

# Chapitre 5

## Analyse des mécanismes d'apprentissage dans des architectures de cartes 1D

### Sommaire

---

<b>5.1</b>	<b>Introduction</b>	<b>96</b>
<b>5.2</b>	<b>Identification des mécanismes d'apprentissage dans une architecture de deux cartes</b>	<b>96</b>
5.2.1	Modèles d'entrées et architecture de cartes	97
5.2.2	Identification de mécanismes d'apprentissage sur un exemple	98
5.2.3	Généralisation des mécanismes d'apprentissage sur les autres modèles d'entrées	103
5.2.4	Étude des mécanismes de formation des deux échelles d'organisation spatiale	107
5.2.5	Discussion	110
<b>5.3</b>	<b>Génération de modalité dans des architectures de trois cartes</b>	<b>112</b>
5.3.1	Méthode expérimentale	114
5.3.2	Résultats	115
5.3.3	Exemple d'application	117
5.3.4	Conclusion	122
<b>5.4</b>	<b>Influence des connexions sur l'apprentissage du modèle d'entrées</b>	<b>122</b>
5.4.1	Influence d'un grand nombre d'entrées contextuelles sur l'organisation d'une carte	123
5.4.2	Influence des connexions distantes sur l'organisation d'une carte	124
<b>5.5</b>	<b>Conclusion</b>	<b>127</b>

---

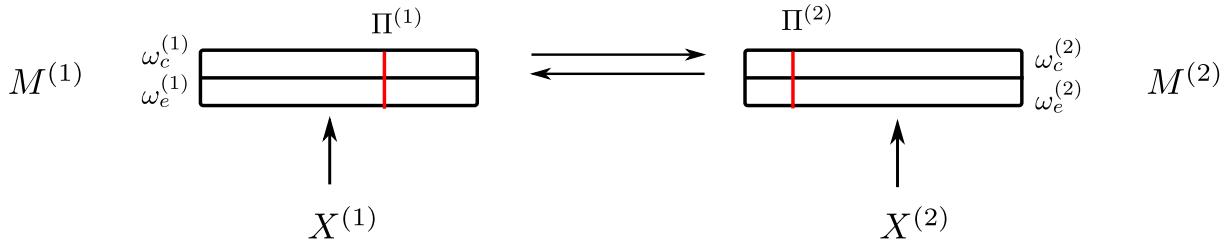


FIGURE 5.1 – Schéma de l'architecture de deux cartes. Chaque carte possède une couche de poids externe  $\omega_e$  et une couche de poids contextuels  $\omega_c$ . Les connexions sont rétroactives : l'entrée contextuelle de  $M^{(1)}$  est la position du BMU de la carte  $M^{(2)}$  et inversement. Chaque carte prend une entrée externe  $X^{(i)}$ .

## 5.1 Introduction

Les représentations introduites au chapitre 4 permettent de mettre en évidence une organisation dans la réponse des cartes d'une architecture lors de phases de test, c'est-à-dire dans ses positions de BMU. Nous utilisons à présent ces représentations pour identifier quels comportements d'organisation sont caractéristiques d'un apprentissage associatif des entrées, dans des architectures élémentaires de deux et trois cartes.

Nous comparons d'abord la réponse d'une même architecture de cartes à différents modèles d'entrées, afin de distinguer l'organisation qui est due au modèle d'entrées, de celle qui est commune aux différents modèles et donc une conséquence directe des règles d'apprentissage de la carte. Nous évaluerons également l'influence des paramètres de l'architecture sur cette organisation.

Nous observerons ensuite qu'une architecture peut encoder le modèle d'entrées et l'utiliser pour prédire une entrée manquante, ce qui confère à l'architecture de cartes un comportement de prise de décision, et non seulement de réaction aux entrées.

## 5.2 Identification des mécanismes d'apprentissage dans une architecture de deux cartes

Nous étudions d'abord les mécanismes d'apprentissage dans une architecture non hiérarchique de deux cartes. Chaque carte prend en entrée le BMU de sa voisine, introduisant une rétroaction entre les deux cartes. De cette façon, nous cherchons à isoler des comportements relatifs à une seule interface entre cartes. Dans cette section, nous reviendrons sur les comportements identifiés au chapitre 4 en cherchant à les généraliser sur plusieurs modèles d'entrées.

### 5.2.1 Modèles d'entrées et architecture de cartes

Le modèle d'architecture étudié dans cette section est présenté en figure 5.1. Chaque carte a une taille fixée de 500 noeuds, indexés entre 0 et 1, et possède deux couches de poids  $\omega_e$  et  $\omega_c$ . Les rayons de voisinage sont fixés à  $r_e = 0.2$  et  $r_c = 0.02$ , sauf si précisé autrement dans l'expérience. Les connexions sont réciproques :  $M^{(1)}$  prend comme entrée contextuelle  $\Pi^{(2)}$  et inversement.

Nous utiliserons comme modèle d'entrées des points en deux dimensions : chaque modalité  $X^{(1)}$  et  $X^{(2)}$  correspond à l'une des deux coordonnées  $x$  et  $y$  de ces points. Nous comparerons la réponse de l'architecture à différents modèles d'entrées jouets générés artificiellement, qui nous permettent de maîtriser les dépendances entre les modalités. Au chapitre 4, nous avons représenté la dépendance entre les modalités par une variable latente  $U$ , en bijection avec les entrées, représentant les paramètres libres du modèle d'entrées. Des points 2D situés sur une courbe sont représentés par une variable  $U$  1D, et des points sur une surface par une variable  $U$  2D. L'apprentissage associatif se traduit alors par l'encodage de  $U$  au sein de l'architecture.

Les modèles d'entrées que nous utilisons sont représentés en figure 5.2. Nous reviendrons d'abord sur le modèle du cercle (**A**), qui a déjà été présenté au chapitre 4. L'intérêt d'utiliser cette courbe est que toute entrée  $X^{(1)}$  correspond à deux valeurs possibles pour  $X^{(2)}$  et inversement.  $U$  est dans ce cas une variable 1D correspondant à l'angle du point sur le cercle. Nous comparerons ensuite les observations réalisées sur ce modèle sur d'autres dispositions d'entrées :

- **B** : Les entrées sont identiques.
- **C** : Une entrée est une fonction de l'autre :  $X^{(2)} = \cos(X^{(1)})$ .
- **D** : Les entrées sont sur une courbe plus complexe que le cercle, ici une courbe de Lissajous : une entrée  $X^{(1)}$  correspond à 4 à 6 valeurs de  $X^{(2)}$  et inversement.  $U$  est une variable 1D paramétrant la courbe.
- **E** : Les entrées sont totalement indépendantes, prises aléatoirement dans le patch  $[0, 1]^2$ .  $U$  est alors une variable 2D, correspondant aux coordonnées des points.
- **F** : Les entrées sont sur un anneau.  $U$  est alors une variable 1D, correspondant à l'angle du point sur le cercle, mais avec du bruit ajouté dans le modèle d'entrées. Une carte de Kohonen classique a comme propriété d'être résistante au bruit sur les données. Ainsi, une carte 1D se dépliant sur un anneau en 2D apprendra d'abord une représentation du cercle sous-jacent. Nous voulons vérifier comment cette propriété se traduit sur l'architecture de deux cartes.

Pour générer ces entrées,  $U$  est tiré uniformément dans  $[0, 1]$ , et tout couple d'entrées  $(X^{(1)}, X^{(2)})$  présenté à l'architecture lors de la même itération est généré à partir de la même valeur de  $U$ .

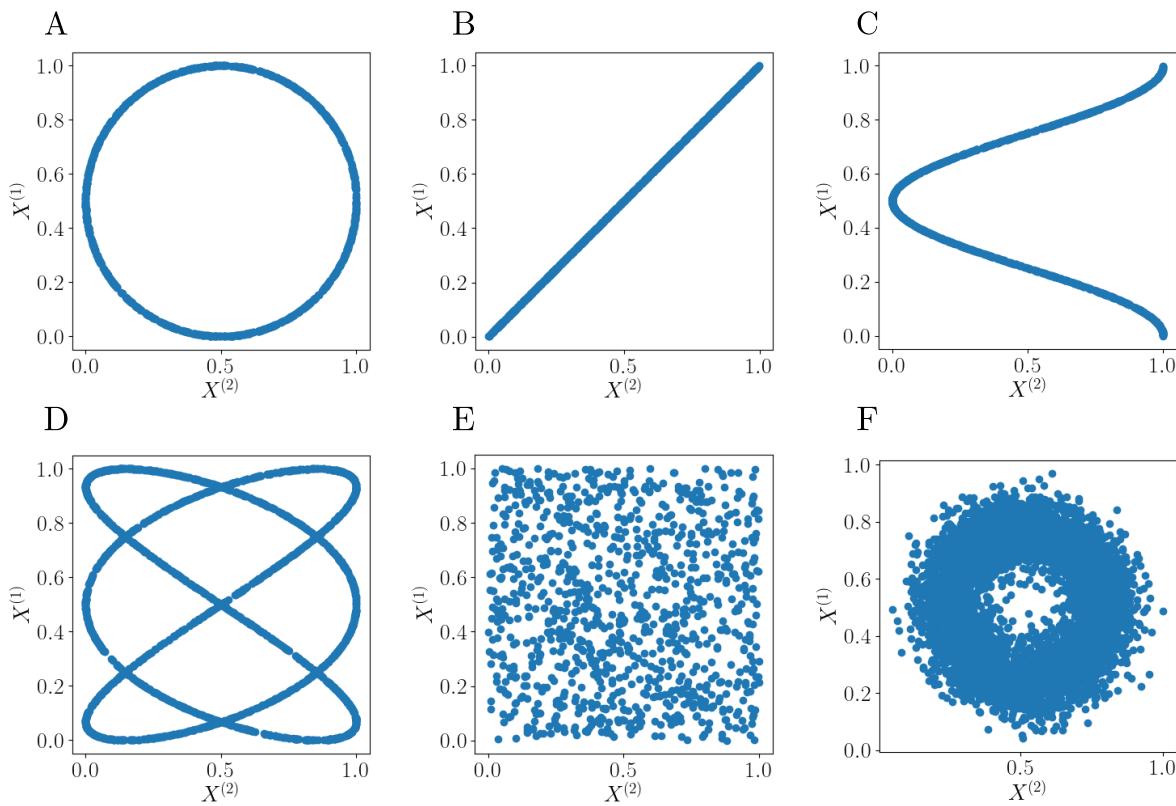


FIGURE 5.2 – Dispositions d'entrées en deux dimensions utilisées dans cette section.  $M^{(1)}$  prend en entrée les valeurs  $X^{(1)}$  et  $M^{(2)}$  les valeurs  $X^{(2)}$ .

### 5.2.2 Identification de mécanismes d'apprentissage sur un exemple

Revenons d'abord sur l'expérience sur le cercle **A**, déjà présentée au chapitre 4. Après avoir vérifié que les poids des cartes convergent au cours de l'apprentissage, nous détaillerons la disposition finale des poids externes et contextuels et l'organisation des BMU de la carte, afin d'identifier les mécanismes traduisant l'apprentissage des entrées et de leurs relations.

#### Convergence des poids

Dans une carte de Kohonen classique, les paramètres d'apprentissage (rayon de voisinage et taux d'apprentissage) sont diminués au cours des itérations d'apprentissage. Cette diminution permet d'assurer une stabilisation des poids. Ces paramètres sont au contraire gardés constants au cours des itérations dans notre architecture, comme expliqué en section 2.2.2 p. 43. Comme nous nous intéressons à l'organisation des cartes après apprentissage, nous nous assurons ici que les poids des cartes convergent effectivement vers une position stable, permettant de définir une fin d'apprentissage. Pour illustrer cette évolution, en figure 5.3, nous traçons la moyenne, sur

## 5.2. Identification des mécanismes d'apprentissage dans une architecture de deux cartes

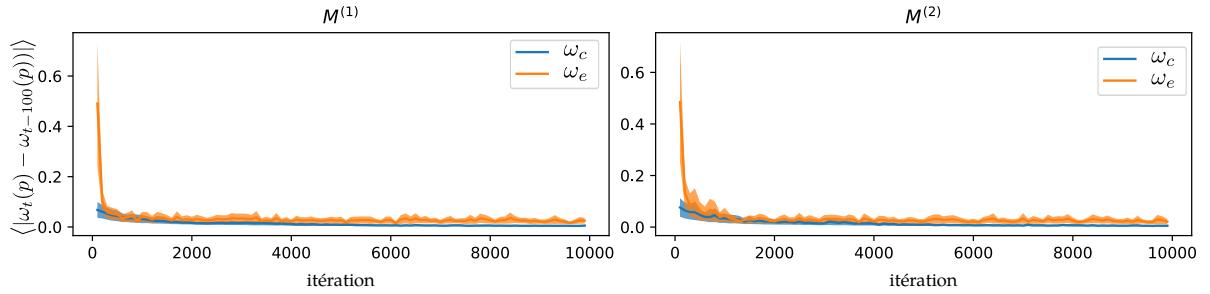


FIGURE 5.3 – Pour chaque carte, nous représentons l'évolution en fonction du temps de la différence moyenne entre les poids d'une carte à l'instant  $t$  et ceux à  $t - 100$ ,  $\langle |\omega_t(p) - \omega_{t-100}(p)| \rangle$ . L'évolution est moyennée sur 10 apprentissages dont les entrées sont tirées aléatoirement selon la même distribution d'entrées (**A**). Ces tracés montrent que les poids externes et contextuels évoluent rapidement vers une position dans laquelle ils ne varient plus que faiblement.

toutes les positions  $p$  d'une carte, de la différence entre  $\omega_t$  et  $w_{t-100}(p)$  :  $\langle |\omega_t(p) - w_{t-100}(p)| \rangle$  à différents instants  $t$  de l'apprentissage. Nous constatons que les poids évoluent rapidement au début de l'apprentissage, puis n'évoluent ensuite que très faiblement, ce qui suggère que les deux couches de poids ont atteint une position stable. Notons que ces tracés ne permettent pas de montrer expérimentalement la convergence des poids, mais donnent une bonne idée de l'évolution générale des poids des cartes. Graphiquement, nous avons également observé que les poids évoluent vers une position stable.

Nous pouvons proposer des éléments d'explication de cette convergence observée empiriquement, en remarquant qu'une carte se comporte principalement comme une carte de Kohonen classique se dépliant sur les entrées externes. En effet, l'activité externe domine dans le calcul de l'activité globale, cf. équation 2.7. L'activité contextuelle est utilisée ici comme un terme de modulation de l'activité externe. La convergence des poids d'une carte de Kohonen classique en 1D sur des données numériques a été démontrée, voir Cottrell, J.-C. Fort et al. 1998. La proximité du comportement de CxSOM avec une carte classique nous permet donc d'envisager que les poids des cartes CxSOM convergent également dans le cas de cartes 1D. La convergence en l'absence de décroissance des paramètres pourrait cependant poser plus de problèmes sur des cartes en deux dimensions.

### Organisation de la réponse des cartes après apprentissage

Maintenant que nous avons observé que les poids convergent vers une organisation stable, nous voulons identifier comment l'organisation des poids et des BMU de l'architecture traduisent un apprentissage du modèle d'entrées **A**. Nous avons présenté les méthodes de représentation et comparé la réponse de CxSOM à celle d'une carte simple au chapitre 4. Nous revenons ici sur ces observations en les complétant.

Nous reprenons la représentation cartographique des valeurs d'entrées, introduite en figure 4.7.

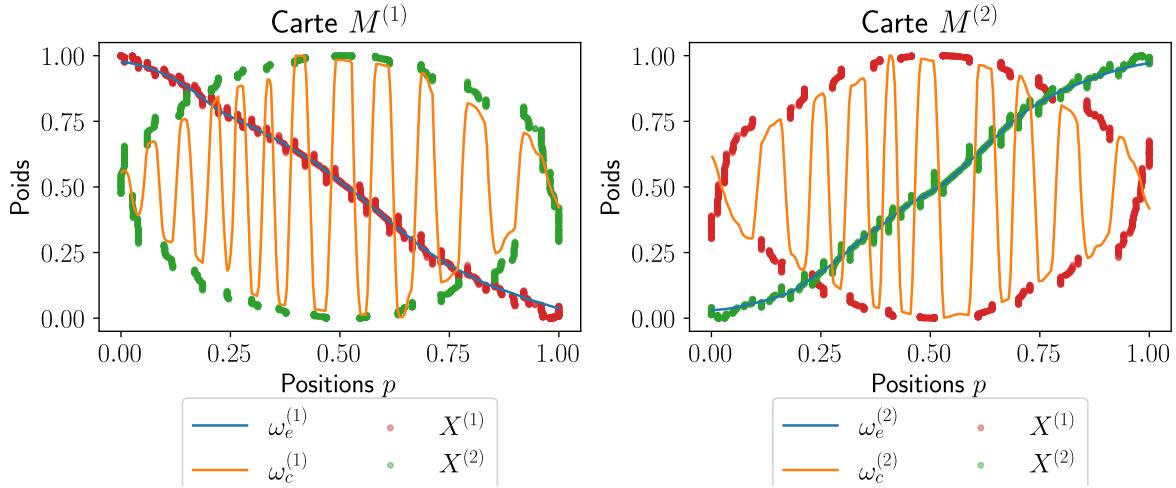


FIGURE 5.4 – Représentation cartographique des poids et entrées lors d'une phase de test selon la position du BMU dans chacune des cartes. Nous remarquons que les poids d'une carte, par exemple la carte  $M^{(1)}$ , s'organisent selon deux échelles. Les poids externes se déplient sur tout l'intervalle  $[0, 1]$ , tandis que les poids contextuels forment des motifs pseudo-périodiques. Les BMU s'organisent en zones, différenciant les valeurs de la paire  $(X^{(1)}, X^{(2)})$  et non seulement la valeur de  $X^{(1)}$ . Deux zones adjacentes codent pour des valeurs de  $X^{(1)}$  proches, mais  $X^{(2)}$  différents. Au sein d'une même zone, les BMU s'organisent sous la forme d'une sous-carte sur les valeurs de l'entrée contextuelle. Ces zones se forment de manière auto-organisée.

Il s'agit d'une représentation comportementale de la carte, qui permet d'observer une organisation des poids et des BMU. Cette représentation est tracée pour les deux cartes de l'architecture en figures 5.4 et 5.5, après apprentissage. Nous constatons que les poids externes, en bleu, présentent une disposition similaire aux poids d'une carte classique : ils sont classés de façon monotone entre 0 et 1. Les poids contextuels, en orange, présentent une disposition pseudo-périodique : ils présentent des oscillations spatiales régulières, mais les amplitudes et la largeur spatiale des oscillations varient.

La valeur des entrées  $X^{(1)}$  et  $X^{(2)}$  sont tracées en rouge et vert en fonction des positions de BMU obtenues lors de la phase de test. Les positions des BMU dans les deux cartes se répartissent en zones compactes, séparées par des zones mortes de la carte dont les nœuds n'ont jamais été BMU. Il s'agit d'une première différence avec une carte classique, pour laquelle toutes les positions seront BMU lorsque les entrées sont distribuées de façon continue. Les zones dans lesquelles les nœuds sont BMU correspondent aux extrema des poids contextuels et leurs alentours.

La figure 5.5 est un agrandissement de la figure 5.4, faisant figurer quatre zones de la carte  $M^{(1)}$ . Elle met en évidence le fait que chaque zone de BMU (en blanc) encode un intervalle du couple  $(X^{(1)}, X^{(2)})$ . Pour illustrer cette propriété, deux points sont indiqués en rouge et bleu. Ils possèdent la même valeur de  $X^{(1)} = 0.7$ , mais une valeur différente de  $X^{(2)}$ . Ils ont ici un

## 5.2. Identification des mécanismes d'apprentissage dans une architecture de deux cartes

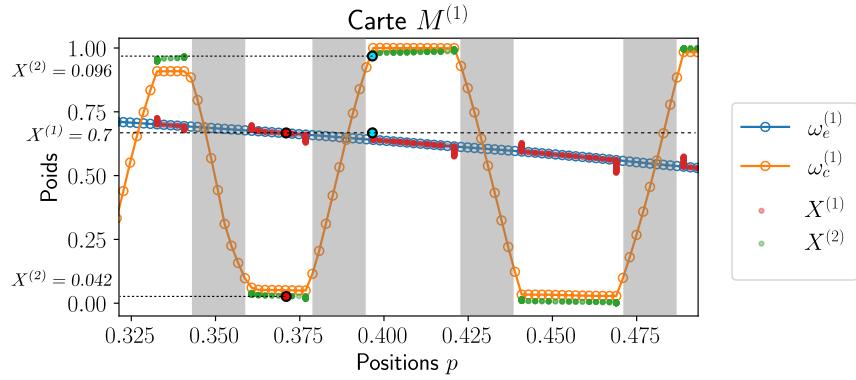


FIGURE 5.5 – Zoom sur la figure 5.4 entre les positions 0.35 et 0.55 de la carte  $M^{(1)}$ . Nous y faisons apparaître la position sur la courbe des nœuds de la carte. Deux zones consécutives seront BMU pour des ensembles d'entrées qui se recouvrent. Par exemple, les deux entrées correspondant aux points bleu et rouge ont les mêmes valeurs de  $X^{(1)}$ , mais des valeurs différentes de  $X^{(2)}$ . Leurs BMU sont alors séparés dans la carte  $M^{(1)}$  dans deux zones consécutives. Entre les zones, quelques unités ne sont jamais BMU, en gris sur la figure. Il s'agit de zones mortes, créant des discontinuités dans la réponse de la carte.

BMU différent dans la carte  $M^{(1)}$ . Ces BMU sont situés dans deux zones adjacentes. Ces tracés montrent que la réponse des cartes présente une organisation à deux échelles, portée par la forme des poids externes et contextuels. Dans chaque carte, le BMU est choisi selon  $X^{(1)}$ , puis localement selon  $X^{(2)}$ . Chaque position se spécialise ainsi en fonction des deux composantes du modèle d'entrées ( $X^{(1)}, X^{(2)}$ ). Chaque zone se spécialise donc pour un petit intervalle de  $U$ . Nous avons ainsi observé, en section 4.3.3, que l'apprentissage des relations entre entrées se traduit par une relation fonctionnelle entre  $U$  et  $\Pi$  dans chaque carte, tracée en figure 4.8. Nous reviendrons plus en détail sur cette représentation et la propriété de relation fonctionnelle entre la variable cachée et la position du BMU au chapitre 6.

### Erreur de quantification vectorielle

Nous nous intéressons enfin à la quantification vectorielle réalisée par la couche de poids externe sur l'entrée externe dans chaque carte. Nous souhaitons que le poids externe du BMU soit une approximation de l'entrée externe, ce qui apporte une possibilité de reconstruction de l'entrée à partir du BMU.

La figure 5.6 présente la valeur de cette approximation  $\omega_e(\Pi^{(i)})$  au sein de chaque carte en fonction de l'entrée correspondante  $X^{(i)}$ . Cette figure montre que la quantification vectorielle est bien réalisée : les valeurs approximées sont proches des valeurs d'entrées. L'erreur de quantification est néanmoins plus importante que celle qu'on obtiendrait avec une carte de même taille et mêmes paramètres apprenant sur l'ensemble des  $X^{(i)}$ . Nous remarquons une disposition en étages, dues aux zones formées par les poids contextuels. En effet, une même valeur d'entrée

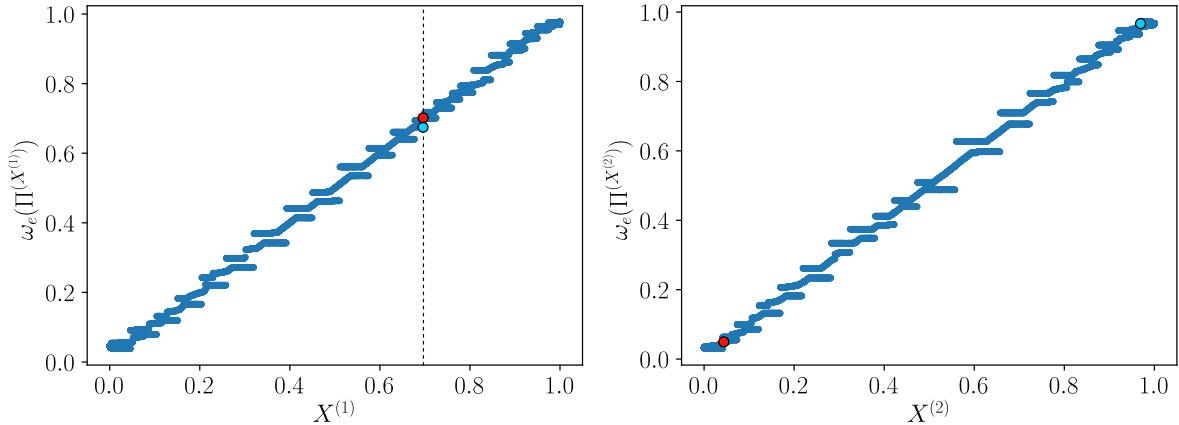


FIGURE 5.6 – Représentation de l'erreur de quantification sur les valeurs de  $X^{(1)}$  et  $X^{(2)}$ . Le poids externe du BMU est proche de la valeur de l'entrée ; chaque carte réalise une bonne quantification vectorielle sur ses entrées. Les mêmes points rouges et bleus représentés en figure 5.5 sont reportés sur le graphique. Bien qu'ils aient la même valeur d'entrée  $X^{(1)}$ , leurs BMU sont placés dans deux zones de la carte. La valeur quantifiée de ces deux entrées est alors légèrement différente. Ce comportement entraîne la disposition en étages observée sur le nuage de points.

peut avoir un BMU dans deux zones de la carte, en fonction de la valeur de l'entrée contextuelle. Comme les poids externes sont strictement croissants ou décroissants, cela induit l'erreur observée dans la prédiction d'entrée.

### Résumé des observations

Les résultats de cette expérience ainsi que les observations présentées au chapitre 4 nous permettent de formuler les hypothèses suivantes concernant les comportements élémentaires d'apprentissage d'une architecture de deux cartes 1D :

- Les poids externes de chaque carte de l'architecture permettent à chacune d'effectuer une bonne quantification vectorielle de leurs entrées externes.
- Les poids externes et contextuels de chaque carte s'organisent selon deux échelles d'organisation. L'organisation des poids contextuels forme des motifs pseudo-périodiques.
- La réponse des cartes est marquée par la présence de zones de BMU, séparées par des zones mortes. Ces zones de BMU correspondent aux pseudo-périodes formées par les poids contextuels. Chaque zone se spécialise pour un même intervalle de valeurs du modèle d'entrées complet, donc de  $U$ .
- L'apprentissage du modèle d'entrées par l'architecture se traduit par l'existence d'une relation fonctionnelle entre  $U$  et  $\Pi$  dans chaque carte, montrant que chaque carte encode le modèle d'entrées et non seulement son entrée externe.

## 5.2. Identification des mécanismes d'apprentissage dans une architecture de deux cartes

---

Nous cherchons dans la suite de ce chapitre à vérifier ces hypothèses sur les autres dispositions d'entrées en deux dimensions, et à compléter les observations. Nous étudierons en particulier comment les deux échelles de quantification se forment et quelles propriétés d'apprentissage elles confèrent à l'architecture de cartes. Nous verrons ensuite en section 5.3 que cette organisation à deux échelles associée à la propriété de quantification vectorielle de l'entrée externe permet à l'architecture de cartes de prédire des entrées manquantes lors d'une phase de test.

### 5.2.3 Généralisation des mécanismes d'apprentissage sur les autres modèles d'entrées

Nous reprenons les modèles d'entrées **B,C,D,E** de la figure 4.7. Pour tous ces modèles, nous avons vérifié que la quantification vectorielle est bien réalisée dans chaque carte sur ses entrées, que nous n'avons pas tracée ici. Nous nous concentrerons sur l'organisation des poids et des BMU des cartes, afin d'identifier les propriétés d'organisation directement liées aux entrées et celles qui sont systématiques au modèle d'architecture CxSOM.

Dans la disposition d'entrées **B**, les deux entrées  $X^{(1)}$  et  $X^{(2)}$  sont en bijection : une valeur de  $X^{(1)}$  correspond à une seule valeur de  $X^{(2)}$  dans le modèle. La représentation cartographique correspondante est tracée en figure 5.7. Les poids externes et contextuels s'organisent tous les deux de façon monotone entre 0 et 1. Ils sont donc sur une même échelle spatiale. Ceci s'expliquerait par le fait que la carte  $M^{(1)}$  n'a pas besoin de distinguer plusieurs valeurs de  $X^{(2)}$  pour une même entrée  $X^{(1)}$ , et inversement. Les deux cartes agissent alors comme des cartes classiques, indépendantes. Leurs BMU se répartissent sur toute la carte, sans former de zones.

Dans la disposition d'entrées **C**, la dépendance entre les entrées n'est plus bijective :  $X^{(2)}$  est toujours une fonction de  $X^{(1)}$ , mais pas le contraire. Sa représentation est tracée en figure 5.8. Les poids externes et contextuels de la carte  $M^{(1)}$  s'organisent sur la même échelle spatiale, ce qui est cohérent avec le comportement observé sur les entrées **B** : une seule valeur de  $X^{(2)}$  correspond à une même valeur de  $X^{(1)}$ . Au contraire, la carte  $M^{(2)}$  doit à présent distinguer deux valeurs de  $X^{(1)}$  possibles pour chaque valeur de  $X^{(2)}$ , ce qu'elle fait en formant une deuxième échelle de quantification, grâce aux poids contextuels. Ce comportement rejoint ainsi celui que nous avons observé sur le cercle. Chaque zone de la carte  $M^{(2)}$  est alors spécialisée pour un intervalle de valeurs de  $(X^{(1)}, X^{(2)})$  et non seulement  $X^{(2)}$ .

D'après ces deux expériences, nous supposons que la formation de deux échelles de quantification permet à une carte de distinguer plusieurs valeurs dans le modèle d'entrées qui correspondent à une même valeur pour son entrée externe. Dans ce cas, une position se spécialise en tant que BMU sur une seule valeur du modèle d'entrées. Lorsqu'une telle division n'est pas nécessaire d'après le modèle d'entrées, les cartes s'organiseront selon leur entrée externe, de façon similaire à une carte de Kohonen classique.

Nous observons ensuite l'organisation des cartes obtenue lorsqu'une valeur de  $X^{(1)}$  correspond à plus de deux valeurs de  $X^{(2)}$  : 4 valeurs dans le cas de la courbe de Lissajous (Entrées **D**), tout l'intervalle  $[0, 1]$  dans le cas du patch  $[0, 1]^2$  (Entrées **E**) ou deux intervalles dans le cas de l'anneau (Entrées **F**). La représentation cartographique est tracée pour ces trois cas en figures 5.9, 5.10 et 5.11. Dans ces trois cas, les poids externes et contextuels présentent encore deux échelles d'organisation spatiale, les poids contextuels formant des pseudo-périodes. La disposition des BMU de ces cartes forment des zones distinctes, dont chacune encode des intervalles différents du modèle d'entrées. Chaque zone de BMU sur  $M^{(1)}$  agit ici comme une sous-carte de toutes les valeurs possible de l'entrée  $X^{(2)}$  qui correspondent aux valeurs de  $X^{(1)}$  dans cette zone. Par exemple, en figure 5.10, chaque zone de BMU forme une carte de tout l'intervalle  $[0, 1]$ , ce qui correspond à la distribution de  $X^{(2)}$  lorsque  $X^{(1)}$  est fixé. Il est intéressant de noter que la forme des poids contextuels diffère entre toutes les dispositions d'entrées, mais que le nombre de pseudo-périodes formées par les poids contextuels reste similaire. Cette observation laisse penser que l'encodage à deux échelles est régulé par des paramètres de l'architecture, tandis que la disposition au sein des zones dépend ensuite des distributions d'entrées.

Nous voulons enfin observer comment est réalisée la quantification vectorielle de l'espace 2D ( $X^{(1)}, X^{(2)}$ ) par les deux cartes. Nous prenons ici comme exemple la disposition d'entrées **E**. En figure 5.12, nous traçons la distorsion des poids externes des cartes. Il s'agit des valeurs  $\omega_e(\Pi^{(1)})$  en fonction de  $\omega_e(\Pi^{(2)})$ , reliées selon l'ordre des connexions de  $M^{(1)}$  (à gauche) ou de  $M^{(2)}$  (à droite). Nous y observons que les cartes quantifient tout l'espace  $[0, 1]^2$ , mais par seulement 90 valeurs de ces vecteurs codes. Le patch est donc seulement quantifié en 90 valeurs, ce qui est plus faible que ce que nous pourrions attendre de deux cartes ayant chacune 500 noeuds. Ces valeurs sont définies par les zones de poids contextuels des cartes. L'apprentissage du modèle d'entrées dans chaque carte semble ainsi réduire la capacité de quantification vectorielle sur les entrées externes.

La dernière observation que nous avions relevée sur le cercle est que  $U$  est une fonction de la position du BMU dans chaque carte, ce qui montre que chaque carte de l'architecture a appris une représentation du modèle d'entrées. Le tracé de  $U$  selon  $\Pi$  a été effectué en figure 4.8 pour le cas des entrées **A**. Nous effectuons ce tracé pour les entrées **D** (courbe de Lissajous) en figure 5.13. Cette figure fait également apparaître  $U$  comme une fonction de la position du BMU dans chaque carte, ce qui étend l'observation réalisée sur le cercle. Nous reviendrons plus en détail sur l'utilisation de  $U$  dans l'analyse des réponses des cartes au chapitre 6.

Grâce à ces expériences, nous postulons que l'organisation de la carte en deux échelles spatiales est un comportement systématique de l'architecture CxSOM. Ces deux échelles sont marquées par une organisation pseudo-périodique des poids contextuels, qui s'organisent selon une échelle spatiale plus réduite que celle formée par les poids externes. Les BMU des cartes se répartissent alors en zones distinctes, séparées par des zones mortes. Une position de BMU se

## 5.2. Identification des mécanismes d'apprentissage dans une architecture de deux cartes

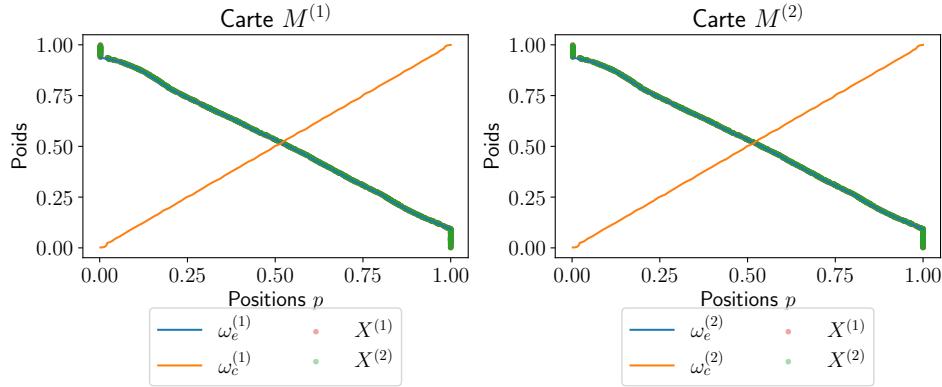


FIGURE 5.7 – Représentation cartographique des poids et entrées pour la disposition  $X^{(1)} = X^{(2)}$  (**B**). Les entrées  $X^{(1)}$  et  $X^{(2)}$  sont identiques, et superposées. Les poids externes et contextuels s'organisent selon une seule échelle spatiale. Les deux cartes agissent comme deux cartes indépendantes qui apprendraient sur les mêmes entrées.

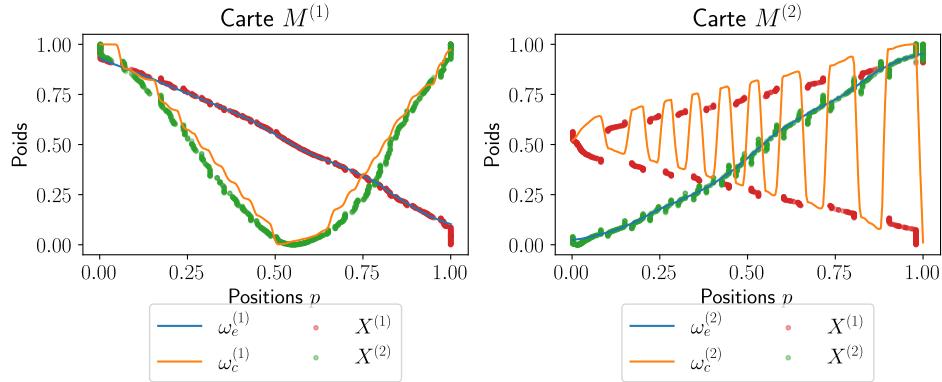


FIGURE 5.8 – Représentation cartographique des poids et entrées pour  $X^{(2)} = \cos(X^{(1)})$  (**C**). Les poids contextuels de la carte  $M^{(1)}$  forment une même échelle spatiale, car une valeur de  $X^{(1)}$  correspond toujours à une seule valeur de  $X^{(2)}$ . Au contraire, les poids de la carte  $M^{(2)}$  forment deux échelles d'organisation spatiale, permettant de gérer une distinction : pour une même valeur de  $X^{(2)}$ , deux  $X^{(1)}$  sont possibles. Les BMU s'organisent alors en zones distinctes.

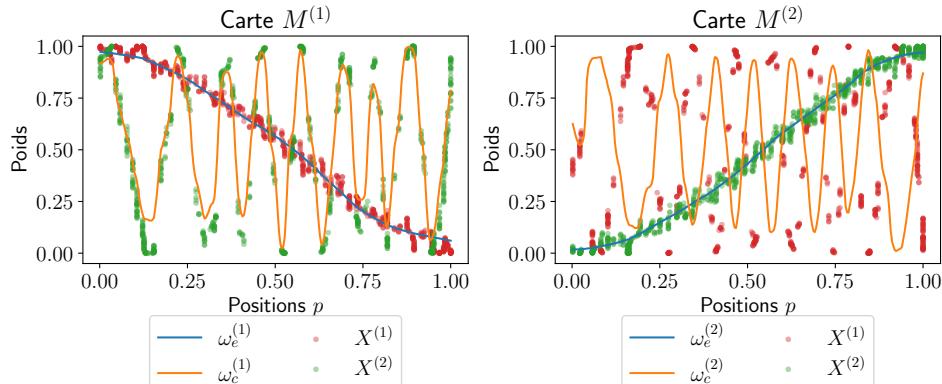


FIGURE 5.9 – Représentation cartographique des poids et entrées pour des entrées sur une courbe de Lissajous, **D**. Comme ci-dessus, les poids externes et contextuels forment deux échelles d'apprentissage et les BMU se répartissent en zones. Chaque zone de  $M^{(1)}$  forme une carte organisée des entrées  $X^{(2)}$  de l'architecture sur une sous-région des entrées  $X^{(1)}$ .

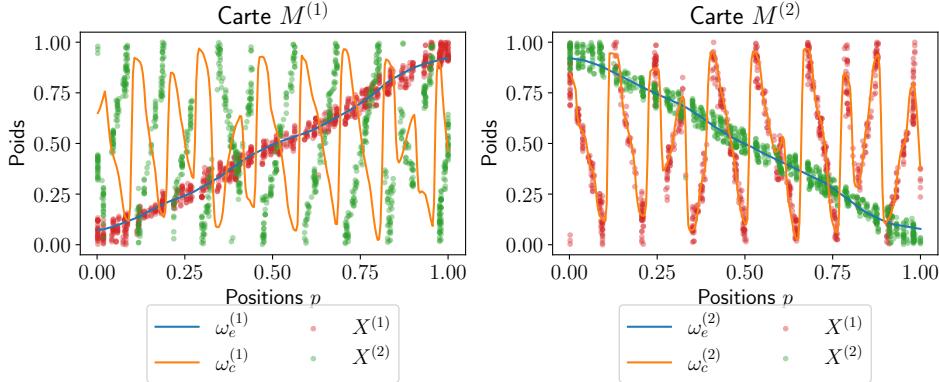


FIGURE 5.10 – Représentation cartographique des poids et entrées dans le patch  $[0, 1]^2$ , **E**. Les poids contextuels s'organisent de façon pseudo-périodique. Chaque zone de BMU définie par ces motifs forme une carte organisée des sous-régions de l'espace d'entrée externe.

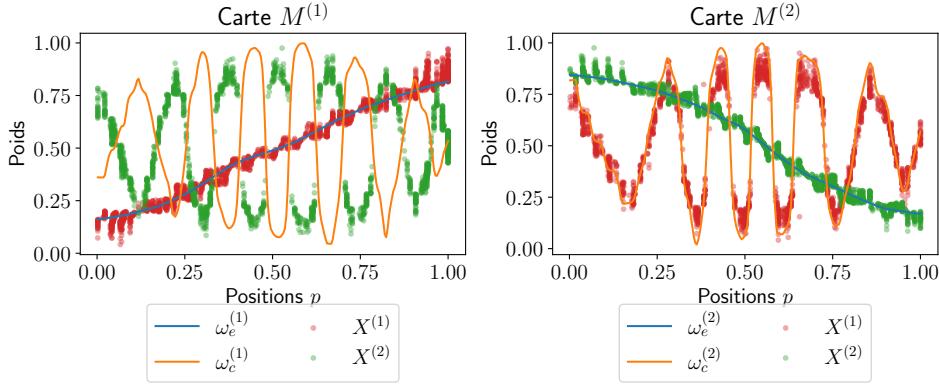


FIGURE 5.11 – Représentation cartographique des poids et entrées pour des entrées sur un anneau, **F**. La disposition des poids et la réponse des cartes est similaire à celle de l'architecture apprenant sur le cercle. Une architecture de cartes est robuste au bruit sur les entrées externes, ce qui étend les propriétés de robustesse au bruit d'une carte de Kohonen classique à CxSOM.

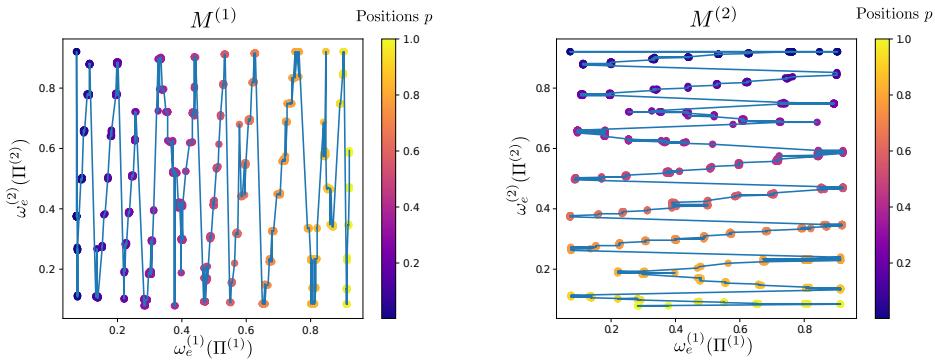


FIGURE 5.12 – Représentation de la distorsion des poids des deux cartes sur le modèle d'entrées **E**. Les cartes s'organisent de façon à représenter tout le patch  $[0, 1]^2$ , l'une selon les  $X^{(1)}$ , l'autre selon les  $X^{(2)}$ . Bien que chaque carte ait 500 nœuds, on observe seulement environ 90 valeurs possibles des paires  $\omega_e(\Pi^{(1)})$ ,  $\omega_e(\Pi^{(2)})$

## 5.2. Identification des mécanismes d'apprentissage dans une architecture de deux cartes

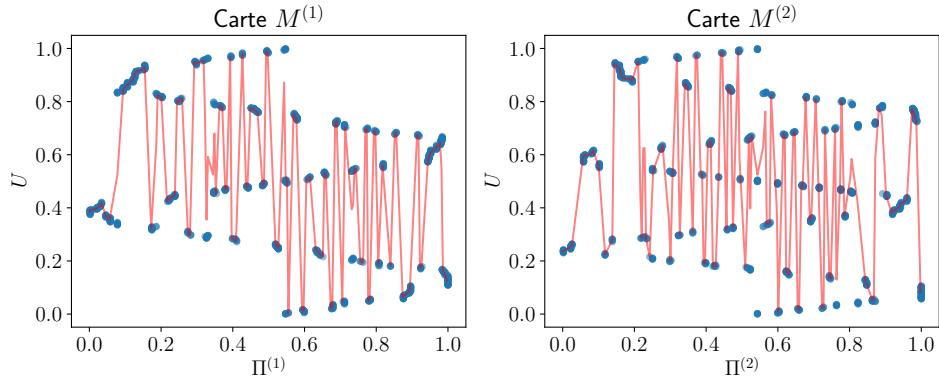


FIGURE 5.13 – Représentation de  $U$  selon le BMU  $\Pi^{(i)}$  dans chaque carte (en bleu) pour des entrées sur une courbe de Lissajous  $\mathbf{D}$ . La courbe en rouge montre l'approximation du nuage de points par une fonction :  $U$  est ici une fonction de la position du BMU dans chaque carte, ce qui vérifie l'observation réalisée sur le cercle.

spécialise en fonction des deux entrées de l'architecture, encodant finalement une valeur du modèle d'entrées  $U$ . Une zone agit comme une sous-carte organisée, à  $X^{(i)}$  fixé, des valeurs possible de l'autre entrée du modèle dans cette zone.

Ces deux niveaux d'indexation se forment de façon auto-organisée, mais émergent de l'organisation seulement lorsqu'il est nécessaire de pouvoir distinguer plusieurs valeurs du modèle  $U$  correspondant à la même entrée externe d'une carte dans le modèle d'entrées. Le nombre de zones reste similaire entre les expériences présentées, ce qui suggère que cette organisation est liée aux paramètres de l'architecture. La forme des zones et la réponse des cartes dépend ensuite de la relation entre les entrées.

Nous observons enfin que la quantification vectorielle globale de l'espace d'entrée choisit peu de valeurs de quantification, par rapport à ce qu'on pourrait attendre de cartes de taille 500. Chaque carte encode donc tout le modèle d'entrées et non seulement l'entrée externe, mais au détriment de la précision de la quantification vectorielle générale des entrées. Ce comportement était prévisible, du fait que l'architecture possède un nombre d'unités de calcul fixé correspondant au nombre de nœuds dans les cartes.

### 5.2.4 Étude des mécanismes de formation des deux échelles d'organisation spatiale

Nous nous intéressons maintenant aux paramètres des cartes influençant la formation des deux échelles d'organisation spatiales et des zones de BMU. Nous avons observé que la présence de zones est liée en particulier au rapport entre les rayons de voisinage externe et contextuel. Ces paramètres contrôlent l'élasticité de chaque couche de poids. Nous comparons dans cette section l'organisation obtenue sur plusieurs architectures ayant des couples de rayons de voisinage

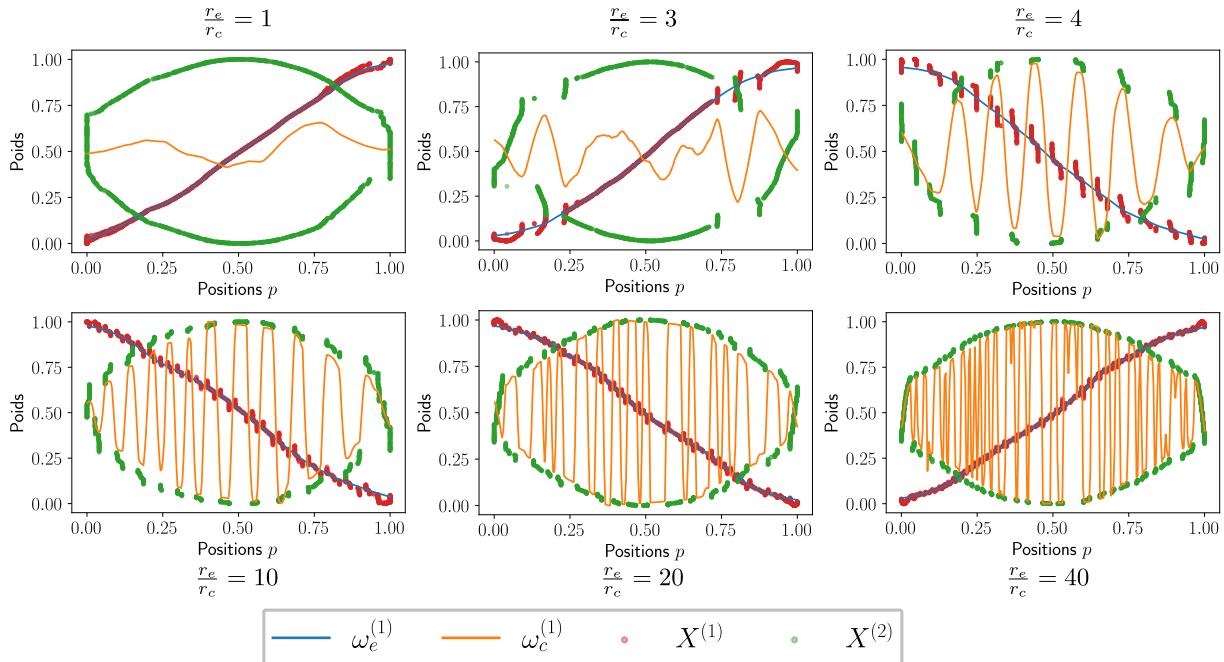


FIGURE 5.14 – Représentation cartographique de la carte  $M^{(1)}$  pour différents rayons de voisinage contextuels, le rayon de voisinage  $r_e$  étant fixé à 0.2. Les entrées sont tirées dans le modèle **A**. Nous observons que la présence de zones dépend du rapport entre les rayons de voisinage. La carte définit des zones de BMU grâce à la forme des poids contextuels. Elles sont de plus en plus nombreuses et contiennent de moins en moins d'unités lorsqu'on augmente le rapport entre rayons de voisinage. La carte  $M^{(2)}$ , non représentée ici, se comporte de façon similaire.

contextuels et externes différents, sur une même disposition d'entrée.

### Influence du rapport entre les rayons de voisinage sur la formation de zones de BMU

Nous reprenons les entrées **A**, en cercle et réalisons l'apprentissage de ce modèle par plusieurs architectures dans lesquelles le rapport  $\frac{r_e}{r_c}$  est différent. Nous fixons pour cela  $r_e$  à 0.2 et faisons varier  $r_c$  de 0.2 à 0.005.

En figure 5.14, nous traçons la représentation cartographique de  $M^{(1)}$  après apprentissage pour ces différents rayons de voisinages. Nous n'avons pas représenté  $M^{(2)}$ , mais son comportement est semblable à  $M^{(1)}$  par la symétrie des entrées. Nous constatons que la disposition des BMU en zones intervient ici pour une valeur de  $\frac{r_e}{r_c} < 3$ . Pour  $r_c = r_e$ , les poids  $\omega_c$  restent centrés autour de 0.5, moyennant les deux positions d'entrées contextuelles possibles pour chaque valeur de  $X^{(1)}$ .

Le nombre de zones de BMU augmente ensuite avec le rapport des rayons de voisinage. Plus il y a de zones, plus la quantification réalisée sur l'entrée externe est précise. La taille du rayon de voisinage contextuel est limité par la taille de la carte. Si celui-ci est trop faible, la notion de

## 5.2. Identification des mécanismes d'apprentissage dans une architecture de deux cartes

---

zone n'a plus vraiment de sens. C'est ce qu'on observe pour  $\frac{r_e}{r_c} = 40$ .  $r_c$  est alors de 0.005, soit 2 nœuds, les cartes ayant chacune 500 nœuds. Chaque zone de BMU ne contient plus que quelques unités, on ne peut donc plus vraiment parler de sous-carte. Néanmoins, les BMU sont toujours spécialisés en fonction des deux entrées du modèle, donc l'apprentissage associatif est toujours réalisé.

Nous verrons dans la suite du chapitre que la disposition en zones de BMU est nécessaire pour permettre l'apprentissage des relations entre entrées. Les rayons de voisinage doivent donc être choisis de façon à permettre la formation de zones. Le choix des paramètres les plus adaptés à une application restera ensuite à déterminer dans les travaux futurs : vaut-il mieux privilégier un grand nombre de petites zones, ou peu de zones contenant de nombreux nœuds ?

Nous pouvons proposer une explication à ces deux échelles d'apprentissage par le fait que le rapport entre rayons de voisinage introduit deux échelles d'élasticités dans les poids, ainsi que deux échelles temporelles de mise à jour. D'une part, les poids externes se déplient plus vite que les poids contextuels, car le grand rayon de voisinage externe permet de mettre à jour plus de prototypes à chaque itération. D'autre part, les poids externes présentent une « attraction » plus forte sur les unités voisines : l'élasticité de la couche de poids externe est plus élevée. Les poids contextuels doivent donc composer avec cette force. Leur organisation est ainsi subordonnée à l'organisation des poids externes.

Dans nos expériences, nous avons pris des  $r_c$  communs à toutes les couches de poids contextuels. Nous pourrions cependant associer à chaque couche de poids d'une carte un rayon de voisinage différent. Enfin, nous pouvons aussi faire varier le taux d'apprentissage  $\alpha$ , ainsi que les paramètres de la fonction d'activation : la pondération de chaque couche d'activité dans l'activation globale ou les largeurs des activations gaussiennes  $\sigma$ . Pour pouvoir adapter ces paramètres automatiquement dans un objectif d'application et réaliser une étude paramétrique approfondie, il nous faudrait définir une fonction caractérisant l'organisation d'une carte, ou une fonction de coût relative à un objectif d'apprentissage. Nous discuterons d'un tel indicateur au chapitre 6, mais soulignons ici que nous n'avons pas encore défini de valeur adéquate. C'est pourquoi nous nous sommes intéressée à la compréhension de l'effet des paramètres dans nos travaux, et n'avons pas cherché à optimiser ces valeurs.

### Influence des deux échelles d'apprentissage sur la recherche de BMU par relaxation

Nous nous interrogeons sur l'influence de la disposition à deux échelles des poids externes et contextuels dans le processus de relaxation. Nous pouvons en effet penser que les périodes formées par les poids contextuels favorisent la recherche de consensus lors de la relaxation et la convergence des poids, ce que nous voulons vérifier ici. Pour cela, nous comparons les indicateurs d'évolution de la convergence de la relaxation, introduits au chapitre 3 sur deux expériences. Nous

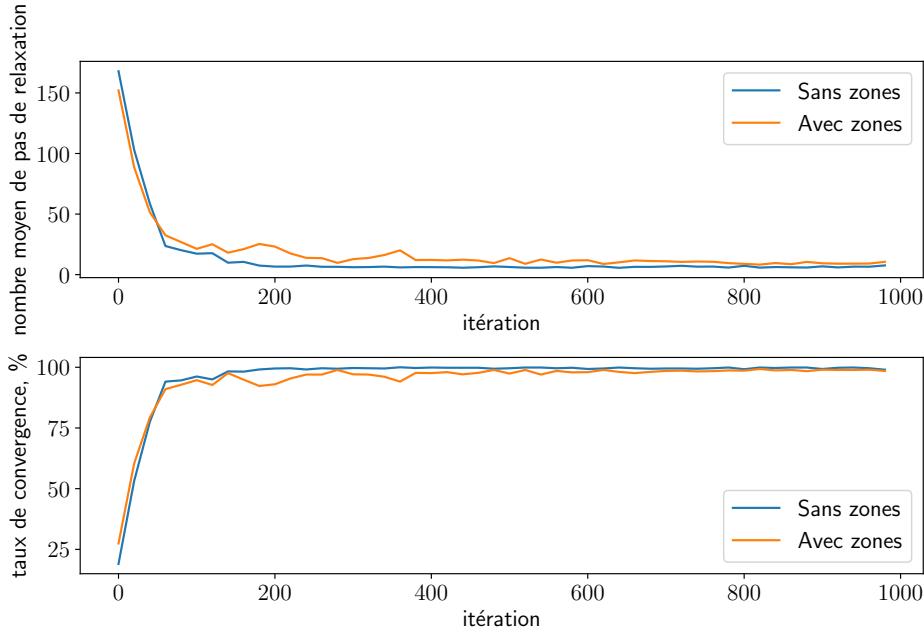


FIGURE 5.15 – Évolution du nombre moyen de pas de relaxation et du taux de convergence pour une organisation de cartes ayant formé des zones de poids contextuels ( $\frac{r_e}{r_c} = 10$ ) et une organisation n'ayant pas formé de zones ( $\frac{r_e}{r_c} = 1$ ). Dans les deux cas, la relaxation mène à un consensus en fin d'apprentissage. La formation de zones ne semble donc pas intervenir dans la qualité de la convergence de la relaxation.

reprenons d'une part l'organisation obtenue en section 5.2.2, dans laquelle les poids contextuels sont organisés en pseudo-périodes. Nous la comparons à un apprentissage réalisé sur les mêmes entrées, mais dans lequel les poids contextuels n'ont pas formé cette organisation périodique :  $\frac{r_e}{r_c} = 1$ . Nous traçons en figure 5.15 les indicateurs de la convergence de la relaxation : en haut, nous représentons l'évolution du nombre de pas moyens nécessaires à la relaxation sur une phase de test, pour des tests réalisés régulièrement au cours des 1000 premiers pas d'apprentissage. En bas, nous traçons le pourcentage d'entrées ayant mené à une convergence de la relaxation au cours de ces mêmes phases de test, chaque phase de test contenant 1000 points. Dans les deux cas, les phases de test en fin d'apprentissage convergent dans plus de 95 % des cas. Cette convergence s'effectue en moyenne en une dizaine de pas de relaxation. Les deux échelles d'organisation des poids ne semblent donc pas avoir d'influence sur la relaxation.

### 5.2.5 Discussion

L'observation des cartes sur ces exemples fait apparaître des motifs pseudo-périodiques formés par les poids contextuels. Ils forment une deuxième échelle d'organisation, s'ajoutant aux poids externes dont l'organisation est similaire à celle d'une carte auto-organisatrice classique qui aurait appris sur les mêmes entrées externes. Ces deux échelles d'organisation sont un mécanisme qui

## 5.2. Identification des mécanismes d'apprentissage dans une architecture de deux cartes

---

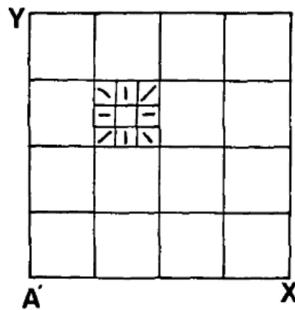


FIGURE 5.16 – Schématisation d'une répartition en indices primaires et secondaires des neurones d'une aire corticale, tirée de [Ballard 1986](#). Les auteurs observent que la réponse des neurones du cortex est organisée en zones selon leur position, formant des indices primaires. Les carrés de la figure représentent cette première indexation. Ces zones reçoivent différentes portions de l'espace d'entrée : les zones situées à gauche du cortex traitent les signaux venant du champ visuel de gauche et ainsi de suite pour couvrir tout le champ visuel. Au sein d'une zone, les neurones cartographient toutes les valeurs possibles de l'entrée sous forme de carte topologiquement ordonnée, formant une indexation secondaire.

émerge du processus d'évolution des cartes. Les tracés des réponses des cartes montrent que les BMU se répartissent en zones distinctes, formées à partir des motifs de poids contextuels. Ces zones sont séparées par des zones mortes faisant office de transition et introduisant des discontinuités dans la réponse des cartes.

La formation de ces zones dépend des paramètres de l'architecture  $r_c$  et  $r_e$  :  $r_c$  doit être inférieur à  $r_e$  pour que la carte permette la formation de telles zones. Ensuite, leur nombre dépend surtout du rapport entre ces deux rayons de voisinage, et peu de l'organisation des entrées : il s'agit d'une propriété induite par les paramètres de l'architecture, qui force les poids de la carte à s'organiser sur deux échelles. Une zone correspond à un intervalle d'entrée externe ; les poids contextuels s'organisent de manière à former une sous-carte des valeurs possibles que prend l'entrée contextuelle pour cet intervalle d'entrée externe. Finalement, chaque zone de la carte se spécialise pour un intervalle de valeur du modèle d'entrées  $U$ , et non seulement de son entrée externe.

Cette organisation à deux échelles fait apparaître une notion d'indices primaires et secondaires dans la carte, l'indice primaire étant l'indice de la zone dans laquelle se trouve le BMU réagissant à une entrée, et l'indice secondaire la position du BMU au sein la zone. Cette notion d'indices primaires et secondaires est également observée en biologie dans le cerveau. Par exemple, la figure 5.16 présente un schéma d'organisation des neurones du cortex V1 décrit en [Ballard 1986](#). Les auteurs observent que des neurones situés à différents emplacements sur le cortex visuel ne reçoivent pas la même partie du champ de vision ; leur emplacement correspond à la partie du champ de vision traitée (gauche - droite, haut-bas). Ces entrées différencieront forment une indexation  *primaire* de V1. Au sein d'une zone de même indice primaire, les neurones s'organisent de façon à représenter tout le sous-espace des entrées ayant été présenté à cette zone. Cette sous-

carte définit alors des indices secondaires. Dans CxSOM, la même entrée est certes présentée à toute la carte, donc la proximité avec ce modèle biologique est limitée. On peut cependant noter qu'après apprentissage, un nœud de la carte ne réagit qu'à un sous-ensemble d'entrées définies par les valeurs des poids externes autour cet emplacement, ce qui se rapproche de ce phénomène de spécialisation d'une zone de neurones observé en biologie.

D'un point de vue des mécanismes de calcul, l'organisation d'une carte s'apparente à une technique de modulation : la valeur de l'activité externe est modulée par l'activité contextuelle. Ici, la forme de poids contextuels induit une activité contextuelle pseudo-périodique, de même forme que les poids contextuels, qui vient moduler l'activité externe au sein de l'activité globale. L'alternance des valeurs hautes et basses dans cette activité contextuelle permet d'encoder en une même valeur, à savoir le BMU  $\Pi$ , à la fois l'entrée externe  $X$  et l'entrée contextuelle  $\gamma$ . Cette modulation émerge de la dynamique d'apprentissage des cartes.

### 5.3 Génération de modalité dans des architectures de trois cartes

Nous avons vu que dans une architecture de deux cartes, chaque carte encode la totalité du modèle d'entrées  $U$  et non seulement son entrée externe.  $U$  est donc encodé dans les deux cartes de l'architecture, apportant une redondance dans l'information apprise par l'architecture sur le modèle d'entrées.

Nous voulons maintenant qu'une architecture de cartes soit directement capable d'utiliser cet encodage du modèle d'entrées dans une tâche de génération de modalité manquante après apprentissage. Même sans qu'une entrée externe ne lui soit présentée, une carte de l'architecture possède une activité contextuelle et donc un BMU. Sur l'architecture de deux cartes, il est envisageable de ne pas présenter  $X^{(2)}$  à la carte  $M^{(2)}$ . La carte  $M^{(2)}$  pourra quand même définir un BMU grâce à son entrée contextuelle. Son poids externe  $\omega_e(\Pi^{(2)})$  appartient à l'espace de la modalité  $X^{(2)}$ , donc la valeur  $\omega_e(\Pi^{(2)})$  peut être considérée comme une valeur générée de cette modalité manquante.

Nous nous plaçons dans un cadre de prédiction d'une modalité manquante : comme l'architecture a appris de l'information redondante sur le modèle d'entrées  $U$ , nous nous attendons à ce qu'elle soit capable de prédire une modalité qui n'a pas été présentée, à partir des autres modalités.

Dans le cas du cercle en 2D, il y a deux valeurs possibles  $X^{(2)}$  pour une même valeur de  $X^{(1)}$ , donc il manque de l'information pour faire de la prédiction. Nous nous intéressons dans cette partie à des modèles d'entrées en trois dimensions dans lesquels la connaissance de  $X^{(1)}$  et  $X^{(2)}$  détermine la valeur de la troisième entrée  $X^{(3)}$ . Nous construirons des architectures de trois cartes sur ces deux modèles d'entrées et vérifierons la capacité de prédiction d'une

### 5.3. Génération de modalité dans des architectures de trois cartes

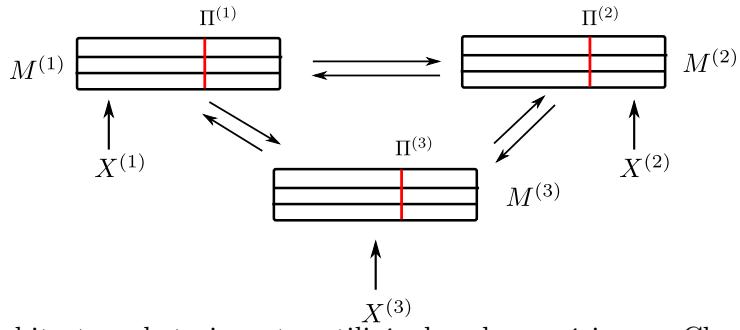


FIGURE 5.17 – Architecture de trois cartes utilisée dans les expériences. Chaque carte prend une entrée externe  $X^{(i)}$  et est connectée aux deux autres. Elle possède ainsi deux couches de poids contextuels et une couche de poids externes.

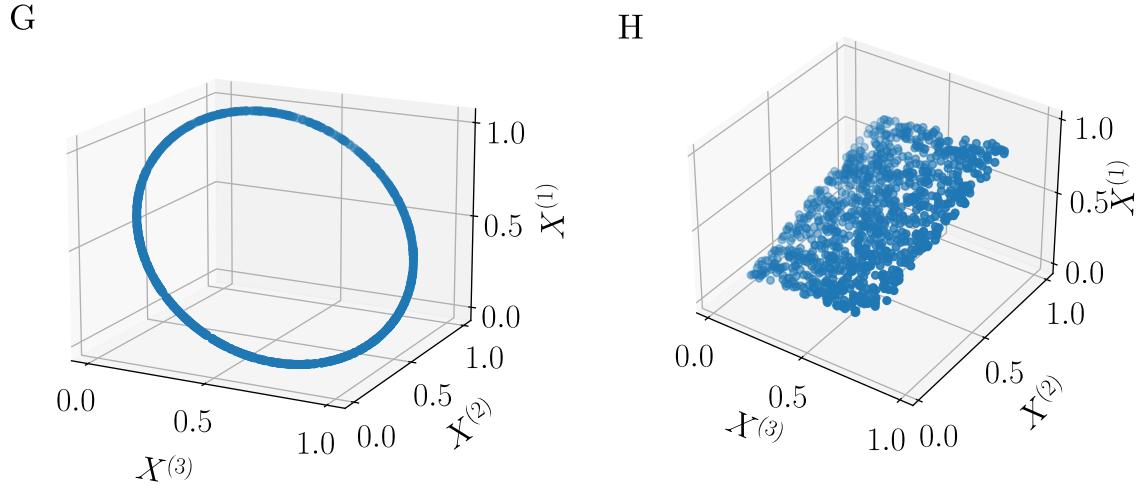


FIGURE 5.18 – Dispositions d’entrées en trois dimensions utilisées dans cette partie. Les entrées **G** sont sur un cercle qui a été pivoté dans l’espace en trois dimensions. Les entrées **H** sont sur un plan pivoté dans l’espace 3D. Chaque carte  $M^{(1)}, M^{(2)}, M^{(3)}$  prend en entrée une coordonnée  $X^{(1)}, X^{(2)}, X^{(3)}$ .

modalité. L’étude d’architectures de trois cartes nous permettra également de vérifier comment les propriétés d’organisation observées sur deux cartes s’appliquent sur une architecture de trois cartes.

Cette capacité de génération d’entrée peut constituer un cadre applicatif, par exemple lorsqu’un capteur serait manquant en robotique, ou pour donner à une carte de l’architecture un rôle de prise de décision et non seulement de réaction à une entrée. Elle permet également de valider l’encodage du modèle appris par une architecture de cartes sans avoir à connaître le modèle  $U$ .

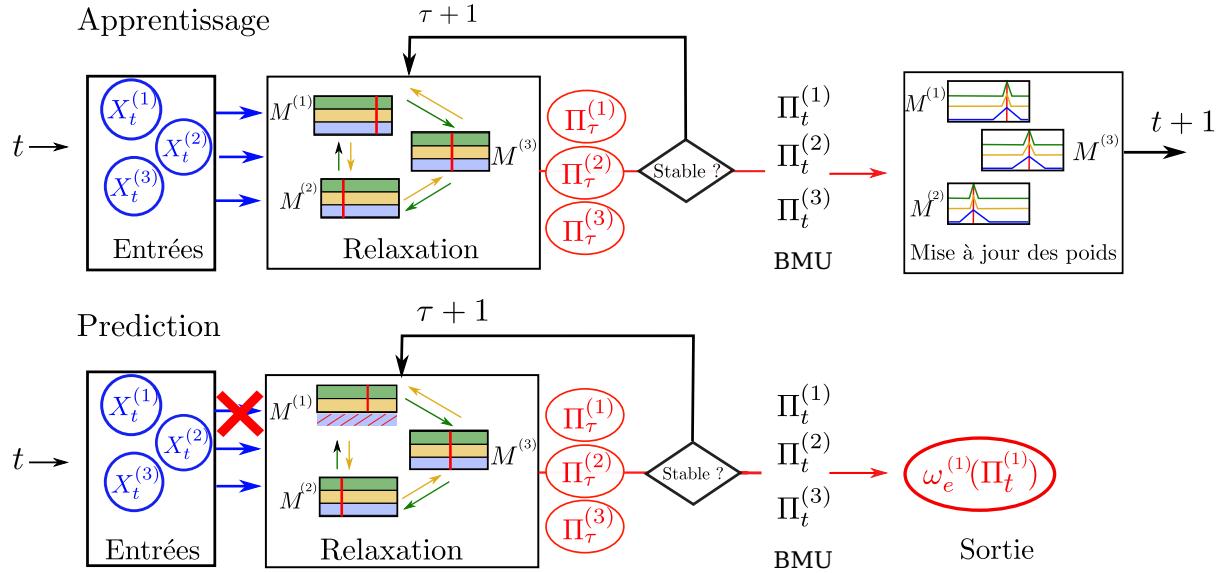


FIGURE 5.19 – Schéma descriptif des opérations effectuées lors de l'apprentissage et de la phase de prédiction. Après une phase d'apprentissage classique, la phase de prédiction est une phase de test durant laquelle l'entrée  $X^{(1)}$  n'est pas présentée à la carte  $M^{(1)}$ . Celle-ci ne prend alors plus en compte d'activité externe dans le calcul du BMU par relaxation.

### 5.3.1 Méthode expérimentale

Les tâches de prédiction de ce chapitre seront réalisées sur une architecture de trois cartes 1D, toutes connectées entre elles ; cette architecture est représentée en figure 5.17. Chacune des trois cartes prend une entrée externe en une dimension et deux entrées contextuelles qui sont les positions des BMU des deux autres cartes. Elle possède donc deux couches contextuelles  $\omega_{c_0}$  et  $\omega_{c_1}$ . Nous construisons deux modèles d'entrées jouets à partir du cercle 2D et du patch  $[0, 1]^2$  en leur ajoutant une troisième dimension. Pour cela, nous pivotons le plan 2D dans lequel se situent ces entrées dans un espace en trois dimensions, de telle sorte à ce que la connaissance de deux entrées sur trois détermine la valeur de la troisième entrée. Ces modèles, référencés par **G** et **H**, sont tracés en figure 5.18. Chaque carte de l'architecture prend en entrée une des coordonnées des points 3D du modèle.

L'algorithme de prédiction d'entrée est schématisé en figure 5.19. Lors de la phase d'apprentissage, les trois cartes reçoivent leurs entrées externes  $X^{(1)}, X^{(2)}, X^{(3)}$ . La phase de prédiction est une phase de test, durant laquelle les poids de toutes les cartes ne sont pas mis à jour. Pour cette phase de prédiction, nous choisissons une carte, ici  $M^{(1)}$ , comme carte prédictive, qui ne reçoit plus son entrée externe, mais seulement ses entrées contextuelles. Les autres cartes reçoivent quant à elles leurs entrées externes  $X^{(2)}$  et  $X^{(3)}$ . La seule différence avec une phase de test classique est que la carte prédictive  $M^{(1)}$  prend comme activité globale sa seule activité

contextuelle. La valeur prédictive récupérée en sortie de l'architecture est le poids externe du BMU de la carte prédictive  $\omega_e^{(1)}(\Pi^{(1)})$ .

Pour ces deux modèles d'entrées, nous vérifierons comment se traduit l'organisation à deux échelles des cartes observée en section précédente. Nous effectuerons ensuite une phase de prédiction de l'entrée  $X^{(1)}$  et vérifierons si la valeur prédictive  $\omega_e(\Pi^{(1)})$  est proche de la valeur attendue  $X^{(1)}$ , qui n'a pas été présentée à la carte.

### 5.3.2 Résultats

Nous traçons en figures 5.20 et 5.21 la représentation cartographique des trois cartes après apprentissage, pour chacun des deux modèles d'entrées. Nous observons d'abord que, comme pour deux cartes, les poids contextuels présentent des motifs pseudo-périodiques qui définissent une seconde échelle d'organisation au sein de la carte. Les BMU se répartissent dans des zones distinctes. Les zones observées sur la représentation cartographique de  $M^{(1)}$  montrent que la carte choisit un BMU réagissant à  $X^{(1)}$  de façon à différencier les deux valeurs correspondantes possibles dans le modèle d'entrées ce qui étend le comportement observé sur deux cartes. L'architecture ayant appris sur les entrées **H** présente également une disposition similaire à celle observée sur deux cartes. Dans ces deux exemples, chaque carte a encodé l'ensemble du modèle d'entrées  $U$ , et non seulement sa seule entrée externe. L'information apprise sur  $U$  est bien redondante : nous nous attendons à ce que l'architecture soit capable de prédire  $X^{(1)}$  à partir des seules entrées  $X^{(2)}$  et  $X^{(3)}$ .

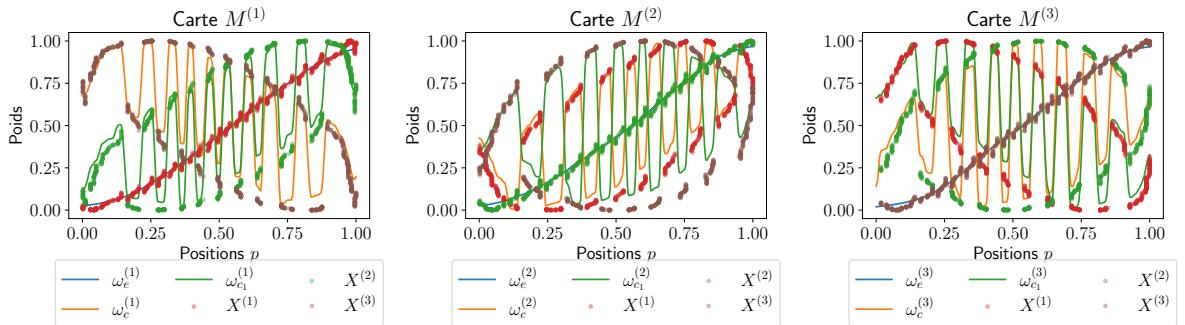


FIGURE 5.20 – Représentation cartographique des poids et entrées dans l'architecture de trois cartes apprenant sur le cercle en trois dimensions **G**. Les motifs pseudo-périodiques des poids contextuels et les zones de BMU sont similaires à celles observées sur l'architecture de deux cartes. Une zone encode un intervalle de valeur de  $(X^{(1)}, X^{(2)}, X^{(3)})$ .

Nous traçons l'erreur obtenue lors d'une phase de prédiction de  $X^{(1)}$  en figure 5.22 pour le modèle du cercle **G**, et 5.23 sur le plan **H**. Nous ajoutons à ces tracés l'erreur de quantification vectorielle obtenue dans les autres cartes qui ont reçu leur entrée externe lors de cette phase de test, afin de comparer la qualité de la prédiction à la qualité de la quantification vectorielle.

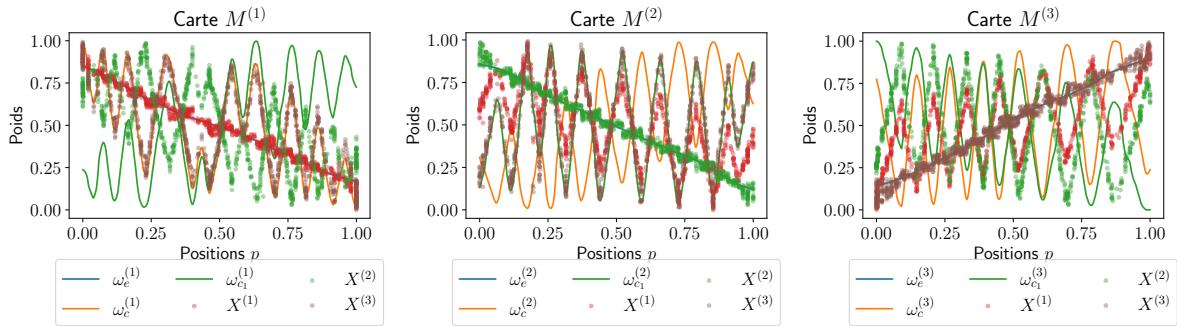


FIGURE 5.21 – Représentation cartographique des poids et entrées dans l'architecture de trois cartes apprenant sur un plan pivoté en 3D. Les zones sont similaires à celles formées par l'architecture de deux cartes.

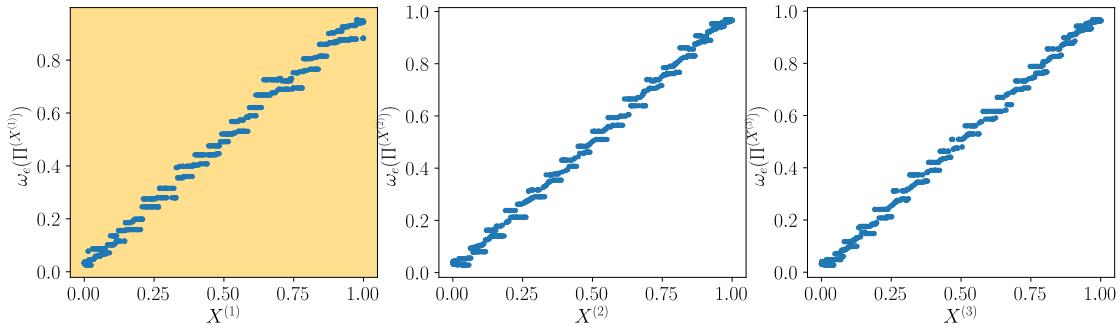


FIGURE 5.22 – Erreur de prédiction de  $X^{(1)}$  par  $\omega_e(\Pi^{(1)})$  lorsque les entrées sont sur le cercle en trois dimensions **G**.  $X^{(1)}$  n'a pas été présenté à  $M^{(1)}$ . Les nuages de points correspondant à  $M^{(2)}$  et  $M^{(3)}$  correspondent à l'erreur de quantification dans les cartes 2 et 3 qui ont reçu leur entrée externe. Ces tracés montrent une bonne prédiction de  $X^{(1)}$  par la carte 1.

Nous observons que la prédiction est très bien réalisée sur les entrées tirées sur le cercle **G**. Les valeurs prédites par la carte  $M^{(1)}$  sont proches de l'entrée théorique  $X^{(1)}$ . Leur prédiction est aussi précise que les valeurs quantifiées par les cartes  $M^{(2)}$  et  $M^{(3)}$ . Dans le cas du plan **H**, la prédiction est également bien réalisée : l'erreur sur la prédiction de l'entrée  $X^{(1)}$  est équivalente à l'erreur réalisée par la quantification vectorielle dans les autres cartes.

La disposition en étages qui apparaît sur les tracés d'erreur de prédiction se rapporte directement à la disposition en zones des cartes de l'architecture. Les entrées contextuelles reçues par  $M^{(1)}$  permettraient donc à une carte de sélectionner une des zones de BMU définies par les poids contextuels. Afin de valider l'importance de cette disposition en zones dans la capacité de prédiction d'entrée, nous nous intéressons à la prédiction réalisée dans le cas où les cartes ne se seraient pas organisées en zones. Pour cela, nous effectuons une phase d'apprentissage et prédiction sur une architecture dans laquelle  $r_c = r_e$ . Nous avions vu que ce choix de paramètres n'engendre pas la formation de zones. L'erreur de prédiction qui en résulte est tracée en figure 5.24. Nous

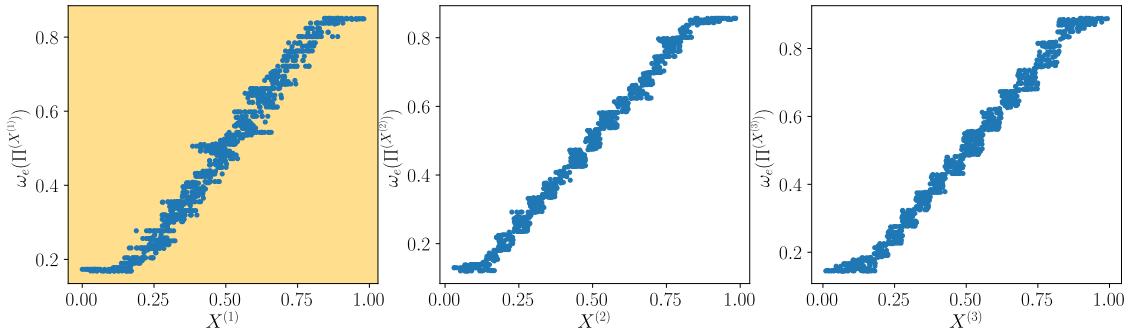


FIGURE 5.23 – Erreur de prédiction de  $X^{(1)}$  dans le cas du plan **H**, montrant une bonne prédiction : l’erreur est équivalente à la quantification vectorielle réalisée par  $M^{(2)}$  et  $M^{(3)}$ . L’architecture de trois cartes est donc capable de prédire  $X^{(1)}$ .

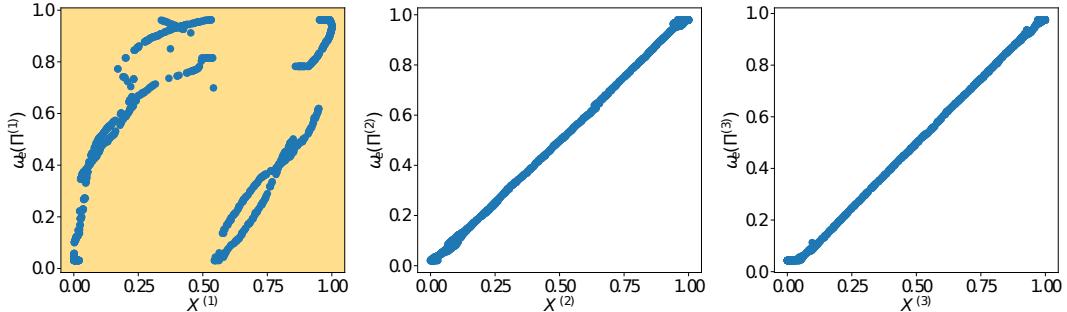


FIGURE 5.24 – Prédiction de l’entrée  $X^{(1)}$  lorsque  $r_c = r_e$ . La prédiction n’est pas effectuée. Ainsi, sans formation de zones, la capacité de prédiction n’est plus réalisable par une carte de l’architecture.

observons que dans cette configuration, l’architecture de cartes n’est pas capable de réaliser la prédiction d’entrée manquante. Bien que cette architecture ait appris les entrées externes du modèle, ne pouvons pas parler d’un apprentissage du modèle, car les cartes ne sont pas capables d’utiliser les relations entre entrées dans la tâche de prédiction. L’organisation à deux échelles des cartes, formant des zones de BMU, permet donc bien à l’architecture de cartes d’encoder les relations entre entrées et de les réutiliser en sortie. Cette organisation est caractéristique de l’apprentissage du modèle d’entrées par l’architecture.

### 5.3.3 Exemple d’application

Nous sortons du cadre des entrées simulées pour nous placer dans un cas de contrôle réel. Dans le cadre de travaux pratiques donnés à CentraleSupélec, nous disposons d’une plateforme expérimentale de contrôle automatique d’un drone quadricoptère. Ce contrôle est réalisé à partir

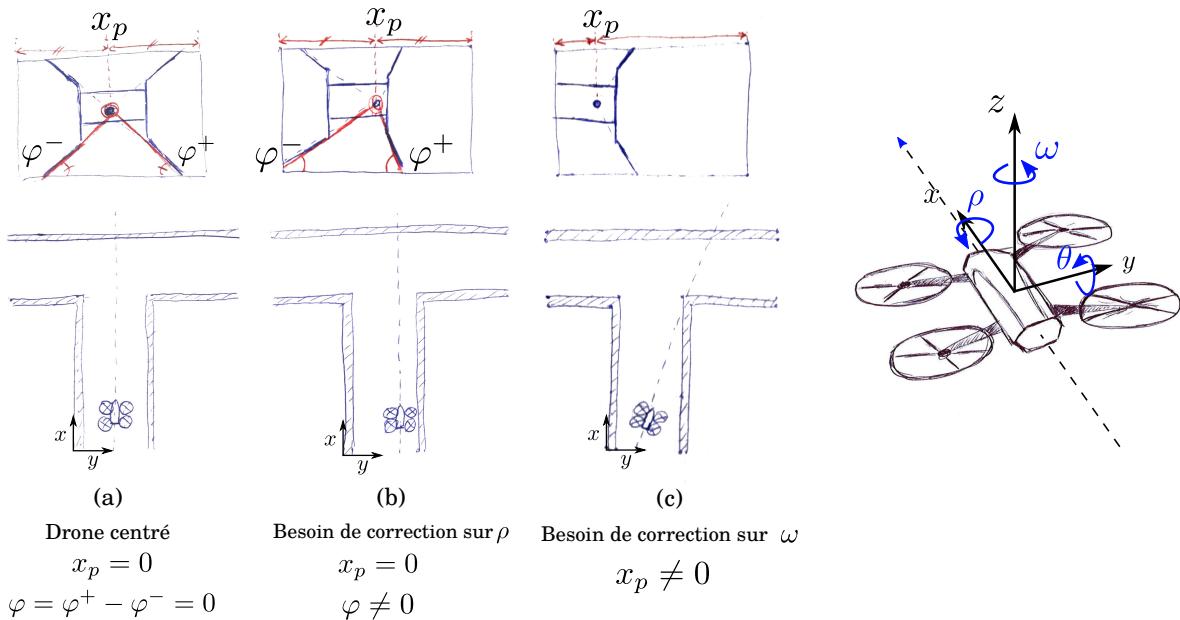


FIGURE 5.25 – À gauche, représentation des situations nécessitant une correction sur la trajectoire du drone. Les images de la caméra frontale correspondant à chaque situation sont représentées en haut. Les commandes du drone sont représentées à droite. Il s'agit de  $\omega$ , la vitesse angulaire autour de  $z$ ,  $\rho$ , l'angle autour de l'axe  $x$  (roulis) et  $\theta$ , l'angle autour de  $y$  (tangage). Pour centrer le drone dans le couloir, ces commandes dépendent de deux valeurs extraites de l'image de la caméra (en haut) :  $x_p$ , abscisse du point de fuite dans le référentiel de la caméra, et  $\varphi = \varphi^+ - \varphi^-$ , la différence entre les angles formés par les lignes de fuite du couloir. Le drone vole en ligne droite lorsqu'il garde  $x_p$  et  $\varphi$  centrés en 0 (a).  $x_p \neq 0$  nécessite une correction sur  $\omega$  (c).  $\varphi \neq 0$  nécessite une correction sur  $\rho$  (b). Enfin, l'accéléromètre du drone nous permet de récupérer la vitesse linéaire du drone selon chaque axe à tout instant,  $v_x, v_y, v_z$ .

du traitement des images issues de sa caméra frontale et de ses capteurs de vitesse internes<sup>1</sup>. Cette section propose un exemple simple d'application de l'architecture en robotique en s'appuyant sur cette plateforme déjà implémentée. Cela nous permet de tester l'apprentissage d'une architecture de cartes sur des entrées bruitées, obtenues en conditions réelles et dont nous ne connaissons pas le modèle de relation.

### Méthode expérimentale

L'objectif de la tâche de contrôle est que le drone se déplace en avant en restant au centre du couloir, malgré les perturbations liées aux turbulences qu'il génère. Il possède une caméra frontale ainsi qu'un ensemble de capteurs de position et vitesse interne et peut être contrôlé à distance par un opérateur.

La figure 5.25 présente les capteurs et commandes que nous voulons utiliser pour la tâche

1. [https://st5drone.metz.centralesupelec.fr/EI/visual\\_processing.html](https://st5drone.metz.centralesupelec.fr/EI/visual_processing.html)

de contrôle. Les commandes envoyées au drone sont les angles de rotation  $\theta$  (tangage)  $\rho$  (roulis) et la vitesse de rotation autour de  $z$ , notée  $\omega$ . À chaque instant, nous extrayons de l'image de la caméra l'abscisse du point de fuite du couloir dans le référentiel de l'image caméra,  $x_p$ , et la différence entre les deux angles formés par les lignes de fuite du couloir,  $\varphi$ . Pour que le drone vole en ligne droite au centre du couloir, on doit avoir  $x_p = 0$  et  $\varphi = 0$ , ce qui correspond au cas (a) sur la figure 5.25. Le centrage de  $x_p$  doit être ajusté en contrôlant  $\omega$ , par exemple dans le cas (c). À  $x_p$  centré, le centrage de  $\varphi$  doit être ajusté grâce à une commande sur  $\rho$ , ce qui correspond au cas (b). Notons que la commande sur  $\rho$  influence en fait l'accélération linéaire du drone selon  $y$  à cause de la structure des hélices : lorsque l'angle est maintenu constant, le drone accélère.

Dans le cadre du TP, nous disposons déjà d'un programme de pilotage automatique du drone lui permettant de rester centré dans le couloir à partir de deux filtres correcteurs sur  $\omega$  et  $\rho$ . La commande  $\omega$  est soumise à un filtre proportionnel à partir de  $x_p$ , afin de le garder centré en 0. La commande  $\rho$  influence l'accélération linéaire selon  $y$ . Elle est donc contrôlée par un filtre proportionnel intégral dérivé qui permet de centrer  $\varphi$  en 0, en prenant également en compte  $v_y$ , la vitesse linéaire selon  $y$  obtenue par les capteurs d'odométrie interne du drone. Ces deux correcteurs créent des relations, à tout instant, entre  $x_p$  et  $\omega$  ainsi qu'entre  $\rho$ ,  $v_y$  et  $\varphi$ . Nous nous retrouvons dans un cas d'entrées multimodales similaires aux entrées jouets que nous avons utilisées. Dans notre expérience, nous voulons apprendre ces relations par une architecture CxSOM, par imitation des trajectoires contrôlées grâce aux filtres. Nous présentons ici le résultat du contrôle de  $\rho$  grâce à CxSOM,  $\omega$  restant contrôlé grâce au correcteur proportionnel. L'architecture que nous avons construite cherche à apprendre les relations entre  $\varphi$ ,  $v_y$  et  $\rho$ , afin de remplacer la partie PID lors de la phase de prédiction.

L'architecture CxSOM que nous avons utilisée est en fait construite sur quatre modalités :  $\varphi$ ,  $v_y$ ,  $\rho$  et  $x_p$ .  $x_p$  n'est a priori pas liée aux valeurs des autres capteurs, car contrôlée par  $\omega$ . Les dépendances entre ces modalités sont présentées en figure 5.26. Nous y voyons que  $\rho$  dépend de  $\varphi$  et  $v_y$  de façon linéaire, mais très bruitée. Nous connectons toutes les cartes entre elles. Elles ont donc chacune une couche de poids externe et trois couches de poids contextuels. Le but de l'architecture sera de capturer les relations entre les capteurs  $\varphi$ ,  $v_y$  et la commande  $\rho$  puis de générer la commande en temps réel lors de la phase de prédiction.

## Résultats

Les poids des cartes obtenus après apprentissage sont représentés en figure 5.27. Constatons d'abord que l'organisation des poids rappelle celle obtenue dans les dispositions d'entrées jouets : les poids externes forment une cartographie ordonnée de l'entrée externe, et les poids contextuels forment des motifs pseudo-périodiques sur la carte. Un exemple de calcul d'activité de la carte  $\rho$  pendant la phase de prédiction est illustré en violet. Le BMU correspond au maximum de

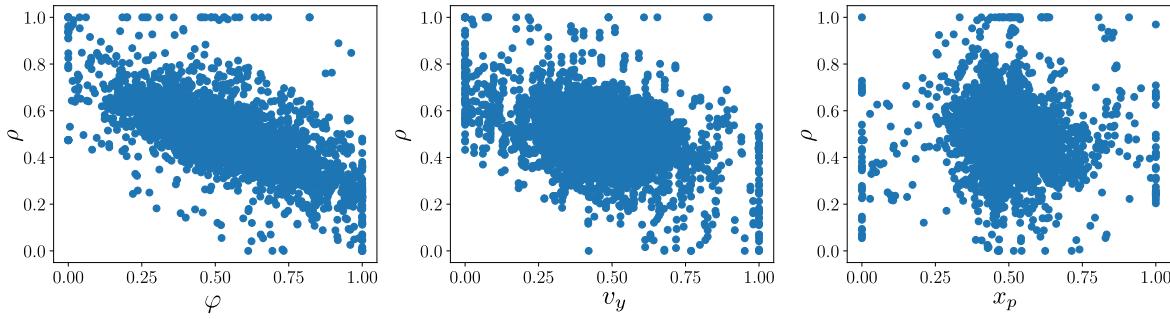


FIGURE 5.26 – Disposition et dépendances des entrées d'apprentissage. Nous chercherons à prédire  $\rho$  : cette valeur dépend bien des autres modalités  $v_y$ ,  $\varphi$  et  $x_p$ . La dépendance est très simple (linéaire) mais très bruitée, et  $\rho$  ne dépend pas de  $x_p$ .

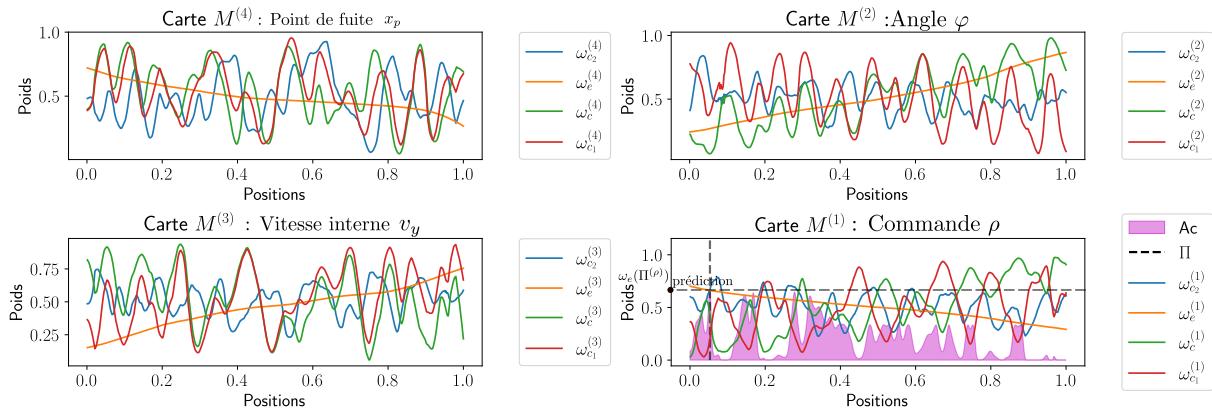


FIGURE 5.27 – Disposition des poids des 4 cartes après apprentissage. La commande prédictive  $\rho$  envoyée au drone est le poids externe du BMU de la carte  $\rho$ , calculé uniquement à partir des activités contextuelles. Cette activité est représentée en violet sur la dernière carte.

l'activité :  $\Pi^{(\rho)} = 0.41$  et la valeur de prédiction est  $\omega_e(\Pi^{(\rho)}) = 0.49$ . Cette valeur, remise à l'échelle, est la commande que nous envoyons au drone à l'instant  $t$ . La relaxation doit être réalisée assez rapidement pour que le drone réagisse en temps réel.

Lors des expériences que nous avons effectuées, le drone apparaît voler correctement dans le couloir sans toucher les murs. Nous avons cependant constaté que cette trajectoire est très imprécise : elle ne lui permet pas de se centrer finement au centre du couloir. Pour illustrer ce comportement, nous avons tracé en figure 5.28 l'erreur de prédiction de  $\rho$ , réalisée sur les données d'apprentissage. La figure fait apparaître une prédiction très bruitée, ce qui correspond à ce que nous avons observé sur le test réel. Elle est moins bonne que la quantification des entrées réalisée dans les autres cartes. Au vu des relations initiales bruitées entre les modalités, nous pouvions nous attendre à une telle imprécision.

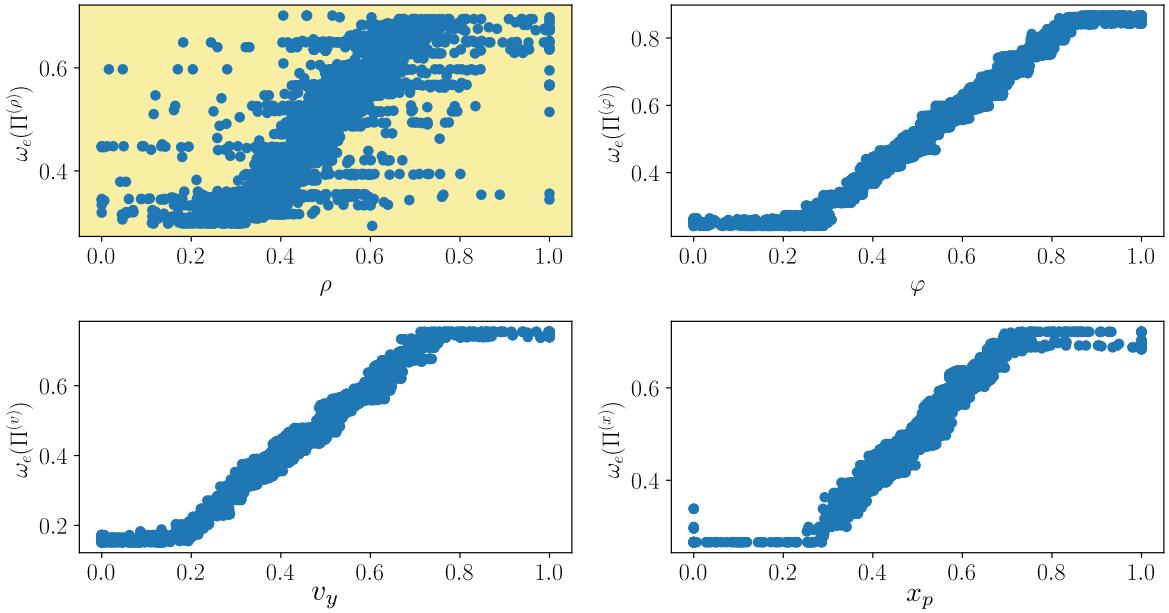


FIGURE 5.28 – Prédiction de  $\rho$  par CxSOM sur les données d’apprentissage. Nous observons que l’entrée a été correctement prédite, bien que très bruitée, ce qui correspond aux observations réalisées en pratique.

Cette application sur une architecture de quatre cartes nous permet d’abord d’étendre les observations réalisées sur deux et trois cartes : les poids externes et contextuels forment deux échelles d’organisation, et les poids contextuels forment des motifs pseudo-périodiques dans la carte. La capacité de prédiction laisse envisager une possibilité future d’application des architectures de cartes sur des données réelles. L’architecture de cartes a en effet détecté une relation entre entrées qui lui permet de prédire une commande à envoyer. De plus, la réactivité de l’envoi de la commande au drone suggère que malgré la relaxation, l’architecture réagit assez rapidement. Cependant, le dispositif expérimental réalisé ici reste à améliorer pour pouvoir envisager une réelle application de CxSOM au contrôle d’un robot. Dans ce sens, il serait intéressant d’étudier une application similaire sur des données moins bruitées et disponibles en simulation afin de mesurer l’erreur de prédiction relative à l’architecture, comme une étape intermédiaire entre les données géométriques et le cadre réel. Une telle application nécessiterait une adaptation fine des paramètres de l’architecture. Nous pouvons imaginer, à plus grande échelle, qu’une architecture de cartes permette à un robot de réaliser une prise de décision par rapport aux valeurs reçues par ses capteurs. La capacité de prédiction laisse aussi envisager des applications relatives au remplacement d’un capteur défaillant ou à l’utilisation de données dans lesquelles il manque parfois l’une ou l’autre des modalités.

### 5.3.4 Conclusion

Cette section présente un comportement induit par les rétroactions dans l'architecture, à savoir la prédiction d'entrée par une carte. Nous avons d'abord montré sur deux expériences jouets qu'une architecture de cartes est capable de générer une modalité manquante dans le modèle grâce aux connexions entre cartes. En effet, chaque carte a encodé la valeur du modèle d'entrées  $U$ , apportant une redondance dans l'information sur  $U$  au sein de l'architecture. Lors d'une telle phase de prédiction, le poids externe du BMU de la carte correspondant à la modalité manquante est utilisé en tant que valeur de prédiction. Sur les deux expériences, cette valeur est bien prédite. L'erreur de prédiction est du même ordre que l'erreur réalisée par la quantification vectorielle dans les cartes qui ont reçu leur entrée externe. Cette capacité de prédiction est permise par l'organisation à deux échelles émergeant des cartes. L'apparition de motifs pseudo-périodiques de poids contextuels et l'organisation en zones de BMU est donc bien un marqueur de l'apprentissage du modèle par une architecture de cartes. Le fait que l'entrée générée corresponde directement à la valeur de l'entrée manquante vient de la quantification vectorielle sur l'entrée externe au sein de chaque carte.

Ce comportement montre que grâce à l'organisation des poids et à la recherche de BMU couplée, une architecture de cartes est non seulement capable d'encoder un modèle d'entrées, mais également de le réutiliser de façon autonome : la génération d'entrée est réalisée par le même algorithme que celui utilisé lors des phases de tests. Par ailleurs, les connexions d'une architecture sont rétroactives, aussi n'importe quelle carte de l'architecture peut être utilisée comme carte prédictive, ce qui constitue un avantage d'une architecture non hiérarchique.

## 5.4 Influence des connexions sur l'apprentissage du modèle d'entrées

Dans toutes les expériences précédentes, nous avons utilisé des architectures dans lesquelles les cartes sont toutes connectées. Or, le nombre d'architectures de cartes possibles augmente exponentiellement avec le nombre de cartes utilisées. L'influence des connexions sur l'apprentissage d'un modèle d'entrées est donc une orientation judicieuse pour des futurs travaux sur CxSOM. Nous présentons dans cette dernière partie deux observations ouvrant des questions sur l'influence des connexions dans une architecture. Nous étudierons d'abord un exemple d'architecture dans laquelle chaque carte possède un grand nombre de connexions. Nous nous intéresserons ensuite à l'influence qu'a une carte sur une autre carte qui ne lui est pas directement connectée.

#### 5.4. Influence des connexions sur l'apprentissage du modèle d'entrées

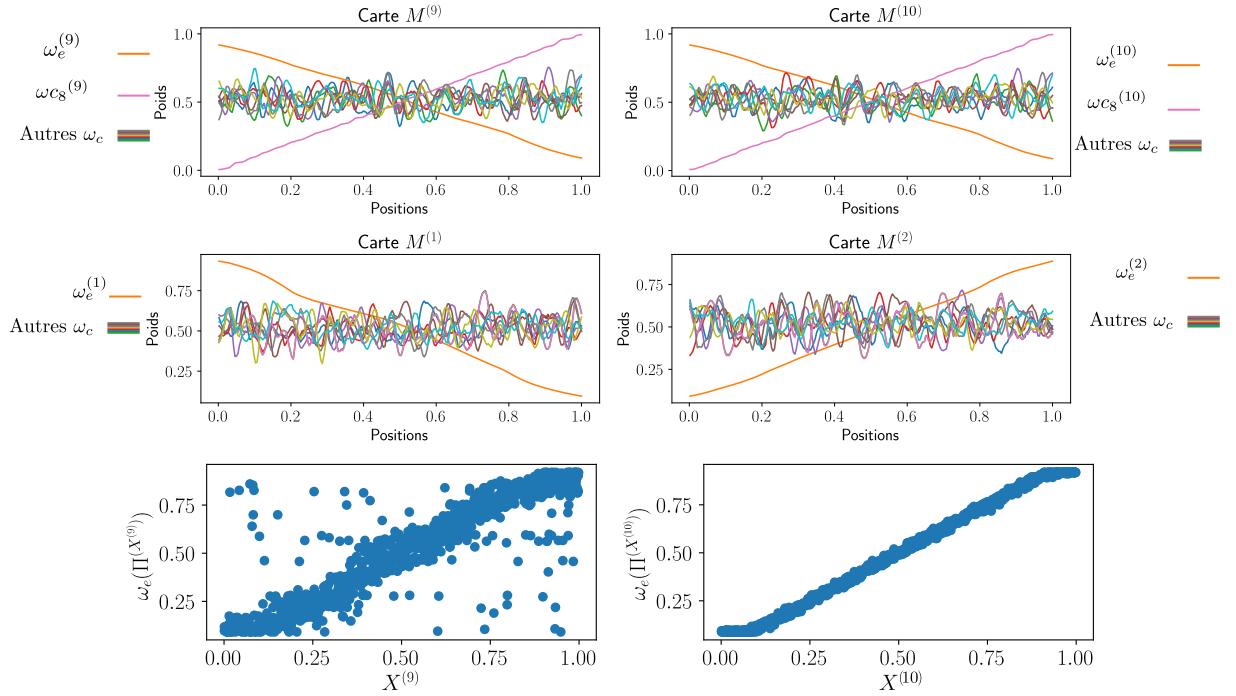


FIGURE 5.29 – En haut, tracé des poids des cartes pour une architecture de 10 cartes toutes connectées. Nous avons ici seulement tracé 4 cartes sur les 10. Les entrées sont toutes indépendantes, sauf les entrées  $X^{(9)}$  et  $X^{(10)}$  qui sont identiques. Les poids contextuels  $\omega_{c8}^{(9)}$  et  $\omega_{c8}^{(10)}$  correspondent à la connexion entre  $M^{(9)}$  et  $M^{(10)}$ . Nous remarquons que les poids contextuels ont évolué vers une valeur moyenne autour de 0.5, sauf ceux correspondant aux entrées dépendantes qui se sont dépliés. En bas, nous traçons l'erreur de prédiction de la carte 9 lorsqu'elle ne reçoit pas d'entrée. La prédiction est assez bien réalisée : les connexions contextuelles inutiles perturbent peu l'encodage du modèle.

##### 5.4.1 Influence d'un grand nombre d'entrées contextuelles sur l'organisation d'une carte

Nous avons vu qu'une architecture de deux cartes apprenant sur le patch  $[0, 1]^2$  se déplie de manière à former deux échelles d'indices. Nous voulons étendre cette expérience pour des entrées de plus grande dimension et une architecture de plus de cartes. Nous choisissons d'effectuer une expérience sur 10 cartes, toutes connectées. Nous utilisons des entrées dans l'hypercube  $[0, 1]^9$ , et ajoutons à ces 9 entrées indépendantes une entrée  $X^{(10)}$  identique à l'entrée  $X^{(9)}$ . Chacune des 10 cartes prend une dimension comme entrée externe. Nous voulons observer deux comportements :

- Comment la présence de nombreuses entrées contextuelles modifient le choix du BMU et l'organisation des cartes ?
- La relation entre les entrées  $X^{(9)}$  et  $X^{(10)}$  est-elle encodée au sein des cartes, ou cette relation est « cachée » par toutes les connexions relatives à des entrées indépendantes ?

La figure 5.29 présente la forme des poids de quatre des cartes dans l'architecture de 10 cartes : les cartes  $M^{(9)}$  et  $M^{(10)}$ , dont les entrées externes sont identiques, et les cartes  $M^{(1)}$  et

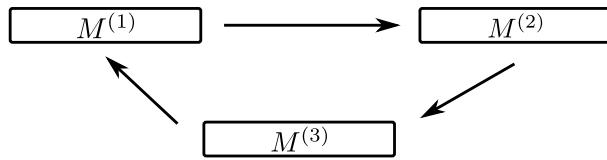


FIGURE 5.30 – Schéma de connexions d'une architecture en « boucle ».

$M^{(2)}$ , dont les entrées sont indépendantes de toutes les autres entrées. Nous pouvons noter que les poids contextuels de chaque carte présentent des motifs périodiques, rappelant ceux observés sur le plan 2D en figure 5.10. L'organisation à deux échelles entre les poids externes et contextuels est donc toujours observée sur une architecture de 10 cartes. Cependant, ces motifs sont de faible amplitude et se rapprochent tous d'une valeur moyenne constante dans chaque carte. Ces couches de poids contextuels interviennent donc peu dans le calcul de l'activité globale, et donc dans le choix du BMU. On peut supposer qu'en augmentant le nombre de connexions, les poids contextuels tendront vers une valeur constante autour de 0.5. Dans ce cas, ils n'interviendraient donc plus du tout dans le choix du BMU.

Les couches de poids contextuels correspondant aux deux cartes dont les entrées sont identiques  $X^{(9)}$  et  $X^{(10)}$  sont représentées en rose dans  $M^{(9)}$  et  $M^{(10)}$ . Elles se déplient totalement, comme nous l'avions observé en figure 5.7 sur les entrées identiques. Cette couche de poids sera ainsi la seule à réellement intervenir dans le choix du BMU. L'architecture de 10 cartes a donc encodé la relation existant entre  $X^{(9)}$  et  $X^{(10)}$  malgré les autres connexions contextuelles inutiles. En bas de la figure, nous traçons également la prédiction de l'entrée  $X^{(9)}$ . Cette prédiction est correctement réalisée, bien que plus bruitée que dans le cas d'une seule connexion : les connexions inutiles perturbent peu l'apprentissage des relations entre entrées. Finalement, les cartes  $M^{(9)}$  et  $M^{(10)}$  ont appris à effacer du calcul de l'activité les entrées indépendantes de leur entrée externe, et gardé les entrées contextuelles qui ont une relation avec leur entrée externe.

Cette expérience propose une idée de comportement à grande échelle à explorer plus en détail. En particulier, on peut se demander si ce comportement peut permettre d'extraire automatiquement des relations entre entrées. Cette détection de relations serait une possibilité de comportement émergent d'une grande architecture de cartes.

#### 5.4.2 Influence des connexions distantes sur l'organisation d'une carte

Nous cherchons maintenant à observer comment une carte d'une architecture peut influencer une carte qui ne lui est pas directement connectée. Dans cette deuxième expérience, nous repassons sur une architecture de trois cartes 1D et reprenons comme modèle d'entrées le cercle en trois dimensions ( $\mathbf{G}$ ). Nous voulons comparer le comportement d'une architecture dans laquelle les cartes sont connectées réciproquement, présentée plus haut, à celui d'une architecture de trois cartes connectées en boucle, dans laquelle chaque carte est uniquement connectée à une seule

#### 5.4. Influence des connexions sur l'apprentissage du modèle d'entrées

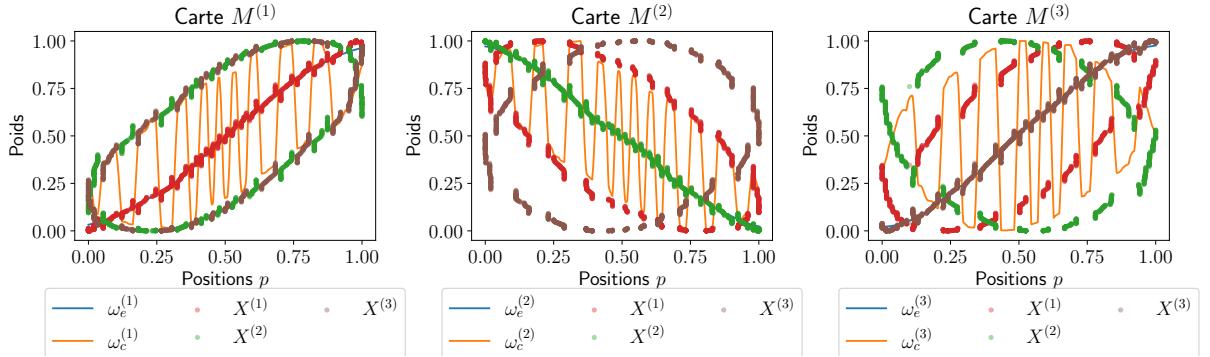


FIGURE 5.31 – Poids à l'issue de l'apprentissage dans une architecture de 3 cartes connectées en boucle. Les poids contextuels forment des zones de BMU similaires à celles observées dans l'architecture avec des connexions réciproques.

autre carte. Ce modèle d'architecture est illustré en figure 5.30 :  $M^{(1)}$  nourrit  $M^{(2)}$ ,  $M^{(2)}$  nourrit  $M^{(3)}$  et  $M^{(3)}$  nourrit  $M^{(1)}$ .

La disposition des poids de cette architecture après apprentissage est tracée en figure 5.31. Les poids des cartes montrent une organisation similaire au cas où les connexions sont réciproques, cf. figure 5.20. Les poids contextuels forment des motifs pseudo-périodiques, et chaque carte différencie ses BMU en fonction du modèle d'entrées  $U$  et non seulement de son entrée externe. D'après les éléments d'étude que nous avons établis précédemment, nous pourrions donc dire que l'architecture a encodé le modèle d'entrées dans chaque carte.

Nous nous demandons si cette architecture en boucle est capable de prédire  $X^{(1)}$ . La carte prédictive  $M^{(1)}$  ne reçoit alors que l'entrée venant de  $M^{(3)}$  :  $M^{(2)}$  intervient seulement de façon distante dans le calcul de l'activité de  $M^{(1)}$ . Afin de dissocier l'influence de  $M^{(2)}$  et  $M^{(3)}$  dans la prédiction de  $X^{(1)}$ , nous réalisons également une phase de prédiction effectuée uniquement à partir des cartes  $M^{(3)}$  et  $M^{(1)}$ , sur les mêmes dispositions de poids que dans l'architecture en boucle. Dans le premier cas,  $M^{(2)}$  contribue à la prédiction via  $M^{(3)}$  ; elle n'intervient pas dans le second cas.

En figure 5.32, nous représentons la valeur de la prédiction dans l'architecture de trois cartes en fonction de l'entrée théorique  $X^{(1)}$ , comparée à une prédiction effectuée uniquement grâce à  $M^{(3)}$ . Nous remarquons d'abord que la prédiction est moins bien réalisée dans l'architecture en boucle, à gauche, que dans l'architecture avec rétroaction (cf. figure 5.22). Les valeurs prédites ayant une erreur de moins de 0.1 par rapport à leur entrée théorique, marquées par les points rouges, représentent seulement 55 % des 2000 entrées présentées lors du test. L'architecture en boucle n'a donc pas aussi bien appris le modèle d'entrées que dans le cas où les trois cartes sont connectées réciproquement, dans laquelle plus de 99 % des valeurs sont prédites avec une erreur de moins de 0.1. Cependant, la prédiction est mieux réalisée dans l'architecture en boucle

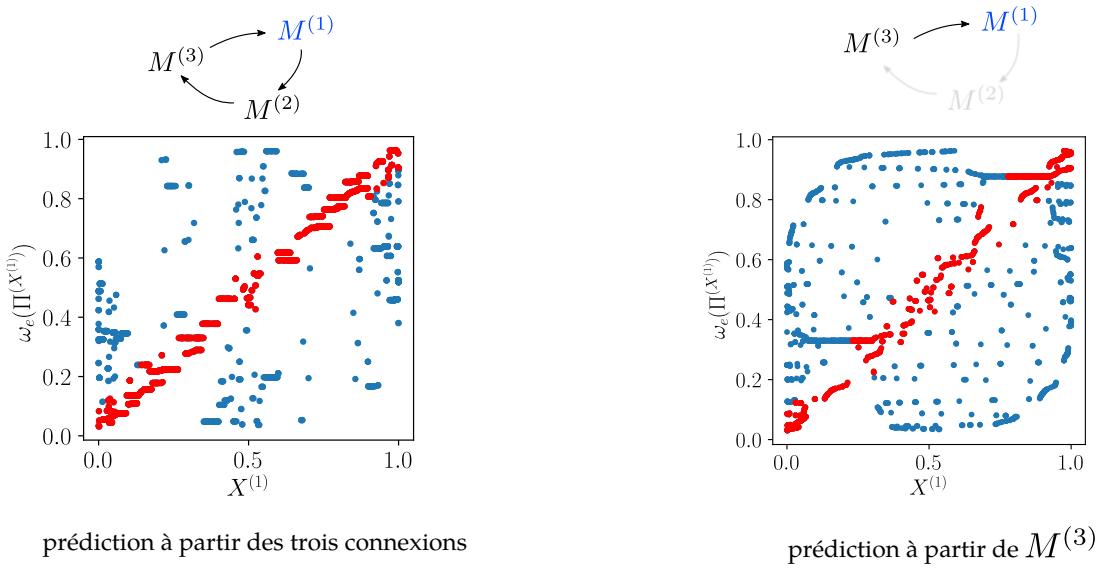


FIGURE 5.32 – À gauche, erreur de prédiction de l'entrée  $X^{(1)}$  par l'architecture en boucle. Les points marqués en rouge représentent les valeurs correctement prédites à moins de 0.1 près. À droite, nous représentons l'erreur de prédiction réalisée à partir des mêmes poids de cartes en ne prenant en compte que  $M^{(1)}$  et  $M^{(3)}$ . La prédiction est moins bien réalisée que dans le cas de gauche, ce qui montre que  $M^{(2)}$  a bien une influence sur le calcul du BMU de  $M^{(1)}$  via  $M^{(2)}$ .

que lorsque nous utilisons seulement  $M^{(3)}$  pour la prédiction. Dans ce dernier cas, seules 40% des entrées ont été correctement prédites par le couple de cartes ; l'erreur est globalement plus importante. Cette observation montre qu'une carte conserve une influence à distance dans une architecture. Par contre, cette influence est très réduite par rapport à l'influence d'une connexion directe.

Nous pouvons conclure de ces deux observations que la disposition des connexions est un point important pour la construction d'architectures. Les cartes ont peu d'influence à distance. Il faudra donc s'assurer que les modalités dont nous voulons apprendre les relations soient directement connectées. Cette observation est également prometteuse : on peut ainsi envisager de traiter plusieurs flux d'information dans différentes parties d'une architecture, sans que le comportement d'une de ces parties n'affecte complètement les autres cartes. A contrario, la présence de nombreuses connexions entre cartes semble moins limitante, l'architecture semblant être capable, grâce aux règles d'organisation, de moins prendre en compte les connexions venant d'une carte qui n'a pas de relation avec son entrée externe. Une possibilité d'étude de la connectivité serait également de construire des architectures adaptant leurs connexions au cours de l'apprentissage.

## 5.5 Conclusion

Ce chapitre de résultats présente une étude de l'organisation d'architectures de 2 et 3 cartes en une dimension. Le but de ce chapitre est d'identifier les mécanismes d'organisation propres à CxSOM qui témoignent d'un apprentissage associatif du modèle d'entrées. Nous avons réalisé cette étude sur des données 1D afin de pouvoir en tracer une représentation facilement interprétable.

Nous avons d'abord observé que l'apprentissage du modèle d'entrées dans une architecture CxSOM est marqué par une organisation à deux échelles : les poids externes se déplient sur l'espace d'entrée comme les poids d'une carte classique. Les poids contextuels se déplient sur des sous-régions de la carte, définies par la valeur de l'entrée externe, formant des motifs pseudo-périodiques : la forme de ces périodes est similaire, mais varie en largeur spatiale et en amplitude. Cette organisation à deux échelles spatiales se retrouve dans l'organisation des BMU de chaque carte. Ces derniers se disposent dans des zones distinctes, séparées par des zones mortes. Cette organisation permet d'encoder l'information sur tout le modèle d'entrées dans chacune des cartes. Nous avons en effet constaté qu'une zone de BMU se spécialise pour un intervalle de valeur du couple  $(X^{(1)}, X^{(2)})$  ou  $(X^{(1)}, X^{(2)}, X^{(3)})$  en 3D, représenté par  $U$ . L'apprentissage des relations entre entrées est alors marqué par le fait que  $U$  est une position du BMU II dans chaque carte.

La notion de réponse à deux échelles se rapproche par ailleurs de certains modèles biologiques du cortex. Cette organisation des cartes dépend des paramètres d'apprentissage, mais n'émerge que lorsque plusieurs points du modèle d'entrées partagent la même valeur sur une des modalités. L'organisation en zones est caractéristique de l'apprentissage du modèle d'entrées par une architecture de cartes. Ce comportement est observé sur plusieurs jeux de données d'entrées jouets ainsi que réelles, sur des architectures de deux et trois cartes, et quel que soit le nombre de connexions entre cartes.

Nous avons ensuite constaté que l'encodage du modèle d'entrées complet dans chacune des cartes, effectué grâce à l'organisation à deux échelles, permet à l'architecture de prédire des valeurs d'entrées manquantes. Après apprentissage, une carte qui ne reçoit plus son entrée externe peut définir son BMU uniquement grâce à ses activités contextuelles. Le poids externe du BMU peut être utilisé comme une valeur générée de la modalité qui n'a pas été présentée. Dans ces tâches de prédiction, l'architecture CxSOM utilise le fait qu'elle ait encodé le modèle d'entrées de façon redondante entre les cartes. Cette capacité de prédiction est une application possible des architectures de cartes. Il s'agit également d'une façon d'évaluer l'apprentissage d'un modèle d'entrées par l'architecture lorsque ce modèle n'est pas connu en théorie. La prédiction est un comportement qui différencie une carte simple ou une architecture hiérarchique de cartes d'une architecture non hiérarchique : les modèles non hiérarchiques présentés au chapitre 1 proposaient également cette capacité de prédiction. Nous voyons ici qu'il s'agit d'une capacité également per-

mise par CxSOM. Dans notre cas, l'information n'est pas centralisée dans une carte associative. De plus, une prédiction peut-être réalisée par n'importe quelle carte grâce aux connexions non hiérarchiques, et s'appuie sur le même algorithme que celui utilisé lors des tests.

Maintenant que nous avons observé les comportements d'apprentissage dans des petites architectures de 2 et 3 cartes, une perspective principale d'étude de CxSOM est la construction d'architectures comportant plus de cartes. Dans ces architectures, le choix des connexions entre cartes constitue un degré de liberté supplémentaire. Nous avons présenté pour cela deux expériences questionnant l'influence des connexions entre les cartes. Nous avons vu que des connexions distantes permettent toujours à l'architecture de cartes d'apprendre un modèle d'entrées. Cependant, lors de la prédiction, les cartes distantes de la carte prédictive ont moins d'influence que les cartes qui lui sont directement connectées. Ensuite, nous avons observé que la présence de nombreuses entrées contextuelles amène les poids contextuels à s'organiser vers une valeur moyenne des entrées. Dans le cas où ces entrées seraient indépendantes au sein du modèle d'entrées, ce comportement de moyennage permettrait aux activités contextuelles de réduire leur contribution dans le calcul de l'activité globale ; le BMU ne prendrait alors en compte que les entrées qui présentent une dépendance. Cependant, si  $U$  est de grande dimension, tout couple de modalité  $(X^{(i)}, X^{(j)})$  ne présentera que peu de dépendance, ce qui pourrait également entraîner ce comportement de moyennage. Ce n'est pas souhaitable, car nous voulons pouvoir extraire une représentation de  $U$ , même de grande dimension.

Plus généralement, il est difficilement souhaitable que chaque carte d'une architecture encode tout le modèle d'entrées lorsque  $U$  est de dimension supérieure à 2. Déjà lorsque  $U$  est 2D, comme dans le patch  $[0, 1]^2$ , la qualité de quantification vectorielle des entrées externes est significativement plus faible que celle obtenue sur un modèle  $U$  1D. On attendrait plutôt de l'architecture qu'elle encode le modèle d'entrées de façon certes redondante, mais distribuée entre les cartes de l'architecture. Cette distribution de la représentation de  $U$  n'apparaît pas clairement dans les expériences sur deux et trois cartes, car les architectures sont encore trop petites. Cet aspect distribué de l'apprentissage sera un point à étudier sur des architectures de plus grande taille, et si besoin à corriger en adaptant les paramètres du modèle pour envisager un développement du modèle CxSOM à grande échelle. Il sera par exemple possible de jouer sur les connexions, les paramètres des cartes ou le calcul d'activité.

Ces constats soulèvent qu'il sera pertinent d'évaluer l'encodage du modèle d'entrées dans les cartes par des valeurs indicatrices. La définition de valeurs indicatrices de cet encodage permettrait d'automatiser l'analyse des paramètres d'une architecture, de comparer des expériences entre elles et d'étudier des expériences dans lesquelles les tracés ne sont pas possibles à cause de la dimension des cartes et des entrées. Ces indicateurs permettraient aussi d'étudier la distribution de l'encodage du modèle d'entrées au sein d'une architecture. Nous étudierons quelques indicateurs d'analyse de l'encodage de  $U$  par l'architecture au chapitre 6.

Enfin, toutes ces expériences ont été menées sur des cartes 1D apprenant à représenter des données en une dimension. Ce cas de figure est rarement rencontré en pratique et il serait intéressant d'évaluer l'apprentissage sur des dimensions d'entrées supérieures. Pour une tâche de quantification vectorielle classique par une SOM, il est plus usuel d'utiliser des cartes en deux dimensions qui forment un bon compromis entre la capacité de quantification vectorielle rendue possible et le coût des calculs pendant l'apprentissage. Le passage de 1D à 2D n'est pas immédiat et pose de nombreuses questions quant à l'organisation des poids et la recherche de BMU par relaxation. Nous présenterons à ce propos au chapitre [7](#) des expériences préliminaires étudiant l'organisation d'une architecture de cartes en deux dimensions.

