

Sommaire

1 Approche modulaire des réseaux de neurones	1
1.1 Modularité	2
1.2 Mémoire associative et multi-modalité	2
1.3 Cartes auto-organisatrices	2
1.3.1 Carte de Kohonen classique	2
1.3.2 Aspect topologique de la carte de Kohonen	3
1.3.3 Inspiration Biologique d'une carte de Kohonen	4
1.4 Motivations de la thèse : pourquoi construire une architecture de cartes auto-organisatrices modulaires	6
1.4.1 Inspiration biologique	6
1.4.2 Systèmes autonomes de cartes auto-organisatrices	7
1.4.3 Mémoire associative et SOM	7
1.4.4 Notion de mémoires	8
1.4.5 Association de réseaux de neurones	8
1.4.6 Calcul décentralisé ?	8
1.5 Architectures de cartes auto-organisatrices	8
1.5.1 Modèles de SOM impulsionnelles	9
1.5.2 The hierarchical self organizing map	9
1.5.3 Modèle CDZ de Dominey	10
1.5.4 A-SOM	11
1.6 Cartes auto-organisatrices récurrentes	11
1.6.1 Cartes récurrentes	11

1.6.2	The Hypermap architecture, TKM, RSOM	11
1.6.3	MSOM	11
2	Modèle d'architecture CxSOM	13
2.1	Carte de Kohonen classique	14
2.1.1	Algorithme et notations	14
2.1.2	Paramétrage d'une carte de Kohonen	15
2.2	Motivations du modèle CxSOM	18
2.2.1	Champ d'application : mémoire associative	18
2.2.2	Description de l'architecture	18
2.3	Présentation de CxSOM : exemple d'une architecture de deux cartes	20
2.3.1	Détail des étapes	20
2.3.2	Résumé	25
2.4	Formalisation : cas pour n cartes	25
2.4.1	Entrées et calcul d'activité	25
2.4.2	Calcul du BMU par relaxation	26
2.4.3	Mise à jour des poids	27
2.5	Choix des paramètres	28
2.5.1	Paramétrage d'une carte	28
2.5.2	Paramètres de l'architecture	29
2.6	Conclusion	29
3	Méthodes de représentation et d'analyse de l'architecture CxSOM	31
3.1	Introduction	31
3.1.1	Présentation de l'expérience d'illustration	32
3.1.2	Représentations et indicateurs classique des cartes de Kohonen	33
3.1.3	Limites de la représentation classique pour CxSOM	33
3.2	Formalisation par variables aléatoires	35
3.2.1	Représentation des entrées	35
3.2.2	Démarche expérimentale	35

3.2.3	Résumé	36
3.3	Représentation graphiques des cartes	36
3.3.1	Erreur de quantification d'une modalité dans une carte	37
3.3.2	Cartographie des entrées	37
3.3.3	Réduction de dimension des entrées	38
3.3.4	Résumé	41
3.4	Un indicateur de l'apprentissage du modèle par l'architecture basé sur l'information mutuelle	41
3.4.1	Information mutuelle et entropie	43
3.4.2	Indicateur	44
3.4.3	Estimation	44
3.4.4	Information mutuelle sur deux cartes	45
3.4.5	Discussion	47
3.5	Conclusion	51
4	Analyse de l'organisation de CxSOM	51
4.1	Analyse de la relaxation	51
4.1.1	Etude de la convergence des BMUs lors de la relaxation	52
4.1.2	Formalisation de l'algorithme de relaxation	53
4.1.3	Etude de l'influence de l'entrée contextuelle sur le BMU	55
4.1.4	Etude de l'unicité du point fixe	56
4.1.5	Influence du pas de convergence	57
4.1.6	Discussion	57
4.1.7	Conclusion	61
4.2	Organisation de structures de deux et trois cartes	61
5	Application : prédiction d'entrée manquante	63
5.1	Prédiction d'entrées géométriques	63
5.1.1	Méthodes	63
5.1.2	Résultats	64

5.1.3	Discussion	64
5.2	Application à la commande de drône en vol	65
5.2.1	Méthode expérimentale	65
5.2.2	Résultats	65
5.2.3	Discussion	65
5.3	Conclusion	65
	Bibliographie	69

Chapitre 3

Méthodes de représentation et d'analyse de l'architecture CxSOM

Sommaire

3.1	Introduction	31
3.1.1	Présentation de l'expérience d'illustration	32
3.1.2	Représentations et indicateurs classique des cartes de Kohonen	33
3.1.3	Limites de la représentation classique pour CxSOM	33
3.2	Formalisation par variables aléatoires	35
3.2.1	Représentation des entrées	35
3.2.2	Démarche expérimentale	35
3.2.3	Résumé	36
3.3	Représentation graphiques des cartes	36
3.3.1	Erreur de quantification d'une modalité dans une carte	37
3.3.2	Cartographie des entrées	37
3.3.3	Réduction de dimension des entrées	38
3.3.4	Résumé	41
3.4	Un indicateur de l'apprentissage du modèle par l'architecture basé sur l'information mutuelle	41
3.4.1	Information mutuelle et entropie	43
3.4.2	Indicateur	44
3.4.3	Estimation	44
3.4.4	Information mutuelle sur deux cartes	45
3.4.5	Discussion	47
3.5	Conclusion	51

3.1 Introduction

Dans le chapitre précédent, nous avons proposé l'algorithme CxSOM, permettant de construire des architectures non-hiéronymiques de cartes auto-organisatrices.

Dans ces architectures non-hiéronymiques, plusieurs cartes sont connectées et effectuent chacune une tâche d'apprentissage sur leur espace d'entrée externe, tout en prenant comme entrée

secondaire les positions des *Best Matching Unit* d'autres cartes. La particularité du modèle CxSOM est d'introduire des rétroactions entre cartes : l'architecture n'est pas un empilement de cartes qui apprennent tour à tour, de façon *feedforward*. Le but de ce modèle est de pouvoir construire des architectures assemblant un grand nombre de cartes ; nous nous concentrerons dans cette thèse sur des petites architectures de deux et trois cartes afin de comprendre les comportements qui émergent d'un tel système.

Nous étudions l'architecture CxSOM dans un cadre de mémoire associative. L'objectif pour une architecture de cartes est alors d'apprendre une représentation des relations existant entre des entrées de différentes modalités, tout en apprenant une représentation de chaque espace d'entrée. Le but de cette thèse est d'analyser comment est effectuée cette représentation des relations et des entrées dans une architecture CxSOM. La compréhension du comportement de structures avec un faible nombres de cartes posera des bases pour la construction d'architectures plus grandes. Ce système de cartes est en effet un système complexe, même dans une architecture de quelques cartes. Chaque carte possède 500 neurones, donc 500 états possibles. Ces états dépendent des cartes voisines.

Cette thèse s'inscrit dans une démarche complètement expérimentale : nous observons l'organisation d'architectures de cartes sur différents espaces d'entrées, différents configurations, différents paramètres pour comprendre leur comportement. Nous voulons également poser un formalisme clair sur des architectures de quelques cartes pour permettre l'adaptation de CxSOM à plus grande échelle. Ce chapitre s'attelle à définir une méthode expérimentale claire et un formalisme pour cette méthode et pose les représentations utilisées.

L'évaluation de l'organisation d'une SOM classique est directe et très visuelle, cependant nous avons besoin de méthodes pour analyser l'organisation de ces cartes sur plusieurs entrées et plusieurs cartes. Nous posons dans ce chapitre la méthode expérimentale que nous utiliserons dans toutes les expériences présentées dans ce manuscrit. Nous présenterons des représentations adaptées à cette méthode expérimentale et le formalisme utilisé. Le but de ce chapitre est de comprendre la démarche, comprendre les représentations et quels comportements sont observés d'après chaque représentation proposés. Nous introduirons enfin un indicateur caractérisant l'apprentissage de ces relations entre entrées, basé sur l'information mutuelle. Toutes ces représentations seront illustrées sur une architecture de deux cartes.

3.1.1 Présentation de l'expérience d'illustration

La méthode expérimentale sera présentée dans toute ce chapitre sur l'exemple très simple d'une architecture de deux cartes. L'architecture est illustrée à droite en figure 3.1 : elle est composée de deux cartes en une dimension. Chaque carte prend une entrée externe. Il s'agit de $X^{(1)} = x$ et $X^{(2)} = y$, les coordonnées de points 2D sur un cercle. Ces deux modalités sont dépendantes : pour une valeur de x , seule deux valeurs sont possible pour y , et symétriquement. Les entrées sont représentées sur le schéma de gauche, figure 3.1. Ces entrées externes sont normalisées entre 0 et 1. Les deux cartes sont des lignes 1D de 500 noeuds. Les rayons de voisinage sont $h_e = 0.2$ et $h_c = 0.02$. Chacune des deux cartes est également connectée à sa voisine, c'est à dire, la carte $M^{(1)}$ prend en entrée contextuelle la position du BMU de $M^{(2)}$, et inversement. Afin de comprendre les tracés que nous présenterons, nous utiliserons deux cartes de Kohonen classiques en tant que témoin. Une carte prend en entrée les valeurs x , et la deuxième les valeurs y , mais ces cartes ne sont pas connectées entre elles. Les paramètres de ces cartes sont les mêmes que les cartes de CxSOM : 500 noeuds et $h_e = 0.2$.

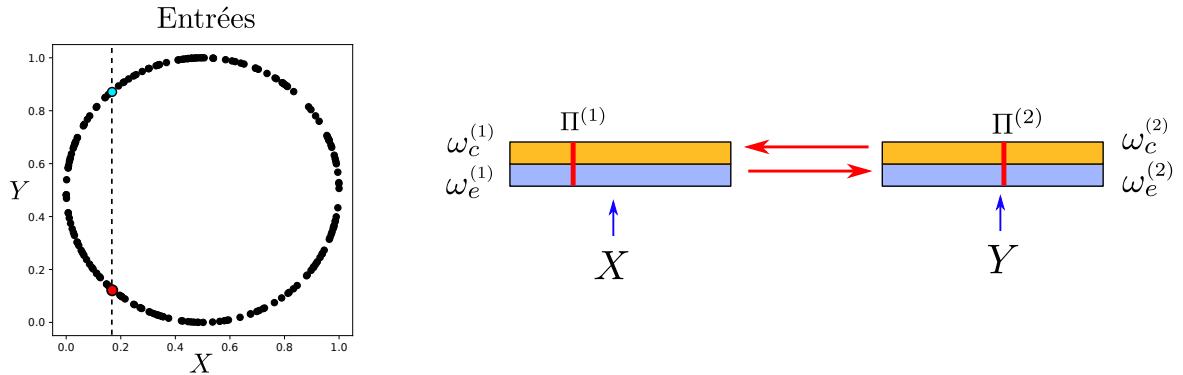


FIGURE 3.1 – Disposition des entrée, sous forme de cercle, à gauche, et architecture de deux cartes en une dimension étudiée et représentée dans ce chapitre.

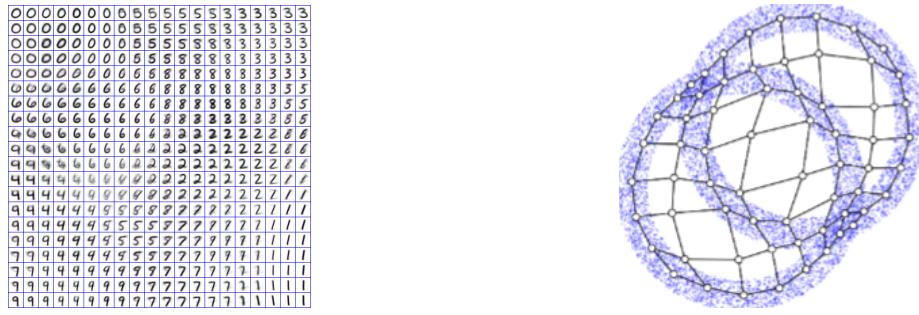


FIGURE 3.2 – Représentations possibles des poids d'une carte de Kohonen classiques, dans le cas d'entrées sous forme d'imagettes ou de points en deux dimensions.

3.1.2 Représentations et indicateurs classique des cartes de Kohonen

Les cartes de Kohonen sont particulièrement associées à une facilité de représentation et de visualisation. Leur nombre réduit de prototypes et leur aspect topologique permet d'en tracer une représentation visuelle interprétable. La manière la plus couramment utilisée de représenter une carte de Kohonen est de tracer les poids de ses prototypes, disposés dans le graphe (ligne ou grille) qu'est la carte. En fonction des dimensions des entrées, cette représentation prennent plusieurs formes. Deux exemples courants de représentation sont les suivants :

- Le graphe qu'est la carte de Kohonen est représenté dans l'espace de ses positions (la grille d'indices (i, j) , ou une ligne indexée par i). Sur chaque noeud est tracé le poids correspondant. C'est le cas sur l'exemple de gauche en figure 3.2 dans lequel les poids des prototypes, qui sont des imagettes, sont affichés en chaque point de la grille.
- Lorsque les données traitées sont des points deux ou trois dimensions, les poids des prototypes peuvent être directement tracés dans l'espace \mathbb{R}^2 ou \mathbb{R}^3 . Ces poids sont alors reliés en fonction des positions des noeuds dans la carte, montrant ainsi la déformation de la carte dans l'espace d'entrée, c'est le cas sur l'exemple de droite en figure 3.2.



3.1.3 Limites de la représentation classique pour CxSOM

Utilisons les représentations classiques mentionnées ci-dessus pour tracer les poids de chacun des cartes d'une architecture CxSOM. Les poids sont tracés à la fin de l'apprentissage. La fin

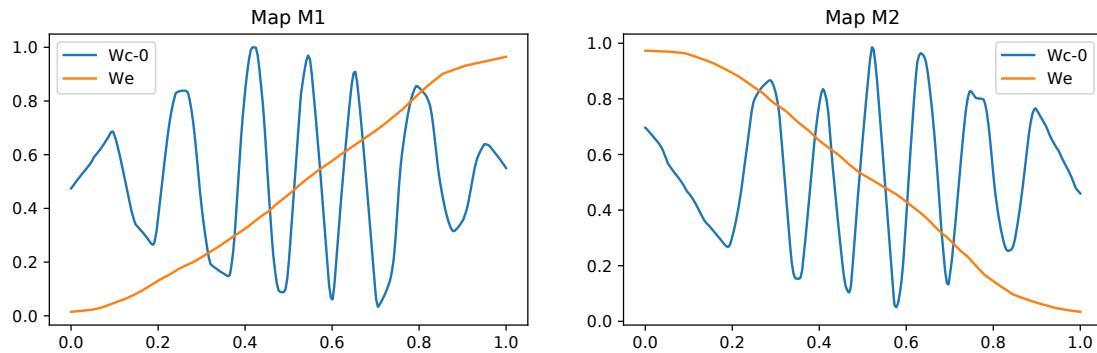


FIGURE 3.3 – Représentation des valeurs des poids d'une carte au sein de CxSOM après apprentissage. La seule représentation de ces poids ne suffit pas à savoir comment la carte se comporte.

de l'apprentissage est définie comme le moment où les poids ont convergé vers une organisation restant stable au cours des itérations t . La figure 3.3 présente le tracé des poids des deux cartes de l'exemple. La courbe orange correspond aux poids externes, se dépliant sur chaque coordonnée x et y des points du cercle, appartenant chacune à $[0, 1]$. Ce tracé permet d'observer que les poids externes couvrent l'intervalle $[0, 1]$, et sont organisés de façon monotone, comme on l'attend dans une carte simple. Les poids contextuels, en bleu, ne présentent pas cette organisation monotone, mais présentent une continuité : deux prototypes proches ont des poids proches. Le tracé nous informe donc sur le caractère continu de l'organisation de chacune de couches de poids.

Notons que nous ne pouvons pas en tirer plus de conclusion : la représentation des poids de la figure 3.3 ne différencie pas les nœuds qui seront effectivement BMUs, des nœuds *morts*. Ces nœuds morts ont bien un poids, mais ne seront jamais BMUs. Dans une carte de Kohonen classique, ces nœuds correspondent à des transitions, liant deux zones denses de l'espace d'entrée séparées par une zone sans points. Par ailleurs, cette représentation concerne une seule carte. Nous ne pourrons pas tirer des informations sur l'influence des connexions entre cartes à partir de ces représentations.

Il est donc nécessaire de trouver un moyen de représenter l'architecture comme un tout. Nous devons définir une représentation qui montre comment l'architecture de cartes est capable d'apprendre les relations entre les entrées multimodales.

Lorsque la dimension des entrées ou le nombre de cartes sont élevés, une représentation graphique ne suffit plus : il y a trop d'éléments à représenter. Cette difficulté de représentation soulève la nécessité de définir des valeurs indicatrices du fonctionnement de la carte qui soient calculables en grande dimension.

Ce chapitre questionne donc la façon de représenter une carte au sein d'une architecture. Nous présenterons en premier lieu le formalisme décrivant les cartes et leurs entrées multimodales associées ainsi que la méthode expérimentale que nous utiliserons pour toutes les expériences présentées dans cette thèse. A partir de ce formalisme, nous proposerons plusieurs représentations et indicateurs permettant de comprendre et représenter ce que calcule une architecture CxSOM sur les données d'entrées.

3.2 Formalisation par variables aléatoires

Nous introduisons dans cette section un formalisme traitant les éléments des cartes et les entrées en tant que variables aléatoires. Ce formalisme a l'avantage de à la fois clarifier les représentations et de permettre le développement d'indicateurs statistiques sur les cartes.

3.2.1 Représentation des entrées

Nous nous plaçons dans une tâche de mémoire associative. Nous considérons plusieurs espaces d'entrée $\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(n)}$ dont seront tirées les entrées présentées au cartes. Chaque espace est une modalité. Les observations multimodales que l'on cherche à apprendre par l'architecture de cartes sont notées ($X^{(i)} \in \mathcal{D}^{(i)}, i = 1 \dots n$). Ces observations $X^{(i)}$ sont modélisées comme des *variables aléatoires*. Chaque variable aléatoire possède une distribution $p^{(i)}$ sur $\mathcal{D}^{(i)}$. Nous notons $\mathbf{X} = (X^{(1)}, \dots, X^{(n)})$ la variable aléatoire jointe. Cette variable appartient à l'espace $\mathcal{D}^{(1)} \times \dots \times \mathcal{D}^{(n)}$. Elle a une distribution jointe. La distribution de probabilité de chaque modalité $X^{(i)}$ sur son espace $\mathcal{D}^{(i)}$ est alors une distribution marginale de \mathbf{X} . A chaque pas de temps, un vecteur $\mathbf{X} = (X_t^{(0)}, \dots, X_t^{(N)})$ est présenté à l'architecture : il s'agit d'une réalisation de la variable jointe \mathbf{X} . On s'intéresse à l'apprentissage de relations entre entrées : les variables $X^{(i)}$ ne sont pas des variables indépendantes.

En pratique, ces variables sont des observations, issues par exemple de capteurs d'un robot. Ces observations sont issues d'un d'environnement général qui est modélisable. Nous introduisons la notion de *modèle d'entrées* se rapportant à cette dépendance entre variables. Le modèle d'entrée fait référence au modèle d'environnement permettant de générer les entrées multimodales fournies en entrées. Dans l'exemple d'illustration, les modalités sont les abscisses $X^{(1)} = x$ et les ordonnées $X^{(2)} = y$; le modèle d'entrées correspond à l'équation du cercle.

Le but de l'apprentissage non-supervisé par des cartes de Kohonen classiques est d'apprendre une représentation discrète de l'espace d'entrée. Avec CxSOM, nous chercherons à la fois à apprendre un représentation discrète des espaces d'entrée mais aussi à apprendre une représentation du modèle d'entrées.

Les tracés et indicateurs que nous développerons dans cette section ont pour but de mesurer comment ce modèle d'entrées est appris par l'architecture.

3.2.2 Démarche expérimentale

Afin d'étudier le comportement de la carte à n'importe quel instant t de l'apprentissage, nous effectuons une pase de *test*, décrit en figure 3.4 Lors de cette phase de test, des entrées sont présentées à la carte, mais seul le processus de recherche de la best matching unit est réalisé, les poids des cartes ne sont pas mis à jour. Cet étape génère un ensemble de réponses de la carte aux entrées présentées. Les entrées utilisées lors du test sont un ensemble de réalisations de la variable aléatoire $(X^{(1)}, \dots, X^{(n)})$.

Définissons aussi des variables aléatoires pour chaque élément des cartes. Nous représentons notamment les positions des BMUs par des variables aléatoires, $\Pi^{(1)}, \dots, \Pi^{(n)}$ et leurs poids $\omega_e^{(1)}(\Pi^{(1)})$ et $\omega_e^{(2)}(\Pi^{(2)})$. Entre deux itérations de test, la valeur de ces éléments ne dépend que de l'entrée, car les poids ne sont pas mis à jour. Une phase de test est donc un ensemble de

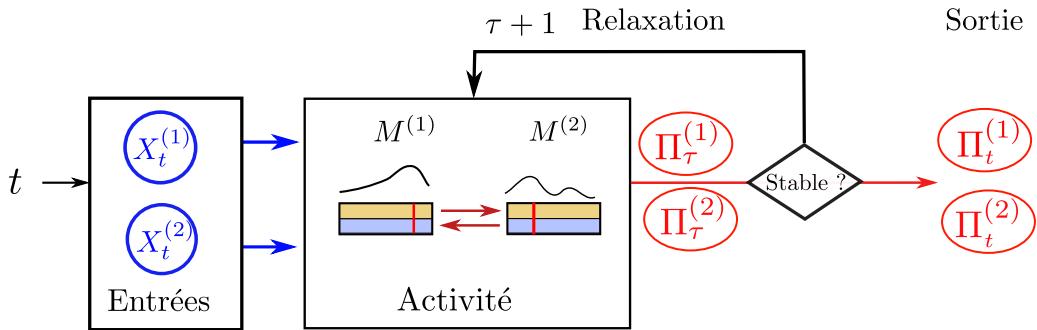


FIGURE 3.4 – Schéma descriptif des tests

réalisations d'une variable aléatoire jointe :

$$(X^{(1)}, \dots, X^{(n)}, \Pi^{(1)}, \dots, \Pi^{(n)}, \omega_e^{(1)}(\Pi^{(1)}), \dots, \omega_e^{(n)}(\Pi^{(n)}))$$

Les composantes de cette variable jointe ne sont pas indépendantes. Les représentations et indicateurs présentés ensuite chercheront à détecter et comprendre au mieux leurs dépendances statistiques. Les variables d'entrées sont à valeurs continues et Π à valeurs discrètes, correspondant aux 500 nœuds d'une carte. Nous considérerons cependant Π comme une variable continue plutôt qu'une grandeur discrète. En effet, l'ensemble des positions du BMU correspondent une discrétisation de l'espace continu $[0, 1]$, et sont ordonnées. Le formalisme par variable aléatoires permet d'utiliser des outils et métriques issus de la théorie de l'information pour qualifier l'organisation des cartes au sein de l'architecture.

3.2.3 Résumé

Pour évaluer le comportement des cartes à une itération t , nous effectuons une phase de test sur les poids à cette itération. Cette phase de test consiste à enregistrer la réponse d'une carte à un grand nombre d'entrées test, sans effectuer de mise à jour des poids. Nous enregistrons ainsi les positions des BMUs résultats de chaque entrée présentées et leur poids externes. Les entrées et les réponses des cartes obtenues lors d'une phase de test sont modélisées comme des variables aléatoires, notées

$$(X^{(1)}, \dots, X^{(n)}, \Pi^{(1)}, \dots, \Pi^{(n)}, \omega_e^{(1)}(\Pi^{(1)}), \dots, \omega_e^{(n)}(\Pi^{(n)}))$$

Une phase de test est un ensemble de réalisation de cette variables aléatoire jointe.

3.3 Représentation graphiques des cartes

A partir des échantillons de tests, nous proposons dans cette section les représentations graphiques que nous utiliserons et leur intérêt. Il s'agit de tracer les dépendances entre les variables

$$(X^{(0)}, \dots, X^{(N)}, \Pi^{(0)}, \dots, \Pi^{(N)}, \omega_e^{(0)}(\Pi^{(0)}), \dots, \omega_e^{(N)}(\Pi^{(N)}))$$

dont les valeurs sont obtenues lors du test. Nous proposons d'abord une représentation pour évaluer la qualité de quantification vectorielle effectuée par une carte de l'architecture. Nous présenterons ensuite des représentations nous permettant de comprendre comment le modèle d'entrées est appris par l'architecture CxSOM.

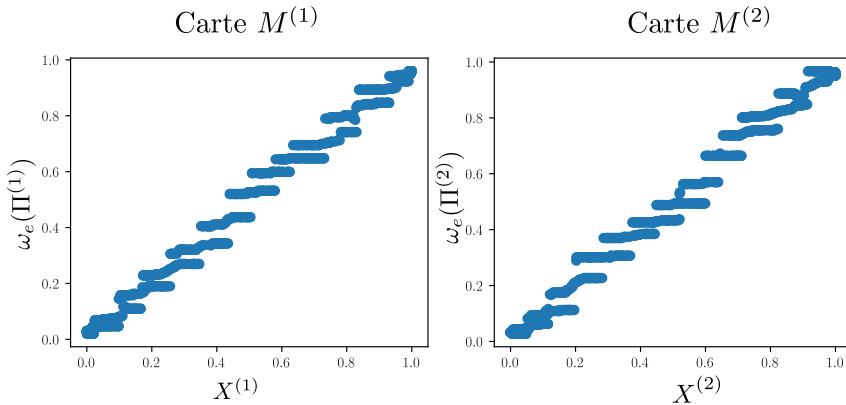


FIGURE 3.5 – Poids du BMU dans chaque carte en fonction de l’entrée présentée. On s’attend à des tracés proches de l’identité, montrant que le poids du BMU d’une carte est une bonne représentation de l’entrée. Sur ce graphique, on se rapproche effectivement de la fonction identité, cependant, une faible erreur est observée. On observe également un découpage des poids en bandes.

3.3.1 Erreur de quantification d’une modalité dans une carte

La première fonction d’une carte de Kohonen est de réaliser de la quantification vectorielle sur son entrée externe. Au sein d’une architecture de cartes, nous nous attendons à ce que chaque carte apprenne une bonne représentation de la modalité qu’elle prend en entrée. Afin de mesurer cette qualité de la quantification vectorielle au sein d’une carte dans CxSOM, nous tracerons le nuage de points correspondant au poids externe du BMU $\omega_e(\Pi^{(i)})$ en fonction de l’entrée présentée $X^{(i)}$. Une carte effectue une quantification vectorielle correcte si ce nuage de points est proche de l’identité. Ces tracés sont réalisés en figure 3.5 pour l’expérience exemple. Ces tracés s’approchent de l’identité : la quantification des entrées est correctement réalisée. On observe néanmoins une erreur assez forte. Cette représentation nous informe sur l’erreur de quantification dans une seule carte relativement à une seule modalité. Il nous faut ensuite définir des méthodes de représentation relatives à l’ensemble de l’architecture et au modèle d’entrée.

3.3.2 Cartographie des entrées

En biologie, les aires du cortex cérébral sont cartographiées en faisant varier le motif d’entrée dans son espace, et en indiquant, pour chacune des valeurs prises par cette entrée, le neurone y réagissant. Cela donne alors une carte ou des valeurs de l’espace d’entrée sont tracées par rapport aux positions de neurones. Par exemple, une carte corticale est tracée pour l’aire visuelle primaire du cortex cérébrale, l’aire v1, en figure 3.6.

De façon similaire, nous tracerons, pour chaque élément de l’échantillon test, la valeur de l’entrée $X^{(i)}$ d’une carte par rapport à la position du BMU $\Pi^{(i)}$ qui a été trouvée par le processus de relaxation. Cette représentation permet d’analyser la quantification des entrées par la carte. Nous mettrons ces éléments en relation avec les poids des cartes en faisant également apparaître les poids externes et contextuels de la carte $M^{(i)}$. On s’attend à ce que les points soient proches de la courbe des poids externes de la carte $M^{(i)}$. Ce tracé fait également apparaître les zones dans lesquelles les neurones ne sont jamais best matching unit, les *zones mortes*.

Sur le même graphique, nous affichons non seulement les couples $(\Pi^{(i)}, X^{(i)})$ mais également les entrées des autres cartes, également en fonction de $\Pi^{(i)}$. En figure 3.7, nous avons ainsi tracé les points $(\Pi^{(1)}, X^{(1)})$ et $(\Pi^{(1)}, X^{(2)})$ issus de l'expérience sur les deux cartes, ainsi que les poids de la carte $M^{(1)}$. Deux échantillons sont mis en couleur rouge et bleue sur chaque graphique. Un point de même couleur correspond au même échantillon dans chaque graphique. Ces deux points partagent la même abscisse, donc l'entrée $X^{(1)}$ est la même pour ces deux échantillons. Par contre, leur ordonnée est différente et $M^{(2)}$ reçoit donc une entrée $X^{(2)}$ différente dans ces échantillons.

Ce tracé nous permet d'abord d'observer que les points $(\Pi^{(1)}, X^{(1)})$ sont proches de la courbe de poids externe : le poids d'un BMU est proche de l'entrée qui a été présentée, le poids du BMU est donc une bonne approximation de cette entrée. Cela permet de conclure que la quantification vectorielle est bien réalisée dans cette carte sur les entrées externes.

Tracer les échantillons de test permet ensuite d'observer la répartition des BMUs sur la carte. Les courbes de poids externes de la carte dans CxSOM (c) et de la carte indépendantes (b) sont indifférenciables ; par contre, l'affichage de l'échantillon de test fait apparaître des zones mortes. Nous observons ainsi que la carte au sein de CxSOM est découpée en plusieurs zones dans lesquelles les unités sont BMUs, séparées par des petites zones mortes. Ce tracé permet donc d'identifier un comportement nouveau à investiguer.

En s'intéressant aux valeurs des entrées, nous observons aussi que les BMUs d'une même zone encodent des entrées sur un intervalle réduit, mais que les intervalles encodés par deux zones adjacentes se recoupent. Les points rouges et bleus, ayant la même valeur de $X^{(1)}$, sont par exemple envoyés dans deux zones différentes.

Ces valeurs peuvent être mises en relation avec celles de l'entrée $X^{(2)}$, également disposées selon $\Pi^{(1)}$. Nous pouvons alors observer que deux zones adjacentes de la carte encodent des entrées proches selon $X^{(1)}$, mais très différentes pour $X^{(2)}$. Cela explique la séparation du point rouge et du point bleu dans deux zones.

La représentation des valeurs d'entrées $X^{(i)}$ selon la position du BMU calculée $\Pi^{(i)}$ permet ainsi d'identifier une répartition des BMUs sur la carte que nous ne pourrions pas observer en traçant simplement les poids.

3.3.3 Réduction de dimension des entrées

Lorsque la dimension totale des entrées dépasse trois, l'interprétation visuelle de toutes les entrées sur le tracé précédemment présenté n'est plus possible. Nous chercherons donc à faire une réduction de dimension des entrées pour pouvoir les tracer. Des méthodes de réduction de dimension sont largement utilisées dans les applications de l'apprentissage automatique, telles que T-SNE [27] ou l'analyse en composantes principales (PCA). A partir de données d'entrée en grande dimension, la réduction de dimension cherche à associer à chaque vecteur d'entrée un vecteur de plus faible dimension, en l'occurrence deux ou trois, pour que l'entrée puisse être représentée sur un graphique. Cette valeur de dimension plus faible doit être une "bonne" représentation des données originale. La définition de la qualité de cette représentation dépend de l'algorithme utilisé. Ainsi, T-SNE priorise la conservation des relations de proximité entre les points, la PCA conserve les axes sur lesquels la covariance est maximale. La réduction de dimension, dans la plupart des cas, perd de l'information sur les entrées. Cette perte d'information doit être prise en compte lors de l'analyse des données.

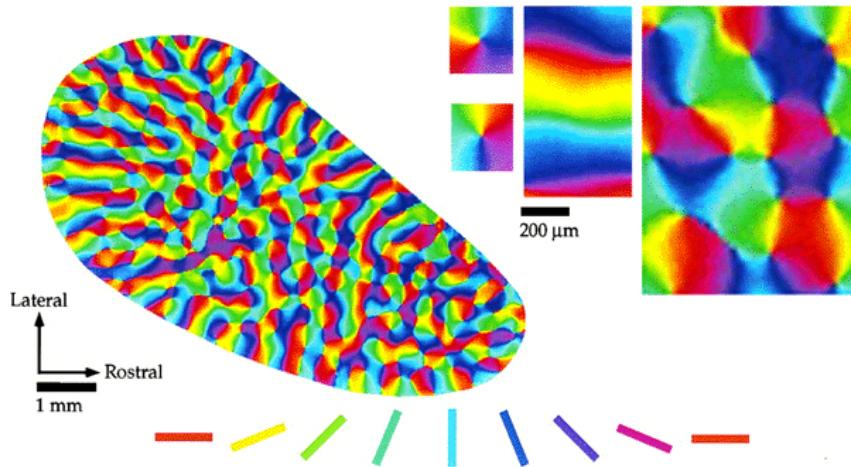


FIGURE 3.6 – Carte corticale de l'aire cérébrale visuelle V1. Pour tracer cette représentation, un ensemble de traits de différentes orientation sont présentés en stimuli visuels au sujet, indiqués en bas de l'image. Le neurone réagissant à une entrée d'orientation particulière est coloré sur la carte de la couleur correspondante à l'entrée. Cette méthode permet de tracer des *cartes corticales* d'une aire cérébrale [2].

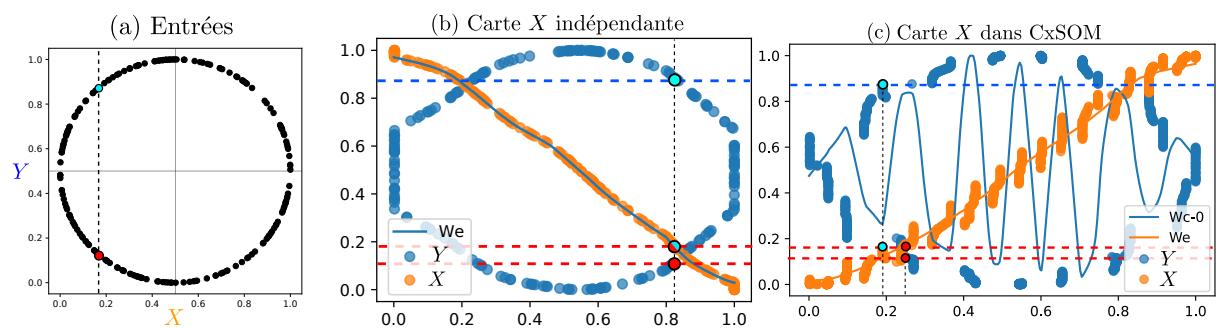


FIGURE 3.7 – Représentation des entrées X, Y d'une architecture de deux cartes relativement au BMU de la carte X après apprentissage. Ces tracés mettent en valeur l'organisation des cartes, différentes dans le cas où les cartes apprennent indépendamment leurs entrées (b) ou sont connectées (c). Les entrées correspondantes sont en figure (a). Les points bleu et rouge reportés sur les tracés correspondent au même échantillon de test.

Nous nous placerons dans un cas particulier de réduction de dimension sans perte d'information. Dans notre cas particulier, les entrées sont générées par un modèle géométrique. Il est alors possible d'effectuer une réduction de dimension en faisant une paramétrisation géométrique du modèle générant les entrées. Nous définissons une variable supplémentaire U , de dimension inférieure ou égale à la somme des dimensions des entrées. Elle choisie de façon à ce que chaque variable $X^{(i)}$ soit une fonction de la variable aléatoire U , et uniquement de cette variable.

$$\forall i, X^{(i)} = f^{(i)}(U) \quad (3.1)$$

Pour que la variable U conserve toute l'information sur le modèle, la fonction $(f^{(1)}, \dots, f^{(N)}) : (X^{(1)}, \dots, X^{(N)}) \rightarrow U$ doit être une bijection. Toute valeur jointe d'entrée correspond à un seul U , toute valeur de U renvoie à une seule valeur d'entrée jointe. Cet aspect est un cas particulier par rapport aux méthodes de réduction de dimension classique, car il n'y a pas de perte d'information. Dans le cas d'exemple, $\mathbf{X} = (X, Y)$, les coordonnées cartésiennes des points du cercle est alors une vecteur aléatoire, dont les composantes sont les variables aléatoires X, Y . En définissant une variable U à valeurs dans $[0, 1]$, chaque point du cercle peut maintenant s'écrire, selon l'équation paramétrique du cercle :

$$\begin{cases} X = r \cos(2\pi U) \\ Y = r \sin(2\pi U) \end{cases} . \quad (3.2)$$

U représente ici l'angle du point sur le cercle (à un facteur 2π près). Les exemples donnés sont scalaires, mais cette représentation générale à n'importe quel dimension et nombre d'entrées.

Cette variable ajoute donc un attribut aux données échantillonées durant une phase de test. Une phase de test donne donc un ensemble de valeurs de U , en plus des positions de BMUs $\Pi^{(i)}$ et de leurs poids.

Nous avons vu lors de la représentation cartographique des entrées que chacune des cartes s'organise en fonction non seulement de son entrée externe, mais aussi de l'entrée de l'autre carte. Chaque carte s'organise donc en fonction du *modèle d'entrée*, donc en fonction de U . Nous tracerons les nuages de points U en fonction de la position $\Pi^{(i)}$ du BMU d'une carte pour représenter comment la position du BMU traduit la relation entre les entrées. Dans le cas de l'expérience à deux cartes, U est en une dimension.

En figure 3.9, nous traçons donc U en fonction de $\Pi^{(1)}$ et U en fonction de $\Pi^{(2)}$. Ce tracé montre U comme une fonction de la position du BMU dans chaque carte, contrairement au cas où les cartes ne sont pas connectées. Cela traduit bien le fait que chaque carte a appris une représentation du modèle d'entrée et non seulement de son entrée externe. Dans le cas où les cartes ne sont pas connectées, il y a plusieurs valeurs de U pour un même Π , une valeur x de $X^{(1)}$ correspondant à deux positions sur le cercle. L'organisation de la carte dans CxSOM rend chaque position Π codant pour une seule valeur de U , c'est à dire une seule position d'échantillon sur le cercle. U est alors une fonction de Π , ce qui est représenté en pointillé sur la figure 3.9. Donc, chaque carte $M^{(1)}$ et $M^{(2)}$ encode tout le modèle d'entrée, et non seulement l'entrée externe. La représentation de U selon la position du BMU d'une carte $\Pi^{(i)}$ permet de représenter comment la carte i a appris l'ensemble d'entrées $(X^{(1)}, X^{(2)})$ et non seulement son entrée externe. Déterminer si l'architecture a appris le modèle d'entrées revient à vérifier si U est une fonction de Π dans chacune des cartes de l'architecture.

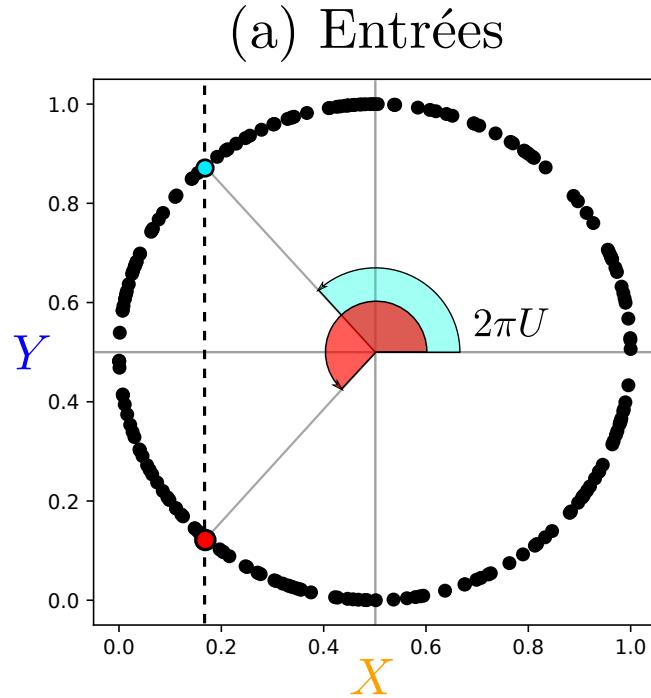


FIGURE 3.8 – Représentation choisie pour le cercle. Le modèle auxquelles appartiennent les modalités $X^{(1)}$ et $X^{(2)}$ est représenté par la variable U .

3.3.4 Résumé

Nous utiliserons donc trois représentations des cartes dans la suite de cette thèse, toutes définies à partir d'un échantillon de test. Le tracé du poids du BMU en fonction de l'entrée externe permet d'évaluer comment une carte a individuellement appris une représentation de son espace d'entrée. Pour observer comment l'architecture a appris le modèle d'entrée, nous traçons une représentation cartographique des entrées. Sur un même graphique nous traçons les nuages de points $(X^{(1)}, \Pi^{(1)})$ et $(X^{(2)}, \Pi^{(1)})$, de même pour $M^{(2)}$, superposés avec les courbes de poids de $M^{(1)}$ (respectivement $M^{(2)}$). Cette représentation permet d'observer quelles valeurs d'entrées chaque position de BMU encode. Enfin, nous réduisons si possible la dimension du modèle d'entrée $(X^{(1)}, \dots, X^{(n)})$ en une valeur U et traçons les nuage de points $(U, \Pi^{(1)})$ et $(U, \Pi^{(2)})$. L'apprentissage du modèle d'entrée doit se traduire par l'observation d'une relation fonctionnelle entre U et Π dans chaque carte.

3.4 Un indicateur de l'apprentissage du modèle par l'architecture basé sur l'information mutuelle

L'étude de tout processus physique s'effectue par un ensemble signaux issus de capteurs. La théorie de l'information de Shannon [24] apporte un modèle mathématique qui permet de les manipuler, les encoder, les décoder et quantifier l'apport ou perte d'information entre eux, en les utilisant en tant que distributions de probabilités. Nous investiguerons dans cette partie comment quantifier l'apprentissage de l'architecture de cartes par des outils d'information. En particulier,

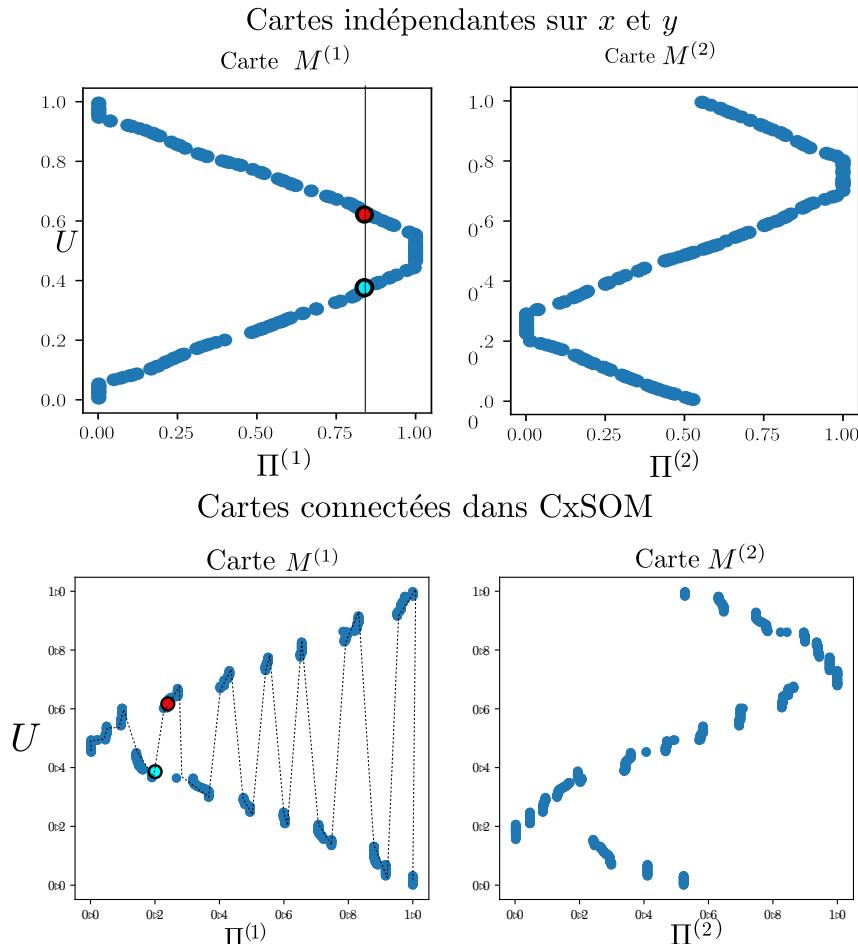


FIGURE 3.9 – Valeur de U en fonction des valeurs du BMU $\Pi^{(i)}$ dans chacune des cartes, pour des entrées prises sur le cercle. Sur la première ligne, nous traçons la réponse de chaque carte à son entrée dans le cas où les cartes ne sont pas connectées. Sur la deuxième ligne, nous traçons la réponse de chaque carte lorsqu'elles ont appris de façon jointe au sein de CxSOM. U apparaît alors comme une fonction de la position du BMU $\Pi^{(i)}$ dans chaque carte, contrairement au cas où les cartes apprendraient indépendamment sur les mêmes entrées. Cette relation fonctionnelle est symbolisée par les pointillés sur les tracés du bas.

l'estimation de ces grandeurs à partir de données est un élément clé et parfois limitant pour utiliser la théorie de l'information dans un cadre expérimental. Nous définirons dans cette partie les quantités que nous mesurerons dans l'architecture de cartes et comment les estimer.

3.4.1 Information mutuelle et entropie

Les notions d'*entropie* et les valeurs associées, telle que l'*information mutuelle* entre des distributions, sont des notions fondamentales de la théorie de l'information de Shannon. Ces quantités donnent des informations concernant la distribution d'une variable aléatoire. Les formules indiquées dans ce paragraphe concernent des variables aléatoires discrètes mais sont transposables en version continue. L'entropie de Shannon d'une variable aléatoire X à valeurs discrètes dans un ensemble E_X , de distribution p , est notée $H(X)$ et définie par la formule :

$$H(X) = - \sum_{x \in E_X} p(x) \log(p(x)) \quad (3.3)$$

Elle se mesure en *bit/symbole* lorsque le logarithme est en base 2, ce qui est généralement utilisé. L'entropie est une mesure de la quantité d'incertitude, ou de surprise, sur la valeur de la variable aléatoire X . Si la distribution de probabilité de X est concentrée autour d'un point, l'entropie est faible : lors d'une réalisation de X , l'observateur est *plutôt certain* du résultat. En revanche, l'entropie est maximale lorsque lorsque X suit une distribution de probabilité uniforme. L'entropie s'interprète également comme la quantité moyenne d'information à fournir, en bits, pour coder la valeur que prend la variable X . De la même manière, on peut définir l'entropie conjointe de deux variables, qui est l'entropie de leur distribution jointe, et l'entropie conditionnelle, qui est l'entropie de leurs distributions conditionnelles.

Outre les entropies jointes et conditionnelles, les relations statistiques entre deux variables aléatoires $X, Y \in E_X, E_Y$ peuvent être mesurées par l'*information mutuelle*. Elle se définit formellement par :

$$I(X, Y) = \sum_{x, y \in E_X, E_Y} p(x, y) \log\left(\frac{p(x, y)}{p(x)p(y)}\right) \quad (3.4)$$

Cette valeur mesure la quantité d'information moyenne apportée par une réalisation de X sur la réalisation de Y .

L'information mutuelle possède les propriétés suivantes :

1. $I(X, Y) = 0 \Leftrightarrow X$ et Y sont indépendantes. L'information mutuelle peut être vue comme une mesure de la distance entre la distribution jointe de (X, Y) , $p(X, Y)$ et la distribution dans laquelle les deux variables sont indépendantes, $p(X)p(Y)$.
2. Elle s'exprime à partir de l'entropie : $I(X, Y) = H(X) + H(Y) - H(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$
3. Elle est symétrique : $I(X, Y) = I(Y, X)$
4. Pour toute fonction f , $I(X, Y) \geq I(X, f(Y))$. L'égalité est atteinte si et seulement si f est *bijective*.

Lors de l'analyse de CxSOM, on s'interroge sur l'information que portent les positions des BMUs II d'une carte sur le modèle d'entrée. Les éléments de la carte ont été définis en termes statistiques ; on peut donc utiliser l'information mutuelle comme une représentation de l'information portée par le BMU d'une carte sur le modèle d'entrée.

3.4.2 Indicateur

Nous avons introduit U , la variable correspondant à une réduction de dimension des entrées ($X^{(i)}$). Dans l'exemple, U est en une dimension pour des entrées 2D ($X^{(1)}, X^{(2)}$). Nous avons observé que si l'architecture a appris le modèle s'entrée, alors U est une fonction de Π dans chacune des cartes. Cette relation fonctionnelle montre qu'une carte a appris non seulement une représentation de son entrée mais aussi une représentation du modèle complet, représenté par ($X^{(1)}, \dots, X^{(n)}$) ou par la variable U . Dans notre cas, U est en bijection avec les entrées. Nous pouvons utiliser $I(\Pi, U)$ comme l'information moyenne qu'une position de BMU d'une carte porte sur U , donc sur le modèle d'entrées, et U sur le BMU. Il est nécessaire ici que la réduction de dimension donnant U soit une relation bijective.

Nous voulons privilégier un indicateur normalisé, qui nous permettrait, sur une échelle de 0 à 1, de quantifier à quel point un BMU porte de l'information sur U . Cet indicateur sera ainsi comparable entre plusieurs expériences. Pour cela, nous normalisons l'information mutuelle $I(\Pi, U)$ par la valeur maximale qu'elle peut prendre dans notre carte. Cette valeur maximale atteinte par $I(\Pi, U)$ est $H(U)$, atteinte lorsque U est fonction de Π , qui est ce qu'on attend de la réponse d'une carte.

En effet, par construction, Π est une fonction de U dans une carte de Kohonen : l'algorithme est déterministe et une sortie est définie pour toute valeur de U . C'est à dire, $I(U, \Pi) = I(U, f(U))$. Par propriété de l'information mutuelle, pour toute fonction f et variables X, Y , $I(X, f(Y)) \leq I(X, Y)$. Donc, $I(U, \Pi) \leq I(U, U) = H(U)$. Cette valeur est atteinte si et seulement si U et Π sont en bijection, autrement dit, si et seulement si U est aussi une fonction de Π .

Nous définissons donc un indicateur de la relation entre U et un BMU comme :

$$I_x(U|\Pi) = \frac{I(\Pi, U)}{H(U)} \quad (3.5)$$

Ce coefficient n'est pas symétrique, et mesure donc l'information portée par le premier terme sur le second, relativement à la valeur maximale qu'elle peut prendre. Dans le cas des cartes CxSOM, $I_x \in [0, 1]$. Cette valeur rappelle le *coefficient d'incertitude* entre U et Π [26].

Ce coefficient peut être élargi à plus de variables : on peut calculer $I_x(U|(\Pi^{(1)}, \Pi^{(2)}, \Pi^{(3)}))$ pour 3 cartes, en considérant la variable jointe $(\Pi^{(1)}, \Pi^{(2)}, \Pi^{(3)})$. Plus largement, pour prouver que l'archicecture a appris un modèle, on souhaite que $I_x(U|\Pi^{(1)}, \dots, \Pi^{(k)})$ soit le plus proche possible de 1.

3.4.3 Estimation

L'information mutuelle et l'entropie sont des grandeurs probabilistes. Elles sont définies à partir de la distribution des variables aléatoire. Lorsque qu'on ne connaît pas les distributions, il est nécessaire d'estimer ces valeurs autrement. Nous passons par une estimation de la distribution des variables U, Π et la distribution de la variable jointe (U, Π) en discrétilisant l'espace par la méthode des *histogrammes*. Cette méthode est représentée en figure 3.10. Les variables U et Π sont discrétilisées en *boîtes* de centres x_k et y_k . La distribution de X est alors estimée par :

$$P(U = x_i) = \frac{n_{xi}}{N}$$

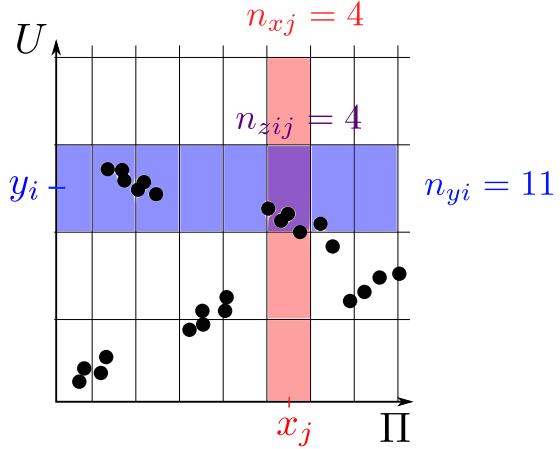


FIGURE 3.10 – Méthode par histogrammes pour estimer les distributions des variables U et Π . Les distributions sont estimées à partir de n_{xj} , n_{yi} et n_{zij} , puis les valeurs de l'entropie H et l'information mutuelle I calculées.

où n_{xi} est le nombre d'échantillons de X tombant dans la boîte de valeur x_i et N le nombre de points. Le même procédé est réalisé pour Π et (U, Π) . La précision de l'estimation peut être améliorée en choisissant des tailles de boîtes variées ; nous utilisons ici la méthode simple avec des boîtes de taille fixe. Pour des variables à valeur dans $[0, 1]$, les centres sont définis par $x_k = \frac{k}{M} + \frac{1}{2M}$, avec M le nombre de boîtes. L'information mutuelle et l'entropie sont ensuite estimées à partir de ces distributions discrètes, par leur formules :

$$I(U, \Pi) = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} P(x_i, y_j) \log \left(\frac{P(U = x_i, \Pi = y_j)}{P(U = x_i) \times P(\Pi = y_j)} \right) \quad (3.6)$$

La méthode par histogramme est simple, mais peu fiable quand la dimension des entrées augmente. Le nombre d'échantillons disponibles pour l'estimation doit augmenter exponentiellement avec la dimension des variables pour éviter le phénomène de "boîtes vides" : à cause de la dispersion des données, de nombreuses boîtes (x_j, y_i) ne contiendront pas de points alors qu'elles auraient dû en contenir selon leur distribution ; l'estimation de la distribution en ce point sera alors nulle, et l'estimation de l'indicateur faussée.

3.4.4 Information mutuelle sur deux cartes

Le fait d'utiliser un indicateur numérique permet de tracer l'évolution de cet indicateur au cours de l'apprentissage des cartes. Nous traçons cette évolution au cours de l'apprentissage dans un système de deux cartes apprenant sur le cercle en deux dimensions.

Pour cette expérience, une phase de test sur 5000 entrées est réalisée toutes les 10 itérations, puis toutes les 200 itérations à partir de l'itération 200. Chaque phase de test donne alors un ensemble d'entrées $X^{(1)}, X^{(2)}, U$ et un ensemble de réponses des cartes $\Pi^{(1)}, \Pi^{(2)}$. On peut alors estimer $I_x(U|\Pi^{(1)})$ et $I_x(U|\Pi^{(2)})$ sur chaque itération considérée, ce qui nous donne la courbe de l'évolution de l'indicateur au long de l'apprentissage. Ces calculs sont réalisés sur 100 apprentissages complets, prenant des entrées d'apprentissage aléatoires sur le même cercle. Les cartes sont initialisées à des poids aléatoires au début de chaque apprentissage. Les tracés présentés

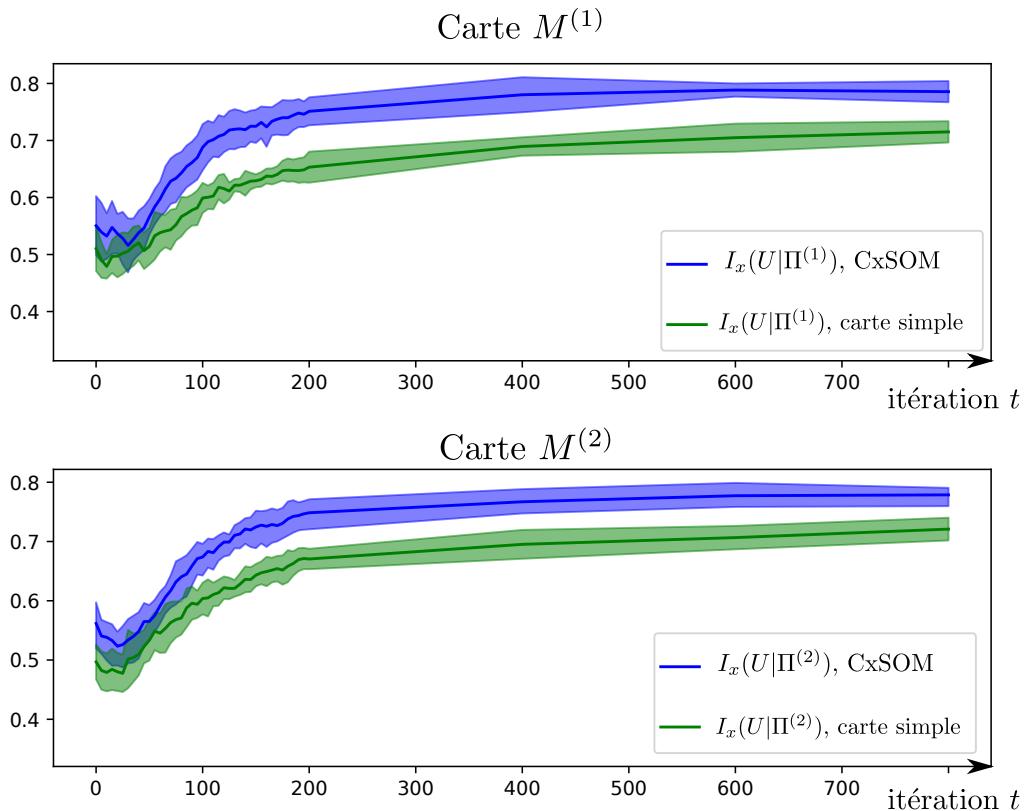


FIGURE 3.11 – Evolution du coefficient d’incertitude dans chaque carte au long de l’apprentissage. La courbe bleue correspond à $I_x(U|\Pi)$ dans l’architecture de cartes $M^{(1)}$ et $M^{(2)}$. On compare cette évolution à l’évolution de l’information d’une seule carte apprenant sur les mêmes entrées X ou Y , sans être connectée.

en figure 3.11, sont la moyenne, à chaque pas de temps, de l’information mutuelle de chaque expérience au pas de temps t . Nous avons représenté ici l’évolution de l’information mutuelle moyenne sur 50 expériences.

L’estimation est réalisée par la méthode des histogrammes. On choisit de découper les valeurs de U en 50 boîtes, et en 500 pour $\Pi^{(i)}$; de cette façon, les points correspondant à des U très proches seront comptés ensemble pour l’estimation de l’information..

Nous comparons les valeurs obtenues pour une carte de CxSOM à celles d’une carte apprenant sur les mêmes entrées $X^{(1)}$ ou $X^{(2)}$, mais sans connexions entre elles. On s’attend à ce que l’information soit plus élevée pour la carte au sein de CxSOM que la carte seule, ce qui montrerait que la carte porte aussi de l’information sur l’autre entrée. On s’attend à ce que cette valeur atteigne 1, ce qui montrerait qu’une seule carte porte de l’information sur tout le modèle : U est une fonction de Π dans chaque carte.

L’observation du tracé montre que les quantités $I_x(U|\Pi^{(1)})$ et $I_x(U|\Pi^{(2)})$ sont bien toutes deux plus élevées à chaque moment de l’apprentissage que dans le cas où les cartes sont séparées. Ces quantités augmentent au cours de l’apprentissage, traduisant bien un gain d’information des cartes sur le modèle.

3.4.5 Discussion

Influence de l'estimation

L'estimation par histogramme est peu fiable en grande dimension. Elle suffit pour des données 2D, mais peut difficilement être passée à l'échelle. Nous avons donc envisagé d'autres estimateurs moins biaisés en grand dimension. Détailons par exemple l'estimateur par KNN (K-nearest neighbors) de Kraskov [19]. Cet estimateur ne passe pas par l'estimation de la densité de probabilité, contrairement aux histogrammes, mais estime directement l'information mutuelle. C'est l'estimation de la densité de probabilité qui pose justement problème en grande dimension. Le découpage de l'espace se fait en recherchant, pour un couple (X, Y) les k plus proches voisins. Une information mutuelle locale est calculée dans cette zone de l'espace, suivant une formule permettant d'approximer les différences de logarithme par la fonction digamma ψ :

$$i_j(X, Y) = \psi(k) - \psi(n_{x_j} + 1) - \psi(n_{y_j} + 1) + \psi(N)$$

Cette information mutuelle locale est ensuite moyennée sur l'ensemble des points :

$$\hat{I}(X, Y) = \psi(k) - \langle \psi(n_{x_j} + 1) + \psi(n_{y_j} + 1) \rangle + \psi(N)$$

Pour estimer $I_x(X|Y)$, on estimera $I(X, Y)$ et $H(Y)$ avec les mêmes paramètres, en notant que $H(Y) = I(Y, Y)$.

Comparons les deux méthodes d'estimation de l'information mutuelle proposées dans ce chapitre : l'estimation par histogrammes, et l'estimation par KNN de Kraskov. En figure 3.13, nous traçons l'évolution de l'information mutuelle moyenne, cette fois estimée par la méthode de Kraskov. Sur le même schéma, nous traçons également l'évolution de l'information mutuelle obtenue par la méthode des histogrammes.

Sur les tracés, l'indicateur calculé avec la méthode de Kraskov converge vers une même valeur à la fin de l'apprentissage pour la carte simple que pour la carte au sein d'une architecture CxSOM (tracés rouges et noirs). Ce résultat est étonnant : cela signifie donc que la carte au sein de CxSOM n'a pas plus d'information sur le modèle que la carte isolée, lorsque cette information est estimée avec la méthode de Kraskov. Ce résultat va également à l'encontre de ce qu'on observe sur l'évolution de l'information mutuelle calculée par les histogrammes, dans laquelle une différence franche est observée entre la carte isolée et la carte au sein de l'architecture.

Proposons une explication. La méthode des noyaux de Kraskov est plus "granulaire" que la méthode des histogrammes au niveau de l'estimation, c'est à dire que les données sont considérées *points par point*. L'avantage est que l'évaluation de la relation fonctionnelle entre U et $\Pi^{(i)}$ est plus précise qu'avec les histogrammes : s'il y a deux valeurs de U pour un même $\Pi^{(i)}$, l'information diminue. Par contre, la contribution de ces deux valeurs sera la même dans le calcul de l'information mutuelle, qu'elles soient proches ou éloignées : la distance entre donnée n'intervient pas dans le calcul. Cette différence de contribution est illustrée en figure 3.14. On calcule l'information mutuelle I_x dans les deux cas de distribution proposées, à l'aide de l'estimateur granulaire de Kraskov. Dans le cas où la relation est proche d'une relation fonctionnelle, mais est très bruitée, l'information $I_x(Y|X)$ est plus faible que dans le cas où cette relation n'est pas fonctionnelle, mais sans bruit parasite.

On observe que l'information $I_x(U|\Pi)$ tend vers une même valeur dans CxSOM et dans une carte isolée quand on la calcule avec l'estimateur granulaire. Cela signifie, qu'on a la même quantité d'information sur U avec le BMU, dans la carte isolée que dans CxSOM. Simplement, cette

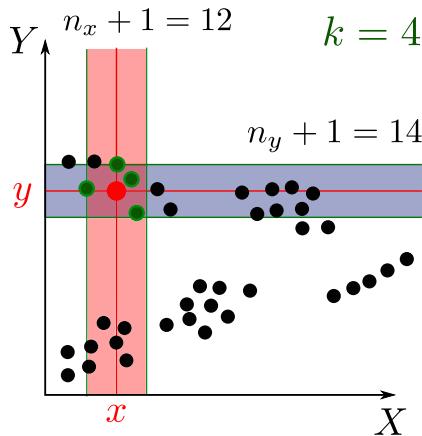


FIGURE 3.12 – Découpage en KNN de Kraskov pour estimer l’entropie et l’information mutuelle des variables X et Y . Les plus proches voisins du point rouge sont trouvés, en vert, et le processus est répété sur tous les points. Les valeurs de n_x et n_y permettent d’estimer directement l’entropie.

information n'est pas répartie de la même façon. Dans une carte isolée, le niveau de quantification vectorielle qu'on effectue sur X est très précis : lorsqu'on présente une entrée X à la carte, le poids du BMU est très proche de cette valeur X . Dans CxSOM, on perd ce niveau de quantification, ce qu'on a observé en figure 3.5. Le fait que l'indicateur, lorsqu'il est estimé avec une méthode très granulaire, prend la même valeur dans les deux expériences traduit alors qu'on a perdu de l'information sur l'entrée X par rapport à la carte isolée, avec la perte de précision, mais qu'on a gagné de l'information sur l'autre entrée Y . Le fait que les deux évolutions de I_x , pour chaque expérience, convergent vers la même valeur montre qu'on est dans une situation de compromis : on gagne de l'information sur le modèle au détriment de l'information sur l'entrée externe.

C'est donc le fait de discréteriser grossièrement la distribution de U qui permet de mesurer le gain d'information sur le modèle complet, sans prendre en compte le fait que la précision sur l'entrée externe est affaiblie. L'indicateur I_x reste un indicateur fiable, mais il doit être utilisé en prenant en compte cet aspect.

Perspectives

L'indicateur que nous proposons, $I_x(Y|X) = \frac{I(X,Y)}{H(Y)}$ traduit bien le fait que les cartes ont appris une relation entre les entrées. Son estimation doit passer par une discréterisation avec des intervalles larges pour U , afin de ne pas prendre en compte la perte d'information sur l'entrée externe X . Les données doivent également être débruitées avant l'estimation. L'information mutuelle et l'entropie étant des quantités fondamentales en théorie de l'information, il existe de nombreuses méthodes d'estimations de ces valeurs malgré la difficulté qu'elle pose, voir [6] pour une revue de différentes méthodes. Ainsi, l'utilisation du coefficient d'incertitude comme indicateur reste robuste pour des données de plus grande dimension ou pour plus de cartes, en utilisant des méthodes d'estimations plus élaborées. Cependant, cette estimation devra être retravaillée en plus grande dimension.

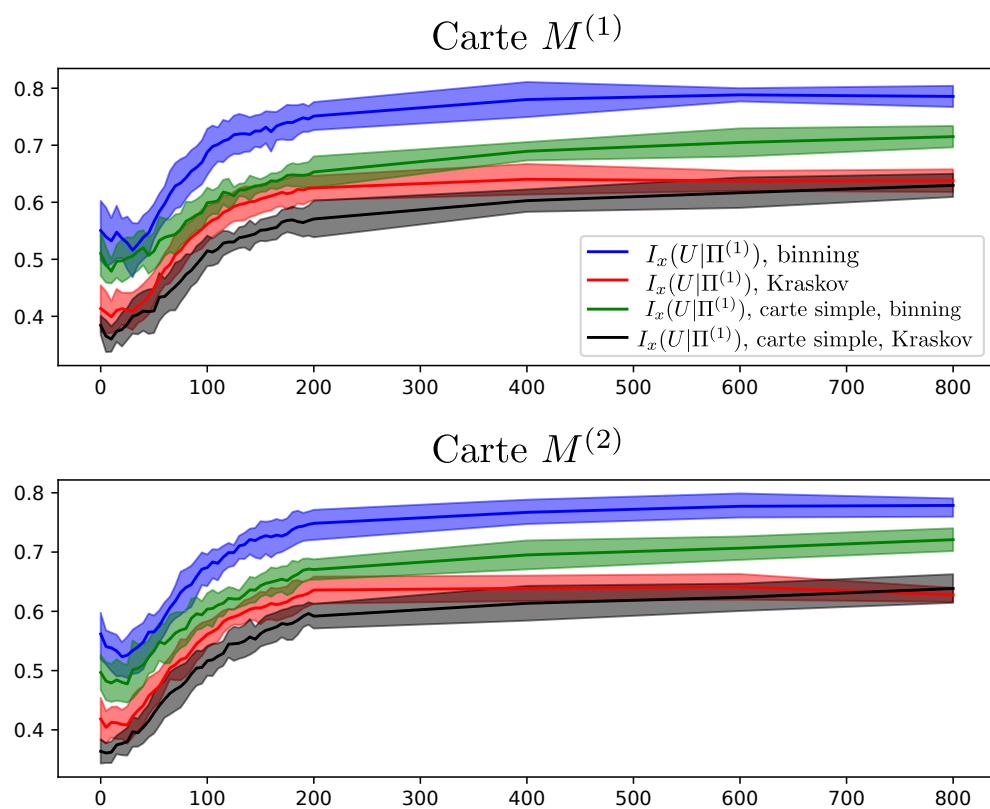


FIGURE 3.13 – Evolution du coefficient d'incertitude dans chaque carte au long de l'apprentissage, en comparant l'estimation par histogrammes et l'estimation par la méthode de Kraskov.

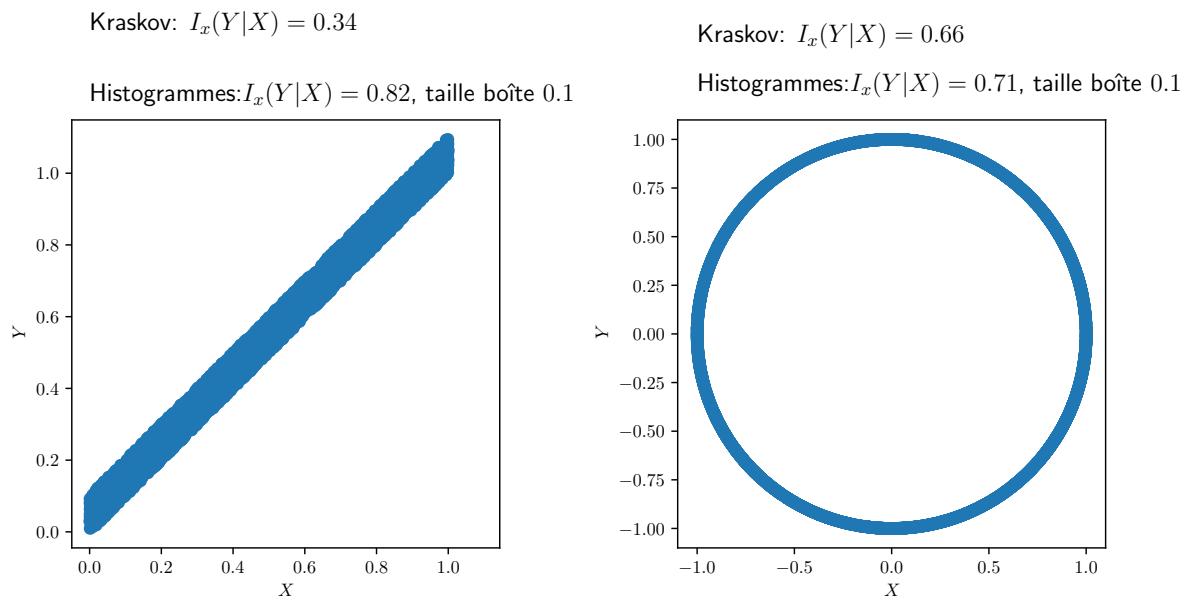


FIGURE 3.14 – Comparaison du calcul de l'indicateur I_x sur deux distributions. A gauche, la relation entre Y et X se rapproche d'une fonction, mais bruitée. A droite, la relation n'est pas fonctionnelle, mais de telle sorte qu'une valeur de X correspond au maximum à deux valeurs de Y . Lorsqu'on calcule l'indicateur avec la méthode très granulaire de Kraskov, cet indicateur est plus élevé dans le cas de droite que de gauche : en effet, le calcul ne prend pas en compte si les points sont condensés ou éloignés. Pour que l'indicateur nous informe correctement sur l'aspect fonctionnel de la relation entre X et Y , il faut enlever manuellement le bruit. Avec la méthode par histogrammes, on prend une taille de boîte de 0.1 selon Y . Dans ce cas, l'indicateur est bien plus élevé à gauche, où Y se rapproche d'une fonction de X , que à droite.

3.5 Conclusion

- La représentation par échantillonnage et variable aléatoire permet de mieux comprendre les mécanismes des cartes, ceux ci ne reposant plus directement sur les courbes de poids
- Ces tracés montrent qu'une architecture de deux cartes s'organise comme une carte simple, mais modulée par l'entrée contextuelle : les poids externes se déplient comme une carte simple, mais les poids contextuels amènent des zones dans la carte. Ces zones séparent les BMUs en fonction de à la fois les entrées externes (organisation générale), et des entrées contextuelles (localement). Observation d'un nombre réduit de zones.
- Perte de précision au niveau de la quantification des poids externes, mais apprentissage d'un modèle. Nécessité de faire un compromis, réalisé de façon auto-organisée. Chaque carte a alors appris le modèle en entier et non seulement son entrée.
- Utilisation d'un indicateur basé sur l'info mutuelle pour évaluer comment une carte apprend le modèle. Indicateur pouvant être utile en grande dimension ; mais l'estimation peut poser problème à ce moment.

Bibliographie

- [1] D. Alahakoon, S.K. Halgamuge, and B. Srinivasan. Dynamic self-organizing maps with controlled growth for knowledge discovery. *IEEE Transactions on Neural Networks*, 11(3) :601–614, May 2000. Conference Name : IEEE Transactions on Neural Networks.
- [2] William H Bosking, Y. Zhang, Brett R. Schofield, and David Fitzpatrick. Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex. *The Journal of Neuroscience*, 17 :2112 – 2127, 1997.
- [3] M. Cottrell, J.C. Fort, and G. Pagès. Theoretical aspects of the SOM algorithm. *Neurocomputing*, 21(1-3) :119–138, November 1998.
- [4] Marie Cottrell, Madalina Olteanu, Fabrice Rossi, and Nathalie Villa-Vialaneix. Theoretical and Applied Aspects of the Self-Organizing Maps. In *Advances in Self-Organizing Maps and Learning Vector Quantization*, volume 428, pages 3–26. Springer International Publishing, Cham, 2016.
- [5] Antonio R. Damasio. Time-locked multiregional retroactivation : A systems-level proposal for the neural substrates of recall and recognition. *Cognition*, 33(1) :25–62, 1989.
- [6] Gauthier Doquire and Michel Verleysen. A comparison of multivariate mutual information estimators for feature selection. In *ICPRAM*, 2012.
- [7] Gerald M. Edelman. Group selection and phasic reentrant signaling a theory of higher brain function. In *The 4th Intensive Study Program Of The Neurosciences Research Program*, 1982.
- [8] Daniel J. Felleman and David C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. 1991.
- [9] Jérémie Fix and Hervé Frezza-Buet. Look and Feel What and How Recurrent Self-Organizing Maps Learn. In *Proc. WSOM’19*, volume 976, pages 3–12. 2020.
- [10] J.C. Fort. SOM’s mathematics. *Neural Networks*, 19(6-7) :812–816, July 2006.
- [11] Noémie Gonnier, Yann Boniface, and Hervé Frezza-Buet. Consensus Driven Self-Organization : Towards Non Hierarchical Multi-Map Architectures. In *Communications in Computer and Information Science, Neural Information Processing, ICONIP 2020*, pages 526–534. November 2020.
- [12] M. Hagenbuchner, A. Sperduti, and Ah Chung Tsoi. A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks*, 14(3) :491–505, May 2003.
- [13] Barbara Hammer, Alessio Micheli, Alessandro Sperduti, and Marc Strickert. Recursive self-organizing network models. *Neural Networks*, 17(8-9) :1061–1085, October 2004.
- [14] Logan Harriger, Martijn P. van den Heuvel, and Olaf Sporns. Rich club organization of macaque cerebral cortex and its role in network communication. *PLoS ONE*, 7, 2012.

- [15] Lyes Khacef, Laurent Rodriguez, and Benoît Miramond. Brain-inspired self-organization with cellular neuromorphic computing for multimodal unsupervised learning. *Electronics*, 9(10), 2020.
- [16] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1) :59–69, 1982.
- [17] Teuvo Kohonen. Self-organizing maps. In *Springer Series in Information Sciences*, 1995.
- [18] P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *1990 IJCNN International Joint Conference on Neural Networks*, pages 279–284 vol.2, June 1990.
- [19] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6), Jun 2004.
- [20] Stephane Lallee and Peter Dominey. Multi-modal convergence maps : From body schema and self-representation to mental imagery. *Adaptive Behavior*, 21(4) :274–285, 2013.
- [21] Jouko Lampinen and Erkki Oja. Clustering Properties of Hierarchical Self-Organizing Maps. *Journal of Mathematical Imaging and Vision*, page 15, 1992.
- [22] V. Mountcastle. The columnar organization of the neocortex. *Brain*, 120(4) :701–722, April 1997.
- [23] Olivier Ménard and Hervé Frezza-Buet. Model of multi-modal cortical processing : Coherent learning in self-organizing modules. *Neural Networks*, 18(5-6) :646–655, July 2005.
- [24] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27 :379–423, 1948.
- [25] Marc Strickert and Barbara Hammer. Merge som for temporal data. *Neurocomputing*, 64 :39–71, 2005.
- [26] Henri Theil, J. S. Cramer, A. Russchen, and H. Moerman. Economic forecasts and policy ed. 2nd. *Birchandra State Central Library,tripura*, 1961.
- [27] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 :2579–2605, 2008.