

3.1. Réseaux de neurone artificiels

Aujourd'hui, plusieurs termes sont utilisés dans la littérature pour montrer le domaine des réseaux de neurones artificiels (RNA), comme le connexionnisme ou la simulation neuronale. Selon nous, chacun de ces noms doit être associé à une sémantique précise. Par conséquent, le réseau de neurones artificiels se réfère uniquement à la manipulation de modèles ; ce n'est ni un domaine d'étude ni une discipline scientifique. Le connexionnisme et la simulation neuronale sont deux domaines d'étude en soi, manipulant tous deux des modèles de RNA, mais avec des objectifs distincts. L'objet poursuivi par les ingénieurs et chercheurs connexionnistes est d'amplifier la puissance de calcul en utilisant des modèles à composants fortement connectés. De leur côté, les neuro-simulateurs utilisent des modèles de RNA afin de valider leurs hypothèses biologiques sur le fonctionnement du système nerveux central [111].

3.1.1. Définitions

Avant d'introduire le concept technique de réseaux de neurones artificiels proposé dans la littérature, commençons par les définitions des trois mots "réseau", "neurone" et "artificiel" dans la langue de Molière :

- **Vocabulaire 1 (Nom) :** Un réseau est un ensemble organisé dont les éléments sont répartis en différents points, interconnectés pour assurer leur communication.
- **Vocabulaire 2 (Nom) :** Les neurones sont des cellules basées sur des tissus nerveux capables de recevoir, d'analyser et de produire des informations.
- **Définition 1 :** Un réseau de neurones est un graphe conduit, composé d'un ensemble d'unités, exécutant des traitements élémentaires, construites sous forme de couches successives reliées capables d'échanger des informations à travers des liens structurés.

Un réseau neuronal artificiel inspiré de concepts biologiques n'est qu'un modèle mathématique de ce graphe. Ainsi, ils ont reproduit certaines caractéristiques de l'analyse neurobiologique (la capacité d'apprendre, de mémoriser des informations et de traiter des informations incomplètes ou corrompues). Depuis l'introduction du concept, plusieurs modèles neuronaux ont été proposés. Les modèles artificiels proposés dans une perspective connexionniste s'appuient sur des modèles biologiques de structure et de comportement global. Ainsi, les principes d'interconnexion et de

parallélisme sont toujours respectés, quels que soient les principes de structuration. Sur la base de ce principe, les réseaux de neurones ont fait leurs preuves.

- **Vocabulaire 3 (Adjectif) :** La manipulation résulte un produit artificiel.
- **Définition 2 :** Un réseau neuronal artificiel est un réseau fortement connecté de processeurs de base fonctionnant en parallèle. Chaque processeur de base calcule une sortie unique basée sur les informations qu'il reçoit en entrée. Toute hiérarchie de réseaux est évidemment un autre réseau [89].

3.1.2. Historique

L'origine de l'inspiration pour les réseaux de neurones artificiels remonte à 1890, lorsque le célèbre psychologue américain W. James a proposé le concept de mémoire associative. Il a proposé ce qui allait devenir la loi de fonctionnement des réseaux de neurones d'apprentissage, plus tard la loi de Hebb. Quelques années plus tard, en 1949, J. Mc Culloch et W. Pitts donnent son nom au modèle du neurone biologique (neurone à comportement binaire). Ils ont d'abord démontré que de simples réseaux de neurones formels peuvent exécuter des fonctions logiques, arithmétiques et symboliques complexes. C'est alors que le physiologiste américain D. Hebb propose en 1949 les propriétés des neurones agissant sur les animaux par conditionnement. Ainsi, le conditionnement pavlovien, comme nourrir le chien à la même heure chaque jour, fait saliver l'animal à ce moment précis, même en l'absence de nourriture. Ce résultat expérimental s'explique en partie par sa proposition de modification des propriétés des connexions entre neurones.

Le premier succès de la discipline remonte à 1957, lorsque F. Rosenblatt développe le modèle perceptron. Il a construit le premier ordinateur neuronal basé sur ce modèle et l'a appliqué au domaine de la reconnaissance de formes. Il faut dire qu'à l'époque, les moyens dont il disposait étaient limités, et c'était une prouesse technique de réussir à faire fonctionner la machine plus de quelques minutes. C'est alors, en 1960, que l'automaticien Widrow développe le modèle Adaline (Adaptive Linear Element). Dans sa structure, le modèle est similaire à un perceptron, mais avec des lois d'apprentissage différentes.

C'est l'origine de l'algorithme de rétro propagation de gradient largement utilisé dans les perceptrons multicouches aujourd'hui. M. Minsky et S. Papert ont publié un article en 1969 qui soulignait les limites théoriques des perceptrons [34]. Ces

limitations sont liées à l'incapacité de gérer des problèmes non linéaires à l'aide de ce modèle.

Suivent quelques années d'ombres, de 1967 à 1982. Le renouveau de la discipline reprend en 1982, grâce au physicien reconnu J. J. Hopfield. Dans un article court, clair et bien écrit, il a développé une théorie sur les capacités et les possibilités des réseaux de neurones. Notez l'introduction non conventionnelle de son article.

Alors que les auteurs ont jusqu'à présent insisté sur la proposition de lois de structure et d'apprentissage, lors de l'étude des propriétés émergentes, JJ Hopfield fixe d'abord le comportement que son modèle est censé atteindre, et à partir de là construit les lois de structure et d'apprentissage correspondant aux résultats attendus. Ce modèle est encore largement utilisé aujourd'hui pour des problèmes d'optimisation. On peut également citer la machine de Boltzmann de 1983, le premier modèle connu qui gère de manière satisfaisante les limitations rencontrées dans le cas du perceptron. Mais il est difficile à utiliser en pratique, le temps de convergence de l'algorithme est extrêmement long (le temps de calcul est assez long). Puis, en 1985, il y a eu une rétro propagation par gradient. C'est un algorithme d'apprentissage adapté aux perceptrons multicouches. Ses conclusions ont été faites par trois groupes de chercheurs indépendants. Sur la base de cette constatation, il est possible d'implémenter une fonction d'entrée/sortie non linéaire sur le réseau en décomposant cette fonction en une série d'étapes linéairement séparables. Enfin, en 1989, Moody et Darken ont proposé un réseau de fonctions de base radiales (RFR) utilisant certains résultats d'interpolation multi variée, le nom anglais Radial Basis Function network (RBF). Ce qui distingue ce type de réseau des autres types de réseaux de neurones est sa représentation locale [25].

3.1.3. Intérêts pour l'utilisation des réseaux de neurones artificiels

Les réseaux de neurones formels, tels que nous les définirons dans les paragraphes suivants, ont des propriétés approximatives et peuvent être formulés comme suit :

Toute fonction délimitée suffisamment régulière peut être approximée avec une précision arbitraire dans le champ fini de son espace variable. Un réseau de neurones se compose d'une couche d'un nombre fini de neurones cachés, qui ont tous la même fonction d'activation et la même sortie linéaire de neurones.

De plus, les réseaux de neurones nécessitent moins de paramètres ajustables (poids de connexion) que d'autres outils mathématiques couramment utilisés. En pratique, les réseaux de neurones ne sont pas utilisés pour effectuer des approximations de fonctions connues. La plupart du temps, le problème qui se pose est de trouver la relation entre un ensemble de sorties dans un processus donné et un ensemble d'entrées correspondantes à travers les mesures effectuées. Nous admettons que cette relation existe malgré : le nombre limité des mesures, ils doivent être pollués par le bruit et toutes les variables qui déterminent le résultat d'un processus ne doivent pas être mesurées.

En d'autres termes, l'ingénieur recherche un modèle du procédé sur lequel il travaille à partir des mesures dont il dispose : disons qu'il effectue une modélisation "boîte noire". Dans la « grammaire » des réseaux de neurones, les données sur lesquelles nous cherchons à construire un modèle sont appelées exemples. Par conséquent, la raison pour laquelle les réseaux de neurones sont considérés comme un bon choix est qu'une approximation peut être trouvée à partir des mesures disponibles du modèle RNA.

En général, les réseaux de neurones utilisent mieux les mesures disponibles que les méthodes d'approximation non linéaires traditionnelles. Ce gain peut être considérable lorsque le procédé à modéliser dépend de plusieurs variables, comme dans le cas de plusieurs types de non-linéarités et de procédés de mise en forme où plusieurs paramètres matériels et techniques interviennent [113].

3.1.4. Concept neurobiologique

Les éléments de base du système nerveux central sont les neurones. Le cerveau se compose d'environ un billion de neurones, chacun avec 1000 à 10000 synapses (connexions).

Les neurones sont des cellules composées d'un corps cellulaire et d'un noyau (Figure 3.1.). Le corps cellulaire se ramifie pour former ce que l'on appelle des dendrites. Celles-ci sont parfois si nombreuses qu'on parle de poils dendritiques ou de structures dendritiques. C'est par les dendrites que l'information est transmise de l'extérieur vers le corps cellulaire (corps neuronal). L'information est ensuite traitée par le corps cellulaire. Si le potentiel d'action dépasse un certain seuil, le corps cellulaire répond au stimulus (Fig. 3.2, (a)). Le signal transmis par le neurone voyage

alors le long de l'axone (unique) vers d'autres neurones. La transmission entre deux neurones n'est pas directe. En effet, il existe un gap cellulaire de plusieurs dizaines d'angströms entre les axones des neurones afférents et les dendrites des neurones efférents. La connexion entre deux neurones s'appelle une synapse (Fig. 3.2, (b.)) [25].

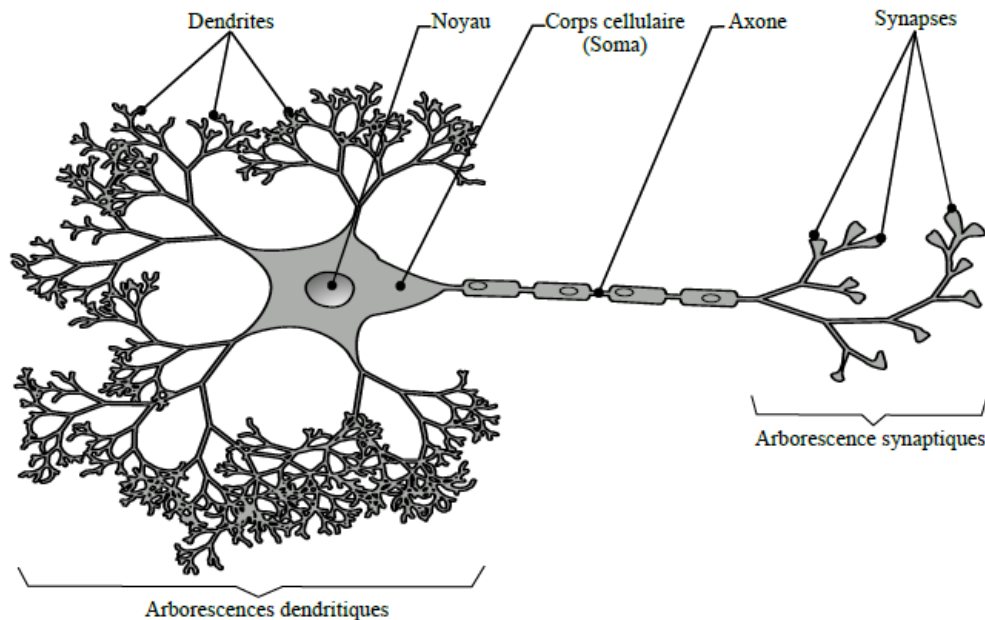


Fig. 3.1. Structure d'un neurone biologique.

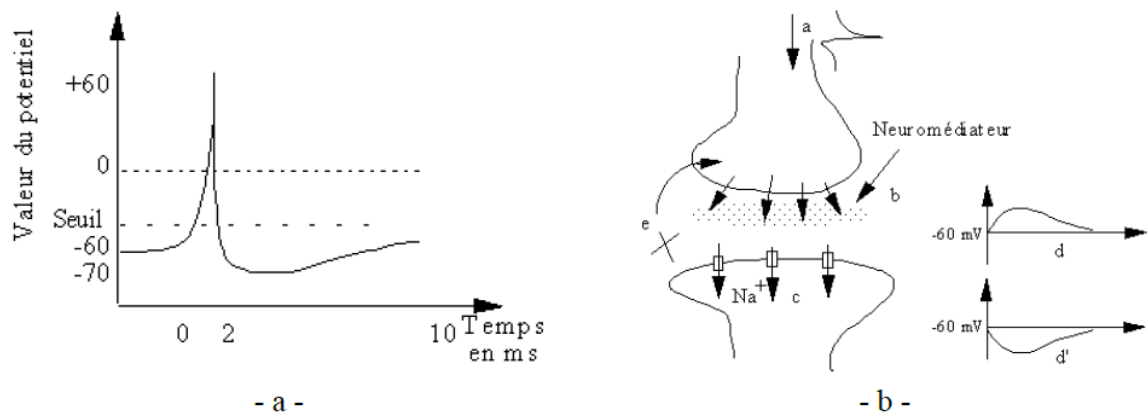


Fig. 3.2, (a). Potentiel d'action. (b) Opération au niveau synaptique (a) L'arrivée d'un potentiel d'action. (b) Libération de neuromédiateurs dans l'espace synaptique.

3.1.5. Les modèles mathématiques

De nombreuses applications telles que la reconnaissance de formes, le traitement du signal, l'apprentissage d'exemples, la mémoire, la généralisation, etc. sont facilement mises en œuvre par les réseaux de neurones biologiques. Cependant,

malgré tous les efforts en algorithmes et en intelligence artificielle, ces applications sont encore à la limite des possibilités actuelles. Les réseaux de neurones artificiels sont basés sur l'hypothèse que le comportement intelligent découle de la structure et du comportement des éléments de base du cerveau. Les RNA sont des modèles, ils peuvent donc être décrits par leurs composants, leurs variables descriptives et les interactions des composants.

3.1.6. Structure de neurone artificiel

La figure 3.3 montre la structure du neurone artificiel. Chaque neurone artificiel est un processeur de base. Il reçoit un nombre variable d'entrées des neurones en amont. Chacune de ces entrées est associée à une abréviation de poids w qui représente le poids de la force de connexion. Chaque processeur de base à une sortie, qui se ramifie ensuite pour fournir un nombre variable de neurones en aval. Chaque connexion est associée à un poids [111].

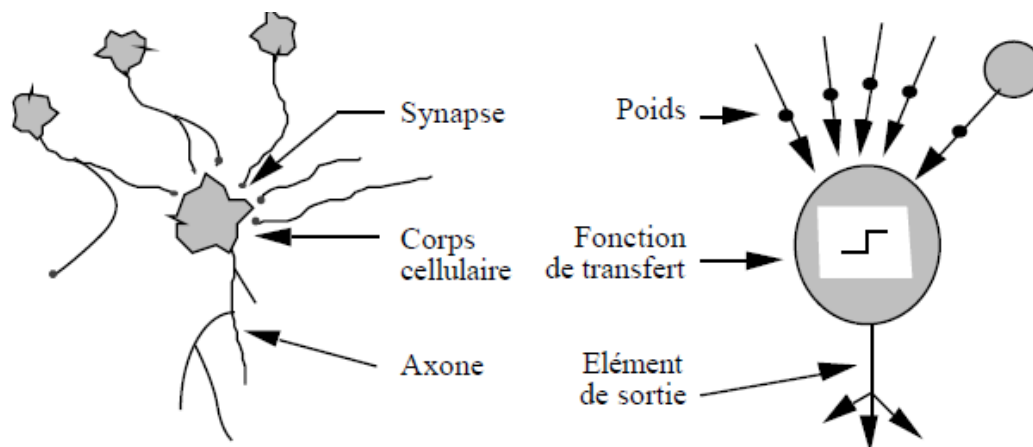


Fig. 3.3. Faites correspondre les neurones biologiques/neurones artificiels.

3.1.7. Comportement

Pendant le fonctionnement d'un réseau de neurones, les neurones formels produisent des sorties variables basées sur les propriétés d'entrée de l'environnement externe ou sur les propriétés d'entrée des neurones voisins en fonction de leur position dans le réseau. La variable de sortie est la valeur différentielle qui permet de distinguer l'état actif ou inactif du neurone.

Après avoir reçu les attributs d'entrée, ces derniers passent par une fonction de pondération avec des poids de connexion. Toutes les valeurs pondérées sont ensuite transmises à la fonction de combinaison. Ainsi, la somme pondérée est calculée.

La somme pondérée est calculée par l'équation suivante :










Enfin, la somme sera envoyée à la fonction d'activation qui est souvent identifiée par la fonction de transfert. Cette fonction permet de générer une valeur de sortie basée sur un seuil introduit en paramètre. Il existe plusieurs formes de fonctions de transfert dans la littérature.

Pour résumer, un neurone calcule sa valeur de sortie à l'aide de l'équation 3. La forme de cette équation reflète la non-linéarité du traitement basé sur le réseau neuronal. La valeur de sortie représente l'état du neurone, qui sera déclenché vers les neurones en aval ou directement vers l'environnement externe [112].

3.1.8. Fonctions d'activation

Les fonctions de transfert jouent un rôle important dans les réseaux de neurones, et le tableau 3.1 montre les différents types de fonctions de transfert et la relation entre les entrées et les sorties des neurones :

Tableau. 3.1. Fonctions de transfert $a = f(n)$.

Nom de la fonction	Relation d'entrée/sortie	Icône
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
linéaire	$a = n$	
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
sigmoïde	$a = \frac{1}{1 + \exp^{-n}}$	
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
compétitive	$a = 1$ si n maximum $a = 0$ autrement	

Parmi ces fonctions, il en existe trois les plus utilisées dans le domaine de l'ingénierie. Ces fonctions sont : "seuil" (a) ("limite dure"), "linéaire" (b) et "sigmoïde" (c) comme le montre la figure 3.4 :

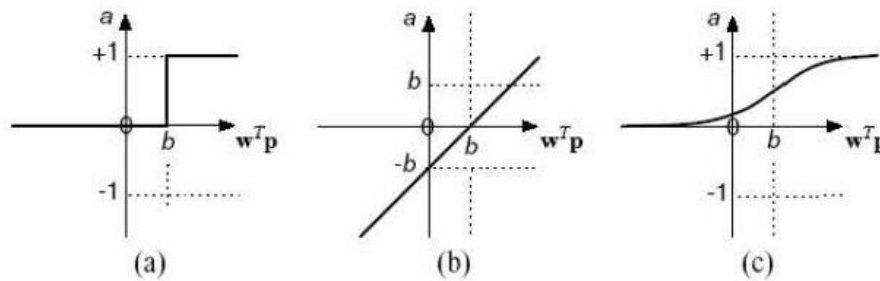


Fig. 3.4. Les fonctions les plus utilisées adaptées en ingénierie.

Comme son nom l'indique, une fonction de seuil applique un seuil à son entrée. Plus précisément, la fonction renvoie la valeur 0 (faux) pour les entrées négatives qui ne dépassent pas le seuil, et 1 (vrai) pour les entrées positives ou vides. Il est évident que ce genre de fonction permet de prendre des décisions binaires.

Une fonction linéaire est simple, elle affecte directement son entrée à sa sortie selon la relation $n = f(n) = n$. Il est clair que la sortie d'un neurone correspond à son niveau d'activation, qui passe par zéro (l'ordonnée de l'origine) lorsque $W.p = b$

La fonction de transfert sigmoïde est q . Les deux sont définis par la relation mathématique :

Cela ressemble à une fonction de seuil ou à une fonction linéaire, selon que nous sommes loin ou près de b , respectivement. La fonction de seuil est très non linéaire car il y a une discontinuité lorsque $W.p = b$. Les fonctions linéaires, en revanche, sont complètement linéaires. Il n'inclut aucun changement de pente. Sigmoïde est un compromis intéressant entre les deux premiers. Enfin, notez que la fonction tangente hyperbolique est une version symétrique de la sigmoïde [114].

3.1.9. Architectures d'un réseau de neurones

Généralement, les réseaux de neurones se composent de trois types de couches :

- **La couche d'entrée :** C'est la première couche. Il recevra les données sources que nous voulons utiliser pour l'analyse. Sa taille est directement déterminée par le nombre de variables d'entrée.

- **La couche cachée** : c'est la couche suivante après la couche d'entrée, nous pouvons avoir une ou plusieurs couches cachées dans le réseau. Nous notons que si la fonction de transfert est linéaire, nous traitons toutes les couches cachées comme une seule couche. Cependant, la fonction d'activation est généralement non linéaire à cette couche, mais n'obéit pas aux règles. Le choix de sa taille n'est pas implicite et doit être ajusté. En général, on peut commencer avec des couches d'entrée et de sortie de taille moyenne, mais ce n'est pas toujours le meilleur choix. Il est généralement préférable d'obtenir de bons résultats et d'en essayer autant que possible.
- **La couche de sortie** : C'est la troisième couche. Il donne le résultat obtenu après que les données entrées dans la première couche ont été compilées par le réseau. Sa taille dépend directement par le nombre de variables que nous voulons en sortie [114].

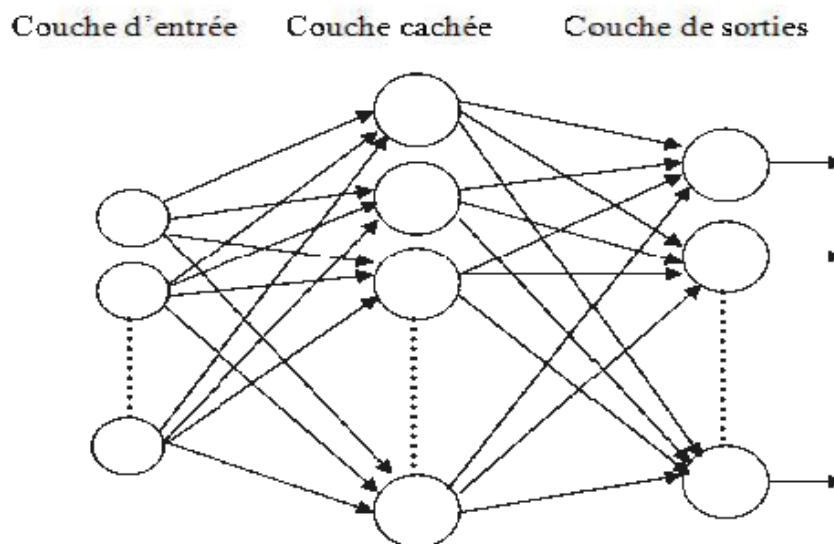


Fig. 3.5. Architecture d'un réseau de neurones.

3.1.10. Apprentissage

L'apprentissage est une étape très importante dans le développement des réseaux de neurones, où le comportement du réseau est modifié de manière itérative jusqu'à l'obtention du comportement souhaité, ce qui est obtenu en ajustant les poids (connexions ou synapses) des neurones à des informations sources bien définies. L'apprentissage comprend également l'extraction de la cohérence entre les données utilisées pour la formation du réseau, mais l'objectif fondamental de l'apprentissage est de résoudre des problèmes par la prédiction, la classification, etc.

Pour les réseaux de neurones artificiels, l'apprentissage peut également être considéré comme le processus de mise à jour des poids (connexions) dans le réseau pour ajuster la réponse du réseau à l'expérience et aux exemples.

Il existe de nombreux types de règles d'apprentissage, qui peuvent être divisées en deux catégories, à savoir l'apprentissage supervisé et l'apprentissage non supervisé.

❖ **Apprentissage supervisé** : Dans ce type d'apprentissage, on essaie d'imposer une fonction donnée au réseau en forçant la sortie du réseau à prendre une valeur donnée (choisie par l'opérateur) et en modifiant les poids synaptiques.

Le réseau se comporte alors comme un filtre dont les paramètres de transfert sont ajustés en fonction du couple entrée-sortie présenté.

L'adaptation des paramètres du réseau est effectuée par un algorithme d'optimisation, et l'activation des poids synaptiques est généralement aléatoire. Des exemples d'utilisation de différentes règles d'apprentissage, architectures et algorithmes associés à ce type de supervision sont présentés dans le tableau 3.2.

Tableau. 3.2. Exemple de différents types d'apprentissages en mode supervisé.

Paradigme	Architecture	Règle d'apprentissage	Algorithme	Taches
Supervisé	Perception Simple ou Multi-Couches	Correction d'erreur	Perceptron, Rétro-Propagation, Adaline, Madaline	Classification, Approximation de fonctions, Prédiction, Contrôle
	Récurrente	Boltzmann	Apprentissage de Boltzmann	Classification
	Multi-Couches non bouclés	Hebb	Analyse de discriminants linéaires	Analyse de données, Classification
	à compétition	Par compétition	LVQ	Catégorisation au sein d'une classe, Compression de données
	ART		ARTMap	Classification, Catégorisation au sein d'une classe

❖ **Apprentissage non supervisé** : Contrairement à l'apprentissage supervisé, cet apprentissage non supervisé est aussi appelé « apprentissage compétitif » où seules les valeurs d'entrée sont disponibles. Dans ce cas, les exemples présentés à l'entrée amènent le réseau à générer de manière adaptative des valeurs de sortie proches en réponse à des valeurs d'entrée similaires.

Des exemples d'utilisation de différentes règles d'apprentissage, architectures et algorithmes associés à ce type de supervision sont présentés dans le tableau 3.3 [115].

Tableau. 3.3. Exemple de différents types d'apprentissages en mode non supervisé.

Paradigme	Architecture	Règle d'apprentissage	Algorithme	Taches
Non supervisé	Multi-couches non bouclé	Correction d'erreur	Projection de Sammon	Analyse de données
	Non bouclé ou à compétition	Hebb	Analyse en composantes	Analyse de données,
	à compétition	Par compétition	VQ	Catégorisation, Compression de données
	Carte de Kohonen		SOM	Catégorisation, Analyse de données
	ART		ART-1, ART-2	Catégorisation

3.1.11. Modèles des réseaux de neurones artificiels

Pour engendrer un réseau de neurones, nous devons mettre des connexions entre les neurones. Nous avons quatre principaux types de connexions : les connexions directes, les connexions récurrentes, les connexions latérales et les connexions différées. Tous les réseaux de neurones utilisent des connexions directes pour transmettre les informations de l'entrée à la sortie. Des connexions récurrentes permettent de transmettre des informations depuis les sorties des neurones de la couche supérieure vers les entrées des neurones précédents.

Les réseaux de neurones qui doivent sélectionner le neurone gagnant utilisent des connexions latérales pour établir la relation entre les neurones de sortie et la maintenir. Enfin, le problème de synchronisation est résolu par un modèle de réseau dynamique avec des connexions temporisées. Les connexions entre les neurones

peuvent être complètes ou partiellement complètes. Quand les neurones de la couche inférieure sont reliés aux neurones de la couche supérieure, la connexion est complète, et quand les neurones des deux couches ne sont pas complètement reliés, la connexion est locale.

Une couche est définie comme un groupe de neurones situé au niveau d'un réseau de neurones. Par exemple, nous avons une couche de neurones de sortie et une couche entre l'entrée-sortie appelée la couche cachée. Les réseaux de neurones ont une ou plusieurs couches de neurones, et leur taille dépend du nombre de couches et du nombre de neurones dans chaque couche [116].

❖ **Perceptron**

Multi-Layer Perceptron (MLP) en anglais, est le réseau de neurones le plus utilisé pour les fonctions d'approximation, de diagnostic et de prédiction. Son architecture est conçue pour que le flux de données voyage dans une seule direction (de l'entrée à la sortie) et à travers les couches intermédiaires du réseau. Ce modèle utilise généralement des règles d'apprentissage supervisé. L'algorithme d'apprentissage le plus couramment utilisé ou employé pour MLP est la rétro propagation des gradients, où les informations sont propagées dans des directions opposées pour s'adapter aux paramètres libres.

- **Perceptron simple** : Un cas particulier de MLP est un perceptron simple. Comme le montre la figure 3.6, un perceptron simple est une architecture constituée d'une seule couche de neurones dont la fonction d'activation est de type «pas unitaire». Tous les neurones d'un perceptron simple sont alimentés par D éléments de la couche d'entrée, où D est également la dimension du vecteur de poids W associé au neurone. L'entrée sera traitée par les neurones et présentée à la fonction d'activation de chaque neurone. Cette fonction va déterminer l'état d'activation du neurone et présenter le résultat sur la sortie S .

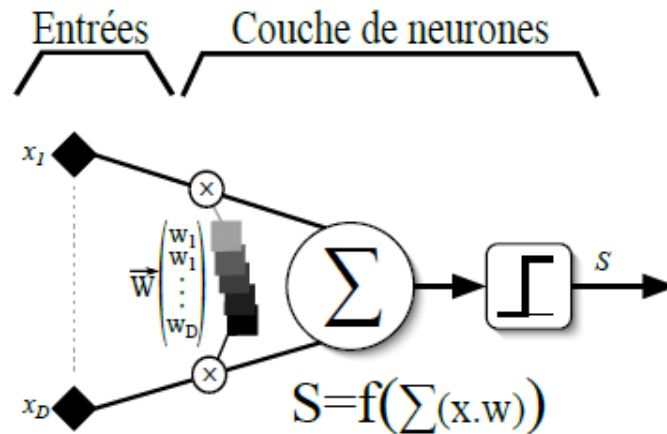


Fig. 3.6. Schéma d'un Perceptron Simple.

Malgré la simplicité de cette architecture et de l'algorithme utilisé, les perceptrons simples sont très utiles pour les fonctions d'ordre linéaires. Cependant, des difficultés surgissent si les deux classes sont linéairement inséparables. Dans ce cas, des couches supplémentaires de neurones doivent être ajoutées. Par conséquent, le perceptron multicouche se présente comme une solution [112].

- **Perceptron multicouches :** (Modèle utilisé dans cette étude) Un perceptron multicouche (PMC) est un ensemble de plusieurs perceptrons simples connectés les uns aux autres en N couches (Figure 3.7). Selon cette implémentation, Le flux de données ne se déplace que dans un seul sens (entrée vers sortie). Les neurones d'une couche sont fournis par les sorties des couches adjacentes. Il existe deux types de calques selon leur disposition :

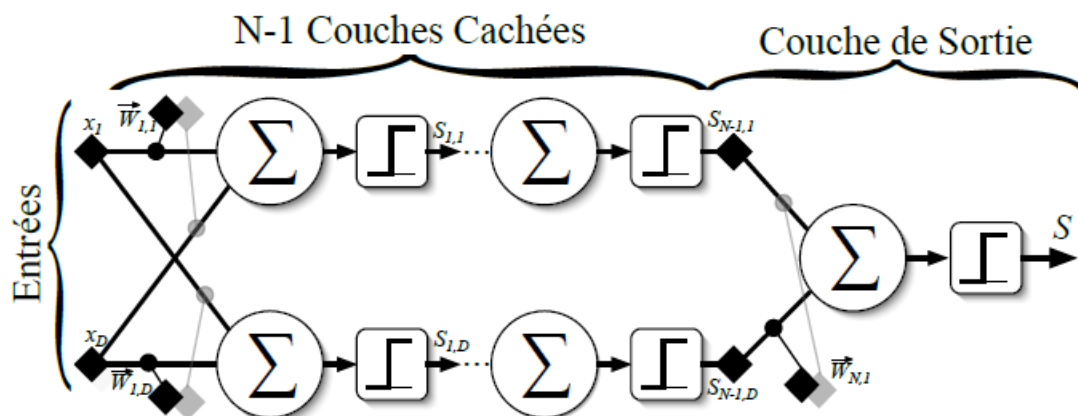


Fig. 3.7. Schéma de principe d'un PMC.

La couche de sortie est souvent appelée couche de décision. C'est la couche la plus à droite. Au niveau de cette couche, le résultat final du traitement est récupéré. Par

conséquent, le nombre de neurones sur la couche de sortie dépend du résultat souhaité. Cette couche peut être rendue par un simple perceptron. Entre autres choses, il peut être présenté par un autre modèle neuronal appelé ADALINE. Ce modèle a la même architecture qu'un perceptron simple, sauf que les neurones utilisés par ce modèle ont des fonctions d'activation linéaires.

Couche cachée ou couche de prétraitement. Ces couches sont situées entre la couche d'entrée et la couche de sortie. Ce qu'ils font, c'est utiliser principalement des fonctions d'activation non linéaires dans les neurones pour préparer les données à présenter à la couche de sortie. Pour que cela fonctionne, les couches doivent être de type perceptron simple, et non de type ADALINE. Car, en combinant plusieurs fonctions linéaires, on obtient une fonction linéaire. Le nombre de neurones sur chaque couche cachée et le nombre de couches cachées dépend du traitement à effectuer et du comportement souhaité du réseau [112].

❖ Réseau RBF

Les réseaux RBF sont des réseaux hiérarchiques dérivés d'une technique d'interpolation appelée méthodes d'interpolation RBF. Le réseau est constitué d'une couche cachée dont la fonction d'activation est appelée fonction noyau ou fonction gaussienne, et d'une couche de sortie avec une fonction d'activation linéaire. La particularité de la méthode RBF est sa réponse utile aux plages restreintes. La réponse de la fonction noyau est la plus grande au niveau du noyau et diminue généralement de manière monotone avec la distance existant entre le vecteur d'entrée et le centre de la fonction noyau. Pour approximer un comportement donné, les fonctions de base sont assemblées pour couvrir toutes les données d'entrée à travers leurs champs récepteurs. Ces fonctions sont ensuite pondérées et leurs valeurs additionnées pour produire la valeur de sortie. Les réseaux RBF sont capables de calculs puissants. Par rapport aux réseaux multicouches, l'apprentissage est plus rapide et plus simple, mais nécessite un grand nombre de neurones. De plus, ils sont beaucoup moins sensibles à la destruction du poids corporel. Leurs champs d'application sont très larges : traitement d'images, reconnaissance de formes, et surtout problèmes de classification [116].

❖ Réseau Hopfield

Un réseau Hopfield implémente une mémoire adressable par son contenu. C'est un réseau de neurones de McCulloch et Pitts (à deux états, -1 et 1 ou 0 et 1), Sa règle d'apprentissage est la règle de Hebb (1949), qui stipule qu'une synapse augmente son activité si et seulement si l'activité de ses deux neurones est courante (c'est-à-dire que lorsque deux neurones sont activés simultanément, la différence entre les neurones i et j est Le poids des connexions w augmenter au même temps) [115].

❖ Les réseaux de Kohonen

Contrairement aux réseaux Hopfield, qui modélisent les neurones de la manière la plus simple, ces réseaux s'inspirent d'observations biologiques du fonctionnement du système nerveux sensoriel des mammifères. Une loi de Hebb modifiée (tenant compte de l'oubli) est utilisée pour l'apprentissage. Cette connexion est renforcée lorsque les neurones connectés sont actifs en même temps, et affaiblie dans le cas contraire. Par conséquent, la loi d'interaction latérale (observée biologiquement) est également simulée. Les neurones très proches (physiquement) interagissent positivement (le poids de la connexion augmente lorsque les synapses sont activées), négativement pour les neurones ultérieurs, et pas du tout pour les neurones distants. Les réseaux Kohonen ont des applications dans la classification, le traitement d'images, l'aide à la décision et l'optimisation [115].

❖ Réseau de neurone récurant

Contrairement aux MLP, les réseaux de neurones récurrents (RNN), comme le montre la figure 3.7, ont des cycles dans le graphe neuronal (Elman, 1990). La principale motivation derrière cette architecture est de pouvoir manipuler une séquence de vecteurs d'entrée, chaque vecteur représentant un événement temporel, et pas seulement des données isolées sans signification temporelle. En déroulant la modélisation compacte des RNN par rapport au temps (voir Figure 3.8), ce type de réseau peut ainsi être vu comme une série temporelle de réseaux MLP reliés entre eux par leurs couches cachées respectives. Cette liaison permet au RNN de coder les dépendances latentes entre les événements de la séquence vectorielle d'entrée.

Selon cette modélisation, le RNN prend une séquence d'événements en entrée $x = (x_1, x_2, \dots, x_T)$ et définit la séquence d'états cachés $h = (h_1, h_2, \dots, h_T)$ pour produire la séquence de vecteurs de sortie $y = (y_1, y_2, \dots, y_T)$ en itérant de $t = 1$ à T :

$$h_t = H(w_{xh}x_t + w_{hh}h_{t-1} + b_h) \quad (3.1)$$

$$y_t = w_{hy}h_t + b_y \quad (3.2)$$

Où T est le nombre total de vecteurs d'entrée, $w_{\alpha\beta}$ est la matrice de poids entre les couches α et β et b_β est le vecteur de biais de la couche β . La fonction H utilisée dans le cas des RNN est généralement la tangente hyperbolique [117].

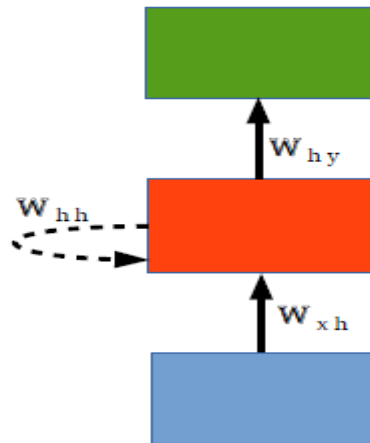


Fig. 3.8. Une représentation compacte d'un RNN. Toutes les flèches représentent des connexions complètes. Les flèches en pointillé indiquent les connexions avec un décalage dans le temps (t -1).

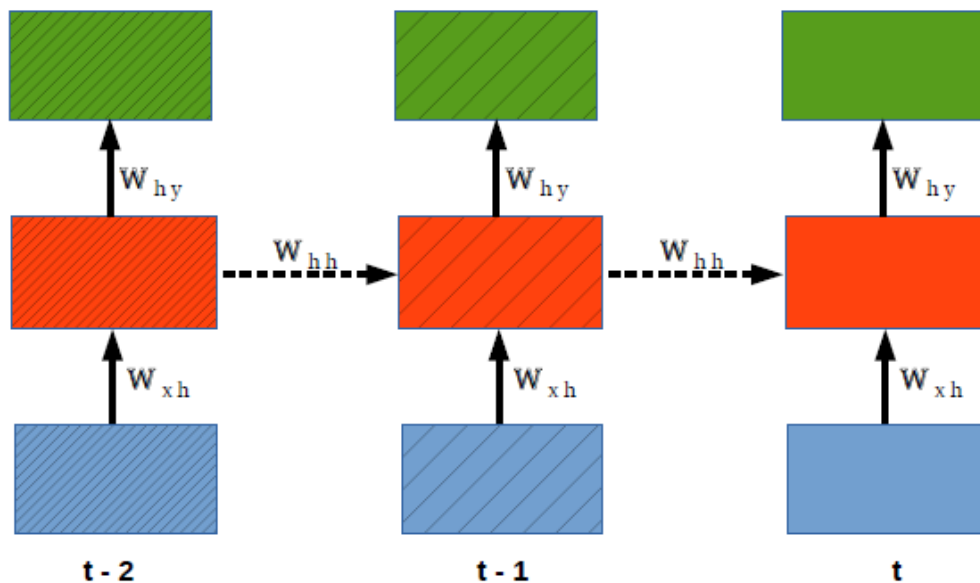


Fig. 3.9. Une représentation élargie d'un RNN.

❖ Deep Learning

Dans les algorithmes classiques d'apprentissage statistique, une fois les données collectées, la première difficulté est de trouver des descripteurs pertinents pour représenter les données et contenir des informations utiles pour la tâche souhaitée. Ainsi, pour chaque modèle considéré, plusieurs types de descripteurs peuvent être investigués jusqu'à trouver une description satisfaisante des données. L'utilisation de descripteurs géométriques ou de moments statistiques est un moyen courant d'obtenir des descripteurs.

Le risque est que le jeu de descripteurs soit incomplet, ou au contraire redondant. Un autre problème concerne la collecte des données, dont la qualité peut varier. De plus, les échantillons de la base d'apprentissage doivent être représentatifs des données sur lesquelles le modèle est construit.

Les réseaux de neurones artificiels correspondent à des associations dans un graphe d'objets élémentaires appelés neurones formels. L'architecture du graphe (par exemple les couches), la complexité (par exemple la présence de boucles de rétroaction), les fonctions d'activation des neurones (par exemple sigmoïde) sont des exemples standard de distinction des réseaux de neurones. Il est possible de distinguer l'entrée d'un neurone en tant que dendrites. Les connexions avec d'autres neurones sont considérées comme des synapses, les fonctions d'activation sont considérées comme des noyaux et les sorties sont activées en fonction des stimuli entrant et sortant des neurones, une analogie avec les réseaux neuronaux biologiques peut être établie comme axone.

L'apprentissage en profondeur (par opposition à l'apprentissage superficiel) est un apprentissage effectué sur un réseau de neurones à plusieurs couches cachées. Le principe du Deep Learning est un apprentissage couche par couche basé sur un apprentissage hiérarchique. Entre chaque couche, une transformation non linéaire se produit et chaque couche reçoit la sortie de la couche précédente en entrée. Dans le Deep Learning, l'extraction des descripteurs est directement pilotée à partir des données [118]. En d'autres termes, l'apprentissage en profondeur repose donc sur un paradigme d'apprentissage que l'on peut qualifier de "supervision des entrées" - où la sortie attendue du modèle est l'entrée elle-même. Dans ce paradigme, l'apprentissage

dépend d'une fonction de coût (comme dans l'apprentissage supervisé) sans fournir de données de sortie au modèle (comme dans l'apprentissage non supervisé).

Les informations contenues dans les données peuvent être représentées de différentes manières. Par exemple, une image peut être encodée sous la forme d'un vecteur de valeurs d'intensité pour chaque pixel, ou d'un ensemble de contours, de régions ayant une forme spécifique. Certaines représentations permettent un meilleur apprentissage de certaines tâches à partir d'exemples. L'un des avantages du Deep Learning est le remplacement de la détermination manuelle des descripteurs par des algorithmes qui extraient des descripteurs hiérarchiques [119].

Il existe plusieurs façons de construire des réseaux de neurones profonds, dont le DBN (Deep Belief Network). Le moyen le plus populaire de former efficacement des réseaux de neurones profonds est l'apprentissage couche par couche via des machines Boltzmann restreintes utilisant des algorithmes gloutons (des algorithmes qui recherchent progressivement des minima locaux). Plus précisément, il s'agit d'un problème d'apprentissage de chaque couche de manière non supervisée afin d'extraire les principaux descripteurs de la distribution des données d'entrée. La première couche cachée correspond donc à la représentation de ces entrées. Cette représentation est ensuite utilisée comme entrée pour la couche suivante.

Les méthodes d'apprentissage profond peuvent être utilisées comme initialisation des poids et des biais avant d'utiliser des algorithmes supervisés tels que la rétro-propagation de gradient (cette méthode calcule le gradient de l'erreur pour chaque neurone du réseau de neurones, de la dernière couche à la première couche. En apprenant une profondeur réseau, la rétro-propagation joue alors le rôle de fine-tuning). L'apprentissage de stratégies utilisant des réseaux aussi profonds est assez efficace. Il a été montré que l'utilisation d'un réseau profond (Deep Belief Network, ou type DBN) pour initialiser les poids d'un perceptron multicouche donne de meilleurs résultats qu'une initialisation aléatoire. Il existe plusieurs modèles d'architecture de Deep Learning, en trouve : Auto-encodeur, CNN, LSTM, ... [120].