

# Sommaire

<b>1 Approche modulaire des réseaux de neurones</b>	<b>1</b>
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
<b>2</b>	<b>Modèle d'architecture CxSOM</b>	<b>13</b>
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 . . . . .	22
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
<b>3</b>	<b>Méthodes de représentation et d'analyse de l'architecture CxSOM</b>	<b>31</b>
3.1	Présentation de l'expérience . . . . .	32
3.2	Représentation des cartes de Kohonen . . . . .	32
3.2.1	Représentations et indicateurs classique des cartes de Kohonen . . . . .	32
3.2.2	Limite de la représentation classique pour CxSOM . . . . .	33
3.3	Formalisme : variables aléatoires . . . . .	34
3.3.1	Représentation des entrées . . . . .	34
3.3.2	Représentation des éléments des cartes . . . . .	35

---

3.4	Représentations graphiques . . . . .	36
3.4.1	Représentation des entrées par rapport au BMU . . . . .	36
3.4.2	Représentation de U par rapport au BMU . . . . .	38
3.4.3	Représentation de l'erreur de quantification dans une carte . . . . .	39
3.5	Information mutuelle comme indicateur statistique . . . . .	39
3.5.1	Information mutuelle et entropie . . . . .	41
3.5.2	Indicateur : Information mutuelle . . . . .	42
3.5.3	Evolution de l'information entre deux cartes . . . . .	43
3.5.4	Discussion . . . . .	44
3.6	Conclusion . . . . .	47
<b>4</b>	<b>Analyse de l'organisation de CxSOM</b>	<b>51</b>
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
<b>5</b>	<b>Application : prédiction d'entrée manquante</b>	<b>63</b>
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
--------------------------	----

<b>Bibliographie</b>	<b>69</b>
----------------------	-----------







## Chapitre 3

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

### Sommaire

---

<b>3.1</b>	<b>Présentation de l'expérience</b>	<b>32</b>
<b>3.2</b>	<b>Représentation des cartes de Kohonen</b>	<b>32</b>
3.2.1	Représentations et indicateurs classique des cartes de Kohonen	32
3.2.2	Limite de la représentation classique pour CxSOM	33
<b>3.3</b>	<b>Formalisme : variables aléatoires</b>	<b>34</b>
3.3.1	Représentation des entrées	34
3.3.2	Représentation des éléments des cartes	35
<b>3.4</b>	<b>Représentations graphiques</b>	<b>36</b>
3.4.1	Représentation des entrées par rapport au BMU	36
3.4.2	Représentation de U par rapport au BMU	38
3.4.3	Représentation de l'erreur de quantification dans une carte	39
<b>3.5</b>	<b>Information mutuelle comme indicateur statistique</b>	<b>39</b>
3.5.1	Information mutuelle et entropie	41
3.5.2	Indicateur : Information mutuelle	42
3.5.3	Evolution de l'information entre deux cartes	43
3.5.4	Discussion	44
<b>3.6</b>	<b>Conclusion</b>	<b>47</b>

---

Dans le chapitre précédent, nous avons proposé l'algorithme CxSOM permettant de construire des architectures non-hierarchiques de cartes auto-organisatrices. Parallèlement à l'objectif d'une SOM, qui consiste à effectuer de la quantification vectorielle sur un espace d'entrée, l'objectif d'une architecture de SOM est de faire de la quantification vectorielle sur plusieurs espaces d'entrées et d'apprendre les relations entre ces entrées. C'est ce qu'on appelle mémoire associative. Le but de cette thèse est d'analyser dans quelle mesure, et de quelle façon un tel apprentissage se déroule grâce au modèle CxSOM.

En utilisant une architecture sur ces entrées, on a deux objectifs. D'abord, on cherche à ce que chaque carte joue un rôle similaire à celui d'une carte de Kohonen classique, par rapport à son entrée externe. Chaque carte prise séparément doit donc avoir appris une représentation de son espace d'entrée, en présentant une continuité entre les prototypes. Ensuite, on souhaite

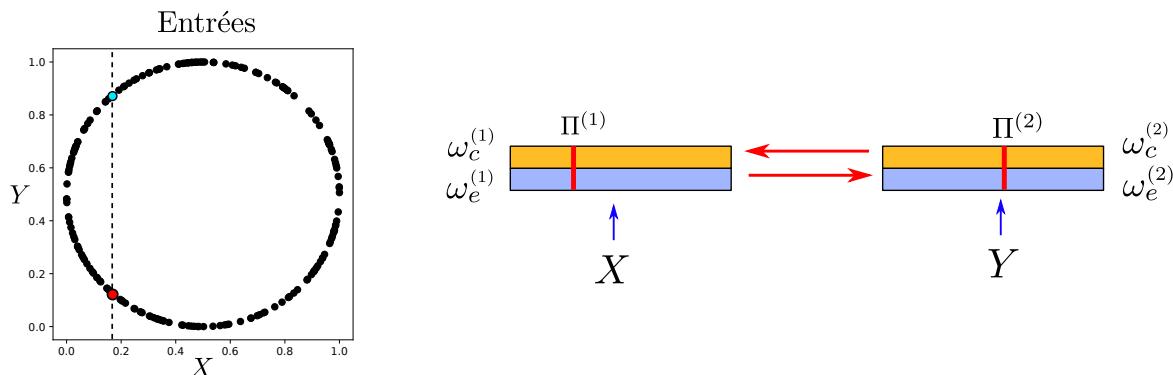


FIGURE 3.1 – Disposition des entrées, sous forme de cercle, à gauche, et architecture de deux cartes en une dimension étudiée et représentée dans ce chapitre.

aussi que le modèle liant les entrées soit appris par l'architecture. Cet aspect est nouveau : nous introduirons des modes de représentation cherchant à définir l'existence d'un apprentissage du modèle, et comment le mesurer.

Ce chapitre a pour but d'étudier l'organisation d'une architecture simple, tout en introduisant les méthodes et les représentations que nous utiliserons par la suite. Nous présenterons la méthode expérimentale sur une architecture de deux cartes, et observerons son comportement grâce aux différents modes de représentations. Nous introduirons également un indicateur caractérisant l'organisation d'une architecture, construit à partir de l'information mutuelle.

### 3.1 Présentation de l'expérience

Nous étudierons dans ce chapitre l'organisation 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 possibles 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. Les données relatives à cette expérience et le code permettant de faire les tracés sont fournies [sur git](#).

### 3.2 Représentation des cartes de Kohonen

#### 3.2.1 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

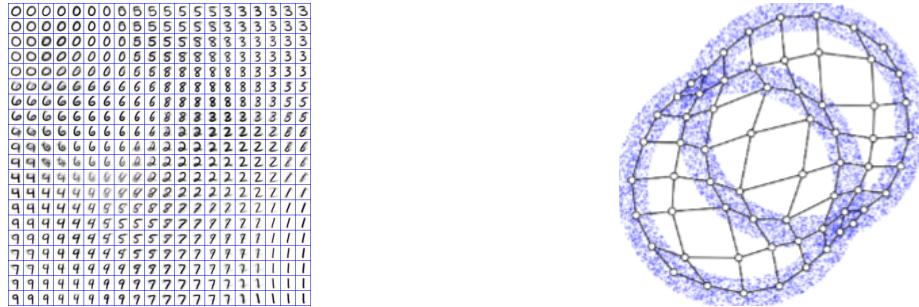


FIGURE 3.2 – Représentations possibles des poids d'une carte de Kohonen classiques, dans le cas d'entrées sous forme d'imagettes ou de points en deux dimensions.

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 3.2.1 dans lequel les poids des prototypes, qui sont des imagettes, sont affichés en chaque point de la grille. Si la dimension d'un poids est trop grande pour être représentée graphiquement, il est également courant d'étiquetter chaque prototype et d'afficher ces étiquettes sur les noeuds de la carte, en tant que représentation.
- 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 3.2.1.



### 3.2.2 Limite de la représentation classique pour CxSOM

Nous pouvons tracer les poids de chaque carte de l'architecture après apprentissage. On considère que les cartes ont atteint la fin de l'apprentissage lorsque les poids ne varient plus que peu. La figure 3.2.2 présente le tracé des poids des deux cartes de l'exemple. Ce tracé permet de conclure que les cartes sont bien dépliées : les poids présentent une continuité dans l'espace d'entrée.



On aimerait mieux comprendre les mécanismes impliqués dans l'apprentissage de ces cartes. Or, le choix du BMU dépend de plusieurs entrées et du processus de relaxation. La représentation des poids seuls ne permet pas de comprendre quels seront les BMUs de chaque carte pour une entrée donnée. Par ailleurs, représenter les cartes une à une laisse de coté leur connexion. Il est donc nécessaire de trouver un moyen de représenter l'architecture comme un tout. La représentation cherchera notamment à montrer comment l'architecture de cartes est capable d'apprendre les relations entre les entrées multimodales.

Enfin, la représentation visuelle d'une cartes d'une architecture est limitée par la dimension des entrées et la dimension des cartes. Dans l'exemple, les entrées et les cartes sont en une dimension, représenter leurs poids est donc réalisable ; en plus grande dimension, il sera nécessaire d'utiliser une représentation telle que celle décrite en figure 3.2.1. Le nombre de connexions contextuelles limitera alors également la lecture d'un tracé. Cette difficulté de représentation soulève la nécessité de définir des valeurs indicatrices du fonctionnement de la carte, calculables en grande dimension.

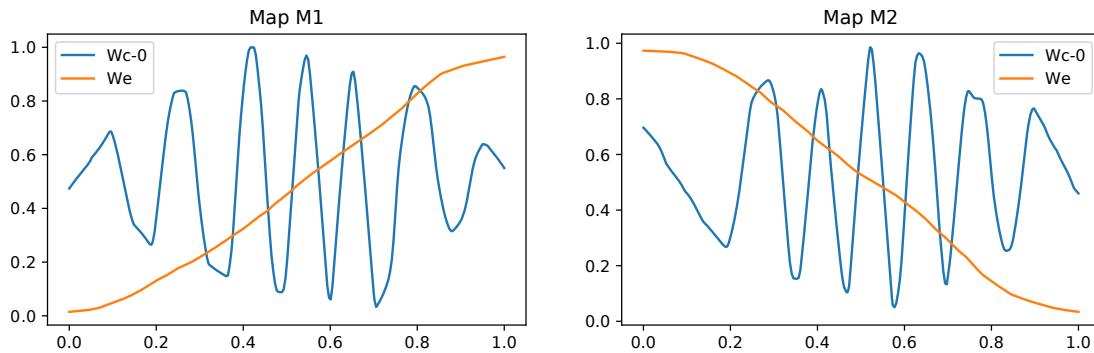


FIGURE 3.3 – Représentation des valeurs des poids d'une carte au sein de CxSOM après apprentissage. La seule représentation de ces poids ne suffit pas à savoir comment la carte se comporte.

Ce chapitre questionne donc la façon de représenter une carte au sein d'une architecture. Nous présenterons en premier lieu un formalisme décrivant les cartes et leurs entrées multimodales associées. 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. Nous utiliserons ces représentations et indicateurs dans les chapitres suivants.

### 3.3 Formalisme : 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.

#### 3.3.1 Représentation des entrées

Les observations multimodales que l'on cherche à apprendre par l'architecture de cartes sont notées  $(X^{(i)}, i = 1 \dots n)$  où  $n$  est le nombre de modalités considérées. On considère les  $X^{(i)}$  comme des variables aléatoires, et le vecteur  $\mathbf{X} = (X^{(1)}, \dots, X^{(N)})$  la variable aléatoire jointe. Cette variable est tirée dans l'espace  $\mathcal{D}^{(1)} \times \dots \times \mathcal{D}^{(N)}$  selon une distribution de probabilité. La distribution de chaque modalité  $X^{(i)}$  sur son espace  $\mathcal{D}^{(i)}$  est appellée 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. On s'intéresse à l'apprentissage de relations entre entrées : les variables  $X^{(i)}$  ne sont pas des variables indépendantes.

On représente cette dépendance en considérant une variable aléatoire cachée  $U$ . Cette variable est multidimensionnelle et 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.

Il s'agit d'une réduction de dimension qui traduit l'existence d'un modèle reliant les observations. Dans le cas d'exemple,  $\mathbf{X} = (X, Y)$ , les coordonnées cartésiennes des points du cercle est

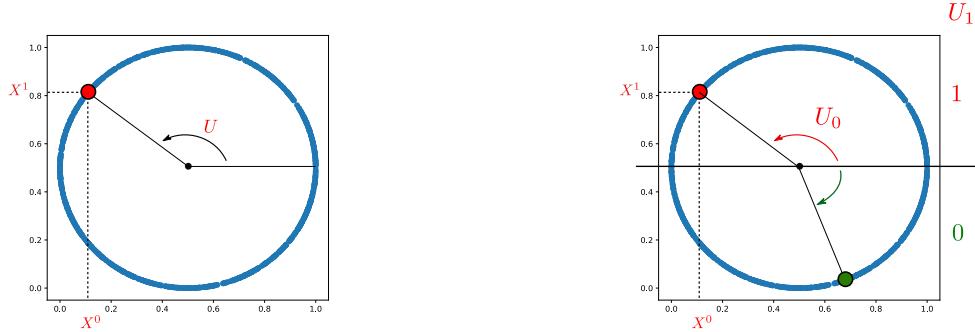


FIGURE 3.4 – Exemples de paramétrisations du cercle. La paramétrisation qui traduit le plus facilement le modèle est naturellement celle dans laquelle  $U$  est à valeurs réelles. Le modèle auxquelles appartiennent les modalités  $X^0$  et  $X^1$  est donc représenté par la variable cachée  $U$ .

alors une vecteur aléatoire, dont les composantes sont les variables aléatoires  $X, Y$ . En définissant une variable  $U$  à valeurs réelles, 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.  $U$  est une variable cachée qui réduit la dimension du modèle. Et contient toute l'information sur l'échantillon.  $U$  et  $f^{(i)}$  ne sont pas uniques. Elle sont choisies en fonction de ce qu'on cherche à traduire dans le modèle. Ainsi, pour le même ensemble de points sous forme de cercle, on pourrait aussi définir une variable  $U$  en deux dimensions, une dimension à valeur réelles paramétrisant un demi cercle, l'autre à valeurs dans 0, 1 indiquant de quel coté de l'axe des abscisses on se situe. Ces paramétrisations sont exprimées en figure 3.3.1. On peut par contre noter que la plus petite dimension qu'on peut choisir pour  $U$  traduit le degré de "dépendance" des entrées. Si les points sont sur une courbe de dimension 1, on peut trouver une variable  $U$  1D; s'il s'agit d'une surface,  $U$  sera 2D, etc. Les exemples donnés sont scalaires, mais cette représentation générale à n'importe quel dimension et nombre d'entrées.

Sur un jeu de données artificielles, on s'appuie sur un modèle connu pour générer les entrées ;  $U$  est donc connu. Sur un jeu de données naturelles, des méthodes de réduction de dimension permettent de trouver une telle variable.

### 3.3.2 Représentation des éléments des cartes

Le jeu de données d'entrée se décompose en jeu d'apprentissage et jeu de tests. Lors de la phase de test, seul le processus de recherche de la best matching unit est réalisé et les poids des cartes ne sont pas mis à jour. Les éléments des cartes peuvent donc, comme les entrées, être considérés comme des variables aléatoires et une itération de test comme une réalisation de celles-ci. La phase de test peut être réalisée après n'importe quelle itération de l'algorithme d'apprentissage. Le processus d'apprentissage et de tests est décrit en figure 3.5.

Nous considérerons plusieurs éléments des cartes ; qui nous intéressent particulièrement dans les représentations, en tant que variables aléatoires :

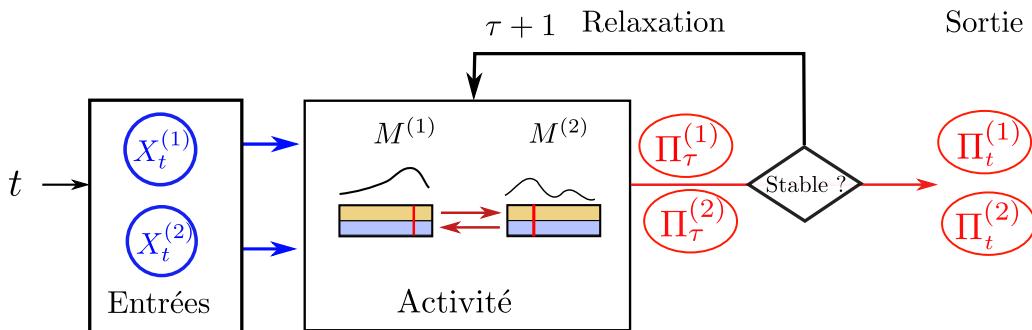


FIGURE 3.5 – Schéma des tests

- Les positions des BMUs  $\Pi^{(0)}, \dots, \Pi^{(N)}$  dans chaque carte
- Les poids externes des BMUs  $\omega_e^{(0)}(\Pi^{(0)}), \dots, \omega_e^{(N)}(\Pi^{(N)})$

Mais tout autre élément d'une carte peut être vu de cette manière, telles que les activités.

Une phase de test est alors un ensemble de réalisations d'une variable aléatoire jointe :

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

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.

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.

### 3.4 Représentations graphiques

La représentation classique d'une carte, qui est de tracer les poids par rapport à la position des unités de la carte, permet de visualiser la continuité des poids de la carte. Si on veut représenter quelle unités seront effectivement BMU et lesquelles ne le seront jamais, on y ajoute le tracé, sur un ensemble de tests assez grand, **des poids des BMU en fonction des BMUs**. Ces points se superposent à la courbe de poids, mais certaines unités ne seront pas représentées : on les appelle *unités mortes*. La position de la best matching unit  $\Pi$  étant la réponse de la carte à une entrée, il s'agit d'un tracé d'un élément de la carte en fonction de la réponse de celle-ci. **Cette façon de représenter les réponses d'une carte rappelle directement la façon de représenter les neurones en biologie pour réaliser des cartes du cortex.** Dans ces représentations, la valeur d'entrée faisant réagir un neurone est tracé en fonction de sa position dans le cortex.

Le poids de la BMU n'est pas le seul élément qu'on peut tracer en fonction de la réponse  $\Pi$ . Nous présentons dans cette partie les choix de tracés que nous utiliserons dans cette thèse.

#### 3.4.1 Représentation des entrées par rapport au BMU

##### Méthode

En première représentation, nous tracerons la valeur de l'entrée  $X^{(i)}$  d'une carte par rapport à la position du BMU  $\Pi^{(i)}$ . Cette représentation permet d'analyser la quantification des entrées

par la carte. Ces tracés sont réalisables pour des cartes une et deux dimensions, et pour des entrées quelconques, que ce soient des réels ou des entrées de plus grande dimension comme des images. Pour mieux comprendre les relations entre entrées et plusieurs cartes, on tracera sur une même figure les entrées de toutes les cartes selon la position du BMU d'une des cartes. Afin de comparer le comportement d'une carte de CxSOM avec une carte de Kohonen classique, on réalise le même tracé d'entrées par rapport au BMU, sur une carte qui aura appris indépendamment les mêmes valeurs de  $X^{(1)}$  que  $M^{(1)}$ , sans avoir de connexions avec  $M^{(2)}$ . Cette dernière carte a donc seulement une couche de poids externes, il s'agit d'une carte de Kohonen classique.

On s'attend à ce que les entrées externes soient proches de la courbe de poids externes. Cela indiquera que le poids externe  $\omega_e^{(1)}(\Pi^{(1)})$  est une approximation de  $X^{(1)}$  et qu'une carte réalise de la quantification vectorielle sur ses entrées externes, malgré les connexions à l'autre carte. On attend également que la présence de la connexion avec  $M^{(2)}$ , par l'entrée contextuelle, influence l'organisation de  $M^{(1)}$ ; et, si possible, qu'un modèle de connexions d'entrée est appris.

Ce tracé est présenté en figure 3.6. On représente la valeur des entrées  $X^{(1)} = x$ , en orange, et  $X^{(2)} = y$ , en bleu, par rapport à la position du BMU dans la carte  $M^{(1)}$ ,  $\Pi^{(1)}$ . Ce tracé sur l'architecture de deux cartes (c) est comparé à ce qu'on obtient si les valeurs de  $x$  étaient présentées à une carte classique (b). Les poids externes sont représentés en bleu, les poids contextuels en orange. Deux échantillons du schéma des entrées (a) sont mis en valeur, en bleu et rouge. Ces deux points ont la même valeur pour  $x$ , mais deux valeurs différentes pour  $y$ . Ces échantillons sont reportés en couleur sur les deux tracés (b) et (c).

## Résultats

Analysons d'abord la forme des poids contextuels de  $M^{(1)}$ , que nous avions tracés en figure 3.2.2 sans les commenter. Les poids externes, en orange, présentent une disposition similaire à ceux observés dans la carte classique (b). Les poids contextuels, en bleu, présentent une forme de vagues, avec 7 valeurs de maximum allant de 0.5 à 1, et 6 minimum allant de 0.5 à 0.1. Ces maximum et minimum sont répartis en zones de taille équivalente sur la carte.

Lorsqu'on s'intéresse au tracés des échantillons, on remarque d'abord que les positions dans la carte  $M^{(1)}$  se répartissent en zones étant BMUs et zones mortes, dans lesquelles aucune entrée n'a gagné. C'est une première différence avec la carte indépendante, pour laquelle toutes les positions gagneront pour des entrées. Les zones dans lesquelles il y a des BMUs correspondent aux extrémum des poids contextuels et leurs alentours. C'est un phénomène inhabituel pour une carte de Kohonen. Les entrées  $X^{(1)}$ , dans la carte classique (b), correspondent à la courbe de poids externe : la valeur du poids du gagnant est toujours très proche de la valeur de l'entrée. Dans la carte  $M^{(1)}$ , les entrées externes  $X^{(1)}$  orange sont proches de la courbe de poids externes, mais avec plus d'erreur de quantification. Les deux points rouge et bleu ayant la même valeur de  $x$  ont un BMU différent dans la carte  $M^{(1)}$ , alors que ces deux échantillons ont le même BMU dans la carte apprenant indépendamment sur les valeurs de  $x$ . Ainsi, la carte connectée au sein de CxSOM différencie les échantillons en fonction de non seulement leur entrée externe, mais aussi de l'entrée de l'autre carte de l'architecture. La plage de valeurs des  $X^{(1)}$  gagnant dans un des zones recoupe les plages de valeurs gagnant dans les zones situées à gauche et à droite. Par exemple, la zone dans laquelle l'échantillon rouge gagne, autour de  $\Pi^{(1)} = 0.25$ . La partie des entrées située en dessous de la courbe de poids externe recoupe les valeurs d'entrées gagnant dans la zone précédente ; la partie située au dessous de la courbe de poids externe recoupe des valeurs gagnant dans la partie suivante. Pour une entrée externe, le choix de la zone de BMU

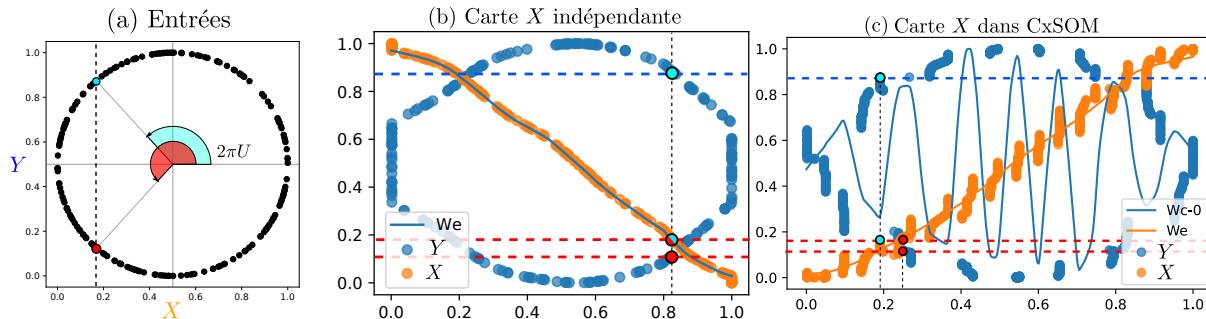


FIGURE 3.6 – Représentation des entrées  $X, Y$  d'une architecture de deux cartes relativement au BMU de la carte  $X$  après apprentissage. Ces tracés mettent en valeur l'organisation des cartes, différentes dans le cas où les cartes apprennent indépendamment leurs entrées (b) ou sont connectées (c). Les entrées correspondantes sont en figure (a). Les points bleu et rouge reportés sur les tracés correspondent au même échantillon de test.

dans laquelle elle gagnera dépend alors de l'entrée contextuelle.

## Conclusion

Dans la carte  $M^{(1)}$ , une unité se spécialise donc par rapport aux deux entrées et non pas une seule comme dans la carte indépendante : les entrées externes et l'entrée contextuelles. C'est bien ce à quoi on s'attendait en ayant deux couches de poids. Ce qui est intéressant est que cette différenciation est réalisée par la répartition des unités en un nombre fini de zones distinctes. Dans chaque zone, les unités sont BMUs pour un segment de valeurs d'entrée externe et contextuelles. Au sein d'une zone, la répartition des entrées externe selon le BMUs est ordonnée, comme ce serait le cas dans une carte auto-organisatrice classique. Le comportement de la carte au sein d'une zone reste donc similaire à celui d'une carte classique.

Deux zones adjacentes correspondent par ailleurs à des segments de valeur d'entrée en partie superposés, et des segments de valeurs d'entrées contextuelles différentes. Il s'agit d'une deuxième échelle d'organisation, qui garde également l'aspect ordonné d'une carte classique. Ces zones sont créées par auto-organisation ; aucun paramètre de la carte n'a été modifié pendant l'apprentissage pour former ces zones, et le nombre d'unités allouées par auto-organisation dans chaque zone est à peu près égal. La carte agit un peu comme une base de données structurée avec des indices primaires et des indices secondaires pour chaque neurone, l'indice primaire étant la zone de la carte, et l'indice secondaire la position dans cette zone.

### 3.4.2 Représentation de $U$ par rapport au BMU

#### Méthode

Dans le cas d'une architecture de deux cartes, il était facile de tracer les deux entrées sur un même graphique. Lorsqu'on augmente la dimension, il est pertinent de dégager des nouvelles valeurs qui représentent le modèle en plus faible dimension : il s'agit ici de la variable  $U$ . Cette variable cachée est en fait une représentation du modèle en dimension plus faible. On peut tracer  $U$  en fonction de la position  $\Pi$  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 : il s'agit de l'angle correspondant à la position d'un point  $x, y$  sur le cercle. On tracera donc  $U$  en fonction de  $\Pi^{(1)}$  et  $U$  en fonction de  $\Pi^{(2)}$ . Nous avons observé sur les tracés en figure 3.6 que chaque carte allouent leurs unités en fonction des deux entrées. En tracant  $U$  en fonction de la position du BMU, on retrouvera d'abord les zones de la cartes.

## Résultats

Le tracé en figure 3.7 montre  $U$  comme une fonction de la position du BMU dans chaque carte, contrairement au cas où les cartes ne sont pas connectées. En effet, dans ce dernier cas, il y a plusieurs valeurs de  $U$  pour un même  $\Pi$ , une valeur  $x$  de  $X^{(1)}$  correspondant à deux positions sur le cercle. L'organisation en zones rend une position  $\Pi$  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  $\Pi$ , ce qui est représenté en pointillé sur la figure 3.7. Donc, chaque carte  $M^{(1)}$  et  $M^{(2)}$  code pour toute l'architecture, par les deux couches de poids, et non plus seulement son entrée.

La forme générale du tracé de  $U$  selon  $\Pi^{(i)}$  dans chaque carte est proche du tracé obtenu dans le cas des cartes indépendantes. Une carte de l'architecture se comporte donc comme une carte classique, dans laquelle les connexions contextuelles viennent moduler la réponse. La réponse d'une carte garde une continuité par rapport à  $U$ .

## Conclusion

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.

### 3.4.3 Représentation de l'erreur de quantification dans une carte

Afin de mesurer la qualité de la quantification vectorielle au sein d'une carte dans CxSOM, nous tracerons, pour un échantillon test, le poids externe du BMU  $\omega_e(\Pi^{(i)})$  en fonction de l'entrée présentée  $X^{(i)}$ . Si une carte a appris bonne représentation correcte de ses entrées externes, la forme de ce tracé doit être proche de l'identité.

En figure 3.8, nous avons tracé le poids du BMU de chaque carte en fonction de l'entrée qui lui a été présentée, dans l'expérience à deux cartes. Ces tracés s'approchent de l'identité : la quantification des entrées est correctement réalisée. On observe une erreur plus grande que dans le cas d'une carte isolée qui aurait le même nombre de neurones, c'est à dire 500 ici. Dans ce dernier cas, les 500 neurones permettent aux poids de se disposer densément entre 0 et 1..

## 3.5 Information mutuelle comme indicateur statistique

L'étude de tout processus physique s'effectue par un ensemble signaux issus de capteurs. La théorie de l'information de Shannon [23] apporte un modèle mathématique qui abstrait ces signaux et 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. Ce modèle mathématique puissant permet de s'abstraire de la nature des signaux pour s'intéresser à leurs relations. Comme son nom l'indique, la théorie de l'information s'appuie sur la notion fondamentale d'information portée par un symbole. Ensuite, cette information se décline en quantités qu'on calcule

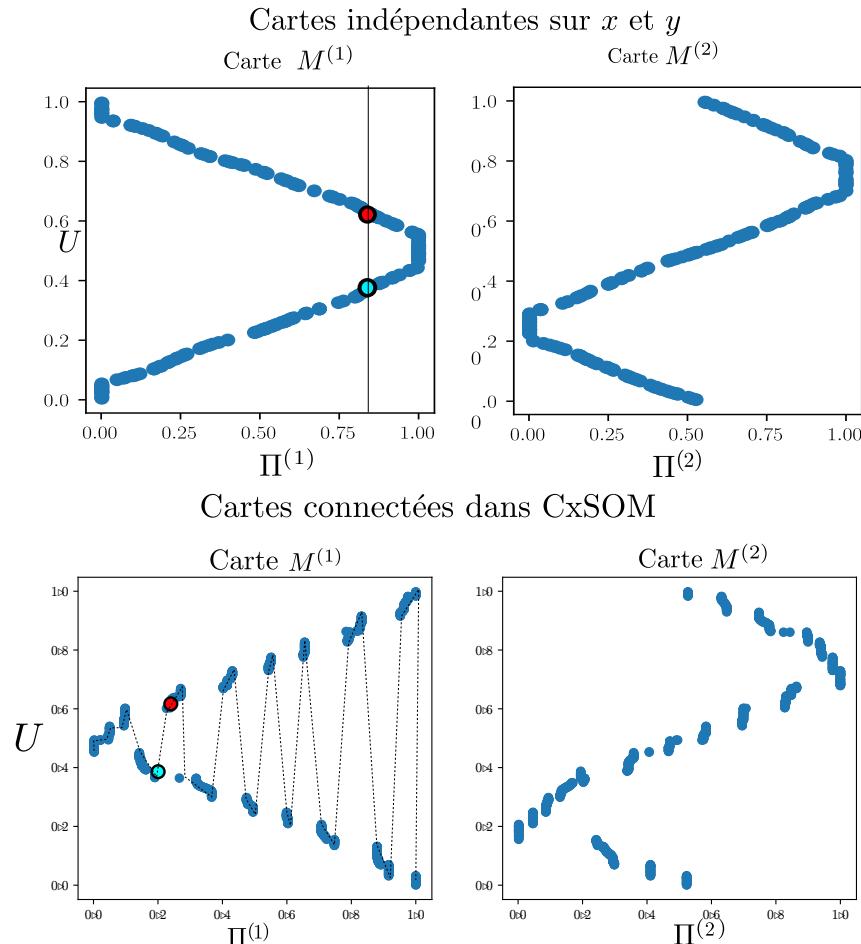


FIGURE 3.7 – Valeur de  $U$  en fonction des valeurs du BMU  $\Pi^{(i)}$  dans chacune des cartes, pour des entrées prises sur le cercle. Sur la première ligne, nous traçons la réponse de chaque carte à son entrée dans le cas où les cartes ne sont pas connectées. Sur la deuxième ligne, nous traçons la 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.

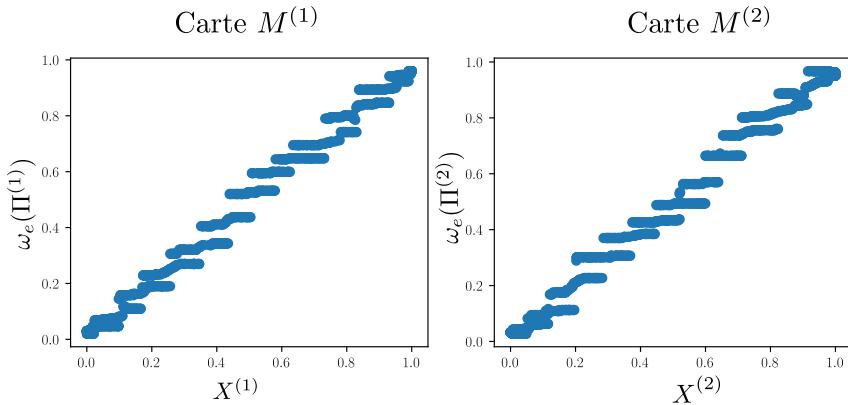


FIGURE 3.8 – Poids du BMU dans chaque carte en fonction de l’entrée présentée. On s’attend à des tracés proches de l’identité, montrant que le poids du BMU d’une carte est une bonne représentation de l’entrée. Sur ce graphique, on se rapproche effectivement de la fonction identité, cependant, une faible erreur est observée. On observe également un découpage des poids en bandes.

en fonction de ce qu’on veut mesurer : l’entropie d’une variable, comme l’information apportée par l’observation de la variable seule ; l’entropie conditionnelle entre deux variables, l’information mutuelle entre deux variables ou un plus grand nombre. Ces mesures définissent une dépendance statistique générale, et ne dépendent pas du type de modèle ou de relation.

Nous investiguerons dans cette partie comment quantifier l’apprentissage de l’architecture de cartes par des outils d’information. Bien que cette théorie soit un outil mathématique puissant, il s’agit d’un modèle s’appuyant sur les probabilités. L’estimation à partir de données est donc un élément clé et parfois limitant lorsqu’on cherche à utiliser des valeurs telles que l’entropie pour quantifier l’information au sein d’un système. Nous définirons donc dans cette partie des quantités à mesurer dans l’architecture de cartes, et chercherons à l’estimer.

### 3.5.1 Information mutuelle et entropie

Les notions d’*entropie* et les valeurs qui en sont dérivées, 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. L’entropie de Shannon d’une variable aléatoire  $X$  à valeurs discrètes dans un ensemble  $E_X$ , de distribution  $p(X)$ , 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*.



### 3.5.2 Indicateur : Information mutuelle

Lors de l'analyse de CxSOM, on s'interroge sur l'information que portent les positions des BMUs  $\Pi$  d'une carte sur le modèle d'entrées. Les éléments de la carte ont été définis en terme de variables aléatoires ; on peut donc utiliser l'information mutuelle comme une représentation de l'information portée par le BMU d'une carte sur **le modèle**. Le modèle est représenté par la variable  $(X, Y, Z)$ , mais aussi par  $U$ .  $I(\Pi, U)$  est alors l'information moyenne que le BMU d'une carte porte sur  $U$ , donc sur le modèle, et  $U$  sur le BMU. On souhaite cependant avoir un indicateur normalisé, qui permettrait, sur une échelle de 0 à 1, de quantifier à quel point un BMU porte de l'information sur  $U$ . On va donc normaliser 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  $\Pi$ . En effet, par construction,  $\Pi$  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  $\Pi$  sont en bijection, autrement dit, si et seulement si  $U$  est aussi une fonction de  $\Pi$ .

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  $\Pi$ , ou  **$U$  de Theil** [25].

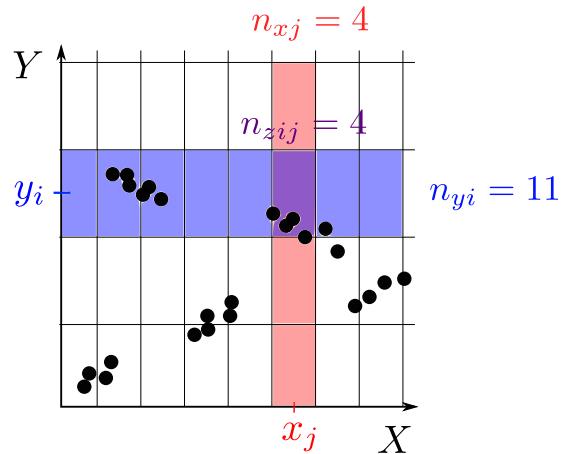


FIGURE 3.9 – Méthode par histogrammes pour estimer les distributions des variables  $X$  et  $Y$ . Les distributions sont estimées à partir de  $n_{xj}$ ,  $n_{yi}$  et  $n_{zij}$ , puis les valeurs de l'entropie  $H$  et l'information mutuelle  $I$  calculées.

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'archictecture a appris un modèle, on souhaite que  $I_x(U|\Pi^{(1)}, \dots, \Pi^{(k)})$  soit le plus proche possible de 1.

### 3.5.3 Evolution de l'information entre deux cartes

#### 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 estimons la distribution des variables  $X, Y$  et leur distribution jointe  $Z = (X, Y)$  en discrétilisant l'espace par la méthode des *histogrammes*, représenté en figure 3.9. Les variables  $X$  et  $Y$  sont donc discrétilisées en *boîtes* de centres  $x_k$  et  $y_k$ . La distribution de  $X$  est alors estimée par :

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

, 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  $Y$  et  $Z = (X, Y)$ . La précision de l'estimation peut être améliorée en choisissant des tailles de boîtes variables ; nous utilisons ici la méthode simple avec des boîtes de taille fixe. L'information mutuelle et l'entropie sont ensuite estimées à partir de ces distributions discrètes, par leur formules :

$$I_x(X, Y) = \sum_{i=0}^{n_x} \sum_{j=0}^{n_y} P(x_i, y_j) \log \left( \frac{P(x_i, y_j)}{P(x_i)P(y_j)} \right) \quad (3.6)$$



### Information mutuelle sur deux cartes

Analysons à présent comment l'information mutuelle évolue 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  $\text{test}$  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 en figure 3.10, sont la moyenne, à chaque pas de temps, de l'information mutuelle de chaque expérience au pas de temps  $t$ . On représente donc l'évolution de l'information mutuelle en moyenne.

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  $\Pi$  dans chaque carte.

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

L'observation du tracé montre qu'en effet, les informations mutuelles  $I_x(U|\Pi^{(1)})$  et  $I_x(U|\Pi^{(2)})$  sont toutes deux plus élevées à chaque moment de l'apprentissage que dans le cas où les cartes sont séparées. C'est également vrai au tout début de l'apprentissage : en effet, les BMUs étant calculés selon les poids contextuels et les poids externes

Cette valeur augmente rapidement et se stabilise autour de 0.8. Cette stabilisation correspond à la stabilisation des poids de l'architecture.

#### 3.5.4 Discussion

##### Influence de l'estimation

Lorsque la dimension augmente, le nombre d'échantillon disponibles 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 pu contenir selon leur distribution ; l'estimation de la probabilité en ce point sera donc nulle, et l'estimation faussée.

Nous avons donc envisagé d'autres estimateurs moins biaisés en grande dimension. Détailons par exemple l'estimateur par KNN (K-nearest neighbors) de Kraskov [18]. 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

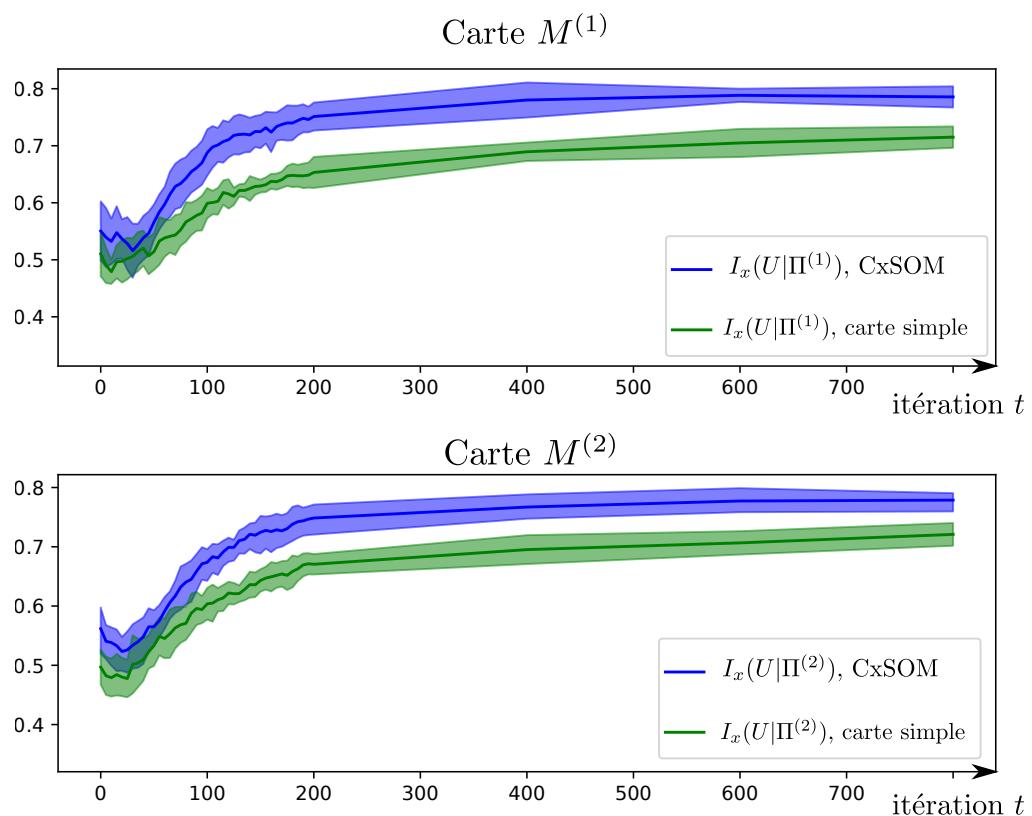


FIGURE 3.10 – Evolution du coefficient d'incertitude dans chaque carte au long de l'apprentissage. La courbe bleue correspond à  $I_x(U|\Pi)$  dans l'architecture de cartes  $M^{(1)}$  et  $M^{(2)}$ . On compare cette évolution à l'évolution de l'information d'une seule carte apprenant sur les mêmes entrées  $X$  ou  $Y$ , sans être connectée.



dans cette zone de l'espace, suivant une formule permettant d'approximer les différences de logarithme par la fonction digamma  $\psi$  :

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

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 3.12, 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 3.13. 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 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 3.8. 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.

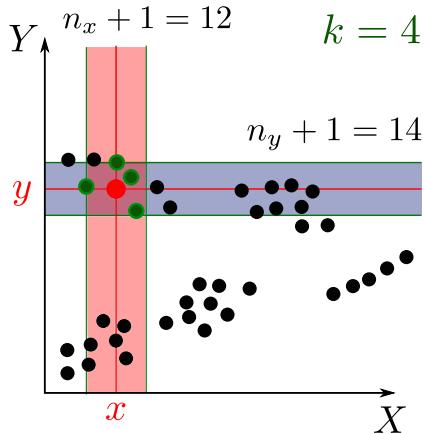


FIGURE 3.11 – Découpage en KNN de Kraskov pour estimer l’entropie et l’information mutuelle des variables  $X$  et  $Y$ . Les plus proches voisins du point rouge sont trouvés, en vert, et le processus est répété sur tous les points. Les valeurs de  $n_x$  et  $n_y$  permettent d’estimer directement l’entropie.

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 discréétisation 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 [5] 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.

## 3.6 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

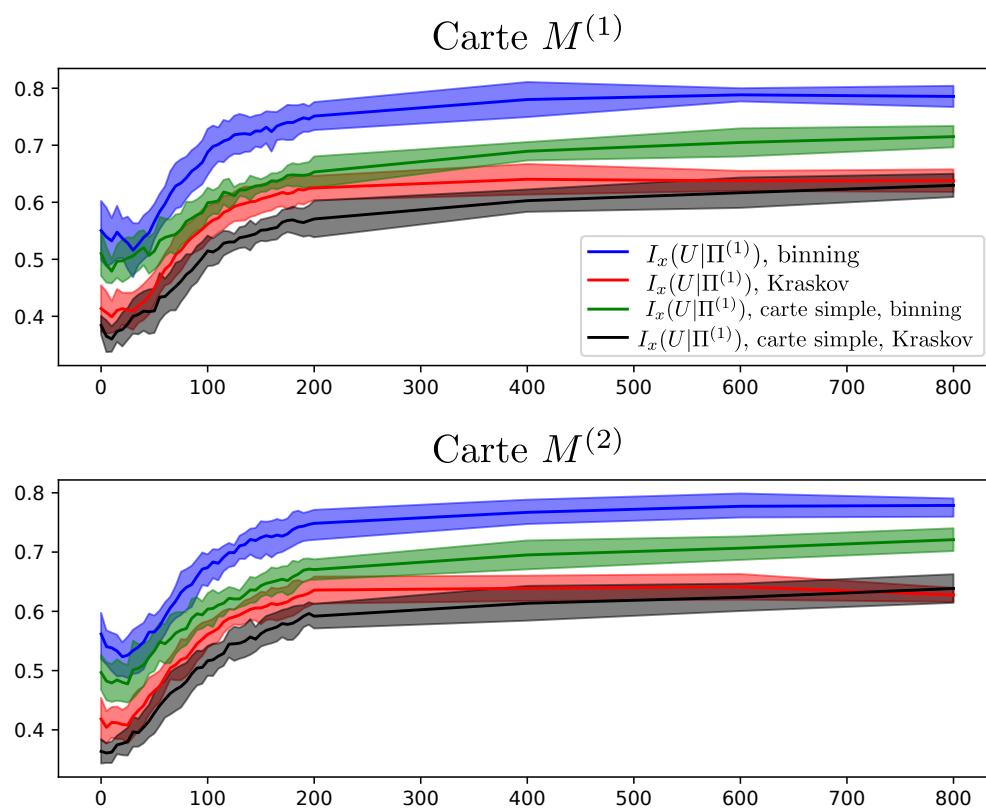


FIGURE 3.12 – Evolution du coefficient d'incertitude dans chaque carte au long de l'apprentissage, en comparant l'estimation par histogrammes et l'estimation par la méthode de Kraskov.

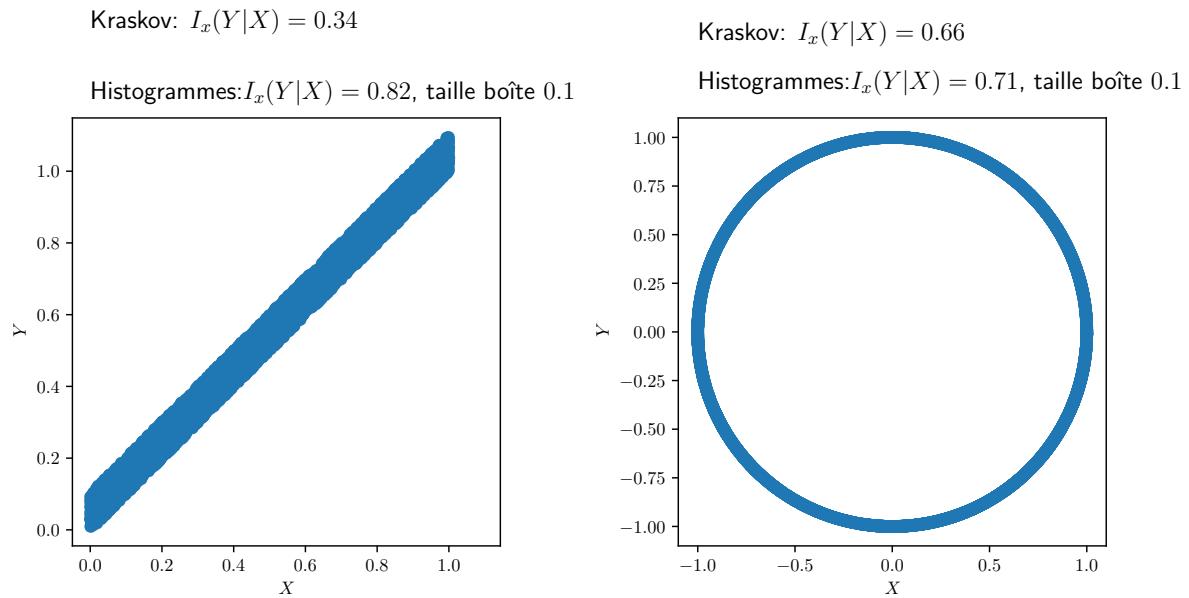


FIGURE 3.13 – Comparaison du calcul de l’indicateur  $I_x$  sur deux distributions. À gauche, la relation entre  $Y$  et  $X$  se rapproche d’une fonction, mais bruitée. À droite, la relation n’est pas fonctionnelle, mais de telle sorte qu’une valeur de  $X$  correspond au maximum à deux valeurs de  $Y$ . 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.

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] M. Cottrell, J.C. Fort, and G. Pagès. Theoretical aspects of the SOM algorithm. *Neurocomputing*, 21(1-3) :119–138, November 1998.
- [3] 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.
- [4] 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.
- [5] Gauthier Doquire and Michel Verleysen. A comparison of multivariate mutual information estimators for feature selection. In *ICPRAM*, 2012.
- [6] 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.
- [7] Daniel J. Felleman and David C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. 1991.
- [8] 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.
- [9] J.C. Fort. SOM’s mathematics. *Neural Networks*, 19(6-7) :812–816, July 2006.
- [10] 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.
- [11] 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.
- [12] Barbara Hammer, Alessio Micheli, Alessandro Sperduti, and Marc Strickert. Recursive self-organizing network models. *Neural Networks*, 17(8-9) :1061–1085, October 2004.
- [13] 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.
- [14] Lyes Khacef, Laurent Rodriguez, and Benoît Miramond. Brain-inspired self-organization with cellular neuromorphic computing for multimodal unsupervised learning. *Electronics*, 9(10), 2020.

- [15] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1) :59–69, 1982.
- [16] Teuvo Kohonen. Self-organizing maps. In *Springer Series in Information Sciences*, 1995.
- [17] 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.
- [18] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6), Jun 2004.
- [19] 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.
- [20] Jouko Lampinen and Erkki Oja. Clustering Properties of Hierarchical Self-Organizing Maps. *Journal of Mathematical Imaging and Vision*, page 15, 1992.
- [21] V. Mountcastle. The columnar organization of the neocortex. *Brain*, 120(4) :701–722, April 1997.
- [22] 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.
- [23] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27 :379–423, 1948.
- [24] Marc Strickert and Barbara Hammer. Merge som for temporal data. *Neurocomputing*, 64 :39–71, 2005.
- [25] Henri Theil, J. S. Cramer, A. Russchen, and H. Moerman. Economic forecasts and policy ed. 2nd. *Birchandra State Central Library,tripura*, 1961.