

Introduction

Les systèmes biologiques qui nous entourent présentent une incroyable diversité de structures leur permettant d'évoluer et de s'adapter à leur environnement par l'échange, le stockage et le traitement d'information venant de ce dernier. Autrement formulé, ces systèmes biologiques présentent des capacités de calcul remarquables. Ces mécanismes ont inspiré de nombreux algorithmes en informatique, imitant par exemple les colonies de fourmis, les essaims d'abeilles, les groupes de chats, les bancs de poissons ou encore les baleines, tous ces groupes d'animaux présentant des méthodes de communication décentralisées efficaces pour accomplir une tâche donnée (Darwish 2018). Un paragon de système biologique de calcul est sans conteste le cerveau humain ou animal, qui est capable d'exécuter des tâches de calcul et d'apprentissage incroyablement sophistiquées via une multitude de signaux électriques et chimiques circulant entre les neurones et au sein des vaisseaux sanguins. L'inspiration biologique occupe ainsi une place de premier rang dans les débuts de la recherche en intelligence artificielle. Dans ce sens, les premiers modèles d'apprentissage automatique ont été développés en s'appuyant sur des modèles biologique de neurones, afin de chercher à imiter les capacités d'évolution et d'adaptation à l'origine de la notion d'apprentissage dans les réseaux de neurones biologiques. Le perceptron, modèle à l'origine des réseaux de Deep Learning les plus performants à l'heure actuelle, s'inspirait par exemple d'un modèle de neurone biologique très simplifié (McCulloch et Pitts 1990). La diversité des applications actuelles de l'apprentissage automatique et le développement de nombreux modèles ont amené la recherche en apprentissage automatique à se concentrer principalement sur une approche algorithmique et computationnelle de ces réseaux, en particulier pour les architectures de Deep Learning. Cet éloignement de la biologie permet de s'extraire des contraintes physiques liées au neurone pour envisager des règles de calcul adaptées à des tâches spécifiques. Néanmoins, la biologie, par sa diversité de comportements encore incompris, reste une source d'inspiration abondante pour apporter des paradigmes alternatifs ou complémentaires aux modèles d'apprentissage existants dans l'état de l'art. De plus, le comportement du cerveau est loin d'être entièrement compris et modélisé, ce qui signifie que les possibilités d'inspiration biologique sont constamment en train d'évoluer. Les réseaux de neurones impulsionnels (*Spiking Neural Networks*) sont un exemple de modèle d'apprentissage illustrant une complémentarité récente entre l'approche bio-inspirée et l'approche computationnelle. Ces réseaux de neurones

ont été développés dès les années 1990 [Maass 1996](#) et s'appuient directement sur un modèle biologique du neurone ; cependant, ils apparaissent dans de nombreux travaux récents comme une méthode montante dans le domaine de l'apprentissage automatique pour la conception systèmes d'apprentissage moins énergivores et distribués grâce à la conception d'architectures matérielles neuromorphiques telles que LOIHI ¹. L'inspiration biologique apporte ainsi de nouveaux paradigmes de calcul pouvant se combiner avec des approches plus appliquées. Nous pensons pour cette raison qu'il est pertinent de continuer à explorer des modèles d'apprentissage automatique inspirés de la biologie.

Parmi les propriétés inspirées de la biologie, nous nous intéressons à la modularité. Cette propriété de modularité est partagée par de nombreux systèmes biologiques et artificiels et présente des avantages en termes de réutilisation, de robustesse aux fautes, de redondance et de traitement local de l'information (Clune et al. [2013](#)). De nombreuses modélisations générales du cortex cérébral, telles que [Binzegger et al. 2005](#) ; [Meunier et al. 2009](#) ; [Sporns 2013](#) ; [Betz et Bassett 2017](#) proposent ainsi que le cortex est une architecture composée de modules auto-organisés. Ces modules communiquent autour des informations sensorielles collectées par l'organisme. Cette communication est réalisée de façon interne, liant des informations sensorielles et abstraites provenant de différentes parties du cortex et à différentes échelles spatiotemporelles. Enfin, bien qu'une hiérarchie de traitement de l'information apparaisse entre ces modules, certains traitant des entrées sensorielles et d'autres des entrées plus abstraites, de nombreux circuits de rétroactions entre les modules semblent présents à différents niveaux de l'architecture. Suivant ce constat, la recherche d'architectures cognitives s'inspirant de l'architecture du cortex cérébral est un enjeu de longue date dans la recherche en apprentissage automatique. Il s'agit de développer des réseaux de neurones autonomes, capables de mémoire et de prise de décision de façon non supervisée, qui s'inspirent des architectures modulaires présentes dans le cerveau humain et qui cherchent à imiter certains comportements ([Kotseruba et Tsotsos 2018](#)). Le développement de la robotique appelle également à envisager de telles architectures cognitives, qui sont directement liées à la perception sensorielle.

Outre l'inspiration biologique liée à la structure corticale, la construction de systèmes d'apprentissage modulaires découle également d'une motivation computationnelle. Attardons-nous sur le concept de modularité, qui peut prendre des significations très vastes, de l'informatique à la biologie. Dans sa définition la plus générale, un système modulaire est un système composé de sous-systèmes, les modules, dont chacun peut être ajouté ou supprimé sans impacter l'architecture des autres modules. Cette vision générale de la modularité est l'approche généralement privilégiée en sciences et ingénierie, lorsque l'on décompose un problème à résoudre en un ensemble sous-problèmes, puis que l'on construit des modules visant à résoudre chacun de ces sous-problèmes. Nous nous plaçons dans une approche plus spécifique de cet aspect mo-

1. <https://www.intel.com/content/www/us/en/research/neuromorphic-computing.html>

dulaire en s'intéressant uniquement à des architectures dont les modules communiquent entre eux de façon locale, sans être supervisée par un processus externe, via le biais d'interfaces bien définies. Cette vision de la modularité se rapproche plus de la vision biologique, dans laquelle aucun processus global ne vient a priori superviser l'organisation des systèmes. Nous pensons que cette approche modulaire, telle que définie ci-dessus, est propice à faire émerger des nouveaux paradigmes d'apprentissage au sein d'architectures neuronales. En effet, la combinaison de principes et d'algorithmes simples peut mener au développement de mécanismes de calcul complexes au sein du système. Le comportement global du système résulte alors de l'interaction entre les modules et non seulement de la somme des comportements des modules pris individuellement : il s'agit de systèmes complexes. Des exemples de modèles artificiels ont exploré cette idée de modularité. Un exemple en robotique autonome est l'architecture de *sumsumption* de [Brooks 1986](#). Ces travaux construisent une architecture robotique composée de modules comportementaux simples, tels que "marcher", "regarder", qui interagissent entre eux. L'interaction de tous ces comportements de base permet au système de réagir de manière autonome à son environnement. Dans cet exemple, les modules ont une structure préétablie. Pour intégrer le contexte d'apprentissage, nous nous intéressons plutôt à des architectures modulaires dont ces modules sont a priori indifférenciés et interchangeables, et vont se spécialiser dans l'architecture au cours de l'apprentissage.

Dans cette démarche d'exploration d'un modèle d'apprentissage modulaire inspiré du cortex, nous proposons dans cette thèse de construire une architecture de cartes auto-organisatrices, ou cartes de Kohonen ([Kohonen 1982](#)). Les cartes de Kohonen sont un exemple de modèle d'apprentissage initialement inspiré de considérations biologiques, modélisant de façon très simple l'organisation spatiale observée dans les aires corticales. Ces modèles sont caractérisés par leur capacité à représenter des données de façon ordonnée sur un espace de dimension plus faible (typiquement une ou deux dimensions). L'algorithme d'apprentissage d'une carte auto-organisatrice suit un principe assez simple : elle est composée de vecteurs de l'espace d'entrée (prototypes) positionnés sur une grille de faible dimension, initialement distribués aléatoirement dans l'espace d'entrée. L'apprentissage est réalisé en présentant les entrées une à une à la carte, en trouvant leur *Best Matching Unit* comme le prototype le plus proche de l'entrée, puis en déplaçant ce prototype ainsi que ses voisins dans la grille vers l'entrée. À l'issue de ce processus d'apprentissage, la grille munie des prototypes est dépliée sur l'espace d'entrée. N'importe quel point de l'espace d'entrée peut alors être représenté par une position sur la grille.

La littérature autour des cartes de Kohonen, bien qu'extensive, s'est principalement attachée à l'augmentation des performances des cartes sur des applications d'apprentissage automatique et de fouille de données, comme de la compression d'image ou du clustering ([Kohonen 2013](#)). Nous pensons que leur inspiration biologique, leurs propriétés d'organisation et de représentation en deux dimensions d'un espace complexe et la simplicité de leurs règles de mise à jour en font des candidates naturelles pour la création d'une architecture modulaire d'apprentissage. Cette

idée d'architecture modulaire de cartes n'est pas nouvelle et est d'ailleurs formulée par Kohonen dès 1995 :

Un objectif à long terme de l'auto-organisation est de créer des systèmes autonomes dont les éléments se contrôlent mutuellement et apprennent les uns des autres. De tels éléments de contrôle peuvent être implémentés par des SOMs spécifiques ; le problème principal est alors l'interface, en particulier la mise à l'échelle automatique des signaux d'interconnexion entre les modules et la collecte de signaux pertinents comme interface entre les modules. Nous laisserons cette idée aux recherches futures.

Traduit de [Kohonen 1995](#)

Depuis, bien que des travaux aient proposé des architectures de carte auto-organisatrices, peu ont effectivement exploré l'aspect topographiquement ordonné et la simplicité des règles mise à jour des poids d'une carte pour les assembler en architectures modulaires comportant des rétroactions : des architectures non-hiérarchiques. C'est pourquoi nous nous intéressons à cet aspect dans cette thèse.

L'approche que nous avons privilégiée pour la construction d'une telle architecture est ascendante : nous définissons un modèle en s'appuyant sur l'état de l'art, puis nous cherchons à évaluer son comportement, afin de guider les améliorations ou applications qui en découleront. Nous proposons dans cette thèse un modèle de cartes auto-organisatrice pouvant être assemblées en tant que module d'une architecture, CxSOM. L'objectif de la thèse est ensuite de fournir un cadre méthodologique visant à comprendre et observer les mécanismes d'apprentissage émergent des règles de calcul que nous avons définies.

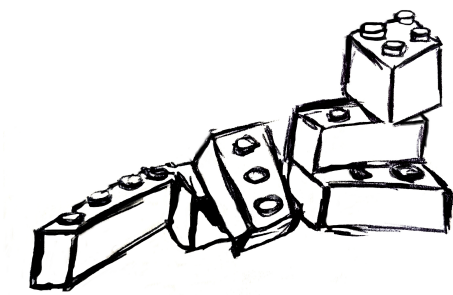
La notion d'architecture cognitive présentée plus haut cherche à explorer de grands principes liés à la cognition tels que l'apprentissage non-supervisé, autonome, le traitement de données temporelles, l'apprentissage sur le long terme sans oubli catastrophique des données précédentes et la fusion de données multimodales, s'inspirant du traitement multisensoriel du cerveau humain. L'architecture que nous proposons, par sa conception, vise à long terme l'application à de telles tâches. Cette problématique étant très vaste, nous avons choisi dans cette thèse de nous concentrer sur la tâche particulière d'apprentissage associatif de données multimodales. Il s'agit d'apprendre des relations existant entre des entrées provenant de différents espaces, en plaçant cet apprentissage de relations à un niveau interne à l'architecture et non en combinant les entrées. Le but est d'apprendre à la fois une représentation des entrées et de leurs relations. Dans cette optique d'apprentissage de représentation de relations entre entrées, nous proposerons des outils de visualisation et des indicateurs permettant de comprendre ces mécanismes.

En résumé, cette thèse vise à répondre à deux problématiques principales entremêlées : proposer une nouvelle architecture non-hiérarchique de cartes auto-organisatrices exploitant l'aspect topographiquement ordonné de ce modèle d'apprentissage, et étudier comment mettre en évidence et évaluer l'apprentissage associatif qui émerge d'une telle architecture.

Le manuscrit est organisé de la façon suivante. Le chapitre 1 présente un état de l’art des architectures de cartes auto-organisatrices existantes dans la littérature. Ces modèles d’architectures sont issus de plusieurs domaines, de l’apprentissage automatique aux neurosciences computationnelles. Le chapitre propose une revue des modèles principaux en s’attachant à unifier les notations et leurs désignations afin d’identifier les points communs et différences principales de conception de ces modèles. Cet état de l’art nous permettra de situer le modèle que nous proposons au regard de la littérature existante.

Nous détaillerons ensuite le modèle d’architecture décentralisée de cartes auto-organisatrices que nous proposons dans cette thèse, que nous avons appelé CxSOM, pour *Consensus-driven Multi-SOM*. Nous définissons un modèle de carte qui peut être assemblé à volonté en architecture non-hiérarchiques. Dans ce modèle, les activités des différentes cartes sont interdépendantes et l’apprentissage s’appuie sur une recherche de consensus entre les cartes pour la recherche d’un BMU. Nous présenterons ce modèle au chapitre 2. Le chapitre 3 s’attarde ensuite sur une analyse plus approfondie de la recherche de consensus constituant l’interface entre les cartes afin de valider ce mécanisme en tant qu’algorithme de choix de BMU pour l’apprentissage.

Si notre approche a pour but à long terme de concevoir une architecture comportant de nombreux modules ainsi que des connexions temporelles, nous avons concentré cette thèse sur l’analyse expérimentale des comportements d’apprentissage dans des architectures de deux et trois cartes. Le pari de l’approche ascendante est de faire émerger des nouveaux comportements ; aussi faut-il pouvoir les mettre en évidence. Nos travaux se sont vite confrontés à une difficulté de visualisation d’une telle architecture de cartes. Cette thèse met l’accent sur une méthodologie d’analyse expérimentale de cette architecture modulaire, ce qui nous permettra d’en tirer des comportements élémentaires qui serviront à poser les jalons de la construction d’une architecture plus complexes. Nous introduisons au chapitre 4 cette méthode expérimentale et un cadre de représentation des entrées, et questionnons comment exprimer qu’une architecture de cartes encode les entrées et leurs relations. Nous présentons ensuite au chapitre 5 les comportements élémentaires d’apprentissage associatif observés sur des architectures de deux et trois cartes en une dimension, à partir des représentations que nous avons proposées. Nous présenterons en particulier un comportement de prédiction d’entrée, rendu possible par les rétroactions présentes dans le modèle d’architecture. Nous explorons au chapitre 6 des indicateurs numériques d’évaluation de l’apprentissage associatif par l’architecture de cartes, dans le but d’étendre l’analyse du modèle à des architectures plus grandes, qui seraient difficilement représentables graphiquement. Le chapitre 7 étend enfin les mécanismes d’apprentissage que nous avons identifiés à des architectures de cartes en deux dimensions, se plaçant comme une étude préliminaire pour saisir la scalabilité du modèle.



Chapitre 1

Architectures de cartes auto-organisatrices

Sommaire

1.1	Introduction	1
1.2	Les cartes auto-organisatrices de Kohonen comme modules d'une architecture	2
1.2.1	Description du modèle de carte auto-organisatrice de Kohonen	2
1.2.2	La SOM, un algorithme d'apprentissage de représentation	4
1.3	Inspiration biologique des architectures de cartes	6
1.3.1	Inspiration biologique des cartes de Kohonen	6
1.3.2	Rétroactions dans le traitement de l'information multisensorielle du cortex	7
1.4	Architectures de cartes auto-organisatrices	9
1.4.1	Éléments de comparaison	10
1.4.2	Architectures hiérarchiques de cartes	11
1.4.3	Architectures non hiérarchiques de cartes auto-organisatrices	20
1.4.4	Apprentissage de séquences et architectures de cartes auto-organisatrices	32
1.5	Discussion et axe de recherche	36

1.1 Introduction

Les travaux que nous présentons dans cette thèse explorent la création d'une architecture non hiérarchique de cartes auto-organisatrices (SOM). Les cartes auto-organisatrices sont principalement utilisées en tant qu'algorithme d'apprentissage non supervisé appliqué à des tâches de réduction de dimension, de visualisation de données ou de classification. Certains travaux ont

étudié l'utilisation de plusieurs cartes collaborant entre elles sur différentes applications, en général afin d'améliorer les performances de classification ou de regroupement de données d'une carte auto-organisatrice classique. Ces travaux se retrouvent sous le terme de SOM hiérarchiques, SOM multi-couches, ou *Deep SOM*. Cependant, peu de travaux ont exploré l'aspect topologiquement ordonné et la simplicité des règles de mise à jour d'une carte pour les assembler en architectures modulaires comportant des rétroactions, c'est-à-dire des architectures non-hiérarchiques.

L'étude d'une architecture non hiérarchique de SOM est motivée par leur inspiration biologique. Le cortex faisant apparaître des aires interagissant entre elles avec des boucles de rétroaction, la création d'une architecture non hiérarchique de cartes s'inscrit dans la continuité de cette inspiration biologique. Aussi, les architectures de cartes bio-inspirées que nous avons relevées dans la littérature se définissent des domaines soit de l'apprentissage automatique, soit des neurosciences computationnelles ou de l'apprentissage incarné en robotique (*Embodied intelligence*) (Smith et Gasser 2005 ; Cangelosi et al. 2015), à la frontière entre étude de la biologie et apprentissage automatique. Ce chapitre se veut une relecture d'un ensemble de travaux définissant des architectures de cartes auto-organisatrices, issues de plusieurs domaines de recherche, en s'intéressant aux différentes formes de modularité qu'elles implémentent.

Nous présenterons le modèle général d'une carte de Kohonen et ses comportements fondamentaux, puis répertorions les différents types de structures se présentant comme des architectures de cartes. Nous nous attacherons en particulier à définir leurs structures et leurs règles d'apprentissage sous une forme unifiée, afin de mieux comparer les mécanismes de calcul présents dans ces architectures. Nous pourrons ainsi définir en pratique la notion d'architecture modulaire non-hiérarchique et comment se placent nos travaux dans cette taxonomie.

1.2 Les cartes auto-organisatrices de Kohonen comme modules d'une architecture

1.2.1 Description du modèle de carte auto-organisatrice de Kohonen

Le modèle de cartes auto-organisatrices a été initialement développé par Kohonen (Kohonen 1982) ; nous utiliserons les termes cartes de Kohonen et SOM de façon équivalente pour désigner ce modèle initial. Une carte de Kohonen est un algorithme de quantification vectorielle. Les algorithmes de quantification vectorielle cherchent à représenter un ensemble de données d'entrées issues d'un espace \mathcal{D} en un nombre réduit de vecteurs codes, appelés prototypes. Les cartes auto-organisatrices y ajoutent une topologie sur laquelle s'appuie cette représentation de l'espace d'entrée. Les prototypes ω sont alors positionnés sur les nœuds d'un graphe et indexés par une position p . Ce graphe se présente en général comme une grille de faible dimension, par exemple une grille 2D, définissant une correspondance directe entre la dimension de la carte et les positions

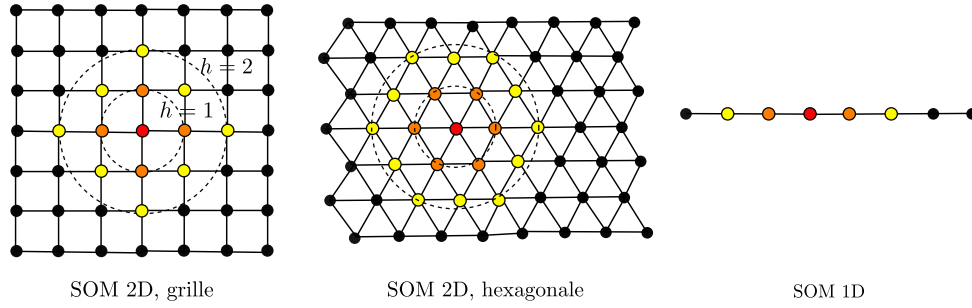


FIGURE 1.1 – Exemples de topologies utilisées pour des SOM. Les SOM en deux dimensions sont les plus communément utilisées dans la littérature, sous forme d'une grille carrée hexagonale. Les SOM une dimension sont parfois utilisées. Cette topologie permet de calculer des distances entre nœuds et un voisinage. Les nœuds orange sont par exemple à une distance $h = 1$ du nœud rouge sur chacune des cartes.

p des nœuds (*mapping*). Des exemples de topologies de SOM 1D et 2D sont par exemple illustrées en figure 1.1.

Avant apprentissage, les prototypes sont initialisés aléatoirement dans l'espace d'entrée. Une itération d'apprentissage comporte ensuite trois étapes :

1. Une entrée X est présentée à toute la carte.
2. Le nœud ayant le prototype le plus proche de X selon une distance $d(\omega(p), X)$ est choisi comme *Best Matching Unit* (BMU) de la carte. Son indice est noté Π .

$$\Pi = \arg \min_p d(\omega(p), X) \quad (1.1)$$

3. Le prototype du BMU $\omega(\Pi)$ ainsi que les prototypes des nœuds voisins sont déplacés vers l'entrée X . Le déplacement est pondéré par leur degré de proximité au BMU par une fonction de voisinage $H(p, \Pi)$:

$$\forall p, \omega(p) \leftarrow \omega(p) + \alpha H(\Pi, p) (X - \omega(p)) \quad (1.2)$$

H définit l'amplitude de modification de chaque prototype $\omega(p)$, en modulant le taux d'apprentissage α par $H(p, \Pi)$. H est maximale à la position du BMU et décroissante autour de cette position, de sorte que les prototypes des nœuds les plus proches de Π soient les plus influencés par le déplacement. Il s'agira par exemple d'une fonction rectangulaire, triangle ou gaussienne, illustrées en figure 1.2.

L'algorithme de mise à jour des cartes de Kohonen repose à la fois sur un mécanisme de compétition par la sélection de la BMU de la carte et un processus de coopération avec le déplacement des unités voisines de la BMU, aussi nommé « Winner Take Most ». Le processus de mise à jour des poids d'une carte de Kohonen se traduit par un dépliement de la carte

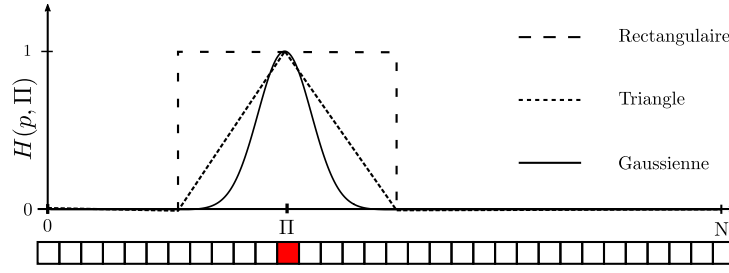


FIGURE 1.2 – Exemples de fonctions de voisinage (Rectangulaire, Triangle ou Gaussienne), centrées sur le BMU II, couramment utilisées sur une carte, ici en une dimension.

dans l'espace d'entrée. On parlera donc aussi de *dépliement* d'une carte lorsque qu'on parle d'apprentissage. Ce dépliement est représenté en figure 1.4 pour des exemples de cartes en une et deux dimensions, se dépliant sur des données en deux dimensions. À la fin de l'apprentissage, la carte conserve la structure topologique des entrées :

- Elle conserve les distances : deux prototypes ayant une distance proche dans la carte seront également proches selon la distance définie dans l'espace d'entrée. On observe alors une continuité des valeurs des poids au sein de la carte.
- Elle conserve les densités. Une zone dense de l'espace d'entrée \mathcal{D} sera représentée par plus de prototypes au sein de la carte, qu'une zone de faible densité.

Le calcul de distances, utilisé dans la version initiale des SOM, est remplacé dans d'autres modèles de cartes par le calcul d'une *activation* $a(p, X)$ liant les poids des nœuds et les entrées. Il s'agit généralement d'une activation gaussienne :

$$a(p, X) = \exp - \frac{\|X - \omega(p)\|^2}{2\sigma^2} \quad (1.3)$$

Ce calcul d'activation est représenté en figure 1.3, et s'apparente à la définition de champs récepteurs gaussiens pour chaque neurone de poids $\omega(p)$, inspirée de la biologie. L'activation est un champ 2D, dépendant de X (en ordonnée) et de p (en abscisse). À p fixé, $a(X)$ serait l'équivalent d'un champ récepteur gaussien, pour le neurone situé en p et de valeur préférentielle $\omega(p)$, tracé sur la figure par la courbe bleue. Le paramètre σ intervenant dans l'équation détermine la largeur de cette gaussienne. L'activation considérée pour le choix du BMU est quant à elle $a(p)$, calculée à X fixé et tracée sur la figure par la courbe en rouge. Le Best Matching Unit est choisi comme l'argmax de cette activation.

1.2.2 La SOM, un algorithme d'apprentissage de représentation

La carte de Kohonen se distingue d'autres algorithmes de quantification vectorielle par la topologie introduite par la carte dans l'ensemble des prototypes. Par sa topologie, une carte de Kohonen permet d'extraire une *représentation* de l'espace d'entrée en faible dimension : une

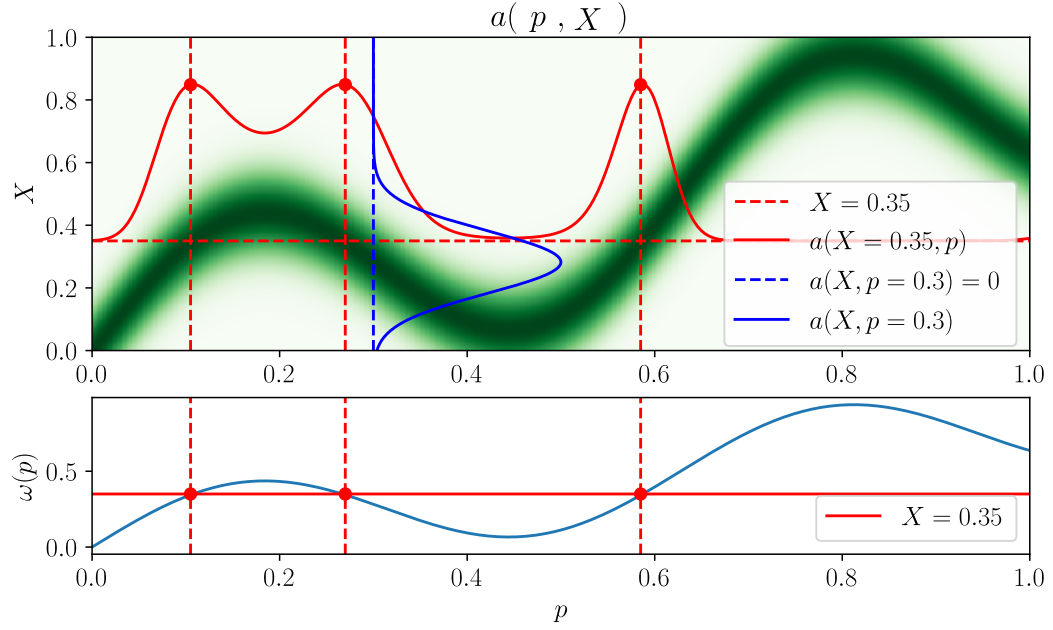


FIGURE 1.3 – Activation $a(p, X)$ selon les valeurs de $X \in [0, 1]$ (en ordonnées) et $p \in [0, 1]$ (en abscisse). À p fixé, $a(X)$ correspond au champ récepteur gaussien (en bleu) du neurone situé en position p , de valeur préférentielle $\omega(p)$, dont les valeurs sont tracées en bas. À X fixé, $a(p)$ est l'activation de la carte, et le BMU est choisi comme le nœuds situé à l'argmax de cette activation.

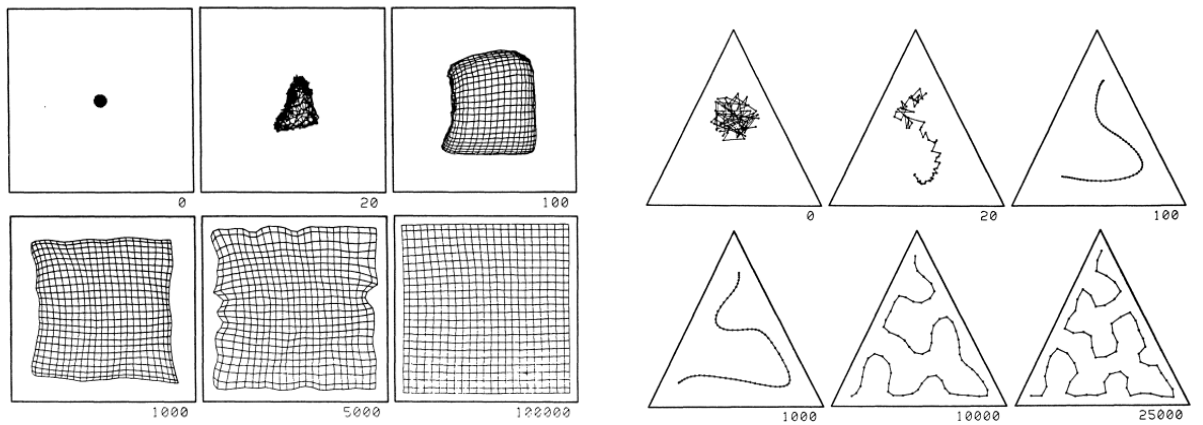


FIGURE 1.4 – Exemples de dépliement d'une SOM 2D sur des données dans un carré (à gauche) et d'une SOM 1D sur des données dans un triangle 2D (à droite) au cours des itérations d'apprentissage (Kohonen 1995)

carte 2D extrait ainsi une paramétrisation en 2D de l'espace des entrées \mathcal{D} . Chaque élément de l'espace d'entrée peut alors être représenté par un vecteur, de la dimension de la carte.

En théorie, les cartes peuvent être en une dimension (ligne), deux dimensions (grilles), ou de dimensions plus grandes. Les cartes peuvent aussi être des graphes de forme plus variable. En pratique, les grilles en deux dimensions sont les supports les plus couramment utilisés. Les cartes de dimensions supérieures sont très rarement utilisées dans la littérature. Le coût de l'algorithme d'apprentissage dépend en effet du nombre de nœuds, et celui-ci augmente exponentiellement lorsqu'on augmente la dimension d'une carte de Kohonen.

Les cartes une dimension sont quant à elles plus limitées que les cartes 2D en termes de représentation des données et sont donc rarement utilisées en pratique ou sur des applications dérivées, par exemple pour de la planification de chemin (Frezza-Buet 2020). Cependant, elles se prêtent mieux à la représentation graphique et au développement de nouveaux modèles de SOM que les cartes 2D. Les travaux conduits en Cottrell, J.-C. Fort et al. 1998 ; J. Fort 2006 ; Cottrell, Olteanu et al. 2016 apportent par exemple une formalisation mathématique de l'algorithme de Kohonen et prouvent la convergence de cartes une dimension. Les auteurs se heurtent cependant à la preuve de convergence pour des cartes en deux dimensions. Les processus intervenant dans des cartes 1D sont donc mathématiquement difficiles à formaliser, et cette difficulté augmente avec les dimensions de la carte.

Si les cartes de forme autre que des grilles 1D ou 2D sont moins couramment utilisées, elles peuvent présenter des avantages. Ainsi, des cartes structurées en arbre telles que développées en Koikkalainen et Oja 1990 permettent une recherche de BMU plus rapide au sein de la carte, adaptées à des données d'entrée présentant une structure hiérarchique. Certains modèles construisent une carte de Kohonen incrémentale en ajoutant des nœuds au fur et à mesure de l'apprentissage, générant une carte de Kohonen sous forme d'un graphe construit par l'algorithme, par exemple en Fritzke 1995 ; Alahakoon et al. 2000 ; Yamaguchi et al. 2010.

1.3 Inspiration biologique des architectures de cartes

1.3.1 Inspiration biologique des cartes de Kohonen

Le développement des cartes auto-organisatrices par Kohonen est initialement inspiré par les cartes topologiques observées dans les aires du cortex cérébral. Le cortex est cartographié en *aires* distinctes selon la fonction principale présumée de la zone correspondante. Ce découpage fonctionnel fait apparaître des grandes catégories d'aires corticales. Certaines aires sont dites sensorielles, car elles reçoivent des entrées sensorielles via le thalamus. Certaines aires sont dites motrices et reliées aux muscles, via des structures sous corticales et permettent ainsi un contrôle moteur. Enfin, des aires sont identifiées comme traitant des informations venant de plusieurs

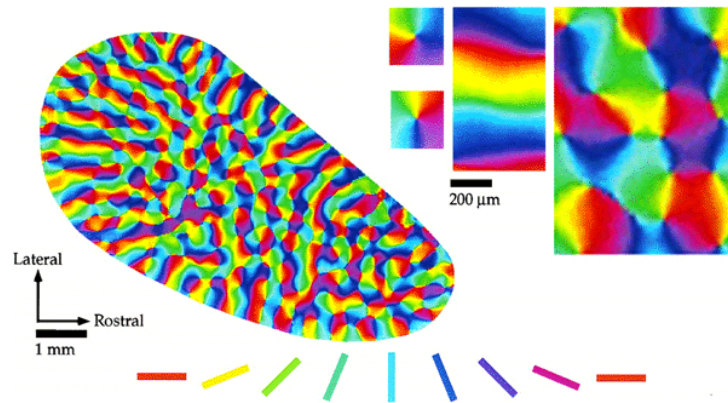


FIGURE 1.5 – Représentation des réponses du cortex visuel V1 à un stimulus visuel (bâtonnets d'orientations spatiales différentes). Les neurones répondant à une certaine orientation sont affichés de la même couleur. On observe une continuité entre les neurones proches dans le cortex et l'orientation à laquelle ils répondent. Cette propriété de continuité est partagée par l'organisation des SOM.

autres aires. De nombreux travaux suggèrent une organisation sous forme de carte topologiquement ordonnées dans différentes aires du cortex cérébral : les neurones proches dans le substrat cortical réagissent à des stimuli proches. Un exemple est ainsi celui du cortex visuel V1, représenté en figure 1.5. L'aire associée à l'audition présente aussi une organisation topographique (Reale et Imig 1980). Cette organisation se retrouve également dans de nombreuses autres aires sensorielles ou de plus haut niveau de traitement de l'information (Kohonen 1995). Une carte de Kohonen ne doit cependant pas être considérée comme une modélisation biologiquement plausible d'une aire du cortex cérébral, mais plutôt comme une adaptation au niveau computationnel d'un concept biologique, ici le concept d'organisation topographique dans les cortex sensoriels.

1.3.2 Rétroactions dans le traitement de l'information multisensorielle du cortex

L'aspect multisensoriel du traitement de l'information s'appuie sur des connexions entre aires corticales. Cette connectivité peut-être étudiée de plusieurs points de vue : d'un point de vue structurel, en se basant sur des éléments anatomiques ou d'un point de vue fonctionnel. Dans le cas fonctionnel, la connexion de deux aires est déduite de l'existence de dépendances statistiques entre l'activation des neurones des deux aires, observées par électroencéphalographie ou IRM fonctionnelle. Il faut noter cependant que ces observations traduisent une relation statistique et pas forcément une relation de cause à effet. La modélisation de la connectivité physique de ces aires à partir des observations reste donc l'objet de différentes théories cherchant à reproduire ces corrélations. Un exemple fonctionnel de traitement multisensoriel de l'information est l'effet ventriloque (Bonath et al. 2007) qui crée l'illusion que la source sonore provient de la marionnette

du ventriloque dont les mouvements de bouche sont coordonnés avec les paroles. Ce phénomène entraîne une activité dans le cortex visuel et auditif au niveau des neurones correspondant à l'emplacement exact de la source des stimuli de chacune des modalités. Après quelques millisecondes, une activité émerge dans le cortex auditif au niveau des neurones sensibles à l'emplacement spatial de la source du stimulus visuel, témoignant d'une interaction entre les cortex visuels et auditifs. Un autre exemple est l'effet McGurk (McGurk et MacDonald 1976) : ces psychologues ont montré que la présentation du son « ba » à un sujet associée à la présentation d'une vidéo d'une bouche prononçant « ga » amènent ce sujet à indiquer qu'il a entendu le son « da ». Historiquement, le traitement de l'information multisensorielle dans le cortex cérébral a été modélisé comme hiérarchique, des aires dites bas niveau alimentant des aires haut niveau permettant le traitement de l'information multimodale. De nombreux travaux montrent également l'existence de connexions directes entre aires sensorielles qui s'ajouteraient à une hiérarchie du traitement de l'information. Par exemple, de nombreuses connexions entre les aires corticales dédiées au traitement d'une modalité sensorielle ont été mises en évidence chez différentes espèces, par exemple en Felleman et Essen 1991 ; Calvert et Thesen 2004 ; Cappe et al. 2009 ; Foxe et Schroeder 2005 ; Schroeder et Foxe 2005. Un exemple d'une telle architecture corticale est illustrée en figure 1.6, faisant apparaître un grand nombre de rétroactions entre aires. Fonctionnellement, des coactivations entre aires corticales sont observées par exemple entre aires tactile et visuelle (Sathian et Zangaladze 2002), ou entre aires visuelle et olfactive (González et al. 2006). Ces connexions s'observent à différents niveaux de la hiérarchie du traitement de l'information : Kiefer et al. 2008 met par exemple en évidence un lien existant entre le cortex sensoriel auditif et l'aire dédiée à la représentation de concepts dans le cerveau humain. La structure du traitement de l'information dans les aires corticales ne se limite donc pas à un aspect hiérarchique, des connexions rétroactives entre aires existant à plusieurs niveaux du traitement de l'information.

En termes de modélisation du cortex, la théorie des zones de convergence divergence (Damasio 1989) suggère que certaines aires corticales servent d'espaces uniquement associatifs agrégeant les signaux des zones corticales prenant des modalités sensorielles en entrée et les propageant vers d'autres zones sensorielles. La théorie de la réentrée (Edelman 1982) postule quant à elle des connexions directes et réciproques entre les neurones de différentes zones sensorielles ou non. Les neurones d'une aire corticale peuvent alors être activés par à la fois un stimulus sensoriel et un stimulus provenant d'une autre aire corticale. Les travaux de Burnod 1989 modélisent le cortex en colonnes corticales et proposent qu'en chaque point du cortex se croisent des flux de connexions venant de neurones d'autres aires sensorielles, organisées en bandes. Ces théories reviennent régulièrement comme inspirant les modèles de calcul des architectures présentées dans ce chapitre.

La carte de Kohonen implémentant des concepts computationnels qu'on retrouve en biologie au niveau de l'aire cérébrale, nous pouvons chercher à pousser l'inspiration biologique au niveau des connexions entre les aires cérébrales, en construisant des connexions entre plusieurs cartes

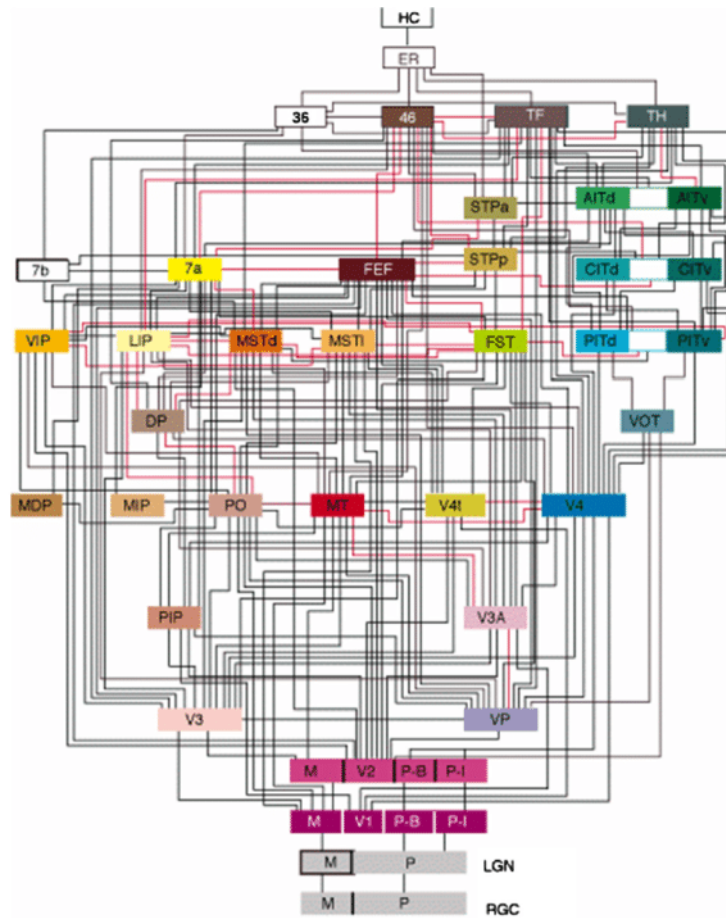


FIGURE 1.6 – Schéma de connexions entre aires sensorielles existant dans le cortex du singe, faisant apparaître des connexions rétroactives entre aires. Ce traitement fait apparaître plusieurs niveaux de hiérarchie tout en incluant des connexions entre aires d’un même niveau (Felleman et Essen 1991)

de Kohonen. De la même façon qu’une carte n’est pas un modèle biologique, il s’agit plutôt de développer un modèle computationnel qui ne soit pas biologiquement plausible au niveau neuronal, mais dont la structure du traitement de l’information est inspirée de celle du cerveau, ici la présence de plusieurs aires connectées entre elles, modélisées par l’utilisation de plusieurs cartes de Kohonen assemblées en une architecture.

1.4 Architectures de cartes auto-organisatrices

La littérature autour des architectures de cartes auto-organisatrices est assez peu fournie, ce qui motive notre travail d’exploration d’un modèle d’architecture. Nous avons constaté que la notion de modularité et de hiérarchie du traitement de l’information prend des significations complètement différentes en fonction des travaux. De plus, la création de modèles d’architec-

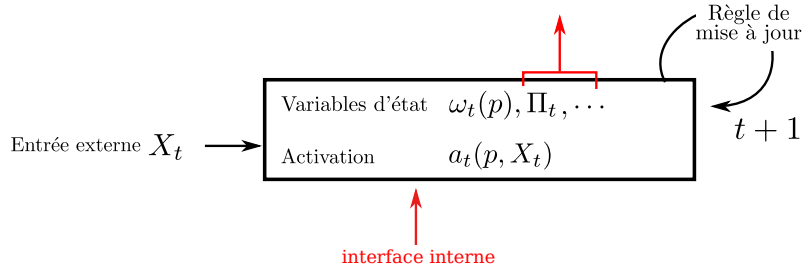


FIGURE 1.7 – Description d’une carte de Kohonen en tant que module d’une architecture modulaire. Un module prend des entrées et possède des variables d’état, ici les poids ω , dont l’évolution dans le temps est régie par des règles de mise à jour ou équations d’évolution. L’interface entre les modules (en rouge) est un ensemble de variable d’états, accessible par d’autres modules. Il peut par exemple s’agir de l’activation ou du BMU.

tures de cartes relève de plusieurs domaines, de l’apprentissage automatique aux neurosciences computationnelles en passant par la robotique.

Cette section présente les principaux types de modèles d’architecture que nous avons rencontrés dans la littérature, en les détaillant sous une notation unifiée, introduite en 1.2.1, afin d’offrir une vue synthétique du domaine. L’analyse de ces modèles nous permet d’apporter une taxonomie classant ces différents types de modèles d’architectures. À partir de cette catégorisation, nous pourrions situer plus clairement dans la littérature l’architecture que nous développons et émettre des hypothèses concernant les comportements qu’on peut en attendre.

1.4.1 Éléments de comparaison

Détaillons l’approche sous laquelle nous abordons l’étude de ces architectures. Il s’agit d’abord de différencier les architectures dédiées au traitement d’un problème particulier et les architectures génériques. Pour résoudre un problème complexe, une démarche courante est de le décomposer en sous-problèmes, puis de créer des structures spécifiquement adaptées à chaque sous-tâche. L’assemblage des résultats de chaque structure propose alors une solution pour résoudre le problème général. Cette méthode est souvent rencontrée dans la conception d’architectures de réseaux de neurones. Nous différencions cette vision d’architecture dédiée à une application, de la notion de modèle d’architecture. On entend par modèle d’architecture le cadre de calcul sous-jacent au modèle, appliqué ou non, défini par ses règles de construction. Dans ce chapitre, nous nous intéressons spécifiquement aux modèles d’architectures et analyserons leur aspect modulaire de ces modèles d’architectures. Par module, nous entendons un élément possédant des règles d’évolution temporelles à partir d’éléments d’entrées. Nous définissons comme architecture modulaire l’assemblage de modules de même type par une interface définie. Des modules doivent pouvoir être ajoutés ou retirés à une architecture modulaire sans modifier la structure interne des autres modules de l’architecture.

Les cartes auto-organisatrices s'interprètent facilement en tant que module d'une architecture : elles prennent des entrées X et ont des variables d'état ω mises à jour par des règles d'évolution. La conception du modèle d'architecture consiste ensuite à définir l'interface entre les modules : il peut s'agir de la position du BMU, des poids ou des valeurs d'activités. Un schéma d'une carte considéré en tant que module d'une architecture est par exemple représenté en figure 1.7. Nous présenterons dans cette section des modèles d'architectures composés uniquement de cartes auto-organisatrices, en se focalisant sur les mécanismes d'apprentissage induits par l'utilisation de plusieurs cartes combinées en tant que modules.

L'étude des architectures nous a amenée à différencier les structures selon plusieurs aspects, résumés en figure 1.8. Nous distinguons deux structures principales d'architectures de cartes : les architectures *hiérarchiques* et les architectures *non hiérarchiques*. Une architecture est dite hiérarchique lorsqu'il n'existe pas de rétroactions, directes ou indirectes, dans les connexions entre cartes. Dans ce cas, on peut définir des niveaux de cartes. Les cartes d'un même niveau ne sont pas connectées entre elles, ont au moins une connexion arrivant du niveau précédent et/ou une connexion sortant vers le niveau suivant. Une architecture est au contraire non hiérarchique lorsqu'il existe au moins une boucle de rétroaction. Cette boucle peut être une connexion bidirectionnelle entre deux cartes ou une boucle comprenant plus de nœuds. Ces boucles implémentent des rétroactions entre les cartes de l'architecture. Au sein des architectures non hiérarchiques, nous différencions deux paradigmes que nous détaillerons dans la section correspondante : les architectures centralisées et décentralisées.

Au sein de ces catégories d'architectures, nous nous intéressons à la façon dont les cartes interagissent. Ces interactions peuvent être gérées par un processus extérieur aux cartes et global à l'architecture, ou plutôt être traitées localement par les mécanismes d'organisation de chaque carte. Nous relèverons également les éléments transmis entre les cartes : certaines architectures de cartes utilisent la position du BMU, son poids, ou un ensemble d'activités de neurones d'une carte. Enfin, la temporalité de l'algorithme de mise à jour de l'architecture peut se présenter de différentes façons. Nous appelons mise à jour séquentielle un algorithme dans lequel l'apprentissage est complètement effectué sur un module, avant de mettre à jour les suivants. La mise à jour est synchrone lorsqu'on peut définir des itérations communes à l'architecture, pendant lesquelles tous les modules seront mis à jour. La mise à jour est asynchrone lorsqu'un module n'est mis à jour que lorsqu'un signal déclencheur lui parvient.

1.4.2 Architectures hiérarchiques de cartes

Présentons d'abord les architectures hiérarchiques de cartes, qui ne présentent pas de rétroactions. Ces architectures se retrouvent également sous les termes de *Deep SOM* ou SOM multicouches. Nous avons relevé deux paradigmes de construction : dans un premier cas, l'architecture s'appuie sur un processus extérieur aux cartes, qui sélectionne et attribue les entrées

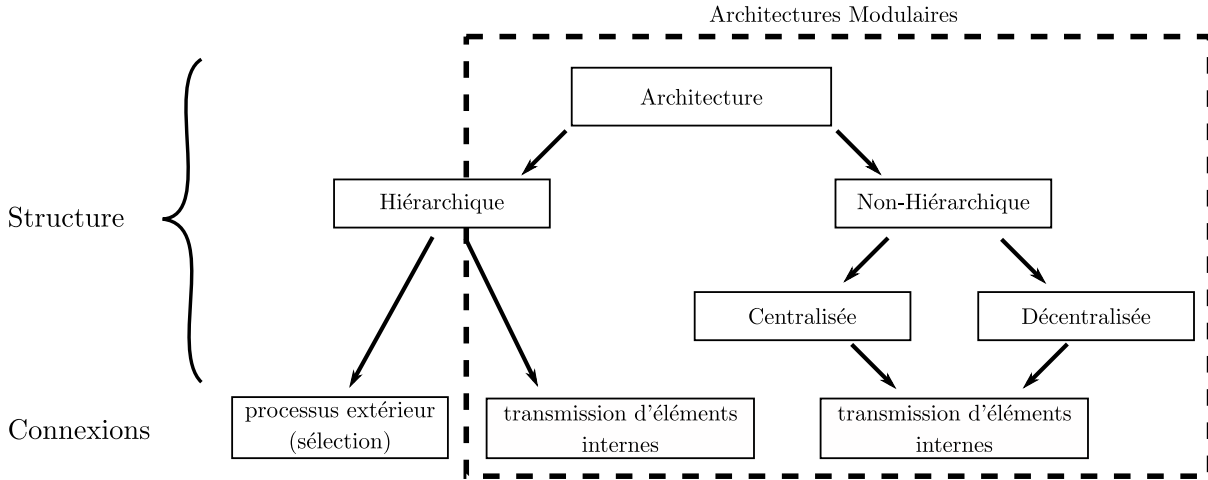


FIGURE 1.8 – Taxonomie des architectures de cartes présentées dans ce chapitre. Nous analyserons comment leurs caractéristiques structurelles : hiérarchiques ou non hiérarchiques, centralisées, décentralisées, façonnent leur comportement d'apprentissage. Nous analyserons également leur interface de communication. Nous n'avons pas relevé d'architecture non hiérarchique s'appuyant sur le principe de sélection, car ce principe est inhérent à une organisation hiérarchique.

à présenter à une carte. Dans l'autre, l'architecture est composée de cartes dont certaines apprennent les entrées externes et d'autre apprennent sur des éléments de sortie d'autres cartes. Par facilité de représentation, les schémas des architectures seront présentés avec des cartes en une dimension. Cependant, toutes les architectures présentées utilisent des cartes en deux dimensions, parfois plus. Une revue des méthodes de clustering basées sur des SOM hiérarchiques a également été effectuée en [Henriques et al. 2012](#).

Architectures hiérarchiques par sélection

Nous appelons architecture par sélection un ensemble de cartes organisés en différents niveaux, et dont la sortie d'un niveau de cartes permet de diviser l'espace d'entrée en sous-espaces qui seront utilisés comme entrées par le niveau de cartes supérieur. Détaillons par exemple l'architecture développée en [Barbalho et al. 2001](#), représentée en figure 1.9. Le premier niveau de cette architecture est une carte classique, prenant des entrées $X \in \mathcal{D}$. Une première étape consiste en un apprentissage complet de la carte du premier niveau. Le second niveau est composé de plusieurs cartes ; chacune de ces cartes est associée à un des nœuds de la première. Lors la deuxième phase de l'apprentissage, les données d'entrées sont réparties en plusieurs sous-ensembles, tels que chaque sous-ensemble \mathcal{D}_i est l'ensemble des entrées X_t ayant i pour position du BMU associé à l'entrée. Chaque carte i du deuxième niveau est alors entraînée sur son espace \mathcal{D}_i , les cartes du premier niveau n'étant plus mises à jour. L'architecture de cartes peut être définie à l'avance comme en [Barbalho et al. 2001](#) ou de façon incrémentale en s'adaptant aux données, comme en

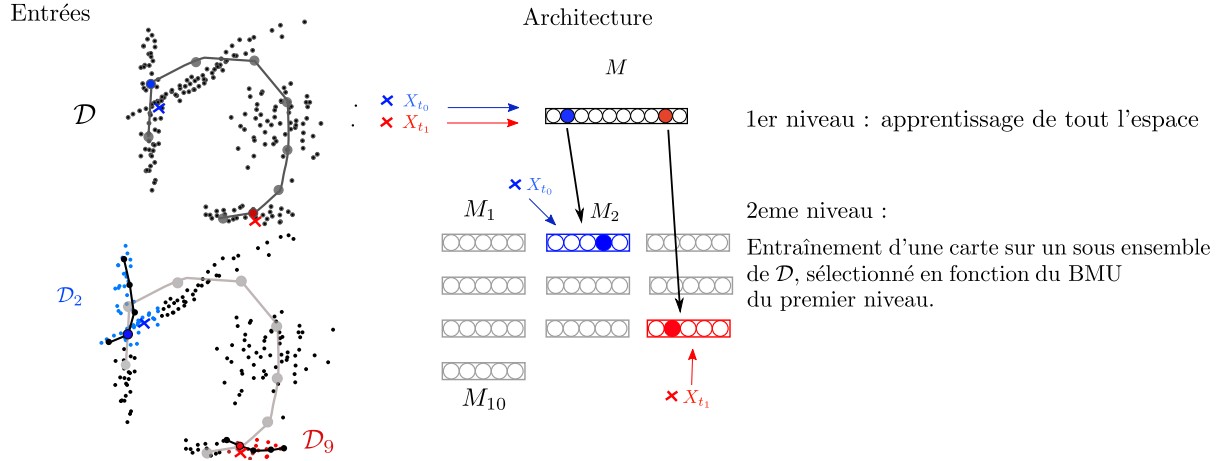


FIGURE 1.9 – Exemple d’architecture hiérarchique sélective. La carte M du premier niveau est entraînée sur tout l’espace d’entrée \mathcal{D} . Après apprentissage, la carte permet de filtrer les entrées pour les envoyer vers une carte du niveau suivant. Dans cet exemple, la position du BMU de la carte du niveau 1 permet de sélectionner une carte du niveau 2, comme c’est le cas en [Barbalho et al. 2001](#). L’entrée permet d’entraîner une carte du deuxième niveau. Chacune des cartes du niveau 2 apprend alors sur un sous-espace d’entrée. Le sous espace \mathcal{D}_1 lié au BMU à la position 1 alimente alors une carte du deuxième niveau, M_1 .

[Costa et al. 2016](#). L’ensemble des prototypes des cartes, non seulement ceux du dernier niveau, forment alors une cartographie plus précise de l’espace d’entrée : l’erreur de quantification vectorielle y est plus faible. Ce processus est sélectif dans la mesure où chaque carte se voit sélectionnée pour l’apprentissage d’une entrée en fonction de l’état du niveau précédent.

Ce principe se retrouve en [Miikkulainen 1992](#). Les auteurs utilisent une architecture du même type, mais pour traiter des données dont la structure est hiérarchique, ici des phrases écrites. La structure de l’architecture est similaire : une carte d’un premier niveau prend une entrée des phrases complètes, et permet d’extraire une représentation globale des entrées. Une fois cette carte entraînée, chaque carte du deuxième niveau apprend sur le sous-espace de phrases ayant un même BMU au premier niveau. Contrairement aux exemples précédents, les auteurs filtrent l’entrée avant de la transmettre à la carte du deuxième niveau, pour en extraire les dimensions pertinentes pour le sous-ensemble en question. L’aspect hiérarchique de l’architecture permet ainsi de découvrir des motifs hiérarchiques dans la dimension des données parallèlement à leur quantification sur leur valeur. Cette découverte de structures hiérarchiques dans les entrées par des architectures de cartes se retrouve en [Ordonez et al. 2010](#) ; [Dittenbach et al. 2000](#). Nous notons que le choix de répartition des sous-ensembles du deuxième niveau repose dans tous les modèles présentés ici sur la position du BMU du premier niveau, avec éventuellement des variantes comme en [Suganthan 2001](#) qui choisit de considérer plusieurs BMU par entrée pour décomposer l’espace en sous-ensembles qui se chevauchent.

L'application privilégiée de ces architectures sélectives est d'améliorer la quantification vectorielle réalisée dans une SOM, en décomposant cette quantification sur un ensemble de cartes qui apprennent chacune sur des sous-groupes de données. Ces sous-groupes sont détectés automatiquement par l'architecture. Cette décomposition peut permettre la découverte de structures hiérarchiques dans les données d'entrées. Le premier niveau seul est déjà une représentation générale de l'espace d'entrée. L'augmentation du nombre de cartes et leur séparation dans les niveaux supérieurs permet une meilleure précision en termes de quantification vectorielle sur chaque sous-espace et cette division du travail réduit le coût du calcul.

Ces travaux montrent que l'agrégation de cartes de Kohonen en architecture permet d'améliorer les performances d'une SOM en tant qu'algorithme de quantification vectorielle et de diversifier les représentations extraites par la carte en y ajoutant un aspect hiérarchique. Cependant, bien que la position du BMU soit utilisée pour la décomposition de l'espace d'entrée en sous-espaces, l'aspect de représentation topologique n'est pas spécialement exploité dans ce type d'architectures. C'est plutôt le principe de *clustering* qui permet de séparer l'espace d'entrée en sous-espaces, ainsi ce type d'architecture pourrait aussi se construire à partir d'autres algorithmes de clustering comme K-means (Linde et al. 1980).

Dans notre analyse, nous ne considérons pas ces architectures comme modulaires, par l'aspect global de l'algorithme. C'est en effet un processus extérieur aux cartes qui permet de sélectionner la carte du niveau suivant, de créer ou non des cartes à ajouter à l'architecture et de décomposer les entrées en sous-espaces. L'algorithme de mise à jour des poids de l'architecture est ainsi sous la dépendance d'un processus global. Seul ce processus traite l'information de chacune des cartes et apporte une connexion entre cartes.

Architectures hiérarchiques par transmission de représentation interne

Certaines architectures implémentent une interface entre cartes gérée directement au niveau de l'algorithme d'organisation de la carte. La gestion de l'interface est alors locale à une carte : aucune surcouche algorithmique globale à l'architecture n'intervient dans les tâches de transmission d'information. Notons que la gestion des itérations peut rester globale à l'architecture. Contrairement aux architectures par sélection, le deuxième niveau de cartes de ces architectures hiérarchiques ne prend plus comme entrée un élément de l'espace d'entrée de l'architecture, mais des éléments des cartes des couches précédentes, tels que la position, le poids du BMU ou un ensemble d'activation. Ces éléments sont une représentation de l'entrée, transmise à la couche supérieure.

Des travaux proposant un modèle de SOM hiérarchique ont été développés assez tôt en [Luttrell 1989](#). Ces travaux proposent un algorithme de quantification vectorielle hiérarchique à partir de cartes de Kohonen, et montrent expérimentalement qu'il s'agit d'une méthode moins

coûteuse qu'une SOM classique pour quantifier des données de grande dimension. Notons que ces travaux ne sont pas applicables seulement aux SOM mais à tout algorithme de quantification vectorielle. Les auteurs utilisent ici le poids du BMU comme interface entre les SOM.

Ensuite, le modèle HSOM (Lampinen et Oja 1992) construit une architecture composée de deux cartes : une première carte $M^{(1)}$ se déploie sur des entrées $X^{(1)}$, et une deuxième carte $M^{(2)}$ reçoit ensuite comme entrée la position du BMU $\Pi^{(1)}$ de la première carte ; cette architecture est illustrée en figure 1.10. Le BMU de la première carte est alors défini par :

$$\Pi^{(1)} = \arg \min_p (||\omega^{(1)}(p) - X^{(1)}||^2)$$

Le BMU de la deuxième couche est ensuite calculé comme :

$$\Pi^{(2)} = \arg \min (||\omega^{(2)}(p) - \Pi^{(1)}||^2)$$

La deuxième carte réalise ainsi de la quantification vectorielle sur les positions du BMU de la première carte. Les auteurs utilisent HSOM dans le cadre du *clustering* et la classification de données : les SOM doivent automatiquement extraire des groupes de données (*clusters*) par similarité. Ces clusters sont définis après apprentissage en utilisant les cellules de Voronoi dont les poids de la carte $\omega(p)$ en sont les centres. Comme les cartes s'organisent de façon à conserver les distances dans l'espace d'entrée au sein de la carte, deux éléments faisant partie d'un même cluster auront des BMU proches dans la première carte ; par conséquent, leurs BMU dans la seconde carte le seront également. Ils notent que la SOM du premier niveau de l'architecture, qui est une SOM classique, générera de nombreux petits clusters, dont plusieurs d'entre eux seront nécessaires pour couvrir une classe entière. Au contraire, la deuxième couche de SOM génère des clusters plus larges, et moins de ces clusters seront alors nécessaires pour couvrir une classe de données. Le fait d'utiliser une architecture de SOM permet dans ce cas d'extraire une représentation différente de celle extraite par une SOM classique.

D'autres travaux par la suite implémentent des modèles similaires transmettant la position du BMU entre cartes, sur des architectures comportant plus de cartes que HSOM, tel que **hagenauer_hierarchical_2013** ; Paplinski et Gustafsson 2005. Dans ces travaux, les auteurs implémentent une architecture en arbre. La carte du niveau i , $M^{(i)}$ reçoit un vecteur de position de BMU de $k + 1$ cartes $M_0^{(i-1)}, \dots, M_k^{(i-1)}$ du niveau inférieur en tant qu'entrée :

$$X^{(i)} = [\Pi_0^{(i-1)}, \dots, \Pi_k^{(i-1)}]$$

L'architecture est dans ce cas utilisée pour l'agrégation de données dépendantes les unes des autres. La carte du niveau supérieur apparaît comme une représentation abstraite des données et de leurs dépendances.

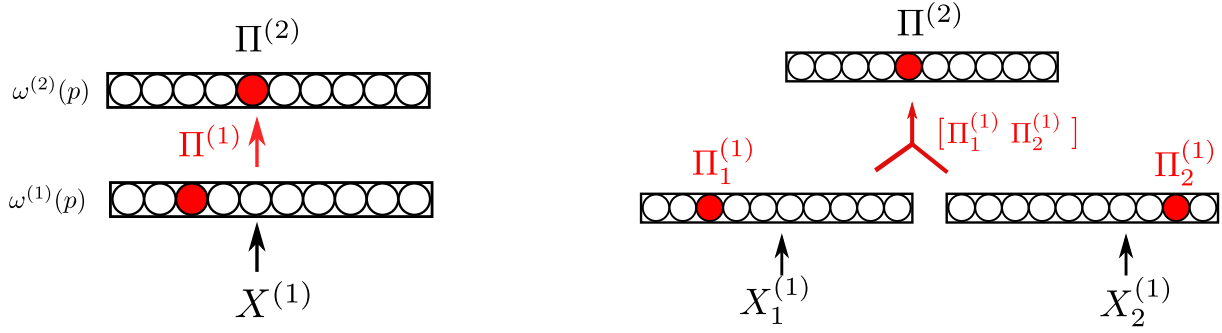


FIGURE 1.10 – Deux exemples d’architectures basées sur HSOM. À gauche, le modèle HSOM original proposé en [Lampinen et Oja 1992](#). L’apprentissage des positions du BMU de la première couche par la seconde permet de mieux détecter les ensembles de données présents dans la distribution des X (*clustering*). La deuxième couche est vue comme un niveau plus abstrait que la première. À droite, une version de HSOM comportant plus de cartes proposée en [Hagenauer et Helbich 2013](#) permettant de faire du clustering sur des entrées provenant de deux espaces $X_1^{(1)}$ et $X_2^{(1)}$. Ces deux espaces sont ici des caractéristiques spatiales et temporelles d’un environnement d’entrée.

Un point clé dans la construction de ces architectures repose sur l’interface entre les cartes. Cette interface doit permettre de véhiculer un maximum d’information entre cartes, tout en étant interprétable par les autres cartes. Les architectures HSOM et ses dérivées utilisent ainsi la position du BMU en tant que représentation. Par le choix de la position du BMU comme vecteur de transmission d’information, HSOM exploite effectivement la préservation de la topologie de l’espace d’entrée qu’offrent les cartes de Kohonen. Cette information est par ailleurs relative à une carte et non un type d’entrée. Enfin, il s’agit d’une position 1D ou 2D, donc une information peu coûteuse à utiliser. Nous avons également relevé le poids du BMU $\omega(\Pi)$ comme une méthode de transmission d’information au sein d’autres modèles de SOM hiérarchiques, comme [Wang et al. 2007](#) ; [Gunes Kayacik et al. 2007](#) ; [Dozono et al. 2016](#). Par exemple, [Dozono et al. 2016](#) décomposent une image d’entrée en imagerie qui sont utilisées en tant qu’entrées d’une première couche de cartes. Après apprentissage de cette couche, l’image est reconstruite grâce aux poids des BMU, puis décomposée en imagerie de tailles différentes pour être soumise à la deuxième couche de cartes. Enfin, les travaux de [Mici et al. 2018](#) s’appuient sur des cartes hiérarchiques pour effectuer de la fusion de données spatio-temporelles. Les auteurs et autrices de ces travaux utilisent comme sortie de la carte temporelle la série de poids des BMU successifs, relatifs à la séquence d’entrée, et comme sortie de la carte spatiale le poids du BMU relatif à l’entrée. L’entrée de la deuxième couche de cartes est alors un mélange entre les deux modalités. L’application de cette architecture mérite d’être soulignée, dans la mesure où elle permet d’associer information spatiale et temporelle. Par contre, le fait d’utiliser les poids du BMU comme interface fait perdre la représentation de chaque entrée apprise par les cartes du premier niveau dans leur topologie.

Le terme de *Deep SOM* est régulièrement rencontré lorsqu'on s'intéresse aux travaux récents portant sur les architectures de cartes auto-organisatrices. Aussi [Liu et al. 2015](#) ; [Dozono et al. 2016](#) ; [Hankins et al. 2018](#) ; [Mici et al. 2018](#) ; [Wickramasinghe et al. 2019](#) ; [Aly et Almotairi 2020](#) ; [Sakkari et Zaied 2020](#) ; [Nawaratne et al. 2020](#) et de nombreux autres travaux sont présentés comme tels. Les travaux se décrivant par ces termes implémentent des structures hiérarchiques puisant leur inspiration des réseaux de neurones profonds (*Deep Learning*), ayant notamment connu leur essor avec les réseaux convolutifs permettant l'apprentissage supervisé d'images (LeCun et al. 2015). Cependant, leur analogie avec les modèles de Deep Learning, qui s'appuient sur le principe de rétropropagation du gradient, s'arrête à la présence de couches de poids et leur application au traitement d'image. Dans leur structure, les modèles de Deep SOM restent bien proches des modèles de SOM et SOM hiérarchiques. Par analogie avec ces réseaux convolutifs, les réseaux présentés comme Deep SOM s'intéressent à l'apprentissage d'images.

Nous pouvons prendre comme exemple le modèle D-SOM introduit en [Liu et al. 2015](#) ; [Wickramasinghe et al. 2019](#), illustré en figure 1.11. Le but d'une telle architecture est de classifier des images X fournies en entrée de l'architecture. Une fenêtre est déplacée sur l'image d'entrée, créant un ensemble de $N \times N$ imagerie de position fixée. La première couche du réseau comporte $N \times N$ cartes, donc chacune prend en entrée l'imagerie de même position i, j . La sortie de la couche donne $N \times N$ positions de BMU $\Pi_{i,j}$. Ces positions représentées comme des valeurs en une dimension sont assemblées en une image intermédiaire, chaque pixel prenant la valeur du BMU de la carte correspondante.

$$X_{int} = \begin{pmatrix} \Pi_{0,0} & \cdots & \Pi_{0,N} \\ \cdots & \cdots & \cdots \\ \Pi_{N,0} & \cdots & \Pi_{N,N} \end{pmatrix}$$

Une deuxième couche de cartes de même structure que la première carte est alors appliquée à cette image intermédiaire. La dernière couche du réseau est composée d'une SOM simple effectuant la quantification vectorielle sur l'image de sortie de la couche précédente, vue comme une représentation abstraite de l'entrée. L'interface entre les couches de cartes est créée à partir des BMU des SOM : ce modèle de transmission rejoint ainsi ceux présentés dans HSOM. La différence apparaît au niveau du prétraitement de l'entrée image, décomposée en imagerie.

Ce type d'architecture utilise bien l'aspect topologique de la carte de Kohonen dans ses calculs, les interfaces entre couches de cartes s'appuyant sur les positions du BMU. L'utilisation des positions au lieu de poids permettent de réduire la dimension des images traitées par les couches successives. L'image de sortie de la dernière couche est alors de taille réduite. Lorsque la dernière SOM classifie les images générées par la dernière couche, elle classifie une représentation plus abstraite de l'image d'entrée. Les auteurs de cette architecture montrent que la classification de cette dernière image permet de bien retrouver les classes présentes dans les données d'entrées.

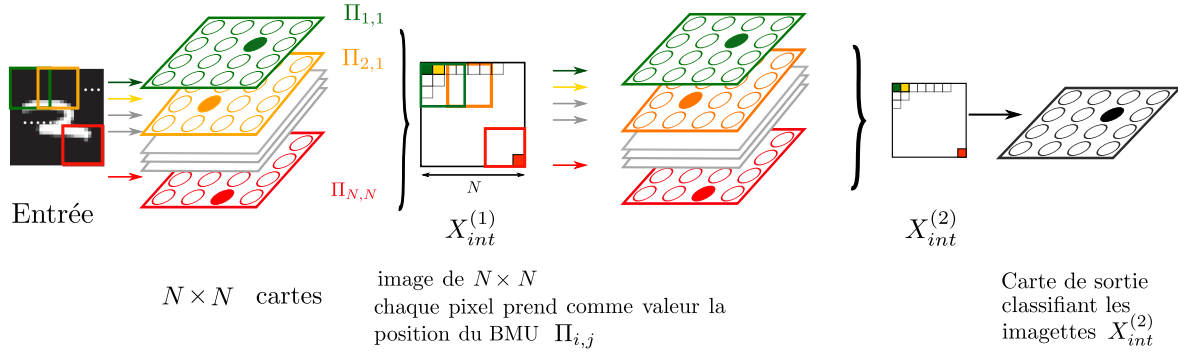


FIGURE 1.11 – Architecture DSOM de SOM « convolutive » (Liu et al. 2015). Les auteurs utilisent les positions des BMU $\Pi_{p,q}$ d’une couche de carte, disposés dans un tableau 2D, comme valeurs d’entrée pour les couches suivantes. Ces couches sont entraînées les unes après les autres.

Ils notent que l’architecture D-SOM possède une erreur de classification sur MNIST plus faible qu’une SOM simple, ce qui montre que l’abstraction générée par les couches successive renforce la séparation entre classes. Par l’assemblage des positions du BMU en tant que représentation intermédiaire de l’entrée, l’architecture D-SOM est très similaire à HSOM.

Toutes les architectures présentées ici, qu’il s’agisse des SOM hiérarchiques ou Deep SOM, comme pour les architectures sélectives, sont ascendantes dans leur mise à jour : chacune des couches de cartes est entraînée l’une après l’autre. Si les tâches de classifications réalisées par ces travaux auraient pu être réalisées avec une SOM classique, l’utilisation d’une architecture plutôt qu’une SOM vise à obtenir de meilleures performances en termes d’erreur de classification.

Discussion

Nous avons distingué deux catégories d’architectures hiérarchiques. Le premier type d’architecture repose sur la sélection d’une carte du niveau supérieur en s’appuyant sur la réponse des cartes d’un même niveau à une entrée, afin de transmettre cette entrée à la carte supérieure. Cette architecture permet de créer un ensemble de cartes s’organisant sur un même espace d’entrée, laissant plus de possibilités au dépliement des cartes qu’une SOM classique. Ces architectures permettent notamment d’améliorer la qualité de la quantification vectorielle générée par une SOM classique et d’ajouter des nœuds de façon peu coûteuse. Les architectures par sélection utilisent une surcouche algorithmique aux cartes, qui décompose successivement l’espace d’entrée en sous-espaces et distribue aux cartes leurs entrées. Ce type d’architecture n’est pas modulaire dans la mesure où la connexion entre carte est gérée par un processus global.

Le second type d’architecture repose sur la modification du principe de calcul d’activité et de mise à jour d’une carte pour prendre en compte les éléments de réponse d’une autre carte, par

exemple en ajoutant cet élément de réponse en tant qu'entrée secondaire d'une carte. Cette communication entre cartes est intégrée au processus d'auto-organisation, ce qui localise le traitement des connexions à l'échelle d'une carte. Cette méthode de construction d'architecture s'appuie sur la transmission d'une représentation de l'entrée interne entre cartes. Cette structure nous intéresse d'un point de vue modulaire : elle autorise l'ajout de modules à une architecture sans avoir à modifier toute la structure de l'architecture, ce qui est un des éléments de définition d'un système modulaire. Nous verrons plus loin que les architectures non hiérarchiques s'appuient également toutes sur la transmission d'éléments internes. Nous avons relevé au sein des architectures hiérarchiques deux représentations internes majoritairement utilisées comme information transmise entre cartes : la position du BMU ou le poids du BMU. La position du BMU est une représentation exploitant totalement l'aspect topologique d'une carte auto-organisatrice, de dimension faible et homogène à toutes les cartes. Il permet d'extraire une représentation abstraite de l'entrée, ce qui est le but recherché d'une architecture, par exemple dans le cas de la classification de données multimodales. En revanche, lorsque le poids du BMU est échangé, les calculs sont plus coûteux. On perd l'aspect générique et homogène de la transmission des positions des BMU, ce qui oblige à particulariser les calculs. Cet échange de poids n'exploite pas l'aspect topologique d'une carte de Kohonen, mais seulement sa capacité de quantification vectorielle.

Qu'elles soient sélectives ou par transmission de représentation, toutes les architectures relevées ici ont une séquence de mise à jour séquentielle : les cartes du premier niveau sont dépliées lors d'une première phase d'apprentissage. Une fois ces cartes dépliées, la deuxième couche est apprise à partir de la première lors d'une deuxième phase, lors de laquelle le premier niveau n'est plus mis à jour. Le champ d'application des architectures hiérarchiques est le même qu'une SOM classique : quantification vectorielle et classification. Dans les deux cas, il s'agit d'améliorer les performances sur une tâche de quantification vectorielle ou de classification pouvant être réalisée par une SOM. Par exemple, les SOM par sélection permettent d'améliorer la quantification vectorielle sur l'espace d'entrée ou de prendre en compte l'aspect hiérarchique des entrées. Les SOM par transmission de représentation permettent de mieux isoler les clusters de données qu'une SOM classique ou d'effectuer une classification sur des données multimodales. Nous pouvons donc considérer ces architectures comme des améliorations de cartes auto-organisatrices sur des applications spécifiques. Elles ont les mêmes types de réponses qu'une SOM simple. L'aspect uniquement ascendant en est la cause : les cartes de l'architecture agissent comme des filtres intermédiaires de l'information donnée en entrée, mais seule la couche finale est considérée en sortie de l'architecture : cette couche finale reste une carte auto-organisatrice classique, apprenant simplement sur des entrées filtrées. Dans une volonté d'étudier une architecture proposant des comportements de calcul différents de ceux réalisés dans une SOM classique, notre attention se portera sur les architectures comportant des boucles de rétroaction : les architectures non hiérarchiques. Ces architectures permettent de diversifier les comportements d'apprentissage qu'il est possible d'obtenir avec des SOM en apportant un aspect dynamique au système

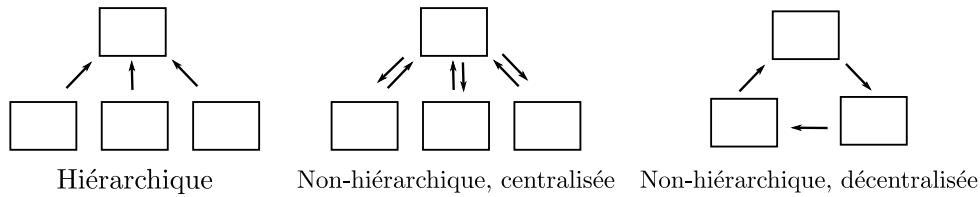


FIGURE 1.12 – Exemples de connexions dans des architectures hiérarchiques et non hiérarchiques centralisées et décentralisées. Un rectangle correspond à un module, ici une carte auto-organisatrice. Une flèche représente l’existence d’une interface entre deux modules : le module destination prend comme entrée contextuelle la sortie de la source et utilise donc de l’information de la source dans ses règles d’évolution.

par les rétroactions. Notons qu’une architecture hiérarchique est un cas particulier d’architecture non hiérarchique : les modèles que nous allons étudier en partie suivante pourraient donc aussi bien être utilisés dans un cadre hiérarchique.

1.4.3 Architectures non hiérarchiques de cartes auto-organisatrices

Les architectures non hiérarchiques de SOM sont des architectures comportant plusieurs cartes communiquant entre elles et dont le graphe de connexion comporte des boucles de rétroaction : une carte A reçoit de l’information d’une carte B, qui elle-même reçoit, directement ou indirectement, de l’information de la carte A.

Notons d’abord que les travaux cherchant à assembler des réseaux de neurones en architecture non hiérarchiques se revendiquent plutôt du domaine des neurosciences computationnelles ou de la robotique, tandis que les architectures hiérarchiques décrites précédemment se positionnaient dans un domaine d’apprentissage automatique, dans un objectif d’amélioration de la classification et quantification vectorielle d’une SOM. Les motivations se situent d’une part au niveau de l’inspiration biologique des modèles. Nous avons vu que le cortex présente des aires dont les activités sont corrélées, suggérant une relation entre les activations des différentes aires. Ces relations sont observées comme bidirectionnelles et intervenant à différents niveaux du traitement de l’information sensorielle. Les travaux de modélisation du cerveau cherchent donc à implémenter des architectures de réseaux de neurones non hiérarchiques. Nous ne détaillerons pas ici les travaux portant sur des modélisations fines du cerveau à base de modèles biologiques de neurones, nous intéressant seulement à ceux présentant des cartes auto-organisatrices. Nous avons par contre vu qu’une carte de Kohonen, sans être une modélisation fine d’une aire cérébrale, est une adaptation informatique d’un concept d’auto-organisation présent dans les aires sensorielles des réseaux de neurones biologiques. Plusieurs travaux de neurosciences computationnelles ont ainsi utilisé des cartes de Kohonen comme un modèle simplifié d’aire cérébrale pour les assembler en architecture.

L’aspect bio-inspiré se retrouve également dans les motivations des modèles robotiques. Ces

modèles se placent dans le paradigme d'embodiment (*Embodied Cognition*), c'est-à-dire le développement d'un système intelligent qui peut interagir avec son environnement. Parmi les éléments de recherche principaux de ce domaine robotique figurent la fusion de données multimodales, le traitement de séquences et l'apprentissage développemental, inspiré du comportement des humains et animaux. (Smith et Gasser 2005). Le recensement des architectures de cartes non hiérarchiques relève ainsi de plusieurs domaines, dans la mesure où ces modèles sont développés dans le contexte des neurosciences computationnelles ou de la robotique cognitive et cherchent à modéliser les aires cérébrales. Nous chercherons à faire une relecture de ces modèles d'un point de vue de la méthode de transmission d'information entre les cartes de ces structures, comme nous l'avons réalisé sur les architectures hiérarchiques.

Nous avons pu distinguer deux structures principales d'architectures non hiérarchiques dans les travaux répertoriés, illustrées en figure 1.12. Certaines architectures comportent des cartes sensorielles qui sont reliées via des cartes associatives ne prenant pas d'entrées sensorielles, mais seulement des éléments de connexion venant des autres cartes. Ces architectures sont *centralisées*, étant donné que les cartes associatives centralisent l'information montant des cartes sensorielles et la redistribuent. Ces architectures centralisées sont souvent désignées par leurs auteurs comme hiérarchiques : les cartes associatives forment effectivement un niveau d'apprentissage différent des cartes sensorielles, apportant une hiérarchie dans l'apprentissage. Néanmoins, nous les classons ici dans la catégorie non hiérarchique. En effet, bien que des niveaux de cartes peuvent être isolés dans ces architectures, les connexions entre les cartes de deux niveaux sont bidirectionnelles, la carte associative étant à l'origine de l'activation de cartes sensorielles, et réciproquement. Nous les différencions ainsi des cartes hiérarchiques uniquement ascendantes que nous avons listées au paragraphe précédent. Un second type d'architectures non hiérarchiques sont les architectures présentant des connexions directes entre cartes sensorielles. Ces architectures sont *décentralisées*, et il n'existe pas de module par lequel toute l'information transite.

La mémoire associative et l'apprentissage développemental comme applications des architectures non hiérarchiques

Les architectures non hiérarchiques proposées dans la littérature ont en commun leur application à la mémoire associative de données multimodales. La fusion de données multimodales est un enjeu actuel des algorithmes d'apprentissage en robotique développementale. Il s'agit d'intégrer les données issues de multiples capteurs au sein d'un même algorithme d'apprentissage. Il est en effet rare que l'information issue d'un seul capteur apporte toute l'information nécessaire à l'apprentissage et la prise de décision dans un environnement réel (Lahat et al. 2015).

Dans la mesure où la recherche en robotique cherche à complexifier les comportements possibles pour les agents et à s'inspirer de la biologie, la prise en compte de données de différentes sources est nécessaire. Ces données proviennent d'espace de différentes dimensions comme des

images, des capteurs audio, des capteurs tactiles, du texte, des actions. Leur temporalité peut varier : on veut pouvoir associer des données séquentielles, c'est-à-dire extraire de l'information d'une succession d'entrées, à des données instantanées dans lesquelles seule la valeur de l'entrée compte. La fréquence d'arrivée des données séquentielles varie également. L'enjeu de la fusion de données multimodales est alors de concilier tous ces aspects lors de l'apprentissage.

La mémoire associative se définit dans le cadre de la fusion de données multimodale par l'action de prise de décision sur une modalité relativement aux autres. Les autres modalités peuvent venir améliorer la prise de décision par rapport à la modalité seule. C'est par exemple le cas dans l'effet McGurk (McGurk et MacDonald 1976), lorsque la vision d'une bouche prononçant "ga" associée au son "ba" amène un sujet à indiquer avoir entendu "da" (voir section 1.3). Il est également montré que le fait de lire sur les lèvres en écoutant une personne améliore la compréhension du discours, par exemple dans un environnement bruyant. Il s'agit ici de mémoire associative entre modalités visuelles et auditives. Cette mémoire associative peut aussi s'utiliser pour prédire une modalité par rapport aux autres : les modalités visuelles et auditives vont générer une prise de décision au niveau de la modalité moteur d'un robot et ainsi générer une action par association.

Les architectures de cartes non hiérarchiques que nous avons relevées se positionnent dans un cadre de mémoire associative, que ce soit par une motivation bio-inspirée ou par leur but d'implémentation en robotique. Leur architecture modulaire apparaît comme un moyen de réaliser de la fusion de données à l'échelle de l'algorithme, par opposition à la fusion de données à l'échelle des entrées. Une modalité est alors traitée par un ensemble de cartes de l'architecture, et les autres cartes de l'architecture n'ont accès qu'à une information filtrée de cette entrée.

Notons que les cartes hiérarchiques apparaissaient déjà comme un moyen de traiter des données multimodales, par exemple en Mici et al. 2018 et Nawaratne et al. 2020. Une carte traite des données spatiales d'un côté, une autre des données temporelles ; l'architecture associe la sortie de ces cartes dans la couche finale pour classifier les motifs spatio-temporels. Les cartes du premier niveau étaient alors consacrées à la représentation d'une modalité, tandis que la dernière carte est une carte associative apprenant des motifs spatio-temporels liant les deux cartes modales. Les cartes non hiérarchiques vont plus loin dans l'application de la mémoire associative, car la présence de rétroactions permet de générer une activité au sein d'une carte modale par ses connexions aux autres cartes, même lorsque l'entrée est manquante. Une carte auto-organisatrice acquiert ainsi une capacité de prise de décision, par son activation alors que la carte hiérarchique permet seulement d'extraire une représentation. Cette prédiction de modalité est utilisée dans les différents travaux présentés dans cette section comme l'application principale de ce type d'architecture et les expériences de validation sont menées autour de la capacité d'une carte modale à prédire de façon précise la modalité à partir des connexions associatives. Notons enfin que la notion de mémoire associative s'étend à l'apprentissage de séquences : il s'agit alors d'extraire une

représentation d'une séquence temporelle complète ou de pouvoir compléter automatiquement une séquence.

Le concept d'apprentissage développemental est un autre enjeu de la robotique et s'intéresse à des systèmes étant mis à jour en ligne, dès qu'ils reçoivent une entrée, et dont l'apprentissage n'a pas de limite temporelle fixée. On doit donc avoir un système qui trouve de lui-même une stabilité dans l'apprentissage et qui est capable de s'adapter à de nouvelles entrées. Dans les applications de robotique, les entrées présentées à une structure d'apprentissage sont des entrées ayant une relation temporelle. Deux images reçues successivement par un capteur visuel seront proches dans l'espace des images. Pour une SOM classique par exemple, cela pose problème : le réseau s'organiserait d'abord sur le sous-espace composé des premières images de la séquence, puis évoluerait en même temps que les entrées en oubliant la séquence vue précédemment. Les architectures développementales cherchent donc une solution à ces problèmes pour créer une structure autonome, évoluant dans le temps et permettant de réaliser la tâche pour laquelle elle est conçue tout en continuant à être mise à jour, sans oublier catastrophiquement les données apprises au début de l'apprentissage. Ces enjeux applicatifs, communs aux architectures présentées dans cette partie, nous motivent également à étudier les cartes non hiérarchiques.

Architecture comportant une carte associative : architecture centralisée

L'idée d'assembler des cartes prenant en entrée une modalité sensorielle par une carte associative a été explorée en [Lallee et Dominey 2013](#) et [Escobar-Juárez et al. 2016](#). Dans ces deux travaux de neurosciences computationnelles, les auteurs construisent une architecture se voulant une modélisation de la théorie de la zone de convergence-divergence (Edelman [1982](#)) avec des cartes auto-organisatrices, en transmettant les positions des BMU entre les cartes multimodales.

Le modèle MMCM de [Lallee et Dominey 2013](#) propose une architecture composée de plusieurs cartes modales chacune associée à une modalité sensorielle et d'une carte associative prenant en entrée les positions des BMU des cartes modales. Cette architecture est représentée en figure [1.13](#). Nous définissons cette architecture comme non hiérarchique, car il existe des rétroactions entre les cartes modales $M^{(1)}$, $M^{(2)}$ et la carte associative $M^{(m)}$. Dans l'exemple d'une architecture à deux cartes modales, l'une reçoit les mouvements de tête d'un robot et une autre les mouvements du bras. Chaque carte du premier niveau possède une couche de poids ω_e liées aux entrées sensorielle ainsi qu'une couche de poids ω_c dédiée aux connexions descendantes, prenant en entrée les positions du BMU de la carte associative. La carte associative prend deux couches de poids, chaque couche correspondant à la position du BMU d'une carte sensorielle.

La mise à jour est réalisée en trois étapes : D'abord, les couches de poids externes des cartes modales sont mises à jour indépendamment sur les entrées. La recherche du BMU est réalisée en prenant en compte une activation $a(p, \omega, X^{(i)})$ dans la carte. Ensuite, les poids ω_e

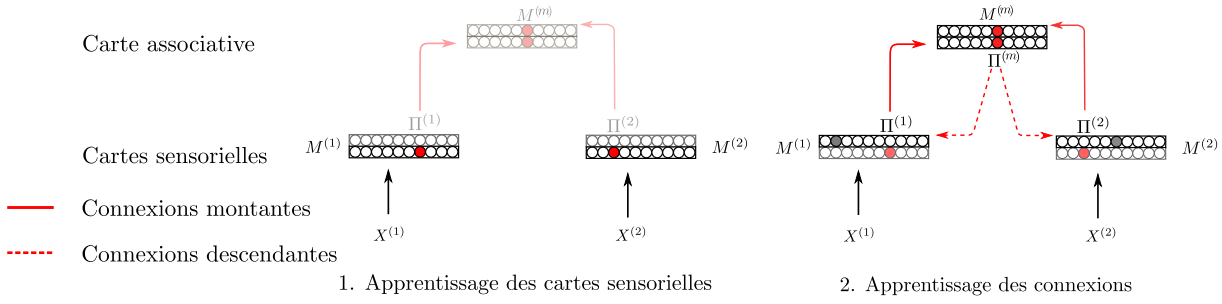


FIGURE 1.13 – L’architecture MMCM (Lallee et Dominey 2013) est une architecture centralisée. Les cartes du premier niveau sont les cartes modales $M^{(1)}, M^{(2)}$ qui reçoivent l’une les mouvements de tête d’un robot, une autre les mouvements de son bras. Une carte associative $M^{(m)}$ reçoit les positions des BMU $\Pi^{(1)}, \Pi^{(2)}$ de chaque carte du premier niveau en tant qu’entrées. Les cartes modales ont ensuite une couche de poids encodant les positions des BMU de la carte associative et permettant leur activation depuis la carte associative.

sont ensuite gelés, et les poids de la carte associative sont mis à jour de façon à apprendre à associer les positions des BMU $(\Pi^{(1)}, \Pi^{(2)})$ correspondant aux cartes modales, rappelant les modèles hiérarchiques HSOM. La carte prend en entrée $X^{(m)} = [\Pi^{(1)}, \Pi^{(2)}]$

Dans un troisième temps, les poids de la carte associative sont figés et les couches de chaque carte modales ω_c dédiées aux connexions sont mises à jour, l’entrée étant le BMU de la carte associative $\Pi^{(m)}$.

$$\begin{cases} \Pi_c^{(1)} = \arg \max_p a(p, \Pi^{(m)}, \omega_c^{(1)}) \\ \Pi_c^{(2)} = \arg \max_p a(p, \Pi^{(m)}, \omega_c^{(2)}) \end{cases}$$

Une carte modale a donc à la fois un BMU relatif aux activités externes et un BMU Π_c relatif aux activités contextuelles pendant l’apprentissage, les deux couches de poids étant décorréées. La rétroaction entre les cartes est en fait découplée lors de la phase d’apprentissage, car la carte associative dépend seulement de la couche externe des cartes sensorielles et transmet des informations seulement à la couche contextuelle des cartes sensorielles. Après ces trois phases d’apprentissage, les entrées modales ne sont pas présentées aux cartes modales. L’activation manuelle d’un neurone de position $p^{(m)}$ de la carte associative entraîne une activité et un BMU dans les deux cartes modales grâce au calcul de l’activation sur la couche de poids contextuelle :

$$\Pi_c^{(1)} = \arg \max_p a(p^{(m)}, \omega_c^{(1)}(p))$$

La valeur $\omega_c^{(1)}(\Pi_c^{(1)})$ est alors une prédiction de la modalité 1. Les auteurs montrent que cette méthode d’activation produit des mouvements coordonnés entre modalités. De la même façon, l’activation d’un neurone d’une carte sensorielle entraîne également une activation coordonnée dans les autres cartes sensorielles en passant par la carte associative. Notons que les cartes

utilisées dans ces travaux sont des cartes 3D.

L'architecture SOIMA (Escobar-Juárez et al. 2016) associe également plusieurs cartes modales avec une carte associative, présentée en figure 1.14. La transmission d'information des cartes modales vers la carte associative est réalisée par la transmission de la position du BMU : la carte associative prend en entrée $(\Pi^{(1)}, \Pi^{(2)})$, le couple de BMU des cartes modales. Afin de gérer les rétroactions, les auteurs ajoutent en tant que connexions descendantes des connexions pondérées neurone à neurone mises à jour par une règle de transmission Hebbienne : le poids de la connexion est renforcé si les deux neurones reliés s'activent lors de la même itération. Les connexions montantes et descendantes sont ici encodées de manière différente ; cela permet aux auteurs d'effectuer la mise à jour des cartes et de leurs connexions en une seule étape. Dans ces travaux, les auteurs associent deux modalités sensorielles et motrices par une carte associative en trois dimensions. L'utilisation de connexions hebbiennes pondérées entre neurones est équivalente à transmettre l'entièreté de l'activation de la carte associative à une carte sensorielle. Prenons l'exemple de cartes 1D. Chaque neurone j de la carte modale reçoit un signal a_i de chacun des neurones i de la carte associative par une connexion de poids ω_{ij} . Tous les neurones de la carte modale reçoivent donc le même ensemble d'entrées $\{a_i, i = 0..N\}$. Les poids des neurones ω_i sont ainsi équivalents à l'ajout d'une couche de poids supplémentaire à la carte modale telle que : $\forall j, \omega_j \in [0, N]$ apprend un champ d'activation de la carte associative.

L'information transmise entre cartes dans l'architecture SOIMA repose sur la position du BMU pour les connexions montant des cartes sensorielles à la carte associative, et sur des champs d'activité neuronales pour les connexions descendantes. La gestion des rétroactions est réalisée de la même façon que pour MMCM : les couches de poids des cartes modales étant décorréliées lors de l'apprentissage, les rétroactions n'ont pas d'influence sur la mise à jour. Elles sont utilisées seulement en phase d'application.

Les modèles mentionnés ci-dessus entrent dans la catégorie non hiérarchique pour leur possibilité d'activation d'une carte par l'autre. La position du BMU apparaît dans les modèles SOIMA et MMCM comme le vecteur de transmission d'information entre cartes. Le modèle SOIMA privilégie la connexion neurone à neurone entre la carte associative et la carte modale. La présence de cartes associatives au sein d'une architecture crée une centralisation de l'information multimodale sur une carte, ce qui nous amène à parler d'apprentissage centralisé. Chaque carte sensorielle ne reçoit aucune information directe d'autres cartes de l'architecture, sauf de la carte associative. Les cartes modales et associatives jouent ainsi un rôle différent dans les calculs.

La présence de rétroactions soulève une problématique de conception supplémentaire dans les cartes non hiérarchiques : l'activité de la carte A influence l'activité de la carte B, mais l'activité de la carte B influence également celle de la carte A, etc. Pour résoudre ce problème, l'architecture MCM et l'architecture SOIMA décorrèlent les couches de poids prenant en compte l'entrée modale et celles relatives à l'entrée descendant de la carte associative lors de l'apprentissage.

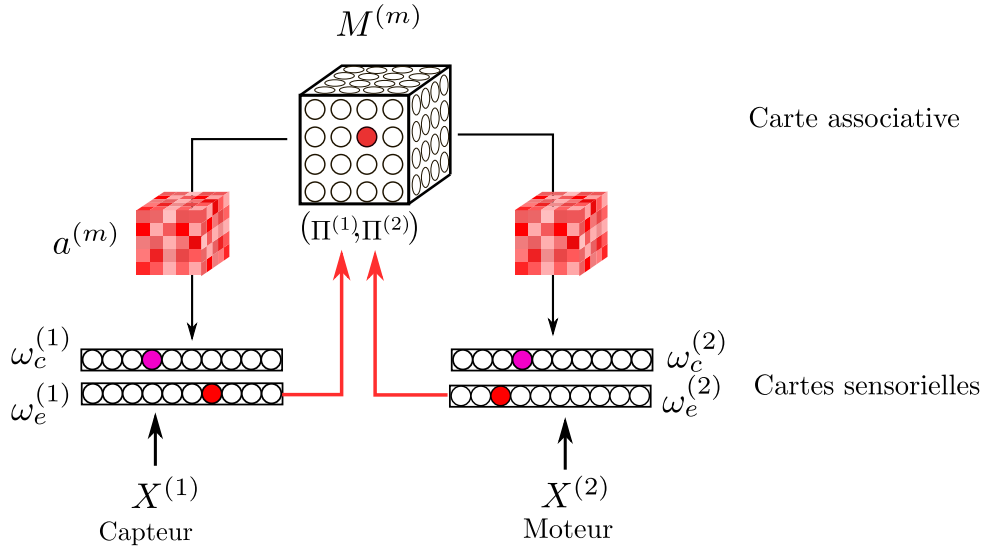


FIGURE 1.14 – Le modèle SOIMA (Escobar-Juárez et al. 2016) propose une architecture centralisée dans laquelle des cartes modales $M^{(1)}$ et $M^{(2)}$ sont connectées par une carte associative $M^{(a)}$. Cette carte associative prend comme entrée les BMU des cartes modales. Les connexions descendantes sont gérées par la transmission du champ d'activation de $M^{(m)}$ vers les cartes modales. Les poids de ces connexions sont stockés dans une deuxième couche de poids.

L'entrée de la carte associative est seulement le BMU de la couche de poids externes des cartes modales. La couche de poids contextuelle des cartes modales a son propre BMU pour la mise à jour. Par ailleurs, les auteurs de ces travaux décomposent l'apprentissage en plusieurs étapes : les cartes modales sont apprises, puis la carte associative, puis les connexions descendantes. La mise à jour est donc séquentielle.

Ces modèles sont des architectures modulaires. Toutes les cartes d'une architecture ont une structure similaire. Cependant, elles prennent des rôles conceptuellement différents par leur position dans l'architecture : certaines cartes sont associatives et d'autres cartes sont modales.

Architectures non hiérarchiques décentralisées

Une architecture non hiérarchique décentralisée est une architecture présentant des rétroactions entre cartes et dont les cartes modales présentent des connexions directes entre elles. Les modèles d'architectures décentralisées sont les plus génériques dans la mesure où ils n'imposent pas de structure spécifique pour l'architecture. La structure des connexions entre cartes devient alors un paramètre sur lequel on peut complètement agir, contrairement aux architectures centralisées. Ces modèles apparaissent comme des architectures modulaires idéales, car aucun a priori n'est associé aux modules, même une fois connectés. Leur spécialisation intervient uniquement grâce à leurs règles d'évolution internes.

Les auteurs de Khacef et al. 2020 utilisent par exemple deux cartes de Kohonen associées

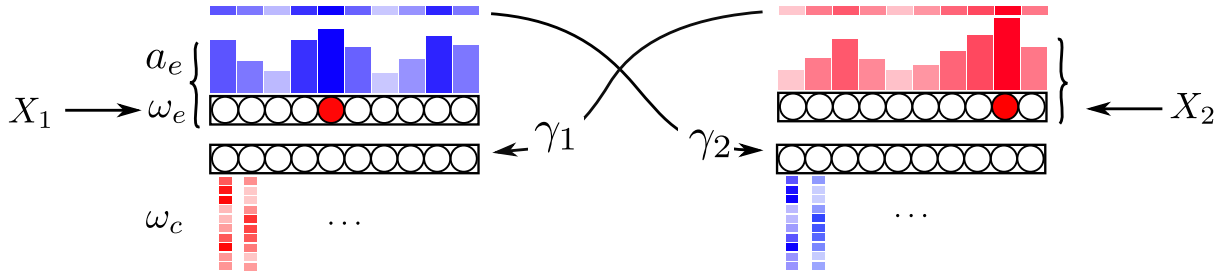


FIGURE 1.15 – Le modèle A-SOM (Johnsson, Balkenius et Hesslow 2009) associe les activités de différentes cartes. Chaque carte prend une entrée modale X_1 ou X_2 . Chacune des cartes possède deux couches de poids, une couche ω_e associée aux entrées modales et une couche ω_c associées aux entrées γ venant de l'autre carte. Lors de l'apprentissage, le calcul des activités sur chaque couche de poids est déconnecté, ce qui permet de gérer les rétroactions. Après apprentissage, une des entrées est supprimée. L'activation de la carte correspondante est alors permise par les connexions contextuelles, amenant la carte à prédire une entrée. Les cartes sont représentées en version 1D pour plus de clarté, mais le modèle utilise des cartes 2D.

par des connexions tous à tous entre neurones. Une carte prend en entrées des images MNIST, et l'autre le son du chiffre prononcé. L'apprentissage des deux cartes modales est réalisé dans un premier temps, puis les connexions entre neurones sont mises à jour dans une seconde étape à partir des mêmes paires d'entrées image-son. Les neurones de chaque carte s'activant sur une même paire d'entrées voient le poids de leur connexion se renforcer, et inversement. Les auteurs utilisent ici la transmission d'un champ d'activité neuronale comme vecteur de communication entre cartes (nous avons vu en effet que la connexion neurone à neurone revenait à transmettre un champ d'activation). Après apprentissage, la présentation d'une image à la carte associée permet de générer une activité cohérente dans la carte associée au son. Le modèle a donc ainsi appris les relations existant entre les deux modalités et est capable de générer une prédiction dans une carte à partir de l'autre. Un modèle similaire d'architecture non hiérarchique de deux cartes par transmission d'activité neuronale est également proposé en Jayaratne et al. 2018, les auteurs utilisant cette fois des SOM incrémentales au lieu de SOM à taille fixe. Dans ces deux modèles, la mise à jour des modules est séquentielle.

Une autre version d'architecture de cartes non hiérarchiques est développée en Johnsson et Balkenius 2008 ; Johnsson, Balkenius et Hesslow 2009, sous le nom de A-SOM, *associative self-organizing map*. La particularité de A-SOM, par rapport à tous les modèles précédemment étudiés est que l'apprentissage des cartes et de leurs interactions est réalisé simultanément et non séquentiellement. Il s'agit d'une mise à jour synchrone : on peut définir une itération globale à toute l'architecture pendant laquelle toutes les cartes seront mises à jour une fois. Ce modèle décentralisé inclut aussi la possibilité de créer une version d'architecture centralisée à partir des mêmes règles d'associations, construite par exemple en Buonamente et al. 2016. A-SOM est illustré en figure 1.15 pour l'exemple de deux cartes associées. Dans ce modèle, chaque SOM

reçoit une entrée X provenant d'une modalité, telle que la texture et l'image d'un objet. Une carte possède alors deux couches de poids : l'une est relative aux entrées externes X et l'autre relative à l'entrée contextuelle provenant de l'autre carte, γ , qui est ici un champ d'activation. Sur ces entrées, les auteurs calculent une activité par couche de poids : a_e et a_c . L'entrée γ correspond au vecteur des activations externes a_e des neurones de l'autre carte. Cette interface par transmission d'activation comme entrée d'une carte est équivalente à des connexions pondérées par des poids ω_{cij} reliant le neurone i d'une carte au neurone j de l'autre. On a alors $w_c(i) = [\omega_{ci1}, \dots, \omega_{ciN}]$. Lors de l'apprentissage, la mise à jour des poids ω_e et ω_c est réalisée de manière indépendante. Le BMU de position Π se situe au maximum de l'activité externe et les poids ω_e sont mis à jour comme dans une SOM classique. Les poids ω_c sont mis à jour en fonction de la différence entre activités externes et contextuelles à la position p :

$$\omega_c(p) \leftarrow \omega_c(p) + \beta \times \gamma_t \times (a_e(p) - a_c(p))$$

Cette règle de mise à jour permet de renforcer le schéma d'activation γ_t appris par un neurone seulement lorsque son activité externe est forte, et de réduire son impact si le neurone a une activité externe faible. Elle équivaut à la règle Hebbienne qui renforce les connexions de deux neurones s'activant en même temps, mais calculée à l'échelle d'une carte. Pendant l'apprentissage, le calcul d'activité est indépendant sur chaque couche de poids, seule la mise à jour insère une dépendance entre les deux couches. Après apprentissage, il est possible de supprimer les entrées externes d'une des cartes, mais de toujours pouvoir l'activer grâce à la seconde. Le BMU d'une telle carte est alors calculé comme le maximum de l'activité contextuelle a_c . Cette activation permet alors de générer des prédictions entre modalités. Le modèle A-SOM est ainsi un modèle d'architecture décentralisée, par transmission d'un champ d'activation. Comme dans les modèles centralisés SOIMA et HCMM, les rétroactions sont prises en charge en découplant les BMU relatifs à a_e et a_c .

Ensuite, les travaux menés précédemment dans notre équipe se sont attachés à la création d'architectures décentralisées de cartes auto-organisatrices. Ainsi, l'architecture Bijama développée en [Ménard et Frezza-Buet 2005](#) et l'architecture SOMMA développée en [Lefort et al. 2011](#) proposent des modèles d'architectures modulaires s'appuyant sur l'association des activités neuronales de cartes cellulaires. Les auteurs se sont appuyés sur des modèles de cartes auto-organisatrices cellulaires, les opérations s'effectuant ainsi à l'échelle même d'un nœud d'une carte. Dans le modèle Bijama, un neurone d'une carte est composé de plusieurs étages d'activation comportant chacun un poids, rappelant les colonnes corticales. Ces activations contribuent au calcul d'une activation globale du neurone. Un poids dit thalamique est relatif à l'entrée sensorielle du neurone, c'est-à-dire l'entrée X . Des poids corticaux sont quant à eux relatifs aux connexions venant des neurones des autres cartes de l'architecture.

Dans ce modèle présenté en figure [1.16](#), chaque neurone d'une carte est connecté via sa couche

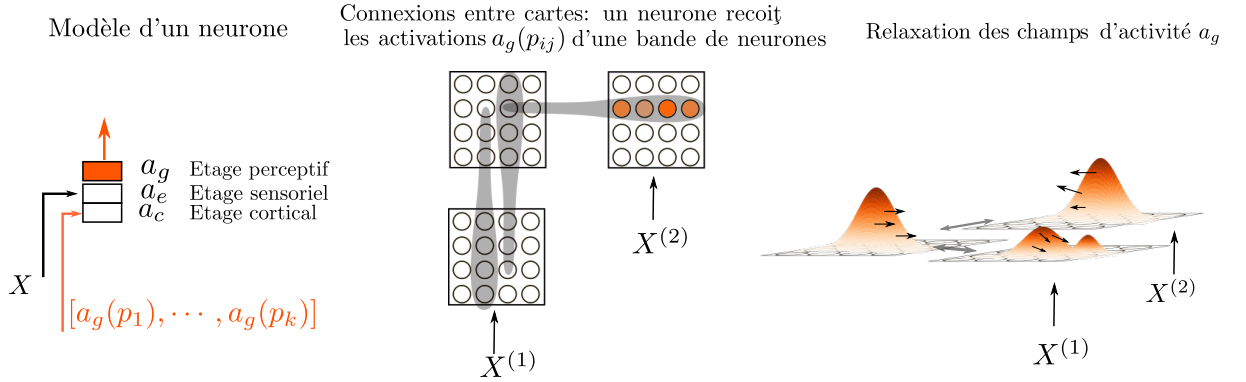


FIGURE 1.16 – Exemple d'architecture construite avec le modèle Bijama (Ménard et Frezza-Buet 2005). Chaque neurone d'une carte i reçoit une entrée sensorielle $X^{(i)}$ et des entrées corticales, indiquées figure de gauche. Dans cet exemple, deux cartes reçoivent des entrées sensorielles et corticales, et la troisième reçoit seulement des entrées corticales. Les activités corticales correspondent aux activités globales $a_g(p)$ des neurones situées dans une bande de même position dans la carte voisine, en gris dans la figure centrale. L'activité globale est une moyenne géométrique des activités sensorielles et corticales d'un neurone. Trois connexions sont représentées sur le schéma, mais tous les neurones reçoivent des connexions. La rétroaction entre les activités neuronales induit un phénomène dynamique de relaxation au sein de l'architecture, au cours duquel les entrées sensorielles restent inchangées et les activités globales des neurones évoluent vers un état limite stable.

corticale à une rangée de neurones d'une carte modale. Cela revient donc à la transmission de champs d'activation entre cartes. La mise à jour est asynchrone : les connexions directes entre neurones génèrent la mise à jour des poids du neurone. Contrairement aux modèles précédemment présentés, l'activité prise en compte lors des connexions neuronales est l'activité globale du neurone. Le calcul de l'activité du neurone de la première carte dépend de l'activation des neurones de la deuxième, qui dépendent de l'activité de la première. Les activités étant calculées de façon asynchrone, la présentation d'une entrée induit un processus dynamique de calcul d'activité au sein des neurones, au cours duquel les entrées thalamiques restent inchangées et les activités des neurones évoluent vers un état limite stable. Cette dynamique est appelée relaxation. Cette activité finale est celle prise en compte par les neurones pour la mise à jour de leurs prototypes.

L'architecture SOMMA implémente également une architecture décentralisée pour de l'apprentissage multimodal (figure 1.17). L'information transmise dans ce cas est une partie de l'activité des neurones, comme en Bijama. Comme dans ce modèle, les neurones sont structurés en étages, et l'étage cortical du neurone reçoit les activations des neurones situés dans une partie d'une autre carte. Il s'agit ici de l'activité située dans un carré centré à la même position que le neurone courant, au lieu des bandes de Bijama. L'information transmise entre cartes est ainsi également un champ d'activation. Les auteurs utilisent ici comme interface un champ d'activité réduit à une zone de la carte. Comme pour le modèle Bijama, SOMMA prend en compte les

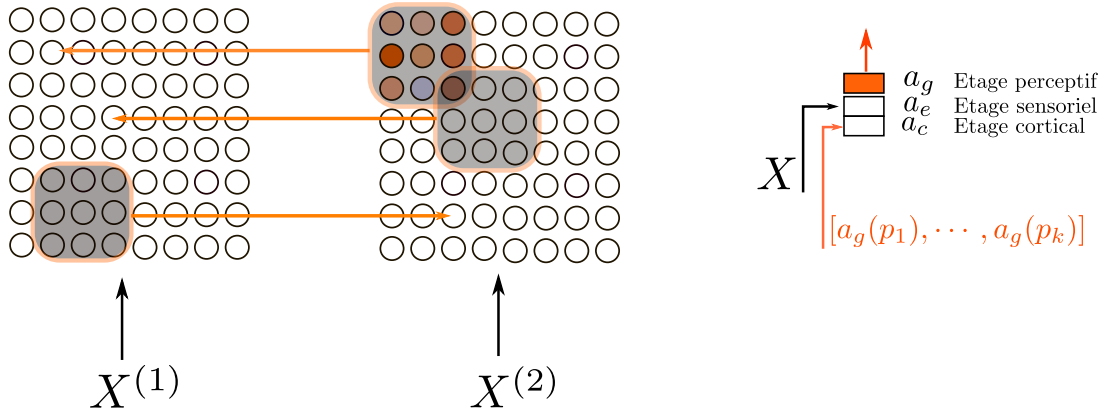


FIGURE 1.17 – Le modèle SOMMA (Lefort et al. 2011 ; Lefort 2012) associe les activités de différentes cartes, mais en réduisant les champs d'activité transmis aux neurones entourant le neurone situé en position courante. À gauche sont représentées les connexions entre les neurones de deux cartes, dans l'architecture SOMMA. À droite, le schéma d'un neurone d'une carte, comportant un étage sensoriel, cortical et global. Comme dans l'architecture Bijama, les rétroactions sont gérées par la dynamique de relaxation, laissant les champs d'activité évoluer vers un état stable.

rétroactions dans le calcul d'activité de chaque carte, utilisant le même mécanisme de relaxation.

Enfin, l'architecture proposée en Baheux et al. 2014 cherche à transformer le modèle cellulaire de Bijama en s'appuyant sur des cartes auto-organisatrices classiques et l'applique au traitement de séquences. Cette architecture, décrite en figure 1.18, est composée de deux cartes. Chacune des cartes est composée de deux couches de poids ω_e et ω_c . Une des cartes prend une entrée X_t correspondant à l'observation courante et relative à la couche de poids ω_e , comme une SOM classique. La deuxième couche de poids est relative à l'information contextuelle descendant de la seconde carte, qui est sa position du BMU $\Pi^{(1)}$. La seconde carte reçoit également deux entrées : l'entrée externe est la position du BMU $\Pi^{(1)}(t-1)$ de l'état précédent et l'entrée contextuelle la position du BMU $\Pi^{(1)}(t)$ de l'état courant. Chaque carte a ainsi des activations externes et contextuelles : $a_e^{(1)} = a_e^{(1)}(p, \omega_e^{(1)}, X)$, $a_c^{(1)} = a_c^{(1)}(p, \omega_c^{(1)}, \Pi_t^{(2)})$, et $a_e^{(2)} = a_e^{(2)}(p, \omega_e^{(2)}, \Pi_{t-1}^{(1)})$, $a_c^{(2)}(p, \omega_c^{(2)}, \Pi_t^{(1)})$.

Ces activations sont combinées dans chaque carte pour former une unique activité permettant de trouver le BMU :

$$a^{(i)} = \sqrt{a_e^{(i)}(\beta a_e^{(i)} + (1 - \beta)a_c^{(i)})}$$

Comme chaque carte reçoit en entrée la position de l'état courant du BMU de l'autre carte dépendant des boucles de rétroactions, le modèle laisse relaxer les activités en déplaçant petit à petit les BMU de chaque carte, jusqu'à obtenir un état stable pour les activités. Cette relaxation est un parallèle dans une carte auto-organisatrice classique à la relaxation proposée en Bijama. Cette position stable est utilisée pour déterminer le BMU final servant à la mise à jour des

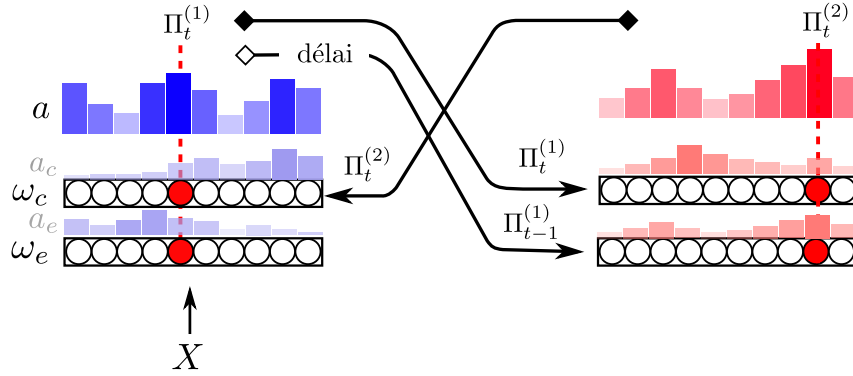


FIGURE 1.18 – Structures de deux cartes auto-organisatrices communicantes, (Baheux et al. 2014). Chaque carte est composée de trois couches d’activités, représentées séparément sur le schéma : sur la première carte, une activité est relative à l’entrée X , l’observation. L’autre activité reçoit une entrée descendant de la seconde carte. Ces deux activités sont fusionnées en une activité globale servant à déterminer un BMU. La seconde carte reçoit ensuite deux entrées venant de la première carte : le BMU de l’état courant et le BMU de l’état précédent. Un système de résonance est mis en place pour gérer les boucles de rétroactions entre BMU, comme chaque carte reçoit le BMU de l’état courant de l’autre carte en entrée. Ce principe laisse évoluer dynamiquement les activités vers un état stable, utilisé ensuite pour la détermination du BMU final.

poids. Ce modèle permet alors d’apprendre des séquences d’entrée. Alors qu’une carte simple différencierait les BMU en fonction de la valeur de l’entrée, ce modèle génère une différenciation des BMU en fonction de la position d’un élément dans la séquence.

Discussion

Les architectures non hiérarchiques de cartes font donc apparaître deux grandes catégories : les architectures centralisées, dans lesquelles une carte associative apprend à associer les activités de cartes sensorielles, et les architectures décentralisées. Les architectures décentralisées apparaissent comme la version la moins contrainte d’une architecture modulaire : lorsque les modules sont assemblés en architecture, aucun rôle ne leur est attribué a priori, contrairement aux architectures centralisées faisant apparaître des cartes associatives sans entrées externes et des cartes modales. Dans ce cas, les interfaces de connexions entre cartes sont de plus différenciés.

Nous avons relevé que la gestion des rétroactions entre cartes est un enjeu clé de la construction d’architectures décentralisées. Une première approche est de déconnecter, au sein d’une carte, l’apprentissage des poids liés aux entrées externes et des poids liés aux entrées contextuelles. Une carte a donc plusieurs BMU, relatifs à une ou plusieurs couches de poids, et la mise à jour s’effectue couche par couche. Cette approche est celle privilégiée dans la plupart des travaux que nous avons relevés. Les travaux menés dans notre équipe ont gardé la structure en couches, mais ont fait le choix de ne pas séparer les activités des couches lors de l’apprentissage

et de chercher un BMU commun. La gestion des rétroactions est réalisée en introduisant un mécanisme de relaxation dans le calcul des activités. Dans les modèles cellulaires, l'architecture laisse les champs neuronaux dynamiques couplés évoluer, en tant que système dynamique, vers un état stable. Cette relaxation se traduit dans la version non-cellulaire par une boucle imbriquée au sein d'une itération, calculant le BMU par petits déplacements dans chaque carte, jusqu'à atteindre une position stable. Nous choisissons dans cette thèse de privilégier cette seconde approche, introduisant un aspect de recherche dynamique du BMU dans les cartes. Un module n'a alors pas besoin de connaître des éléments de structure des autres modules : il ne perçoit pas de différence entre un autre module qui aurait des entrées externes et un module qui n'aurait que des entrées contextuelles.

1.4.4 Apprentissage de séquences et architectures de cartes auto-organisatrices

La capacité de traitement de séquences est une autre problématique d'utilisation des architectures de cartes. Nous avons d'ailleurs relevé des architectures permettant d'unir données spatiales et temporelles en un seul algorithme d'apprentissage. L'objectif d'un algorithme implémentant l'apprentissage de séquence est alors soit de prédire l'élément suivant d'une séquence de données, soit d'extraire des motifs temporels ou spatio-temporels se répétant dans les séquences de données d'entrée. La figure 1.19 illustre par exemple ce qu'on attend de l'apprentissage de séquences d'images d'un sportif : en s'appuyant sur la succession des images présentées à l'algorithme, le but est d'extraire des catégories de mouvements comme « tirer » ou « marcher », ce qui correspond à de la classification des séquences, ou de pouvoir compléter la vidéo en prédisant l'image suivante dans la séquence. Ainsi, la création d'architectures pour le traitement de données multimodales se rapproche de la question du traitement de séquences. Rappelons également que l'enjeu de la multimodalité en robotique développementale inclue le traitement de données séquentielles.

Cette similarité entre données multimodales et séquentielles n'est pas seulement présente au niveau des objectifs d'application des architectures de cartes auto-organisatrices, mais bien dans la structure même du traitement des données. Une solution pour faire de l'apprentissage de séquences peut être de fournir en entrée d'un réseau non plus une donnée instantanée mais une suite de données, sous forme par exemple de fenêtre temporelle ou d'une représentation de la séquence. Une autre solution est de prendre en compte l'état du réseau à l'instant précédent pour effectuer la mise à jour du réseau à l'état courant. Cette solution, implémentée dans de nombreux modèles d'apprentissage, se rapproche de la notion de transmission d'information entre modules d'une architecture dans laquelle les modules seraient la même carte, mais à deux instants. Ces réseaux prenant en compte leur instant précédent pour calculer leur état actuel sont appelés réseaux récurrents ou récursifs. Plusieurs modèles de cartes auto-organisatrices récurrentes, destinés à l'apprentissage de séquences, ont ainsi été proposés dans la littérature.

L'analyse des cartes récurrentes apparaît à la fois comme un enjeu de création d'une archi-

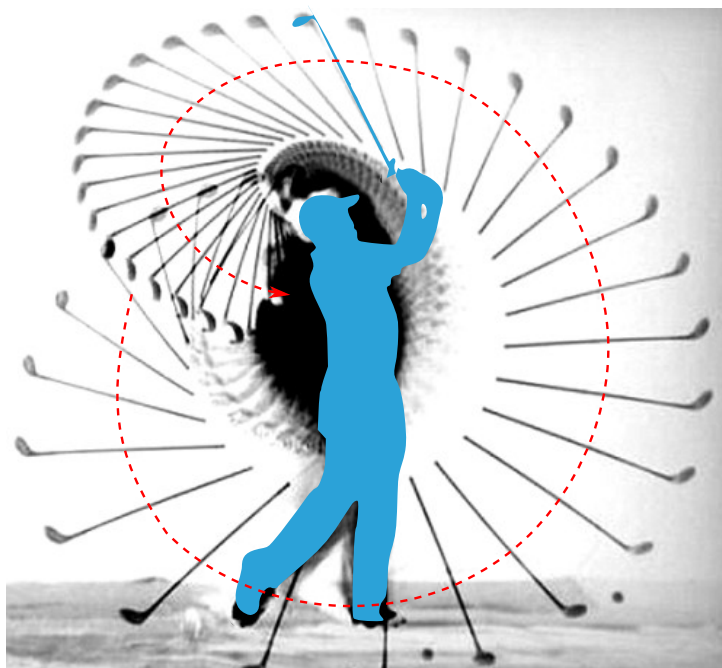


FIGURE 1.19 – L'image présentée à un réseau (en bleu) correspond à un instant d'une séquence. L'objectif de l'apprentissage non supervisé de séquences est d'extraire une représentation d'une séquence d'entrée. Une utilisation est par exemple la classification de mouvements. La séquence « tirer » sera différente de la séquence « marcher ».

architecture générale de cartes auto-organisatrices dans la mesure où il s'agit de créer un modèle qui permet d'associer des modules et se laisse la possibilité d'y intégrer des connexions récurrentes au même titre qu'une connexion intercartes, et comme une source supplémentaire sur laquelle s'appuyer pour catégoriser les modes de transmission d'information entre cartes. Dans cette section, nous passons en revue différents modèles de cartes récurrentes et nous intéresserons aux modèles multi-cartes implémentant des connexions récurrentes au même titre que des connexions inter-cartes.

Cartes auto-organisatrices récurrentes

Les modèles de cartes récurrentes existant dans la littérature s'appuient sur la transmission de représentation interne entre deux itérations. L'information transmise entre ces itérations rejoint celles relevées dans les architectures multi-cartes : transmission de la position du BMU, transmission d'une activité, transmission du poids du BMU.

Parmi les premiers travaux autour des cartes auto-organisatrices, les cartes de Kohonen Temporelles TKM, dérivées ensuite en *recurrent SOM* (Varsta et al. 2001) utilisent l'activité a d'une carte à l'instant précédent dans le calcul de l'activité à l'instant courant par un modèle d'intégrateur à fuite.

$$a(p, t) = (1 - \alpha)a(p, t - 1) + \alpha \times (X_t - \omega_t(p))$$

Avec $\omega(p)$ les poids de la carte et $\alpha \in [0, 1]$ un coefficient de fuite. a correspond ici à une distance modifiée entre un prototype et l'entrée, et le BMU est alors calculé par $\Pi_t = \arg \min_p (a(p, t)^2)$. Au lieu de chercher la position dont le prototype $\omega(p)$ est le plus proche de l'entrée X_t , Le calcul de l'activité revient ici à chercher une position pour laquelle la distance entre le prototype et l'entrée est faible et pour laquelle la distance calculée par rapport aux entrées précédentes l'était aussi. Il s'agit d'un intégrateur à fuite dans la mesure ou l'influence de l'activité d'un instant sur les suivants décroît au cours du temps.

D'autres travaux reposent plutôt sur la transmission du BMU ou du poids du BMU comme entrée courante d'une carte. À chaque instant t , ces SOM fusionnent deux entrées : l'entrée venant de la séquence à apprendre, X_t et l'entrée de contexte γ_t interne à la carte. Ainsi, les *recursive SOM* de Voegtlin 2002 utilisent en tant qu'entrée de contexte un vecteur contenant l'ensemble des activations des neurones de la carte à l'état précédent $\gamma_t = [a(t - 1, p), p \in [0, N]^k]$ avec $k = 1, 2$ la dimension de la carte. Les travaux de Buonamente et al. 2013 proposent une version récurrente du modèle A-SOM présenté en section précédente. Le contexte considéré est alors également un ensemble d'activités de neurones. MSOM, proposée en Strickert et Hammer 2005 s'appuie sur le poids du BMU. À chaque instant, l'entrée de contexte à transmettre à l'état suivant est définie comme une combinaison linéaire entre le poids du BMU courant et le contexte courant $\cdot \gamma_t = \lambda \gamma_{t-1} + (1 - \lambda) \omega(\Pi_{t-1})$. Enfin, le modèle SOMD, initialement proposé pour le traitement de données structurées (Hagenbuchner et al. 2003) puis étendu au traitement de séquences en Hammer, Micheli, Sperduti et al. 2004 ; Hammer, Micheli, Neubauer et al. 2005 réduit ce contexte à la position de la Best Matching Unit :

$$\gamma_t = \Pi_{t-1}$$

Les mécanismes de transmission de contexte entre instants dans les cartes récurrentes s'appuient donc sur les mêmes mécanismes que ceux proposés dans le cadre d'architectures de cartes : sélection de région de la carte, transmission d'activation, et enfin transmission du BMU.

Architectures incluant des connexions temporelles

Certains modèles s'appuient sur plusieurs cartes de Kohonen connectées, en y ajoutant une notion de traitement de séquences. En Parisi et al. 2018, les auteurs développent une architecture de deux réseaux auto-organisés appelés *grow when required networks* (GWR), voir figure 1.20. Ces réseaux sont des versions incrémentales de cartes de Kohonen dans lesquelles des neurones sont ajoutés au cours de l'apprentissage, le processus de recherche de BMU restant ensuite similaire à une SOM classique. Cette architecture utilise deux réseaux GWR pour apprendre des

séquences, formant une mémoire épisodique et une mémoire sémantique. La carte associée à la mémoire épisodique (G-EM) est une version récurrente du GWR, prenant en entrée courante X_t et en entrée de contexte $\omega(\Pi_{t-1})$, le poids du BMU à l'instant précédent. La deuxième carte est une version classique du GWR. Elle prend en entrée le poids du BMU de la carte épisodique ainsi que la classe de la séquence courante, afin de mettre à jour ses poids. Les auteurs utilisent leur architecture pour de la reconnaissance d'objets. Cependant, lors de l'apprentissage, les données ne sont pas présentées après un tirage aléatoire dans l'espace, mais sont présentées classe par classe : tous les objets d'une même classe d'abord, etc. Les auteurs montrent que l'architecture est capable de bien prédire la classe d'un objet lors d'un test sur toutes les classes apprises. À titre de comparaison, une SOM classique apprendrait la classe du premier objet, puis l'oublierait pour se déplier entièrement sur la deuxième classe ; à terme, seule la dernière classe serait gardée en mémoire. Ce type de structure prenant des entrées évoluant dans le temps et les gardant en mémoire s'inscrit dans l'apprentissage développemental. Nous entrevoyons ainsi l'intérêt que peuvent présenter des structures rassemblant connexions temporelles et intercartes au sein d'une même architecture. On ne peut pas vraiment parler d'architecture modulaire dans ces travaux, les deux couches de cartes étant différentes et spécialement conçues pour l'application d'apprentissage de séquence réalisée par les auteurs. Une logique de vérification externe aux cartes est par ailleurs utilisée pour ajouter ou non des neurones dans la couche supervisée. La carte récurrente est donc une manière de filtrer les entrées avant d'effectuer de l'apprentissage supervisé. Par contre, la motivation de ce modèle est intéressante : il s'agit cette fois de voir les deux cartes comme des modules d'apprentissage à différentes échelles temporelles. Avec les bonnes règles de mise à jour, cette propriété pourrait émerger dans des architectures modulaires.

Les travaux autour du modèle A-SOM mentionné précédemment ont également dérivé une version récurrente du modèle (Buonamente et al. 2015) dans le but d'associer cartes récurrentes et multimodales en architecture. Cette version récurrente est similaire à la version multi-cartes. Elle calcule alors son activité par rapport à son entrée et possède une seconde couche de poids qu'elle met à jour relativement au champ d'activation de l'instant précédent. Cette structure est appliquée à la prédiction de mouvement. De la même façon qu'une architecture est capable, à partir d'une modalité, de prédire les valeurs correspondant à l'autre modalité, l'architecture incluant une version récurrente peut prédire la fin d'une séquence à partir de son début. Nous n'avons cependant pas relevé de travaux les intégrant effectivement dans une architecture multi-cartes.

Enfin, nous pouvons revenir sur les travaux menés précédemment dans notre équipe décrits plus haut, qui proposent des architectures multi-cartes appliquées au traitement de séquence. Bassem Khouzam 2014 utilise le modèle Bijama, permettant de construire des architectures de cartes auto-organisatrices cellulaires, dans un cadre de traitement de séquences. Les connexions corticales des neurones d'une carte proviennent alors des activités des neurones de cette même carte au pas de temps précédent. Le modèle à base de cartes auto-organisatrices classiques

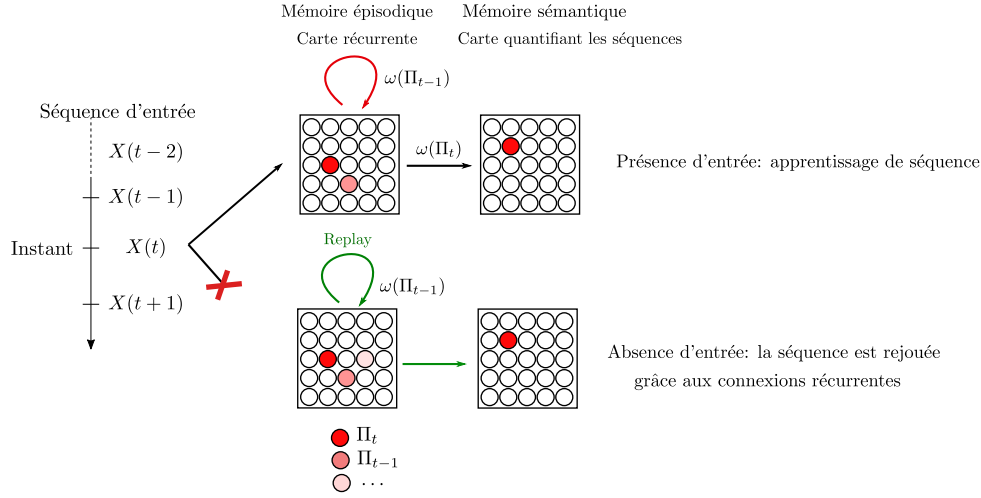


FIGURE 1.20 – Architecture à double mémoire proposée en Parisi et al. 2018. La couche de mémoire épisodique, permettant la différenciation de séquences, prend en entrée externe un instant de la séquence d'entrée X et en entrée de contexte le poids du BMU de l'instant précédent. La couche de mémoire sémantique est entraînée à partir du poids des BMU de la couche épisodique. Les auteurs ajoutent des conditions de classification supervisant la mise à jour de ces cartes, que nous ne détaillons pas sur ce schéma. Ce modèle est un exemple d'architecture assemblant cartes récurrentes et cartes classiques ; il s'agit ici d'un modèle semi-supervisé. Les auteurs utilisent cette architecture dans le cadre de l'apprentissage à long-terme.

proposé en Baheux et al. 2014 que nous avons décrit plus haut s'inscrit également dans un cadre de traitement de séquences. Le modèle montre qu'une carte de Kohonen au sein d'une architecture récurrente permet de différencier les BMU d'une carte non seulement en fonction de la valeur de l'entrée mais également en fonction de la position dans la séquence. Ces résultats sont similaires à ce qu'on obtient avec une carte récurrente s'appuyant sur la transmission du BMU, sans relaxation, décrite par exemple en J. Fix et Frezza-Buet 2020. La transformation du modèle récurrent en modèle multi-cartes montre la similarité existant entre récurrence et multimodalité et la possibilité d'inclusion de connexions récurrentes au sein d'un modèle multi-cartes.

1.5 Discussion et axe de recherche

Dans ce chapitre, nous avons présenté la littérature portant sur les architectures de SOM en la divisant en deux grandes catégories. D'une part, des travaux ont exploré des architectures hiérarchiques ou multicouches, se plaçant dans une optique d'amélioration des performances d'une SOM sur des applications de quantification vectorielle et de classification. Ces travaux relèvent du domaine de l'apprentissage automatique. Dans certains travaux, l'assemblage des cartes est régi par une surcouche algorithmique globale, ce qui nous amène à ne pas les considérer comme modulaires. D'autres travaux gèrent au contraire les connexions entre cartes à l'échelle d'une carte. Une carte prend alors le rôle de module d'une architecture. Chaque carte évolue

alors uniquement grâce aux règles d'évolution internes qui ont été définies et grâce aux interfaces venant d'autres modules. Cependant, des conditions sur la structure hiérarchique de l'architecture sont préétablies.

D'autre part, certains travaux portent sur la création d'architectures comportant des rétroactions, que nous appelons architectures non hiérarchiques. Ces architectures non hiérarchiques apportent moins de précondition sur la structure de l'architecture et sont donc plus génériques. Ces travaux se placent plutôt dans le domaine des neurosciences computationnelles ou de la robotique. La création de ces architectures est en effet motivée par des considérations biologiques, les neurosciences suggérant que les aires du cerveau présentent des connexions rétroactives. Ces architectures à rétroactions permettent l'activation d'une carte par une autre et apportent aux SOM une capacité de prise de décision et de prédiction lorsqu'elles sont au sein d'une architecture. Elles se présentent soit sous la forme d'une architecture centralisée, dans laquelle une carte associative permet d'associer des cartes sensorielles, ou sous la forme d'une architecture décentralisée. Ce dernier cas est la forme la plus générique d'architecture modulaire de cartes de Kohonen : chaque carte est un module autonome que l'on peut ajouter à une architecture existante sans différencier les modules a priori en fonction de leur position dans l'architecture. Les modèles de cartes récurrentes, adaptées au traitement de séquence, se rapprochent par ailleurs des architectures multi-cartes par leur structure s'appuyant sur une transmission d'information entre itérations d'apprentissage. Une architecture modulaire générique de cartes devrait ainsi laisser la possibilité d'intégrer de façon indifférenciée des connexions classiques ou récurrentes au sein d'une architecture, dans une motivation d'apprentissage développemental.

Dans la démarche de construction d'une architecture non-hiérarchique, nous avons souligné les différences de conception existant dans les différentes architectures, notamment au niveau du choix d'interface entre cartes et dans la séquentialité de la mise à jour. Le tableau 1.1 présente une comparaison des structures des principales architectures modulaires et récurrentes que nous avons relevées au cours de cette revue. Nous avons remarqué que seulement peu de travaux ont exploré l'idée d'associer des SOM en architectures non hiérarchiques. Parmi ces quelques travaux, les interfaces entre cartes considérées par leurs auteurs s'appuient principalement sur la transmission de champs d'activation neuronaux. La transmission de la position du BMU comme information entre cartes apparaît au contraire comme un paradigme principalement utilisé dans le domaine des cartes hiérarchiques ainsi que certains modèles récurrents. Ce mode de transmission exploite pleinement l'aspect topologique de la carte de Kohonen, est indépendante du type d'entrée fourni à une carte et est une valeur de faible dimension, donc intéressante pour le calcul. Ce paradigme permet aux architectures s'y appuyant de réaliser de bonnes performances en terme d'apprentissage automatique. Les architectures décentralisées sont principalement proposées sous le prisme d'une inspiration biologique, ce qui justifie l'utilisation privilégiée de transmission d'activité : les neurones biologiques sont connectés par des connexions neurones à neurones et se transmettent une activation. Seuls les travaux de [Lallee et Dominey 2013](#) s'appuient sur transmission de la

position du BMU dans le cadre des SOM non hiérarchiques. Enfin, nous avons constaté des différences dans le traitement des rétroactions entre cartes. Les architectures SOIMA, MCMM et A-SOM découplent les poids externes et contextuels des cartes lors de l'apprentissage, en différenciant les BMU sur chaque couche de poids, pour ne combiner de fait les activités que lors de la phase de prédiction d'entrée. Cette gestion des rétroactions ne va pas dans le sens d'une architecture autonome, car les règles de calcul changent entre la phase d'apprentissage et la phase de prédiction.

Le travail de recherche proposé dans cette thèse consiste à construire et étudier un modèle non-hiérarchique d'architecture de cartes, en s'inspirant de l'architecture corticale. Nous cherchons à nous placer plutôt du côté du calcul informatique, laissant la biologie comme une inspiration et non en cherchant à la modéliser. Pour ses avantages en termes de coût de transmission et d'homogénéité des calculs, nous nous tournerons vers la transmission de la position du BMU comme information entre modules. La plupart des architectures relevées dans la littérature s'appuient sur des mises à jour séquentielles. Une architecture générale incluant le traitement de séquences doit nécessairement gérer ses mises à jour de façon synchrone ; aussi nous choisissons de diriger nos recherches dans ce sens.

L'architecture CxSOM que nous proposons et étudions dans cette thèse est une architecture non hiérarchique décentralisée de cartes de Kohonen, dont les mises à jour sont réalisés de façon synchrone. Nous voulons également que les rétroactions soient prises en compte dès l'apprentissage, dans un objectif d'architecture autonome : nous ne différencions pas les règles d'évolution des modules entre phases d'apprentissage et phase de test ou de prédiction. Ce modèle fait suite aux travaux amorcés en [Baheux et al. 2014](#), qui définissent un modèle d'architecture récurrente utilisant la transmission de la position du BMU entre des cartes de Kohonen, et utilisent un mécanisme de relaxation pour gérer les rétroactions entre cartes. Ces travaux exploitent des connexions intercartes mais restent appliqués au traitement de séquences. Par leur motivation, qui est le développement d'un système multi-cartes générique, nos travaux se rapprochent aussi des travaux conduits sur l'architecture A-SOM [Johnsson et Balkenius 2008](#) ; [Johnsson, Balkenius et Hesslow 2009](#) ; [Gil et al. 2015](#) ; [Buonamente et al. 2015](#) ; les choix de modèle sont cependant différents.

Nous dirigerons les travaux de cette thèse vers l'identification des mécanismes d'apprentissage associatifs dans de petites architectures de deux et trois cartes. Pour cela, nous élaborerons une méthode expérimentale et des représentations adaptées du modèle, dans une optique de création d'architecture comportant de nombreuses cartes.

TABLE 1.1 – Comparaison des principaux modèles d’architectures relevés dans ce chapitre. Nous n’y faisons pas figurer les architectures sélectives, étant non modulaires. Les modèles très similaires sont regroupées sur une seule ligne. Les cartes récurrentes ne sont pas concernées par la question de séquence de mise à jour.

Modèle	Type	Mise à jour	Mode de transmission
HSOM ¹	Hiérarchique	Séquentielle	Position du BMU
Deep SOM ²	Hiérarchique	Séquentielle	Position du BMU
Autres ³	Hiérarchique	Séquentielle	Poids du BMU
MMCM ⁴	non hiérarchique, Centralisée	Séquentielle	Positions BMU
SOIMA ⁵	non hiérarchique, Centralisée	Séquentielle	Champ d’activité
Bijama ⁶	non hiérarchique, Décentralisée	Asynchrone	Champ d’activité partiel
A-SOM ⁷	non hiérarchique, Décentralisée	Synchrone	Champ d’activité
SOMMA ⁸	non hiérarchique, Décentralisée	Synchrone	Champ d’activité partiel
Jayaratne et al. 2018	non hiérarchique, Décentralisée	Séquentielle	Champ d’activité
Khacef et al. 2020	non hiérarchique, Décentralisée	Séquentielle	Champ d’activité
RSOM ⁹	Récurrente		Champ d’activité
MSOM ¹⁰	Récurrente		Poids du BMU
A-SOM ¹¹	Récurrente		Champ d’activité
Recursive SOM ¹²	Récurrente		Champ d’activité
SOMD ¹³	Récurrente		Position du BMU
Parisi et al. 2018	Récurrente, Hiérarchique	Synchrone	Poids du BMU
Baheux et al. 2014	Récurrente, non hiérarchique	Synchrone	Position du BMU

1 [Lampinen et Oja 1992](#); [Hagenauer et Helbich 2013](#)[Papinski et Gustafsson 2005](#)

2 [Liu et al. 2015](#),[Wickramasinghe et al. 2019](#)

4 [Lallee et Dominey 2013](#)

3 [Dozono et al. 2016](#); [Mici et al. 2018](#); [Nawaratne et al. 2020](#); [Aly et Almotairi 2020](#); [Wang et al. 2007](#); [Gunes Kayacik et al. 2007](#); [Luttrell 1989](#)

5 [Escobar-Juárez et al. 2016](#)

6 [Ménard et Frezza-Buet 2005](#); [B. Khouzam et H.Frezza-Buet 2013](#)

7 [Johnsson et Balkenius 2008](#)

8 [Lefort et al. 2011](#)

9 [Varsta et al. 2001](#)

10 [Strickert et Hammer 2005](#)

11 [Buonamente et al. 2013](#)

12 [Voegtlin 2002](#)

13 [Hammer, Micheli, Neubauer et al. 2005](#)

Bibliographie

- Alahakoon, D., S.K. Halgamuge et B. Srinivasan. “Dynamic self-organizing maps with controlled growth for knowledge discovery”. In : *IEEE Transactions on Neural Networks* 3 (2000). DOI : [10.1109/72.846732](https://doi.org/10.1109/72.846732) (6).
- Aly, Saleh et Sultan Almotairi. “Deep Convolutional Self-Organizing Map Network for Robust Handwritten Digit Recognition”. In : *IEEE Access* 8 (2020), p. 107035-107045. DOI : [10.1109/ACCESS.2020.3000829](https://doi.org/10.1109/ACCESS.2020.3000829) (17, 39).
- Amari, Shun-ichi. “Dynamics of pattern formation in lateral-inhibition type neural fields”. In : *Biological Cybernetics* 27 (1977), p. 77-87 (60).
- Baheux, D., J. Fix et H. Frezza-Buet. “Towards an effective multi-map self organizing recurrent neural network”. In : *Proc. ESANN*. 2014 (30, 31, 36, 38, 39, 49, 170).
- Ballard, Dana H. “Cortical connections and parallel processing : Structure and function”. en. In : *Behavioral and Brain Sciences* 9.01 (1986), p. 67. DOI : [10.1017/S0140525X00021555](https://doi.org/10.1017/S0140525X00021555) (111, 112).
- Barbalho, J.M. et al. “Hierarchical SOM applied to image compression”. In : *IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*. T. 1. ISSN : 1098-7576. 2001, 442-447 vol.1. DOI : [10.1109/IJCNN.2001.939060](https://doi.org/10.1109/IJCNN.2001.939060) (12, 13).
- Betz, Richard F. et Danielle S. Bassett. “Multi-scale brain networks”. In : *NeuroImage* 160 (2017), p. 73-83. DOI : <https://doi.org/10.1016/j.neuroimage.2016.11.006> (vi, 170).
- Binzegger, Tom, Rodney J. Douglas et Kevan A. C. Martin. “Cortical Architecture”. In : *Brain, Vision, and Artificial Intelligence*. Sous la dir. de M. De Gregorio et al. Springer-Verlag, 2005 (vi).
- Bonath, Bjoern et al. “Neural Basis of the Ventriloquist Illusion”. In : *Current Biology* (2007) (7).
- Brooks, Rodney A. “A robust layered control system for a mobile robot”. In : *IEEE J. Robotics Autom.* 2 (1986) (vii).

- Buonamente, Miriam, Haris Dindo et Magnus Johnsson. “Discriminating and simulating actions with the associative self-organising map”. In : *Connection Science* 27 (2015), p. 118-136 ([35](#), [38](#)).
- “Hierarchies of Self-Organizing Maps for action recognition”. In : *Cognitive Systems Research* 39 (2016), p. 33-41. DOI : [10.1016/j.cogsys.2015.12.009](#) ([27](#)).
- “Simulating Actions with the Associative Self-Organizing Map”. In : *AIC@AI*IA*. 2013 ([34](#), [39](#)).
- Burnod, Yves. “An adaptive neural network - the cerebral cortex”. In : 1989 ([8](#)).
- Calvert, Gemma A. et Thomas Thesen. “Multisensory integration : methodological approaches and emerging principles in the human brain”. In : *Journal of Physiology-Paris* 98 (2004), p. 191-205 ([8](#)).
- Cangelosi, Angelo et al. “Embodied Intelligence”. In : *Springer Handbook of Computational Intelligence*. Sous la dir. de Janusz Kacprzyk et Witold Pedrycz. Berlin, Heidelberg : Springer Berlin Heidelberg, 2015, p. 697-714. DOI : [10.1007/978-3-662-43505-2_37](#) ([2](#)).
- Cappe, Catharine, Eric M. Rouiller et Pascal Barone. “Multisensory anatomical pathways”. In : *Hearing Research* 258 (2009), p. 28-36 ([8](#)).
- Clune, Jeff, Jean-Baptiste Mouret et Hod Lipson. “The evolutionary origins of modularity”. en. In : *Proceedings of the Royal Society B : Biological Sciences* 280.1755 (2013), p. 20122863 ([vi](#)).
- Costa, José Alfredo Ferreira, Adrião Duarte Dória Neto et Márcio Luiz De Andrade Netto. “A New Structured Self-Organizing Map with Dynamic Growth Applied to Image Compression”. In : *Proceedings of the VI Brazilian Conference on Neural Networks*. 2016 ([13](#)).
- Cottrell, Marie, Jean-Claude Fort et Gilles Pagès. “Theoretical aspects of the SOM algorithm”. In : *Neurocomputing* 21 (1998), p. 119-138 ([6](#), [45](#), [99](#)).
- Cottrell, Marie, Madalina Olteanu et al. “Theoretical and Applied Aspects of the Self-Organizing Maps”. In : *Advances in Self-Organizing Maps and Learning Vector Quantization*. T. 428. Cham : Springer International Publishing, 2016, p. 3-26. DOI : [10.1007/978-3-319-28518-4_1](#) ([6](#)).
- Cover, Thomas M. et Joy A. Thomas. “Elements of Information Theory : Cover/Elements of Information Theory, Second Edition”. In : 2005 ([134](#)).
- Damasio, Antonio R. “Time-locked multiregional retroactivation : A systems-level proposal for the neural substrates of recall and recognition”. In : *Cognition* 33.1 (1989). DOI : [https://doi.org/10.1016/0010-0277\(89\)90005-X](#) ([8](#)).

- Darwish, Ashraf A. “Bio-inspired computing : Algorithms review, deep analysis, and the scope of applications”. In : *Future Computing and Informatics Journal* (2018) ([v](#)).
- Dittenbach, M., D. Merkl et A. Rauber. “The growing hierarchical self-organizing map”. In : *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing : New Challenges and Perspectives for the New Millennium*. Como, Italy : IEEE, 2000, 15-19 vol.6. DOI : [10.1109/IJCNN.2000.859366](#) ([13](#)).
- Doquire, Gauthier et Michel Verleysen. “A Comparison of Multivariate Mutual Information Estimators for Feature Selection”. In : *ICPRAM*. 2012 ([135](#), [147](#)).
- Dozono, Hiroshi, Gen Niina et Satoru Araki. “Convolutional Self Organizing Map”. In : *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2016, p. 767-771. DOI : [10.1109/CSCI.2016.0149](#) ([16](#), [17](#), [39](#)).
- Edelman, Gerald M. “Group selection and phasic reentrant signaling a theory of higher brain function”. In : *The 4th Intensive Study Program Of The Neurosciences Research Program*. 1982 ([8](#), [23](#)).
- Ellefsen, Kai Olav, Jean-Baptiste Mouret et Jeff Clune. “Neural Modularity Helps Organisms Evolve to Learn New Skills without Forgetting Old Skills”. In : *PLoS Computational Biology* 11 (2015) ([170](#)).
- Escobar-Juárez, Esau et al. “A Self-Organized Internal Models Architecture for Coding Sensory–Motor Schemes”. en. In : *Frontiers in Robotics and AI* (2016) ([23](#), [25](#), [26](#), [39](#)).
- Felleman, Daniel J. et David C. Van Essen. “Distributed Hierarchical Processing in the Primate Cerebral Cortex”. In : (1991) ([8](#), [9](#)).
- Fix, Jérémy. “Template based black-box optimization of dynamic neural fields”. In : *Neural Networks* 46 (2013), p. 40-49 ([62](#)).
- Fix, Jérémy et Hervé Frezza-Buet. “Look and Feel What and How Recurrent Self-Organizing Maps Learn”. In : *Proc. WSOM’19*. T. 976. 2020, p. 3-12 ([36](#), [170](#)).
- Fix, Jeremy David, Nicolas P. Rougier et Frédéric Alexandre. “A Dynamic Neural Field Approach to the Covert and Overt Deployment of Spatial Attention”. In : *Cognitive Computation* 3 (2011), p. 279-293 ([61](#)).
- Fort, J.C. “SOM’s mathematics”. en. In : *Neural Networks* 19.6-7 (2006). DOI : [10.1016/j.neunet.2006.05.025](#) ([6](#)).
- Foxe, John J. et Charles E. Schroeder. “The case for feedforward multisensory convergence during early cortical processing”. In : *NeuroReport* 16 (2005), p. 419-423 ([8](#)).

- Frezza-Buet, Hervé. “Self-organizing maps in manifolds with complex topologies : An application to the planning of closed path for indoor UAV patrols”. In : *ESANN*. 2020 (6).
- Fritzke, Bernd. “Growing Grid — a self-organizing network with constant neighborhood range and adaptation strength”. In : *Neural Processing Letters* 2 (1995), p. 9-13 (6).
- Gao, Weihao et al. “Estimating Mutual Information for Discrete-Continuous Mixtures”. In : *NIPS*. 2017 (147).
- Gil, David et al. “SARASOM : a supervised architecture based on the recurrent associative SOM”. en. In : *Neural Computing and Applications* 26.5 (2015) (38).
- González, Julio et al. “Reading cinnamon activates olfactory brain regions”. In : *NeuroImage* 32 (2006), p. 906-912 (8).
- Gunes Kayacik, H., A. Nur Zincir-Heywood et Malcolm I. Heywood. “A hierarchical SOM-based intrusion detection system”. In : *Engineering Applications of Artificial Intelligence* 20.4 (2007), p. 439-451. DOI : [10.1016/j.engappai.2006.09.005](https://doi.org/10.1016/j.engappai.2006.09.005) (16, 39).
- Hagenauer, Julian et Marco Helbich. “Hierarchical self-organizing maps for clustering spatio-temporal data”. In : *International Journal of Geographical Information Science* 27.10 (2013), p. 2026-2042. DOI : [10.1080/13658816.2013.788249](https://doi.org/10.1080/13658816.2013.788249) (16, 39).
- Hagenbuchner, M., A. Sperduti et Ah Chung Tsoi. “A self-organizing map for adaptive processing of structured data”. In : *IEEE Transactions on Neural Networks* 14.3 (2003), p. 491-505. DOI : [10.1109/TNN.2003.810735](https://doi.org/10.1109/TNN.2003.810735) (34, 166).
- Hammer, Barbara, Alessio Micheli, Nicolas Neubauer et al. “Self-Organizing Maps for Time Series”. In : *Proc. WSOM'2005*. 2005, p. 8 (34, 39).
- Hammer, Barbara, Alessio Micheli, Alessandro Sperduti et al. “Recursive self-organizing network models”. In : *Neural Networks* 17.8-9 (2004), p. 1061-1085. DOI : [10.1016/j.neunet.2004.06.009](https://doi.org/10.1016/j.neunet.2004.06.009) (34, 58).
- Hankins, Richard, Yao Peng et Hujun Yin. “SOMNet : Unsupervised Feature Learning Networks for Image Classification”. In : *2018 International Joint Conference on Neural Networks (IJCNN)*. Rio de Janeiro : IEEE, 2018, p. 1-8. DOI : [10.1109/IJCNN.2018.8489404](https://doi.org/10.1109/IJCNN.2018.8489404) (17).
- Henriques, Roberto, Victor Lobo et Fernando Bacao. “Spatial Clustering Using Hierarchical SOM”. en. In : *Applications of Self-Organizing Maps*. Sous la dir. de Magnus Johnsson. InTech, 2012 (12).
- Hjelm, R. Devon et al. “Learning deep representations by mutual information estimation and maximization”. In : *ArXiv abs/1808.06670* (2018) (147).

- Jayaratne, Madhura et al. “Bio-Inspired Multisensory Fusion for Autonomous Robots”. In : *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. ISSN : 2577-1647. 2018, p. 3090-3095 ([27](#), [39](#)).
- Johnsson, Magnus et Christian Balkenius. “Associating SOM Representations of Haptic Submodalities”. In : *Computer Science* (2008) ([27](#), [38](#), [39](#)).
- Johnsson, Magnus, Christian Balkenius et Germund Hesslow. “Associative Self-Organizing Map”. In : *Proc. IJCCI*. 2009 ([27](#), [38](#)).
- Khacef, Lyes, Laurent Rodriguez et Benoit Miramond. “Brain-inspired self-organization with cellular neuromorphic computing for multimodal unsupervised learning”. In : *ArXiv :2004.05488 [cs, Q-bio]* (2020). arXiv : 2004.05488 ([26](#), [39](#)).
- Khouzam, B. et H.Frezza-Buet. “Distributed recurrent self-organization for tracking the state of non-stationary partially observable dynamical systems”. In : *Biologically Inspired Cognitive Architectures* (2013) ([39](#), [49](#)).
- Khouzam, Bassem. “Neural Networks as Cellular Computing Models for Temporal Sequence Processing”. Thèse de doct. Supélec, 2014 ([35](#), [60](#), [61](#)).
- Kiefer, Markus et al. “The Sound of Concepts : Four Markers for a Link between Auditory and Conceptual Brain Systems”. In : *The Journal of Neuroscience* 28 (2008), p. 12224-12230 ([8](#)).
- Kohonen, Teuvo. “Essentials of the self-organizing map”. In : *Neural Networks* 37 (2013), p. 52-65. DOI : <https://doi.org/10.1016/j.neunet.2012.09.018> ([vii](#)).
- “Self-organized formation of topologically correct feature maps”. In : *Biological Cybernetics* 43.1 (1982) ([vii](#), [2](#), [42](#)).
- “Self-Organizing Maps”. In : *Springer Series in Information Sciences*. 1995 ([viii](#), [5](#), [7](#), [42](#)).
- Koikkalainen, P. et Erkki Oja. In : *1990 IJCNN International Joint Conference on Neural Networks*. 1990. DOI : [10.1109/IJCNN.1990.137727](https://doi.org/10.1109/IJCNN.1990.137727) ([6](#)).
- Kotseruba, Iuliia et John K. Tsotsos. “40 years of cognitive architectures : core cognitive abilities and practical applications”. In : *Artificial Intelligence Review* 53 (2018), p. 17-94 ([vi](#)).
- Kraskov, Alexander, Harald Stögbauer et Peter Grassberger. “Estimating mutual information”. In : *Physical Review E* 69.6 (2004). DOI : [10.1103/physreve.69.066138](https://doi.org/10.1103/physreve.69.066138) ([135](#)).
- Lahat, Dana, Tülay Adalı et Christian Jutten. “Multimodal Data Fusion : An Overview of Methods, Challenges and Prospects”. In : *Proceedings of the IEEE*. Multimodal Data Fusion 103.9 (2015), p. 1449-1477 ([21](#)).

- Lallee, Stephane et Peter Dominey. “Multi-modal convergence maps : From body schema and self-representation to mental imagery”. In : *Adaptive Behavior* 21.4 (2013), p. 274-285 ([23](#), [24](#), [37](#), [39](#)).
- Lampinen, Jouko et Erkki Oja. “Clustering Properties of Hierarchical Self-Organizing Maps”. In : *Journal of Mathematical Imaging and Vision* (1992), p. 15 ([15](#), [16](#), [39](#), [47](#), [165](#)).
- LeCun, Yann, Yoshua Bengio et Geoffrey Hinton. “Deep learning”. In : *Nature* 521.7553 (2015), p. 436-444. DOI : [10.1038/nature14539](#) ([17](#)).
- Lefort, Mathieu. “Apprentissage spatial de corrélations multimodales par des mécanismes d’inspiration corticale”. Thèse de doct. Université de Lorraine, 2012 ([30](#)).
- Lefort, Mathieu, Yann Boniface et Bernard Girau. “Unlearning in the BCM Learning Rule for Plastic Self-organization in a Multi-modal Architecture”. In : *Artificial Neural Networks and Machine Learning, ICANN 2011*. Berlin, Heidelberg, 2011 ([28](#), [30](#), [39](#)).
- Linde, Y., A. Buzo et R. Gray. “An Algorithm for Vector Quantizer Design”. In : *IEEE Transactions on Communications* 28.1 (1980), p. 84-95. DOI : [10.1109/TCOM.1980.1094577](#) ([14](#)).
- Liu, Nan, Jinjun Wang et Yihong Gong. “Deep Self-Organizing Map for visual classification”. In : *2015 International Joint Conference on Neural Networks (IJCNN)*. ISSN : 2161-4407. 2015, p. 1-6. DOI : [10.1109/IJCNN.2015.7280357](#) ([17](#), [18](#), [39](#)).
- Luttrell, S.P. “Hierarchical vector quantisation”. In : *IEE Proceedings I Communications, Speech and Vision* (1989) ([14](#), [39](#)).
- Maass, Wolfgang. “Networks of Spiking Neurons : The Third Generation of Neural Network Models”. In : *Electron. Colloquium Comput. Complex.* TR96 (1996) ([vi](#)).
- McCulloch, Warren S. et Walter H. Pitts. “A logical calculus of the ideas immanent in nervous activity”. In : *Bulletin of Mathematical Biology* 52 (1990), p. 99-115 ([v](#)).
- McGurk, Harry et John MacDonald. “Hearing lips and seeing voices”. In : *Nature* 264 (1976), p. 746-748 ([8](#), [22](#)).
- Ménard, Olivier et Hervé Frezza-Buet. “Model of multi-modal cortical processing : Coherent learning in self-organizing modules”. In : *Neural Networks* 18.5-6 (2005), p. 646-655 ([28](#), [29](#), [39](#), [49](#), [60](#), [61](#)).
- Meunier, David et al. “Hierarchical Modularity in Human Brain Functional Networks”. In : *Frontiers in Neuroinformatics* 3 (2009) ([vi](#)).

- Mici, Luiza, German I. Parisi et Stefan Wermter. “A self-organizing neural network architecture for learning human-object interactions”. In : *Neurocomputing* 307 (2018), p. 14-24. DOI : [10.1016/j.neucom.2018.04.015](https://doi.org/10.1016/j.neucom.2018.04.015) (16, 17, 22, 39).
- Miikkulainen, Risto. “Script Recognition with Hierarchical Feature Maps”. In : *Connection Science* 2 (1992), p. 196-214 (13).
- Nawaratne, Rashmika et al. “Hierarchical Two-Stream Growing Self-Organizing Maps With Transience for Human Activity Recognition”. In : *IEEE Transactions on Industrial Informatics* 16.12 (2020). Conference Name : IEEE Transactions on Industrial Informatics, p. 7756-7764. DOI : [10.1109/TII.2019.2957454](https://doi.org/10.1109/TII.2019.2957454) (17, 22, 39).
- Ordonez, Diego et al. “Hierarchical Clustering Analysis with SOM Networks”. In : *International Journal of Computer and Information Engineering* 4.9 (2010), p. 1393-1399 (13).
- Papilinski, Andrew P. et Lennart Gustafsson. “Multimodal FeedForward Self-organizing Maps”. In : *CIS*. 2005 (15, 39).
- Parisi, German I. et al. “Lifelong Learning of Spatiotemporal Representations With Dual-Memory Recurrent Self-Organization”. In : *Frontiers in Neurorobotics* (2018) (34, 36, 39).
- Pless, Robert et Richard Souvenir. “A Survey of Manifold Learning for Images”. In : *IPSJ Trans. Comput. Vis. Appl.* 1 (2009), p. 83-94 (84, 85).
- Reale, Richard A. et Thomas J. Imig. “Tonotopic organization in auditory cortex of the cat”. In : *Journal of Comparative Neurology* 192 (1980) (7).
- Ross, Brian C. “Mutual Information between Discrete and Continuous Data Sets”. In : *PLoS ONE* 9 (2014) (135, 147).
- Sakkari, Mohamed et Mourad Zaied. “A Convolutional Deep Self-Organizing Map Feature extraction for machine learning”. In : *Multimedia Tools and Applications* 79.27-28 (2020), p. 19451-19470. DOI : [10.1007/s11042-020-08822-9](https://doi.org/10.1007/s11042-020-08822-9) (17).
- Sandamirskaya, Yulia. “Dynamic neural fields as a step toward cognitive neuromorphic architectures”. In : *Frontiers in Neuroscience* 7 (2014) (61).
- Sathian, Krish et Andro Zangaladze. “Feeling with the mind’s eye : contribution of visual cortex to tactile perception”. In : *Behavioural Brain Research* 135 (2002), p. 127-132 (8).
- Schroeder, Charles E. et John J. Foxe. “Multisensory contributions to low-level, ‘unisensory’ processing”. In : *Current Opinion in Neurobiology* 15 (2005), p. 454-458 (8).
- Shannon, Claude E. “A mathematical theory of communication”. In : *Bell Syst. Tech. J.* 27 (1948) (132).

- Shwartz-Ziv, Ravid et Naftali Tishby. "Opening the Black Box of Deep Neural Networks via Information". In : *ArXiv* abs/1703.00810 (2017) ([147](#)).
- Smith, Linda B. et Michael Gasser. "The Development of Embodied Cognition : Six Lessons from Babies". In : *Artificial Life* 11 (2005), p. 13-29 ([2](#), [21](#)).
- Sporns, Olaf. "Structure and function of complex brain networks". en. In : *Dialogues in Clinical Neuroscience* 15.3 (2013), p. 247-262. DOI : [10.31887/DCNS.2013.15.3/osporns](#) ([vi](#)).
- Strickert, Marc et Barbara Hammer. "Merge SOM for temporal data". In : *Neurocomputing* 64 (2005), p. 39-71 ([34](#), [39](#)).
- Suganthan, P N. "Pattern classification using multiple hierarchical overlapped self-organising maps". In : *Pattern Recognition* (2001), p. 7 ([13](#)).
- Theil, Henri et al. "Economic Forecasts And Policy Ed. 2nd". In : *Birchandra State Central Library,tripura* (1961) ([135](#), [136](#)).
- Varsta, Markus, Jukka Heikkonen et Jouko Lampinen. "Temporal Kohonen Map and the Recurrent Self-Organizing Map : Analytical and Experimental Comparison". In : *Neural Processing Letters* (2001), p. 15 ([33](#), [39](#)).
- Voegtlin, Thomas. "Recursive self-organizing maps". In : *Neural Networks : the Official Journal of the International Neural Network Society* 15 8-9 (2002), p. 979-991 ([34](#), [39](#)).
- Wang, Liang, Eliathamby Ambikairajah et Eric H.C. Choi. "A comparisonal study of the multi-layer Kohonen self-organizing feature maps for spoken language identification". In : *2007 IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*. 2007, p. 402-407. DOI : [10.1109/ASRU.2007.4430146](#) ([16](#), [39](#)).
- Wickramasinghe, Chathurika S., Kasun Amarasinghe et Milos Manic. "Deep Self-Organizing Maps for Unsupervised Image Classification". In : *IEEE Transactions on Industrial Informatics* 15.11 (2019), p. 5837-5845. DOI : [10.1109/TII.2019.2906083](#) ([17](#), [39](#)).
- Williams, Paul L. et Randall D. Beer. "Nonnegative Decomposition of Multivariate Information". In : *ArXiv :1004.2515 [math-ph, Physics :physics, Q-bio]* (2010). arXiv : 1004.2515 ([146](#), [147](#)).
- Yamaguchi, Takashi, Takumi Ichimura et Kenneth James Mackin. "Adaptive Learning Algorithm in Tree-Structured Self-Organizing Feature Map". In : *SCIS&ISIS, Okayoma*. 2010, p. 6 ([6](#)).