

peut-être dire qu'en dehors de l'apprentissage et de sa mesure (comme d'après la fonctionnalité sur dimension manquante) c'est la dynamique de l'architecture que l'on souhaite étudier ?

## Chapitre 3

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

### Sommaire

---

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

---

### 3.1 Introduction

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

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

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

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

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

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

### 3.1.1 Présentation de l'expérience d'illustration

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

paris  
évoque  
du  
parag

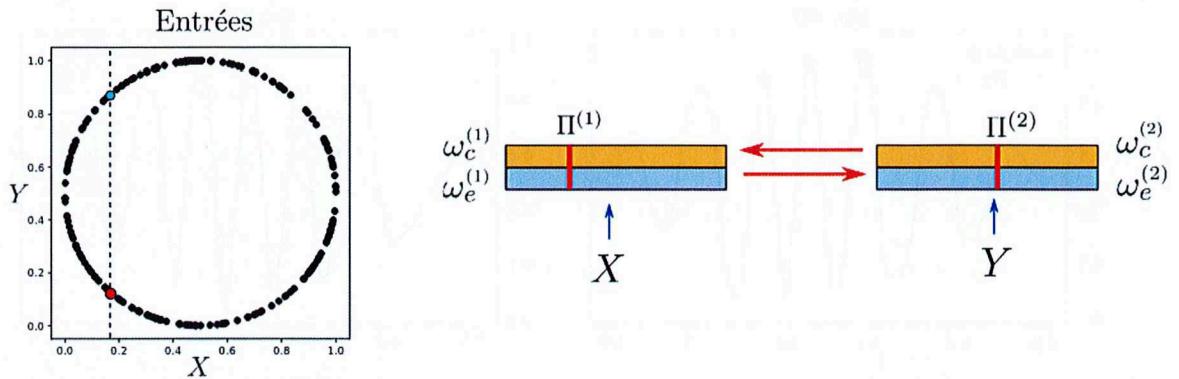


FIGURE 3.1 – Disposition des entrées, sous forme de cercle, à gauche, et architecture de deux cartes en une dimension étudiée et représentée dans ce chapitre.

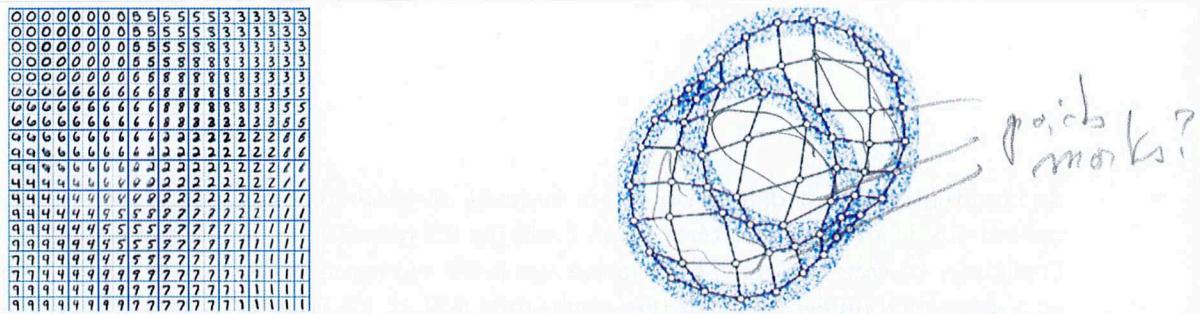


FIGURE 3.2 – Représentations possibles des poids d'une carte de Kohonen classiques, dans le cas d'entrées sous forme d'imagettes ou de points en deux dimensions.

### 3.1.2 Représentations et indicateurs classique des cartes de Kohonen

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

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

### 3.1.3 Limites de la représentation classique pour CxSOM

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

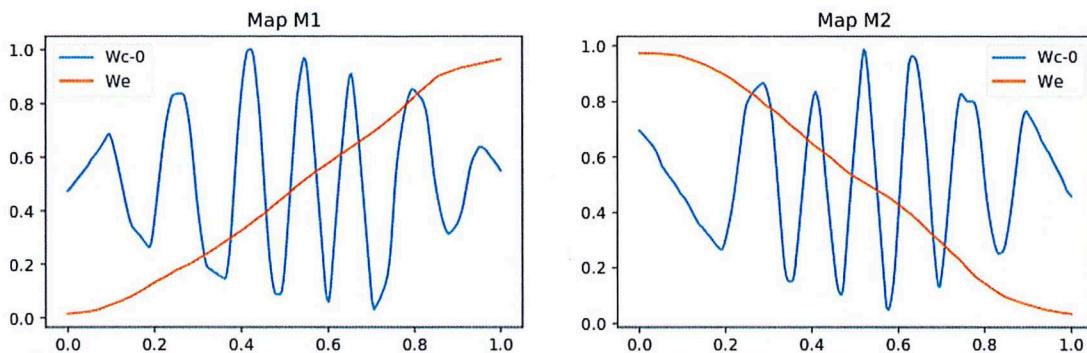


FIGURE 3.3 – Représentation des valeurs des poids d'une carte au sein de CxSOM après apprentissage. La seule représentation de ces poids ne suffit pas à savoir comment la carte se comporte.

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

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

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

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

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

## 3.2 Formalisation par variables aléatoires

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

### 3.2.1 Représentation des entrées

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

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

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

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

### 3.2.2 Démarche expérimentale

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

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

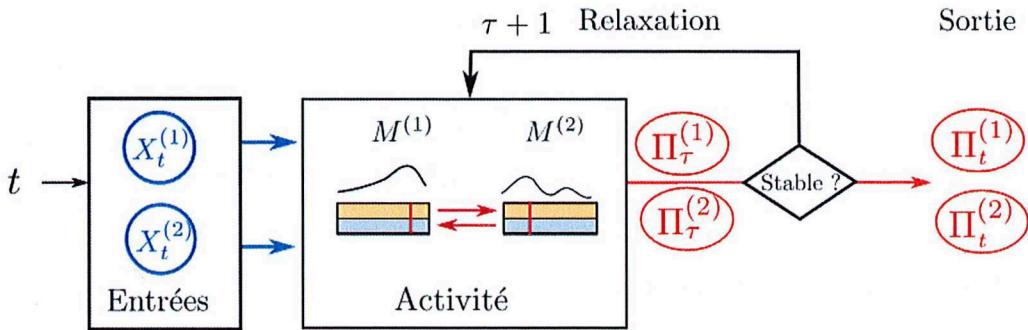


FIGURE 3.4 – Schéma des tests

réalisations d'une variable aléatoire jointe :

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

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

### 3.2.3 Résumé

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

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

Une phase de test est un ensemble de réalisations de cette variable aléatoire jointe.

## 3.3 Représentation graphiques des cartes

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

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

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

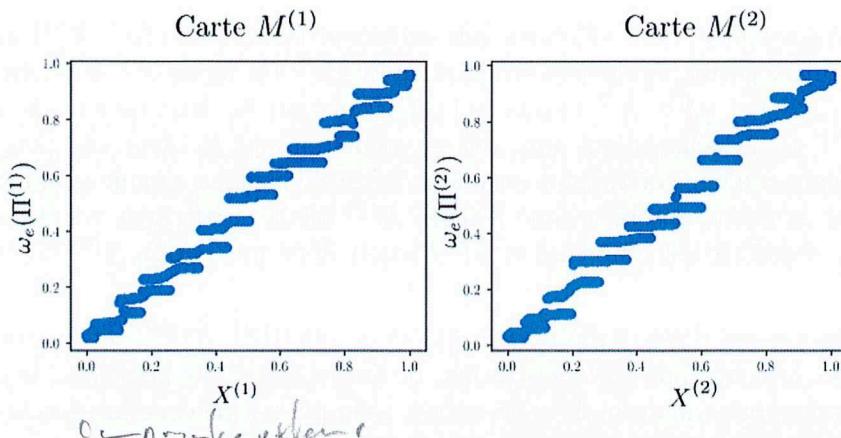


FIGURE 3.5 – Poids du BMU dans chaque carte en fonction de l’entrée présentée. On s’attend à des tracés proches de l’identité, montrant que le poids du BMU d’une carte est une bonne représentation de l’entrée. Sur ce graphique, on se rapproche effectivement de la fonction identité, cependant, une faible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

### 3.3.1 Erreur de quantification d'une modalité dans une carte

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

### 3.3.2 Cartographie des entrées

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

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

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

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

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

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

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

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

### 3.3.3 Réduction de dimension des entrées

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

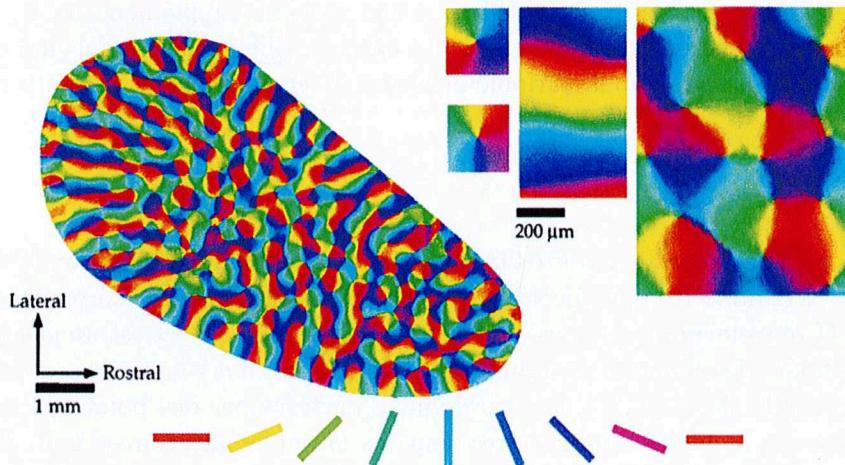


FIGURE 3.6 – Carte corticale de l'aire cérébrale visuelle V1. Pour tracer cette représentation, un ensemble de traits de différentes orientations sont présentés en stimuli visuels au sujet, indiqués en bas de l'image. Le neurone réagissant à une entrée d'orientation particulière est coloré sur la carte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

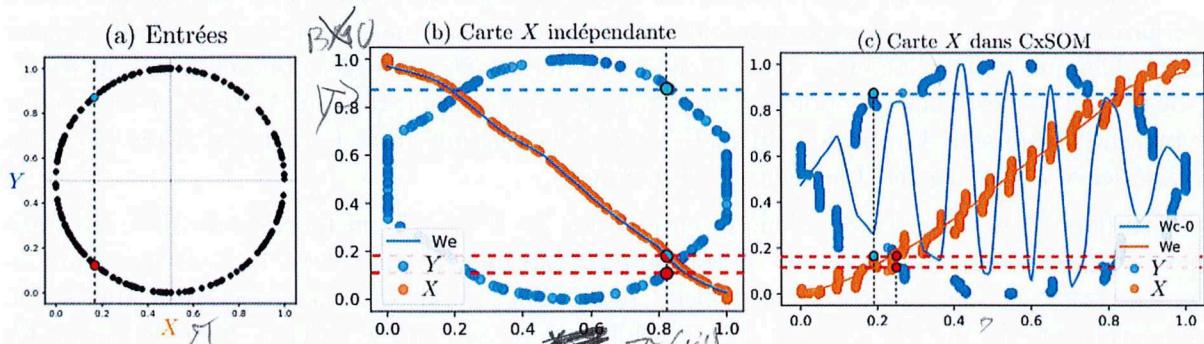


FIGURE 3.7 – Représentation des entrées  $X, Y$  d'une architecture de deux cartes relativement au BMU de la carte  $X$  après apprentissage. Ces tracés mettent en valeur l'organisation des cartes, différentes dans le cas où les cartes apprennent indépendamment leurs entrées (b) ou sont connectées (c).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

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

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

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

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

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

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

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

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

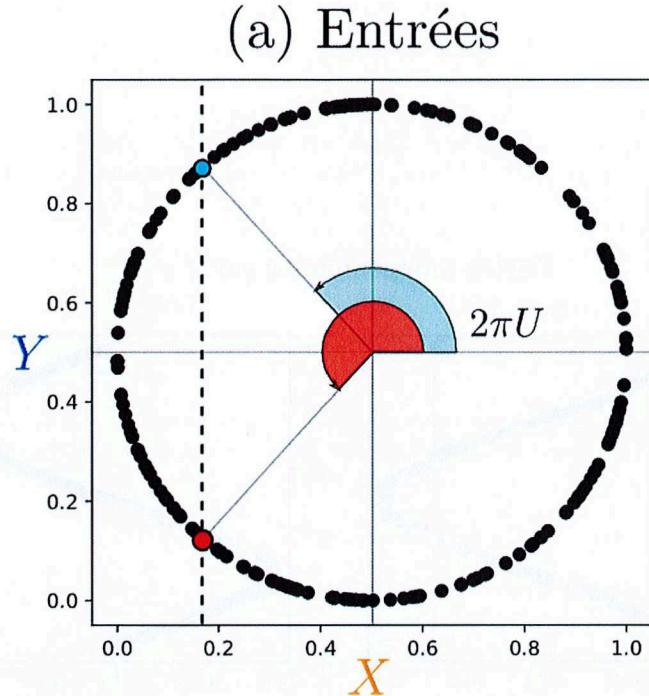


FIGURE 3.8 – Représentation choisie pour le cercle. Le modèle auxquelles appartiennent les modalités  $X^{(1)}$  et  $X^{(2)}$  est représenté par la variable  $U$ .  $\rightarrow$  cible de recherche?

### 3.3.4 Résumé

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

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

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

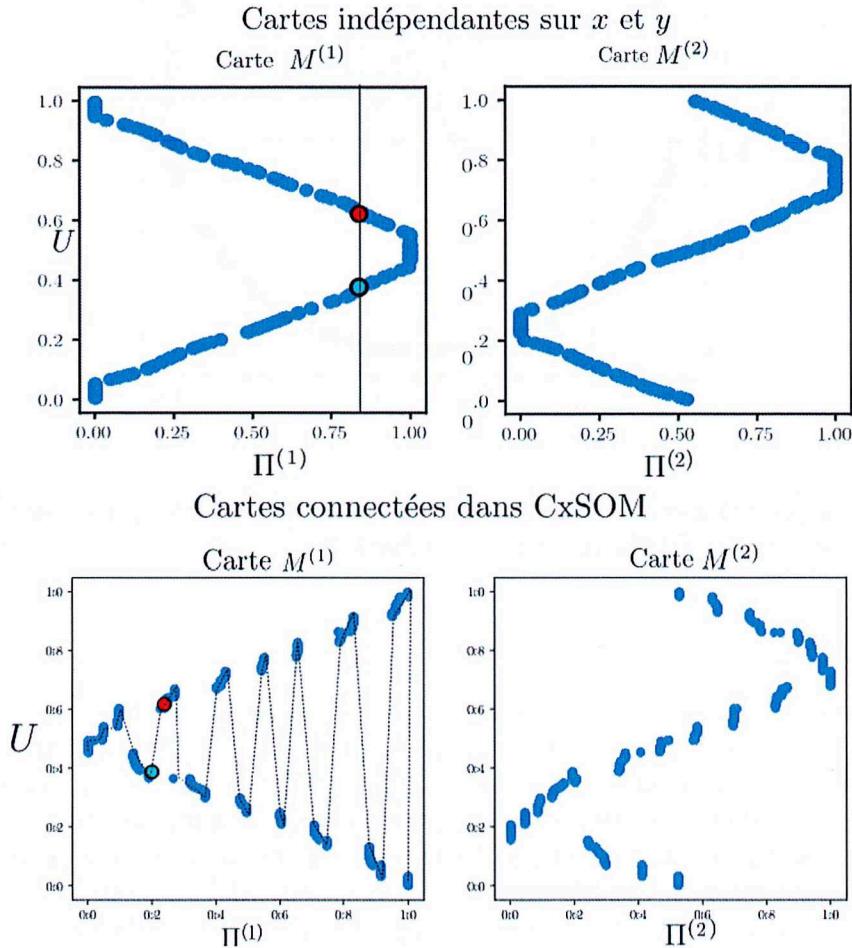


FIGURE 3.9 – Valeur de  $U$  en fonction des valeurs du BMU  $\Pi^{(i)}$  dans chacune des cartes, pour des entrées prises sur le cercle. Sur la première ligne, nous traçons la réponse de chaque carte à son entrée dans le cas où les cartes ne sont pas connectées. Sur la deuxième ligne, nous traçons la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