

17 juin 2022

Sommaire

I Approche modulaire pour construire des réseaux de neurones bio-inspirés 1

1	Approche modulaire des réseaux de neurones	3
1.1	Systèmes complexes et émergence de calcul	4
1.2	Inspiration biologique	5
1.3	Mémoire associative et multi-modalité	5
1.4	Les cartes auto-organisatrices, un modèle d'apprentissage inspiré de la biologie	5
1.4.1	Carte de Kohonen classique	5
1.4.2	Aspect topologique de la carte de Kohonen	6
1.4.3	Inspiration Biologique d'une carte de Kohonen	7
1.5	Les cartes de Kohonen comme modules d'une architecture d'apprentissage complexe ?	10
1.5.1	Inspiration biologique	10
1.5.2	Systèmes autonomes de cartes auto-organisatrices	10
1.5.3	Mémoire associative et SOM	11
1.5.4	Notion de mémoires	11
1.5.5	Association de réseaux de neurones	11
1.5.6	Calcul décentralisé?	11
6	Architectures de cartes auto-organisatrices	12
6.1	Modèles de SOM impulsionales	12
6.2	The hierarchical self organizing map	13

6.3	Modèle CDZ de Dominey	13
6.4	A-SOM	14
7	Cartes auto-organisatrices récurrentes	14
7.1	Cartes récurrentes	15
7.2	The Hypermap architecture, TKM, RSOM	15
7.3	MSOM	15
 I Un nouveau modèle multi-cartes : CxSOM et ses méthodes d'analyse		 1
1 Modèle d'architecture CxSOM		3
1.1	Carte de Kohonen classique	4
1.1.1	Algorithme et notations	4
1.1.2	Paramétrage d'une carte de Kohonen	5
1.2	Motivations du modèle CxSOM	7
1.2.1	Champ d'application : mémoire associative	8
1.2.2	Description de l'architecture	8
1.3	Présentation de CxSOM : exemple d'une architecture de deux cartes	10
1.3.1	Détail des étapes	10
1.3.2	Résumé	15
1.4	Formalisation : cas pour n cartes	15
1.4.1	Entrées et calcul d'activité	15
1.4.2	Calcul du BMU par relaxation	16
1.4.3	Mise à jour des poids	17
1.5	Choix des paramètres	18
1.5.1	Paramétrage d'une carte	18
1.5.2	Paramètres de l'architecture	19
1.6	Conclusion	19
2 Méthodes de représentation et d'analyse de l'architecture CxSOM		21
2.1	Introduction	21

2.1.1	Motivations	21
2.1.2	Présentation d'une expérience multimodale minimale illustrant les représentations	22
2.1.3	Représentations et indicateurs classiques des cartes de Kohonen	23
2.1.4	Limites des représentations classiques dans le cas d'une architecture CxSOM	24
2.1.5	Problématique	24
2.2	Formalisation statistique des entrées et sorties des cartes	25
2.2.1	Formalisation des entrées	25
2.2.2	Formalisation du modèle d'entrée par une variable cachée	25
2.2.3	Démarche expérimentale	27
2.3	Représentations graphiques	27
2.3.1	Erreur de quantification d'une modalité dans chaque carte	28
2.3.2	Représentation cartographique des valeurs d'entrées préférentielles des BMUs	28
2.3.3	Représentation de la variable cachée selon les positions des BMUs	31
2.3.4	Dépliement d'une carte dans l'espace d'entrée multimodal	31
2.4	Conclusion	33
3	Un nouvel indicateur de l'apprentissage des données multimodales basé sur l'information mutuelle	35
3.1	Rappel des éléments de la théorie de l'information	36
3.2	Indicateur : information mutuelle normalisée	37
3.2.1	Définition de l'indicateur	37
3.2.2	Estimation de l'indicateur	38
3.3	Application de l'indicateur au cas d'exemple multimodal simple	41
3.3.1	Evolution de l'indicateur normalisé dans une architecture de cartes au cours de l'apprentissage	41
3.3.2	Comparaison des deux méthodes d'estimation	41
3.4	Conclusion	41

II Expériences de validation et application	35
3 Analyse de l'organisation de CxSOM	37
3.1 Identification des comportements de blocs élémentaires de cartes	37
3.1.1 Auto-organisation émergeant d'architectures de deux et trois cartes	38
3.1.2 Conclusion	39
3.2 Analyse de la relaxation	41
3.2.1 Etude de la convergence des BMUs lors de la relaxation	41
3.2.2 Formalisation de l'algorithme de relaxation	44
3.2.3 Etude de l'influence de l'entrée contextuelle sur le BMU	45
3.2.4 Etude de l'unicité du point fixe	46
3.2.5 Influence du pas de convergence	47
3.2.6 Discussion	47
3.2.7 Conclusion	50
4 Application : prédiction d'entrée manquante	53
4.1 Prédiction d'entrées géométriques	54
4.1.1 Algorithme de prédiction	54
4.1.2 Résultats	55
4.1.3 Discussion	55
4.2 Application à la commande de drône en vol	55
4.2.1 Méthode expérimentale	55
4.2.2 Résultats	56
4.2.3 Discussion	56
4.3 Conclusion	56
Bibliographie	83

Chapitre 3

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

Sommaire

2.1	Introduction	21
2.1.1	Motivations	21
2.1.2	Présentation d'une expérience multimodale minimale illustrant les représentations	22
2.1.3	Représentations et indicateurs classiques des cartes de Kohonen	23
2.1.4	Limites des représentations classiques dans le cas d'une architecture Cx-SOM	24
2.1.5	Problématique	24
2.2	Formalisation statistique des entrées et sorties des cartes	25
2.2.1	Formalisation des entrées	25
2.2.2	Formalisation du modèle d'entrée par une variable cachée	25
2.2.3	Démarche expérimentale	27
2.3	Représentations graphiques	27
2.3.1	Erreur de quantification d'une modalité dans chaque carte	28
2.3.2	Représentation cartographique des valeurs d'entrées préférentielles des BMUs	28
2.3.3	Représentation de la variable cachée selon les positions des BMUs	31
2.3.4	Dépliement d'une carte dans l'espace d'entrée multimodal	31
2.4	Conclusion	33

3.1 Introduction

3.1.1 Motivations

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. Chaque carte a pour but d'extraire une représentation de ses entrées externes, tout en prenant comme entrée secondaire les positions des Best

Matching Unit d'autres cartes afin de mettre en relation les activités relatives aux différentes modalités. 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 à terme 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 particulier 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 au sein de chaque carte d'un espace d'entrée. 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 un système complexe, même dans une architecture de quelques cartes. Chaque carte possède 500 unités ; son état, représenté par son BMU, peut alors prendre 500 valeurs, et l'état d'une carte dépend des cartes voisines.

Cette thèse s'inscrit alors dans une démarche complètement expérimentale d'étude d'un système complexe : nous observerons l'organisation d'architectures de cartes sur différents espaces d'entrées et différents configurations et observerons comment l'organisation qui en émerge traduit l'apparition d'un phénomène de mémoire associative. Pour analyser l'organisation de ces cartes, nous aurons besoin d'introduire de nouvelles représentations par rapport à l'étude d'une SOM classique. Nous voulons également poser un formalisme clair sur des architectures de quelques cartes pour permettre l'adaptation de CxSOM à plus grande échelle.

Nous posons ainsi dans ce chapitre la méthode expérimentale que nous utiliserons dans toutes les expériences présentées dans ce manuscrit. Nous y présenterons les représentations adaptées à cette méthode expérimentale ainsi que le formalisme utilisé. Ces représentations ont pour but non seulement de qualifier la qualité de l'apprentissage mais surtout de mettre en lumière les propriétés et l'organisation des cartes émergeant de l'algorithme d'apprentissage.

3.1.2 Présentation d'une expérience multimodale minimale illustrant les représentations

La méthode expérimentale sera présentée dans toute ce chapitre sur l'exemple minimal 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 , seules 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 points sont donc sur un cercle de centre $x_c, y_c = 0.5, 0.5$ et de rayon 0.5. 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 1D classiques en tant que témoin. Ces cartes prennent en entrée les valeurs x , et la deuxième les valeurs y , mais sans être connectées entre elles. Il s'agit de cartes classiques, prenant une entrée. Les paramètres de ces cartes sont les mêmes que les cartes de CxSOM : 500 noeuds, de rayon de voisinage constant $h_e = 0.2$.

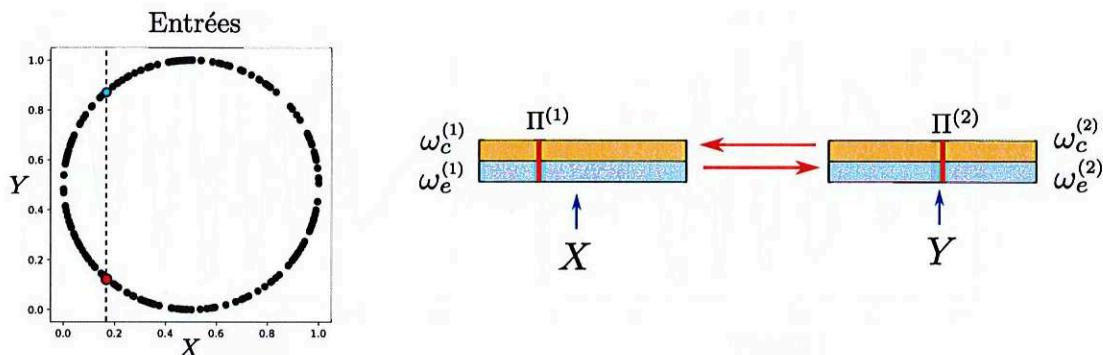


FIGURE 3.1 – A gauche, disposition des entrées dans l'exemple illustratif, sous forme de cercle. A droite, l'architecture de deux cartes en une dimension utilisée sur ces entrées dans ce chapitre pour illustrer les méthodes de représentation.

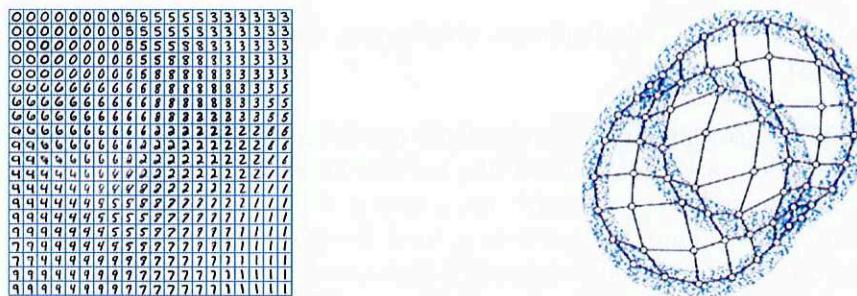


FIGURE 3.2 – Représentations possibles des poids d'une carte de Kohonen classiques, dans le cas d'entrées sous forme d'imagettes ou de points en deux dimensions.

3.1.3 Représentations et indicateurs classiques 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. 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 reliés en fonction des positions des nœuds 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 pour une grille en deux dimensions.

Nous utiliserons des représentations adaptées au cas d'une architecture à plusieurs cartes à partir de ces deux modes de représentation. Cette adaptation dans le cas précis d'une architecture CxSOM est l'objet de cette section.

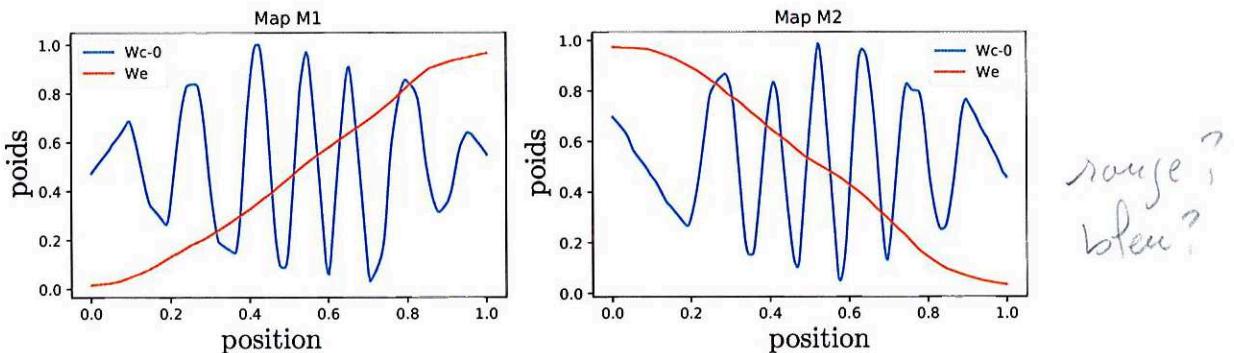


FIGURE 3.3 – Représentation des valeurs des poids d'une carte au sein de CxSOM après apprentissage en fonction de leur position dans la carte. (La seule représentation de ces poids ne suffit pas à savoir comment la carte se comporte.)

3.1.4 Limites des représentations classiques dans le cas d'une architecture CxSOM

Utilisons d'abord les représentations classiques mentionnées ci-dessus pour tracer les poids de ~~chaque~~ des cartes d'une architecture CxSOM à la fin de l'apprentissage. La fin 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 ~~des cartes~~, se dépliant sur chaque coordonnée $X^{(1)} = x$ et $X^{(2)} = y$ 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 : ce comportement est celui qu'on attend dans une carte de Kohonen classique. Les poids contextuels, en bleu, ne présentent pas cette organisation monotone. Ils présentent toutefois une continuité : deux prototypes proches ont des poids proches. Le tracé des deux couches de poids 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 dits *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 de conclusion sur l'influence des connexions entre cartes à partir de cette seule représentation.

Au regard des insuffisances des représentations classiques, que nous avons révélées sur un cas très simple de deux cartes mono-dimensionnelles, nous constatons qu'il est nécessaire de trouver un moyen de représenter l'architecture comme un tout. Nous devons ainsi définir des représentations qui montrent comment l'architecture de cartes est capable d'apprendre les relations entre les entrées multimodales.

3.1.5 Problématique

Suite aux limites montrées par les méthodes de représentation classiques utilisées dans le domaine des SOMs, ce chapitre propose plusieurs façons 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 permettant de comprendre et représenter ce que calcule une architecture CxSOM sur les données d'entrées. Ces représentations seront illustrées sur un exemple minimal de deux cartes pour faciliter la compréhension. Nous utiliserons ces méthodes dans les chapitres suivants pour analyser le comportement des cartes.

3.2 Formalisation statistique des entrées et sorties des cartes

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

3.2.1 Formalisation des entrées

Plaçons nous dans le cas général d'une architecture de N cartes pour formaliser davantage. Nous considérons des entrées multimodales tirées d'un ensemble d'espaces d'entrée $\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(n)}$. Chaque espace $\mathcal{D}^{(i)}$ est une modalité. Les observations multimodales que l'on cherche à apprendre par l'architecture de cartes sont notées $(X^{(i)} \in \mathcal{D}^{(i)}, i = 1 \dots N)$. Nous choisissons de modéliser ces entrées $X^{(i)}$ 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 $\mathcal{D}^{(1)} \times \dots \times \mathcal{D}^{(n)}$. A chaque pas de temps, un vecteur $\mathbf{X} = (X_t^{(1)}, \dots, X_t^{(n)})$ est présenté à l'architecture : il s'agit d'une réalisation de la variable jointe \mathbf{X} .

En pratique, ces variables sont des observations, issues par exemple de capteurs d'un robot. Ces observations sont issues d'un environnement général et sont donc liées par des relations au sein de ce modèle d'environnement : les variables $X^{(i)}$ ne sont pas des variables indépendantes. 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 au cercle, modélisé par une équation.

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 que nous développerons dans cette section ont pour but de mesurer comment ce modèle d'entrées est représenté par l'architecture.

3.2.2 Formalisation du modèle d'entrée par une variable cachée

Nous cherchons à évaluer expérimentalement la capacité du modèle CxSOM à apprendre à partir d'un environnement multimodal. Dans ce cadre expérimental particulier, nous choisissons des modèles d'entrées dont les relations sont connues. Ces modèles peuvent être paramétrisés

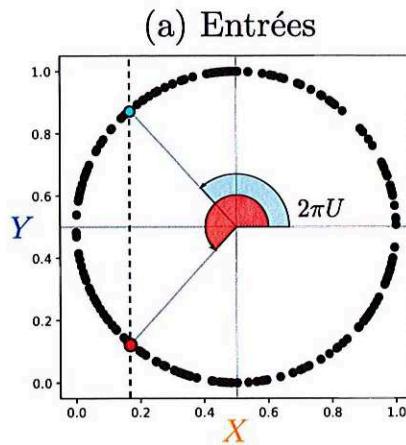


FIGURE 3.4 – Représentation choisie pour le cercle. Le modèle auxquelles appartient les modalités $X^{(1)}$ et $X^{(2)}$ est représenté par la variable U . ✓

par une variable multidimensionnelle U . Les modalités sont alors définies comme des fonctions de cette variable : $X^{(i)} = f^{(i)}(U)$ U est une nouvelle variable aléatoire représentant le modèle.

Pour que la variable U conserve toute l'information sur le modèle d'entrées, la fonction $(f^{(1)}, \dots, f^{(n)}) : (X^{(1)}, \dots, X^{(n)}) \rightarrow U$ doit être une bijection. Toute valeur d'entrées jointes correspond à un seul U , toute valeur de U renvoie à une seule valeur d'entrées jointes. La définition de U peut aussi être considérée comme un cas particulier de réduction de dimension du modèle, dans lequel la variable U , de dimension plus faible que \mathbf{X} , réduit la dimension des entrées sans perte d'information.

Dans le cas d'exemple, $\mathbf{X} = (X^{(1)}, X^{(2)})$ est un vecteur aléatoire prenant comme valeurs les coordonnées cartésiennes des points du cercle de centre $x_c, y_c = 0.5, 0.5$ et de rayon $r = 0.5$. En définissant une variable U scalaire à valeurs dans $[0, 1]$, ces variables aléatoires peuvent maintenant s'écrire, selon l'équation paramétrique du cercle :

$$\begin{cases} X^{(1)} = x_c + r \cos(2\pi U) \\ X^{(2)} = y_c + r \sin(2\pi U) \end{cases} . \quad (3.1)$$

U représente ici l'angle du point sur le cercle (à un facteur 2II près). Ce modèle n'étant pas fourni à l'architecture de cartes lors de l'apprentissage, il s'agit d'un modèle *latent*. Nous cherchons, par l'architecture de cartes, à apprendre les entrées et les relations entre entrées : nous cherchons donc à extraire une structure dans le modèle latent. La relation entre U et \mathbf{X} est bijective ; de ce fait, étudier comment l'architecture de cartes a appris U est équivalent à étudier comment l'architecture a appris le modèle d'entrées.

Cet exemple est scalaire mais la représentation sous forme de variable cachée est générale à ✓ n'importe quel dimension et nombre d'entrées. En effet, toute configuration d'entrée multimodale dépend d'un environnement global. La variable U correspond alors aux paramètres de cet environnement. Néanmoins, dans le cas de variables non artificielles, cette variable d'environnement est inconnue. Nous utiliserons donc la variable cachée U uniquement dans les expériences sur données artificielles. Elle nous permettra d'étudier et de représenter comment les cartes apprennent le modèle d'entrées.

3.2.3 Démarche expérimentale

Afin d'étudier le comportement de la carte à n'importe quel instant t de l'apprentissage, nous effectuons une phase de *test*, décrit en figure 3.5. 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 échantillon de taille 1000 de la variable aléatoire $(X^{(1)}, \dots, X^{(n)})$. La distribution des entrées test est identique à la distribution des entrées d'apprentissage ayant servi au déploiement de la carte.

Nous formalisons non seulement les entrées mais aussi chaque élément de réponse des cartes d'une architecture comme une variable aléatoire. Un élément de réponse d'une carte est une valeur régissant aux entrées lors d'une itération : positions du BMU, activité, poids du BMU. Nous choisissons notamment de s'intéresser aux position des BMUs $\Pi^{(1)}, \dots, \Pi^{(n)}$ et à leurs poids $\omega_e^{(1)}(\Pi^{(1)})$ et $\omega_e^{(2)}(\Pi^{(2)})$. La valeur de ces éléments est indépendante entre deux itérations de test, car les poids ne sont pas mis à jour. Grâce à cette indépendance entre itérations, les valeurs obtenues lors d'une phase de test forment un échantillon de la variable aléatoire jointe :

$$\underbrace{(X^{(1)}, \dots, X^{(n)}, U)}_{\text{Entrées}}, \underbrace{\Pi^{(1)}, \dots, \Pi^{(n)}, \omega_e^{(1)}(\Pi^{(1)}), \dots, \omega_e^{(n)}(\Pi^{(n)})}_{\text{Eléments de réponse}}$$

Les composantes de cette variable jointe ne sont pas indépendantes. Maintenant que la démarche expérimentale est formalisées, les représentations présentées par la suite sont simplement des tracés de dépendances au sein d'un échantillon de test entre composantes de la variable jointe définie ci-dessus.

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 à une discréétisation de l'espace continu $[0, 1]$, et sont ordonnées. Le déplacement par relaxation n'est pas limité aux positions discrètes des BMUs. Les variables aléatoires considérées sont bien à valeurs continues. Le formalisme par variable aléatoires permettra aussi d'utiliser des outils et métriques issus de la théorie de l'information pour qualifier l'organisation des cartes au sein de l'architecture, ce que nous ferons dans le chapitre 4.

3.3 Représentations graphiques

A partir des échantillons de test, nous proposons dans cette section les représentations graphiques que nous utiliserons pour évaluer expérimentalement les architectures de cartes. Ces représentations sont toutefois un tracé de dépendances entre des composantes de la variable

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

dont un échantillon est obtenu 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.

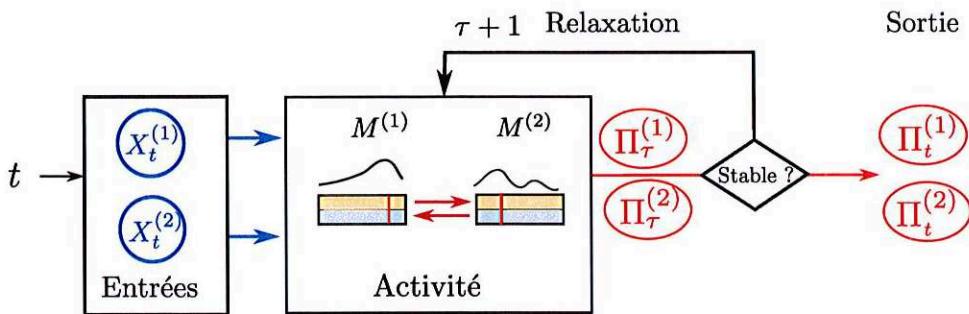


FIGURE 3.5 – Schéma décrivant une étape de test. Un test consiste à présenter successivement des réalisations de \mathbf{X} , notées $(X_t^{(1)}, X_t^{(2)})$. Nous laissons le processus de relaxation stabiliser les BMUs. Quand la stabilité est atteinte, la valeur des positions de BMU $\Pi_t^{(1)}$ et $\Pi_t^{(2)}$ est obtenue. Un échantillon de test complet est ainsi obtenu en présentant un ensemble de réalisations de \mathbf{X} . Les poids ne sont pas mis à jour entre chaque itération, ce qui permet de considérer une phase de test comme un échantillonnage de la variable aléatoire $(X^{(1)}, X^{(2)}, U, \Pi^{(1)}, \Pi^{(2)})$

3.3.1 Erreur de quantification d'une modalité dans chaque carte

La première fonction d'une carte de Kohonen est de réaliser une tâche de quantification vectorielle sur son entrée externe. Au sein d'une architecture de cartes, nous nous attendons à ce que chaque carte extraie une représentation de la modalité qu'elle prend en entrée externe. Afin de mesurer cette qualité de quantification vectorielle au sein d'une carte dans CxSOM, nous traçons le nuage de points correspondant au poids externe du BMU $\omega_e(\Pi^{(i)})$ en fonction de l'entrée externe présentée $X^{(i)}$. Une carte effectue une quantification vectorielle correcte si ce nuage de points est proche de la fonction identité. Ces tracés sont réalisés en figure 3.6 pour l'expérience exemple. Ces tracés s'approchent de l'identité : la quantification des entrées est correctement réalisée. On pourrait mesurer une erreur quadratique moyenne pour déterminer numériquement cette erreur de quantification mais la représentation en nuage de points est, à défaut d'être quantitative, plus qualitative. En effet, ici, on observe que le nuage montre une structure "filamenteuse". Nous reviendrons sur ce point par la suite, nous contentant de souligner ici que la représentation graphique exprime une propriété que la simple mesure d'erreur n'aurait pas mise en évidence.

Cette représentation nous informe ainsi sur la qualité de quantification dans une seule carte relativement à une seule modalité. Cette seule représentation est insuffisante à elle seule pour comprendre plus en détail le comportement d'une architecture de cartes : il nous faut également définir des méthodes de représentation permettant d'évaluer comment la structure globale du modèle d'entrées est apprise par l'architecture dans son ensemble.

3.3.2 Représentation cartographique des valeurs d'entrées préférentielles des BMUs

En biologie, les aires du cortex cérébral sont cartographiées en faisant varier le motif d'entrée dans son espace, et en indiquant pour chaque neurone la valeur d'entrée préférentielle à laquelle il réagit. Cela donne alors une représentation cartographique où des valeurs de l'espace d'entrée sont tracées par rapport à la position sur le substrat neuronal du neurone qui y réagit. Par

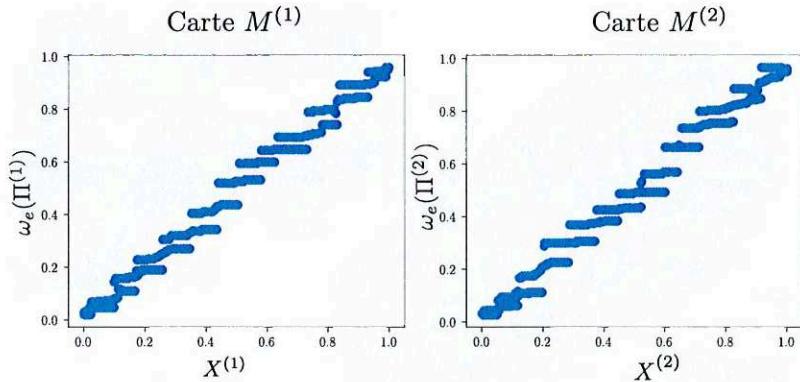


FIGURE 3.6 – Poids du BMU dans chaque carte en fonction de l’entrée présentée. On s’attend à des tracés proches de l’identité, montrant que le poids du BMU d’une carte est une bonne représentation de l’entrée. Sur ce graphique, on se rapproche effectivement de la fonction identité, cependant, une faible erreur est observée. On observe également un découpage des poids en bandes.

exemple, une carte corticale est tracée pour l’aire visuelle primaire du cortex cérébral, l’aire v1, en figure 3.7.

Un échantillon test donne un ensemble de valeurs de variable jointe $(X^{(1)}, \dots, X^{(n)}, \Pi^{(1)}, \dots, \Pi^{(n)})$. Dans la même idée que les cartes corticales, nous tracerons à partir de l’échantillon de test le nuage de points correspondant à 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. 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*. En plus des couples $(\Pi^{(i)}, X^{(i)})$, nous traçons également les entrées externes des autres cartes en fonction de $\Pi^{(i)} : (\Pi^{(i)}, X^{(j)})$. Enfin, sur le même graphique, nous mettons ces valeurs d’entrées externes en relation avec les poids de la carte $M^{(i)}$ en traçant les poids externes et contextuels de la carte $M^{(i)}$ en fonction de leur position. Une représentation cartographique fait ainsi apparaître les poids externes et contextuels d’une carte de l’architecture. À ces poids s’ajoutent plusieurs nuages de points correspondant aux valeurs d’une entrée $X^{(j)}$ en fonction du BMU de la carte $M^{(i)}$, $\Pi^{(i)}$.

En figure 3.8, nous avons ainsi tracé cette représentation cartographique pour l’exemple à deux cartes. Nous y voyons deux nuages de points : $(\Pi^{(1)}, X^{(1)})$ et $(\Pi^{(1)}, X^{(2)})$ issus de l’expérience sur les deux cartes, ainsi que deux courbes : les poids externes et contextuels de la carte $M^{(1)}$. Deux valeurs issues de l’échantillon de test sont mise en évidence en couleur rouge et bleue sur chaque graphique. Un point de même couleur correspond à la même itération de test 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 : $M^{(2)}$ reçoit donc une entrée $X^{(2)}$ différente entre de ces deux itérations.

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, comme le montrait déjà la figure 3.6.

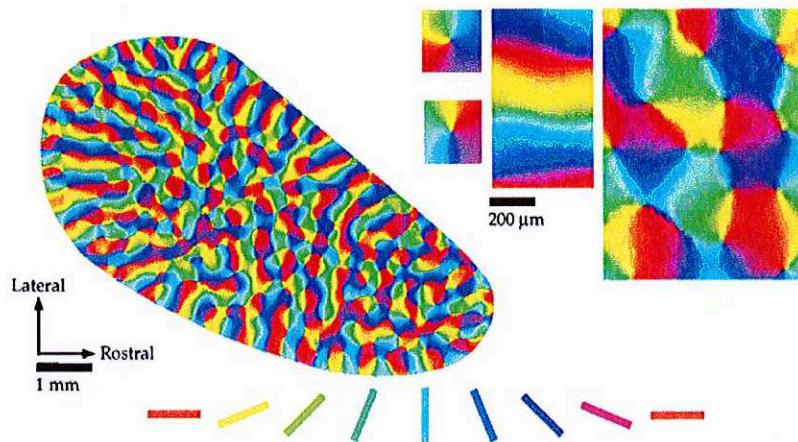


FIGURE 3.7 – Carte corticale de l'aire cérébrale visuelle V1. Pour tracer cette représentation, un ensemble de traits de différentes orientation sont présentés en stimuli visuels au sujet, indiqués en bas de l'image. Le neurone réagissant à une entrée d'orientation particulière est coloré sur la carte de la couleur correspondante à l'entrée. Cette méthode permet de tracer des *cartes corticales* d'une aire cérébrale [2].

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, sans BMUs. 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 dans une architecture de cartes, par rapport à une carte classique.

Les nuages de points $(\Pi^{(1)}, X^{(1)})$ et $(\Pi^{(1)}, X^{(2)})$ nous permettent d'observer quelles valeurs d'entrées sont codées dans les zones observées sur les positions des BMUs. Nous observons notamment qu'une zone encode pour un intervalle de valeurs spécifique pour le couple $(X^{(1)}, X^{(2)})$ et non seulement pour l'entrée externe de la carte, comme ce serait le cas dans une carte seule (b). Les points rouges et bleus illustrent ce comportement. Ces échantillons correspondent à la même valeur d'entrée externe présentée à $M^{(1)}$, mais $X^{(2)}$ est différent. Dans la carte indépendante, le BMU sera identique pour ces deux valeurs. Sur la figure (c), le point rouge et le point bleu ont leurs BMUs dans deux zones adjacentes, mais séparées, sur la carte.

La représentation des valeurs d'entrées $X^{(j)}$ selon la position du BMU $\Pi^{(i)}$ permet ainsi d'identifier une répartition des BMUs sur une carte que nous ne pourrions pas observer en traçant simplement les poids. Nous voyons notamment que la carte s'auto-organise en zones de BMUs, séparées par des zones mortes. Les ~~nuages~~ d'une zone réagissent à des valeurs spécifiques du couple d'entrées et non de seulement l'entrée externe. Cette organisation en zone observée sur ce graphique traduit l'existence de deux échelles d'indices : une organisation primaire selon la valeur de l'entrée externe, et une organisation secondaire au sein des zones selon la valeur de l'entrée de l'autre carte. Nous étudierons dans les chapitres suivants, grâce à cette représentation, si ce comportement se généralise pour différentes architectures et distributions d'entrées.

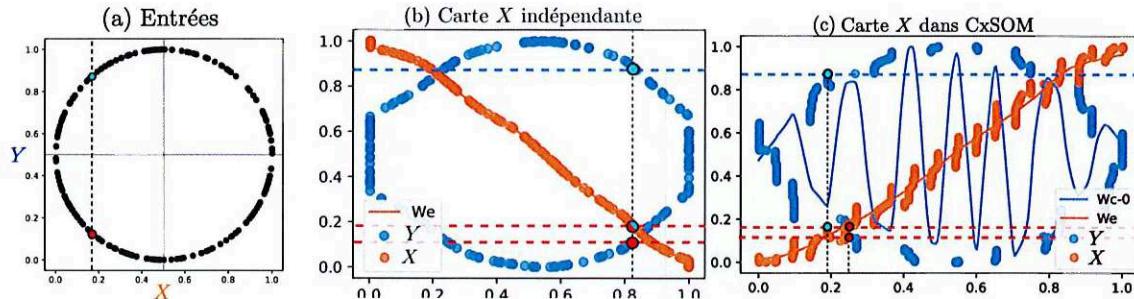


FIGURE 3.8 – Représentation des entrées $X^{(1)}, X^{(2)}$ d'une architecture de deux cartes relativement au BMU de la carte $M^{(1)}$ après apprentissage. Ces tracés mettent en valeur l'organisation des cartes, différentes dans le cas où les cartes apprennent indépendamment leurs entrées (b) ou sont connectées (c). Les entrées correspondantes sont en figure (a). Les points bleus et rouges reportés sur les tracés correspondent au même échantillon de test.

3.3.3 Représentation de la variable cachée selon les positions des BMUs

Nous avons vu lors de la représentation cartographique des entrées que chacune des cartes choisit son BMU 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 . Afin de mettre en valeur ce comportement, nous tracerons les nuages de points représentant U en fonction de la position $\Pi^{(i)}$ du BMU d'une carte pour afficher comment la position du BMU traduit la relation entre les entrées.

En figure 3.9, nous traçons 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 pour lequel deux valeurs de U correspondent à un même BMU. Cela traduit bien le fait que chaque carte a appris une représentation du modèle d'entrée et non seulement de son entrée externe. L'organisation de la carte dans CxSOM montre donc chaque position Π codant pour une seule valeur de U , c'est à dire une seule position d'échantillon sur le cercle. 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 alors à vérifier si U est une fonction de Π dans chacune des cartes de l'architecture.

Cette représentation fait apparaitre U comme une fonction de Π dans chaque carte. Ce comportement traduit le fait que les cartes s'organisent en fonction de tout le modèle d'entrée, non seulement de leur entrée externe. Il s'agit là d'un comportement que nous pouvons qualifier de comportement de base pour CxSOM.

3.3.4 Dépliement d'une carte dans l'espace d'entrée multimodal

Une représentation classique utilisée pour représenter une carte de Kohonen est de tracer les poids du graphe qu'est la carte dans l'espace de ses entrées, telle qu'en figure de droite en 3.2. Dans CxSOM, chaque carte prend en entrée une seule modalité. Par contre, grâce à l'échantillon de test, il est possible de représenter comment la carte se déplie dans l'espace de toutes les modalités.

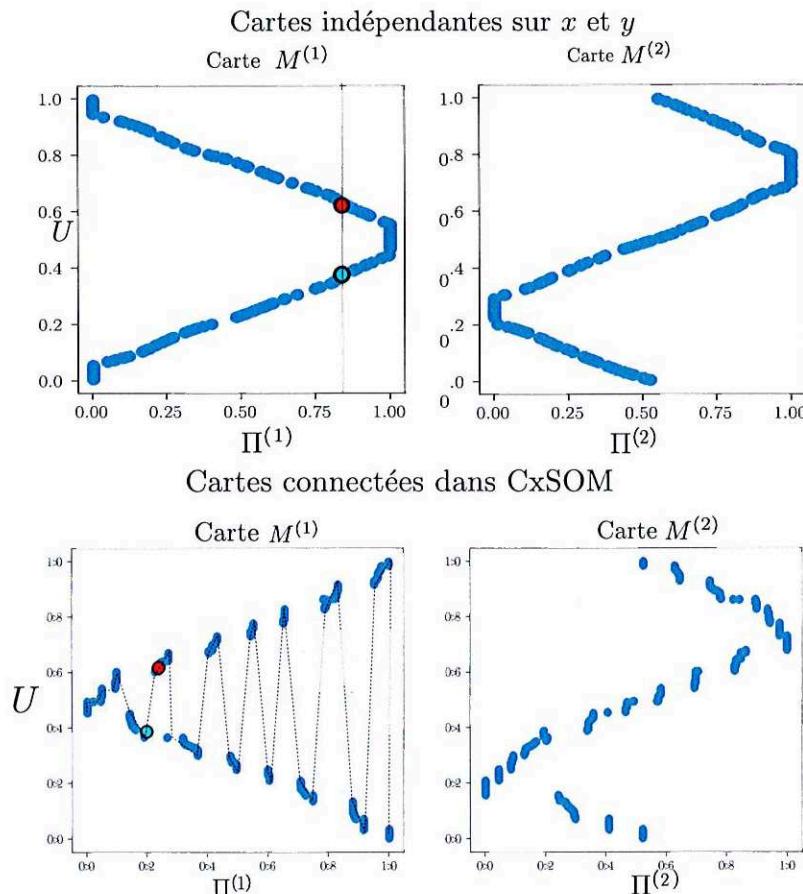


FIGURE 3.9 – Valeur de U en fonction des valeurs du BMU $\Pi^{(i)}$ dans chacune des cartes dans l'exemple d'illustration. Sur la première ligne, nous traçons la réponse de chaque carte à son entrée dans le cas où les deux cartes ne sont pas connectées. Sur la deuxième ligne, nous traçons la réponse 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. Les mêmes échantillons rouge et bleu mis en évidence en figure 3.8 sont reportés sur les tracés.

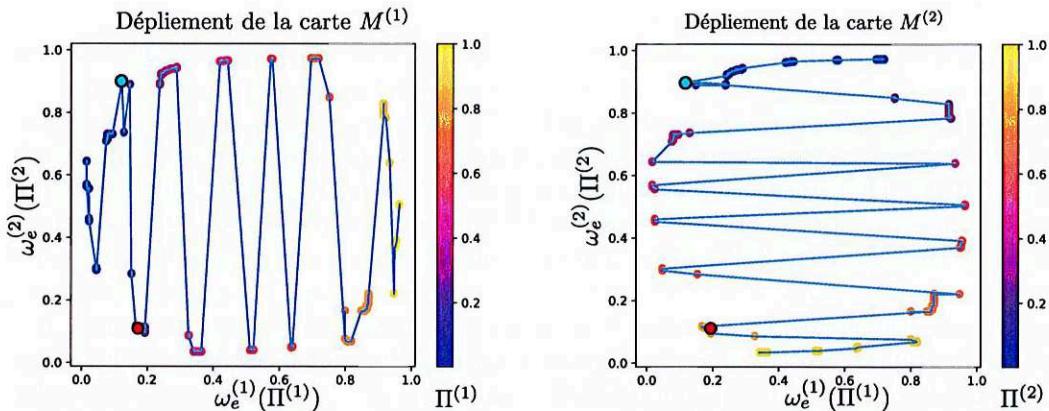


FIGURE 3.10 – Dépliement des cartes $M^{(1)}$ et $M^{(2)}$, reliés dans l'ordre de leurs positions selon $M^{(1)}$ figure de gauche et $M^{(2)}$ figure de droite. Le dépliement de chacune des cartes est alors représenté dans l'espace complet des entrées . L'indice du BMU est signalé par la carte de couleurs, différenciant ainsi les extrémités en 0 et 1 des cartes.

Nous définissons alors une façon de représenter le dépliement d'une seule carte de CxSOM dans l'espace global des entrées. Dans l'exemple, il s'agit de représenter le dépliement de $M^{(1)}$ dans l'espace des entrées $X^{(1)}, X^{(2)}$. Au lieu de s'appuyer sur les poids des cartes comme en 3.2, nous utilisons les valeurs de l'échantillon de test. Nous traçons le nuage de poids correspondant au poids des BMUs : $\omega_e^{(1)}(\Pi^{(1)}, \omega_e^{(2)}(\Pi^{(2)})$, puis relierons ces points selon l'ordre de leurs positions dans la carte $M^{(1)}$ ou $M^{(2)}$. Ainsi, nous obtenons une représentation des cartes $M^{(1)}$ ou $M^{(2)}$ dépliées dans l'espace global des entrées. Notons que les unités mortes ne peuvent pas être représentées sur la carte. Nous ne représentons que les morceaux de cartes étant effectivement BMUs. Cette représentation est tracée en 3.10 pour l'exemple à deux cartes.

La carte représentant les poids externes des BMUs, on s'attend à ce que la forme du nuage de points corresponde à la structure des entrées. Les tracés mettent en lumière les propriétés observées lors de la représentation cartographique des entrées, à savoir l'apparition de zones dans chaque carte. Les poids suivent la structure circulaire des entrées en découplant l'espace en zones. Les parties du cercle effectivement représentées par les poids sont réduites : le cercle est comme discréteisé en petits morceaux. Dans l'exemple, on remarque la façon dont est parcouru l'espace dans chaque carte : en fonction de x dans la carte $M^{(1)}$, et en fonction de y dans la carte $M^{(2)}$. Cette représentation offre la possibilité de visualiser comment une carte se déplie dans l'espace d'autres modalités.

3.4 Conclusion

Afin d'extraire des comportements d'une architecture de cartes, nous modélisons les éléments des cartes comme des variables aléatoires et l'étape de test comme l'échantillonnage d'une variable jointe. Nous introduisons notamment, dans le cas des exemples artificiels, une variable cachée U non présentée comme entrée à la carte, portant toute l'information sur le modèle d'entrées. La représentation par échantillonnage et variable aléatoire permet de mieux comprendre les mécanismes des cartes, ceux-ci reposant plus directement sur les courbes de poids.

Nous utiliserons donc quatre 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 chaque carte extrait une représentation de son espace d'entrée. La représentation cartographique des entrées selon le BMU de chaque carte fait apparaître des zones mortes dans les cartes CxSOM et affiche les valeurs préférentielles d'entrées selon leur BMU. Cette représentation met en lumière un comportement émergeant de l'exemple à deux cartes : de façon auto-organisée, chaque carte définit des zones de BMUs correspondant à intervalle de valeurs ($X^{(1)}, X^{(2)}$), et non seulement $X^{(1)}$. Cette représentation fait également apparaître une notion d'organisation selon un indice primaire sur les poids externes, et secondaire grâce aux poids contextuels. Nous étudierons plus extensivement ce comportement dans la partie suivante grâce à cette représentation. Afin d'évaluer la façon dont un BMU traduit le modèle d'entrée, nous traçons la variable cachée U en fonction du BMU d'une carte $\Pi^{(i)}$. L'apprentissage du modèle d'entrée se traduit alors par l'observation d'une *relation fonctionnelle* entre U et Π dans chaque carte. Enfin, nous proposons une représentation du dépliement de chaque carte dans l'espace de toutes les entrées en traçant les poids des BMUs de chaque carte, ordonnés selon les BMUs d'une des cartes. Cette représentation apporte une vision globale de la façon dont une carte encode le modèle d'entrées. Ces représentations permettront d'étudier plus en détail les comportements observés dans l'exemple illustratif. Cette étude sera traitée au chapitre 5.

Même avec des représentations adaptées, l'analyse d'architectures comportant de nombreuses cartes ne peut pas simplement s'effectuer à l'aide de graphiques, qui deviendraient trop complexes. La comparaison d'un grand nombre d'expériences est aussi difficilement réalisable graphiquement. Cette difficulté de représentation et le besoin de comparer des expériences soulève la nécessité de définir une valeur indicative du fonctionnement de la carte, que nous proposerons dans le chapitre suivant.

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] D. Meunier, R. Lambiotte, and E. Bullmore. Modular and hierarchically modular organization of brain networks. *Frontiers in Neuroscience*, 4, 2010.
- [23] V. Mountcastle. The columnar organization of the neocortex. *Brain*, 120(4) :701–722, April 1997.
- [24] 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.
- [25] Edmund T. Rolls and Gustavo Deco. Computational neuroscience of vision. 2002.
- [26] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27 :379–423, 1948.
- [27] Marc Strickert and Barbara Hammer. Merge som for temporal data. *Neurocomputing*, 64 :39–71, 2005.
- [28] Henri Theil, J. S. Cramer, A. Russchen, and H. Moerman. Economic forecasts and policy ed. 2nd. *Birchandra State Central Library,tripura*, 1961.