

# A propos des architectures de cartes auto-organisatrices stylées

## THÈSE

présentée et soutenue publiquement le le plus tard possible en 2022

pour l'obtention du

**Doctorat CentraleSupélec**  
**(mention informatique)**

par

Noémie Gonnier

### Composition du jury

*Président :* Le président du jury

*Rapporteurs :* Le rapporteur 1 du laboratoire  
Le rapporteur 2  
Le rapporteur 3

*Examinateurs :* L'examinateur 1  
L'examinateur 2

---

Laboratoire Lorrain de Recherche en Informatique et ses Applications





# Sommaire

<b>1</b>	<b>Approche modulaire des réseaux de neurones</b>	<b>1</b>
<b>2</b>	<b>Cartes de Kohonen et modèle d'architecture CxSOM</b>	<b>3</b>
2.1	De la biologie au calcul . . . . .	3
2.2	Algorithme Général . . . . .	3
2.3	Approche topologique des cartes de Kohonen . . . . .	4
2.4	Description de l'algorithme . . . . .	4
2.4.1	Carte de Kohonen classique . . . . .	4
2.4.2	Modèle : CxSOM . . . . .	6
2.5	A trier . . . . .	8
<b>3</b>	<b>Représentation des cartes de Kohonen</b>	<b>9</b>
3.1	Représentation et explicabilité des cartes de Kohonen . . . . .	9
3.1.1	Représentation classique des cartes de Kohonen . . . . .	10
3.1.2	Que cherche t-on à représenter dans CxSOM ? . . . . .	11
3.2	Formalisme : variables aléatoires . . . . .	12
3.2.1	Représentation des entrées . . . . .	12
3.2.2	Représentation des éléments des cartes . . . . .	13
3.3	Représentations graphiques . . . . .	14
3.3.1	Représenter les entrées par rapport à une carte . . . . .	14
3.3.2	Représentation de U par rapport au BMU . . . . .	14
3.3.3	Dépliement d'une carte en plusieurs dimensions . . . . .	15
3.4	Information mutuelle comme indicateur statistique . . . . .	17
3.4.1	Information mutuelle et entropie . . . . .	17
3.4.2	Indicateur : coefficient d'incertitude. . . . .	18
3.4.3	Estimation . . . . .	19
3.4.4	Perspectives . . . . .	20
3.5	Prédiction d'entrée . . . . .	21

<b>4 Expériences : analyse provisoire et liste</b>	<b>23</b>
4.1 Entrées sur un cercle . . . . .	23
4.1.1 Cercle 2 dimensions . . . . .	23
4.1.2 Cercle trois dimensions . . . . .	23
4.1.3 Entrées dans un anneau . . . . .	24
4.2 Entrées dans un carré . . . . .	25
4.3 Entrées en clusters . . . . .	25
4.3.1 "petits" clusters . . . . .	26
4.3.2 "gros" clusters . . . . .	26
4.4 Influence de l'architecture sur des entrées . . . . .	26
4.4.1 Cartes intermédiaires . . . . .	26
4.4.2 Et si on a la même architecture, avec une carte centrale, mais cette fois les cartes M1, M2, M3 sont également connectées ? . . . . .	29
4.4.3 Boucle vs rétroaction à trois cartes . . . . .	29
4.4.4 Conclusion et perspectives : influence de l'architecture . . . . .	29
4.5 Influence des paramètres des cartes . . . . .	29
4.5.1 rayons de voisinage . . . . .	29
4.5.2 Combinaison des activités externes et contextuelles par moyenne ou moyenne géométrique . . . . .	29
4.5.3 Influence de la relaxtion . . . . .	29
4.6 Conclusion des expériences, perspectives et limites . . . . .	29
<b>Bibliographie</b>	<b>31</b>

## Introduction

Cette thèse propose une construction d'une architecture modulaire

## Chapitre 1

# Approche modulaire des réseaux de neurones

Les systèmes d'information sont le fruit d'un travail de construction et de recherche, cherchant à créer de nouveaux comportements dans une architecture globale. En particulier, la recherche en systèmes d'intelligence artificielles cherche sans cesse à créer de nouvelles structures afin de générer des comportements. Nous introduisons des règles de calcul entre des modules plus simples qui sont des cartes auto-organisatrices.

Ces règles utilisent des propriétés topologiques des cartes de Kohonen et sont créées dans l'idée d'avoir un système décentralisé entre les cartes. On espère ainsi ouvrir les cartes de Kohonen à du calcul décentralisé. L'idée que l'interaction entre des cartes amène de nouveaux comportement n'est pas nouvelle : elle se base sur l'émergence de comportements dans les systèmes complexes.

Un système complexe est un système dans lequel de nombreux composants interagissent de façon non-linéaire, avec des rétro-actions. De ce fait, il est difficile d'appréhender et de comprendre un tel système sans le simuler. Ces systèmes présentent notamment des propriétés d'auto-organisation.

Cette émergence de comportement est étudiée dans les réseaux biologique, et a été beaucoup utilisée pour la création de nouveaux paradigmes d'apprentissage automatiques.



## Chapitre 2

# Cartes de Kohonen et modèle d'architecture CxSOM

Idée du chapitre :

"Qu'est ce qu'on veut faire avec des cartes de Kohonen?" " A quoi servent les cartes de Kohonen ?" ok on les utilise pour de la visualisation, de la réduction de dimension. La visualisation est bien pour un observateur humain, la réduction de dimension peut impliquer qu'on va utiliser un algorithme derrière. Mais les cartes de Kohonen vont plus loin dans l'apprentissage : on a une approximation de l'espace d'entrée par un graphe. Cela veut dire qu'une entrée est associée à un prototype dans la carte, mais inversement : un prototype est associé à un ensemble d'entrée continu ou contigu. Une entrée est alors représentée par notamment sa position dans la carte : un nombre donc, ou une paire. Il est possible de faire du calcul sur ces positions au sein d'algorithmes.

Dans cette thèse, on a pensé à utiliser cette propriété pour construire un réseau de cartes auto-organisatrices. Par ce réseau, on peut exploiter les positions pour générer des dynamiques au sein de la carte qui permettront une prise de décision, ou des représentations de données différentes.

// Kohonen : il faut surprendre encore ! Par quel bout le prendre ? → Appuyer sur les cartes 1D → Comment ça se fait qu'on les utilise pas de ouf ? → Intérêt de la topologie de la carte. Dans une carte seule, est ce que c'est vraiment utile ? → Questionnement informatique : qu'est ce qui se passe en fait dedans, mais c'est quand même rigolo.

### 2.1 De la biologie au calcul

De la biologie au calcul : patterns temporels des neurones impulsoriels vs SOM

### 2.2 Algorithme Général

Une carte de Kohonen est un graphe dans lequel chaque noeud possède un poids  $\omega$  appartenant à l'espace des entrées. L'algorithme repose ensuite sur l'adaptation de ces poids, en prenant en compte les connexions dans le graphe, afin de représenter les données d'entrées. Ainsi, n'importe quel graphe pourrait être considéré ; le plus souvent, une grille 2D est utilisée.

Mettre ici algo

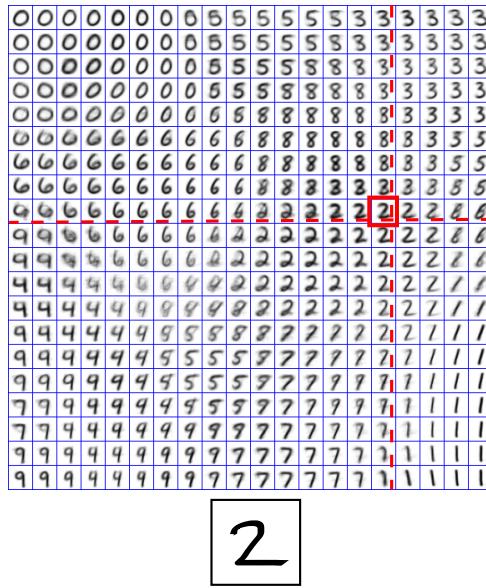


FIGURE 2.1 – Une carte de Kohonen s’organise en zones dont les poids sont proches dans l’espace des entrées. Chaque entrée présentée à la carte peut alors être représentée par la valeur de la position du BMU correspondant dans la carte. Les entrées sont projetées sur le carré  $[0, 1] \times [0, 1]$ .

## 2.3 Approche topologique des cartes de Kohonen

La notion de voisinage et de topologie est un élément clé des cartes de Kohonen. Le voisinage est en effet pris en compte lors de l’apprentissage et lors de l’interprétation des cartes. Cependant, ce voisinage est généralement défini, dans les applications des cartes, comme un bonus par rapport aux KMeans, une aide à la convergence et à la vitesse de dépliement. Pourtant c’est la l’essence même d’une carte de Kohonen : projeter des éléments sur un graphe, ce qui nous permet de faire des calculs sur des positions plutôt que des données de grandes dimensions.

## 2.4 Description de l’algorithme

Le but de cette thèse est de proposer un modèle permettant d’associer des cartes auto-organisatrices dans n’importe quel type d’architecture, comme une sorte de brique de base. En particulier, on cherchera à construire des architectures non-hiéarchiques de cartes, exemple en figure 2.2. Nous nous placons donc dans le cadre de modules pré-établis, dont les entrées ont été connectées par avance. Les poids de chaque carte seront quant à eux appris, avec comme objectif que les cartes apprennent leurs entrées mais puissent également distinguer un état global de l’architecture. Pour les entrées, nous nous placons dans un cadre de multi-modalité, détaillé au chapitre suivant. Les différentes cartes prendront des données d’entrées sur différents espaces.

### 2.4.1 Carte de Kohonen classique

Rappelons les notations concernant une carte de Kohonen standard. Prenons un ensemble de données d’entrées, dans lequel chaque élément est un vecteur d’un espace  $D$ . On a défini une distance  $d$  sur  $D$ , généralement la distance euclidienne. La carte de Kohonen construite sur ces entrées est un graphe, généralement une ligne 1D ou une grille 2D de  $N$  noeuds. Chaque noeud

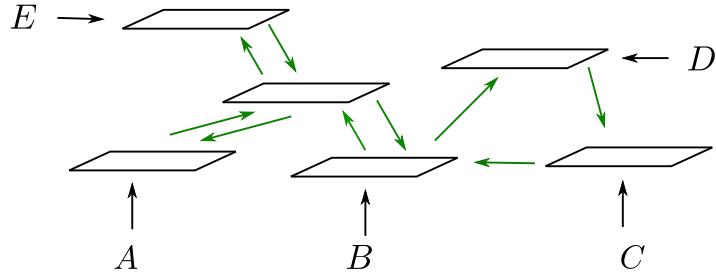


FIGURE 2.2 – Exemple d'architecture modulaire *non-hiéarchique* de cartes de Kohonen. Les entrées sont  $A, B, C, D, E$  quelconques. Chaque carte peut ou non prendre une entrée ; les connexions sont réciproques ou non.

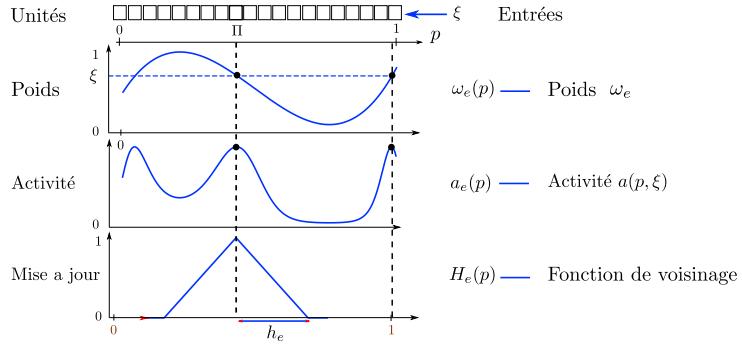


FIGURE 2.3 – Notations utilisées dans une carte de Kohonen simple

possède un poids associé  $\omega_e$  dans  $D$  ou *prototype*, du même espace que les entrées. et une *position*  $i$  dans la carte. Ces positions sont ensuite indexées entre 0 et 1 par  $p = \frac{i}{N}$  pour l'homogénéité des calculs. L'ensemble des poids est noté  $\omega_e(p)$ ,  $p \in [0, 1]$ . L'algorithme se décompose de la façon suivante :

1. Une entrée  $\xi_t$  est présentée à la carte.
2. L'unité ayant le poids le plus proche de  $\xi_t$  selon la distance  $d$  est choisie comme *Best Matching Unit* de la carte. Sa position est notée  $\Pi$ .
3. Chaque poids  $\omega_e$  est déplacé vers l'entrées  $\xi$ , en fonction de sa distance dans la carte à la best matching unit :

$$\omega_e(p, t + 1) = \omega_e(p, t) + \alpha h(\Pi, p)(\xi - \omega_e(p, t)) \quad (2.1)$$

$h(\Pi, p)$  est la *fonction de voisinage*. Elle est maximale en  $p = \Pi$  et décroissante autour de cette position. Dans notre étude, les fonctions de voisinage sont triangulaire, donc maximale en  $\Pi$ , décroissante sur le rayon de voisinage  $h_e$  et nulle après.

Lors de l'étape 2 de l'algorithme, une activité peut être calculée, au lieu d'une distance pour choisir le BMU. Ce dernier est alors choisi comme  $\Pi = \arg \max_p(a(\xi, p))$ . Nous utiliserons cette solutions dans notre modèle. Les notations au sein d'une carte sont résumées en figure 2.3.

### 2.4.2 Modèle : CxSOM

Décrivons maintenant le modèle CxSOM étudié dans cette thèse. Dans ce modèle, l'algorithme original de Kohonen est modifié afin de connecter des cartes entre elles, et d'autoriser des connexions non-hiéronymiques. Définissons la connexion entre deux cartes. Une carte A est connectée à une carte B lorsque la carte B prend en entrée la position du BMU de la carte A. Considérons  $G$ , le graphe de connexions des cartes. Ce graphe est *orienté* et les *boucles* sont autorisées. C'est ce qu'on appellera *architecture non-hiéronymique* de cartes, par opposition à des architectures comme HSOM dans laquelle le BMU d'une carte A nourrit une carte B de façon unidirectionnelle. Chaque carte aura ainsi plusieurs entrées : une entrée *externes* dans un espace d'entrée, facultative, et  $k$  entrées *contextuelles* qui sont les positions des BMU des cartes qui lui sont connectées. Par ailleurs, la recherche du BMU doit être modifiée par rapport à l'originale : les rétroactions entre les cartes sont autorisées, la position du BMU de la carte A va donc influencer la position du BMU de la carte B, lequel modifie à nouveau le BMU de la carte A, etc. Notre algorithme présente donc deux modifications principales :

- Les cartes possèdent plusieurs entrées, externes et contextuelles. Le calcul de l'activité est donc modifié afin de prendre en compte ces différentes couches d'entrées.
- La recherche du BMU est modifiée afin de gérer les rétroactions entre cartes.

La description du modèle CxSOM est détaillée en figure 2.5, dans un cas où une carte reçoit deux connexions, et l'algorithme explicité en ??.

#### Gestion des entrées externes et contextuelles

À un pas d'apprentissage  $t$ , une carte  $M$  reçoit en entrée une entrée *externe* notée  $\xi_t$  et  $K$  entrées *contextuelles* notées  $\gamma_{0t}, \dots, \gamma_{Kt}$ , qui sont les BMU II des cartes qui lui sont connectées. La carte possède donc  $k + 1$  couches de poids.  $\omega_e$  correspond à l'entrée externe et  $\omega_{c0}, \dots, \omega_{cK}$  aux entrées contextuelles. On calcule une activité séparément sur chaque couche de poids selon la formule suivante :

$$a(p, x) = \exp\left(\frac{(\omega(p) - x)^2}{2\sigma^2}\right) \quad x = \xi_t \text{ ou } \gamma_{kt}, \quad \omega = \omega_e \text{ ou } \omega_{ck} \quad (2.2)$$

Les activités contextuelles sont moyennées en une activité  $a_c(p, \gamma_t)$ , avec  $\gamma_t = (\gamma_{0t}, \dots, \gamma_{Kt})$ . Les activités externes et contextuelles sont enfin fusionnées en une activité globale :

$$a_g(p, \xi_t, \gamma_t) = \sqrt{a_e(p, \xi_t)(\beta a_e(p, \xi_t) + (1 - \beta)a_c(p, \gamma_t))} \quad (2.3)$$

Une convolution est appliquée sur cette activité globale. Cela évite les effets de plateau. Cette activation globale est utilisée pour déterminer le BMU de la carte.

#### Gestion des rétroactions dans l'architecture

Contrairement à une carte simple, on ne peut pas calculer tous les BMUs de l'architecture en prenant l'argmax de  $a_g$  dans chaque carte. À cause des influences mutuelles entre cartes, calculer le BMU d'une des cartes modifie les entrées des autres cartes de l'architecture, et donc leur BMU. Cette recherche est donc réalisée par un processus dynamique que l'on appellera *relaxation*, menant à un consensus entre cartes : on cherche le point, s'il existe, où chaque BMU maximise l'activité globale de chaque carte.

Le processus de relaxation est donc une boucle imbriquée dans un pas d'apprentissage de l'architecture, indexée par  $\tau$ . Notons  $\Pi^{(\tau)i}$  la position du BMU de la carte  $i$ , et  $\Pi =$

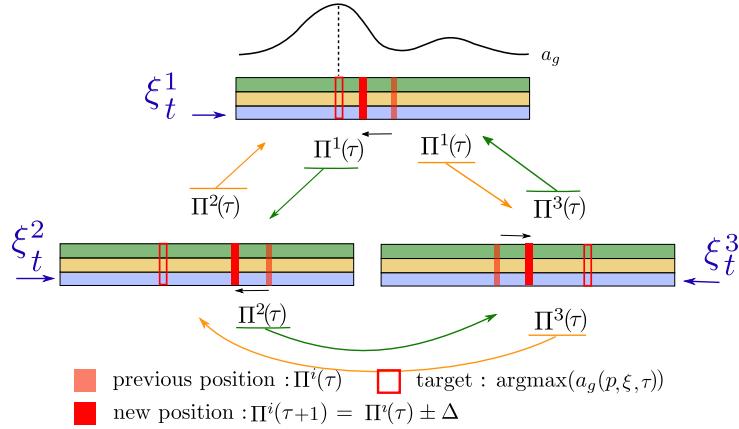


FIGURE 2.4 – description d'une étape de la relaxation dans l'architecture, aboutissant à un consensus entre cartes. Au sein d'une même itération  $t$ , les positions des BMU  $\Pi$  sont légèrement déplacées jusqu'à ce que toutes les positions  $\Pi$  des cartes de l'architecture soient stables. Ces positions maximisent collectivement les activités globales de chaque carte.

$(\Pi^{(0)}, \dots, \Pi^{(n)})$ , avec  $n$  le nombre de cartes de l'architecture. Au début d'un pas d'apprentissage, chaque carte est nourrie avec une entrée externe  $\xi_t^i$ , et les activités externes  $a_e^i(\xi_t^i, p)$  de chaque carte peuvent être calculées. La recherche du BMU suit donc le processus de relaxation suivant :

1. Dans chaque carte  $i$ , la position  $\Pi^i$  est initialisée à  $\arg \max_p (a_e^i(\xi_t^i, p))$ . Les entrées contextuelles sont alors initialisées en prenant le BMU correspondant aux connexions de l'architecture.
2. Tant que toutes les positions  $\Pi^i$  ne sont pas stables,
  - (a) Dans chaque carte  $i$ , calculer les activités contextuelles et globales, définissant ainsi  $p^{*i} = \arg \max_p (a_g(p, \gamma^i, \xi^i))$
  - (b) Déplacer  $\Pi^i$  vers  $p^{*i}$  :  $\Pi^i \leftarrow \Pi^i \pm \Delta$  si  $|\Pi^i - p^{*i}| \geq \Delta$ ,  $\Pi^i \leftarrow p^{*i}$  sinon
3. Le BMU de chaque carte est pris comme la valeur finale stable de ce processus dynamique. Cette valeur est utilisée pour les mise à jour des poids.

Il peut arriver que les positions se stabilisent sur un cycle limite. Dans ce cas, on arrêtera la relaxation arbitrairement ; ce phénomène étant ponctuel, il n'influencera pas l'apprentissage. Les paramètres des cartes de l'architecture sont choisis pour éviter de telles situations.

### Mise à jour des poids

Les poids sont mis à jour par rapport à leurs entrées respectives suivant l'équation 2.1. Le BMU d'une carte est ainsi commun à toutes les couches. Les rayons de voisinage  $h_e$  et  $h_c$  ont des valeurs différentes ; celles-ci seront détaillées en partie suivante.

### Tests

Les expériences faites sur l'architecture se décomposent en une phases d'apprentissage et phases de test. Pendant les tests, la mise à jour des poids des cartes est gelée et seuls le calcul des activités et le processus dynamique de sélection du BMU sont effectués.

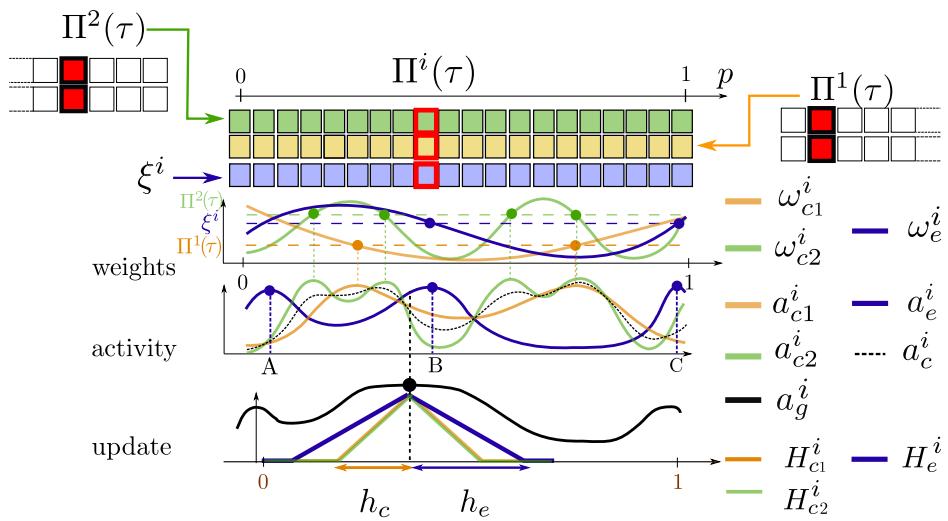


FIGURE 2.5 – Description d'une carte au sein d'une architecture CxSOM. La carte reçoit deux connexions de cartes voisines, et possède donc deux couches contextuelles

## 2.5 A trier

## Chapitre 3

# Représentation des cartes de Kohonen

### Sommaire

---

<b>3.1</b>	<b>Représentation et explicabilité des cartes de Kohonen</b>	<b>9</b>
3.1.1	Représentation classique des cartes de Kohonen	10
3.1.2	Que cherche t-on à représenter dans CxSOM ?	11
<b>3.2</b>	<b>Formalisme : variables aléatoires</b>	<b>12</b>
3.2.1	Représentation des entrées	12
3.2.2	Représentation des éléments des cartes	13
<b>3.3</b>	<b>Représentations graphiques</b>	<b>14</b>
3.3.1	Représenter les entrées par rapport à une carte	14
3.3.2	Représentation de U par rapport au BMU	14
3.3.3	Dépliement d'une carte en plusieurs dimensions	15
<b>3.4</b>	<b>Information mutuelle comme indicateur statistique</b>	<b>17</b>
3.4.1	Information mutuelle et entropie	17
3.4.2	Indicateur : coefficient d'incertitude.	18
3.4.3	Estimation	19
3.4.4	Perspectives	20
<b>3.5</b>	<b>Prédiction d'entrée</b>	<b>21</b>

---

### 3.1 Représentation et explicabilité des cartes de Kohonen

Les algorithmes d'apprentissage sont composés de structures complexes. Leurs règles d'évolution et leurs structures sont certes connues et conçues par leur développeur, mais leur état au cours de l'apprentissage dépend de tellement de paramètres que le concepteur ne peut plus prévoir son état. Celui-ci doit alors être étudié, observé et mesuré afin de pouvoir utiliser le système pour des tâches d'apprentissages. La représentation d'un algorithme d'apprentissage est ainsi un défi posé depuis quelques années. S'intéresser à une compréhension et une représentation fine des mécanismes impliqués dans un algorithme d'apprentissage est important pour une transparence de l'algorithme, mais aussi pour permettre une amélioration de ces algorithmes. Dans la mesure où cette thèse cherche à construire un réseau de neurones, il semble important de poser des métriques claires pour l'analyse de l'apprentissage et permettre un développement futur. [2]

Lorsqu'on s'intéresse à des algorithmes supervisés, dans lesquels l'évolution dépend d'une fonction de coût et d'un objectifs, des métriques assez évidentes existent, en s'intéressant à l'erreur de prédiction. Mais même en situation supervisée, le problème est largement ouvert

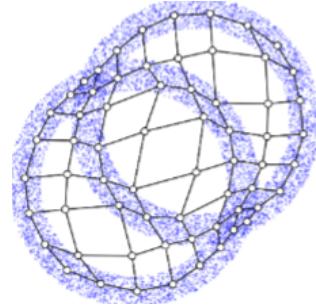
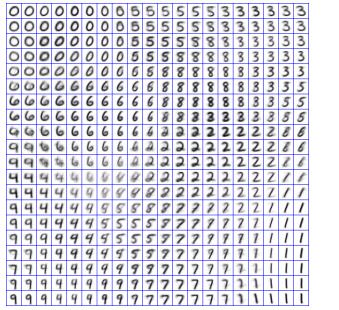


FIGURE 3.1 – Représentations possibles des poids d'une carte de Kohonen classiques, dans le cas d'entrées sous forme d'imagettes ou de points en deux dimensions.

quand il s'agit de comprendre les mécanismes d'apprentissage à l'oeuvre dans la structure. Cette question de représentation est notamment soulevée dans l'étude de l'explicabilité de l'intelligence artificielle. Quand on s'intéresse aux algorithmes non-supervisés, la question de représentation de l'algorithme devient centrale : *Que cherche-t-on à représenter et comment déterminer si on en a extrait une bonne représentation ?*

En particulier, "Est ce que les prototypes ont extrait une information pertinente des données" n'a en fait que des réponses partielles dans la littérature. Il s'agit d'abord de déterminer ce qu'on cherche à apprendre dans un cas spécifique. Nous nous poserons ainsi cette question pour l'architecture CxSOM.

### 3.1.1 Représentation classique des cartes de Kohonen

Les cartes de Kohonen sont un algorithme d'apprentissage non-supervisé, mais sont particulièrement associées à une facilité de représentation et de visualisation. En effet, leur nombre réduit de prototypes et leur aspect topologique permet d'en tracer une représentation visuelle interprétable. La manière la plus utilisée de représenter une carte de Kohonen est de tracer les poids de ses prototypes, disposés dans le graphe qu'est la carte. En fonction des dimensions des entrées, cette représentation prennent plusieurs formes. Deux exemples courants de représentation sont les suivants :

- Le graphe qu'est la carte de Kohonen est représenté dans l'espace de ses positions (la grille d'indices  $(i, j)$ , ou une ligne indexée par  $i$ ). Sur chaque noeud est tracé le poids correspondant. C'est le cas sur l'exemple de gauche en figure 3.1.1 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 de labeliser chaque prototype et d'afficher ces labels 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 3.1.1.

On parle ici d'interprétation visuelle humaine. Pour l'oeil humain, cette facilité d'interprétation est limitée à un domaine d'utilisation : celui dans lequel les éléments qui nous intéressent sont les distances euclidiennes entre les données, ou plus généralement dans lequel la distance considérée pour la mise à jour des cartes possède un aspect graphique facilement interprétable.

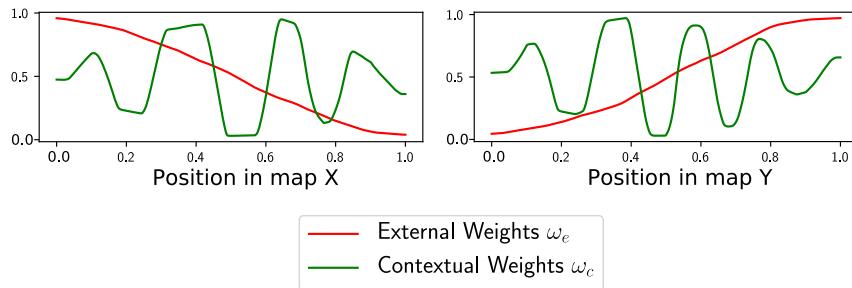


FIGURE 3.2 – Représentation des valeurs des poids d'une carte au sein de CxSOM. La seule représentation de ces poids ne suffit pas à savoir comment la carte se comporte.

Essayez par exemple de vous représenter des distances dans un espace non-euclidien, ou des distances . Savoir quels points sont les plus proches nécessite alors un effort mental important et non une seule intuition ; finalement la façon la plus simple de le savoir est de faire le calcul.

La représentation d'une carte cherche à répondre à la question : "est-ce que la carte est bien dépliée sur toutes les données ? Est-ce qu'un prototype représente correctement une donnée ?". Y répondre en visualisant ses prototypes revient au processus intellectuel suivant : l'observateur imagine une donnée, par exemple une imagette d'un chiffre à main levée, et reproduit le processus de sélection du BMU qui a eu lieu lors de l'apprentissage de la carte pour trouver le poids qui lui correspond le mieux. Via ce processus mental, on est capable d'évaluer si une carte est dépliée sur les données. Imaginons à présent que les distances considérées entre les éléments d'une carte ne soient plus euclidiennes : cette évaluation du dépliement de la carte repose maintenant sur soit une capacité d'abstraction phénoménale de l'observateur, soit des calculs de distances entre les points. La représentation de la carte doit ainsi être ajustée en fonction du processus d'organisation.

Finalement, pour représenter un algorithme d'apprentissage non-supervisé et en particulier une carte de Kohonen, on doit d'abord bien poser ce qu'on cherche à évaluer ; ensuite, cette représentation doit être adaptée aux règles de calcul de l'algorithme, ici de l'espace de la carte.

### 3.1.2 Que cherche t-on à représenter dans CxSOM ?

Pour pouvoir étudier le comportement d'une architecture de cartes, il nous faut donc répondre à ces deux questions de représentation.

La représentation des prototypes dans chaque carte n'est plus un bonne représentation de l'architecture. En effet, le choix du BMU se fait suivant plusieurs activités, et plus encore, suivant un processus de relaxation. Il est bien entendu possible de tracer les poids d'une carte après apprentissage, comme représenté en figure 3.1.2. Cependant, le processus intellectuel menant à la représentation mentale d'une carte, en regardant les poids, n'est plus possible. En imaginant une donnée, on ne pourra pas trouver le BMU sélectionné. La simulation du processus d'activité et relaxation est nécessaire pour la représentation compréhensible par l'humain d'une carte au sein de CxSOM. Par ailleurs, chaque unité d'une carte a plusieurs poids. Il est donc compliqué de comprendre directement le rôle de ces poids en regardant leur valeur.

De plus, la représentation visuelle des cartes d'une architecture est limitée par la dimension des entrées et la dimension des cartes. Ici s'ajoutent à la dimension des entrées la dimension d'une carte et le nombre de carte. Il sera difficile de représenter graphiquement des architectures de plus de trois cartes, et encore plus lorsque les entrées sont en grande dimension. Cette difficulté de représentation soulève la nécessité de définir des valeurs indicatrices du fonctionnement de la carte, calculables en grande dimension.

Mais, qu'est ce qu'une carte *qui fonctionne*? L'intérêt de CxSOM réside dans la communication entre cartes. Représenter les cartes une à une laisse donc de coté leur connexion. Il est donc nécessaire de trouver un moyen de représenter l'architecture comme un tout. On s'intéressera notamment à comment l'architecture de carte est capable de représenter les relations entre les entrées multimodales.

Ce chapitre questionne donc la façon de représenter une carte de Kohonen, et plus particulièrement la façon de représenter une carte au sein d'une architecture. Nous présenterons donc en premier lieu un formalisme pour la carte et les entrées multimodales associées, et à partir de ce formalisme nous proposerons plusieurs représentations et indicateurs cherchant à comprendre ce que l'architecture apprend sur les données d'entrée, et de quelle façon.

## 3.2 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.

### 3.2.1 Représentation des entrées

Les observations multimodales que l'on cherchera à apprendre par l'architecture de cartes sont notées  $X^i, i = 0 \dots N$  où  $N$  est le nombre de modalités considérées. Lors de l'apprentissage et du test, elles sont échantillonnées ; ainsi, à chaque pas de temps, l'architecture se voit présentée un vecteur  $(X_t^0, \dots, X_t^N)$ . Lorsqu'elles sont considérées en tant que *entrée externe* d'une carte, on les notera plutôt  $\xi^{(i)}, i = 0 \dots N$ , avec  $i$  l'indice de la carte dont  $\xi^{(i)}$  est l'entrée. Pour tout  $i$ ,  $X^i$  et  $\xi^{(i)}$  sont des variables aléatoires, et  $\mathbf{X} = (X^0, \dots, X^N)$  et  $\xi = (\xi^{(0)}, \dots, \xi^{(N)})$  sont les vecteurs aléatoires correspondants.

Pour les entrées CxSOM, on s'intéresse à l'apprentissage de relations entre entrées. Les variables  $X^i$  ne sont a priori donc pas des variables indépendantes. Afin de mieux comprendre comment les cartes apprennent des relations entre les entrées, on introduit une autre variable aléatoire  $U$ . Cette variable est multidimensionnelle et est choisie de façon à ce que chaque variable  $X^i$  soit une fonction quelconque de la variable aléatoire  $U$ , et uniquement de cette variable. Il s'agit en fait d'une réduction de dimension :

$$\forall t, \forall i, X_t^i = f_t^i(U_t) \quad (3.1)$$

Cette variable traduit l'existence d'un modèle reliant les observations. Prenons un exemple géométrique ; considérons des points tirés sur un cercle quelconque dans l'espace en deux dimensions.  $\mathbf{X} = (X^0, X^1)$ , les coordonnées cartésiennes des points du cercle, est alors une vecteur aléatoire, dont les composantes sont les variables aléatoires  $X^0, X^1$ . En définissant une variable  $U$  à valeurs réelles, chaque point du cercle peut maintenant s'écrire, selon l'équation paramétrique du cercle :

$$\begin{cases} X_t^0 = r \cos(U_t) \\ X_t^1 = r \sin(U_t) \end{cases} .$$

$U$  représenterait ici l'angle du point sur le cercle.  $U$  est une variable cachée qui *réduit la dimension* du modèle. Elle contient toute l'information sur l'échantillon.

$U$  et  $f^i$  ne sont pas uniques. Elles 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 peut aussi définir

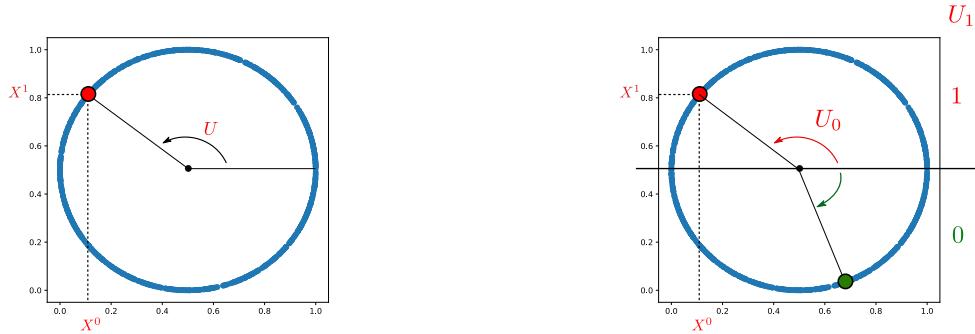


FIGURE 3.3 – Exemples de paramétrisations du cercle. La paramétrisation qui traduit le plus facilement le modèle est naturellement celle dans laquelle  $U$  est à valeurs réelles. Le modèle auxquelles appartiennent les modalités  $X^0$  et  $X^1$  est donc représenté par la variable cachée  $U$ .

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 côté de l'axe des abscisses on se situe.

Notons par contre que la plus petite dimension possible de  $U$  dépend du degré de liberté du modèle. Si toutes les observations se situent sur une courbe de dimension 1, alors il existe une variable  $U$  en une dimension satisfaisant l'équation 3.1. Si les observations se situent sur une surface de dimension 2, alors,  $U$  sera en deux dimensions, et ainsi de suite.

### 3.2.2 Représentation des éléments des cartes

Comme dans de nombreux algorithmes d'apprentissages, on peut décomposer le jeu de données d'entrée 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 la partie mise à jour des cartes de Kohonen n'est plus opérée. Dans le cadre des variables aléatoires, chaque itération est alors un tirage indépendant. Les éléments des cartes peuvent donc être considérés comme des variables aléatoires et une itération de test comme la 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 3.4.

Nous considérerons alors plusieurs éléments des cartes en tant que variables aléatoires, notamment :

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

Notons que tout élément d'une carte pourrait être vu de cette manière. Une phase de test est donc un grand nombre de réalisations d'une variable aléatoire jointe :

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

Les composantes de cette variable jointe ne sont pas indépendantes. Les représentations et indicateurs présentés ensuite chercheront à détecter et comprendre au mieux ces dépendances statistiques.

Ainsi, dans ce formalisme par variable aléatoires, à chaque pas d'apprentissage peut-être associé un ensemble de réalisations de variables aléatoires. Ceci permet alors d'utiliser des outils et métriques issus de la théorie de l'information pour qualifier l'organisation des cartes au sein de l'architecture. Cette approche ne se limite pas à l'architecture CxSOM :

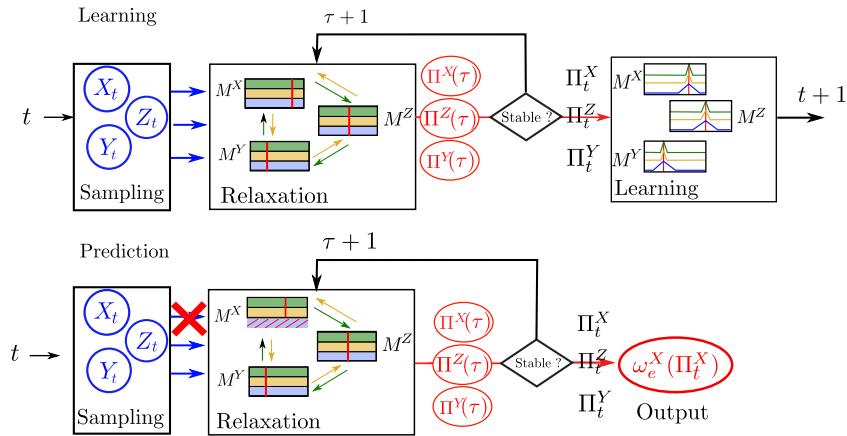


FIGURE 3.4 – Schéma descriptif de l'apprentissage et des tests.

### 3.3 Représentations graphiques

*Qu'est ce qu'une carte a appris des données ?*

Que cherche t-on à représenter dans les représentations classiques des poids des cartes de Kohonen, que ce soit sous la forme d'un tableau de prototypes ou d'une projection dans l'espace d'entrée ? On cherche en fait à visualiser comment les poids sont répartis en fonction de leur *position* dans la carte. En d'autres termes, on cherche à comprendre si les positions de la carte correspondent à tous les éléments de l'espace d'entrée, si une continuité est réalisée. D'une façon similaire, on peut faire le choix de représenter le poids de la best matching unit par rapport à sa position. Cela donne la même représentation que le fait de tracer le poids de chaque prototype par rapport à sa position dans la carte ; à la seule différence qu'elle fera la distinction entre les *unités mortes de la carte*, c'est à dire les unités qui ne sont jamais best matching unit et qui ne seront donc pas affichée dans la représentation des tests, et les autres. Cette représentation prend en compte la façon de calculer le BMU, donc le coeur de l'algorithme.

La question de la répartition des valeurs d'une carte par rapport à la position de leur BMU va plus loin que les poids : il est intéressant d'étudier la répartition de n'importe quel élément d'une carte de cette façon, afin de comprendre De façon plus générale, on peut représenter, à partir d'un échantillon test, la dépendance de n'importe quelle variable par rapport à la position de la best matching unit correspondante. Nous détaillerons dans cette partie quelques représentations qui paraissent pertinentes.

#### 3.3.1 Représenter les entrées par rapport à une carte

Une première représentation d'une carte est de tracer la valeur de son entrée  $\xi^{(i)}$  par rapport à la position du BMU. 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, on peut tracer sur une même figure les entrées de toutes les cartes selon la position du BMU d'une des cartes.

#### 3.3.2 Représentation de U par rapport au BMU

*Chercher à apprendre des relations entre les données*

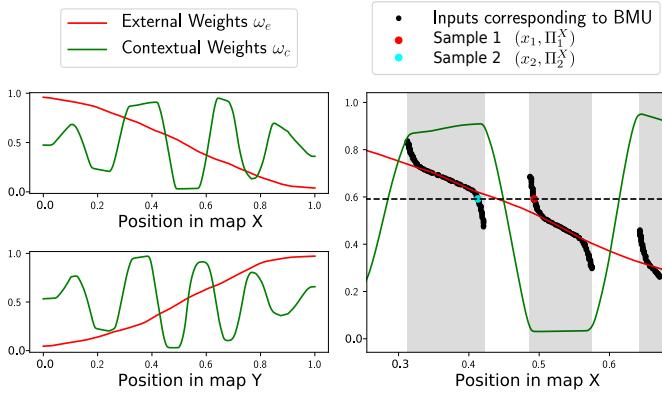


FIGURE 3.5 – A gauche, les poids externes et contextuels dans une architecture de deux cartes connectées réciproquement. Chaque carte a donc un poids externe et un poids contextuel. A droite, on a tracé en supplément la valeur de l’entrée (scalaire) de la première carte, en fonction de la position du BMU correspondant. Cette représentation fait apparaître les unités mortes de la carte et la façon dont les entrées sont projetées sur les poids.

Pour une, deux, trois entrées, les relations entre entrées se déduisent assez directement. Lorsqu’on augmente la dimension, il paraît pertinent de dégager des nouvelles valeurs qui représentent le modèle : il s’agit ici de la variable  $U$ . Cette variable cachée est en fait une représentation du modèle en dimension plus faible, par une transformation non linéaire. On peut alors tracer  $U$  en fonction de la position  $\Pi$  du BMU d’une carte.

### 3.3.3 Dépliement d'une carte en plusieurs dimensions

Nous proposons dans cette thèse une façon de représenter les poids d'une carte de Kohonen au sein d'une architecture. Cette représentation est créée à partir des échantillons de test. Il s'agit de tracer les poids des BMUs ( $\omega_e(\Pi^{(1)}), \dots, \omega_e(\Pi^{(k)})$ ) dans l'espace en  $k$  dimensions correspondant -  $k$  correspondant ici à 2 ou 3 dimensions. Les échantillons sont ensuite reliés suivant l'ordre des positions dans *une des cartes*. On obtient ainsi le *dépliement* d'une carte de l'architecture dans l'espace multimodal à plusieurs dimensions. Un exemple de carte ainsi dépliée est présenté en figure 3.8.

Ces figures sont équivalentes à tracer une carte dans l'espace de ses entrées : les poids des BMUs de l'échantillon sont les prototypes des cartes ; seuls les poids des unités mortes ne sont pas représentés. Cette représentation est limitée par la dimension des entrées, ces dernières devant représenter au total moins de trois dimensions. Cette représentation peut malgré tout être étendue : il est possible de tracer le dépliement de la carte selon un *élément* de chacune des modalités. Par ailleurs, l'étude du comportement de cartes sur des données 3D s'inscrit dans la démarche de construction d'un modèle que nous suivons dans cette thèse. Leur visualisation est alors un élément clé dans la compréhension des comportements possibles de l'architecture. A partir de cette visualisation, on peut envisager de construire des indicateurs permettant l'analyse de l'architecture en dimension supérieure. Le second avantage de ces tracés est qu'il est possible de représenter graphiquement une carte qui ne prend pas d'entrée externe, ou de représenter une carte dans l'espace des poids d'autres cartes.

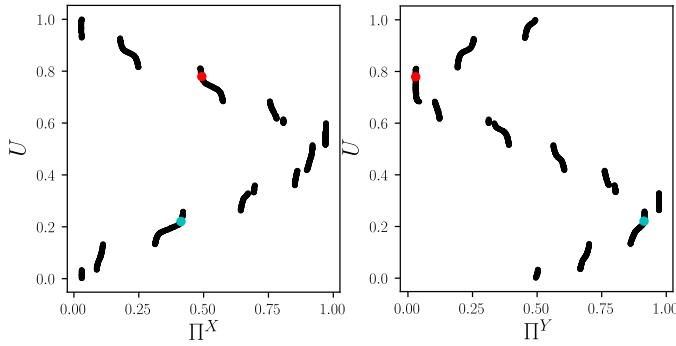


FIGURE 3.6 – Pour l'échantillon de test, valeur de  $U$  en fonction des valeurs du BMU  $\Pi$  dans chacune des cartes. On voit que  $U$  est une fonction du BMU dans chaque carte, contrairement au cas où les cartes apprendraient indépendamment sur les mêmes entrées, voir figure 3.7.

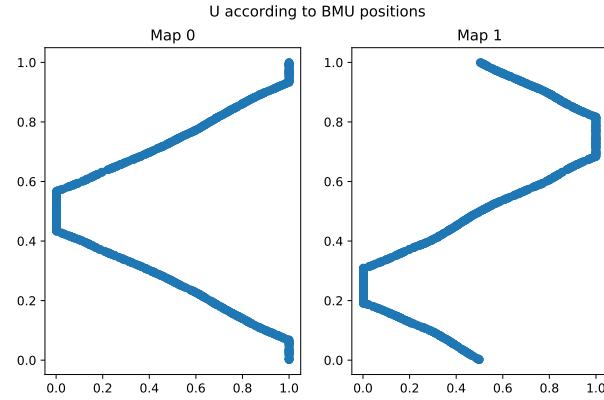


FIGURE 3.7 – Pour l'échantillon de test, entrée sur un cercle, valeur de  $U$  en fonction des valeurs du BMU  $\Pi$  dans chacune des cartes, lorsque les cartes  $M_x$  et  $M_y$  ne sont pas connectées. Chacune des cartes n'a aucune information de plus que celle portée par son entrée sur l'état global du système  $U$ , et  $\Pi$  n'est donc pas une fonction de  $U$  dans chaque carte.

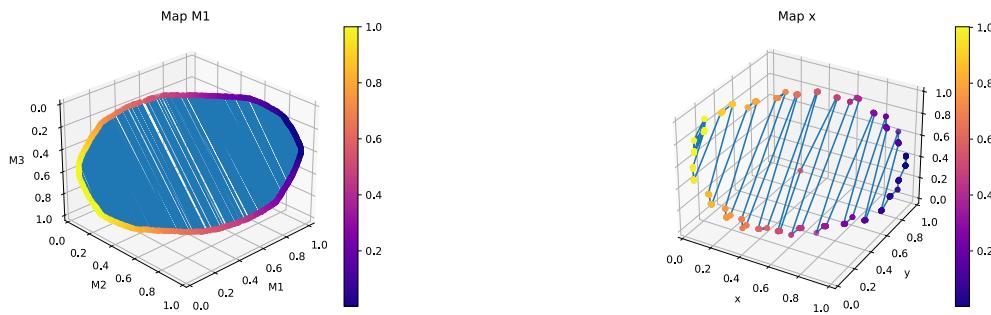


FIGURE 3.8 – Représentation des poids finaux de trois cartes prenant en entrée  $X, Y$  et  $Z$ , reliés selon les positions de la carte  $X$ . À gauche, les cartes de l'architecture ont appris séparément sur les données. À droite, disposition lorsque les cartes ont été connectées au sein d'une architecture. Un échantillon de 1000 points a été utilisé pour les tracés.

## 3.4 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 [5] 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 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.

La théorie de l'information est non seulement applicable, mais en fait profondément liée aux algorithmes l'apprentissage automatique [4]. L'informatique prend sa source dans les travaux de recherche en la théorie de l'information au XXème siècle. L'apprentissage automatique en est un aspect. Ces algorithmes cherchent en effet à apprendre un modèle, un signal liant des entrées à des sorties, ou des entrées entre elles. Ce modèle est un encodage du signal d'entrée. De son côté, la théorie de l'information apporte un langage fondamental pour quantifier la dépendance entre signaux, les encoder et les décoder. Les algorithmes d'apprentissage étant fondamentalement des encodeurs et des décodeurs, les quantités issues de la théorie de l'information ont donc une traduction directe dans les modèles d'apprentissage, et inversement. D'ailleurs, de nombreux modèles d'apprentissage automatique se reposent directement sur des règles de calcul d'information pour encoder le signal d'entrée. On pense notamment aux approches bayésiennes de l'apprentissage qui estiment des distributions, et passent par des calculs d'information pour les estimer.

L'analyse d'un système d'apprentissage via la théorie de l'information peut donc avoir deux approches :

- Utiliser des quantités estimables pour quantifier les relations entre entrées et sorties, et ainsi en tirer des conclusions en terme d'encodage et de pertes. Par exemple, si l'information portée par la sortie sur l'entrée est élevée, l'encodage est bien réalisé.
- Abstraire le système d'apprentissage dans le modèle mathématique de la théorie de l'information.

Nous investiguerons dans cette partie la première approche, soit, comment quantifier le système apprenant 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 répondrons donc à ces questions :

- Quelle quantité cherche-t-on à mesurer dans l'architecture de cartes ? Cette quantité dépend de la fonction qu'on cherche à réaliser.
- Peut-on l'estimer, et comment ?

### 3.4.1 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 aléatoire discrètes. L'entropie de Shannon d'une variable aléatoire  $X$ , de distribution  $p(X)$ , est notée  $H(X)$  et définie par la formule :

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

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 peuvent être mesurées par *l'information mutuelle*. Elle se définit formellement par :

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

Cette valeur mesure la quantité d'information moyenne apportée par une réalisation de  $X$  sur la réalisation de  $Y$ . L'information mutuelle possède notamment les propriétés suivantes :

$$I(X, Y) = 0 \Leftrightarrow X \text{ et } Y \text{ sont indépendantes}$$

L'information mutuelle est donc une mesure de la distance entre la distribution jointe de  $(X, Y)$  et leur indépendance.

Elle s'exprime à partir de l'entropie :

$$I(X, Y) = H(X) + H(Y) - H(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Elle est symétrique :

$$I(X, Y) = I(Y, X)$$

Pour toute fonction  $f$ ,  $I(X, Y) \geq I(X, f(Y))$ . L'égalité est atteinte si  $f$  est *bijective*.

### 3.4.2 Indicateur : coefficient d'incertitude.

Lors de l'analyse de CxSOM, l'information que portent les positions des BMUs d'une carte sur le modèle d'entrées est une valeur intéressante. Les éléments de la carte ont été définis en terme de variables aléatoires ; l'information mutuelle peut alors être une représentation pertinente 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  $U$ . Dans ce sens,  $I(\Pi, U)$  est l'information moyenne que le BMU d'une carte porte sur  $U$  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.

**Propriété 1.** *La valeur maximale atteinte par  $I(\Pi, U)$  est  $H(U)$ , atteinte lorsque  $U$  est fonction de  $\Pi$ .*

*Démonstration.* Par construction,  $\Pi$  est une fonction de  $U$  dans une carte de Kohonen. En effet, notre algorithme est déterministe et une sortie est définie pour toute valeur de  $U$ . Par propriété de l'information mutuelle, pour toute fonction  $f$  et variable  $X, Y$ ,  $I(X, f(Y)) \leq I(X, Y)$ . 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, ssi  $U$  est aussi une fonction de  $\Pi$ .

□

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

$$UC(U|\Pi) = \frac{I(\Pi, U)}{H(U)} \quad (3.2)$$

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,  $UC \in [0, 1]$ .

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

### 3.4.3 Estimation

L'information mutuelle et l'entropie sont des grandeurs probabilistes. Elles sont définies à partir de la distribution des variables aléatoires. Lorsque qu'on ne connaît pas les distributions, il est nécessaire d'estimer ces valeurs autrement. Une première façon d'estimer l'entropie et l'information mutuelle entre  $X$  et  $Y$  est d'estimer la distribution des variables  $X, Y$  et leur distribution jointe  $Z = (X, Y)$  en discrétilisant l'espace par *binning*, 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  $Z = (X, Y)$ . L'information mutuelle par binning est calculée à partir de ces distributions discrètes. Lorsque la dimension des variables est faible (typiquement 1D), l'estimateur est ?? (fiable ?? comment). Des termes de corrections peuvent éventuellement être apportés. La précision de l'estimation peut être améliorée en choisissant des tailles de boîtes variables. Cependant, 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 théoriquement pu contenir ; l'estimation de la probabilité en ce point sera donc nulle, et l'indicateur faussé.

Un autre estimateur que le binning est l'estimation par noyaux de Kraskov [3]. Cet estimateur passe par une estimation directe de l'entropie au lieu des densités de probabilités. Le découpage de l'espace se fait en recherchant, pour un couple  $(X, Y)$  les  $k$  plus proches voisins. Une information mutuelle locale est calculée dans cette zone de l'espace, suivant une formule permettant d'approximer la somme de logarithmes par la fonction digamma :

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

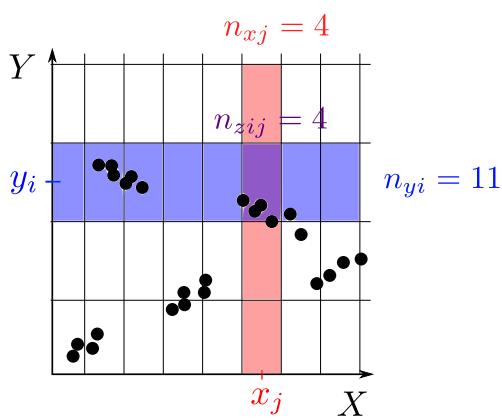


FIGURE 3.9 – Procédé de binning pour estimer les distributions des variables  $X$  et  $Y$ . Les distributions sont estimées à partir de  $n_{xj}$ ,  $n_{yi}$  et  $n_{zij}$ , puis les valeurs de  $H$  et  $I$  calculées.

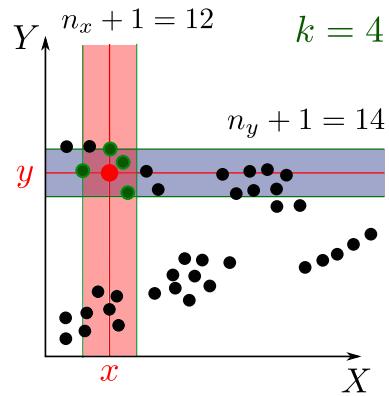


FIGURE 3.10 – Découpage en KNN de Kras-kov pour estimer l'entropie et l'information mutuelle des variables  $X$  et  $Y$ . Les plus proches voisins du point rouge sont trouvés, en vert, et le processus est répété sur tous les points. Les valeurs de  $n_x$  et  $n_y$  permettent d'estimer directement l'entropie.

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

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

Cet estimateur est meilleur que le binning, car il évite les boîtes vides du binning, car on n'explorera que l'espace des points. Il est donc préférable d'utiliser cet estimateur en plus grande dimension. Nous utiliserons cet estimateur par la suite.

L'estimation de  $UC(X|Y)$  nécessite d'estimer  $I(X, Y)$  et  $H(Y)$ .

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. On peut notamment citer [1], qui utilise une approche neuronale pour estimer l'information mutuelle. Ainsi, l'utilisation du coefficient d'incertitude comme indicateur semble réutilisable pour des données de plus grande dimension ou pour plus de cartes, quitte à utiliser des méthodes d'estimations plus élaborées.

### 3.4.4 Perspectives

#### Erreurs sur les données bruitées

Le coefficient d'incertitude mesure une forte relation statistique entre les données. Cela mesure bien ce qu'on apprend entre cartes ; cependant, il ne prend pas en compte l'aspect continu des variables. Ainsi, prenons une distribution  $X$  quelconque et deux distributions  $Y_1$  et  $Y_2$ , telles que représentées en figure 3.12.  $Y_1$  et  $X$  ne sont pas en bijection, mais ont une forte dépendance statistique : lorsqu'on connaît la valeur de  $X$ , seules deux valeurs sont possibles pour  $Y_1$ .  $Y_2$  et  $X$  ne sont pas en bijection, mais sont proches : leur relation est en fait une bijection, mais avec du bruit. Lorsqu'on analyse des données de CxSOM, les entrées sont généralement bruitées, tout comme les sorties. Par contre, on voudrait privilégier l'existence d'une dépendance forte entre entrées et sorties qui ne tiennent pas compte de ce bruit. Or, dans l'exemple cité,  $UC(Y_1|X) = 0.6$  et  $UC(Y_2|X) = 0.4$ . L'indicateur n'est donc pas complètement approprié dans le cas de variables avec bruit.

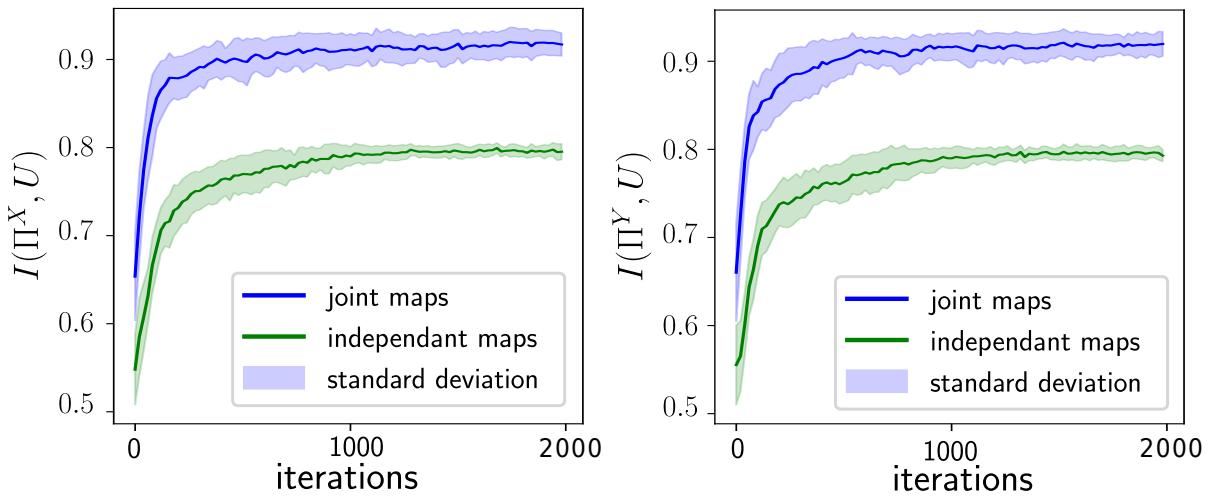


FIGURE 3.11 – Evolution de l’indicateur relatif à l’information mutuelle entre  $\Pi$  et  $U$  dans chaque carte au cours de l’apprentissage. Cet indicateur est comparé à celui calculé dans le cas où les cartes apprennent séparément.

## Discussion

### Correlation Ratio

## 3.5 Prédiction d’entrée

Au sein d’une architecture de cartes, il est possible de ne pas présenter à une ou plusieurs cartes de l’architecture leur entrée externe  $\xi^{(i)}$ . Dans ce cas, une best matching unit peut quand même être calculée par leurs entrées contextuelles. Le poids de cette best matching unit peut alors être vu comme une prédiction de l’entrée manquante. Cette capacité de prédiction peut être à la fois vue comme une application possible de l’architecture, mais aussi comme une façon de représenter *ce que les autres cartes connaissent d’une autre*. Tracer les prédictions d’une carte est donc un indicateur de la façon dont une architecture a appris des relations.

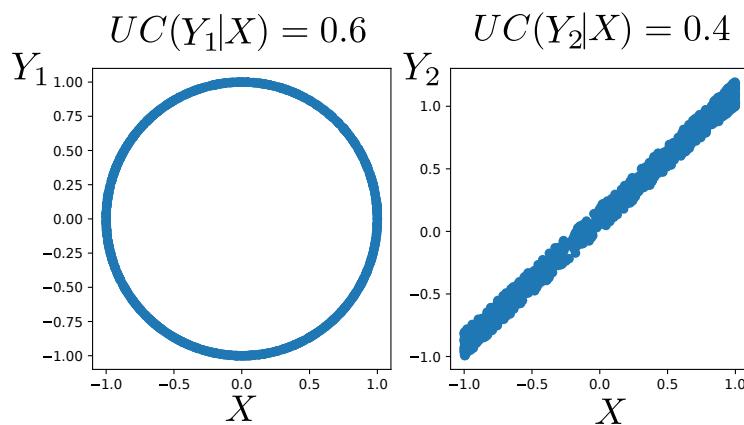


FIGURE 3.12 – Pour ces deux distributions  $X$  et  $Y$ , le coefficient d'incertitude mesure une meilleure dépendance statistique dans le premier cas que le second. On voudrait au contraire privilégier le score de la seconde relation.

## Chapitre 4

# Expériences : analyse provisoire et liste

Cette section présente les différentes expériences réalisées afin de mieux comprendre l'architecture CxSOM. C'est une section provisoire, elle permet de mettre en relation les expériences afin d'ensuite tirer des conclusions claires a propos de l'architetecture CxSOM. En utilisant les indicateurs et tracés présentés dans le chapitre précédent, nous détaillons dans cette partie les résultats obtenus et les pistes d'interprétation.

Les jeux de données présentés aux cartes de l'architecture sont des nuages de points en deux ou trois dimensions, tirés selon une distribution. Les données d'entrée à l'architecture de carte seront notées  $X, Y, Z$ , sauf mention contraire. Ces trois valeurs sont scalaires, sont les coordonnées d'un point du nuage de point, et correspondent chacune à l'entrée d'une des cartes de l'architecture.

Dans cette version, les notations ne sont pas encore homogènes : l'autrice s'en excuse d'avance. Notez donc que : les modalités  $X^{(i)}$  sont ici notées  $I^i$ . L'entrée  $I^i$  est toujours assignée à la carte  $M^i$ . Lorsqu'on utilise la variable  $M^i$ , il s'agit de la position du BMU  $\Pi^{(i)}$ .

### 4.1 Entrées sur un cercle

Une architecture de deux cartes, connectées mutuellement, est l'exemple le plus simple d'architecture CxSOM. Son étude permet de dégager des comportements facilement représentables d'un point de vue graphique et ayant peu de liberté possible.

#### 4.1.1 Cercle 2 dimensions

Prenons en entrée  $(X, Y)$  situé sur un cercle de centre 0.5 et de rayon 0.5, tel que  $X, Y \in [0, 1]$ . Les entrées de chacune des cartes sont donc dépendantes. Nous regarderons la distribution des valeurs selon les BMU de chaque carte.

#### 4.1.2 Cercle trois dimensions

Les entrées sont  $(X, Y, Z)$ , trois cartes connectées mutuellement. Les entrées sont dépendantes de telle sorte à ce que connaître l'une d'entre elle laisse 50% d'erreur sur les deux autres, mais deux d'entre elles permettent de déterminer totalement la troisième.

En figure 4.2

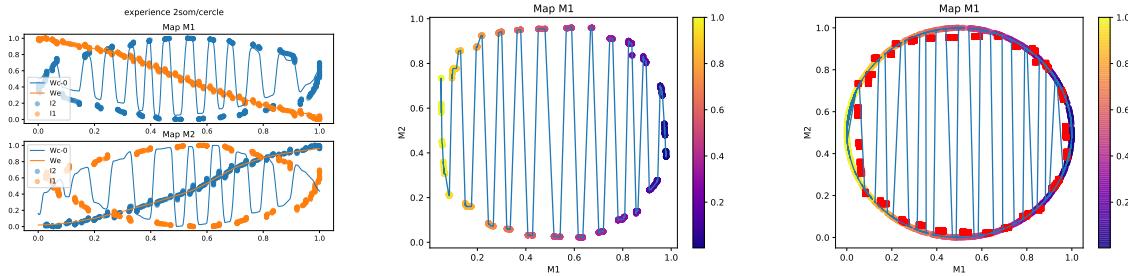
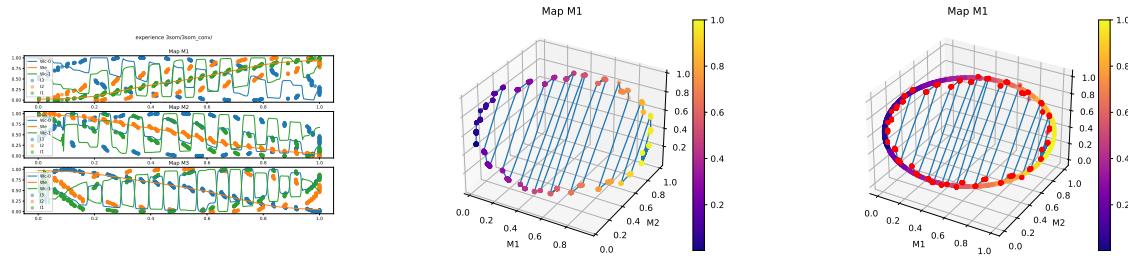


FIGURE 4.1 – Tracé des poids de M1 et M2, dépliement des poids de M1 dans l'espace 2D et dépliement des entrées



#### 4.1.3 Entrées dans un anneau

On peut, au lieu de considérer un cercle, ajouter du bruit à chaque dimension. Les entrées sont alors tirées dans un fin anneau ou tore. Dans ces conditions, le modèle peut toujours être considéré comme un cercle avec bruit. On peut donc prendre  $U$  en une dimension :

$$\begin{cases} X_t = r \cos(U_t) + \epsilon_X \\ Y_t = r \sin(U_t) + \epsilon_Y \end{cases} .$$

Cependant, on peut aussi considérer que le modèle n'est plus réductible en une seule dimension cachée. Il est donc intéressant de regarder comment les cartes considèrent le bruit : a-t-on toujours une quantification vectorielle du cercle, ou la carte se déplie t-elle autrement ? Autrement dit, une carte sépare t-elle le bruit de la forme des entrées ?

	I3	I2	U	I1	M1	M2	M3
I3	1.000000	0.547491	0.739482	0.557418	0.591741	0.596191	0.582069
I2	0.547491	1.000000	0.738534	0.608069	0.576468	0.564915	0.600864
U	0.739482	0.738534	1.000000	0.739812	0.749590	0.746406	0.765705
I1	0.557418	0.608069	0.739812	1.000000	0.566214	0.580723	0.599703
M1	0.714445	0.698738	0.907380	0.683431	0.994537	0.886474	0.885367
M2	0.720026	0.685787	0.905187	0.705475	0.897010	0.999703	0.862628
M3	0.690424	0.713113	0.904939	0.713265	0.858226	0.847641	0.998612

FIGURE 4.2 – Valeurs de  $UC$  entre les différents éléments des cartes. Pour un élément  $i, j$  du tableau est indiqué  $UC(i|j)$  ( $M_i$  correspond à  $\Pi^{(i)}$ ). Le tableau sera mis au propre s'il est pertinent. Estimation par la méthode de Kraskov, voisinage de 7. Ce voisinage est arbitraire, il produit une variance plus grande mais biais moins important. Il faudrait refaire l'expé sur de multiples échantillons pour avoir une valeur sûre.

## 4.2 Entrées dans un carré

### Expérience

Prenons en entrée  $(X, Y) \in [0, 1]^2$ , tirés selon une distribution uniforme. Les entrées de chacune des cartes sont donc indépendantes. Nous regarderons la distribution des valeurs selon les BMU de chaque carte. Cette distribution de valeurs dépendra donc uniquement de l'architecture, étant données que les entrées n'ont aucun lien entre elles. Cela nous permet de visualiser quels comportements sont uniquement issus du modèle.

### Résultats

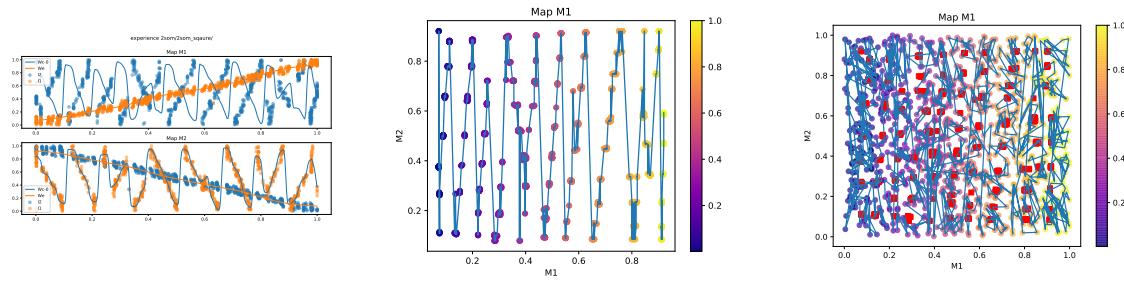


FIGURE 4.3 – Tracé des poids de M1 et M2, dépliement des poids de M1 dans l'espace 2D et dépliement des entrées

Cette expérience montre que la disposition en vagues des cartes est un comportement lié à l'architecture et non aux données en elles même : quelles que soient les données, la carte effectuera toujours une séparation en indices primaires et secondaires. Ici, pour un X donné, on a toutes les valeurs possibles pour Y. Cependant, les poids sont un peu différent de l'expérience avec X, Y sur un cercle : les bmus au sein d'une zone secondaire sont répartis tout le long de cette zone, alors que pour un cercle ils sont centrés dans la zone, car peu de valeurs possibles.

## 4.3 Entrées en clusters

On considère des entrées  $X, Y$  distribuées dans  $[0, 1]^2$ , telles que les points sont regroupés autour de 6 centres. Autour de ces centres, les points sont tirés aléatoirement dans un cercle. On a donc des clusters de points. La distribution de  $(X, Y)$  peut être considérée comme fonction des centres des clusters. A-t-on un regroupement des points des clusters dans la carte, ou la carte se déplie t-elle sans prendre en compte les regroupements ? Nous présentons ici deux expériences, chacune prenant des entrées distribuées selon des tailles différentes de clusters autour de leurs centres, tracées en figure 4.4.  $U$  est ici considéré comme l'indice (entre 0 et 1) du cluster.



FIGURE 4.4 – Entrées présentées aux cartes dans chaque expériences. A gauche, "petits" clusters, à droite, "grands" clusters

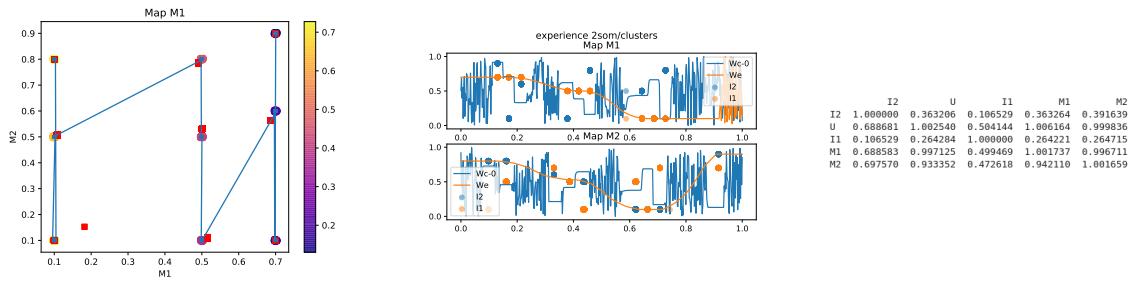


FIGURE 4.5 – Dépliement, poids et information mutuelle des cartes.

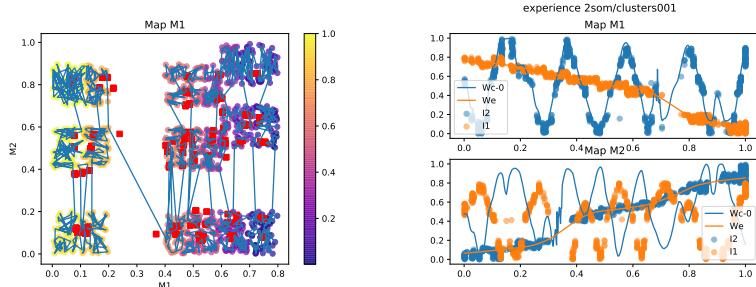


FIGURE 4.6 – Dépliement, poids et information mutuelle des cartes.

#### 4.3.1 "petits" clusters

#### 4.3.2 "gros" clusters

### 4.4 Influence de l'architecture sur des entrées

Sur des entrées identiques, on testera différentes architectures afin d'évaluer le rôle des connexions. On utilisera toujours comme entrée dans cette section un cercle en deux ou trois dimensions.

#### 4.4.1 Cartes intermédiaires

mettre line 4 som, etc. information mutuelle dans ce cas :) comparer a la version une carte intermédiaire. Tableau info mutuelle : faire les comparaisons. Comparaison des entrées, ce sont a

peu près les mêmes donc on doit avoir les memes valeurs ok, donc ce qui nous intéresse c'est la comparaison entre M1 M2 M3 M4 etc.

Dans cette expérience, les entrées sont  $X, Y$  ou  $X, Y, Z$ , sur un cercle en 2D ou 3D ; l'architecture est composée de 3 cartes, une prenant  $X$ , une prenant  $X$  et une connectée aux deux autres ne prenant que les BMUs en entrée. Les cartes  $X$  et  $Y$  ne sont pas directement connectées. On testera l'architecture avec une et deux cartes intermédiaires pour la version 2D (figure 4.7). Pour la version 3D, on a une carte centrale.

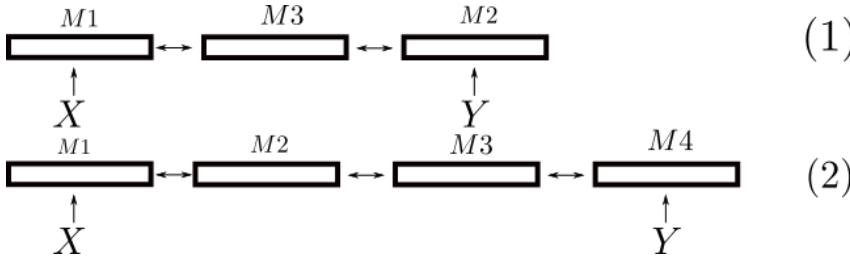


FIGURE 4.7 – Architectures avec une (1) et deux (2) cartes intermédiaires sans entrées, présentées dans cette section.

	I2	U	I1	M1	M2	M3		I4	U	I1	M1	M2	M3	M4	
I2	1.000000	0.706968	0.502862	0.504594	0.542851	0.545108		I4	1.000000	0.712408	0.499305	0.396878	0.397598	0.484030	0.688904
U	0.706968	1.000000	0.713136	0.679871	0.692786	0.721094		U	0.712377	1.000000	0.705074	0.602633	0.549592	0.532868	0.569440
I1	0.502862	0.713136	1.000000	0.592889	0.526895	0.541757		I1	0.499305	0.705074	1.000000	0.600599	0.511068	0.384278	0.366529
M1	0.644014	0.867133	0.758627	0.995997	0.731338	0.772175		M1	0.529678	0.804086	0.982251	1.002759	0.697677	0.393710	0.364293
M2	0.684167	0.873095	0.665419	0.720377	1.002729	0.854391		M2	0.669543	0.931094	0.859126	0.880271	0.999815	0.485324	0.473787
M3	0.707711	0.938401	0.704817	0.787571	0.883355	1.003478		M3	0.826151	0.907546	0.656624	0.505375	0.487672	0.995807	0.829614
								M4	0.965341	0.797884	0.515359	0.378048	0.392343	0.685453	1.002398

FIGURE 4.8 – coefficients d'incertitude entre les éléments des cartes, pour l'architecture (1)(gauche) et (2)(droite). Le nombre de voisins utilisés pour l'estimation est 10. L'estimation est réalisée sur 1000 échantillons.  $I1$  est l'entrée de la carte  $M1$  et correspond à  $X$ ,  $I2$  ou  $I4$  est l'entrée de la carte 2 ou 4 et correspond à  $Y$ .

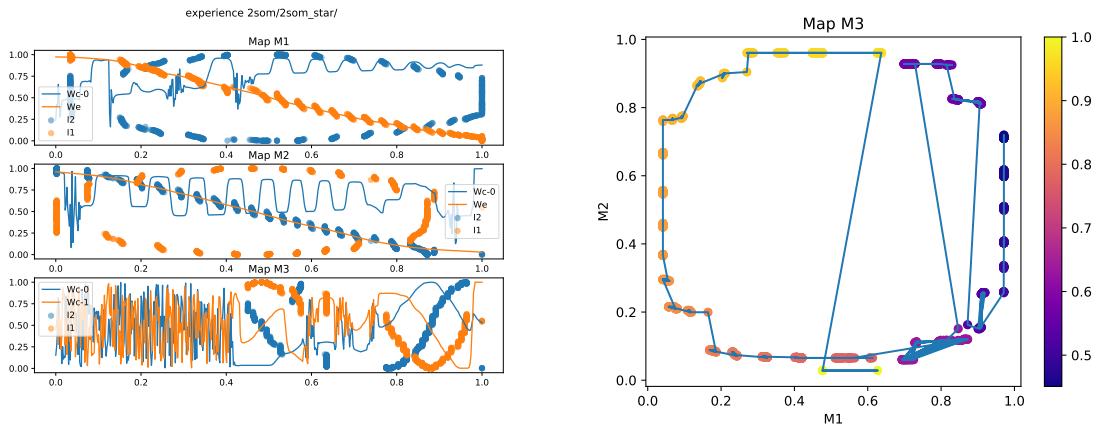


FIGURE 4.9 – Poids et dépliement des deux cartes lorsqu'elles sont connectées via une carte intermédiaire (architecture 1).

Les coefficients d'incertitudes sont ici un élément de comparaison particulièrement bon entre les cartes. Les relations entre les entrées  $X, Y$  sont identiques, on a en effet la même distribution. On peut porter notre attention sur  $UC(I1|M1)$  dans les deux cas : ce coefficient est plus élevé

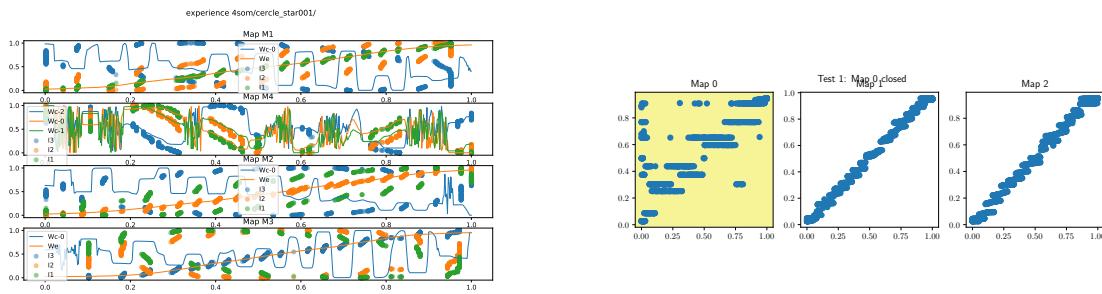


FIGURE 4.10 – Poids et dépliement de trois cartes connectées avec une intermédiaire.

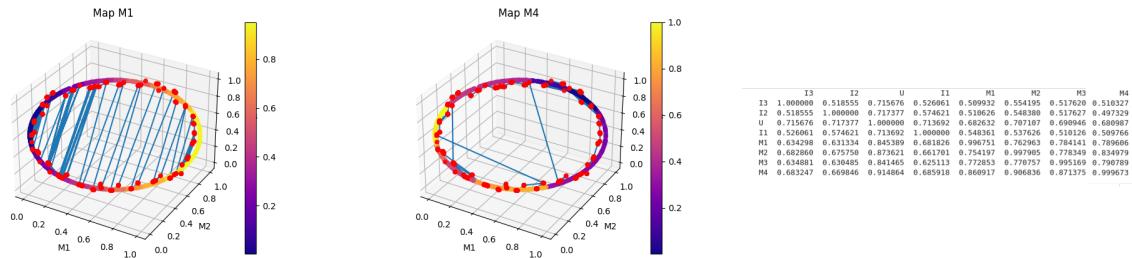


FIGURE 4.11 – Poids et dépliement de trois cartes connectées avec une intermédiaire et information mutuelle. Les cartes 1,2,3 reçoivent des entrées, la carte 4 les connecte.

dans le cas 2. Cela traduit une meilleure quantification vectorielle par les cartes ayant une entrée externe lorsqu'on met 2 cartes intermédiaires qu'une seule. On se rapproche en fait d'une architecture dans laquelle les cartes ne seraient pas connectées. Il semblerait donc que seules les connexions directes soient vraiment utiles. Par contre, on peut regarder  $UC(U|M3)$  pour (1), ou  $UC(U|M2)$  et  $UC(U|M3)$  pour (2). Dans le cas (1) : le coefficient est de 0.93, vs 0.9 sur les deux cartes ayant une entrée externe, donc la différence n'est pas si significative. Les trois cartes ont une information sur le modèle. Dans le cas (2) : le coefficient est de 0.9 sur M2 et M3, vs 0.8 sur les cartes du bout. Il semble donc que les cartes intermédiaires ont appris une représentation de la variable cachée.

On peut maintenant s'intéresser au dépliement des cartes, et à leur capacité de prédiction dans le cas en 3 dimensions. ( La prédiction n'étant en effet pas possible dans le cas de deux cartes). A trois cartes, la prédiction semble moins bonne que lorsque les trois cartes sont directement connectées. Si on regarde les coefficients d'incertitude, on remarque que les BMUs de la carte 4, qui connecte les trois autres, ont une info de 0.9 sur U et sur les bmus des trois autres cartes. Elle possède également un peu plus d'infos sur I1, I2, I3 que les cartes séparées. Donc, d'un coté la carte centrale possède les infos sur le modèle, mais d'un autre coté l'architecture effectue mal la prédiction lorsqu'une entrée manque.

On peut donc imaginer utiliser des cartes centrales dans ce genre d'architecture pour résumer les trois autres. Ainsi, on pourra travailler sur l'info d'une seule des cartes au lieu de trois.

**4.4.2 Et si on a la même architecture, avec une carte centrale, mais cette fois les cartes M1, M2, M3 sont également connectées ?**

**4.4.3 Boucle vs rétroaction à trois cartes**

On a vu avec les cartes intermédiaires que les connexions lointaines semblent jouer un rôle moins important dans la dynamique que les connexions directe. Ainsi, on peut regarder comment se comporte une architecture de trois cartes connectées en boucle.

**4.4.4 Conclusion et perspectives : influence de l'architecture**

## **4.5 Influence des paramètres des cartes**

Dans cette section,

**4.5.1 rayons de voisinage**

**4.5.2 Combinaison des activités externes et contextuelles par moyenne ou moyenne géométrique**

**4.5.3 Influence de la relaxtion**

**4.6 Conclusion des expériences, perspectives et limites**

## **Conclusion**



# Bibliographie

- [1] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, R. Devon Hjelm, and Aaron C. Courville. Mutual information neural estimation. In *ICML*, 2018.
- [2] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning : A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35 :1798–1828, 2013.
- [3] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6), Jun 2004.
- [4] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.
- [5] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27 :379–423, 1948.