

# Chapitre 1

## Architectures de cartes auto-organisatrices

### Sommaire

---

<b>1.1</b>	<b>Les cartes auto-organisatrices de Kohonen comme modules d'une architecture . . . . .</b>	<b>2</b>
1.1.1	Description du modèle de carte auto-organisatrice de Kohonen . . . . .	2
1.1.2	La carte de Kohonen, un algorithme d'apprentissage de représentation non-supervisé . . . . .	3
<b>1.2</b>	<b>Inspiration biologique des architectures de cartes . . . . .</b>	<b>7</b>
1.2.1	Inspiration biologique des cartes de Kohonen . . . . .	7
1.2.2	Rétroactions dans le traitement de l'information multisensorielle du cortex . . . . .	7
<b>1.3</b>	<b>Architectures de cartes auto-organisatrices . . . . .</b>	<b>10</b>
1.3.1	Méthode d'analyse . . . . .	10
1.3.2	Architectures hiérarchiques de cartes . . . . .	12
1.3.3	Architectures non-hiérarchiques de cartes auto-organisatrices . . . . .	20
1.3.4	Connexions temporelles et architectures de cartes auto-organisatrices . . . . .	28
<b>1.4</b>	<b>Axe de recherche . . . . .</b>	<b>32</b>

---

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 de données ou de classification. Cependant, peu de travaux ont exploré leur aspect topologique et la simplicité de leurs règles de calcul pour les assembler en architectures modulaires, formant un système dynamique et générant d'autres comportements que ceux d'apprentissage automatique décrits ci-dessus. Une motivation de création d'une telle architecture se place par exemple au niveau de l'apprentissage développemental. Une telle motivation est ainsi formulée par Kohonen en 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)

Bien que de nombreux travaux en apprentissage automatique aient fait avancer le domaine de l'apprentissage développemental depuis ces 30 dernières années, peu de travaux ont exploré la création d'architectures autonomes de cartes auto-organisatrices. La simplicité des règles de mise à jour de ces algorithmes leur confère pourtant un intérêt computationnel digne de ne pas être oublié. Nous étudierons dans cette thèse un modèle de création d'architectures modulaires décentralisées de cartes auto-organisatrices. Le but de cette thèse est ainsi d'une part, de proposer un modèle de carte qui puisse être utilisée en tant que module et de définir l'interface entre modules et d'autre part d'en extraire des comportements de calcul en émergent. Avant toute chose, nous pouvons chercher des éléments de réponse à ces deux questions en étudiant les travaux traitant des architectures de cartes de Kohonen ou plus généralement de réseaux d'apprentissage utilisant des règles d'auto-organisation dans leur évolution. Cela nous permettra de privilégier un type d'interface au regard des travaux existants et d'émettre des hypothèses quant aux comportements que nous pouvons attendre de l'architecture que nous proposerons.

Nous présentons dans ce chapitre un modèle général d'une carte de Kohonen et ses comportements fondamentaux. Nous passons ensuite en revue différentes architectures de SOM proposées dans la littérature. Nous listerons les différents types de structure que nous pouvons relever et analyserons leur aspect modulaire et les comportements d'apprentissage générés par ces structures. Nous comparerons aussi le choix du modèle d'interfaces entre cartes. A l'issue de ce chapitre, nous aurons une vue d'ensemble organisée de différents modèles d'architectures de SOMs existantes et définirons où se place le modèle que nous étudierons.

pariel ?  
qui en émerge ?

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

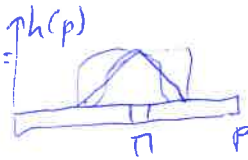
Le modèle de cartes auto-organisatrice a été initialement développé par Kohonen (Kohonen 1982); nous utiliserons les termes cartes de Kohonen et SOM de façon équivalente pour désigner ce modèle initial. De nombreux modèles dérivés ont ensuite été développés à partir de ce modèle, sur diverses applications. Nous présentons dans cette section le modèle de carte de Kohonen et introduisons les notations que nous utiliserons dans cette partie de revue. Nous réintroduirons le modèle plus en détail au chapitre suivant.

### 1.1.1 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 1.1. Le principe de quantification vectorielle cherche à représenter 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 grille en deux dimensions. Les nœuds du graphe possèdent alors chacun un prototype  $\omega$  et sont *indexés* par  $p$ , un réel ou un vecteur en deux dimensions lorsque que la carte est une grille. Cette indexation et le format de graphe permet 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.

fais une petite figure:  $h(p)$



index ou indice?

## 1.1. LES CARTES AUTO-ORGANISATRICES DE KOHONEN COMME MODULES D'UNE ARCHITECTURE

3

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 choisie comme *Best Matching Unit* (BMU) de la carte. Son index est noté  $\Pi$ . La distance  $d$  généralement utilisée est la distance euclidienne.
3. Le prototype du BMU  $\omega(\Pi)$  est déplacé vers l'entrée  $X$ , ainsi que les prototypes des nœuds voisins de  $\Pi$  dans un voisinage  $H(p, \Pi)$ .

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

$H$  n'est pas un voisinage!

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

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. Toutes les données d'entrées sont tirées dans un même espace  $\mathcal{D}$ .

La version originale de l'algorithme de Kohonen s'appuie sur le calcul de distances entre entrées et prototypes. Ce dernier est remplacé dans de nombreux 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 de *dépliement* d'une carte lorsque qu'on parle d'apprentissage. Ce dépliement est représenté en figures 1.2 et 1.3 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 1.4 représente le dépliement d'une carte sur des images MNIST. Par son aspect ordonné, une carte est une représentation en faible dimension d'un espace d'entrée de grande dimension.

### 1.1.2 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 1.5.

c'est un degré de proximité...  
le voisinage, c'est l'ensemble des nœuds de fort degré de proximité

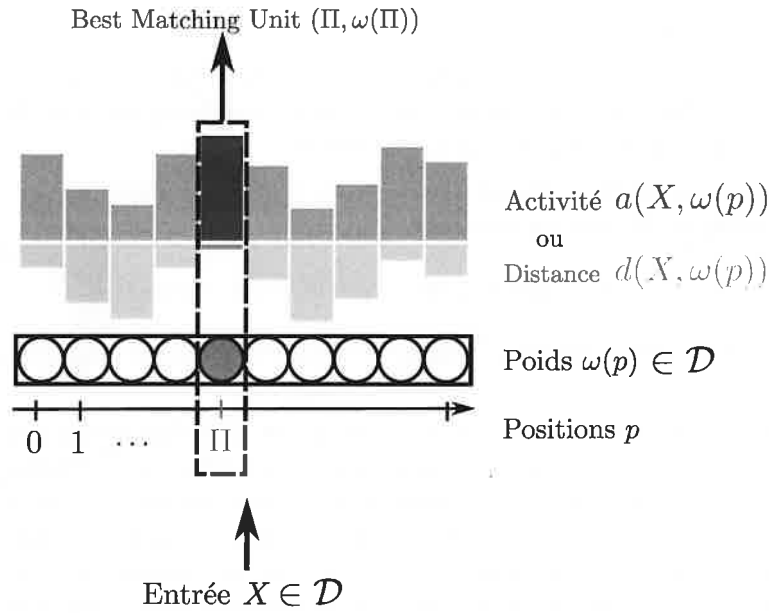


FIGURE 1.1 – Elements principaux composant une carte de Kohonen : une carte prend une entrée  $X$ , un ensemble d'unités de poids  $\omega$  et indexées par une position  $p$ ; une activité  $a$  ou une distance  $d$  est calculée pour chaque unité par rapport à l'entrée. La Best Matching Unit, abrégée en BMU, est calculée comme l'unité d'activité maximale sur les positions (ou de distance minimale). Sa position est notée  $\Pi$  et son poids est ainsi  $\omega(\Pi)$ .

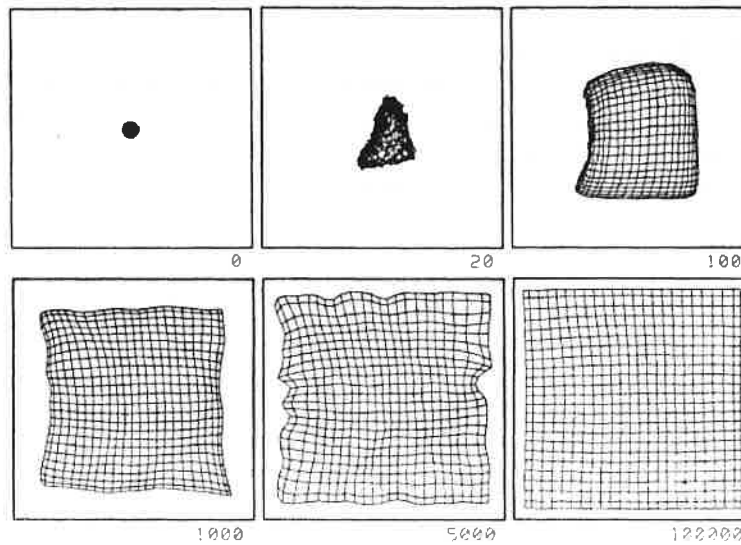


FIGURE 1.2 – Dépliement d'une SOM 2D sur des données dans le plan  $[0, 1]^2$ , (Kohonen 1995)

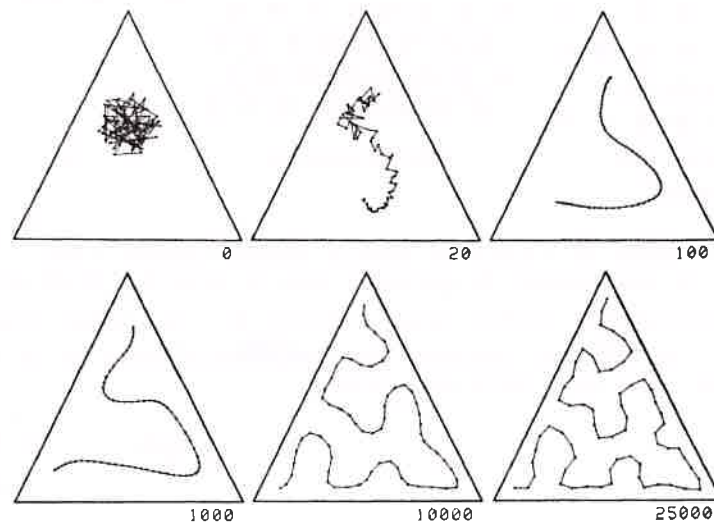


FIGURE 1.3 – Déploiement d'une SOM 1D sur des données dans un triangle 2D, (Kohonen 1995)

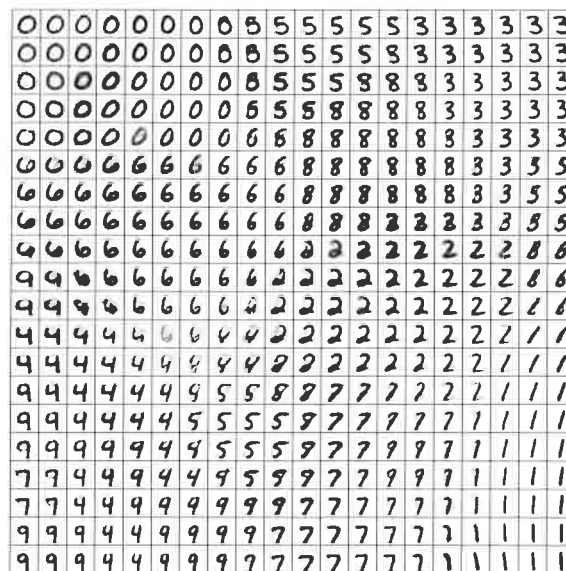


FIGURE 1.4 – Représentation de la base de données MNIST, images de chiffres écrits à main levées, par une SOM en deux dimensions. Une continuité est observée dans la forme des images lorsqu'on se déplace dans la carte : le 0 se transforme en 6, etc.

↑ on risque de te demander d'où vient cette figure... TP Centrale Supélec?



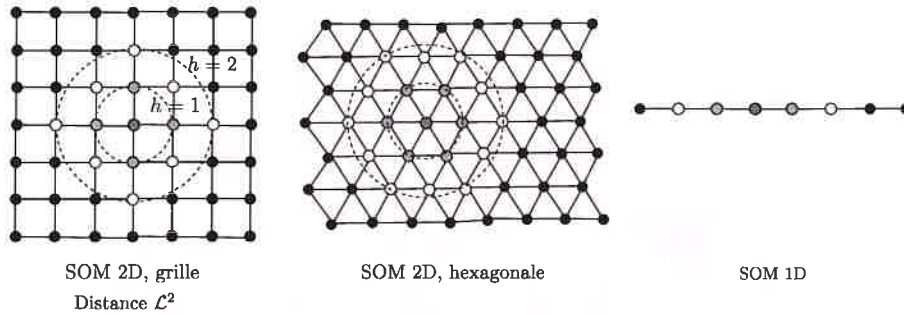


FIGURE 1.5 – Exemples de connexions dans le graphe support d'une SOM. Deux nœuds connectés sont ici à une distance de une unité dans la carte. Les SOM en deux dimensions sont les plus communément utilisées dans la littérature, sous forme d'une grille ou d'une grille hexagonale. Les SOM une dimension sont également utilisées. La distance utilisée dans l'exemple de grille carrée est la distance euclidienne.

*parfois*      *car le papier sur voyageur de commerce!*

En théorie, les cartes peuvent être 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 nœuds, et celui-ci augmente exponentiellement lorsqu'on augmente la dimension d'une carte de Kohonen. Les calculs deviennent alors rapidement coûteux. Les cartes une dimension sont quant à elles limitées en termes de représentation des données et sont donc rarement utilisées en pratique. Cependant, elles se prêtent mieux à la représentation graphique et au développement de nouveaux modèles 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.

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 al. 1990) permettant une recherche de BMU structurée. Certains modèles construisent une carte de Kohonen 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 d'algorithmes de quantification vectorielle classiques tels que K-means.

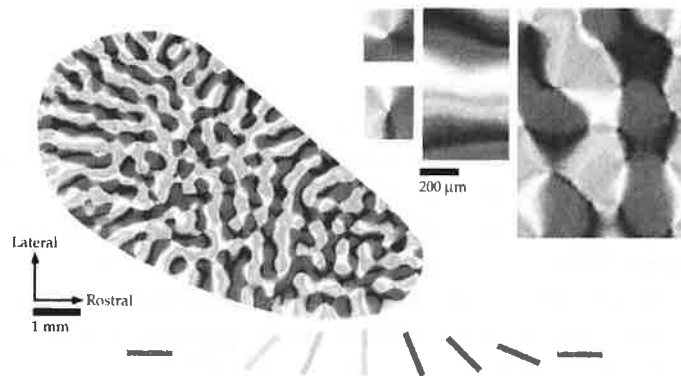


FIGURE 1.6 – Représentation des réponses du cortex visuel V1 à un stimulus visuel (bâtonnets d'orientations spatiales différentes). Les neurones répondant à une certaine orientation sont affichés de la même couleur. On observe une continuité entre les neurones proches dans le cortex et l'orientation à laquelle ils répondent. Cette propriété d'organisation est l'inspiration biologique des cartes de Kohonen.

## 1.2 Inspiration biologique des architectures de cartes

### 1.2.1 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 1.6. L'aire associée à l'audition présente aussi une organisation topographique (Reale et al. 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.

### 1.2.2 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 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. Fonctionnellement, un traitement multisensoriel de l'information incluant des activation rétroactives entre aires corticales permettrait par exemple d'expliquer l'effet ventriloque (Bonath et al. 2007). Lors de cet effet, une activité apparaît dans les cortices visuel et auditif pour les neurones sensibles à l'emplacement exact de la source des stimuli dans chacune des modalités. Après quelques millisecondes, correspondant au temps de trajet de l'aire visuelle à l'aire auditive, on observe une activité auditive pour les neurones sensibles à l'emplacement spatial de la source du stimulus visuel. Historiquement, le traitement de l'information multisensorielle dans le cortex cérébral apparaissait comme hiérarchique, des aires dites bas niveau alimentant des aires haut niveau 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 vérifié, de nombreux travaux montrent également l'existence de connexions directes entre aires sensorielles, dites bas-niveau, illustrées en figure 1.7 Anatomiquement, de nombreuses connexions entre les aires corticales dédiées au traitement d'une modalité sensorielles ont été mises en évidence chez différentes espèces, par exemple en (Calvert et al. 2004 ; Cappe et al. 2009 ; Foxe et al. 2005 ; Schroeder et al. 2005) De nombreux exemples de co-activations fonctionnelles entre aires corticales sont observées dans les travaux récents, par exemple entre aires tactile et visuelle (Sathian et al. 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 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.

La théorie de zone 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 cortical peuvent donc 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 apparaissent 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.



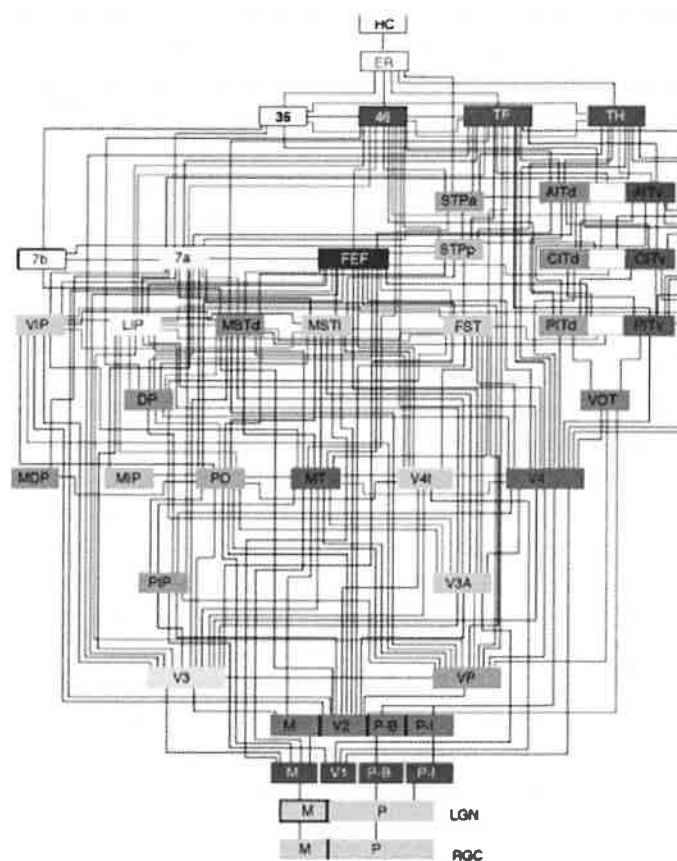


FIGURE 1.7 – Schéma de connexions entre aires sensorielles existant dans le cortex du singe, faisant apparaître des connexions rétroactives entre aires. Ce traitement fait apparaître plusieurs niveaux de hiérarchie tout en incluant des connexions entre aires d'un même niveau (Felleman et al. 1991)

### 1.3 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 abordons ces modèles d'un point de vue structurel en s'intéressant notamment à comment s'effectue 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 chque structure est spécifiquement conçue pour résoudre un porblème particulier, l'architecture est très appliquée : elle est uniquement conçue pour un problème particulier. Nous différencions cette vision d'architecture appliquée de la notion d'architecture générique modulaire. 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 est générique et utilisable sur n'importe quelles données d'application. C'est cette deuxième approche que nous étudions dans cette thèse.

Un module se définit ainsi comme un élément possédant des règles d'évolution à partir d'entrées et une sortie. L'assemblage de ces modules par leur interface 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 ainsi d'assembler un nombre variable d'éléments de même type, des modules, dans une architecture. 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 1.8. Les éléments de l'architecture, les modules, n'ont pas de rôle défini a priori avant d'être insérés dans l'architecture, mais 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. Cette notion d'entrée et sorties interne à l'architecture constitue les règles d'interface entre cartes.

#### 1.3.1 Méthode d'analyse

L'étude des architectures développées dans les travaux précédents nous amènent à différencier les structures selon plusieurs aspects. 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 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

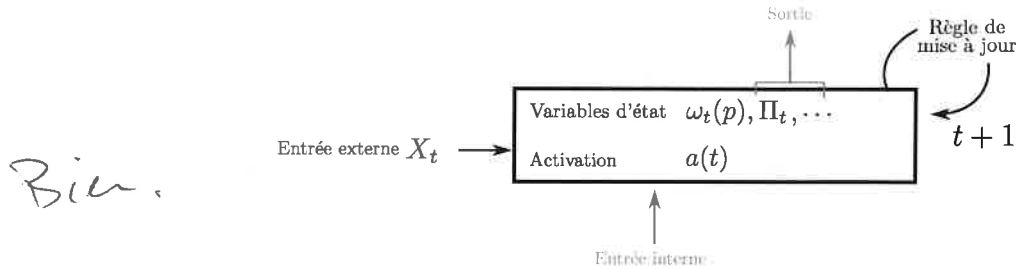


FIGURE 1.8 – Description de la notion de module d’une architecture modulaire. Un module prend des entrées externes à l’architecture ou internes, et possède des variable d’état dont l’évolution dans le temps est régie par des règles de mise à jour ou équations d’évolution. La sortie d’un module est un ensemble de variable d’états. La sortie d’un module et l’entrée interne d’une autre constitue l’interface entre modules. Les modules étudiés dans ce chapitre sont des cartes de Kohonen. Leurs variables d’état sont entre autre leur poids et leur BMU.

graphe orienté, dans lequel les nœuds 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 1.9.

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 donc les modèles du plus spécifique au plus générique.

**Granularité du calcul et temporalité de l’apprentissage** Pour chaque catégorie d’architecture, nous analyserons le mode de connexion entre cartes. Cette communication est réalisée soit par une surcouche algorithmique, soit est interne à l’organisation d’une carte. Cette surcouche algorithmique se présente sous forme de sélection ou ajout de cartes pour l’apprentissage des données d’entrées. Nous présenterons des exemples de modèles de ce type dans le paragraphe suivant. La communication effectuée par une telle surcouche est alors globale à l’architecture. La communication est interne à l’organisation d’une carte lorsque les données transmises sont directement prises en compte dans l’algorithme de mise à jour. Ainsi, les cartes ou les nœuds implémentant une telle communication prennent en tant qu’entrée des

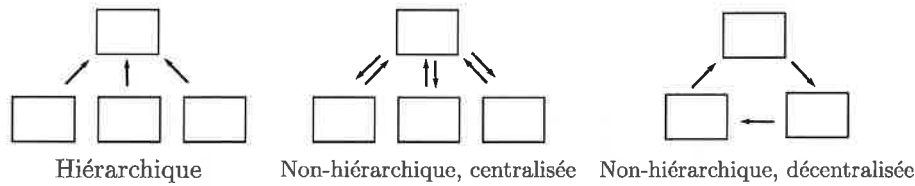


FIGURE 1.9 – Exemples de connexions dans des architectures hiérarchiques et non-hiérarchiques centralisées et décentralisées. Un rectangle correspond à un module, ici une carte auto-organisatrice. Une flèche représente l'existence d'une interface entre deux modules : le module destination prend comme entrée interne la sortie de la source et utilise donc de l'information de la source dans ses règles d'évolution.

éléments de sortie des autres cartes, telles que la position du BMU, son poids ou une activité neuronale. La communication réalisée de cette manière est donc plutôt locale à une carte. La construction d'architecture par la méthode globale nous intéresse peu dans un cadre 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 cartes varient. Certaines architectures utilisent ainsi la position du BMU, son poids, ou encore un ensemble d'activités de neurones d'une carte. Nous releverons ainsi 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 du 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 impulsionnels, 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 permet 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.

### 1.3.2 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 multi-couches* : ces modèles se rapportent tous au même concept d'architecture hiérarchique de SOMs. Nous les divisons en deux catégories : la première est celle des architectures

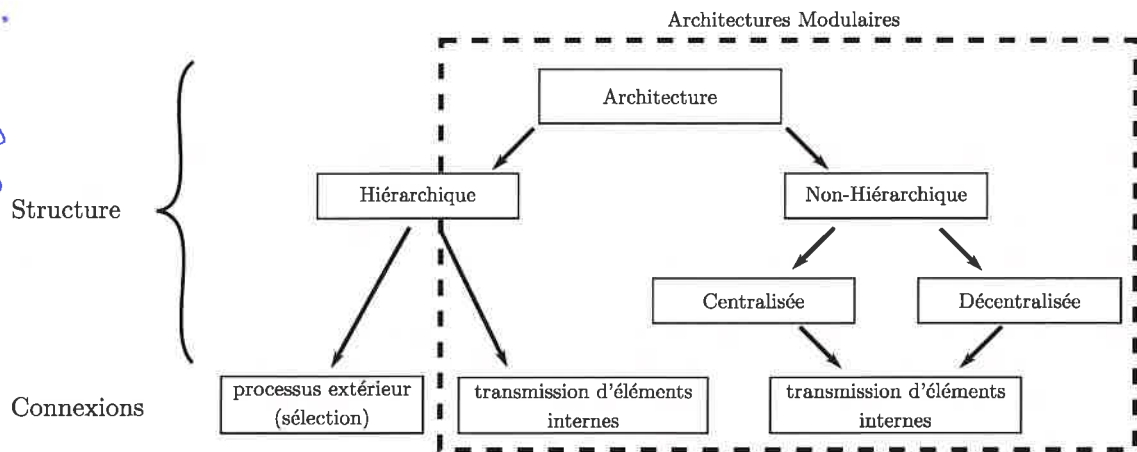


FIGURE 1.10 – Taxonomie des architectures de cartes présentées dans ce chapitre. Nous analyserons comment leurs caractéristiques structurelles : hiérarchiques ou non-hiérarchiques, centralisées, décentralisées, façonnent leur comportement d'apprentissage. Nous analyserons également leur interface de communication. Nous n'avons pas relevé d'architecture non-hiérarchique s'appuyant sur le principe de sélection, car ce principe est fondamentalement hiérarchique.

par sélection, dans laquelle le premier niveau est généralement une carte seule et le nombre de cartes augmente au fur et à mesure des niveaux. L'autre catégorie est celle des architectures par transmission de représentation. Dans ce cas, les cartes du deuxième niveau prennent en entrée la sortie du premier niveau et non les données d'entrées. Le nombre de cartes diminue alors au parcours des niveaux le dernier niveau formant une représentation plus abstraite de l'espace d'entrée. Nous présenterons les équations des différents modèles dans le formalisme introduit en section 1.1. 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.

### Architectures hiérarchiques par sélection

Nous appelons une architecture par sélection un ensemble de cartes organisées de façon hiérarchiques 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 1.11. Ce modèle est 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 entrée  $X \in \mathcal{D}$ . 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 sous-ensemble, tel que chaque sous-ensemble  $\mathcal{D}_i$  est l'ensemble des entrées  $X_i$  ayant  $i$  pour position du BMU associé à l'entrée. Chaque carte  $i$  du deuxième niveau est alors entraînées 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 également 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-ensemble 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 sélective 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 la mémoire nécessaire pour le calcul.

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.

Nous ne considérons pas ces architectures comme modulaires. En effet, chacune des cartes d'un tel modèle apprend sur des données d'entrées qui leur sont fournies par un processus extérieur global : la décomposition en sous-espaces. Seul ce processus traite l'information de chacune des cartes et apporte une connexion entre cartes. Dans ce sens, on ne peut pas parler de modularité de par l'aspect global de l'algorithme.

### 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

TR!



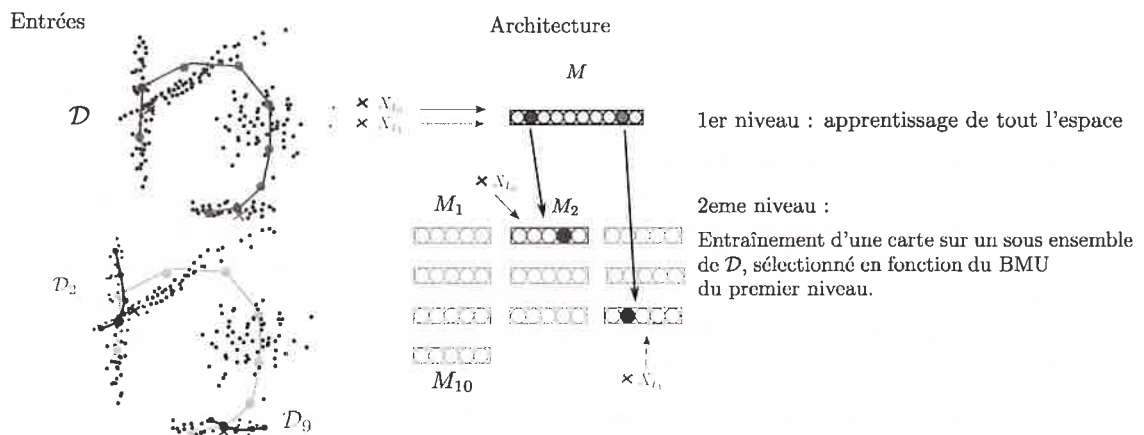


FIGURE 1.11 – Exemple d'architecture hiérarchique sélective. La carte du premier niveau est entraînée sur tout l'espace d'entrée. Après apprentissage, la carte permet de filtrer les entrées pour les renvoyer vers une carte du niveau suivant. Dans cet exemple, la position du BMU de la carte du niveau 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  $M_1$ .

sélection, la deuxième couche de carte de ces architectures hiérarchiques ne prend plus comme entrée un élément de l'espace d'entrée de l'architecture, mais prend en entrée 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.

Les 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 al. 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  $\Pi$  de la première carte ; cette architecture est illustrée en figure 1.12. Le BMU de la première carte est alors défini par :

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

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*)

⚠ Très pas introduit que  $\Pi^{(i)}$  signifie le  $\square$  de la carte  $i$

par similarité, sachant que les données sont effectivement regroupées en classes distinctes. Ces clusters sont définis après apprentissage par les cellules de Voronoï 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 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 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.

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 al. 2013 ; Paplinski et al. 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 couche. 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 légère. 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 imagerie qui sont utilisées en tant qu'entrées d'une première couche de cartes. Après apprentissage de cette couche, l'image est reconstruite grâce aux poids des BMU, puis décomposée en imagerie de tailles différentes pour être soumise à la deuxième couche de carte. Enfin, (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 carte 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 (N. Liu et al. 2015 ; Dozono et al. 2016 ; Hankins et al. 2018 ; Mici et al. 2018 ; Wickramasinghe et al. 2019 ; Aly et al. 2020 ; Sakkari et al. 2020 ; Nawaratne et al. 2020) et de nombreux autres travaux sont présentés comme tels. Les travaux se décrivant par ces termes implémentent des structures

hiérarchiques puisant leur inspiration des réseaux de neurones profonds (*Deep Learning*), ayant notamment connu leur essor avec les réseaux convolutifs permettant l'apprentissage supervisé d'images (LeCun et al. 2015). Cependant, leur analogie avec les modèles de Deep Learning, qui s'appuient sur le principe de rétropropagation du gradient, s'arrête à la présence de couches 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 (N. Liu et al. 2015 ; Wickramasinghe et al. 2019) est illustré en figure 1.13. Le but d'une telle architecture est de classifier des images  $X$  fournies en entrée de l'architecture. Une fenêtre est déplacée sur l'image d'entrée, créant un ensemble de  $N \times N$  imagerie de position fixée. La première couche du réseau comporte  $N \times N$  cartes, donc chacune prend en entrée l'imagerie de même position  $i, j$ . La sortie de la couche donne  $N \times N$  positions de BMU  $\Pi_{i,j}$ . Ces positions représentées comme des valeurs en une dimension sont assemblées en une image intermédiaire, chaque pixel prenant la valeur du BMU de la carte correspondante.

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

Une deuxième couche de cartes de même structure que la première carte est alors appliquée à cette image intermédiaire. La dernière couche du réseau est composée d'une SOM simple effectuant la quantification vectorielle sur l'image de sortie de la couche précédente, vue comme une représentation abstraite de l'entrée. L'interface entre les couches de cartes est créée à partir des 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 imagerie.

Ce type d'architecture utilise bien l'aspect topologique de la carte de Kohonen dans ses calculs, les interfaces entre couches de cartes s'appuyant sur les positions du BMU. L'utilisation des positions au lieu de poids permettent de réduire la dimension des images traitées par les couches successives. L'image de sortie de la dernière couche est alors de taille réduite. Lorsque la dernière SOM classifie cette imagerie, elle classifie 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, comme pour les architectures sélectives, sont ascendantes dans leur mise à jour : chacune des couches de cartes sont entraînées les unes après les autres. Ces travaux sont appliqués à des tâches de classification qu'on pourrait aussi réaliser avec une SOM classique. Les motivations pour utiliser une architecture plutôt qu'une SOM est d'abord le fait la couche finale du réseau possède de meilleures performances en termes de classification que si on avait utilisé une SOM simple.

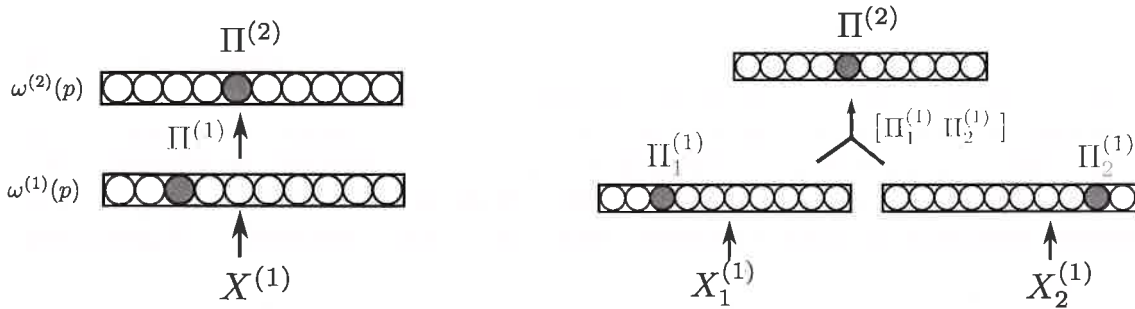


FIGURE 1.12 – Deux exemples d'architectures basées sur HSOM. A gauche, le modèle HSOM original proposé en (Lampinen et al. 1992). L'apprentissage des positions du BMU de la première couche par la seconde permet de mieux détecter les ensembles de données présents dans la distribution des  $X$  (*clustering*). La deuxième couche est vue comme un niveau plus abstrait que la première. A droite, une version de HSOM comportant plus de cartes proposée en (Hagenauer et al. 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.

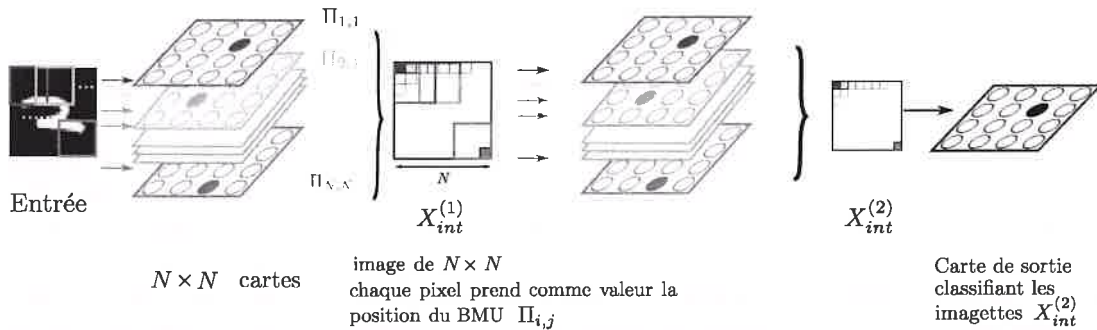


FIGURE 1.13 – Architecture DSOM de SOM "convolutive" (Nan Liu et al. 2015). Les auteurs utilisent les positions des BMUs  $\Pi_{p,q}$  d'une couche de carte, disposés dans un tableau 2D, comme valeurs d'entrée pour les couches suivantes. Les couches sont entraînées les unes après les autres.

## Discussion

Nous avons donc 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 dépliement 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 nœuds 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 carte est gérée par un processus global.

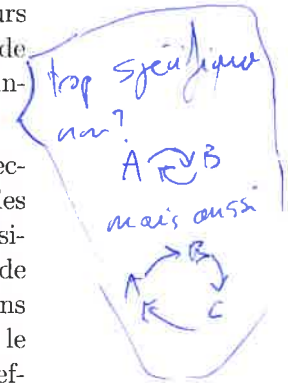
Le second mode de connexion repose sur la modification du principe de calcul d'activité et de mise à jour d'une carte pour prendre en compte les éléments de réponse d'une autre carte, par exemple en ajoutant cet élément de réponse en tant qu'entrée secondaire d'une carte. Ce mode de connexion est intégré au processus d'auto-organisation et non une couche algorithmique supplémentaire comme les architectures par sélection, ce qui localise le traitement des connexions à l'échelle d'une carte. Cette méthode de construction d'architecture s'appuie sur la transmission d'une représentation de l'entrée interne entre cartes. Ce type de connexions nous intéresse d'un point de vue modulaire : il autorise en effet 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 terme 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. 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 carte auto-organisatrices sur des applications spécifiques. Elles n'ont pas la capacité de faire d'autres types de calcul que ceux originalement 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.

### 1.3.3 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 schéma de connexion comporte des boucles de rétroaction : une carte A reçoit de l'information d'une carte B, qui elle-même reçoit 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 ces 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 naturellement 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 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*), dont parmi les enjeux principaux figurent la fusion de données multimodales, le traitement de séquences et l'apprentissage développemental (Smith et al. 2005), inspiré de la biologie. 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. Dans la mémoire associative, elles sont alors utilisées pour prédire une modalité à partir d'une autre.

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 interne. Ces architectures sont *centralisées* dans la mesure où ces 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.

### 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 al. 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, commun 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 al. 2013) et (Escobar-Juárez et al. 2016). Dans ces deux travaux de neuroscience computationnelle, les auteurs construisent une architecture se voulant une modélisation du cadre CDZ, mais avec des cartes auto-organisatrices classiques, en transmettant les positions des BMU entre les cartes multimodales.

Le modèle MMCM de (Lallee et al. 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

?  
espace  
inséparable  
stg...

1.14. Nous définissons cette architecture comme non-hiérarchique car il existe des rétroactions entre les cartes modales  $M^{(1)}, M^{(2)}$  et la carte associative  $M^{(m)}$ . Dans l'exemple d'une architecture à deux cartes modales, l'une reçoit les mouvements de tête d'un robot et une autre les mouvements du bras. Chaque carte du premier niveau possède une couche de poids  $\omega_c$  liées aux entrées sensorielle ainsi qu'une couche de poids  $\omega_c$  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. Cette carte se veut la représentation de la zone de convergence-divergence du modèle CDZ.

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  $A$  dans la carte. Les poids  $\omega_c$  sont ensuite gelés, et la carte associative est ensuite entraînée pour apprendre à associer les positions des BMUs correspondant aux cartes modales :

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

Enfin, les couches des cartes modales  $\omega_c$  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  $\Pi_c$  relatif aux activités internes pendant l'apprentissage, les deux couches de poids étant décorréliées. Cette décorrélation permet de ne pas avoir à gérer les rétroactions. Après ces trois phases d'apprentissage, les entrées modales ne sont pas présentées aux cartes modales. L'activation manuelle d'un neurone de position  $p^{(a)}$  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 interne :

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

La valeur  $\omega_c^{(1)}(\Pi^{(1)})$  est alors une prédiction de la modalité 1.

Les auteurs montrent que cette méthode d'activation produit des mouvements coordonnés entre modalités. L'activation d'un neurone d'une carte sensorielle entraîne également une activation coordonnée dans les autres cartes sensorielles. 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 1.15. 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 neurones à neurones mis 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

mis à jour

qui mais est-ce que c'est clair ?

$a = m$  ?

2

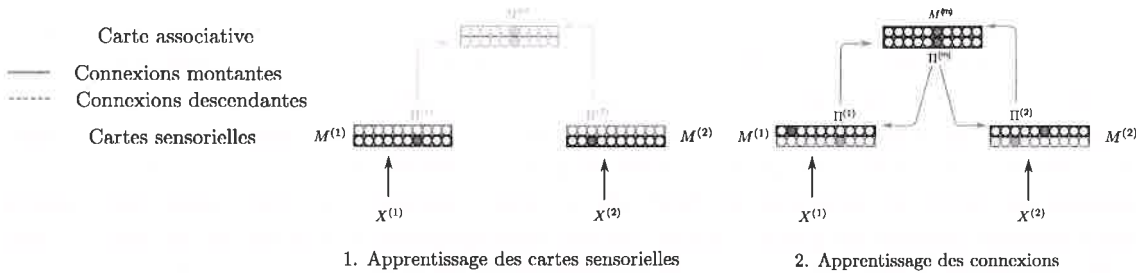


FIGURE 1.14 – L'architecture MMCM (Lalée et al. 2013) est une architecture centralisée. Les cartes du premier niveau sont les cartes modales  $M^{(1)}$ ,  $M^{(2)}$  qui reçoivent l'une les mouvements de tête d'un robot, une autre les mouvements de son bras. Une carte associative  $M^{(n)}$  reçoit les positions des BMUs  $\Pi^{(1)}$ ,  $\Pi^{(2)}$  de chaque carte du premier niveau en tant qu'entrées.

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 modale 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  $\omega_{ij}$ . Tous les neurones de la carte modale reçoivent donc le même ensemble d'entrées  $\{a_i, i = 0..N\}$ . En termes de finalité, c'est équivalent à l'ajout d'une couche de poids supplémentaire à la carte modale tels que  $\forall j, \omega_j \in [0, N]$  représentent 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 décorréliées lors de l'apprentissage, les rétroactions n'ont pas d'influence sur la mise à jour. Elles sont utilisées seulement en phase d'application.

L'architecture Bijama développée en (Ménard et al. 2005 ; Khouzam et al. 2013) s'appuie quand à elle uniquement sur l'association des activités neuronales. Dans ce modèle, les auteurs utilisent une version de carte auto-organisatrice cellulaire reposant sur des calculs locaux au niveau même des neurones. Une entrée  $X$  est présentée à tous les neurones d'une carte, liés par des mécanismes locaux. 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. Nous pouvons souligner le parallèle existant entre la version cellulaire et la version centralisée d'une carte de Kohonen. Dans notre étude, nous étudions les cartes de Kohonen classiques, mais les mécanismes occurring à l'échelle d'une carte dans une carte de Kohonen cellulaire ressemblent à ce qu'on observe dans une carte classique.

Dans l'architecture Bijama, la connexion entre carte modale et carte associative s'effectue par des connexions neurones à neurones. Chaque neurone de la carte associative prend en entrée une rangée d'une carte modale. Cela revient donc à la transmission de champs d'activation entre cartes. La mise à jour dans ce cas est asynchrone : les connexions entre neurones génèrent la mise à jour des paramètres du neurone. Les mécanismes locaux d'évolution des neurones fait apparaître une notion de relaxation au sein des cartes. Contrairement aux architectures précédentes, l'activation prise en compte en entrée de la carte associative est l'activité externe modulée par l'activité revenant de la carte associative.

assoc ?

font

pas d'arr.

avec fortif  
du

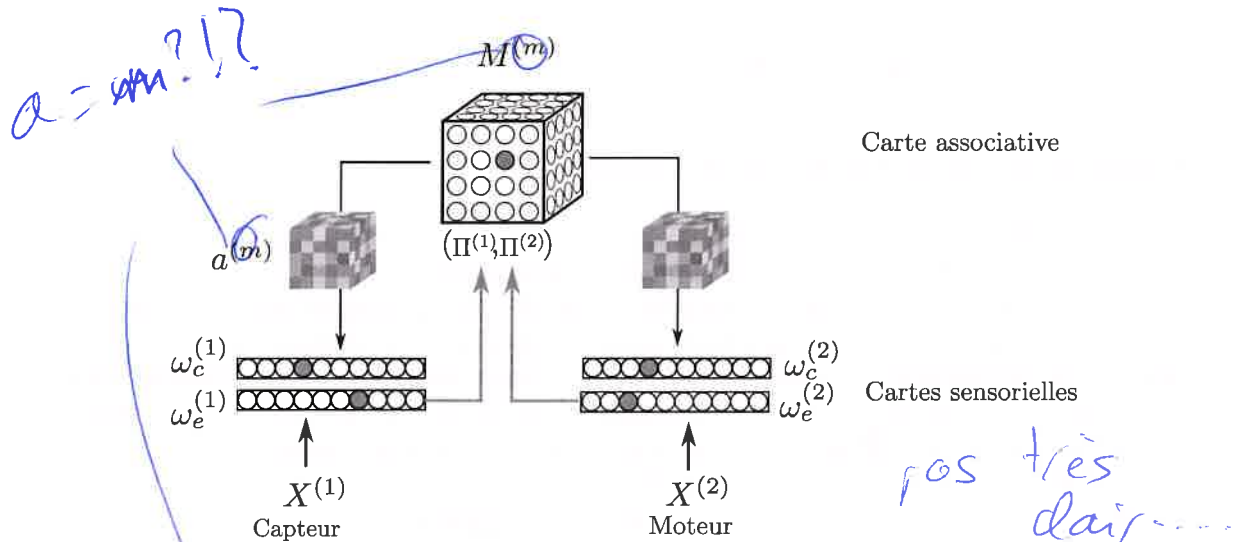


FIGURE 1.15 – Le modèle SOIMA (Escobar-Juárez et al. 2016) propose une architecture centralisée dans laquelle des cartes modales  $M^{(1)}$  et  $M^{(2)}$  sont connectées par une carte associative  $M^{(a)}$ . Cette carte associative prend comme entrée les BMUs des cartes modales. Les connexions descendantes 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.

Les modèles mentionnés ci-dessus rentrent 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 et le modèle Bijama privilégient 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 supplémentaires dans les cartes non-hiérarchiques : l'activité de la carte A influence l'activité de la carte B, mais l'activité de la carte B influence également celle de la carte A, 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 interne 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. Le modèle Bijama, étant un modèle cellulaire, gère ses mises à jour de manière asynchrone, se rapprochant plus de la biologie.

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, elles prennent des rôles différents a priori. Des cartes sont associatives et des cartes sont



modales.

→ autant que toi pour Bijama!

### Architectures non-hiérarchiques décentralisées

Une architecture non-hiérarchique décentralisée est une architecture présentant des rétroaction 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, dans la mesure où aucun a priori n'est associé aux modules, même une fois connectés en architecture. Leur spécialisation intervient uniquement grâce à leurs règles d'évolution interne.

Les auteurs de (Khacef et al. 2020) utilisent par exemple deux cartes de Kohonen associées par des connexions neurones à 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-hiérarchique est développée en (Magnus Johnsson et al. 2008; M. Johnsson et al. 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 1.16 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 interne provenant de l'autre carte,  $\gamma$ . Sur ces entrées, les auteurs calculent une activité par couche de poids :  $a_e$  et  $a_c$ . L'entrée  $\gamma$  correspond au vecteur des activations externes  $a_c$  des neurones de l'autre carte. L'interface entre cartes utilisée en A-SOM est donc le champ d'activation des neurones, interface qui avait été utilisée dans les architectures centralisées SOIMA ou Bijama. Cette interface par transmission d'activation comme entrée d'une carte est équivalente à des connexions pondérées par des poids  $w_{cij}$  reliant le neurone  $i$  d'une carte au neurone  $j$  de l'autre. On a alors  $w_c(i) = [w_{ci1}, \dots, w_{ciN}]$ . Lors de



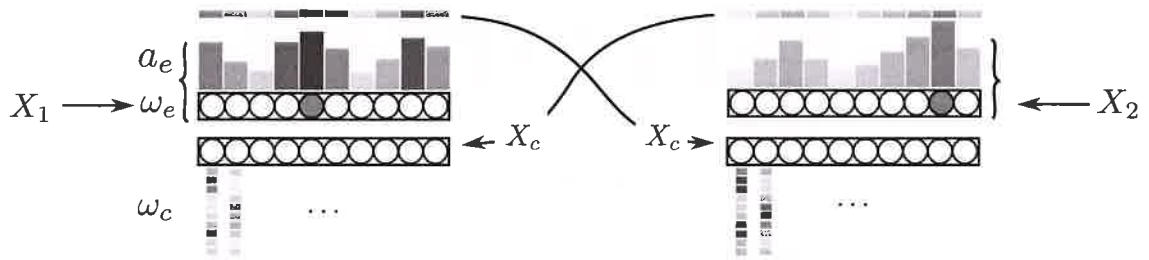


FIGURE 1.16 – Le modèle A-SOM (M. Johnsson et al. 2009) associe les activités de différentes cartes. Chaque ASOM prend une entrée modale  $X_1$  et  $X_2$ . Chacune des cartes possède deux couches de poids, une couche  $\omega_e$  associée aux entrées modales et une couche  $\omega_c$  associées aux entrées  $\gamma$  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 coupées. L'activation de cette carte 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'apprentissage, la mise à jour des poids  $\omega_e$  et  $\omega_c$  est réalisée de manière indépendante. Le BMU de position  $\Pi$  se situe au maximum de l'activité externe et les poids  $\omega_c$  sont mis à jour comme dans une SOM classique :

$$\omega_c(p) \leftarrow \omega_c(p) + \alpha H(\Pi, p)(\omega_c(p) - X_c)$$

Les poids  $\omega_c$  sont mis à jour en fonction de la différence entre activités externes et internes :

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

Cette règle de mise à jour permet de renforcer le schéma d'activation 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 couper 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.

L'architecture développée en (M. Lefort et al. 2015) implémente également une architecture décentralisée pour de l'apprentissage multimodal. De façon similaire à A-SOM, les auteurs associent plusieurs poids aux neurones d'une carte, chacun des poids étant relatif à un type d'entrée : l'entrée modale et l'entrée venant d'autres cartes. L'information transmise dans ce cas est une partie de l'activité des neurones, celle située dans un carré centré à la même position que le neurone courant. 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, rappelant le principe proposé en (Ménard et al. 2005). Comme pour le modèle Bijama, SOMMA prend en compte les rétroactions dans le calcul d'activité de chaque carte. Nous ne nous attarderons pas sur les détails du modèle qui est à base de neurones impulsionnels, éloignés de notre champ d'étude. Néanmoins, le principe de coactivation et de gestion des rétroactions se doit d'être soulevé.

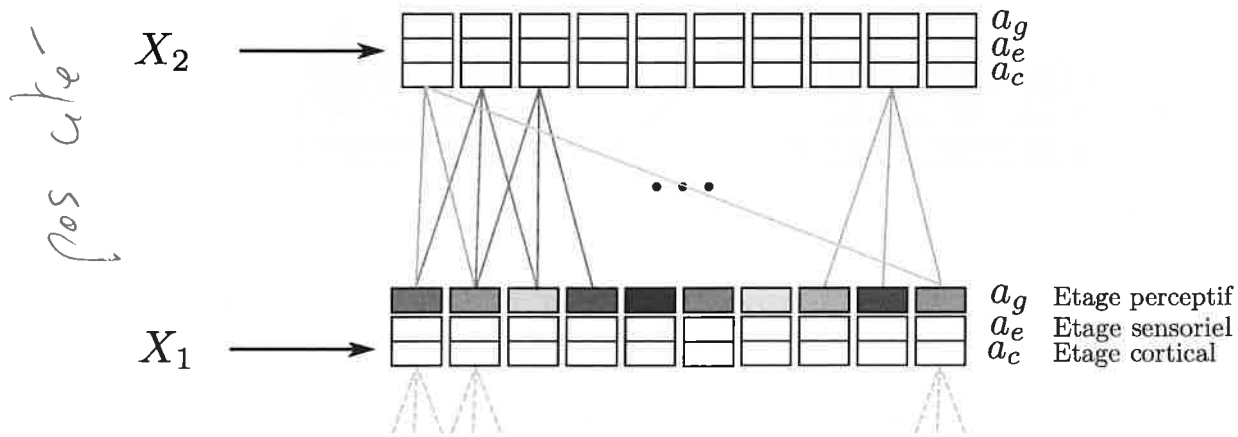


FIGURE 1.17 – Le modèle SOMMA (Mathieu Lefort et al. 2011 ; Mathieu Lefort 2012) associe les activités de différentes cartes, mais en réduisant les champs de réception d'activité aux neurones entourant le neurone situé en position courante. Une seule connexion est représentée, mais toutes les connexions sont réciproques. Contrairement au modèle ASOM, l'activité considérée lors de la transmission est l'activité totale d'une colonne. Les auteurs introduisent un principe de résonance permettant de gérer les rétroactions.

## 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 cas, les modèles de connexions entre cartes sont différenciés. Nous nous intéresserons donc dans cette thèse à un modèle d'architecture décentralisée de carte de Kohonen.

### 1.3.4 Connexions temporelles et architectures de cartes auto-organisatrices

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 1.18 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 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 séquence.

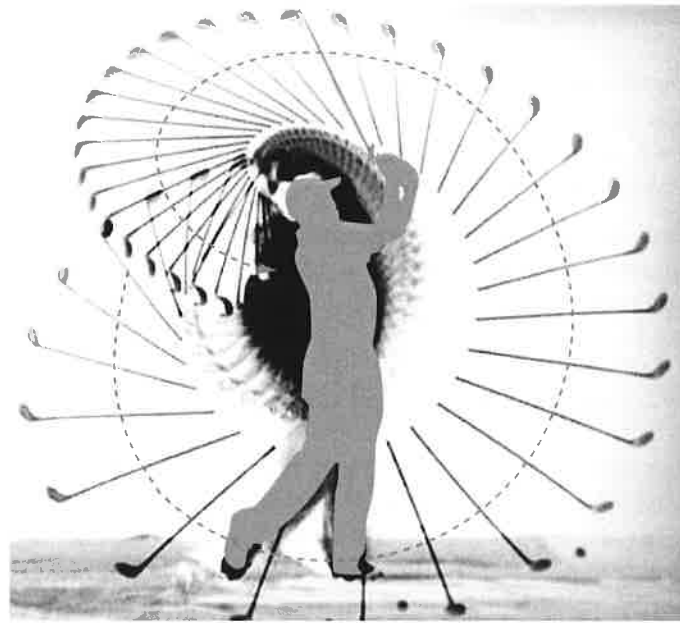


FIGURE 1.18 – L'image présentée à un réseau (en bleu) correspond à un instant d'une séquence. L'objectif de l'apprentissage non-supervisé de séquences est d'extraire une représentation d'une séquence d'entrée. Une utilisation commune est la classification de mouvements. La séquence "tirer" sera différente de la séquence "marcher".

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'applications des architectures de cartes auto-organisatrices mais bien dans la structure même du traitement des données. Une solution pour faire de l'apprentissage de séquence 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, implémentée 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 actuels sont appelés réseaux récurrents ou récurrents. Plusieurs modèles de cartes auto-organisatrices récurrentes, destiné à l'apprentissage de séquence, 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 intéressons aux modèles multicartes implémentant des connexions récurrentes au même titre que des connexions inter-cartes.

### Cartes auto-organisatrices récurrentes

Les modèles de cartes récurrentes existant dans la littérature s'appuient sur la transmission de représentation interne entre 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  $\gamma_t$  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 al. 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 proposé 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 1.19. Ces réseaux sont des versions incrémentales de cartes de Kohonen dans lesquelles des neurones sont ajoutés au cours de l'apprentissage, le processus de recherche de BMU restant ensuite similaire à une SOM classique. Cette architecture utilise deux réseaux GWR pour apprendre des séquences, formant une mémoire épisodique et une mémoire sémantique. La carte associée à la mémoire épisodique (G-EM) est une version récurrente du GWR, prenant en entrée courante  $X_t$  et en entrée de contexte  $\omega(\Pi_{t-1})$ , le poids du BMU à l'instant précédent. La deuxième carte est une version classique du GWR. Elle prend en entrée le poids du BMU de la carte épisodique ainsi que la classe de la séquence courante, afin de mettre à jour ses poids. Les auteurs utilisent leur architecture pour de la reconnaissance d'objets. Cependant, lors de l'apprentissage, les données ne sont pas présentées après un tirage aléatoire dans l'espace, mais sont présentées classe par classe : tous les objets d'une même classe d'abord, etc. Les auteurs montrent que l'architecture est capable de bien prédire la classe d'un objet lors d'un test sur toutes les classes apprises. À titre de comparaison, une SOM classique apprendrait la classe du premier objet, puis l'oublierait pour se déplier entièrement sur la deuxième classe ; à terme, seule la dernière classe serait gardée en mémoire. Ce type de structure prenant des entrées évoluant dans le temps et les gardant en mémoire s'inscrit dans l'apprentissage développemental. Nous entrevoyons ainsi l'intérêt que peuvent présenter des structures 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 multicartes. Elle calcule alors son activité par rapport à son entrée, et possède une seconde couche de poids qu'elle met à jour relativement au champ d'activation de l'instant précédent. Cette structure est appliquée à la prédiction de mouvement. De la même façon qu'une architecture est capable, à partir d'une modalité, de prédire les valeurs correspondant à l'autre modalité, l'architecture incluant une version récurrente peut prédire la fin d'une séquence à partir de son début. Nous n'avons cependant pas relevé de travaux les intégrant effectivement dans une architecture multi-cartes.

Enfin, les travaux menés précédemment dans notre équipe en (Baheux et al. 2014) proposent une architecture multicarte dédiée au traitement de séquence, décrite en figure 1.20. L'architecture proposée est composée de deux cartes. Chacune des cartes est composée de deux couches de poids  $\omega_e$  et  $\omega_c$ . Une des cartes prend une entrée  $X_t$  correspondant à l'observation courante et relative à la couche de poids  $\omega_e$ , comme une SOM classique. La deuxième couche de poids est relative à l'information interne descendant de la seconde carte, qui est la position du BMU  $\Pi_2$  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

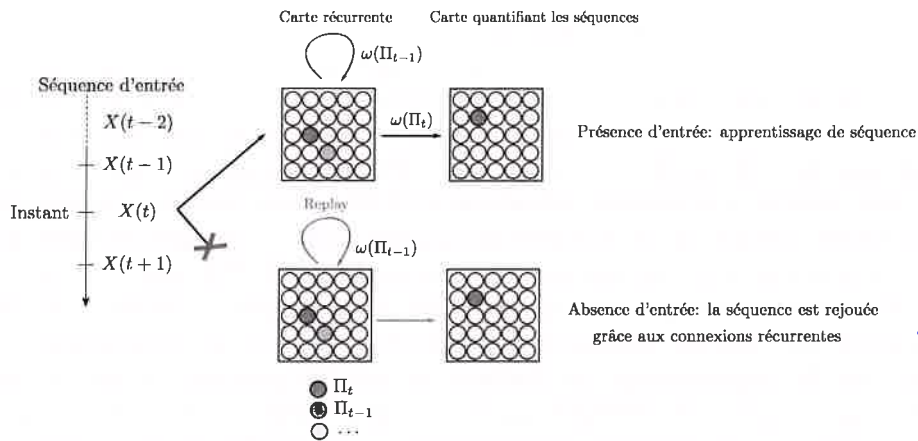


FIGURE 1.19 – Architecture à double mémoire proposée en (Parisi et al. 2018). La couche de mémoire épisodique, permettant la différenciation de séquences, prend en entrée externe un instant de la séquence d'entrée  $X$  et en entrée de contexte le poids du BMU de l'instant précédent. La couche de mémoire 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.

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 "résonner" les activités en déplaçant petit à petit les BMUs de chaque carte, jusqu'à obtenir un état stable pour les activités. 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 en plus de sa valeur.

## 1.4 Axe de recherche

La littérature portant sur les architecture 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 catégoriser de 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

il manque son-sd, et le papier de  
Guthrie de Barbara Hammer.

il y a des  
trucs récurrents  
avec seulement  
le BAV.  
je l'envoie le  
lien

il y a aussi autre  
papier avec j'étais  
pour la résonance (avec baron  
sur recsom). Si on, ici,

tu pourrais  
insister un  
peu sur la  
résonance...  
ou en un  
moins dire  
que tu vas  
mettre en place  
un truc  
similaire.

il y a des

considérer comme



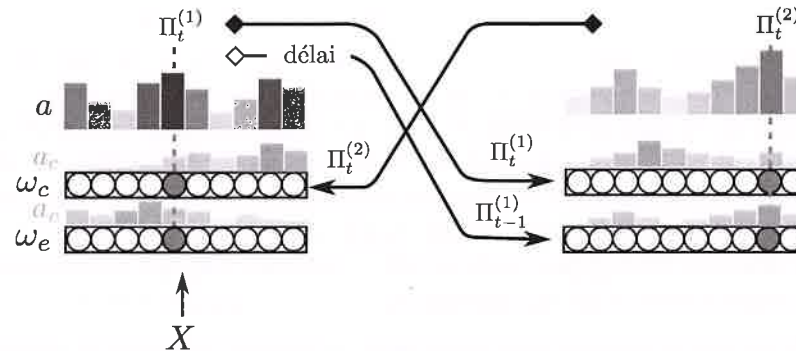


FIGURE 1.20 – Structures de deux cartes auto-organisatrices communicantes, (Baheux et al. 2014). Chaque carte est composée de trois couches d'activités, représentées séparément sur le schéma : sur la première carte, une activité est relative à l'entrée  $o$ , l'observation. L'autre activité reçoit une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

d'évolution internes qui ont été définies et grâce aux interfaces venant d'autres modules.

D'autre part, certains travaux portent sur la création d'architecture comportant des rétroactions, que nous appelons architectures non-hiérarchiques. Ces travaux se revendiquent du 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 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 séquence, se rapprochent par ailleurs des architectures de cartes par leur structure s'appuyant sur une transmission d'information entre itération. 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 champ d'activation. La transmission de la position du BMU comme information entre carte apparaît au contraire comme un paradigme principalement utilisé dans le domaine des cartes hiérar-

pos de temps?

↓  
synchronisme avec  
SON-SD donc...

chiques. Ce mode de transmission exploite pleinement l'aspect topologique de la carte de Kohonen, est indépendante du type d'entrée fournie, 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 (Lalle et al. 2013) s'appuient sur transmission de la position du BMU dans le cadre des SOM 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 en effet connectés par des connexions neurones à neurones et se transmettent donc une activation.

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.

D'autre part, la plupart des architectures relevées dans la littérature s'appuient sur des mises à jour séquentielles. Une architecture générale incluant le traitement de séquence 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 à (Baheux et al. 2014) sur des architectures récurrentes multimodales utilisant la transmission de la position du BMU entre des cartes de Kohonen, exploitant la position du BMU. Les travaux ~~commencés~~ en (Baheux et al. 2014), bien qu'ils exploitent des connexions intercartes, sont similaires à ce qu'on obtiendrait avec une carte récurrente simple, telle que celle décrite en (Fix et al. 2020). Par leur motivation, qui est le développement d'un système multicarte générique, nos travaux se rapprochent aussi des travaux conduits sur l'architecture A-SOM (Magnus Johnsson et al. 2008 ; M. Johnsson et al. 2009 ; Gil et al. 2015 ; Buonamente et al. 2015).

Nous dirigerons les travaux vers le traitement de données multimodales dans de petites architectures. Le but de cette thèse est d'identifier et de formaliser les comportements d'apprentissage générés par une petite structure non-hiérarchique, dans une optique de création d'architecture comportant de nombreuses cartes.

✓  
Baheux  
fait explicitement  
le lien avec  
BMU

présentées



aux travaux  
amorçés par...

TABLE 1.1 – Comparaison des principaux modèles d'architectures relevés dans ce chapitre. Nous n'y faisons pas figurer les architectures sélectives, étant non-modulaires. Les modèles très similaires sont regroupées sur une seule ligne.

Modèle	Type	Mise à jour	Mode de transmission
HSOM <sup>1</sup>	Hiérarchique	Séquentielle	Position du BMU
Deep SOM <sup>2</sup>	Hiérarchique	Séquentielle	Position du BMU
HSOM - poids du BMU <sup>3</sup>	Hiérarchique	Séquentielle	Poids du BMU
MMCM <sup>4</sup>	Non-Hiérarchique, Centralisée	Séquentielle	Positions BMU
SOIMA <sup>5</sup>	Non-Hiérarchique, Centralisée	Séquentielle	Champ d'activité
Bijama <sup>6</sup>	Non-Hiérarchique, Centralisée	Asynchrone	Champ d'activité partiel
A-SOM <sup>7</sup>	Non-Hiérarchique, Décentralisée	Synchrone	Champ d'activité
SOMMA <sup>8</sup>	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 <sup>9</sup>	Récurrente		Champ d'activité
MSOM <sup>10</sup>	Récurrente		Poids du BMU
A-SOM <sup>11</sup>	Récurrente		Champ d'activité
Recursive SOM <sup>12</sup>	Récurrente		Champ d'activité
SOMSD <sup>13</sup>	Récurrente		Position du BMU
(Parisi et al. 2018)	Récurrente, Hiérarchique	Synchrone	Poids du BMU
(Baheux et al. 2014)	Récurrente, non hiérarchique	Synchrone	Position du BMU

1 (Lampinen et al. 1992; Hagenauer et al. 2013)(Paplinski et al. 2005)

2 (N. Liu et al. 2015),(Wickramasinghe et al. 2019)

4 (Lallee et al. 2013)

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

5 (Escobar-Juárez et al. 2016)

6 (Ménard et al. 2005; Khouzam et al. 2013)

7 (Magnus Johnsson et al. 2008)

8 (M. Lefort et al. 2015)

9 (Varsta et al. 2001)

10 (Strickert et al. 2005)

11 (Buonamente et al. 2013)

12 (Voegtlin 2002)

13 (Hammer, Micheli, Neubauer et al. 2005)

je ne sais pas comment t'as fait...  
mais c'est cool!