

Chapitre 1

Architectures de cartes auto-organisatrices

Nous nous concentrerons dans cette thèse sur la création d’architectures de cartes auto-organisatrices ou SOM (*Self-Organizing Maps*) comme modules d’une architecture décentralisée. Les cartes auto-organisatrices et notamment le modèle de Kohonen sont largement utilisées en tant qu’algorithme d’apprentissage non supervisé sur diverses applications, notamment la réduction de dimension et la visualisation de données. Cependant, peu de travaux ont exploré l’idée de les assembler en architecture comportant des rétroactions, formant ainsi un système dynamique. L’aspect bio-inspiré des cartes de Kohonen et la présence de motifs auto-organisés motivent pourtant cet axe d’utilisation en tant que système dynamique, motivations que nous présentons dans ce premier chapitre. Kohonen écrivait par exemple dans son livre à propos des enjeux des cartes de 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. (**Kohonen1995SelfOrganizingM**)

L’idée de trouver de nouveaux paradigmes de calculs non conventionnels dans les SOMs à l’aide de leur assemblage en structure modulaire dynamique rejoint l’idée de Kohonen sur les systèmes autonomes. Il soulève la question de l’interface entre les modules, qui sera centrale dans notre construction. 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’en extraire des comportements de calcul en émergeant. 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 attendus de l’architecture que nous proposerons.

Nous présentons ainsi dans ce chapitre le modèle général d’une carte de Kohonen et ses comportements de base ; notre modèle sera ensuite décrit plus en détail au chapitre ??.

passons ensuite en revue différentes architectures de SOM proposées dans la littérature. Nous analyserons notamment comment la structure d’une architecture définit le type d’apprentissage effectué par un système. 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. A la lumière des résultats des différents travaux, nous pourrions proposer des hypothèses sur les comportements que nous pouvons attendre de notre modèle d’architecture. Ces hypothèses motivent les expériences conduites dans la suite de nos travaux.

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 (**Kohonen1982**) ; nous utiliserons ainsi 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 initial, sur diverses applications. Nous présentons dans cette section le modèle de carte de Kohonen et détaillons les possibilités qu’il offre en tant que module d’une architecture.

1.1.1 Carte de Kohonen classique

Une carte de Kohonen est un algorithme de quantification vectorielle. Le but de la quantification vectorielle est de représenter un ensemble de données d’entrées issues d’un espace \mathcal{D} en un nombre fini de vecteurs de l’espace d’entrée, les prototypes. Dans une SOM, ces prototypes sont disposés sur les nœuds d’un graphe, en général une grille en deux dimensions. Les nœuds du graphe possèdent alors chacun un prototype et sont *indexés*, par 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 dans la carte et une notion de voisinage entre nœuds. Nous appellerons carte de Kohonen le graphe assorti de ses prototypes.

Au début de l’apprentissage, les prototypes prennent une valeur aléatoire dans l’espace d’entrée. L’apprentissage est ensuite réalisé en 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 Π . La distance d généralement utilisée est la distance euclidienne.
3. Le prototype de la BMU est déplacé vers l’entrée X , ainsi que les prototypes des nœuds voisins de Π dans un voisinage défini à l’avance. On peut interpréter cette étape comme le déplacement d’une zone de la carte centrée en Π . Des exemples de voisinages sont ainsi indiqués en figure ?? . Ce voisinage est défini par une fonction de voisinage 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, avec 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

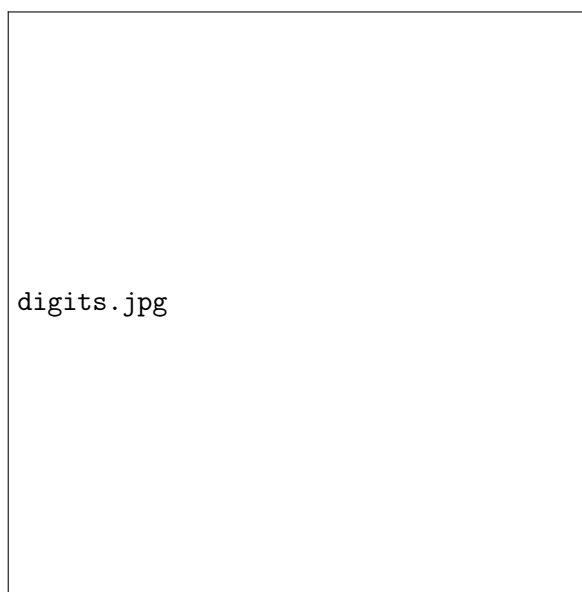


FIGURE 1.1 – Représentation de la base de données MNIST, images de chiffres écrits à main levées, par une SOM en deux dimensions. Une continuité est observée dans la forme des images lorsqu'on se déplace dans la carte : le 0 se transforme en 6, etc.

\mathcal{D} . L'utilisation de distances est la version originale de l'algorithme de Kohonen ; elle est remplacée dans de nombreux modèles de cartes par le calcul d'une fonction d'activité liant les poids des nœuds et les entrées.

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 ?? et ?? pour des exemples de cartes en une et deux dimensions, se dépliant sur des données en deux dimensions. A 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 ?? présente par exemple le dépliement d'une carte sur des imagerie MNIST. Par son aspect ordonné, une carte est une représentation en faible dimension d'un espace d'entrée de grande dimension.

1.1.2 Aspect topologique de la carte de Kohonen

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 du support de la carte. La plupart des implémentations de SOMs de la littérature utilisent comme support une grille en deux dimensions. L'indexation des nœuds est alors un ensemble de positions 2D.

En théorie, les cartes peuvent être une dimension (ligne), deux dimensions (grilles), ou de

FIGURE 1.2 – 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.

FIGURE 1.3 – Dépliement d’une SOM 2D sur des données dans le plan $[0,1]^2$, tiré de (Kohonen1995SelfOrganizingM)

dimensions plus grandes. Les cartes peuvent aussi être des graphes de forme plus variable. En pratique, les grilles deux dimensions sont les plus couramment utilisées. Elles 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 neurones, 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 que les cartes 2D. Les travaux conduits en (cottrell_theoretical_2016; fort_soms_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 fortement 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 avoir des avantages. Ainsi, des cartes structurées en arbre telles que développées en (koikkalainen_self-organizing_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_dynamic_2000; yamaguchi_adaptive_2010).

1.1.3 Inspiration biologique

Le développement des cartes auto-organisatrices par Kohonen est initialement inspiré par les cartes topologiques observées dans les aires du cerveau. Le cerveau est cartographié en *aires corticales* distinctes selon la fonction principale présumée de la zone du cortex correspondante. Le découpage fonctionnel du cerveau 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

FIGURE 1.4 – Dépliement d’une SOM 1D sur des données dans un triangle 2D, tiré de (Kohonen1995SelfOrganizingM)

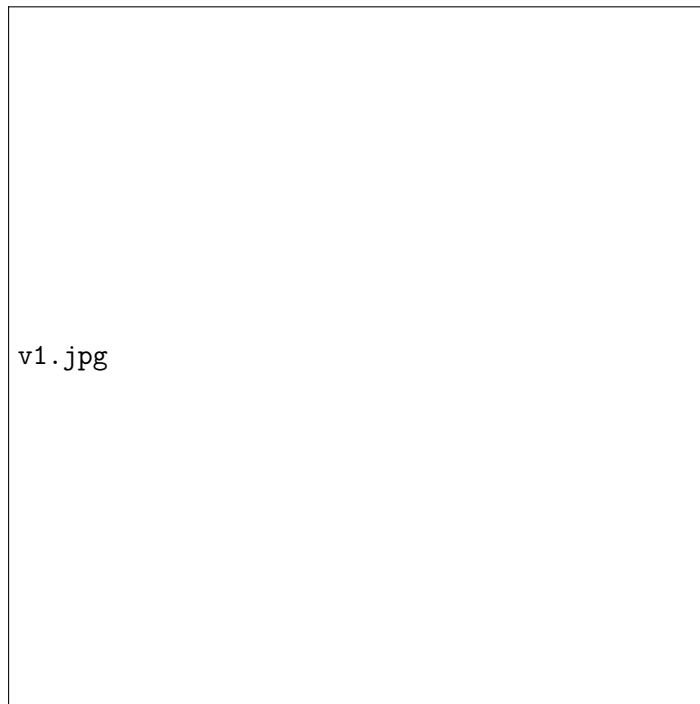


FIGURE 1.5 – Représentation des réponses du cortex visuel V1 à un stimulus visuel (bâtonnets d'orientations spatiales différentes). Les neurones répondant à une certaine orientation sont affichés de la même couleur. On observe une continuité entre les neurones proches dans le cortex et l'orientation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

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 ?? . L'aire associée à l'audition présente aussi une organisation topographique (**Reale1980TonotopicOI**), ainsi que de nombreuses autres aires, directement sensorielles ou plus abstraites (**Kohonen1995SelfOrganizingM**). 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.

Les aires du cerveau sont connectées entre elles. Notons que cette connectivité du cerveau peut-être étudiée de plusieurs points de vue : d'un point de vue structurel, en se basant sur des éléments anatomiques ou 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. Des modèles communs

pour cette communication entre neurones sont par exemple la zone de convergence-divergence de Damasio (CDZ) (**damasio_time-locked_1989**), et le modèle de boucles de ré-entrées de Edelman (**Edelman1982GroupSA**). La zone de convergence-divergence suggère que certaines aires corticales servent d'aires associatives pour associer d'autres zones corticales prenant des modalités sensorielles en entrée. Ces aires associatives rassemblent les signaux en provenance des zones sensorielles et les propagent vers d'autres zones. La théorie de la ré-entrée postule quant à elle des connexions directes et réciproques entre les neurones de différentes zones sensorielles ou non. Ces connexions sont à l'origine de la coactivation de neurones dans différentes cartes.

La carte de Kohonen implémentant des concepts computationnels qu'on retrouve en biologie au niveau de l'aire cérébrale, nous pouvons chercher à pousser l'inspiration biologique d'une carte de Kohonen au niveau des connexions entre les aires cérébrales, se transcrivant par des connexions entre plusieurs cartes de Kohonen. De la même façon qu'une carte n'est pas un modèle biologique, il s'agit plutôt de développer un modèle computationnel qui ne soit pas biologiquement plausible au niveau neuronal, mais dont la structure du traitement de l'information est inspirée de celle du cerveau, ici la présence de plusieurs aires connectées entre elles, modélisées par l'utilisation de plusieurs cartes de Kohonen en architecture.

1.2 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 et de créer des algorithmes pour résoudre ces sous-problèmes. Par construction, l'assemblage des résultats de chacun de ces sous-problèmes permettrait de résoudre le problème général. Nous différencions cette vision d'architecture appliquée de la notion d'architecture modulaire générique. Dans ce deuxième cas, la notion d'architecture cherche à proposer un cadre de calcul utilisable sur n'importe quelles données d'application. C'est cette deuxième approche que nous étudions dans cette thèse.

A la revue des modèles existants, nous distinguerons deux types d'architectures de cartes : les architectures *hiérarchiques* et les architectures *non-hiérarchiques*. La connectivité d'une architecture de cartes auto-organisatrice peut se représenter comme un graphe orienté, dans lequel les nœuds correspondent à 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 lorsque son graphe de connexion peut se représenter comme un arbre. Toutes les connexions sont alors dans un même sens : de la racine vers les feuilles ou inversement. 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 sur plus de nœuds. Ces boucles implémentent des rétroactions entre les cartes de l'architecture. Ces deux types d'architectures se ressemblent dans

leur conception : notons qu'une architecture hiérarchique se classe comme un cas particulier d'architecture hiérarchique. Néanmoins, les architectures non-hiérarchiques doivent répondre à plus de problèmes de construction, notamment la gestion des rétroactions dans les méthode de communication et d'apprentissage. Ainsi, nous observerons une différence de domaine d'application entre architecture hiérarchiques et non-hiérarchiques.

Ainsi, la structure d'une architecture influence son champ d'application et ses capacités de calculs. Nous analyserons les architectures de cartes sous le prisme de leur structure, et en particulier leur mode de connexion entre cartes. Cette communication est réalisée soit par une surcouche algorithmique, soit 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 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 éléments de sortie des autres cartes, telles que la position du BMU, son poids ou une activité neuronale. Enfin, la temporalité de l'algorithme de mise à jour de l'architecture peut se présenter de plusieurs 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 alors effectué sur un groupe avant de faire l'apprentissage des groupes suivants. Les mises à jour peuvent au contraire être réalisées en une seule étape, lors de laquelle les éléments sont tous mis à jour en tenant en compte des dépendances. Ces mises à jour peuvent être synchrones : une itération d'apprentissage peut être défini pour toute l'architecture, lors de laquelle tous les éléments seront mis à jour, ou encore asynchrone : les mises à jour sont effectuées au fil de l'eau. C'est par exemple le cas des réseaux de neurones impulsionnels, dans lequel les paramètres d'un neurone sont mis à jour seulement lorsqu'une impulsion lui parvient. Nous verrons ainsi que les modèles existants d'architecture de cartes se placent tous 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 d'un tel modèle et ainsi de guider les expériences que nous effectuerons.

1.2.1 Architectures hiérarchiques de cartes

Dès les débuts du développement des SOMs dans les années 1980 des travaux proposent des architectures à base de cartes auto-organisatrices. Ces premiers travaux font apparaître des architectures qu'on peut qualifier de hiérarchiques. Toutes les architectures hiérarchiques comportent plusieurs cartes, et ont comme point commun qu'il est possible de définir des *niveaux* dans l'architecture, comprenant une ou plusieurs cartes. L'apprentissage des cartes dans toutes ces architectures est effectuée niveau par niveau : nous nous plaçons dans les architectures à mise à jour séquentielles. Nous les qualifierons ainsi d'architecture *ascendantes*, dans la mesure où le premier niveau est appris lors d'une première phase d'apprentissage, puis le second niveau est appris en s'appuyant sur les résultats du premier.

Architectures sélectives

Certains travaux s'appuient sur des architectures hiérarchiques qu'on qualifiera de "sélective". Toutes les architectures présentées peuvent être séparées en niveaux, dont un niveau d'entrée, un niveau de sortie et des niveaux intermédiaires. L'exemple de l'architecture sélective (**barbalho_hierarchical_2001**) est ainsi illustrée en figure ?? ; ce procédé se retrouve dans d'autres architectures.

Par exemple, l'architecture développée en (**barbalho_hierarchical_2001**) ainsi que sa version dynamique (**Costa2016ANS**) est une architecture hiérarchique sélective. Le premier niveau de l'architecture est une carte classique, prenant des données X dans un espace d'entrée \mathcal{D} . La première étape de l'apprentissage est une étape d'apprentissage complet de la carte du premier niveau. Après apprentissage du premier niveau, le second niveau est composé de plusieurs cartes ; chacune de ces cartes est associée à un des noeuds de la première. Pour 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_t ayant i pour position du BMU associé à l'entrée. Cette répartition est effectuée lors d'un passage de test des entrées, pendant lequel les poids ne sont pas mis à jour. Chaque carte i du deuxième niveau est alors entraînée sur son espace \mathcal{D}_i ; les cartes du premier niveau n'étant plus mises à jour. L'ensemble des cartes, non seulement le dernier niveau, forme alors une cartographie plus précise de l'espace d'entrée. Dans la version de (**barbalho_hierarchical_2001**), l'architecture de carte est définie à l'avance. Dans la version dynamique, l'ajout de niveau hiérarchique est soumis à la condition que la carte enfant d'un nœud doit permettre de réduire d'au moins 20% l'erreur de quantification sur le sous-espace associé. Autrement dit, si tous les points ayant le même BMU sont à proximité immédiate du poids du BMU, aucune carte n'est générée pour le niveau suivant. Dans cette version dynamique, les auteurs font également varier la taille de la carte d'un niveau en fonction de la cardinalité du sous-ensemble de données utilisé pour l'apprentissage de cette carte.

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. Dans ces travaux, la position du BMU est utilisée comme information pour la sélection. Dans ce type d'architecture, c'est un processus extérieur aux cartes qui permet de sélectionner la carte du niveau suivant. Ce processus extérieur est encore plus utilisé dans la version dynamique : le processus extérieur permet de choisir des cartes et de créer ou non des cartes à ajouter à l'architecture. L'architecture est ainsi sous la dépendance d'un processus unique.

Ce procédé sélectif est retrouvé dans d'autres architectures telles que (**suganthan_pattern_2001**), (**miikkulainen_script_1992**) pour de la classification de phrases, (**dittenbach_growing_2000** ; **ordonez_hierarchical_2010**), ou encore en (**zhao_stacked_2015**) pour de la détection d'arrière-plan. La façon de sélectionner une carte d'un niveau différent ensuite en fonction des travaux.

L'architecture utilisée en (**zhao_stacked_2015**) utilise plusieurs couches de cartes pour du traitement de séquence. Les auteurs, pour chaque niveau, Leur architecture est spécifiquement développée ici pour l'application de détection d'arrière-plan.

Les connexions permettent à chacune des cartes de se spécialiser sur une partie de l'espace d'entrée.

Face à ces structures sélectives, certaines structures présentent une interface entre cartes

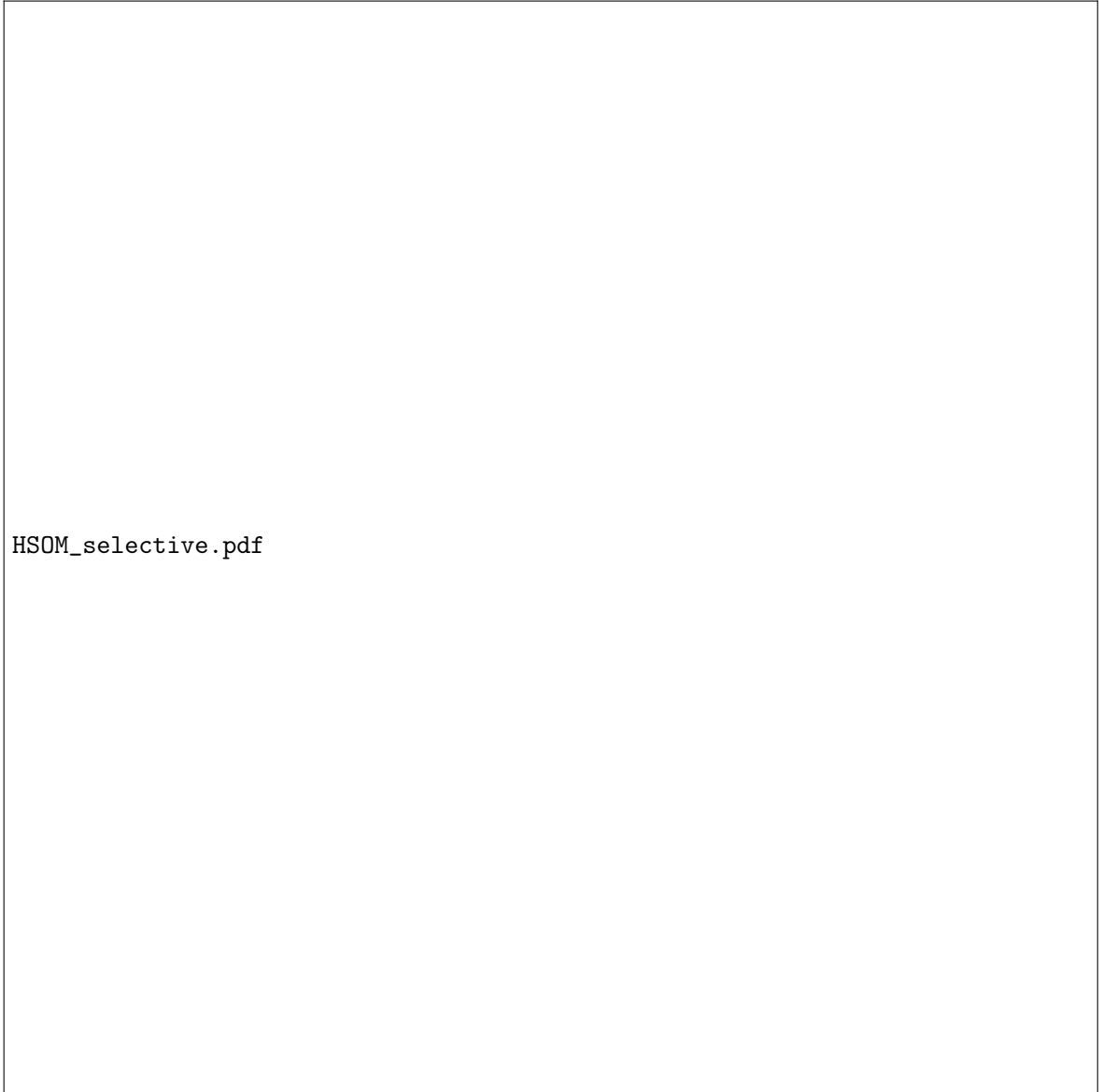


FIGURE 1.6 – Exemple d’architecture hiérarchique sélective. La carte du premier niveau est entraînée sur tout l’espace d’entrée. Après apprentissage, la carte permet de filtrer les entrées pour les renvoyer vers une carte du niveau suivant. Dans cet exemple, la position du BMU de la carte du niveau 1 permet de sélectionner une carte du niveau 2, comme c’est le cas en (**barbalho_hierarchical_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 $\mathcal{D}_{1,1}$ lié au BMU à la position $(1, 1)$ alimente alors une carte du deuxième niveau $M_{(1,1)}$.

gérée directement au niveau du processus d'auto-organisation de la carte. Cette gestion des connexions qui était collective dans les architectures sélectives devient locale : une carte a seulement accès aux éléments d'information transmis. Ce traitement local de l'information se rapproche plus de notre approche modulaire.

Architectures hiérarchiques par transmission de représentation intermédiaire

Les architectures que nous positionnons dans cette section restent hiérarchiques et ascendantes. Les connexions entre carte s'apparentent à un arbre.

Contrairement aux architectures par sélection, la deuxième couche de carte ne prend plus comme entrée un élément de l'espace d'entrée de l'architecture, mais travaille sur 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 intermédiaire de l'entrée, transmise à la couche supérieure. Les niveaux supérieurs de ce type d'architecture ont moins de cartes que les premiers niveaux et peuvent être considérés comme traitant l'information à un niveau plus abstrait que les cartes du premier niveau.

L'architecture HSOM (**lampinen_clustering_1992**) proposée dès 1992 est composée de deux cartes : une carte apprenant sur des entrées X , et une carte prenant comme entrée la position du BMU Π de la première carte ; cette architecture est illustrée en figure ?? . La position du BMU est ici la représentation intermédiaire transmise aux cartes du niveau suivant. 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 groupe de données (cluster) auront des BMUs proches dans la première carte ; par conséquent, leurs BMUs dans la seconde carte le seront également. Les auteurs notent que l'architecture HSOM permet de bien détecter des clusters de données, avec une séparation des clusters meilleure qu'une SOM classique. La présence de la carte intermédiaire change la façon de séparer les données par rapport à une SOM classique ; les auteurs notent une meilleure séparation des clusters dans HSOM, mais une moins bonne quantification vectorielle au sein d'un cluster. Le fait d'utiliser deux 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_hierarchical_2013**). Dans leurs travaux, les auteurs implémentent une architecture en arbre. Une carte d'un niveau reçoit alors un vecteur de position de BMUs du niveau inférieur en tant qu'entrée.

Autant le type d'information transmis entre carte avait assez peu d'influence dans les exemples d'architecture sélective, autant il devient un point central dans le cas des architectures par transmission de représentation. Cette information transmise doit permettre de véhiculer un maximum d'information entre cartes, tout en étant interprétable par les autres cartes. Les architectures HSOM utilisent la position du BMU en tant que représentation. Par le choix de la position du BMU comme vecteur de transmission d'information, les auteurs de HSOM exploitent 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. Aisni, d'autres travaux, tels que ceux conduits en (**wang_comparisnal_2007** ; **gunes_kayacik_hierarchical_2007**) utilisent plutôt le poids du BMU comme transmission d'information.

On observe un regain de publications sur les architectures de cartes auto-organisatrices après 2015, cette fois sous la terminologie de *Deep SOM*. La plupart des travaux portant sur ces *Deep SOM* implémentent des cartes hiérarchiques similaires aux travaux mentionnés précédemment. Mais cette fois, ces travaux s'inspirent des réseaux de neurones profonds (*Deep Learning*), ayant connu leur essor cette même année avec les réseaux convolutionnels permettant l'apprentissage supervisé d'images de façon inégalée ([lecun_deep_2015](#)). Par analogie avec les réseaux convolutionnels, les *Deep SOM* s'intéressent souvent à l'apprentissage d'images par des SOMs. Ainsi ([Liu2015DeepSM](#) ; [hankins_somnet_2018](#) ; [wickramasinghe_deep_2019](#) ; [aly_deep_2020](#) ; [sakkari_convolutional_2020](#) ; [dozono_convolutional_2020](#) ; [nawaratne_hierarchical_2020-1](#) ; [mici_self-organizing_2018](#)) et sont présentés comme des "SOMs convolutionnelles". La façon d'associer ces cartes en architectures rappellent cependant les SOMs hiérarchiques.

Par exemple, le modèle introduit en ([Liu2015DeepSM](#)) est illustré en figure ?? et s'inspire des réseaux de neurones convolutionnels. Le but d'une telle architecture est de classifier des images. Une fenêtre est déplacée sur l'image d'entrée et chaque imagelette nourrit alors une carte d'une première couche, donnant $N_{maps} \times N_{maps}$ positions de BMU $j_{p,q}$. 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. Une deuxième étape de fenêtrage peut alors être appliquée sur cette image, et ainsi de suite. La dernière couche du réseau est composée d'une SOM qui effectue alors la tâche de classification de l'image intermédiaire, vue comme une représentation abstraite de l'entrée. L'interface entre les couches "convolutionnelles" est créée à partir des BMUs des SOMs : l'architecture DSOM s'inscrit ainsi directement dans la lignée de HSOM, à la différence qu'un vecteur de positions de BMUs est utilisé comme entrée pour la couche suivante et non la position d'un seul BMU. Les auteurs montrent que ce modèle est meilleur qu'une SOM classique dans des tâches de classification sur MNIST ; les couches supérieures apportent un niveau d'abstraction tandis que les couches inférieures apprennent les motifs.

Toutes les architectures présentées ici, comme pour les architectures sélectives, sont ascendantes : les couches sont entraînées les unes après les autres. La plupart des travaux sont appliqués à des tâches de classification qu'on pourrait réaliser avec une SOM classique. La motivation d'utiliser une architecture plutôt qu'une SOM est alors que la couche finale du réseau possède de meilleures performances en termes de classification que si on avait utilisé une SOM simple.

Nous pouvons conclure de ces travaux qu'il existe différentes façons de transmettre des informations entre couches. Les plus courantes sont la transmission de la position du BMU comme entrée pour la carte suivante, ou un calcul d'activation prenant en compte l'activation de la couche précédente par une connexion neurone à neurone. La transmission de la position du BMU dans HSOM, *Deep SOM* et de nombreux autres travaux nous semble particulièrement intéressante car elle exploite totalement l'aspect topologique d'une carte de Kohonen. Par ailleurs, il s'agit d'une valeur de faible dimension, donc se prêtant à des calculs. Les architectures telles que *Deep SOM* nous montrent qu'une architecture utilisant seulement ce type d'information comme interface est capable de bonnes performances en reconnaissance d'image. Le champ d'application des architectures ascendantes est le même qu'une SOM classique : quantification vectorielle et classification. Leurs performances dans ces domaines surpassent alors celles d'un SOM classique, la présence de couche multiples créant un nouveau niveau d'abstraction.

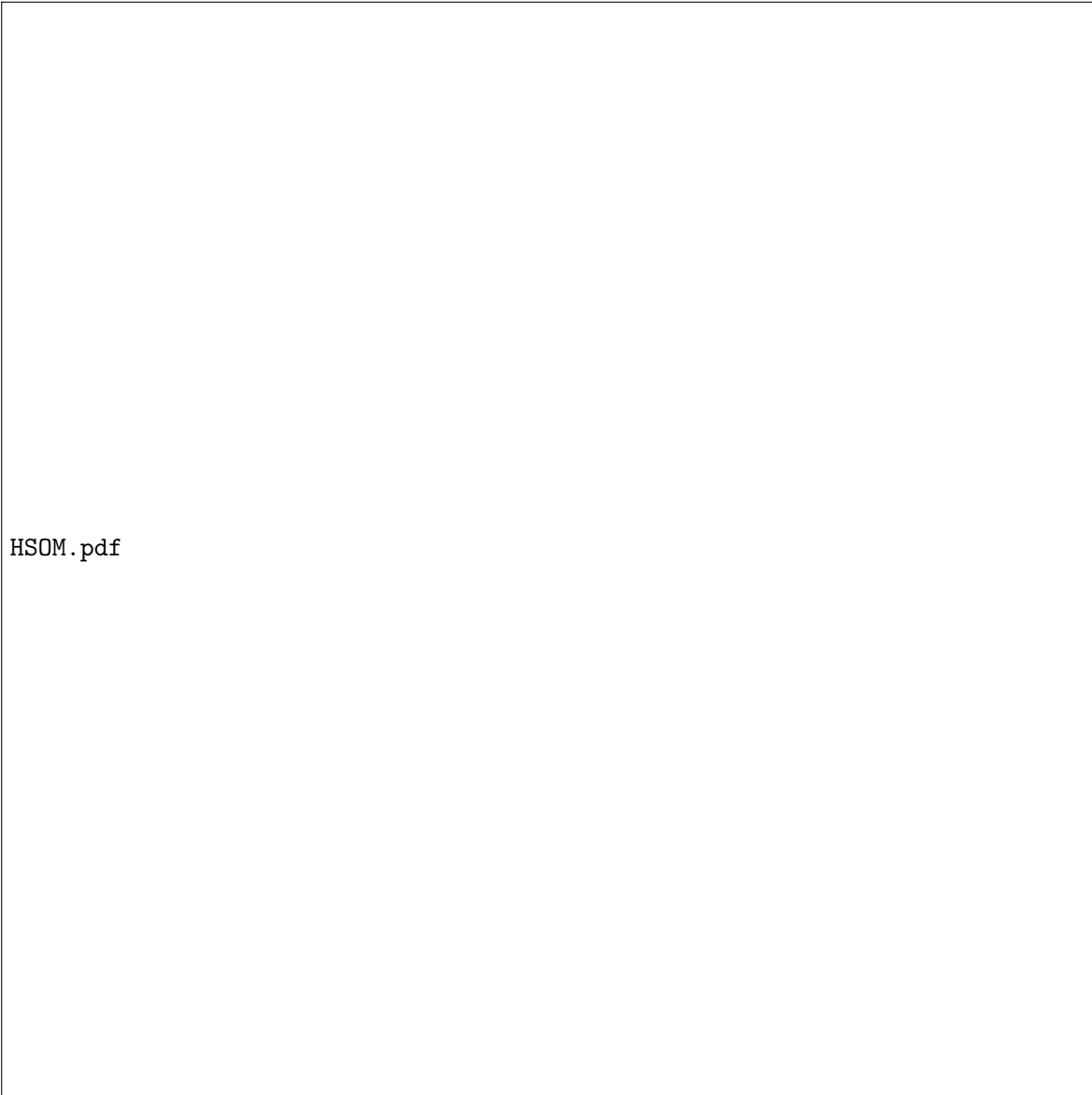
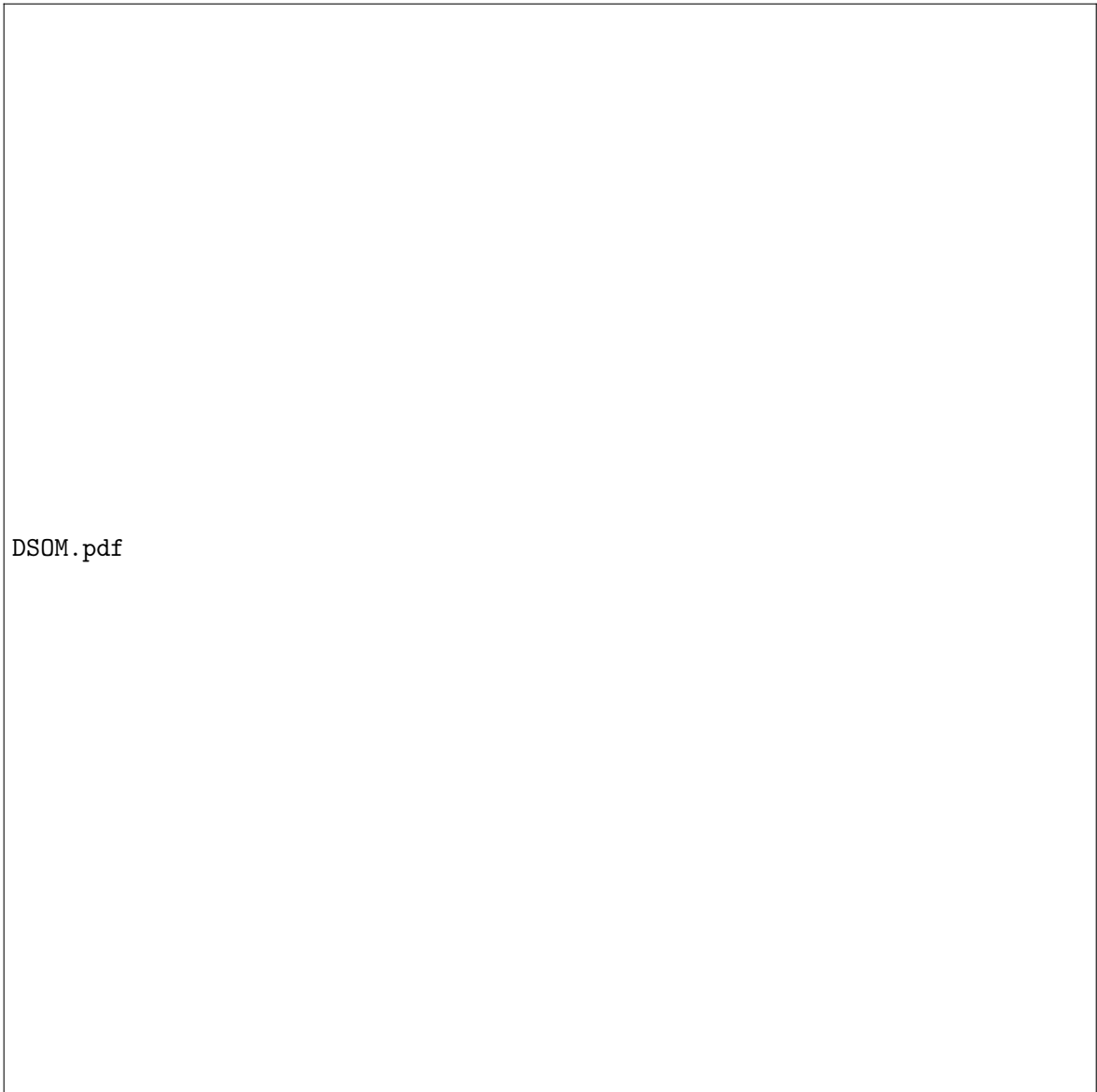


FIGURE 1.7 – Architecture HSOM (**lampinen_clustering_1992**). L'apprentissage des positions du BMU de la première couche par la seconde permet de mieux détecter les ensembles de données, dans une tâche de clustering. La deuxième couche est vue comme un niveau plus abstrait que la première.



DSOM.pdf

FIGURE 1.8 – Architecture DSOM de SOM "convolutionnelle" (**liu_deep_2015**). Les auteurs utilisent les positions des BMUs de la couche de carte $j_{p,q}$ comme valeurs d'entrée pour les couches suivantes, dans la lignée de HSOM (**lampinen_clustering_1992**). Les couches sont entraînées les unes après les autres. Cette architecture est *feedforward* et *hiérarchique* par transmission de représentation intermédiaire.

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 feed-forward en est la cause : les cartes intermédiaires agissent comme des filtres intermédiaires de l'information donnée en entrée, mais la couche finale reste une carte auto-organisatrice classique. Cela nous amène à étudier des architectures comportant des boucles de rétroaction ; nous qualifierons ces architectures de non-hiérarchiques. Ces architectures, nous le verrons, 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.

1.2.2 Structures de cartes auto-organisatrices non-hiérarchiques

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. Ces architectures sont souvent proposées par une motivation d'inspiration biologique, dans le domaine des neurosciences computationnelles, en s'appuyant sur les théories mentionnées précédemment de la réentrée (**Edelman1982GroupSA**) et des zones de convergence-divergence (CDZ) (**damasio_time-locked_1989**). L'assemblage de réseaux de neurones en architecture fait l'objet de plus nombreux travaux dans le domaine des neurosciences computationnelles. Nous ne détaillerons pas ces travaux ici, nous intéressant spécifiquement aux modèles computationnels destinés à être appliqués en informatique et non modélisant finement la biologie.

Nous avons pu distinguer deux types d'architectures non-hiérarchiques dans l'état de l'art. 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 feed-forward que nous avons listé 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.

Un point commun de toutes ces architectures non-hiérarchiques est leur champ d'application : contrairement aux architectures hiérarchiques feed-forward 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 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.

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 (**dominey13**) et (**escobar-juarez_self-organized_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. Chacune des cartes possède plusieurs couches, chacune prenant une modalité en entrée. Une activité est calculée sur ces modalités en une activité commune. La position du BMU, en l'occurrence un vecteur 3D, sera utilisée comme modalité hiérarchique pour connecter des cartes entre elle. Les auteurs assemblent alors plusieurs cartes sensorielles grâce à des cartes associatives. Dans le cas d'une hiérarchie de cartes, les auteurs entraînent les couches de l'architecture séparément : les cartes modales du premier niveau sont apprises, puis la carte amodale les connectant est apprise dans un second temps. Une fois toutes les cartes apprises, la structure est utilisée pour activer une ou des modalités en activant la carte amodale. Cette carte représente la zone de convergence divergence des modèles cérébraux. Dans ce modèle, les cartes sensorielles sont d'abord entraînée, puis les cartes associatives sont apprises dans un second temps. Nous l'avons quand même classée comme non-hiérarchique, car une fois que toutes les connexions sont apprises, elle permet des rétroactions entre cartes.

Les travaux conduits précédemment dans notre équipe de recherche sur les architectures de cartes auto-organisatrices se classent dans cette catégorie. Ainsi, l'architecture Bijama développée en (**menard05 ; khouzam_2013**), associe des cartes modales par une carte associative par des connexions neurone à neurones, en renforçant les connexions entre les groupes de neurones s'activant au même moment dans les cartes modales, via la carte associative. Notons que Bijama repose sur des calculs complètement locaux, y compris au sein d'une carte auto-organisatrice.

Les modèles mentionnés ci-dessus rentrent dans la catégorie non-hiérarchique pour leur possibilité d'activation d'une carte par l'autre. Encore une fois, la position du BMU apparaît chez (**dominey13**) comme le vecteur de transmission d'information entre cartes et suffit pour que la coactivation des cartes permettent de réaliser de la mémoire associative. Le modèle SOIMA et le modèle Bijama privilégient la connexion neurone à neurone entre la carte associative et la carte modale, avec une règle d'apprentissage Hebbienne. Cette mémoire associative est utilisée dans un cadre de données multimodales, avec une notion d'activation d'une carte par l'autre, contrairement aux architectures hiérarchiques citées en section précédentes, utilisées plutôt pour des tâches de classification, autrement dit des tâches supervisées. Dans ces exemples architectures présentées ici, on considère les cartes comme des représentations de leur espace de données qui permettent de la coactivation entre cartes : une carte de Kohonen prend une fonction générative.

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.

Face aux architectures centralisées, nous pouvons imaginer des architectures décentralisées implémentant des connexions directes entre cartes modales.



FIGURE 1.9 – Exemple général d’architecture décentralisée comportant une carte associative. Ces architectures sont utilisées dans des tâches de traitement de données multimodales. Des cartes appelées cartes *sensorielles* ou *modales* prennent des entrées dans plusieurs modalités. Une carte *associative* reçoit des connexions montantes de ces cartes et apprend à associer les activités. Les cartes sensorielles sont connectées à la carte associative par des connexions descendantes pouvant générer une activation dans la carte. Dans la plupart des modèles, les connexions montantes et descendantes n’ont pas le même rôle : les cartes sensorielles ne s’influencent pas entre elles lors de l’apprentissage. Lors de l’utilisation de l’architecture pour de la génération d’entrée sensorielle, alors les connexions descendantes permettent à la carte associative d’activer une carte sensorielle.

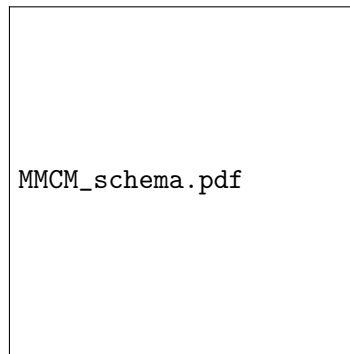


FIGURE 1.10 – Architecture MMCM. Les cartes du premier niveau reçoivent l’une les mouvements de tête d’un robot, l’autre les mouvements de main. Une carte amodale (MMCM) reçoit les positions des BMUs de chaque carte du premier niveau. Chaque carte du premier niveau possède également une couche "hiérarchique" prenant en entrée les positions du BMU de MCMM. Chaque niveau est entraîné séparément. Après apprentissage, l’activation de la carte MCMM produit des mouvements coordonnés tête/mains (**dominey13**)

Architectures non-hiérarchiques décentralisées

Nous avons relevé peu de travaux portant sur des architectures décentralisées, c’est-à-dire comportant des connexions directes entre cartes sensorielles. Un exemple générique d’architecture décentralisée est tracée en figure ?? . Les architectures proposées dans tous les travaux que nous avons relevés sont appliquées à de l’apprentissage multimodal.

Une première façon d’associer des cartes de Kohonen est de connecter les neurones d’une carte aux neurones d’une autre par des connexions pondérées. C’est par exemple ce qui est proposé dans (**khacef_brain-inspired_2020**). Les auteurs utilisent ici des cartes de Kohonen impulsives, plus directement inspirées de la biologie que les cartes de Kohonen classiques, mais les mêmes principes d’auto-organisation se retrouvent entre les deux modèles. Dans ces travaux, les auteurs apprennent ainsi deux modalités d’un espace multimodal sur deux cartes auto-organisatrices, l’image et le son, puis dans un second temps apprennent les connexions entre neurones en mettant leurs poids à jour à partir des 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 montrent que cette architecture permet d’apprendre les associations d’entrées et peut générer une entrée à partir de l’autre. Ce modèle est très simple mais sa mise à jour doit être réalisée en plusieurs étapes : d’abord les poids des cartes, puis les poids des connexions. Cet apprentissage en deux temps était également présent dans l’architecture centralisée de (**dominey13**).

Nous remarquons par contre deux modèles qui cherchent à créer une architecture décentralisée de carte dont l’apprentissage est réalisé d’une seule traite.

Une telle version d’architecture de cartes non-hiérarchiques est développée en (**johnsson_associating_johnsson_associative_2009**), sous le nom de A-SOM, *associative self-organizing map*. Encore une fois, le but d’une telle architecture est de faire de l’apprentissage multimodal, en apprenant à associer les activités de cartes sur différentes modalités. La particularité de A-SOM, par rapport à tous les modèles précédemment étudiés, est que l’apprentissage de toutes les cartes et de leurs interactions est réalisé simultanément. Ce modèle décentralisé inclut la



FIGURE 1.11 – Exemple général d’architecture décentralisée de cartes. Chacune des cartes de l’architecture prend une entrée sensorielle A,B,C. Des connexions entre cartes permettent l’apprentissage d’associations entre modalités. Chacune des cartes peut donc être vue comme une carte multimodale. La façon de gérer les rétroactions entre cartes varie en fonction des travaux et est une problématique majeure dans la construction d’une telle architecture. Ainsi, les cartes apprennent à associer leurs activités seulement après avoir appris les modalités en (khacef_brain-inspired_2020) et conjointement en (johnsson_associative_2009) et (baeux_towards_2014).

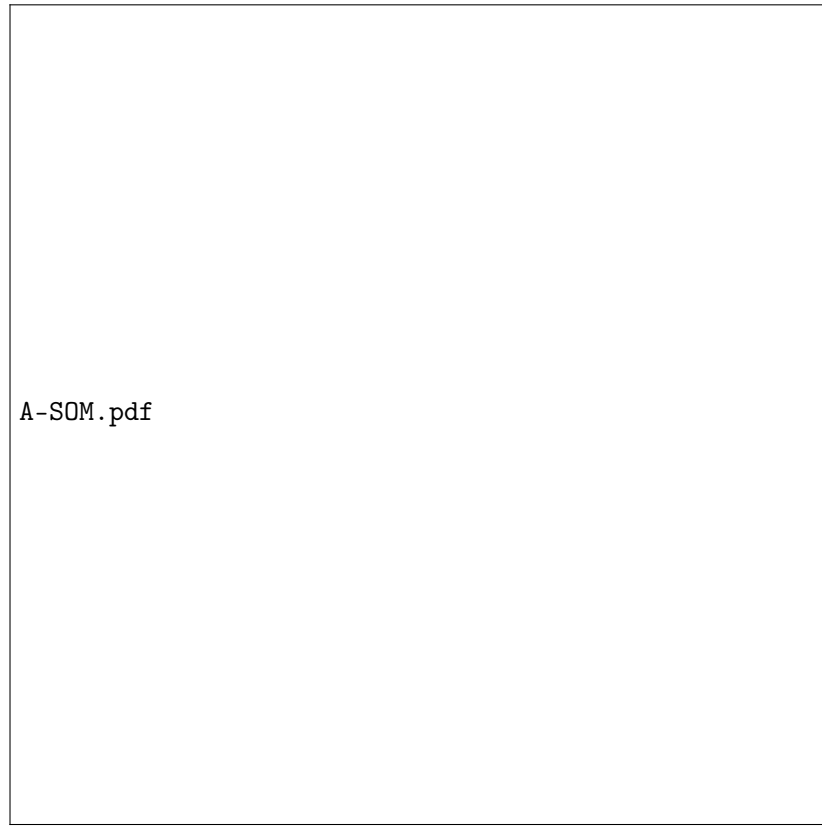


FIGURE 1.12 – Le modèle A-SOM ([johnsson_associative_2009](#)) associe les activités de différentes cartes. Chacune des cartes prend une modalité A,B,C en entrée. Contrairement aux modèles précédemment cités, les trois cartes apprennent simultanément. L’association est prise en compte lors du calcul des activités de chaque carte.

possibilité de créer une version d’architecture centralisée à partir des règles d’associations, par exemple ([buonamente_hierarchies_2016](#)). Ainsi, un modèle d’architecture décentralisé est intéressant pour la recherche de nouveaux comportements dans la mesure où il n’impose 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. Une carte du modèle A-SOM possède plusieurs couches de poids. L’une de ces couches est relative à l’entrée de la SOM provenant du capteur. L’autre couche de poids reçoit une entrée provenant d’une autre carte de l’architecture. L’information transmise ici est l’activation de tous les neurones de l’autre carte. chaque carte reçoit ainsi $N \times M$ valeurs d’activité. L’interface entre cartes utilisée en A-SOM est donc le champ d’activation des neurones.

L’architecture développée en ([lefort_active_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.

Des modèles multicartes incluant des connexions temporelles

Certains modèles s'appuient sur plusieurs cartes de Kohonen connectées, en y ajoutant une notion de traitement de séquences. en (**parisiLL**), les auteurs développent une architecture de deux réseaux auto-organisés appelés *grow when required networks* (GWR). 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, dans laquelle des connexions temporelles entre neurones sont mises à jour en supplément des poids associés aux neurones. Le BMU est alors choisi en fonction de l'entrée courante ainsi que des BMUs précédents. La deuxième carte est une version classique du GWR. Elle prend en entrée une séquence de BMUs de la carte G-EM, ainsi que la classe de la séquence courante, afin de mettre à jour ses poids. Cette architecture associe ainsi des connexions temporelles récurrentes sur une carte ainsi que des connexions entre cartes. Cette architecture permet des tâches de *lifelong learning*. Le concept d'apprentissage sur le long terme s'intéresse à des systèmes étant mis à jour en ligne, dès qu'ils reçoivent une entrée, et dont l'apprentissage n'a pas de limite temporelle fixée. On doit donc avoir un système qui trouve de lui-même une stabilité dans l'apprentissage et qui est capable de s'adapter à de nouvelles entrées. Dans la plupart des applications en robotique, les entrées présentées à une structure d'apprentissage sont par ailleurs des entrées ayant une relation temporelle. Deux images recues successivement par un capteur visuel seront proches dans l'espace des images. Pour une SOM par exemple, cela pose problème. Les architectures de *lifelong learning* 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 les données vues au début de l'apprentissage. 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, etc. À terme, seule la dernière classe apprise est gardée en mémoire.

La motivation de ces modèles multicartes est intéressante : il s'agit cette fois de voir les deux cartes comme de l'apprentissage à différentes échelles temporelles. L'architecture mélange connexions récurrentes et connexions inter-cartes, ce qui est pertinent dans le cadre de l'apprentissage de séquence, et dans le but de création de systèmes autonomes de cartes auto-organisatrices évoquées par Kohonen. Cependant, si sa motivation nous intéresse, le modèle décrit précédemment utilise une logique de vérification externe aux cartes pour ajouter ou nommer des neurones.

Notre démarche s'inscrit dans un objectif d'allier connexions temporelles et connexions intercartes au sein d'une même architecture, sans forcément avoir à différencier ces connexions. Les travaux autour du modèle A-SOM mentionné précédemment en ont ainsi dérivé une version récurrente (**Buonamente2015DiscriminatingAS**). Cette version récurrente est similaire à la version multicarte. Au lieu de considérer l'activité d'une autre carte pour le calcul de l'activité courante d'une carte, les auteurs utilisent l'activité de la même carte à l'état 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. La motivation de développer une version récurrente de A-SOM est alors de pouvoir développer des architectures comportant connexions récurrente et connexions intercartes. Nous n'avons pas encore relevé cependant de travaux les intégrant effectivement dans une architecture multicartes.

Les travaux menés en ([baheux_towards_2014](#)) ont également cherché à insérer des connexions temporelles au sein d'une architecture de deux cartes, présentée en ???. Chaque carte est composée de deux couches de poids. Une des cartes prend une entrée correspondant à l'observation courante, relative à une première couche de poids, 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 de la deuxième carte. La seconde carte reçoit deux entrées de la première : une entrée est la position du BMU de l'état précédent et la seconde la position du BMU de l'état courant. Une activité est calculée sur chaque couche de poids relativement à son entrée et ces deux activités sont fusionnées en une activité globale à chaque carte. Comme chaque carte reçoit en entrée la position de l'état courant du BMU de l'autre carte, il existe une boucle de rétroaction entre carte. Les auteurs laissent alors "résonner" les activités en déplacement petit à petit les BMUs de chaque carte, jusqu'à obtenir un état stable pour les activités, qui est utilisé 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.

Ces exemples nous amènent à pousser la bibliographie de cette thèse sur les cartes de Kohonen récurrentes. On voit en effet la similarité existant dans le traitement de séquences, ou le choix d'un BMU doit prendre en compte le contexte des états précédents, aux modèles multimodaux dans lequel le choix d'un BMU doit prendre en compte le contexte des entrées d'autres modalités.

1.2.3 Cartes auto-organisatrices récurrentes

On appelle réseaux récurrents des réseaux de neurones qui prennent en compte leur état précédent pour calculer leur état actuel. Ces réseaux sont utilisés pour le traitement et l'apprentissage de signaux temporels. Citons par exemple, en Deep Learning, les RNN (*recurrent neural networks*), dont les neurones reçoivent leur état précédent en entrée. Les cartes de Kohonen ont ainsi également des versions récurrentes.

Nous nous intéressons aux architectures multicartes, mais les cartes récurrentes répondent à des problèmes très similaires à ceux rencontrés dans la conception d'architectures de cartes pour faire de la mémoire associative, comme vu en section précédente. Dans une carte récurrente, le problème principal est de trouver comment communiquer à la carte de l'information sur son état précédent et comment utiliser cette information dans l'apprentissage de l'état courant. Cela rejoint la problématique posée dans les architectures de cartes de Kohonen, qui est de comment communiquer à une autre carte son état, afin de l'utiliser dans l'apprentissage de l'état courant. L'étude des modèles de cartes récurrentes existant nous permettrons de compléter cette bibliographie.

Notre volonté de créer un modèle général d'architecture de cartes auto-organisatrice motive

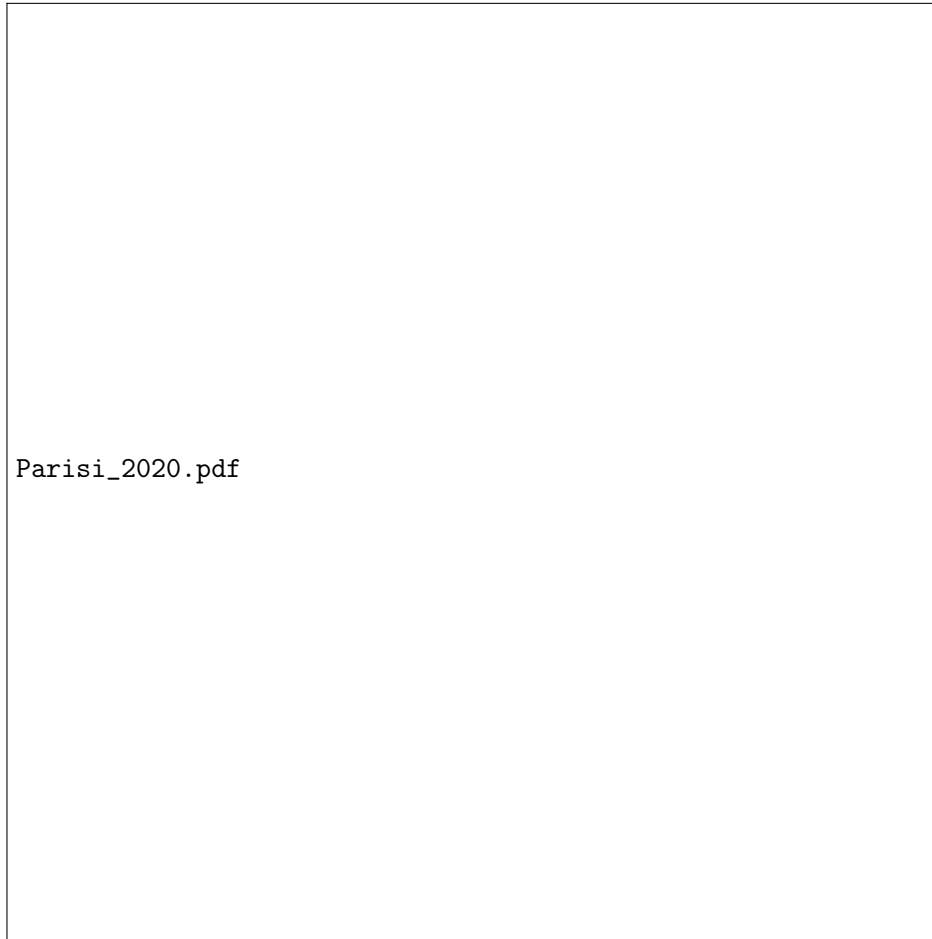


FIGURE 1.13 – Architecture à *double mémoire* proposée en (**parisiLL**). La couche de mémoire épisodique est un réseau *grow when required (GWR)*, un réseau auto-organisé similaire à une carte de Kohonen, à la différence que les neurones sont ajoutés au fur et à mesure de l'apprentissage. La couche de mémoire sémantique est également un réseau GWR, entraîné à partir des BMUs de la couche épisodique et de la classe de la séquence jouée. L'architecture apprend à reconnaître plusieurs séquences.

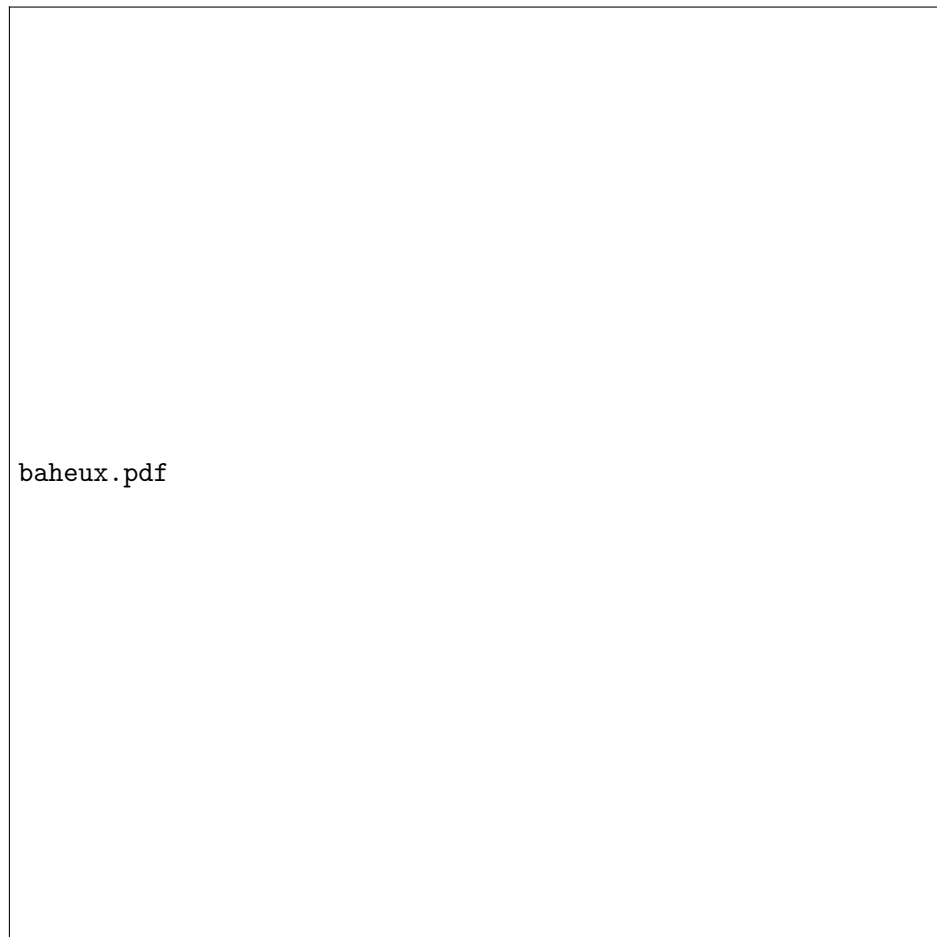


FIGURE 1.14 – Structures de deux cartes auto-organisatrices communicantes, (baheux_towards_2014). Chaque carte est composée de trois couches d'activités, représentées séparément sur le schéma : sur la première carte, une activité est relative à l'entrée o , l'observation. L'autre activité reçoit une 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 résonnance 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.



FIGURE 1.15 – 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".

également le fait de s'intéresser aux cartes récurrentes. On souhaite en effet créer un modèle qui puisse unir cartes récurrentes et cartes normales au sein d'une même architecture. L'aspect bio-inspiré du modèle et son aspect multimodal ciblent plutôt des applications d'un tel réseau en robotique. Or, la plupart des données traitées par des réseaux de neurones, en particulier dans les applications robotiques, sont temporelles : vidéos, signaux sonores, capteurs de position. Il est donc important de s'appuyer autant sur les modèles de cartes récurrentes existantes que sur les architectures afin de créer un modèle général.

Les modèles de cartes récurrentes existant dans la littérature se classent en catégories similaires aux modèles multicartes que nous avons listé. D'une part, certains modèles de cartes utilisent l'état précédent de la carte lors du calcul de l'*activité* de l'état courant. De l'autre, des modèles réutilisent plutôt des éléments de la carte à l'instant précédent directement en tant qu'*entrée* de l'état courant.

Parmi les premiers travaux autour des cartes auto-organisatrices, l'Hypermap (Kohonen1991THEHA dérivée ensuite en *recurrent SOM* (varsta_temporal_2001) conditionnent la recherche de BMU d'une carte à un contexte dépendant de l'état précédent, lié à l'entrée précédente dans la séquence. Ce contexte repose sur la limitation d'une zone de la carte dans laquelle faire la recherche du BMU.

D'autres travaux reposent sur la transmission d'un contexte en tant qu'entrée complétant l'entrée courante. Ainsi, les *recursive SOMs* de (Voegtlin2002RecursiveSM) prennent deux entrées : l'élément de la séquence ainsi qu'un vecteur contenant l'ensemble des activations des neurones de la carte à l'état précédent. MSOM, de (Strickert2005MergeSF) 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. Le modèle SOMSD (hagenbuchner_self-organizing_2003 ; hammer_recursive_2004 ; hammer_self-organizing_2005) réduit ce contexte à la position de la Best Matching Unit. Les travaux de (Buonamente2013Simul) proposent une version récurrente du modèle A-SOM présenté en section précédente. Le contexte considéré est alors un ensemble d'activités de neurones. Les mécanismes de transmission de contexte entre instants dans les cartes récurrentes s'appuient sur les mêmes mécanismes que ceux proposés dans le cadre d'architectures de cartes : sélection de région de la carte, transmission d'activation, et enfin transmission du BMU. Les travaux menés en (fix20) sur le modèle SOMSD montrent qu'une carte récurrente parvient à différencier ses BMUs en fonction de la position de l'entrée dans une séquence et non seulement de la valeur de l'entrée.

1.3 Axe de recherche

Nous avons détaillé la littérature existante en terme d'architecture de SOMs et plus généralement de réseaux de neurones auto-organisés. Nous avons divisé ces architectures en deux grandes catégories : un format hiérarchique et *feedforward*, et un format non-hiérarchique incluant des rétroactions. Le format feed forward implique généralement un apprentissage couche par couche. Ce format est très appliqué et permet d'améliorer les capacités de clustering d'une SOM classique, principalement dans le cas où les données traitées présentent elles-mêmes une structure hiérarchique, telles que des images ou des phrases. Nous cherchons à développer une architecture plus générale de cartes auto-organisatrices et ne nous plaçons ainsi pas dans le contexte des Deep SOM mentionné ci-dessus. Cependant, nous notons que la position du BMU comme interface entre couche de cartes permet des capacités de calcul.

Nombre de ces architectures sont développées directement dans un but applicatif. On peut ainsi faire la distinction entre un modèle d'architecture, tel que HSOM, qui est générique et applicable à tout type de données, et une structure appliquée, développées spécifiquement pour un type de données.

Opposées à ces architectures hiérarchiques, des architectures reposent sur de l'interaction entre cartes, avec des boucles de rétroaction. Ces architectures sont moins présentes dans la littérature que les Deep SOM, et cherchent en général à se rapprocher d'un contexte biologique. Nous nous plaçons plutôt dans la lignée des cartes non-hiérarchiques, sans vouloir cependant copier un aspect biologique. De façon intéressante, nous remarquons que plusieurs structures non-hiérarchiques sont associées à l'apprentissage de données temporelles. Ces architectures se rapprochent des modèles appelés cartes auto-organisatrices récurrentes, dans lesquels des éléments de calcul d'une carte à une itération données sont réutilisés pour le calcul des itérations suivantes. Ces architectures non-hiérarchiques dynamiques se divisent en architectures centralisées et décentralisées. La décentralisation des calculs va dans un sens de l'informatique non conventionnelle.

Ces modèles soulèvent également une problématique dans les algorithmes d'apprentissage d'architecture non-hiérarchiques comportant des rétroactions. Dans le cas des neurones impulsifs, les impulsions des neurones arrivent en différé, une connexion réciproque entre neurones ne pose pas de problème : les neurones sont traités dans l'ordre des impulsions. Dans le cas de cartes de Kohonen, qui ne repose pas sur des séries d'impulsions, l'information arrive simultanément dans les différentes cartes. 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. Les architectures mentionnées font le choix d'apprendre d'abord les cartes sensorielles de façon indépendante, puis d'apprendre seulement leurs connexions dans un second temps, décomposant ainsi la boucle. La rétroaction est ensuite utilisée en phase applicative pour générer une modalité : seule une des cartes sensorielle prend une entrée, l'autre carte est considérée comme la sortie de l'architecture et réagit à l'activité des autres cartes. Les rétro-actions ne sont alors pas non plus un problème en phase d'application.

Notons enfin que limites entre les catégories d'architectures que nous avons différenciées dans ce chapitre sont floues. Une architecture décentralisée peut contenir des sous-structures hiérarchiques ou au contraire, une structure hiérarchique de modules décentralisés peut être imaginée.

Parmi ces axes de développement d'architectures, nous choisissons de nous intéresser dans cette thèse à des architectures non-hiérarchiques de cartes. Les architectures hiérarchiques nous paraissent de bonnes candidates à améliorer les performances d'une SOM sur une application spécifique comme du traitement d'images, tandis que les architectures non-hiérarchiques offrent de nouvelles possibilités de calcul, non envisageables par des SOM classiques, telles que la génération d'entrée, et c'est cet axe que nous souhaitons explorer. Peu de travaux ont par ailleurs exploré l'idée d'associer des SOMs en architectures comportant des rétroactions. Les choix pour la construction d'une telle architecture se situent au niveau de l'interface entre cartes. De nombreux travaux autour des cartes de Kohonen utilisent la position du BMU comme vecteur de l'information transmise entre plusieurs cartes. Nous faisons ce choix également. Il apparaît comme une valeur peu coûteuse à communiquer entre cartes, mais qui contient beaucoup d'information sur l'état d'une carte. Cette valeur se présente également comme un cadre homogène de communication intercarte : quelles que soient les en-

trées sur lesquelles apprend une carte, il sera toujours possible de la connecter à d'autres cartes de l'architecture. Enfin, on retrouve l'utilisation de la position du BMU à la fois dans des architectures multicartes et dans les cartes récurrentes comme SOMSD. Le cadre choisi permettrait donc d'intégrer à la fois des cartes classiques et des cartes récurrentes au sein d'une même architecture, offrant encore plus de possibilités de calcul. Ce modèle de connexions par transmission de position de BMU n'a pas été exploré pour créer des architectures non-hiérarchiques décentralisées. Cette thèse se veut l'exploration de ce modèle. Nos travaux font ainsi suite à (**baheux_towards_2014**) sur des architectures récurrentes multimodales utilisant la transmission de la position du BMU entre des cartes de Kohonen, exploitant la position du BMU comme les travaux sur les cartes récurrentes SOMSD (**hagenbuchner_self-organizing_2003** ; **Strickert2003UnsupervisedRS** ; **fix20**). Les travaux commencés en (**baheux_towards_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 (**fix20**). Nous voulons continuer sur le même modèle, utilisant la transmission du BMU, qui nous apparaît comme une solution compacte et facilement extensible à grande échelle pour construire des grandes architectures, mais en explorant cette fois l'aspect des connexions intercartes. Par leur motivation, qui est le développement d'un système multi-som, nos travaux se rapprochent aussi des travaux conduits sur l'architecture A-SOM (**johnsson_associating_2008** ; **johnsson_associative_2009** ; **gil_sarasom_2015** ; **Buonamente**). Notre modèle de carte et d'interface est par contre différent.

