

Rien sur les échages d'activation?

Chapitre 1

Architectures de cartes auto-organisatrices

Sommaire

1.1	Les cartes auto-organisatrices de Kohonen comme modules d'une architecture	ii
1.1.1	Carte de Kohonen classique	ii
1.1.2	Aspect topologique de la carte de Kohonen	iii
1.1.3	Inspiration biologique	iv
1.2	Architectures de cartes auto-organisatrices	vii
1.2.1	Architectures hiérarchiques de cartes	viii
1.2.2	Structures de cartes auto-organisatrices non-hiéarchiques	xi
1.2.3	Cartes auto-organisatrices récurrentes	xvii
1.3	Axe de recherche	xx

comme *baf*

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. (Kohonen 1995)

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 ??. Nous 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 proposerons des hypothèses sur les comportements que nous pouvons attendre de notre modèle d'architecture. Ces hypothèses motivent les expériences conduites dans la suite de cette thèse.

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 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 grappe 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 choisi 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 1.2. 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 \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. \rightarrow exemple ?

Le processus de mise à jour des poids d'une carte de Kohonen se traduit visuellement par un déploiement de la carte dans l'espace d'entrée. On parlera donc de *dépliement* d'une carte lorsque qu'on parle d'apprentissage. Ce déploiement est représenté en figures 1.3 et 1.4 pour des exemples de cartes en une et deux dimensions, se dépliant sur des données en deux dimensions. On observe d'ailleurs que les valeurs des poids à l'issue de l'apprentissage correspondent aux centres des cellules de Voronoï de l'espace d'entrée. 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 1.1 présente par exemple le déploiement d'une carte sur des imagettes MNIST. Par son aspect ordonné, une carte est une représentation en faible dimension d'un espace d'entrée de grande dimension.

+ représentation discrète d'un espace continu?

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 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



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.

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 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 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 et E. Oja 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).

1.1.3 Inspiration biologique

Le développement des cartes auto-organisatrices par Kohonen est initiallement 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

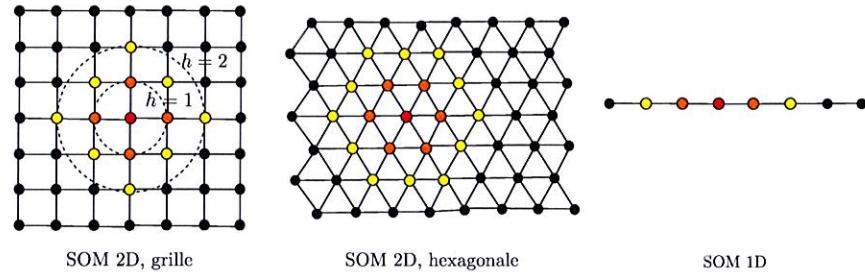


FIGURE 1.2 – Exemples de connexions dans le graphe support d'une SOM. Deux nœuds connectés sont ici à une distance de une unité dans la carte. Les SOM en deux dimensions sont les plus communément utilisées dans la littérature, sous forme d'une grille ou d'une grille hexagonale. Les SOM une dimension sont également utilisées.

Y

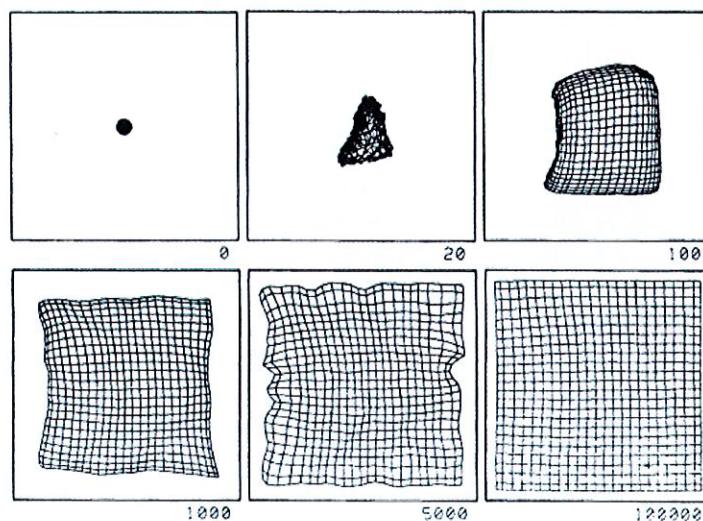


FIGURE 1.3 – Dépliement d'une SOM 2D sur des données dans le plan $[0, 1]^2$, tiré de (Kohonen 1995)

dvp_p!

↓ uniformes

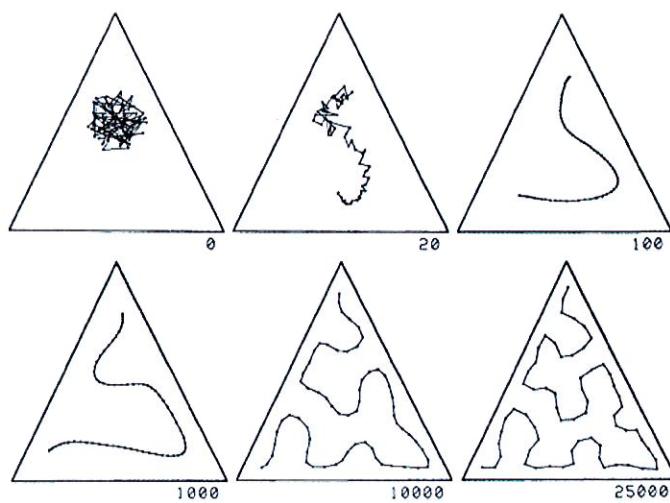


FIGURE 1.4 – Dépliement d'une SOM 1D sur des données dans un triangle 2D, tiré de (Kohonen 1995)

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 1.5. L'aire associée à l'audition présente aussi une organisation topographique (Reale et Imig 1980), ainsi que de nombreuses autres aires, directement sensorielles ou plus abstraites (Kohonen 1995). Une carte de Kohonen ne doit cependant pas être considérée comme une modélisation biologiquement plausible d'une aire du cortex cérébral, mais plutôt comme une adaptation au niveau computationnel d'un concept biologique, ici le concept d'organisation topologiquement correcte dans les cortex sensoriels, tel que le cortex visuel ou auditif.

Vue ?

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. Les modèles de base les plus communs pour cette communication entre neurones sont la zone de convergence-divergence de Damasio (CDZ) (Damasio 1989), et le modèle de boucles de réentrées de Edelman (Edelman 1982). 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 assemblent les signaux en provenance

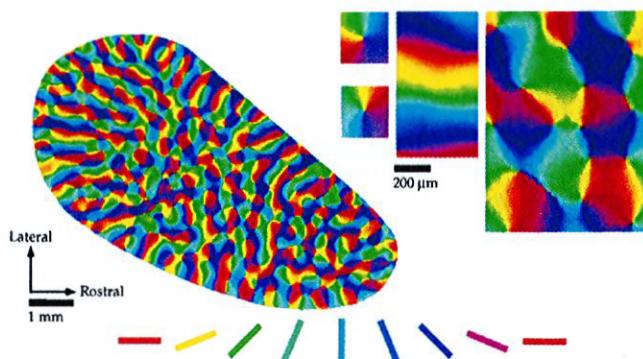


FIGURE 1.5 – Représentation des réponses du cortex visuel V1 à un stimulus visuel (bâtonnets d'orientations spatiales différentes). Les neurones répondant à une certaine orientation sont affichés de la même couleur. On observe une continuité entre les neurones proches dans le cortex et l'orientation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

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. Cette section passe en revue certains modèles. 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. À la lecture des modèles existants, nous avons pu différencier deux grandes classes d'architectures de cartes : les architectures *hiérarchiques feed-forward* et les architectures *non-hiérarchiques*, qui peuvent elles-mêmes être *centralisées* ou *décentralisées*. Nous détaillerons dans cette section ce qu'on appelle architecture hiérarchique et non-hiérarchique et analyserons les comportements d'apprentissage émergeant des différentes structures. 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.

→ est-ce que ? (% centralisé)

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 les qualifierons ainsi d'architecture *feed-forward*.

communication à sens unique?

Architectures hiérarchiques sélectives

Certains travaux s'appuient sur des architectures hiérarchiques qualifiées de "sélective", dont le principe général est illustré en figure 1.6. Toutes les architectures présentées peuvent être séparées en niveaux, dont un niveau d'entrée, un niveau de sortie et des niveaux intermédiaires. Le premier niveau d'une telle architecture est une carte classique, prenant des données en entrée. Après apprentissage du premier niveau, le second niveau est composé de plusieurs cartes. L'activation ou le BMU de la carte du premier niveau à une entrée permet de sélectionner une carte du deuxième, à laquelle sera présentée cette même entrée. Chacune des cartes du deuxième niveau est alors entraînée sur un sous-ensemble des données d'entrée. Pour chaque entrée, la carte à mettre à jour est sélectionnée en fonction de la réponse du premier niveau à l'entrée, d'où qualification de sélective pour le mode de communications entre cartes ; le premier niveau n'est plus mis à jour lors de cette phase. Le processus est répété ainsi de suite si besoin sur d'autres niveaux. Les niveaux supérieurs ont alors plus de cartes que les niveaux inférieurs. Ce procédé est retrouvé dans (Barbalho et al. 2001 ; Suganthan 2001) (Miikkulainen 1992) pour de la classification de phrases, (Dittenbach et al. 2000 ; Ordonez et al. 2010), ou encore en (Zhao et al. 2015) pour de la détection d'arrière-plan.

La façon de sélectionner une carte d'un niveau différent ensuite en fonction des travaux. (Barbalho et al. 2001) utilisent la position du BMU calculée au premier niveau pour choisir la carte du second niveau à utiliser. L'architecture utilisée en (Zhao et al. 2015) utilise plusieurs couches de cartes pour du traitement de séquence. Cette fois, le processus de sélection s'opère en choisissant ou non de transférer l'entrée courante au niveau suivant, en s'appuyant sur la distance moyenne de la réponse des cartes à l'entrée.

Dans ce type d'architecture, c'est un processus extérieur aux cartes qui permet de sélectionner la carte du niveau suivant. Les connexions permettent à chacune des cartes de se spécialiser sur une partie de l'espace d'entrée. Face à ces structures sélectives, on peut imaginer une structure présentant une interface entre cartes qui soit une transmission d'une représentation des données générée par une carte.

avantage/particularité d'un point de vue fondamental? computation?

Architectures hiérarchiques par transmission de représentation intermédiaire

Contrairement aux architectures par sélection, la deuxième couche de carte 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

?

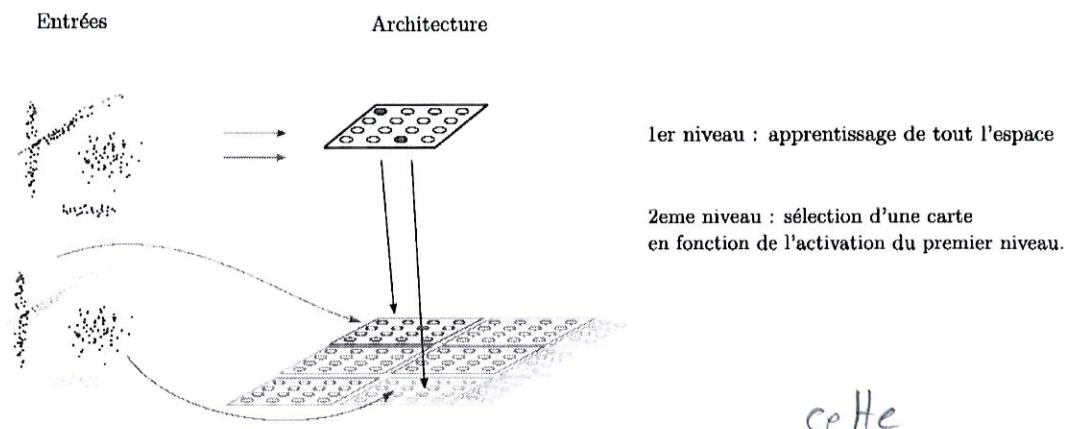


FIGURE 1.6 – Exemple d’architecture hiérarchique sélective. La carte du premier niveau est entraînée sur tout l’espace d’entrée. Après apprentissage, la carte permet de filtrer les entrées pour les renvoyer vers une carte du niveau suivant. Dans cet exemple, la position du BMU de la carte du niveau 1 permet de sélectionner une carte du niveau 2, comme c’est le cas en (Barbalho 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.

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 et Erkki Oja 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 de la première carte ; cette architecture est illustrée en figure 1.7. 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 BMU 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 un peu meilleure qu’une SOM classique. La présence de la carte intermédiaire change un peu 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 d’extraire une représentation un peu différente que celle extraite par une SOM classique. 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. D’autres travaux par la suite implémentent des modèles similaires transmettant la position du BMU entre cartes, sur des architectures comportant plus de cartes que HSOM, tel que (Hagenauer et Helbich 2013). Comme dans les modèles sélectifs, la représentation intermédiaire considérée peut également être les poids des BMUs du premier niveau, comme c’est le cas en (Wang et al. 2007 ; Gunes Kayacik et al. 2007).

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 assez similaires aux travaux mentionnés précédemment. Mais cette fois, ces travaux s’inspirent des réseaux de neurones profonds

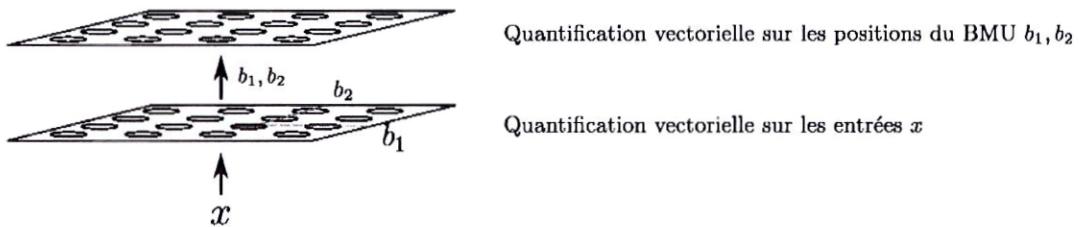


FIGURE 1.7 Architecture HSOM (Lampinen et Erkki Oja 1992). L'apprentissage des positions du BMU de la première couche par la seconde permet de mieux détecter les ensembles de données, dans une tâche de clustering. La deuxième couche est vue comme un niveau plus abstrait que la première.

(*Deep Learning*), ayant connu leur essor cette même année avec les réseaux convolutionnels permettant l'apprentissage supervisé d'images de façon inégalée (LeCun et al. 2015). Par analogie avec les réseaux convolutionnels, les Deep SOM s'intéressent souvent à l'apprentissage d'images par des SOMs. Ainsi (N. Liu et al. 2015; Hankins et al. 2018; Wickramasinghe et al. 2019; Aly et Almotairi 2020; Sakkari et Zaiied 2020; Dozono et al. 2016; Nawaratne et al. 2020; Mici et al. 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 (N. Liu et al. 2015) est illustré en figure 1.8 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 imagette 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 feed-forward : les couches sont entraînées les unes après les autres. Par conséquent, elles ne se prêtent pas à un apprentissage en ligne, tout comme une carte de Kohonen classique. 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 particuliè-

In nén
caus en pas
avav!

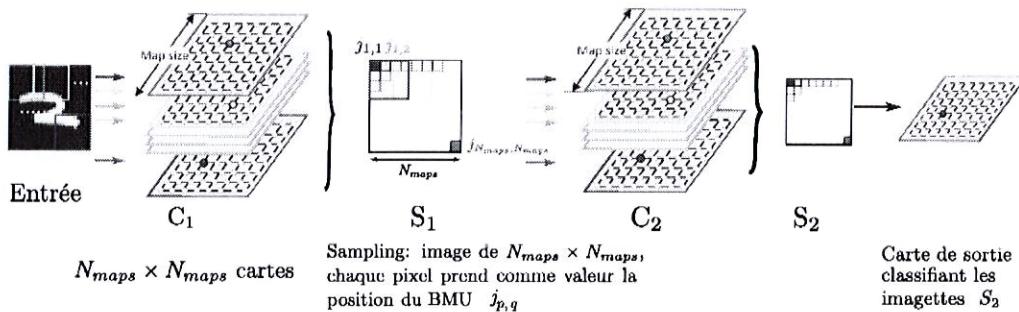


FIGURE 1.8 Architecture DSOM de SOM "convoluttionnelle" (Nan Liu et al. 2015). Les auteurs utilisent les positions des BMUs de la couche de carte $j_{p,q}$ comme valeurs d'entrée pour les couches suivantes, dans la lignée de H SOM (Lampinen et Erkki Oja 1992). Les couches sont entraînées les unes après les autres. Cette architecture est *feedforward* et *hiérarchique* par transmission de représentation intermédiaire.

ment 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 feed-forward est le même qu'une SOMs 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.

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 initialement 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-hiéronymiques. 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-hiéronymiques

Les architectures non-hiéronymiques 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 (Edelman 1982) et des zones de convergence-divergence (CDZ) (Damasio 1989). L'assemblage de réseaux de neurones en architecture fait l'objet de 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-hiéronymiques 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-hiéronymique. 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-hiéronymiques 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-hiéronymiques 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-hiéronymiques 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 (Lallee et Dominey 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. 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 elles. 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 entraînées, puis les cartes associatives sont apprises dans un second temps. Nous l'avons quand même classée comme non-hiéronymique, 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 (Ménard et Frezza-Buet 2005 ; Khouzam et H.Frezza-Buet 2013) associe des cartes modales par une carte associative par des connexions neurone à neurones, en renforçant

à ?

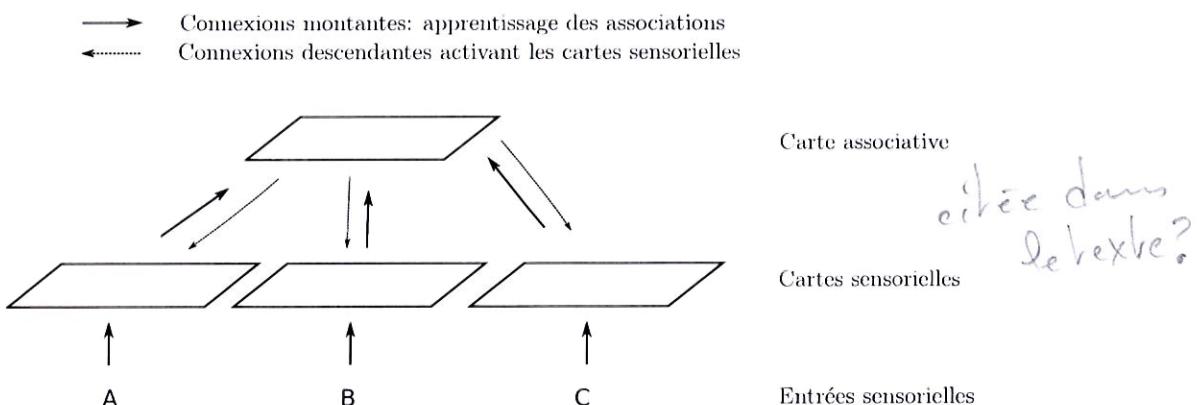


FIGURE 1.9 – Exemple général d’architecture décentralisée comportant une carte associative. Ces architectures sont utilisées dans des tâches de traitement de données multimodales. Des cartes appelées cartes *sensorielles* ou *modales* prennent des entrées dans plusieurs modalités. Une carte *associative* 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.

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-hiéarchique pour leur possibilité d’activation d’une carte par l’autre. Encore une fois, la position du BMU apparaît chez (Lallee et Dominey 2013) 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.

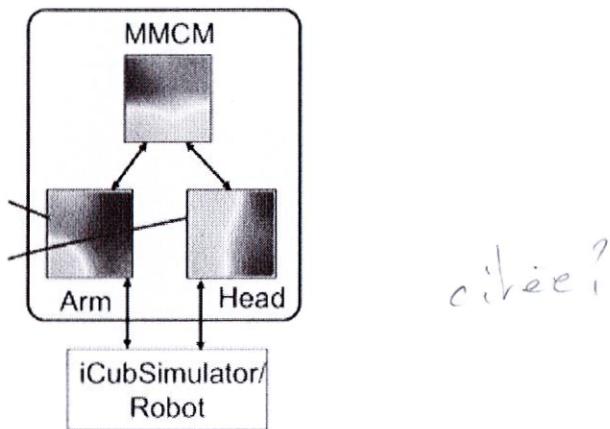


FIGURE 1.10 – Architecture MMCM. Les cartes du premier niveau reçoivent l'une les mouvements de tête d'un robot, l'autre les mouvements de main. Une carte amodale (MMCM) reçoit les positions des BMUs de chaque carte du premier niveau. Chaque carte du premier niveau possède également une couche "hiérarchique" 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 (Lallee et Dominey 2013)

Architectures non-hiéronymiques 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 tracé~~e~~ en figure 1.11. 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 et al. 2020). Les auteurs utilisent ici des cartes de Kohonen impulsionales, 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~~s~~ 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 (Lallee et Dominey 2013).

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-hiéronymiques est développée en (Magnus

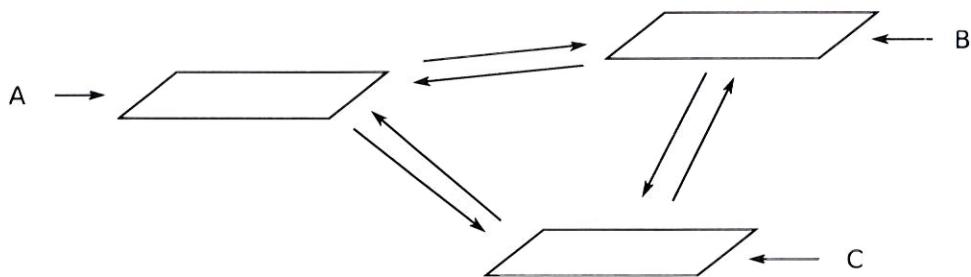


FIGURE 1.11 – Exemple général d’architecture décentralisée de cartes. Chacune des cartes de l’architecture prend une entrée sensorielle A,B,C. Des connexions entre cartes permettent l’apprentissage d’associations entre modalités. Chacune des cartes peut donc être vue comme une carte multimodale. La façon 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 et al. 2020) et conjointement en (M. Johnsson et al. 2009) et (Baheux et al. 2014).

Johnsson et Balkenius 2008 ; M. Johnsson et al. 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

decentralisé inclut la possibilité de créer une version d’architecture centralisée à partir des règles d’associations, par exemple (Buonamente et al. 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 et Gepperth 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 (Parisi et al. 2018), les auteurs développent une architecture de deux réseaux auto-organisés appelés *grow when required networks* (GWR). Ces

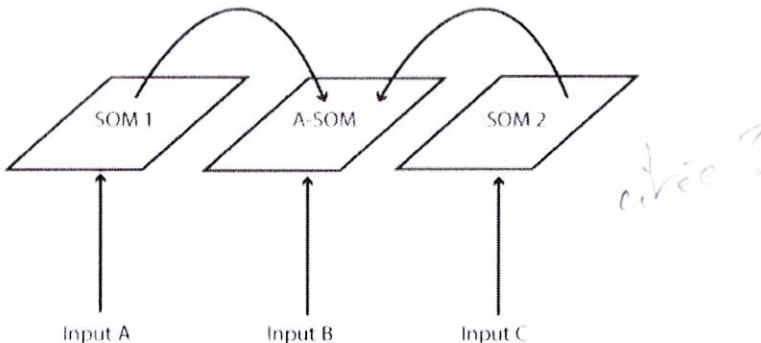


FIGURE 1.12 – Le modèle A-SOM (M. Johnsson et al. 2009) associe les activités de différentes cartes. Chacune des cartes prend une modalité A,B,C en entrée. Contrairement aux modèles précédemment cités, les trois cartes apprennent simultanément. L'association est prise en compte lors du calcul des activités de chaque carte.

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 précédent. 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 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 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.) un peu vide, non ?

? deux app juis de objets à batch
peut le faire en continu?

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 nom 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 (Buonamente et al. 2015). 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 et al. 2014) ont également cherché à insérer des connexions temporelles au sein d'une architecture de deux cartes, présentée en 1.14. 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 existante dans le traitement de séquences, où le choix d'un BMU doit prendre en compte le contexte des états précédents, aux modèles multimodals 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'ap-

la suite ?

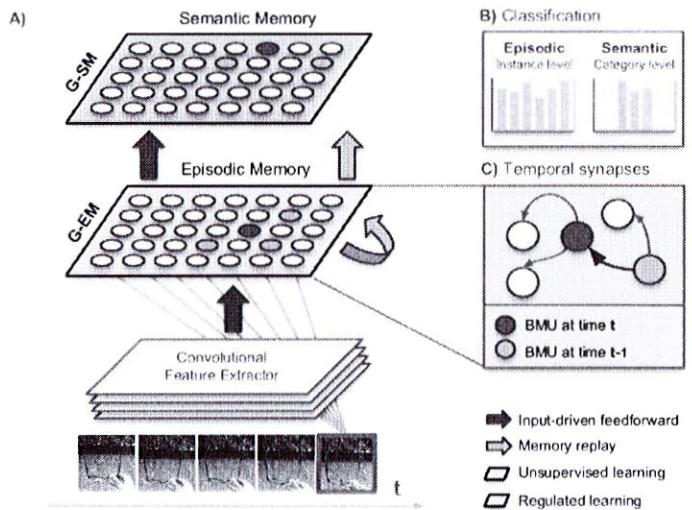


FIGURE 1.13 – Architecture à *double mémoire* proposée en (Parisi et al. 2018). La couche de mémoire épisodique est un réseau *grow when required* (*GWR*), un réseau auto-organisé similaire à une carte de Kohonen, à la différence que les neurones sont ajoutés au fur et à mesure de l'apprentissage. La 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.

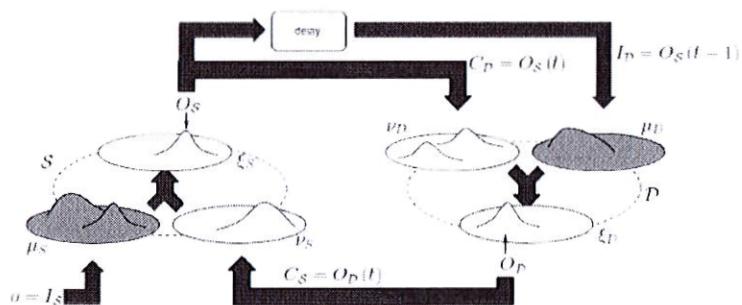


FIGURE 1.14 – Structures de deux cartes auto-organisatrices communicantes, (Baheux et al. 2014). Chaque carte est composée de trois couches d'activités, représentées séparément sur le schéma : sur la première carte, une activité est relative à l'entrée o , l'observation. L'autre activité reçoit une entrée 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.

résonance ?

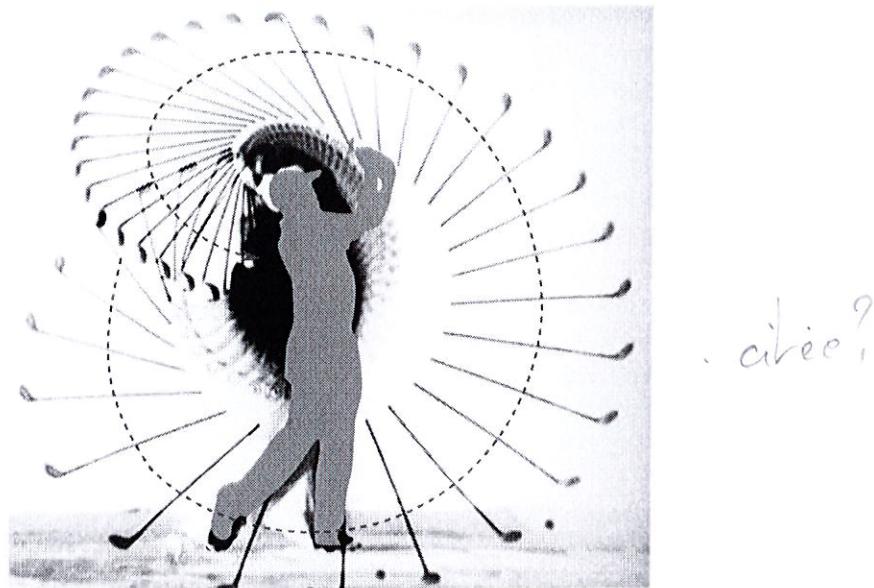


FIGURE 1.15 – L'image présentée à un réseau (en bleu) correspond à un instant d'une séquence. L'objectif de l'apprentissage non-supervisé de séquences est d'extraire une représentation d'une séquence d'entrée. Une utilisation commune est la classification de mouvements. La séquence "tirer" sera différente de la séquence "marcher".

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 existants nous permettront de compléter cette bibliographie.

- ✓ Notre volonté de créer un modèle général d'architecture de cartes auto-organisatrice motive 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 existants dans la littérature se classent en catégories similaires aux modèles multicartes que nous avons listés. 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 (Kohonen 1991), dérivée ensuite en *recurrent SOM* (Varsta et al. 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 (Voegtlin 2002) 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 (Strickert et Hammer 2005) s'appuie sur le poids du BMU. ~~A~~ chaque instant, l'*entrée* de contexte à transmettre à l'état suivant est définie comme une combinaison linéaire entre le poids du BMU courant et le contexte courant. Le modèle SOMSD (Hagenbuchner et al. 2003 ; Hammer, Micheli, Sperduti et al. 2004 ; Hammer, Micheli, Neubauer et al. 2005 ; Fix et Frezza-Buet 2020) réduit ce contexte à la position de la Best Matching Unit. Les travaux de (Buonamente et al. 2013) 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 (Fix et Frezza-Buet 2020) 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*. X

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-hiéarchical 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-hiéarchicals, sans vouloir

*→ apprendre ?
→ peu explicatif
au-delà*

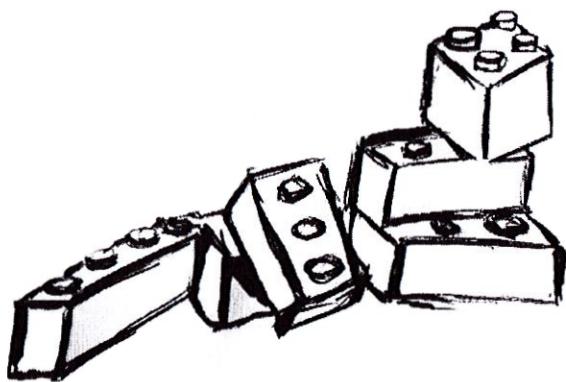
cependant copier un aspect biologique. De façon intéressante, nous remarquons que plusieurs structures non-hiérarchiques sont associées à l'apprentissage de données temporelles. Ces architectures se rapprochent des modèles appelés cartes auto-organisatrices récurrentes, dans lesquels des éléments de 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 sensorielles 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 entrées 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 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 comme les travaux sur les cartes récurrentes SOMSD (Hagenbuchner et al. 2003 ; Strickert, Hammer et Blohm 2003 ; Fix et Frezza-Buet 2020). 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 Frezza-Buet 2020). 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 (Magnus Johnsson et Balkenius 2008 ; M. Johnsson et al. 2009 ; Gil et al. 2015 ; Buonamente et al. 2015) ; notre modèle de carte et d'interface est par contre différent.



Bibliographie

- Alahakoon, D., S.K. Halgamuge et B. Srinivasan. "Dynamic self-organizing maps with controlled growth for knowledge discovery". In : *IEEE Transactions on Neural Networks* 3 (2000).
- Aly, Saleh et Sultan Almotairi. "Deep Convolutional Self-Organizing Map Network for Robust Handwritten Digit Recognition". In : *IEEE Access* 8 (2020), p. 107035-107045.
- Baheux, D., J. Fix et H. Frezza-Buet. "Towards an effective multi-map self organizing recurrent neural network". In : *Proc. ESANN*. 2014.
- Barbalho, J.M. et al. "Hierarchical SOM applied to image compression". In : *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*. T. 1. ISSN : 1098-7576. 2001, 442-447 vol.1.
- Buonomente, Miriam, Haris Dindo et Magnus Johnsson. "Discriminating and simulating actions with the associative self-organising map". In : *Connection Science* 27 (2015), p. 118-136.
- "Hierarchies of Self-Organizing Maps for action recognition". In : *Cognitive Systems Research* 39 (2016), p. 33-41.
- "Simulating Actions with the Associative Self-Organizing Map". In : *AIC@AI*IA*. 2013.
- Cottrell, Marie et al. "Theoretical and Applied Aspects of the Self-Organizing Maps". In : *Advances in Self-Organizing Maps and Learning Vector Quantization*. T. 428. Cham : Springer International Publishing, 2016, p. 3-26.
- Damasio, Antonio R. "Time-locked multiregional retroactivation : A systems-level proposal for the neural substrates of recall and recognition". In : *Cognition* 33.1 (1989).
- Dittenbach, M., D. Merkl et A. Rauber. "The growing hierarchical self-organizing map". In : *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing : New Challenges and Perspectives for the New Millennium*. Como, Italy : IEEE, 2000, 15-19 vol.6.
- Dozono, Hiroshi, Gen Niina et Satoru Araki. "Convolutional Self Organizing Map". In : *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2016, p. 767-771.
- Edelman, Gerald M. "Group selection and phasic reentrant signaling a theory of higher brain function". In : *The 4th Intensive Study Program Of The Neurosciences Research Program*. 1982.
- Escobar-Juárez, Esau et al. "A Self-Organized Internal Models Architecture for Coding Sensory-Motor Schemes". en. In : *Frontiers in Robotics and AI* (2016).
- Fix, Jérémie et Hervé Frezza-Buet. "Look and Feel What and How Recurrent Self-Organizing Maps Learn". In : *Proc. WSOM'19*. T. 976. 2020, p. 3-12.
- Fort, J.C. "SOM's mathematics". en. In : *Neural Networks* 19.6-7 (2006).
- Gil, David et al. "SARASOM : a supervised architecture based on the recurrent associative SOM". en. In : *Neural Computing and Applications* 26.5 (2015).

- Gunes Kayacik, H., A. Nur Zincir-Heywood et Malcolm I. Heywood. "A hierarchical SOM-based intrusion detection system". In : *Engineering Applications of Artificial Intelligence* 20.4 (2007), p. 439-451.
- Hagenauer, Julian et Marco Helbich. "Hierarchical self-organizing maps for clustering spatiotemporal data". In : *International Journal of Geographical Information Science* 27.10 (2013), p. 2026-2042.
- Hagenbuchner, M., A. Sperduti et Ah Chung Tsoi. "A self-organizing map for adaptive processing of structured data". In : *IEEE Transactions on Neural Networks* 14.3 (2003), p. 491-505.
- Hammer, Barbara, Alessio Micheli, Nicolas Neubauer et al. "Self-Organizing Maps for Time Series". In : *Proc. WSOM'2005*. 2005, p. 8.
- Hammer, Barbara, Alessio Micheli, Alessandro Sperduti et al. "Recursive self-organizing network models". In : *Neural Networks* 17.8-9 (2004), p. 1061-1085.
- Hankins, Richard, Yao Peng et Hujun Yin. "SOMNet : Unsupervised Feature Learning Networks for Image Classification". In : *2018 International Joint Conference on Neural Networks (IJCNN)*. Rio de Janeiro : IEEE, 2018, p. 1-8.
- Johnsson, M., C. Balkenius et G. Hesslow. "Associative Self-Organizing Map". In : *Proc. IJCCI*. 2009.
- Johnsson, Magnus et Christian Balkenius. "Associating SOM Representations of Haptic Sub-modalities". In : *Computer Science* (2008).
- Khacef, Lyes, Laurent Rodriguez et Benoit Miramond. "Brain-inspired self-organization with cellular neuromorphic computing for multimodal unsupervised learning". In : *ArXiv :2004.05488 /cs, Q-bio/* (2020). arXiv : 2004.05488.
- Khouzam, B. et H. Frezza-Buet. "Distributed recurrent self-organization for tracking the state of non-stationary partially observable dynamical systems". In : *Biologically Inspired Cognitive Architectures* (2013).
- Kohonen, Teuvo. "Self-organized formation of topologically correct feature maps". In : *Biological Cybernetics* 43.1 (1982).
- "Self-Organizing Maps". In : *Springer Series in Information Sciences*. 1995.
- "THE HYPERMAP ARCHITECTURE". In : 1991.
- Koikkalainen, P. et E. Oja. "Self-organizing hierarchical feature maps". In : *1990 IJCNN International Joint Conference on Neural Networks*. 1990.
- Lallee, Stephane et Peter Dominey. "Multi-modal convergence maps : From body schema and self-representation to mental imagery". In : *Adaptive Behavior* 21.4 (2013), p. 274-285.
- Lampinen, Jouko et Erkki Oja. "Clustering Properties of Hierarchical Self-Organizing Maps". In : *Journal of Mathematical Imaging and Vision* (1992), p. 15.
- LeCun, Yann, Yoshua Bengio et Geoffrey Hinton. "Deep learning". In : *Nature* 521.7553 (2015), p. 436-444.
- Lefort, M. et A. Gepperth. "Active learning of local predictable representations with artificial curiosity". In : *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. 2015.
- Liu, N., Jinjun Wang et Yihong Gong. "Deep Self-Organizing Map for visual classification". In : *2015 International Joint Conference on Neural Networks (IJCNN)* (2015).
- Liu, Nan, Jinjun Wang et Yihong Gong. "Deep Self-Organizing Map for visual classification". In : *2015 International Joint Conference on Neural Networks (IJCNN)*. ISSN : 2161-4407. 2015, p. 1-6.
- Ménard, Olivier et Hervé Frezza-Buet. "Model of multi-modal cortical processing : Coherent learning in self-organizing modules". In : *Neural Networks* 18.5-6 (2005), p. 646-655.

- Mici, Luiza, German I. Parisi et Stefan Wermter. "A self-organizing neural network architecture for learning human-object interactions". In : *Neurocomputing* 307 (2018), p. 14-24.
- Miikkulainen, Risto. "Script Recognition with Hierarchical Feature Maps". In : *Connection Science* 2 (1992), p. 196-214.
- Nawaratne, Rashmika et al. "Hierarchical Two-Stream Growing Self-Organizing Maps With Transience for Human Activity Recognition". In : *IEEE Transactions on Industrial Informatics* 16.12 (2020). Conference Name : IEEE Transactions on Industrial Informatics, p. 7756-7764.
- Ordonez, Diego et al. "Hierarchical Clustering Analysis with SOM Networks". In : *International Journal of Computer and Information Engineering* 4.9 (2010), p. 1393-1399.
- Parisi, German I. et al. "Lifelong Learning of Spatiotemporal Representations With Dual Memory Recurrent Self-Organization". In : *Frontiers in Neurorobotics* (2018).
- Reale, Richard A. et Thomas J. Imig. "Tonotopic organization in auditory cortex of the cat". In : *Journal of Comparative Neurology* 192 (1980).
- Sakkari, Mohamed et Mourad Zaied. "A Convolutional Deep Self-Organizing Map Feature extraction for machine learning". In : *Multimedia Tools and Applications* 79.27-28 (2020), p. 19451-19470.
- Strickert, Marc et Barbara Hammer. "Merge SOM for temporal data". In : *Neurocomputing* 64 (2005), p. 39-71.
- Strickert, Marc, Barbara Hammer et Sebastian Blohm. "Unsupervised Recursive Sequence Processing". In : *ESANN*. 2003.
- Suganthan, P N. "Pattern classification using multiple hierarchical overlapped self-organising maps". In : *Pattern Recognition* (2001), p. 7.
- Varsta, Markus, Jukka Heikkonen et Jouko Lampinen. "Temporal Kohonen Map and the Recurrent Self-Organizing Map : Analytical and Experimental Comparison". In : *Neural Processing Letters* (2001), p. 15.
- Voegtlin, Thomas. "Recursive self-organizing maps". In : *Neural Networks : the Official Journal of the International Neural Network Society* 15 8-9 (2002), p. 979-991.
- Wang, Liang, Eliathamby Ambikairajah et Eric H.C. Choi. "A comparisonal study of the multi-layer Kohonen self-organizing feature maps for spoken language identification". In : *2007 IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*. 2007, p. 402-407.
- Wickramasinghe, Chathurika S., Kasun Amarasinghe et Milos Manic. "Deep Self-Organizing Maps for Unsupervised Image Classification". In : *IEEE Transactions on Industrial Informatics* 15.11 (2019), p. 5837-5845.
- Yamaguchi, Takashi, Takumi Ichimura et Kenneth James Mackin. "Adaptive Learning Algorithm in Tree-Structured Self-Organizing Feature Map". In : *SCIS&ISIS, Okayama*. 2010, p. 6.
- Zhao, Zhenjie, Xuebo Zhang et Yongchun Fang. "Stacked Multilayer Self-Organizing Map for Background Modeling". In : *IEEE Transactions on Image Processing* 24.9 (2015). Conference Name : IEEE Transactions on Image Processing, p. 2841-2850.

