

Chapitre 5

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

Sommaire

5.1	Méthode expérimentale	2
5.1.1	Modèle d'entrées géométriques	2
5.1.2	Matériel	3
5.1.3	Paramètres des architectures	4
5.2	Mécanismes d'auto-organisation jointe sur une architecture de deux cartes	5
5.2.1	Mécanismes observés sur les entrées en cercle	5
5.2.2	Organisation des cartes en fonction de la distribution d'entrées	9
5.2.3	Discussion : mécanismes de formation des zones de poids contextuels	10
5.3	Prédiction d'entrée dans des architectures de trois cartes 1D	13
5.3.1	Architecture de trois cartes et modèle d'entrées	13
5.3.2	Résultats	14
5.4	Influence des connexions et des paramètres	17
5.5	Prédiction d'entrée réelles : application au contrôle d'un drone	19
5.5.1	Méthode expérimentale	20
5.5.2	Résultats	22
5.6	Conclusion	23

À partir de la méthode expérimentale et de représentation du comportement de l'architecture que nous avons présentée précédemment, nous nous concentrons dans ce chapitre sur l'identification des dynamiques et comportements d'apprentissage d'une architecture CxSOM. Nous étudions encore ici les cartes en une dimension ; nous ouvrirons l'étude aux cartes en deux dimensions au chapitre suivant. Nous chercherons d'abord à caractériser l'organisation d'architecture de deux et trois cartes en une dimension en étudiant différentes dispositions d'entrées. Nous introduirons ensuite un comportement spécifique à CxSOM : la prédiction d'entrée. Nous montrerons que la relaxation permet de prédire une entrée manquante, conférant à des cartes de Kohonen une capacité de prise de décision. Nous appliquerons cette propriété de prédiction de CxSOM au contrôle d'un drone.

5.1 Méthode expérimentale

Nous reprenons dans ce chapitre l'expérience déjà présentée sur le cercle en deux dimensions et étudions quelques autres dispositions géométriques d'entrées. Ces modèles géométriques nous permettent de maîtriser les dépendances sur des modalités en basse dimension et ainsi de visualiser les liens entre organisation et apprentissage. Cela nous permettra également de mesurer l'apprentissage de cette dépendance connue au sein des structures de cartes.

5.1.1 Modèle d'entrées géométriques

Rappelons la définition des entrées multimodales : il s'agit d'un ensemble d'entrées $\mathbf{X} = (X^{(1)}, \dots, X^{(n)})$. 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 évenutuel sur les entrées. La dépendance entre modalités est définie par la dimension choisie pour U . Une variable U en une dimension paramètre des points placés sur une courbe en une dimension ; U en 2 dimensions paramètre une surface 2D.

Dans cette série d'expériences, nous étudions un modèle dans lequel chaque modalité est en une dimension. Dans ce chapitre, nous considérons des architectures de deux et trois cartes et prenons donc des entrées de dimension totale 2 et 3. Afin de définir une dépendance entre entrées, nous les prenons sur une variété de dimension k inférieure à la dimension totale des entrées. Les données sont ainsi paramétrées par $U \in [0, 1]^k$. Ce modèle est général : au pire, la dimension de U correspond à la dimension totale des entrées. Par ailleurs, de nombreux modèles d'entrées réelles se placent effectivement sur une variété de dimension réduite. Le choix d'un modèle d'entrées géométriques situées sur une variété de dimension inférieure est donc justifié comme modèle expérimental simplifié.

Dans une architecture de deux cartes, nous considérons d'abord des points pour lesquels U est en une dimension : ils sont situés sur une courbe du plan 2D. Nous reviendrons d'abord sur l'expérience sur des données disposée en cercle (A). L'intérêt de cette courbe est que la disposition est symétrique : toute entrée $X^{(1)}$ correspond à deux valeurs possibles pour $X^{(2)}$ et inversement. Nous testerons ensuite si les observations réalisées sur cette disposition d'entrées se retrouvent pour d'autres dispositions d'entrées listées ci-dessous, représentées en figure 5.1 :

- Une entrée est fonction de l'autre : $X^{(2)} = \cos(X^{(1)})$ (B)
- Entrées identiques (cas dégénéré). Ces points sont toujours sur une courbe 1D, mais leur dépendance est bijective. (C)
- Entrées sur une courbe de Lissajous (D) : une entrée $X^{(1)}$ correspond à 4 à 6 valeurs de $X^{(2)}$ et inversement.
- Entrées totalement indépendantes, prises aléatoirement dans un carré $[0, 1]^n$ (E). U est alors une variable 2D correspondant aux entrées.
- Une carte de Kohonen classique a comme propriété d'être résistante au bruit des données. Ainsi, une carte 1D se dépliant sur un anneau fin en 2D apprendra d'abord la représentation du cercle. Nous voulons vérifier comment cette propriété se vérifie sur l'apprentissage de données par plusieurs cartes ; nous prendrons ainsi des points sur un anneau (F). U est ici en une dimension, et chaque dimension est bruitée.

Sur ces différents jeux de données, nous étudierons l'organisation après apprentissage de deux cartes connectées réciproquement. Nous suivons la méthode expérimentale et les représentations élaborées au chapitre ?? Dans un premier temps, nous étudions des architectures de deux cartes 1D,

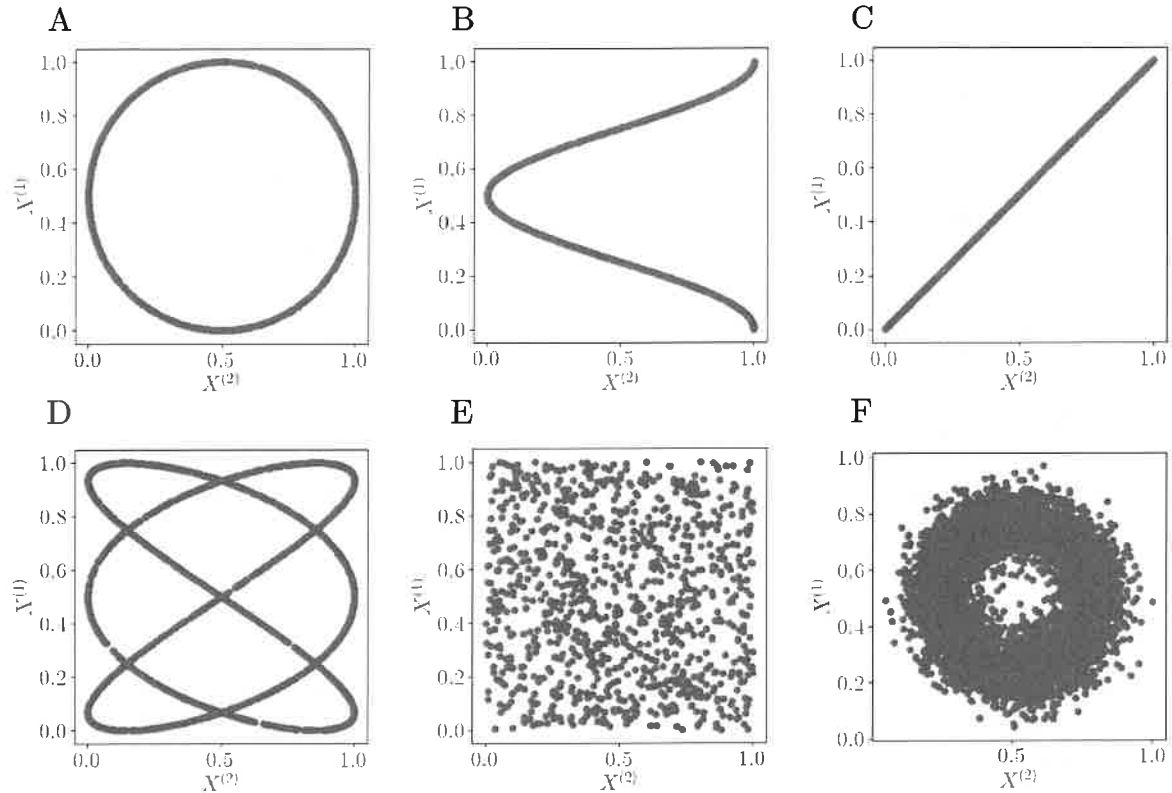


FIGURE 5.1 – Dispositions d’entrées en deux dimensions. $M^{(1)}$ prend en entrée l’ordonnée $X^{(1)}$ et $M^{(2)}$ prend en entrée l’abscisse $M^{(2)}$.

ce qui nous permettra de mieux comprendre certains mécanismes d’organisation. Nous étendrons ensuite notre étude à des modèles d’architectures de trois cartes. Nous prendrons deux configurations d’entrées construites à partir d’objets en deux dimensions plongés en trois dimensions par la transformation illustrée en figure 5.2 :

- Les entrées sont dans un plan 2D pivoté en trois dimensions
- Les entrées sont sur un cercle en 2D, pivoté en trois dimensions

Dans ces deux cas, la connaissance de deux des entrées ainsi que du modèle détermine la valeur de la troisième entrée. Nous étudierons ces configurations dans un cadre de prédiction d’entrées : nous donnons en entrées $X^{(1)}$, $X^{(2)}$ à la structure et regardons si la valeur de $X^{(3)}$ correspondante est correctement prédite par l’architecture. Une bonne prédiction témoigne de l’apprentissage du modèle d’entrées par l’architecture de cartes. Les architectures de cartes en deux dimensions seront traitées par la suite au chapitre ??.

5.1.2 Matériel

Le modèle d’architecture CxSOM est implémenté en C++ en s’appuyant sur la librairie CxSOM¹, développée au sein de notre équipe. Cette librairie permet d’implémenter des cartes de Kohonen simples ainsi que des architectures CxSOM spécifiquement. Cette librairie s’interface avec

1. <https://github.com/HerveFrezza-Buet/cxsom>

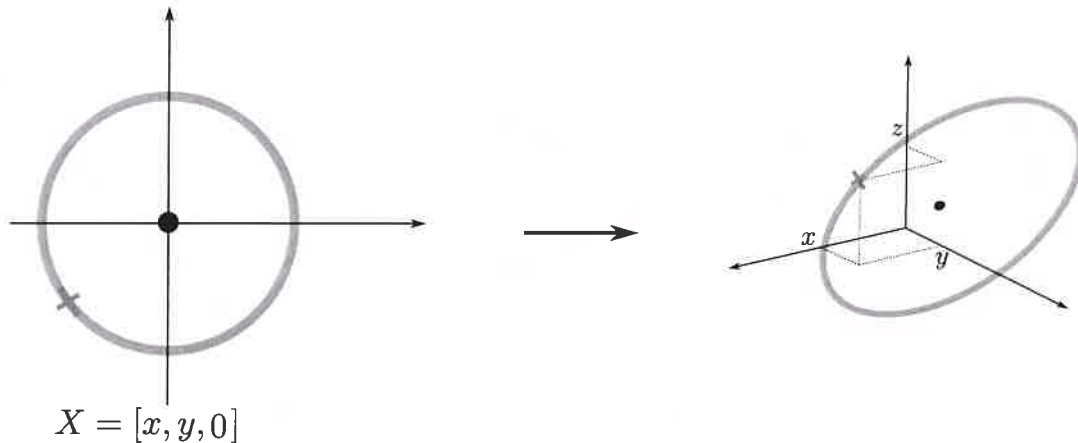


FIGURE 5.2 – Exemple de courbe plongée en trois dimensions. La figure 2D est pivotée en trois dimensions. Chaque coordonnée est normalisée de façon à s'étendre entre 0 et 1 en trois dimensions.

un module python (*pycxsom*) afin de faciliter les représentations et manipulation des cartes pendant et après l'apprentissage. CxSOM permet de paralléliser au maximum les opérations indépendantes. Par exemple, la phase d'apprentissage des cartes est séquentielle car le calcul des valeurs des poids pour l'itération i dépend de $i - 1$, mais toutes les opérations de tests dans lesquelles le temps n'intervient pas tournent en parallèle. Notons que la gestion du consensus lors de la relaxation passe également par des mécanismes locaux aux cartes dans notre implémentation. Nous devons en effet vérifier lors de chaque pas de relaxation si les cartes ont atteint un consensus. Pour cela, chaque carte envoie un signal supplémentaire aux cartes voisines indiquant si son BMU a été modifié. Lorsque qu'une carte reçoit de toutes ses voisines que leur BMU n'est plus modifié et que le BMU de la carte n'a pas non plus été modifié lors de l'étape, la relaxation s'arrête dans cette carte. Les codes C++ et python que nous avons utilisé pour générer les expériences présentées dans ce chapitre sont disponibles sur git : REF. Toutes les expériences présentées ici tournent sur un processeur i7 4 cœurs.

Tu peux dire que tu as dégrossi cxsom-builder avant que ces fonctionnalités soient mûrées. Mais ne t'étila pas, mais c'est du dev que tu as fait quand-même.

5.1.3 Paramètres des architectures

Nous nous concentrons d'abord sur des architectures de deux cartes, prenant chacune en entrée une modalité en une dimension. Chaque carte a une taille fixée de 500 nœuds, indexées 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 autrement dans l'expérience. La génération des entrées suit le processus suivant : U est tiré uniformément dans $[0, 1]$, puis les entrées $X^{(1)}$ et $X^{(2)}$ sont calculées à partir de la valeur de U . L'apprentissage est réalisé sur un échantillon de 20000 itérations, générés aléatoirement. Notons que ce nombre d'itérations est souvent largement supérieur au nombre d'itérations effectivement nécessaires à la convergence des poids. Les tests sont ensuite réalisés sur 1000 points générés aléatoirement selon la même distribution d'entrées.

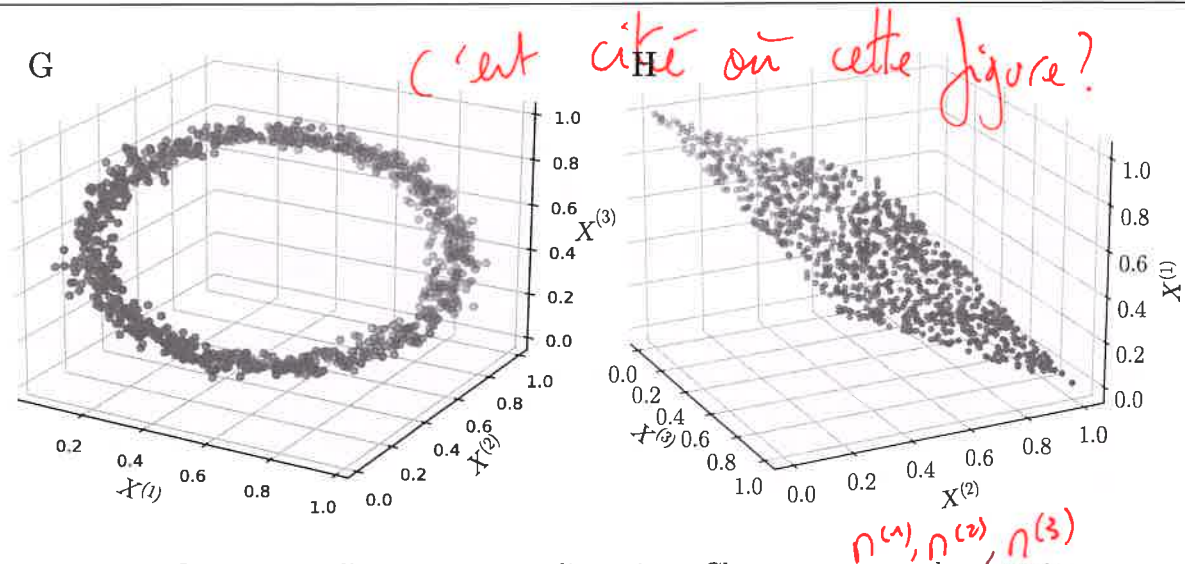


FIGURE 5.3 – Dispositions d'entrées en trois dimensions. Chaque carte prend en entrée une coordonnée $X^{(1)}, X^{(2)}, X^{(3)}$.

5.2 Mécanismes d'auto-organisation jointe sur une architecture de deux cartes

Le premier but de cette étude est d'identifier des comportements *systémiques* émergeant d'une architecture simple à deux et trois cartes, sur des entrées en deux et trois dimensions.

5.2.1 Mécanismes observés sur les entrées en cercle

Revenons d'abord sur l'expérience précédemment présentée au chapitre ??, réalisée sur les entrées disposées selon un cercle (A). Nous avons alors relevé plusieurs comportements se différenciant de ce qu'on observe dans une carte auto-organisatrice classique.

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 voulons en effet laisser la possibilité d'intégrer des connexions récurrentes temporelles aux architectures, dans quel cas il ne serait pas possible de diminuer les paramètres au cours de l'apprentissage. La question de la convergence des poids des cartes est donc à vérifier.

La figure 5.4 présente l'évolution de la modification des poids ω_e et ω_c dans chaque carte au cours de l'apprentissage. Tous les 1000 itérations, nous calculons la différence maximale entre w_t et w_{t-1000} . Nous observons que cette courbe tend vers 0 pour chaque courbe de poids. Cela montre que tous les poids de la carte tendent vers une position stable. Cette convergence est favorisée par la différence de contribution dans l'activité globale des activations externes et contextuelles, calculée

utilise les guillemets français !

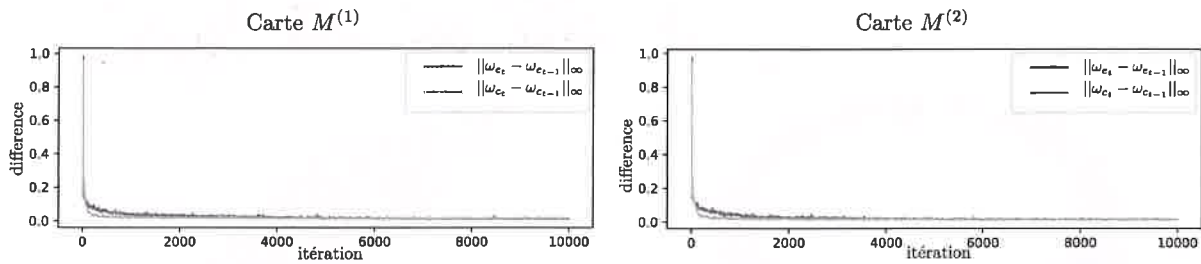


FIGURE 5.4 – Pour chaque carte, nous représentons l'évolution en fonction du temps d'apprentissage, de la différence maximale en valeur absolue entre les poids à l'instant t et ceux à $t - 1$ ω_t et ω_{t-1} . Les entrées sont ici un cercle en deux dimensions. L'évolution est moyennée sur 10 apprentissages 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 stable.

par :

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

La carte se comporte donc d'abord comme une carte de Kohonen classique apprenant sur des entrées externes, qui pour des cartes 1D sur des entrées 1D converge sans difficulté même en l'absence de décroissance des paramètres d'apprentissage. Les entrées contextuelles viennent seulement moduler le calcul d'activité. Notons toutefois que nous sommes sur un cas particulier de cartes 1D sur des entrées 1D. La convergence en l'absence de décroissance de paramètres peut poser plus de problèmes sur des cartes en deux dimensions. Nous verrons que pour des rayons de voisinage bien choisis, la convergence sera possible également en deux dimensions.

Disposition des poids

Analysons maintenant la forme des poids contextuels des deux cartes, tracés en figure 5.5. Les poids externes, en orange, 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 bleu, présentent une forme de "vagues". Nous ajoutons à ces deux courbes le tracé des valeurs des entrées selon la position de leur BMU lors de la phase test, en vert et rose sur la figure.

Nous remarquons d'abord 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 nœ sont BMUs correspondent aux extrema des poids contextuels et leurs alentours.

Dans la carte $M^{(1)}$, les entrées externes $X^{(1)}$ sont proches de la courbe de poids externes, mais avec plus d'erreur de quantification que ce qu'on attendrait dans une carte simple. Une quantification vectorielle correcte est souhaitable, car elle permet de lier de retrouver la valeur de l'entrée à partir de la réponse d'une carte. Les zones définies par les poids contextuels se partagent les positions des BMUs en fonction de la valeur des entrées. Deux points ayant la même valeur de $X^{(1)}$ mais une valeur différente de $X^{(2)}$ auront un BMU différent dans la carte $M^{(1)}$, mais ces BMUs sont situés dans deux zones adjacentes.

Dans la carte $M^{(1)}$, une position se spécialise donc en tant que BMU par rapport aux entrées

c'est tout au pluriel ou il y a une stabilité qui m'échappe?

5.2. Mécanismes d'auto-organisation jointe sur une architecture de deux cartes

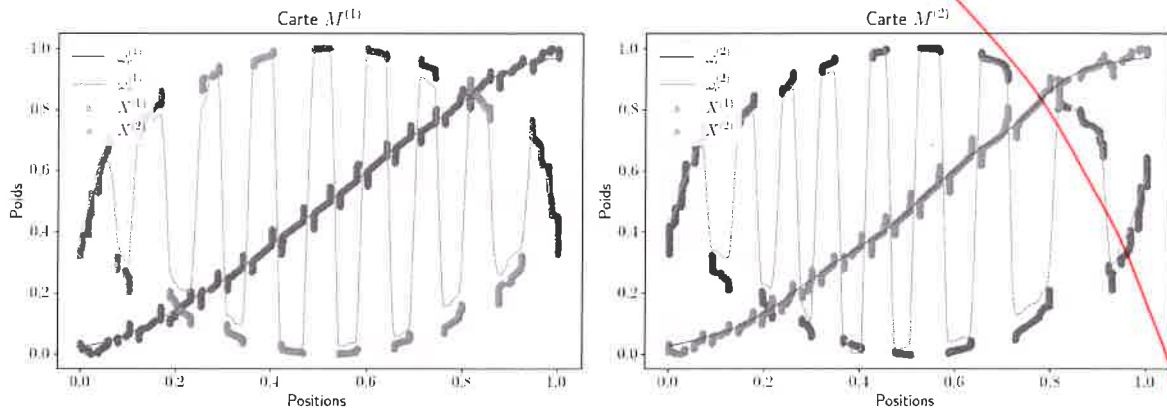


FIGURE 5.5 – Représentation cartographique des poids et entrées lors d'une phase de test selon la position dans chacune des cartes. Nous remarquons que les poids d'une carte, par exemple la carte $M^{(1)}$ s'organisent en zones différenciant les valeurs de la paire $X^{(1)}, X^{(2)}$ et non seulement de 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.

externes et ~~l'entrée~~ contextuelles. C'est bien ce à quoi on s'attendait en ayant deux couches de poids. Cette différenciation est réalisée par la répartition des unités 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 classique. Deux zones adjacentes correspondent par ailleurs à des segments de valeur d'entrée qui se recouvrent, et des segments de valeurs d'entrées contextuelles différentes. 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, et le nombre d'unités allouées par auto-organisation dans chaque zone est à peu près égal.

Erreur de quantification vectorielle

La Figure 5.7 présente la valeur de l'entrée quantifiée au sein de chaque carte $\omega_e(\Pi^{(i)})$ en fonction de l'entrée présentée $X^{(i)}$. Nous pouvons observer que la quantification vectorielle est bien réalisée. Cette capacité de quantification permet de retrouver la valeur de l'entrée à partir du BMU de la carte et donne donc une sortie interprétable dans l'espace des entrées à l'architecture CxSOM. Cependant, l'erreur de quantification est plus forte que ce qu'on peut attendre avec une carte de même taille et mêmes paramètres apprenant sur l'ensemble de $X^{(i)}$. Nous remarquons une disposition en étages, dues aux zones formées par les poids contextuels. En effet, les nœuds de zones adjacentes de la carte sont en effet des Best Matching Unit pour des intervalles d'entrée qui se recoupent. Les poids externes entre deux zones sont différents : il n'y a pas de plateaux dans les poids externes. 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. La carte prend son BMU de façon à ce que

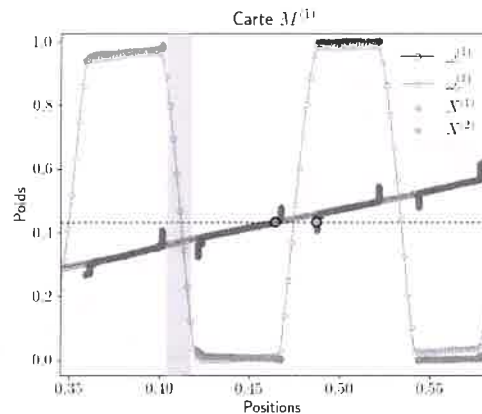


FIGURE 5.6 – Zoom sur la figure 5.5 entre les positions 0.35 et 0.55 de la carte $M^{(1)}$. Nous y faisons apparaître la position sur la courbe des nœuds de la carte. Deux zones consécutives seront BMUs pour des ensembles d'entrées qui se recouvrent. Par exemple, les deux entrées correspondant aux points bleu et rouge 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, introduisant un semblant de discontinuité.

!!!

son poids se rapproche au maximum de l'entrée, mais il sera soit un peu trop bas par rapport à la valeur de l'entrée, soit un peu trop haut.

Conclusion

Les résultats de cette expérience ainsi que les observations présentées au chapitre ?? nous permettent donc de formuler les hypothèses suivantes concernant le comportement de cartes en une dimension :

- Chaque carte de l'architecture présente une faible erreur de quantification vectorielle sur ses entrées externes 5.7
- Les poids contextuels de chaque carte s'organisent en zones distinctes. Une zone correspond à un même intervalle de valeur pour X et U . Deux zones adjacentes encodent le même intervalle de valeurs pour X mais des valeurs distinctes de U . 5.5 Ces zones se caractérisent par un étirement de la carte entre deux valeurs éloignées, apportant un aspect discontinu. Cette discontinuité passe en fait par la présence d'une zone peu dense de la carte contenant des nœuds qui ne seront jamais BMUs 5.6. Nous verrons en section Prédiction que la formation de ces zones permet à la carte de prendre des décisions lors d'une phase de prédiction d'entrée.
- 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 des entrées externes.

Nous chercherons à vérifier et compléter ces observations sur d'autres dispositions d'entrées dans des structures de deux cartes. Nous étudierons 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 introduisons ensuite un nouveau comportement possible dans une architecture : la

pas de fig pour sa?

met le numéro

possible que 5.7 et 5.6 permettent? ça serait cohérent avec la lecture.

5.2. Mécanismes d'auto-organisation jointe sur une architecture de deux cartes

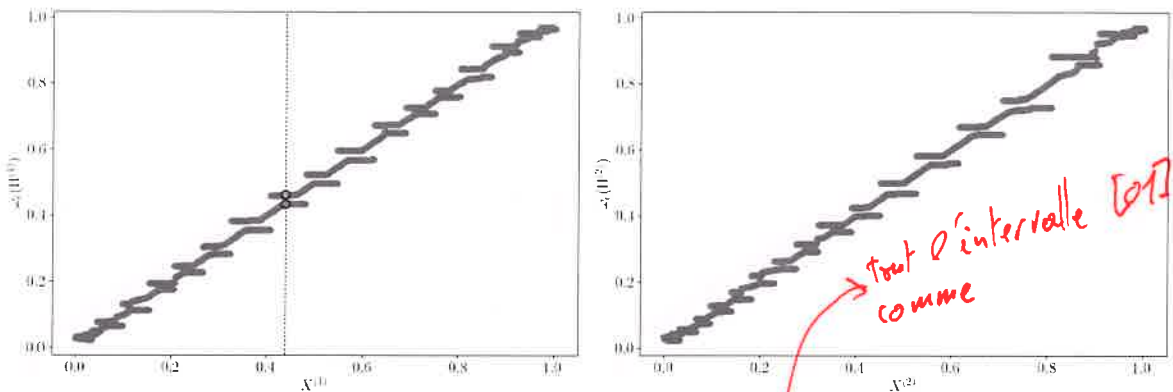


FIGURE 5.7 – Représentation de l'erreur de quantification sur les valeurs de $X^{(1)}$ et $X^{(2)}$. Le poids externe du BMU est proche de la valeur de l'entrée; chaque carte réalise ainsi une bonne quantification vectorielle sur ses entrées. Les poids rouges et bleus représentés en figure 5.6 sont reportés sur le graphique.

prédiction d'entrée manquante. L'étude de ce comportement nous permettra de compléter l'étude de la dynamique d'apprentissage et d'envisager des applications.

5.2.2 Organisation des cartes en fonction de la distribution d'entrées

Présentons maintenant l'organisation des poids et des BMUs sur les différentes dispositions d'entrées. Dans toutes ces dispositions, la quantification vectorielle est bien réalisée dans chaque carte sur ses entrées. Nous nous concentrerons sur la présence ou non de zones de poids contextuels.

La figure 5.8 présente la disposition des poids et entrées des cartes lorsque $X^{(1)}$ et $X^{(2)}$ sont identiques. Nous observons que les poids externes et contextuels ne forment pas de zones et les deux cartes se comportent comme une seule carte simple sur $X^{(1)} = X^{(2)}$. En figure 5.9, la dépendance entre les entrées présentée n'est plus bijective mais $X^{(2)}$ est fonction de $X^{(1)}$. La carte $M^{(1)}$ ne forme pas de zones, car une seule valeur de $X^{(2)}$ correspond à une même valeur de $X^{(1)}$. 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 qui rejoint les observations présentées sur le cercle.

Nous traçons en Figure 5.11 l'organisation obtenue pour des points placés sur une courbe de Lissajous, dans laquelle une valeur de $X^{(1)}$ correspond à 4 à 6 valeurs de $X^{(2)}$ et inversement. Enfin, la figure 5.10 présente l'organisation obtenue lorsque les points sont dans le carré $[0, 1]^2$: les entrées sont indépendantes, une valeur de $X^{(1)}$ correspond à une infinité de valeurs pour $X^{(2)}$.

Dans ces deux derniers cas, les cartes présentent une organisation en zones des poids contextuels, tout comme le comportement observé sur le cercle. 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. Contrairement au cercle, les BMUs se répartissent sur toutes les valeurs de ω_c dans une zone, réduisant le nombre de nœuds morts entre les zones. Nous pouvons en conclure que la présence de zones en tant que telle est un comportement systématique de la carte étant donné qu'elles sont observées même lorsque les entrées sont indépendantes. Par contre, la forme des zones dépend de la relation entre entrées.

Sur la distribution indépendante, la carte ne présente pas de zone morte. La totalité d'une zone

En fait le doit: "on voit bien que le zonage est mobilisé par la carte quand il est nécessaire, c'est émergeant!" (à dire rien x)

fait en sorte que les numéros soient dans l'ordre de lecture.

tout d'intervalle [0,1] comme

dire plaque, cette cart joue le tour.

("raies" verticales).

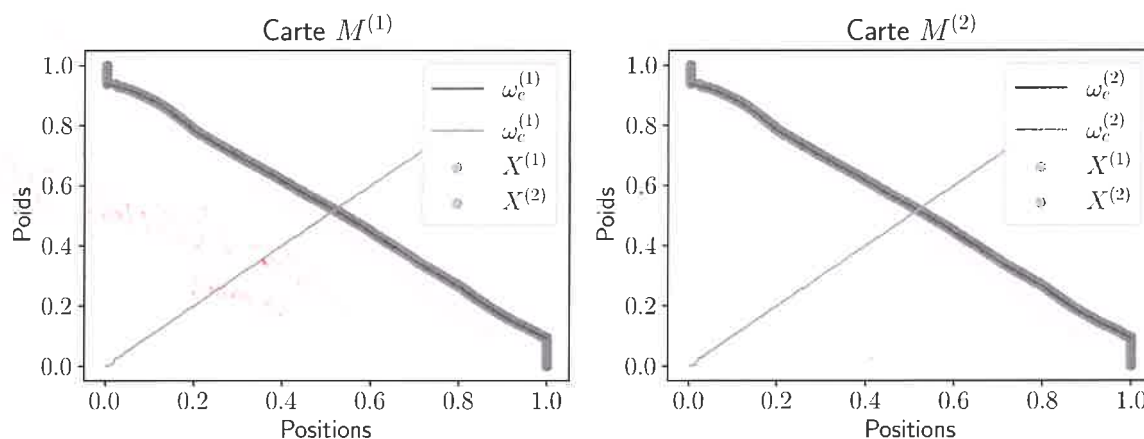


FIGURE 5.8 – Représentation cartographique des poids et entrées pour la disposition identité. Les poids externes et contextuels sont superposés, et les poids contextuels n'ont pas besoin de former de zones

se déploie de manière à couvrir l'ensemble des valeurs de U correspondant à cette zone, ce qui est également observé en figure 5.11 pour les courbes de Lissajous. Une zone agit alors comme une petite carte d'une sous-région de l'espace d'entrée. Nous pouvons le constater sur la Figure 5.12 présentant la distorsion des poids externes des cartes dans l'espace $X^{(1)}; X^{(2)}$ lorsque les entrées sont dans le carré. Les poids des cartes quantifient tout l'espace $[0, 1]^2$, en notant qu'une centaine de points servent à la quantification, pour deux cartes de taille 500. La carte $M^{(1)}$ parcourt l'espace selon les valeurs de $X^{(1)}$ tandis que $M^{(2)}$ le parcourt selon les valeurs de $X^{(2)}$.

Enfin, nous voulons vérifier si les cartes sont robustes au bruit. Sur des données en forme d'anneau (Entrées (**E**)), représentées en Figure 5.13, nous observons le même comportement que sur les courbes de Lissajous ou le carré : le nombre de zones de poids contextuels est défini par l'organisation de la carte. Chaque zone correspond à un même intervalle de valeurs pour le couple (X, U) . Enfin, notons que dans toutes ces dispositions d'entrées, la quantification vectorielle sur les entrées externes est bien réalisée. Le tracé de U en fonction de la position du BMU dans chaque carte montre une relation fonctionnelle entre U et les positions du BMU, ce qui montre que chaque carte de l'architecture a appris une représentation du modèle d'entrée.

5.2.3 Discussion : mécanismes de formation des zones de poids contextuels

Nous avons observé sur ces distributions d'entrées en 2D les comportements suivants qui vérifient et complètent les hypothèses formulées sur la disposition d'entrées en cercle. Tout d'abord, la quantification vectorielle reste correctement réalisée sur les différentes dispositions d'entrées. Les valeurs prises par les poids sont toujours disposées en étages à cause des zones. Cette organisation en zones intervient dès que la disposition des entrées implique d'avoir à séparer au moins deux valeurs de U pour une même entrée externe. Le fait que des zones existent n'implique pas la détection d'une relation entre entrées : la carte forme des zones lorsque les entrées sont indépendantes. Cette formation de zones est systémique aux mécanismes d'auto-organisation des cartes. Le nombre de zones ne varie pas en fonction des distributions d'entrées. Cela implique que les mécanismes d'évolution de la carte conduisent à la formation des zones, et que l'organisation s'adapte ensuite

du fait distinguer

C'est vraiment bien ça, bravo!

5.2. Mécanismes d'auto-organisation jointe sur une architecture de deux cartes

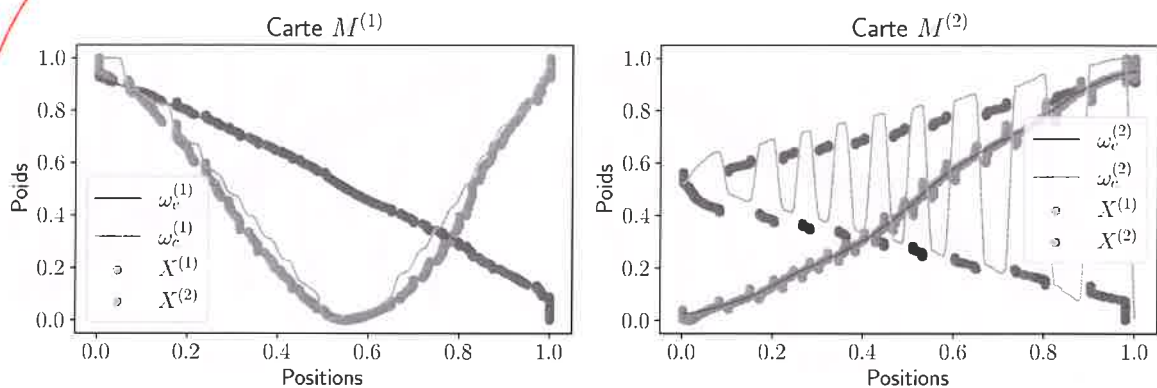


FIGURE 5.9 – Représentation cartographique des poids et entrées pour $X^{(2)} = \cos(X^{(1)})$. Les poids contextuels de la carte $M^{(1)}$ ne forment pas de zones car une seule valeur de $X^{(2)}$ correspond à une entrée $X^{(1)}$. Au contraire, les poids de la carte $M^{(2)}$ s'organisent pour gérer la distinction.

à cette disposition. ... *on se rend de la bête: "OK, mais par quel miracle à tous ces zones ???"*

Nous avons relevé dans nos expériences que la formation de zones dépend notamment des valeurs des rayons externes et contextuels. Pour $r_c \geq r_e$, la carte ne s'organise pas en zones. La formation de celles-ci intervient pour $r_c \ll r_e$ et leur nombre dépend ensuite du rapport entre les rayons de voisinage. La disposition des cartes repose ainsi complètement sur les zones formées par les poids contextuels. Ces zones apparaissent grâce au fait que la proximité des poids externes est priorisée par rapport aux poids contextuels par le grand rayon de voisinage externe. Ce rapport introduit une relation subordonnée entre les poids. Il semble que le nombre de zones dépend en partie du rapport entre r_c et r_e . Une étude plus approfondie devrait être réalisée pour analyser l'influence des paramètres sur l'organisation. Par contre, l'organisation au sein d'une zone et la forme des poids contextuels dépend de la dépendance entre les entrées.

Nous pensons que ces zones favorisent la recherche de consensus lors de la relaxation et la convergence des poids; mais nous n'avons pas observé significativement cet avantage en une dimension, dans laquelle les poids des cartes convergent très facilement. Le nombre de pas moyen nécessaire à la recherche du BMU par relaxation et les indicateurs de convergence des poids des cartes sont similaires dans les cas d'une carte formant des zones et d'une carte n'ayant pas formé de zones. (Voir Figure 5.15, dans laquelle cette comparaison est effectuée sur une architecture de trois cartes).

Au sein d'une même zone, dans laquelle les poids externes ont des valeurs très proches, les poids contextuels s'organisent de manière à former une sous-carte des valeurs possibles de l'entrée contextuelle. On pourrait donc introduire une notion d'indices primaires et secondaires, 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 également observée dans le cerveau, proposé en 1986 par (Ballard 1986). Les neurones du cortex V1, par exemple, gèrent leurs connexions et leur organisation comme schématisé en figure 5.14. Les neurones situés à différents emplacements sur cortex V1 ne reçoivent pas la même partie de l'entrée visuelle. Ces entrées différenciées 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 la zone. Cette sous-carte définit alors des indices secondaires. Ici, la même entrée est certes présentée à toute la carte, donc la proximité avec

ex: "Il se pose alors légitimement la question de ce qui préside à la formation de ces zones, nous explorons les ordinaux suivants..."

OK, c'est le processus suivant, j'ai donc sur une transition

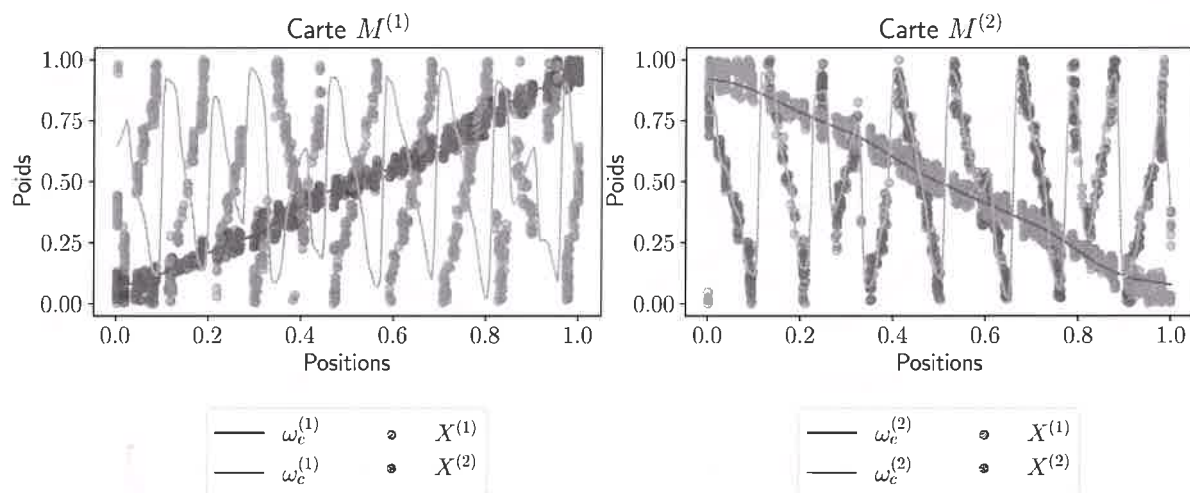


FIGURE 5.10 – Représentation cartographique des poids et entrées pour la disposition carré

le modèle biologique est limitée; mais les BMUs des zones réagissent en fait à un sous-ensemble d'entrées externes définies par les poids contextuels. Il s'agit d'une sorte de mécanisme d'attention qu'on peut comparer au fait qu'une zone de même indice primaire chez Ballard reçoit seulement un sous-ensemble d'entrées.

D'un point de vue computationnel, la forme des poids contextuels rappelle également la notion de *positional encoding*, une technique utilisée pour définir un mécanisme d'attention en apprentissage de séquence dans les réseaux de type *Transformers* (Vaswani et al. 2017). Cette méthode module la valeur d'un vecteur représentant un mot par un poids sinusoïdal en fonction de la position du mot dans la séquence, ce qui permet d'encoder facilement à la fois la valeur du mot et sa position en un seul réel par l'alternance des valeurs du sinus. L'organisation d'une carte encode finalement une méthode similaire à cette technique d'encodage : la valeur de l'activité externe est modulée par une activité contextuelle alternant des valeurs hautes et basses, qui encode ainsi en une seule valeur à la fois la valeur de X et la valeur de U , le modèle d'entrée.

Le comportement des cartes jointes introduit enfin un phénomène de discontinuité dans les poids contextuels, chose peu courante dans une carte auto-organisatrice classique. Cette discontinuité est en fait une fausse discontinuité puisque des zones mortes comptant quelques nœuds n'étant jamais BMU séparent deux zones de poids contextuels. On observe que plus on a de valeurs possibles pour $X^{(2)}$ pour une même valeur de $X^{(1)}$, moins on aura de nœuds morts dans les zones de discontinuité, comme le montre l'exemple du carré dans lequel il n'y a pas de nœuds morts entre zones. Les zones sont à peu près réparties équitablement sur la carte. Les zones forment ainsi un deuxième niveau de quantification vectorielle, moins précis que le premier, mais qui encode la valeur de U et non seulement de l'entrée X . Ce deuxième niveau de quantification vectorielle encode l'information sur U dans chaque carte et nous permet l'utilisation de l'architecture comme prédiction d'une entrée manquante.

NON! ça fait référence à trop de choses. "mécanisme de quantification" plutôt non?

là OK, ça va.

donc ça devient une vraie discontinuité?

les pigé

globalement, c'est pas mal sinon.

↑ pas pas un $\begin{cases} figure \end{cases} [t]$ par hasard?
toutes les fig sont au top de page
et elles s'éloignent du texte.

5.3. Prédiction d'entrée dans des architectures de trois cartes 1D

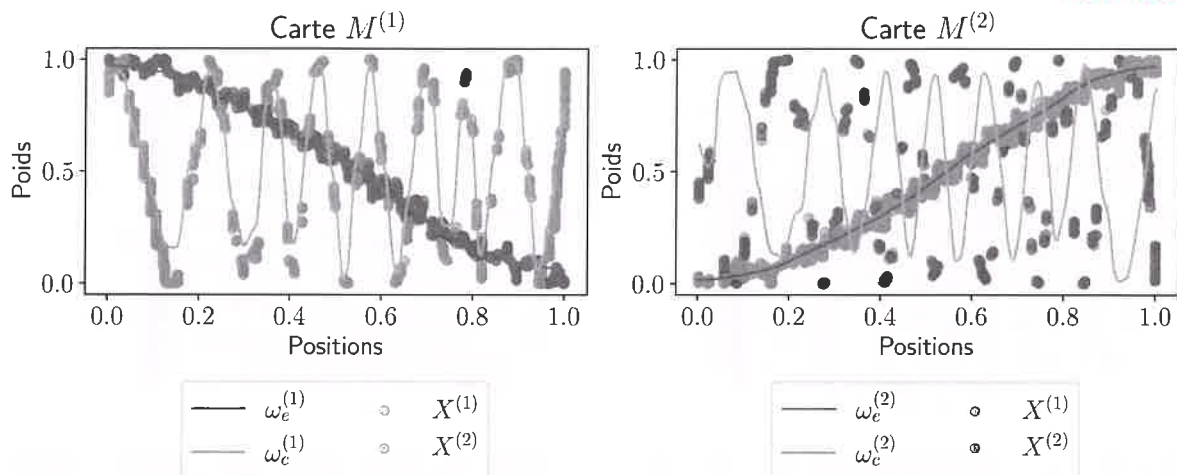


FIGURE 5.11 – Représentation cartographique des poids et entrées pour des entrées sur une courbe de Lissajous. Les poids contextuels continuent de former des zones.

5.3 Prédiction d'entrée dans des architectures de trois cartes 1D

L'observation des mécanismes d'organisation à l'œuvre dans des architectures de deux cartes nous confirme que le modèle d'entrées est bien appris dans chacune des cartes de l'architecture grâce aux connexions entre cartes. Nous utilisons maintenant le fait qu'une architecture ait appris le modèle dans une tâche de prédiction d'entrée.

Cette utilisation en tant que prédiction a certes une valeur applicative, car il s'agit d'un cas d'utilisation possible d'une architecture CxSOM pour une tâche de mémoire associative. Néanmoins, la prédiction permet avant tout de valider l'apprentissage du modèle par l'architecture d'un point de vue d'une carte. Par exemple, dans le cas de données réelles, le modèle d'entrée n'est pas forcément connu. La prédiction permet alors de valider l'apprentissage du modèle : étant donné un modèle d'entrées dans lequel la connaissance de deux entrées et du modèle détermine la valeur de la troisième entrée ; si l'architecture de cartes, à partir de ces deux entrées est capable de prédire la valeur de la troisième entrée, alors nous pouvons conclure que l'architecture a bien appris les relations entre entrées.

5.3.1 Architecture de trois cartes et modèle d'entrées

La structure utilisée pour la tâche de prédiction est présentée en Figure 5.16. Dans un premier temps, les cartes CxSOM sont mises à jour sur l'ensemble des entrées dans une phase d'apprentissage. Ensuite, nous choisissons une des cartes comme carte prédictive. Lors de la phase de prédiction, cette carte ne reçoit plus d'entrée externe, mais seulement ses entrées contextuelles. Les autres cartes reçoivent quant à elles toujours leurs entrées externes et contextuelles. La phase de prédiction est une phase de test durant laquelle les poids de toutes les cartes ne sont pas mis à jour. Les règles de calcul et paramètres sont conservés entre apprentissage et prédiction. La carte ne recevant pas d'entrée externe prend comme activité globale son activité contextuelle seulement, c'est à dire la moyenne des activités liées à chaque entrée contextuelle reçue. La relaxation nous permet de trouver un BMU dans cette carte. Nous choisissons comme valeur de prédiction le poids externe

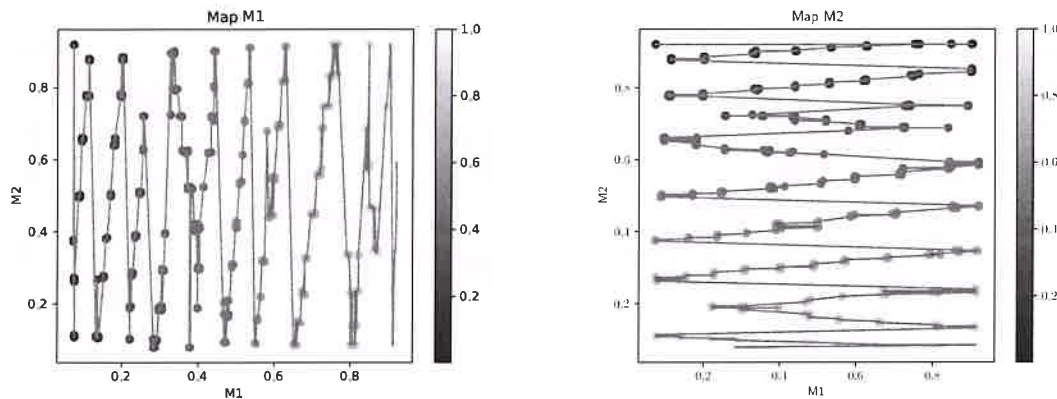


FIGURE 5.12 – Représentation de la distortion des poids des deux cartes dans l'espace d'entrée $X^{(1)}, X^{(2)}$ lorsque les entrées sont indépendantes. Les cartes s'organisent de façon à quadriller le carré, l'une selon les $X^{(1)}$, l'autre selon les $X^{(2)}$. Bien que chaque carte a 500 nœuds, on observe seulement environ 90 valeurs possibles pour les paires $\omega_c(\Pi^{(1)}), \omega_c(\Pi^{(2)})$

du BMU de la carte prédictive : $\omega_c^{(p)}(\Pi^{(p)})$. Ce BMU a été choisi uniquement grâce aux entrées contextuelles de la carte. Nous vérifierons alors si cette prédiction des cartes en une dimension est proche de l'entrée théorique $X^{(p)}$, qui n'a pas été présentée à la carte.

5.3.2 Résultats

Nous testons d'abord la qualité de la prédiction sur des dispositions d'entrées géométriques. Pour qu'une carte puisse faire une prédiction correcte, nous choisissons des dispositions d'entrées telles que la connaissance des entrées et du modèle détermine l'entrée manquante. Nous prendrons ainsi des entrées disposées sur un cercle en deux dimensions plongé et pivoté dans l'espace en 3D ainsi que sur un plan pivoté en trois dimensions. Ainsi, la connaissance de deux des trois coordonnées permet de déterminer la troisième avec précision.

Ces tâches de prédictions sont réalisées sur une architecture de trois cartes 1D, toutes connectées entre elles. Chaque carte prend donc deux entrées contextuelles, les positions des BMUs des deux cartes voisines, et de ce fait possède deux couches contextuelles ω_{c_0} et ω_{c_1} . Nous étendrons la capacité de prédiction aux cartes deux dimensions au chapitre suivant.

La figure 5.17 présente la disposition des poids des trois cartes de l'architecture après apprentissage ainsi que des entrées associées. Comme dans la version à deux cartes, les poids contextuels s'organisent en plusieurs zones au sein desquelles la valeur de U est située dans une même plage de valeur. Nous pouvons remarquer que les deux couches de poids contextuels forment les mêmes zones. Le nombre de zones est équivalent à celui observé pour l'architecture de deux cartes. La figure 5.18 présente ensuite l'erreur obtenue lors de la phase de prédiction. Ici, l'entrée $X^{(1)}$ n'est pas présentée à la carte et sa valeur est prédite par $\omega_c^{(1)}(\Pi^{(1)})$. Cette figure nous montre que la prédiction est correctement réalisée par l'architecture de cartes. Remarquons que la disposition de l'erreur en lignes horizontales correspond aux zones définies par les valeurs des poids contextuels d'une carte.

Nous traçons également en Figure 5.20 la disposition des cartes et l'erreur de prédiction obtenue pour des entrées situées sur un plan 2D de l'espace en trois dimensions. Ici encore, La connaissance

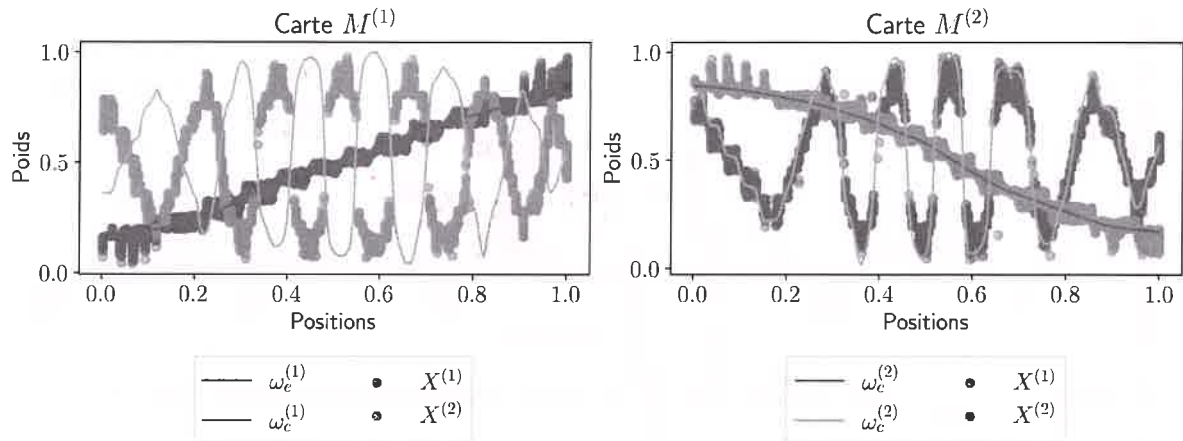


FIGURE 5.13 – Représentation cartographique des poids et entrées pour des entrées sur un anneau.

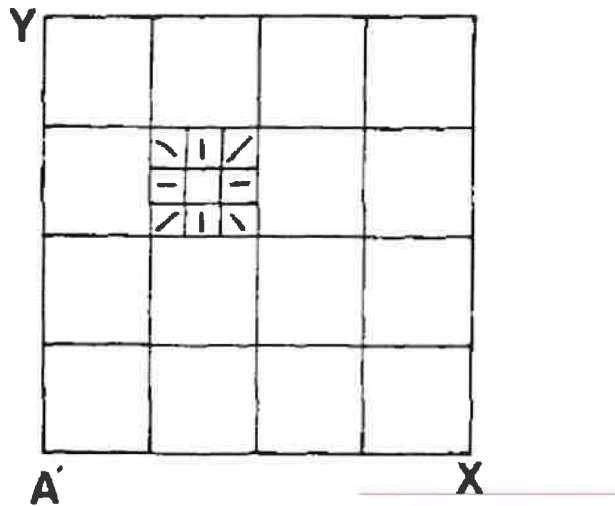


FIGURE 5.14 – Schéma d'une répartition en indices primaires et secondaires des neurones d'une aire corticale, tirée de (Ballard 1986). Les auteurs observent que sur une carte rétinotopique des neurones, c'est à dire tracées en fonction de la position des neurones sur le cortex, la réponse de neurones est organisée en zones, les indices primaires, recevant en connexion différentes portions de l'espace d'entrée. Au sein d'une même zone, les neurones cartographient toutes les valeurs possibles de l'entrée sous forme de carte topologiquement ordonnée, formant des indices secondaires.

retinotopie = organisation sur la surface
du cortex en fonction de la zone de la
rétine qui nourrit le bout de
cortex en question.

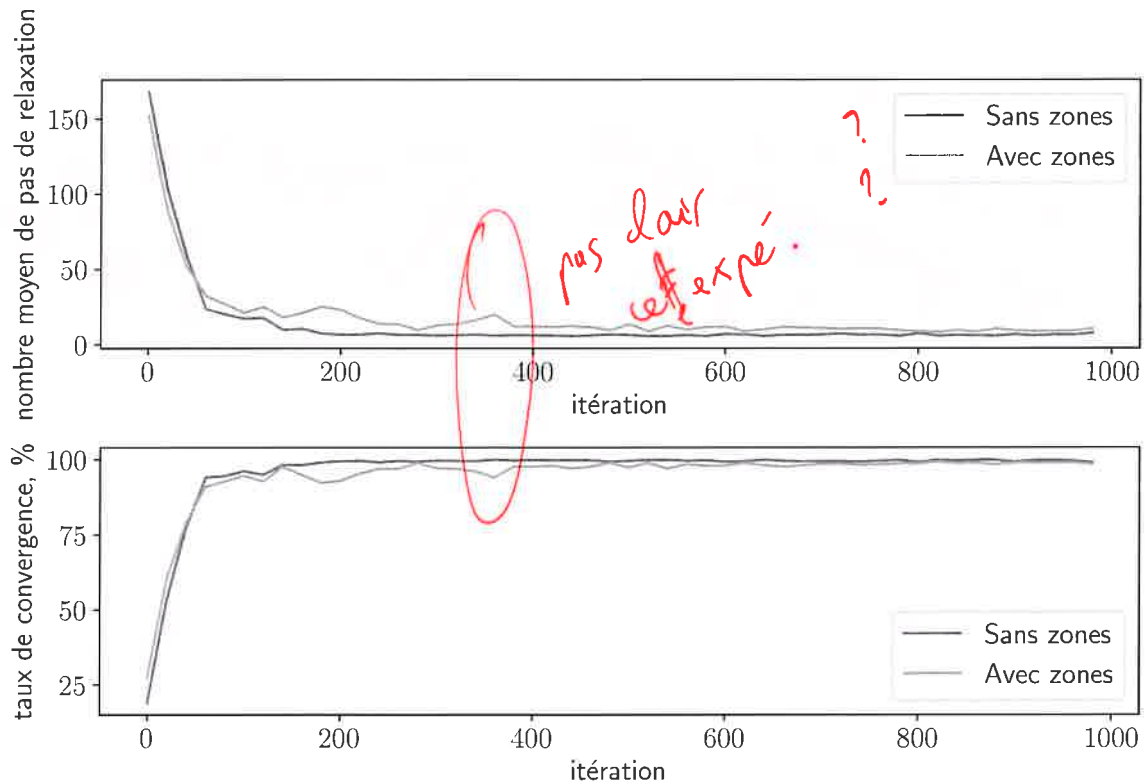


FIGURE 5.15 – Évolution du nombre moyen de pas de relaxation et du taux de convergence pour une organisation de cartes ayant formé des zones de poids contextuels ($\frac{r_e}{r_c} = 10$) et une organisation n'ayant pas formé de zones ($\frac{r_e}{r_c} = 1$). Dans les deux cas, la relaxation mène à un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

de $X^{(2)}$ et $X^{(3)}$ définit bien une seule valeur possible de $X^{(1)}$. La prédiction est bien réalisée, en remarquant une erreur assez élevée. Nous avons vu que les poids externes des cartes s'étaient étalés sur le plan en discrétisant l'espace en une centaine de points seulement, bien que chaque carte soit de taille 500 (Voir Figure 5.12). Cette discrétisation se retrouve dans l'erreur de prédiction pour trois cartes.

Cette capacité de prédiction est peu précise, mais il s'agit d'une prise de décision induite par l'auto-organisation des cartes. La formation de zones discrètes apparaît comme nécessaire à cette capacité prédictive. Par exemple, nous présentons en Figure 5.19 l'erreur de prédiction dans une architecture de 3 cartes sur un cercle 2D, mais dans lesquelles on a pris $r_c = r_e$. Ce jeu de paramètre ne fait pas apparaître de zones dans les poids contextuels. Nous voyons alors qu'aucune prédiction n'est réalisée. Dans ce sens, bien que les cartes se soient organisées, on ne peut pas parler d'un apprentissage du modèle par l'architecture.

Lorsqu'il manque des informations pour prédire correctement l'entrée, par exemple dans le cas du cercle en deux dimensions et d'une architecture de deux cartes, l'activation est cohérente. Si on ne présente pas l'entrée $X^{(2)}$ à la carte $M^{(2)}$, deux bulles d'activité se formeront aux deux emplacements possibles pour $\Pi^{(2)}$, sans qu'une des deux options ne soit systématiquement privilégiée par

pas sûr qu'on pige.
deux pics d'activité $a(p)$

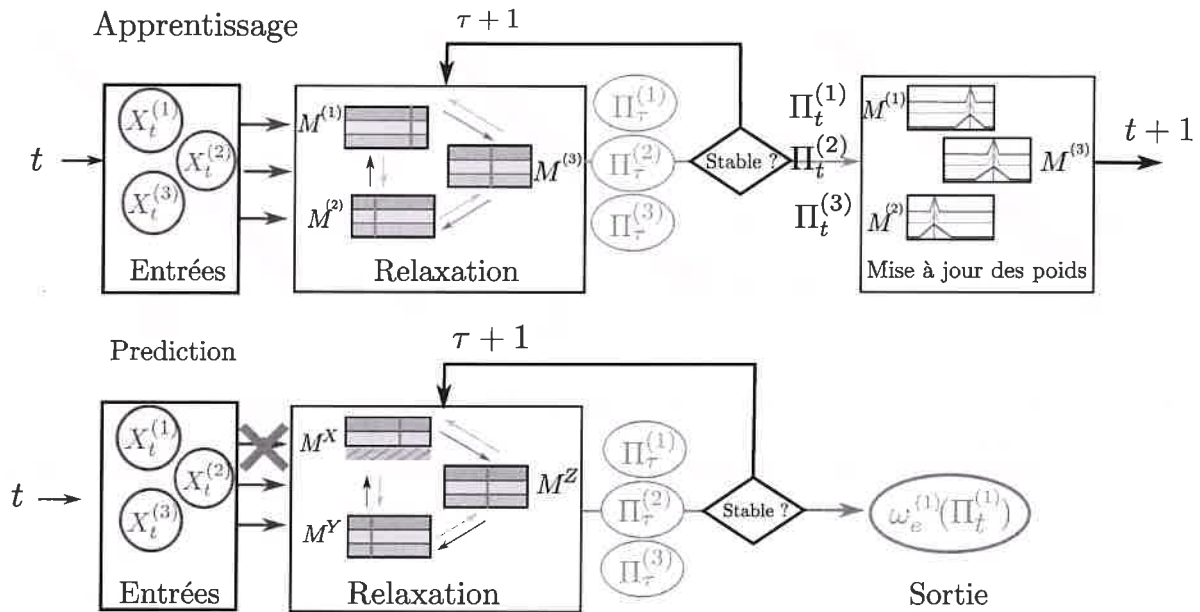


FIGURE 5.16 – Schéma descriptif des opérations effectuées lors de l'apprentissage et de la phase de prédiction.

l'architecture. *Les des exps y m' sa!*

5.4 Influence des connexions et des paramètres

Dans toutes les expériences précédentes, nous avons utilisé une architecture de cartes toutes connectées. Nous présentons maintenant quelques observations ouvrant des questions sur l'influence des connexions dans une architecture.

Nous nous intéressons tout d'abord à des grandes architectures de cartes. Nous avons vu qu'une architecture de deux cartes apprenant le carré $[0, 1]^2$ se déploie de manière à former deux indices. Nous pouvons effectuer la même expérience pour une architecture de K cartes apprenant sur tout

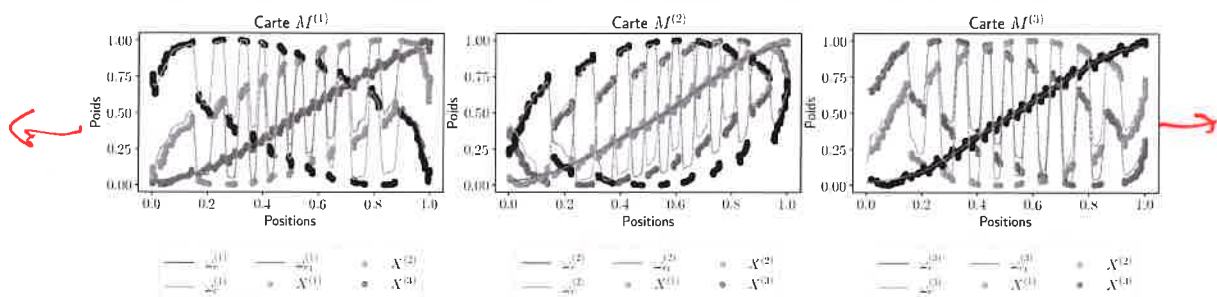


FIGURE 5.17 – Représentation cartographique des poids et entrées dans l'architecture de trois cartes apprenant sur un cercle en trois dimensions. Nous observons la formation de zones similaires au cas en deux dimensions.

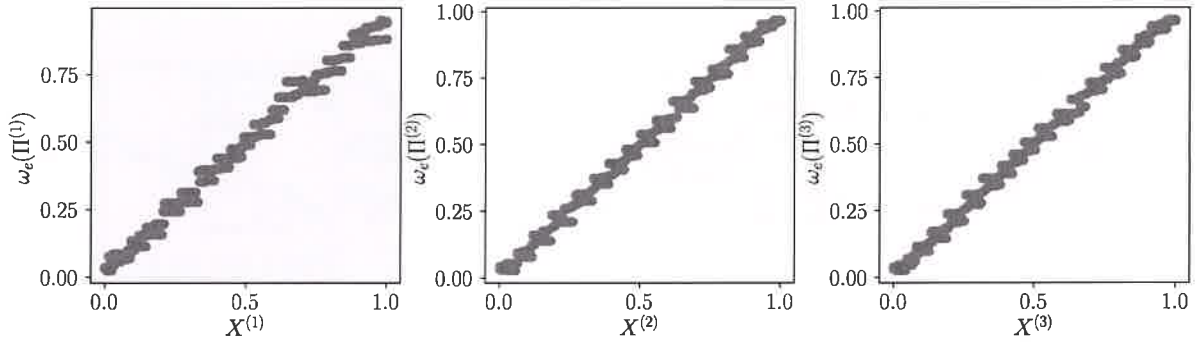


FIGURE 5.18 – Erreur de prédiction de $X^{(1)}$ par $\omega_c(\Pi^{(1)})$ lorsque les entrées sont sur un cercle en trois dimensions. $X^{(1)}$ n'a pas été présenté à $M^{(1)}$. Les nuages de points correspondant à $M^{(2)}$ et $M^{(3)}$ correspondent à l'erreur de quantification dans les cartes 2 et 3 qui ont reçu leur entrée externe. Ces tracés montrent une bonne prédiction de $X^{(1)}$ par la carte 1. *sens?*

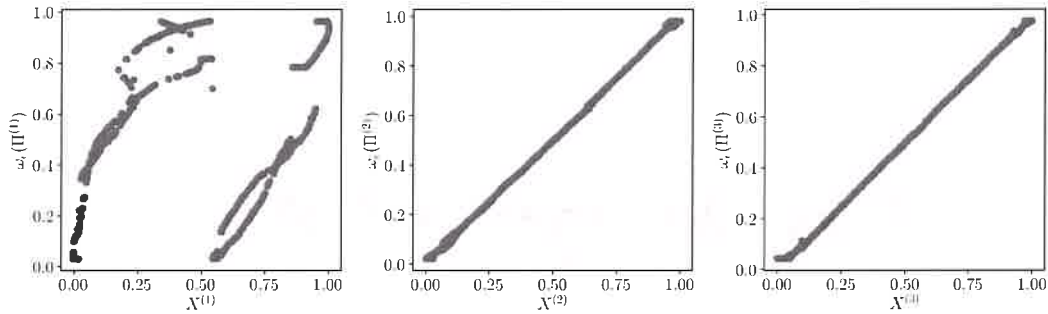


FIGURE 5.19 – Prédiction de l'entrée $X^{(1)}$ lorsque $r_c = r_e$. La prédiction n'est pas effectuée. Ainsi, sans formation de zones, la capacité de prise de décision n'est plus réalisable par une carte de l'architecture.

l'espace $[0, 1]^K$, toutes connectées. La figure 5.21 présente la forme des poids d'une telle disposition, sur une architecture de 10 cartes toutes connectées. Les entrées sont tirées dans le cube $[0, 1]^9$, et les entrées $X^{(9)}$ et $X^{(10)}$ sont identiques. Nous remarquons que les poids contextuels tendent vers une valeur moyenne de 0.5 dans chaque carte lorsque les entrées sont indépendantes. L'architecture équivaut donc à 9 cartes indépendantes, ce qui est logique par rapport à la disposition des entrées qui sont ici toutes indépendantes. Par contre, les poids contextuels correspondant aux entrées identiques se déplient totalement, comme en figure 5.8. La prédiction de l'entrée 9 est bien réalisée lorsque la carte ne reçoit pas l'entrée. Ainsi, il semble que les connexions correspondant à des entrées indépendantes finissent par se moyenniser pour ne plus jouer dans l'apprentissage des connexions correspondant à des entrées dépendantes et ne polluent donc pas l'apprentissage des relations. Cette expérience mérite d'être étudiée plus en détail pour plus de types de connexions entre entrées. En particulier, on peut se demander si le comportement de l'architecture sur des entrées en grande dimension permet d'extraire automatiquement des relations entre entrées, ce qui serait un comportement émergeant d'un grand nombre d'interactions.

Dans une deuxième expérience, nous reprenons en entrée le cercle tourné en trois dimensions (G). Nous comparons le comportement d'une architecture de trois cartes connectées réciproque-

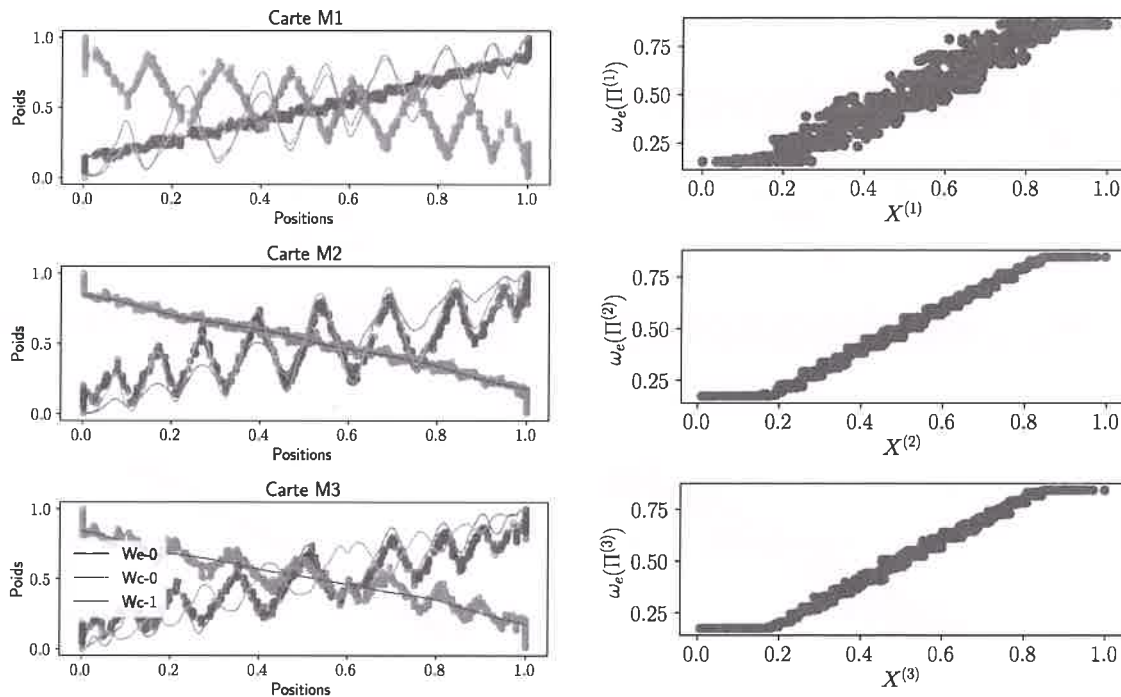


FIGURE 5.20 – Représentation cartographique des poids et entrées des trois cartes après apprentissage d'un plan pivoté en 3D et erreur de prédiction. Le découpage de l'espace par les cartes permet de prédire correctement une entrée. La précision faible de la quantification au niveau des zones explique l'erreur de prédiction moins précise dans ce cas.

ment, présenté plus haut, à celui d'une architecture de trois cartes connectées en boucle : $M^{(1)}$ est connectée à $M^{(2)}$, connectée à $M^{(3)}$, connectée à $M^{(1)}$. La phase d'apprentissage forme des motifs similaires à ceux observés sur la disposition avec rétroactions. Les poids contextuels forment des zones et U est une fonction du BMU dans chaque carte. Cependant, lorsque nous effectuons la phase de prédiction, la carte $M^{(1)}$ ne reçoit plus d'entrée externe et son activité contextuelle, relative à une seule couche de poids, dépend donc seulement de $M^{(3)}$. La carte $M^{(2)}$ intervient seulement via l'activité de cette dernière carte. Nous observons que la prédiction est moins précise que dans l'architecture avec rétroaction, mais est globalement réalisée. Des connexions distantes influencent donc une carte de manière moindre. L'influence des connexions sera à réaliser pour un passage sur une grande architecture.

fig?
ah si,
22.
Cite-la!

5.5 Prédiction d'entrée réelles : application au contrôle d'un drone

Nous avons vu une possibilité d'application d'une architecture CxSOM dans une tâche de prédiction d'entrée. Nous sortons du cadre des entrées simulées pour nous placer dans un cas de contrôle réel. 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 de ces capteurs peut être considéré comme une modalité d'un espace multimodal. À ces modalités s'ajoute une modalité correspondant à la commande envoyée au drone à chaque instant. La commande est ici une valeur d'accélération angulaire autour de chaque axe de rotation du drone en vol.

non.

font des cartes univariées

> font des dessins.



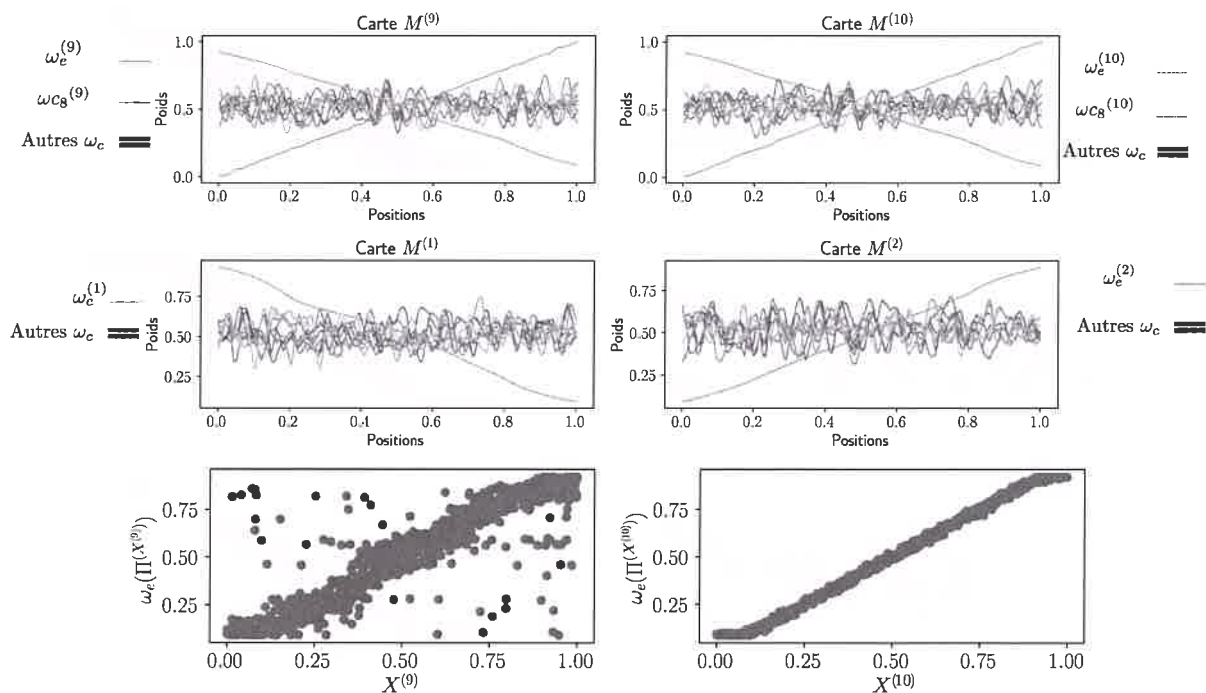


FIGURE 5.21 – En haut, tracé des poids des cartes pour une architecture de 10 cartes toutes connectées. Nous avons ici seulement tracé 4 cartes sur les 10. Les entrées sont indépendantes, sauf les entrées $X^{(9)}$ et $X^{(10)}$ qui sont identiques. Nous remarquons que les poids contextuels se moyennent 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 polluent pas l'apprentissage.

Le principe de cette expérience est d'apprendre, à l'aide d'une architecture de cartes, les relations existant entre les modalités des capteurs et de la commande. Dans un deuxième temps, nous utiliserons les poids appris afin prédire la commande à envoyer à partir des valeurs des capteurs, en temps réel sur une trajectoire du drone. Afin que les relations entre la commande et les capteurs soient significatives, nous nous plaçons dans un cas d'application particulier : le drone vole dans un couloir étroit, en ligne droite. Le but du drone est alors de voler en avant dans le couloir, sans toucher les murs en corrigeant sa trajectoire si besoin.

Lors des expériences géométriques, les données étaient peu bruitées et nous nous étions assurés que chaque modalité contribuait à l'apprentissage du modèle. Dans cette application, les relations entre entrées sont simples, mais les données sont très bruitées. Par ailleurs, certaines entrées ne sont pas du tout déterminantes pour le modèle et peuvent ainsi polluer l'apprentissage. Nous évaluerons grâce à cette expérience la robustesse de l'algorithme à des données bruitées et la capacité de CxSOM à réagir en temps réel malgré les étapes de relaxation.

5.5.1 Méthode expérimentale

Le drone utilisé pour l'expérience est un quadricoptère. Il possède une caméra frontale. Nous le contrôlons à distance par un ordinateur ; la commande est réalisée en envoyant l'accélération angulaire du drone autour de ses trois axes de rotation. Nous avons accès aux données des capteurs

redite - rose bien 6 pb... 1 fois!

ou c'est trop bavarde

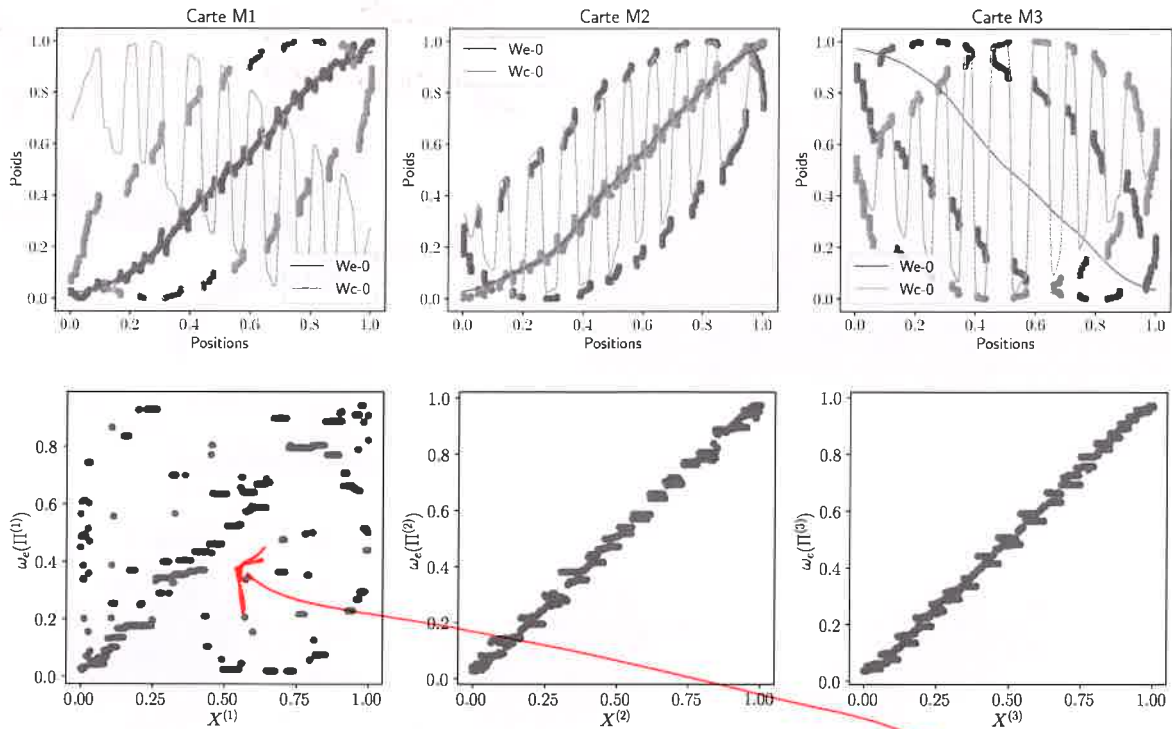


FIGURE 5.22 – Poids à l'issue de l'apprentissage et erreur de prédiction de $X^{(1)}$ dans une architecture de 3 cartes connectées en boucle. Bien que les poids s'organisent de façon similaire aux expériences dans lesquelles les cartes sont connectées réciproquement, la prédiction est moins bien réalisée. L'influence des connexions diminue donc rapidement avec la distance dans l'architecture, mais reste présente.

internes, notamment la vitesse linéaire courante selon chaque axe de déplacement. Le drone se déplace dans un couloir étroit en ligne droite, à hauteur constante et vitesse constante.

Dans le cadre de l'expérience, nous extrayons deux éléments visuels spécifiques au couloir à partir de la caméra du drone : l'abscisse du point de fuite du couloir x et la différence entre les angles des lignes du couloir, notée φ . Ces valeurs sont illustrées en figure 5.23.

La commande générant le déplacement en avant (tangage) est maintenue constante. Les commandes permettant le déplacement en largeur sont alors ω et ρ . Dans le cadre de cette expérience, nous contrôlerons uniquement ρ . Enfin, la vitesse linéaire en largeur du drone est récupérée à chaque instant ; nous la notons v . Nous utilisons ainsi quatre modalités lors du déplacement du drone : x , φ , ρ et v . Nous construisons une architecture CxSOM sur ces quatre modalités, composée de quatre cartes connectées chacune aux trois autres.

Une phase d'apprentissage est réalisée sur des déplacements du drone contrôlés humainement. 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 pour que les données soient présentées aléatoirement lors de l'apprentissage. Nous normalisons les entrées entre 0 et 1 pour favoriser l'apprentissage des cartes. Après apprentissage, nous effectuons une phase de prédiction. Lors de cette étape, la commande ρ n'est plus présentée à la carte correspondante. Nous envoyons alors le

ben oui, c'est
mal dit car c'est pas bien
posé la description.

pas vraiment...
c'est la vitesse qui l'est.

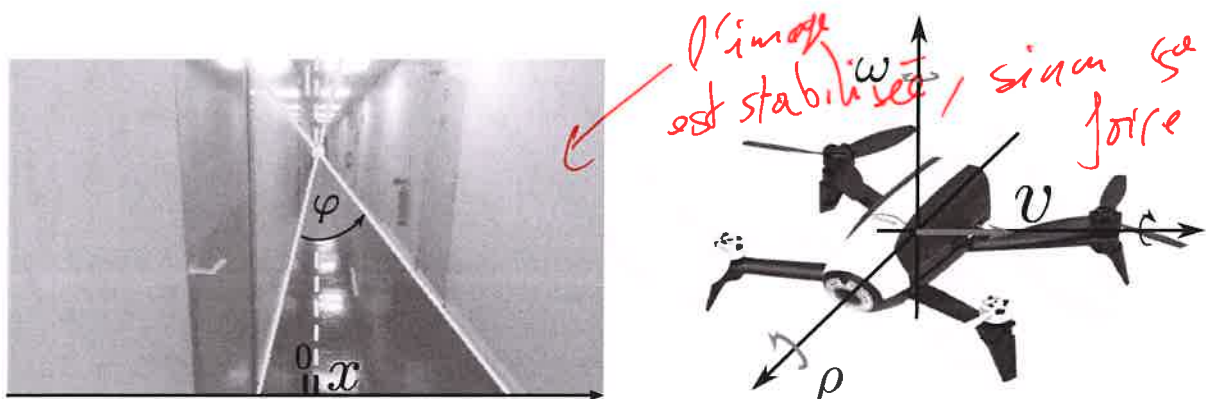


FIGURE 5.23 – Disposition des capteurs utilisés pour l'expérience

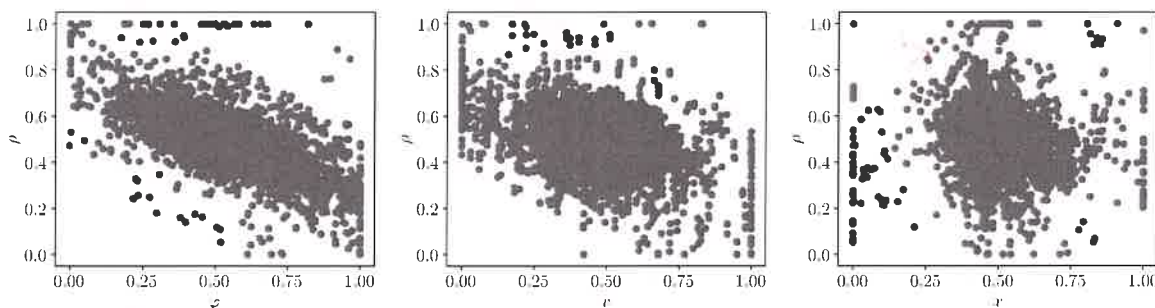


FIGURE 5.24 – Disposition et dépendances des entrées d'apprentissage. Nous chercherons à prédire ρ : cette valeur dépend bien des autres modalités v , φ et x . La dépendance est très simple (linéaire) mais très bruitée.

poids externe du BMU de cette carte comme commande du drone. Cette étape est réalisée en ligne en temps réel sur la trajectoire du drone.

La figure 5.24 présente la répartition des entrées présentées au drone. Nous avons tracé les dépendances entre chaque modalité. Nous nous intéressons ici aux dépendances entre la commande en y ρ , sujet de la prédiction 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 . Par contre, les entrées correspondant à l'abscisse du point de fuite varient peu, et la commande ρ dépend peu de x .

Le contrôle du robot est réalisé à distance via le logiciel ROS. Les données sont récupérées et envoyées au robot à une fréquence de **20Hz** ?? La vitesse interne v est seulement obtenue à une fréquence de 5Hz. Pour l'apprentissage, nous complétons la valeur de la vitesse interne dans les échantillons d'entrées en considérant sa dernière valeur prise. Elle évolue donc par paliers.

5.5.2 Résultats

La carte associée à ρ possède une couche de poids externe et trois couches de poids contextuels. Ces poids sont représentés en figure 5.25.

Nous observons d'abord que l'organisation des poids contextuels rappelle celle observé dans des conditions géométriques : les poids contextuels définissent des zones. Ces zones sont moins précisément définies que la configuration sur un cercle, certainement à cause de la grande variabilité des données : le modèle d'entrée est très bruité.

A chaque pas de temps, les valeurs v , x et φ sont récupérées ; le BMU est cherché par une phase de relaxation, puis la prédiction de la commande ρ , le poids externe du BMU dans la carte correspondante, est envoyé comme commande au drone. La figure 5.25 présente ainsi en violet la valeur de l'activité contextuelle de la carte ρ , utilisée comme activité globale, lorsque la relaxation est terminée. Le BMU est pris au maximum de l'activité, $\Pi = 0.41$. Cela correspond à une valeur de commande $\omega_e(\Pi^{(\rho)}) = 0.49$. Cette valeur est remise à l'échelle et envoyée au drone.

L'expérience en temps réel montre que l'architecture de cartes réagit correctement aux entrées et envoie une commande cohérente au drone. Le drone apparaît voler correctement dans le couloir sans toucher les murs. Nous observons cependant une imprécision sur la trajectoire. La prédiction est donc correctement réalisée, toutefois assez imprécise. Cependant, le fait que les entrées soient très bruitées peut aussi expliquer le manque de précision. En figure 5.26, nous avons représenté l'erreur de prédiction de l'architecture en effectuant une phase de test sur les mêmes données d'entraînement afin d'illustrer la qualité de la prédiction. Cette figure confirme que la prédiction est correctement effectuée mais très bruitée.

Cette capacité de prédiction montre une possibilité d'application des architectures de cartes sur des données réelles. Nous avons utilisé ici une architecture de quatre cartes, contrairement aux parties précédentes où nous avons seulement utilisé des architectures de deux et trois cartes. L'apprentissage a été effectué seulement sur 4890 points, présentés une seule fois, contrairement aux expériences géométriques dans lesquelles nous avons laissé aux cartes de nombreuses itérations pour s'organiser. Les résultats des expériences montrent donc que les comportements des architectures de deux et trois cartes sont encore observés sur 4 cartes, apprenant sur des données très bruitées. Peu d'itérations sont nécessaires en pratique pour déplier des cartes 1D. Rappelons que les poids externes et contextuels sont appris en une seule et même étape. Enfin, la réactivité de l'envoi de la commande au drone montre donc que le nombre de pas de relaxation est assez faible et donc que la relaxation permet une réponse en temps réel, sans chercher une réactivité instantanée. Ces observations, bien que sur un cas simple d'entrées en une dimension laissent la porte ouverte à une application possible des architectures CxSOM en pratique, par exemple en robotique.

Du travail supplémentaire est cependant nécessaire pour une réelle application de l'architecture. Cette expérience se place comme une première illustration du modèle dans un cadre réel. Tout d'abord, 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 réel. Plus généralement, les entrées d'une application robotique sont en pratique en plus grande dimension, par exemple une représentation d'une image. Il sera nécessaire d'étudier l'organisation des cartes sur ce type de données, en utilisant dans ce cadre des cartes en deux dimensions.

5.6 Conclusion

Dans ce chapitre, nous avons d'abord observé que l'apprentissage du modèle d'entrée dans une architecture CxSOM passe par la formation de zones de poids contextuels créant une seconde échelle d'indices et donc de quantification vectorielle au sein de chaque carte. Ces zones ont été observées

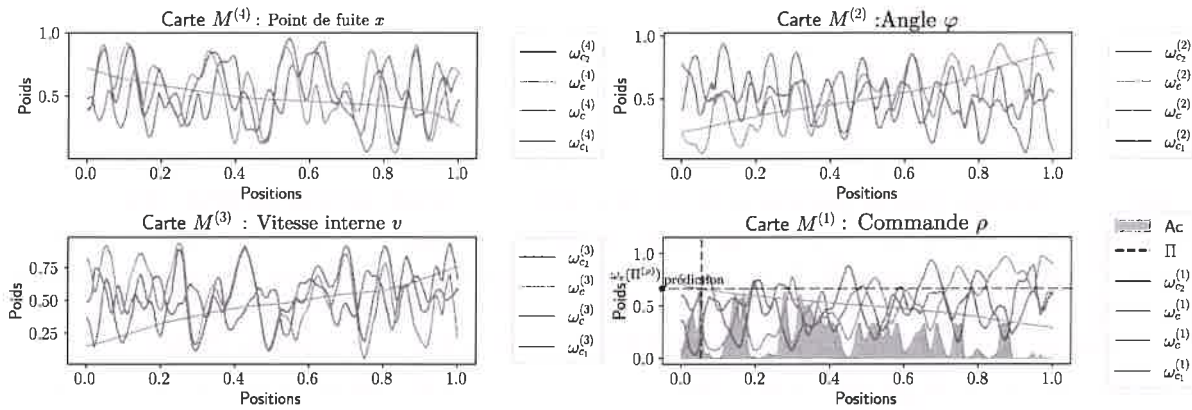


FIGURE 5.25 – Disposition des poids des 4 cartes après apprentissage.

2 dans chaque carte sur toutes les dispositions d'entrées en deux et trois dimensions étudiées, dès lors que deux points différents du modèle d'entrées correspondent à la même entrée externe d'une carte. Ces zones de poids contextuels encodent l'information sur le modèle U dans chaque carte. Elles permettent à une carte de l'architecture d'acquiescer un comportement prédictif en l'absence de son entrée externe. Une carte ne recevant pas d'entrée externe possède quand même un BMU grâce à ses activités contextuelles. Nous avons montré que le poids externe de ce BMU correspond bien à la valeur de l'entrée externe manquante. Cette prédiction est directement due à la présence des zones poids contextuelles. Cette capacité de prédiction, avant d'être une application possible des cartes, permet d'évaluer l'apprentissage du modèle d'entrée par l'architecture.

Cette étude se veut l'identification de comportements élémentaires de cartes, dans le but de construire des architectures comportant plus de cartes. Dans de telles architectures, la disposition des connexions est un degré de liberté supplémentaire pour la conception de l'architecture. Nous avons identifié sur l'exemple des cartes connectées en boucle que des connexions distantes dans l'architecture jouent bien un rôle dans l'activité d'une carte, tout en perdant leur influence. Par ailleurs, les observations réalisées sur le drone et sur l'architecture de 10 cartes montrent qu'une entrée ne participant pas à la définition du modèle ne vient pas polluer l'apprentissage. Au contraire, il semble même que les connexions de l'architecture reliée à la carte prenant cette entrée inutile apprennent à effacer leur participation au calcul de l'activité des autres cartes. Une étude plus générale de l'influence des connexions permettra d'ajouter cette dimension à la conception de systèmes CxSOM.

Une perspective d'étude nécessaire est d'évaluer la distribution de l'encodage du modèle d'entrée dans les cartes. Dans les expériences présentées dans ce chapitre, plus globalement dans cette thèse, le modèle U est une variable 1D. Nous avons montré que l'apprentissage se traduit par le fait que U est une fonction de Π dans chaque carte. Dans une grande architecture, donc des entrées multimodales de dimension totale supérieure, on ne peut pas attendre que chaque carte encode complètement 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 pour U de dimension 10. 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. Cet aspect distribué de l'apprentissage est un point à étudier 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

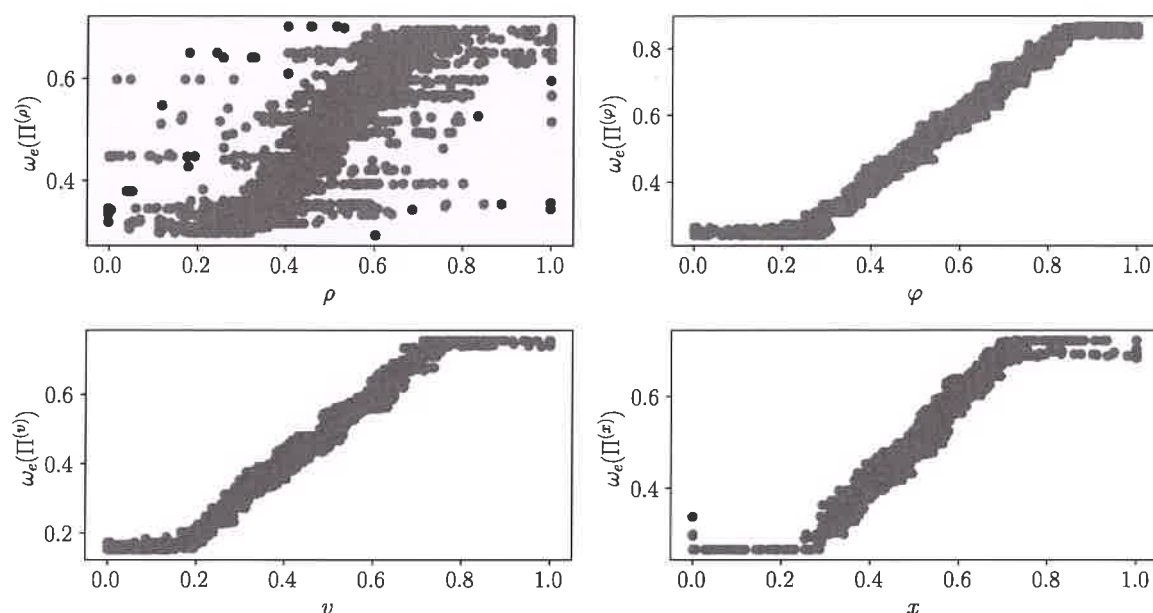


FIGURE 5.26 – Prédiction des valeurs de la commande ρ du drone sur des données de test. La prédiction, bruitée, est correctement réalisée par rapport aux relations initiales existant entre les entrées.

ou le calcul d'activité. Par exemple, nous avons toujours pris la même valeur de r_c pour toutes les couches de poids contextuels dans toutes les cartes. Ces paramètres et leur disposition constituent de nombreux degrés de libertés supplémentaires dans la construction d'une architecture et sont des pistes possibles d'étude et d'adaptation de l'architecture.

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 pertinent d'utiliser des cartes en deux dimensions. Nous chercherons donc également 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 à la fin de ce manuscrit des expériences préliminaires étudiant l'organisation des poids dans une architecture de cartes en deux dimensions.