

! Gr° pour ° des images !

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	3
1.2.1	Description du modèle de carte auto-organisatrice de Kohonen	3
1.2.2	La carte de Kohonen, un algorithme d'apprentissage de représentation non supervisé	7
1.3	Inspiration biologique des architectures de cartes	8
1.3.1	Inspiration biologique des cartes de Kohonen	8
1.3.2	Rétroactions dans le traitement de l'information multisensorielle du cortex	9
1.4	Architectures de cartes auto-organisatrices	10
1.4.1	Méthode d'analyse	12
1.4.2	Architectures hiérarchiques de cartes	15
1.4.3	Architectures non hiérarchiques de cartes auto-organisatrices	23
1.4.4	Connexions temporelles et architectures de cartes auto-organisatrices	36
1.5	Axe de recherche	40

1.1 Introduction

Les travaux que nous présentons dans cette thèse se concentrent sur la création d'une architecture non hiérarchique modulaire de cartes auto-organisatrices (SOM). Les cartes auto-organisatrices et notamment le modèle de Kohonen sont largement utilisées en tant qu'algorithme d'apprentissage non supervisé appliqué à des tâches de réduction de dimension, de visualisation

soit pour... soit pour?

Chapitre 1. Architectures de cartes auto-organisatrices

de données ou de classification. D'une part, de nombreux travaux étudient 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 topologique et la simplicité des règles de calcul d'une carte pour les assembler en architectures modulaires comportant des rétroactions, formant un système dynamique et conférant à une architecture de cartes un comportement de prise de décision.] ?

def?

les aspects topo, les règles de m. mais permet-
de supprimer?

L'étude d'une architecture non hiérarchique de cartes est d'une part motivée par leur inspiration biologique. Leur organisation rappelle en effet celle qu'on peut observer dans des aires cérébrales. 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. Ensuite, l'étude des systèmes biologiques et la robotique sont liées : la biologie sert d'inspiration à la robotique, que ce soit pour le mouvement d'un bras ou la prise de décision, et la robotique permet de tester des théories cherchant à modéliser des comportements biologiques (Oudeyer 2010). Aussi les architectures de cartes bio-inspirées que nous avons relevées dans la littérature se placent aussi dans les domaines des neurosciences computationnelles ou de l'apprentissage incarné (*Embodied intelligence*) en robotique (Smith et Gasser 2005 ; Cangelosi et al. 2015), à la frontière entre étude de la biologie et apprentissage automatique.

Nous passerons en revue dans ce chapitre un ensemble de travaux définissant des architectures de cartes auto-organisatrices, en s'intéressant aux différences de conception existant au sein des architectures. Nous définirons ainsi la notion d'architecture non hiérarchique modulaire. Cette notion d'architecture modulaire de cartes est bien résumée 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.
(Kohonen 1995)

en s'appliquant
sur ...

Ces éléments de contrôle sont les modules d'une architecture. L'objectif de nos travaux est ainsi de proposer un modèle de carte qui puisse être utilisée en tant que module, de définir l'interface entre les modules, afin de créer une architecture et de comprendre les comportements de calcul qui émergent de l'association des modules. Nous présentons dans ce chapitre 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. À l'issue de ce chapitre, nous aurons une vue d'ensemble des catégories d'architectures de cartes auto-organisatrices existant

dans la littérature et de leurs principales propriétés d'apprentissage. Nous pourrons ainsi définir en pratique la notion d'architecture modulaire et comment se placent nos travaux dans cette taxonomie.

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

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. De nombreux modèles dérivés ont ensuite été développés à partir de ce modèle, sur diverses applications. Nous présentons dans cette section le modèle initial de carte de Kohonen et introduisons les notations que nous utiliserons dans cette partie de revue.

1.2.1 Description du modèle de carte auto-organisatrice de Kohonen

Une carte de Kohonen est un algorithme de quantification vectorielle. Les éléments principaux de ce modèle décrits ci-après sont représentés en figure [1.2](#). Les algorithmes de quantification vectorielle cherchent à représenter un ensemble de données d'entrées issues d'un espace \mathcal{D} en un nombre fini de vecteurs de l'espace d'entrée, les prototypes. Dans une SOM, ces prototypes sont disposés sur les nœuds d'un graphe, en général une ligne 1D ou une grille en deux dimensions. Les nœuds du graphe possèdent alors chacun un prototype ω et sont *indexés* par p , vecteur aux dimensions de la carte, soit un réel pour une carte 1D ou un vecteur en deux dimensions lorsque que la carte est une grille. Cette indexation et la connectivité du graphe permettent de définir une distance d dans la carte et donc une notion de *voisinage* entre les nœuds. Nous appellerons carte de Kohonen le graphe assorti de ses prototypes $\omega(p)$.

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

1. Une entrée X est présentée à la carte.
2. Le nœud ayant le prototype le plus proche de X selon une distance d est choisi comme *Best Matching Unit* (BMU) de la carte. Son indice est noté Π . La distance d généralement utilisée est la distance euclidienne.
3. Le prototype du BMU $\omega(\Pi)$ ainsi que les prototypes des nœuds voisins sont déplacés vers X . Le déplacement est pondéré par leur degré de proximité au BMU, défini par la fonction de voisinage $H(p, \Pi)$

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

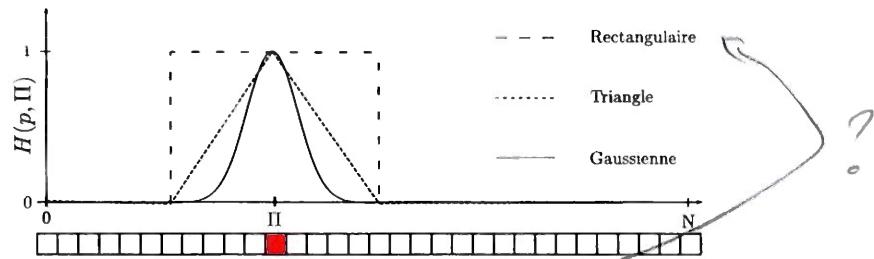


FIGURE 1.1 – Exemples de fonctions de voisinage (Rectangle, Triangle ou Gaussienne), centrées sur le BMU II, couramment utilisées sur une carte, ici en une dimension.

On peut interpréter cette étape comme le déplacement d'une zone de la carte centrée en II.

Ce voisinage est défini par la fonction de voisinage H dans l'algorithme qui dépend de la distance d'un nœud au BMU et associe à chaque nœud un coefficient multiplicatif pour la mise à jour des poids. Cette fonction est maximale à la position du BMU et décroissante autour de cette position. Il s'agira par exemple d'une fonction rectangulaire, triangle ou gaussienne. L'algorithme de Kohonen repose donc 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.

La version originale de l'algorithme de Kohonen s'appuie sur le calcul de distances entre entrées et prototypes. Ce dernier est remplacé dans d'autres modèles de cartes par le calcul d'une *activation* $a(X, \omega(p))$ liant les poids des nœuds et les entrées. La BMU est alors l'unité située au maximum de l'activation, au lieu du minimum des distances.

Le processus de mise à jour des poids d'une carte de Kohonen se traduit visuellement par un dépliement de la carte 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 figures 1.3 et 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 donc une continuité des valeurs des poids au sein de la carte.
- Elle conserve les densités. Une zone dense de \mathcal{D} aura plus d'unités correspondant à cette zone de valeurs dans la carte qu'une zone moins dense.

La figure 1.5 représente le dépliement d'une carte sur des imagettes MNIST. Par son aspect ordonné, une carte est une représentation en dimension 2 d'un espace d'entrée de grande dimension.

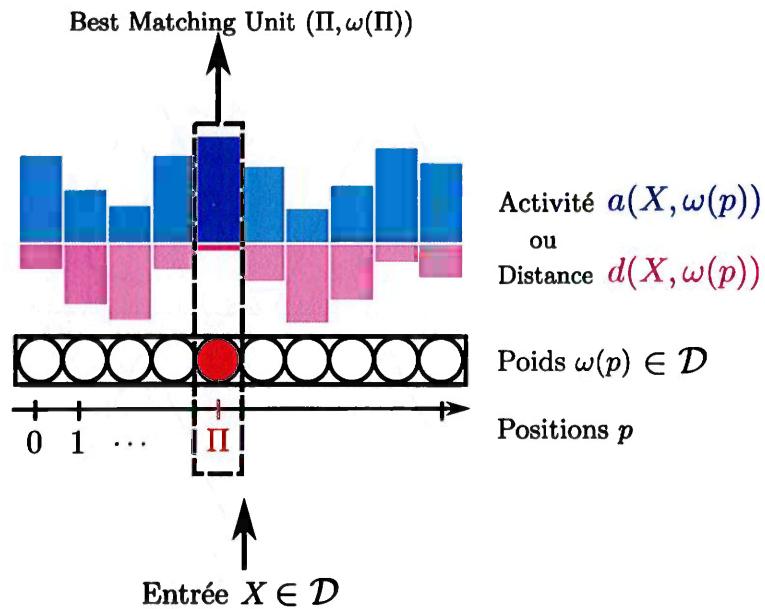


FIGURE 1.2 – Éléments principaux composant une carte de Kohonen : une carte possède un ensemble d'unités de poids ω , indexées par une position p . En présence d'une entrée X , une activité a ou une distance d est calculée pour chaque unité par rapport à l'entrée. La *Best Matching Unit*, abrégée en BMU, est calculée comme l'unité d'activité maximale sur les positions (ou de distance minimale). Sa position est notée Π et son poids est $\omega(\Pi)$.

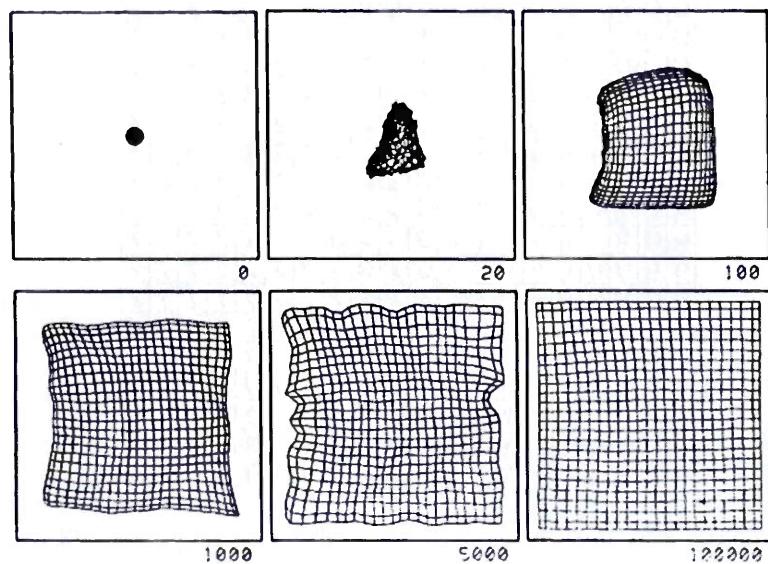


FIGURE 1.3 – Dépliement d'une SOM 2D sur des données dans le plan $[0, 1]^2$, (Kohonen 1995)
t + n b itératio?

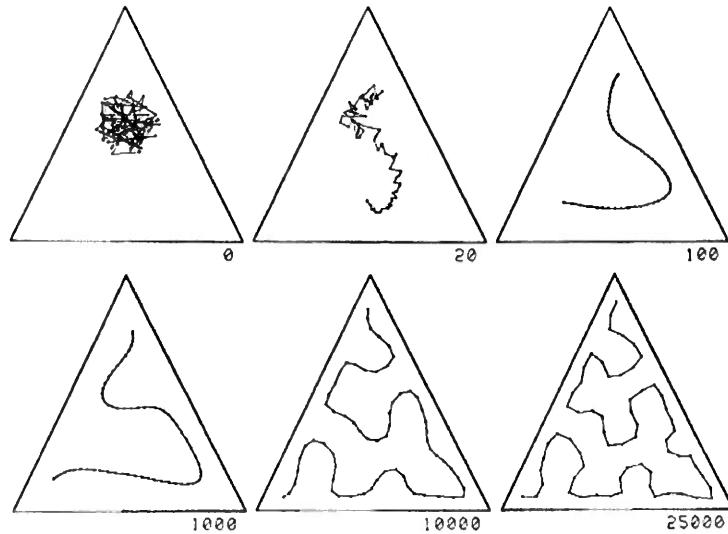


FIGURE 1.4 – Dépliement d'une SOM 1D sur des données dans un triangle 2D, (Kohonen 1995)

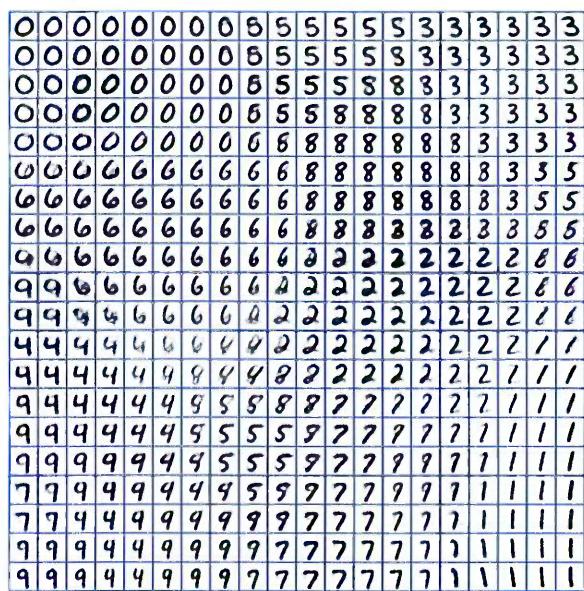


FIGURE 1.5 – Représentation de la base de données MNIST, images de chiffres manuscrits, par une SOM en deux dimensions. Une continuité est observée dans la forme des prototypes lorsqu'on se déplace dans la carte.

1.2.2 La carte de Kohonen, un algorithme d'apprentissage de représentation non supervisé

La carte de Kohonen se distingue d'autres algorithmes de quantification vectorielle par la topologie introduite par la carte dans l'ensemble des prototypes. Cette topologie dépend du voisinage utilisé par l'algorithme et de la dimension de la carte. La plupart des implémentations de SOMs de la littérature utilisent un voisinage s'appuyant sur une grille en deux dimensions. L'indexation des nœuds est alors un ensemble de positions 2D. Des exemples de topologies en ligne et grilles sont présentées en figure 1.6.

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. La carte peut ainsi être indexée par des positions 1D, définissant un voisinage seulement sur une dimension, des positions 2D ou d'autres positions. En pratique, les grilles en deux dimensions sont les supports les plus couramment utilisés. Ces grilles permettent d'effectuer une réduction de dimension, tout en étant facile à visualiser [sur un écran]. 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 noeuds, et celui-ci augmente exponentiellement lorsqu'on augmente la dimension d'une carte de Kohonen. Les cartes une dimension sont quant à elles limitées 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 en (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 et al. 2016 ; Fort 2006) 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. ~~Donc, les processus intervenant dans des cartes 1D sont déjà mathématiquement difficiles à formaliser, difficulté qui augmente avec les dimensions.~~ L'étude des cartes 1D a ainsi l'intérêt d'envisager un modèle simplifié dans le cadre de développement d'un nouveau modèle de SOM, ce que nous chercherons à faire dans cette thèse, avant de proposer une extension aux cartes 2D.

er Pa
+ pédagogie théorique

Si Les cartes de forme autre que des grilles 1D ou 2D sont moins couramment utilisées, mais peuvent présenter des avantages. Ainsi, des cartes structurées en arbre telles que développées en (Koikkalainen et Oja 1990) permettant une recherche de BMU structurée. 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 (Alahakoon et al. 2000 ; Yamaguchi et al. 2010).

Par sa topologie, une carte de Kohonen permet d'extraire une représentation de l'espace d'entrée dans la répartition de ses poids sur son graphe. Cet algorithme se différencie ainsi

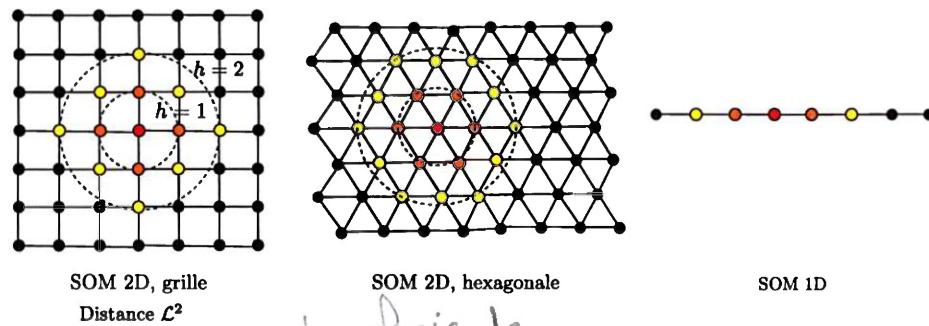


FIGURE 1.6 – Exemples de connexions dans le graphe support d'une SOM. Deux noeuds connectés sont ici à une distance de une unité dans la carte. Les SOM en deux dimensions sont les plus communément utilisées dans la littérature, sous forme d'une grille ou d'une grille hexagonale. Les SOMs une dimension sont parfois utilisées. La distance utilisée dans l'exemple de grille carrée est la distance euclidienne.

d'algorithmes de quantification vectorielle classiques tels que K-means.

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 autres aires. De nombreux travaux montrent la présence de cartes 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 1.7. L'aire associée à l'audition présente aussi une organisation topographique (Reale et Imig 1980), ainsi que de nombreuses autres aires, directement sensorielles ou plus abstraites (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 topologiquement correcte dans les cortex sensoriels, tel que le cortex visuel ou auditif.

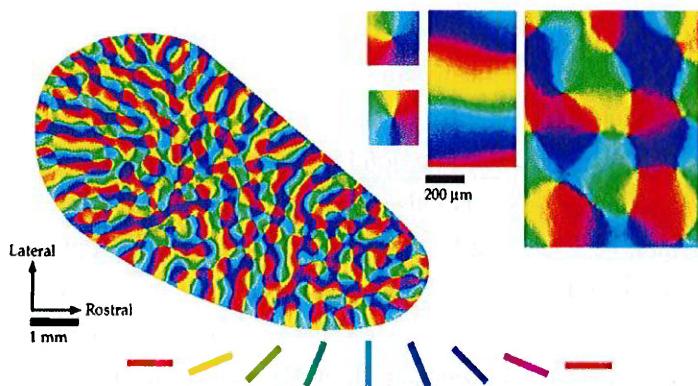


FIGURE 1.7 – Représentation des réponses du cortex visuel V1 à un stimulus visuel (bâtonnets d'orientations spatiales différentes). Les neurones répondant à une certaine orientation sont affichés de la même couleur. On observe une continuité entre les neurones proches dans le cortex et l'orientation à laquelle ils répondent. Cette propriété d'organisation est l'inspiration biologique des cartes de Kohonen.

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

L'aspect multisensoriel du traitement de l'information par le cerveau vient 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. Dans tous les cas, la présence d'aires distinctes communicantes fait l'objet d'un consensus. Nous pouvons citer comme exemple fonctionnel de traitement multisensoriel de l'information l'effet ventriloque (Bonath et al. 2007), qui correspond à l'illusion que la source sonore provient de la marionnette du ventriloque, dont la bouche s'agit de façon coordonnée aux paroles, alors qu'elle provient en réalité du ventriloque lui-même. Lors de cet effet, une activité apparaît dans le cortex visuel et auditif pour les neurones sensibles à l'emplacement exact de la source des stimuli dans chacune des modalités. Après quelques millisecondes, on observe une activité auditive pour les neurones sensibles à l'emplacement spatial de la source du stimulus visuel (Lefort 2012). Un autre exemple est celui de 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 le 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

→ élaborage de chez Mathieu pour une observation?

permettant le traitement de l'information multimodale. Si l'existence d'une hiérarchie du traitement de l'information par des aires haut niveau et bas niveau reste vérifiée, de nombreux travaux montrent également l'existence de connexions directes entre aires sensorielles, dites bas-niveau, illustrées en figure 1.8. Anatomiquement, 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). 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. (Burnod 1989) modélise le cortex en colonnes corticales et propose encore 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 [d'une carte de Kohonen] au niveau des connexions entre les aires cérébrales, se transcrivant par des connexions entre plusieurs cartes 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 en architecture.

1.4 Architectures de cartes auto-organisatrices

Plusieurs travaux dans la littérature informatique autour des SOMs cherchent ainsi à construire des architectures de cartes auto-organisatrices, que nous passons en revue dans cette section. Nous

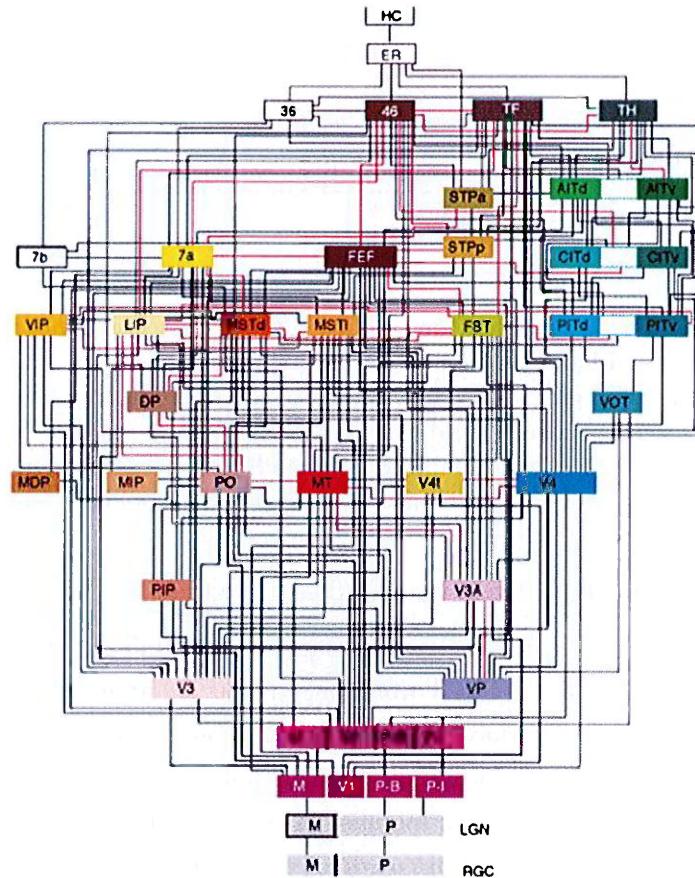


FIGURE 1.8 – Schéma de connexions entre aires sensorielles existant dans le cortex du singe, faisant apparaître des connexions rétroactives entre aires. Ce traitement fait apparaître plusieurs niveaux de hiérarchie tout en incluant des connexions entre aires d'un même niveau (Felleman et Essen 1991)

abordons ces modèles d'un point de vue structurel en nous intéressant notamment à l'interface entre les cartes dans chacun des modèles.

Il s'agit d'abord de différencier les architectures appliquées au traitement d'un problème particulier et les architectures génériques de cartes de Kohonen. Pour résoudre un problème compliqué, une démarche courante est de le décomposer en sous-problèmes puis de créer des structures pour résoudre ces sous-problèmes. L'assemblage des résultats de chaque structure donne alors une solution pour résoudre le problème général. Lorsque chaque structure est spécifiquement conçue pour résoudre un problème particulier, l'architecture est appliquée : elle est uniquement conçue pour résoudre le problème en question. Nous différencions cette vision d'architecture appliquée de la notion de modèle d'architecture. Dans cette deuxième approche, on entend par modèle d'architecture le cadre de calcul sous-jacent au modèle, appliqué ou non, défini par ses règles de construction. Ce cadre se veut générique et utilisable sur n'importe quelles données d'application.

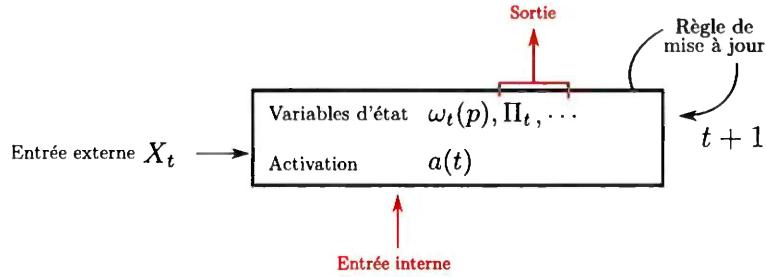


FIGURE 1.9 – Description de la notion de module d'une architecture modulaire. Un module prend des entrées externes à l'architecture ou contextuelles, et possède des variables d'état dont l'évolution dans le temps est régie par des règles de mise à jour ou équations d'évolution. La sortie d'un module est un ensemble de variable d'états. La sortie d'un module et l'entrée contextuelle d'une autre constituent l'interface entre modules. Les modules étudiés dans ce chapitre sont des cartes de Kohonen. Leurs variables d'état sont, entre autres, leur poids et leur BMU.

C'est cette deuxième approche que nous étudions dans cette thèse. *de ce Paul ? De ma j?*

Nous nous intéressons aux modèles d'architectures dits modulaires. Un module se définit comme un élément possédant des règles d'évolution à partir d'entrées, et une sortie. L'assemblage de ces modules par une interface définie constitue une architecture modulaire. D'après la définition de la modularité en informatique, des modules doivent pouvoir être ajoutés ou retirés à une architecture modulaire sans modifier la structure interne des autres modules de l'architecture. Le principe d'une architecture modulaire est alors d'assembler un nombre variable d'éléments de même type, des modules, dans une architecture. Ici les modules seront des cartes auto-organisatrices, dont le modèle a été adapté pour permettre la construction d'architectures. Un schéma d'une carte en tant que module est par exemple représenté en figure 1.9. Les éléments de l'architecture, les modules, n'ont pas de rôle défini a priori avant d'être insérés dans l'architecture, mais seulement des règles d'activation et de mise à jour. Les variables du module qui seront accessibles à d'autres modules de l'architecture sont les sorties du module. Dans le cas des SOMs, il s'agit par exemple de la position du BMU, des poids ou des valeurs d'activités. Les modules définissent par ailleurs leurs règles d'activation à partir d'entrées externes à l'architecture ou internes à l'architecture : les entrées contextuelles. Cette notion d'entrée et sorties internes à l'architecture constitue les règles d'interface entre cartes.

1.4.1 Méthode d'analyse

L'étude des architectures développées dans les travaux précédents nous amène à différencier les structures selon plusieurs aspects, résumés en figure 1.11. Nous les classerons d'abord selon leur structure hiérarchique ou non hiérarchique. Nous analyserons dans ce cadre leur mode de communication et les attributs servant de vecteur de transmission d'information entre modules. Nous noterons également leurs différences lors de la séquence de mise à jour dans un contexte

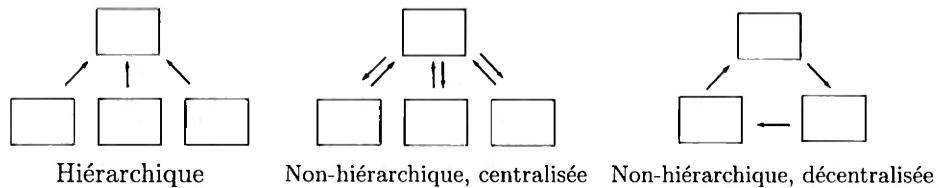


FIGURE 1.10 – Exemples de connexions dans des architectures hiérarchiques et non hiérarchiques centralisées et décentralisées. Un rectangle correspond à un module, ici une carte auto-organisatrice. Une flèche représente l'existence d'une interface entre deux modules : le module destination prend comme entrée contextuelle la sortie de la source et utilise donc de l'information de la source dans ses règles d'évolution.

d'apprentissage.

Architectures hiérarchiques et non hiérarchiques Nous distinguerons deux structures d'architectures de cartes : les architectures *hiérarchiques* et les architectures *non hiérarchiques*. La structure d'une architecture de cartes auto-organisatrice peut se représenter comme un graphe orienté, dans lequel les noeuds correspondent à un module, c'est-à-dire une carte. Lorsque la carte B utilise des informations provenant de la carte A lors de son apprentissage, une arête de A vers B est présente dans le graphe. Ce graphe est le graphe de connexion de l'architecture.

Une architecture est dite hiérarchique lorsqu'il n'existe pas de cycle dans le graphe de connexions. Dans ce cas, les arêtes sont orientées dans le même sens et on peut définir des niveaux de cartes. Toutes 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 non hiérarchique lorsqu'il existe au moins une boucle dans le graphe de connexions. 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 verrons qu'il existe deux paradigmes que nous détaillerons dans la section correspondante : les architectures centralisées et décentralisées, voir figure 1.10. *→ voir après 1.11*

Ces types d'architectures se ressemblent dans leur conception : une architecture hiérarchique est ainsi un cas particulier d'architecture non hiérarchique ; toute architecture non hiérarchique est un cas particulier d'architecture décentralisée. Une architecture décentralisée est alors le modèle le plus générique d'architecture modulaire. La construction d'une architecture plus générique amène des contraintes supplémentaires, notamment la gestion des rétroactions dans les méthodes de communication et d'apprentissage. Nous classerons les modèles du plus spécifique au plus générique.

Granularité du calcul et temporalité de l'apprentissage Pour chaque catégorie d'architecture, nous analyserons la façon dont les cartes interagissent. Nous différencierons les interactions gérées par une surcouche algorithmique de celles qui sont intégrées aux mécanismes d'organisation de chaque carte. La surcouche algorithmique se présente sous forme d'un algorithme ayant accès à l'état de l'architecture complète et permettant la sélection ou l'ajout de cartes pour l'apprentissage des données d'entrées. L'interaction entre les cartes générée par une telle surcouche est alors globale à l'architecture. L'interaction est au contraire interne à l'organisation d'une carte lorsque les données transmises sont directement prises en compte dans l'algorithme de mise à jour d'une carte. Ainsi, les cartes d'une architecture utilisant une interaction interne prennent en tant qu'entrée des éléments de sortie des autres cartes, tels que la position du BMU, son poids ou une activité neuronale. La communication réalisée de cette manière est ainsi locale à une carte de l'architecture. La construction d'architectures par la méthode globale nous intéresse peu dans un cadre d'architecture modulaire ; aussi, nous nous concentrerons sur les architectures dans lesquelles la communication est traitée à l'échelle d'une carte.

Au sein de cette échelle de connexion, les éléments transmis entre les cartes varient. Certaines architectures utilisent ainsi la position du BMU, son poids, ou encore un ensemble d'activités de neurones d'une carte. Nous relèverons les éléments de communication utilisés dans la littérature.

Enfin, la temporalité de l'algorithme de mise à jour de l'architecture peut se présenter de différentes façons. Nous distinguerons d'abord des architectures ayant une mise à jour séquentielle : l'architecture peut être décomposée en groupes d'éléments indépendants, comme les niveaux d'une architecture hiérarchique. Un apprentissage complet est d'abord effectué sur un groupe avant de passer à l'apprentissage des groupes suivants. Les mises à jour peuvent au contraire être réalisées en une seule étape, lors de laquelle les éléments seront tous mis à jour en tenant en compte des dépendances. Dans ces cas, ces mises à jour peuvent être synchrones : une itération d'apprentissage peut être définie de façon globale à toute l'architecture, lors de laquelle tous les éléments seront mis à jour au moins une fois. Les signaux lançant l'opération de mise à jour d'une carte sont régis par un processus extérieur aux cartes. Les mises à jour seraient asynchrones si elles sont effectuées [seulement] lorsqu'un signal déclenchant la mise à jour est transmis à un élément (carte ou neurone). C'est notamment le cas dans certains réseaux de neurones impulsifs, dans lequel les paramètres d'un neurone sont mis à jour seulement lorsqu'une impulsion lui parvient. Ce mode de mise à jour repose encore plus sur une notion de calcul local. Nous verrons ainsi que les modèles existants d'architecture de cartes se placent dans ces quelques catégories. L'analyse de ces modèles nous permettra de nous situer dans la littérature et d'émettre des hypothèses concernant les comportements attendus de l'architecture décentralisée que nous étudions dans cette thèse.

de quoi ?
le mode asynchrone ?
la m.a.j

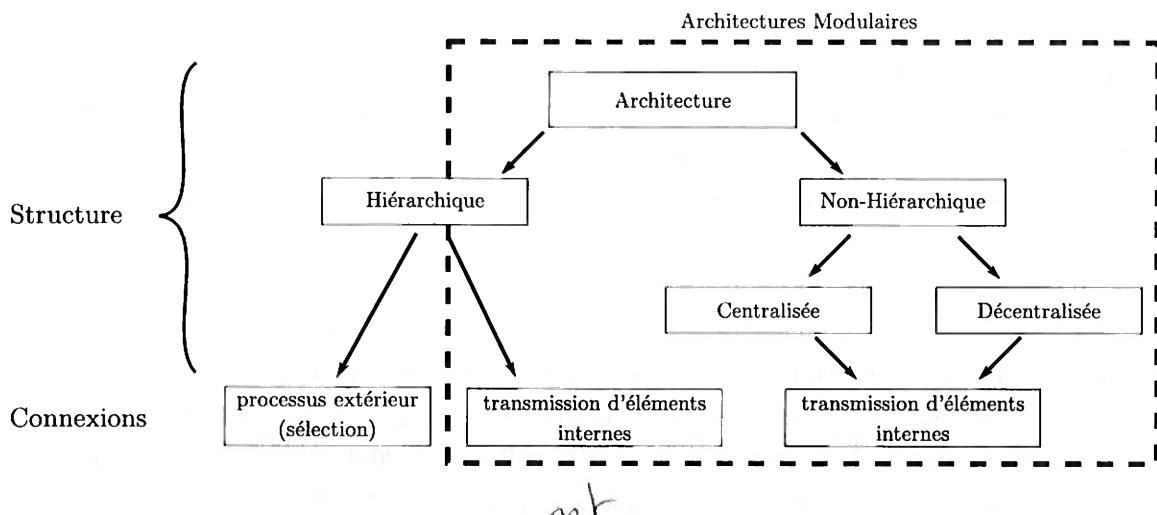


FIGURE 1.11 – Taxonomie des architectures de cartes présentées dans ce chapitre. Nous analyserons comment leurs caractéristiques structurelles : hiérarchiques ou non hiérarchiques, centralisées, décentralisées, façonne leur comportement d'apprentissage. Nous analyserons également leur interface de communication. Nous n'avons pas relevé d'architecture non hiérarchique s'appuyant sur le principe de sélection, car ce principe est fondamentalement hiérarchique.

1.4.2 Architectures hiérarchiques de cartes

Nous nous intéressons en premier lieu aux architectures hiérarchiques de cartes, c'est-à-dire sans rétroaction. Ces architectures se retrouvent également sous les termes de *Deep SOM* ou *SOM multicouches* : ces modèles se rapportent tous au même concept d'architecture hiérarchique de SOMs. Nous les divisons en deux catégories selon leur mode d'interface. Dans un premier cas, l'architecture s'appuie sur un processus extérieur aux cartes qui sélectionne et attribue les entrées 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, donc des représentations internes à l'architecture des entrées. Nous présenterons les équations des différents modèles dans le formalisme introduit en section 1.2. Par facilité de représentations, 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. Le lecteur peut également se référer à (Henriques et al. 2012) pour une revue appliquée des méthodes de clustering basées sur des SOMs hiérarchiques. Nous abordons de notre côté le sujet d'un point de vue plus structurel qu'appliqué, comparant l'aspect modulaire et l'information transmise dans les cartes.

Architectures hiérarchiques par sélection

Nous appelons une architecture par sélection un ensemble de cartes organisées de façon hiérarchique et dont la sortie d'une carte permet de diviser l'espace d'entrée en sous-espaces qui

seront utilisés par le niveau de cartes suivant. Détaillons par exemple l'architecture développée en (Barbalho et al. 2001), représentée en figure 1.12. Ce modèle est également décliné en une version dynamique dans laquelle les paramètres des cartes dépendent des données, en (Costa et al. 2016). 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 tel 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 carte peut être définie à l'avance comme en (Barbalho et al. 2001) ou de façon incrémentale en s'adaptant aux données, comme en (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 ici une architecture du même type mais pour traiter des données à structure hiérarchiques, ici des phrases écrites. La structure est similaire : une carte d'un premier niveau prend une entrée qui est une phrase 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, en retirant les dimensions communes à toutes les entrées d'un sous-ensemble. 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 structure hiérarchique 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 BMUs 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 en un ensemble de cartes apprenant sur des sous-groupes de données détectées dans l'espace d'entrée. Cette décomposition peut permettre la découverte de structures hiérarchiques dans les données d'entrées. Le premier niveau seul est alors une représentation générale de l'espace d'entrée. L'augmentation et la séparation des cartes 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.

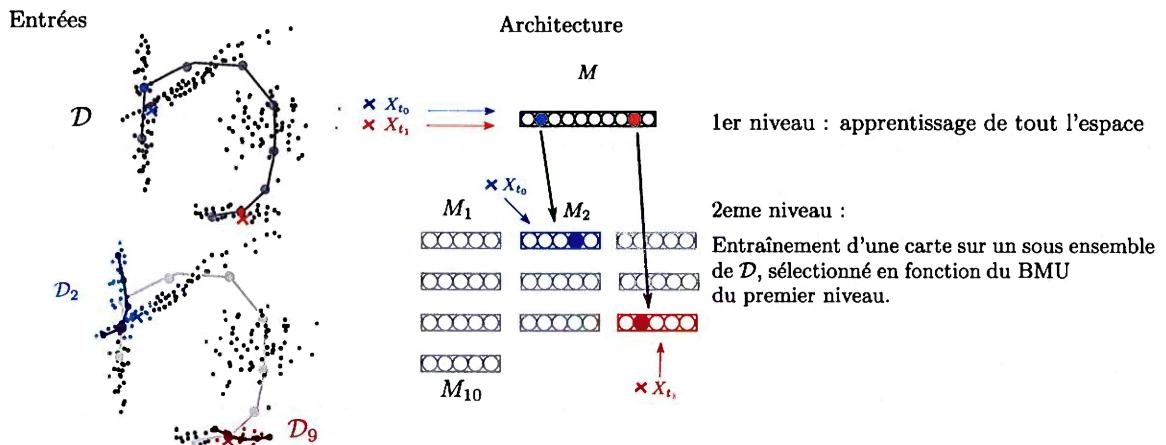


FIGURE 1.12 – Exemple d’architecture hiérarchique sélective. La carte du premier niveau est entraînée sur tout l’espace d’entrée. Après apprentissage, la carte permet de filtrer les entrées pour [les renvoyer vers] une carte du niveau suivant. Dans cet exemple, la position du BMU de la carte du niveau 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. La carte M est représentée sur les données d’entrée (disposition exemple, non générée par une simulation). Le sous espace D_1 – lié au BMU à la position 1 alimente alors une carte du deuxième niveau M_1 .

Ces travaux montrent ainsi 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 permet 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 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 ou EM. *Quelles sont les différences entre ces algos ?*

- ❖ Nous ne considérons pas ces architectures comme modulaires. Dans ce type d’architecture, c’est 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. [Dans ce sens, on ne peut pas parler d’architecture modulaire de par l’aspect global de l’algorithme.] *→ nedire ?*

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. Cette gestion des connexions, qui était collective dans les architectures sélectives, devient locale : 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, la deuxième couche de carte~~s~~ 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 une intensité d'activité neuronale. Ces éléments sont une représentation latente de l'entrée transmise à la couche supérieure.

Des premiers travaux proposant un modèle de SOM hiérarchique apparaissent chez (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. Les travaux ne sont pas applicables seulement aux SOMs mais à tout algorithme de quantification vectorielle. Les auteurs utilisent ici le poids du BMU comme interface entre les SOMs.

Parmi ces premiers travaux, 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 , et une deuxième carte $M^{(2)}$ vient ensuite prendre comme entrée la position du BMU $\Pi^{(1)}$ de la première carte ; cette architecture est illustrée en figure 1.13. Le BMU de la première carte est alors défini par :

$$\text{ne quel ?} \quad \Pi^{(1)} = \arg \min_p (\|\omega^{(1)}(p) - X\|^2)$$

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 SOMs doivent automatiquement extraire des groupes de données (*clusters*) par similarité, sachant que les données sont effectivement regroupées en classes distinctes. Ces clusters sont définis après apprentissage par les cellules de Voronoi dont les poids de la carte 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 BMUs proches dans la première carte ; par conséquent, leurs BMUs 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~~clusters~~ seront nécessaires pour couvrir une classe entière. Au contraire, la deuxième couche de SOMs génère des clusters plus larges, et moins de

1.4. Architectures de cartes auto-organisatrices

ces clusters seront alors nécessaires pour couvrir une classe de données. Le fait d'utiliser une architecture de SOMs permet ainsi d'extraire une représentation différente que celle extraite par une SOM classique.

L de ?

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 et Helbich 2013; Paplinski et Gustafsson 2005). Dans leurs travaux, les auteurs implémentent une architecture en arbre. La carte du niveau i , $M^{(i)}$ reçoit alors un vecteur de position de BMUs de k 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.

Un point clé dans la construction de ces architectures repose sur l'information transmise entre couches. Cette information 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 totalement l'aspect topologique qu'offrent les cartes de Kohonen. Cette information est par ailleurs relative à une carte et non un type d'entrée ce qui en fait une interface très générique. Enfin, il s'agit d'une position 1D ou 2D, donc une information peu coûteuse à utiliser. Le poids du BMU apparaît également comme une méthode de transmission d'information au sein des SOMs hiérarchiques (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 imagettes 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 imagettes 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.

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 (Le-Cun 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.

Par exemple, le modèle D-SOM introduit en (Liu et al. 2015 ; Wickramasinghe et al. 2019) est illustré en figure 1.14. 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 $N \times N$ imagettes de position fixée. La première couche du réseau comporte $N \times N$ cartes, donc chacune prend en entrée l'imagette 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 BMUs des SOMs : ce modèle de transmission rejoint ainsi ceux présentés dans HSOM. La différence apparaît au niveau du prétraitement de l'entrée image, décomposée en imagettes.

- ✓ 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 classe les images générées par la dernière couche, elle classe une représentation plus abstraite de l'image d'entrée. L'étude montre ainsi que la classification de cette dernière image permet de bien retrouver les classes présentes dans les données d'entrées. L'architecture D-SOM 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 donc très similaire à HSOM.

Toutes les architectures présentées ici, qu'il s'agisse des SOMs hiérarchiques ou Deep SOM,



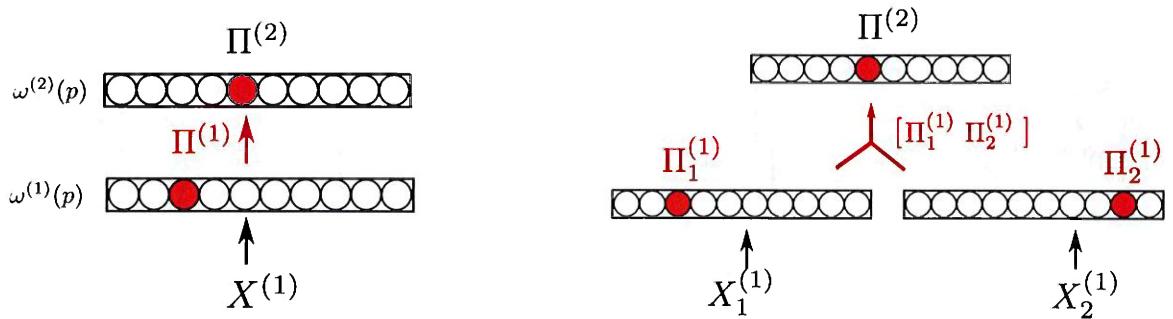


FIGURE 1.13 – Deux exemples d’architectures basées sur HSOM. A gauche, le modèle HSOM original proposé en (Lampinen et Oja 1992). L’apprentissage des positions du BMU de la première couche par la seconde permet de mieux détecter les ensembles de données présents dans la distribution des X (*clustering*). 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.

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 donne de meilleures performances en termes d’erreur de classification.

Discussion

Nous avons distingué deux catégories d’architectures hiérarchiques en s’appuyant sur le mode de connexion entre cartes dans l’algorithme de mise à jour. Le premier mode repose sur la sélection d’une carte du niveau supérieur en s’appuyant sur la réponse des cartes du niveau courant à une entrée, afin de transmettre cette entrée à la carte supérieure. Ce mode de transmission permet de créer un ensemble de cartes s’organisant sur un même espace d’entrée, laissant plus de possibilités au déploiement des cartes qu’une SOM classique. Ces types d’architectures permettent notamment d’améliorer la qualité de la quantification vectorielle générée par une SOM classique et d’ajouter des noeuds de façon peu coûteuse. Le mode de transmission par sélection s’appuie sur 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 cartes est gérée par un processus global.

Le second mode de connexion 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. Ce mode de connexion est intégré au processus d’auto-organisation et non une couche algorithmique

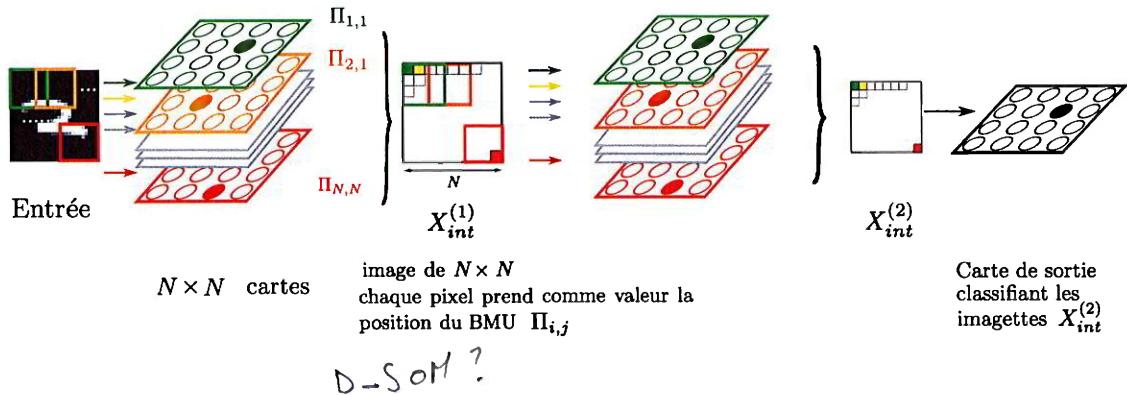


FIGURE 1.14 – Architecture DSOM de SOM « convolutive » (Liu et al. 2015). Les auteurs utilisent les positions des BMUs $\Pi_{p,q}$ d'une couche de carte, disposés dans un tableau 2D, comme valeurs d'entrée pour les couches suivantes. Ces couches sont entraînées les unes après les autres.

supplémentaire comme les architectures par sélection, ce qui localise le traitement des connexions à l'échelle d'une carte. Cette méthode de construction d'architectures s'appuie sur la transmission d'une représentation de l'entrée interne entre cartes. Ce type de connexions nous intéresse d'un point de vue modulaire : il 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 toutes sur ce mode de transmission. 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 entre cartes. Il permet aussi 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. Le poids du BMU est moins spécifique à une carte de Kohonen, peut être de dimension élevée et donc plus coûteux en termes de calcul dans une carte. Enfin, si les cartes de l'architecture prennent des entrées dans des espaces différents, cette représentation n'est pas homogène. Nous reviendrons sur ces modes de représentations en analysant les architectures non hiérarchiques.

Qu'elles soient sélectives ou par transmission de représentation, toutes les architectures relevées ici ont une séquence de mise à jour ascendante : 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 classification pouvant être réalisée par une SOM. Par exemple, les SOMs par sélection permettent d'améliorer la quantification vectorielle sur l'espace d'entrée ou de prendre en compte l'aspect hiérarchique des entrées.

1.4. Architectures de cartes auto-organisatrices

Les SOMs 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 n'ont pas la capacité de faire d'autres types de calcul que ceux originellement réalisés dans une SOM. 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 d'une SOM classique, notre attention se portera donc 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 SOMs en apportant un aspect dynamique au système 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.

→ Rappelons ?

1.4.3 Architectures non hiérarchiques de cartes auto-organisatrices

Les architectures non hiérarchiques de SOMs 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 raisons derrière 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 de cause à effet entre les activations des différentes aires, 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 en effet dans le paradigme d'*embodiment* (*Embodied Cognition*), c'est-à-dire de développer 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). L'inventaire 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. Toutes ces architectures non hiérarchiques ont en commun leur champ d'application : contrairement aux architectures hiérarchiques ascendantes qui cherchent à améliorer les performances de classification ou de *clustering* d'une SOM classique, les SOM non hiérarchiques que nous avons relevées dans la littérature sont plutôt appliquées à des tâches de *mémoire associative* sur des données *multimodales*. Ces cartes sont des systèmes dynamiques par leurs rétroactions et ont la capacité de générer une valeur de sortie de façon autonome. Elles sont alors utilisées pour prédire une modalité à partir d'une autre, ce qui correspond à une tâche de mémoire associative.

Nous avons pu distinguer deux structures principales d'architectures non hiérarchiques dans les travaux réalisés jusqu'à présent. 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 contextuels. 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 : en effet, les cartes associatives forment un niveau d'apprentissage différent des cartes sensorielles, apportant une hiérarchie dans l'apprentissage. Nous les classons ici dans la catégorie non hiérarchique. 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 ascendantes que nous avons listées au paragraphe précédent. Le second type d'architectures non hiérarchiques sont celles [utilisant des connexions directes entre cartes sensorielles]. Ces architectures sont *décentralisées* : il n'existe pas de module par lequel toute l'information transite.

↳ concerne les archi non hiérar.

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). Biologiquement, notre comportement est multisensoriel, influencé par toutes les sources d'informations dont nous disposons.

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 multimodale est alors de concilier tous ces aspects [au sein d'un même algorithme d'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). Alors que la perception de la parole pourrait être uniquement auditive, les psychologues montrent que la présentation du son "ba" à un sujet associée à la présentation d'une vidéo d'une bouche prononçant "ga" amènent le sujet à indiquer qu'il a entendu le son "da". 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 [d'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 ; les autres cartes de l'architecture n'ont accès qu'à une information filtrée et compréhensible par leurs règles d'évolution

internes.

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 activation étant lié à des poids représentant la modalité, il est alors possible de prédire une valeur pour la modalité manquante. 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 fixé. 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 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

1.4. Architectures de cartes auto-organisatrices

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 1.15. 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 l'autre les mouvements du bras. Chaque carte du premier niveau possède une couche de poids ω_e liées aux entrées sensorielles 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. La recherche du BMU est réalisée en prenant en compte une activation dans la carte. Les poids ω_e sont ensuite gelés, et la carte associative est ensuite entraînée pour apprendre à associer les positions des BMUs correspondant aux cartes modales :

$$\begin{cases} \Pi^{(1)} = \arg \max_p a(X^{(1)}, \omega_e^{(1)}(p)) \\ \Pi^{(2)} = \arg \max_p a(X^{(2)}, \omega_e^{(2)}(p)) \\ X^{(m)} = [\Pi^{(1)}, \Pi^{(2)}] \end{cases} \quad (1.1)$$

Enfin, les couches des cartes modales ω_c dédiées aux connexions sont mises à jour en répétant les étapes de l'équation 1.2 :

$$\begin{cases} \Pi_c^{(1)} = \arg \max_p a(\Pi^{(m)}, \omega_c^{(1)}(p)) \\ \Pi_c^{(2)} = \arg \max_p a(\Pi^{(m)}, \omega_c^{(2)}(p)) \end{cases} \quad (1.2)$$

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 décorrélées. La rétroaction entre les cartes n'en est donc pas une lors de la phase d'apprentissage : 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 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 :

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

a. ²⁷ activation associative ?
m?

La valeur $\omega_e^{(1)}(\Pi^{(1)})$ 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 1.16. 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 également 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 donc 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. Les connexions hebbiennes pondérées entre neurones sont en fait équivalentes 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 $\{a_i, i = 0..N\}$. Ces connexions sont équivalentes à l'ajout d'une couche de poids supplémentaire à la carte modale telle que : $\forall j, \omega_j \in [0, N]$ représente un champ d'activation de la carte associative.

L'information transmise dans SOIMA repose ainsi 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 décorrélées 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 privilie 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

1.4. Architectures de cartes auto-organisatrices

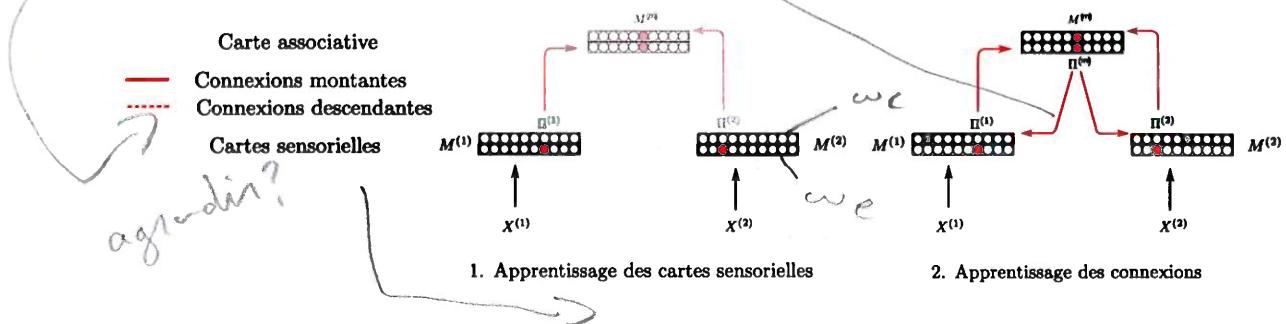


FIGURE 1.15 – L’architecture MMCM (Lallee et Dominey 2013) est une architecture centralisée. Les cartes du premier niveau sont les cartes modales $M^{(1)}, M^{(2)}$ qui reçoivent l’une les mouvements de tête d’un robot, une autre les mouvements de son bras. Une carte associative $M^{(m)}$ reçoit les positions des BMUs $\Pi^{(1)}, \Pi^{(2)}$ de chaque carte du premier niveau en tant qu’entrées.
et les entrées descendantes ?

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, formant une boucle de rétroaction potentiellement infinie. Pour résoudre ce problème, l’architecture MCMM 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. 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, à partir du moment où elles sont placées dans l’architecture, avant même l’apprentissage, elles prennent des rôles différents : des cartes sont associatives et des 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’architecture 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 [en architecture]. 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

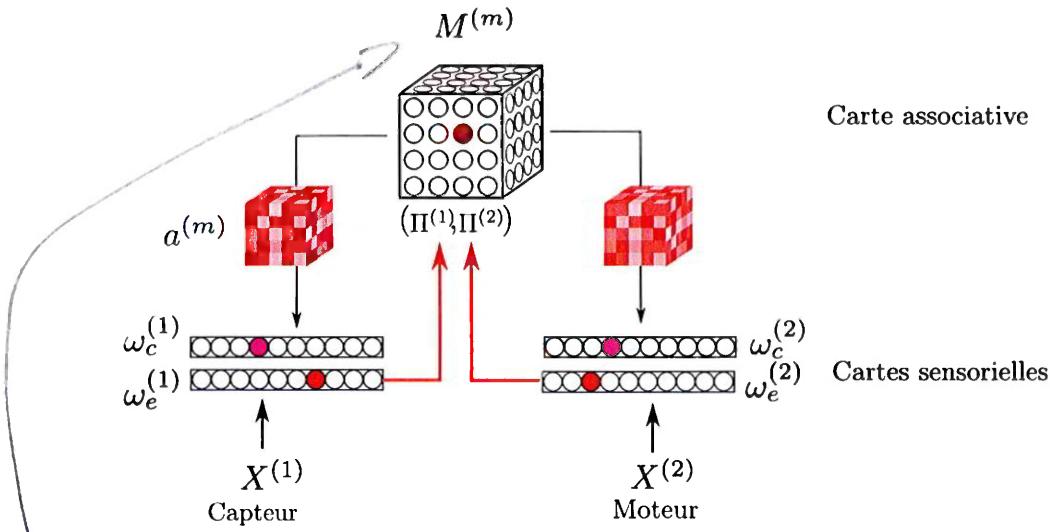


FIGURE 1.16 – Le modèle SOIMA (Escobar-Juárez et al. 2016) propose une architecture centralisée dans laquelle des cartes modales $M^{(1)}$ et $M^{(2)}$ sont connectées par une carte associative $M^{(a)}$. Cette carte associative prend comme entrée les BMUs des cartes modales. Les connexions descendantes sont gérées par la transmission du champ d'activation de $M^{(a)}$ vers les cartes modales. Les poids de ces connexions sont stockés dans une deuxième couche de poids.

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.

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 du mode de 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, par exemple (Buonamente et al. 2016). A-SOM est illustré en figure 1.17 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 . 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. L'interface entre cartes utilisée en A-SOM est donc le champ d'activation des neurones. 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 $w_c(i) = [\omega_{c11}, \dots, \omega_{c1N}]$. Lors de l'apprentissage, la mise à jour des poids ω_e et ω_c est réalisée de manière indépendante. Le BMU de position II 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 l'échelle d'une carte et non du neurone, à la règle Hebbienne qui renforce les connexions de deux neurones s'activant en même temps. 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.

Les travaux menés précédemment dans notre équipe se sont attachés à la création d'architectures décentralisées de cartes auto-organisatrices. Les auteurs se sont initialement appuyés sur des modèles de cartes auto-organisatrices cellulaires, les opérations s'effectuant ainsi à l'échelle même d'un nœud d'une carte. Une entrée X est présentée à tous les neurones d'une carte. La notion de BMU se définit alors par le neurone s'activant le plus fortement à la présentation d'une entrée, inhibant alors les autres neurones de la carte grâce à des calculs de champs neuronaux dynamiques. Nous pouvons souligner le parallèle existant entre la version cellulaire et la version centralisée d'une carte de Kohonen. Les champs neuronaux dynamiques implémentent [en effet] un mécanisme de *Winner Take All* similaire au choix du BMU par argmax dans une carte classique. Dans nos travaux, nous étudions les cartes de Kohonen classiques, mais les mécanismes occurant à l'échelle d'une carte dans une carte de Kohonen cellulaire ressemblent à ce qu'on observe dans une carte classique. Ainsi, l'architecture Bijama développée en (Ménard et Frezza-Buet 2005) et

✓ ✎

ref?

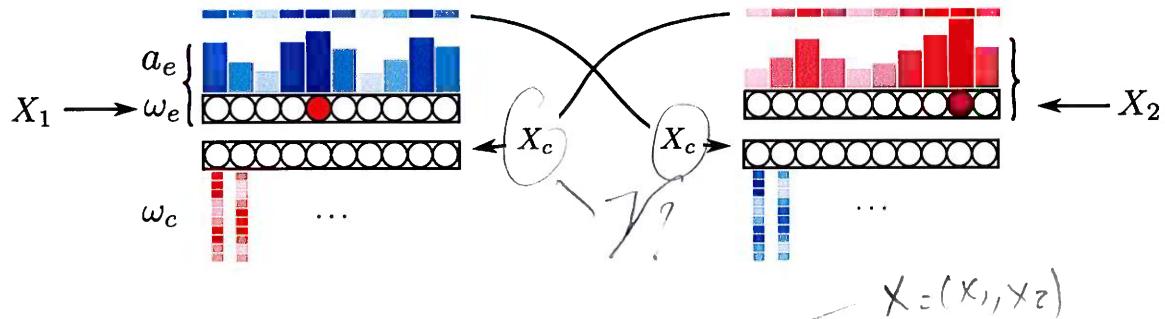


FIGURE 1.17 – Le modèle A-SOM (Johnsson, Balkenius et Hesslow 2009) associe les activités de différentes cartes. Chaque ASOM prend une entrée modale X_1 et X_2 . Chacune des cartes possède deux couches de poids, une couche ω_e associée aux entrées modales et une couche ω_c associées aux entrées γ venant 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.

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. 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 le modèle Bijama, présenté en figure 1.18, chaque neurone d'une carte est connecté via sa couche 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 directe 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 1.19). 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 rétroactions dans le calcul d'activité de chaque carte, utilisant le même mécanisme de relaxation.

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 1.20, 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 la position du BMU Π_2 de la deuxième carte. La seconde carte reçoit deux entrées de la première : une entrée est la position du BMU $\Pi_1(t-1)$ de l'état précédent et la seconde la position du BMU $\Pi_1(t)$ de l'état courant. Une activité a , gaussienne, est calculée sur chaque couche de poids relativement à son entrée et ces deux activités sont fusionnées en une activité globale à chaque carte :

$$\begin{cases} a^{(1)}(p) = a(\Pi_t^{(2)}, p) + a(X_t, p) \\ a^{(2)}(p) = a(\Pi_t^{(1)}, p) + a(\Pi_{t-1}^{(2)}, p) \end{cases}$$

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 alors relaxer les activités en déplaçant petit à petit les BMUs 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 poids. Ce modèle permet alors d'apprendre des séquences d'entrée. Alors qu'une carte simple différencierait les BMUs en fonction de la valeur de l'entrée, ce modèle génère une différenciation des BMUs 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 plus pure 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

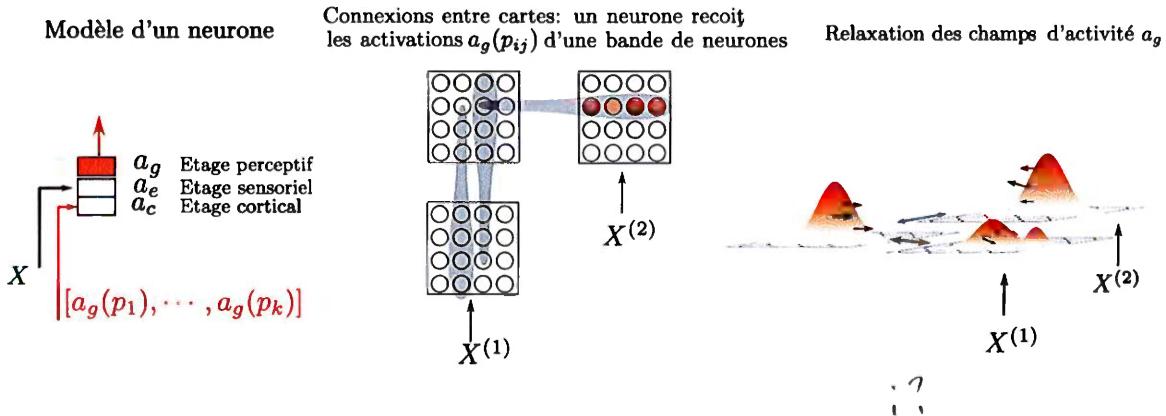


FIGURE 1.18 – Exemple d'architecture construite avec le modèle Bijama (Ménard et Frezza-Buet 2005). Chaque neurone d'une carte reçoit une entrée sensorielle $X^{(1)}$ et des entrées corticales, indiquées figure de gauche. Dans cet exemple, deux cartes reçoivent des entrées sensorielles et corticales, 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.

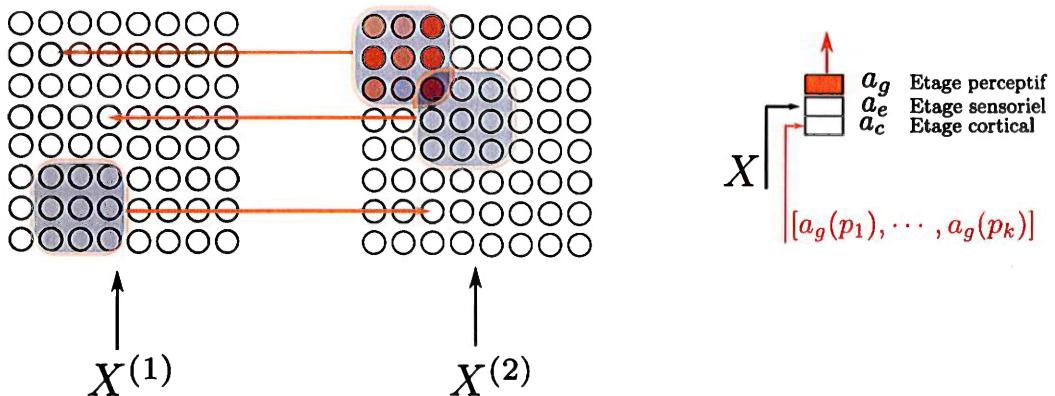


FIGURE 1.19 – Le modèle SOMMA (Lefort et al. 2011; Lefort 2012) associe les activités de différentes cartes, mais en réduisant les champs d'activité transmis aux neurones entourant le neurone situé en position courante. À gauche sont représentées les connexions entre les neurones de deux cartes, dans 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.

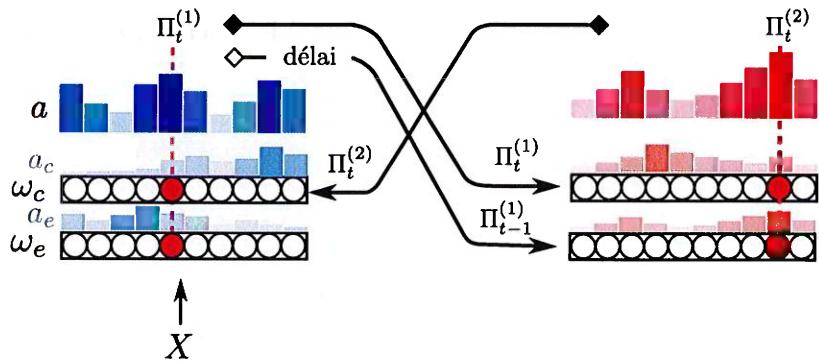


FIGURE 1.20 – Structures de deux cartes auto-organisatrices communicantes, (Baheux et al. 2014). Chaque carte est composée de trois couches d’activités, représentées séparément sur le schéma : sur la première carte, une activité est relative à l’entrée X , l’observation. L’autre activité reçoit une entrée 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 BMUs, 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.

cas, les modèles de connexions entre cartes sont différenciés.

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 BMUs, 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. [Cette déconnexion relève plutôt de l’apprentissage automatique que d’une inspiration biologique] Les travaux menés dans notre équipe, bien que les neurones soient toujours structurés en couches, ont fait le choix de ne pas différencier les activités des couches lors de l’apprentissage. 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 activités évoluer, en tant que système dynamique, vers un état stable grâce aux règles de calcul d’activités. 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. 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 Connexions temporelles et architectures de cartes auto-organisatrices

L'apprentissage de séquences temporelles
Données temporelles?
? une autre problématique liée aux arch...

La capacité de traitement de séquences est un des enjeux de l'application des architectures de cartes. Nous avons également relevé des architectures cherchant à 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 1.21 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 l'enjeu de la multimodalité en robotique développementale incluant 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 taille fixe de données, sous forme de fenêtre temporelle. Une autre solution au contraire 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, implantée dans de nombreux modèles d'apprentissage, se rapproche de la notion de transmission d'information entre modules d'une architecture, les modules étant ici les états de la carte 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, destiné à l'apprentissage de séquences ont ainsi été proposés dans la littérature.

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



FIGURE 1.21 – 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 commune est la classification de mouvements. La séquence "tirer" sera différente de la séquence "marcher".

Cartes auto-organisatrices récurrentes

Les modèles de cartes récurrentes existant dans la littérature s'appuient sur la transmission de représentation interne entre itérations. L'information transmise entre ces états rejoint les mécanismes relevés dans les architectures multi-cartes : transmission du BMU en tant qu'entrée, 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.

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

avec a équivalent ici à une distance modifiée entre un prototype et l'entrée. Le BMU est alors calculé par $\Pi_t = \min(\|a(p, t)\|^2)$. Le calcul de l'activité revient donc, au lieu de chercher la position donc le prototype $\omega(p)$ est le plus proche de l'entrée X_t , 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 où l'influence de l'activité d'un instant sur les suivants décroît au cours du temps.

- D'autres travaux reposent sur la transmission d'une information en tant qu'entrée, prise en compte dans le calcul de l'activité. A chaque instant t , ces SOMs 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 SOMs* 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. A 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.

$$\gamma_t = \lambda \gamma_{t-1} + (1 - \lambda) \omega(\Pi_{t-1})$$

Enfin, le modèle SOMSD, 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 1.22. 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

1.4. Architectures de cartes auto-organisatrices

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 assemblant 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.

Une carte ?

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) utilisant 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 proviennent alors des activités des neurones de cette même carte au pas de temps précédent. Le modèle à base de cartes auto-organisatrices classiques proposé en (Baheux et al. 2014) que nous avons décrit plus haut s'inscrit également dans un cadre de traitement de séquence. Le modèle montre qu'une carte de Kohonen au sein d'une architecture récurrente permet de différencier les BMUs 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 existante entre récurrence et multimodalité et la possibilité d'inclusion de connexions récurrentes au sein d'un modèle multi-cartes.

Est-ce normal cette écriture bizarre des rep's ?

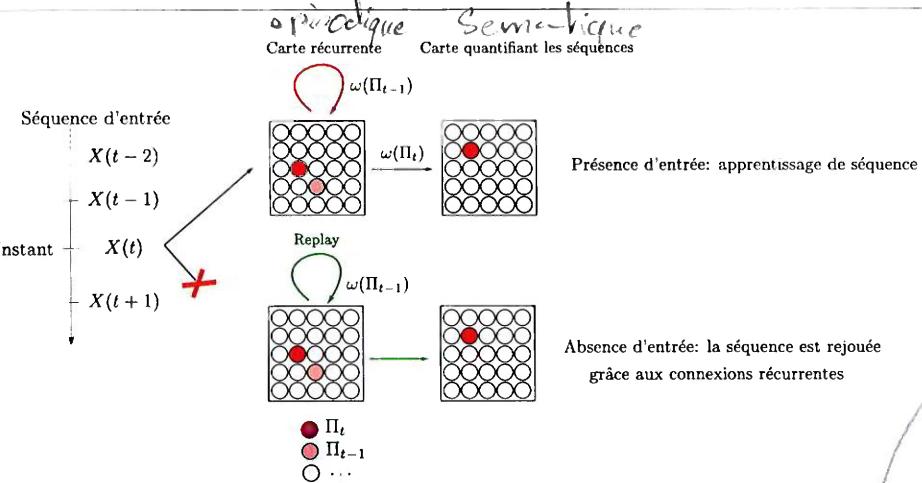


FIGURE 1.22 – Architecture à double mémoire proposée en (Parisi et al. 2018). La couche de mémoire épisodique, permettant la différenciation de séquences, prend en entrée externe un instant de la séquence d'entrée X et en entrée de contexte le poids du BMU de l'instant précédent. La couche de mémoire 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 cette couche de carte 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.

1.5 Axe de recherche

La littérature portant sur les architectures de SOMs se divise donc 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 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 de structure. Ces travaux se placent plutôt dans 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 rétroaction permettent l'activation d'une carte par une autre et apportent [alors] aux SOMs une capacité de prise de décision et

de prédiction lorsqu'elles sont au sein d'une architecture. Ces architectures non hiérarchiques 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 : la 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 séquences, se rapprochent par ailleurs des architectures multi-cartes par leur structure s'appuyant sur une transmission d'information, cette fois entre une même carte à deux pas de temps. Une architecture modulaire générique de cartes doit 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. Le tableau 1.1 présente ainsi 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 SOMs 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. Seuls les travaux de (Lallee et Dominey 2013) s'appuient sur transmission de la position du BMU dans le cadre des SOMs non hiérarchiques. Les architectures décentralisées sont en effet 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. Or, le concept de BMU et de processus *Winner Take All* intervenant dans une carte auto-organisatrice permet en fait de remplacer les champs neuronaux, proches de la biologie, par un mécanisme global à la carte. Ce mécanisme s'éloigne de la plausibilité biologique mais permet des calculs plus rapides dans une SOM, tout en conservant les propriétés d'auto-organisation d'une carte. La transmission du BMU entre cartes au lieu d'activités apparaît donc comme un pendant plus informatique de la transmission d'activité.

Le travail de recherche proposé dans cette thèse consiste à construire et étudier un modèle d'architecture de cartes bio-inspiré. Nous cherchons à nous placer plutôt du côté du calcul informatique, laissant la biologie comme une inspiration et non en cherchant à la modéliser. Par cette inspiration biologique, il nous apparaît pertinent d'explorer la construction d'architectures non hiérarchiques décentralisées de cartes de Kohonen. 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 choisirons de diriger nos recherches dans ce sens. Nous proposerons donc une architecture non hiérarchique décentralisée de cartes de Kohonen, dont les mises à jour se font de façon synchrone. Les travaux que nous avons réalisés portent sur la définition et l'analyse des comportements générés par un tel modèle. Nos travaux font ainsi suite aux travaux amorcés en (Baheux et al. 2014) sur des architectures récurrentes multimodales utilisant la transmission de la position du BMU entre des cartes de Kohonen et utilisant un mécanisme de relaxation pour gérer les rétroactions entre cartes. Ces travaux exploitent des connexions intercartes mais restent appliqués au traitement de séquences. Nous explorerons donc la construction d'architectures multi-cartes et continuerons le développement de ce modèle mais dans un cadre multimodal, afin de comprendre d'autres mécanismes intervenant dans une telle architecture. 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 le traitement de données multimodales dans de petites architectures de deux et trois cartes. Le but de cette thèse est d'identifier et de formaliser les comportements d'apprentissage générés par une petite structure non hiérarchique et de développer une méthode expérimentale et des représentations adaptées du modèle, dans une optique de création d'architectures comportant de nombreuses cartes.

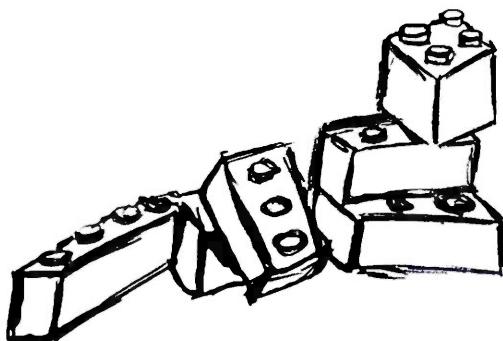


TABLE 1.1 – Comparaison des principaux modèles d'architectures relevés dans ce chapitre. Nous n'y faisons pas figurer les architectures sélectives, étant non modulaires. Les modèles très similaires sont regroupés sur une seule ligne. Les cartes récurrentes ne sont pas concernées par la question de séquence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

Modèle	Type	Mise à jour	Mode de transmission
HSOM	Hiérarchique	Séquentielle	Position du BMU
Deep SOM	Hiérarchique	Séquentielle	Position du BMU
HSOM - poids du BMU	Hiérarchique	Séquentielle	Poids du BMU
MMCM	non hiérarchique, Centralisée	Séquentielle	Positions BMU
SOIMA	non hiérarchique, Centralisée	Séquentielle	Champ d'activité
Bijama	non hiérarchique, Décentralisée	Asynchrone	Champ d'activité partiel
A-SOM	non hiérarchique, Décentralisée	Synchrone	Champ d'activité
SOMMA	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	Récurrente		Champ d'activité
MSOM	Récurrente		Poids du BMU
A-SOM	Récurrente		Champ d'activité
Recursive SOM	Récurrente		Champ d'activité
SOMSD	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

- Menquez
les n° ?
- 1 (Lampinen et Oja 1992; Hagenauer et Helbich 2013)(Paplinski 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)

