

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ MOHAMED KHIDER, BISKRA

FACULTÉ des SCIENCES EXACTES et des SCIENCES de la NATURE et de la VIE

DÉPARTEMENT DE MATHÉMATIQUES



Mémoire présenté en vue de l'obtention du Diplôme :

Master en **Mathématiques**

Option : **Analyse**

Par

SAYAH OUANASSA

Titre :

**Une approche pratique des réseaux de
neurones artificiels**

Membres du Comité d'Examen :

Dr.	HOUAS AMRANE	UMKB	Président
Pr.	KHELIL Naceur	UMKB	Encadreur
Dr.	GUIDAD DERRADJI	UMKB	Examineur

juin2022

Dédicace

je dédie ce modeste travail à :

ma fleur de ma vie ma mère Haouchia Djaffale

Et le roi de mon coeur mon père Rabah

Pour les aimables mes frères Taher , Faisal , Bilkassim , Zino , Sadam, Ward

Et pour les nobles filles mes soeurs : Aida, Aicha , Razika

Aux femmes et aux enfants de mes chers frères : Abd Al Hai, choaib, Retaj, Amira,

Badre adin et Ishak

Pour tout grand familles, tout mes oncles

Pour tous les amies soit leurs l'étude ou dans la city universitaire :

Nesrine, Nedjet, Radia, Hada, Abir, Zahia, Najwa, Sara, Ahlam...

A tous ceux qui cherchent la connaissance parmi les attributs de cett note

A tous ceux qui ont été oubliés par ma plume et non oubliés par mon coeur

REMERCIEMENTS

Je tiens tout d'abord à remercier Pr KHELIL Naceur à l'Université de Biskra, qui m'a donné ce sujet et m'a encadré pendant mon Master. Il a toujours été à mon écoute et son point de vue complémentaire est souvent été très utile.

Je suis très reconnaissante envers Dr HOUAS AMRANE, et Dr GUIDAD DERRAJI d'avoir manifesté de l'intérêt pour mon travail en acceptant de participer à ce jury. Leurs remarques et commentaires constructifs m'ont permis d'en améliorer le manuscrit. Enfin, que toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail trouvent ici l'expression de mes sincères remerciements.

Mes derniers remerciements vont à mes parents, mes frères et soeurs et tous les amies.

Table des matières

Dédicace	i
Remerciements	ii
Table des matières	iii
Liste des figures	v
Liste des tableaux	vi
Introduction	1
1 Concepts de réseau neuronal	3
1.1 Notes historiques	3
1.1.1 Caractérisation des réseaux de neurones artificiels	5
1.1.2 Concepts mathématiques	7
2 Mémoire auto-associative bidirectionnelle (BAM)	9
2.1 Mémoire auto-associative bidirectionnelle (BAM)	10
2.2 Architecture BAM	10
2.3 Mémoire auto-associative	11
3 Modèle de Hopfield discret	14
3.1 Processus d'apprentissage	15

3.2 Principe de fonctionnement	16
3.3 Concept énergétique dans le modèle hopfield discret	17
3.4 Exemple de traitement	17
Conclusion	21
Annexe A : Mathématiques pour le chapitre 2 et 3	23
3.5 Mémoire associative	23
3.6 BAM (Bidirectional Associative Memory)	24
3.6.1 Architecture BAM	24
3.6.2 Dynamique de la BAM	25
3.6.3 Energie de la BAM	26
3.6.4 Commentaires	27
3.7 Mémoire de Hopfield discrète	28
3.7.1 Caractéristiques	29
3.7.2 Energie de la Mémoire Hopfield discrète	29
Annexe B : Abréviations et Notations	31

Table des figures

1.1 Neurone naturel	3
1.2 Modèle de neurone mathématique	7
2.1 Mémoire associative bidirectionnelle	11
2.2 Mémoire auto-associative	12
3.1 Modèle de Hopfield discret	15
3.2 La structure de la mémoire auto-associative	28
3.3 La structure du réseau de Hopfield	29

Liste des tableaux

2.1 Analogie entre le neurone biologique et le neurone formel	9
---	---

Introduction

Motivation

Au cours de la dernière décennie, les réseaux de neurones, ou mieux les réseaux de neurones artificiels (*ANN*), ont reçu de plus en plus l'attention dans la physique et les sciences de l'ingénieur. La motivation pour cet intérêt est la capacité d'apprentissage des *ANN* qui permet, sous conditions, de déplacer le fardeau d'effectuer des tâches de calcul difficiles de l'ingénieur au réseau de neurones, la « machine à apprendre », dans des domaines tels que le contrôle, le traitement du signal et reconnaissance de formes. Le présent travail s'occupe de la présentation de la mémoire auto-associative bidirectionnelle (*BAM*) afin de poser les bases nécessaires pour comprendre le modèle de réseau dynamique proposé en 1982 par Hopfield.

Organisation du mémoire

La partie principale de ce mémoire est composée de trois chapitres, Le mémoire comporte également, un résumé, une introduction et deux annexes, l'annexe [3.4](#) dans lequel on donne les précisions mathématiques pour le chapitre [2](#), chapitre [3](#) et l'annexe [3.7.2](#) est réservé pour les abréviations et les notations.

Dans le chapitre [1](#); nous parlerons du neurone comme étant un élément de base d'une structure complexe du cerveau

Le chapitre [2](#) présente un bref rappel de la mémoire auto-associative bidirectionnelle (*BAM*) afin de poser les bases nécessaires pour comprendre le modèle de réseau dynamique proposé en 1982 par Hopfield.

Dans le chapitre [3](#) nous aborderons les modèles discrets et continus proposés par Hopfield, partant de l'architecture de ce type de réseau, nous poursuivrons le traitement des données et des algorithmes d'apprentissage respectifs.

À la fin, nous montrerons quelques applications dans l'optimisation du système et la reconnaissance et la classification des formes.

Et on termine par une conclusion [3.4](#).

Chapitre 1

Concepts de réseau neuronal

1.1 Notes historiques

La forme et la fonction du cerveau humain fascinent les chercheurs depuis des siècles. L'une des premières contributions à la neurophysiologie moderne a été le travail de Ramon y Cajal en 1911, qui a suggéré les neurones (Fig.1.1) comme éléments de base d'une structure complexe du cerveau [1].

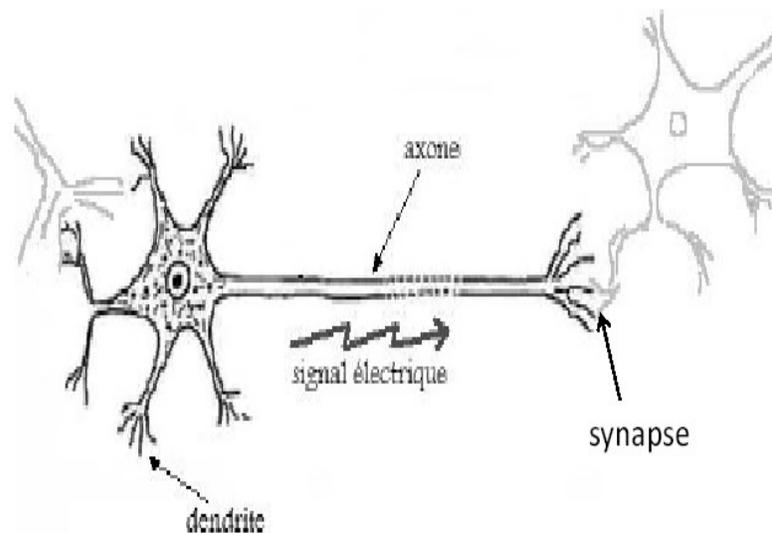


FIG. 1.1 – Neurone naturel

Le modèle neuronal introduit par McCulloch et Pitts (1943) considère également les neurones comme l'élément informatique de base du cerveau. Leur modèle de neurone est encore largement utilisé, que ce soit dans sa version originale ou forme modifiée, comme base pour les modèles informatiques du cerveau. En particulier, le neurone de McCulloch-Pitts incluait la notion de connexions synaptiques des neurones comme facteurs de pondération, ce qui est une simplification clé pour la représentation des synapses dans les réseaux de neurones artificiels. Au fil des décennies, notre connaissance de la structure du cerveau s'est affinée et bien que nous soyons loin d'une compréhension complète du fonctionnement du cerveau humain, il est certain que son organisation comporte un traitement de l'information distribué dans un grand nombre de neurones et les schémas de connexion synaptique extrêmement riches. Cela permet au cerveau d'effectuer des tâches complexes de traitement de l'information, qui n'ont pas encore été accomplies avec des ordinateurs conventionnels.

La connaissance est acquise dans le cerveau humain via un processus d'apprentissage qui se produit de deux manières différentes. Le cerveau d'un nouveau-né ne montre qu'un modèle de connexion clairsemé, mais se développe rapidement au cours des deux premières années suivant la naissance. Le modèle de connexion se développe en fonction de l'environnement perçu par l'enfant, la croissance la plus spectaculaire se produisant au cours des premiers mois. Ce développement structurel peut être considéré comme un « câblage » de la structure cérébrale selon l'expérience de l'individu enfant [6]. Le deuxième type d'apprentissage utilise la modification de la synaptique connexions entre les neurones adjacents. La première et la plus célèbre règle l'apprentissage qui décrit ce dernier phénomène a été suggéré par Hebb (1949). L'apprentissage de la langue postule que la synapse entre deux neurones actifs simultanément (corrélés activités neuronales) est renforcée. Ce processus d'apprentissage peut être considéré comme un « logiciel conception » selon l'expérience de l'individu. Selon la définition de « connaissance » de Fischler et Firschein [4] (1987), les informations stockées sont alors disponibles pour interpréter, prédire et répondre de

manière appropriée au monde extérieur.

À partir des années 1950, les ingénieurs et informaticiens ont modifié les modèles de neurones et théories d'apprentissage pour leurs objectifs spécifiques. Les architectures de ces premiers *ANN* peuvent être séparés selon deux concepts de base : les architectures des réseaux feedforward (d'anticipation) et les réseaux de feedback (retour) d'information. Des réseaux de neurones feedforward ont été appliqués par exemple. par Uttley (1956) pour la classification des motifs binaires, ou par Gabor (1954) pour les non-linéaires filtrage adaptatif. Rosenblatt (1958) a inventé le perceptron classique, un seul neurone, et l'a utilisé pour la reconnaissance de formes. Il a également présenté le théorème de convergence perceptron, la première preuve de convergence pour un processus d'adaptation de poids. Webos (1974) ont décrit un algorithme d'apprentissage en mode inverse dans le contexte des réseaux généraux avec les réseaux de neurones comme cas particulier. Cette méthode d'apprentissage a été redécouverte et popularisé comme l'algorithme de rétro-propagation(en anglais «backpropagation»)d'erreurs par Rumelhart, Hinton et Williams (1986). La deuxième classe d'ANN, les réseaux de neurones Feedback, a d'abord été étudiée comme mémoires associatives. Cette ligne de recherche a été établie par Grossberg (1967) et par Taylor (1956), qui a simulé les activités fonctionnelles du système nerveux avec un circuit électrique. La mémoire matricielle de corrélation basée sur le produit extérieur. La règle a fait l'objet d'une enquête indépendante par Anderson, Kohonen et Nakano. Le modèle de l'état du cerveau dans une boîte a été proposé en (1977) [3], posant le fondation du célèbre article de Hopfield(1982) [5] . Hopfield a établi une classe de réseaux de neurones de rétroaction dynamiquement stables qui sont aujourd'hui appelés réseaux de Hopfield.

1.1.1 Caractérisation des réseaux de neurones artificiels

Bien que les théories des réseaux de neurones artificiels aient été motivées par leur origines, de nombreux modèles et algorithmes d'apprentissage développés à partir d'une ingénierie perspective ne sont pas trop préoccupés par la plausibilité biologique. Au cours des

années, la recherche sur les réseaux neuronaux dans la communauté de l'ingénierie et de l'informatique a dévié (plus ou moins) de ses racines d'inspiration biologique. Mais il est évident qu'une expression explicite. L'objectif de la recherche sur les réseaux neuronaux a toujours été de trouver des principes par lesquels un grand nombre d'unités de traitement simples (« traitement local ») peuvent effectuer des calculs, et par lequel les connaissances nécessaires pour effectuer ces calculs est acquis par un processus d'apprentissage. Les modèles neurophysiologiques ont motivé la conception de paradigmes de réseaux de neurones artificiels, qui intègrent généralement les éléments suivants caractéristiques :

- Élément de traitement (traitement local). Variations du modèle de neurone de McCulloch-Pitts est principalement utilisé comme élément de base de traitement *ANN (PE)*.
- Couches de *PEs* (parallélisme). Les *ANN* sont organisés en couches de *PE* parallèles, mettant ainsi en œuvre un traitement parallèle et distribué.
- Mémoire distribuée. Les informations sont stockées dans un (grand) nombre de connexion poids entre les couches de *PE*.
- Algorithme d'apprentissage. pendant le processus d'apprentissage, les informations peuvent être stockées soit dans les poids de connexion (via le réglage du poids) soit dans la structure des connexions (via un apprentissage à structure variable).

Résumant le propriétés ci-dessus, la définition suivante des réseaux de neurones artificiels est présentée comme une version étendue de la définition dans [\[2\]](#) [*Aleksander et Morton 1990*].

Définition 1.1.1 : *Un réseau de neurones artificiels est un processeur massivement parallèle qui a la capacité de stocker des connaissances expérimentales et de les rendre disponibles pour utilisation. Il ressemble au cerveau à deux égards : 1) La connaissance s'acquiert par un processus d'apprentissage. 2) Les poids de connexion entre les neurones, et ou le modèle de connexion entre les neurones est adapté pour stocker les informations apprises.*

1.1.2 Concepts mathématiques

Cette section passe en revue les concepts mathématiques pertinents des paradigmes *ANN*. Un modèle mathématique de neurone (élément de traitement, ou *PE*), est montré dans Figure 1.2. Les entrées x_i de *PE* sont multipliées par leurs poids de connexion correspondants w_i . En notation vectorielle, il s'agit du produit scalaire du vecteur d'entrée $X \in R^P$ et du vecteur poids $W \in R^P$, ce qui donne la valeur d'activation a (équation 1.1). La sortie de *PE* est donnée par la fonction d'activation (ou fonction de transfert) $g(a)$ dans l'équation 1.2.

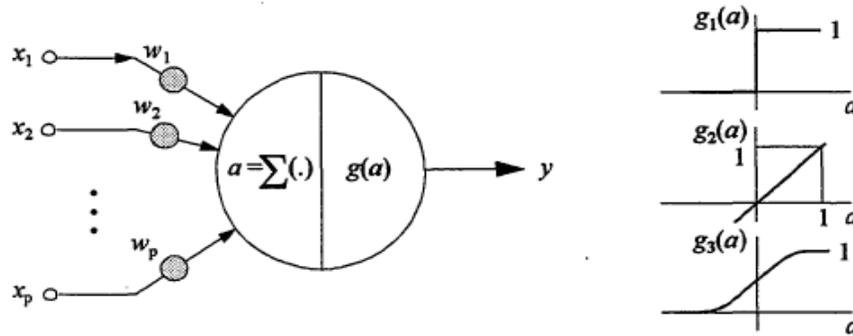


FIG. 1.2 – Modèle de neurone mathématique

$$a = w^T x \tag{1.1}$$

$$y = g(a) \tag{1.2}$$

Des exemples de fonctions d'activation courantes sont la fonction échelon, la fonction linéaire et la fonction sigmoïde, étiquetée g_1 , g_2 et g_3 , respectivement, sur la figure 1.2.

Le neurone de McCulloch-Pitts utilisait à l'origine une fonction échelonnée. L'activation linéaire et sigmoïde les fonctions utilisées dans ce travail sont définies comme

$$g_2(a) = a \tag{1.3}$$

$$g_3(a) = \frac{1}{1 + e^{-a}} \tag{1.4}$$

Le modèle de connexion entre les *PE* d'un réseau de neurones est appelé l'architecture ou la structure de l'*ANN*.

Chapitre 2

Mémoire auto-associative bidirectionnelle (BAM)

Un réseau de neurones est un modèle informatique dont la structure en couches est similaire à la structure en réseau des neurones du cerveau (Tableau ci-dessous), avec des couches de nœuds connectés.

Nourone biologique	Nourone formel
synapses	Poids des connexions
Axones	Signal sortie
Dendrites	Signal d'entrée
Noyau ou Somma	Fonction d'activation

TAB. 2.1 – Analogie entre le neurone biologique et le neurone formel

Un réseau de neurones peut apprendre à partir de données ; il peut ainsi être entraîné à reconnaître des tendances, classer des données et prévoir des événements à venir.

Un réseau de neurones décompose vos données en couches d'abstraction. Il peut être entraîné sur de nombreux exemples en vue de reconnaître des modèles au niveau de la

parole ou des images, par exemple, à l'instar du cerveau humain. Son comportement est défini par la façon dont ses éléments individuels sont reliés et par la solidité (ou poids) de ces liaisons. Ces poids sont automatiquement ajustés au cours de l'entraînement selon une règle d'apprentissage spécifiée jusqu'à ce que le réseau de neurones exécute correctement la tâche souhaitée.

Dans ce chapitre, nous commencerons par un bref rappel de la mémoire auto-associative bidirectionnelle (*BAM*) afin de poser les bases nécessaires pour comprendre le modèle de réseau dynamique proposé en 1982 par Hopfield.

2.1 Mémoire auto-associative bidirectionnelle (BAM)

L'une des caractéristiques les plus puissantes de l'intelligence humaine c'est la capacité d'associer des faits, des données, des situations, etc. Par exemple, Si nous posons une question, qui est Larbi ? Quelles réactions pourrions-nous attendre à ? . Comme il est naturel dans la communauté en général, très probablement, il n'y aura pas de réaction particulière et très sûrement, ils l'associeront au nom d'une personne d'origine arabe. Mais parmi nous, lecteurs de sujets liés aux réseaux de neurones artificiels, nous espérons qu'une sorte de réaction ou d'association spéciale a été générée ; sans crainte de se tromper, en entendant ce nom très sûrement à nos esprits le nom du réseau de neurones de type Perceptron vient immédiatement. Cette merveilleuse capacité de nous, êtres humains, à accomplir associations entre des idées, des concepts, des choses, etc. nous allons essayer de l'imiter avec ce nouveau type de réseau de neurones appelé *BAM*.

2.2 Architecture BAM

La mémoire associative bidirectionnelle (*BAM*) présentée dans la figure Fig. [2.1](#), Il est composé de deux couches de neurones qui sont les éléments de base de traitement d'infor-

mations. Ces couches sont complètement interconnectées et le flux d'informations va de la couche d'entrée à la couche de sortie et de la couche de sortie à la couche d'entrée, les pondérations sont donc bidirectionnelles.

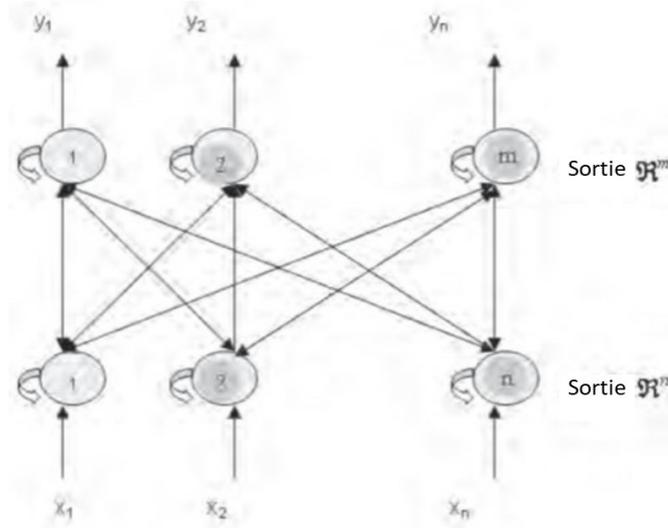


FIG. 2.1 – Mémoire associative bidirectionnelle

Si le problème est bien défini, alors nous connaissons tous les vecteurs d'apprentissage à l'avance et pouvons déterminer la matrice de poids W à l'aide de l'équation [2.1](#), où p est le nombre de modèles d'entraînement.

$$W = y_1 x_1^T + \dots + y_i x_i^T + \dots + y_p x_p^T \quad (2.1)$$

Avec cette expression on obtient la matrice W (Voir [3.5.3](#)) des poids synaptiques du réseau les données neuronales des paires ordonnées de vecteurs d'entrée et de sortie de l'ensemble d'apprentissage.

2.3 Mémoire auto-associative

Comme *BAM*, la mémoire auto-associative est composée de deux couches de neurones qui sont les éléments de base du traitement de l'information (figure [2.2](#)). Ces couches sont

complètement interconnectées et le flux de données va de la couche d'entrée à la couche de sortie et de la couche de sortie vers la couche d'entrée, donc les connexions et leurs poids respectifs sont bidirectionnels.

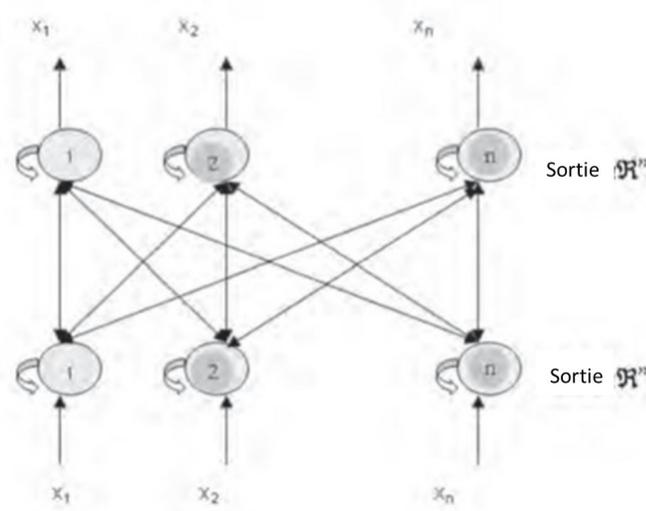


FIG. 2.2 – Mémoire auto-associative

Si nous connaissons tous les vecteurs d'apprentissage à l'avance, nous pouvons déterminer les poids à l'aide de l'équation [2.2](#).

$$W = x_1x_1^T + \dots + x_ix_i^T + \dots + x_px_p^T \quad (2.2)$$

Avec cette expression, nous obtenons la matrice W [3.5.3](#) des poids synaptiques du réseau de neurones, basée sur les paires ordonnées de vecteurs d'entrée et de sortie de l'ensemble d'apprentissage. Notez que puisqu'il s'agit d'un processus d'auto-association, les vecteurs d'entrée et de sortie utilisés pour l'apprentissage sont les mêmes. Dans ce cas, les deux couches sont de même dimension, donc la matrice de poids W est symétrique et de dimension $n \times n$.

Une fois la matrice de poids obtenue, nous pouvons utiliser le réseau pour associer les vecteurs d'entrée et de sortie, garantissant qu'il est capable de répondre aux données contaminées par le bruit.

Traitement de l'information dans le *BAM*

Une fois que la matrice de poids a été construite sur la base de l'équation [2.1](#), le *BAM* peut être utilisé pour mémoriser les informations stockées par les vecteurs de l'ensemble d'apprentissage. Les étapes que nous effectuerons pour traiter les informations sont les suivantes :

1. Nous appliquons une paire de vecteurs (x_i, y_i) aux neurones du *BAM*.
2. Nous propageons les informations de la couche x à la couche y et elles sont mises à jour les sorties des unités de couche et. Dans ce cas, il part de x à y , mais il peut aller de y à x et est connu sous le nom contre-propagation.
3. On propage les informations et mises à jour à l'étape précédente vers la couche x , et ces unités sont mises à jour.
4. Nous répétons les étapes 2 et 3 jusqu'à ce qu'il n'y ait aucun changement dans les sorties des deux couches.

Si l'apprentissage est adéquat, le système convergera vers un point qui correspondra à l'un des vecteurs utilisés pour construire la matrice de poids en phase d'apprentissage. La sortie pour toute entrée x sera la plus proche $\hat{O}(xi) = yi$.

Cet algorithme nous montre le caractère bidirectionnel du traitement des données dans le *BAM*. Pour élargir ce concept, considérons un problème où il est nécessaire de concevoir un système avec ce type de réseau qui associe l'empreinte et le nom d'un utilisateur, la différence avec une table traditionnelle est que si l'on introduit une empreinte digitale contaminée par le bruit , le système est capable de récupérer le nom correct, contrairement à une solution basée sur une table.

Le cas inverse peut aussi être résolu par le système, par exemple, si l'on introduit le patronyme "Larba" au *BAM*, celui-ci sur plusieurs itérations par les connexions du même serait en mesure de récupérer le nom de famille correct "Larbi" et son empreinte digitale correspondante. Une base de données avec un système de recherche traditionnel ne serait pas en mesure de se remettre de ces types d'erreurs.

Chapitre 3

Modèle de Hopfield discret

Le modèle discret du réseau de neurones de Hopfield est assimilé à une mémoire auto associative, où la proposition est de réduire les deux couches de neurones du *BAM* à une seule, où la sortie de ces neurones est amenée à l'entrée. Sur la figure Fig. [3.1](#), nous montrons ce modèle de réseau dynamique proposé par Hopfield et dont il tire son nom. L'entrée I_i atteint chacun des neurones, qui sont transformés par l'unité de traitement avec les poids synaptiques w_{ij} , pour générer la sortie y_i .

L'architecture correspond à un réseau d'une couche de neurones ou d'unités de traitement, ce réseau appartient donc à ce qu'on appelle réseaux de neurones monocouches. Cette couche est constituée de n neurones et la sortie de ces neurones constitue l'état du réseau. La sortie de chaque neurone se propage vers l'entrée de chacun des autres neurones, mais pas elle-même, il n'y a donc pas d'auto-récurrance ; c'est-à-dire que la sortie d'un neurone n'affecte pas l'entrée de lui-même.

Pour représenter les poids du réseau de neurones de Hopfield, nous utiliserons la notation matricielle.

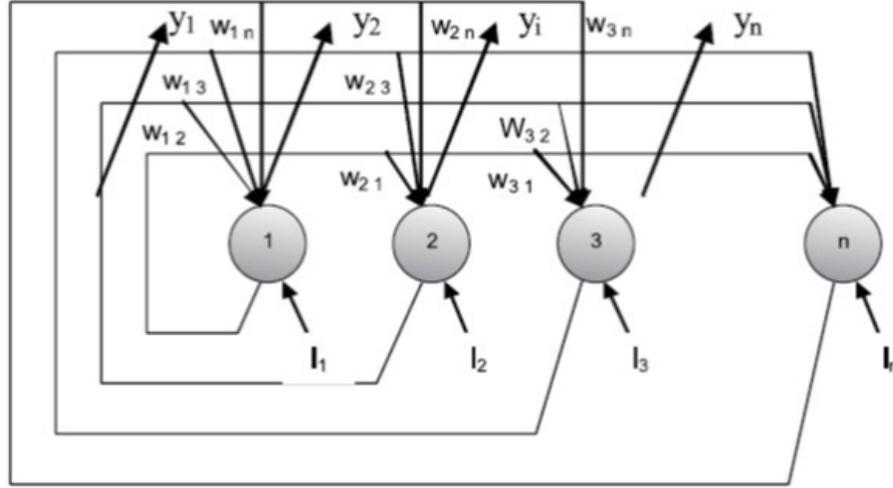


FIG. 3.1 – Modèle de Hopfield discret

3.1 Processus d'apprentissage

Dans cette section, nous présentons le processus d'apprentissage d'un réseau de Hopfield discret. L'information que ce réseau va traiter est de type binaire, pouvant coder les données comme zéro et un (0, 1), ou comme plus un et moins un (+1, -1).

Dans l'équation [3.1](#), nous présentons la façon de calculer le poids w_{ij} pour les motifs codés avec les valeurs 0 et 1, et dans l'équation [3.2](#) l'expression mathématique pour calculer le poids w_{ij} pour les régularités codées avec les valeurs -1 et 1.

Où e_i^k et e_j^k , sont les i -ième et j -ième composantes du k -ième modèle d'apprentissage, n est le nombre de modèles à mémoriser dans le réseau de neurones [3.6.2](#).

$$w_{ij} = \begin{cases} \sum_{k=1}^n (2e_i^k - 1)(2e_j^k - 1) \implies 1 \leq i, j \leq n; i \neq j \\ 0 \implies 1 \leq i, j \leq n; i = j \end{cases} \quad (3.1)$$

$$w_{ij} = \begin{cases} \sum_{k=1}^n (e_i^k e_j^k) \implies 1 \leq i, j \leq n; i \neq j \\ 0 \implies 1 \leq i, j \leq n; i = j \end{cases} \quad (3.2)$$

En conclusion :

L'apprentissage est un processus dynamique et itératif permettant de modifier les paramètres d'un réseau en réaction avec les stimuli qu'il reçoit de son environnement. Le type d'apprentissage est déterminé par la manière dont les changements de paramètre surviennent.

3.2 Principe de fonctionnement

Comme nous sommes déjà capables de calculer la matrice de poids d'un réseau de Hopfield discret, nous allons maintenant étudier son fonctionnement, c'est-à-dire que nous allons voir comment mettre à jour la sortie du réseau.

Comme nous l'avons décrit tout au long du texte, chacun des neurones traite les informations d'entrée en calculant son entrée totale comme la somme des entrées par leurs poids respectifs, mais maintenant nous ajoutons l'effet de la récurrence qui se fait à partir des sorties des autres neurones, comme le montre l'expression [3.3](#) qui calcule l'entrée nette net_i .

Une fois que la valeur totale d'entrée a été calculée avec l'équation [3.3](#), nous sélectionnons aléatoirement un neurone du réseau pour calculer la sortie qu'il va générer. Dans l'étape suivante, nous calculons la valeur de l'entrée nette dudit neurone. Prenant la valeur du revenu net calculé, Nous mettons à jour la sortie du neurone avec l'équation [3.4](#), pour les valeurs codées comme $\{-1, +1\}$. Dans ces expressions, I_i correspond à l'entrée externe du i ème neurone, et y_j est la sortie du j -ième neurone. Quand on dit que $i \neq j$ c'est parce que ce type de réseau n'admet pas d'auto-récurrence. Si les entrées sont codées en $\{0, 1\}$, l'équation [3.4](#) conserve son structure, en considérant que lorsque le Net est négatif, la sortie est nulle.

$$net_i = \sum_{k=1}^n y_k w_{ik} + I_i \text{ avec } i \neq j \quad (3.3)$$

$$x_i(t+1) = \begin{cases} 1; net \succ 0 \\ x_i(t); net = 0 \\ -1; net \prec 0 \end{cases} \quad (3.4)$$

3.3 Concept énergétique dans le modèle hopfield discret

Le réseau Hopfield est un système dynamique car les sorties du réseau peuvent être assimilées à un vecteur d'état qui, au fur et à mesure que l'information est traitée, évolue dans le temps. Pour vérifier sa convergence, le critère de stabilité de Lyapunov est utilisé. Afin de montrer la conformité avec le critère de stabilité de Lyapunov, nous définissons dans l'équation [3.5](#), une fonction d'énergie pour le rapporter. Nous pouvons vérifier que comme l'état d'un réseau Hopfield évolue l'énergie du réseau devient de moins en moins. En d'autres termes, l'énergie du réseau a tendance à décroître asymptotiquement.

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij} x_i x_j + \sum_{i=1}^N \theta_i x_i \quad (3.5)$$

Où,

w_{ij} Valeur du poids entre le i-ième neurone et le j-ième neurone

x_i Sortie du i-ième neurone

x_j Sortie du j-ième neurone

N Nombre de neurones dans la couche

θ_i Seuil du ième neurone

3.4 Exemple de traitement

Afin de clarifier la procédure de calcul de la matrice de poids, le fonctionnement du réseau et la procédure pour l'énergie, passons en revue l'exemple suivant où le problème est décrit

par deux vecteurs d'apprentissage x_1 et x_2 .

$$x_1^T = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix} \text{ et } x_2^T = \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix}$$

Si nous appliquons les expressions vues dans les sections précédentes, nous pouvons calculer

la matrice des poids, comme ceci :

$$w = x_1 x_1^T + x_2 x_2^T = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 & -2 & -2 \\ 2 & 0 & -2 & -2 \\ -2 & -2 & 0 & 2 \\ -2 & -2 & 2 & 0 \end{bmatrix}$$

Une fois la matrice de poids calculée, on peut passer à la phase d'exploitation et pour cela,

supposons que l'on présente le motif à l'entrée du réseau.

Le réseau répondra au stimulus de ce modèle d'entrée en modifiant ses sorties dans chacun de ses neurones :

Pour le neurone 1, nous calculons l'entrée nette :

$$net_1 = x_2 w_{12} + x_3 w_{13} + x_4 w_{14} + I_1 = 1 \times 2 + 1 \times -2 + -1 \times -2 + 1 = 3$$

Ensuite, nous évaluons cette entrée dans la fonction d'activation et obtenons leur sortie respective :

$$x_1 = 1$$

Pour le neurone deux, nous calculons l'entrée nette :

$$net_2 = x_1 w_{21} + x_3 w_{23} + x_4 w_{24} + I_2 = 1 \times 2 + 1 \times -2 + -1 \times -2 + 1 = 3$$

Ensuite, nous évaluons cette entrée dans la fonction d'activation et obtenons leur sortie respective :

$$x_2 = 1$$

Pour le neurone trois, nous calculons l'entrée nette :

$$net_3 = x_1 w_{31} + x_2 w_{32} + x_4 w_{34} + I_3 = 1 \times -2 + 1 \times -2 + -1 \times 2 + 1 = -5$$

Ensuite, nous évaluons cette entrée dans la fonction d'activation et obtenons sa sortie respective :

$$x_3 = -1$$

Notez que dans ce cas la sortie x_3 a changé de valeur, donc pour calculer le réseau du

neurone quatre cette nouvelle valeur doit être prise en compte :

$$net_4 = x_1w_{41} + x_2w_{42} + x_3w_{43} + I_4 = 1 \times -2 + 1 \times -2 + -1 \times 2 - 1 = -7$$

Ensuite, nous évaluons cette entrée dans la fonction d'activation et obtenons sa sortie respective :

$$x_4 = -1$$

Enfin, la sortie du réseau est déterminée par le vecteur x_f :

$$x_f = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}$$

Comme nous pouvons le voir, la sortie du réseau converge vers le premier modèle qui a été stocké dans le réseau. Voyons maintenant comment est le comportement de l'énergie du réseau, pour cela on revient au vecteur d'entrée x_0 .

$$x_0 = \begin{bmatrix} 1 & 1 & 1 & -1 \end{bmatrix}$$

Lors du démarrage de la sortie du réseau, c'est le même schéma d'entrée et, par conséquent, l'énergie du réseau dans ce cas, nous la calculons comme ceci :

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N w_{ij}x_i x_j, i \neq j$$

pour $i = 1$

$$E_1 = w_{21}x_1x_2 + w_{31}x_1x_3 + w_{41}x_1x_4 = 2$$

pour $i = 2$

$$E_2 = w_{12}x_2x_1 + w_{32}x_2x_3 + w_{42}x_2x_4 = 2$$

pour $i = 3$

$$E_3 = w_{13}x_3x_1 + w_{23}x_3x_2 + w_{43}x_3x_4 = -6$$

pour $i = 4$

$$E_4 = w_{14}x_4x_1 + w_{24}x_4x_2 + w_{34}x_4x_3 = 2$$

$$E = -\frac{1}{2} \sum_{i=1}^4 E_i = 2 + 2 - 6 + 2 = 0$$

Calculons maintenant l'énergie du réseau pour la sortie finale x_f , vers l'endroit où le réseau converge.

$$x_f = \begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}$$

pour $i = 1$

$$E_1 = w_{21}x_1x_2 + w_{31}x_1x_3 + w_{41}x_1x_4 = 6$$

pour $i = 2$

$$E_2 = w_{12}x_2x_1 + w_{32}x_2x_3 + w_{42}x_2x_4 = 6$$

pour $i = 3$

$$E_3 = w_{13}x_3x_1 + w_{23}x_3x_2 + w_{43}x_3x_4 = 6$$

pour $i = 4$

$$E_4 = w_{14}x_4x_1 + w_{24}x_4x_2 + w_{34}x_4x_3 = 6$$

$$E = -\frac{1}{2} \sum_{i=1}^4 E_i = -\frac{1}{2} (6 + 6 + 6 + 6) = -12$$

Notez que la sortie vers laquelle le réseau converge a une énergie inférieure à la sortie initiale. C'est logique car lorsque le réseau de Hopfield modifie sa sortie c'est parce qu'il converge vers un minimum de la fonction énergie.

L'explication en est que le réseau Hopfield a été conçu de telle sorte qu'à lui seul il constitue un système stable. Cela signifie que la sortie du réseau converge vers une valeur à laquelle le réseau se stabilise. Ce que nous espérons, c'est que cette sortie vers laquelle le réseau converge, soit l'un des modèles qui y sont stockés. Cependant, il est à noter que ce comportement est garanti si les motifs à mémoriser sont orthogonaux. Si cette condition n'est pas remplie, le réseau peut se stabiliser à un débit qui n'a pas été stocké, mais qui constitue un minimum de la fonction énergie.

Conclusion

Ce mémoire avait un objectif principal de poser les bases nécessaires pour comprendre le modèle de réseau dynamique proposé en 1982 par Hopfield. à partir de la mémoire auto-associative bidirectionnelle (*BAM*). Il s'agit d'une mémoire hétéro-associative, pour un motif d'entrée, elle renvoie un autre motif qui est potentiellement d'une taille différente . Ce phénomène est très similaire au cerveau humain. La mémoire humaine est nécessairement associative. Il utilise un ensemble d'associations mentales pour récupérer une mémoire perdue comme des associations de visages avec des noms, dans des questions d'examen avec des réponses, etc.

Bibliographie

- [1] Haykin, S. (1994) Neural Networks : A Comprehensive Foundation. Macmillan Publishing, New York.
- [2] Aleksander I. and H. Morton (1990), An Introduction to Neural Computing, Chapman & Hall, London.
- [3] Anderson, M. J., & Cohen, M. W. (1977). Nerve-induced and spontaneous redistribution of acetylcholine receptors on cultured muscle cells. *The Journal of Physiology*, 268(3), 757-773.
- [4] Fischler M.A. and O. Firschein (1987), *Intelligence : The Eye, the Brain, and the Computer*, Addison-Wesley, Reading, Massachusetts.
- [5] Hopfield J.J. (1982), "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the U.S.A.*, Vol. 79, pp. 2554-2558.
- [6] Vester F. (1975), *Denken, Lernen, Vergessen*, Deutscher Taschenbuch Verlag, Stuttgart, Germany
- [7] Hopfield J.J. (1984), "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences of the U.S.A.*, Vol. 81, pp. 3088-3092.

Annexe A : Mathématiques pour les chapitres 2 et 3

3.5 Mémoire associative

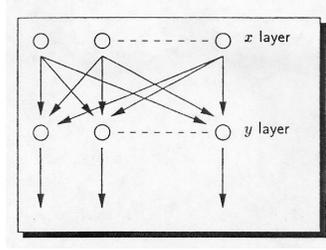
Définition 3.5.1 soit P des paires de vecteurs : $\{(x_1, y_1) \cdots (x_p, y_p)\}$, avec $x_p \in R^N, y_p \in R^K$ définissant un échantillon ; alors la fonction $M : R^N \longrightarrow R^K$ avec N, K et $P \in N^+$ est appelée mémoire hétéroassociative ssi :

- $M(x_p) = y_p \quad \forall p = 1, \dots, P$
- $M(x) = y_p \quad \forall x : \|x - x_p\| < \|x - x_l\| \quad l = 1, \dots, P, l \neq p$

Définition 3.5.2 soit P des paires de vecteurs : $\{(x_1, y_1) \cdots (x_p, y_p)\}$, avec $x_p \in R^N, y_p \in R^K$ définissant un échantillon ; alors la fonction $M : R^N \longrightarrow R^K$ avec N, K et $P \in N^+$ est appelée mémoire associative interpolative ssi :

- $M(x_p) = y_p \quad \forall p = 1, \dots, P$
- $\forall d \implies M(x_p + d) = y_p + e \quad d \in R^N, e \in R^K$,ie si si $x \neq x_p$ alors $y = M(x) \neq y_p, \forall p = 1, \dots, P$

Définition 3.5.3 Soit $M : R^N \longrightarrow R^K$ est appelée mémoire associative interpolative ssi : peut être construit à partir d'un ensemble orthonormé de vecteurs $\{x_p\}, p = 1, \dots, P$, alors la fonction M est définie comme $\implies M(x) = \left(\sum_{p=1}^P y_p x_p^T \right) x$



Définition 3.5.4 un ensemble de P vecteurs $\{x_1, \dots, x_p\}$ avec $x_p \in R^N$, $N, P \in N^+$ ensemble nommé échantillon alors on dit que la fonction M implémente une mémoire auto-associative interpolative ssi :

- $M(x_p) = x_p \forall p = 1, \dots, P$
- $M(x) = x_p \forall x : \|x - x_p\| \prec \|x - x_l\| \quad l = 1, \dots, P, l \neq p$

3.6 BAM (Bidirectional Associative Memory)

L'implimentation d'une mémoire associative interpolative consiste en deux couches de neurones totalement connectés

- l'entrée et la sortie peuvent être commutées de manière interchangeable, ie les sens des flèches sont inversés.

3.6.1 Architecture BAM

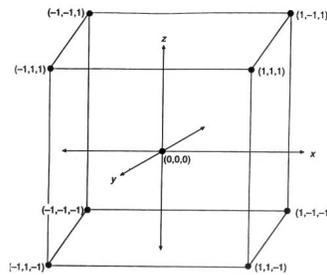
- Matrice des poids $W = \sum_{p=1}^P y_p x_p^T$, $\{x_p\}$, $p = 1, \dots, P$ ensemble orthogonal
- sortie réseau : $y = Wx$
- fonction d'activation $f(x) = x$
- si $\{y_p\}$ est orthogonal, alors le réseau est réversible : $x = W^t y$
- le réseau peut être utilisé comme mémoire auto associative en considérant $x = y$, alors

$$W = \sum_{p=1}^P x_p x_p^T$$

3.6.2 Dynamique de la BAM

- Dans l'ANN-BAM, les poids ne sont pas ajustés pendant la période d'entraînement. Ils sont calculés dès le départ, en fonction de l'ensemble des vecteurs stockés $(x_p, y_p)_{p=1, \dots, P}$
- Des vecteurs bipolaires sont utilisés (avec des valeurs "-1" ou "1") appartenant à l'espace de Hamming),
- $x = 2x^* - 1$ (avec des valeurs "0" ou "1")
- De $\{x_p\}$ et $\{y_p\}$ orthonormés, BAM
- L'espace de travail en temps discret

Distance de Hamming



Processus

- A $t = 0, x = 0$ s'applique au réseau y se calcule $y(0) = W x(0)$.
- La sortie des couches x et y est propagée dans les deux sens jusqu'à ce qu'un état stable soit atteint en utilisant

$$x_i(t+1) = f\left(W(:, i)^T y(t)\right) = \begin{cases} +1 & \text{si } W(:, i)^T y(t) \succ 0 \\ x_i(t) & \text{si } W(:, i)^T y(t) = 0, i = 1, \dots, N \\ -1 & \text{si } W(:, i)^T y(t) \prec 0 \end{cases} \quad (3.6)$$

$$y_i(t+1) = f\left(W(:,i)^T x(t+1)\right) = \begin{cases} +1 & \text{si } W(:,j)^T x(t+1) \succ 0 \\ y_i(t) & \text{si } W(:,j)^T x(t+1) = 0, i = 1, \dots, K \\ -1 & \text{si } W(:,j)^T x(t+1) \prec 0 \end{cases} \quad (3.7)$$

Forme matricielle

$$x(t+1) = \text{sign}(W^T y(t)) + |\text{sign}(W^T y(t))|^c \otimes x(t)$$

$$y(t+1) = \text{sign}(W^T x(t+1)) + |\text{sign}(W^T x(t+1))|^c \otimes y(t)$$

$$\text{Condition de stabilité : } \text{sign}(W^T y(t)) = \text{sign}(W^T x(t+1)) = \widehat{0}$$

– Lorsque l'on travaille dans le processus inverse $y(0)$ est appliqué au réseau, $x(0) =$

$$W^T y(t) \text{ est calculé à partir de : } y(t+1) = \text{sign}(W^T x(t)) + |\text{sign}(W^T x(t))|^c \otimes y(t)$$

$$x(t+1) = \text{sign}(W^T y(t+1)) + |\text{sign}(W^T y(t+1))|^c \otimes x(t) \quad \text{Condition de stabilité :}$$

$$\text{sign}(W^T x(t)) = \text{sign}(W^T y(t+1)) = \widehat{0}$$

- Le système résultat est stable
- Il existe une solution stable
- Le procesus converge vers la solution en un temps fini.

3.6.3 Energie de la BAM

La fonction energie de la BAM

$$E(x, y) = -y^t W x$$

Théorème : La fonction d'énergie possède les propriétés suivantes :

- $E_{t+1}(x(t+1), y(t+1)) \preceq E_t(x(t), y(t))$
- $E_{\min} = \sum_{i,i} |W_{ji}|$
- $\Delta E = E_{i+1} - E_i \prec \infty$

1. On peut vérifier que la fonction énergétique est une fonction de Lyapunov (qui décroît toujours pendant l'évolution libre du réseau de neurones de Hopfield ; les états stables de celui-ci sont donc des minima de cette fonction d'énergie) et donc le système dynamique a une solution stable.
2. Essentiellement, la matrice de poids détermine une surface avec des vallées (dépressions avec attracteurs) et des collines similaires à BPL.
3. BAM ressemble à un système physique dissipatif dans lequel la fonction E , correspond à l'énergie du système physique.
4. Initialement, les changements de $E(x, y)$ sont importants et lorsque les vecteurs x et y atteignent leur état stable, la valeur de $E(x, y)$ a des changements de plus en plus petits.

Proposition 3.6.1 *Si le motif d'entrée x_i est égal au $\{x_p\}$ stocké, alors y_l sera renvoyé*

Proof. $y = \text{sign}(W(x(t))) = \text{sign}\left(\sum_{p=1}^P y_p x_p^t x_l\right)$ ■

$$\text{sign}\left(\sum_{p=1}^P y_p \delta_{pl}\right) = \text{sign}(y_l) = y^l$$

3.6.4 Commentaires

- Le processus d'exécution est convergent et la solution est atteinte en un temps fini.
- Le nombre maximum de vecteurs pouvant être enregistrés est 2^{N-1}
- Les vecteurs de Hamming sont symétriques par rapport à la notation ± 1 . Par conséquent, le vecteur de Hamming porte la même information que son complément x comme $x^c = -x$ et $y_p = Wx_p$
vous avez: $y_p^c = -y_p = Wx_p = W(-x_p) = Wx_p^c$
- La BAM enregistre la direction des vecteurs échantillons et non leurs valeurs.

Algorithme

1. Initialisation du réseau
2. calculer la matrice de poids W
3. réseau récursif en avant
 - (a) étant donné $x(0)$, calculer $y(0) = Wx(0)$
 - (b) Calculer $x(t+1) = \text{sign}(W^T y(t)) + |\text{sign}(W^T y(t))|^c \otimes x(t)$ $y(t+1) = \text{sign}(W^T x(t+1)) + |\text{sign}(W^T x(t+1))|^c \otimes y(t)$
 - (c) Jusqu'à ce que le réseau soit établi
4. Réseau récursif vers l'arrière
 - (a) étant donné $y(0)$, calculer $x(0) = W^T y(0)$
 - (b) Calculer $y(t+1) = \text{sign}(W^T x(t)) + |\text{sign}(W^T x(t))|^c \otimes y(t)$ $x(t+1) = \text{sign}(Wy(t+1)) + |\text{sign}(Wy(t+1))|^c \otimes x(t)$
 - (c) Jusqu'à ce que le réseau se stabilise

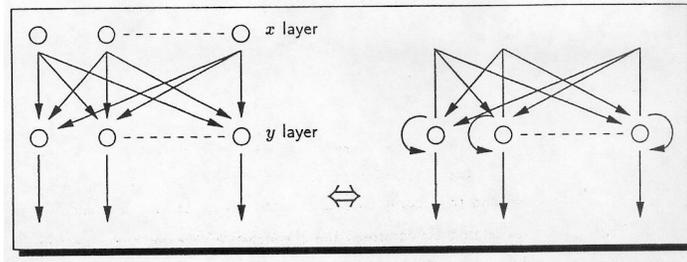


FIG. 3.2 – La structure de la mémoire auto-associative

3.7 Mémoire de Hopfield discrète

Il se compose d'une mémoire auto-associative avec une couche entièrement connectée qui reçoit en outre un signal d'entrée externe x ,

$$W = \sum_{p=1}^P y_p y_p^T$$

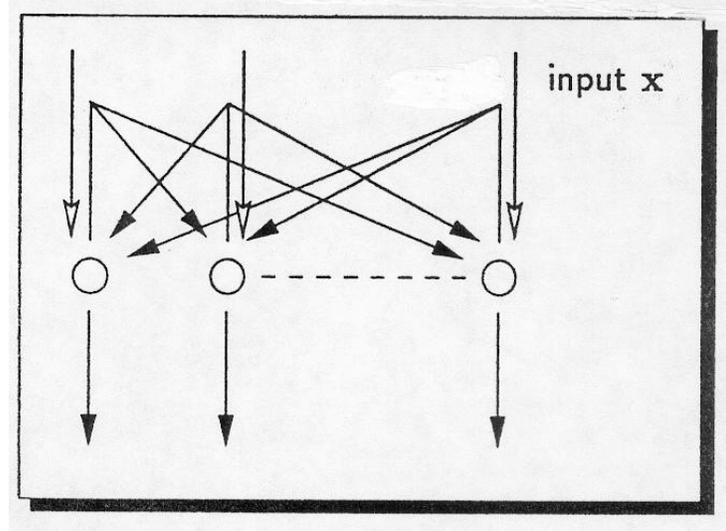


FIG. 3.3 – La structure du réseau de Hopfield

3.7.1 Caractéristiques

- Travaille avec des vecteurs binaires $\{0, 1\}$

- Matrice poids $W = \sum_{p=1}^P (2y_p - \hat{1})(2y_p - \hat{1})^T$ avec une diagonale égal 0.

- La fonction d'actualisation $y_j(t+1) = \begin{cases} +1 & \text{si } \sum_{k=1}^K w_{jk}y_k + x_j \succ t_j \\ y_j(t) & \text{si } \sum_{k=1}^K w_{jk}y_k + x_j = t_j \\ 0 & \text{si } \sum_{k=1}^K w_{jk}y_k + x_j \prec t_j \end{cases}$ avec $\{t_j\}_{j=1,K} =$

t est le vecteur symbolique

- Sous forme matricielle $A(t) = \text{sign}(Wy(t) + x - t)$

$$y(t+1) = \frac{1}{2} \left[A(t) + \hat{1} - |A(t)|^c \right] + |A(t)|^c \otimes y(t)$$

3.7.2 Energie de la Mémoire Hopfield discrète

La fonction energie

$$E = -\frac{1}{2}y^T W y - y^T (x - t)$$

Théorème 3.7.1 : La fonction d'energie possède les propriétés suivantes :

- $E_{t+1}(y(t+1)) \preceq E_t(y(t))$
- $E_{\min} = -\frac{1}{2} \sum_{i,i} |W_{ji}| - K$
- $\Delta E = E_{i+1} - E_i \prec \infty$

Le processus d'itération converge en temps fini

Annexe B : Abréviations et Notations

Les différentes abréviations et notations utilisées tout au long de ce mémoire sont expliquées ci-dessous.

<i>ANN</i>	Artificial Neural Networks
<i>BAM</i>	Mémoire auto-associative bidirectionnelle
<i>PE</i>	Processing Element
<i>BPL</i>	Back-Propagation Learning

المخلص

واحدة من أقوى خصائص الذكاء البشري هي القدرة على ربط الحقائق والبيانات والمواقف وما إلى ذلك. تخيل أنه طلب منك حفظ 10 أرقام هاتف (10 صور مختلفة ، 10 جمل ، إلخ)، و بعد فترة التعلم، يتم تقديم أحد هذه الأرقام الأولى (أو جزء من إحدى الصور أو الجمل)، وتتذكر تلقائيًا الرقم أو الصورة المحفوظة المقابلة. ما هي الآلية العصبية الكامنة وراء هذا النوع من الذاكرة والتي تسمى "الذاكرة الترابطية"؟

تعتبر شبكة هوبفيلد العصبية نموذجًا عصبيًا للذاكرة الترابطية. يتعامل العمل الحالي مع عرض ذاكرة الترابط الذاتي ثنائية الاتجاه (BAM) من أجل وضع الأسس اللازمة لفهم نموذج الشبكة الديناميكي الذي اقترحه هوبفيلد في عام 1982.

الكلمات المفتاحية: شبكة هوبفيلد، الذاكرة الترابطية ثنائية الاتجاه (BAM)، الشبكات العصبية الاصطناعية (ANN).

RESUME

L'une des caractéristiques les plus puissantes d'intelligence humaine c'est la capacité d'associer des faits, des données, des situations, etc. Imaginez que l'on vous demande de mémoriser 10 numéros de téléphone (10 images différentes, 10 phrases, etc). Après une période d'apprentissage on vous présente les premiers chiffres d'un de ces numéros (ou une partie d'un des images ou des phrases). Vous vous rappelez automatiquement le numéro ou l'image mémorisée correspondant. Quel est le mécanisme neuronal sous-tendant ce type de mémoire, appelé " mémoire associative"?

Le réseau de neurones de Hopfield est considéré comme un modèle neuronal de la mémoire associative. Le présent travail s'occupe de la présentation de la mémoire auto-associative bidirectionnelle (BAM) afin de poser les bases nécessaires pour comprendre le modèle de réseau dynamique proposé en 1982 par Hopfield.

Mots clés: Réseau Hoffield, Mémoire Associative Bidirectionnelle (BAM), Réseaux de Neurones Artificiels (ANN)

ABSTRACT

One of the most powerful characteristics of human intelligence is the ability to relate facts, data, situations, etc. Imagine being asked to memorize 10 telephone numbers (10 different images, 10 sentences, etc..). After one learning period you are presented with the first digits of one of these numbers (or a part of one of the images or sentences), You automatically remember the number or the corresponding memorized image. What is the neural mechanism underlying this type of memory, called "associative memory"?

Hopfield's neural Network is a neural model of memory associative. The present work deals with the presentation of auto-associative memory (BAM) in order to lay the foundations for necessary to understand the model of dynamic network proposed in 1982 by Hopfield.

Keywords: Hopfield Network, Bidirectional Associative Memory (BAM), Artificial Neural Networks (ANN).