *embedding* sub-symbolique \mathbf{x}_i se re-entraîne et prend de nouvelles valeurs, modifiant les **distances** ou **similarités** $\|\mathbf{x}_i - \mathbf{x}_j\|$ ou encore $\mathbf{x}_i \cdot \mathbf{x}_j$. De même, un nœud symbolique peut ajouter ou retrancher une règle (logique), entraînant un **changement** dans la fonction $S_{\text{sym}}(i, j)$. On assiste alors à une **synergie** redéfinie en cours de route, ce qui oblige la mise à jour $\omega_{i,j}$ à se poursuivre en s'adaptant à cette nouvelle donne.

Dans des scénarios temps réel (ex. robotique, environnement capteur), les conditions (luminosité, bruit, configuration spatiale, etc.) varient de façon incessante. La synergie $S(i, j, t)$ — qui peut refléter la cohérence instantanée de deux signaux sensoriels — doit être recalculée en continu, empêchant le SCN de “figer” un arrangement final.

B. Impact sur la Dynamique du Réseau

Contrairement au **cas stationnaire** où l'on peut exhiber un point fixe ω^* , ici la loi

$$\boldsymbol{\omega}(t+1) = \mathbf{F}(\boldsymbol{\omega}(t), S(\cdot, \cdot, t))$$

est *non autonome*. On ne peut plus parler de $\boldsymbol{\omega}^* = \mathbf{F}(\boldsymbol{\omega}^*, S(\cdot, \cdot))$ de manière permanente, car S varie à chaque étape. En conséquence, le réseau peut évoluer *indéfiniment* :

- Il peut se reconfigurer au gré des insertions de nœuds,
- Il peut réajuster ses liens quand un embedding \mathbf{x}_i change,
- Il peut basculer d'un type de partition à un autre selon l'arrivée de données ou l'évolution d'un paramètre (inhibition, température de recuit, etc.).

Cette logique rejoint l'idée d'un **apprentissage en ligne** où, à chaque itération ou petit batch, on :

- **Observe** les changements, qu'il s'agisse de nouveaux nœuds ou de modifications d'embeddings,
- **Recalcule** au moins partiellement la fonction $S(i, j, t)$,
- **Réalise** une mise à jour incrémentale (*delta update*) $\Delta\omega_{i,j}(t)$ pour ajuster la pondération et maintenir une nouvelle cohérence.

Le **SCN** reste ainsi *flexible* et *vivant*, réorientant ses clusters ou sous-structures au fil du temps.

Du fait que S n'est plus constant, le **SCN** peut parfois **osciller**, par exemple lorsqu'un nœud voit sa synergie vis-à-vis d'un cluster se renforcer temporairement, puis s'affaiblir, puis se renforcer à nouveau, entraînant une dynamique cyclique dans la structure du réseau. On peut assister à des “phases” transitoires de stabilisation, suivies d'une “transition” quand une nouvelle *vague* de données survient.

C. Considérations Mathématiques : Systèmes Dynamiques Non Autonomes

Le système s'écrit

$$\boldsymbol{\omega}(t+1) = \mathbf{F}(\boldsymbol{\omega}(t), \mathbf{p}(t)),$$

où $\mathbf{p}(t)$ dénote la configuration (paramètres, représentations) à l'instant t . Les propriétés de stabilité ou de convergence dépendent du *trajectoire* $\{\mathbf{p}(t)\}$. Si $\mathbf{p}(t)$ varie lentement, on peut espérer un “quasi-équilibre” mobile ; si elle varie rapidement, la $\omega(t)$ peut rester en turbulence, sans se poser.

On peut mobiliser des cadres “skew-product” en théorie des systèmes dynamiques non autonomes, ou des approches d'**adaptation** en contrôle. En pratique, la plupart des implémentations reposent sur des *simulations*, où l'on met à jour $\omega_{i,j}(t)$ de façon incrémentale, tout en recomputant $S(\cdot, \cdot, t)$. On observe l'existence ou non d'un “pseudo-ordre”, de *phases* où le réseau se stabilise partiellement, puis se réadapte, etc.

Dans un système de recommandation, chaque itération introduit un *nouvel utilisateur* ou un *nouvel item*. La synergie $S(u, i)$ entre l'utilisateur u et l'item i doit être définie (ou recalculée). Cela conduit à des réajustements $\omega_{u,i}$, puis redistribue la force sur d'autres liens déjà existants. Cette dynamique se poursuit indéfiniment, car la base de données utilisateurs/items ne cesse d'évoluer.

D. Applications Pratiques

Recommandation en Ligne.

On ne peut jamais finaliser un “training” une fois pour toutes, car l'arrivée constante d'utilisateurs ou d'items (ou la modification des préférences) rend la matrice ω obsolète si elle n'est pas ajustée. Le **DSL** apporte une **plasticité native** où chaque nouveau flux de données modifie la synergie S , permettant au **SCN** de réapprendre et de réorganiser les **clusters**, tout en restant opérationnel.

Robotique.

Les flux sensoriels évoluent (conditions d'éclairage, types d'objets manipulés), les capteurs ou actions peuvent se paramétrier. Le SCN recalcule la *pertinence* des liens sensorimoteurs. Il n'existe pas d'état final, seulement une suite d'états adaptatifs.

Dialogue.

Un chatbot perfectionne ses embeddings (BERT, GPT) à mesure qu'il ingère de nouvelles conversations. La *synergie textuelle* S n'est donc pas fixe. Le SCN en charge d'associer des thèmes ou des intentions réorganise ses liaisons ω en continu, parfois sans stabilisation.

4.3.3.3. Exemples de courbes d'évolution $\omega_{i,j}(t)$ montrant la formation progressive d'un cluster

Pour mieux **visualiser** ce qui se passe durant la **dynamique** d'auto-organisation, il est souvent instructif de tracer au fil du temps l'**évolution** de quelques pondérations $\omega_{i,j}(t)$. Dans un **SCN** (Synergistic Connection Network) à grande échelle où n est élevé, on se limite généralement à un sous-ensemble représentatif de liens ou à des **indicateurs** agrégés, mais le principe reste le même : observer comment certains liens $\omega_{i,j}(t)$ passent de valeurs proches de 0, ou très faibles, à des niveaux élevés voire saturés, tandis que d'autres se réduisent ou stagnent à des valeurs négligeables.

L'idée est de **montrer la formation progressive** d'un **cluster** où, au départ, les liens sont dispersés ou quasi nuls, puis la dynamique sélectionne progressivement certains couples (i,j) qui se renforcent, révélant ainsi un **groupe** cohérent émergent.

A. Scénario simple à cinq entités

Considérons un exemple **pédagogique** où l'on dispose de cinq entités $\{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4, \mathcal{E}_5\}$. Supposons qu'en **réalité** la **synergie** $S(i,j)$ soit nettement **élevée** entre les paires du **sous-groupe** $\{1,2,3\}$ et plus **faible** pour leurs liens avec $\{4,5\}$. Autrement dit, il existe un "vrai" petit cluster $\{1,2,3\}$, alors que $\{4,5\}$ en est assez distant.

Pour amorcer la simulation, on pose :

$$\omega_{i,j}(0) \approx 0 \quad (\text{avec éventuellement un faible bruit aléatoire}).$$

Cette **condition initiale** illustre l'idée que, avant toute itération, les pondérations $\omega_{i,j}$ sont de niveau très bas (ou nuls) ; la dynamique du **DSL** va alors opérer progressivement la mise en place de liens plus ou moins forts.

Lorsque l'on **applique** la règle de mise à jour (par exemple la variante additive) :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

les paires $(1,2)$, $(1,3)$ et $(2,3)$ commencent à s'**amplifier** si $S(1,2)$, $S(1,3)$, et $S(2,3)$ sont élevés. En pratique, dès les premières itérations, on observe une **croissance** notable de $\omega_{1,2}(t)$, $\omega_{1,3}(t)$, $\omega_{2,3}(t)$. La dynamique va rapidement les rapprocher de leur "point d'équilibre" $\frac{S(i,j)}{\tau}$ (hypothèse sans inhibition).

En revanche, pour les paires mixtes (c'est-à-dire entre $\{1,2,3\}$ et $\{4,5\}$), la valeur de la synergie $S(i,j)$ est plus faible. De ce fait, les mises à jour successives :

$$\Delta \omega_{i,j}(t) = \eta [S(i,j) - \tau \omega_{i,j}(t)]$$

restent proches de zéro, voire négatives si $\omega_{i,j}$ a eu un léger sursaut de bruit initial. Les liens $(1,4)$, $(2,5)$, etc., finissent donc par **stagner** à un niveau très faible ou décroître pour s'approcher de zéro.

Pour **visualiser** cette progression, on peut tracer, à chaque itération t , la valeur de $\omega_{i,j}(t)$ pour diverses paires (i,j) . Un exemple :

- $\omega_{1,2}(t)$: part d'environ 0 au temps $t = 0$, **monte** assez vite dans les dix ou vingt premières itérations, puis **se stabilise** dans une région proche de $S(1,2)/\tau$.
- $\omega_{1,4}(t)$: demeure très faible ; si le bruit initial la poussait légèrement, la correction $\eta [S(1,4) - \tau \omega_{1,4}(t)]$ devient vite négative ou quasi nulle, la ramenant vers zéro.

Cette différence de **trajectoire** entre les paires intra-cluster $\{1,2,3\}$ et les paires hors-cluster, comme $\{1,4\}$ ou $\{2,5\}$, révèle le mécanisme **sélectif** où une synergie forte enclenche un **renforcement local** des liens, tandis qu'une synergie faible conduit à leur **quasi-extinction**.

En examinant les **graphes** $\omega_{i,j}(t)$ en fonction du temps, on distingue alors deux “familles” de courbes :

- **Famille haute** : $\omega_{1,2}(t), \omega_{1,3}(t), \omega_{2,3}(t)$. Ces liens croissent puis se maintiennent à un plateau élevé, typique de la **cohésion** interne au cluster {1,2,3}.
- **Famille basse** : $\omega_{1,4}(t), \omega_{2,5}(t), \omega_{3,4}(t)$, etc. Celles-ci restent très proches de **zéro** ou se tassent rapidement, montrant la **séparation** d’avec {4,5}.

Après un certain nombre d’itérations, on obtient une **matrice ω** où {1,2,3} s’affiche comme un “**micro-bloc**” densément interconnecté, tandis que {4,5} n’y est lié que par des poids très bas (ou à un autre cluster si {4,5} se rapprochent entre eux).

Lorsque la **dynamique** se stabilise (ou atteint un état stationnaire), on a, pour les paires intra-{1,2,3}, des valeurs $\omega_{i,j}^* \approx \frac{S(i,j)}{\tau}$ (si le modèle est purement additif, sans inhibition). Entre {1,2,3} et {4,5}, on constate $\omega \approx 0$.

Dans un **SCN** plus vaste, on observe la **même** logique se reproduire sur des groupes plus grands où, si la synergie est suffisante, ces groupes émergent comme **clusters** plus ou moins compacts, et la visualisation des pondérations confirme une structure en **blocs**.

Cet **exemple** simplifié illustre la **progression** d’un **DSL** qui part d’une initialization “faible” $\omega(0) \approx 0$:

- Les **liens** correspondant à de **fortes** synergies se **renforcent**,
- Les autres se **réduisent** ou restent proches de zéro,
- Au bout d’un nombre d’**itérations**, on **observe** clairement la formation d’un ou plusieurs **clusters** (ici, {1,2,3}).

C’est en examinant les **courbes** $\omega_{i,j}(t)$ que l’on **saisit** la nature dynamique de la convergence où chaque liaison évolue selon la différence $[S(i,j) - \tau \omega_{i,j}(t)]$. Ce **principe** se généralise à de plus grands ensembles avec des patterns plus complexes, mais la mécanique reste la même : une synergie forte $S(i,j)$ entraîne un **renforcement**, tandis qu’une synergie faible conduit à une **quasi-extinction** des liaisons, aboutissant ainsi à une **auto-organisation** en groupes cohérents.

B. Exemples de variations en cas d’inhibition ou de multiplicatif

Dans le cadre du **DSL** (Deep Synergy Learning), l’évolution des pondérations $\omega_{i,j}(t)$ peut prendre des profils temporels sensiblement différents, notamment lorsqu’on introduit des **mécanismes** tels que l’**inhibition compétitive** ou une **mise à jour multiplicative**. Il s’ensuit des **dynamiques** qui peuvent produire des comportements parfois plus contrastés, allant de “va-et-vient” marqués à de véritables “sauts” exponentiels.

Lorsque l’on ajoute de l’**inhibition latérale** (voir la section 4.2.2.2) dans la règle de mise à jour, on introduit un **terme** $-\gamma \sum_{k \neq j} \omega_{i,k}(t)$ qui pénalise la croissance conjointe de plusieurs liens partant de la même entité \mathcal{E}_i . La mise à jour $\omega_{i,j}(t+1)$ peut alors s’écrire :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta \left[S(i,j) - \tau \omega_{i,j}(t) - \gamma \sum_{k \neq j} \omega_{i,k}(t) \right],$$

où $\gamma > 0$ représente l'intensité de l'inhibition.

Sur le plan dynamique, si un lien $\omega_{i,2}$ se met à croître rapidement (parce que la synergie $S(i, 2)$ est forte), il “inhibe” ou freine la croissance d'autres liens partant de la même entité \mathcal{E}_i . Les courbes $\omega_{i,j}(t)$ peuvent alors révéler un **comportement compétitif** où, tandis que $\omega_{i,2}$ se renforce, $\omega_{i,3}$ ou $\omega_{i,5}$ stagnent ou diminuent, même si leur synergie $S(i, 3)$ ou $S(i, 5)$ n'est pas complètement négligeable.

Il est possible d'observer des **va-et-vient** où, à certains moments, $\omega_{i,2}(t)$ progresse, puis si la synergie $S(i, 3)$ n'est pas trop faible, $\omega_{i,3}(t)$ profite d'un **relâchement d'inhibition** pour se renforcer à son tour, ce qui à nouveau rabaisse $\omega_{i,2}$, et ainsi de suite, entraînant une oscillation entre les connexions. Ces oscillations (ou successions de prises d'avantage) sont plus ou moins marquées selon les valeurs de γ , η et τ .

Sur un graphe “ $\omega_{i,j}(t)$ vs. t ”, on discerne clairement des courbes qui croissent, puis s'infléchissent sous l'action de l'inhibition, favorisant l'émergence d'un “gagnant” local. Cette situation peut conduire à des clusters plus nets (car un nœud \mathcal{E}_i n'entretient qu'un faible nombre de liens forts), mais aussi à des **phases de compétition** prolongées.

La **règle multiplicative** (voir la section 4.2.2.1) modifie la façon dont un lien $\omega_{i,j}$ se met à jour, en introduisant une **proportionnalité** à sa valeur actuelle. Au lieu d'ajouter un terme $\eta[S(i,j) - \tau \omega_{i,j}(t)]$ (cas additif), on applique souvent une formule de la forme :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) \times (1 + \eta S(i,j)) - \beta \omega_{i,j}(t),$$

ou même une exponentiation directe (par exemple $\omega_{i,j}(t+1) = \omega_{i,j}(t) \exp\{\alpha[S(i,j) - \tau \omega_{i,j}(t)]\}$).

Une caractéristique majeure de la mise à jour multiplicative tient à sa **dépendance** au niveau actuel de $\omega_{i,j}$. Si un lien est déjà modérément fort, la rétroaction positive agit de manière accélérée (“the rich get richer”). Inversement, si $\omega_{i,j}$ est initialement très faible, il peut rester collé à presque zéro, car l'incrément dépend de sa valeur courante.

Les simulations montrent souvent des **sauts exponentiels** où une liaison $\omega_{i,j}$ peut demeurer quasi nulle pendant plusieurs itérations, puis, dès qu'un léger bruit ou une fluctuation l'élève au-dessus d'un petit **seuil**, elle monte subitement de façon exponentielle jusqu'à atteindre un **plateau** ou une **saturation** ω_{\max} . Sur un graphe “ $\omega_{i,j}(t)$ vs. t ”, la courbe paraît stable proche de 0, puis effectue un **bond** rapide.

Ce type de mise à jour produit des **clusters** très tranchés où, une fois qu'un lien est enclenché parce qu'un nœud a franchi le **seuil de synergie**, il grandit fortement, reléguant d'autres **liaisons concurrentes** à un niveau bas. Dans un DSL, on peut donc aboutir à des partitions plus brutales où quelques liens dominent massivement, tandis que d'autres restent quasi inexistantes.

Imaginons que $\omega_{1,2}(t)$ dépasse 0.02 à l'itération $t = 10$ (à cause d'une petite synergie favorable). La multiplicativité enclenche alors une **croissance** en quelques pas, menant $\omega_{1,2}$ à un niveau élevé (0.7, 0.8...). Dans le même temps, $\omega_{1,3}$ ou $\omega_{2,4}$ ne franchissent pas leur seuil, de sorte qu'ils demeurent sous 0.01. Le cluster “(1,2)” se forme ainsi quasi instantanément, tandis que (1,3) reste “ignoré”.

C. Mise en forme de ces courbes

Lorsqu'on souhaite **visualiser** l'évolution des pondérations $\omega_{i,j}(t)$ au cours des itérations, on peut procéder en plusieurs étapes afin de tirer le meilleur parti des **courbes** tracées. Les sous-sections suivantes décrivent ce processus et mettent en évidence son intérêt pour **comprendre** la dynamique d'**auto-organisation** du réseau.

Un **premier** aspect consiste à **stocker** les valeurs de chaque liaison $\omega_{i,j}$ à chaque itération $t = 0, 1, \dots, T$. Dans la pratique, on pourra :

- Définir un tableau ou un dictionnaire pour les $\omega_{i,j}(t)$.
- Après chaque mise à jour, on mémorise $\omega_{i,j}(t)$ pour un petit nombre de paires (i,j) d'intérêt (pour ne pas trop alourdir la mémoire si le réseau est grand).
- À l'issue des itérations (par exemple au bout de 50, 100 ou 500 pas), on dispose alors de la **trajectoire** complète $\omega_{i,j}(0), \omega_{i,j}(1), \dots, \omega_{i,j}(T)$.

Pour **sélectionner** les paires (i,j) à tracer, on identifie généralement :

- **Une paire** à forte synergie a priori (ou bien dont on anticipe qu'elle va se renforcer si les entités sont très similaires).
- **Une paire** à synergie moyenne (où l'on veut vérifier si le lien monte ou descend).
- **Une paire** à synergie faible (pour constater si $\omega_{i,j}$ demeure au voisinage de 0 ou s'éteint).

On peut ainsi **comparer** les comportements de quelques liaisons représentatives, sans tracer la totalité des $O(n^2)$ courbes du SCN.

Après avoir récupéré ces **trajectoires** $\omega_{i,j}(t)$, un usage courant est de produire un **graphique** (par exemple via matplotlib ou un outil similaire en Python) où l'axe horizontal représente l'itération t et l'axe vertical la valeur de $\omega_{i,j}(t)$.

On peut tracer **plusieurs** liaisons sur un **même** diagramme :

$$\omega_{1,2}(t), \quad \omega_{1,3}(t), \quad \omega_{4,5}(t), \quad \dots$$

Ces différentes **courbes** de couleur distincte illustrent alors, sur une même figure, comment certains liens **montent** rapidement quand la synergie est forte (et donc la mise à jour DSL les renforce), tandis que d'autres **restent** très bas ou **progressent** lentement.

Cette **représentation** multi-lignes met en évidence :

- Les **phases** d'augmentation : par exemple, une courbe qui s'élève dès les 5 premières itérations, suggérant une forte cohésion entre les entités correspondantes.
- Les **paliers** ou saturations : il arrive qu'après un certain temps, une courbe se fixe autour d'une valeur (ex. $\omega_{1,2}(t) \rightarrow 0.8$), marquant la stabilisation d'un lien fort.
- Les **variations plus erratiques** ou "yoyo" : elles indiquent soit des oscillations (chap. 4.3.2.2), soit un changement de la fonction synergie (scénario évolutif, chap. 4.3.3.2).

En examinant ces **trajectoires** de $\omega_{i,j}(t)$, on obtient une **indication** très claire de la situation dynamique :

- Si, au bout d'un nombre raisonnable d'itérations (par exemple 50 ou 100), les **valeurs** $\omega_{i,j}(t)$ montrent toutes un **plateau** ou une zone presque stationnaire, c'est un signe fort de **convergence** (scénario stationnaire du chap. 4.3.3.1). On peut alors considérer que le **SCN** a "appris" sa structure finale.
- À l'inverse, si certaines courbes **oscillent** (montées et descentes successives) ou **changent** de palier à plusieurs reprises, cela indique soit un **phénomène oscillatoire** au sens du chap. 4.3.2.2, soit un **scénario évolutif** (chap. 4.3.3.2) où la synergie S se modifie (arrivée de nouvelles entités, rafraîchissement des embeddings, etc.), empêchant un état unique de s'installer définitivement.

Cette **distinction** (plateau vs. oscillations/progression continue) s'avère cruciale pour **comprendre** le régime dans lequel le **SCN** se situe :

- **Régime stationnaire** : la topologie des liens se "fige" et on peut alors extraire de manière stable les clusters.
- **Régime non stationnaire** : les courbes ne se stabilisent pas, reflétant un **réseau** "vivant" ou "en apprentissage continu", potentiellement réactif aux nouveaux flux de données.

Un **bénéfice** direct de tracer ces courbes est d'observer, "pas à pas", **comment** un cluster se **construit** dans le SCN.

Si l'on voit **plusieurs** liaisons $\omega_{i,j}(t)$ associées à un **même** groupe d'entités \mathcal{C} (par exemple $\{1,2,3\}$) **augmenter** quasi simultanément, c'est la **marque** qu'un bloc se **renforce**. Dans le même temps, les liens de ces entités vers l'extérieur peuvent **stagner** ou **décroître**.

CCela illustre de manière **graphique** le phénomène d'**isolement** d'un cluster où $\omega_{1,2}(t)$ et $\omega_{1,3}(t)$ deviennent grandes, tandis que $\omega_{1,k}$ pour $k \notin \{2,3\}$ restent faibles. La dynamique DSL favorise les "intra-liens" cohérents et punit (ou néglige) les "extra-liens" divergents.

Exemple d'évolution

- **Étapes initiales** : $\omega_{i,j}(0) \approx 0$. Pendant quelques itérations, toutes les courbes sont proches de zéro.

- **Montée** : pour les paires à haute synergie, leurs courbes **décollent** plus ou moins vite (selon η, τ) et peuvent atteindre une zone de plateau intermédiaire (ex. 0.5–0.7).
- **Stabilisation** : si rien ne change, on voit un palier final (par ex. 0.8) pour ces liens forts, alors que les liens “faibles” conservent une valeur modeste (0.1 ou <0.05).
- **Cluster** : on interprète $\omega_{1,2}, \omega_{1,3}, \omega_{2,3}$ comme hautes, signe d'un **cluster** {1,2,3}. Les liaisons de ces entités vers d'autres indices {4,5, ... } ne dépassant pas 0.15, on obtient un “mur” séparant {1,2,3} du reste.

Selon la **présence** ou non d'inhibition (compétition pour les liaisons sortantes), quelques **tweaks** peuvent survenir :

- *Inhibition latérale* : force un **choix** plus exclusif, où un lien se stabilise un peu au détriment d'un autre. $\omega_{1,3}$ peut donc baisser de 0.5 à 0.3 si $\omega_{1,2}$ grimpe à 0.8, etc.
- *Saturation ou clipping* : empêche une courbe de dépasser ω_{\max} . Visuellement, on voit la courbe buter sur une valeur-limite.

Au final, la **lecture** de ces tracés $\omega_{i,j}(t)$ donne une “**histoire**” de la construction d'un cluster, au lieu de juste observer l'état final $\omega_{i,j}(T)$.

Cette **analyse** offre en outre un **témoignage** direct de l'émergence d'un **cluster** où l'on voit concrètement *quand* et *comment* les liaisons associées à un groupe s'élèvent en parallèle, apportant ainsi une **compréhension dynamique** du processus d'**auto-organisation** au sein du **SCN**.

4.4. Oscillations, Pseudo-Chaos et Méthodes de Contrôle

Les sections précédentes (4.3) ont montré que la **dynamique** d'un SCN (Synergistic Connection Network) pouvait, dans certains cas, conduire à une **stabilisation** (formation de clusters fixes), tandis que dans d'autres, elle pouvait demeurer en mouvement (scénario évolutif). Mais il existe une autre possibilité où le réseau entre dans des **oscillations** plus ou moins régulières ou même un régime "pseudo-chaotique". Dans ce chapitre 4.4, nous allons :

- **Analyser les causes** de ces oscillations ou de comportements complexes (4.4.1),
- **Proposer des stratégies** pour les **rompre** ou les **contrôler** (4.4.2) lorsque le but est de favoriser la convergence,
- **Présenter** enfin quelques **cas stochastiques** (4.4.3) où l'injection de bruit (recuit simulé, perturbations) peut aider le SCN à échapper à des minima ou à se stabiliser de façon plus globale.

Ce thème s'inscrit dans la continuité de la question des **attracteurs** (4.3.2) où, au-delà des **points fixes** ou des **cycles simples**, on peut observer des régimes plus complexes comme des **oscillations prolongées** ou des formes de **pseudo-chaos**, lorsque la rétroaction est suffisamment puissante et que la **non-linéarité** du système est marquée.

4.4.1. Pourquoi des Oscillations ou un Pseudo-Chaos ?

Malgré la tendance naturelle à la stabilisation (lorsqu'un terme $-\tau \omega_{i,j}$ freine la croissance), il peut arriver que la **mise à jour** des pondérations $\omega_{i,j}$ entretienne un **cycle** ou un **mouvement** quasi cyclique. De plus, dans des réseaux de forte dimension et dotés de mécanismes non linéaires, on observe parfois des régimes encore plus complexes, assimilables à du **pseudo-chaos** (une sensibilité marquée aux conditions initiales, une trajectoire ne se répétant pas exactement mais restant dans un comportement "erratique").

4.4.1.1. Rétroactions Positives Trop Intenses, Absence d'Inhibition (2.3.2.2)

Principe Général : un Effet de Boule de Neige

Dans les **formulations** de la mise à jour (qu'elles soient additives ou multiplicatives, voir chapitre 4.2), il est fréquent de disposer d'un terme du type $+\eta S(i,j)$ qui **encourage** la croissance du lien $\omega_{i,j}$ dès lors que la **synergie** $S(i,j)$ est élevée. Une fois que ce renforcement positif l'emporte sur la décroissance $-\eta \tau \omega_{i,j}$ (parce que τ est trop faible ou parce que $S(i,j)$ dépend déjà de $\omega_{i,j}$ de manière auto-amplificatrice), on peut observer un **effet boule de neige** :

$$\omega_{i,j}(t+1) > \omega_{i,j}(t) > \dots \text{ pendant plusieurs itérations.}$$

Autrement dit, chaque itération accentue encore plus la croissance d'un lien déjà en essor, de sorte que $\omega_{i,j}$ s'élève presque sans limite. Cette **dynamique** peut conduire à un **emballlement** des pondérations et, par ricochet, à l'émergence de **cycles** ou d'**oscillations**, surtout si d'autres liens liés à $\omega_{i,j}$ se trouvent également encouragés à croître.

Absence d'Inhibition et Instabilités

Lorsqu'on n'a **pas** incorporé de mécanismes d'inhibition compétitive (chap. 4.2.2.2) ou de saturation dans la mise à jour, plusieurs liens $\omega_{i,j}$ peuvent simultanément connaître cette croissance soutenue. On se retrouve alors avec un **blocage**, où la dynamique entre dans un **renforcement collectif** où chaque lien élevé en entraîne un autre, modifiant la synergie d'autres paires et perpétuant ainsi une boucle de rétroaction qui peut empêcher la stabilisation du système. Le réseau peut **basculer** successivement d'un groupement à un autre (un cluster domine, puis un autre s'impose), s'inscrivant dans un **cycle** de transitions ou de résonances.

Dans un **système** purement linéaire, la présence d'un **gain** trop important sur les boucles de rétroaction positive suscite des phénomènes de **résonance** ou de divergence. Dans un **système** non linéaire, on peut voir apparaître des attracteurs cycliques ou chaotiques, témoignant d'un équilibre impossible à stabiliser.

Exemple de Rétroaction Positive

Considérons un **mini-système** à deux pondérations ω_1 et ω_2 . On suppose :

$$\omega_1(t+1) = \omega_1(t) + \eta [S_1(\omega_2(t))\tau \omega_1(t)], \quad \omega_2(t+1) = \omega_2(t) + \eta [S_2(\omega_1(t))\tau \omega_2(t)].$$

Si $S_1(\omega_2)$ et $S_2(\omega_1)$ sont des **fonctions croissantes** (par exemple, $S_1(\omega_2) = a \omega_2$ avec a grand), alors la croissance de ω_2 **stimule** ω_1 , et la croissance de ω_1 **stimule** ω_2 . Sans **inhibition**, ce couplage conduit aisément à un **cycle** (un lien monte pendant que l'autre descend, puis vice versa) ou à une **explosion** de leurs valeurs. Dans d'autres cas, on obtient des oscillations autour d'un point ou d'une courbe, et ces oscillations peuvent ne jamais s'amortir si τ est trop faible ou si $\eta S_1, S_2$ sont trop grands.

Ce **phénomène** explique l'**emballlement** de certaines pondérations dans le réseau si rien n'est là pour les contenir (inhibition, saturation, etc.). Une fois enclenché, la rétroaction positive se nourrit d'elle-même et empêche le système de revenir vers un état stable.

Analyse Mathématique d'Exemple Simplifié

Prenons deux pondérations ω_1, ω_2 avec mise à jour :

$$\begin{cases} \omega_1(t+1) = \omega_1(t) + \eta [A \omega_2(t) - \tau \omega_1(t)], \\ \omega_2(t+1) = \omega_2(t) + \eta [B \omega_1(t) - \tau \omega_2(t)]. \end{cases}$$

Ici, $A, B > 0$ modélisent la rétroaction positive, et $\tau \omega_i(t)$ la décroissance. Les points fixes ω_1^*, ω_2^* vérifient :

$$\begin{cases} A \omega_2^* - \tau \omega_1^* = 0, \\ B \omega_1^* - \tau \omega_2^* = 0, \end{cases}$$

d'où $\omega_1^*(AB - \tau) = 0$. On voit qu'il existe un **point fixe non trivial** seulement si $AB = \tau$. La stabilité dépendra de la jacobienne :

$$J = \begin{pmatrix} 1 - \eta\tau & \eta A \\ \eta B & 1 - \eta\tau \end{pmatrix}.$$

Si les **valeurs propres** de J sortent du cercle unité, ω_1, ω_2 peuvent **osciller** ou diverger, engendrant cycles ou chaos.

Dans un **SCN** à grande dimension, ce mécanisme s'étend et des **oscillations** complexes ou pseudo-chaotiques peuvent s'installer si la rétroaction positive est prépondérante ou si aucune forme d'inhibition globale ne vient canaliser la croissance.

Conclusion Partielle : Nécessité d'un Frein ou d'une Inhibition

Lorsque la **rétroaction positive** est **excessive**, le SCN risque de développer des **oscillations** ou un **emballlement** où certains liens se rehaussent indéfiniment, souvent en boucle avec d'autres liens croissants. Pour nombre d'applications, c'est un **problème** (on désire la stabilisation de clusters), d'où l'introduction d'un **terme** d'inhibition compétitive (chap. 4.2.2.2) ou de **saturation** (limiter $\omega_{i,j} \leq \omega_{\max}$).

À l'inverse, dans certaines approches exploratoires, on peut **vouloir** conserver un degré de mouvement oscillatoire afin d'explorer diverses partitions temporaires. Mais la plupart du temps, si le but est d'obtenir un **arrangement stable**, il est crucial de **contenir** la rétroaction positive (régler η, τ de sorte que $\eta\tau \ll 1$) ou d'implémenter un mécanisme d'inhibition. Dans la section 4.4.2, on exposera des stratégies pour **endiguer** ces oscillations et viser la **stabilité** désirée.

4.4.1.2. Couplages non linéaires (ex. synergie dépendant des ω elles-mêmes)

Lorsqu'on conçoit un **DSL** (Deep Synergy Learning) de base, on postule souvent que la **synergie** $S(i,j)$ entre deux entités i et j découle de leurs **représentations** internes (\mathbf{x}_i pour le sub-symbolique, R_i pour le symbolique, etc.) sans dépendre de la **valeur courante** du lien $\omega_{i,j}$. Toutefois, dès qu'on autorise un **coupillage** où $S(i,j)$ devient **fonction** (directe ou indirecte) de la configuration $\{\omega_{p,q}\}$, on introduit une boucle de **rétroaction** supplémentaire. Cela peut, d'une part, **enrichir** la capacité d'auto-organisation et, d'autre part, **augmenter** le risque d'oscillations ou de comportements pseudo-chaotiques.

A. Principe : $S(i,j)$ dépend de ω

Dans le schéma de mise à jour standard,

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

on considère $S(i,j)$ comme **exogène**, c'est-à-dire qu'elle ne varie pas selon la valeur de $\omega_{i,j}$. Mais si, au contraire, $S(i,j)$ dépend de $\boldsymbol{\omega}(t)$, alors on obtient :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j, \boldsymbol{\omega}(t)) - \tau \omega_{i,j}(t)].$$

La **dépendance** $S(i, j, \omega(t))$ peut être **directe** (S explicitement lié à $\omega_{i,j}$) ou **indirecte** (ex. un module symbolique, activé seulement si certains liens sont déjà au-dessus d'un seuil, modifie la synergie).

On peut imaginer une **synergie** de la forme :

$$S(i, j) = S_0(i, j) + \alpha \sum_{k \neq i, j} \omega_{i,k} \omega_{k,j}.$$

Cela signifie que plus un *tiers* k connecte déjà i et j , plus la “coopération” potentielle entre i et j grandit. C'est un **effet d'auto-renforcement** où, si i et k sont liés et que k et j sont également liés, alors la synergie $S(i, j)$ augmente, renforçant ainsi indirectement la connexion entre i et j .

Mathématiquement, le **DSL** prend alors la forme d'un système **non linéaire** :

$$\omega(t+1) = \omega(t) + \eta [S(\omega(t)) - \tau \omega(t)].$$

La **variable** ω agit sur $S(\cdot, \cdot)$, qui *en retour* modifie ω . Sur le plan **dynamique**, cette boucle interne peut créer des régimes oscillatoires ou amplifiés si rien ne vient limiter cette rétroaction.

B. Conséquences Mathématiques : Non-linéarités et Risques d'Emballage

Dans un DSL où un **lien** $\omega_{i,j}$ peut *lui-même* accroître $S(i, j)$, la “correction” $\Delta\omega_{i,j} \approx \eta [S(i, j) - \tau \omega_{i,j}]$ risque de devenir **positive** au-delà d'un simple seuil, augmentant **exponentiellement** $\omega_{i,j}$. Sans un mécanisme de **saturation** ou d'**inhibition** (voir 4.4.2), $\omega_{i,j}$ peut monter sans bound ou osciller de façon importante.

Prenons un cas simplifié où :

$$S(i, j, \omega) = S_0(i, j) + \alpha \omega_{i,j}.$$

Alors la mise à jour additive devient :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S_0(i, j) + \alpha \omega_{i,j}(t) - \tau \omega_{i,j}(t)].$$

On obtient :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta S_0(i, j) + \eta (\alpha - \tau) \omega_{i,j}(t).$$

- Si $\alpha > \tau$, alors $\alpha - \tau > 0$, et l'on peut avoir $\eta (\alpha - \tau) > 1$, créant un **terme d'amplification** qui fait diverger $\omega_{i,j}$.
- À l'inverse, si $\eta (\tau - \alpha) > 1$, des oscillations ou un comportement erratique peut se produire selon le signe et la dynamique pas-à-pas.

Ainsi, de **petits** changements de paramétrage peuvent entraîner de grands effets.

Dans un **réseau** de grande dimension, des couplages non linéaires peuvent conduire à des **dynamiques** pseudo-chaotiques, dans lesquelles un **petit** écart initial sur $\omega(0)$ se traduit, au bout de plusieurs itérations, par de grandes différences (sensibilité exponentielle aux conditions initiales). Sur un plan théorique, ce type de comportement exige de solides **dispositifs** (inhibition,

limites de croissance, etc.) pour canaliser la rétroaction positive et préserver une trajectoire “stable” ou quasi-stable.

C. Exemples d’Applications

Même si les couplages non linéaires augmentent le risque de comportements complexes, ils **intéressent** certains scénarios.

Dans les **réseaux sociaux ou viraux**, un lien “viral” entre i et j renforce la probabilité que d’autres liens autour s’activent, ce qui, en retour, augmente la **popularité** de $\omega_{i,j}$. Sur un plan **DSL**, cela se traduit par une fonction $S(i,j,\omega)$ qui croît avec $\omega_{i,j}$. Sans mécanismes d’inhibition, un **emballage** peut se produire, entraînant la formation d’un cluster hautement **viralisé**.

Dans une **logique symbolique contexte-dépendante**, certaines **règles** ne s’appliquent que si un lien existant $\omega_{i,j}$ dépasse un certain **seuil**. Cela modifie **localement** la valeur de $S(i,j)$ lorsque la configuration $\{\omega_{k,\ell}\}$ franchit ce seuil. Sur le plan **dynamique**, cela crée des **transitions** d’état où un lien excédant le seuil déclenche une série de modifications en cascade, réorganisant ainsi la structure du réseau.

Dans un contexte de **collaboration multi-agent**, comme un essaim robotique, la **coopération** entre deux robots peut être **amplifiée** si un troisième robot interagit de manière synergique avec chacun d’eux. Cela se modélise par un **terme**

$$\sum_k \omega_{i,k} \omega_{k,j}$$

qui vient augmenter la synergie $S(i,j)$, ce qui favorise des **renforcements indirects** entre agents connectés via des intermédiaires.

Cependant, ce mécanisme comporte un **risque** où la formation de **boucles de renforcement** peut rendre le trio $\{i,j,k\}$ **instable** ou l’entraîner dans une **dynamique oscillatoire**, empêchant ainsi la stabilisation de l’ensemble du réseau.

D. Comment Contrôler ou Réguler ces Couplages ?

Pour éviter des **oscillations** ou une **explosion** des liens, on recourt souvent à :

- **Inhibition** (chap. 4.2.2.2) : Un coût $-\gamma \sum_k \omega_{i,k}$ empêche qu’un nœud i accroisse tous ses liens en même temps.
- **Saturation** ou “clipping” : On borne $\omega_{i,j} \leq \omega_{\max}$.
- **Seuil adaptatif** : On coupe toute $\omega_{i,j} < \theta$.

Ces **outils** limitent la rétroaction positive et bloquent la dérive exponentielle.

On peut analyser la **stabilité** via une étude du **système dynamique** :

$$\omega_{i,j}(t+1) - \omega_{i,j}(t) = \eta [S(i,j, \omega(t)) - \tau \omega_{i,j}(t)].$$

En dérivant les **conditions** sous lesquelles $\nabla_{\omega}S(\cdot)$ est dominée par le terme $-\tau \mathbf{I}$, on obtient des **critères** (genre $\eta \parallel \nabla S \parallel < \tau$) assurant un équilibre stable.

Dans certains cas, on *veut* cette rétroaction positive pour encourager un **cluster** à s'auto-former vigoureusement (ex. pour modéliser des “communautés fortes”). On l'associe à un **frein global** sous forme d'inhibition ou de compétition. Le résultat est l'émergence de **petits** clusters internes très soudés, plutôt que la formation de liens dispersés et diffus.

4.4.2. Stratégies pour Rompre les Oscillations

Dans la section 4.4.1, nous avons décrit **pourquoi** des oscillations ou un pseudo-chaos pouvaient apparaître dans un **SCN** (Synergistic Connection Network) où des rétroactions positives trop intenses, une absence d'inhibition ou des couplages non linéaires conduisent à des dynamiques instables. Pour de nombreuses applications du **DSL** (Deep Synergy Learning), on souhaite cependant **stabiliser** la dynamique, car des oscillations prolongées ou un chaos déterministe peuvent rendre la structure (clusters, attracteurs) peu exploitable.

Heureusement, il existe plusieurs **méthodes** pour **rompre** ou **atténuer** ces oscillations, en introduisant notamment :

- De l'**inhibition latérale** (4.4.2.1),
- Des **règles de sparsification** (4.4.2.2),
- Une **adaptation** fine des paramètres η, τ (4.4.2.3).

Ces stratégies visent à **contenir** l'emballlement de certains liens et à maintenir la rétroaction dans un domaine maîtrisé. Elles s'appliquent que la mise à jour soit **additive** ou **multiplicative**, et peuvent parfois se cumuler pour un meilleur effet stabilisateur.

4.4.2.1. Inhibition Latérale (2.4.4) : contrainte de ressources, pousse le réseau à choisir un nombre limité de liens forts

L'**inhibition latérale** – parfois appelée **inhibition compétitive** – vise à contrôler la **croissance simultanée** de multiples liens sortant d'une même entité \mathcal{E}_i . Au lieu de laisser $\omega_{i,j}$ s'élever librement lorsque la synergie $S(i,j)$ est satisfaisante, on y ajoute un **terme négatif** qui dépend de la **somme** (ou d'un autre agrégat) des connexions $\omega_{i,k}$ sortant du même noeud i . Une **forme** typique de cette mise à jour a été discutée en (4.2.2.2) et peut se formaliser, pour une règle additive, de la façon suivante :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t),$$

où $\gamma > 0$ est un **coefficent** d'inhibition, et $\sum_{k \neq j} \omega_{i,k}(t)$ désigne la **masse** totale des autres liens sortants de \mathcal{E}_i .

A. Principe et influence sur les oscillations

Lorsque \mathcal{E}_i dispose d'un **budget** de ressources limité, que ce soit en énergie neuronale dans un système biologique ou en bande passante dans un réseau technique, l'inhibition latérale impose une **compétition interne**. Si $\omega_{i,j}$ tente de s'élever, l'effet $-\gamma \sum_{k \neq j} \omega_{i,k}$ contraint la croissance simultanée d'autres liens, limitant ainsi l'expansion incontrôlée des connexions. Sur le plan **comportemental**, cela force \mathcal{E}_i à "choisir" un petit nombre de connexions privilégiées, évitant qu'il ne se connecte trop fortement à de multiples nœuds en parallèle.

Dans le DSL sans inhibition, un **lien** $\omega_{i,j}$ peut parfois s'amplifier fortement dès que la synergie $S(i,j)$ est supérieure à la "penalty" $\tau \omega_{i,j}$, en particulier s'il existe des boucles positives. L'inhibition latérale introduit un **terme négatif** supplémentaire proportionnel à la somme $\sum_{k \neq j} \omega_{i,k}$, ce qui limite la capacité de $\omega_{i,j}$ à croître de manière exponentielle. Ce **frein** empêche également un ensemble de liens de s'amplifier **simultanément**, l'un d'eux finissant par "dominer" les autres.

De ce fait, toute **oscillation** qui apparaît lorsque plusieurs liens tentent d'atteindre de grandes valeurs à tour de rôle se trouve en partie **amortie** par la compétition. Plus un lien s'élève, plus il pèse sur l'ensemble des autres, limitant leur croissance, et inversement, ce qui régule naturellement la dynamique du réseau. On observe ainsi, mathématiquement, une **dissipation** de l'énergie potentielle d'oscillation.

B. Analyse mathématique simplifiée

Pour voir comment l'inhibition latérale agit sur la **stabilité**, considérons la règle :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t).$$

Un **point fixe** $\omega_{i,j}^*$ s'obtient lorsque $\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega_{i,j}^*$, i.e. :

$$S(i,j) - \tau \omega_{i,j}^* - \gamma \sum_{k \neq j} \omega_{i,k}^* = 0.$$

Sur l'ensemble des entités, cela se traduit par un système d'équations linéairement couplées (ou, selon la forme de S , non linéaires), imposant une **compétition** pour qu'un nœud \mathcal{E}_i ne puisse "allouer" trop de force à plusieurs liens concurrents.

Sur le plan **stabilité**, la matrice jacobienne \mathbf{J} de la transformation $\omega_{i,j} \mapsto \omega_{i,j} + \dots$ inclut des **termes croisés** liés au $\sum_{k \neq j} \omega_{i,k}$. En un **point fixe**, ces dérivées croisées induisent un **effet de concurrence** où, lorsqu'un lien se renforce, il exerce une pression sur les autres, rendant ainsi plus difficile un **emballage synchronisé** et favorisant une stabilisation relative du réseau. On diminue ainsi la probabilité de boucles positives incontrôlées, et, dans la majorité des cas pratiques, on aboutit à un **régime** stable ou faiblement oscillatoire (des oscillations amorties).

C. Impact sur la formation de clusters

L'inhibition latérale incite chaque entité \mathcal{E}_i à ne conserver qu'un **nombre restreint** de liaisons fortes. Cela se rapproche d'un **modèle winner-takes-most** où, si $\omega_{i,j}$ devient très élevé pour un certain j , les autres $\omega_{i,k}$ avec $k \neq j$ subissent une **pénalisation accrue**, limitant leur croissance et renforçant la domination de la liaison la plus forte. On obtient donc un **réseau économe** en connexions, renforçant la **lisibilité** des structures émergentes (éviter que tout soit moyennement connecté).

Dans les algorithmes de **clustering**, on recherche souvent des **groupes** ou **communautés** clairement définis où les entités sont fortement reliées **entre elles** et faiblement connectées **aux autres**. L'inhibition latérale contribue à ce **clivage** en limitant la dispersion des connexions. Lorsqu'un nœud \mathcal{E}_i rejoint un **cluster principal**, il devient moins susceptible de développer des liens vers un autre cluster, car cette nouvelle croissance se heurterait au coût $\gamma \sum_{k \neq j} \omega_{i,k}$, rendant plus difficile l'expansion de connexions concurrentes. Cela **stabilise** les partitions émergentes.

D. Conclusion : un mécanisme crucial de stabilité

L'**inhibition latérale**, inspirée des phénomènes **biologiques** (2.4.4), remplit deux rôles essentiels :

Rôle de stabilisation : En limitant l'**auto-amplification** de multiples liens, on **réduit** le risque d'oscillations ou de comportements chaotiques. La dynamique DSL reste **contrôlée** et converge souvent plus sûrement.

Rôle de structuration : Chaque nœud se **focalise** sur quelques liens dominants, produisant des **clusters** plus nets et plus faciles à interpréter. La compétition incite le réseau à "faire des choix" tranchés, plutôt qu'à disperser ses connexions.

Dans les sections suivantes (4.4.2.2, 4.4.2.3) et plus loin (chap. 7.4), on verra comment compléter cette inhibition latérale par d'autres techniques (seuil adaptatif, saturation, recuit simulé) pour **approfondir** la maîtrise de la dynamique et **éviter** les fluctuations excessives. L'inhibition demeure toutefois l'un des **leviers** centraux pour maintenir un **SCN** (Synergistic Connection Network) dans un état **auto-organisé**, cohérent et **parcimonieux**.

4.4.2.2. Sparsification : imposer un maximum de k liaisons par entité (k plus forts liens), ε -radius, etc.

La **sparsification** vise à **réduire** le nombre de liaisons $\omega_{i,j}$ actives dans un **SCN** (Synergistic Connection Network), afin de **limiter** la complexité des rétroactions susceptibles de créer des oscillations ou d'autres comportements chaotiques. L'idée est qu'en empêchant chaque nœud \mathcal{E}_i d'entretenir de multiples liens à intensité moyenne, on obtient un **réseau** plus clairsemé (sparse), ce qui **stabilise** la dynamique et **clarifie** la formation de clusters. Les méthodes les plus courantes sont :

- **k-liaison** : on ne garde, pour chaque entité i , que les k plus forts liens $\omega_{i,j}$.

- **ε -seuil** : on annule tout lien $\omega_{i,j}$ dont la valeur tombe sous un seuil ε .

Dans un cas, on contrôle le **degré** sortant du nœud i (exactement k liens), dans l'autre, on impose un **niveau** minimal pour conserver une liaison.

A. Principe général de la sparsification

En l'absence de **restriction**, un nœud \mathcal{E}_i pourrait entretenir des connexions non négligeables vers la quasi-totalité des autres nœuds. Dans les réseaux de grande dimension, cela génère un très grand nombre de **circuits de rétroaction** où le surplus de liens de force **moyenne** augmente la **probabilité** d'oscillations ou de comportements complexes, rendant la dynamique du système plus imprévisible et potentiellement instable.

La **sparsification** empêche cette prolifération en forçant chaque nœud à **faire des choix**, c'est-à-dire à ne conserver qu'un **sous-ensemble** restreint de connexions jugées essentielles, limitant ainsi le risque d'oscillations et de comportements complexes dus à un excès de rétroactions. En conséquence, de nombreux liens, dont les valeurs sont trop faibles ou moins utiles, sont **coupés** (fixés à 0). On obtient alors un **réseau** moins densément connecté, limitant mécaniquement la complexité dynamique et réduisant le risque d'oscillation.

B. Méthodes pratiques

Après chaque cycle de mise à jour $\omega_{i,j}(t+1) \dots$, on “trie” les liaisons sortantes de chaque entité \mathcal{E}_i par ordre décroissant (ou selon un critère de pertinence). Seuls les k premiers liens $\omega_{i,j}$ (ceux de plus grande intensité) sont **préservés**. Les autres, classés au-delà du rang k , sont mis à 0.

$$\omega_{i,j}(t+1) = \begin{cases} \omega_{i,j}(t+1), & \text{si } \omega_{i,j} \text{ est dans le top } k, \\ 0, & \text{sinon.} \end{cases}$$

Cette **stratégie** garantit que le **degré** (sortant) de \mathcal{E}_i ne dépasse pas k . On parle parfois de “k-sparsification”. Cette forme de contrainte **réduit** fortement le nombre de liens actifs lorsque $k \ll n$, ce qui **simplifie** la dynamique du réseau. Chaque nœud ne s'occupe alors que d'un **petit ensemble** de voisins jugés les plus forts, limitant ainsi la complexité des interactions et stabilisant l'évolution des pondérations.

Ici, on fixe un $\varepsilon > 0$. Après la mise à jour, si $\omega_{i,j} < \varepsilon$, alors on **coupe** le lien $\omega_{i,j}$ en le mettant à 0 :

$$\omega_{i,j}(t+1) = \begin{cases} \omega_{i,j}(t+1), & \text{si } \omega_{i,j}(t+1) \geq \varepsilon, \\ 0, & \text{sinon.} \end{cases}$$

On obtient un “ ε -radius” où seuls les liens **au-dessus** du seuil ε subsistent. Le **degré** de chaque nœud n'est pas directement contrôlé, pouvant fluctuer, mais les connexions trop faibles sont systématiquement éliminées, limitant ainsi la prolifération de liens de force intermédiaire. Cette technique se rapproche du “**hard thresholding**” (voir 7.4.3) où, en deçà d'un certain seuil, un lien est considéré comme inexistant, renforçant la **sparsification** et la lisibilité de la structure du réseau.

On peut :

- **Appliquer** la coupe (k-liaison ou ε -seuil) à chaque itération (version stricte),

- Ou **périodiquement** (tous les p cycles, ou quand la densité du réseau dépasse un certain seuil).

De plus, on peut **faire varier** k ou ε au fil du temps en autorisant d'abord un plus grand nombre de liens, puis **en resserrant** progressivement la contrainte. Cela s'apparente à un recuit “inhibiteur”, où la parcimonie augmente à mesure que le réseau s'organise.

C. Impact sur la limitation des oscillations

Lorsqu'un **grand** nombre de liens coexistent dans un SCN, la **rétroaction** non linéaire peut entraîner des **oscillations** localisées ou globales (voir 4.3.2). En imposant un k -liaison ou un ε -seuil, on **casse** de nombreux arcs, réduisant ainsi drastiquement les chemins de rétroaction possibles. Moins de boucles = moins d'opportunités d'osciller.

Par ailleurs, chaque nœud n'a plus la “capacité” de faire croître simultanément un grand nombre de liens, car la plupart seront coupés s'ils sont trop faibles. La **dynamique** se concentre alors sur un ensemble restreint d'arcs — souvent, ceci rend la **trajectoire** du réseau plus lisible et moins sujette à des allers-retours.

D. Exemples d'usage et bénéfices

- **Clustering** plus franc : avec peu de liaisons conservées, les **clusters** deviennent plus évidents, car on voit nettement qui se connecte à qui.
- **Calcul plus rapide** : dans un réseau volumineux (n grand), manipuler une matrice ω dense peut être coûteux ($O(n^2)$ liens). Si, pour chaque i , on ne garde que k liens, on tombe à $O(nk)$, fréquemment $O(n)$ si k est constant. Cela accélère la mise à jour, en plus de calmer la dynamique.
- **Contrôle** de la densité : si le réseau tend à trop se connecter (souvent constaté dans l'entraînement DSL), la sparsification agit comme un “filet” coupant la majorité des petites connexions.

4.4.2.3. Adaptation de η, τ : régulation fine de la vitesse de mise à jour et de la décroissance.

La **dynamique** d'un SCN (Synergistic Connection Network) repose en grande partie sur deux **paramètres** fondamentaux où le **taux d'apprentissage** η et le **taux de décroissance** τ jouent un rôle central. Ces paramètres apparaissent dans l'équation de mise à jour (chap. 4.2) :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

ou, dans une version multiplicative, sous une forme proche de

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) \times [1 + \eta(\dots)] - \dots .$$

D'un point de vue **mathématique**, η contrôle la **vitesse** à laquelle chaque lien $\omega_{i,j}$ se met à jour ; τ détermine la **force** de “rappel vers zéro”, autrement dit la **décroissance** qui empêche le lien de grandir sans limite. Un mauvais choix de η ou de τ peut générer des **oscillations**, voire un

emballlement ; à l'inverse, des valeurs trop “timides” peuvent **freiner** exagérément la formation des clusters.

Dans les sections précédentes (4.4.2.1, 4.4.2.2), nous avons décrit l'**inhibition latérale** et la **sparsification** comme des solutions pour **endiguer** les oscillations. Ici, nous montrons qu'un **réglage** précis de η et τ est un levier tout aussi crucial, souvent complémentaire à ces mécanismes.

A. Rôle de η : vitesse de mise à jour

Dans le cas **additif** :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

L'analyse (4.2.1.2) révèle que pour garantir une convergence locale (sans oscillations), il faut

$$|1 - \eta \tau| < 1,$$

ou au moins $\eta \tau < 2$. Au-delà de cette borne, le système peut entrer en **oscillation** et, pour $\eta \tau$ encore plus grand, **diverger**.

Lorsque le SCN comprend **plusieurs** entités et couplages, l'analyse de stabilité se raffine par l'étude de la **matrice jacobienne** globale. Néanmoins, l'idée essentielle subsiste où un η trop fort accélère énormément la correction, risquant de **sauter** par-dessus la valeur d'équilibre et créant ainsi des **rebonds** successifs menant à des oscillations. En revanche, un η trop faible retarde la mise en place des **clusters**, rendant la convergence (ou la pseudo-convergence) plus lente.

À l'instar de nombreuses méthodes d'apprentissage, on peut **décroître** η au fil des itérations (un “annealing” du learning rate). On commence avec un η suffisamment élevé pour **différencier** rapidement les liens (ceux qui ont une synergie élevée se renforcent plus vite), puis on réduit η afin de **stabiliser** la configuration et éviter que les fluctuations ne perdurent.

B. Rôle de τ : force de la décroissance

Le paramètre τ agit comme un **frein**, à chaque mise à jour,

$$\eta [S(i,j) - \tau \omega_{i,j}(t)]$$

inclut la composante $-\eta \tau \omega_{i,j}(t)$, qui ramène $\omega_{i,j}$ vers zéro si la synergie $S(i,j)$ n'est pas assez grande. Si τ est trop **faible**, les liens risquent de croître exagérément ; si τ est trop **fort**, la plupart des liens restent proches de zéro, le SCN échouant à **exploiter** la synergie potentielle.

- $\tau \approx 0.1$: peut être trop faible si, par exemple, $S(i,j) \approx 1$. Dans ce cas, la décroissance (multipliée par $\omega_{i,j}$) n'est pas assez forte pour empêcher un emballlement (surtout si η n'est pas minime).
- $\tau \approx 10$: écrase presque toute **croissance** des liens, car $\tau \omega_{i,j}(t)$ dépasse rapidement $S(i,j)$. Les liaisons finissent par stagner très près de zéro.

Comme pour η , on peut **faire évoluer** τ en commençant avec un τ modéré, laissant le réseau former de premiers clusters, puis en l'augmentant pour consolider la configuration en bloquant les

variations tardives. Dans un SCN évolutif, un module de surveillance détecte d'éventuelles oscillations. Si elles sont trop fortes, on relève τ pour **amortir** davantage la dynamique.

C. Lien avec l'inhibition et la sparsification

Ces mécanismes se **superposent**. Avec **inhibition latérale** (4.4.2.1), la mise à jour inclut un terme

$$-\gamma \sum_{k \neq j} \omega_{i,k}(t),$$

qui limite aussi la croissance simultanée de plusieurs liens. Cela peut autoriser un η un peu plus élevé ou un τ plus bas, puisque la compétition prend en charge une partie de la **stabilisation**.

Si l'on **coupe** régulièrement les liens trop faibles (ε -seuil) ou ne garde que les k plus gros liens sortants, la dimension effective du système diminue (moins de boucles de rétroaction). Par conséquent, le couple (η, τ) peut être choisi un peu plus librement. Dans un réseau très dense, il faut souvent "serrer" η et τ de manière prudente.

D. Approche algorithmique pour η, τ : essais, annealing, jacobienne

Approche algorithmique pour η, τ : essais, annealing, jacobienne

L'ajustement des paramètres η (taux d'apprentissage) et τ (taux de décroissance) joue un rôle clé dans la dynamique du SCN. Une première méthode repose sur des **essais empiriques**, où l'on teste plusieurs paires (η, τ) en observant l'évolution du réseau. Si des oscillations excessives apparaissent, on réduit η ou on augmente τ . À l'inverse, si la formation de clusters est trop lente ou incomplète, on augmente η ou on diminue τ . Pour les SCN de grande taille, cette expérimentation est souvent menée d'abord sur un sous-problème réduit avant d'être généralisée.

Une alternative consiste à ajuster dynamiquement les paramètres selon une stratégie d'**annealing**. On commence avec un η relativement élevé pour accélérer la différenciation des liaisons, puis on le diminue progressivement, de manière linéaire ou exponentielle, une fois que la structure du réseau commence à émerger. En parallèle, on augmente τ pour accentuer l'effet de décroissance et stabiliser la configuration finale. Cette approche permet d'éviter un blocage initial tout en contrôlant la dynamique des ajustements tardifs.

Enfin, une analyse plus formelle peut être menée à travers l'**étude de la Jacobienne**. Sur un système réduit, il est possible d'analyser la matrice J , définie par $J = (\partial \Delta \omega / \partial \omega)$, afin de vérifier que sa norme reste inférieure à 1 et que son spectre en module ne dépasse pas cette valeur, assurant ainsi une convergence locale. Dans les SCN de grande échelle, cette analyse devient rapidement trop coûteuse. On privilégie alors les simulations pour observer l'évolution des pondérations ω et garantir la stabilité du réseau.

4.4.3. Cas Stochastiques : Recuit, Perturbations

Après avoir présenté (4.4.2) diverses stratégies **déterministes** (inhibition, sparsification, ajustement η, τ) pour rompre ou limiter les oscillations, nous abordons à présent des **approches stochastiques**. L'idée est parfois **d'injecter du bruit** (des perturbations aléatoires) dans la

dynamique, afin de sortir de configurations sous-optimales ou d'éviter une oscillation stable qui ne nous satisfait pas (chap. 2.4.5.1 abordait déjà ce concept). On retrouve ici l'analogie avec le **recuit simulé** en physique statistique, qui secoue la configuration $\omega(t)$ avant de “refroidir” pour se stabiliser dans un attracteur potentiellement plus global.

4.4.3.1. Injection volontaire de “bruit” pour échapper aux minima (2.4.5.1)

Il existe des situations où le **Synergistic Connection Network** (SCN) se trouve **piégé** dans un **minimum local** ou entretient une **oscillation** persistante qui empêche la dynamique de converger vers une configuration plus satisfaisante. Dans un tel contexte, l'idée d'introduire un **terme stochastique** dans la mise à jour des pondérations peut sembler contradictoire, d'autant plus que l'on cherche souvent à **diminuer l'instabilité** ou la **non-linéarité**. Cependant, cette démarche conserve une **logique** solide où, de la même manière que l'on applique un **recuit** en métallurgie pour franchir une barrière d'énergie, l'**injection de bruit** agit comme un **apport d'énergie** aléatoire permettant au SCN de se libérer d'un attracteur local ou de rompre un **cycle** d'oscillation (cf. section 2.4.5.1).

Une **contradiction** apparente peut se lire dans le fait que l'on introduise délibérément des **perturbations aléatoires**, alors même que des mécanismes tels que l'**inhibition** ou la **sparsification** visent à stabiliser la dynamique. Pour comprendre cette démarche, il faut garder à l'esprit que l'objectif premier est de **favoriser l'exploration** de configurations nouvelles, qui seraient difficilement atteignables depuis un état trop stable mais de mauvaise qualité. Les **systèmes physiques** fournissent une analogie où un solide en cours de recuit a besoin d'un **excès** d'énergie pour franchir le **seuil** séparant deux **vallées** de potentiel. Dans le cadre d'un **SCN**, le bruit joue un rôle semblable en **déstabilisant** ponctuellement la **trajectoire** des pondérations $\omega_{i,j}$. On peut alors espérer que la structure glisse vers un autre **bassin d'attraction**, plus proche d'une configuration optimale.

Il est fréquent, dans les **réseaux** adaptatifs et non linéaires, de se trouver en présence d'**oscillations** du type “ping-pong”, où quelques liaisons alternent périodiquement entre des valeurs élevées et faibles, sans jamais converger vers un point fixe stable. L'**injection** d'un **terme aléatoire** peut dans ce cas **briser** la synchronisation de l'oscillation. Si l'on modélise ce phénomène, on peut considérer qu'à chaque itération t , la mise à jour

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] + \xi_{i,j}(t)$$

vient remplacer la formule usuelle $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$. Le **terme** $\xi_{i,j}(t)$ est un **bruit** gaussien ou uniforme, en général de moyenne nulle, dont l'**amplitude** peut être fixée ou décroissante au fil du temps. Dans certains cas, on peut aussi cibler seulement un **sous-ensemble** de paires (i,j) pour ne pas perturber l'ensemble du réseau.

Sur le **plan mathématique**, on peut relier cette injection de bruit à la **descente de gradient** perturbée, très souvent employée pour **échapper** aux minima locaux. Bien que le **DSL** et son **SCN** ne correspondent pas toujours à une **descente de gradient** globale, puisqu'il s'agit souvent d'une mise à jour locale ne minimisant pas explicitement une fonction (voir chapitre 2.4.3 sur la notion d'**énergie**), le **raisonnement** demeure valable. Tout **puits** local où le réseau serait “coincé” dispose d'une **barrière** à franchir, et le bruit aléatoire fournit la **quantité** d'énergie nécessaire pour la

surmonter. Par analogie, on peut dire que l'**état** du SCN suit un “processus stochastique” à l’intérieur d’un **paysage d’énergie**, et qu’en l’absence d’un tel bruit, l’état risquerait de converger dans la première vallée rencontrée, sans avoir la possibilité d’explorer un autre **bassin** à énergie plus basse.

D’un point de vue **dynamique**, on peut observer que dans des conditions de bruit modéré, la **dérive** aléatoire permet d’**éviter** la cristallisation prématuée d’un **mauvais** attracteur local. Si le bruit est trop élevé, on peut toutefois provoquer un **comportement** erratique qui empêche de **stabiliser** réellement la structure. En pratique, on paramètre la **variance** de $\xi_{i,j}(t)$ ou sa **fenêtre** d’uniformité de façon à autoriser quelques **sauts** hors des vallées, mais sans plonger le réseau dans un mouvement totalement brownien.

4.4.3.2. Recuit simulé : un protocole pour secouer la matrice ω avant de “refroidir” et se stabiliser

Le recours à un **recuit simulé** (`\emph{simulated annealing}`) permet de structurer plus finement l’**injection de bruit** au sein d’un **Synergistic Connection Network** (SCN). Plutôt que d’ajouter des perturbations stochastiques ponctuelles ou uniformément réparties dans le temps, le **recuit** propose de “chauffer” d’abord le réseau à une **température** élevée pour lui donner la capacité d’**explorer** un vaste éventail de configurations, puis de “refroidir” graduellement, laissant le SCN se **figer** dans un état final qu’on espère “globalement optimal” ou, à tout le moins, **meilleur** qu’un minimum local trivial. Cette approche puise ses racines dans la **physique statistique**, où l’on chauffe un matériau (agitation thermique forte) avant de le refroidir lentement, de sorte à franchir des barrières d’énergie et atteindre une structure cristalline stable.

A. Analogies et principes du recuit.

Le recuit s’inspire de la métallurgie où un matériau chauffé voit ses atomes disposer d’une énergie thermique suffisante pour franchir de nombreuses barrières locales et se réarranger. En le refroidissant graduellement, on stabilise la configuration dans une cristallisation souvent plus homogène et moins sujette aux défauts. Transposé à un SCN, le “chauffage” correspond à l’adjonction d’un **terme aléatoire** de grande amplitude à la formule de mise à jour des pondérations $\omega_{i,j}$. Ceci permet au réseau de “sauter” entre plusieurs bassins d’attraction, sans se bloquer prématurément dans l’un d’eux. Le “refroidissement” opère en réduisant progressivement ce **bruit**, faisant tendre la dynamique vers une structure où les liens $\omega_{i,j}$ se figent dans une configuration stable.

Dans la perspective d’une **fonction énergie** $J(\omega)$ (voir la section 2.4.3 sur la vision “descente d’énergie”), un important niveau de bruit favorise les remontées énergétiques transitoires, que l’on peut formaliser par un facteur $\exp[-(J(\omega') - J(\omega))/T]$. Le réglage d’une **température** T régule le degré de tolérance aux hausses d’énergie où, à haute température, on accepte aisément de “monter” en énergie pour fuir un puits local médiocre, tandis qu’à basse température, on rejette de tels mouvements, ce qui stabilise le SCN.

B. Protocole concret dans un SCN.

Dans une formulation pratique, on pose une **température** T dont la valeur décroît au fil des itérations selon une loi typique telle que

$$T(t) = T_0 \alpha^{\lfloor t/\Delta t \rfloor}, \quad 0 < \alpha < 1,$$

avec un pas de temps Δt . On ajoute alors, à chaque mise à jour $\omega_{i,j}(t+1)$, un **bruit** $\xi_{i,j}(t)$ dont la **variance** dépend de T . Ainsi, on obtient la relation

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] + \xi_{i,j}(t),$$

où $\xi_{i,j}(t) \sim \mathcal{N}(0, \sigma(t)^2)$ avec $\sigma(t) \propto T(t)$. Dans les premières itérations, $\sigma(t)$ est grande où le SCN est agité et “saute” volontiers hors des puits locaux. Puis, à mesure que T baisse, $\sigma(t)$ diminue, ce qui fige progressivement la structure émergente.

C. Analyse mathématique du recuit.

Sur le plan formel, l'on connaît des **résultats** garantissant qu'un refroidissement suffisamment lent (c'est-à-dire un α très proche de 1) peut conduire le système à **converger** vers un minimum global de l'énergie \mathcal{J} . En pratique, on arbitre entre la qualité de la solution (un refroidissement plus lent favorise l'exploration) et le temps de calcul (un refroidissement trop lent peut être inabordable). Dans un **SCN**, on n'a pas toujours besoin de suivre l'exact protocole de la physique statistique (Markov Chain Monte Carlo, acceptation/rejet), car un simple **bruit additif** “pondéré” par $\exp[-\Delta \mathcal{J}/T]$ peut remplir la fonction d'exploration.

En tout état de cause, le recuit simulé fournit un **cadre** méthodique pour **monter** d'abord la “température”, explorer le **paysage** d'états plus librement, puis “refroidir” en limitant graduellement les perturbations.

D. Avantages et inconvénients.

Le recuit simulé présente l'**avantage** majeur de **décoller** le réseau d'un minimum local (ou d'un attracteur) inopportun. Là où un DSL déterministe, limité à

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

risquerait de s'arrêter prématurément, la “secousse” apportée par un niveau élevé de bruit autorise l'exploration d'autres **partitions** ou **clusters** potentiels. Cette dimension d'exploration peut aussi, dans les phases “haute température”, dévoiler plusieurs organisations concurrentes, dont l'une s'avérera plus globalement satisfaisante.

Ce **bénéfice** se paie par un **coût** temporel accru. D'une part, on doit maintenir des itérations “bruyantes” en nombre suffisant pour vraiment sauter entre bassins d'attraction ; d'autre part, il faut prévoir un **refroidissement** progressif, ce qui rallonge la durée de la simulation. Un autre **risque** vient de la configuration initiale de $\sigma(t)$ où, si la température de départ est trop basse, le caractère stochastique ne sera pas suffisamment exploité, tandis que si elle est trop haute, la trajectoire pourra demeurer chaotique ou écarter certaines bonnes configurations faute d'un contrôle approprié.

E. Conclusion.

Le recuit simulé se révèle un protocole pertinent pour “**secouer**” la matrice $\{\omega_{i,j}\}$ avant de “\textbf{refroidir}” la dynamique, cherchant ainsi à **éviter** les minima locaux et à **brisier** des oscillations éventuelles. Il complète les approches plus déterministes (inhibition, saturation) qui visent à **stabiliser** la structure. En veillant à la cohérence du rythme de décroissance de la température, on apporte au **Synergistic Connection Network** un instrument supplémentaire pour **explorer** l'espace des configurations, déjouer les verrous locaux et produire, in fine, une **organisation** potentiellement plus satisfaisante à l'échelle du réseau.

4.4.3.3. Choix de la “température” stochastique, liens avec la physique statistique

Lorsqu'un **recuit simulé** ou plus généralement une **injection de bruit** (voir 4.4.3.1 et 4.4.3.2) est implémentée dans un **Synergistic Connection Network** (SCN), la notion de **température** T joue un rôle central. Cette **température**, inspirée des concepts de la **physique statistique**, contrôle l'**ampleur** des perturbations aléatoires ajoutées aux pondérations $\omega_{i,j}$. Un bon réglage de T est donc déterminant pour que le réseau puisse, dans un premier temps, **explorer** suffisamment l'espace des configurations, puis, à l'inverse, se **figer** à la fin du processus, permettant ainsi une convergence vers un état stable.

A. Rappel : analogie avec la physique statistique.

En physique, la **température** T mesure l'**agitation** des atomes ou des particules. Un solide porté à haute température dispose de suffisamment d'**énergie** pour dépasser des barrières locales et se réorganiser librement. Dans le **SCN**, la “température” joue un rôle similaire où, lorsqu'elle est élevée, on autorise l'**injection** de perturbations stochastiques de grande amplitude dans la mise à jour des pondérations. Ces perturbations permettent au réseau de “sauter” hors de puits d'énergie trop étroits ou de configurations peu satisfaisantes. À mesure qu'on réduit progressivement T , l'agitation baisse et le SCN se **stabilise**, à l'image d'un refroidissement qui fige les atomes d'un métal dans une structure cristallisée.

Il existe, dans les modèles inspirés de la physique statistique (Ising, Hopfield, etc.), une relation de type $\exp[-(\mathcal{J}(\omega') - \mathcal{J}(\omega))/T]$ régissant la probabilité d'accepter une modification augmentant l'énergie du système. Même si un **DSL** peut s'écartier de ces protocoles Markoviens classiques, la **logique** reste identique : un niveau de température suffisamment haut donne au SCN la capacité de franchir de petites “vallées” d'énergie et d'explorer son espace d'états de manière étendue.

B. Comment définir T et l'amplitude de bruit.

Le principe consiste à associer l'**injection** de bruit à la **température**. Concrètement, si $\xi_{i,j}(t)$ désigne un bruit gaussien de moyenne nulle, sa variance σ^2 peut être reliée à la température via $\sigma = \sigma_0 T$. Lorsque T est grand, les fluctuations $\xi_{i,j}(t)$ atteignent une amplitude notable ; à l'inverse, pour une température faible, elles deviennent minimales. Dans un **recuit simulé**, la température elle-même suit un schéma de décroissance contrôlé, par exemple

$$T(t) = T_0 \alpha^{\lfloor t/4 \rfloor},$$

avec $\alpha \in (0,1)$. Ainsi, au démarrage, T_0 est suffisamment élevé pour permettre une exploration étendue, puis, au fil du temps, on abaisse σ et on réduit graduellement le mouvement aléatoire autour de la configuration courante.

Cette stratégie reproduit l'idée que la **dynamique** d'un SCN doit d'abord être **libérée** (phase de haute température) pour échapper à des attracteurs locaux trop précoces, puis **ralentie** (phase de basse température) afin de stabiliser la structure. Dans certains contextes, on maintient même un petit bruit résiduel pour conserver un léger "tremblement" qui peut se révéler bénéfique, par exemple pour empêcher de retomber dans des oscillations cycliques.

C. Sélection de T_0 et vitesse de décroissance.

Le choix de la **température initiale** T_0 a un impact crucial. Si elle est trop basse, la perturbation stochastique est insuffisante pour sortir le réseau de son attracteur initial ; si elle est trop élevée, toute structuration précoce de $\omega_{i,j}$ est détruite, et le SCN passe son temps en réarrangements chaotiques. Il convient, en pratique, d'ajuster T_0 par essais, parfois en s'appuyant sur une estimation du "coût" des barrières typiques.

Le rythme de **refroidissement** (vitesse à laquelle α fait décroître T) détermine la portée de l'exploration. Un refroidissement extrêmement lent peut tendre vers un optimum global, selon les théorèmes classiques du recuit simulé, mais cela s'avère onéreux en temps de calcul. Les approches plus pragmatiques s'accordent d'une décroissance modérée qui équilibre la qualité de la solution et l'efficacité de la simulation.

Les **critères** d'arrêt (seuil minimal de température, durée maximum d'itérations, stabilité de la matrice ω) contribuent à définir la durée du recuit. Il faut laisser au SCN le temps d'**explorer**, tout en évitant de prolonger trop inutilement une phase de bruit intense qui empirerait le coût de calcul.

D. Liens formels avec la physique statistique.

Dans un **DSL** disposant d'une fonction "énergie" $\mathcal{J}(\omega)$, on peut imaginer un protocole de type Metropolis–Hastings, où l'on calcule la différence $\Delta\mathcal{J} = \mathcal{J}(\omega') - \mathcal{J}(\omega)$ et on accepte, selon la probabilité $\min\{1, \exp[-\Delta\mathcal{J}/T]\}$, une modification qui augmente l'énergie. Cet artifice modélise la possibilité de s'extraire d'un puits local, contrôlée par la température. Bien que la mise à jour d'un SCN ne soit pas toujours décrite en ces termes, la **logique** demeure où un niveau de T élevé inflige des **sauts** plus fréquents et plus importants, tandis qu'un faible T "gèle" l'état.

En physique, on observe des **transitions** de phase (paramagnétique à ferromagnétique, par exemple). Parallèlement, dans un SCN, un fort bruit peut maintenir un ensemble de liaisons "déstructuré", puis un abaissement de T induit une **cristallisation** sous forme de **clusters** affirmés.

E. Conclusion.

Le **choix** de la "température" T est un **levier essentiel** pour piloter l'équilibre entre l'exploration (grâce à un bruit significatif) et la stabilisation (grâce à un bruit quasi nul). En début d'apprentissage, on privilégie une "phase chaude" (élevée T_0) qui facilite les sauts hors des minima locaux et la rupture d'oscillations non désirées. Puis, en réduisant progressivement $\sigma(t) \propto T(t)$, le réseau se "fige" dans une configuration plus stable, comparable à un métal cristallisé après un recuit lent.

Cette démarche **stochastique**, inspirée de la **physique statistique**, complète les mécanismes plus déterministes (inhibition, parsimonie) déjà déployés pour maîtriser la dynamique du SCN. Le paramétrage judicieux de T_0 , de α et du temps de refroidissement offre un **contrôle** fin sur l'ensemble du processus et peut considérablement améliorer le **résultat** final, notamment dans les situations où la fonction énergie \mathcal{J} possède de nombreuses vallées locales ou où le SCN développe des **oscillations** difficiles à résorber autrement.

4.5. Héritage de la Physique Statistique et des Systèmes Dynamiques

Tout au long des sections précédentes (4.2, 4.3, 4.4), la **dynamique** d'un **SCN** (Synergistic Connection Network) a été soulignée en mettant en évidence l'évolution des pondérations $\omega_{i,j}$, la formation des clusters, l'apparition d'oscillations et les mécanismes permettant de les contrôler, tels que l'inhibition, la sparsification et le recuit simulé. Derrière ces mécanismes se cachent des **concepts** issus de la **physique statistique** (modèles d'énergie, recuit, transitions de phase) et de la **théorie des systèmes dynamiques** (points fixes, cycles, attracteurs). Cette section (4.5) met en évidence cet **héritage** et explique pourquoi on parle souvent de "descente d'énergie" ou de "contraction" locale pour caractériser la stabilité d'un point fixe.

4.5.1. Notions de Contraction, Point Fixe, Attracteurs

La grande force d'un **SCN** (Deep Synergy Learning) réside dans sa **capacité** à faire émerger des **états** (clusters, arrangements) stables ou quasi stables. Pour analyser ces états, on s'appuie sur des outils mathématiques liés aux **points fixes** et aux **attracteurs** d'une application **F** (le "règlement" qui, à chaque itération, met à jour $\omega_{i,j}(t)$).

4.5.1.1. Rappels mathématiques : $\|DF(\Omega^*)\| < 1$, etc.

Dans l'analyse des dynamiques d'auto-organisation dans un **Synergistic Connection Network (SCN)**, il est essentiel de disposer d'un cadre mathématique permettant d'étudier la stabilité des pondérations $\omega_{i,j}$. Ce cadre s'appuie sur la notion d'**équilibre** ou de **point fixe** d'une fonction de mise à jour globale, ainsi que sur la condition de **contraction locale** caractérisée par l'opérateur dérivé de cette fonction. Nous rappelons ici les principaux éléments de ce formalisme.

A. Formalisation de la mise à jour

On considère que la mise à jour globale des pondérations peut être décrite par une fonction

$$F: \mathcal{W} \rightarrow \mathcal{W},$$

où \mathcal{W} désigne l'espace des matrices $\{\omega_{i,j}\}$, souvent assimilé à $\mathbb{R}^{n \times n}$ si l'on compte n entités dans le SCN. Ainsi, à l'itération t , le vecteur (ou la matrice) des pondérations est mis à jour selon

$$\omega(t+1) = F(\omega(t)).$$

Un **point fixe** ω^* est défini par la condition

$$\omega^* = F(\omega^*),$$

ce qui signifie que si le système atteint cet état, aucune mise à jour ultérieure ne le modifie. Ce concept constitue la base pour étudier la convergence et la stabilité du SCN.

B. Condition de contraction locale

Pour qu'un point fixe ω^* soit stable, il est nécessaire que la fonction de mise à jour contracte les petites perturbations autour de ce point. En termes mathématiques, on examine la **matrice jacobienne** de \mathbf{F} évaluée en ω^* , notée $D\mathbf{F}(\omega^*)$. On impose qu'il existe une **norme** $\|\cdot\|$ sur l'espace \mathcal{W} telle que

$$\| D\mathbf{F}(\omega^*) \| < 1.$$

Cette inégalité indique que toute petite perturbation δ autour de ω^* est réduite par l'application de $D\mathbf{F}$, ce qui se traduit par un retour progressif vers le point fixe. La condition de contraction, souvent résumée par $\| D\mathbf{F}(\Omega^*) \| < 1$, est donc le critère de **stabilité locale**.

C. Exemple élémentaire en dimension 1

Considérons un système unidimensionnel où la mise à jour s'exprime par

$$\omega(t+1) = F(\omega(t)).$$

Un point fixe ω^* satisfait

$$\omega^* = F(\omega^*).$$

La condition de stabilité locale pour ce système s'écrit alors

$$|F'(\omega^*)| < 1,$$

ce qui assure qu'une petite perturbation autour de ω^* sera contractée par l'itération. En dimension supérieure, cette condition se généralise par l'exigence que la norme de la matrice jacobienne $D\mathbf{F}(\omega^*)$ soit strictement inférieure à 1.

D. Lien avec la dynamique additive

Dans le DSL, la mise à jour des pondérations se fait généralement par une règle additive de la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où $S(i,j)$ est la synergie entre les entités \mathcal{E}_i et \mathcal{E}_j , η le taux d'apprentissage, et τ le coefficient de décroissance. Pour analyser la stabilité autour du point fixe, nous supposons que $S(i,j)$ est constant et que la dynamique converge vers $\omega_{i,j}^*$. En posant

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega_{i,j}^*,$$

nous avons

$$\omega_{i,j}^* = \omega_{i,j}^* + \eta [S(i,j) - \tau \omega_{i,j}^*].$$

En simplifiant, nous obtenons

$$S(i,j) - \tau \omega_{i,j}^* = 0 \quad \Rightarrow \quad \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Pour étudier la stabilité locale, on examine la dérivée locale de la fonction de mise à jour. Dans ce schéma simplifié, la dérivée par rapport à $\omega_{i,j}$ s'exprime par

$$\frac{d}{d\omega_{i,j}} [\omega_{i,j} + \eta(S(i,j) - \tau \omega_{i,j})] = 1 - \eta \tau.$$

La condition de convergence locale exige que

$$|1 - \eta \tau| < 1.$$

Cette inégalité traduit la **contraction** autour du point fixe, c'est-à-dire que toute perturbation sera atténuée au fil des itérations, garantissant la stabilité du système.

Conclusion partielle

Le critère de contraction, symbolisé par l'inégalité

$$\| D\mathbf{F}(\boldsymbol{\omega}^*) \| < 1,$$

constitue la base pour vérifier la **stabilité** d'un point fixe dans un SCN. Lorsque l'on applique la règle de mise à jour additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

et que l'on considère un cas stationnaire où $S(i,j)$ est constant, le système converge vers un équilibre donné par

$$\omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

La dérivée locale $1 - \eta \tau$ doit être inférieure à 1 en valeur absolue, ce qui assure que toute perturbation autour de l'équilibre est contractée, garantissant ainsi la stabilité locale du SCN. Cette interprétation en termes de **descente d'énergie** ou de **gradient** offre une vision intuitive de la manière dont le DSL équilibre le renforcement des liens avec la nécessité de réguler leur croissance, évitant ainsi une explosion des pondérations. Bien entendu, dans des cas plus complexes où $S(i,j)$ varie dans le temps, l'analyse se complique, mais le principe fondamental demeure : la stabilité du point fixe repose sur la condition de contraction de la fonction de mise à jour.

4.5.1.2. Critères de stabilité locale ou globale, cas non linéaires

Dans l'analyse des systèmes dynamiques qui sous-tendent un **Deep Synergy Learning (DSL)**, la question de la stabilité des points fixes et des attracteurs est centrale. En effet, pour garantir que les pondérations $\omega_{i,j}$ convergent vers des valeurs qui reflètent correctement la **synergie** entre entités, il convient de s'appuyer sur des critères mathématiques issus de la théorie des systèmes dynamiques. Ces critères se distinguent en deux catégories : la **stabilité locale**, qui examine le comportement du système à proximité d'un point fixe, et la **stabilité globale**, qui concerne la convergence de la dynamique depuis presque toute condition initiale. La présence de **non-linéarités** telles que l'inhibition, les couplages multiplicatifs ou les mécanismes de saturation complique l'analyse, mais le principe de base demeure celui de la contraction autour de l'attracteur.

A. Stabilité locale : La contraction au voisinage du point fixe

Pour étudier la stabilité locale d'un point fixe Ω^* dans un SCN, on considère la fonction de mise à jour globale

$$\mathbf{F}: \mathcal{W} \rightarrow \mathcal{W},$$

définie sur l'espace \mathcal{W} des matrices de pondérations. À chaque itération, le vecteur de pondérations se met à jour selon

$$\omega(t+1) = \mathbf{F}(\omega(t)).$$

Un point fixe Ω^* satisfait alors la condition

$$\Omega^* = \mathbf{F}(\Omega^*).$$

La **stabilité locale** se caractérise par la capacité du système à ramener toute perturbation infinitésimale autour de Ω^* vers ce point fixe. Pour cela, on linéarise \mathbf{F} autour de Ω^* en introduisant la **matrice jacobienne** $D\mathbf{F}(\Omega^*)$. La condition de contraction est alors exprimée par

$$\| D\mathbf{F}(\Omega^*) \| < 1,$$

où $\|\cdot\|$ désigne une norme appropriée (par exemple, la norme spectrale) et où le rayon spectral $\rho(D\mathbf{F}(\Omega^*))$ est inférieur à 1. En d'autres termes, toute perturbation δ autour du point fixe satisfait, pour une itération :

$$\omega(t+1) - \Omega^* \approx D\mathbf{F}(\Omega^*)(\omega(t) - \Omega^*),$$

et la norme de l'écart se contracte à chaque itération, garantissant que

$$\lim_{t \rightarrow \infty} \| \omega(t) - \Omega^* \| = 0.$$

Cette condition de contraction, analogue à la condition en dimension 1 $|F'(x^*)| < 1$, assure que dans un voisinage de Ω^* , la dynamique est **attractive** et les petites perturbations sont rapidement annulées par le processus itératif.

B. Stabilité globale : Convergence depuis presque toute condition initiale

La **stabilité globale** implique que, quel que soit l'état initial $\omega(0)$ (à l'exception d'un ensemble de mesure nulle), la trajectoire converge vers le point fixe Ω^* , c'est-à-dire

$$\lim_{t \rightarrow \infty} \| \omega(t) - \Omega^* \| = 0.$$

Pour démontrer la stabilité globale, il faut que la fonction de mise à jour \mathbf{F} soit **contractante** sur l'ensemble de l'espace \mathcal{W} , et non seulement dans un voisinage de Ω^* . Cette condition est beaucoup plus forte et rarement vérifiée dans des systèmes non linéaires à haute dimension, où l'on observe souvent la coexistence de plusieurs attracteurs ou de **bassins d'attraction** distincts. Dans ces cas, le système présente une **multi-stabilité** et la convergence dépend fortement de la condition initiale.

C. Cas non linéaires et phénomènes complexes

Les systèmes réels de DSL intègrent fréquemment des **non-linéarités** telles que des mécanismes d'inhibition, des couplages multiplicatifs ou des saturations. Par exemple, une mise à jour du type multiplicatif peut être exprimée par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) \exp(\eta [S(i,j) - \tau \omega_{i,j}(t)]),$$

ou encore des termes d'inhibition peuvent être ajoutés pour moduler la dynamique en fonction des interactions globales, donnant lieu à une mise à jour plus complexe du type

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t).$$

Ces non-linéarités rendent l'analyse de la stabilité plus subtile. La **matrice jacobienne** de la fonction de mise à jour devient alors un opérateur qui intègre des termes croisés et dont les valeurs propres peuvent être complexes ou dépasser 1 en module, ce qui conduit à des phénomènes tels que les **oscillations**, les **bifurcations** ou même le **chaos**. Dans ces régimes, le système ne converge pas nécessairement vers un point fixe stable, mais peut osciller ou présenter des attracteurs étranges. Les méthodes de **contrôle** telles que l'inhibition, la **sparsification** ou l'ajustement dynamique des paramètres η et τ sont alors nécessaires pour maintenir une dynamique raisonnablement stable.

4.5.1.3. Illustrations : Petit SCN (3 ou 4 entités) pour montrer stabilisation exponentielle ou oscillations

Afin de rendre concrètes les notions de stabilité, d'attracteurs et de convergence évoquées dans les sections précédentes (notamment 4.5.1.1 et 4.5.1.2), il est instructif d'étudier des cas à petite échelle, par exemple un SCN constitué de trois ou de quatre entités. Ces illustrations permettent de mettre en évidence, de manière numérique et graphique, comment la dynamique de mise à jour des pondérations $\omega_{i,j}$ peut converger de manière exponentielle vers un point fixe ou, au contraire, présenter des oscillations périodiques, en fonction du choix des paramètres d'apprentissage et de décroissance.

A. Cas à 3 entités : Stabilisation exponentielle

Considérons un SCN formé de trois entités, notées \mathcal{E}_1 , \mathcal{E}_2 et \mathcal{E}_3 . Pour simplifier l'analyse, nous supposerons que la **synergie** $S(i,j)$ entre chaque paire d'entités est stationnaire, c'est-à-dire constante au cours du temps. Par exemple, nous définissons les synergies par les valeurs suivantes :

$$S(1,2) = 0.8, \quad S(1,3) = 0.7, \quad S(2,3) = 0.2,$$

et par symétrie, $S(2,1) = 0.8$, $S(3,1) = 0.7$ et $S(3,2) = 0.2$. Dans ce scénario, la mise à jour des pondérations est réalisée selon la règle additive classique

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où η est le **taux d'apprentissage** et τ est le **coefficient de décroissance**. Pour illustrer la stabilisation exponentielle, nous choisissons par exemple $\eta = 0.05$ et $\tau = 1.0$. L'initialisation des pondérations est faite avec des valeurs proches de zéro, par exemple, $\omega_{i,j}(0) \approx 0$ (avec une légère perturbation aléatoire pour simuler du bruit).

Au cours des itérations, les pondérations $\omega_{1,2}(t)$ et $\omega_{1,3}(t)$ augmentent progressivement car $S(1,2) = 0.8$ et $S(1,3) = 0.7$ sont relativement élevées, tandis que $\omega_{2,3}(t)$ croît plus lentement en raison d'une synergie moindre $S(2,3) = 0.2$. Par l'analyse de la règle de mise à jour, nous pouvons observer qu'en supposant une convergence stationnaire, c'est-à-dire en posant $\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega_{i,j}^*$, nous obtenons :

$$\omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Ainsi, pour le lien entre \mathcal{E}_1 et \mathcal{E}_2 , on a $\omega_{1,2}^* = 0.8$; pour le lien entre \mathcal{E}_1 et \mathcal{E}_3 , $\omega_{1,3}^* = 0.7$; et pour le lien entre \mathcal{E}_2 et \mathcal{E}_3 , $\omega_{2,3}^* = 0.2$. En outre, la dynamique de convergence se caractérise par une décroissance exponentielle de l'écart, puisque la mise à jour linéaire entraîne que la différence $|\omega_{i,j}(t) - \omega_{i,j}^*|$ diminue en proportion d'un facteur constant $(1 - \eta \tau)$ à chaque itération. Graphiquement, en traçant $\omega_{1,2}(t)$ en fonction de t , on observe une courbe exponentielle ascendante se rapprochant asymptotiquement de 0.8. Ce cas illustre ainsi la **stabilité locale** dans un SCN simple, où la dynamique converge de manière prévisible vers des points fixes déterminés par la synergie stationnaire.

B. Cas à 4 entités : Apparition d'oscillations

Pour examiner un comportement différent, considérons maintenant un SCN comportant quatre entités, notées $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3$ et \mathcal{E}_4 . Supposons des synergies stationnaires qui varient de façon à créer des interactions compétitives, par exemple :

$$S(1,2) = 0.9, \quad S(2,3) = 0.8, \quad S(3,4) = 0.7, \quad S(4,1) = 0.8,$$

et pour certaines paires moins influentes, $S(1,3) = 0.1$ et $S(2,4) = 0.2$. Dans ce scénario, nous adoptons une **mise à jour multiplicative** afin d'exacerber les effets non linéaires, en utilisant une règle du type

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) [1 + \eta S(i,j) - \eta \tau \omega_{i,j}(t)].$$

Nous choisissons ici des paramètres plus agressifs, par exemple $\eta = 0.8$ et $\tau = 0.2$, et nous initialisons les pondérations avec une valeur faible, par exemple $\omega_{i,j}(0) \approx 0.01$. Dans ce régime, en raison du taux d'apprentissage élevé et de la faible décroissance, le système peut entrer dans une phase oscillatoire. Concrètement, les pondérations des liens avec des synergies élevées, telles que $\omega_{1,2}(t), \omega_{2,3}(t), \omega_{3,4}(t)$ et $\omega_{4,1}(t)$, vont d'abord augmenter rapidement. Cependant, lorsque l'un de ces liens atteint un niveau trop élevé, l'effet multiplicatif couplé à la rétroaction négative induite par $-\eta \tau \omega_{i,j}(t)$ provoque un sur-correction qui fait diminuer ce lien, tandis qu'un autre lien compense en augmentant, créant ainsi un cycle de **ping-pong** entre les différentes pondérations. Par exemple, $\omega_{1,2}(t)$ peut augmenter jusqu'à un certain point puis décroître brusquement lorsque $\omega_{2,3}(t)$ prend le relais, et vice versa. Ces oscillations se traduisent par des

cycles quasi-périodiques dans la dynamique du SCN, et la convergence n'est pas atteinte de manière statique, mais le système évolue autour d'un attracteur cyclique.

Ce comportement oscillatoire, qui peut être observé en traçant les valeurs de $\omega_{i,j}(t)$ en fonction de t sur un graphique, démontre que le système, en l'absence d'inhibition ou de régulation supplémentaire, peut se comporter de manière non monotone, illustrant ainsi la sensibilité du DSL aux paramètres η et τ , et l'importance de mécanismes de contrôle supplémentaires (tels que l'inhibition latérale ou le clipping) pour éviter des comportements indésirables.

C. Illustration par simulation en Python

Pour mettre en évidence ces deux régimes de comportement, on peut se référer à un exemple de simulation en Python. Dans le cas à 3 entités, la mise à jour additive est appliquée avec un taux d'apprentissage modéré et un coefficient de décroissance suffisant pour assurer une convergence exponentielle. En revanche, dans le cas à 4 entités, un modèle multiplicatif avec des paramètres agressifs engendre des oscillations. Voici un extrait de pseudo-code illustratif pour le cas à 3 entités :

```

import numpy as np
import matplotlib.pyplot as plt

# Définition des synergies stationnaires pour 3 entités
S = {(1,2): 0.8, (1,3): 0.7, (2,3): 0.2}
# Assurer la symétrie
for (i, j), val in list(S.items()):
    S[(j, i)] = val

eta = 0.05 # Taux d'apprentissage
tau = 1.0 # Coefficient de décroissance
num_iterations = 30

# Initialisation des poids (petit bruit autour de 0)
np.random.seed(42)
weights = {(i, j): np.random.uniform(-0.01, 0.01) for (i, j) in S.keys()}

# Enregistrement de l'évolution des poids pour chaque paire
trajectories = {key: [] for key in weights.keys()}

for t in range(num_iterations):
    for (i, j) in weights.keys():
        omega = weights[(i, j)]
        delta = eta * (S[(i, j)] - tau * omega)
        weights[(i, j)] += delta
        trajectories[(i, j)].append(weights[(i, j)])

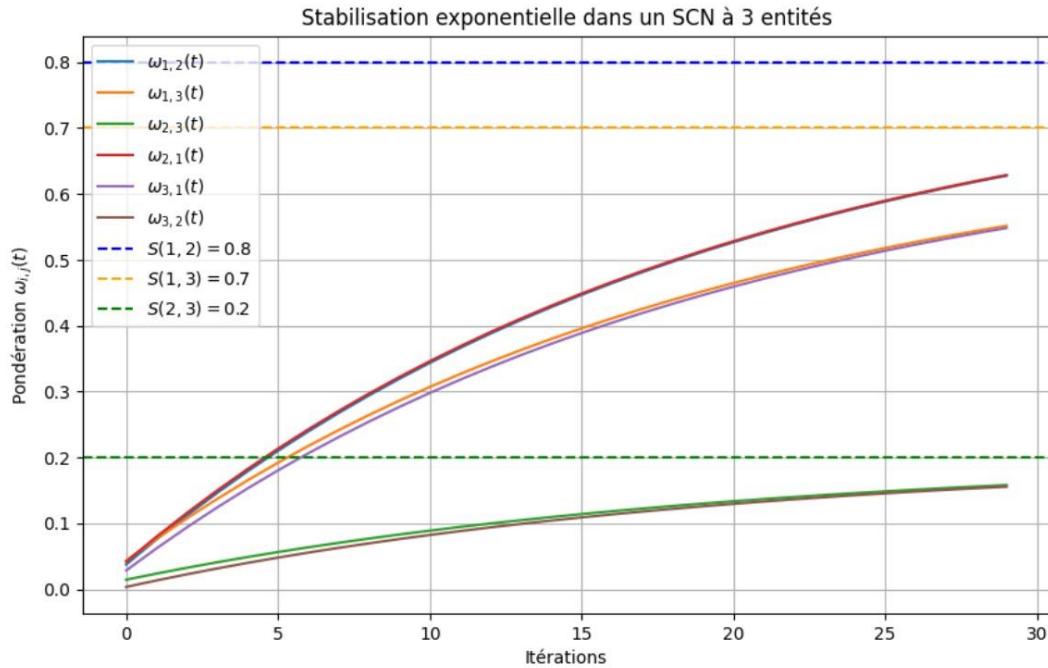
# Tracé des trajectoires
plt.figure(figsize=(10, 6))
for (i, j), traj in trajectories.items():
    plt.plot(traj, label=f'$\omega_{\{i\},\{j\}}(t)$')
plt.axhline(0.8, color='blue', linestyle='--', label='$S(1,2)=0.8$')

```

```

plt.axhline(0.7, color='orange', linestyle='--', label='$S(1,3)=0.7$')
plt.axhline(0.2, color='green', linestyle='--', label='$S(2,3)=0.2$')
plt.title("Stabilisation exponentielle dans un SCN à 3 entités")
plt.xlabel("Itérations")
plt.ylabel("Pondération $\omega_{i,j}(t)$")
plt.legend()
plt.grid()
plt.show()

```



Pour le cas à 4 entités, une variante multiplicative avec des paramètres plus élevés (par exemple, $\eta = 0.8$ et $\tau = 0.2$) et l'introduction de termes non linéaires peut être utilisée pour simuler des oscillations, révélant un comportement cyclique dans la mise à jour des pondérations.

D. Conclusion

Les exemples illustratifs sur de petits SCN, comportant 3 ou 4 entités, démontrent de manière concrète comment la dynamique de mise à jour des pondérations $\omega_{i,j}$ peut soit converger de manière exponentielle vers un point fixe (dans le cas stationnaire où $S(i,j)$ est constant et les paramètres η et τ sont bien choisis), soit présenter des oscillations persistantes (dans un régime non linéaire ou avec des paramètres trop agressifs). Ces comportements illustrent l'importance des choix paramétriques et des mécanismes additionnels (comme l'inhibition et le clipping) pour garantir la stabilité et la robustesse de l'auto-organisation du SCN dans un DSL. Ces exemples à petite échelle servent de laboratoire expérimental permettant de vérifier théoriquement la condition de contraction et de comprendre, par analogie, comment la dynamique se généralisera dans des systèmes de plus grande taille comportant de nombreux attracteurs et des interactions complexes.

4.5.2. Énergie ou Pseudo-Énergie

Une façon très féconde de **comprendre** la dynamique d'un **SCN** (Synergistic Connection Network) est d'introduire l'idée d'une **fonction d'énergie** (ou **pseudo-énergie**) qui, en quelque sorte, **mesure** la qualité de la configuration ω . Cette idée, inspirée de la **physique statistique** (chap. 2.4), permet d'envisager la mise à jour $\omega(t)$ comme une forme de "descente" (ou de quasi-descente) dans un **paysage** énergétique — ou du moins comme un processus régulé par un critère qui ressemble à une minimisation.

$$4.5.2.1. \text{ Cas simple : } \mathcal{J}(\Omega) = -\sum \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum (\omega_{i,j})^2 \quad (2.4.3)$$

Dans cette section, nous développons de manière approfondie le modèle énergétique qui sous-tend la mise à jour des pondérations dans un **Deep Synergy Learning (DSL)**. L'idée centrale est de définir une **fonction d'énergie** ou **fonction potentielle** $\mathcal{J}(\Omega)$ sur l'ensemble des pondérations Ω du **Synergistic Connection Network (SCN)**, de telle sorte que la dynamique d'auto-organisation puisse être interprétée comme une descente vers un minimum de cette énergie. La forme simple considérée ici est

$$\mathcal{J}(\Omega) = -\sum_{(i,j)} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{(i,j)} (\omega_{i,j})^2,$$

où $S(i,j)$ désigne la **synergie** entre les entités \mathcal{E}_i et \mathcal{E}_j , $\omega_{i,j}$ représente la **pondération** ou la force du lien entre ces entités, et τ est un paramètre qui module la décroissance des pondérations. Cette fonction \mathcal{J} traduit l'intuition que l'on souhaite **maximiser** la synergie globale entre les entités tout en empêchant une croissance excessive des pondérations.

D'un point de vue interprétatif, le terme $-\sum \omega_{i,j} S(i,j)$ favorise l'augmentation des liens lorsque la synergie est élevée. En effet, plus $S(i,j)$ est grand, plus le produit $\omega_{i,j} S(i,j)$ contribue négativement à \mathcal{J} ; puisque l'objectif de la descente consiste à minimiser \mathcal{J} , la dynamique incite à augmenter $\omega_{i,j}$ pour réduire la valeur absolue de ce terme négatif, ce qui correspond à renforcer les connexions entre entités fortement synergiques. À l'inverse, le terme de régularisation $\frac{\tau}{2} \sum (\omega_{i,j})^2$ pénalise la croissance excessive des pondérations, en augmentant l'énergie si les valeurs de $\omega_{i,j}$ deviennent trop grandes. Ce compromis permet ainsi d'éviter une **croissance non contrôlée** des liens, garantissant que la solution convergera vers un équilibre.

Pour rendre cette intuition plus formelle, nous pouvons envisager une analyse variationnelle dans un cadre continu. Supposons que la dynamique de mise à jour soit modélisée par une équation différentielle approchée

$$\frac{d \omega_{i,j}}{dt} = \eta [S(i,j) - \tau \omega_{i,j}],$$

où η est le taux d'apprentissage, et où l'on suppose que $S(i,j)$ ne dépend pas de $\omega_{i,j}$. Dans ce cas, la dérivée partielle de \mathcal{J} par rapport à $\omega_{i,j}$ est donnée par

$$\frac{\partial \mathcal{J}}{\partial \omega_{i,j}} = -S(i,j) + \tau \omega_{i,j}.$$

La descente de gradient, qui tend à minimiser \mathcal{J} , correspond à une dynamique

$$\frac{d \omega_{i,j}}{dt} = -\eta \frac{\partial \mathcal{J}}{\partial \omega_{i,j}} = \eta [S(i,j) - \tau \omega_{i,j}].$$

Cette équation confirme que la dynamique de mise à jour des pondérations est en effet équivalente à une **descente de gradient** sur \mathcal{J} . Au point fixe stationnaire, on pose $\frac{d \omega_{i,j}}{dt} = 0$ et l'on obtient

$$S(i,j) - \tau \omega_{i,j}^* = 0 \quad \Rightarrow \quad \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Ainsi, la solution d'équilibre, ou **attracteur**, est directement liée à la synergie $S(i,j)$ et régulée par le coefficient τ .

Il convient toutefois de noter que cette formulation constitue un **cas simple** dans lequel la synergie $S(i,j)$ est supposée constante au cours du temps et indépendante des pondérations. Dans des situations réelles, notamment lorsque des mécanismes non linéaires (inhibition latérale, couplages multiplicatifs, etc.) interviennent, la fonction \mathcal{J} peut inclure des termes supplémentaires, et la descente de gradient n'est plus exacte. C'est pourquoi on parle parfois de **pseudo-énergie**, car la dynamique réelle du DSL tente de réduire \mathcal{J} mais peut ne pas garantir une décroissance monotone à chaque itération en raison de la présence de rétroactions complexes.

La présence du terme de régularisation $\frac{\tau}{2} \sum (\omega_{i,j})^2$ joue un rôle essentiel, car il agit comme un **coût** associé au maintien d'un lien fort. Ce mécanisme empêche une croissance illimitée des pondérations et assure qu'en l'absence de synergie suffisante, les liens se désactivent progressivement. Ainsi, l'équilibre atteint, $\omega_{i,j}^* = \frac{S(i,j)}{\tau}$, représente le compromis optimal entre le renforcement des liens et la régulation nécessaire pour préserver la **stabilité** du système.

En résumé, la fonction d'énergie

$$\mathcal{J}(\boldsymbol{\Omega}) = - \sum_{(i,j)} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{(i,j)} (\omega_{i,j})^2,$$

formalise l'intuition selon laquelle le système cherche à **maximiser** la synergie globale $\sum \omega_{i,j} S(i,j)$ tout en **pénalisant** la croissance excessive des liens via le terme quadratique. La dynamique induite par la mise à jour additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$$

correspond alors à une descente de gradient sur \mathcal{J} dans un cas simple. Cette approche permet d'atteindre un point fixe stable caractérisé par

$$\omega_{i,j}^* = \frac{S(i,j)}{\tau},$$

ce qui offre une interprétation claire en termes d'**équilibre** entre l'activation induite par la synergie et la régulation imposée par la décroissance. Cette formulation constitue la base théorique sur laquelle s'appuient des mécanismes plus complexes dans le DSL, et sert de référence pour analyser la stabilité et la convergence des pondérations dans le réseau, même lorsque des non-linéarités supplémentaires sont introduites.

4.5.2.2. Descente locale vs. paysage évolutif si $S(i,j)$ change

Dans l'analyse de la dynamique d'un **Deep Synergy Learning (DSL)**, la fonction d'énergie $\mathcal{J}(\omega)$ joue un rôle central pour comprendre comment la mise à jour des pondérations $\omega_{i,j}$ tend à converger vers un minimum. Dans la partie précédente, nous avons étudié le cas simple où $S(i,j)$ est considéré comme constant, ce qui conduit à une descente locale dans un paysage énergétique fixe. Cependant, dans de nombreux systèmes réels, la synergie $S(i,j)$ varie au fil du temps, en raison de la mise à jour continue des représentations (embeddings) ou de l'évolution des règles logiques. Cette variabilité entraîne la formation d'un **paysage évolutif** où la fonction d'énergie se modifie continuellement, et, par conséquent, la dynamique d'auto-organisation ne se stabilise pas de manière statique. Nous détaillons ci-après les deux scénarios et leurs implications.

A. Descente locale lorsque $S(i,j)$ est constant

Lorsque la représentation des entités est stable, la synergie $S(i,j)$ est considérée comme une valeur fixe, ce qui signifie que, pour toute paire d'entités, le score $S(i,j)$ reste inchangé au cours des itérations. Dans ce cas, la fonction d'énergie

$$\mathcal{J}(\omega) = - \sum_{i,j} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{i,j} (\omega_{i,j})^2$$

devient indépendante du temps. La mise à jour des pondérations est alors réalisée par la règle additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

et le système se comporte comme une descente de gradient sur \mathcal{J} . En effet, en posant l'équilibre stationnaire $\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega_{i,j}^*$, on obtient immédiatement

$$S(i,j) - \tau \omega_{i,j}^* = 0 \quad \Rightarrow \quad \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Dans ce contexte, toute perturbation autour de ce point fixe est contractée par le système, de sorte que la dynamique converge de manière exponentielle vers $\omega_{i,j}^*$. La vitesse de convergence est essentiellement dictée par le facteur $1 - \eta \tau$, et l'écart à l'équilibre se réduit comme $|\omega_{i,j}(t) - \omega_{i,j}^*| \sim (1 - \eta \tau)^t$. Ce cas de **descente locale** est idéal pour l'analyse théorique, car il offre un cadre simple et linéaire où le paysage énergétique \mathcal{J} est fixe, et où les mécanismes d'auto-organisation peuvent être décrits de manière analytique.

B. Paysage évolutif lorsque $S(i, j)$ change

En réalité, de nombreux **DSL** opèrent dans des environnements non stationnaires, où la synergie $S(i, j)$ est recalculée à chaque itération. Cette situation peut se produire lorsque :

- Les **embeddings** des entités sont continuellement mis à jour par des processus de fine-tuning ou d'apprentissage continu, entraînant ainsi une modification de $S(i, j)$ basée sur la proximité vectorielle.
- De nouvelles entités sont introduites dans le système, modifiant la distribution globale et redéfinissant ainsi les relations entre les entités existantes.
- Les **règles symboliques** évoluent (par exemple, par ajout ou suppression d'axiomes), ce qui modifie la composante logique du score de synergie.

Dans un tel scénario, la fonction d'énergie devient dépendante du temps et s'écrit de manière implicite comme

$$\mathcal{J}(\boldsymbol{\omega}, t) = - \sum_{i,j} \omega_{i,j} S(i, j, t) + \frac{\tau}{2} \sum_{i,j} (\omega_{i,j})^2,$$

où $S(i, j, t)$ varie à chaque itération. La dynamique de mise à jour s'adapte alors en temps réel :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j, t) - \tau \omega_{i,j}(t)].$$

Dans ce **paysage évolutif**, le minimum de \mathcal{J} n'est pas fixe mais se déplace continuellement. Le système tente alors de « suivre » ce minimum mouvant, ce qui peut conduire à des ajustements constants des pondérations. Si les variations de $S(i, j, t)$ sont **douces**, la dynamique peut suivre le déplacement du minimum local, et le SCN se réorganise progressivement sans perdre sa cohérence. Cependant, si $S(i, j, t)$ subit des variations brutales – par exemple, en cas d'insertion massive de nouvelles entités ou de modifications importantes dans les représentations – le paysage énergétique peut se transformer de manière significative, et le système peut alors se retrouver dans une situation de **rebondissement** où les pondérations oscillent ou se réajustent continuellement, sans parvenir à une stabilisation définitive.

Ces phénomènes peuvent être analysés en considérant la dérivée temporelle de \mathcal{J} et en observant que, contrairement au cas stationnaire, le gradient $-\nabla \mathcal{J}(\boldsymbol{\omega}, t)$ varie avec t . Le système est alors contraint de s'adapter à un environnement énergétique qui évolue, et la dynamique d'auto-organisation peut présenter des comportements complexes, tels que des transitions de phase ou des oscillations persistantes.

C. Conséquences pratiques et stratégies d'adaptation

Dans un DSL opérant dans un paysage évolutif, la stabilité du SCN est moins garantie que dans le cas stationnaire. La non-stationnarité de $S(i, j, t)$ requiert des mécanismes supplémentaires pour assurer la **robustesse** du système. Parmi ces mécanismes, on peut citer :

- **Lissage des embeddings** : en réévaluant les représentations de manière graduelle, par exemple en utilisant une mise à jour du type

$$\mathbf{x}_i(t+1) \leftarrow \alpha \mathbf{x}_i(t) + (1 - \alpha) g_{t+1}(\mathbf{d}_i),$$

où g_{t+1} représente la fonction d'extraction mise à jour et $\alpha \in [0,1]$ un coefficient de lissage, on atténue les variations brusques de $S(i,j,t)$.

- **Réindexation partielle** : lorsqu'un grand changement se produit (nouveaux modèles, nouvelles entités), il peut être nécessaire de recalculer globalement ou partiellement les pondérations $\omega_{i,j}$ pour réaligner le SCN avec la nouvelle configuration du paysage énergétique.
- **Utilisation de mécanismes de contrôle** : des stratégies inspirées du recuit stochastique ou de la régulation adaptative peuvent être mises en place pour permettre au système de “suivre” le minimum local tout en évitant des oscillations excessives ou des divergences.

Ces stratégies permettent d'adapter le SCN à un environnement où $S(i,j,t)$ évolue continuellement, assurant ainsi une **adaptabilité** optimale du DSL même lorsque le paysage énergétique est en mouvement.

D. Conclusion

En résumé, lorsque la synergie $S(i,j)$ est constante, la dynamique de mise à jour des pondérations $\omega_{i,j}$ s'apparente à une **descente locale** dans un paysage énergétique fixe, conduisant à la convergence vers un point fixe défini par

$$\omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Cette situation permet une analyse simple de la stabilité du système via la descente de gradient. En revanche, lorsque $S(i,j)$ varie au fil du temps – en raison de mises à jour des représentations ou d'insertion de nouvelles données – le paysage énergétique $\mathcal{J}(\boldsymbol{\omega}, t)$ devient mobile, et le SCN doit s'adapter continuellement. Dans ce cas, le système ne se fige pas dans un attracteur fixe, mais réorganise ses pondérations pour suivre le minimum local en évolution. Ce comportement dynamique, qui peut se traduire par des réajustements constants voire des oscillations, souligne l'importance de mécanismes de contrôle supplémentaires pour garantir la **robustesse** du DSL dans des environnements non stationnaires. La distinction entre descente locale et paysage évolutif permet ainsi de mieux comprendre comment la variabilité de $S(i,j)$ influence la **stabilité**, la **convergence** et l'**adaptabilité** du système, fournissant une base théorique pour l'analyse des performances d'un DSL dans des contextes évolutifs et réels.

4.5.2.3. Cas stationnaire : $\Delta\omega_{i,j} = 0 \Rightarrow \omega_{i,j}^* = \frac{S(i,j)}{\tau}$

Dans ce cas, nous nous intéressons à la situation idéale dans laquelle la **synergie** $S(i,j)$ entre chaque paire d'entités est considérée comme **constante** au cours du temps, ce qui conduit à une dynamique de mise à jour des pondérations qui ne dépend que de la valeur actuelle de $\omega_{i,j}$ et des paramètres du système. Cette hypothèse simplificatrice permet de décrire le comportement stationnaire du système, c'est-à-dire le point fixe vers lequel tend la dynamique d'auto-organisation du **Synergistic Connection Network (SCN)**.

Soit la fonction d'**énergie** (ou pseudo-énergie) définie par

$$\mathcal{J}(\boldsymbol{\omega}) = - \sum_{i,j} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{i,j} (\omega_{i,j})^2,$$

où $S(i,j)$ est un score de synergie constant pour la paire (i,j) et τ est le paramètre de décroissance. L'objectif est de minimiser \mathcal{J} afin de trouver une configuration optimale des pondérations. En effet, le premier terme, $-\sum \omega_{i,j} S(i,j)$, encourage le renforcement des liens lorsque la synergie est élevée, tandis que le terme de régularisation $\frac{\tau}{2} \sum (\omega_{i,j})^2$ pénalise une croissance excessive des pondérations. Le compromis entre ces deux contributions conduit à un minimum d'énergie qui définit l'équilibre du système.

Pour étudier ce cas stationnaire, nous utilisons la **règle additive** de mise à jour des pondérations donnée par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où η représente le **taux d'apprentissage**. Dans un état stationnaire, on suppose que la dynamique ne fait plus évoluer les pondérations, c'est-à-dire que

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) = \omega_{i,j}^*.$$

En imposant la condition stationnaire, nous avons :

$$\omega_{i,j}^* = \omega_{i,j}^* + \eta [S(i,j) - \tau \omega_{i,j}^*].$$

La seule solution non triviale de cette équation est obtenue en annulant le terme entre parenthèses, ce qui conduit directement à

$$S(i,j) - \tau \omega_{i,j}^* = 0 \quad \Rightarrow \quad \omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Cette équation montre que, dans un environnement stationnaire, la pondération entre deux entités converge vers une valeur proportionnelle à la synergie mesurée, modulée par le paramètre de décroissance τ . En d'autres termes, si la synergie est élevée, le lien se renforce jusqu'à atteindre une valeur d'équilibre donnée par $\omega_{i,j}^* = \frac{S(i,j)}{\tau}$. À l'inverse, si $S(i,j)$ est nul ou très faible, alors le lien tendra à disparaître (i.e., $\omega_{i,j}^*$ sera proche de 0).

Pour approfondir cette analyse, on peut interpréter la règle de mise à jour comme une **descente de gradient** sur la fonction d'énergie \mathcal{J} . En effet, la dérivée partielle de \mathcal{J} par rapport à $\omega_{i,j}$ est donnée par

$$\frac{\partial \mathcal{J}}{\partial \omega_{i,j}} = -S(i,j) + \tau \omega_{i,j}.$$

La dynamique de mise à jour du DSL, écrite en termes continus, s'exprime par

$$\frac{d \omega_{i,j}}{dt} = -\eta \frac{\partial \mathcal{J}}{\partial \omega_{i,j}} = \eta [S(i,j) - \tau \omega_{i,j}],$$

ce qui est exactement la forme de notre règle de mise à jour. Par conséquent, l'évolution des pondérations est guidée par le gradient de \mathcal{J} et tend à réduire l'énergie du système jusqu'à ce que le gradient soit nul, c'est-à-dire quand $\omega_{i,j}$ atteint l'équilibre.

Un **exemple numérique** simple peut illustrer ce point. Supposons que pour une paire d'entités, la synergie stationnaire est $S(i,j) = 0.8$ et que $\tau = 1.0$. Si l'on initialise $\omega_{i,j}(0)$ à une valeur proche de zéro, la mise à jour s'effectuera de manière à faire croître $\omega_{i,j}$ vers

$$\omega_{i,j}^* = \frac{0.8}{1.0} = 0.8.$$

On observe alors, par exemple, une convergence exponentielle telle que

$$\omega_{i,j}(t) \approx 0.8[1 - (1 - \eta\tau)^t],$$

ce qui traduit la réduction exponentielle de l'écart par rapport à l'équilibre. Graphiquement, une courbe de $\omega_{i,j}(t)$ en fonction de t montrerait une ascension rapide suivie d'un plateau asymptotique à 0.8, confirmant que la dynamique d'auto-organisation a atteint un état stationnaire.

Dans le cadre de cette approche, le terme $\tau \omega_{i,j}^2$ dans la fonction d'énergie sert de **pénalité** qui empêche la croissance excessive des pondérations. C'est ce terme régulateur qui assure la **stabilité** du système en imposant un coût à la maintenance de liens trop forts, et qui, conjointement avec le terme de synergie $\omega_{i,j} S(i,j)$, détermine l'équilibre optimal.

Conclusion

En conclusion, dans un **cas stationnaire** où la synergie $S(i,j)$ reste constante, la règle de mise à jour additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$$

permet d'atteindre un point fixe $\omega_{i,j}^*$ donné par

$$\omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Cette solution correspond à un minimum local de la fonction d'énergie

$$\mathcal{J}(\boldsymbol{\omega}) = - \sum_{i,j} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{i,j} (\omega_{i,j})^2,$$

et reflète l'équilibre entre l'effet de renforcement induit par la synergie et la pénalité qui empêche la croissance illimitée des liens. La dynamique de mise à jour agit alors comme une descente de gradient sur \mathcal{J} , assurant que toute perturbation autour du point fixe est progressivement amortie, ce qui garantit la stabilité du SCN dans un environnement stationnaire. Cette analyse fournit ainsi une base théorique solide pour comprendre la convergence et la stabilité des pondérations dans un DSL, dans des conditions idéales où le paysage énergétique reste fixe.

4.5.3. Renvoi vers Chapitre 2.3 et 2.4

Les sections précédentes (4.5.1 et 4.5.2) ont souligné l'importance de certains **concepts** (attracteurs, point fixe, énergie ou pseudo-énergie) pour comprendre la dynamique d'un SCN (Synergistic Connection Network). Mais l'origine de ces notions se situe déjà en **Chapitre 2**, lorsque les **fondements théoriques** du DSL (Deep Synergy Learning) ont été abordés, notamment la possibilité de multi-stabilité (2.3.2) ainsi que l'analogie avec la physique statistique et les systèmes dynamiques (2.4.3). Cette sous-section (4.5.3) vise à faire le lien explicite avec ces passages et à voir **comment** on exploite concrètement (i.e. de manière “appliquée”) ces idées dans un DSL.

4.5.3.1. Bref rappel des sections 2.3.2 (attracteurs multiples) et 2.4.3 (notion d'énergie)

Dans l'analyse théorique du Deep Synergy Learning (DSL), les chapitres 2.3.2 et 2.4.3 fournissent respectivement les fondations conceptuelles concernant la co-existence d'attracteurs multiples et la formulation d'une fonction d'énergie sur l'espace des pondérations. Ces deux approches, bien que relevant de domaines différents de la théorie des systèmes dynamiques, se rejoignent pour expliquer la manière dont le Synergistic Connection Network (SCN) se structure et converge vers des configurations stables.

A. Attracteurs multiples (Chap. 2.3.2)

La notion d'attracteurs multiples trouve son origine dans la théorie des systèmes dynamiques, où l'on démontre que, pour un système non linéaire, il peut exister plusieurs points fixes ou cycles stables, c'est-à-dire plusieurs attracteurs locaux. Mathématiquement, si l'on définit la fonction de mise à jour globale du SCN par

$$\mathbf{F}: \mathcal{W} \rightarrow \mathcal{W},$$

un point fixe Ω^* vérifie

$$\Omega^* = \mathbf{F}(\Omega^*).$$

Dans des systèmes complexes tels que le SCN, il est souvent constaté que l'espace des pondérations se divise en différents bassins d'attraction. Ainsi, l'initialisation de $\omega(0)$ détermine vers quel attracteur le système convergera. Cette multi-stabilité se manifeste par la formation de **clusters alternatifs** ou de configurations distinctes, chacune étant localement stable. Par analogie avec les modèles de spin (tels que le modèle d'Ising ou de Potts) et les réseaux de Hopfield, ces attracteurs multiples illustrent la possibilité pour le système d'atteindre des états globalement différents, bien que chacun satisfasse la condition d'équilibre local.

L'existence de plusieurs attracteurs implique également que la dynamique du SCN peut être sensible aux perturbations et que de légères variations dans l'état initial peuvent conduire à des résultats radicalement différents. Cette propriété, bien que complexe à analyser de manière exhaustive dans des espaces de grande dimension, constitue la base de nombreux phénomènes de **phase** et de **transition** observés dans les systèmes d'apprentissage non supervisé.

B. Notion d'énergie (Chap. 2.4.3)

Le chapitre 2.4.3 introduit l'idée d'une fonction d'énergie ou fonction potentielle $\mathcal{J}(\omega)$ définie sur l'ensemble des pondérations du SCN. Une forme simple de cette fonction est donnée par

$$\mathcal{J}(\omega) = - \sum_{i,j} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{i,j} (\omega_{i,j})^2,$$

où $S(i,j)$ représente la synergie entre les entités \mathcal{E}_i et \mathcal{E}_j et τ est le coefficient de décroissance, jouant ainsi le rôle d'un terme de régularisation. Cette fonction d'énergie est conçue de manière à ce que, lorsque l'on cherche à la minimiser (ou, de manière équivalente, à maximiser $\sum \omega_{i,j} S(i,j)$ tout en pénalisant la croissance excessive des pondérations), le système converge vers un état stable où les pondérations $\omega_{i,j}$ atteignent des valeurs optimales.

En effet, dans le cas stationnaire où $S(i,j)$ est constant, la descente de gradient sur \mathcal{J} conduit à la condition d'équilibre

$$\frac{\partial \mathcal{J}}{\partial \omega_{i,j}} = -S(i,j) + \tau \omega_{i,j} = 0,$$

ce qui implique que

$$\omega_{i,j}^* = \frac{S(i,j)}{\tau}.$$

Cette relation fournit une interprétation élégante du compromis entre le renforcement d'un lien (favorisé par une synergie élevée) et la régulation de la croissance des pondérations (imposée par le terme quadratique). La fonction d'énergie permet également de visualiser le **paysage** dans lequel évolue le SCN. Dans ce paysage, chaque attracteur correspond à un minimum local de \mathcal{J} . Des analogies fortes peuvent être établies avec des systèmes physiques, tels que les réseaux de neurones de Hopfield, où la dynamique du système converge vers des états d'énergie minimale.

C. Lien entre attracteurs multiples et fonction d'énergie

L'héritage théorique posé dans le Chapitre 2 montre que la co-existence de plusieurs attracteurs découle naturellement de la forme de la fonction d'énergie. En effet, dans un système non linéaire, $\mathcal{J}(\omega)$ peut présenter plusieurs vallées ou minima locaux, chacun correspondant à un ensemble stable de pondérations. Ainsi, en fonction de l'initialisation du système, le SCN convergera vers l'un de ces attracteurs, ce qui explique la **multi-stabilité** observée. Les attracteurs multiples sont alors interprétés comme des **clusters alternatifs** ou des configurations distinctes du réseau, chaque minimum local traduisant une organisation particulière des liens, qui peut être analysée en termes de similarité des entités et de la cohérence de leur interaction.

Ces concepts se retrouvent dans de nombreux modèles d'auto-organisation et d'apprentissage non supervisé. Par exemple, dans les réseaux de Hopfield, la dynamique converge vers des états mémorisés qui représentent des minima de l'énergie, tandis que dans les modèles de spin, la configuration du système est déterminée par l'équilibre entre les interactions locales et l'influence de l'environnement.

D. Résumé du lien entre Chapitre 2 et Chapitre 4

Le **Chapitre 2** pose les fondations théoriques en introduisant la notion d'attracteurs multiples et en définissant une fonction d'énergie $\mathcal{J}(\omega)$ qui guide la dynamique d'auto-organisation du SCN. Cette approche conceptuelle permet d'interpréter la dynamique de mise à jour des pondérations comme une **descente d'énergie** qui, dans le cas stationnaire, conduit à des points fixes stables, et dans des systèmes plus complexes, à une multi-stabilité ou à des comportements oscillatoires.

Le **Chapitre 4** poursuit cette analyse en développant de manière plus "ingénierie" les mécanismes de mise à jour concrets, tels que la règle additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

et en explorant les stratégies de contrôle (inhibition, clipping, etc.) qui permettent d'ajuster la dynamique du SCN pour éviter des comportements indésirables. Ainsi, la théorie des attracteurs et de la fonction d'énergie fournie au Chapitre 2 constitue la **base conceptuelle** sur laquelle s'appuie la mise en pratique détaillée dans le Chapitre 4, illustrant ainsi la continuité entre une vision théorique et une application pratique dans le cadre d'un DSL.

4.5.3.2. Comment on opérationnalise ces idées maintenant dans un cadre "appliqué"

Dans cette section, nous développons en détail comment les concepts théoriques introduits dans le Chapitre 2 — tels que la notion d'énergie $\mathcal{J}(\omega)$, les attracteurs multiples et la descente d'énergie — sont traduits en procédures concrètes dans un système de **Deep Synergy Learning (DSL)**. L'objectif est de montrer comment, à travers des algorithmes implémentés en code, on peut mettre en œuvre la mise à jour des pondérations $\omega_{i,j}$ de manière à faire émerger des clusters stables ou des configurations attractives, tout en adaptant le système aux évolutions de la représentation.

A. Mise en œuvre de la descente d'énergie

L'idée fondamentale est de considérer la mise à jour des pondérations comme une descente de gradient sur une fonction d'énergie, qui dans sa forme simple est donnée par

$$\mathcal{J}(\omega) = - \sum_{i,j} \omega_{i,j} S(i,j) + \frac{\tau}{2} \sum_{i,j} (\omega_{i,j})^2,$$

où $S(i,j)$ représente la synergie entre les entités \mathcal{E}_i et \mathcal{E}_j et τ contrôle la pénalité appliquée aux grandes pondérations. Dans un cadre appliqué, cette fonction d'énergie sert à guider la mise à jour des poids par une procédure itérative. Plus précisément, l'algorithme se base sur la règle de mise à jour additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

qui peut être interprétée comme une descente de gradient sur \mathcal{J} lorsque l'on observe que

$$\frac{\partial \mathcal{J}}{\partial \omega_{i,j}} = -S(i,j) + \tau \omega_{i,j}.$$

En pratique, on initialise le vecteur des pondérations $\omega(0)$ (par exemple, en partant d'un bruit faible autour de zéro) et on exécute la boucle itérative suivante :

- **Calcul du gradient local** : Pour chaque paire (i, j) , le gradient local $-S(i, j) + \tau \omega_{i,j}(t)$ est évalué. Ce terme indique la direction dans laquelle il faut modifier la pondération pour réduire l'énergie.
- **Mise à jour** : On ajuste la pondération selon

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) - \eta \frac{\partial \mathcal{J}}{\partial \omega_{i,j}} = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)].$$

- **Clipping et régularisation** : Pour éviter des valeurs aberrantes ou une croissance non contrôlée, on applique des mécanismes tels que le clipping (par exemple, forcer $\omega_{i,j}(t+1) \in [0, \omega_{\max}]$) et éventuellement des termes d'inhibition.

L'ensemble de ces opérations s'exécute de manière itérative jusqu'à ce que la variation entre deux itérations consécutives, $\| \omega(t+1) - \omega(t) \|$, devienne inférieure à un seuil prédéfini, indiquant ainsi que le système a atteint un état quasi-stationnaire.

B. Gérer la multi-stabilité et l'adaptation dans un environnement évolutif

Dans des applications réelles, le système peut être amené à évoluer, que ce soit par l'insertion de nouvelles entités ou par l'actualisation des représentations (embeddings ou règles). Cela se traduit par une modification du paysage énergétique $\mathcal{J}(\omega, t)$ au fil du temps. Pour gérer ce phénomène, plusieurs stratégies opérationnelles peuvent être mises en place :

- **Mécanismes de lissage temporel** : Afin de limiter les variations brutales des synergies $S(i, j)$ dues à des mises à jour continues des représentations, on peut utiliser un schéma de lissage, par exemple en mettant à jour l'embedding \mathbf{x}_i de manière progressive selon la formule

$$\mathbf{x}_i(t+1) \leftarrow \alpha \mathbf{x}_i(t) + (1 - \alpha) g_{t+1}(\mathbf{d}_i),$$

où g_{t+1} est la fonction d'extraction mise à jour et $\alpha \in [0, 1]$ est un coefficient de lissage. Cela permet de faire évoluer $S(i, j, t)$ de manière graduelle, de sorte que la descente de gradient sur \mathcal{J} ne soit pas perturbée par des sauts trop brusques dans le paysage énergétique.

- **Réindexation et recalibration** : En cas de modification significative du système (par exemple, l'arrivée de nombreuses nouvelles entités), il peut être nécessaire de réinitialiser ou recalibrer partiellement les pondérations $\omega_{i,j}$ pour aligner le SCN avec la nouvelle configuration des représentations.
- **Utilisation de techniques de recuit stochastique** : Pour aider le système à sortir des minima locaux peu pertinents et explorer de nouveaux attracteurs plus profonds, on peut injecter du bruit contrôlé dans la mise à jour des pondérations. Ceci est analogue à un recuit stochastique où la "température" décroît progressivement afin de permettre au système d'échapper aux attracteurs peu stables tout en se stabilisant finalement dans un minimum de \mathcal{J} .

Ces mécanismes assurent que le SCN reste **adaptatif** et capable de suivre l'évolution des représentations, même lorsque le paysage énergétique est en constante mutation.

C. Mise en œuvre opérationnelle dans un cadre appliqué

Pour traduire ces idées dans un cadre appliqué, il convient de développer un pipeline algorithmique qui intègre à la fois la descente de gradient sur \mathcal{J} et les stratégies d'adaptation nécessaires à un environnement non stationnaire. Concrètement, le pipeline peut être structuré en plusieurs étapes clés :

1. Initialisation des pondérations : Définir $\omega(0)$ avec des valeurs faibles, éventuellement avec un bruit aléatoire pour simuler des conditions réelles.

2. Calcul de la synergie : Pour chaque paire d'entités, calculer $S(i,j)$ en fonction des représentations actuelles. Dans un environnement appliqué, ce calcul peut être basé sur des embeddings extraits par des modèles pré-entraînés ou sur la vérification de règles logiques pour des entités symboliques.

3. Boucle itérative de mise à jour :

- **Évaluation du gradient** : Pour chaque lien, calculer la variation $\Delta\omega_{i,j} = \eta[S(i,j) - \tau\omega_{i,j}(t)]$.
- **Application de la mise à jour** : Actualiser $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \Delta\omega_{i,j}$.
- **Régulation** : Appliquer des mécanismes de clipping et, le cas échéant, des termes d'inhibition ou de recuit pour contrôler la croissance.

4. Surveillance de la convergence : Calculer la différence $\|\omega(t+1) - \omega(t)\|$ et suivre l'évolution de la valeur de $\mathcal{J}(\omega(t))$. Une fois ces valeurs stabilisées, le système est considéré comme convergent.

5. Adaptation en ligne : En cas de changement dans les représentations (mise à jour des embeddings ou des règles), relancer le processus d'actualisation, éventuellement en réutilisant un schéma de lissage pour éviter des réajustements trop brusques.

Ces étapes peuvent être implémentées dans un cadre logiciel (par exemple en Python, avec des bibliothèques de calcul scientifique comme NumPy et des frameworks de deep learning), permettant ainsi d'opérationnaliser les concepts théoriques issus du Chapitre 2 dans un environnement appliqué.

4.6. Extraction et Visualisation des Clusters Émergents

Au cours de la dynamique d'un **SCN** (Synergistic Connection Network), les pondérations $\omega_{i,j}$ évoluent et finissent souvent par révéler un **partitionnement** implicite (voir chapitres précédents). Une question cruciale consiste alors à **observer** et **visualiser** ces pondérations, de sorte à comprendre **comment** les clusters apparaissent et **quand** la structure du réseau se stabilise ou se modifie. Cette section (4.6) présente les principales **méthodes** pour :

- **Suivre** l'évolution de la matrice $\{\omega_{i,j}\}$ en temps réel (4.6.1),
- **Identifier** de façon automatique les clusters à partir de cette matrice (4.6.2),
- **Analyser** les évolutions temporelles éventuelles de ces regroupements (4.6.3).

4.6.1. Méthodes d'Observation

La première étape pour extraire et comprendre les **clusters** consiste à **observer** la matrice $\{\omega_{i,j}\}$ et ses changements au fil du temps. Cette approche est particulièrement importante lorsque la dynamique est complexe (multiplicative, inhibition, etc.) ou qu'il y a un **apprentissage continu** (synergie $S(i,j)$ qui se met à jour).

4.6.1.1. Suivi de la Matrice $\{\omega_{i,j}\}$ au Fil du Temps

Dans la dynamique d'**auto-organisation** d'un **Synergistic Connection Network (SCN)**, la matrice $\{\omega_{i,j}(t)\}$ joue un rôle fondamental en reflétant l'état courant du réseau à chaque itération t .

Les pondérations $\omega_{i,j}(t)$, ajustées selon une équation locale (ex. $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$), traduisent la **force de liaison** entre deux entités \mathcal{E}_i et \mathcal{E}_j . Observer leur évolution au fil du temps permet d'**analyser** la progression du SCN dans sa quête de **clusters** ou de **sous-réseaux** cohérents, tout en signalant d'éventuels déséquilibres (phénomènes d'oscillation, croissance excessive de liens, inertie). Cette démarche de *monitoring* revêt donc une valeur heuristique considérable pour comprendre si le réseau tend vers un **état stable** ou si, au contraire, il oscille ou se fragmente.

Le **suivi** de $\omega_{i,j}(t)$ consiste à archiver ou à sonder la matrice à intervalle régulier, afin de détecter comment les liens les plus significatifs se renforcent ou s'affaiblissent.

Dans un SCN de taille modeste (n de l'ordre de quelques centaines), il est envisageable de **stocker** l'intégralité de la matrice $\{\omega_{i,j}(t)\}$ à chaque itération, pour reconstituer a posteriori l'évolution complète. On dispose ainsi d'un véritable "film" de la progression, ce qui facilite la mise en évidence de la formation progressive de **blocs** ou de "zones actives" où les pondérations montent en flèche.

À l'inverse, dans un **réseau** plus volumineux (n allant jusqu'à plusieurs milliers ou dizaines de milliers), l'espace requis pour conserver la totalité des $O(n^2)$ liens à chaque étape s'avère prohibitif, et l'on recourt alors à un **échantillonnage** partiel (en ne conservant la matrice que toutes les 10 ou 50 itérations, ou en ne stockant qu'un sous-ensemble restreint de liens jugés pertinents). On peut aussi suivre des **indicateurs statistiques** (moyenne, variance, quantiles de $\omega_{i,j}$) pour dresser un portrait global de la dynamique, sans conserver chaque élément de la matrice.

Cette observation régulière renseigne non seulement sur la **formation** des **clusters** (par la concentration de liens forts à l'intérieur de certaines sous-parties), mais aussi sur la **différenciation** progressive des liens faibles et des liens forts.

Les résultats typiques montrent qu'au départ, toutes les $\omega_{i,j}$ se situent dans un intervalle homogène, souvent proche de zéro si la matrice est initialisée avec un faible bruit. Progressivement, la distribution se "bimodalise" où certains liens demeurent à des niveaux faibles tandis que d'autres s'élèvent, révélant ainsi l'émergence de groupes denses.

Parfois, un suivi fin permet de repérer des **oscillations** récurrentes dans la force de certains liens, symptôme d'une configuration non stabilisée ou d'un paramétrage problématique (par exemple, un η trop grand ou un τ trop faible). Dans ces cas, le *monitoring* constitue un outil de **diagnostic** où, si le réseau ne converge pas ou oscille sur une large amplitude, il devient possible d'ajuster les paramètres ou de vérifier que la fonction de synergie $S(i,j)$ ne présente pas de discontinuité excessive.

Le simple **observationnel** des pondérations ne suffit toutefois pas à extraire automatiquement les **communautés** ni à fournir une répartition claire de l'espace. C'est la raison pour laquelle, dans des sections ultérieures (telles que la section 4.6.2), l'on recourra à des méthodes plus **systématiques** d'extraction de clusters ou de détection de structures.

Néanmoins, l'examen visuel ou statistique régulier de la matrice $\omega_{i,j}(t)$ garde tout son intérêt en phase de développement, de **test** ou de **recherche de bugs**, puisqu'il permet de diagnostiquer l'évolution globale du SCN et de repérer s'il se dirige vers un état stable, un cycle, ou un brouillard de valeurs moyennes peu discriminantes.

Cette première étape de *monitoring* jette ainsi les bases d'une analyse plus poussée où, sans elle, il existerait un risque d'utiliser des algorithmes de détection de communautés ou de hiérarchisation sans s'apercevoir que le réseau ne s'est pas encore stabilisé ou qu'une oscillation compromet l'interprétabilité des clusters.

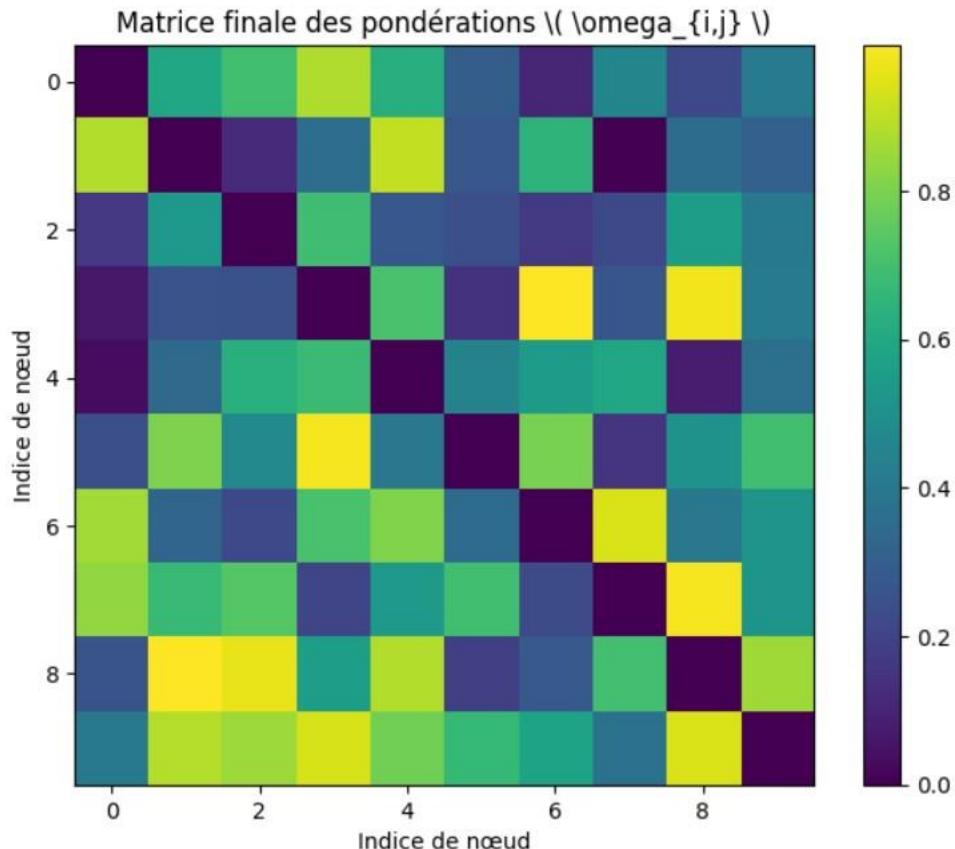
4.6.1.2. Visualisations en Heatmap, Courbes de Liaisons, ou Graphes Dynamiques

Le **Synergistic Connection Network (SCN)** met en jeu un ensemble de **pondérations** $\omega_{i,j}(t)$ dont l'évolution permet de comprendre la genèse de **clusters** ou de communautés internes. Lorsque le nombre d'entités n reste modéré, la simple consultation de la matrice $\omega(t)$ peut suffire, mais dès que l'on dépasse quelques centaines de nœuds, cette matrice devient difficile à interpréter si on se limite à un tableau numérique. Il est donc souvent essentiel d'adopter des **représentations visuelles** pour saisir le comportement du réseau, apprécier la montée ou la chute des liaisons $\omega_{i,j}(t)$ et identifier la formation progressive de blocs.

A. Heatmap (Matrice Colorée)

L'une des techniques les plus directes pour **visualiser** la matrice $\omega(t)$ consiste à en dresser une **carte colorée**. Les lignes représentent les entités \mathcal{E}_i et les colonnes \mathcal{E}_j , tandis que chaque cellule (i,j) est teintée selon la valeur de $\omega_{i,j}(t)$. Une **échelle de couleurs** allant du clair au foncé, par exemple, traduit l'intensité du lien où plus la cellule est saturée, plus la pondération est élevée. Il est possible de générer cette carte pour différentes itérations t , de manière à constituer une séquence ou un “film” illustrant l’émergence ou l’extinction de régions de forte connectivité.

Cette approche convient bien lorsque le **nombre** d'entités n demeure dans des limites raisonnables, comme quelques centaines de nœuds. On observe alors la formation de **blocs** compacts correspondant à des **clusters**, identifiables par des carrés colorés le long de la diagonale (ou, après réordonnancement, dans d'autres zones de la matrice). Lorsque n devient plus grand, il est courant de ne conserver qu'un **échantillonnage** ou de n'afficher que les liaisons dépassant un certain seuil. La heatmap reste alors un outil macroscopique, utile pour distinguer rapidement un état quasi homogène (aucun bloc net), une structure bipartite, ou plusieurs communautés distinctes.

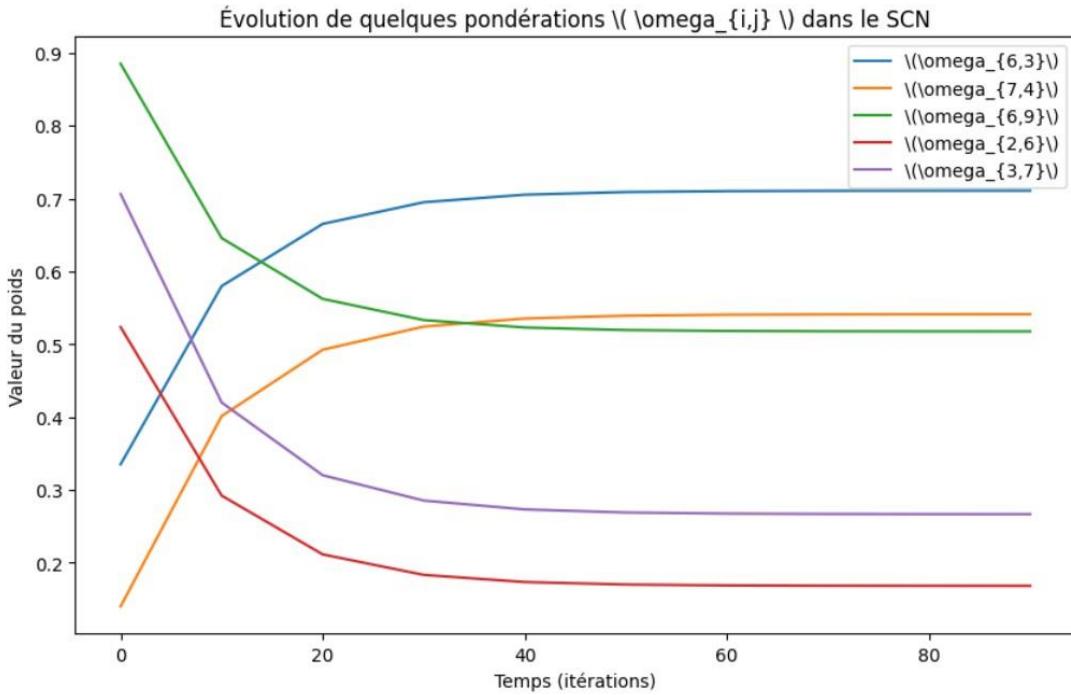


B. Courbes de Liaisons

Une autre méthode consiste à **sélectionner** un sous-ensemble de liens $\omega_{i,j}(t)$ et à tracer leur valeur au cours du temps. Cette démarche permet un suivi plus **fin** d'un petit nombre de paires jugées représentatives, qu'il s'agisse de certains liens présumés intra-cluster, de liaisons inter-cluster ou

encore de connexions ayant un rôle spécifique. On produit alors, pour chaque lien (i, j) choisi, une courbe indiquant $\omega_{i,j}(t)$ en fonction de t .

Cette représentation est particulièrement instructive pour **comprendre** la dynamique de montée ou de descente d'un lien donné. Un lien **intra-cluster** se met à croître rapidement jusqu'à un plateau, alors qu'un lien **inter-cluster** peut demeurer faible ou décroître progressivement. Parfois, on décèle des **oscillations** si $\omega_{i,j}(t)$ monte puis redescend en boucle, symptôme d'un déséquilibre dans la règle de mise à jour ou d'un paramètre mal calibré (taux d'apprentissage η trop grand, par exemple). Les *courbes de liaisons* offrent donc un moyen succinct de diagnostiquer la formation de blocs ou les difficultés de convergence, sans afficher la totalité de la matrice.



C. Graphes Dynamiques

Il est également possible de **représenter** le SCN comme un graphe dont les nœuds incarnent les entités $\{E_i\}$, et où chaque arête (i, j) est dotée d'un poids $\omega_{i,j}(t)$. On peut alors faire évoluer l'épaisseur de l'arête ou sa couleur en fonction de $\omega_{i,j}(t)$, voire employer un algorithme de placement (layout) dynamique pour rapprocher les nœuds fortement reliés et éloigner ceux qui sont faiblement connectés.

Cette **visualisation** en "graphe dynamique" offre une vision très intuitive où l'on observe littéralement le rassemblement progressif des nœuds en sous-ensembles denses. Un nœud peut être intégré simultanément à plusieurs blocs s'il dispose de liaisons importantes réparties, ou bien on voit apparaître des "communautés" distinctes dans différentes régions du plan si le layout *force-directed* est utilisé. Lorsque les pondérations $\omega_{i,j}(t)$ convergent, le graphe se fige en clusters stables ; si des oscillations surviennent, les liens changent périodiquement de force, et le graphe ne stabilise pas sa topologie.

Cette technique devient toutefois coûteuse pour de grands n . On doit le plus souvent filtrer les liaisons (ne garder que celles au-dessus d'un seuil, ou les k plus fortes par noeud) afin de limiter l'encombrement visuel. Dans un DSL de taille moyenne, l'animation du graphe dynamique constitue un atout pédagogique puissant pour visualiser la **coalescence** de communautés internes.

D. Intérêt pour l'Analyse de la Dynamique

Le choix de la **représentation** visuelle dépend du nombre d'entités, de la granularité qu'on veut observer et de la possibilité de filtrer ou non les liens. Les **heatmaps** indiquent la globalité de la matrice, les **courbes de liaisons** donnent un portrait plus détaillé de certaines paires, et les **graphes dynamiques** procurent une image structurale et esthétique du réseau en mutation. Toutes ces méthodes servent un même objectif où il s'agit d'**appréhender** la progression du SCN, **déetecter** les moments critiques tels que la croissance explosive de certaines pondérations ou l'effondrement d'autres, et **repérer** les régimes d'oscillation ou de convergence.

Dans la suite (section 4.6.2), des outils plus formels d'**identification automatique** des clusters seront présentés. Les approches de visualisation décrites ici restent néanmoins un complément précieux pour la **compréhension** et la **validation** empirique de ce qui se produit dans un DSL. Elles confèrent une dimension exploratoire indispensable pour ajuster les hyperparamètres (η, τ) ou évaluer la pertinence de la fonction de synergie $S(i, j)$. Elles constituent en somme un pont entre l'analyse intuitive (repérer des blocs colorés, des arcs épais) et l'**extraction** algorithmique de structures.

4.6.2.1. Seuil sur $\omega_{i,j}$: si $\omega_{i,j} > \theta$, on considère l'arête comme “active” → composantes connexes

Le recours à un **seuil** θ appliqué aux pondérations $\omega_{i,j}$ représente l'une des stratégies les plus élémentaires pour **extraire** des **clusters** au sein d'un **Synergistic Connection Network (SCN)**. L'idée consiste à convertir la matrice $\{\omega_{i,j}\}$ en un graphe binaire en imposant une valeur critique θ où toute liaison (i, j) dont $\omega_{i,j}$ dépasse θ est qualifiée de “active” et prend la valeur 1, tandis que les autres sont ramenées à 0. Cette binarisation aboutit à un réseau non pondéré dans lequel il devient aisément de repérer des **composantes connexes**, chacune reflétant un **cluster** potentiel.

A. Définition du seuil et matrice binaire

La règle de transformation repose sur la relation

$$A_{i,j} = \begin{cases} 1 & \text{si } \omega_{i,j} > \theta, \\ 0 & \text{sinon.} \end{cases}$$

Ce passage d'une représentation pondérée $\omega_{i,j} \in \mathbb{R}^+$ à une **matrice** binaire **A** simplifie considérablement l'analyse. Dès qu'une liaison franchit la **barre** θ , elle est considérée comme un **arc actif** dans le graphe, sans plus distinguer les nuances de valeurs au-dessus de θ .

Ce procédé reste très intuitif où seules les connexions jugées “fortes” ou “significatives” sont conservées, tandis que la multitude de liens résiduels, souvent de poids faible, est écartée afin d'éviter de brouiller la lisibilité de la structure.

B. Rôle de θ et formation des clusters

Ce paramètre θ joue un rôle crucial, car il influe directement sur la **taille** et le **nombre** de composantes qu'on verra émerger dans le graphe binaire. Si l'on choisit un seuil trop élevé, la plupart des $\omega_{i,j}$ seront en deçà de θ ; on risque alors d'obtenir de nombreuses petites composantes, voire des nœuds totalement isolés.

Au contraire, un seuil trop faible a pour effet de conserver trop de liaisons, transformant le graphe en un réseau quasi complet qui ne laisse apparaître aucune partition claire.

La distribution des valeurs $\{\omega_{i,j}\}$ peut guider l'ajustement de θ où l'on peut, par exemple, fixer θ au percentile 80 ou 90 % des valeurs non nulles, ou encore analyser l'histogramme afin de repérer un coude permettant de distinguer les liaisons fortes des liaisons faibles.

C. Détection des composantes connexes

Une fois la matrice binaire établie, le **réseau** se conçoit comme un graphe simple où $\omega_{i,j} \in \{0,1\}$. L'extraction des **composantes** connexes devient alors un problème classique en théorie des graphes.

On peut recourir à un algorithme de parcours en profondeur (*Depth-First Search*) ou en largeur (*Breadth-First Search*) où, pour chaque nœud, on visite successivement les nœuds qui lui sont reliés par des arêtes actives, puis ceux reliés à ces derniers, et ainsi de suite. L'ensemble de nœuds atteints de la sorte forme une **composante**, souvent qualifiée de cluster dans ce contexte d'auto-organisation.

Il suffit alors de relancer cette procédure pour chaque nœud non encore visité afin de recenser la totalité des composantes. La **cohésion** interne d'une composante reflète la densité d'arêtes ayant dépassé le seuil θ . Ainsi, si θ est bien choisi, on obtient en général une partition reflétant très fidèlement les groupes de nœuds fortement reliés (intra-cluster), tout en isolant les liens inter-cluster jugés insuffisants.

D. Avantages et limites

La principale **force** de cette méthode réside dans sa **simplicité** et sa **transparence** où il suffit de couper tous les liens en deçà d'un certain poids. Les *clusters* détectés comme composantes connexes sont alors très lisibles, ne nécessitent pas de calcul sophistiqué et garantissent une séparation nette entre les liens traduisant une véritable affinité et ceux considérés comme parasites ou négligeables. Toutefois, cette binarisation empêche de nuancer les liaisons supérieures au seuil où un lien à 0,95 est traité de la même manière qu'un lien à 0,55, bien que leur intensité réelle puisse différer de façon significative. De plus, l'existence d'un **unique** paramètre θ rend la solution très sensible à ce choix où un seuil mal ajusté peut conduire soit à un trop grand nombre de micro-clusters, soit à un unique bloc, ce qui peut s'éloigner de la structure sous-jacente du SCN.

4.6.2.2. Mesures de Modularity (Inspirées des Graphes) ou “Community Detection” dans le SCN

Les **méthodes** décrites précédemment (en particulier le recours à un **seuil** θ pour binariser la matrice $\omega_{i,j}$) constituent des approches rapides pour **délimiter** des blocs dans un **Synergistic Connection Network (SCN)**. Cependant, elles peinent parfois à exploiter la **graduation** subtile des valeurs $\omega_{i,j}$. Dès lors que l'on souhaite distinguer des liens à 0.95 de ceux à 0.55, il est naturel de se tourner vers des **outils** de *community detection* issus de la **théorie des graphes pondérés**. Cette famille d'algorithmes recourt souvent à une **mesure de modularité**, initialement conçue pour des graphes non pondérés, mais aisément étendue à des poids réels.

L'idée générale est de **maximiser** un indicateur, la **modularité** Q , qui évalue dans quelle mesure une partition en communautés s'écarte d'une configuration purement aléatoire. Appliquée à un SCN, où $\{\omega_{i,j}\}$ traduit la **synergie** entre entités, la modularité fournit un critère solide pour **valider** la cohésion de chaque bloc interne tout en préservant l'hétérogénéité des liens.

A. Rappel de la Notion de Modularité dans un Graphe Pondéré

Dans un graphe non pondéré, la modularité (Q) se définit par le formalisme de Newman-Girvan. On compare la fraction de liens **intra-communautés** réelle d'une partition à la fraction **attendue** si la distribution des degrés était maintenue de façon aléatoire. En notation succincte, pour un graphe non pondéré, on a

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{i,j} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

où $A_{i,j}$ décrit la matrice d'adjacence binaire ($A_{i,j} = 1$ s'il existe une arête, 0 sinon), m est le nombre total d'arêtes, et k_i le degré du nœud i . Le terme $\delta(c_i, c_j)$ vaut 1 si les nœuds i et j appartiennent à la **même** communauté, 0 dans le cas contraire.

Dans le cadre **pondéré** pertinent pour un **SCN**, on remplace $\{A_{i,j}\}$ par $\{\omega_{i,j}\}$, et on définit le degré pondéré comme $d_i = \sum_j \omega_{i,j}$. La somme totale de poids W s'identifie alors à la moitié de $\sum_{i,j} \omega_{i,j}$ pour un graphe non orienté. La **modularité** prend la forme

$$Q = \frac{1}{2W} \sum_{i,j} \left[\omega_{i,j} - \frac{d_i d_j}{2W} \right] \delta(c_i, c_j).$$

Le raisonnement demeure identique où l'on quantifie la somme des poids “intra-communauté” qui excède la somme que l'on obtiendrait dans un régime aléatoire respectant les degrés $\{d_i\}$.

B. Application dans un Synergistic Connection Network

Dans un **SCN**, chaque entité \mathcal{E}_i possède une somme pondérée $d_i = \sum_j \omega_{i,j}$. La totalité des poids est $W = 1/2 \sum_{i,j} \omega_{i,j}$. Considérer $\omega_{i,j}$ comme la matrice de poids d'un *graphe pondéré* rend possible l'emploi direct des algorithmes de *community detection* tels que Louvain, Leiden ou d'autres techniques spectrales ou basées sur la betweenness.

Leur but est de **maximiser** la modularité Q . Une “communauté” au sens modulaire se conçoit donc comme un ensemble de nœuds (entités) dont les liens internes $\omega_{i,j}$ sont significativement supérieurs à ce que proposerait un modèle d’attachement aléatoire. Intuitivement, si des entités forment un cluster dans le SCN, on doit y constater un renforcement local $\{\omega_{i,j}\}$, ce qui se traduira par un niveau élevé de modularité intra-communauté.

C. Avantages pour le SCN

Comparer cette méthode de *community detection* à la binarisation via un **seuil** θ souligne deux atouts majeurs. D’abord, la modularité prend **pleinement** en compte les variations de poids entre différents liens, et ne les réduit pas à un simple 0/1. Ensuite, de nombreux **algorithmes** éprouvés – comme Louvain ou Leiden – sont disponibles dans des bibliothèques spécialisées (NetworkX, igraph, etc.), avec une **implémentation** optimisée même pour des graphes de grande taille.

Si l’on souhaite confirmer la cohérence d’un découpage donné (par exemple, si on a une partition candidate trouvée par un autre moyen), on peut en calculer la **modularité** Q . Plus la valeur de Q s’approche de 1, plus la structuration en communautés est considérée “prononcée”. Dans les cas extrêmes, un Q très proche de 0 ou négatif indique une absence de véritables communautés au sens de la pondération interne.

D. Limites et Alternatives

La **modularité** est victime d’un phénomène connu sous le nom de “résolution limit”, qui tend à faire émerger des communautés parfois plus grandes que la “réalité” de terrain ou, au contraire, à rater certains clusters de taille réduite. D’autres métriques (comme la *partition density*, la *surprise*, ou des méthodes de *infomap*) peuvent pallier partiellement ces écueils, mais la modularité demeure un **standard** pour la *community detection* dans de nombreux scénarios.

En sus, on peut envisager des **méthodes** alternatives pour repérer des blocs pondérés où un **algorithme spectral** construit la Laplacienne du graphe $\mathbf{L} = \mathbf{D} - \mathbf{W}$, où \mathbf{W} est la matrice $\{\omega_{i,j}\}$ et \mathbf{D} la matrice diagonale des degrés pondérés. Les *vecteurs propres* associés aux plus petites valeurs propres de \mathbf{L} permettent d’identifier, via un k -means, des ensembles fortement connectés. D’autres algorithmes, comme *Girvan–Newman* adapté aux poids, suppriment itérativement les liens de forte betweenness jusqu’à l’apparition de groupes.

4.6.2.3. Lien avec le Clustering “Offline” (**k-means**, **DBSCAN**), mais ici c’est issu d’une Dynamique Interne

Le **clustering** en tant que méthode de segmentation de données se rencontre fréquemment sous forme d’algorithmes “offline” tels que **k-means**, **DBSCAN** ou encore l’approche **agglomérative**. Ces techniques s’appliquent habituellement à un **ensemble** de points $\{x_i\} \subset \mathbb{R}^d$, soumis à une distance explicite (euclidienne, cosinus, densité, etc.), dans le but de **découper** l’ensemble en blocs suffisamment cohérents pour un critère global.

Dans un **Synergistic Connection Network (SCN)**, le repérage des **clusters** repose sur une **dynamique interne** où la mise à jour itérative de $\omega_{i,j}(t)$ construit progressivement une structure de poids, révélant des **liaisons** fortes entre entités $\mathcal{E}_i, \mathcal{E}_j$. Il est alors légitime de se demander dans

quelle mesure ces approches de clustering classiques recoupent, ou se différencient, des blocs obtenus par la **lecture** de la matrice $\{\omega_{i,j}\}$ (comme décrit en 4.6.2.1 ou 4.6.2.2).

A. Rappel : k-means, DBSCAN et Clustering Offline

Lorsque l'on applique **k-means** à un ensemble de points $\{x_1, \dots, x_n\} \subset \mathbb{R}^d$, l'algorithme procède par minimisation d'une fonction d'énergie, généralement la somme intra-cluster des distances au centroïde. La taille k de la partition est fixée au préalable, et le résultat final dépend de la position initiale des centroïdes et de la distribution des données. **DBSCAN**, à l'inverse, se fonde sur une notion de **densité** (rayon ε et minPts) afin de regrouper les points "denses" et laisser les autres comme du "bruit".

Ces algorithmes sont dits **offline**, car ils nécessitent un **lot** de données fixe avant exécution et cherchent le partitionnement optimal en se basant sur une **distance** ou une **affinité** statique entre entités. Toute correction (changement de paramètres) exige de relancer l'algorithme.

B. Dynamique Interne des Pondérations dans un SCN

Dans un **SCN**, au contraire, les entités \mathcal{E}_i ne sont pas placées dans un espace fixe ; elles se relient par des pondérations $\omega_{i,j}(t)$ qui évoluent au fil d'une **règle** de mise à jour dépendant de la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$. Le SCN ne procède pas par minimisation explicite d'un critère global comme k-means ; il s'appuie plutôt sur un **processus local** qui renforce ou atténue chaque liaison. Les *clusters* émergent alors naturellement, et l'on peut distinguer :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)].$$

Les *clusters* sont donc un **résultat** de l'**auto-organisation** et non l'objet d'un algorithme de partition. Il est possible de considérer qu'une fois la matrice $\{\omega_{i,j}\}$ stabilisée (ou lue à un instant donné), on détient un "**graph pondéré**" dont on peut extraire des communautés via différentes méthodes (seuil, modularité, etc.).

C. Possibilité de Relier l'Approche "Offline" et "Online"

Un **DSL** présente d'autres subtilités, notamment la possibilité de prélever la **matrice** $\{\omega_{i,j}(t)\}$ à intervalles réguliers pour effectuer une **analyse offline**. Une approche consiste à exploiter directement les **profils de connexion** de chaque entité. En définissant pour chaque entité \mathcal{E}_i un vecteur $\boldsymbol{\omega}_i \in \mathbb{R}^n$ composé des pondérations de ses liens, on obtient une description complète de sa synergie avec l'ensemble des autres nœuds du réseau :

$$\boldsymbol{\omega}_i = (\omega_{i,1}, \omega_{i,2}, \dots, \omega_{i,n}).$$

Ces vecteurs capturent la manière dont chaque entité est connectée au sein du SCN. Une technique de **clustering offline** (comme k-means ou spectral clustering) appliquée sur l'ensemble des $\{\boldsymbol{\omega}_i\}$ permet alors d'identifier des groupes d'entités partageant des profils de connexion similaires. Ces entités, bien que distinctes en nature, manifestent des **affinités structurelles** proches en raison de la dynamique auto-organisée du réseau.

Cette technique revient à considérer que la *distance* entre deux entités $\mathcal{E}_i, \mathcal{E}_j$ peut être mesurée via une norme $\|\boldsymbol{\omega}_i - \boldsymbol{\omega}_j\|$. Par ce biais, l'on fait un pont entre la **dynamique interne** (qui a fait croître

ou décroître les $\omega_{i,j}$) et un **algorithme** de segmentation standard (k-means, DBSCAN) appliqué “après coup”.

D. Intérêt et Différences

Il existe un **intérêt** certain à comparer une **partition** issue de cette démarche (ou via la modularité sur le graph pondéré) à celle qu’on obtiendrait en prenant directement les **données brutes** de \mathcal{E}_i (ex. leurs embeddings ou leurs attributs) et en appliquant k-means/DBSCAN. Si les deux partitions coïncident, on peut conclure que la **fonction** de synergie $S(i, j)$ et la **dynamique** du SCN n’ont fait que reproduire la structure spatiale originale. S’il y a un écart notable, cela indique que la logique interne du SCN (éventuellement imprégnée de symbolique, ou d’une synergie non triviale) a façonné une structuration différente, possiblement plus riche.

La différence essentielle est que dans le **clustering offline**, on cherche à **minimiser** explicitement un critère global (somme des distances intra-cluster, densité, etc.), tandis qu’un **SCN** exécute un **processus local d’auto-organisation**. Ce processus peut faire intervenir différents paramètres (symboliques ou sub-symboliques), permettant d’autres formes de “convergence” qu’une simple structure euclidienne dans \mathbb{R}^d .

Un utilisateur pourrait se demander pourquoi ne pas appliquer un simple k-means ou DBSCAN au lieu d’un **DSL** avec mise à jour des $\omega_{i,j}$. La différence tient à la finalité. Un **DSL** gère un **flux évolutif** où les relations entre entités se modifient au fil du temps. Les **clusters** ne sont pas des partitions statiques mais émergent d’un **processus auto-organisé** intégrant des **synergies multimodales** et des **couplages symbolique–sub-symbolique** ce qui permet une adaptation continue de la structure du réseau.

4.6.3. Évolutions dans le Temps

Une fois les **clusters** identifiés (4.6.2), il est également pertinent de **suivre** leur **évolution** au fil des itérations ou des mises à jour (particulièrement si $S(i, j)$ n’est pas figé). Dans un **SCN** (Synergistic Connection Network), les changements de pondérations $\omega_{i,j}$ peuvent amener un cluster à **apparaître**, à **fusionner** avec un autre, ou encore à se **scinder** en deux ou plusieurs sous-groupes. L’observation de ces **dynamiques** se révèle cruciale pour comprendre la façon dont le réseau s’auto-organise ou se réorganise lorsque les entités, les synergies ou les paramètres du modèle varient.

4.6.3.1. Les Clusters Peuvent Apparaître, Fusionner, se Scinder

Les **clusters** formés au sein d’un **Synergistic Connection Network (SCN)** ne constituent pas nécessairement des entités fixes et immuables. Sous l’effet de la **dynamique** de mise à jour des pondérations $\omega_{i,j}(t)$ et de l’évolution potentielle de la **synergie** $S(\mathcal{E}_i, \mathcal{E}_j)$ (dans un scénario de données ou de règles changeantes), ces regroupements peuvent se **créer**, se **réunifier** ou se **dissocier** au fil du temps.

A. Apparition de nouveaux clusters

Un groupe restreint d'entités, initialement dispersées au sein du SCN, est susceptible de voir leurs liaisons $\omega_{i,j}$ progresser de manière conjointe. Les liens intra-groupes s'élèvent peu à peu, se stabilisant au-dessus d'un certain **niveau**. À mesure que ces pondérations franchissent un seuil critique (éventuellement implicite) ou qu'elles surpassent nettement les liens externes, ce **groupe** se distingue nettement du reste du réseau. Un nouveau **cluster** "apparaît" alors, témoignant de la coalescence progressive de ses membres autour d'affinités fortes.

Si les poids $\{\omega_{i,j}\}$ pour (i,j) dans un certain sous-ensemble \mathcal{C} atteignent des valeurs significatives, alors les entités $\mathcal{E}_i \in \mathcal{C}$ forment un bloc cohésif. Il suffit par exemple qu'à l'itération t les liaisons $\omega_{i,j}(t)$ dépassent successivement un seuil θ , rendant évident le fait que $\{i,j\}$ forment désormais un **noyau** ou qu'un triplet se renforce.

B. Fusion de clusters préexistants

Il arrive également que deux ou plusieurs **communautés** déjà consolidées se rapprochent l'une de l'autre. Dans ce cas, un sous-ensemble d'entités appartenant à deux blocs différents se met à voir ses pondérations inter-blocs croître de manière notable. Les liens $\omega_{i,j}$ entre ces deux agrégats, autrefois faibles, s'intensifient à la suite de modifications dans la **synergie** $S(i,j)$.

Lorsqu'un nombre suffisant de liaisons entre ces groupes dépasse un niveau appréciable, le système peut converger vers une situation où les deux ensembles précédents ne forment plus qu'un seul **cluster** fusionné. Sur le plan **graphique**, on interprète la scène comme deux composantes qui se **rejoignent**, des arêtes inter-blocs qui se renforcent, et une nouvelle macro-communauté qui émerge.

C. Scission d'un cluster en plusieurs

Inversement, un **cluster** jusque-là stable peut se **scinder** si, dans sa structure interne, certaines liaisons se mettent à décroître ou si un sous-groupe d'entités voit ses pondérations internes s'intensifier au détriment des liens avec le reste. Le sous-groupe finit par se "détacher" du noyau originel.

Sur un plan formel, on suppose qu'un cluster \mathcal{C} se fragmente en $\mathcal{C}_1 \cup \mathcal{C}_2 = \mathcal{C}$ parce que les valeurs $\{\omega_{i,j}\}_{i \in \mathcal{C}_1, j \in \mathcal{C}_2}$ baissent progressivement, tandis que les pondérations $\{\omega_{i,j}\}_{(i,j) \in \mathcal{C}_1^2}$ demeurent élevées. La discontinuité se révèle lorsqu'on observe la formation d'un véritable sous-cluster \mathcal{C}_1 qui ne conserve presque plus d'attaches fortes avec \mathcal{C}_2 .

D. Causes et conditions de ces évolutions

Les **transformations** qui mènent à l'apparition, la fusion ou la scission de clusters peuvent s'expliquer par plusieurs facteurs. L'évolution de la **synergie** $S(i,j)$ constitue une première cause. Dans un **DSL** multimodal ou symbolique, un changement d'environnement ou une mise à jour des caractéristiques d'une entité entraînent une redistribution des valeurs de similarité ou de compatibilité, ce qui modifie les $\{\omega_{i,j}\}$. Les **paramètres** internes (le taux d'apprentissage η , le coefficient τ , ou encore l'ajout d'une inhibition latérale γ) peuvent aussi favoriser la ségrégation en petits blocs ou, au contraire, la coalescence en ensembles plus vastes. Enfin, des **perturbations**

volontaires (recuit, injection de bruit) peuvent redéfinir la topologie du SCN et engendrer de nouvelles partitions.

E. Observation empirique et diagnostic

Sur le plan **pratique**, on suit la matrice $\{\omega_{i,j}(t)\}$ (section 4.6.1) ou on en extrait des communautés (section 4.6.2). À chaque itération ou après un certain nombre de pas, on compare la partition courante à la partition précédente, ce qui met en évidence l'apparition ou la fusion de blocs, la fragmentation d'une communauté, etc. Des mesures telles que la **NMI** (Normalized Mutual Information) ou l'indice de Rand offrent un moyen quantitatif de comparer deux partitions successives. On peut ainsi repérer les instants où un cluster a “**disparu**” (scission) ou où deux communautés se sont unies.

Cette **observation** permet également de comprendre si le **SCN** se stabilise ou maintient un régime oscillatoire. Lorsqu'un cluster apparaît mais se délite quelques itérations plus tard, l'auto-organisation demeure dans un état transitoire, suggérant qu'il n'existe pas encore d'équilibre.

4.6.3.2. Notation d'un “Cluster Tracking” : Indexer les Entités qui Changent de Communauté

La dynamique d'un **Synergistic Connection Network (SCN)** peut conduire à des **transformations** répétées de la structure de **clusters** ou communautés. Un groupe d'entités peut émerger, deux communautés peuvent fusionner ou, au contraire, un bloc jusqu'alors soudé peut se scinder en plusieurs parties. Dans l'optique d'analyser un tel système en évolution, il est souvent crucial d'exploiter un mécanisme de **cluster tracking**, c'est-à-dire un dispositif consignant dans quelle communauté chaque entité se situe à divers instants. Ce mécanisme autorise un suivi temporel des recompositions et, en cas de changements, il rend possible l'évaluation du degré d'instabilité ou de transition dans le SCN.

A. Principe Général du “Cluster Tracking”

Il est fréquent de ne pas recalculer la partition en communautés à chaque itération t , surtout si la dynamique est potentiellement rapide ou coûteuse.

On définit plutôt une suite de **palières** $\{t_0, t_1, \dots, t_K\}$ où l'on exécute une procédure d'**extraction** de clusters (comme décrit en 4.6.2). Chaque entité \mathcal{E}_i reçoit un label de communauté $\text{clusterID}_i(t_k)$ pour chaque instant de référence t_k . La confrontation des partitions successives fournit des indications précises sur l'évolution des communautés.

Si une entité \mathcal{E}_i change de label, cela signifie qu'elle migre d'une communauté à une autre. Si un cluster se divise, l'analyse de la partition au temps t_{k+1} permet d'observer que le groupe identifié comme “C1” au temps t_k est désormais réparti en deux nouvelles communautés distinctes.

B. Comparaison Formelle Entre Deux Partitions

Pour comparer la partition $\mathcal{P}(t_k)$ à la partition $\mathcal{P}(t_{k+1})$, il est souvent utile de recourir à des **indices** standard tels que l'**indice de Rand** ou la **NMI** (Normalized Mutual Information). Ces mesures quantifient la similarité entre deux partitions, de sorte qu'un indice élevé indique une partition presque inchangée alors qu'un indice bas reflète une fragmentation ou une fusion importante.

Sur le plan **algorithmique**, on doit parfois résoudre un **matching** entre les labels des deux partitions, puisque le cluster 3 dans $\mathcal{P}(t_k)$ peut correspondre majoritairement au cluster 2 dans $\mathcal{P}(t_{k+1})$. Il se peut qu'un algorithme de *Hungarian matching* (ou un équivalent) soit utilisé pour trouver un mapping optimal, minimisant les désaccords entre les deux partitions.

C. Changements Possibles et Suivi dans le Temps

Les changements majeurs identifiés sont :

- **Apparition d'un nouveau cluster** : un sous-ensemble d'entités jusqu'alors éparpillées ou relevant d'autres communautés se regroupent en un bloc cohésif.
- **Fusion de clusters** : deux blocs établis voient leurs liens inter-communautés $\{\omega_{i,j}\}$ croître au point de se confondre en un unique cluster.
- **Scission** : un bloc relativement stable se fragmente en deux sous-blocs distincts, dont les liens internes respectifs se renforcent et se détachent de l'ensemble initial.
- **Migration partielle** : quelques entités se déplacent d'une communauté à une autre, sans que la communauté source ni la communauté cible ne disparaissent ou ne fusionnent.

La connaissance de ces changements successifs permet, par exemple, de constituer un **journal** retracant l'historique de l'évolution du SCN. Cela se conçoit sous la forme d'une table où l'on indique à quel pas de temps \mathcal{E}_i quitte un label pour un autre, ou sous forme de diagrammes qui représentent visuellement la "trajectoire" de chaque bloc dans un plan temporel, un peu à la manière d'un diagramme de Sankey.

D. Avantages et Limites

Le *cluster tracking* éclaire la **stabilité** du SCN en révélant quelles entités demeurent ancrées dans le même cluster sur une longue période et quelles entités se montrent plus volatiles. Cela facilite un diagnostic de la **convergence** où, si la partition fluctue très peu d'une itération à l'autre, on en déduit une stabilisation. Inversement, si une proportion non négligeable de nœuds change de label à chaque palier, on soupçonne soit des oscillations internes, soit une évolution rapide de la synergie $\{\omega_{i,j}\}$.

Sur le plan computationnel, le *cluster tracking* exige d'**exécuter** ou de **répéter** régulièrement un algorithme de détection de communautés (ou un seuil, ou une modularité). Cette opération ne doit pas nécessairement être effectuée à chaque itération où l'on peut opter pour un pas plus lâche, réduisant ainsi la charge. Il subsiste un flou potentiel lorsqu'une communauté se scinde en plusieurs morceaux de taille comparable où les algorithmes de matching entre labels peuvent alors décider arbitrairement d'un label pour la majorité, et on perd une correspondance limpide pour l'autre fragment.

4.7. Exemples Concrets et Études de Cas

Les sections précédentes (4.5, 4.6) ont présenté la dynamique d'un **SCN** (Synergistic Connection Network) et les moyens d'observer/extraire les clusters. Pour aller plus loin, il est souvent instructif de mettre en pratique ces notions sur des **cas concrets** ou des **simulations**. Ce chapitre 4.7 illustre ainsi plusieurs **exemples** de configurations (sous forme de petites études de cas) où l'on voit clairement comment la synergie (artificielle, semi-aléatoire ou adaptée à une tâche) déclenche l'auto-organisation du réseau.

4.7.1.1. 10 Entités Sub-Symboliques + 5 Entités Logiques, Synergie Artificielle ou Semi-Aléatoire

Il peut être instructif, pour illustrer les mécanismes d'un **Deep Synergy Learning (DSL)** à taille **modeste**, de constituer un petit jeu de données mélangeant plusieurs entités **sub-symboliques** et quelques entités **symboliques**. L'objectif est de mettre en évidence la **cohérence** qui peut émerger entre groupes de vecteurs (sous-cluster A ou B) et blocs logiques (règles, axiomes) au terme de la dynamique de mise à jour $\omega_{i,j}(t)$.

A. Configuration des Entités

L'expérimentation suppose la présence de dix entités **sub-symboliques** $\{\mathcal{E}_1, \dots, \mathcal{E}_{10}\}$, chacune décrite par un **embedding** en dimension $d = 5$. La génération de ces vecteurs peut s'effectuer selon un **tirage** aléatoire contrôlé, en faisant apparaître deux **sous-groupes** (A et B) plus proches intérieurement (typiquement, leurs centres de gravité se distinguent). L'on obtient ainsi une répartition semi-artificielle, de sorte que la *similarité sub–sub* prenne des valeurs significativement plus élevées (par exemple, autour de 0.7) dans chacun des deux sous-groupes, tandis qu'elle tombe plutôt autour de 0.2 pour des paires inter-groupes.

On introduit en complément cinq entités **logiques** $\{\mathcal{L}_1, \dots, \mathcal{L}_5\}$. Chacune incarne un **ensemble de règles** ou un **bloc de prédicts**, susceptibles d'être plus ou moins compatibles avec les caractéristiques de A ou B. On peut, par exemple, faire en sorte que $\{\mathcal{L}_1, \mathcal{L}_2\}$ correspondent à des axiomes particulièrement alignés sur le sous-cluster A, alors que $\{\mathcal{L}_3, \mathcal{L}_4\}$ trouvent une plus grande cohérence avec B, tandis que \mathcal{L}_5 demeure relativement **neutre**.

B. Génération de la Synergie

Le calcul de la synergie $S(i, j)$ peut se structurer selon une logique **hybride** (chapitre 3), c'est-à-dire :

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

où $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^5$ représentent des embeddings sub-symboliques et $\{R_i, R_j\}$ les blocs logiques associés. Pour deux entités sub-symboliques, on se limite à la **similarité sub–sub** (par exemple un cosinus normalisé ou un noyau gaussien), alors que pour deux entités **symboliques**, un **score** binaire (0 ou 1) ou trinitaire (0, 0.5, 1) quantifie la **compatibilité** des règles. Enfin, pour un couple

sub-sym, l'on convient d'une formule adaptée (voir chap. 3.5) ou d'un paramètre β modérant l'influence d'un vecteur sub-symbolique sur un bloc logique.

La distribution semi-aléatoire est paramétrée de telle sorte que, dans chaque sous-groupe sub-symbolique A ou B, on obtienne des similarités sub-sub moyennes de 0.6–0.8, tandis que les paires inter-groupes se maintiennent à des valeurs plus faibles (0.1–0.3). Les entités logiques $\mathcal{L}_1, \mathcal{L}_2$ pourront ainsi être fortement cohérentes avec A ($S_{\text{sym}}(\mathcal{L}_1, \mathcal{E}_i) \approx 1$ si $\mathcal{E}_i \in A$), et moyennement ou faiblement cohérentes avec B, etc.

C. Lecture Attendue du SCN

Lorsque la **dynamique** de pondérations $\omega_{i,j}(t)$ se déroule (au sens de la mise à jour $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \dots$), il est prévisible de voir émerger au moins deux **communautés** majeures : la première associant les entités de **sous-cluster A** (car elles sont proches sub-symboliquement) et les entités logiques compatibles $\{\mathcal{L}_1, \mathcal{L}_2\}$, la seconde regroupant les entités de **sous-cluster B** liées à $\{\mathcal{L}_3, \mathcal{L}_4\}$. La cinquième entité logique \mathcal{L}_5 , dotée d'un profil plus neutre, pourra :

- Rejoindre l'un des deux groupes si sa compatibilité symbolique penche assez nettement dans ce sens,
- Rester relativement isolée dans un micro-bloc, ou
- Établir un pont faiblement pondéré entre les deux macros-clusters.

L'observation de la matrice $\{\omega_{i,j}(t)\}$ devrait mettre en évidence deux **blocs** de fortes pondérations internes, reflétant les deux sous-groupes sub-sym coalisés, et laisser à part une zone de liaisons plus ténues avec \mathcal{L}_5 .

D. Reproductibilité et Variations

Cette configuration, étant essentiellement **semi-artificielle**, s'apprête bien à des expérimentations comparatives. En changeant la graine aléatoire qui distribue les embeddings sub-symboliques et en modifiant légèrement la logique de $\mathcal{L}_1, \dots, \mathcal{L}_5$, on peut évaluer :

- La robustesse de la formation des mêmes blocs A– $\{\mathcal{L}_1, \mathcal{L}_2\}$ et B– $\{\mathcal{L}_3, \mathcal{L}_4\}$,
- L'existence de configurations multiples (multi-stabilité), dans lesquelles \mathcal{L}_5 intègre tel ou tel bloc en fonction des initialisations $\{\omega_{i,j}(0)\}$.

E. Extension et Généralisation

Ce **mini-prototype** avec dix entités sub-symboliques et cinq entités logiques peut aisément s'agrandir pour mieux faire ressortir le **phénomène**. En portant le nombre de sub-symboliques à 30 ou 50, réparties en deux ou trois sous-groupes artificiels, on intensifie la dimension “clustering sub-sub”. Simultanément, on peut augmenter le nombre d'entités logiques en leur associant diverses compatibilités $\{R_i\}$ afin d'obtenir un scénario plus riche, tout en restant dans une échelle accessible au *monitoring* (tracé de heatmaps, courbes, etc.). Les étapes subséquentes (sections 4.7.1.2, 4.7.1.3) détailleront comment **visualiser** l'évolution des $\omega_{i,j}(t)$ et **régler** des paramètres tels que η et τ pour mettre en évidence la trajectoire menant à la formation de **clusters** logiquement et statistiquement cohérents.

4.7.1.2. Mise en Œuvre de la Dynamique $\omega_{i,j}(t + 1)$. Courbes de Convergence, Émergence de 2 ou 3 Clusters

La configuration décrite en (4.7.1.1) – réunissant dix entités **sub-symboliques** et cinq entités **logiques** avec une **synergie** semi-aléatoire – constitue un cadre propice pour illustrer la **dynamique** interne d'un **Synergistic Connection Network (SCN)**. Après avoir fixé la structure des entités (embeddings, règles) et leur **synergie** (sub–sub, sym–sym, sub–sym), la prochaine étape consiste à **exécuter** l'algorithme de mise à jour $\omega_{i,j}(t + 1)$. L'observation de la **convergence** (ou non-convergence) et la **visualisation** des pondérations au fil des itérations révèlent la formation potentielle de deux ou trois **clusters** stabilisés.

A. Choix de la Règle de Mise à Jour

Il s'agit ici de mettre en œuvre la **formulation additive**, telle que :

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)],$$

où η est le **taux d'apprentissage**, τ la **décroissance** et $S(i, j)$ la **synergie** présumée fixe pour cet exemple (les valeurs issues du modèle sub-symbolique, symbolique ou mixte). Le système démarre avec $\omega_{i,j}(0) \approx 0$ (ou un bruit faible dans $[0, 0.05]$). En option, une **inhibition compétitive** peut s'ajouter, du type

$$-\gamma \sum_{k \neq j} \omega_{i,k}(t),$$

afin de limiter la prolifération des liens, bien que l'on puisse initialement ne pas l'inclure (ou la fixer à un faible $\gamma \approx 0.01$).

B. Boucle de Simulation

La procédure la plus classique consiste à répéter l'**itération** suivante pour un nombre fixé T_{\max} de pas de temps (par exemple 500). À chaque itération t , on évalue pour toutes les paires (i, j) :

$$\begin{aligned} \omega_{i,j}(t + 1) &= \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)] \\ &- \gamma \sum_{k \neq j} \omega_{i,k}(t) \quad (\text{si l'inhibition latérale est activée}), \end{aligned}$$

suivi éventuellement d'un **clipping** $\max\{0, \cdot\}$ si la convention exige que $\omega_{i,j} \geq 0$. Une **mise à jour synchrone** (calcul des $\omega_{i,j}(t + 1)$ à partir de toutes les $\omega_{i,j}(t)$) est préférée ici pour la simplicité. Entre deux itérations, on peut **enregistrer** la matrice $\{\omega_{i,j}(t)\}$ ou ses indicateurs sommaires (moyennes, variances).

C. Courbes de Convergence et Émergence de Blocs

En début de simulation, la plupart des pondérations restent très faibles. Puis, à mesure que l'on répète la mise à jour, les **paires** (i, j) pour lesquelles la **synergie** $S(i, j)$ est élevée s'accroissent nettement, tandis que d'autres stagnent ou régressent. Après plusieurs dizaines ou centaines

d’itérations, deux ou trois ensembles **cohérents** ont tendance à se distinguer. Si l’on considère deux **sous-groupes** sub-symboliques A et B ainsi qu’un ensemble de **règles** $\{\mathcal{L}_1, \dots, \mathcal{L}_5\}$ dont certaines s’alignent avec A et d’autres avec B, la dynamique du réseau révèle une structuration progressive des liaisons.

Le premier groupe, composé de A et des logiques $\mathcal{L}_1, \mathcal{L}_2$, voit ses liens internes s’intensifier et dépasser un seuil critique, par exemple 0.5–0.7, formant un **cluster cohérent**. Simultanément, l’autre groupe, constitué de B et des logiques $\mathcal{L}_3, \mathcal{L}_4$, suit une évolution similaire et se consolide en un bloc distinct avec des niveaux d’intensité comparables.

Quant à \mathcal{L}_5 , son positionnement reste plus incertain. Selon la synergie qu’elle entretient avec les deux clusters, elle peut rester en **marge** avec des connexions faibles ou être progressivement absorbée par l’un des deux ensembles si ses liens se renforcent.

On peut **documenter** la convergence via plusieurs moyens :

- **Courbes** $\omega_{i,j}(t)$: on choisit quelques paires (i, j) caractéristiques (intra-groupe, inter-groupe) et on trace $\omega_{i,j}(t)$ vs. t . Les liaisons intra-groupes montrent souvent une augmentation plus ou moins rapide jusqu’à un palier stable, tandis que les liaisons inter-groupe demeurent faibles ou oscillent en deçà d’un seuil.
- **Heatmap** : on extrait régulièrement la matrice $\{\omega_{i,j}(t)\}$ et on la présente sous forme de *carte colorée*. Les “zones” correspondantes aux clusters apparaissent clairement, occupant des blocs plus saturés.
- **Algorithmes** de détection de communautés (4.6.2.2) : si l’on souhaite une lecture plus automatisée, on peut appliquer, à chaque instant notable (ou à la fin de la simulation), un critère de modularité ou un algorithme Louvain sur la matrice $\{\omega_{i,j}\}$. En pratique, cela confirmera l’émergence de deux ou trois groupes majeurs.

D. Deux ou Trois Clusters au Final

En théorie, l’expérience montre que l’on obtient souvent **deux** clusters correspondant aux regroupements A et B, chacun couplé aux entités logiques qui partagent leur synergie. Toutefois, des ajustements subtils de la synergie ou des paramètres (η, τ, γ) peuvent introduire :

- **Un troisième** bloc, si une portion de A ou B se détache sous l’influence de certaines logiques, ou si \mathcal{L}_5 forme un embryon de communauté autonome.
- **Un regroupement** en un seul super-bloc si la différence sub-sym est insuffisante pour maintenir une réelle scission (par exemple τ trop faible, η trop élevé).

Dans une configuration “parfaite” (sous-cluster A resserré, B bien séparé, et logiques polarisées), la dynamique converge la plupart du temps vers **deux** agglomérations centrales, et éventuellement un “satellite” composé d’entités moins typées.

4.7.1.3. Discussion des Paramètres : η , τ , γ (Inhibition), etc.

Le **mini-exemple** présenté aux sections 4.7.1.1 et 4.7.1.2 a montré comment un **SCN** (Synergistic Connection Network) relativement modeste – rassemblant dix entités sub-symboliques et cinq entités logiques – se développe au travers d'une **dynamique** de mise à jour $\omega_{i,j}(t+1)$. L'expérience a simultanément mis en relief l'importance des **paramètres** η , τ et γ . Ces coefficients déterminent la **vitesse** à laquelle les pondérations $\omega_{i,j}$ se réajustent, la **hauteur** qu'elles peuvent atteindre avant saturation et l'**interaction compétitive** qui peut encourager chaque nœud à privilégier certains liens au détriment d'autres. Cet ensemble de réglages exerce une influence directe sur la **structure** finale : nombre de **clusters**, clarté de la partition, potentiel de bifurcation ou d'oscillation.

A. Rôle de η (Taux d'Apprentissage)

La **formulation additive** classique suppose :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où η contrôle l'ampleur de la correction appliquée à la pondération $\omega_{i,j}(t)$ à chaque itération. La manière dont η influe sur la **vitesse** et la **stabilité** de la convergence est cruciale :

- Un η **trop faible** peut rallonger considérablement la **phase transitoire** : la croissance ou la décroissance de $\omega_{i,j}$ demeurant minime à chaque pas, plusieurs centaines ou milliers d'itérations peuvent être nécessaires pour voir émerger des blocs cohérents.
- Un η **trop élevé** peut au contraire causer des **oscillations** ou une divergence si la boucle de rétroaction devient excessive. La matrice $\omega_{i,j}(t)$ se met alors à “sauter” autour de la valeur d'équilibre sans s'y stabiliser, ou elle s'aventure vers des amplitudes anormales.

Un compromis pratique consiste à essayer des valeurs de $\eta \in [0.01, 0.1]$. Dans le **scénario** illustré (4.7.1.2), un $\eta = 0.05$ s'avère un point de départ correct permettant d'atteindre un **plateau** en quelques centaines d'itérations sans oscillations majeures.

B. Rôle de τ (Décroissance)

La mise à jour comporte un terme $\tau \omega_{i,j}(t)$ que l'on **soustrait** partiellement à la pondération. Cette “force de **décroissance**” empêche $\omega_{i,j}$ de croître indéfiniment en cas de synergie $S(i,j)$ positive. Dans un cadre stationnaire simple, on retrouve un **point fixe** $\omega_{i,j}^* = S(i,j)/\tau$. Ainsi, plus τ est grand, plus ce point d'équilibre demeure **bas**, et inversement.

- Lorsque $\tau \approx 0$, le mécanisme se rapproche d'une croissance potentiellement illimitée, risquant des saturations hors de portée ou des oscillations.
- Lorsque τ est élevé, la possibilité de liens forts exige que $S(i,j)$ soit vraiment substantiel. Autrement, $\omega_{i,j}^* \approx 0$ pour la plupart des paires.

Dans l'exemple, fixer $\tau = 1$ et $\eta = 0.05$ donne un produit $\eta \tau = 0.05$. Dans un schéma additif linéaire, cette configuration garantit généralement une **convergence** stable pour un large éventail

de synergies. Pour des schémas plus sophistiqués (inhibition, non-linéarités), la stabilité peut nécessiter des conditions plus élaborées.

C. Rôle de γ (Inhibition Compétitive)

L'**inhibition latérale** (section 4.2.2.2) introduit un terme supplémentaire $-\gamma \sum_{k \neq j} \omega_{i,k}(t)$ dans la mise à jour $\omega_{i,j}(t+1)$. L'effet de cette composante est de **forcer** chaque nœud \mathcal{E}_i à limiter le nombre ou la force totale de ses liaisons en compétition. Lorsque γ est nul, chaque nœud peut, en théorie, entretenir de multiples liens “moyennement élevés”. Une inhibition plus **forte** ($\gamma \approx 0.05$) incite chaque nœud à se focaliser sur moins de liens, ou à “opter” pour un cluster plus net. Cela réduit les configurations mixtes et favorise l'apparition de **communautés** mutuellement exclusives, au risque de scinder des regroupements potentiellement plus larges si γ atteint un niveau trop élevé.

D. Autres Paramètres Pratiques

Outre η , τ et γ , d'autres choix peuvent impacter la dynamique :

- **Clipping** $\omega_{i,j} \geq 0$. Il est fréquent d'imposer $\max\{0, \cdot\}$ pour éviter que certaines pondérations deviennent négatives, ce qui peut heurter l'interprétation de “lien fort”.
- **Sparsification**. On peut décider de conserver seulement les k liaisons les plus fortes par nœud, ou celles dépassant un certain seuil dynamique (voir 4.4.2.2), pour favoriser l'émergence de blocs distincts.
- **Bruitage** ou “recuit” (4.4.3.2). Injecter un léger bruit aléatoire dans la mise à jour $\omega_{i,j}$ aide parfois à échapper à des minima locaux ou à surmonter des phénomènes de multi-stabilité non désirés.

E. Synthèse des Réglages dans l'Exemple

Dans la simulation illustrée en 4.7.1.2, des valeurs modérées $\eta = 0.05$, $\tau = 1.0$ et $\gamma \approx 0$ (ou $\gamma = 0.01$) produisent typiquement la formation de deux à trois **clusters** en moins de 300 itérations, s'accompagnant d'une stabilisation visible dans les courbes $\omega_{i,j}(t)$. En revanche, si η est trop fort (0.2 ou 0.3), le réseau peut osciller ou hésiter plus longtemps avant de trouver un arrangement stable. Si τ est trop petit (0.1 ou 0.2), beaucoup de liaisons gonflent rapidement et les clusters manquent de “relief” ; si τ est trop grand (> 2), la plupart des pondérations restent faibles et l'on ne parvient pas à dégager des blocs cohérents, sauf si $S(i,j)$ est très prononcé. L'ajout d'une inhibition γ plus substantielle (0.05 ou 0.1) va forcer le réseau à “choisir” de manière plus drastique, potentiellement conduisant à une partition plus scindée, avec des blocs plus purs.

4.7.2. Cas Robotique ou Multi-Agent

En plus des scénarios de **simulation numérique** purement abstraits (4.7.1), un **SCN** (Synergistic Connection Network) peut s'appliquer à des contextes plus “réels” ou plus “pragmatiques”, comme la coordination de **robots** ou d'**agents multiples**. Dans ce type de cadre, la **synergie** $S(i,j)$ n'est plus simplement une similarité vectorielle ou une compatibilité logique ; elle peut représenter la **complémentarité** de tâches ou la **cohérence** d'actions entre robots. On souhaite alors que le **DSL**

(Deep Synergy Learning) favorise la formation de **sous-groupes** de robots qui collaborent efficacement, tout en laissant d'autres robots moins impliqués si leur contribution ne s'imbrique pas bien.

4.7.2.1. Rappel : Flottes de Robots, Ex. 5 Robots Simulés, Synergie = Réussite Conjointe de Tâches

Les principes du **Deep Synergy Learning (DSL)**, décrits dans les sections précédentes, peuvent s'appliquer de manière naturelle à des scénarios **multi-agents**, en particulier lorsqu'on considère plusieurs **robots** ou entités dotées de capacités complémentaires. Pour illustrer cette mise en pratique, on imagine une flotte réduite de cinq robots $\{r_1, \dots, r_5\}$, chacun disposant de capteurs, effecteurs ou aptitudes particulières, et l'on définit la synergie $S(r_i, r_j)$ sur la base de leur **réussite conjointe** dans des tâches passées ou de leur **compatibilité** fonctionnelle. L'**auto-organisation** d'un **SCN** édifié autour de ces robots peut alors faire émerger un partitionnement clair : un ensemble manipulatoire, un ensemble d'exploration, ou toute autre combinaison reflétant la meilleure configuration de collaboration selon les missions à accomplir.

A. Motivation Générale

Dans une configuration à cinq robots, il existe souvent un **partage** des rôles : certains se spécialisent dans la **manipulation** (bras robotisés), d'autres dans l'**inspection** (caméras, scanners, capacité de déplacement rapide). Les missions collectives imposent parfois une coordination de robots complémentaires, par exemple un robot manipulateur ayant besoin de l'assistance d'un robot d'inspection ou d'un robot de transport. La **synergie** $S(r_i, r_j)$ quantifie alors la probabilité (ou la qualité mesurée) que les robots r_i et r_j réalisent efficacement une tâche ensemble. Pour alimenter cette synergie, on peut s'appuyer sur la **complémentarité** des modules, où capteurs et actionneurs se renforcent mutuellement. Par exemple, la présence d'un bras manipulateur couplé à un détecteur de défauts améliore certaines missions. Un **historique** de performances peut également jouer un rôle clé : si deux robots ont souvent coopéré avec succès, la valeur de $\text{Hist}(r_i, r_j)$ s'en trouve renforcée. Enfin, la **distance** entre robots peut être prise en compte si la proximité spatiale impacte la communication ou la facilité d'interaction, entraînant une décroissance de $S(r_i, r_j)$ à mesure que $\text{dist}(r_i, r_j)$ augmente.

Dans tous les cas, l'idée est de traduire la complémentarité et la réussite en un **score** S au sens du DSL, qui alimentera la mise à jour de $\omega_{i,j}(t)$.

B. Exemple Concret de Configuration

Supposons cinq robots $\{r_1, \dots, r_5\}$ dont trois $\{r_1, r_2, r_3\}$ sont dotés de bras manipulateurs. Les deux autres, $\{r_4, r_5\}$, privilégient l'inspection et la reconnaissance mobile, éventuellement plus rapides mais dépourvus de bras de préhension. Une **matrice** de synergie peut être construite en fonction des types d'interaction entre robots. Par exemple, $S(r_i, r_j) \approx 0.7$ pour deux robots manipulateurs travaillant ensemble sur une tâche d'assemblage, indiquant une forte coopération. Un manipulateur associé à un explorateur peut afficher $S(r_i, r_j) \approx 0.4$, traduisant une collaboration modérée. En

revanche, deux robots explorateurs impliqués dans une mission de montage peuvent avoir une synergie plus faible, autour de $S(r_i, r_j) \approx 0.1$ ou 0.2 , reflétant un faible apport mutuel.

En pratique, on peut enrichir la formulation :

$$S(r_i, r_j) = \alpha \text{Comp}(r_i, r_j) + \beta \text{Hist}(r_i, r_j) + \gamma \text{Dist}(r_i, r_j),$$

avec Dist prenant la forme d'une fonction décroissante de la distance géographique (si la proximité influe). Par souci de simplicité, on peut rester dans un cas stationnaire où $\{S(r_i, r_j)\}$ ne varie pas au cours du temps, pour observer la formation d'un SCN.

C. Pourquoi un SCN ?

Dans une solution **centralisée**, il serait aisément de programmer explicitement que les robots r_1, r_2, r_3 travailleront ensemble pour les missions d'assemblage, tandis que r_4, r_5 couvriront l'inspection. Cependant, l'approche DSL propose une **auto-organisation** où chaque liaison $\omega_{i,j}(t+1)$ est mise à jour localement suivant la synergie $S(r_i, r_j)$ et un terme de décroissance ou d'inhibition. Cette mise à jour répétée conduit les robots partageant une forte synergie à renforcer leurs liens, alors que des paires moins avantageuses demeurent ou se rapprochent de zéro. Au final, un **cluster** regroupe naturellement les trois manipulateurs, tandis que les deux explorateurs ou inspecteurs conservent des pondérations réduites avec ceux-ci, ou forment un autre sous-bloc.

D. Exemple Numérique

Si l'on initialise $\omega_{i,j}(0) \approx 0$, puis qu'on applique, par exemple, la **règle additive** déjà évoquée :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(r_i, r_j) - \tau \omega_{i,j}(t)],$$

avec $\eta = 0.05$ et $\tau = 1.0$, alors les paires $\{(r_1, r_2), (r_1, r_3), (r_2, r_3)\}$ bénéficieront d'un **score** $S \approx 0.7$, ce qui leur permet de croître rapidement. Les paires (r_1, r_4) , (r_2, r_5) , etc., avec $S \approx 0.2$, demeureront plus modestes ou retomberont près de zéro. Après 100 à 200 itérations, on constate l'apparition d'un bloc $\{r_1, r_2, r_3\}$ relativement isolé du bloc $\{r_4, r_5\}$. Sur le plan **graphique**, la heatmap $\{\omega_{i,j}\}$ exhibe deux carrés plus saturés (ou un seul pour $\{r_4, r_5\}$, si ces deux explorateurs se connectent l'un à l'autre avec un certain niveau de synergie).

E. Analyse et Extensions

Un tel dispositif peut naturellement évoluer. Si l'on introduit un **changement** de mission, rendant la collaboration entre un manipulateur et un explorateur tout à fait pertinente, la synergie $S(r_i, r_j)$ peut progresser pour ce couple, et $\omega_{i,j}$ s'élever au point de faire fusionner des sous-blocs antérieurement disjoints. C'est là que réside la force d'un **SCN** où l'**adaptation** se fait en continu, sans qu'un planificateur centralisé impose le regroupement. On peut aussi envisager d'intégrer un **terme d'inhibition latérale** γ pour forcer chaque robot à ne cultiver que peu de liens forts, ou un **clipping** pour éviter la saturation hors de propos.

Cet **exemple** à 5 robots illustre comment un **DSL** capture une **auto-organisation** multi-agent où la **synergie** définit la probabilité ou la qualité de réussite conjointe, et la mise à jour $\omega_{i,j}$ assure la consolidation ou l'abandon progressif d'une liaison. À l'échelle plus large, on peut imaginer des

dizaines ou centaines de robots, répartis entre diverses missions. Le modèle DSL s'étend alors de la même façon, exploitant la dimension partiellement locale et réactive de la mise à jour pour assigner de facto les robots à des **clusters** de tâches.

4.7.2.2. Observations : Un Cluster se Forme entre 3 Robots Complémentaires, les 2 Autres se Marginalisent ou se Lient Plus Faiblement

Dans le cadre d'une flotte de cinq robots, comme illustré en section 4.7.2.1, la dynamique des pondérations $\omega_{i,j}(t)$ appliquée par le Synergistic Connection Network (SCN) permet de mettre en évidence comment, sous l'hypothèse d'une synergie inégale entre certains robots, un **cluster principal** se forme spontanément. Ce cluster se compose typiquement de trois robots qui, en raison de leurs capacités complémentaires, développent des liens forts et cohérents, tandis que les deux autres robots présentent des synergies faibles avec ce groupe et avec eux-mêmes, conduisant ainsi à leur marginalisation ou à des connexions moins robustes. Nous allons maintenant développer en détail les différents aspects de cette observation en plusieurs parties.

A. Mise en Place et Déroulé de la Simulation

Le déploiement de la dynamique du SCN s'effectue en initialisant les pondérations $\omega_{i,j}(0)$ à des valeurs faibles, souvent proches de zéro, ce qui permet d'introduire un bruit initial qui simule des conditions réelles. La mise à jour des poids se fait selon la formule suivante :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(r_i, r_j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t),$$

où :

- η représente le **taux d'apprentissage** qui détermine la vitesse d'évolution des pondérations,
- $S(r_i, r_j)$ est la **synergie** entre les robots r_i et r_j , fonction qui varie de 0 à 1 en fonction de la complémentarité de leurs capacités,
- τ est le **coefficent de décroissance** qui pénalise les liens trop forts pour éviter une croissance illimitée,
- γ est le paramètre d'**inhibition** qui sert à limiter la diffusion excessive des pondérations sur plusieurs liens, encourageant ainsi une répartition plus sélective des connexions.

Dans la simulation, les paramètres sont choisis de sorte que $\eta \approx 0.05$, $\tau \approx 1.0$ et $\gamma \approx 0.02$. L'algorithme s'exécute pour environ 200 à 300 itérations, ou jusqu'à ce que les modifications des pondérations deviennent négligeables, indiquant une stabilisation du réseau.

La synergie $S(r_i, r_j)$ est définie de manière à refléter la qualité de la collaboration entre les robots. Par exemple, pour le sous-groupe de trois robots complémentaires, on attribue des valeurs élevées, typiquement comprises entre 0.7 et 0.8, alors que pour les paires impliquant les deux robots restants, les scores peuvent être de l'ordre de 0.2 à 0.3, voire moins. Cette différence dans les

valeurs de synergie conditionne la dynamique où les liens entre les robots complémentaires se renforcent tandis que les autres restent faibles ou nuls.

B. Le Cluster de Trois Robots Complémentaires

Au cours des itérations, les pondérations reliant les robots du sous-groupe complémentaire — disons r_1 , r_2 et r_3 — augmentent progressivement. Par exemple, les liens $\omega_{1,2}(t)$ et $\omega_{1,3}(t)$ vont converger vers des valeurs proches de 0.6 à 0.7, ce qui correspond, en équilibre stationnaire, à la relation

$$\omega_{i,j}^* = \frac{S(r_i, r_j)}{\tau}.$$

Avec $\tau = 1$, si $S(1,2) = 0.8$ et $S(1,3) = 0.7$, on obtient respectivement $\omega_{1,2}^* \approx 0.8$ et $\omega_{1,3}^* \approx 0.7$. Ce phénomène de renforcement des liens parmi ces trois robots traduit la formation d'un **cluster principal**. L'effet d'inhibition, via le terme $-\gamma \sum_{k \neq j} \omega_{i,k}(t)$, contribue à concentrer l'énergie de chaque robot sur les connexions les plus profitables. En conséquence, ces robots renforcent préférentiellement leurs liens entre eux, se détachant ainsi du reste de la flotte.

C. Les Deux Autres Robots : Marginalisation ou Liens Faibles

Les robots r_4 et r_5 ne bénéficient pas des mêmes synergies. Deux scénarios principaux peuvent se présenter :

- Dans un premier cas, si la synergie entre r_4 et r_5 est très faible (par exemple, $S(r_4, r_5) \approx 0.1$ ou 0.2) et que leurs liens avec le groupe principal sont également faibles (autour de 0.3), alors les pondérations associées à ces robots restent faibles. Graphiquement, les valeurs de $\omega_{4,j}(t)$ et $\omega_{5,j}(t)$ stagnent près de zéro ou oscillent à de faibles niveaux, indiquant que ces entités ne participent pas activement à la formation du cluster principal.
- Dans un second cas, si r_4 et r_5 présentent une synergie intermédiaire entre eux (par exemple, $S(r_4, r_5) \approx 0.5$), ils peuvent former un petit sous-cluster distinct, avec des pondérations convergeant vers une valeur modérée (par exemple, 0.4 à 0.45). Toutefois, leurs liens avec le trio principal restent inférieurs, ce qui les marginalise vis-à-vis du cluster dominant.

Ces situations illustrent comment, selon la répartition des synergies, le SCN peut organiser hiérarchiquement les interactions entre robots, favorisant les collaborations les plus pertinentes et isolant ou réduisant l'influence des entités moins complémentaires.

D. Stabilisation de la Configuration et Visualisation

Après un nombre suffisant d'itérations (typiquement 200 à 300), la dynamique converge vers une configuration stable où la matrice de pondérations $\{\omega_{i,j}\}$ présente un bloc de fortes connexions entre r_1 , r_2 et r_3 , et des liens significativement plus faibles pour les interactions impliquant r_4 et r_5 . La visualisation sous forme de **heatmap** permet de constater cette structuration où le bloc 3×3 correspondant au trio manipulateur affiche des valeurs proches de 0.6 à 0.8 tandis que les liens inter-clusters ou les connexions entre r_4 et r_5 apparaissent en contraste, souvent en dessous de 0.3. Des méthodes d'extraction de communautés, telles que l'indice de Silhouette ou l'algorithme de

Louvain, confirment que le cluster principal se distingue nettement, tandis que les deux autres robots restent soit isolés, soit regroupés dans un sous-ensemble à faible densité.

4.7.2.3. Visualisation Dynamique (Cycles, Stabilisation)

Les **expériences** menées dans le cadre robotique (voir 4.7.2.1 et 4.7.2.2) mettent en évidence la capacité d'un **Synergistic Connection Network** (SCN) à faire émerger, au sein d'une flotte de cinq robots, un **cluster** principal regroupant trois manipulateurs dotés d'une synergie élevée, tandis que les deux autres, moins pertinents à la même tâche, se marginalisent ou conservent des pondérations intermédiaires. Afin de **comprendre** plus finement la dynamique menant à cet état, il est précieux d'adopter des **techniques de visualisation** qui révèlent la progression pas à pas des pondérations $\omega_{i,j}(t)$, la survenue éventuelle de **cycles** ou oscillations, ainsi que le moment précis où se consolide la structure en sous-groupes.

A. Représentation sous forme de "Graph Dynamique"

Pour représenter l'évolution d'un **SCN**, on peut choisir un **graphe** pondéré dont les nœuds sont les robots $\{r_1, r_2, r_3, r_4, r_5\}$ et les arêtes sont les pondérations $\omega_{i,j}(t)$. Le principe est de produire une **animation** en temps discret où à chaque itération ou palier t , on met à jour l'**épaisseur** ou la couleur de chaque arête en fonction de $\omega_{i,j}(t)$. Les liaisons fortes apparaissent plus visibles avec des arêtes plus larges et une couleur saturée, tandis que les liaisons faibles restent minces ou quasi invisibles. Le **positionnement** des nœuds dans le plan peut être ajusté si l'on recourt à un layout type *force-directed*, ce qui permet de voir les nœuds fortement connectés se rapprocher progressivement. Certains utilisateurs préfèrent néanmoins conserver un positionnement fixe pour faciliter la comparaison dans le temps.

Cette représentation dynamique clarifie la formation du **cluster** des trois robots manipulateurs. Au départ, toutes les arêtes sont fines avec $\omega_{i,j} \approx 0$, puis celles reliant $\{r_1, r_2, r_3\}$ s'épaissent nettement, tandis que les connexions $\omega_{4,k}$ et $\omega_{5,k}$ demeurent maigres. Une fois l'itération 100 ou 150 atteinte, on constate un **noyau** visuel constitué de $\{r_1, r_2, r_3\}$, et le graphe se stabilise autour de ce bloc principal.

Cycles ou pseudo-oscillations.

Dans le cas où la dynamique de mise à jour (paramètres η, τ, γ) permet des **oscillations**, on percevrait un "va-et-vient" dans l'épaisseur de certaines arêtes, l'union de robots se renforçant brièvement puis chutant au profit d'un autre regroupement. Cet état se traduirait visuellement par une "valse" des nœuds dans le layout *force-directed*, ou par une alternance de couleurs pour des liens rivaux. Dans la configuration simple où la synergie des trois manipulateurs dépasse nettement celle des deux explorateurs, un tel phénomène survient peu, à moins que η ou γ ne soient mal calibrés.

B. Heatmap et Courbes de Pondérations

En parallèle d'un *graph* dynamique, on peut stocker et afficher la matrice $\{\omega_{i,j}(t)\}$ sous forme de **heatmap** (section 4.6.1.2). Du fait que la flotte compte seulement cinq robots, la matrice 5×5 est aisément lisible. Dès que les liens $\{\omega_{i,j}\}$ intra-cluster ($r_1 \leftrightarrow r_2, r_2 \leftrightarrow r_3$, etc.) grimpent, les cellules

correspondantes du carré deviennent plus “chaudes” (couleurs saturées), alors que les autres restent plus froides (teintes pâles). On suit ainsi la montée de la diagonale 3×3 correspondant au noyau manipulateur, et la stagnation de $\omega_{4,j}$ et $\omega_{5,j}$.

Un autre niveau de détail s’obtient en **traçant** les courbes $\omega_{i,j}(t)$ pour quelques paires repères. Pour la paire (r_1, r_2) ou (r_2, r_3) , une courbe monotone ou faiblement oscillatoire croît de 0 vers un plateau proche de 0.6–0.7, tandis que pour la paire (r_4, r_1) , elle plafonne à 0.2 ou 0.3. Le moment où l’on atteint 90 % du plateau renseigne précisément sur l’itération où la **structure** se scelle. Si aucune oscillation n’apparaît, on parle de stabilisation ; si, au contraire, on voit un comportement fluctuant, c’est signe d’un **régime cyclique** ou multi-stable.

C. Intérêt de la Visualisation pour le Diagnostic

Cette **visualisation** dynamique permet de déterminer la **temporalité** de la formation du cluster manipulateur. On aperçoit à quelle itération les liaisons $\omega_{1,2}$ et $\omega_{2,3}$ commencent à surpasser un seuil implicite, par exemple 0.5, scellant ainsi la communauté $\{r_1, r_2, r_3\}$. Les deux robots restants $\{r_4, r_5\}$ conservent des pondérations plus modestes, se trouvant donc marginalisés en fin de simulation.

En complément, la visualisation dévoile si la **dynamique** traverse une **phase transitoire** agitée (fortes oscillations puis stabilisation tardive) ou se fixe rapidement en un **cluster** ferme. À partir de là, il devient possible d’ajuster η , τ , γ afin d’obtenir un compromis satisfaisant entre **rapidité** de convergence, **contrôle** des oscillations et **clarté** de la séparation des robots.

4.7.3. Liens Pratiques

Après avoir illustré deux types de cas (4.7.1 : simulation semi-artificielle ; 4.7.2 : application robotique), il est naturel de se demander **comment** mettre en œuvre concrètement un **SCN** (Synergistic Connection Network) pour **reproduire** ces expériences ou en concevoir de nouvelles. La présente section (4.7.3) aborde quelques **pistes pratiques**, notamment :

- **Le code Python/C++ minimal** pour gérer la dynamique $\omega_{i,j}(t + 1)$ (4.7.3.1),
- **Un outil de monitoring** (“SCN Dashboard”) qui offre un **suivi** en temps réel de la formation des clusters (4.7.3.2).

4.7.3.1. Ce qu’on Pourrait Coder en Python/C++ pour Reproduire cette Dynamique en Quelques Centaines de Lignes

Il est souvent instructif de passer d’un **exemple théorique** à une **implémentation pratique**, même minimale, afin de rendre tangibles les principes de mise à jour $\omega_{i,j}(t + 1)$ d’un **Synergistic Connection Network (SCN)**. L’objectif** d’un tel prototype est de **montrer** comment, dans un code concis (Python ou C++), on peut :

- Définir l’ensemble d’entités $\{\mathcal{E}_1, \dots, \mathcal{E}_n\}$.

- Construire la **matrice** de synergie $\mathbf{S} = (S(i,j))$.
- Gérer la **matrice** $\boldsymbol{\omega}(t)$ représentant les pondérations.
- Boucler sur un nombre d'itérations T_{\max} afin d'**exécuter** la dynamique de mise à jour en s'appuyant sur les formules du **DSL**.

De cette façon, il devient possible, en quelques centaines de lignes, de **reproduire** les scénarios décrits précédemment. Que ce soit le cas **mixte** des entités sub-symboliques et logiques de la section 4.7.1, l'exemple **robotique** de la section 4.7.2, ou tout autre schéma visant à explorer la formation de clusters par **auto-organisation**, cette approche permet une implémentation souple et adaptable.

A. Structure Globale du Code

Le **cœur** de l'implémentation consiste à :

- **Définir** le **nombre** d'entités, noté n .
- **Allouer et remplir** la matrice $\mathbf{S} \in \mathbb{R}^{n \times n}$, qui fixe la **synergie** $S(i,j)$.
- **Initialiser** la matrice $\boldsymbol{\omega}(0) \approx 0$ (ou un léger bruit), éventuellement en imposant $\omega_{i,i} = 0$ s'il n'existe pas de liaison réflexive.
- Choisir les **paramètres** : η (taux d'apprentissage), τ (décroissance), γ (inhibition compétitive éventuelle) et le **nombre** d'itérations T_{\max} .
- Entrer dans la boucle principale sur $t \in \{0, \dots, T_{\max} - 1\}$, où l'on met à jour $\omega_{i,j}(t+1)$ à partir de $\omega_{i,j}(t)$ et de $S(i,j)$.
- (Optionnel) Stocker régulièrement des instantanés de la matrice $\boldsymbol{\omega}(t)$ pour les besoins de visualisation ou d'analyse.

Cette démarche se transpose aussi bien en **Python** qu'en **C++**. Dans Python, on utilise souvent **numpy** pour manipuler les matrices ; en C++, on recourra à des tableaux 2D ou à des conteneurs appropriés.

B. Exemple de Pseudo-Code (Version Additive)

Le pseudo-code ci-dessous, essentiellement en Python, illustre la **formulation additive** vue au chapitre 4.2.1 :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t)$$

(avec un éventuel γ pour l'inhibition compétitive). Le code est exprimé de manière lisible, sans prétendre à une optimisation poussée :

```
import numpy as np
```

```
# 1) Définition des paramètres et données
```

```

n = 15          # nombre d'entités
eta = 0.05      # taux d'apprentissage
tau = 1.0        # décroissance
gamma = 0.0      # si on veut ajouter l'inhibition compétitive
T_max = 300      # nombre d'itérations

# Matrice de synergie S (n x n), initialisée à zéro
S = np.zeros((n, n))
#=> Ici, on doit remplir S[i,j] selon le scénario :
# - sub-symbolique / logique mixte,
# - ou robots, etc.

# Matrice des pondérations w (n x n)
w = np.zeros((n, n))
#=> on peut ajouter un petit bruit :
# w = np.random.uniform(0.0, 0.05, (n, n))
#for i in range(n):
#    w[i,i] = 0.0

# 2) Boucle de mise à jour
for t in range(T_max):
    w_next = np.copy(w)
    for i in range(n):
        for j in range(n):
            if i != j:
                # Terme principal
                delta = eta * (S[i,j] - tau*w[i,j])
                # Inhibition compétitive
                if gamma > 1e-12:
                    # somme de w[i,k] pour k != j
                    inhibition_term = gamma * (np.sum(w[i,:]) - w[i,j])
                    delta -= inhibition_term
                w_next[i,j] = w[i,j] + delta

        # Option : clipping pour éviter w < 0
        if w_next[i,j] < 0.0:
            w_next[i,j] = 0.0

    w = w_next

# (Option) Stocker ou imprimer un résumé de la matrice w(t)
# if t % 10 == 0:
#     print(f"Iteration {t}: some stats, e.g. mean={w.mean():.3f}, max={w.max():.3f}")

```

3) A la fin, w contient la matrice des pondérations finales

Ce squelette en Python, d'une centaine de lignes ou moins, fournit déjà le cœur d'un SCN dynamique. Pour un scénario plus complexe, on étoffe la partie construction de la matrice **S** (par exemple, pour des robots ou des entités logiques, on calcule les scores sub–sub, sym–sym, sub–sym). De même, on peut rendre la décroissance ou l'inhibition plus sophistiquées.

C. Extensions et Version C++

Pour reproduire cette simulation en C++, on définit des **tableaux** ou **vecteurs** 2D (par exemple `std::vector<std::vector<double>>`) pour **S** et $\omega(t)$. Le schéma principal reste identique : une double boucle `for i in range(n): for j in range(n): ...` actualise $\omega_{i,j}$. On peut stocker la matrice à la fin de la simulation ou à intervalles réguliers.

Dans les deux langages, un **point crucial** est la manière dont la synergie **S** se construit, selon l'hétérogénéité des entités (partie sub-symbolique, symbolique, etc.). Le code de mise à jour reste lui-même identique, témoignant de la simplicité structurelle d'un SCN une fois que la **fonction** $S(i,j)$ est donnée.

D. Paramétrisation et Scénarios Multiples

Dans un **mini-programme** de ce type, on peut aisément :

10. Laisser η , τ , γ être passés en **argument** (ou chargés depuis un fichier de config) afin d'expérimenter la sensibilité de la convergence.
11. Générer **S** selon des distributions semi-aléatoires : par exemple, disjoindre des **sous-groupes** d'entités sub-symboliques avec un niveau de similarité 0.7–0.8 en interne et 0.2–0.3 à l'externe, rattacher à cela quelques entités logiques 0–1, etc.
12. Introduire un **mode évolutif** où **S** se modifie au fil du temps, ce qui exigerait un recalcul de $S(i,j)$ avant chaque itération.

Ainsi, quelques douzaines ou centaines de lignes de code suffisent à couvrir tous les **scénarios** des exemples (4.7.1) et (4.7.2), qu'il s'agisse de **robots** ou d'entités logiques/sub-symboliques mixtes.

E. Observation et Post-Traitements

Un code minimal comme ci-dessus se borne à mettre à jour ω . Dans la pratique, on procède à des **observations** pour :

- **Sauvegarder** la matrice $\omega(t)$ toutes les 10 ou 20 itérations.
- Produire une **heatmap** ou un **graph** (utilisant, par exemple, NetworkX en Python) pour illustrer l'évolution visuelle (4.6.1).
- Mesurer la **modularité** ou d'autres indices de partition (4.6.2) afin de vérifier la formation de clusters.

Ces étapes ne dépassent pas quelques lignes supplémentaires, tout en révélant la **dynamique** interne et la structure émergente.

4.7.3.2. Aperçu du “SCN Dashboard” : Un Outil de Monitoring des $\omega_{i,j}$ pour Détecer la Formation des Clusters en Temps Réel

Dans un cadre de **Deep Synergy Learning (DSL)**, l’implémentation d’un **Synergistic Connection Network (SCN)** nécessite non seulement l’élaboration d’une dynamique de mise à jour des pondérations $\omega_{i,j}$, mais également un outil permettant de **suivre** en temps réel l’évolution de ces poids afin de détecter la formation des clusters et d’identifier les points critiques de réorganisation du réseau. Cet outil, que l’on désigne par le terme “**SCN Dashboard**”, offre une interface de monitoring et de visualisation qui traduit en temps réel la dynamique d’auto-organisation du système. Dans cette section, nous décrivons en détail les principes, l’architecture et les modalités opérationnelles de ce dashboard, en nous appuyant sur des formules mathématiques et des mécanismes algorithmiques concrets.

A. Principes Généraux du SCN Dashboard

Le **SCN Dashboard** a pour objectif de fournir une **vue synthétique** de l’évolution de la matrice de pondérations $\omega(t) = \{\omega_{i,j}(t)\}$ au cours des itérations. Pour ce faire, le dashboard récupère régulièrement des instantanés de la matrice et les présente sous forme de visualisations interactives. Parmi ces visualisations, on retrouve notamment une **heatmap** de la matrice ω , des **courbes d’évolution** pour certaines liaisons représentatives, ainsi que des indicateurs quantitatifs tels que la **moyenne** et la **variance** des poids, ou encore des indices de **modularité** issus d’algorithmes de détection de communautés.

Matériellement, pour chaque itération t de la dynamique définie par

$$\begin{aligned} & \omega_{i,j}(t+1) \\ &= \omega_{i,j}(t) \\ &+ \eta[S(i,j) - \tau \omega_{i,j}(t)] \quad (\text{éventuellement complétée par des termes d’inhibition ou de clipping}), \end{aligned}$$

le dashboard capte les valeurs de $\omega_{i,j}(t)$ et calcule des métriques résumées, telles que

$$\begin{aligned} \bar{\omega}(t) &= \frac{1}{n(n-1)} \sum_{i \neq j} \omega_{i,j}(t) \\ \sigma_\omega(t) &= \sqrt{\frac{1}{n(n-1)} \sum_{i \neq j} (\omega_{i,j}(t) - \bar{\omega}(t))^2}. \end{aligned}$$

Ces scalaires, ainsi que d’autres indicateurs (par exemple, le nombre de liens dépassant un seuil ω_{\min}), sont affichés en temps réel pour permettre à l’utilisateur d’évaluer la **stabilité** du système.

B. Architecture Technique du Dashboard

Pour construire un **SCN Dashboard** opérationnel, plusieurs architectures techniques sont envisageables. La mise en œuvre peut se faire dans un environnement distribué ou en local, selon le volume des données et la nécessité d’une actualisation en temps réel. Deux approches principales sont :

Serveur intégré avec interface web : Le système de simulation du SCN s'exécute sur un serveur (par exemple, en Python ou Node.js) et stocke périodiquement les instantanés de la matrice $\omega(t)$ dans une base de données ou un fichier de log. Une interface web, développée en JavaScript (avec des bibliothèques telles que D3.js ou Plotly), interroge régulièrement ce serveur pour mettre à jour les visualisations. Cette approche permet une **interaction** en temps réel, où l'utilisateur peut ajuster dynamiquement les paramètres (par exemple, les valeurs de η , τ ou γ) via des **contrôles** graphiques (sliders, menus déroulants).

Processus séparé et visualisation post-hoc : Dans un second schéma, la simulation du SCN est exécutée en tant que processus distinct, qui enregistre périodiquement les matrices $\omega(t)$ dans un fichier. Un outil de visualisation, également développé en Python (par exemple, en utilisant Matplotlib ou Seaborn), lit ces fichiers à intervalles réguliers et met à jour les graphiques. Cette solution est plus adaptée à des analyses postérieures, mais peut être complétée par des mécanismes d'actualisation en temps réel pour des applications nécessitant une surveillance continue.

Dans les deux cas, le dashboard intègre des **mécanismes de filtrage** pour réduire la charge visuelle. Par exemple, lorsque le nombre d'entités n est très élevé, le dashboard peut n'afficher que les liens dont les pondérations dépassent un certain seuil ou utiliser des techniques de **clustering** pour regrouper visuellement les entités.

C. Visualisations et Indicateurs

Le dashboard se doit de présenter plusieurs types de visualisations pour rendre la dynamique du SCN intelligible :

- **Heatmaps** : La matrice $\omega(t)$ est représentée sous forme d'une heatmap, où chaque cellule (i, j) est colorée en fonction de la valeur de $\omega_{i,j}(t)$. Cette représentation permet de visualiser instantanément la formation de clusters (zones de couleur intense) et de détecter des zones de faibles connexions.
- **Graphes dynamiques** : Une représentation sous forme de graphe, avec des nœuds représentant les entités et des arêtes pondérées par $\omega_{i,j}(t)$, offre une perspective plus intuitive de la topologie émergente. Les algorithmes de layout de type « force-directed » repositionnent les nœuds en fonction de la force des liens, de sorte que les clusters apparaissent comme des groupes de nœuds rapprochés.
- **Courbes temporelles** : Pour certaines paires d'entités sélectionnées, le dashboard affiche l'évolution de $\omega_{i,j}(t)$ en fonction du temps, permettant d'identifier des régimes de convergence, d'oscillations ou d'instabilité.
- **Indicateurs scalaires** : Des statistiques globales telles que la moyenne, la variance et des indices de modularité (issus d'algorithmes de community detection) sont calculées et affichées. Ces indicateurs permettent de quantifier la qualité de la partition en clusters et de surveiller la stabilité du système.

L'utilisateur peut également disposer d'**outils interactifs** pour modifier les paramètres du système en direct. Par exemple, des sliders peuvent permettre de varier η , τ ou γ et d'observer immédiatement l'impact de ces modifications sur la convergence ou la formation de clusters.

D. Mise en Pratique et Exemples d'Utilisation

Pour illustrer le fonctionnement du dashboard, considérons l'exemple d'une flotte de robots dont les pondérations évoluent selon la règle du **SCN**. Le dashboard permet alors de suivre en temps réel la matrice $\omega(t)$ et de détecter, par exemple, la formation d'un **cluster principal** de robots complémentaires. Au fur et à mesure que la simulation progresse, la **heatmap** révèle l'émergence d'un bloc de valeurs élevées correspondant aux liens internes du cluster. Parallèlement, un **graphe dynamique** montre que les nœuds représentant ces robots se rapprochent physiquement, tandis que les robots moins complémentaires se déplacent en périphérie.

En complément, des **courbes temporelles** affichées pour certains liens illustrent la convergence exponentielle vers des **points fixes**, tandis que d'autres liens montrent des **oscillations**, indiquant des transitions entre différents régimes d'interaction. Ces visualisations permettent ainsi d'interpréter les dynamiques sous-jacentes et d'anticiper d'éventuels changements dans la structure du réseau.

Un tel **dashboard** ne se limite pas à la simple visualisation de la dynamique, il joue aussi un rôle essentiel d'**outil de diagnostic** et de **pilotage**. Par exemple, si les indices de **modularité** diminuent brusquement ou si la **variance** des pondérations augmente, cela peut alerter l'opérateur sur une **instabilité potentielle** ou sur la nécessité de **réajuster** les paramètres du SCN. Cela peut se faire par le biais d'une **inhibition plus forte** ou d'un **ajustement du taux d'apprentissage**.

Ce retour d'information en temps réel est particulièrement précieux dans des **applications critiques**, telles que la **coordination multi-robots en environnement dynamique**, où l'adhérence à une configuration stable et cohérente est essentielle pour la réussite de la mission.

4.8. Stabilisation Avancée : Couplages avec Chapitre 5 (Architecture SCN)

Les sections antérieures du chapitre 4 ont mis en lumière la **dynamique** d'un SCN (Synergistic Connection Network) : les mises à jour $\omega_{i,j}$, la formation de clusters, la gestion de paramètres comme η , τ , γ . Toutefois, pour qu'un **DSL** (Deep Synergy Learning) fonctionne de manière **pérenne et scalable**, il est aussi crucial de **concevoir l'architecture logicielle** appropriée, détaillée au **Chapitre 5**. Cette architecture doit non seulement soutenir la **dynamique auto-organisée** mais également permettre :

- Un **déploiement** clair des différents modules (calcul de $S(i,j)$, mise à jour de $\omega_{i,j}$, inhibition globale, etc.),
- Une **gestion** ordonnée des ressources (threads, verrous, données),
- Une **intégration** flexible (possibilité de changer de schéma de mise à jour ou d'inhibition).

4.8.1. Rôle de l'Architecture

4.8.1.1. Comment l'Organisation Logicielle (Chap. 5) Facilite le Déploiement de la Dynamique Auto-Organisée

La mise en œuvre efficace d'un **Synergistic Connection Network** (SCN) repose sur une **infrastructure logicielle** soigneusement conçue qui permet de séparer les responsabilités au sein du système et de gérer la dynamique d'auto-organisation de manière modulable et évolutive. L'organisation logicielle, telle que décrite au Chapitre 5, joue un rôle déterminant dans le déploiement de la dynamique de mise à jour des pondérations $\omega_{i,j}(t)$ et dans l'intégration des divers modules de calcul de la **synergie** $S(i,j)$, de l'**inhibition** et du **contrôle des paramètres**.

A. Modularisation et Séparation des Rôles

L'un des principes fondamentaux de l'**organisation logicielle** appliquée à un SCN est la **modularisation**. Plutôt que d'imbriquer l'ensemble des calculs dans un unique bloc de code, on divise le système en modules indépendants, chacun ayant un rôle spécifique. Par exemple, le **Module de Calcul de Synergie** est chargé d'extraire et de combiner les informations issues de diverses sources (embeddings sub-symboliques, règles symboliques, etc.) afin de produire un score $S(i,j)$. Ce module peut utiliser diverses méthodes, telles que la **similarité cosinus** ou des noyaux RBF, pour obtenir une mesure quantitative de la proximité entre deux entités. De même, le **Module de Mise à Jour des Pondérations** applique la règle d'auto-organisation, généralement de la forme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où η est le **taux d'apprentissage** et τ le **coefficient de décroissance**. Enfin, un **Module d'Inhibition/Régulation** peut être ajouté pour introduire des mécanismes de sparsification ou de

limitation (par exemple, via un terme $-\gamma \sum_{k \neq j} \omega_{i,k}(t)$) qui garantissent que la structure du réseau reste compacte et que les liens moins pertinents ne perturbent pas la dynamique globale.

Cette séparation permet d'isoler les modifications. Un changement dans la méthode de calcul de $S(i,j)$, comme le passage d'un modèle sub-symbolique à une approche hybride, n'impacte pas directement la logique de mise à jour des pondérations. Cela facilite la maintenance et l'évolution du code.

B. Flexibilité dans le Choix des Schémas

L'**organisation logicielle** favorise également la flexibilité dans le choix des schémas de mise à jour. Par exemple, la règle de mise à jour additive

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)]$$

peut être aisément remplacée par une variante multiplicative si le contexte l'exige, comme

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) \exp(\eta [S(i,j) - \tau \omega_{i,j}(t)]).$$

L'architecture logicielle modulaire permet de modifier uniquement le **Module de Mise à Jour des Pondérations** sans avoir à réécrire la totalité du système. Par ailleurs, les paramètres globaux η , τ et γ peuvent être centralisés dans un **Module de Gestion des Paramètres**, qui fournit des interfaces permettant d'ajuster ces valeurs en fonction des besoins de l'application ou des observations en temps réel.

C. Gestion Cohérente des Itérations et du Flux de Données

Un autre aspect clé de l'organisation logicielle est la gestion structurée de la boucle itérative qui orchestre le SCN. Une **boucle centrale** ou **engine** est mise en place pour :

- Appeler le **Module de Calcul de Synergie** pour recalculer $S(i,j)$ (si nécessaire, notamment dans des systèmes non stationnaires),
- Appliquer la règle de mise à jour aux pondérations $\omega_{i,j}$ à chaque itération,
- Activer des mécanismes de **monitoring** et de **visualisation** (par exemple, en enregistrant périodiquement la matrice $\omega(t)$ pour en générer des heatmaps ou des graphes dynamiques).

Ce processus est rendu encore plus robuste par l'introduction d'**observateurs** (observers) qui analysent la convergence, par exemple en mesurant la norme $\|\omega(t+1) - \omega(t)\|$ et en déclenchant des ajustements si la convergence se révèle trop lente ou si des oscillations apparaissent. De plus, la modularité de la boucle permet d'intégrer des mécanismes de contrôle comme le recuit stochastique pour échapper aux minima locaux peu satisfaisants.

D. Stabilisation et Contrôle de la Dynamique

L'**organisation logicielle** favorise une gestion fine de la dynamique d'auto-organisation grâce à des modules spécifiques qui surveillent et ajustent le comportement du SCN. Par exemple, un **Module de Stabilisation** peut intervenir en temps réel pour ajuster les paramètres η et τ en fonction des indicateurs de convergence et des oscillations détectées. Ce module peut également appliquer

des techniques de **clipping** et de **sparsification** pour empêcher une prolifération excessive de liens, garantissant ainsi la robustesse et la lisibilité de la matrice ω .

L'approche permet de tester différents scénarios. Dans un système multi-entités, la même infrastructure logicielle peut être utilisée pour différents types d'applications comme la robotique collaborative, les agents conversationnels ou les systèmes de recommandation. Il suffit de modifier le module de calcul de synergie $S(i, j)$ et d'adapter les paramètres de la boucle d'itération. Cette capacité à réutiliser et à adapter le **moteur d'auto-organisation** est l'un des avantages majeurs de l'organisation logicielle présentée au Chapitre 5.

E. Intégration Multi-Scénario et Maintenance

Enfin, l'architecture logicielle permet une **intégration multi-scénario** efficace. Qu'il s'agisse de systèmes sub-symboliques, symboliques ou hybrides, le même cadre d'auto-organisation peut être déployé avec des modules spécifiques pour le calcul de la synergie et pour la régulation des pondérations. Cette uniformisation facilite non seulement le développement initial mais aussi la maintenance à long terme, puisque chaque composant évolue de manière indépendante. Par ailleurs, les interfaces de monitoring et de contrôle offertes par le SCN Dashboard (voir section 4.7.3.2) permettent aux opérateurs de visualiser en temps réel l'évolution du réseau et d'ajuster les paramètres via des interfaces ergonomiques, rendant ainsi l'ensemble du système adaptable à des environnements en constante évolution.

4.8.1.2. Notion de Modules Distincts : “Calcul de Synergie”, “Update ω ”, “Inhibition Global ou Local”, etc.

Dans la mise en œuvre d'un **Synergistic Connection Network (SCN)**, il apparaît primordial de structurer le code en **modules distincts** afin de garantir la **clarté**, la **flexibilité** et la **maintenabilité** de l'architecture. Cette approche modulaire, qui sera développée plus en détail au Chapitre 5, permet de séparer les responsabilités liées au **calcul de la synergie**, à la **mise à jour des pondérations** $\omega_{i,j}$, et aux mécanismes de **régulation** tels que l'inhibition (globale ou locale). En procédant ainsi, le système se construit comme un ensemble de briques fonctionnelles interopérables, chacune étant aisément remplaçable ou ajustable selon le contexte d'application. Nous détaillons ci-après les principaux modules qui constituent ce cadre opérationnel.

A. Module “Calcul de Synergie”

Le **module de Calcul de Synergie** est chargé de déterminer le score $S(i, j)$ qui mesure la qualité de la relation entre deux entités \mathcal{E}_i et \mathcal{E}_j . Ce score peut être obtenu à partir de représentations sub-symboliques (telles que des **embeddings** issus de CNN, Transformers, etc.), de critères symboliques (basés sur des règles logiques, axiomes ou ontologies), ou d'une combinaison hybride de ces deux approches. Mathématiquement, on peut écrire la fonction de synergie de manière générique comme

$$S(i, j) = f(\mathbf{r}(i), \mathbf{r}(j)),$$

où $\mathbf{r}(i)$ représente la **représentation** de l'entité i (qui peut être un vecteur, un ensemble de règles, ou une concaténation de ces deux types d'informations) et f est une fonction mesurant la similarité

ou la compatibilité. Par exemple, dans un contexte purement sub-symbolique, on utilisera typiquement la **similarité cosinus** :

$$S_{\text{sub}}(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|},$$

tandis que dans un contexte symbolique, $S_{\text{sym}}(i, j)$ pourra être définie en fonction de la concordance entre deux ensembles de règles R_i et R_j . Dans un système hybride, on opère une fusion pondérée telle que

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(i, j) + (1 - \alpha) S_{\text{sym}}(i, j),$$

où $\alpha \in [0, 1]$ détermine l'importance relative des deux composantes. La modularité de ce calcul permet de modifier ou de substituer le module de synergie sans perturber la structure de la mise à jour des pondérations, rendant ainsi le système adaptable à divers types de données et de contextes.

B. Module “Update ω ”

Le **module de mise à jour des pondérations** est le cœur de la dynamique d'auto-organisation du SCN. Il applique les règles d'évolution des liens qui relient les entités entre elles. La règle de mise à jour additive classique s'exprime par :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i, j) - \tau \omega_{i,j}(t)],$$

où η est le **taux d'apprentissage** et τ représente un terme de **décroissance** qui pénalise une croissance excessive des liens. Dans certains cas, pour obtenir une dynamique plus exponentielle, on peut opter pour une variante multiplicative :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) \exp(\eta[S(i, j) - \tau \omega_{i,j}(t)]).$$

Ce module est conçu pour être indépendant du module de calcul de la synergie ; il reçoit en entrée la matrice $\mathbf{S} = \{S(i, j)\}$ et les paramètres η et τ , et renvoie la nouvelle matrice de pondérations. Par cette séparation, il devient trivial de changer le schéma de mise à jour (passer d'un modèle additif à un modèle multiplicatif, ou encore y intégrer des mécanismes de clipping ou d'inhibition) sans affecter le calcul du score de synergie.

C. Module “Inhibition Global ou Local”

Dans des systèmes complexes, il est souvent nécessaire d'introduire une **inhibition** afin d'empêcher qu'une entité ne se lie de manière excessive à un trop grand nombre de partenaires. Cette inhibition peut être implémentée de deux manières principales :

- **Inhibition locale** : Chaque entité \mathcal{E}_i se voit imposer une contrainte sur la somme des pondérations associées à ses liens. Par exemple, on peut imposer que

$$\sum_{j \neq i} \omega_{i,j} \leq \Omega_{\max},$$

où Ω_{\max} est un seuil maximal. Cela force l'entité à concentrer ses liens sur les partenaires les plus synergiques.

- **Inhibition globale** : On ajoute directement un terme d'inhibition dans la règle de mise à jour, tel que

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t),$$

où γ est un coefficient d'inhibition qui réduit la pondération si l'entité est déjà fortement connectée à plusieurs partenaires.

Ce module peut être conçu comme une fonction à part entière, qui s'applique en post-traitement de la mise à jour des $\omega_{i,j}$. En isolant ce mécanisme, on peut facilement expérimenter avec différentes stratégies d'inhibition (par exemple, en modifiant le seuil Ω_{\max} ou en ajustant le coefficient γ) sans impacter la logique de calcul de la synergie ou la mise à jour des pondérations.

D. Synergie d'Ensemble et Ordonnancement

L'architecture logicielle d'un SCN s'organise de manière séquentielle en un pipeline d'itération. À chaque itération t , le **moteur** de l'auto-organisation procède de la manière suivante :

13. Le **Module de Calcul de Synergie** est appelé pour déterminer les scores $S(i,j)$ pour toutes les paires d'entités.
14. Le **Module de Mise à Jour** applique la règle de mise à jour sur la matrice $\omega(t)$ pour générer $\omega(t+1)$.
15. Le **Module d'Inhibition** intervient ensuite pour appliquer des ajustements supplémentaires, tels que la régulation de la somme des liens ou le clipping.
16. Des **observateurs** ou **logs** sont activés pour suivre l'évolution de la matrice $\omega(t)$ et calculer des indicateurs de convergence, par exemple la norme de la différence $\|\omega(t+1) - \omega(t)\|$ ou des indices de modularité issus d'algorithmes de détection de communautés.

Ce séquençage permet d'intégrer aisément des mécanismes de **contrôle adaptatif**. Par exemple, un **Module de Paramètres** central peut ajuster dynamiquement η et τ en fonction des indicateurs de stabilité, ou activer un **recuit stochastique** pour sortir des minima locaux peu intéressants.

E. Avantages de la Modularisation

La division du système en modules distincts présente plusieurs **avantages** majeurs :

- **Clarté et maintenabilité** : Chaque module est développé et testé indépendamment, ce qui facilite la compréhension globale du système et simplifie les modifications ou mises à jour.
- **Flexibilité** : Il devient trivial de substituer ou d'améliorer un module particulier sans affecter les autres parties du SCN. Par exemple, l'algorithme de calcul de synergie peut être remplacé par une version hybride (fusionnant sub-symbolique et symbolique) tout en conservant la même interface avec le module de mise à jour des ω .
- **Extensibilité** : Dans des scénarios multi-entités ou multi-domaines, le même cadre de mise à jour peut être réutilisé en adaptant uniquement le module de calcul de synergie ou les

paramètres de régulation, ce qui permet une intégration homogène de divers types de données et de logiques.

- **Contrôle et adaptation** : La structure modulaire permet l'implémentation de mécanismes de surveillance et de contrôle (inhibition, clipping, recuit) qui peuvent être activés ou désactivés indépendamment, offrant ainsi une gestion fine de la dynamique d'auto-organisation.

4.8.2. Gestions des Ressources et Threads

Dans un **SCN** (Synergistic Connection Network) à grande échelle, la boucle de mise à jour $\omega_{i,j}(t+1)$ peut rapidement devenir un **goulet d'étranglement** si elle s'exécute de façon **strictement séquentielle**. On peut souhaiter **paralléliser** certaines opérations ou recourir à des **threads** multiples, surtout si n (le nombre d'entités) se compte en milliers ou plus. Toutefois, cette parallélisation doit être orchestrée avec soin pour éviter les **incohérences** dans le calcul de $\Delta\omega_{i,j}$.

4.8.2.1. Mise à Jour Parallèle ou Asynchrone : Chaque Entité Calcule Localement $\Delta\omega_{i,j}$

Dans les systèmes de **Deep Synergy Learning (DSL)**, la mise à jour des pondérations $\omega_{i,j}(t)$ entre entités se réalise traditionnellement par le biais d'une double boucle itérative sur toutes les paires (i, j) . Cependant, lorsque le nombre d'entités n devient grand, le coût de calcul, de l'ordre de $\mathcal{O}(n^2)$ opérations par itération, peut rapidement devenir prohibitif. Pour pallier cette limitation, il est naturel d'envisager une **mise à jour parallèle ou asynchrone** des pondérations, dans laquelle chaque entité calcule localement sa contribution $\Delta\omega_{i,j}$ aux mises à jour, permettant ainsi de répartir la charge de calcul entre plusieurs threads ou processeurs. Nous présentons ici de manière détaillée les mécanismes, les avantages et les risques associés à cette approche.

A. Principe de la Mise à Jour Parallèle

L'idée fondamentale consiste à **diviser** le calcul de la matrice des pondérations $\boldsymbol{\omega}(t)$ en sous-tâches indépendantes, qui peuvent être exécutées simultanément sur différents coeurs ou processeurs. Soit n le nombre d'entités, la mise à jour standard s'exprime par :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

où $S(i,j)$ représente la synergie entre les entités i et j , η est le taux d'apprentissage et τ le coefficient de décroissance. Pour paralléliser ce calcul, plusieurs approches sont possibles. Par exemple, l'on peut découper la matrice $\boldsymbol{\omega}$ en **sous-blocs** ou en attribuant à chaque entité i la responsabilité de mettre à jour tous les liens sortants $\omega_{i,j}$ pour $j = 1, \dots, n$. Formellement, chaque entité réalise le calcul :

$$\Delta\omega_{i,j} = \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

puis met à jour localement

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \Delta\omega_{i,j}.$$

L'utilisation d'un **double-buffer** est souvent recommandée pour éviter les conflits d'accès en écriture. On définit ainsi une matrice $\mathbf{w}^{(t+1)}$ dans laquelle chaque thread écrit ses mises à jour, alors que la matrice $\mathbf{w}^{(t)}$ est lue en lecture seule pendant toute l'itération.

B. Synchronous vs. Asynchronous Update

Dans le **mode synchrone**, toutes les entités ou tous les threads réalisent leur calcul de $\Delta\omega_{i,j}$ en se basant sur la même version de la matrice $\omega(t)$. La procédure se décompose en deux phases distinctes :

- **Phase de lecture** : Chaque thread lit la matrice $\omega(t)$ pour calculer les variations locales $\Delta\omega_{i,j}$ pour les indices qui lui sont assignés.
- **Phase de mise à jour** : Une fois que toutes les variations ont été calculées, un **barrier** ou **synchronisation** est appliquée, puis la nouvelle matrice $\omega(t+1)$ est construite en copiant les résultats du buffer de sortie.

Mathématiquement, on définit deux matrices $\mathbf{w}^{(t)}$ et $\mathbf{w}^{(t+1)}$ telles que, pour tous i et j ,

$$\mathbf{w}_{i,j}^{(t+1)} = \mathbf{w}_{i,j}^{(t)} + \eta [S(i,j) - \tau \mathbf{w}_{i,j}^{(t)}].$$

Ce mode garantit une **cohérence** absolue puisque chaque entité travaille sur une version identique de $\omega(t)$. Le principal inconvénient réside dans la nécessité d'attendre que tous les threads aient terminé leur calcul, ce qui peut introduire un délai de synchronisation dans des environnements massivement parallèles.

En **mode asynchrone**, chaque entité met à jour ses propres liens dès que son calcul de $\Delta\omega_{i,j}$ est terminé, sans attendre la fin de l'itération pour l'ensemble du réseau. Cela peut se formaliser par l'équation :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

où la lecture et l'écriture de $\omega_{i,j}$ se font de manière **concurrente** et **lock-free**. Dans ce contexte, il arrive que différentes entités lisent des versions légèrement différentes de la matrice ω (certaines mises à jour étant déjà effectuées tandis que d'autres non). Ce mode peut accélérer le temps de calcul global, surtout dans des architectures distribuées, mais il soulève des problèmes de **cohérence** et de **stabilité**. Par exemple, une mise à jour asynchrone peut être analysée comme une itération de type Gauss-Seidel, où l'**ordre** d'exécution influe sur la convergence. Des techniques telles que le **verrouillage léger** ou l'emploi de structures atomiques peuvent être nécessaires pour minimiser les conflits, sans pour autant imposer une synchronisation complète.

C. Avantages et Risques du Calcul Parallèle/Asynchrone

L'**avantage principal** de la mise à jour parallèle ou asynchrone réside dans la **réduction du temps de calcul** global. En divisant la charge de mise à jour sur plusieurs threads, le coût de $\mathcal{O}(n^2)$ opérations peut être réduit de manière quasi linéaire avec le nombre de processeurs disponibles, permettant ainsi d'aborder des systèmes à grande échelle. Par ailleurs, un mode asynchrone, s'il

est correctement contrôlé, permet une **adaptation continue** et une mise à jour plus fluide de ω , même dans des environnements distribués.

Cependant, ces gains de performance ne vont pas sans risques. En mode asynchrone, l'absence d'une synchronisation stricte peut conduire à des **incohérences** où certaines mises à jour sont appliquées sur des données déjà modifiées, générant des écarts imprévus et potentiellement des oscillations dans les valeurs des pondérations. La garantie théorique de convergence devient alors plus complexe, nécessitant l'emploi d'algorithmes de contrôle tels que les techniques basées sur le **Gossip** ou des schémas de verrouillage léger pour assurer que la dynamique globale demeure stable.

D. Stratégies pour la Mise en Œuvre Asynchrone

Afin d'assurer la **stabilité** tout en bénéficiant des avantages du parallélisme, plusieurs stratégies peuvent être mises en œuvre :

17. Double Buffering :

Chaque itération lit une copie immuable de la matrice $\mathbf{w}^{(t)}$ pour effectuer les calculs et écrit les résultats dans une nouvelle matrice $\mathbf{w}^{(t+1)}$. Cette approche garantit que toutes les mises à jour de la phase t sont basées sur des données cohérentes, éliminant ainsi les conflits d'accès simultanés.

18. Utilisation de Verrous Atomiques ou Lock-Free :

Dans des environnements multi-threadés, l'emploi de verrous atomiques permet de s'assurer que les mises à jour sur un élément $\omega_{i,j}$ ne sont pas interrompues ou écrasées par d'autres threads. Bien que cette approche puisse introduire un certain coût en termes de performance, elle offre une **garantie** de cohérence. Des algorithmes lock-free sophistiqués, inspirés des techniques de la programmation concurrente, peuvent également être envisagés pour minimiser les surcharges tout en maintenant la stabilité.

19. Décomposition en Blocs :

En divisant la matrice \mathbf{w} en sous-blocs disjoints, chaque thread peut être affecté à un bloc spécifique, réduisant ainsi les risques de conflits. Cette approche est particulièrement efficace si l'on peut isoler des sous-ensembles d'entités ayant peu d'interaction avec d'autres sous-ensembles.

20. Ajustement Dynamique des Paramètres :

Les paramètres η et τ jouent un rôle crucial dans la vitesse de convergence et la stabilité. Un **Module de Gestion des Paramètres** peut surveiller en temps réel la variation des pondérations et ajuster η de manière à atténuer les oscillations lorsque des mises à jour asynchrones induisent des incohérences.

4.8.2.2. Conflits Possibles, Besoin de Cohérence Globale (Verrous, ou Algorithmes Lock-Free ?)

Dans un environnement de mise à jour parallèle ou asynchrone d'un **Synergistic Connection Network (SCN)**, la **cohérence** globale des pondérations $\omega_{i,j}$ est mise à rude épreuve par la concurrence des opérations de lecture et d'écriture. Dès lors que plusieurs threads ou processus interviennent simultanément pour calculer les incrémentations $\Delta\omega_{i,j}$ et mettre à jour les valeurs correspondantes, il apparaît impératif d'assurer une synchronisation adéquate afin de préserver l'intégrité de la dynamique globale. Dans cette section, nous analysons les conflits qui peuvent surgir, les approches traditionnelles basées sur des verrous (mutex) et les alternatives lock-free, et nous discutons de leur impact sur la stabilité et la convergence du SCN.

A. Nature des Conflits en Mise à Jour Concurente

Le SCN repose sur la règle de mise à jour additive donnée par

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)],$$

où $S(i,j)$ représente le score de synergie entre les entités i et j . Dans un environnement parallèle, la matrice ω est partagée entre plusieurs threads, et plusieurs d'entre eux peuvent tenter de lire ou d'écrire simultanément la même valeur $\omega_{i,j}(t)$.

Par exemple, si un thread lit $\omega_{i,j}(t)$ pour calculer $\Delta\omega_{i,j}$ pendant qu'un autre thread a déjà effectué une mise à jour sur la même case, alors la valeur utilisée pour le calcul peut être obsolète ou partiellement mise à jour. Ce phénomène de **races conditions** (conditions de concurrence) peut conduire à des incohérences, notamment lorsque la somme d'une inhibition globale dépend de la lecture simultanée de plusieurs $\omega_{i,k}(t)$.

En d'autres termes, des mises à jour concurrentes non coordonnées peuvent aboutir à une matrice qui ne reflète plus fidèlement la dynamique théorique du système, générant ainsi des déphasages, des oscillations imprévues ou même la divergence de la procédure de convergence.

B. Stratégies de Synchronisation Basées sur les Verrous

La méthode la plus intuitive pour prévenir ces conflits est d'utiliser des **verrous (mutex)**, lesquels garantissent que lorsqu'un thread est en train d'accéder ou de modifier une partie de la matrice ω , aucun autre thread n'y a accès. Deux approches se distinguent :

- **Verrous Globaux** : On peut appliquer un verrou global sur l'ensemble de la matrice ω pendant la phase de mise à jour. Autrement dit, dès qu'un thread commence à mettre à jour une valeur $\omega_{i,j}$, il acquiert un verrou qui empêche toute autre modification sur ω jusqu'à la fin de l'itération. Cette approche garantit une cohérence parfaite, mais elle réduit fortement le parallélisme, car les autres threads doivent attendre la libération du verrou global.
- **Verrous Fins** : Une approche plus raffinée consiste à appliquer des verrous sur des sous-ensembles de la matrice, par exemple sur chaque ligne ou sur des blocs spécifiques. Ainsi, si un thread met à jour la ligne i de la matrice, il acquiert un verrou sur cette ligne, tandis que d'autres threads peuvent mettre à jour des lignes différentes simultanément. Cette granularité fine permet une meilleure **concurrence** tout en assurant une cohérence locale,

même si la gestion de multiples verrous peut devenir complexe dans un système à très haute dimension.

La mise en œuvre des verrous s'appuie souvent sur des primitives fournies par les bibliothèques de concurrence (par exemple, le module *threading* en Python ou les classes *std::mutex* en C++). Bien que cette approche soit efficace pour assurer la cohérence, elle peut introduire un surcoût en termes de **latence** et de **contention**, en particulier lorsque le nombre d'entités est grand.

C. Approches Lock-Free et Algorithmes Gossip

Pour surmonter les limitations imposées par les verrous, des approches **lock-free** ont été développées. Ces méthodes reposent sur l'utilisation d'opérations atomiques, telles que le **compare-and-swap (CAS)**, qui permettent de mettre à jour les valeurs de $\omega_{i,j}$ sans verrou explicite. L'idée est de garantir que chaque opération de mise à jour soit effectuée de manière indivisible, de sorte qu'un thread vérifie que la valeur de $\omega_{i,j}$ n'a pas changé depuis sa lecture avant d'y écrire la nouvelle valeur. Formulée de manière schématique, une mise à jour lock-free s'effectue ainsi :

$$\text{if } \omega_{i,j} = \text{old_value} \text{ then } \omega_{i,j} \leftarrow \text{new_value} \text{ (atomiquement).}$$

Cette approche permet une **concurrence** très élevée, particulièrement dans des environnements distribués ou sur des architectures multicœurs. Toutefois, la garantie de **convergence** en mode lock-free nécessite des conditions strictes, notamment un paramétrage très faible de η et τ pour limiter les déphasages. Par ailleurs, la complexité de l'analyse mathématique augmente, car la dynamique asynchrone introduit des retards variables et des mises à jour non uniformes, qui peuvent, dans certains cas, conduire à des oscillations accrues.

Une autre méthode est l'utilisation d'algorithmes **Gossip-based**, dans lesquels chaque nœud partage ses mises à jour de manière probabiliste et locale avec ses voisins. Cette approche permet de propager l'information de manière distribuée, avec des garanties de convergence sous certaines conditions sur le réseau de communication. Cependant, les algorithmes de type Gossip introduisent également des retards et des incohérences temporaires qui doivent être absorbés par la dynamique globale du SCN.

D. Impact sur la Convergence et la Stabilité Globale

Le choix entre des verrous, des approches lock-free ou des algorithmes Gossip influence directement la **convergence** du SCN. En mode synchrone, la garantie de convergence est aisée à démontrer grâce à l'hypothèse que toutes les mises à jour se font sur une version immuable $\omega(t)$. Le double-buffering, qui consiste à utiliser une matrice $\omega^{(t)}$ en lecture seule et une autre $\omega^{(t+1)}$ pour les écritures, est souvent utilisé pour préserver la **cohérence** tout en permettant une parallélisation. En mode asynchrone, la lecture de valeurs légèrement décalées peut induire des fluctuations dans les mises à jour. Pour assurer la stabilité, il est alors nécessaire d'ajuster le taux d'apprentissage η pour qu'il soit suffisamment faible, garantissant que même en présence de retards, l'opération de mise à jour reste contractante dans la norme choisie (i.e., $|1 - \eta \tau| < 1$).

L'utilisation de techniques de contrôle, telles que l'injection de bruit contrôlé (similaire à un recuit stochastique) ou l'application périodique de mécanismes de réindexation, peut aider à compenser les déphasages induits par une mise à jour asynchrone. Ainsi, l'architecture lock-free, bien que plus

rapide en termes de débit de mise à jour, requiert une analyse plus fine des conditions de convergence, et peut parfois se traduire par des comportements oscillatoires ou des divergences temporaires qu'il faudra ensuite amortir par des correctifs algorithmiques.

4.8.3. Transition

Les sections précédentes (4.8.1, 4.8.2) ont souligné l'importance de l'**architecture logicielle** (modularisation, gestion des ressources, parallélisme) pour soutenir la dynamique d'un **SCN** (Synergistic Connection Network). Il apparaît clairement qu'au-delà de la simple mise à jour $\omega_{i,j}$, un cadre organisationnel robuste est nécessaire. Il s'agit de définir comment instancier les modules de synergie, comment gérer l'inhibition et comment orchestrer la boucle de calcul en mode synchrone ou asynchrone.

4.8.3.1. Ce qu'on Pourrait Coder en Python/C++ pour Reproduire cette Dynamique en Quelques Centaines de Lignes

Dans cette section, nous présentons une approche concrète pour implémenter la dynamique d'un **Synergistic Connection Network (SCN)** dans un langage de programmation tel que Python ou C++. L'objectif est de traduire la théorie développée dans les chapitres précédents en un code minimaliste mais complet, permettant de simuler la mise à jour des pondérations $\omega_{i,j}(t)$ et d'observer l'émergence de structures telles que des clusters, en quelques centaines de lignes. Cette démarche met en œuvre la règle de mise à jour additive ainsi que des mécanismes complémentaires comme l'inhibition et le clipping, tout en assurant une modularisation pour une meilleure maintenance.

A. Définition des Paramètres et Initialisation

Le système comporte n entités, chacune identifiée par un indice $i \in \{1, 2, \dots, n\}$. La dynamique du SCN est régie par la règle de mise à jour :

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t),$$

où :

- η représente le **taux d'apprentissage**,
- τ est le **facteur de décroissance** qui module la pénalisation d'un lien trop élevé,
- γ est le **coefficent d'inhibition** destiné à limiter la prolifération des connexions non sélectives,
- $S(i, j)$ désigne le score de **synergie** entre les entités i et j .

Le premier module du code consiste à définir ces paramètres et à initialiser la matrice de pondérations $\omega(0)$. Par convention, on initialise $\omega_{i,j}(0)$ à des valeurs faibles (par exemple, un bruit aléatoire autour de zéro) pour simuler l'état initial d'un système non organisé.

En Python, on utilisera la bibliothèque NumPy pour manipuler efficacement les matrices. Par exemple :

```
import numpy as np

# Nombre d'entités
n = 15

# Paramètres globaux
eta = 0.05    # Taux d'apprentissage
tau = 1.0     # Facteur de décroissance
gamma = 0.02   # Coefficient d'inhibition
T_max = 300   # Nombre d'itérations

# Initialisation de la matrice des pondérations, avec un léger bruit aléatoire
np.random.seed(42)
omega = np.random.uniform(0.0, 0.05, (n, n))
np.fill_diagonal(omega, 0.0)

# Initialisation de la matrice de synergie S (à définir selon le contexte)
S = np.zeros((n, n))
# Par exemple, pour simplifier, on peut définir S(i,j) statique selon des critères prédéfinis
# Ici, S est symétrique et les valeurs sont entre 0 et 1.
for i in range(n):
    for j in range(n):
        if i != j:
            S[i, j] = np.random.uniform(0.1, 0.8)
```

B. Implémentation de la Boucle de Mise à Jour

La dynamique d'auto-organisation est réalisée via une boucle itérative. À chaque itération t , la mise à jour des pondérations se fait de manière synchrone sur l'ensemble des paires (i, j) . Afin d'éviter les conflits de lecture-écriture, on utilise le **double-buffering**. La matrice $\omega(t)$ est lue en lecture seule tandis que les nouvelles valeurs sont écrites dans un buffer temporaire ω_{next} .

La mise à jour additive est alors formulée comme suit :

$$\omega_{i,j}(t + 1) = \omega_{i,j}(t) + \eta [S(i, j) - \tau \omega_{i,j}(t)] - \gamma \sum_{k \neq j} \omega_{i,k}(t).$$

Le pseudo-code Python ci-dessous illustre ce processus :

```
# Liste pour enregistrer l'évolution de la moyenne des pondérations (pour monitoring)
omega_means = []

# Boucle principale de mise à jour
for t in range(T_max):
```

```

omega_next = np.copy(omega)
for i in range(n):
    for j in range(n):
        if i != j:
            # Calcul de la variation (delta) de omega[i,j]
            delta = eta * (S[i, j] - tau * omega[i, j])
            # Application de l'inhibition : somme des pondérations de la ligne i sauf pour j
            inhibition = gamma * (np.sum(omega[i, :]) - omega[i, j])
            # Mise à jour additive
            omega_next[i, j] = omega[i, j] + delta - inhibition
            # Clipping pour garantir que omega reste positif
            omega_next[i, j] = max(0.0, omega_next[i, j])
    omega = omega_next
    # Enregistrement d'une statistique pour monitoring
    omega_means.append(np.mean(omega))

```

À la fin de la boucle, omega contient la configuration finale du SCN.

Ce code, bien que succinct, encapsule le cœur de la **dynamique auto-organisée** du SCN. On observe que la mise à jour est effectuée de manière synchrone via un double-buffering, assurant la cohérence de la lecture de $\omega(t)$ pendant toute l'itération.

C. Visualisation et Analyse des Résultats

Pour évaluer la **convergence** et la **formation des clusters**, il est utile de visualiser l'évolution de la matrice ω et des indicateurs tels que la moyenne ou la variance des pondérations. En Python, on peut utiliser des bibliothèques telles que Matplotlib pour tracer des graphiques en temps réel ou après simulation :

```

import matplotlib.pyplot as plt

# Tracé de la moyenne des pondérations au cours des itérations
plt.figure(figsize=(8, 5))
plt.plot(omega_means, label='Moyenne de $\omega(t)$')
plt.xlabel('Itérations (t)')
plt.ylabel('Valeur moyenne de $\omega$')
plt.title('Convergence de la matrice de pondérations')
plt.legend()
plt.grid(True)
plt.show()

```

De plus, pour une analyse plus fine, on peut générer une **heatmap** de la matrice ω à la fin de la simulation pour identifier visuellement les clusters :

```

import seaborn as sns

plt.figure(figsize=(10, 8))
sns.heatmap(omega, cmap='viridis', annot=True, fmt=".2f")
plt.title('Heatmap finale de la matrice $\omega$')
plt.xlabel('Entités')

```

```

plt.ylabel('Entités')
plt.show()

```

Ces visualisations permettent de confirmer que les pondérations se sont organisées en clusters stables, conformément aux attentes théoriques.

D. Extensions Possibles et Adaptations en C++

La même logique de mise à jour peut être implémentée en C++ en utilisant, par exemple, la bibliothèque STL et des structures de données adaptées (comme `std::vector<std::vector<double>>` pour représenter ω). La boucle de mise à jour sera similaire à celle présentée en Python, avec des boucles *for* imbriquées pour parcourir les indices i et j . Voici un exemple schématique en C++ :

```

#include <vector>
#include <iostream>
#include <algorithm>
#include <cstdlib>

// Définition des paramètres
const int n = 15;
const int T_max = 300;
const double eta = 0.05;
const double tau = 1.0;
const double gamma = 0.02;

// Fonction pour initialiser la matrice avec un bruit faible
std::vector<std::vector<double>> initializeMatrix(int n) {
    std::vector<std::vector<double>> omega(n, std::vector<double>(n, 0.0));
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            if (i != j) {
                omega[i][j] = ((double) rand() / RAND_MAX) * 0.05;
            }
        }
    }
    return omega;
}

// La fonction principale de mise à jour
void updateOmega(std::vector<std::vector<double>> &omega, const std::vector<std::vector<double>> &S) {
    int n = omega.size();
    std::vector<std::vector<double>> omega_next = omega;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            if (i != j) {
                double sum_inhibition = 0.0;
                for (int k = 0; k < n; ++k) {
                    if (k != j) {
                        sum_inhibition += omega[i][k];
                    }
                }
                omega_next[i][j] = S[i][j] * (1 - tau * sum_inhibition) + gamma * omega[i][j];
            }
        }
    }
    omega = omega_next;
}

```

```

        }
        double delta = eta * (S[i][j] - tau * omega[i][j]) - gamma * sum_inhibition;
        omega_next[i][j] = std::max(0.0, omega[i][j] + delta);
    }
}
omega = omega_next;
}

int main() {
    // Initialisation des matrices omega et S
    std::vector<std::vector<double>> omega = initializeMatrix(n);
    std::vector<std::vector<double>> S(n, std::vector<double>(n, 0.0));
    // Remplissage simple de S, par exemple avec des valeurs aléatoires entre 0.1 et 0.8
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            if (i != j) {
                S[i][j] = 0.1 + ((double) rand() / RAND_MAX) * 0.7;
            }
        }
    }

    // Boucle de mise à jour
    for (int t = 0; t < T_max; ++t) {
        updateOmega(omega, S);
    }

    // Affichage final de la matrice omega
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            std::cout << omega[i][j] << " ";
        }
        std::cout << std::endl;
    }

    return 0;
}

```

Ce code C++ illustre la structure de base du pipeline de mise à jour, de la même manière que la version Python, et démontre que la **logique** sous-jacente est indépendante du langage de programmation. La modularité du code permet ensuite d'y intégrer facilement des mécanismes de **visualisation** ou d'**exportation** des résultats pour une analyse plus poussée.

4.8.3.2. Les Concepts de ce Chapitre 4 (Mise à Jour, Inhibition, Clusters Émergents) s'Inséreront dans une Architecture Logicielle Solide

L'ensemble des notions développées dans le chapitre 4 – notamment la dynamique de mise à jour des pondérations $\omega_{i,j}(t+1)$, les mécanismes d'inhibition (locaux ou globaux) et l'émergence des clusters – doit être opérationnalisé dans une infrastructure logicielle modulaire pour permettre un

déploiement à grande échelle et une maintenance aisée. Le passage de la théorie à la pratique exige la conception d'un framework où chaque composant de la dynamique est isolé sous forme de modules interchangeables. Cette approche facilite non seulement l'évolution et l'extension du code, mais également l'intégration de nouvelles méthodes de calcul ou de nouveaux algorithmes de contrôle. Nous détaillons ci-après les axes majeurs de cette intégration.

A. Intégration Modulaire des Composants de la Dynamique

Dans un premier temps, il convient de séparer distinctement les responsabilités fonctionnelles de la dynamique du SCN. On distingue principalement trois modules :

- **Module de Calcul de Synergie** : Ce module est chargé de déterminer la fonction $S(i, j)$, qui représente la qualité ou l'intensité de la synergie entre deux entités \mathcal{E}_i et \mathcal{E}_j . La fonction $S(i, j)$ peut être définie à partir de critères sub-symboliques (par exemple, la similarité cosinus ou une distance gaussienne entre embeddings) ou symboliques (compatibilité de règles ou axiomes). On peut formuler, pour une approche hybride, la synergie comme

$$S_{\text{hybrid}}(i, j) = \alpha S_{\text{sub}}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) S_{\text{sym}}(R_i, R_j),$$

où $\alpha \in [0, 1]$ module l'importance relative des composantes. Ce module est conçu pour être interfaçable avec différents types de données et peut être mis à jour indépendamment, par exemple en recalculant périodiquement les embeddings ou en ajustant les règles symboliques.

- **Module de Mise à Jour des Pondérations ω** : Ce module encapsule la règle d'auto-organisation appliquée à la matrice $\boldsymbol{\omega}$. Qu'il s'agisse d'un schéma additif classique

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i, j) - \tau \omega_{i,j}(t)],$$

ou d'une version multiplicative (par exemple,

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) \exp(\eta [S(i, j) - \tau \omega_{i,j}(t)]),$$

ce module reçoit en entrée la matrice S issue du module précédent, ainsi que les paramètres η et τ , et produit la nouvelle matrice $\boldsymbol{\omega}(t+1)$. L'architecture modulaire permet de remplacer aisément la règle de mise à jour en modifiant uniquement le code de ce module, sans impacter le calcul de la synergie.

- **Module de Régulation (Inhibition, Sparsification, Clipping)** : Pour éviter une croissance non contrôlée des pondérations et pour favoriser la formation de clusters cohérents, il est souvent nécessaire d'introduire des mécanismes d'inhibition et de clipping. Par exemple, l'ajout d'un terme inhibiteur tel que

$$-\gamma \sum_{k \neq j} \omega_{i,k}(t)$$

dans la mise à jour permet de pénaliser la distribution excessive de liens par une entité. Ce module peut être implémenté soit comme une étape intégrée dans le module de mise à jour, soit comme une étape postérieure appliquant un filtrage sur la matrice $\boldsymbol{\omega}$. Le recours à une méthode de clipping, où l'on borne $\omega_{i,j}(t)$ à une valeur maximale ω_{\max} , peut également être intégré ici pour éviter des valeurs aberrantes. Cette modularisation garantit qu'en cas

de besoin, le comportement de régulation pourra être ajusté sans modifier les autres composants du système.

Ces modules interagissent au sein d'un **pipeline** orchestré par un **engine** central, dont le rôle est de gérer la séquence d'exécution, la synchronisation (via par exemple un double-buffering pour garantir la cohérence) et la persistance des données entre les itérations. Le code principal de ce pipeline appelle successivement le module de calcul de synergie, puis le module de mise à jour des pondérations, et enfin le module de régulation, avant de stocker le résultat ou de déclencher une phase de visualisation.

B. Gestion Dynamique et Paramétrage

Une des forces de l'architecture modulaire est la capacité à gérer dynamiquement les paramètres η , τ et γ . Un **Param Manager** est ainsi mis en place, centralisant la configuration de ces valeurs. Ce gestionnaire peut, par exemple, surveiller la convergence en calculant à chaque itération des indicateurs tels que la norme $\|\omega(t+1) - \omega(t)\|$ ou la variation de l'énergie potentielle $J(\omega)$. Si une oscillation excessive est détectée, le Param Manager peut alors automatiquement réduire η (ou augmenter τ) pour stabiliser la dynamique. La flexibilité apportée par ce module permet non seulement d'ajuster les paramètres de manière adaptative en fonction des données entrantes, mais également de tester divers schémas de mise à jour sans réécriture globale du code.

C. Persistance et Interface d'Intégration

La persistance des données dans un SCN est cruciale, surtout lorsqu'on déploie le système sur des volumes de données conséquents ou sur des plateformes distribuées. Une fois que la boucle de mise à jour est exécutée, la matrice ω doit être sauvegardée pour permettre une reprise ultérieure ou une analyse post-hoc. Ce processus s'effectue via des modules de sauvegarde qui peuvent écrire la matrice dans un format binaire optimisé (par exemple, en utilisant le format HDF5) ou en format sparse si la majorité des liens tend à être nulle. Par ailleurs, une API d'intégration permet de brancher le SCN au sein d'un framework plus large – par exemple, pour interfacer avec des simulateurs robotiques ou des systèmes de monitoring en temps réel. Ce type d'interface se traduit par des endpoints REST ou par l'injection de flux de données dans le SCN, facilitant ainsi son déploiement dans des environnements industriels.

D. Orchestration de la Dynamique par un Engine Central

Le cœur de l'implémentation repose sur une boucle maîtresse ou un **scheduler** qui orchestre les appels successifs aux modules. Ce scheduler, implémenté par exemple en Python via des bibliothèques de multi-threading (comme *multiprocessing* ou *concurrent.futures*), effectue les opérations suivantes à chaque itération t :

- Il appelle le module de **calcul de synergie** pour obtenir la matrice $S(t)$ ou pour la mettre à jour en fonction des nouvelles données.
- Il déclenche le module de **mise à jour de ω** en passant les paramètres courants et la matrice $S(t)$, puis recueille la nouvelle matrice $\omega(t+1)$ via un système de double-buffering afin d'assurer une lecture cohérente.

- Il active ensuite le module de **régulation** (inhibition et clipping) pour appliquer des ajustements supplémentaires sur la matrice $\omega(t + 1)$.
- Enfin, il enregistre l'état actuel de $\omega(t + 1)$ pour permettre des analyses ultérieures (par exemple, via des heatmaps ou des calculs d'indicateurs de modularité).

Cet ensemble d'opérations, en étant modulé et orchestré par un engine central, garantit que la **dynamique globale** du SCN reste stable, tout en permettant une adaptation rapide aux changements de paramètres ou à l'insertion de nouveaux modules.

4.9. Conclusion et Transition

Tout au long de ce **chapitre 4**, nous avons examiné en détail comment un **SCN** (Synergistic Connection Network) s'**auto-organise** et fait émerger des **clusters** cohérents via la mise à jour adaptative des pondérations $\omega_{i,j}$. Les divers mécanismes (inhibition, saturation, injection de bruit, parallélisation...) ont montré la **richesse** et la **flexibilité** de cette dynamique. Il est maintenant temps de faire la **synthèse** (4.9.1) et de préparer le passage vers les **chapitres** suivants (4.9.2 et 4.9.3) où l'on abordera l'architecture globale (Chap. 5) et des développements plus avancés (Chaps. 6, 7, 8).

4.9.1. Synthèse

Dans l'ensemble des développements qui précèdent, le **cœur** de la dynamique d'un **Synergistic Connection Network** (SCN) repose sur la mise à jour locale des pondérations $\omega_{i,j}(t)$ selon des règles inspirées du schéma $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$. Une telle relation, parfois agrémentée d'un terme d'**inhibition** ou d'un **recuit**, définit un processus d'**auto-organisation** qui tend à renforcer les liens $\omega_{i,j}$ pour lesquels la **synergie** $S(i,j)$ est jugée élevée, tout en affaiblissant ceux qui se révèlent peu utiles ou inconsistants. Il est apparu, au fil des sections, que cette mécanique engendre la **formation** de **clusters** cohésifs, c'est-à-dire des ensembles d'entités fortement connectées par des pondérations $\omega_{i,j}$ élevées, alors que les connexions inter-groupes demeurent faibles ou presque nulles.

La règle additive simple, $\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta[S(i,j) - \tau \omega_{i,j}(t)]$, se situe au fondement du processus. Dans un régime stationnaire, elle conduit souvent à un équilibre $\omega_{i,j}^* \approx S(i,j)/\tau$ pour chaque paire (i,j) , à condition que les entités n'évoluent pas et que les paramètres restent constants. Les **variantes** multiplicatives ou inhibitrices enrichissent ce tableau, en introduisant soit des effets de proportionnalité (cas multiplicatif) soit des phénomènes de compétition plus marqués (cas d'une inhibition latérale ou globale). Il a également été mis en évidence que la présence de **recuit** ou l'injection de bruit pouvaient aider à franchir des barrières ou à sortir de minima locaux, favorisant un **cluster** plus pertinent à l'échelle globale.

Le **contrôle** des oscillations se révèle un aspect capital pour la stabilité. Un choix cohérent de η (taux d'apprentissage), τ (décroissance) et γ (inhibition) circonscrit la zone d'attraction évitant les emballements, tout en permettant une différenciation nette des liaisons fortes et faibles. Il subsiste néanmoins une possibilité de **multi-stabilité** lorsque plusieurs configurations de clusters satisfont la matrice de synergie \mathbf{S} . Le système peut alors basculer vers l'un ou l'autre attracteur suivant la condition initiale ou les perturbations reçues au fil des itérations.

À l'issue de cette analyse, il apparaît clairement que la dynamique d'un SCN ne saurait se réduire à une suite d'équations isolées. Dans un contexte pratique ou à grande échelle, elle doit s'inscrire dans une **architecture logicielle** modulaire, où l'on distingue le **module** affecté au **calcul** de la synergie, le **module** chargé de la **mise à jour** ω , le **module** gérant l'**inhibition** (ou la sparsification), et enfin les composantes d'**observation** ou de **clustering** (détection de communautés, visualisation). Cette séparation en blocs distincts, qui sera explicitée plus avant dans

le chapitre 5, facilite la **maintenance**, l'**évolution** et la **persistence** des données. Elle permet également d'orchestrer de manière ordonnée la boucle itérative en intégrant des mécanismes comme la mise à jour parallèle, le double-buffering et la gestion des logs.

En conclusion, la **mise à jour** $\omega_{i,j}(t + 1)$, complétée par les mécanismes d'**inhibition**, de **recuit** ou de **sparsification**, constitue le **noyau** d'un **SCN** permettant d'aboutir à des **clusters** stables ou à un régime oscillatoire si les paramètres le favorisent. La prochaine étape, telle que le décrit le **chapitre 5**, est de **concrétiser** cette dynamique dans un cadre logiciel structuré, garantissant la **cohérence** des opérations, la **scalabilité** face à un grand nombre d'entités et l'**interopérabilité** avec d'autres modules ou frameworks. Cette articulation entre la **théorie** (chapitre 4) et l'**architecture** (chapitre 5) rend possible l'**implémentation** efficace d'un SCN, depuis les principes de base jusqu'au déploiement à grande échelle.

4.9.2. Ouverture

Les développements du présent chapitre 4 ont présenté la **dynamique** d'un **Synergistic Connection Network (SCN)**, en détaillant les principales règles de **mise à jour** $\omega_{i,j}(t + 1)$ (qu'elles soient additives ou multiplicatives), l'**inhibition** sous diverses formes, les mécanismes de **recuit** ou de **sparsification**, ainsi que l'**émergence** de **clusters** et les questions de **parallélisation**. Ces éléments constituent les fondations d'un système d'**auto-organisation** robuste. Néanmoins, ils ne sont que la première étape vers un **DSL (Deep Synergy Learning)** pleinement intégré. Les chapitres ultérieurs, à commencer par le **Chapitre 5**, franchiront un palier supplémentaire en décrivant la manière dont cette logique s'inscrit dans une **architecture** modulaire et scalable, apte à gérer de gros volumes de données, à s'interfacer avec d'autres modules, et à évoluer vers des scénarios avancés.

A. Chapitre 5 : Architecture Logicielle d'un SCN

Le chapitre suivant s'attachera à montrer **comment** transformer la simple “boucle de mise à jour” $\omega_{i,j}(t + 1)$ et ses règles (inhibition, recuit, etc.) en un **environnement logiciel** :

- **Organisation en modules.** Le cœur du système (mise à jour de ω , calcul de synergie **S**, stratégies d'inhibition ou de sparsification) sera découpé en **briques** clairement définies, chacune remplissant un rôle précis.
- **Persistiance de la matrice ω .** Il s'agira notamment de gérer le stockage (sur disque, en RAM distribuée, ou sous format sparse) pour des matrices éventuellement très volumineuses, ainsi que de prévoir la reprise du SCN là où il s'était arrêté (enregistrant $\omega(t)$, les paramètres, etc.).
- **API et intégration.** On décrira les interfaces (fonctions, classes, endpoints) permettant de brancher un SCN dans un framework existant (multi-agent, robotique, deep learning, etc.), de recevoir les flux de données exogènes (évolution de la synergie **S**), ou d'exporter les clusters émergents vers d'autres composants.

Cette approche **mature** du SCN assure la **maintenance** du code, la possibilité de tester facilement de nouveaux modules (par exemple un module d'inhibition différent), et une **adaptation** aisée à des cas pratiques (robotique, streaming de données, etc.).

B. Chapitres 6, 7, et 8 : Extensions et Scénarios Avancés

La base théorique et algorithmique mise en place dans le chapitre 4 (dynamique, clusters, multi-stabilité) et concrétisée architecturalement dans le chapitre 5 trouvera des prolongements au sein des chapitres 6, 7 et 8 :

- **Chapitre 6** : on traitera de l'**apprentissage multi-échelle**, ou de la structure **hiérarchique** d'un SCN, où l'on combine différents niveaux de granularité (micro-clusters, macro-clusters) et où la **fractalité** ou l'organisation emboîtée jouent un rôle.
- **Chapitre 7** : on explorera les **algorithmes d'optimisation** plus complexes (au-delà de la simple règle additive) et les techniques d'**adaptation dynamique**, qu'il s'agisse d'ajuster automatiquement η ou τ au fil du temps, ou de concevoir des approches globales (recuit multi-nœuds, stratégies d'exploration).
- **Chapitre 8** : on se tournera vers la **fusion multimodale**, c'est-à-dire des SCN manipulant des entités décrites par plusieurs modalités (texte, image, son, capteurs divers). Cette extension mettra en évidence la **flexibilité** du DSL pour unifier différents canaux de synergie au sein d'une même architecture.

4.9.3. Place du Chapitre 4

Ce chapitre a présenté la **dynamique** propre à un **Synergistic Connection Network** (SCN), en montrant de quelle manière la **synergie** $S(i,j)$, introduite au chapitre 2 et liée aux représentations d'entités décrites au chapitre 3, conduit à la **mise à jour** adaptative de $\omega_{i,j}(t)$. Ce quatrième volet joue ainsi un **rôle de charnière** entre les **fondements** conceptuels (chapitres 2 et 3) et les **perspectives** d'implémentation et d'extension (chapitres 5 à 8).

A. Passage des Concepts à la Dynamique Opérationnelle

Les deux premiers chapitres ont posé les **fondations** d'un SCN en introduisant les notions essentielles d'entités, de synergie $S(i,j)$ et de co-occurrences ou similarités logiques et sub-symboliques. Sans la **dynamique** décrite ici, ces idées se limiteraient à une théorie ou une cartographie d'affinités statiques. Le chapitre 4 a explicité comment, itération après itération, les pondérations $\omega_{i,j}$ s'**auto-adaptent** en fonction de $\omega_{i,j}(t)$ et de $S(i,j)$. Les notions de **clusters** émergents, de **multi-stabilité**, de **rôle** des paramètres η, τ, γ et de **convergence** (ou oscillations) illustrent la mise en mouvement concrète des entités, faisant passer le SCN d'une théorie de la synergie à un **réseau vivant**.

Cette transition se manifeste par la **formule** de mise à jour de base

$$\omega_{i,j}(t+1) = \omega_{i,j}(t) + \eta [S(i,j) - \tau \omega_{i,j}(t)],$$

éventuellement agrémentée d'un terme d'**inhibition** compétitive, d'un **recuit**, ou d'autres variantes non linéaires. Ces mécanismes répondent directement à la volonté de concrétiser la synergie en des liaisons qui se renforcent ou s'affaiblissent au fil du temps, selon leur utilité mutuelle.

B. Mise en Relation avec le Chapitre 5

Le chapitre 5 abordera la **conception** d'une **architecture** logicielle modulaire, destinée à rendre cette dynamique exploitable à grande échelle ou dans des systèmes complexes. Les principes détaillés ici — tels que l'inhibition locale ou globale, les règles additives ou multiplicatives, les stratégies de **sparcification**, la détection de clusters (section 4.6), la parallélisation ou l'asynchronisme (section 4.8) — s'intègrent dans un “**engine**” de mise à jour, que le chapitre 5 décrira selon des modules de calcul de **S**, de mise à jour ω , de persistance ou de connexion avec des frameworks externes.

L'**insertion** de ces éléments dans une architecture modulaire permet de **séparer** clairement la fonction “calculer la synergie” (chap. 3) de la fonction “faire évoluer la matrice ω ” (chap. 4). Le chapitre 5 montrera que cette discipline, impliquant parfois un double-buffer ou un mécanisme synchrone, évite bien des conflits ou pertes de cohérence. C'est grâce à de telles dispositions qu'il devient possible de **maintenir** ou de **faire évoluer** un SCN au fil du temps, éventuellement dans un contexte distribué ou multimodal.

C. Reliance aux Chapitres Avancés (6, 7, 8)

Les phénomènes d'**auto-organisation** décrits en ce chapitre sont aussi la **base** sur laquelle s'appuient les travaux ultérieurs :

- Le **multi-niveau** ou **multi-échelle** (chapitre 6), où l'on construit un SCN à plusieurs paliers hiérarchiques ;
- Les **algorithmes** d'optimisation et d'adaptation (chapitre 7), poussant plus loin la gestion de la multi-stabilité ou l'autoconfiguration de η, τ ;
- La **fusion multimodale** (chapitre 8), montrant que la mise à jour $\omega_{i,j}$ peut agréger simultanément plusieurs types de synergie (similitudes de vecteurs visuels, textes, sons), prolongeant la simple règle additive de base vers des combinaisons plus riches.

Ces développements avancés réutilisent la **dynamique** présentée dans le chapitre 4, parfois en l'agrémentant de mécanismes supplémentaires ou de variantes de la formule de mise à jour.

Conclusion

Le **chapitre 4** assure la **charnière** essentielle entre les **concept**s (chapitres 2 et 3) et la **réalisation** logicielle (chapitre 5) ainsi que les **extensions** (chapitres 6 à 8). Il a permis de voir comment la **synergie** se transforme, via des équations locales de mise à jour, en un **processus** d'auto-organisation conduisant à la **naissance de clusters**. Les règles d'**inhibition**, de **recuit** ou de **sparcification** assurent un contrôle précis du comportement de la dynamique, permettant d'éviter les effets négatifs (divergence, oscillations incontrôlées) et de stabiliser le système. Les résultats en attestent où de petits **clusters** émergent dans des cas simples et des communautés plus complexes apparaissent dans des situations multimodales ou multi-agent.

Le chapitre suivant explicitera de manière détaillée la **structure modulaire** d'un SCN, ouvrant la voie à la mise en œuvre d'un environnement professionnel, évolutif et apte à intégrer des flux de données massifs ou permanents. Cette transition du **concret** algorithmique (chapitre 4) à l'**ingénierie** (chapitre 5) parachève la logique du **Deep Synergy Learning**, associant auto-organisation théorique et déploiement logiciel robuste.