

# Chapitre Trois

## LES RESEAUX DES NEURONES ARTIFICIEL

<b>3.1. Introduction</b>	<b>52</b>
<b>3.2. Eléments de Base</b>	<b>53</b>
<b>3.3. Fonctionnement des Neurones</b>	<b>54</b>
<b>3.4 Réseaux de Neurone Artificiel</b>	<b>55</b>
<b>3.5 Propriétés des Réseaux de Neurones</b>	<b>59</b>
<b>3.6 Topologie des Réseaux de Neurones</b>	<b>60</b>
<b>3.7 Le Perceptron</b>	<b>63</b>
<b>3.8 Perceptron Simple</b>	<b>64</b>
<b>3.9 L'Apprentissage des Réseaux de Neurones</b>	<b>65</b>
<b>3.10 Les Principales Règles d'Apprentissage</b>	<b>66</b>
<b>3.11 Algorithme de Retro propagation</b>	<b>70</b>
<b>3.12 Conclusion</b>	<b>79</b>

## *LES RESEAUX DES NEURONES ARTIFICILS*

### **3.1. Introduction**

Les réseaux des 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.

Classiquement, l'histoire commence ainsi : en 1943 que, 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 » excitatrices ou inhibitrices. Une assemblée de tels opérateurs en interaction devait avoir des propriétés collectives émergentes, c'est-à-dire capables de certain « calcul » que chacun d'eux séparément est capable d'exécuter.

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 neurone responsables de changement permanent de leurs propriétés collectives, ce que l'on peut appeler « l'apprentissage ».

Son hypothèse, limitée à l'augmentation du coefficient de couplage entre deux neurones réels qui sont excités simultanément, a été étendue aux neurones artificiels comme une règle par laquelle les couplages se modifient proportionnellement aux corrélations entre neurones, que ces corrélations soient positives « activation » ou négatives « inhibition ».

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.

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 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 réseaux 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.

En 1986, Rumelhart, Hinton et Williams publient, l'algorithme de la rétropropagation 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 connaît un essor fulgurant et les applications commerciales de ce succès académique suivent au cours des années 90.

Actuellement, les chercheurs s'intéressent beaucoup plus à l'exploitation des structures d'identification et de contrôle mono et multivariable de la commande adaptative avec les réseaux de neurones [14].

## **3.2. Eléments de Base**

### **3.2.1. Modélisation biologique**

Le cerveau humain, est le meilleur modèle de la machine, polyvalente incroyablement rapide et surtout douée d'une incomparable capacité d'auto organisation. Son comportement est beaucoup plus mystérieux que le comportement de ses cellules de base. Il est constitué d'un grand nombre d'unités biologiques élémentaires (1000 à 10000 synapse par neurone)[15].

Les cellules nerveuses, appelées neurones, sont les éléments de base du système nerveux central. Celui-ci en posséderait environ cent milliards. Les neurones possèdent de nombreux points communs dans son organisation générale et leur système biologique avec les autres cellules. Ils présentent cependant des caractéristiques qui leurs sont propres et se retrouvent au niveau de cinq fonctions spécialisées qu'ils assurent [15].

- Recevoir des signaux en provenance de neurones voisins.
- Intégrer ces signaux.
- Engendrer un influx nerveux.
- Le conduire.
- Le transmettre à un autre neurone capable de le recevoir.

### **3.2.2. Structure de neurone**

Un neurone est constitué de trois parties :

#### **3.2.2.1. Le corps cellulaire**

Il contient le noyau du neurone et effectue les transformations biochimiques nécessaires à la synthèse des enzymes et des autres molécules qui assurent la vie de neurone. Sa forme est pyramidale ou sphérique dans les plupart des cas. Elle dépend souvent de sa position dans le cerveau, ce corps cellulaire fait quelques microns de diamètre [16].

#### **3.2.2.2. Les dendrites**

Chaque neurone possède une « chevelure » de dendrites. Celles-ci sont de fines extensions tubulaires, de quelques dixièmes de microns de diamètre et d'une longueur de quelques dizaines de microns. Elles sont les récepteurs principaux du neurone qui servent à capter les signaux qui lui parviennent [16].

#### **3.2.2.3. L'axone**

L'axone, qui est à proprement parler la fibre nerveuse, sert de moyen de transport pour les signaux émis par le neurone. Il se distingue des dendrites par sa forme et par les propriétés de sa membrane externe. En effet, il est généralement plus longue que les dendrites, et se ramifie à son extrémité, là où il communique avec les autres neurones, alors que les ramifications des dendrites se produisent plutôt près du corps cellulaire.

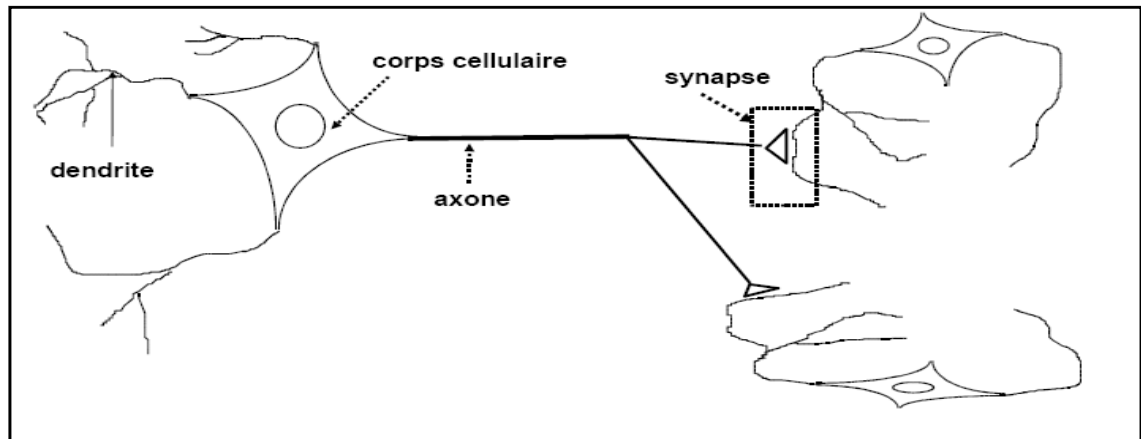
Pour former le système nerveux, les neurones sont connectés les uns aux autres suivant des répartitions spatiales complexes. Les connexions entre deux neurones se font à des endroits appelés synapses ou ils sont séparés par un petit espace synaptique de l'ordre d'un centième de micron [16].

#### **3.2.2.4. Les synapses**

Le rôle des synapses est fondamental pour permettre aux cellules nerveuses de communiquer entre elles. Les signaux qui se propagent dans les neurones sont de nature électrique. Cependant, il n'existe pas de liaisons directes entre deux cellules nerveuses.

Celles-ci sont séparées par un espace appelé fente synaptique que l'influx électrique ne peut traverser.

Le relais s'effectue à ce niveau par l'intermédiaire d'un médiateur chimique [16].



**Figure 3.1**– Schéma du neurone biologique.

### 3.3. Fonctionnement des Neurones

D'une façon générale, le soma de neurone traite les courants électriques (information) qui lui proviennent de ses dendrites et qu'il transmet le courant électrique résultant de ce traitement aux neurones auxquelles il est connecté par l'intermédiaire de son axone.

D'après le modèle classique, le soma effectuant une sommation des influx nerveux transmise par les dendrites, Si la sommation dépasse un seuil, le neurone répond par un influx nerveux au potentiel d'action qui se propage le long de son axone. Si la sommation est inférieure à ce seuil, le neurone reste inactif.

Lorsqu'un potentiel d'action est parvenu au synapse, provoque à travers la membrane de libération d'un médiateur chimique celle-ci se diffuse jusqu'à la membrane du dendrite qui provoque naissance un potentiel [17].

### 3.4. Réseaux de Neurone Artificiel

L'idée générale des réseaux de neurones artificiels et 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.

Cette nouvelle technique semble plus humaine que mécanique. L'homme à toujours rêver de la machine qu'est parle, pense, oublie et se rappelle. Le but n'est pas de produire une machine humaine ce que impossible, mais de copier et imiter l'approche naturelle de cerveau humaine au profite de la science. Un RNA 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 calculs organisée sous forme de couches interconnectées [18].

D'autre définition sont donnés comme suite :

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 [19].

### 3.4.1. Modélisation

La première modélisation d'un neurone date des années quarante. Elle a été présentée par Mac CULLOCH et PITTS. S'inspirant de leurs travaux sur les neurones biologiques, ils ont proposé le modèle suivant : un neurone formel fait une somme pondérée des potentiels d'action qui lui parviennent (chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qui la émis), 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 (sous forme de potentiel d'action) dont la valeur est celle de son activation, si le neurone n'est pas activé il ne transmet rien.

La fonction de transfert est un simple effet de seuil. Alors, si une cellule « i » est reliée à « N » autres cellules, le neurone formel donné par la figure 3.2, effectue une somme pondérée des « N » entrées, augmentée d'une entrée externe désignant le seuil de neurone « i ».

Le tout est suivi d'un élément de décision qui représente la fonction d'activation non linéaire qui détermine la sortie du neurone. La fonction de décision peut avoir les formes les plus utilisées données par la figure 3.3 [20].

Si on désigne par « X » le vecteur d'entrées du neurone « i », «  $W_i$  » représente la force de la connexion simulant les poids synaptiques des neurones, tels que :

$$X=[X_1, X_2 \dots\dots, X_n]$$

$$W_j=[W_{j1}, W_{j2}\dots\dots W_{jn}]$$

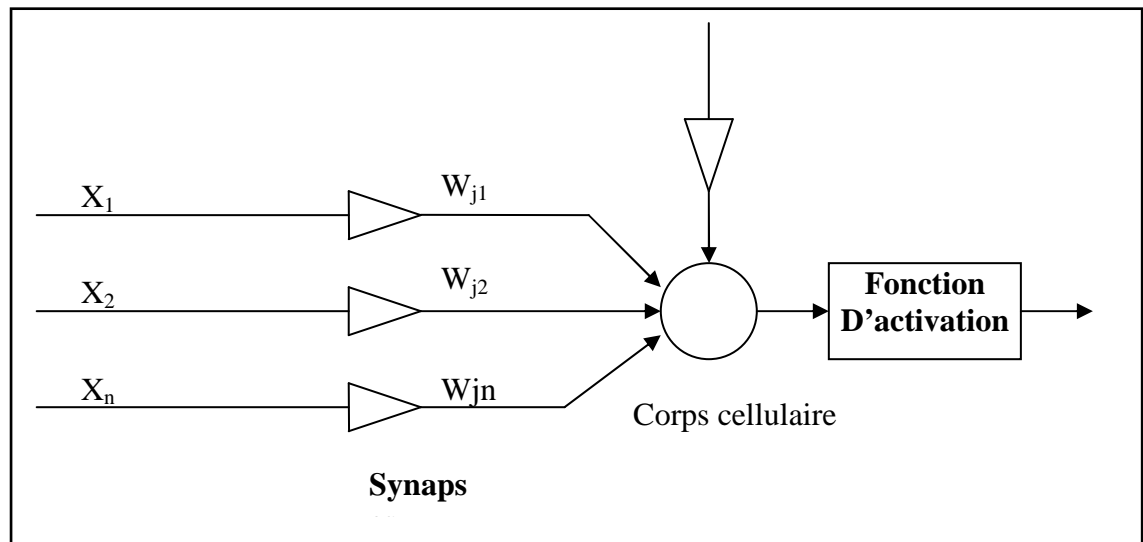
La sortie intermédiaire «  $S_i$  » est donnée par :

$$S_i = W_i * (X) - \theta_i \quad (3.1)$$

Et la sortie de neurone «  $O_i$  » est donnée par :

$$O_i = f (s_i) \quad (3.2)$$

Où les «  $\theta_i$  » sont les seuils et «  $f$  » est la fonction d'activation (ou de décision).



**Figure 3.2**– Modèle architectural d'un neurone formel

En général, un réseau de neurone artificiel peut être défini par les éléments suivants :

#### 3.4.1.1. Nature des entrées

Elle peuvent être soit binaires (-1, +1) ou (0, 1), ou bien réelles.

#### 3.4.1.2. La fonction d'entrée totale

Elle définit le prétraitement effectué sur les entrées.

#### 3.4.1.3. La fonction d'activation

La fonction de transfert est en général, une fonction non linéaire monotone croissante ; Par ailleurs, les fonctions de transfert sont de qualités diverses : elles peuvent être déterministes, continues, discontinues ou aléatoires [21]. La figure 3.3 donne les modèles de fonctions d'activation utilisées.

Fonction de Heaviside	
Fonction Linéaire sans saturation	
Fonction linéaire avec seuil	
Fonction a seuils multiples	
Fonction sigmoïde $\frac{1}{1 + e^{-s}} = F1(s)$	
Fonction sigmoïde $f2(s) = \frac{1 - e^{-s}}{1 + e^{-s}}$	
Fonction de stochastique	

**Figure 3.3**– Les modèles de fonctions d'activation

Le choix de la fonction d'activation dépend de l'application. S'il faut avoir des sorties binaires c'est la première fonction que l'on choisit habituellement. Une entrée spéciale est pratiquement toujours introduite pour chaque neurone. Cette entrée, normalement appelée biais, sert pour déplacer le pas de la fonction d'activation sur l'axe S.

#### 3.4.1.4. La fonction de sortie

Elle calcule la sortie de neurone en fonction de son état d'activation, en général cette fonction est considérée comme la fonction identité, par la suite confondrons toujours activation et sortie du neurone.



Les fonctions d'entrée et d'activation sont choisies en fonction des types d'automates utilisés, c'est-à-dire tout dépend de la nature des entrées et des sorties. En général, ils sont au nombre de six [22]:

#### **3.4.1.5. Les automates booléens**

- Les entrées et les sorties sont booléennes.
- La sortie est une fonction booléenne des entrées.

#### **3.4.1.6. Les automates à seuil**

- La sortie est binaire alors, que les entrées peuvent être binaires ou réelles.
- La fonction d'entrée totale est affine.
- La fonction d'activation est la fonction signe ou la fonction de heaviside.

#### **3.4.1.7. Les automates linéaires**

- Les entrées et les sorties sont réelles.
- La fonction d'entrée totale est linéaire.
- La fonction de sortie est l'identité.

#### **3.4.1.8. Les automates à saturation**

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

Si les entrées et les sorties sont entières, on obtient un automate multi seuils.

#### **3.4.1.9. Les automates continus**

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

#### **3.4.1.10. Les automates probabilistes**

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

### **3.5. Propriétés des Réseaux de Neurones**

Un réseau de neurone est un ensemble d'éléments de traitement de l'information, avec une topologie spécifique d'interconnexions entre ces éléments et une loi d'apprentissage pour adapter les points de connexions. D'une manière générale, un réseau de neurones possède les propriétés suivantes [17]:

### **3.5.1. Le parallélisme**

Cette notion se situe à la base de l'architecture des réseaux de neurones considérés comme ensembles d'entrées élémentaires qui travaillent simultanément.

### **3.5.2. La résistance aux pannes**

A cause de l'abondance des entrées et la structure du réseau. Les données bruitées ou les pannes locales dans un certain nombre de ses éléments n'affectent pas ses fonctionnalités. Cette propriété résulte de fonctionnement collectif et simultané des neurones qui les composent.

### **3.5.3. La capacité d'adaptation**

Celle-ci se manifeste tout d'abord dans les réseaux de neurones par la capacité d'apprentissage qui permet au réseau de tenir compte des nouvelles contraintes ou des nouvelles données du monde extérieur. De plus, ils se caractérisent par leur capacité d'auto organisation qui assure leur stabilité en tant que système dynamique.

### **3.5.4. La généralisation**

La capacité de généralisation d'un réseau de neurone est son aptitude de donner une réponse satisfaisante à une entrée qui ne fait pas partie des exemples à partir desquels il a appris

### **3.5.5. Structure de connexion**

Les connexions entre les neurones qui composent le réseau décrivent la "topologie" du modèle. Elles sont très variées, le nombre de connexions étant énorme. Cette topologie fait apparaître une certaine régularité de l'arrangement des neurones.

## **3.6. Topologie des Réseaux de Neurones**

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

### **3.6.1. Les réseaux statiques ou réseau à couche (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.

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.

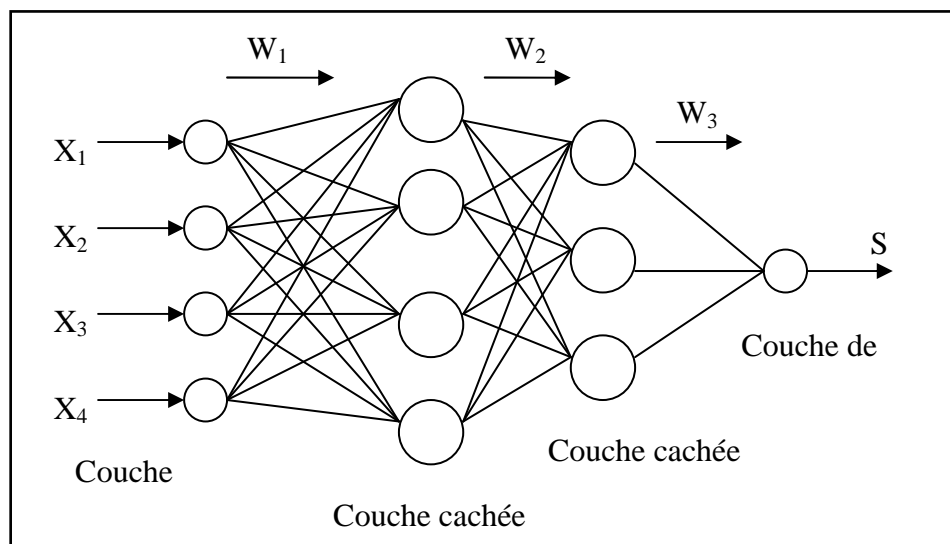
Les signaux des entrées se propagent de la première couche à la couche de sortie en passant par les couches cachées, Il n'y a pas des communications entre les unités de la même

couche, d'où le nom de feedforward. Les liens dirigés connectant les neurones sont appelés les interconnexions [14].

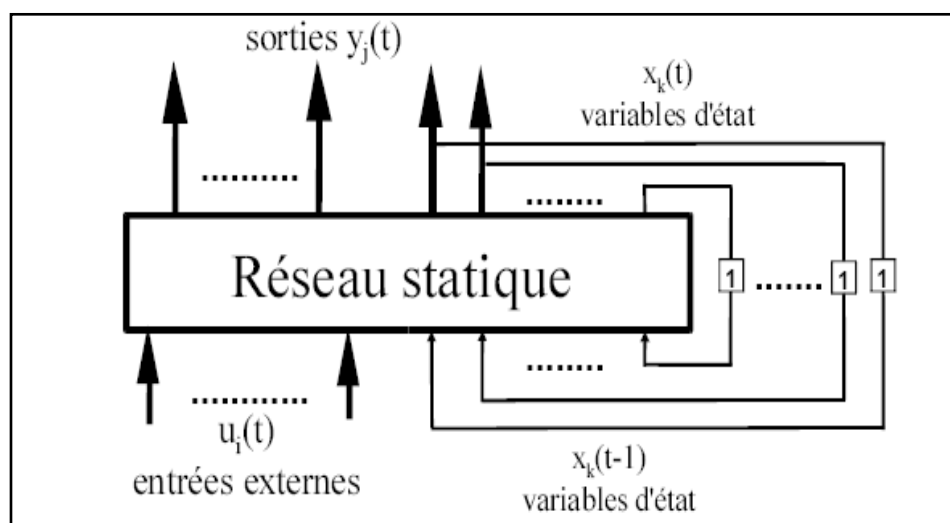
On distingue des réseaux à deux couches tel que le perceptron et l'adaline (adaptive linear neurone) qui sont caractérisés par :

- la simplicité de réglage d'apprentissage.
- la facilité de détermination de l'influence d'un neurone d'entrée sur l'erreur d'un neurone de sortie d'en déduire les modifications à apporter au lien qui les relie.
- La limitation au calcul de fonction très simple.

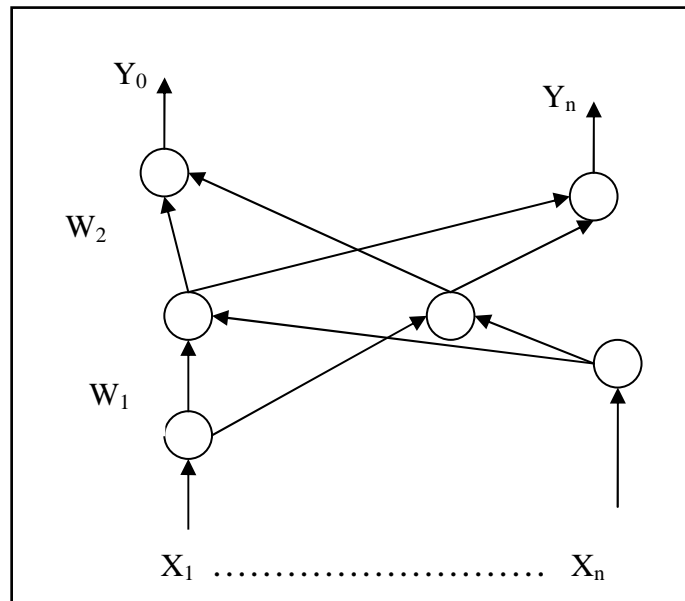
Ces réseaux ne pouvaient résoudre que des problèmes simples de classification. Pour des problèmes complexes, une solution consiste à organiser le réseau en plusieurs couches[23].



**Figure 3.4**– Structure d'un réseau statique avec deux couches cachées.



**Figure 3.5**– Forme canonique des réseaux de neurones



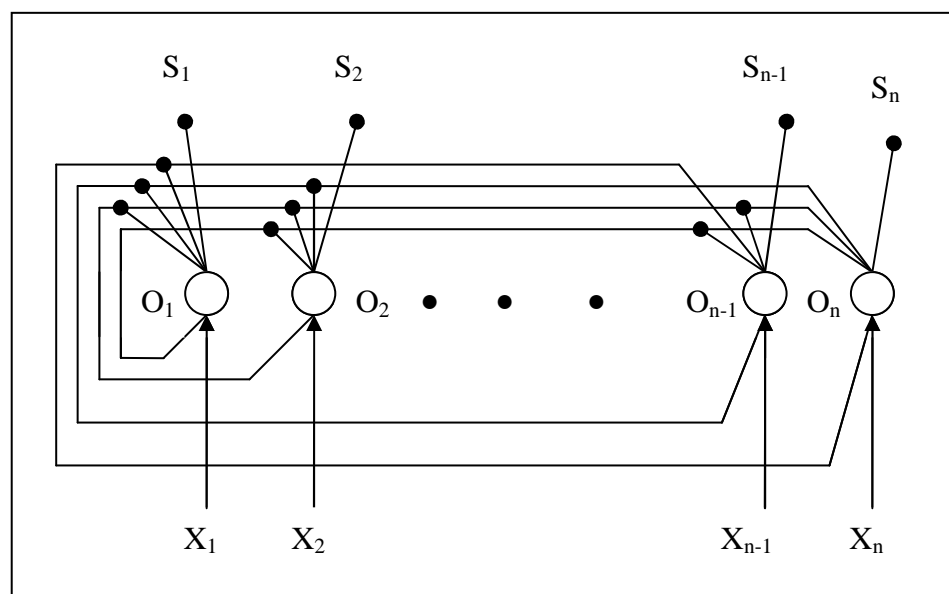
**Figure 3.6**– Un réseau FeedForward avec une couche cachée.

### 3.6.2. Les réseaux dynamiques (récurrents)

Comme leurs noms l'indiquent, contiennent des débouclages partiels ou totaux entre neurones, ils représentent donc une évolution dépendante du temps.

Il faut bien distinguer la dépendance théorique, pour laquelle l'état du réseau à un certain instant dépend de son état à l'instant ou aux instants précédents, du 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** ». Ils sont principalement utilisés pour l'implémentation des mémoires associatives « BAM : bidirectional associative memory » [14].



**Figure 3.7**– Réseau de neurone de Hop Field.

Le perceptron multicouches ordinaire ou la carte auto organisatrice sont des réseaux statiques. Par contre, le réseau du HOPFIELD ou le perceptron avec rebouclage est des réseaux dynamiques.

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 [14].

L'opérateur non linéaire réalisé par un réseau, bouclé ou non, dépend des valeurs des coefficients de pondération du réseau.

Pour qu'un réseau effectue une tâche donnée, il faut donc ajuster la valeur des ses coefficients. Une tâche est définie par un ensemble d'exemples, ou couples (valeurs des entrées et valeurs des sorties désirées correspondantes), tels les couples (forme classe) en classification, ou les couples (commande sortie mesurée du processus) en modélisation ces couples constituent l'ensemble d'apprentissage.

### 3.7. Le Perceptron

A la fin des années **50**, la neurobiologie ne disposait pas encore des techniques modernes de l'électrochimie pour étudier les structures du cerveau. L'idée de certains chercheurs fut alors d'utiliser les modélisations des neurones et de l'évolution des synapses pour simuler des réseaux de neurones. Ceux-ci pourraient exhiber des comportements proches de ceux du cerveau et ainsi fournir des explications sur le fonctionnement de celui-ci.

Le premier modèle fut présenté par ROSEBLATT c'était le perceptron ce modèle a pour propriétés [18] :

- D'être spécifié en termes suffisamment précis pour permettre le test des performances annoncées.
- D'être suffisamment complexe pour que l'on puisse espérer des comportements intéressants.
- D'être suffisamment simple pour que ses performances puissent être prédites et analysées.
- Enfin et surtout d'être en accord avec les faits biologiques

Un perceptron comprend trois éléments principaux :

#### 3.7.1. Rétine

Elle est constituée de cellules, sur lesquelles s'inscrit le stimulus.

### 3.7.2. Couche de cellules d'association

Chacune de ces cellules peut être connectée à des cellules de la rétine, et à d'autres cellules de décision.

Elle fait la somme des impulsions qui lui viennent et des cellules aux quelles elle est connectée.

### 3.7.3. Couche de cellule de décision

Ces cellules fonctionnent comme les cellules d'association, elles reçoivent leurs entrées des cellules d'association ou d'autres cellules de décision. Elles représentent la sortie du perceptron. Le sens des connexions entre les cellules d'associations et les cellules de décisions est bi directionnel, ce qui un FEED BACK de la sortie sur le réseau.

**Remarque :** Les neurones constituant le perceptron sont des réseaux linéaires à seuil.

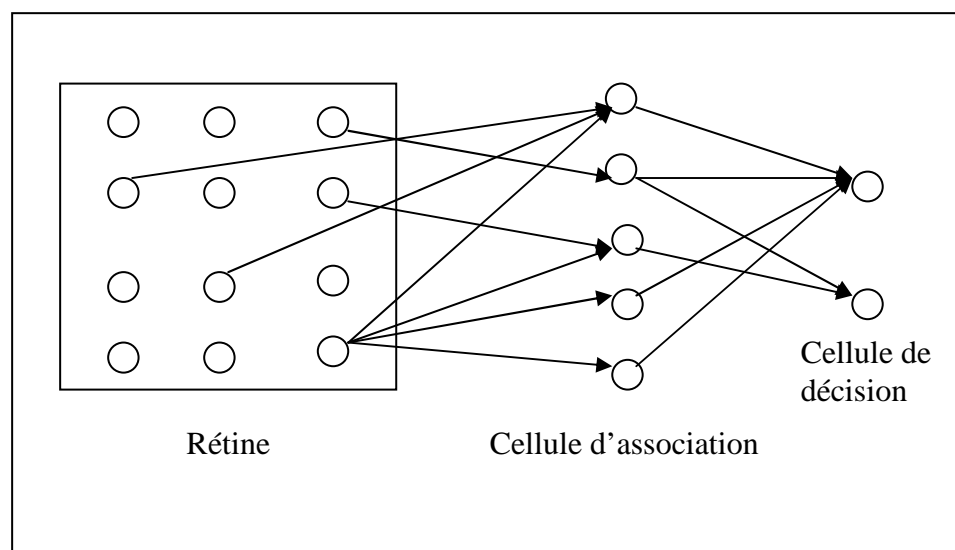
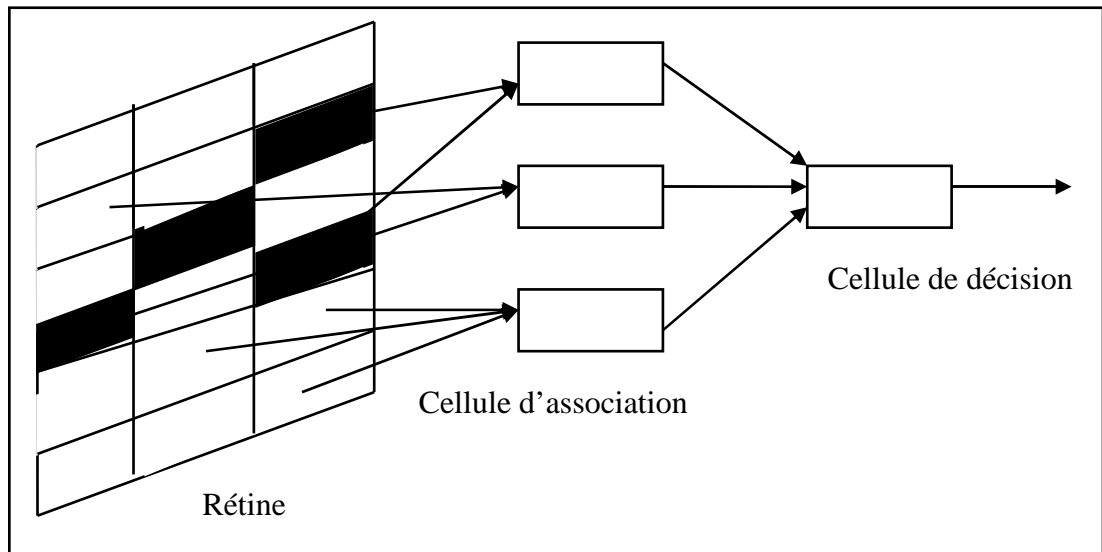


Figure 3.8– Schéma d'un perceptron

## 3.8. Perceptron Simple

Pour faciliter l'étude des comportements du perceptron, il est courant d'utiliser un perceptron d'une forme plus simple que celle définie par le modèle générale. Dans ce cas :

- Les valeurs de sorties de toutes les cellules (rétine, association, décision) sont binaires en 0 et 1.
- Les fonctions réalisées par les cellules d'associations sont des fonctions booléennes et les cellules de décision sont des automates à seuil.
- Il n'existe pas de connexion entre deux cellules d'association, ni entre deux cellules de décisions.
- Les connexions entre cellules d'association et cellules de décision sont à sens unique association décision.



**Figure 3.9**– perceptron simple [22]

Un tel perceptron va être utilisé pour essayer de reconnaître des formes qui lui seront présentées sur sa rétine. Si l'on n'utilise qu'une cellule de décision, celle-ci devra répondre 1 lorsque la forme qui lui sera présentée correspond à une forme qu'on veut lui faire reconnaître et 0 dans tous les autres cas.

L'apprentissage du perceptron va se faire suivant le perceptron de HEBB, c'est-à-dire en modifiant ses connexions [18].

### 3.9. L'Apprentissage des Réseaux de Neurones

L'apprentissage se conduit avec une méthodologie précise. La première, est la plus simple, connue sous le nom de règle de HEBB, date de 1949. Elle indique que lorsque deux neurones sont excités en même temps, il faut modifier les coefficients synaptiques pour renforcer cette excitation simultanée. La règle de WIDROW HOFF, établie en 1960, indique que la modification de ses coefficients est proportionnelle à l'erreur entre le résultat souhaité et le résultat réel et aux valeurs d'entrée.

La phase d'apprentissage consiste à modifier les poids jusqu'à obtention d'une stabilisation du réseau. C'est-à-dire, jusqu'à ce que le poids ne se modifie plus que d'une façon minime. L'apprentissage mathématique, basé sur ce concept, sert à minimiser une fonction de coûts formulée autour de l'erreur de sortie. Alors l'adaptation commence, par les neurones de la couche de sortie, forcés de la bonne valeur, puis on fait varier légèrement les poids des neurones des couches précédentes :

Il y a autant d'algorithmes d'apprentissage que de types de réseau. Mais la question qui se pose, est ce que on est absolument sûr de la stabilité du réseau ? Question classique

qui revient à assurer la convergence des algorithmes d'apprentissage utilisés. Les procédures d'apprentissage peuvent se subdiviser, elles aussi, en deux grandes catégories.

### **3.9.1. 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.

L'apprentissage supervisé détermine les poids synaptiques à partir d'exemples étiquetés de formes auquel un professeur a associé des réponses ou des cibles désirées également étiquetées et grâce à une stratégie spécifique. Après la phase d'apprentissage et la convergence des calculs, il n'est plus nécessaire, en général, de mémoriser le jeu d'apprentissage. Ces algorithmes sont particulièrement dédiés aux problèmes de classification des données bruitées ou incomplètes [14].

### **3.9.2. Apprentissage non supervisé**

Ne nécessite pas « d'enseignant » externe et se base uniquement sur les valeurs d'entrée sans fournir de réponse désirée. Le réseau s'auto organise de façon à optimiser une fonction de coût [24].

L'apprentissage non supervisé copie le fonctionnement du cerveau humain qui retrouve les informations par association. On présente à l'entrée du réseau des exemples connus et le réseau s'organise lui-même autour d'attracteurs qui correspondent à des configurations stables du modèle dynamique non linéaire associé au réseau. L'apprentissage est accompli à l'aide de règles qui changent ou adaptent le poids des coefficients synaptiques en fonction des exemples présentés à l'entrée et dans certains cas en fonction des sorties désirées.

## **3.10. Les Principales Règles d'Apprentissage**

Les stratégies des modifications des poids synaptiques sont dérivées des règles générales suivantes :

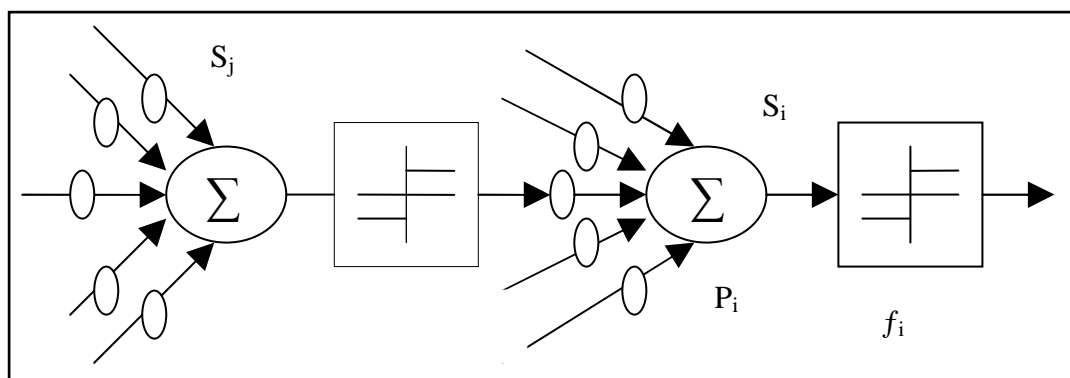
- La règle de HEBB ou les poids des connexions entre deux processeurs élémentaires est renforcé si les deux processeurs élémentaires sont activés simultanément
- La règle de WIDROW HOFF ou règle delta règle ou les poids synaptique est adaptée pour obtenir la diminution de l'erreur entre la sortie réelle du processeur élémentaire et la sortie désirée.



- L'apprentissage compétitif dans lequel tous les processeurs élémentaires sont en compétition. La sélection est réalisée à l'aide d'un processus de compétition basé sur une mesure de similarité entre l'exemple de référence associé à chaque processeur et l'exemple d'entrée.

Les expressions précises des règles dépendent très étroitement de la nature des fonctions d'entrée totale, des fonctions d'activation et de la nature continue ou discrète des informations contenues dans les cellules des neurones.

La figure 3.10 représente la cellule émettrice  $P_j$  connectée à la cellule réceptrice  $P_i$  par l'intermédiaire du poids synaptique  $W_{ij}$



**Figure 3.10**– Modèle d'apprentissage de la connexion  $W_{ij}$  entre deux neurones

Rumelhart [RUME 86] a proposé une règle générale de modification de poids synaptique  $\Delta W_{ij}$  valable pour de nombreux mécanismes d'apprentissage dans le cas d'une cellule  $P_i$  avec éventuellement une entrée «professeur »  $t_i$ .

$$\Delta W_{ij} = \alpha f_i(S_i, t_i) g_j(S_i, W_{ij}) \quad (3.3)$$

Avec :  $\alpha$  : le pas d'adaptation

$f_i$  et  $f_j$  : fonctions d'activations de  $P_i$  et  $P_j$ .

$S_i$  et  $S_j$  : valeurs d'activations de  $P_i$  et  $P_j$ .

$t_i$  : entrée professeur de  $P_i$ .

Dans la règle de HEBB, il n'y a pas de professeur et les fonctions  $f_i$  sont linéaires : le mécanisme d'adaptation se réduit à :

$$\Delta W_{ij} = \alpha S_i S_j \quad (3.4)$$

La règle de WIDROW HOFF, connue également sous le nom de Delta règle (ou  $\Delta$  règle) nécessite un professeur et se formule en présence de fonctions d'activation linéaire sous la forme :

$$\Delta W_{ij} = \alpha (t_i - S_i) S_j \quad (3.5)$$

Le modèle compétitif de type KOHONEN met en œuvre l'expression :

$$\Delta W_{ij} = \alpha (t_i - W_{ij}) \quad (3.6)$$

Après la phase d'apprentissage, toute la mémorisation des connaissances est gravée dans le réseau. C'est la phase la plus longue en calcul connexionniste. Par contre, en phase de reconnaissance, si la forme à reconnaître a déjà été apprise, le réseau fournira instantanément la bonne sortie.

Certains réseaux dédiés aux problèmes de classification automatique ont été conçus pour rejeter de façon certains les exemples qui n'ont pas été mémorisés par le réseau [21].

### 3.10.1. Les mécanismes de classification

La cellule de décision avec fonction de seuil se comporte comme un adaptateur linéaire qui réalise la séparation des exemples en deux classes  $w_1$  et  $w_2$  à l'aide de la fonction discriminante linéaire :

$$g(x) = w^{\text{tr}}x \quad (3.7)$$

$w = (W_{1j}, W_{2j}, \dots, W_{nj}, W_{0j})^{\text{tr}}$ , vecteur des poids synaptiques.

$X = (X_1, X_2, \dots, X_n, 1)^{\text{tr}}$ .

Cette fonction permet la séparation de l'ensemble des vecteurs  $x$  en deux classes  $w_1$  et  $w_2$  :

$$x \in w_1 \text{ si } g(x) \geq 0.$$

$$x \in w_2 \text{ si } g(x) < 0.$$

La figure 3.11 montre l'interprétation géométrique du mécanisme de classification.

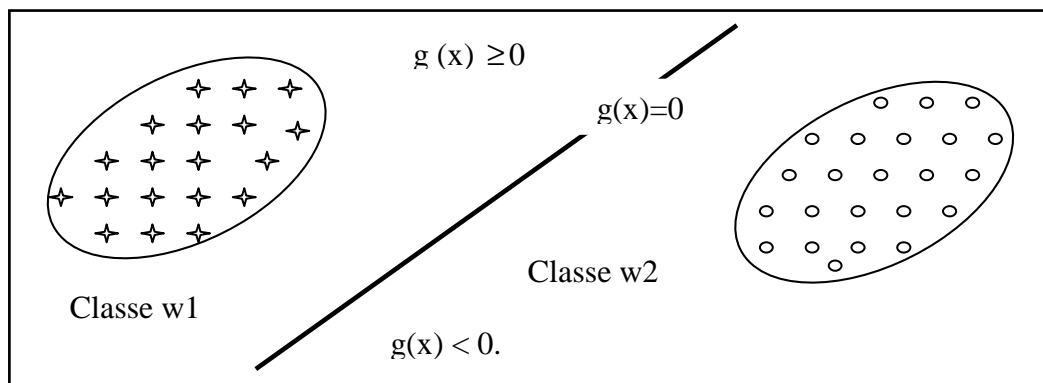


Figure 3.11 - Séparation de deux classes avec adaptateur linéaire

### 3.10.2. Les mémoires associatives

Les ordinateurs classiques de type Neumann traitent de façon séquentielle des informations qui sont codées sous forme binaire et rangées dans des mémoires accessibles par leur adresse indépendamment de leur contenu. La défaillance d'une seule mémoire provoque l'interruption de la chaîne de traitement séquentiel des opérations. Une

caractéristique importante des réseaux de neurones réside dans la manière avec laquelle sont stockées les informations et dans la façon de les associer.

En règle générale, la mémoire des réseaux de neurones est distribuée et toute ou partie des poids de connexion du réseau constitue la mémoire du réseau. Ainsi, un exemple de la base des exemples d'apprentissage sera mémorisé par un sous ensemble particulier des cellules qui contient également la mémoire de certains autres exemples de la base d'apprentissage. La majorité des réseaux de neurones utilisés en diagnostic industriel pour la classification des défauts sont appelés des mémoires associatives ou classifieurs. On distingue deux familles de réseau de neurones : les réseaux à base de neurones appelés souvent neurones « grand-mère » et les mémoires auto associatives et les mémoires hétéro associatives

Dans ce type de réseau, les neurones répondent spécifiquement à un seul exemple de la base d'apprentissage. Le traitement de l'information est réalisé par plusieurs neurones indépendants les uns des autres. Le surnom de tels réseaux repose sur l'hypothèse que le cerveau humain stocke l'image de notre grand-mère dans un neurone particulier placé dans un endroit précis du cerveau. Ces réseaux sont particulièrement adaptés aux problèmes de classification [21].

### **3.10.3. Mémoires auto associatives et hétéro associatives**

Dans une mémoire associative, les informations ne sont pas localisées à des emplacements déterminés, mais la mémorisation de chaque exemple est distribuée sur l'ensemble de la structure du réseau qui constitue ainsi la mémoire. Les seules sorties possibles sont également des contenus des mémoires. Les mémoires auto associatives ont pour caractéristique essentielle de reproduire en sortie, l'image apprise de l'entrée à partir d'un exemple incomplet ou bruité. Les mémoires hétéro associatives apprennent une relation entre un exemple donné et une sortie désirée différente de l'exemple. Si un nouvel exemple présenté est proche de l'exemple appris, il restituera la sortie désirée. Ces mémoires sont particulièrement performantes pour la reconnaissance de caractères avec des données incomplètes ou bruitées [21].

## **3.11. Algorithme de Retro propagation**

### **3.11.1. Introduction**

Le rétro propagation est le paradigme des R N A le plus utilisé.

Le terme se réfère à un algorithme pour ajuster les poids de connections en un R N A multi couches, ce paradigme a été appliqué avec succès dans différents domaines tel que le domaine militaire, médicale, synthèse de la parole, traitement de signal...etc.

Le rétro propagation est basé sur des principes mathématiques. Cette méthode a donné de bons résultats dans beaucoup d'applications. Pour l'appliquer il suffit d'avoir des données d'entrées et des données de sortie.

Cependant, toute information que possède un réseau de neurone est représentée par les poids d'interconnections, cette information est acquise durant la phase d'apprentissage.

Cette propriété d'apprendre a permis l'application des tels réseaux dans plusieurs domaines, contrôle, identification, diagnostic,...etc [18].

### **3.11.2. Présentation**

Cet algorithme que l'on désigne couramment par «Back propagation» est une généralisation de la règle de « WIDROW HOFF » pour un réseau multi couches.

Il a été mis au point simultanément par deux équipes indépendantes en France «FEGELAM N SAULIE, GALLINARI, LECUN» et aux Etats-Unis «RUMELHART, HITON, WILLIAMS ».

L'idée simple qui est à la base de cet algorithme, et qui permet de lever la difficulté du « crédit assignent problème » est l'utilisation d'une fonction dérivable (fonction sigmoïde) en remplacement de la fonction de seuil utilisée dans le neurone linéaires a seuil.

Mathématiquement, cet algorithme utilise simplement les règles de dérivation composée et ne présente aucune difficulté particulière.

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.

### **3.11.3. Position de problème**

L'apprentissage supervisé consiste à ajuster les coefficients synaptiques pour que les sorties du réseau 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 de l'erreur pour l'apprentissage des réseaux statiques multi couches.

### 3.11.4. Modèle et équation du réseau

Le réseau utilisé est un réseau multi couches, comportant une couche d'entrée qui corresponde à la rétine, une couche de sortie qui corresponde à la décision, et un certain nombre de couches dites cachées. Ces couches cachées constituent de véritable représentation interne des problèmes [18].

Le neurone utilisé dans le réseau est fondamentalement de même nature que le neurone linéaire à seuil du perceptron, il applique une fonction à la somme pondérée de ses entrées, cette fonction est une version lissée de la fonction à seuil on utilise en général fonction sigmoïde qui s'écrit :

$$f(S) = 1/(1+e^{-S}) \quad (3.8)$$

Les états des différents neurones dans un réseau ayant « L » couches (couches cachées et couches de sortie) données ayant « N » entrées et « M » sorties, sont données par les équations suivantes :

$$O_{pi} = f(S_{pi}) \quad (3.9)$$

Avec :

$$S_{pi} = \sum W_{ij} \cdot O_{ij} \quad (3.10)$$

Telle que

$O_{pi}$  : La sortie de neurone i pour l'exemple (entrée, sortie désirée) p.

$S_{pi}$  : Le potentiel somatique du neurone i pour l'exemple (entrée, sortie désirée) p.

$W_{ij}$  : Coefficient synaptique (poids) de la  $j^{\text{ème}}$  entrée du neurone i.

### 3.11.5. Principe de rétro propagation

La rétro propagation est basée sur l'adaptation des coefficients synaptiques dites encore des coefficients de pondération dans le but de minimiser une fonction de coût (performance) donnée par :

$$E(W) = \sum_{P=1}^P E_P(W) \quad (3.11)$$

$$E_P(W) = \frac{1}{2} \sum_{i=1}^n (t_{pi} - O_{pi})^2 \quad (3.12)$$

Ou :  $t_{pi}$  et  $O_{pi}$  Représentant respectivement la sortie désirée (Target) et la sortie (output) du réseau.

P : Le nombre d'exemple ou de la longueur de l'ensemble d'entraînement.

La minimisation de l'erreur se fait par une approximation d'une descente de gradient comme dans la règle de Widrow Hoff (règle delta). Toute la difficulté pour effectuer cette descente dans un réseau multi couches était de pouvoir calculer la dérivée de l'erreur quadratique par rapport à un poids donné.

### 3.11.6. Adaptation des poids

L'adaptation (ajustement, mise en forme) des coefficients synaptique, se fait par la méthode du gradient basée sur la formule itérative suivante :

$$W_{ij}(n+1) = W_{ij}(n) + \Delta W_{ij}^{(époque)} \quad (3.13)$$

$$\Delta_p W_{ij} = -\eta \cdot \frac{\partial E_p}{\partial W_{ij}} \quad (3.14)$$

$n$  : Représente le numéro d'itération.

$\eta$  : Représente la pas d'apprentissage. Ce facteur influe sur la vitesse de convergence du réseau.

La dérivée partielle de  $E(W)$  par rapport à chaque poids du réseau, pour cela et on utilisant la règle de delta généralisée. Nous devons charger notre poids liés à :

$$\Delta_p W_{ij} = \eta \cdot \delta_{pi} \cdot O_{pj} \quad (3.15)$$

Avec :

$O_{pj}$  : Sortie du neurone  $j$  pour un exemple  $p$ .

$\delta_{pi}$  : L'erreur commise à la sortie du neurone  $i$  pour l'exemple  $p$ , posant maintenant :

$$\delta_{pi} = -\frac{\partial E_p}{\partial S_{pi}} = (t_{pi} - O_{pi}) \quad (3.16)$$

L'astuce est de calculer  $\delta_{pi}$  pour chaque neurone dans le réseau. Le résultat intéressant que nous avons dérivé maintenant, est qu'il y a un simple calcul récursif de ces  $\delta$  qui peuvent être implémentés par la rétro propagation du signal d'erreur à travers le réseau.

Pour calculer :  $\delta_{pi} = -\frac{\partial E_p}{\partial S_{pi}}$

Nous appliquons la décomposition en chaîne pour exprimer dérivation partielle, nous obtenons :

$$\delta_{pi} = -\frac{\partial E_p}{\partial S_{pi}} = -\frac{\partial E_p}{\partial O_{pi}} * \frac{\partial O_{pi}}{\partial S_{pi}}$$

Calculons le deuxième facteur de l'équation (3.15) nous voyons que :

$$\frac{\partial O_{Pi}}{\partial S_{Pi}} = f'(S_{Pi})$$

Où :  $f'(S_{Pi})$  : Est la dérivée de la fonction d'activation du neurone  $I$  évaluée à  $S_{Pi}$ .

Pour calculer le premier facteur nous considérons deux cas :

Supposant que le neurone  $U_i$  est un neurone de sortie du réseau, dans ce cas, il paraît de la définition de  $E_p$  que :

$$\frac{\partial E_p}{\partial O_{Pi}} = -(t_{Pi} - O_{Pi})$$

Substituons les deux facteurs dans l'équation (16) nous obtenons :

$$\delta_{Pi} = (t_{Pi} - O_{Pi}) f'(S_{Pi}) \quad (3.17)$$

Pour tout neurone de sortie  $U_i$ .

Si  $U_i$  n'est pas un neurone de sortie, nous utilisons la décomposition en chaîne pour écrire :

$$\sum \frac{\partial E_p}{\partial O_{Pi}} * \frac{\partial S_{Pk}}{\partial O_{Pi}} = \sum \frac{\partial E_p}{\partial S_{Pk}} * \frac{\partial}{\partial O_{Pi}} \sum W_{ki} O_{Pi} = \sum \frac{\partial E_p}{\partial S_{Pk}} W_{ki} = \sum \delta_{Pk} W_{ki}$$

Dans ce cas, substituons les deux facteurs dans l'équation (16) nous obtenons :

$$\delta_{Pi} = f'(S_{Pi}) \sum \delta_{Pk} * W_{ki} \quad (3.18)$$

Où :  $U_i$  est un neurone appartenant aux couches cachées

Les équations (3.17) et (3.18) donnent une procédure récursive pour calculer les  $\delta$  pour tout neurone dans le réseau liée à l'équation (3.14), cette procédure constitue « la règle delta généralisée » pour un réseau statique, aussi pour le neurone non linéaire.

Pour minimiser l'erreur total sur l'ensemble d'entraînement, les poids du réseau peuvent être ajustés par la présentation de l'ensemble d'apprentissage en entier.

La variation des poids  $\Delta_p W_{ij}(n)$  peut alors s'écrire ainsi :

$$\Delta_p W_{ij}(n) = \eta \cdot \frac{\partial E_p(W)}{\partial W_{ij}(n)} \quad (3.19)$$

### 3.11.7. Les 'étapes d'algorithme de rétro propagation

L'algorithme de rétro propagation est représenté comme suit :

**Etape 01** : Initialiser les poids  $W_{ij}$  et les seuils internes des neurones à de petites valeurs aléatoires.

**Etape 02** : Présenter l'ensemble des couches (entrée, sortie désirée).

**Etape 03** : Présenter le première couche (entrée, sortie désirée) de l'ensemble.

**Etape 04** : Calculer :

- 1- La somme des entrées des neurones de la couche cachée par l'équation (3.10)
- 2- Les sorties des neurones de la couche cachée par l'équation (3.9)

3- La somme des entrées de la couche de sortie par l'équation (3.10).

4- Les sorties de réseau par l'équation (3.9).

**Etape 05 :** Calculer :

1- Les termes de l'erreur pour les neurones de la couche de sortie par l'équation (3.17).

2- Les termes de l'erreur pour les neurones de la couche cachée par l'équation (3.18).

**Etape 06 :** Calculer la variation des poids par l'équation (3.15).

**Etape 07 :** Calculer l'erreur  $E_p$  par l'équation (3.12).

**Etape 08 :** Présenter un autre couple (entrée, sortie désirée) et aller à l'étape quatre.

**Etape 09 :** Si tout l'ensemble des couple est présenté, calculer la variation total des poids par époque, en utilisant l'expression :

$$\Delta W_{ij}^{(époque)} = \sum_{P=1}^P \Delta_P W_{ij}$$

**Etape 10 :** Ajuster les poids par l'équation (3.13).

**Etape 11 :** Calculer l'erreur totale par l'équation (3.11).

**Etape 12 :** Comparer l'erreur totale avec une erreur prédéterminée ( $E_{PR}$ ).

$$[E - E_{PR} < \mathcal{E}].$$

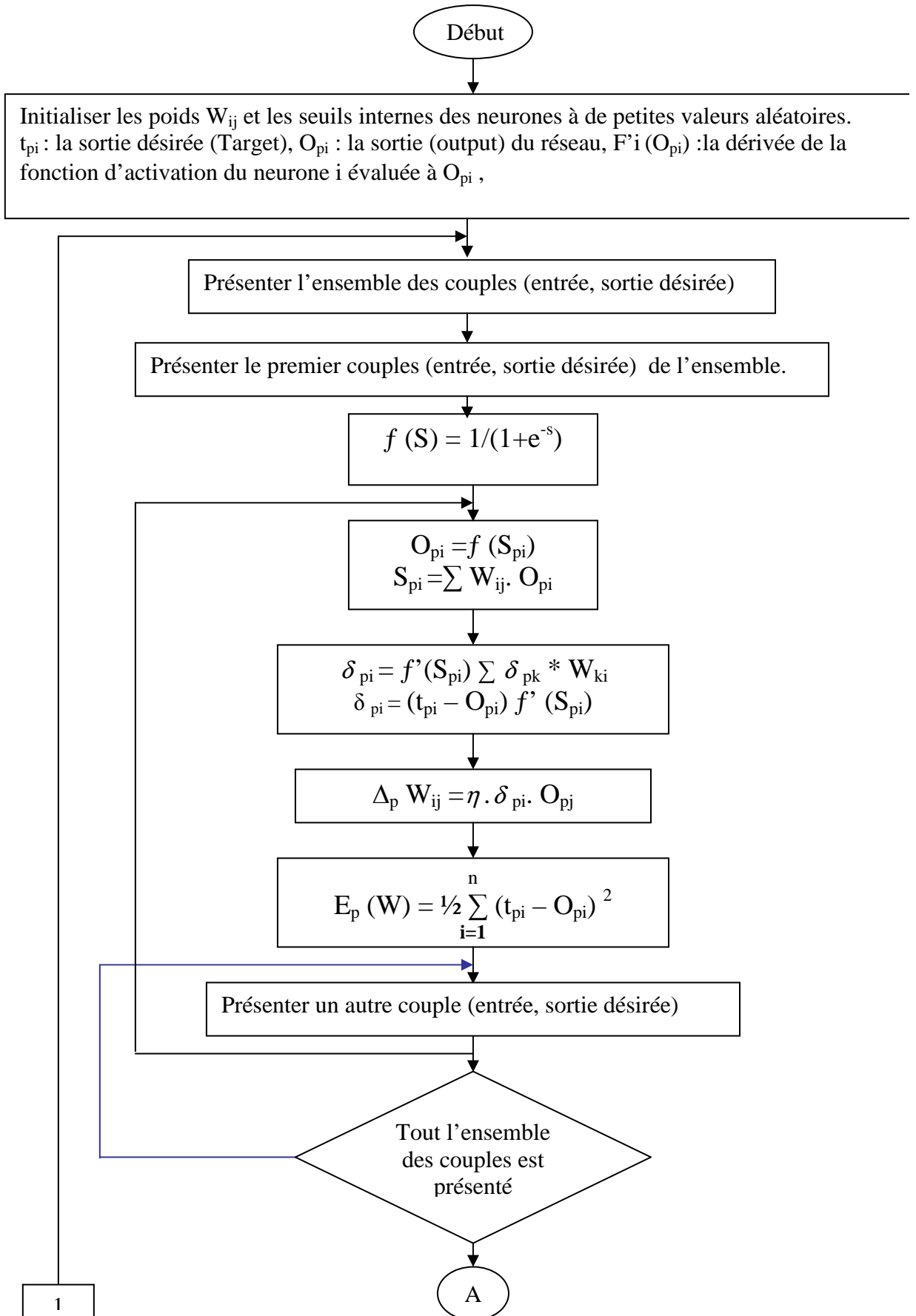
Si la condition est vérifiée aller à l'étape 13.

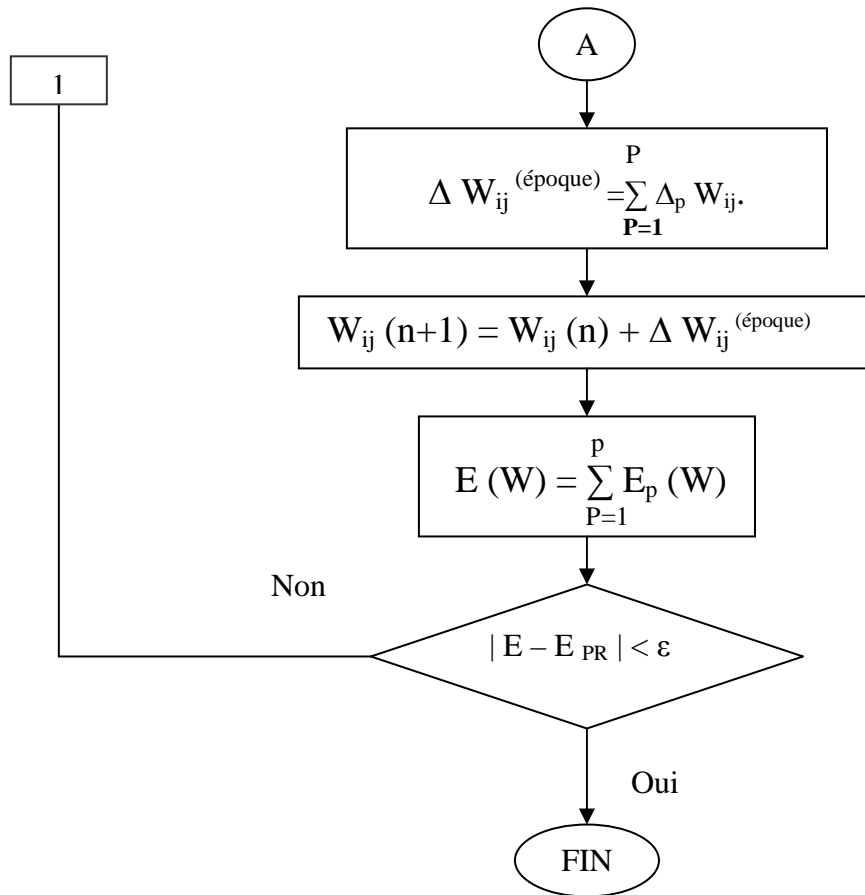
Si non aller à l'étape 03.

**Etape 13 :** Fin



**3.11.8. Algorithme de rétro propagation**





### 3.12. Domaines d'application des Réseaux de Neurones

Les propriétés d'application et d'approximation des réseaux de neurones ont permis une importante application de ces derniers dans les différents domaines pratiques, notamment, les domaines suivants :

#### 3.12.1. Classification des signaux :

Classer des signaux dans les catégories différentes en fonction des caractéristiques de forme (amplitude, fréquence, phase, etc. ...) peut être réalisé aisément par l'utilisation d'un réseau de neurones.

Supposons qu'on a deux catégories de signaux A et B et qu'on souhaite classer de différentes formes dans ces deux catégories en fonction des caractéristiques de ces formes.

On peut simplement définir la fonction non linéaire F qui vaut +1 pour les formes de la classe A et -1 pour celles de la classe B, le problème de classification revient à l'approximation de la fonction non linéaire F qui peut être résolue par un réseau de neurones qui possède des entrées et une seule sortie pour fournir la décision.

### 3.12.2. Identification des processus

Identifier un processus ; c'est d'essayer de donner un modèle plus proche au processus réel qui permet de reproduire avec la précision voulue, la sortie en fonction des entrées du processus, c'est typiquement un problème d'approximation qui peut être résolu par un réseau de neurones.

### 3.12.3. Contrôle des systèmes

Contrôler un système; c'est donner un ensemble d'entrées adéquates qui permettent d'avoir la sortie désirée (position, vitesse, etc....), et comme étant la fonction liant la sortie du système avec l'ensemble de commande est non linéaire, cette fonction peut être approchée avantageusement par un réseau de neurones.

### 3.12.4. L'application au diagnostic :

Les réseaux de neurones sont bien adaptés à la résolution des problèmes de diagnostic, utilisant la classification automatique des signaux et des formes. Dans ce contexte, on distingue plusieurs applications des réseaux de neurones pour le diagnostic des défaillances et en particulier, pour le diagnostic des pannes des machines électriques.

## 3.13. Les Caractéristique d'une Bonne Application

Nous pouvons déterminer quelques unes des caractéristiques des problèmes bien adaptés à une résolution par les réseaux de neurones :

- 1- Les règles qui permettraient de résoudre le problème sont inconnues ou très difficiles à expliciter ou à formaliser. Cependant, on dispose d'un ensemble d'exemples qui correspondent à des entrées du problème et aux solutions qui leur sont données par des experts.
- 2- Le problème fait intervenir des données bruitées.
- 3- Le problème peut évaluer, par exemple en faisant varier son champ de conditions initiales.
- 4- Le problème nécessite une grande rapidité de traitement, il doit par exemple être traité en temps réel.
- 5- Il n'existe pas de solutions technologiques courantes. Cette approche amène à dresser la liste suivante des domaines d'applications privilégiés :
  - Reconnaissance des formes.
  - Traitement de signal.
  - Vision, parole.
  - Prévision et modélisation.

- Aide à la décision.
- Robotique.

Ces quelques domaines possèdent pratiquement toutes les caractéristiques exposées précédemment, c'est pourquoi ils constituent le cœur des applications des réseaux de neurones.

Il ne faut cependant pas oublier le rôle que peuvent jouer les réseaux de neurones formels dans l'étude du cerveau en tant qu'instrument de simulation, et dans la recherche sur les neuro-ordinateurs et les nouvelles architectures d'ordinateurs en tant qu'instruments de formalisation et de simulation [16].

### **3.14. 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.

Nous avons présenté une étude sur l'algorithme du rétro propagation qui a prouvé son efficacité dans la pratique avec ses capacités d'apprentissage et d'identification.

Pour ces raisons, cet algorithme sera la base du travail qui sera présenté dans les prochains chapitres qui traite la commande d'un bras manipulateur par des technique neuro-floue (ANFIS).