

Chapitre 5

Analyse de l'auto-organisation de CxSOM sur des cartes en une dimension

Sommaire

5.1	Introduction	98
5.2	Identification des mécanismes d'auto-organisation dans une architecture de deux cartes	98
5.2.1	Modèles d'entrée et architecture de cartes	99
5.2.2	Observations réalisées sur le modèle d'entrées en cercle	101
5.2.3	Évaluation de l'organisation sur les autres distributions d'entrées	105
5.2.4	Étude des mécanismes de formation des zones de poids contextuels	109
5.2.5	Discussion	113
5.3	Génération de modalité dans des architectures de trois cartes 1D	114
5.3.1	Méthode expérimentale	115
5.3.2	Résultats	116
5.3.3	Conclusion	119
5.4	Application de la prédiction d'entrée à la commande d'un drone	120
5.4.1	Méthode expérimentale	120
5.4.2	Résultats	122
5.4.3	Conclusion	123
5.5	Influence des connexions d'une architecture sur l'apprentissage du modèle d'entrée	124
5.5.1	Influence d'un grand nombre d'entrées contextuelles sur l'organisation d'une carte	125
5.5.2	Influence des connexions distantes sur l'organisation d'une carte	126

5.1 Introduction

Le but de cette thèse est d'identifier les comportements d'auto-organisation de l'architecture CxSOM qui traduisent l'apprentissage d'une représentation du modèle d'entrée multimodal et des relations entre les entrées par l'architecture. En effet, avant d'être capable de caractériser l'apprentissage d'une architecture de cartes quelconque, nous voulons ~~à~~ identifier ce qu'apprend une architecture, comment cet apprentissage se traduit sur l'organisation et la réponse de l'architecture, et quels paramètres et éléments de l'architecture influencent cet apprentissage. À partir des éléments de représentation de l'architecture décrits au chapitre 4, nous identifierons dans ce chapitre des dynamiques et comportements d'apprentissage sur des architectures de deux et trois cartes 1D. Nous chercherons d'abord à caractériser l'organisation d'architecture de deux cartes en une dimension selon leur réponse à différentes dispositions d'entrées. Nous introduirons ensuite un comportement spécifique à CxSOM : la génération de modalité par l'architecture. Nous verrons en effet qu'une architecture peut non seulement encoder le modèle d'entrées mais aussi l'utiliser de façon autonome pour prédire une entrée manquante. Nous appliquerons cette propriété de prédiction de CxSOM sur des entrées jouet et des entrées réelles.

5.2 Identification des mécanismes d'auto-organisation dans une architecture de deux cartes

Nous avons choisi de nous intéresser d'abord aux comportements occourant dans un modèle d'architecture non-hiérarchique de deux cartes, dans laquelle chaque carte prend en entrée le BMU de sa voisine. De cette façon, nous isolerons des comportements relatifs à une seule interface entre cartes. Pour faciliter les représentations, nous utiliserons des cartes 1D, prenant chacune une entrée externe en 1D. Nous avons donc un modèle très simple d'architecture, contenant une seule interface entre modules et facile à représenter graphiquement.

Afin d'identifier l'apprentissage des relations entre modalités, nous utiliserons comme modèle d'entrée des points en deux dimensions : chaque modalité est l'une des deux coordonnées x et y d'un point. Ces modèles d'entrées géométriques nous permettent de maîtriser les dépendances entre les modalités et ainsi de cibler des comportements d'organisation spécifiques à un modèle d'entrées.

Dans cette section, nous identifierons d'abord les comportements d'apprentissage sur un modèle d'entrée : le cercle 2D, déjà utilisé au chapitre 4, puis nous vérifierons et étendrons les observations sur d'autres modèles d'entrées. Nous comparerons enfin l'influence des paramètres

d'une carte sur l'apprentissage de l'architecture.

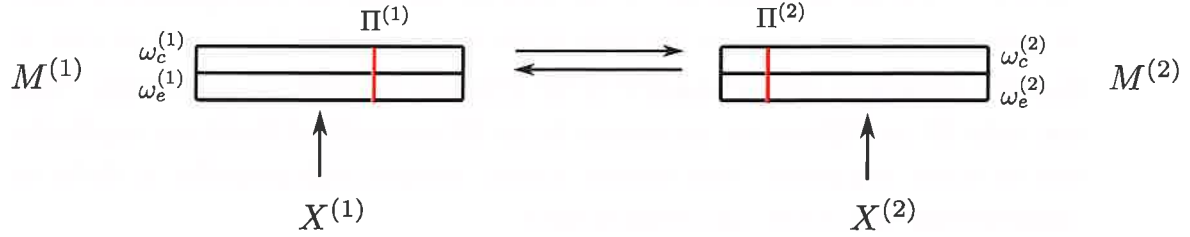


FIGURE 5.1 – Schéma de l'architecture de deux cartes étudiée et rappel des notations. Chaque carte possède une couche de poids externe et une couche de poids contextuels. Les connexions sont rétroactives : l'entrée contextuelle de $M^{(1)}$ est la position du BMU de la carte $M^{(2)}$ et inversement.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

5.2.1 Modèles d'entrée et architecture de cartes

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

Rappelons la définition des entrées multimodales : il s'agit d'un ensemble d'entrées $\mathbf{X} = (X^{(1)}, \dots, X^{(K)})$. Nous représentons la dépendance entre entrées en choisissant une variable latente du modèle U telle que :

$$\forall i, X^{(i)} = f^{(i)}(U) + \epsilon^{(i)}$$

avec $\epsilon^{(i)}$ un bruit sur les entrées. La dimension de U détermine la dépendance entre les entrées.

Dans cette série d'expériences, nous comparons la réponse de l'architecture à différents modèles d'entrées à deux modalités, dans lequel chaque modalité est une valeur 1D. 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 la disposition est symétrique : 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 testerons ensuite si les observations réalisées sur le modèle du cercle se retrouvent pour d'autres dispositions d'entrées, représentées en figure 5.2 :

- Une entrée est une fonction de l'autre : $X^{(2)} = \cos(X^{(1)})$ (B).
- Les entrées sont identiques (C).
- Les entrées sont sur une courbe plus complexe que le cercle, ici une courbe de Lissajous (D) : une entrée $X^{(1)}$ correspond à 4 à 6 valeurs de $X^{(2)}$ et inversement. U est une variable 1D paramétrant la courbe.
- Les entrées sont totalement indépendantes, prises aléatoirement dans le carré $[0, 1]^2$ (E).

U est alors une variable 2D.

- Les entrées sont sur un anneau (**F**). 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 fin en 2D apprendra d'abord une représentation du cercle sous-jacent. Nous voulons vérifier comment cette propriété se vérifie sur l'apprentissage de données par plusieurs cartes.

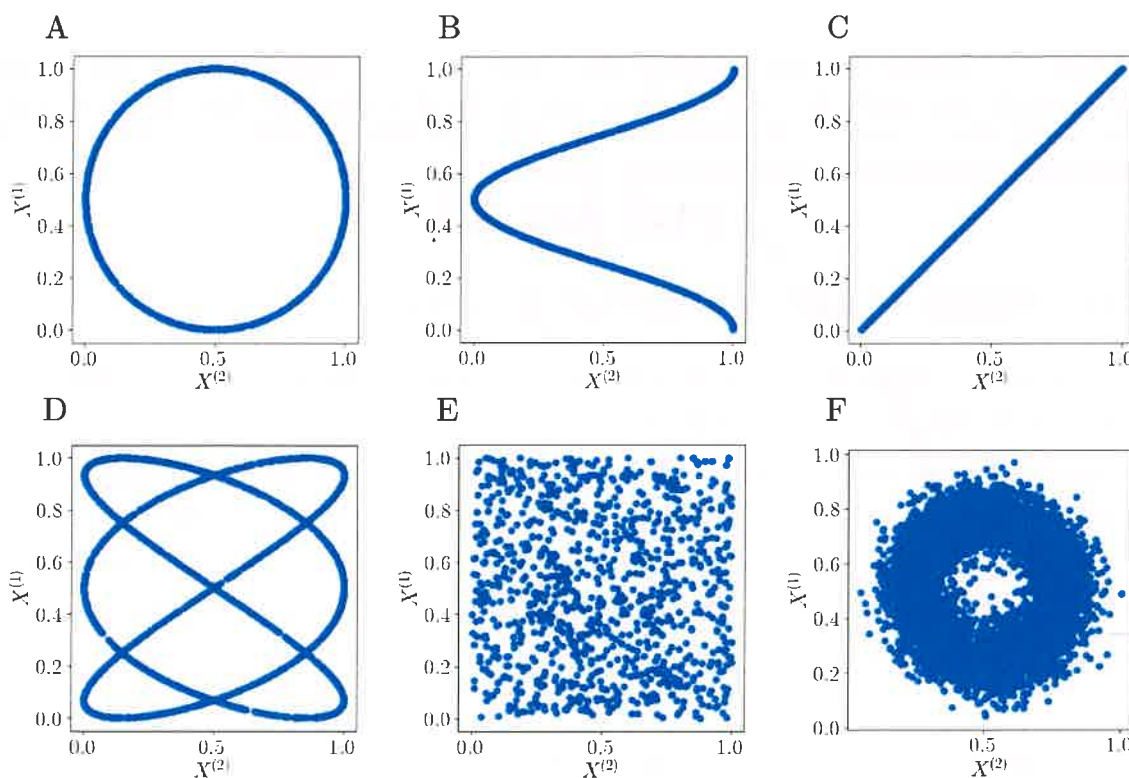


FIGURE 5.2 – Dispositions d'entrées en deux dimensions utilisées dans cette section. $M^{(1)}$ prendra en entrée les valeurs $X^{(1)}$ et $M^{(2)}$ les valeurs $X^{(2)}$.

La génération des entrées suit le processus suivant : U est tiré uniformément dans $[0, 1]$. Le couple d'entrées $(X^{(1)}, X^{(2)})$ généré à partir d'une même valeur de U est présenté à l'architecture lors de la même itération. Lors de chaque expérience, nous réalisons l'apprentissage sur un échantillon de 20000 points, générés aléatoirement et présentés une fois à l'architecture. Les tests sont ensuite réalisés sur 1000 points générés aléatoirement selon la même distribution d'entrées que l'apprentissage. Le code utilisé pour générer les expériences et les représentations est disponible en ligne.³

3. [todo.github.com](https://github.com/todo/todo)

5.2.2 Observations réalisées sur le modèle d'entrées en cercle

Revenons d'abord sur l'expérience précédemment présentée au chapitre 4, réalisée sur les entrées disposées selon un cercle (A). Après avoir vérifié que les poids des cartes convergent bien au cours de l'apprentissage, nous détaillerons la disposition finale des poids externes et contextuels et la réponse de la carte.

Convergence des poids

Dans une carte de Kohonen classique, le rayon de voisinage et le taux d'apprentissage sont diminués de façon prédéfinie au cours des itérations. Cette opération permet d'assurer un dépliement des cartes au début de l'apprentissage puis assure la convergence des poids ω des cartes lorsque les paramètres sont faibles. Dans notre étude, nous choisissons au contraire de ne pas modifier les paramètres d'apprentissage au cours des itérations. Nous avons donc d'abord vérifié que les poids des cartes convergent vers une position stable au cours de l'apprentissage. Sur des cartes 1D, cette propriété a été observée dans toutes les expériences.

Nous illustrons cette convergence en figure 5.3, qui présente l'évolution des variations des poids ω_e et ω_c dans chaque carte au cours de l'apprentissage. Toutes les 100 itérations, nous calculons la moyenne des différences sur toutes les positions p d'une carte entre les poids $\omega_t(p)$ et les poids $\omega_{t-100}(p)$. Cette valeur a été calculée sur 10 apprentissages et nous représentons sur la figure la valeur moyenne et l'écart type de cette différence sur les 10 expériences. Nous pouvons constater que les poids évoluent rapidement au début de l'apprentissage, puis n'évoluent ensuite que très faiblement : à l'itération 10000, chaque prototype $\omega_e(p)$ se déplace en moyenne de 0.05 tous les 100 itérations, et chaque $\omega_c(p)$ de 0.005. Les cartes évoluent donc vers un état dans lequel les poids ne se déplacent que très faiblement. Notons que ces tracés ne permettent pas de montrer expérimentalement une convergence des poids mais donnent une bonne idée de l'évolution générale des poids des cartes.

Nous pouvons proposer des éléments d'explication de cette convergence en remarquant qu'une carte se comporte principalement comme une carte de Kohonen classique apprenant sur les entrées externes. En effet, l'activité externe domine dans le calcul de l'activité globale :

$$a_g = \sqrt{a_e \cdot (\beta a_e + (1 - \beta) a_c)}$$

L'activité contextuelle est utilisée ici comme une modulation d'amplitude de l'activité externe. La convergence des poids d'une carte de Kohonen classique en 1D sur des données numériques a été prouvée en (Cottrell et al. 2016). On peut donc raisonnablement envisager que les cartes CxSOM 1D convergent également. La convergence en l'absence de décroissance de paramètres pourra cependant poser plus de problèmes sur des cartes en deux dimensions.

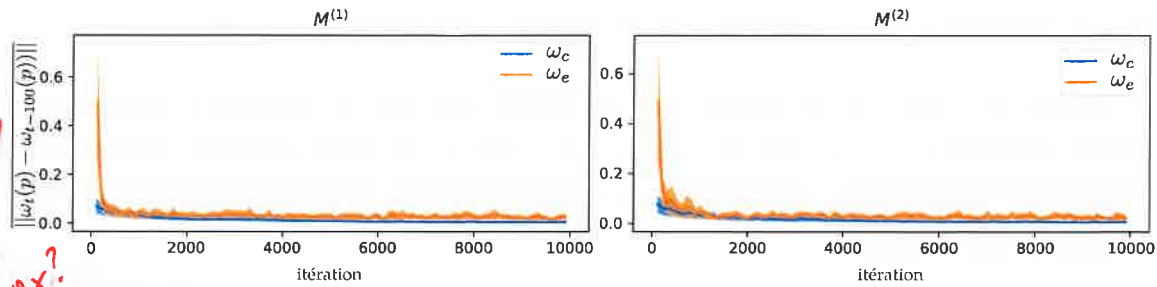


FIGURE 5.3 – Pour chaque carte, nous représentons l'évolution en fonction du temps d'apprentissage de la différence maximale en valeur absolue entre les poids à l'instant t et ceux à $t-100$, et $t-100$. Les entrées sont un cercle en deux dimensions. L'évolution est moyennée sur 10 apprentissages dont les entrées sont tirées aléatoirement selon la même distribution, un cercle en deux dimensions. Ces tracés montrent que les poids externes et contextuels convergent rapidement vers une position dans laquelle ils ne varient plus que faiblement.

Organisation des cartes après apprentissage

Nous nous intéressons ici à l'organisation des poids et la réponse des cartes après apprentissage de la disposition d'entrées **A**. Les poids, les entrées et les BMUs associés sont tracés en figure 5.4 et 5.5 selon la représentation cartographique décrite au chapitre 4. Les poids externes, en bleu, présentent une disposition similaire à ceux observés dans une carte classique : ils sont classés de façon monotone entre 0 et 1. Les poids contextuels, en orange, présentent une forme de « vagues ».

En rouge et vert, nous traçons la valeur des entrées en fonction des positions de BMUs obtenues lors de la phase de test. Nous observons que les positions des BMUs dans la carte $M^{(1)}$ se répartissent en zones, séparées par des zones mortes dont les nœuds n'ont jamais été BMUs. C'est une première différence avec une carte classique, pour laquelle toutes les positions seront BMUs lorsque les entrées sont distribuées de façon continue. Les zones dans lesquelles les nœuds sont BMUs correspondent aux extrema des poids contextuels et leurs alentours.

La figure 5.5 est un agrandissement du tracé sur quatre zones de la carte $M^{(1)}$. Elle met en évidence le fait que les zones définies par les poids contextuels se partagent les positions des BMUs en fonction de la valeur de l'entrée contextuelle : Les deux points représentés en rouge et bleu ont la même valeur de $X^{(1)}$ mais une valeur différente de $X^{(2)}$. Ils ont ici un BMU différent dans la carte $M^{(1)}$. Par ailleurs, ces BMUs sont situés dans deux zones adjacentes. Deux zones adjacentes correspondent à des segments de valeur d'entrée externes qui se recouvrent.

Dans chaque carte, une position se spécialise donc en tant que BMU par rapport à son entrée externe et à son entrée contextuelle. Cette différenciation est réalisée par une organisation de la carte en un nombre fini de zones distinctes. Dans chaque zone, les unités sont BMUs pour un segment de valeurs d'entrée externe et contextuelles. Au sein d'une zone, la répartition des entrées externe selon le BMUs est ordonnée, comme ce serait le cas dans une carte auto-organisatrice classique. Le comportement de la carte au sein d'une zone reste donc similaire à celui d'une carte

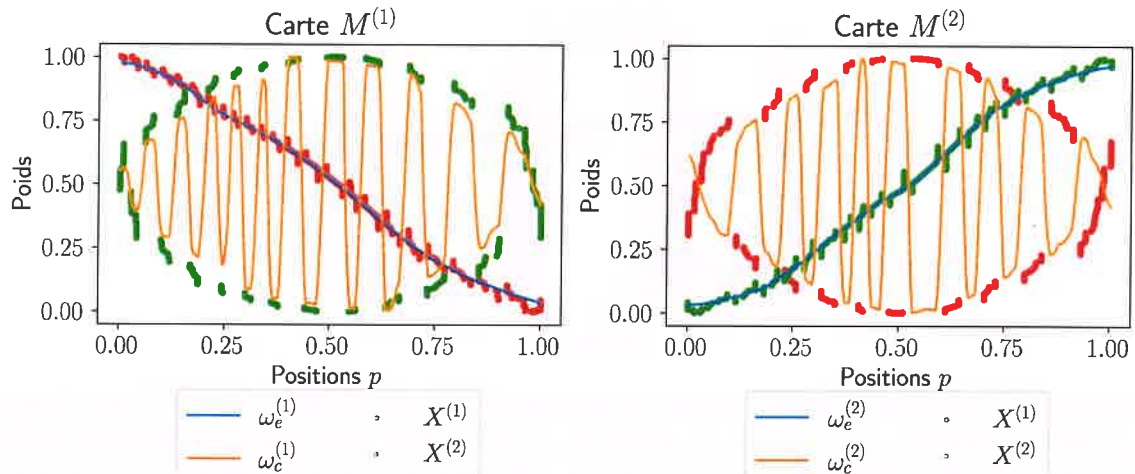


FIGURE 5.4 – Représentation cartographique des poids et entrées lors d'une phase de test selon la position du BMU dans chacune des cartes. Nous remarquons que les poids d'une carte, par exemple la carte $M^{(1)}$, s'organisent en zones différenciant les valeurs de la paire $X^{(1)}, X^{(2)}$ et non seulement 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 BMUs s'organisent sous la forme d'une sous-carte auto-organisée sur les valeurs de l'entrée contextuelle. Ces zones se forment de manière auto-organisée.

classique. Il s'agit d'une deuxième échelle d'organisation, qui garde également l'aspect ordonné d'une carte classique. Ces zones sont créées par auto-organisation ; aucun paramètre de la carte n'a été modifié pendant l'apprentissage pour former ces zones.

Erreur de quantification vectorielle

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

La Figure 5.6 présente la valeur de cette approximation au sein de chaque carte, $w_e(\Pi^{(i)})$, 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 $X^{(i)}$. Il s'agit d'un compromis réalisé par la carte pour pouvoir à la fois encoder son l'entrée externe et son de entrée contextuelle en une seule position de BMU. Nous remarquons enfin une disposition "en étages", dues aux zones formées par les poids contextuels. Les nœuds de deux zones adjacentes de la carte sont en effet des BMU pour des intervalles d'entrée qui se recoupent. Une même valeur d'entrée peut avoir un BMU dans deux zones consécutives de la carte en fonction de la valeur de l'entrée contextuelle. Comme

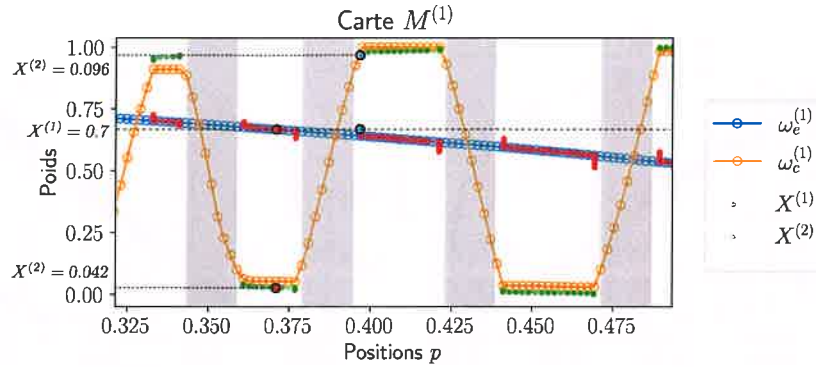


FIGURE 5.5 – Zoom sur la figure 5.4 entre les positions 0.35 et 0.55 de la carte $M^{(1)}$. Nous y faisons apparaître la position sur la courbe des nœuds de la carte. Deux zones consécutives seront BMUs pour des ensembles d'entrées qui se recouvrent. Par exemple, les deux entrées correspondant aux points bleus et rouges ont les mêmes valeurs de $X^{(1)}$, mais des valeurs différentes de $X^{(2)}$. Leurs BMUs 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.

les poids externes sont strictement croissants ou décroissants, cela induit l'erreur observée dans la prédiction d'entrée.

pas d'air

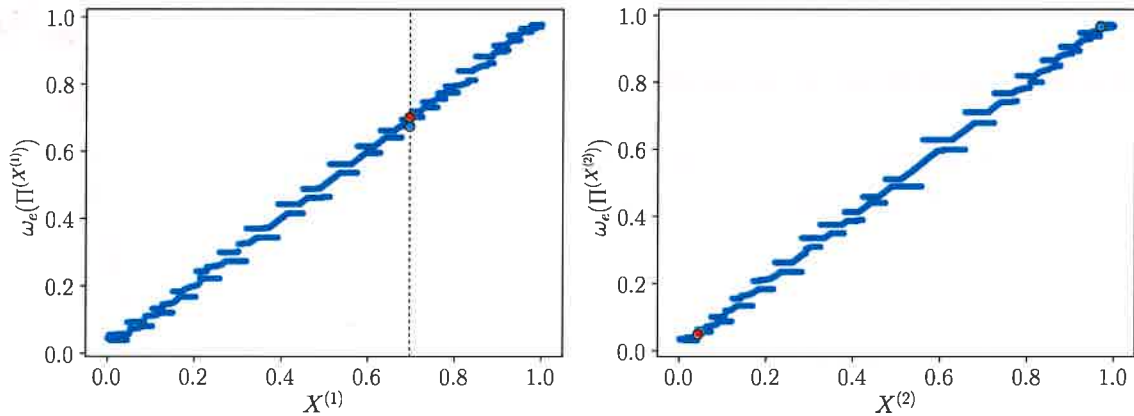


FIGURE 5.6 – Représentation de l'erreur de quantification sur les valeurs de $X^{(1)}$ et $X^{(2)}$. Le poids externe du BMU est proche de la valeur de l'entrée; chaque carte réalise ainsi une bonne quantification vectorielle sur ses entrées. Les poids rouges et bleus représentés en figure 5.5 sont reportés sur le graphique.

Apprentissage du modèle par chaque carte

Enfin, au 4, nous avons observé que l'apprentissage des relations entre entrées se traduit par une relation fonctionnelle entre U et Π dans chaque carte, tracée en figure 7.4. Cette propriété traduit l'observation qu'un BMU se spécialise en fonction de l'entrée externe et des entrées contextuelles.

ake la section.

???

Résumé des observations

Les résultats de cette expérience ainsi que les observations présentées au chapitre 4 nous permettent de formuler les hypothèses suivantes concernant le comportement d'une architecture de deux cartes en une dimension :

- Chaque carte de l'architecture effectue une quantification vectorielle sur ses entrées externes.
- Les poids contextuels de chaque carte définissent des zones distinctes de BMU. Une zone réagira à un même intervalle de valeur pour X et pour U . Deux zones adjacentes encodent le même intervalle de valeurs de X , mais des valeurs distinctes de U . Entre chaque zone de BMUs, nous observons des zones mortes dans lesquelles les nœuds ne sont jamais BMUs.
- L'apprentissage du modèle d'entrée par l'architecture se traduit par l'existence d'une relation fonctionnelle entre U et Π dans chaque carte, montrant que chaque carte encode l'état de toute l'architecture et non seulement l'état des entrées externes.

Nous chercherons à vérifier ces hypothèses sur les autres dispositions d'entrées en deux dimensions, et à compléter les observations. Nous étudierons en particulier si les zones dépendent du modèle d'entrées, comment ces zones se forment et quelles propriétés d'apprentissage elles confèrent à l'architecture de cartes. Nous verrons ensuite en section ?? que la formation de ces zones ainsi que 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 Évaluation de l'organisation sur les autres distributions d'entrées

(fig 5.2)

Nous reprenons les dispositions d'entrées B, C, D, E. Dans toutes ces dispositions, nous avons vérifié que la quantification vectorielle est bien réalisée dans chaque carte sur ses entrées. Nous nous concentrerons ici sur la présence ou non de zones de poids contextuels dans l'organisation finale des poids en fonction de la distribution des entrées.

Dans la disposition d'entrées C, une valeur de $X^{(1)}$ correspond à une seule valeur de $X^{(2)}$. En théorie, l'architecture de cartes n'a pas besoin de séparer les BMUs. La figure 5.7 présente la disposition des poids et entrées des cartes résultant de l'apprentissage. Dans ce cas, les poids externes et contextuels ne forment effectivement pas de zones, ce qui est ce que nous attendions.

En figure 5.8, la dépendance entre les entrées présentées n'est plus bijective : $X^{(2)}$ est toujours fonction de $X^{(1)}$, mais pas l'inverse (Entrées B). Nous observons que la carte $M^{(1)}$ ne forme pas de zones, car une seule valeur de $X^{(2)}$ correspond à une même valeur de $X^{(1)}$, ce qui est cohérent avec le comportement sur les entrées identiques. Au contraire, la carte $M^{(2)}$ doit à présent se diviser pour apprendre les deux valeurs de $\Pi^{(1)}$ possibles correspondant à $X^{(2)}$. Ce comportement rejoint ainsi celui que nous avons observé sur le cercle. La formation de zones semble ainsi intervenir seulement lorsqu'une carte doit séparer au moins deux points du modèle d'entrée ayant la même valeur sur la modalité X .

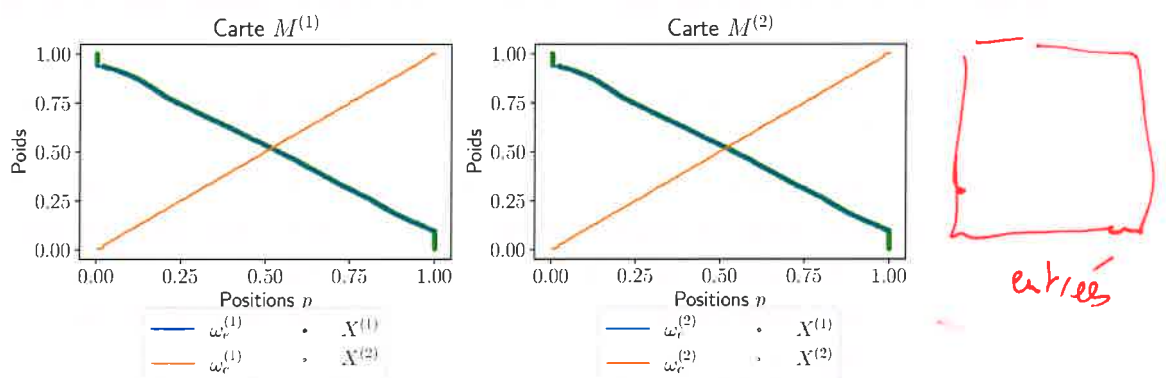


FIGURE 5.7 – Représentation cartographique des poids et entrées pour la disposition identité. Les poids externes et contextuels sont superposés, et les poids contextuels n'ont pas besoin de former de zones

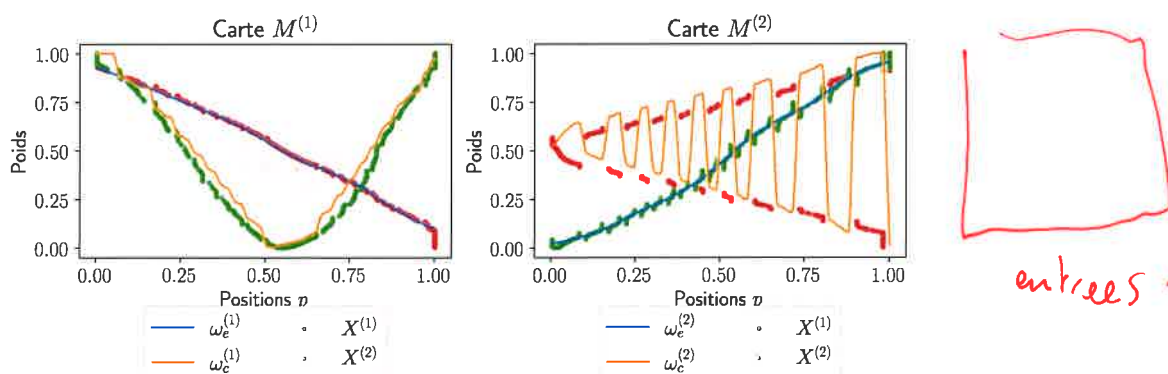


FIGURE 5.8 – Représentation cartographique des poids et entrées pour $X^{(2)} = \cos(X^{(1)})$. Les poids contextuels de la carte $M^{(1)}$ ne forment pas de zones car une seule valeur de $X^{(2)}$ correspond à une entrée $X^{(1)}$. Au contraire, les poids de la carte $M^{(2)}$ s'organisent pour gérer la distinction.

5.2. Identification des mécanismes d'auto-organisation dans une architecture de deux cartes

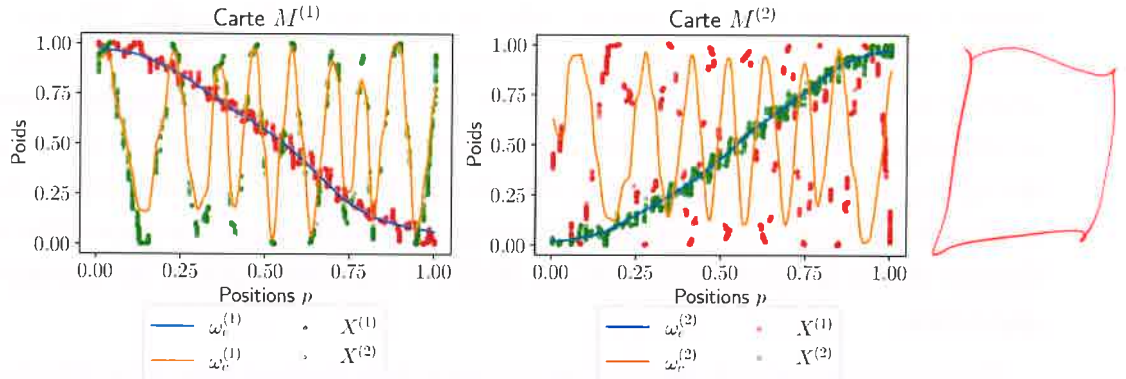


FIGURE 5.9 – Représentation cartographique des poids et entrées pour des entrées sur une courbe de Lissajous. Les poids contextuels se disposent en zones afin de différencier les BMUs selon l'entrée externe et l'entrée contextuelle. Une zone est une carte d'une sous-région des entrées externes

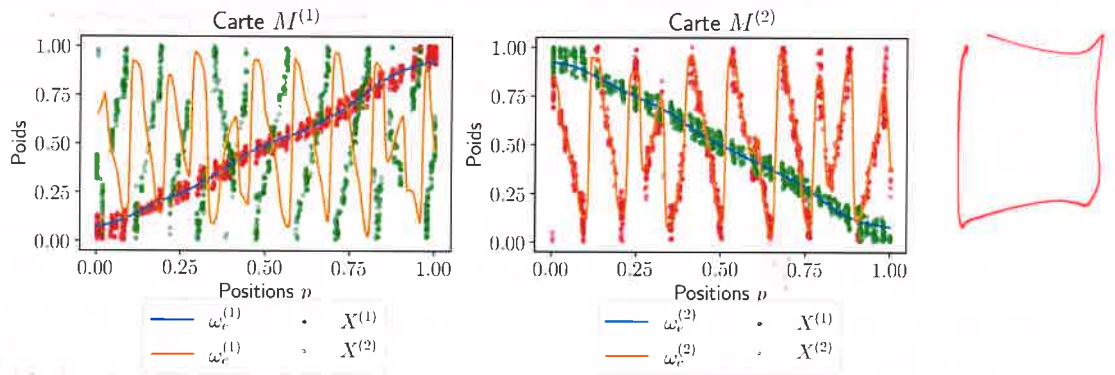


FIGURE 5.10 – Représentation cartographique des poids et entrées dans le patch $[0,1]^2$. Les poids contextuels s'organisent en zones qui cartographient des sous-région de l'espace d'entrée.

Regardons maintenant l'organisation des cartes lorsqu'une valeur de $X^{(1)}$ correspond à plus de deux valeurs de $X^{(2)}$: 4 dans le cas de la courbe de Lissajous (Entrées D) ou tout l'intervalle $[0,1]$ dans le cas du patch $[0,1]^2$ (Entrées E). Ces organisations sont tracées en figure 5.9 et 5.10. Dans ces deux derniers cas, les cartes présentent encore une organisation en zones des poids contextuels. Le nombre de zones est similaire à ce qui est observé sur le cercle, alors que la répartition des entrées est différente. Par contre, la forme des zones varie légèrement par rapport aux observations précédentes. Nous observons clairement en figure 5.10 qu'une zone de BMUs dans $M^{(1)}$ agit comme une sous-carte de toutes les valeurs possible de l'entrée $X^{(2)}$ correspondant à un même intervalle de l'entrée $X^{(1)}$.

Nous pouvons conclure de ces expériences que la présence de zones est un comportement systématique de la carte étant donné qu'elles sont observées même lorsque les entrées sont indépendantes. Cependant, elles émergent de l'organisation seulement lorsque qu'elles sont né-

cessaires, lorsqu'une carte doit pouvoir différencier au moins deux points différents du modèle correspondant à une même valeur d'entrée externe. La forme des zones et la réponse des cartes dépend ensuite de la relation entre les entrées. Ainsi, sur la distribution indépendante, la carte ne présente pas de zone morte. La totalité d'une zone se déploie de manière à couvrir l'ensemble des valeurs de U correspondant à cette zone, ce qui est également observé en figure 5.9 pour les courbes de Lissajous. Une zone agit alors comme une petite carte d'une sous-région de l'espace d'entrée. Sur la distribution en cercle, les BMUs se concentrent autour des extrema des poids contextuels.

Intéressons-nous plus en détail à l'apprentissage de la représentation du modèle d'entrée dans le cas des entrées indépendantes **E**. En figure 5.11, nous traçons la distorsion des poids externes des cartes : $\omega_e(\Pi^{(1)})$ en fonction de $\omega_e(\Pi^{(2)})$ et reliées selon l'ordre des connexions de $M^{(1)}$ puis $M^{(2)}$. Ce tracé nous permet de visualiser la quantification vectorielle dans l'espace d'entrée multimodal et non seulement sur chaque carte. Nous y observons que les cartes quantifient tout l'espace $[0, 1]^2$, mais que seulement une centaine de points servent à la quantification, alors que les deux cartes sont de taille 500. Ces points sont définis par les zones de poids contextuels des cartes. Ces zones définissent donc une échelle de quantification plus large selon X . La carte $M^{(1)}$ classe ensuite les entrées selon les valeurs de $X^{(1)}$ tandis que $M^{(2)}$ les classe selon les valeurs de $X^{(2)}$.

Enfin, nous voulons vérifier si les cartes sont robustes au bruit en prenant des données en forme d'anneau (Entrées **E**), représentées en figure 5.12. Nous nous attendons à un comportement similaire à celui observé sur le cercle, mais avec une marge de quantification un peu plus élevée, ce qui est bien le cas sur le tracé.

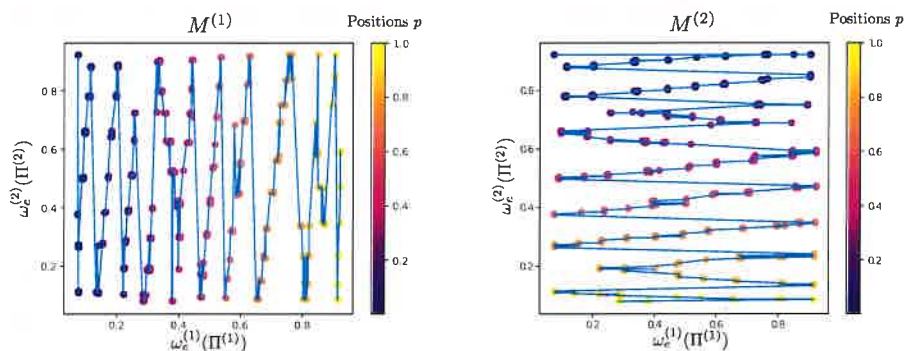


FIGURE 5.11 – Représentation de la distorsion des poids des deux cartes sur le modèle d'entrée **E**. Les cartes s'organisent de façon à représenter tout le patch $[0, 1]^2$, l'une selon les $X^{(1)}$, l'autre selon les $X^{(2)}$. Bien que chaque carte ait 500 nœuds, on observe seulement environ 90 valeurs possibles des paires $(\omega_e(\Pi^{(1)}), \omega_e(\Pi^{(2)}))$.

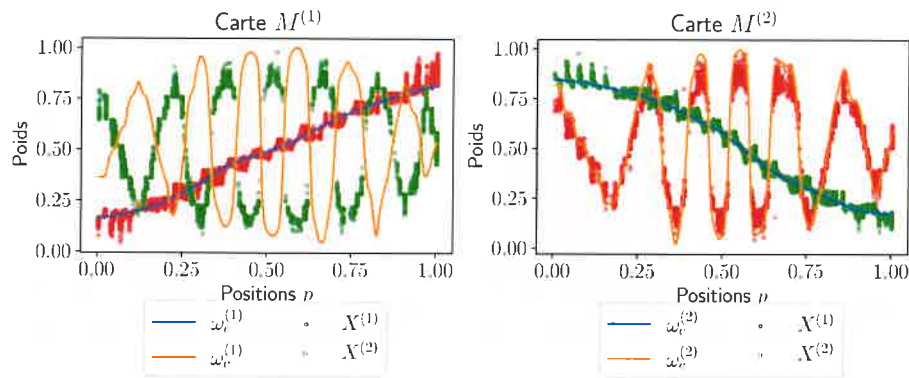


FIGURE 5.12 – Représentation cartographique des poids et entrées pour des entrées sur un anneau. La disposition des poids et la réponse des cartes est similaire à celle de l'architecture apprenant sur le cercle. Une architecture de cartes est robuste au bruit sur les entrées externes de la même façon qu'une carte de Kohonen classique.

ce qui étend à casom les propriétés de robustesse d'une

La dernière observation que nous avons 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'entrée. Cette observation est vérifiée sur les 6 dispositions d'entrées que nous avons étudiées. Nous reviendrons plus en détail sur l'utilisation de U dans l'analyse des réponses des cartes au chapitre 6.

D'après ces tracés, nous concluons que la formation de zones de BMUs grâce aux poids contextuels est une propriété d'organisation systématique d'une architecture de deux cartes Cx-SOM. Une position de BMU encode alors non seulement une entrée externe mais aussi l'entrée contextuelle grâce à un deuxième niveau d'indices dans la carte. Ces deux niveaux d'indexation se forment de façon auto-organisée et seulement lorsqu'il est nécessaire de pouvoir distinguer plusieurs valeurs du modèle U correspondant à la même entrée externe dans une carte.

5.2.4 Étude des mécanismes de formation des zones de poids contextuels

L'organisation en zones observée sur les distributions d'entrées en 2D nous ont montré que les poids des cartes s'organisent en fonction de l'entrée externe, puis des zones se forment systématiquement pour différencier les différentes valeurs possibles du modèle d'entrée U . Les zones se forment dès lors que plusieurs valeurs d'entrées contextuelles correspondent à une même valeur d'entrée externe dans une carte : leur présence et forme dépend du modèle d'entrée. L'organisation et le nombre de zones sont ~~par contre~~ liées aux mécanismes d'apprentissage de la carte. Nous nous intéressons maintenant aux paramètres des cartes influençant la formation de zones. Nous avons observé que la présence de zones est liée en particulier aux rapports entre rayons de voisinage externes et contextuels. Nous comparons dans cette section l'organisation obtenue sur plusieurs architectures ayant des couples de rayons de voisinage contextuels et externes diffé-

en revanche

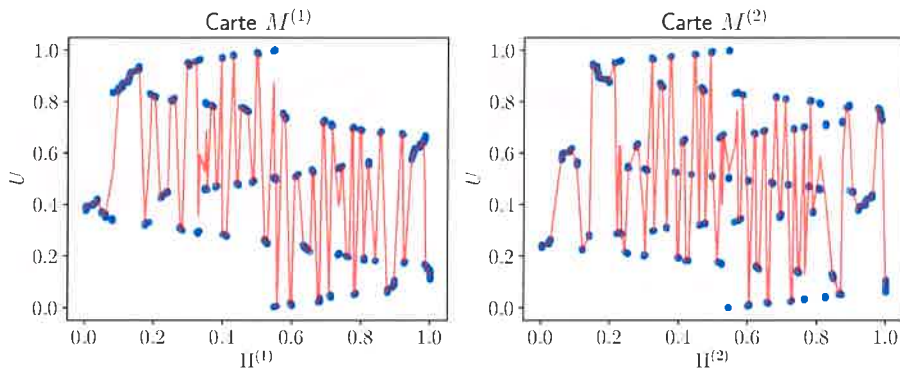


FIGURE 5.13 – Représentation de U selon le BMU $\Pi^{(i)}$ dans chaque carte pour des entrées sur une courbe de Lissajous (D). La courbe en rouge montre l'approximation du nuage de points par une fonction : U est ici une fonction de la position du BMU dans chaque carte, ce qui vérifie l'observation réalisée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

rents, sur une même disposition d'entrée. Nous reprenons ici la disposition A. Nous regarderons également si les zones ont une influence sur la convergence de la relaxation.

Influence du rapport entre rayons de voisinage sur la formation de zones

Nous reprenons les entrées A, en cercle et lançons l'apprentissage de plusieurs architectures dans lesquelles le rapport $\frac{r_e}{r_c}$ est différent. Nous fixons r_e à 0.2 et faisons varier r_c de 0.2 à 0.005. Notons que nous avons observé que l'organisation dépend bien du rapport entre rayons de voisinage et non de la valeur du rayon de voisinage contextuel en elle-même. C'est pourquoi nous fixons r_e à 0.2 sur cet exemple et faisons seulement varier r_c .

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. Pour $r_c = r_e$, ainsi que $r_c \geq r_e$, la carte ne s'organise pas en zones. La formation de celles-ci intervient pour $r_c < r_e$. Sur la figure, nous commençons à voir des zones de poids contextuels apparaître pour $r_c = 3r_e$, sans que les BMUs ne marquent encore de séparation. Les zones sont réellement marquées à partir de $r_c = 4r_e$. Le nombre de zones de BMU augmente ensuite avec le rapport des rayons de voisinage. Plus il y a de zones, plus la quantification de l'entrée externe est précise. La taille du rayon de voisinage contextuel est ensuite limité par la taille de la carte. Si celui-ci est trop faible, la notion de zone de poids contextuel n'a plus lieu d'être. C'est ce qu'on observe pour $\frac{r_e}{r_c} = 40$. r_c est de 2 unités. Chaque zone de BMU ne compte que quelques unités, on ne peut donc plus vraiment parler de sous-carte.

En réalisant la même expérience pour les autres dispositions d'entrées, nous pouvons observer que pour un même couple (r_e, r_c) , le nombre de zones reste similaire quelle que soit l'expérience.

tu le montres ici?

explique.

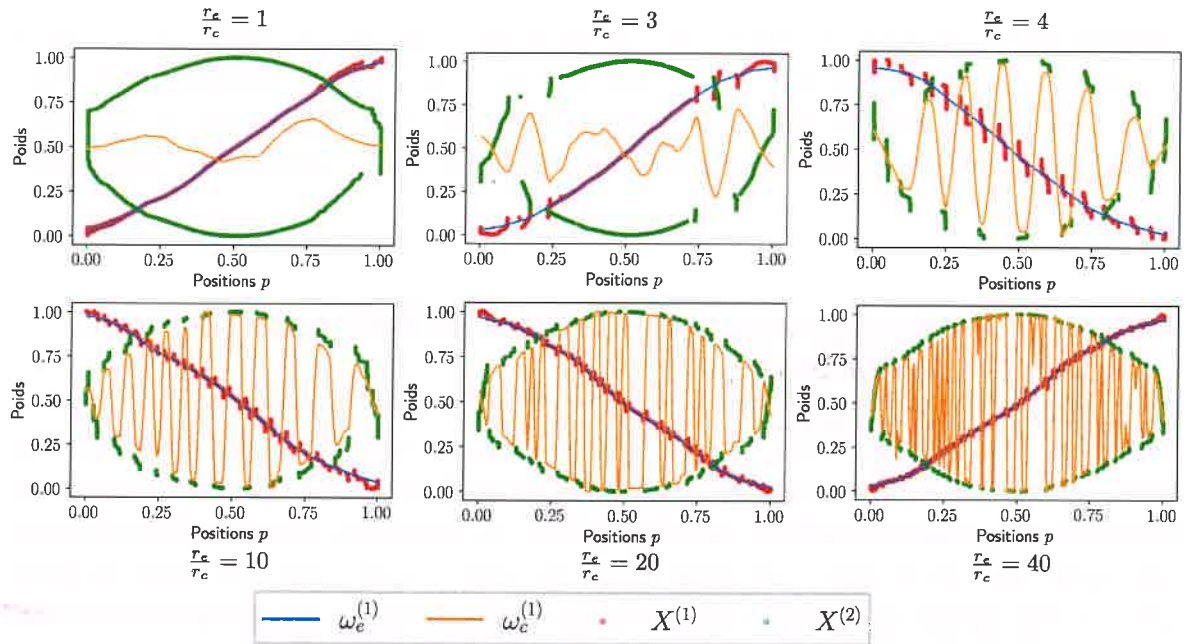


FIGURE 5.14 – Représentation cartographique de la carte $M^{(1)}$ pour différents rayons de voisinage contextuels, le rayon de voisinage r_c étant fixé à 0.2. Nous observons que la présence de zones dépend du rapport entre les rayons de voisinage. La carte définit des zones de BMUs grâce à la forme poids contextuels, de plus en plus nombreuses et contenant 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.

Leur forme diffère en fonction des entrées. C'est donc bien les paramètres des cartes qui induisent donc l'organisation en zones. Le choix du nombre de zones adaptées en vue d'une application reste une question ouverte pour des travaux futurs. L'organisation au sein d'une zone et la forme des poids contextuels dépendent ensuite du modèle d'entrées.

Nous pouvons expliquer la formation des zones par le fait que le rapport entre rayons de voisinage introduit à la fois une relation subordonnée entre les poids lors de la mise à jour et deux échelles temporelles de mise à jour. D'une part, les poids externes se déplient plus vite que les poids contextuels, puis gardent d'autre part une « attraction » plus forte sur les unités voisines ; les poids contextuels doivent donc composer avec cette force. L'organisation des poids contextuels est ainsi subordonnée à l'organisation des poids externes.

Dans nos expériences, nous avons pris r_c identiques pour 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 α , ainsi que les paramètres de la fonction d'activation β , σ , etc. 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 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 intéressés à la compréhension de l'effet des paramètres dans nos travaux, et n'avons pas cherché à optimiser ces valeurs.

Influence de la présence de zones sur la recherche de BMU par relaxation

fais des transitions!
 Nous nous intéressons au passage à l'influence de la formation de zones sur le processus de relaxation. Nous pourrions penser que ces zones favorisent la recherche de consensus lors de la relaxation et la convergence des poids. Nous traçons en figure 5.15 le nombre de pas moyen nécessaire à la recherche du BMU par relaxation et les indicateurs de convergence des poids des cartes, dans le cas d'une carte ayant formé des zones et d'une carte n'en ayant pas formé ($\frac{r_e}{r_c} = 1$). Ces valeurs sont similaires dans les deux cas. Sur des cartes 1D, nous n'observons donc pas d'amélioration de la relaxation grâce à la formation de zones. Cela montre que la convergence de la relaxation n'est pas spécifique aux valeurs de paramètres choisies dans les expériences et est ainsi une méthode générale de connexion entre cartes.

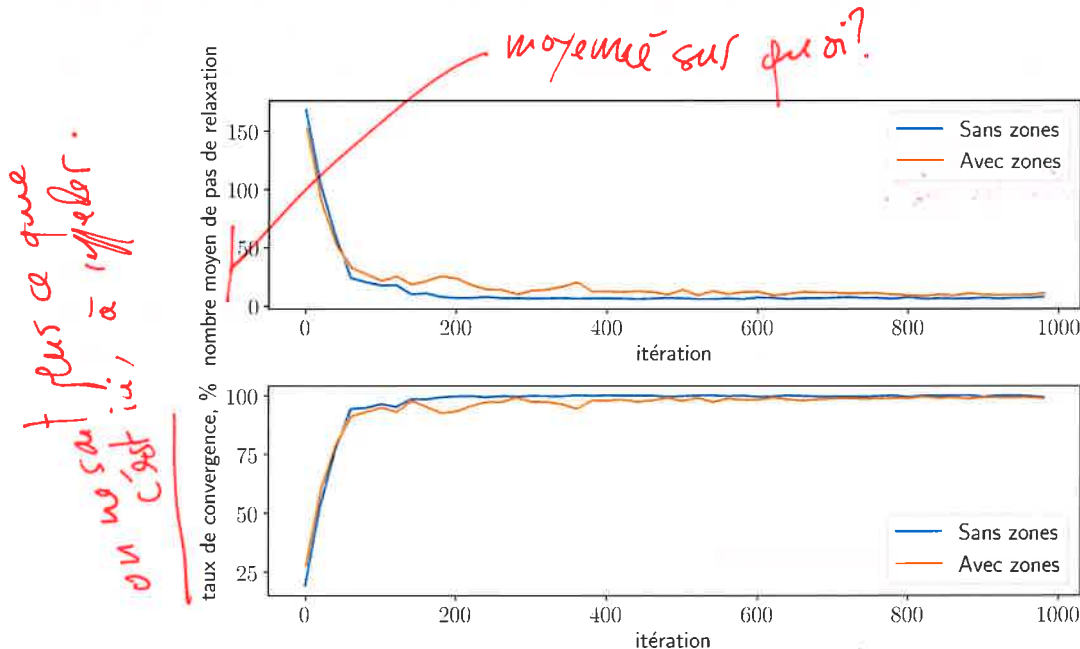


FIGURE 5.15 – Évolution du nombre moyen de pas de relaxation et du taux de convergence pour une organisation de cartes ayant formé des zones de poids contextuels ($\frac{r_e}{r_c} = 10$) et une organisation n'ayant pas formé de zones ($\frac{r_e}{r_c} = 1$). Dans les deux cas, la relaxation mène à un consensus en fin d'apprentissage. Donc, dans des cartes en une dimension, la formation de zones ne favorise pas significativement la convergence de la relaxation. Le mécanisme de relaxation a donc un sens général, quels que soient les paramètres des cartes.

ne semble pas favoriser
tu es trop affirmative alors que c'est un principe.

5.2.5 Discussion

Les motifs en zones formés par les poids contextuels sont un mécanisme qui émerge du processus d'évolution des cartes. Le nombre de zones dépend des paramètres de l'architecture, puis l'organisation au sein des zones dépend ensuite de l'organisation des entrées. Au sein d'une même zone, les poids externes ont des valeurs très proches et les poids contextuels s'organisent de manière à former une sous-carte des valeurs possibles de l'entrée contextuelle pour un même intervalle de valeurs d'entrée externe.

On pourrait donc introduire une notion d'indices primaires et secondaires dans la carte, l'indice primaire étant celui de la zone et l'indice secondaire la position dans la zone. Cette notion d'indices primaires et secondaires est 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, et leur emplacement correspond à la partie du champ de vision traitée (gauche - droite, etc). Ces entrées différenciées forment une indexation *primaire* de V1. Au sein d'une zone de même indice primaire, les neurones s'organisent alors de façon à représenter tout le sous-espace des entrées ayant été présenté à la 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 quand même 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.

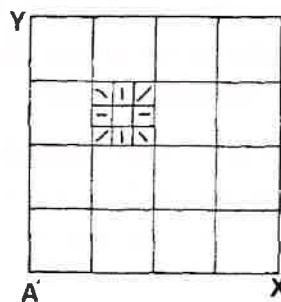


FIGURE 5.16 – Schématisation d'une répartition en indices primaires et secondaire des neurones d'une aire corticale, tirée de (Ballard 1986). Les auteurs observent que la réponse des neurones du cortex est organisée en zones selon leur position, formant des indices primaires. Les carrés de la figure 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.

D'un point de vue computationnel, l'organisation d'une carte s'apparente à une technique de

modulation : la valeur de l'activité externe est modulée par l'activité contextuelle. La forme des poids permet à cette modulation d'encoder en une seule valeur l'entrée externe X et U , le modèle d'entrée. Elle émerge de la dynamique d'apprentissage des cartes. Les zones forment un deuxième niveau de quantification vectorielle au sein d'une carte. Ces deux niveaux de quantifications encodent à la fois l'entrée X et le modèle U en une seule position Π du de BMU.

dit les choses
une fois, afin qu'on
sache où les voir en vérité.

177

5.3 Génération de modalité dans des architectures de trois cartes 1D

Nous avons vu que dans une architecture de deux cartes, apprenant sur un modèle d'entrée liées tel que le cercle en 2D, chaque carte encode à la fois la valeur de son entrée externe et le paramètre du modèle, U , réalisant ainsi un apprentissage associatif. Nous voulons maintenant qu'une architecture de cartes soit directement capable d'utiliser cet encodage du modèle d'entrée 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 son BMU grâce aux entrées contextuelles. Son poids externe $\omega_e(\Pi^{(2)})$ appartient à l'espace de la modalité $X^{(2)}$. La valeur $\omega_e(\Pi^{(2)})$ peut alors être considérée comme une génération de modalité. Nous voulons observer dans cette section si les cartes sont capables d'utiliser la représentation du modèle appris pour générer une valeur correspondant au modèle d'entrée. Pour faciliter l'étude de ce comportement, nous nous plaçons dans un cadre de prédiction d'une modalité $X^{(p)}$ manquante. Il nous suffira de comparer la valeur de l'entrée prédite à celle l'entrée théorique pour vérifier que l'architecture utilise bien le modèle appris pour la génération d'entrée.

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 plutôt 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 avons choisi d'étudier des modèles en trois dimensions plutôt que de rester sur des modèles en deux dimensions, afin de vérifier au passage si les propriétés d'organisation observées sur deux cartes s'appliquent également sur une architecture de trois cartes. Nous construirons ainsi des architectures de trois cartes sur ces modèles d'entrées 3D et étudierons si après apprentissage de toutes les modalités, l'architecture est capable de générer une prédiction précise de la valeur de la modalité manquante à partir des deux modalités qui lui sont présentées. 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 .

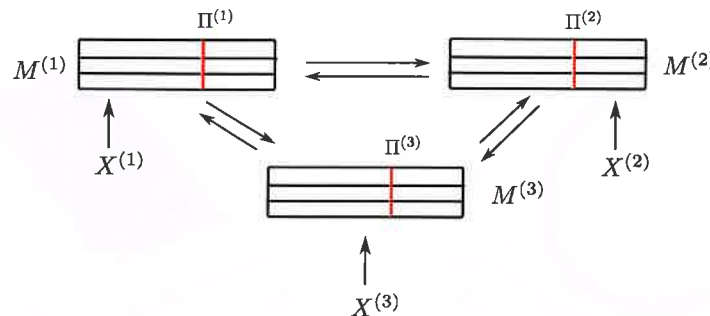


FIGURE 5.17 – Architecture de trois cartes utilisée dans les expériences. Chaque carte prend une entrée externe $X^{(i)}$ et est connectée aux deux autres. Elle possède ainsi deux couches de poids contextuels et une couche de poids externes.

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 BMUs des deux autres cartes. Elle possède donc deux couches contextuelles ω_{c_0} et ω_{c_1} . Nous construisons deux modèles d'entrées jouet à partir du cercle 2D et du patch $[0, 1]^2$ en leur ajoutant une troisième dimension, de telle sorte à ce que la connaissance de deux entrées sur trois détermine la valeur de la troisième entrée. Pour cela, nous pivotons le plan 2D dans lequel se situent ces entrées dans un espace en trois dimensions. Ces modèles sont tracés en figure 5.18. Chaque carte de l'architecture prend en entrée une des coordonnées des points 3D du modèle.

L'architecture de deux cartes ayant appris sur le cercle était capable de réaliser une quantification vectorielle précise de l'entrée externe. Nous nous attendons donc à une bonne prédiction sur trois cartes dans le cas du modèle d'entrée **H**. Dans le cas du plan, la quantification vectorielle sur l'entrée externe était plus large : les cartes distinguent une centaine de points sur le plan. Nous voulons observer comment cela se traduit sur la qualité de la prédiction.

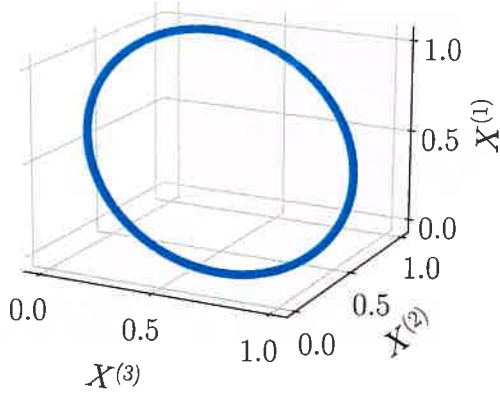
L'algorithme de prédiction d'entrée est schématisé en Figure 5.19. La phase d'apprentissage du modèle est la même que dans les expériences précédentes : les trois cartes reçoivent leurs entrées externes $X^{(1)}, X^{(2)}, X^{(3)}$. La phase de prédiction est une phase de test, durant laquelle les poids 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. Cette carte ne reçoit plus son entrée externe, mais seulement ses entrées contextuelles. Les autres cartes reçoivent toutes leurs entrées. La seule différence avec une phase de test classique est que la carte prédictive $M^{(1)}$ prend comme activité globale sa seule activité contextuelle. Si la carte possède plusieurs couches de poids contextuels, il s'agit de la moyenne des activités contextuelles. La valeur prédite récupérée en sortie de l'architecture est le poids externe du BMU de la carte prédictive $\omega_e^{(1)}(\Pi^{(1)})$.

chaque externe.
Note: je pense que le fait qu'on garde les contextuelles est évident.

vague
intention confuse

références G et H,

G



H

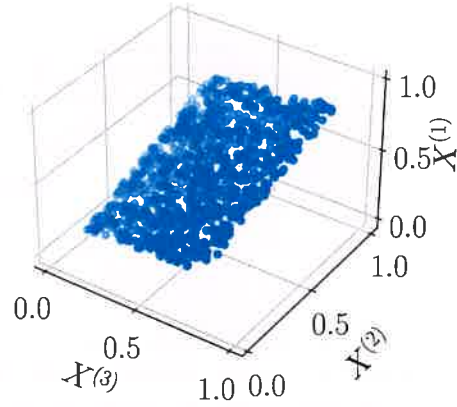


FIGURE 5.18 – Dispositions d'entrées en trois dimensions utilisées dans cette partie. Chaque carte $M^{(1)}$, $M^{(2)}$, $M^{(3)}$ prend en entrée une coordonnée $X^{(1)}$, $X^{(2)}$, $X^{(3)}$.

écrivons { Pour les deux modèles d'entrées, nous tracerons la disposition des poids externes et contextuels après apprentissage et vérifierons si les poids contextuels définissent également des zones de BMUs dans chaque carte. Nous effectuerons dans chaque cas une phase de prédiction de l'entrée $X^{(1)}$ et vérifierons si la valeur prédite $\omega_e(\Pi^{(1)})$ 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 disposition des poids des trois cartes de l'architecture après apprentissage ainsi que des entrées associées pour les entrées dans le cercle 3D (G) et dans le plan 3D (H). Comme dans la version à deux cartes, les poids contextuels s'organisent en plusieurs zones permettant de séparer les BMUs en fonction de la valeur du modèle d'entrée et non seulement de l'entrée externe. Nous concluons de ces deux tracés qu'une architecture de trois cartes a un comportement semblable à une architecture de deux cartes.

Nous traçons ensuite l'erreur obtenue lors de la phase de prédiction de $X^{(1)}$: en figure 5.22 pour le modèle du cercle G, et 5.23 sur le plan H. Dans les deux cas, l'entrée $X^{(1)}$ n'a pas été présentée à la carte et sa valeur est prédite par $\omega_e(\Pi^{(1)})$. Nous ajoutons aux tracés l'erreur de quantification vectorielle dans les autres cartes, qui ont reçu leur entrée, afin de comparer la qualité de la prédiction à la qualité de la quantification vectorielle. Nous observons que la prédiction est très bien réalisée dans le cas du cercle. Les valeurs prédites par la carte $M^{(1)}$ sont aussi précises que les valeurs quantifiées par les cartes $M^{(2)}$ et $M^{(3)}$. L'architecture de cartes a donc encodé le modèle d'entrée et est capable de prédire l'entrée manquante.

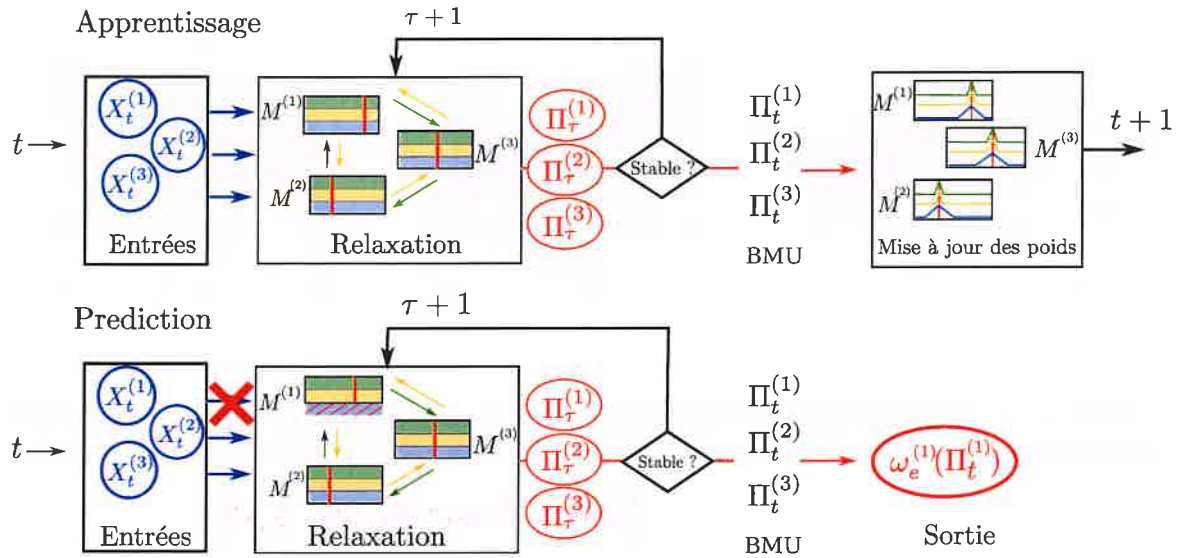


FIGURE 5.19 – Schéma descriptif des opérations effectuées lors de l'apprentissage et de la phase de prédiction. Après une phase d'apprentissage classique, la phase de prédiction est une phase de test durant laquelle l'entrée $X^{(1)}$ n'est pas présentée à la carte $M^{(1)}$. Celle-ci ne prend alors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

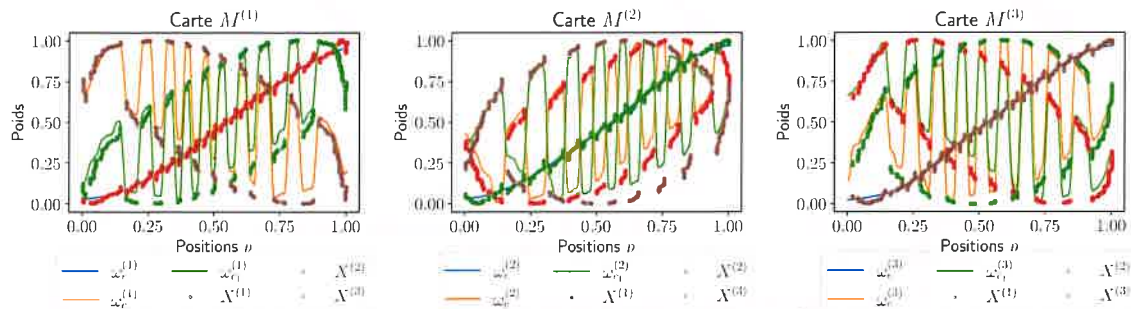


FIGURE 5.20 – Représentation cartographique des poids et entrées dans l'architecture de trois cartes apprenant sur un cercle en trois dimensions. Nous observons la formation de zones similaires au cas en deux dimensions.

Dans le cas du plan, la prédiction est également bien réalisée. L'encodage de U était alors plus large que dans le cas du cercle. Cet encodage plus large se manifeste sur la marge d'erreur de prédiction.

La disposition en étages qui apparaît sur les tracés d'erreur de prédiction vient directement de la disposition en zones des cartes de l'architecture. On peut donc supposer que la relaxation permet de choisir une zone ; et la valeur prédite est prise dans cette zone selon seulement les valeurs des poids contextuels, d'où les « étages » dans les tracés de prédiction. Afin de valider

encodage "large" ? ! ! !

(entrée 6 sur la fig 5.18)

pas clair

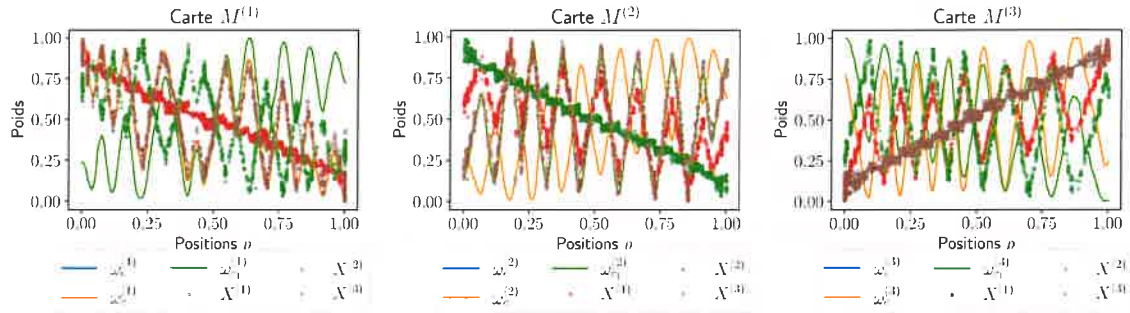


FIGURE 5.21 – Représentation cartographique des poids et entrées dans l'architecture de trois cartes apprenant sur un plan pivoté en 3D. Les zones sont similaires au cas en deux dimensions.

(entrée 1 sur la fig 5.18)

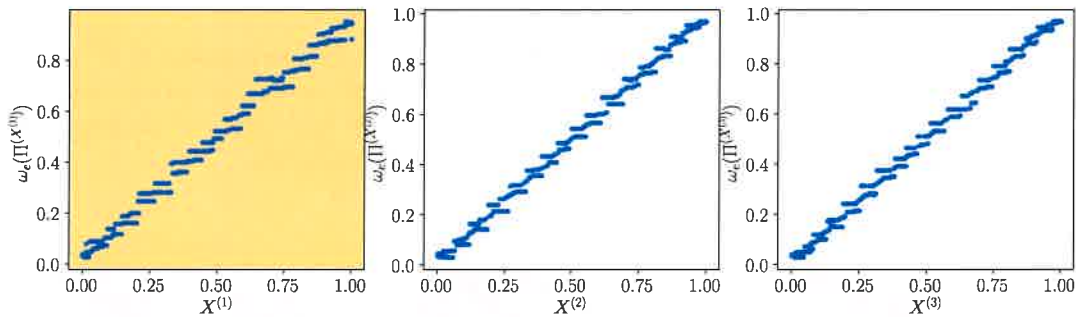


FIGURE 5.22 – Erreur de prédiction de $X^{(1)}$ par $\omega_e(\Pi^{(1)})$ lorsque les entrées sont sur un cercle en trois dimensions. $X^{(1)}$ n'a pas été présenté à $M^{(1)}$. Les nuages de points correspondant à $M^{(2)}$ et $M^{(3)}$ correspondent à l'erreur de quantification dans les cartes 2 et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

(entrée 6 sur fig 5.18)

l'influence des zones dans la capacité de prédiction d'entrée, nous nous intéressons à la prédiction dans un 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 avons 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 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 sur les trois entrées du modèle, ne pouvons pas parler d'un apprentissage du modèle dans ce cas, car les cartes ne sont pas capables d'utiliser les relations entre entrées dans la tâche de prédiction. Les zones permettent donc bien aux cartes d'encoder les relations entre entrées et de les réutiliser en sortie. Elles sont caractéristiques de l'apprentissage du modèle d'entrée par l'architecture.

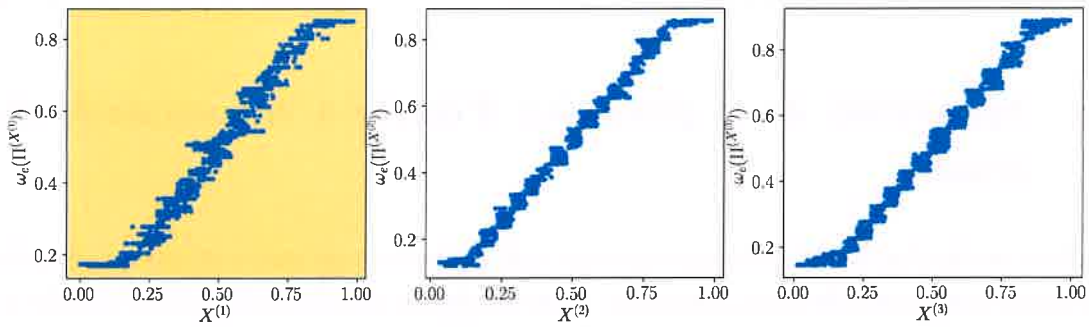


FIGURE 5.23 – Erreur de prédiction de $X^{(1)}$ dans le cas du plan pivoté en 3D. La prédiction est plus large car U est quantifié plus grossièrement que dans le cas du cercle, voir figure 5.11, mais elle est bien réalisée sur toutes les entrées.

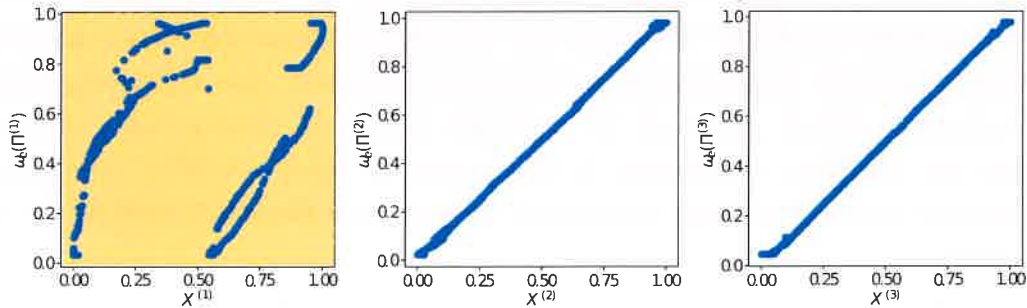


FIGURE 5.24 – Prédiction de l'entrée $X^{(1)}$ lorsque $r_c = r_e$. La prédiction n'est pas effectuée. Ainsi, sans formation de zones, la capacité de prise de décision n'est plus réalisable par une carte de l'architecture.

5.3.3 Conclusion

Nous avons donc montré qu'une architecture de cartes est capable de générer une modalité manquante dans le modèle grâce aux connexions entre cartes. La valeur utilisée pour la prédiction est dans ce cas le poids externe du BMU de la carte qui n'a pas reçu son entrée externe. Ce comportement montre que grâce aux poids appris et à la recherche de BMU, une architecture de cartes est non seulement capable d'encoder un modèle d'entrée, 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. 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.

Nous avons observé que les zones de poids contextuels permettent cette capacité de prédiction. Ce comportement 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 qui est effectuée au sein de chaque carte.

5.4 Application de la prédiction d'entrée à la commande d'un drone

Nous avons observé qu'après apprentissage, une architecture de cartes est capable de générer une entrée manquante de façon autonome. Cette entrée est en adéquation avec le modèle d'entrées apprises par l'architecture : il s'agit d'une prédiction. Nous sortons du cadre des entrées simulées pour nous placer dans un cas de contrôle réel. Cette expérience est un exemple simple d'application de l'architecture en robotique, et nous permet de tester les cartes sur des entrées bruitées, dont nous ne connaissons pas le modèle de relation.

Nous disposons d'un ~~drone~~ ^{quadricoptère} contrôlé à distance par ordinateur. Ce drone possède une caméra frontale ainsi qu'un ensemble de capteurs internes. Chacun des capteurs du drone est une modalité d'un espace multimodal. Leurs valeurs dépendent les unes des autres : elles dépendent par exemple de la position du drone à un instant. La commande envoyée au drone à chaque instant peut également être considérée une modalité de l'environnement. Lorsque le drone cherche à suivre une trajectoire donnée, la commande envoyée à chaque instant devient dépendante des valeurs des capteurs.

À l'aide d'une architecture de cartes, nous voulons ainsi apprendre les relations existant entre les capteurs et la commande, afin de pouvoir générer la commande à envoyer au drone à partir des valeurs des capteurs lors d'une phase de prédiction. Nous effectuerons cette phase de prédiction en temps réel, pendant laquelle la commande prédite sera envoyée au drone à chaque instant. Le but de cette expérience est de tester l'architecture de cartes sur des cas d'entrées réelles, donc bruitées, en se plaçant dans un cadre de contrôle en temps réel d'un drone. Nous évaluerons la robustesse de l'algorithme à des données bruitées et la capacité de CxSOM à réagir en temps réel malgré les étapes de relaxation.

5.4.1 Méthode expérimentale

Le drone utilisé pour l'expérience est un quadricoptère. Il possède une caméra frontale non orientable. Nous le contrôlons à distance par un ordinateur. Lors de cette expérience, le drone naviguera dans un couloir en ligne droite. L'objectif du contrôle est que le drone se déplace au centre du couloir, en ligne droite.

La figure 5.26 présente les capteurs et commandes disponibles. Le drone est commandé par la définition d'un angle de rotation autour de chaque axe. L'angle autour de l'axe z , noté ω , contrôle la vitesse de rotation angulaire du drone autour de cet axe (lacet). L'angle autour de l'axe y (tangage) contrôle la vitesse de rotation haut/bas. Enfin, l'angle autour de x , que

perturbations due aux turbulences qu'il génère.

l'expérience est mal décrite

5.4. Application de la prédiction d'entrée à la commande d'un drone

nous notons ρ (roulis), influence l'accélération du drone selon y . Dans cette expérience, nous contrôlerons uniquement l'angle de roulis ρ à l'aide de l'architecture de cartes. L'angle de lacet ω sera contrôlé à l'aide d'un PID de façon à ce que la caméra du drone reste dirigée vers le milieu du couloir. L'angle de tangage y est maintenu à 0 à l'aide d'un PID, de façon à ce que le drone se déplace à une altitude constante. La commande ρ constitue une première modalité des entrées que nous allons considérer pour l'apprentissage de l'architecture de cartes.

Nous considérons trois éléments extraits des capteurs du drone et qui sont relatifs à sa position par rapport au couloir. Une analyse de l'image issue de la caméra frontale nous permet de récupérer l'abscisse du point de fuite du couloir x et la différence entre les angles des lignes du couloir, notée φ . Enfin, les capteurs internes du drone nous permettent de récupérer les vitesses linéaires v_x, v_y, v_z à chaque instant selon chaque axe de déplacement. Comme nous contrôlerons l'angle ρ , donc une l'accélération linéaire selon y , nous nous intéresserons particulièrement à la vitesse linéaire en y en tant que modalité. Finalement, nous utiliserons quatre modalités lors du déplacement du drone : l'abscisse du point de fuite x , l'angle du couloir φ , la commande en angle de roulis ρ et la vitesse linéaire courante en y v_y .

Nous construisons une architecture CxSOM sur ces quatre modalités, composée de quatre cartes 1D connectées chacune aux trois autres, tracée en figure 5.25. Chaque carte prend une des modalités comme entrée externe. Le but de l'architecture sera de capturer les relations entre les capteurs et la commande puis générer la commande lors de la phase de prédiction. La phase d'apprentissage est réalisée sur des trajectoires du drone dans le couloir lorsque la commande est réalisée par un humain. Lors de cette phase, nous avons utilisé un système de contrôle PID pour assister la commande humaine. Cette phase d'apprentissage est réalisée hors ligne : nous normalisons d'abord les entrées puis les présentons aléatoirement lors de la phase d'apprentissage. Chaque carte est de taille 500. Les paramètres sont $r_e = 0.2$ et $r_c = 0.02$. L'apprentissage a été effectué sur 4890 points, obtenus sur des trajectoires effectuées dans le même couloir que les trajectoires de test. Nous avons présenté ces points une seule fois, ce qui a suffi à un bon dépliement des cartes.

Après apprentissage, nous effectuons une phase de prédiction en ligne et en temps réel sur le contrôle du drone. Lors de cette étape, la commande ρ n'est pas présentée à la carte $M^{(\rho)}$. $\omega_e^{(\rho)}(\Pi^{(\rho)})$ et la valeur de la prédiction est utilisé comme commande ρ envoyée au drone en temps réel. Les angles de tangage et de lacet sont quant à eux contrôlés par un PID afin de maintenir l'axe du drone pointant vers le centre du couloir et une altitude constante.

Afin de mieux visualiser les conditions de l'expérience, nous avons tracé les dépendances entre chaque modalité en figure 5.27. Nous nous intéressons ici aux dépendances entre la commande ρ qui est celle que nous prédirons et des autres modalités. On remarque que ρ dépend linéairement de l'angle du couloir φ , mais que cette dépendance est très bruitée. Elle dépend également de la vitesse interne du drone v_y . Par contre, les entrées correspondant à l'abscisse du point de fuite

l'altitude n'est pas contrôlée via un angle sur un drone ??

sur l'image
humain assiste ??
confus !
confus

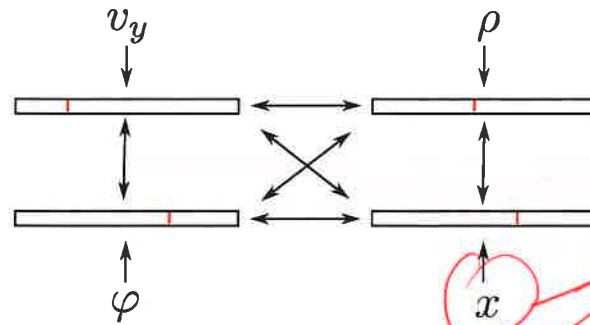


FIGURE 5.25 – Architecture de quatre cartes utilisée dans l'expérience. Chaque carte prend une modalité en entrée externe lors de l'apprentissage : v_y , ρ , φ , x . Lors de la phase de prédiction, l'entrée ρ n'est plus présentée à l'architecture et la commande envoyée au drone est $\omega_e^{(\rho)}(\Pi^{(\rho)})$

ne sert à rien puisqu'il est asservi même en test. ou le voit en 5.27

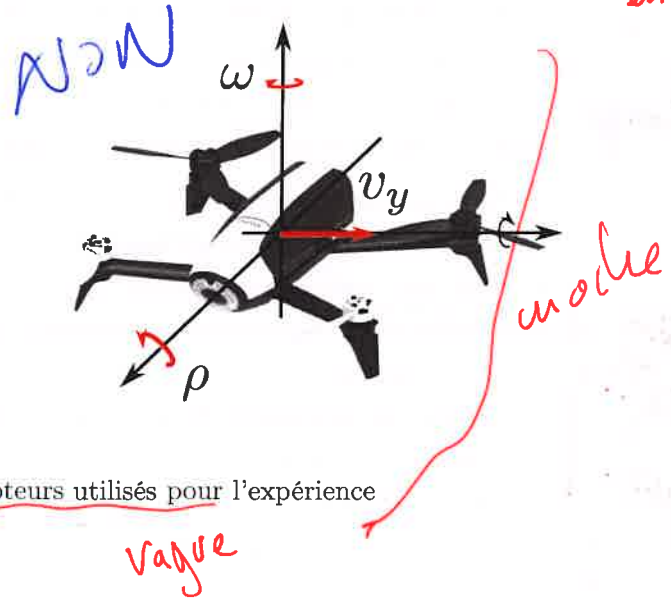
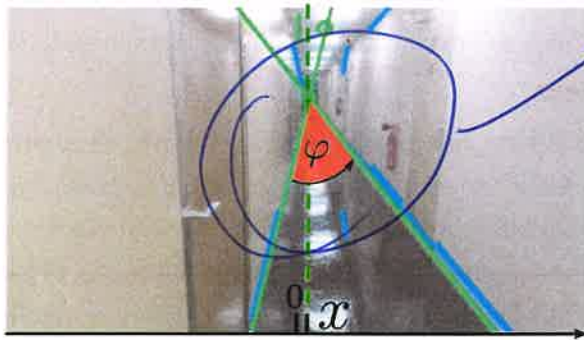


FIGURE 5.26 – Disposition des capteurs utilisés pour l'expérience

varient peu ; la commande ρ dépend peu de x .

5.4.2 Résultats

La carte associée à ρ possède une couche de poids externe et trois couches de poids contextuels. Ces poids sont représentés en figure 5.28. Nous observons d'abord que l'organisation des poids contextuels rappelle celle observée dans les dispositions d'entrées jouets : les poids contextuels définissent des zones. La figure 5.28 présente en violet un exemple de calcul d'activité de la carte ρ pendant la phase de prédiction. A un instant t , l'architecture reçoit les trois modalités v_y , x , φ . L'activité de $M^{(\rho)}$ représentée est son activité contextuelle globale après la relaxation, calculée seulement grâce à ses entrées contextuelles. Le BMU correspond au maximum de 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 . Tous ces calculs doivent être réalisés assez rapidement pour que le drone réagisse en temps réel.

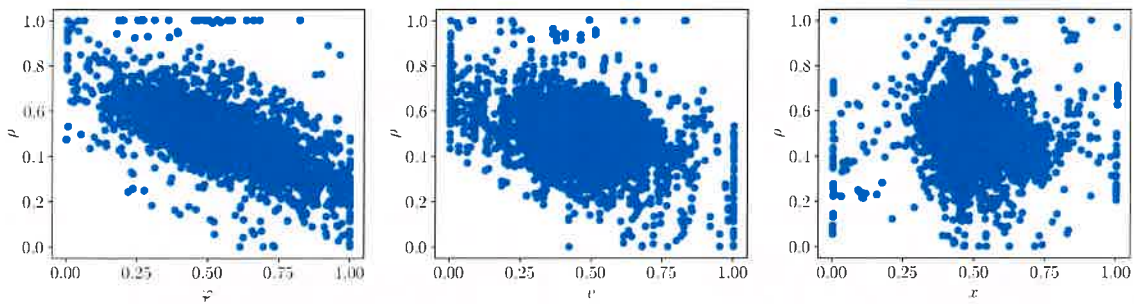


FIGURE 5.27 – Disposition et dépendances des entrées d'apprentissage. Nous chercherons à prédire ρ : cette valeur dépend bien des autres modalités v , φ et x . La dépendance est très simple (linéaire) mais très bruitée, et ρ ne dépend pas de x .

Lors des expériences que nous avons effectuées, le drone apparaît voler correctement dans le couloir sans toucher les murs. Nous observons cependant que cette trajectoire est assez imprécise : la commande prédite permet au drone de ne pas toucher les murs, mais ne lui permet pas de se centrer finement au centre du couloir. La prédiction est donc correctement réalisée, toutefois assez imprécise. Le fait que les entrées soient très bruitées peut expliquer ce manque de précision.

5.4.3 Conclusion

Cette application sur une architecture de quatre cartes nous permet d'abord d'étendre les observations réalisées sur deux et trois cartes et montre que les comportements des architectures de deux et trois cartes, à savoir la formation de zones, sont également observés sur les 4 cartes. Ensuite, la capacité de prédiction observée sur le drone constitue une possibilité conceptuelle d'application des architectures de cartes sur des données réelles. Malgré les données bruitées, l'architecture de cartes détecte une relation entre entrées qui lui permet de prédire de façon large la commande à envoyer. Enfin, la réactivité de l'envoi de la commande au drone montre que la relaxation permet quand même une réponse en temps réel.

Ces observations sont sur un cas simple d'application en une dimension, mais laissent la porte ouverte à une application possible des architectures CxSOM en pratique, par exemple en robotique. Nous pouvons imaginer, à plus grande échelle, que l'architecture de carte permette à un robot une prise de décision par rapport à la disposition de ses capteurs. La capacité de prédiction laisse également envisager des applications de robustesse à un capteur cassé ou l'utilisation de données dans lesquelles il manque parfois une modalité. Une réelle application de l'architecture peut donc être une piste d'étude future. Dans ce sens, il serait intéressant d'étudier l'application 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

fait qu'on rediscute de l'expé
du drone...

!!!
???

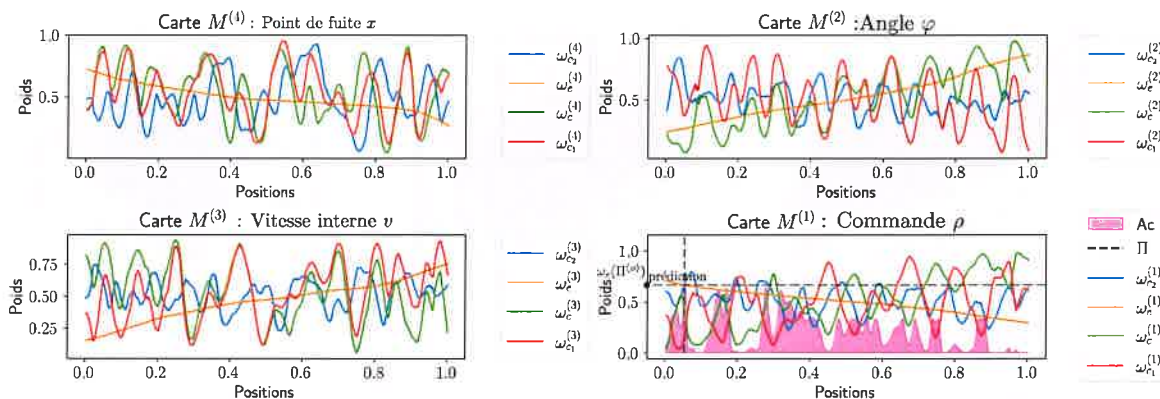


FIGURE 5.28 – Disposition des poids des 4 cartes après apprentissage. La commande prédite ρ envoyée au drone est le poids externe du BMU de la carte ρ , calculé uniquement à partir des activités contextuelles. Cette activité est représentée en violet sur la dernière carte.

réel. L'architecture de carte pourra être combinée à d'autres modèles de contrôle automatique pour stabiliser les commandes envoyée au robot. Enfin, les entrées d'une application en robotique sont en pratique en plus grande dimension : ce sont par exemple directement des images. Il sera nécessaire d'étudier l'organisation des cartes sur des données en grande dimension pour envisager une application, en utilisant dans ce cadre des cartes en deux dimensions.

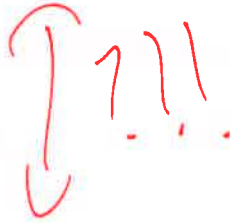
mon

tu pourrais critiquer l'approche supervisée vs RL non? Je ne le sens pas le drone...

5.5 Influence des connexions d'une architecture sur l'apprentissage du modèle d'entrée

Dans toutes les expériences précédentes, nous avons utilisé des architectures dans lesquelles les cartes sont toutes connectées. Nous avons une seule possibilité de graphe de connexions dans une architecture de deux cartes ; le nombre d'architectures de cartes possibles augmente ensuite exponentiellement avec le nombre de cartes considérées pour une architecture. L'influence des connexions sur l'apprentissage d'un modèle d'entrée est donc une orientation judicieuse pour les travaux d'études de CxSOM futurs. 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 au cas d'une architecture de trois cartes dans laquelle chaque carte est liée à une seule autre et non aux deux autres comme c'était le cas plus haut.

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



pas forcément
je pense

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 pouvons donc effectuer la même expérience sur des entrées indépendantes, mais dans une architecture de 9 cartes toutes connectées apprenant sur tout l'hypercube $[0, 1]^9$.

Nous nous demandons si les poids contextuels continuent de s'organiser en zones ou si le nombre de connexions modifie ce comportement. Nous tirons donc des entrées dans l'hypercube $[0, 1]^9$. Nous ajoutons à ces 9 entrées indépendantes une entrée $X^{(10)}$ identique à l'entrée $X^{(9)}$ et construisons finalement une architecture de 10 cartes, toutes connectées. Nous vérifierons sur les cartes $M^{(9)}$ et $M^{(10)}$ si la relation entre ces deux entrées identiques est encodée par l'architecture ou si les connexions « inutiles » de $M^{(9)}$, c'est-à-dire correspondant à des modalités qui n'ont pas de dépendances avec $X^{(9)}$, viennent empêcher l'apprentissage de cette relation.

La figure 5.29 présente la forme des poids de l'architecture de 10 cartes. Nous remarquons que les poids contextuels correspondant à des entrées indépendantes se rapprochent tous d'une valeur moyenne constante de 0.5 dans chaque carte, par exemple dans $M^{(1)}$ et $M^{(2)}$ représentées en bas de la figure. À la fin de l'apprentissage, les activités contextuelles correspondant à ces couches de poids seront constantes sur toute la carte et n'interviendront alors plus dans le choix du BMU. L'architecture équivaut donc à 9 cartes indépendantes, ce qui est cohérent par rapport à la disposition des entrées. Par contre, les couches de poids contextuels correspondant aux deux entrées identiques $X^{(9)}$ et $X^{(10)}$ sont représentées en rose dans $M^{(9)}$ et $M^{(10)}$. Elles se déplient totalement, comme nous l'avons observé en figure 5.7. Cette couche de poids sera donc la seule à réellement intervenir dans le choix du BMU. Nous pouvons également observer la prédiction de l'entrée $X^{(9)}$, représentée en bas de la figure. Elle est correctement réalisée lorsque la carte $M^{(9)}$ ne reçoit pas son entrée externe. Les connexions « inutiles » ne perturbent donc pas l'apprentissage des relations existantes. 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 mériterait d'être étudiée plus en détail pour plus de types de dépendances entre entrées. En particulier, on peut se demander si ce comportement de l'architecture sur des entrées en grande dimension permet d'extraire automatiquement des relations entre entrées. Cette détection de relations serait un comportement émergent d'une architecture de cartes.

un peu non ?

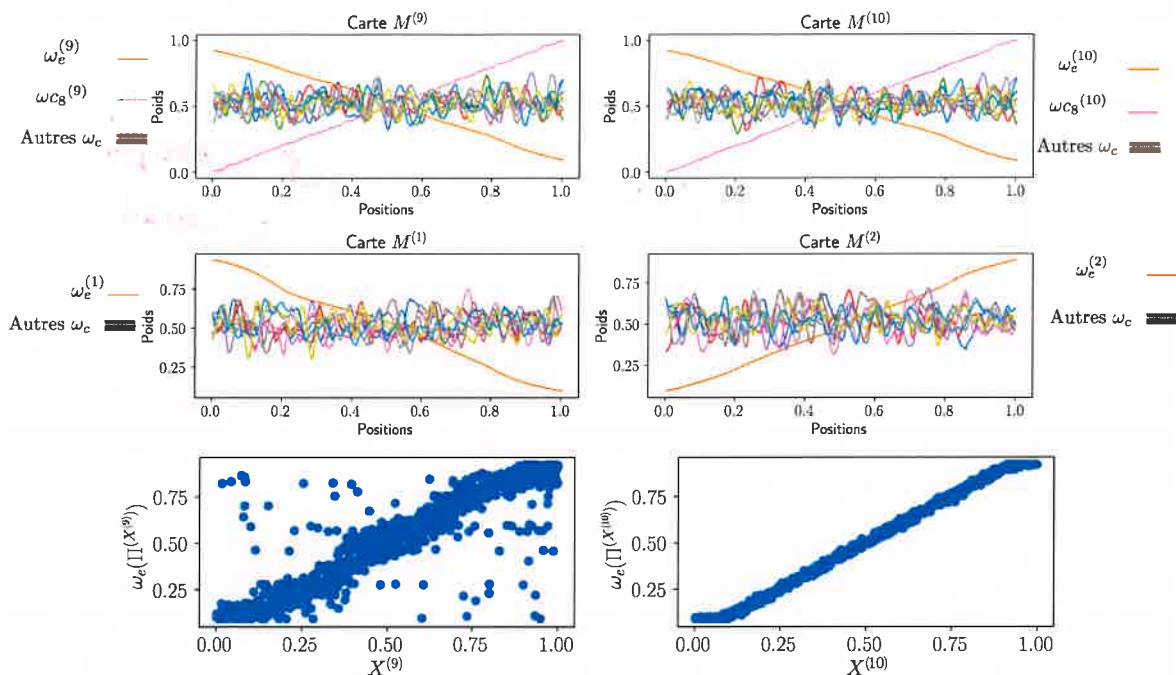


FIGURE 5.29 – En haut, tracé des poids des cartes pour une architecture de 10 cartes toutes connectées. Nous avons ici seulement tracé 4 cartes sur les 10. Les entrées sont toutes indépendantes, sauf les entrées $X^{(9)}$ et $X^{(10)}$ qui sont identiques. Nous remarquons que les poids contextuels se moyennent autour de 0.5, sauf ceux correspondant aux entrées dépendantes. En bas, nous traçons l'erreur de prédiction de la carte 9 lorsqu'elle ne reçoit pas d'entrée. La prédiction est bien réalisée : les connexions contextuelles inutiles ne perturbent pas l'apprentissage.

5.5.2 Influence des connexions distantes sur l'organisation d'une carte

A contrario d'un grand nombre de connexions, nous cherchons 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 entrée le cercle tourné en trois dimensions (**G**). Nous comparons le comportement de l'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 autre carte. Ce modèle d'architecture est illustré en figure 5.30 : $M^{(1)}$ est connectée à $M^{(2)}$, $M^{(2)}$ connectée à $M^{(3)}$ et $M^{(3)}$ connectée à $M^{(1)}$. Lors d'une phase de prédiction, nous observerons si l'architecture est capable de prédire $X^{(1)}$. Dans ce cas, la carte prédictive $M^{(1)}$ ne reçoit en entrée que la position du BMU de $M^{(3)}$: la position du BMU de $M^{(2)}$ intervient de façon distante dans le calcul de l'activité de $M^{(1)}$. Nous comparerons les résultats obtenus à une prédiction effectuée uniquement à partir des cartes $M^{(3)}$ et $M^{(1)}$, sur les mêmes entrées et les mêmes dispositions de poids que dans l'architecture : nous ne relançons pas d'apprentissage pour cette phase de prédiction. $M^{(2)}$ contribue à la prédiction dans le premier

cas via $M^{(3)}$, mais n'intervient pas dans le second cas. Cela nous permettra de dissocier la contribution de $M^{(3)}$ de celle de $M^{(2)}$ dans les représentations.

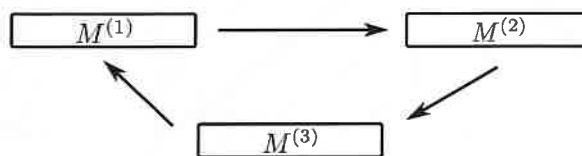


FIGURE 5.30 – Schéma de connexions d'une architecture en « boucle ».

La disposition des poids est tracée en figure 5.31. Les poids des cartes montrent une organisation similaire au cas où les connexions sont réciproques : les poids contextuels forment des zones et chaque carte différencie ses BMUs en fonction du modèle d'entrée U et non seulement de son entrée externe. D'après la méthode d'étude présentée plus haut, nous pourrions dire que l'architecture a encodé le modèle d'entrées.

En figure 5.32, nous représentons la valeur de la prédiction dans l'architecture de trois cartes en fonction de l'entrée théorique $X^{(1)}$. Nous comparons cette prédiction 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 que dans l'architecture avec rétroaction : 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'entrée que dans le cas où les trois cartes sont connectées réciproquement. Cependant, la prédiction est mieux réalisée dans l'architecture en boucle à partir de $X^{(3)}$ et $X^{(2)}$ à distance, que lorsqu'on ne prend en compte que la valeur de $X^{(3)}$ pour la prédiction. Dans ce dernier cas, seules 40% des entrées sont correctement prédites par le couple de cartes et l'erreur est globalement plus importante. Cela montre que $M^{(2)}$ intervient malgré tout dans la prédiction de $X^{(1)}$, dans une faible mesure.

Cette observation montre qu'une carte a une influence à distance dans une architecture. Par contre, cette influence est minime par rapport à l'influence d'une connexion directe. L'influence de ces connexions distantes est ainsi une piste d'étude pour un passage sur une grande architecture, dans laquelle nous pourrions agir sur les connexions entre cartes.

5.6 Conclusion

Ce chapitre de résultats présente l'analyse de l'organisation d'architectures de 2 et 3 cartes 1D sur des données d'entrées en deux et trois dimensions. Nous avons d'abord observé que l'apprentissage du modèle d'entrée dans une architecture CxSOM est marqué par une organisation à deux échelles : les poids externes se déploient sur l'espace d'entrée comme les poids d'une carte classique. Les poids contextuels se déploient sur des sous-régions de la carte définies par la va-

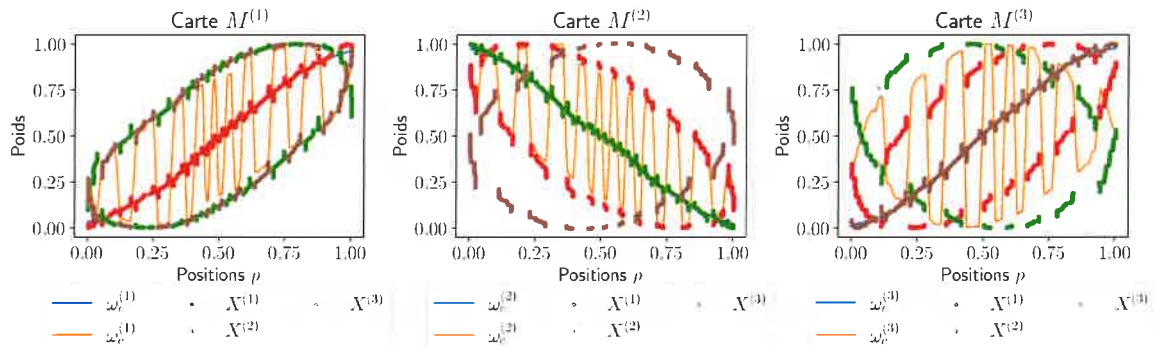


FIGURE 5.31 – Poids à l'issue de l'apprentissage dans une architecture de 3 cartes connectées en boucle. Les poids contextuels forment des zones de BMUs similaires à celles observées dans l'architecture avec des connexions réciproques.

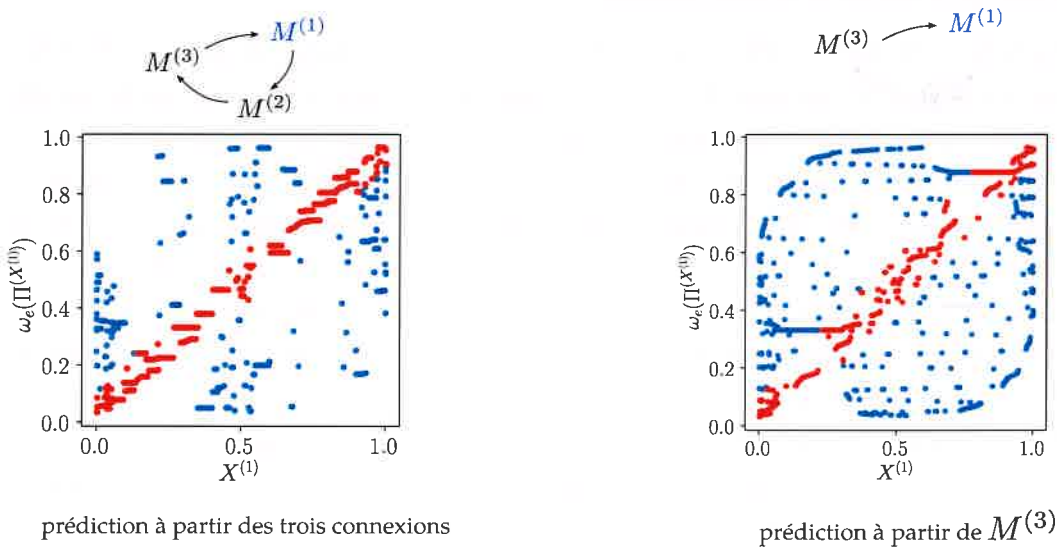


FIGURE 5.32 – À gauche, erreur de prédiction de l'entrée $X^{(1)}$ par l'architecture en boucle. Les points marqués en rouge représente les valeurs correctement prédites à moins de 0.1 près. À droite, nous représentons l'erreur de prédiction réalisée à partir des mêmes poids de cartes en ne prenant en 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 $M^{(2)}$.

leur de l'entrée externe, formant une disposition en « zones ». Ces zones de poids contextuels permettent d'encoder l'information sur tout le modèle d'entrée dans chacune des cartes : chaque carte choisit son BMU en fonction à la fois de son entrée externe et de ses entrées contextuelles. Une position de BMU encode donc en une seule valeur Π l'ensemble du modèle d'entrée et non seulement l'entrée externe de la carte. Ces zones sont caractéristiques de l'apprentissage du modèle d'entrée par une architecture de cartes et sont observées sur de plusieurs jeux de données d'entrées jouet ou réelles, sur des architectures de deux et trois cartes, et quel que soit le nombre

de connexions.

Nous avons ensuite constaté que l'organisation en zones permet à l'architecture d'utiliser le modèle appris pour générer des valeurs d'entrées manquantes. Après apprentissage, une carte qui ne reçoit plus son entrée externe peut définir un BMU grâce à ses activités contextuelles et le poids externe du BMU est ~~perçu comme une~~ valeur générée de la modalité qui n'a pas été présentée. Dans ces tâches de prédiction, l'architecture CxSOM utilise de façon autonome le fait qu'elle ait encodé le modèle d'entrée. Nous avons observé que cette capacité de prédiction est directement liée à la présence des zones de poids contextuels, ce qui valide le fait que l'encodage du modèle vient bien de cette organisation en zones. Cette capacité de prédiction est une application possible des architectures de cartes, mais permet également d'évaluer l'apprentissage d'un modèle d'entrée par l'architecture lorsque ce modèle n'est pas connu. !!!

Maintenant que nous avons observé les comportements d'apprentissage dans des 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. Pour mener à bien cette construction, une étude plus générale de l'influence des connexions permettra d'ajouter cette dimension à la conception de systèmes CxSOM. Nous avons présenté dans ce chapitre deux perspectives d'influence des connexions entre les cartes. D'une part, nous avons vu que des connexions distantes permettent toujours à l'architecture de cartes d'apprendre un modèle d'entrée. Cependant, lors de la prédiction, les cartes distantes de la carte prédictive ont moins d'influence que les cartes qui lui sont directement connectées. Cette observation montre que la connectivité d'une architecture est bien un paramètre ayant une grande influence sur l'apprentissage. 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 sont indépendantes au sein du modèle d'entrée, ce comportement de moyennage permet aux activités contextuelles d'effacer leur contribution dans le calcul de l'activité globale ; cette dernière ne prendra alors en compte que les entrées qui ont une dépendance réelle. Cependant, ce comportement de moyennage n'est pas souhaitable si les entrées ont toutes une relation entre elles mais que U est en grande dimension.

Une perspective d'étude de CxSOM est maintenant d'évaluer la distribution de l'encodage du modèle d'entrée dans les cartes. D'une part, 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. Ensuite, dans les expériences présentées dans ce chapitre, le modèle U est une variable 1D. Nous avons observé que l'apprentissage se traduit par le fait que U est une fonction de Π dans chaque carte. Dans une grande architecture apprenant sur des entrées multimodales de dimension totale supérieure, on ne peut pas attendre que chaque carte encode la totalité du modèle U : le nombre de nœuds est limité et l'architecture

se contenterait d'apprendre la valeur moyenne de U , comme dans l'architecture de 10 cartes de ce chapitre. On voudrait donc que U ait une représentation distribuée au travers des 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 trop petites pour pouvoir étudier cette propriété. Cet aspect distribué de l'apprentissage est 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. Nous étudierons des méthodes 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. Nous chercherons donc à utiliser des cartes 2D dans une architecture CxSOM. 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 des poids dans une architecture de cartes en deux dimensions.