



# A propos des architectures de cartes auto-organisatrices stylées

## THÈSE

présentée et soutenue publiquement le le plus tard possible en 2022

pour l'obtention du

**Doctorat CentraleSupélec**

(mention informatique)

par

Noémie Gonnier

### Composition du jury

<i>Président :</i>	Le président	du jury
<i>Rapporteurs :</i>	Le rapporteur 1	du laboratoire
	Le rapporteur 2	
	Le rapporteur 3	
<i>Examineurs :</i>	L'examineur 1	
	L'examineur 2	

---

Laboratoire Lorrain de Recherche en Informatique et ses Applications





# Sommaire

<b>1</b>	<b>Approche modulaire des réseaux de neurones</b>	<b>1</b>
1.1	La modularité, répandue dans les système biologiques . . . . .	1
1.1.1	Systèmes complexes -> Mettre dans introduction ? . . . . .	1
1.1.2	Le cerveau, réseau modulaire fondamental . . . . .	2
1.1.3	Des réseaux et des modules dans tous les systèmes biologiques . . . . .	3
1.2	Quelle définition de la modularité ? . . . . .	3
1.2.1	Modularité structurelle . . . . .	3
1.2.2	Modularité fonctionnelle . . . . .	4
1.2.3	Modularité temporelle ? . . . . .	4
1.3	Intérêt computationnel des réseaux modulaires . . . . .	4
1.3.1	Types de connexions . . . . .	4
1.3.2	Modularité et émergence d'un apprentissage ? . . . . .	4
1.4	Réseaux de neurones modulaires . . . . .	4
1.4.1	Deep Learning . . . . .	4
1.4.2	Réseaux auto-organisés . . . . .	5
1.4.3	Construire une architecture modulaire . . . . .	6
1.5	Enjeux d'une architecture modulaire de SOMs . . . . .	6
<b>2</b>	<b>Un réseau de neurones auto-organisé : les cartes de kohonen</b>	<b>7</b>
2.1	Principe Général . . . . .	7
2.2	Approche topologique des cartes de Kohonen . . . . .	7
2.3	Principes d'organisation . . . . .	7
2.4	Travaux préliminaires sur les architectures de cartes de Kohonen . . . . .	7
2.5	A trier . . . . .	7
<b>3</b>	<b>Construction d'une architecture modulaire de cartes auto-organisatrices</b>	<b>9</b>
3.1	Description de l'algorithme . . . . .	9
3.1.1	Carte de Kohonen classique . . . . .	9
3.1.2	Modèle : CxSOM . . . . .	11
3.2	Choix des paramètres . . . . .	13
3.2.1	Influence des rayons de voisinage . . . . .	13
3.2.2	Influence des autres paramètres . . . . .	13
3.2.3	Compatibilité en 2D . . . . .	13

3.3	Analyse de la relaxation . . . . .	13
3.3.1	Analyse expérimentale . . . . .	13
3.3.2	Champs de BMU . . . . .	13
3.3.3	Limitations et possibilités en grande dimension . . . . .	13
3.4	Implémentation . . . . .	13
3.5	Perspectives d'évolutions . . . . .	14
<b>4</b>	<b>Analyser l'organisation : une approche par variables aléatoires</b>	<b>15</b>
4.1	Cas d'utilisation : les entrées multimodales . . . . .	15
4.1.1	Définition et inspiration biologique . . . . .	15
4.1.2	Formalisme . . . . .	15
4.1.3	Perspectives . . . . .	15
4.2	Représentation des entrées . . . . .	15
4.3	Information apprise par une carte . . . . .	15
4.4	Représenter une carte au sein d'une architecture . . . . .	16
4.5	Prédiction d'entrée . . . . .	16
	<b>Bibliographie</b>	<b>19</b>

## Introduction

Cette thèse propose une construction d'une architecture modulaire

# Chapitre 1

## Approche modulaire des réseaux de neurones

De nombreux modèles biologiques présentent des architectures modulaires. Notre conception du monde se présente en fait sous une forme de modularité. Présentation d'exemples de théories dérivées d'une modularité bio-inspirée. Brooks [3]

Réseaux associés aux systèmes complexes, interactions

The mind, the brain and complex adaptive systems : how a mind resides in a brain (1962) [?]. Propose la "near decomposability"

En tant qu'humain, nous comprenons le monde de notre point de vue, en le décomposant pour qu'il semble accessible : en effet notre raisonnement est modulaire, notre système social, groupes d'individus, etc. Nos constructions sont modulaires : programme informatique ... Il est difficile de concevoir et surtout de comprendre, en tant qu'individu un système qui ne serait pas décomposable en modules. Prenons comme exemple les réseaux de neurones profonds : la compréhension et l'interprétabilité ces programmes est un défi de la recherche actuelle. Pour cette interprétation, on cherche des éléments symboliques, des groupes, des communautés. La décomposition des sciences elles même, par exemple, nous permet de trouver des solutions aux problèmes à des échelles différentes. Le programmeur n'a pas besoin de comprendre en profondeur quels transistors composent les circuits ; l'expert.e en électronique n'a pas besoin de d'abord résoudre les équations qui régissent les mouvements ioniques au sein des transistors pour concevoir des circuits, etc. Seuls les principes régissant les comportement globaux d'un système comme le transistors ont besoins d'être connus pour utiliser ces systèmes dans une tâche ; tâche qui consistera également un module dans son ensemble et qui sera régie par des principes généraux, du transistor à l'utilisation d'un logiciel de dessin graphique. Mais est ce que cette hiérarchie modulaire est essentiellement subjective ? A priori non. Une organisation modulaire est présente et calculable dans de nombreux systèmes.

### 1.1 La modularité, répandue dans les système biologiques

#### 1.1.1 Systèmes complexes -> Mettre dans introduction ?

Les systèmes, dans leur échelles, sont donc complexes. Si on peut approximer certains systèmes pour les étudier, ils restent liés au sein d'un vaste écosystème. Ainsi, l'étude de la modularité des systèmes est présente au sein des approches biologiques.

"Un système complexe est un ensemble constitué d'un grand nombre d'entités en interaction dont l'intégration permet d'achever une mission commune. Les systèmes complexes sont caractérisés par des propriétés émergentes qui n'existent qu'au niveau du système et ne peuvent pas être observées au niveau de ces constituants. (wikipedia)" -> Exemple fourmilière, L'Emergence de patterns et de phénomènes d'auto-organisation (turing patterns) [14] qui sont même utilisés pour qualifier le système complexe.

Il n'y a a priori pas de définition commune concernant ce qu'on appelle un système complexe et comment le mesurer. On pourrait plutôt voir la complexité comme une façon d'étudier un objet.

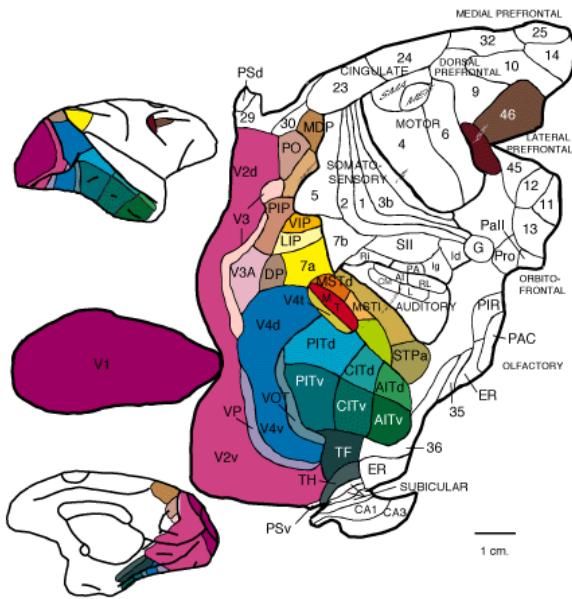


FIGURE 1.1 – carte aplanée des aires du cortex visuel du macaque [5]

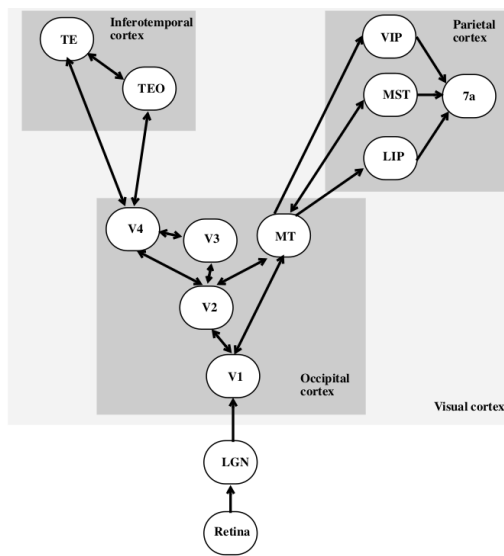


FIGURE 1.2 – pathways dans le cortex visuel du macaque [13]

Exemple de définition : complexité de Kolmogorov qui évalue la complexité par la taille du plus petit module permettant d'engendrer l'objet ; une façon de définir la complexité parmi d'autres. Autre exemples : ??

Complexe ne veut pas dire relation et interaction difficiles à exprimer : un système est complexe du point de vue d'un observateur, mais les règles locales sont souvent simples.

Systèmes liés par des équations d'évolution et des conditions initiales.

Qui dit système complexe dit système modulaire ?

### 1.1.2 Le cerveau, réseau modulaire fondamental

L'étude des réseaux biologiques, notamment, a fait émerger l'idée d'architectures modulaires. Et si on rajoute l'idée d'apprentissage, l'étude des cerveaux est une mine d'information et de questionnement quant aux structures et propriétés le permettant. Les premières propositions d'études du cerveau humain datent du début du XXème siècle. Déjà, un découpage en aires est proposé pour expliquer le fonctionnement de cet organe, notamment avec les travaux de Broca et Wernicke qui mettent en lumière des zones du cerveau qui semblent responsables du langage. Le modèle connexionniste du cerveau, formalisé à partir de ces travaux par Geschwind dans les années 60, décompose ainsi le langage en plusieurs fonctions : la compréhension, la lecture et l'action de parler. Ce modèle n'est plus vraiment utilisé, mais la décomposition. Avec l'avènement des outils d'imagerie, le cerveau a pu être cartographié plus précisément en un ensemble d'aires, agissant comme modules fonctionnels au sein d'une structure complexe ; ces aires sont elles mêmes composées de modules distincts, etc.

- rétroaction entre les modules [5], les pathways sont soit réciproque, soit non, à 50-50

Le cerveau est ainsi présenté en modèle et motivation principale lorsqu'il s'agit de construire des réseaux de neurones artificiels. - Nombreuses études s'appuient sur le système modulaire : le cerveau [5, 11, 2]

- Modularité structurelle : exemple du cortex visuel

- Connexions : hiérarchiques ou non, et feedback, ex boucle baso thalamo corticale, ou au sein des aires comme le cortex visuel.

- Modularité hiérarchique (modules de modules) -> Ajouter infos depuis l'article qui parle de modules hiérarchiques.

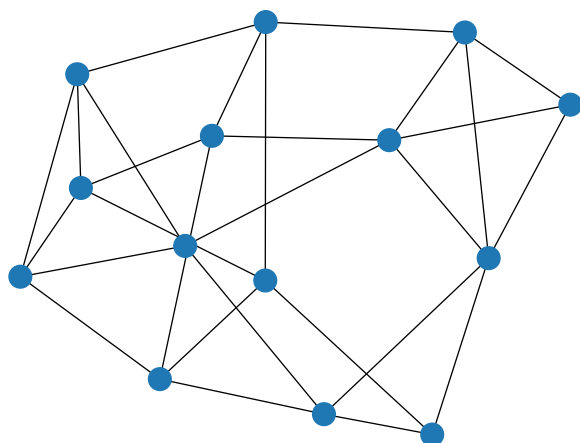


FIGURE 1.3 – Réseau aléatoire "petit-monde" créé avec l'algorithme de Watts-Strogatz

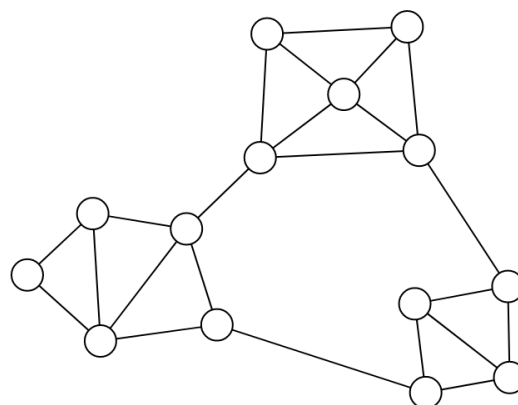


FIGURE 1.4 – Réseau modulaire : il possède une propriété de "petit monde" mais des modules, zones fortement connectées, sont identifiables.

TODO : que signifient pathways streams (vite fait) -> on a une co de 1 vers 2 si 2 est une activation post-synaptique de 1. Mais en moyenne ? associer aires des deux figures -> TEO juste à côté de TF et TH, les autres ok.

Finalement : Cerveau présente ce qu'on peut appeler des modules, avec des connexions entre eux, avec rétroaction.

Leur mesure peut être structurelle (regarder les structures de neurones), ou fonctionnelle : quelles aires s'activent pour une fonction ?

Détail des types de modularité en parties suivantes.

### 1.1.3 Des réseaux et des modules dans tous les systèmes biologiques

## 1.2 Quelle définition de la modularité ?

Il s'agit de tirer une définition claire de ce qu'on appelle modularité. Entre l'étude des systèmes biologiques et l'ingénierie des réseaux de neurones, l'aspect modulaire des réseaux et plus généralement des systèmes peut se définir de plusieurs manières.

### 1.2.1 Modularité structurelle

Lorsque le système possède une structure définie de réseau, typiquement un réseau de neurones, on peut définir une modularité en terme de graphe.

#### Réseaux en "petit-monde" et réseaux sans échelle

Definition : type de graphe dans lequel la distance moyenne entre deux noeuds est proportionnelle à  $\log(N)$ ,  $N$  le nombre de noeuds. C'est à dire, un graphe dans lequel on trouvera un chemin assez court entre n'importe quels noeuds. Particularités : forme souvent des cliques (sous graphes fortement connexes), donc une structure modulaire. Par contre, tous les réseaux small world ne sont pas nécessairement modulaire. Souvent admis que le cerveau en tant que réseau de neurones est small world.

#### Modularité auto-similaire

Exemples ?? [10] Modularité hiérarchique présente l'avantage de maintenir une activité dans le réseau sans que ça ne colonise tout ni ne s'éteigne, ce qui est nécessaire pour la computation.

**Mesurer la modularité d'un réseau**

clustering coefficient proximity ratio scale free networks

**Def**

[6] : différencie réseaux small world et réseaux hiérarchique modulaires, et statue qu'un réseau peut être "hiérarchique modulaire" même s'il n'est pas small world, mais possède une dimension topologique finie. [10] : statue que un réseau modulaire est small world,

**1.2.2 Modularité fonctionnelle****1.2.3 Modularité temporelle ?**

La modularité s'associe aux séquences dans le cerveau. On peut donc raisonnablement définir

**1.3 Intérêt computationnel des réseaux modulaires**

Barabasi : Hypothèse que les réseaux small world présentent un avantage évolutif.

**réponse à un problème de contrainte physiques, énergétiques**

- Parallélisme, calcul et small world networks : réponse à un problème de contrainte physiques, énergétiques. - Calcul distribué - Automates cellulaires ?

**1.3.1 Types de connexions**

Opaque vs tout savoir

**1.3.2 Modularité et émergence d'un apprentissage ?**

La modularité est liée à la complexité des systèmes, donc l'émergence de comportements chaotiques et/ou synchronisés. SYSTEMATIC GENERALIZATION : WHAT IS REQUIRED AND CAN IT BE LEARNED ? : Our findings show that the generalization of modular models is much more systematic and that it is highly sensitive to the module layout, i.e. to how exactly the modules are connected.

Simplicité de la modularité : exemple de construction des fractales, exemple du rigaudon

**1.4 Réseaux de neurones modulaires**

- Challenges et intérêt potentiel des réseaux de neurones modulaires ? -> Evolution (Wertner, Meunier) : pousser plus loin pour être au jus sur les challenges actuels (voir du côté du deep : quelles sont les motivations et les challenges ?)

**1.4.1 Deep Learning**

Boîtes noires qui ont des performances remarquables sur tous les domaines, leur représentation et compréhension est quant à elle toujours un challenge.

**Il existe des réseaux modulaires**

- Réseaux qui apprennent à s'organiser en modules. Intérêt. Limites ? Performances ? [1, 8] "The NMN approach is intuitively appealing but its widespread adoption has been hindered by the large amount of domain knowledge that is required to decide (Andreas et al., 2016) or predict (Johnson et al., 2017; Hu et al., 2017) how the modules should be created (parametrization) and how they should be connected



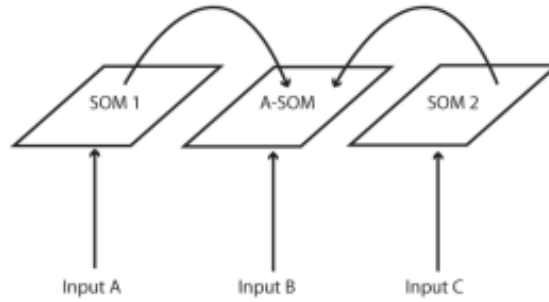


FIGURE 1.5 – A-SOM model [7]

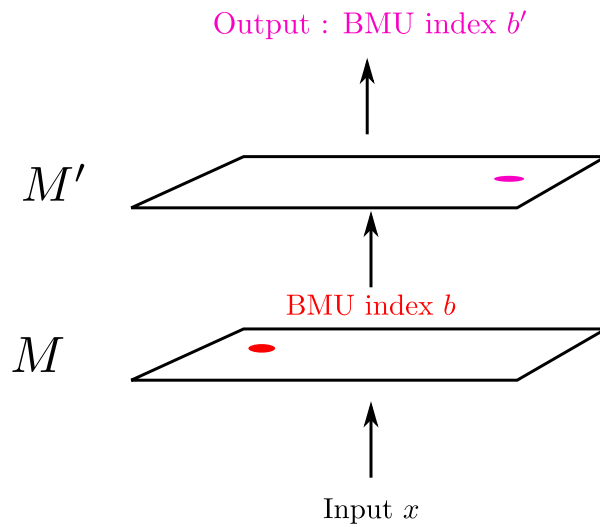


FIGURE 1.6 – HSOM model [9]

(layout) based on a natural language utterance. Besides, their performance has often been matched by more traditional neural models" ( systematic generalization article )

#### Utiliser des modules pour mieux représenter les réseaux de neurones profond

- Trouver des modules dans les réseaux pour les expliquer? [15, 4] are neural net modular : "it uses different modules for very different functions = Pspecialize," et "it uses the same module for identical functions that may have to be performed multiple times = Preuse" - Reconciling deep learning with symbolic artificial intelligence : representing objects and relations(2019) Pb du deep learning = Data inefficiency (comparé a l'humain) ;Poor generalisation ; Lack of interpretability.

#### 1.4.2 Réseaux auto-organisés

Les réseaux auto-organisés sont directement inspirés des réseaux biologiques. Au sein de ces réseaux Plus qu'en deep learning, les réseaux de neurones auto-organisés - Auto-organisation prend une profonde inspiration biologique, tout comme les modules. - Exemple de réseaux auto-organisés modulaires : développer dans la partie suivante.

[9] [12]

### 1.4.3 Construire une architecture modulaire

A mettre dedans :

- Réseaux top down / modulaires ? définition, à quel point un réseau est modulaire, qu'est ce qu'on appelle réseau modulaire ? Fonctions définies au préalable vs émergence des fonctions. Modules définis au préalable vs émergence des modules.

## 1.5 Enjeux d'une architecture modulaire de SOMs

On connaît plutôt bien les SOM, mais on sait que des comportements nouveaux peuvent émerger lorsqu'on met en interaction des systèmes étudiés séparément. On peut donc se poser la question des comportements qui peuvent survenir dans ce cas.

Dans les structures de cartes étudiées, modularité forte dans le sens où les fonctions des modules sont prédéfinies. Si on ne fournit que les règles d'interaction, où est-ce qu'on se situe ?

Position du manuscrit : architecture

Plan de la partie !

1. Intro : Notre monde est modulaire, en tout cas nous l'interprétons en tant que tel. Proposé déjà dans les années 60 - brève histoire des réseaux de neurones ? Remarquer que nous, observateur, raisonnons et nous construisons modulairement. Il nous est difficile de concevoir les choses autrement. Les disciplines scientifiques par ex, un domaine obéit à des principes
2. Qu'appelle-t-on la modularité ? Définitions claires et propriétés
  - Définition
    - (a) Structurelle, dans les systèmes réseaux
    - (b) Fonctionnelle, dans les systèmes dont on ne connaît pas la structure ?
    - (c) Temporelle / mais est-ce que le temps ce n'est qu'une dimension de plus
    - (d) Modularité hiérarchique attention : deux définitions à hiérarchiques. Soit c'est une histoire de connexions, soit d'auto-similarité. On parle nous de l'auto-similarité !!! Au contraire l'autre forme de hiérarchie n'est pas observée en bio.
  - Propriétés de la modularité
    - (a) Auto-organisation
    - (b) Types de connexions
    - (c) Émergence
3. Exemples Biologiques et computationnels
  - (a) Biologie : optimisation évolution a priori...
  - (b) Computationnel : pareil.
4. Maintenant qu'on sait ce que sont les systèmes modulaires, en quoi on peut faire de l'apprentissage avec ? Quels sont les intérêts ?
  - (a) Exemples de réseaux auto-organisés
  - (b) Exemples de réseaux de neurones : deep learning

## Chapitre 2

# Un réseau de neurones auto-organisé : les cartes de kohonen

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

### 2.1 Principe Général

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

### 2.2 Approche topologique des cartes de Kohonen

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

### 2.3 Principes d'organisation

### 2.4 Travaux préliminaires sur les architectures de cartes de Kohonen

### 2.5 A trier

De la biologie à la computation : patterns temporels des neurones impulsionnels vs SOM.



## Chapitre 3

# Construction d'une architecture modulaire de cartes auto-organisatrices

Le but de cette thèse est de proposer un modèle permettant d'associer des cartes auto-organisatrices dans n'importe quel type d'architecture, comme une sorte de brique de base. En particulier, on cherchera à construire des architectures non-hiéarchiques de cartes, exemple en figure 3.1. Nous nous plaçons donc dans le cadre de modules pré-établis, dont les entrées ont été connectées par avance. Les poids de chaque carte seront quant à eux appris, avec comme objectif que les cartes apprennent leurs entrées mais puissent également distinguer un état global de l'architecture. Pour les entrées, nous nous plaçons dans un cadre de multi-modalité, détaillé au chapitre suivant. Les différentes cartes prendront des données d'entrées sur différents espaces.

### 3.1 Description de l'algorithme

#### 3.1.1 Carte de Kohonen classique

Rappelons les notations concernant une carte de Kohonen standard. Prenons un ensemble de données d'entrées, dans lequel chaque élément est un vecteur d'un espace  $D$ . On a défini une distance  $d$  sur  $D$ , généralement la distance euclidienne. La carte de Kohonen construite sur ces entrées est un graphe, généralement une ligne 1D ou une grille 2D de  $N$  noeuds. Chaque noeud possède un poids associé  $\omega_e$  in  $D$  ou *prototype*, du même espace que les entrées. et une *position*  $i$  dans la carte. Ces positions sont ensuite indexées entre 0 et 1 par  $p = \frac{i}{N}$  pour l'homogénéité des calculs. L'ensemble des poids est noté  $\omega_e(p), p \in [0, 1]$ . L'algorithme se décompose de la façon suivante :

1. Une entrée  $\xi_t$  est présentée à la carte.

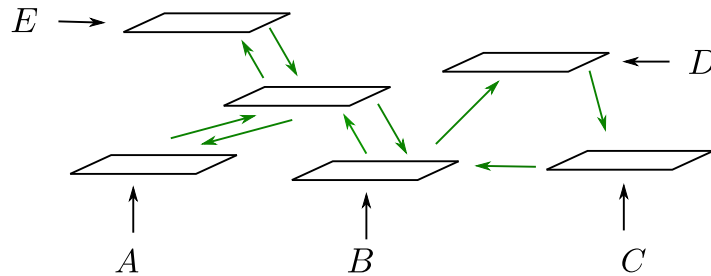


FIGURE 3.1 – Exemple d'architecture modulaire *non-hiéarchique* de cartes de Kohonen. Les entrées sont  $A, B, C, D, E$  quelconques. Chaque carte peut ou non prendre une entrée ; les connexions sont réciproques ou non.

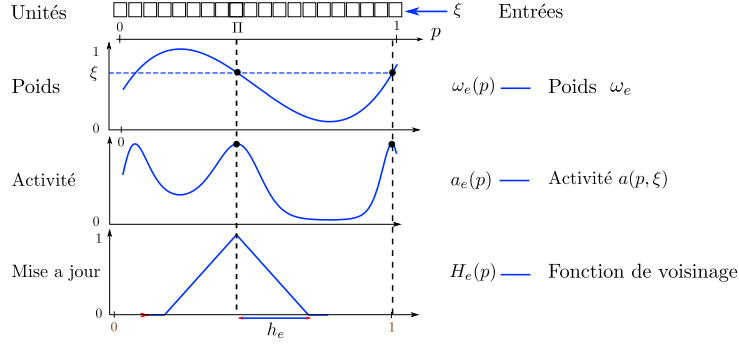


FIGURE 3.2 – Notations utilisées dans une carte de Kohonen simple

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

$$\omega_e(p, t + 1) = \omega_e(p, t) + \alpha h(\Pi, p)(\xi - \omega_e(p, t)) \quad (3.1)$$

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

Lors de l'étape 2 de l'algorithme, une activité peut être calculée, au lieu d'une distance pour choisir le BMU. Ce dernier est alors choisi comme  $\Pi = \arg \max_p (a(\xi, p))$ . Nous utiliserons cette solutions dans notre modèle. Les notations au sein d'une carte sont résumées en figure 3.2.

### 3.1.2 Modèle : CxSOM

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

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

La description du modèle CxSOM est détaillée en figure 3.4, dans un cas ou une carte reçoit deux connexions, et l'algorithme explicité en 1.

#### Gestion des entrées externes et contextuelles

A un pas d'apprentissage  $t$ , une carte  $M$  reçoit en entrée une entrée *externe* notée  $\xi_t$  et  $K$  entrées *contextuelles* notées  $\gamma_{0t}, \dots, \gamma_{Kt}$ , qui sont les BMU  $\Pi$  des cartes qui lui sont connectées. La carte possède donc  $k + 1$  couches de poids.  $\omega_e$  correspond à l'entrée externe et  $\omega_{c0}, \dots, \omega_{cK}$  aux entrées contextuelles.

**Input:**  $\xi^1, \dots, \xi^K \leftarrow (o^1, \dots, o^K) \in \mathcal{D}^1 \times \dots \times \mathcal{D}^K$   
 $t \leftarrow 0$   
 $\forall i, \Pi^i \leftarrow \arg \max_p a_e(\xi^i, p)$   
**while**  $\Pi(t) \neq \Pi(t-1)$  **do**  
  **forall**  $Map\ i$  **do**  
     $\gamma_1^i, \dots, \gamma_k^i \leftarrow$  BMUs from connected maps  
    Computation of  $a_g^i$  (equation ??)  
     $p^{*i} \leftarrow \arg \max_p a_g^i(p)$   
     $\Pi^i \leftarrow \Pi^i + \min(\Delta, |p^{*i} - \Pi^i|) \times \text{sgn}(p^{*i} - \Pi^i)$   
  **end**  
   $t \leftarrow t + 1$   
**end**  
 $\forall i, \omega_e^i(p) \leftarrow \omega_e^i(p) + H_e(\Pi^i, p)(\omega_e^i(p) - \xi^i)$   
 $\forall i, \forall k, \omega_{ck}^i(p) \leftarrow \omega_{ck}^i(p) + H_c(\Pi^i, p)(\omega_{ck}^i(p) - \gamma^i)$

**Algorithm 1:** Learning iteration with relaxation process

On calcule une activité séparément sur chaque couche de poids selon la formule suivante :

$$a(p, x) = \exp\left(-\frac{(\omega(p) - x)^2}{2\sigma^2}\right) \quad x = \xi_t \text{ ou } \gamma_{kt}, \quad \omega = \omega_e \text{ ou } \omega_{ck} \quad (3.2)$$

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

$$a_g(p, \xi_t, \gamma_t) = \sqrt{a_e(p, \xi_t)(\beta a_e(p, \xi_t) + (1 - \beta)a_c(p, \gamma_t))} \quad (3.3)$$

Cette activation globale est utilisée pour déterminer le BMU de la carte.

### Gestion des rétroactions dans l'architecture

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

Le processus de relaxation est donc une boucle imbriquée dans un pas d'apprentissage de l'architecture, indexée par  $\tau$ . Notons  $\Pi^i$  la position du BMU de la carte  $i$ , et  $\Pi = (\Pi^1, \dots, \Pi^n)$ , avec  $n$  le nombre de cartes de l'architecture. Au début d'un pas d'apprentissage, chaque carte est nourrie avec une entrée externe  $\xi_t^i$ , et les activités externes  $a_e^i(\xi_t^i, p)$  de chaque carte peuvent être calculées. La recherche du BMU suit donc le processus de relaxation suivant :

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

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

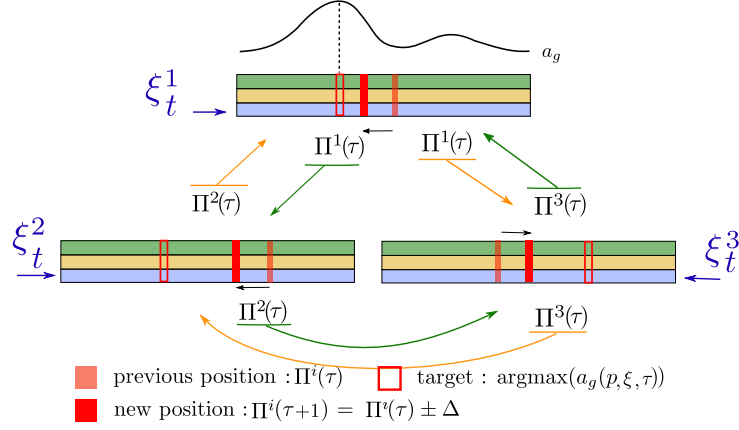


FIGURE 3.3 – description d’une étape de la relaxation dans l’architecture, aboutissant à un consensus entre cartes. Au sein d’une même itération  $t$ , les position des BMU  $\Pi$  sont légèrement déplacées jusqu’à ce que toutes les positions  $\Pi$  des cartes de l’architecture soient stable. Ces positions maximisent collectivement les activités globales de chaque carte.

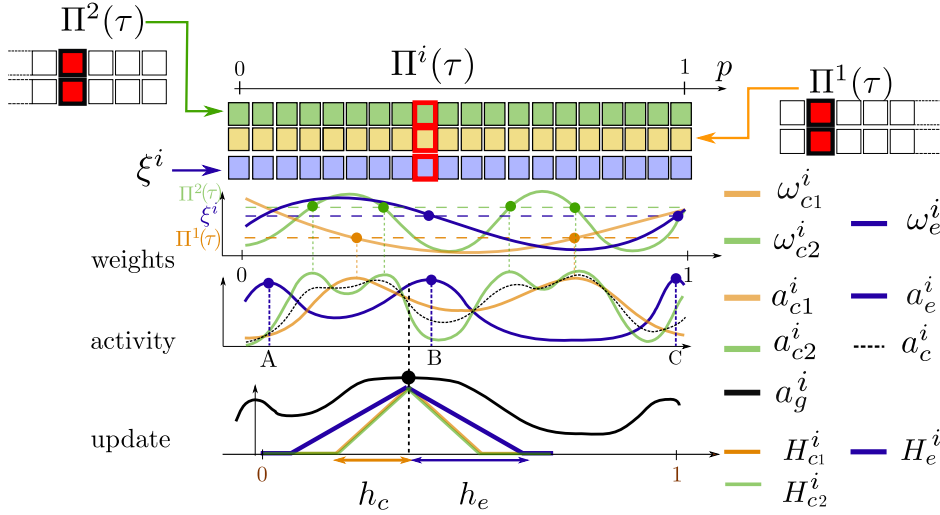


FIGURE 3.4 – Description d’une carte au sein d’une architecture CxSOM. La carte reçoit deux connexions de cartes voisines, et possède donc deux couches contextuelles



### Mise à jour des poids

Les poids sont mis à jour par rapport à leurs entrées respectives suivant l'équation 3.1. Le BMU d'une carte est ainsi commun à toutes les couches. Les rayons de voisinage  $h_e$  et  $h_c$  ont des valeurs différentes ; celles-ci seront détaillée en partie suivante.

### Tests

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

## 3.2 Choix des paramètres

### 3.2.1 Influence des rayons de voisinage

### 3.2.2 Influence des autres paramètres

### 3.2.3 Compatibilité en 2D

## 3.3 Analyse de la relaxation

L'apprentissage conjoint des cartes repose sur la relaxation au sein d'une itération. On cherche donc à vérifier si la relaxation converge vers une valeur quelle que soit l'entrée, et si elle est pertinente en large dimension avec de nombreuses cartes.

### 3.3.1 Analyse expérimentale

### 3.3.2 Champs de BMU

### 3.3.3 Limitations et possibilités en grande dimension

## 3.4 Implémentation

L'histoire de la recherche de consensus dans le graphe de cartes, permet que ce soit vraiment décentralisé

## 3.5 Perspectives d'évolutions



## Chapitre 4

# Analyser l'organisation : une approche par variables aléatoires

Avant de présenter les performances d'un algorithme, il s'agit de définir plus précisément ce qu'on attend de ce système et comment le représenter. L'architecture CxSOM se présente comme une construction qui répond à un questionnement structurel des réseaux de neurones. Mais au juste, qu'attend t-on de ce réseau de neurones ? De la prédiction, de l'organisation ? Les cartes de Kohonen sont habituellement utilisées dans un objectif de clustering, ou associées à d'autres algorithmes de prédiction utilisant leurs propriétés structurelles. En étude préliminaire pour CxSOM, il s'agit de comprendre le comportement de l'architecture de cartes.

### 4.1 Cas d'utilisation : les entrées multimodales

#### 4.1.1 Définition et inspiration biologique

#### 4.1.2 Formalisme

#### 4.1.3 Perspectives

Le formalisme présenté, avec des entrées multimodale comme fonction de variable cachées n'est pas forcément général.

### 4.2 Représentation des entrées

### 4.3 Information apprise par une carte

Une idée est de déterminer si une carte a gagné de l'information sur le modèle générant les entrées. Dans le cas simple, ce modèle peut être entièrement représenté par  $U$  ; chaque carte peut être représentée par son BMU, considéré comme la seule sortie de la carte. En traçant  $U$  en fonction de  $\Pi$ , le BMU d'une carte, on observe directement si une carte a été capable de lever l'ambiguïté sur le modèle en distinguant les entrées selon leur variable cachée  $U$ . Cette ambiguïté est levée si  $U$  est une fonction de  $\Pi$ . Cette fonction est observée dans le cas des cartes jointes.

Cette propriété, dans le cas 1D, peut être calculée par l'information mutuelle entre  $U$  et  $\Pi$ . Plus précisément, par  $\frac{I(U, \Pi)}{H(U)}$ , avec  $H(U)$  l'entropie de  $U$ . En effet, dans le meilleurs des cas,  $U$  est une fonction parfaite de  $\Pi$  et donc  $H(U|\Pi) = 0$  : en connaissant  $\Pi$ , on connaît totalement  $U$ . Alors,  $I(U, \Pi) = H(U) - H(U|\Pi) = H(U)$ . Notre indicateur vaut alors 1 lorsque  $U$  est une fonction parfaite de  $\Pi$ . De plus  $\Pi$  est forcément une fonction de  $U$  car l'algorithme est déterministe : à une entrée correspond une sortie, toujours la même, donc  $(I(U, \Pi) = H(\Pi))$ . Notre indicateur estimant l'information portée par le

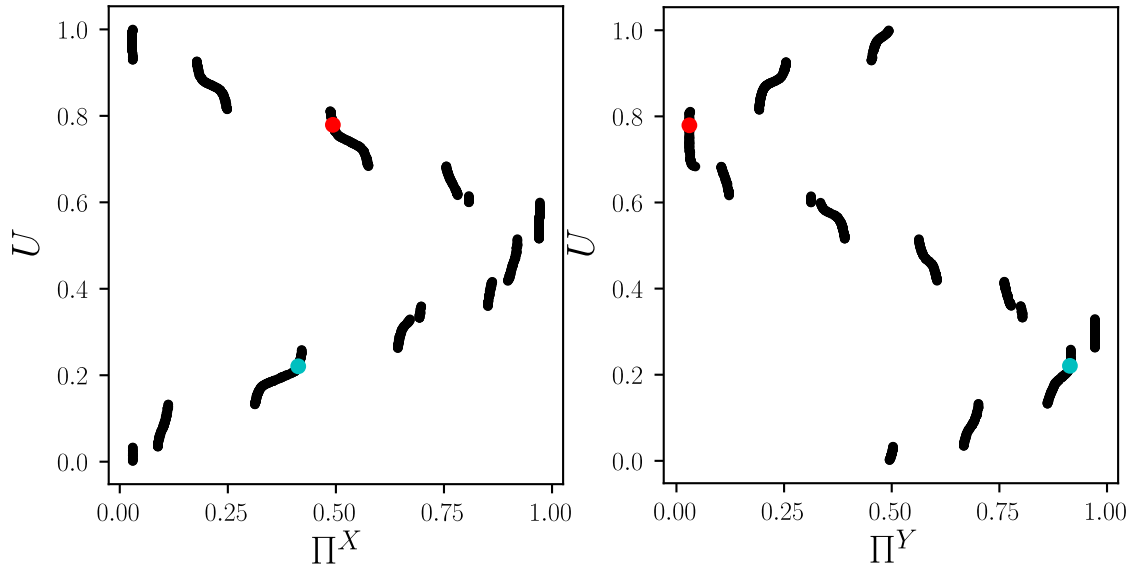


FIGURE 4.1 – Pour l'échantillon de test, valeur de  $U$  en fonction des valeurs du BMU  $\Pi$  dans chacune des cartes. On voit que  $U$  est une fonction du BMU dans chaque carte, contrairement au cas où les cartes apprendraient indépendamment sur les mêmes entrées, voir figure 4.2.

BMU d'une carte sur la variable cachée du modèle  $U$  est donc

$$\frac{H(\Pi)}{H(U)}$$

Cet indicateur doit être estimé en discrétisant les variables, donnant une entropie nécessairement positive et strictement supérieure à 0. L'évolution de l'indicateur au cours de l'apprentissage est donnée en figure 4.3. Cet indicateur est calculé en moyenne pour 100 réalisations de l'apprentissage, avec des poids initiaux différents.

#### Choses à faire

- Cette valeur est uniquement calculée pour un modèle connu, et en 1 dimension forcément. Peut on avoir des équivalents en plus de dimension ?
- Il existe des quantités mesurant l'information portée par un symbole sur une variable, une sorte d'info mutuelle locale. On sait que  $I(U, \Pi) = H(\Pi)$ , et on veut que  $I(U, \Pi) = H(U)$ , mais comment est elle répartie entre les BMUs ? Est-ce pertinent de se pencher sur ces quantités ?

### 4.4 Représenter une carte au sein d'une architecture

Représentation des poids, des entrées, des BMU - analyse

### 4.5 Prédiction d'entrée

Prédiction sur des données jouets

Prédiction sur drone

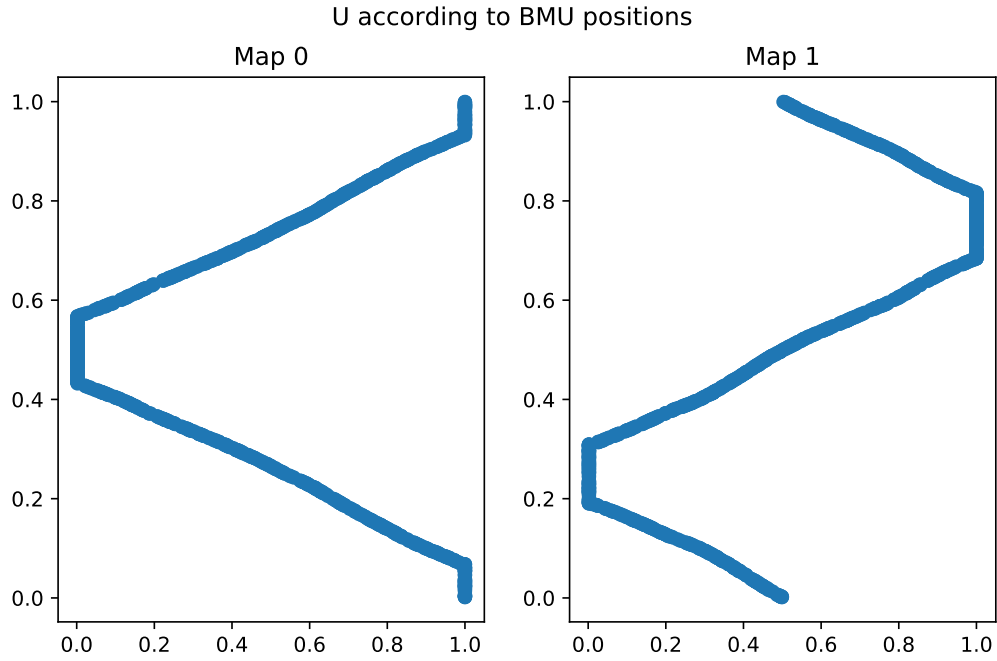


FIGURE 4.2 – Pour l'échantillon de test, entrée sur un cercle, valeur de  $U$  en fonction des valeurs du BMU  $\Pi$  dans chacune des cartes, lorsque les cartes  $M_x$  et  $M_y$  ne sont pas connectée. Chacune des cartes n'a aucune information de plus que celle portée par son entrée sur l'état global du système  $U$ , et  $\Pi$  n'est donc pas une fonction de  $U$  dans chaque carte.

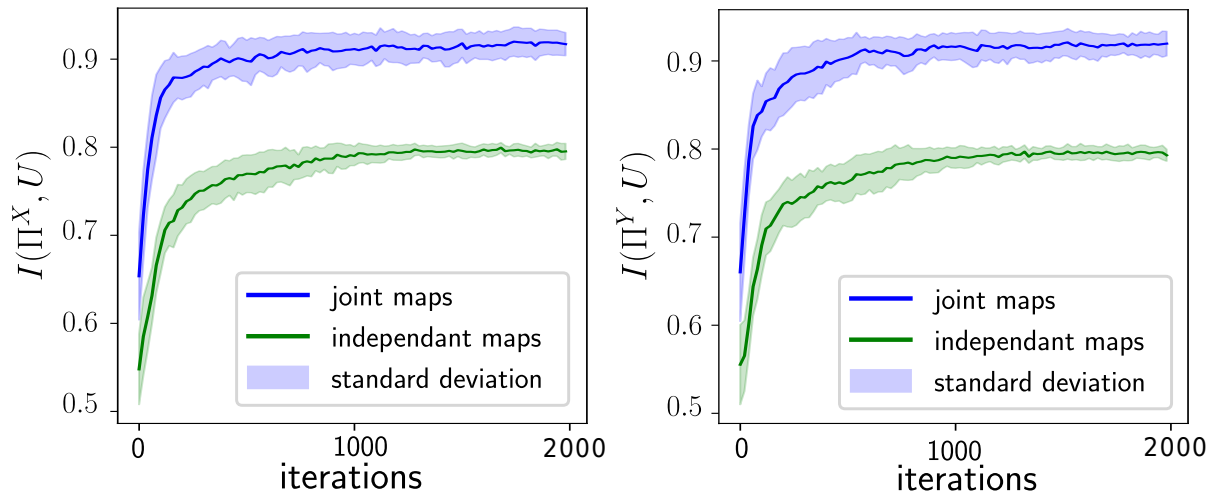


FIGURE 4.3 – Evolution de l'indicateur relatif à l'information mutuelle entre  $\Pi$  et  $U$  dans chaque carte au cours de l'apprentissage. Cet indicateur est comparé à celui calculé dans le cas où les cartes apprennent séparément.

Bien se placer dans le contexte “on va chercher à comprendre ce système dynamique”.  
Formaliser le problème en terme de variables aléatoires

## Conclusion

# Bibliographie

- [1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and D. Klein. Neural module networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, 2016.
- [2] Tom Binzegger, Rodney J. Douglas, and Kevan A. C. Martin. Cortical architecture. In M. De Gregorio, V. Di Maio, M. Frucci, and C. Musio, editors, *Brain, Vision, and Artificial Intelligence*. Springer-Verlag, 2005.
- [3] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE J. Robotics Autom.*, 2 :14–23, 1986.
- [4] R’obert Csord’as, Sjoerd van Steenkiste, and J. Schmidhuber. Are neural nets modular ? inspecting functional modularity through differentiable weight masks. *ArXiv*, abs/2010.02066, 2021.
- [5] Daniel J. Felleman and David C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. 1991.
- [6] Claus C. Hilgetag and Alexandros Goulas. Is the brain really a small-world network ? *Brain Structure & Function*, 221 :2361 – 2366, 2015.
- [7] M. Johnsson, C. Balkenius, and G. Hesslow. Associative self-organizing map. In *Proc. IJCCI*, 2009.
- [8] Louis Kirsch, Julius Kunze, and D. Barber. Modular networks : Learning to decompose neural computation. In *NeurIPS*, 2018.
- [9] J. Lampinen and E. Oja. Clustering properties of hierarchical self-organizing maps. *Journal of Mathematical Imaging and Vision*, 1992.
- [10] D. Meunier, R. Lambiotte, and E. Bullmore. Modular and hierarchically modular organization of brain networks. *Frontiers in Neuroscience*, 4, 2010.
- [11] V. Mountcastle. The columnar organization of the neocortex. *Brain*, 120(4) :701–722, April 1997.
- [12] German I. Parisi, Jun Tani, Cornelius Weber, and Stefan Wermter. Lifelong learning of spatio-temporal representations with dual-memory recurrent self-organization. *Frontiers in Neurorobotics*, 2018.
- [13] Edmund T. Rolls and Gustavo Deco. Computational neuroscience of vision. 2002.
- [14] A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641) :37–72, 1952.
- [15] C. Watanabe, Kaoru Hiramatsu, and K. Kashino. Modular representation of layered neural networks. *Neural networks : the official journal of the International Neural Network Society*, 97 :62–73, 2018.