

**Chapitre IV**  
**OUTILS ARCHITECTURAUX POUR LES RESEAUX DE**  
**NEURONES**

## **Outils Architecturaux pour les Réseaux de Neurones**

### **IV.1 Introduction**

L'Intelligence Artificielle est une approche visant à émuler, dans des systèmes artificiels, les capacités des espèces vivantes pouvant identifier l'environnement et prendre les décisions adéquates en présence d'incertitudes et d'imprécisions. Les réseaux neuronaux sont des techniques qui disposent aujourd'hui d'une théorie bien fondée et qui ont été efficacement utilisées dans divers domaines [14]. Les réseaux de neurones formels sont des structures la plupart du temps simulés par des algorithmes exécutés sur les ordinateurs d'usage général, parfois sur des machines ou même des circuits spécialisés, qui prennent leurs inspirations à partir du fonctionnement élémentaire des systèmes nerveux. Ils sont utilisés essentiellement pour résoudre des problèmes de classification, de reconnaissance de formes, d'association, d'extraction de caractéristique, et d'identification, etc. [47, 48].

Ces tâches sont accomplies avec une grande rapidité, et le réseau adapte son comportement aux situations nouvelles, principalement selon deux processus, l'apprentissage, et l'adaptation rapide et automatique [49].

### **IV.2 Historique**

Classiquement, l'histoire commence ainsi : en 1943 dans un article resté fameux, MC CULLOCH et PITTS ont émis l'idée simplificatrice du neurone formel, c'est-à-dire une opération binaire interconnectée à ses semblables par des « synapses ». En 1949, D.O. HEBB, dans un livre resté lui aussi fameux introduisit la notion de « plasticité synaptique », c'est-à-dire le mécanisme de modification progressive des couplages inter-neurones responsables du changement permanent de leurs propriétés collectives, ce que l'on peut appeler « l'apprentissage ». Les travaux sur les réseaux de neurones formels ont connu des débuts prometteurs vers la fin des années 50. C'est en 1958 que ROSENBLATT décrivit le premier modèle opérationnel de réseaux de neurones, mettant en œuvre les idées de HEBB, MC CULLOCH et PITTS [47].

En 1969, MINSKY et PEPART, démontrèrent les limites théoriques du perceptron, qui est capable d'apprendre à calculer certaines fonctions logiques, mais un certain manque d'approfondissement de la théorie et le mouvement concurrent de l'intelligence artificielle, ont conduit à une mise en sommeil assez prolongée des travaux sur le connexionnisme. Assez peu d'articles furent publiés entre 1970 et 1980, avec un peu de ralentissement en raison du faible nombre de chercheurs intéressés. Ils ont cependant permis une certaine continuité et surtout ont servi de germes à une renaissance frappante de l'activité dans le début des années 80. S'il faut citer des travaux de cette période, c'est celui de HOPFIELD en 1982, qui détaille dans un article le bon usage du réseau de neurones formels, en s'appuyant sur l'organisation en couches et la simulation numérique sur ordinateur et ceux de WIDROW et HOFF sur les algorithmes adaptatifs [48]. En 1986, Rumelhart, Hinton et Williams publient, l'algorithme de la rétro-propagation de l'erreur qui permet d'optimiser les paramètres d'un réseau de neurones à plusieurs couches. À partir de ce moment, la recherche sur les réseaux de neurones a connu un ressort fulgurant et les applications commerciales de ce succès académique ont suivie au cours des années 90. Actuellement, les chercheurs s'intéressent beaucoup plus à l'exploitation des structures d'identification, et le contrôle de la commande adaptative avec les réseaux de neurones [48].

### **IV.3 Modèle d'un neurone**

#### **IV.3.1 Modélisation biologique**

Un neurone biologique est constitué, d'un corps cellulaire appelé soma, de plusieurs prolongements d'entrée appelés dendrites et un prolongement de sortie appelé axone. Les interactions entre les neurones se font au niveau du corps cellulaire ou dans les dendrites à des points de jonctions appelés synapses [47].

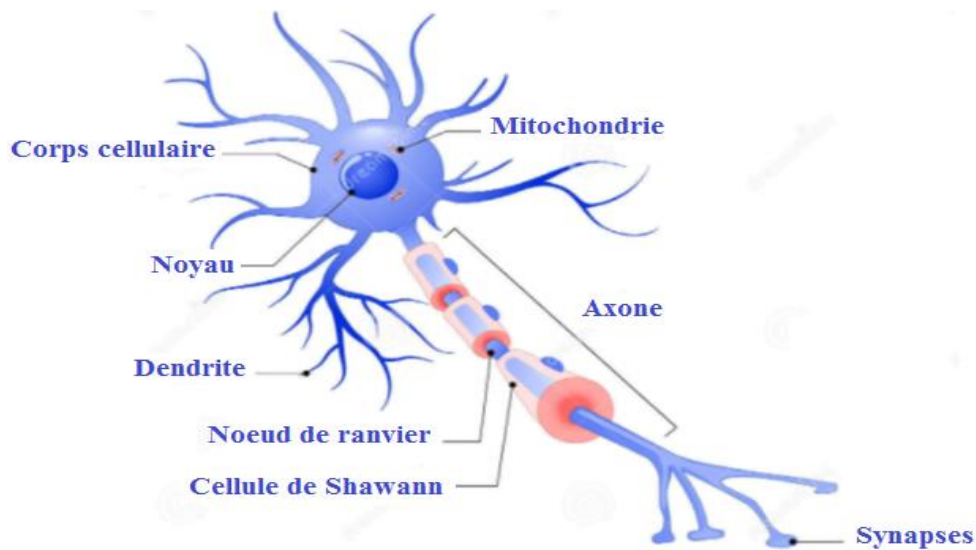


Fig. IV.1 : Anatomie d'un neurone

### IV.3.2 Le neurone formel

Le premier modèle du neurone formel date des années quarante. Il a été présenté par McCulloch et Pitts. S'inspirant de leurs travaux sur les neurones biologiques ils ont proposé le modèle suivant :

Un neurone forme une somme pondérée des potentiels d'action qui lui parviennent. Puis s'active suivant la valeur de cette sommation pondérée. Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien [47,48]. Conceptuellement, ceci peut être représenté par le modèle de la Figure (IV.1).

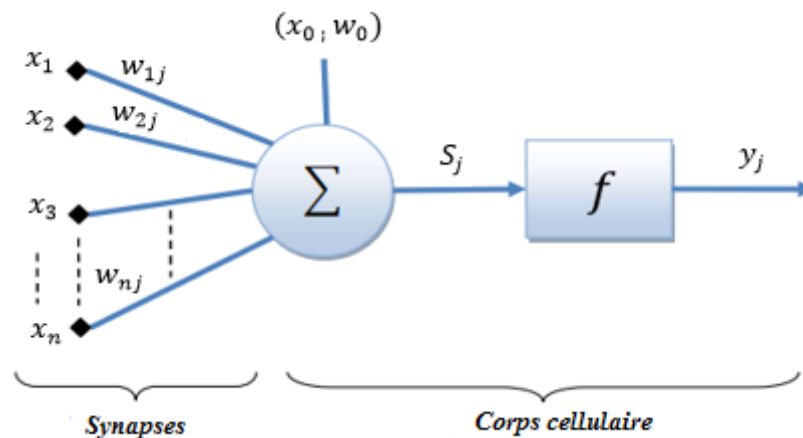


Fig. IV.2 : Schéma Bloc du modèle d'un neurone formel.

Le tableau ci-dessous tableau (IV.1) [50], nous permet de voir clairement la transition entre le neurone biologique et le neurone formel.

Neurone biologique	Neurone artificiel
Synapses	Poids de connexion
Axone	Signal d'entrée
Dendrite	Signal de sortie
somma	Fonction d'activation

**Tableau IV.1 :** Transition entre le neurone biologique et le neurone formel.

Le neurone formel peut être considéré comme un opérateur (ou un élément processeur) qui collecte les entrées et fournit une sortie seulement quand la somme dépasse un certain seuil interne. L'évaluation de la sortie se fait typiquement par la somme pondérée des entrées, soustraction du seuil et le passage du résultat à travers une non-linéarité. Mathématiquement, ceci peut être modélisé par les équations suivantes :

$$S_j = \sum_{i=0}^n w_{ij} \cdot x_i \quad (4-1)$$

$$y_j = f(S_j) \quad (4-2)$$

$x_i$  et  $w_{ij}$  représentent respectivement les composantes du vecteur d'entrée et du vecteur des poids synaptiques. Le terme  $(x_0; w_0)$  représente la valeur du seuil interne qui doit être dépassée pour l'activation de la sortie du neurone. La non linéarité  $f(.)$  désigne la fonction d'activation du neurone.

#### IV.4 Fonctions d'activation

Différentes fonctions de transfert pouvant être utilisées comme fonction d'activation du neurone sont énumérées dans le tableau (2). Les fonctions d'activations les plus utilisées sont les fonctions «seuil», «linéaire» et «sigmoïde». Comme son nom l'indique, la fonction seuil applique un seuil sur son entrée. Plus précisément, une entrée négative ne passe pas le seuil, la fonction retourne alors à la valeur 0, on peut interpréter ce 0 comme signifiant faux, alors qu'une entrée positive ou nulle dépasse le seuil, et la fonction

retourne à 1 (vrai). La figure (IV.2) donne quelques modèles de fonctions d'activation utilisées [50].

Nom de la fonction	Relation entrée/sortie	Icône	Nom MATALB
Seuil	$y = 0$ si $s < 0$ $y = 1$ si $s \geq 0$		hardlim
Seuil symétrique	$y = -1$ si $s < 0$ $y = 1$ si $s \geq 0$		hardlims
Linéaire	$y = s$		purelin
Linéaire saturée	$y = 0$ si $s \leq 0$ $y = s$ si $0 \leq s \leq 1$ $y = 1$ si $s \geq 1$		satlin
Linéaire saturée symétrique	$y = -1$ si $s < -1$ $y = s$ si $-1 \leq s \leq 1$ $y = 1$ si $s > 1$		satlins
Linéaire positive	$y = 0$ si $s \leq 0$ $y = s$ si $s \geq 0$		poslin
Sigmoïde	$y = \frac{1}{1 + \exp^{-s}}$		logsig
Tangente hyperbolique	$y = \frac{e^s - e^{-s}}{e^s + e^{-s}}$		tansig
Compétitive	$y = 1$ si $s$ maximum $y = 0$ autrement		compet

**Tableau IV.2 :** les modèles de fonctions d'activations

Ce vaste choix de fonctions permet aux modèles neuronaux de posséder des caractéristiques très variées. Le choix de la fonction d'activation dépend des types d'automates utilisés, c'est-à-dire tout dépend de la nature des entrées et des sorties [51].

#### IV.4.1. Les automates à seuil

- La sortie est binaire alors, que les entrées peuvent être binaires ou réelles.
- La fonction d'activation est la fonction de Heaviside.

#### IV.4.2 Les automates à saturation

- Les entrées et les sorties sont comprises dans un intervalle  $[u, v]$ .
- La fonction d'activation est la fonction saturée.

#### IV.4.3 Les automates continus

- Les entrées et les sorties sont réelles.
- La fonction d'activation est stochastique.

#### IV.4.4 Les automates probabilistes

- Les sorties sont binaires et les entrées sont quelconques.
- La fonction d'activation est stochastique.

### IV.5 Réseaux de Neurones Artificiels

L'intérêt de ces cellules de traitements simples apparaît lorsqu'on les agence sous forme d'un réseau. L'idée générale d'un réseau de neurones artificiel ( le NN dans le langage de programmation MATLAB) est de trouver une nouvelle procédure de calcul pour produire une nouvelle génération d'ordinateur. Cette idée, inspirée de la biologie du cerveau humain, est différente de la méthode conventionnelle utilisée sur les ordinateurs. Un réseau de neurones peut être considéré comme une boîte noire, qui reçoit des signaux d'entrée et produit des signaux de sortie, c'est un modèle mathématique composé d'un grand nombre d'éléments de calcul organisés sous forme de couches interconnectées. D'autre définition sont donnés comme suit : Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur calcule une sortie unique sur la base des informations qu'il reçoit. Toute structure hiérarchique de réseaux est évidemment un réseau [50].

### IV.5.1. Topologie des Réseaux de Neurones

On peut classer les réseaux de neurones en deux grandes catégories, selon la dépendance de l'évolution de ces réseaux en fonction explicite du temps.

#### IV.5.1.1. Les réseaux statiques ou réseau à couches (Feed Forward)

C'est le cas de réseaux statiques, ou le temps n'est pas un paramètre significatif. En d'autres termes, la modification d'entrée n'entraîne que la modification stable de la sortie, mais elle n'entraîne pas le retour de l'information de cette entrée, c'est-à-dire qu'il n'y a pas de rebouclage au niveau d'un neurone.

Les réseaux statiques (FEED FORWARD) sont des réseaux à couches, constitués d'une couche d'entrée, une couche de sortie et entre les deux au moins une couche composée de nombreux éléments de traitements non linéaires, appelée couche cachée. Les signaux des entrées se propagent de la première couche à la couche de sortie en passant par les couches cachées, Il n'y a pas de communication entre les unités de la même couche, d'où le nom de feed forward. Les liens dirigés connectant les neurones sont appelés les interconnexions [52].

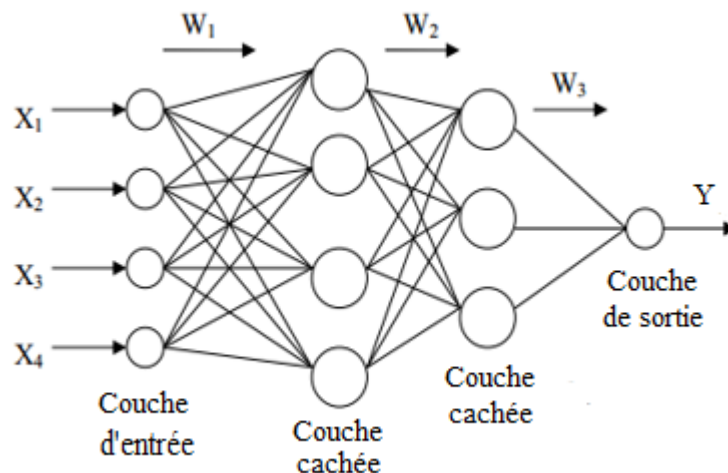
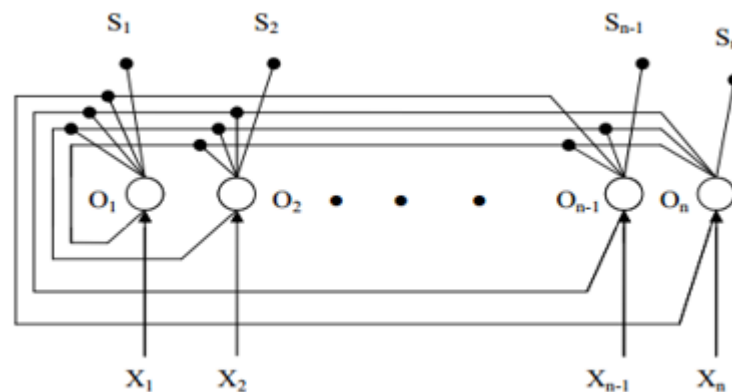


Fig. IV.3 : Structure d'un réseau statique avec deux couches cachées.

#### IV.5.1.2. Les réseaux dynamiques (récurrents)

Comme leur nom l'indique, contiennent des rebouclages partiels ou totaux entre neurones, ils représentent donc une évolution dépendante du temps. Il faut bien distinguer

l'état du réseau à un certain instant de son état, ou aux instants précédents, et de distinguer le temps nécessaire à obtenir une réponse, dans le cas d'une réalisation matérielle ou d'une simulation sur ordinateur. La théorie de ces réseaux fut publiée par HOPFIELD dans les années 1980 et sont souvent appelées « HOPFIELD nets ».



**Fig. IV.4 :** Réseau de neurones de Hopfield

Le perceptron multicouche ordinaire est un réseau statique. Par contre, le réseau du HOPFIELD ou le perceptron avec rebouclage est un réseau dynamique. Les critères motivant les choix d'un type de réseau sont la simplicité de mise en œuvre et l'efficacité des algorithmes d'adaptation appelés à répondre aux performances désirées du système, quelle que soit sa complexité. L'opération réalisée par un réseau, bouclé ou non, dépend des valeurs des coefficients ( $W_i$ ) de pondération du réseau. Pour qu'un réseau effectue une tâche donnée, il faut donc ajuster la valeur de ses poids. Une tâche est définie par un ensemble d'exemples, ou couples (valeurs des entrées et valeurs des sorties désirées correspondantes), ces couples constituent l'ensemble d'apprentissage qui représente une propriété du réseau de neurones.

#### IV.5.2 Propriétés des réseaux de neurones

Ce qui rend les réseaux de neurones différents des méthodes classiques dans la quête de la robustesse, c'est qu'ils sont fondamentalement distingués de quatre points principaux [47] :

#### IV.5.2.1. Parallélisme

Le caractère distribué, parallèle et simultané du traitement par le réseau, offre des avantages de robustesse par rapport à des données incertaines incomplètes, et même par rapport à des défauts locaux du réseau.

#### IV.5.2.2. La généralisation

L'intérêt des réseaux de neurones réside précisément dans leur capacité à la généralisation, c'est-à-dire leur aptitude de bien se comporter vis à vis des vecteurs qui n'ont pas été appris. La capacité de généralisation d'un réseau de neurones est son aptitude de donner une réponse satisfaisante à une entrée qui ne fait pas partie des exemples à partir desquels il a appris.

#### IV.5.2.3. Capacité d'adaptation

C'est la capacité d'apprentissage du réseau, elle se manifeste par leurs capacités d'auto-organisation en adaptant ces poids synaptiques par une méthode d'apprentissage qui assure leurs stabilités.

#### IV.5.2.4 La résistance aux pannes

Les données bruitées ou les pannes locales dans un certain nombre d'éléments du réseau de neurones n'affectent pas ses fonctionnalités. Cette propriété résulte essentiellement du fonctionnement collectif simultané des neurones qui le composent.

#### IV.5.2.5. Capacité de modélisation

Le plus souvent, le problème qui se pose à l'ingénieur est le suivant, il cherche à établir un « model » du processus (physique, chimique, économique ...) qu'il étudie à partir des mesures dont il dispose, et d'elles seules : on dit qu'il effectue une modélisation (boite noire). Cette modélisation peut être effectuée en utilisant un réseau de neurones comme l'illustre cet exemple.

**Exemple :** Le model neuronal représenté sur la figure (IV. 4), modélise l'équation suivante:  $f(x) = 4 - 0.9 \cdot \sin(7 \cdot x + 0.2) + 8 \cdot \sin(x + 3) - 0.6 \cdot \sin(8 \cdot x + 0.1)$  (4.3)

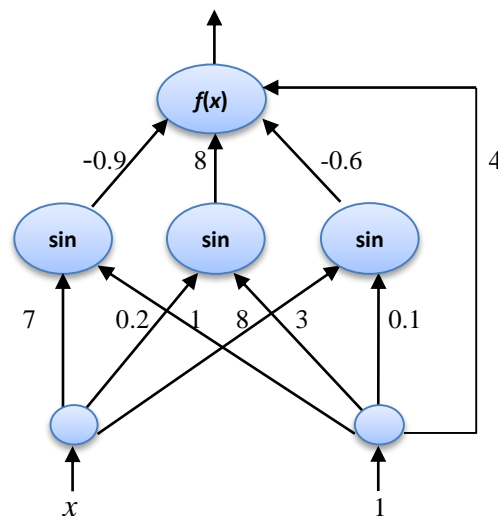


Fig. IV.5 : modélisation par un réseau neuronal

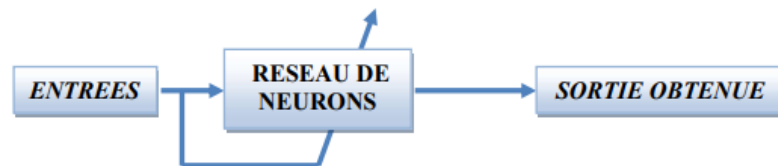
### IV.5.3 L'Apprentissage

La phase d'apprentissage consiste à ajuster les poids du réseau jusqu'à l'obtention d'une stabilisation du réseau. C'est-à-dire, jusqu'à ce que le réseau présente certains comportements désirés, de telle manière que le poids ne se modifie plus que d'une façon minimale [48]. L'apprentissage mathématique, basé sur ce concept, sert à minimiser une fonction de coût formulée autour de l'erreur de sortie. Les procédures d'apprentissage peuvent se subdiviser, elles aussi, en deux grandes catégories [50] :

#### IV.5.3.1 Apprentissage non supervisé

L'apprentissage non supervisé copie le fonctionnement du cerveau humain qui retrouve les informations par association. Cet apprentissage consiste à découper l'ensemble des vecteurs d'entrées en classes d'équivalences, sans qu'il soit nécessaire de donner au réseau neuronal les noms des classes pour chaque exemple. La séparation en classes d'équivalence s'opère par mesure de ressemblance entre les vecteurs proposés. La contrainte que l'on doit imposer au réseau neuronal est le nombre de classes d'équivalence. Pendant la phase d'apprentissage, le réseau neuronal construit une topologie de l'espace des vecteurs d'entrée. Dans ce cas, la connaissance à priori de la sortie désirée n'est pas nécessaire et la procédure d'apprentissage est basée uniquement sur les valeurs d'entrée. Le réseau s'auto-organise de manière à optimiser une certaine fonction de coût, sans qu'on lui présente la réponse désirée. L'apprentissage est accompli

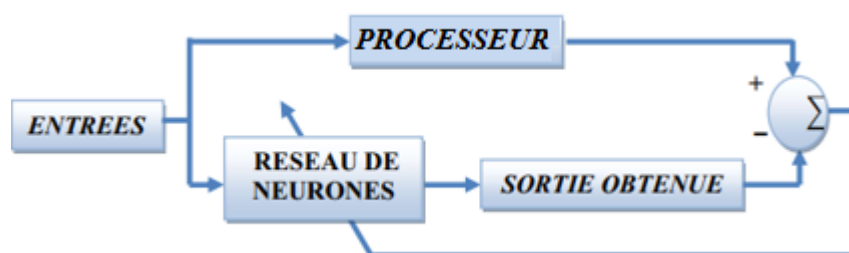
à l'aide de règles qui changent ou adaptent les poids des coefficients synaptiques en fonction des exemples présentés à l'entrée [53]. L'intérêt de cette approche est que l'on n'a pas besoin de disposer d'exemple de problèmes résolus, c'est à dire, de vecteur de sortie correspondant à chaque vecteur d'entrée utilisé pour l'apprentissage.



**Fig IV.6 :** Apprentissage non supervisé

#### IV.5.3.2 Apprentissage supervisé

Dans ce cas, la connaissance a priori de la sortie désirée est nécessaire. On présente au réseau le vecteur d'entrée puis on calcule sa sortie, cette dernière sera comparée avec la sortie désirée, ensuite les poids sont ajustés de façon à réduire l'écart entre elles. Cette procédure est répétée jusqu'à ce qu'un critère de performance soit satisfait. Après la phase d'apprentissage et la convergence des calculs, il n'est plus nécessaire, en général, de mémoriser le jeu d'apprentissage. La méthode appropriée à cet apprentissage est la méthode de rétro-propagation. Cette méthode a donné de bons résultats dans beaucoup d'applications, Domaines de contrôle, identification, diagnostic, etc. [54].



**Fig. IV.7 :** Apprentissage supervisé

#### IV.5.4 Principe de la méthode de rétro- propagation

L'apprentissage supervisé consiste à ajuster les coefficients synaptiques pour que les sorties des réseaux soient les plus proches possibles des sorties de l'ensemble d'entraînement. Donc il faut spécifier une règle d'apprentissage pour l'adaptation de ces

paramètres. Pour remédier à ce problème, on utilise la méthode de rétro-propagation. La rétro propagation est basée sur des principes mathématiques [54, 55]. Pour l'appliquer, il suffit d'avoir des données d'entrées et des données de sortie. Cependant, toute information que possède un réseau de neurones est représentée par les poids d'interconnexions, cette information est acquise durant la phase d'apprentissage.

Le principe de cet algorithme est que, de même que l'on est capable de propager un signal provenant de cellules d'entrée vers la couche de sortie, on peut, en suivant le chemin inverse, rétro-propager l'erreur commise en sortie vers les couches internes du réseau [55, 56].

La loi d'adaptation des poids est donnée par l'équation suivante :

$$W_{ij}^k(t+1) = W_{ij}^k(t) - \mu \frac{\partial E(W)}{\partial W_{ij}^k(t)} + \alpha (W_{ij}^k(t) - W_{ij}^k(t-1)) \quad (4-4)$$

$W_{ij}^k(t)$  : est le poids entre le neurone  $j$  de la couche  $k$  et le neurone  $i$  de la couche  $(k-1)$ .

$t$  : indice d'itération.

$0 \leq \alpha < 1$  : le momentum.

$\mu$  : est le pas d'apprentissage.

Cet algorithme est largement appliqué dans le problème d'identification de paramètres, de commande optimale, etc. [57].

## IV.6. Conclusion

Dans ce chapitre, nous avons présenté les réseaux de neurones, en spécifiant leurs définitions, les concepts de base de cette technique, l'apprentissage des réseaux de neurones, et ses différents domaines d'utilisation. Pour mettre en œuvre un réseau de neurones, cela nécessite que l'on dispose de données suffisamment nombreuses et représentatives pour effectuer l'apprentissage, qui pourrait être émis par le système d'inférence flou. Ainsi l'association ou l'hybridation des techniques neuronales avec la logique floue, et la connaissances disponibles sur le comportement de systèmes à traiter, pourrait améliorer les performances du réseau et permet aussi de réaliser efficacement les tâches désirées. L'étude de cette hybridation fera l'objet du prochain chapitre.