

## **I.1. Introduction**

Les réseaux neurones sont des modèles mathématiques et informatiques des assemblages d'unités de calculs appelés "neurones formels" et dont l'inspiration originelle était un modèle de la cellule nerveuse humaine. Cet héritage de la neurobiologie forme une composante importante de la matière, et le souci de maintenir une certaine correspondance avec le système nerveux humain a amené et continue à amener une part importante des recherches dans le domaine.

Malgré cet héritage, l'essentiel des travaux d'aujourd'hui ont pour objet le réseau de neurones formels et non son corrélat neurobiologique. Vu les systèmes de calcul, les réseaux de neurones possèdent plusieurs propriétés qui les rendent intéressants d'un point de vue théorique, et fort utile en pratique.

## **I.2 Historique**

**1943** : Mc Culloch et Pitts présentent le premier neurone formel.

**1949** : Hebb propose un mécanisme d'apprentissage.

**1958** : Rosenblatt présente le premier réseau de neurones artificiels : le Perceptron. Il est inspiré du système visuel, et possède deux couches de neurones : perceptive et décisionnelle.

Dans la même période, le modèle de l'Adaline (Adaptive Linear Element) est présenté par Widrow. Ce sera le modèle de base des réseaux multicouches.

**1969** : Minsky et Papert publient une critique des perceptrons en montrant leurs limites, ce qui va faire diminuer la recherche sur le sujet.

**1972** : Kohonen présente ses travaux sur les mémoires associatives.

**1982** : Hopfield démontre l'intérêt d'utiliser les réseaux récurrents pour la compréhension et la modélisation des fonctions de mémorisation.

**1986** : Rumelhart popularise l'algorithme de rétropropagation du gradient, conçu par Werbos, qui permet d'entraîner les couches cachées des réseaux multicouches.

Les réseaux neuronaux ont été depuis beaucoup étudiés, et ont trouvé énormément d'applications [8].

## **I.3 Neurone biologique**

### **I.3.1 Modèle de neurone biologique**

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 leur organisation générale et leur système biochimique avec les autres cellules. Ils présentent cependant des caractéristiques qui leur sont propres et se retrouvent au niveau des cinq fonctions spécialisées qu'ils assurent :

- ✓ 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.

### **I.3.2 Structure des neurones**

Un neurone est constitué de: le corps cellulaire, les dendrites, l'axone, les synapses.

#### **I.3.2.1 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 du neurone. Sa forme est pyramidale ou sphérique dans la plus part des cas, elle dépend souvent de sa position dans le cerveau. Ce corps cellulaire fait quelques microns de diamètre.

#### **I.3.2.2 Les dendrites**

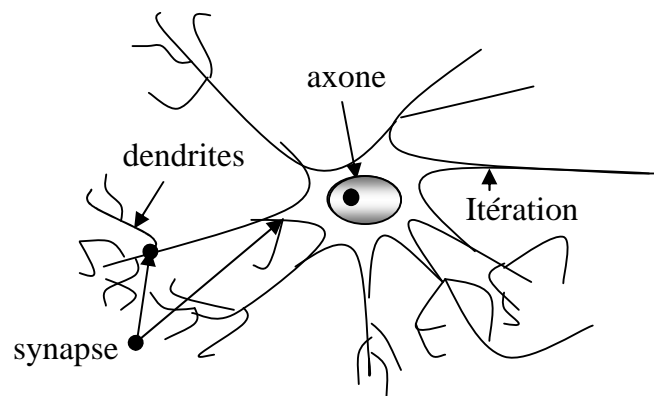
Chaque neurone possède une chevelure de dendrites. Celles-ci sont de fines extensions tubulaires, de quelques dizaines de microns de diamètre et d'une longueur de quelques dizaines de microns. Elles se ramifient, ce qui les amène à former une espace d'arborescence autour du corps cellulaires. Elles sont les récepteurs principaux du neurone pour capter les signaux qui lui parviennent.

### I.3.2.3 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 long (sa longueur varie d'un millimètre à plus d'un mètre) que les dendrites, et se ramifie à son extrémité, là où il communique avec d'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 en des endroits appelés synapses où ils sont séparés par un petit espace synaptique de l'ordre d'un centième de microns [2].

### I.3.2.4 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 éditeur chimique [6].



**Figure (I.1) Structure d'un neurone biologique.**

### **I.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 transmis par les dendrites. Si la sommation dépasse un seuil, le neurone répond par un influx nerveux ou 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 la libération d'un médiateur chimique celle-ci se diffuse jusqu'à la membrane du dendrite qui provoque naissance un potentiel [3].

### **I.4 Réseaux de neurones artificiels**

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 qui 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[6].

#### **I.4.1 Le neurone formel**

Le premier modèle du neurone formel a été présenté par Culloch et Pitts figure (I.2). D'une façon générale, un neurone formel est un élément de traitement qui fait une sommation pondérée de  $n$  entrées  $x_1, \dots, x_n$ . Si cette somme dépasse un certain seuil (fonction de seuillage  $F$ ), le neurone est activé et transmet une réponse dont la valeur est celle de son activation. Si le neurone n'est pas activé, il transmet rien.

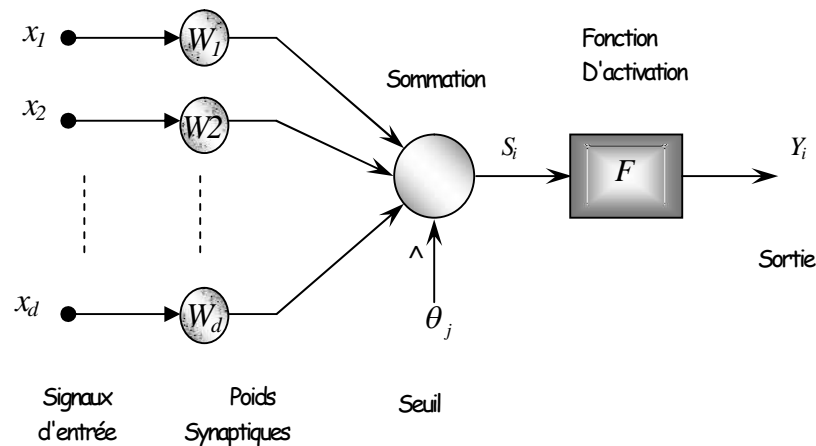
$$Y_i = F\left(\sum_{j=1}^n W_{ij} X_j\right) \quad (\text{I} - 1)$$

$W_{ij}$  : Coefficient synaptique ou poids associé à la  $i$  ème entrée du neurone  $j$ .

Parfois, il y a un terme additionnel  $\theta_j$  représentant le seuil interne du neurone, ce terme est considéré comme un poids  $W_0$  associé à une entrée constante, l'expression (I-1) devient donc [3] :

$$S_i = \sum_{i=1}^n W_{ij} X_i + \theta_j \quad (\text{I} - 2)$$

$$Y_i = F(S_i) \quad (\text{I} - 3)$$



**Figure (I.2) Le neurone formel.**

#### I.4.2 Modélisation générale

D'une façon plus générale, on peut définir un neurone formel par les cinq éléments suivants :

- La nature de ses entrées.
- La fonction d'entrée totale qui définit le prétraitement effectué sur les entrées.
- La fonction d'activation (ou d'état) du neurone qui définit son état interne en fonction de son entrée totale.

- La fonction de sortie qui calcule la sortie du neurone en fonction de son état d'activation ;
- La nature de la sortie du neurone.

#### I.4.2.1 La nature des entrées et de sortie

Elles peuvent être :

- Binaires : (0, 1).
- Réelles.

#### I.4.2.2 La fonction d'entrée totale

$h$  Peut être :

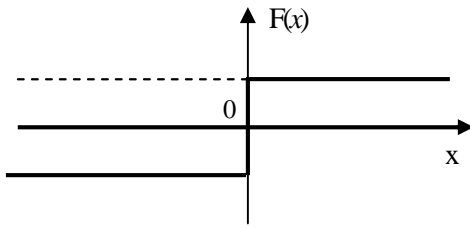
- Booléenne ;
- Linéaire :  $h(e_1, \dots, e_n) = \sum_{i=1}^n W_i \cdot e_i$  (I – 4)
- Affine:  $h(e_1, \dots, e_n) = \sum_{i=1}^n W_i \cdot e_i - a$  (I – 5)

Ces deux derniers cas sont les plus fréquents. Dans le deuxième on peut interpréter  $-a$  comme l'utilisation d'un neurone qui fournirait toujours l'entrée -1 au neurone étudié. Cette notation sera particulièrement utile lorsque nous aborderons les problèmes d'apprentissage avec des fonctions d'activations à seuil.

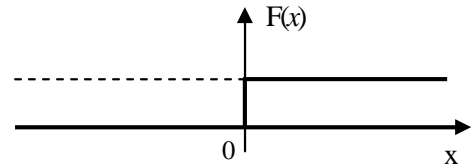
- Polynomiale de degré supérieur à deux.

#### I.4.2.3 Fonction d'activation

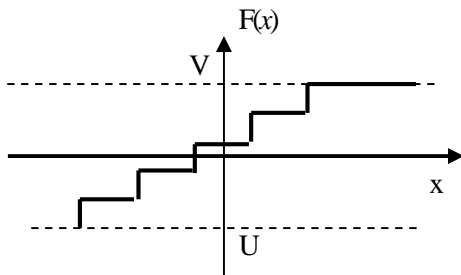
Les fonctions d'activations les plus utilisées :



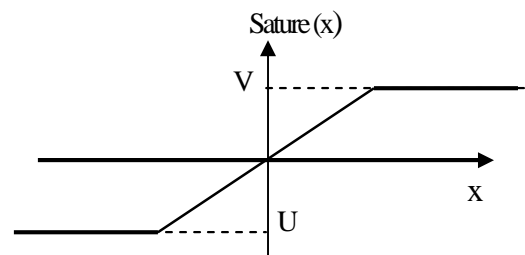
a) Fonction signe.



b) Fonction Heaviside.



c) Fonction à seuils.

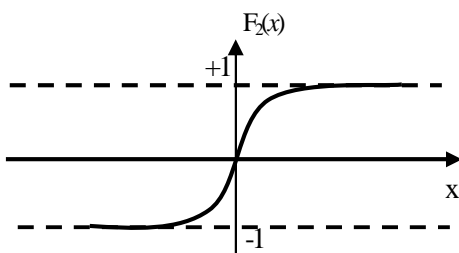


d) Fonction linéaire multi seuils.

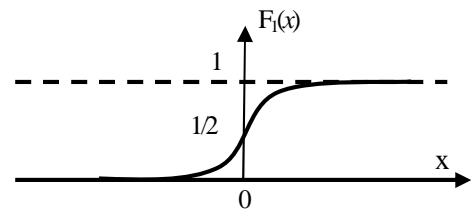
Une fonction sigmoïde :

$$F_1(x) = a(e^{kx} - 1) / (e^{kx} + 1) \quad (I-6)$$

$$F_2(x) = 1 / (e^{-x} + 1) \quad (I-7)$$



e) Fonction sigmoïde (-1, +1).



f) Fonction sigmoïde (0, 1).

Figure (I.3) Les fonctions d'activations.

**I.4.3 Propriétés des réseaux de neurones**

D'une manière générale un réseau de neurones possède les propriétés suivantes :

**I.4.3.1 Le parallélisme**

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

**I.4.3.2 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, elle se caractérise par leur capacité d'auto- organisation qui assure leur stabilité en tant que système dynamique.

**I.4.3.3 La résistance aux panne**

À cause de l'abondance des entrées et la structure du réseau, les données bruyantes ou les pannes locales dans certains nombres de ses éléments n'affectent pas ses fonctionnements collectifs et simultanés des neurones qui le composent.

**I.4.3.4 La généralisation**

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 est pris [4].

**I.4.3.5 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 [5].

**I.3.3. 5.1 Cas général**

Les structures qui peuvent être utilisées sont très variées.

Si l'on se réfère aux études être biologiques du cerveau, on constate que le nombre de connexions est énorme. Par exemple, des chercheurs ont montré que le cortex était divisé en différentes couches. A l'intérieur d'une même couche les interactions entre



neurones sont très grandes, mais les neurones d'une couche sont aussi reliés aux neurones d'autres couches, le tout formant un système d'une complexité.

D'une manière générale, l'architecture des réseaux de neurones formels peut aller d'une connectivité totale (tous les neurones sont reliés les uns aux autres), à une connectivité locale où les neurones ne sont reliés qu'à leurs plus proches voisins. Il est courant d'utiliser des réseaux à structure régulière pour faciliter leurs utilisations [2].

## **I.5 Architecture des réseaux de neurones**

### **I.5.1 Les réseaux à couches**

Aujourd'hui les réseaux de neurones à couches ou multicouches sont les plus utilisés en diagnostic industriel en raison de leur capacité à résoudre les problèmes de classification non linéaires par l'utilisation de l'algorithme de rétropropagation du gradient dérivé de la règle de Widrow- Hoff pour le calcul des poids des connexions des neurones des couches internes.

Le réseau à couches comprend :

- Une couche d'entrée qui reçoit l'ensemble des formes à classer.
- Une ou plusieurs couches cachées intermédiaires.
- Une couche de sortie qui doit restituer l'ensemble des sorties après l'apprentissage [5].

C'est pourquoi on utilise plutôt une structure de réseaux à couches telle que les neurones qui appartiennent à une même couche ne soient pas connectés entre eux, chacune des couches recevant des signaux de la couche précédente, et transmettant le résultat de ses traitements à la couche suivante.

Les deux couches extrêmes correspondent à la couche qui reçoit ses entrées du milieu extérieur d'une part, et à la couche qui fournit le résultat des traitements effectués d'autre part. Les couches cachées, leur nombre est variable[2].

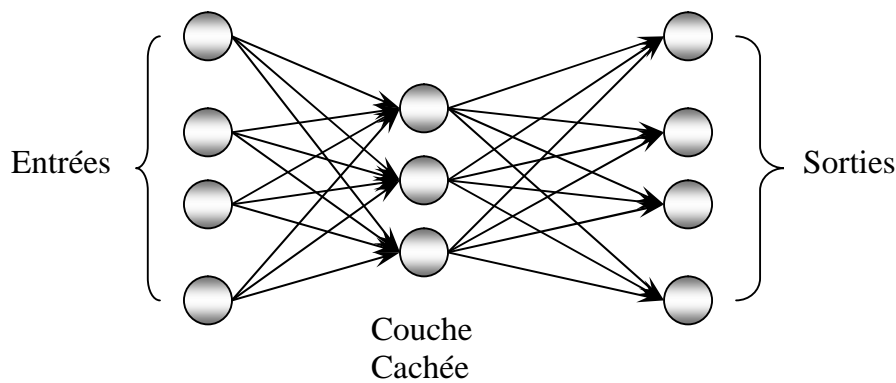


Figure (I.4) Réseau à couche

### I.5.2 Les réseaux entièrement connectés "Feed back network "

Dans ces réseaux, chaque cellule est reliée à toutes les autres et possède même un retour sur elle.

L'importance de ces réseaux n'est pas due à une origine biologique comme cela l'était pour les réseaux à couches cachées, mais au fait qu'on a pu établir une analogie entre les réseaux de ce type constitués d'automate à seuil et le comportement d'un verre de spin. Ceci a permis de mettre à la disposition des chercheurs tout le formalisme de la mécanique statistique et entraîné un regain d'intérêt pour l'étude des réseaux de neurones.

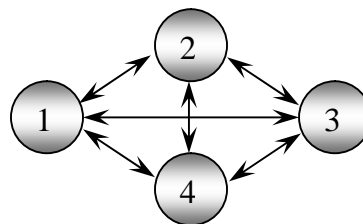


Figure (I.5) Réseaux entièrement connectés (bouclés)

### I.6 L'apprentissage

L'objectif de l'apprentissage est de fournir une méthode au réseau afin qu'il puisse ajuster ces paramètres lorsqu'on lui présente des exemplaires à traiter. Certains réseaux font la différence entre une phase d'exploitation (cas du perceptron).

On distingue habituellement trois paradigmes d'apprentissage :

### **I.6.1 Apprentissage supervisé**

Dans ce cas on fournit au réseau la donnée à traiter mais aussi la réponse attendue. Le réseau effectue une évaluation de la donnée, puis compare la valeur obtenue avec la valeur désirée, il va en suite modifier ses paramètres internes afin de minimiser l'erreur constatée [7].

#### **I.6.1.1 Problèmes d'apprentissage supervisé**

L'apprentissage supervisé consiste à ajuster les paramètres du réseau pour que ses sorties soient suffisamment proches des sorties désirées pour des ensembles d'entraînement.

L'objectif est d'adapter le réseau pour qu'il soit capable de reconnaître des exemples n'appartenant pas nécessairement à l'ensemble d'apprentissage pour résoudre un tel problème, deux étapes sont nécessaires :

1. Spécifier une topologie "équation et connexion" du réseaux. La relation entre les entrées et les sorties doit dépendre d'un ensemble de paramètres (poids) ajustables.
2. Spécifier une règle d'apprentissage, qu l'on utilise pour l'adaptation de ses paramètres.
3. La rétropropagation est actuellement la règle d'apprentissage la plus utilisée pour l'apprentissage supervisé des réseaux de neurones. Elle généralement utiliser pour des réseaux multicouches mais et elle peu être appliquée à n'importe quelle architecture à fonctions dérivables [1].

### **I.6.2 Apprentissage non supervisé**

Dans ce type d'apprentissage la connaissance de la sortie désirée n'est pas nécessaire, et la procédure d'apprentissage est basée uniquement sur les valeurs d'entrées. Le réseau s'auto organise de façon à optimiser une certaine fonction de coût, sans qu'on lui fournir la réponse désirée. Cette propriété est appelée auto organisation [4].

### **I.6. 3 Apprentissage par renforcement**

L'idée de base de l'apprentissage par renforcement est inspirée des mécanismes d'apprentissage chez les animaux, dans ce type l'apprentissage on suppose qu'il n'existe pas de maître qui peut fournir la réponse correcte, mais le système à entraîner est informé, d'une manière indirecte, sur l'effet de son action choisie. Cette action est renforcée si elle conduit à une amélioration des performances du système entraîné et les éléments qui contribuent dans la génération de cette action sont soit récompensés ou punis [4].

### **I.7 Les principales règles d'apprentissage**

Les stratégies de modification des poids synaptiques sont dérivées des règles générales suivantes :

- La règle de Hebb ou le poids des connexions entre deux processeurs élémentaires est renforcée si les deux processeurs élémentaires sont activés simultanément ;
- La règle Widrow-Hoff ou règle Delta règle ou' le poids synaptique est adapté 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 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 continues dans les cellules des neurones [5].

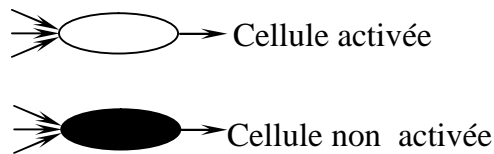
#### **I.7.1 La règle de Hebb**

Comme nous l'avons vu dans les fondements biologiques, la règle de Hebb est le premier mécanisme d'évolution proposé pour les synapses.

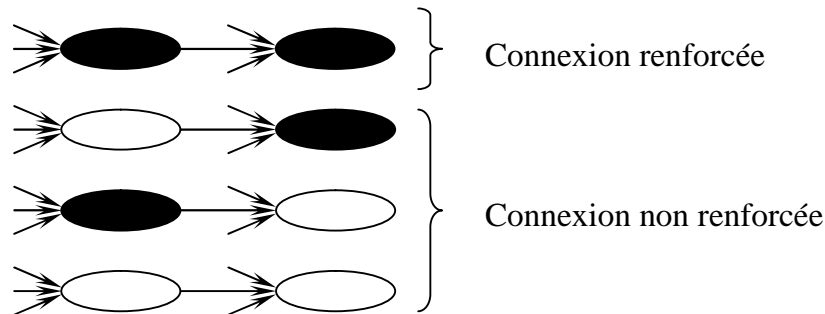
Son interprétation pour les réseaux de neurone formels est la suivante :

On considère que si deux neurones connectés entre eux sont activés au même moment, la connexion qui les relie doit être renforcée. Dans les cas contraire, elle n'est pas modifiée.

Si nous prenons les conventions suivantes [2] :



La règle de Hebb donne alors:



Ceci traduit bien le fait que lorsque la connexion entre deux cellules est très forte, quand la cellule émettrice s'active, aussi. Il faut donc, pour lui permettre de jouer ce rôle déterminant lors du mécanisme d'apprentissage, augmenter le poids de cette connexion. En revanche, si la cellule émettrice s'active sans que la cellule réceptrice le fasse, ou si la cellule émettrice ne s'était pas activée, cela traduit bien le fait que la connexion entre ces deux cellules n'est pas prépondérante dans le comportement de la cellule réceptrice. On peut donc, dans la phase d'apprentissage, laisser un poids faible à cette connexion.

Notons  $W_{ij}(t)$  le poids de la connexion entre le neurone  $j$  et le neurone  $i$  à l'instant  $t$ .

Supposons qu'entre les instants  $t$  et  $t + \delta t$ , on ait soumis le réseau à un stimulus, et que ce stimulus ait provoqué l'activation  $A_i$  du neurone  $i$ , et l'activation  $A_j$  du neurone  $j$ .

Supposons de plus que  $A_i$  et  $A_j$  soient booléens (d'un point de vue général, il est complètement équivalent que l'on prenne les valeurs des activations dans  $(0, 1)$  ou  $(-1, +1)$ ), la règle de Hebb s'écrit

$$W_{ij}(t + \delta t) = W_{ij}(t) + \mu A_i A_j \quad (\text{I-8})$$

Où  $\mu$  ( $\mu > 0$ ) est un paramètre de l'intensité de l'apprentissage [7].

### I.7.2 La règle de Windrow Hoff

La règle de Widrow Hoff ou règle delta proposée en 1960, consiste à modifier, à chaque pas, les poids et les biais afin de minimiser la somme des carrés des erreurs en sortie en utilisant la règle suivante :

$$w(k+1) = w(k) + \eta(t_k - y_k)x_k^T \quad (\text{I-9})$$

A chaque pas d'apprentissage  $k$ , l'erreur en sortie est calculée comme la différence entre la cible recherchée  $t$  et la sortie  $y$  du réseau [6].

$$E_k = e_k^T e_k = (t_k - y_k)^T (t_k - y_k) = \frac{1}{2}(t_k^T t_k + y_k^T y_k - 2y_k^T t_k) \quad (\text{I-10})$$

Le gradient de cette qualité par rapport à la matrice de poids  $w$  est donné par :

$$\frac{\nabla E_k}{w} = \frac{1}{2} \cdot \frac{\nabla(y_k y_k^T - 2y_k^T t_k)}{w} \quad (\text{I-11})$$

Le gradient se calcul comme suit :

$$\frac{\nabla E_k}{w} = \frac{\partial E_k}{\partial w} = \frac{\partial E_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial w} \quad (\text{I-12})$$

D'après l'expression de  $E_k$  et avec  $y_k = w x_k + b$  les dérivées partielles sont :

$$\frac{\partial E_k}{\partial y_k} = y_k - t_k \quad (\text{I-13})$$

$$\frac{\partial (w x_k + b)}{\partial w} = x_k^T \quad (\text{I-14})$$

La mise à jour des poids se fait par l'équation :

$$W(k+1) = w(k) - \eta \frac{\nabla E_k}{w} \quad (\text{I-15})$$

Avec  $\eta$  : le gain d'apprentissage ( $0 < \eta < 1$ ) .

De même, on obtient l'expression de la modification du biais [6] :

$$b(k+1) = b(k) + \eta(t_k - y_k) = b(k) - \eta \frac{\nabla E_k}{w} \quad (\text{I-16})$$

**I.8 Quelques modèles de réseau de neurones****I.8.1 Le modèle du perceptron****I.8.1.1 Genèse**

A la suite des recherches de Mac Culloch et Pitts qui aboutirent à la définition du neurone formel, et de cellules de Hebb qui fournirent une exploitation synaptique du mécanisme d'apprentissage, les chercheurs se retrouvaient une fois de plus confrontés au problème qui consiste à expliquer les fonctionnalités du cerveau par celles de ses structures. A la fin des années de 1950, la neurobiologie ne disposait pas encore des techniques modernes de l'électro- chimie 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. Ils espéraient que si la construction de tels réseaux ne voilait aucune contrainte biologique, 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 solide fut présenté par Rosenblatt, c'était le perceptron. Ce modèle a des propriétés :

- 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 .

**I.8.1.2 Organisation générale**

Un perceptron comprend trois éléments principaux :

**I.8.1.2.1 Une rétine**

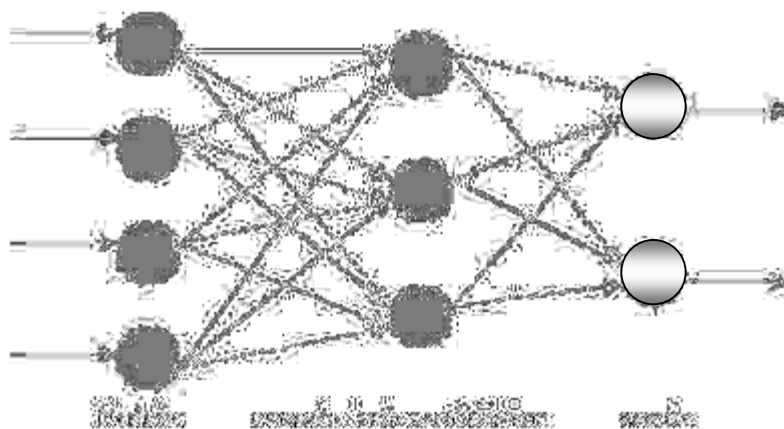
Il est constituée de cellules, sur les quelles s'inscrit le stimulus dans la plupart des modèles, ces cellules répondent suivant une loi du tout- ou- rien, mais elles peuvent aussi fournir une réponse modulée suivant l'intensité du stimulus.

### I.8.1.2.2 Une couche de cellules d'association

Chacune de ces cellules peut être connectée à des cellules de la rétine, à d'autres cellules d'association, et à des cellules de décision (voir explication suivante). Elle fait la somme des impulsions qui lui viennent des cellules aux quelles elle est connectée. Elle répond généralement suivant une loi de tout – ou- rien après avoir comparé la somme effectuée précédemment avec un seuil. Elle peut cependant de la rétine vers les cellules d'association [2].

### I.8.1.2.3 Une couche de cellules de décision

Ces cellules fonctionnent comme les cellules d'association. Elles reçoivent leurs entrées des cellules d'association ou d'autre cellules de décision. Elles représentent la sortie du perceptron. Le sens des connections entre les cellules D'association et les cellules de décision est bi- directionnel, ce qui permet un feed- back de la sortie sur le réseau [2].



**Figure (I.6) Schéma d'un perceptron.**

## I.8.2 Le perceptron simple

Pour faciliter l'étude des comportements du perceptron, il est courant d'utiliser un perceptron d'une forme plus simple que cette définie par le modèle général. Dans ce cas, les valeurs de sorties de toutes les cellules (de la rétine, d'association, de décision) sont binaires en 0 et 1. Les fonctions réalisées par les cellules d'association sont des fonctions booléennes, et les cellules de décision sont automates à seuil.



On considère d'autre part qu'il n'existe pas de connexions entre deux cellules d'association, ni entre deux cellules de décision. Enfin, les connexions entre cellules d'association et cellules de décision sont à sens unique association vers décision.

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 [7].

### **I.8. 3 L'adaline**

Le modèle Adaline a été mis au point par Bernard Widrow pour la résolution du filtrage adaptatif du signal en 1959. Il appartient à la famille des classifieurs à deux classes.

#### **I.8. 3.1 Structure de l'adaline**

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 différente 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.7).

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 les télécommunications pour améliorer le signal sur bruit en prenant en compte la variation de l'impédance des différentes lignes téléphoniques [5].

#### **I.8. 3.2 Règle d'apprentissage**

La règle d'apprentissage de l'adaline, appelée règle de Widrow Hoff ou LMS utilise la descente de gradient de la fonction de la somme des erreurs quadratiques, donnée par [6].

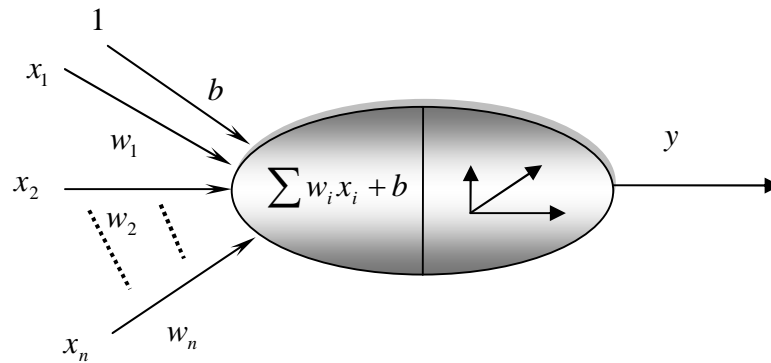
$$E(w) = \frac{1}{2} \sum_{k=1}^m (t_k - y_k)^2 \quad (\text{I} - 17)$$

Avec :

$$y_k = w^t x_k \quad (\text{I} - 18)$$

La descente du gradient donne la direction qui minimise  $E(w)$  :

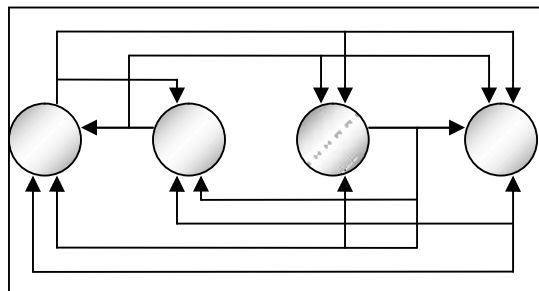
$$w(t+1) = w(t) - \eta \nabla E(w) \quad (\text{I} - 19)$$



**Figure (I.7) Architecture de l'Adaline**

#### I.8. 4 Le réseau de hopfield

Le modèle de hopfield a été proposé en 1982, sont des réseaux récurrents et entièrement connectés. Dans ce type de réseau, chaque neurone est connecté à chaque autre neurone et il n'y a aucune différenciation entre les neurones d'entrée et de sortie. Ils fonctionnent comme une mémoire associative non linéaire et sont capables de trouver un objet stocké en fonction de représentations partielles ou bruitées. L'application principale des réseaux de Hopfield est l'entrepôt de connaissances mais aussi la résolution de problèmes d'optimisation. Le mode d'apprentissage utilisé ici est le mode non supervisé [7].



**Figure (I.8) Le réseau de Hopfield**

### I.8. 5 La carte auto organisatrice de Kohonen

Le modèle a été proposé dès les années 70 par Von der Marsberg, puis par Kohonen ce modèle a pour objectif de représenter des données complexes. Il est réalisé à partir d'un réseau de K neurone à n entrées et donc k sorties. Les entrées sont des vecteurs à n composantes toutes connectées totalement K neurones du réseau par nk connexions modifiables. Les neurones du réseau sont placés en général dans un espace de 1 ou 2 dimensions [6].

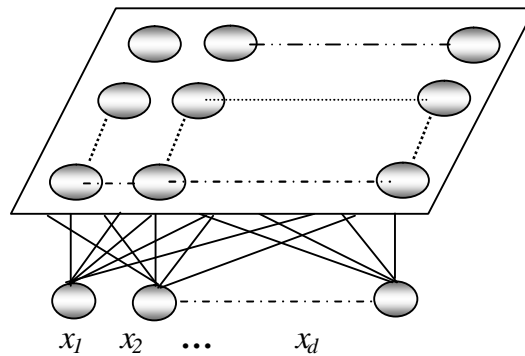


Figure (I.9) Réseaux de Kohonen

### I.9 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 W1 et W2 à l'aide de la fonction discriminante linéaire :

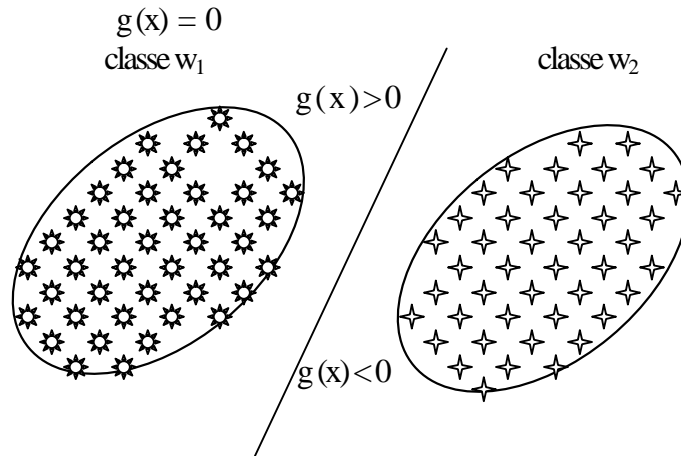
$$g(x) = W^T x \quad (\text{I} - 20)$$

$$w = (W_{1j}, W_{2j}, \dots, W_{nj}, W_{0j})^T, \text{ vecteurs des poids synaptiques.} \quad (\text{I} - 21)$$

$$x = (X_1, X_2, \dots, X_n, 1)^T \quad (\text{I} - 22)$$

Cette fonction permet la séparation de l'ensemble des vecteurs x en deux classes W1 et W2 :

$$\begin{cases} x \in w_1 & \text{si } g(x) \geq 0 \\ x \in w_2 & \text{si } g(x) < 0 \end{cases} \quad (\text{I} - 23)$$



**Figure (I.10) Séparation de deux classes avec adaptateur linéaire.**

### I.10 Les mémoires associatives

Les ordinateurs classiques de type Von 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 tout 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 contiennent 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 associatifs ou classifieurs. On distingue cependant deux familles de réseaux de neurones : les réseaux à base de neurones appelés souvent neurones « grand- mère » et les mémoires auto- associatives et les mémoires hétéro- associatives [5].

#### I.10.1 Réseaux avec neurones grand- mère

Dans ce type de réseaux, les neurones répondent spécifiquement à un seul exemple de la base d'apprentissage. Le traitement de l'information est réalisé par plusieurs neurones indépendants des uns des autres. Le sur nom 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 [5].

### **I.10.2 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 produire 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 [5].

## **I.11 Le rétropropagation**

### **I.11.1 Introduction**

Le problème de calcul des erreurs de sortie pour les neurones cachés, limite l'efficacité des algorithmes d'apprentissage supervisé pour les réseaux multi couches. Donc à cause de l'absence d'une règle d'apprentissage convenable, les perceptrons étaient limités à une seule couche pendant plusieurs années, le problème du calcul des erreurs associées aux neurones cachés a été levé par la découverte de l'algorithme de rétro propagation qui peut être appliqué à n'importe quel système composé de plusieurs sous systèmes élémentaires qui peuvent être représentés par des fonctions connues, continues dérivables.

Avant de définir la règle d'apprentissage, on définit la relation entre les sorties du réseau d'une part, les entrées et les poids d'autre part. on considère dans ce qui suit les réseaux non récurrents (statiques) multicouches. Pour un réseau multicouches (N) entrées et (M) sorties, composé de (L) couches {couches cachées plus couche de sortie}, la somme des entrées de la  $j^{\text{ème}}$  unité cachée est [3].

### I.11.2.1 Principe

Pour converger vers le minimum global de la surface d'erreur, on procède par la méthode du gradient, cette technique ajuste les poids dans la direction opposée du gradient instantané de la fonction d'erreur  $\Delta_K = \frac{\partial E_K}{\partial W_K}$

$$e(t) = [y_d(t) - R(t)] \quad (\text{I} - 24)$$

$$E = \frac{1}{2} \sum_{i=1}^T E(t) = \frac{1}{2} \sum_{i=1}^T [y_d(t) - R(t)]^2 \quad (\text{I} - 25)$$

T : est l'ensemble des exemples d'entraînement.

$y_d(t)$  : Est le vecteur de sortie désiré.

$R(t)$  : Celui du réseau.

W : vecteur des poids

### I.11.2.1 Equation du réseau

Pour un réseau multicouche à N entrées et M sorties composé de L couches (couches cachées, couche de sortie), la somme des entrées de la  $j^{\text{ème}}$  neurone caché est :

$$S_j^L(t) = \sum_{i=1}^N W_{ji}^L x_i(t) + \theta_j^L \quad (\text{I} - 26)$$

L : Indice de la couche cachée.

$W_{ji}^L$  : Poids de la connexion de i ème entrée.

$\theta_j^L$  : Valeur du seuil interne du neurone.

F : Fonction d'activation telle que :

$$I_j^L(t) = F_j^L(S_j^L(t)) = F_j^L\left(\sum_{i=1}^N W_{ji}^L x_i(t) + \theta_j^L\right) \quad (\text{I} - 27)$$

Les équations de la couche de sortie sont :

$$S_K^R(t) = \sum_{j=1}^M W_{Kj}^R I_j(t) + \theta_K^R \quad (\text{I} - 28)$$

$$R_K^R(t) = F_K^R(S_K^R(t)) \quad (\text{I} - 29)$$

K : Numéro du neurone dans la couche de sortie

### I.11.2.2 Adaptation des poids

Les paramètres (poids) sont ajustés par la méthode de la descente du gradient :

$$W_{ji}^L(n+1) = W_{ji}^L(n) + \Delta W_{ji}^L(n) \quad (\text{I} - 30)$$

Où :

$$\Delta W_{ji}^L(n) = -\mu \frac{\partial E}{\partial W_{ji}^L(n)} \quad (\text{I} - 31)$$

Avec :

$n$  : C'est le numéro de l'itération.

$\mu$  : Pas d'apprentissage désignant la vitesse de convergence, sa valeur est généralement choisie expérimentalement.

Le coefficient d'apprentissage  $\mu$  joue un rôle important puisqu'il règle la vitesse avec laquelle se fait la descente du gradient; trop petit, il ne permet d'atteindre une valeur suffisamment faible de l'erreur qu'au bout d'un très grand nombre d'itérations, et même fait encourir le risque d'un minimum local (insatisfaisant) duquel il est impossible de sortir, trop grand il peut conduire à s'éloigner du minimum d'erreur recherché et donc peut empêcher une solution acceptable d'être atteinte .

Pour la couche de sortie :

$$\text{D'après (I-25) : } E = \frac{1}{2} \sum_{t=1}^T E(t) = \frac{1}{2} \sum_{t=1}^T (Y_K(t) - R_K^R(t))^2$$

La dérivée de l'erreur  $E(t)$  par rapport au poids  $W_{Kj}^R$  nous donne :

$$\frac{\partial E(t)}{\partial W_{Kj}^R} = -(Y_K(t) - R_K(t)) - \frac{\partial R_K}{\partial S_K^R(t)} - \frac{\partial S_K^R(t)}{\partial W_{Kj}^R(n)} \quad (\text{I} - 32)$$

Le dernier facteur de (I.-32) est :

$$\frac{\partial S_K^R(t)}{\partial W_{Kj}^R(n)} = \frac{\partial}{\partial W_{Kj}^R(n)} \left[ \sum_{J=1}^M W_{Kj}^R(n) \cdot I_j^R(t) + \theta_K^R \right] = I_j^L(t) \quad (\text{I} - 33)$$

En combinant (I-32) et (I-33), on a pour l'opposé du gradient :

$$-\frac{\partial E(t)}{\partial W_{Kj}^R(n)} = (y_K(t) - R_K(t)) F_K^R(S_K^R(t)) \cdot I_j^L(t) \quad (\text{I} - 34)$$

La règle d'adaptation est donnée par :

Avec :

$$W_{Kj}^R(n+1) = W_{Kj}^R(n) + \Delta W_{Kj}^R(n) \quad (\text{I} - 35)$$

$$\Delta W_{Kj}^R(n) = -\mu \frac{\partial W_{Kj}^R(n)}{\partial E} \quad (\text{I} - 36)$$

Si f est une fonction linéaire :  $F_{Kj}^R = 1$

L'équation (I-34) devient :

$$\frac{-\partial E(t)}{\partial W_{Kj}^R(n)} = (Y_K(t) - R_K(t)).I_j^L(t)$$

Alors :

$$\Delta W_{Kj}^R(n) = -\mu \frac{\partial E}{\partial W_{Kj}^R(n)} = \mu \sum_{t=1}^T (Y_K(t) - R_K(t)).I_j^L(t) \quad (\text{I} - 37)$$

$$W_{Kj}^R(n+1) = W_{Kj}^R(n) + \mu \sum_{t=1}^T (Y_K(t) - R_K(t)).I_j^L(t) \quad (\text{I} - 38)$$

Si F est une fonction sigmoïde :

$$F_K^R = F_K^R(1 - F_K^R) = R_K(1 - R_K)$$

L'équation devient :

$$\frac{-\partial E(t)}{\partial W_{Kj}^R(n)} = (Y_K(t) - R_K(t)).R_K(t).(1 - R_K(t)).I_j^L(t)$$

Alors :

$$\Delta W_{Kj}^R(n) = -\mu \frac{\partial E}{\partial W_{Kj}^R(n)} = \mu \sum_{t=1}^T (Y_K(t) - R_K(t)).R_K(t).(1 - R_K(t)).I_j^L(t)$$

$$W_{Kj}^R(n+1) = W_{Kj}^R(n) + \mu \sum_{t=1}^T (Y_K(t) - R_K(t)).R_K(t).(1 - R_K(t)).I_j^L(t) \quad (\text{I} - 39)$$

En définissant la quantité  $\delta_K^R(t)$  telle que :

$$\delta_K^R(t) = (Y_K(t) - R_K(t)).F_K^R(S_K^R(t)) \quad (\text{I} - 40)$$

$$\text{Alors on a : } W_{Kj}^R(n+1) = W_{Kj}^R(n) + \mu \sum_{t=1}^T \delta_K^R(t).I_j^L(t) \quad (\text{I} - 41)$$

**Pour les couches cachées**

$$E = \frac{1}{2} \sum_{t=1}^T E(t) = \frac{1}{2} \sum_{t=1}^T (Y_K(t) - R_K^R(t))^2$$

$$E = \frac{1}{2} \sum_{t=1}^T (Y_K(t) - F_K^R(S_K^R(t)))^2$$



D'après les équations (I-26) et (I-27) nous savons que  $I_j$  dépend des poids de la couche cachée, nous pouvons exploiter ce fait pour calculer le gradient de  $E$ .

$$\begin{aligned} \frac{\partial E(t)}{\partial W_{ji}^L(n)} &= \frac{\partial}{\partial W_{ji}^L(n)} (Y_K(t) - R_K(t))^2 \\ &= \sum_{K=1}^M (Y_K(t) - R_K(t)) \cdot \frac{\partial R_K}{\partial S_K^R(t)} \cdot \frac{\partial S_K^R(t)}{\partial I_j^L(t)} \cdot \frac{\partial I_j^L(t)}{\partial S_j^L(t)}. \end{aligned} \quad (I-42)$$

Les facteurs de l'équation (I-42) peuvent être calculés à partir des équations précédentes

(I-26), (I-27) pour donner :

$$\frac{\partial E(t)}{\partial W_{ji}^L(n)} = \sum_{K=1}^M (Y_K(t) - R_K(t)) \cdot F_K'^R(S_K^R(t)) \cdot W_{Kj}^R \cdot F_j'^L(S_j^L(t)) \cdot X_i(t) \quad (I-43)$$

$$\Delta W_{ji}^L(n) = \sum_{t=1}^T \mu F_j'^L(S_j^L(t)) \cdot X_i(t) \cdot \sum_{K=1}^M (Y_K(t) - R_K(t)) \cdot F_K'^R(S_K^R(t)) \cdot W_{Kj}^R(n) \quad (I-44)$$

D'après l'équation (I - 40) :

$$\Delta W_{ji}^L(n) = \sum_{t=1}^T F_j'^L(S_j^L(t)) \cdot X_i(t) \cdot \sum_{K=1}^M \delta_K^R(t) \cdot W_{Kj}^R(n) \quad (I-45)$$

$$\Delta W_{ji}^L(n) = \sum_{t=1}^T \mu \delta_K^L(t) X_i(t)$$

Avec :

$$\delta_K^L(t) = F_j'^L(S_j^L(t)) \cdot \sum_{K=1}^M \delta_K^R(t) \cdot W_{Kj}^R \quad (I-46)$$

L'équation d'adaptation des poids dans ce cas est :

$$W_{ji}^L(n+1) = W_{ji}^L(n) + \mu \sum_{t=1}^T \delta_j^L(t) \cdot X_i(t) \quad (I-47)$$

## I.12 Les étapes d'algorithme de rétro propagation

L'algorithme de rétro propagation est représenté comme suit :

**Etape 01 :** Initialiser toutes les connexions par des valeurs aléatoires faibles.

**Etape 02 :** Présenter une entrée ( $X_1, X_2, \dots, X_n$ ) et la sortie souhaitée

( $Y_1, Y_2, \dots, Y_m$ ).

**Etape 03 : Calculer :**

- 1-La somme des entrées des neurones de la couche par l'équation (I-26).
- 2-Les sorties des neurones de la couche cachée par l'équation (I-27)
- 3-La somme des entrées de la couche de sortie par l'équation (I-28).
- 4-Les sorties de réseau par l'équation (I-29).

**Etape 04 : Calculer :**

- 1-Les termes de l'erreur pour les neurones de la couche de sortie par l'équation (I-40)
- 2-Les termes de l'erreur pour les neurones de la couche cachée par l'équation (I-46)

**Etape 05 : Ajuster :**

- 1-Les poids de la couche de sortie par l'équation (I-39).
- 2-Les poids de la couche cachée par l'équation (I-47).

**Etape 06 : Revenir à l'étape 03 jusqu'à la stabilisation du système [10].****I.12.1 Exemple d'application**

On choisi une fonction  $y(x) = \sin(x)$  , et après plusieurs essais et choix d'architecture (nombre de couches, nombre de neurones dans chaque couche) avec un pas d'apprentissage variable, on a pu obtenir l'architecture suivante :

- Quatre neurones dans la première couche.
- Trois neurones dans la deuxième couche.
- Deux neurones dans la troisième couche.

Et avec un pas d'apprentissage  $\mu = 0.01$  et après quelques itérations égales à 2000 avec une erreur de  $4.3 \times 10^{-5}$  , voire la figure (I. 11).

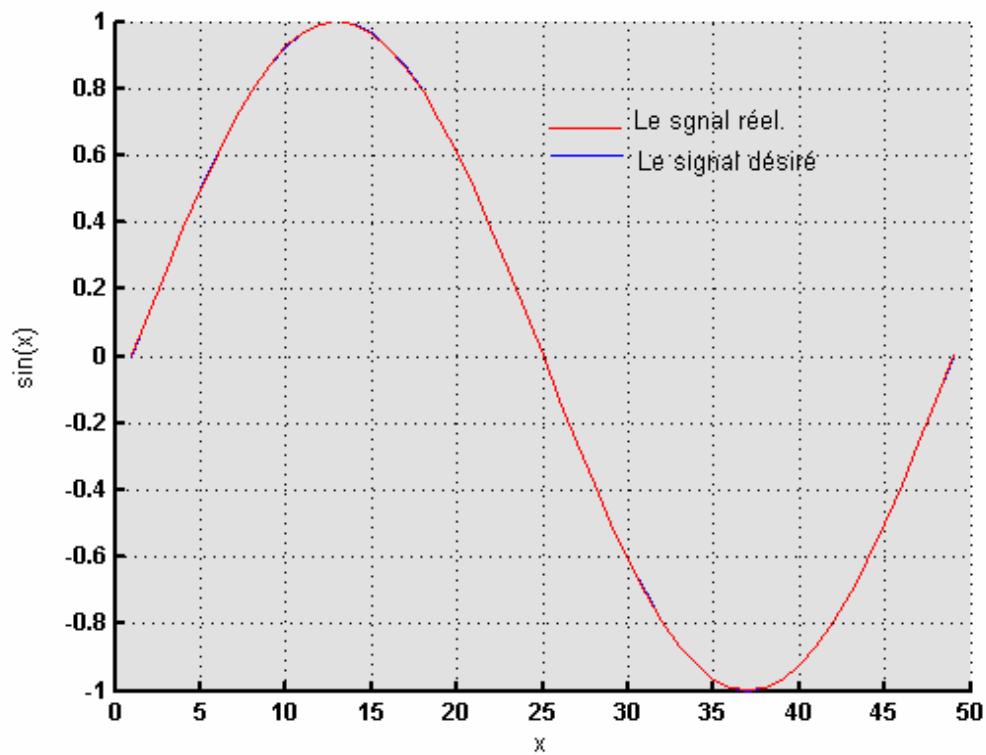


Figure (I.11) l'identification de la fonction  $\sin(x)$  par les réseaux de neurones

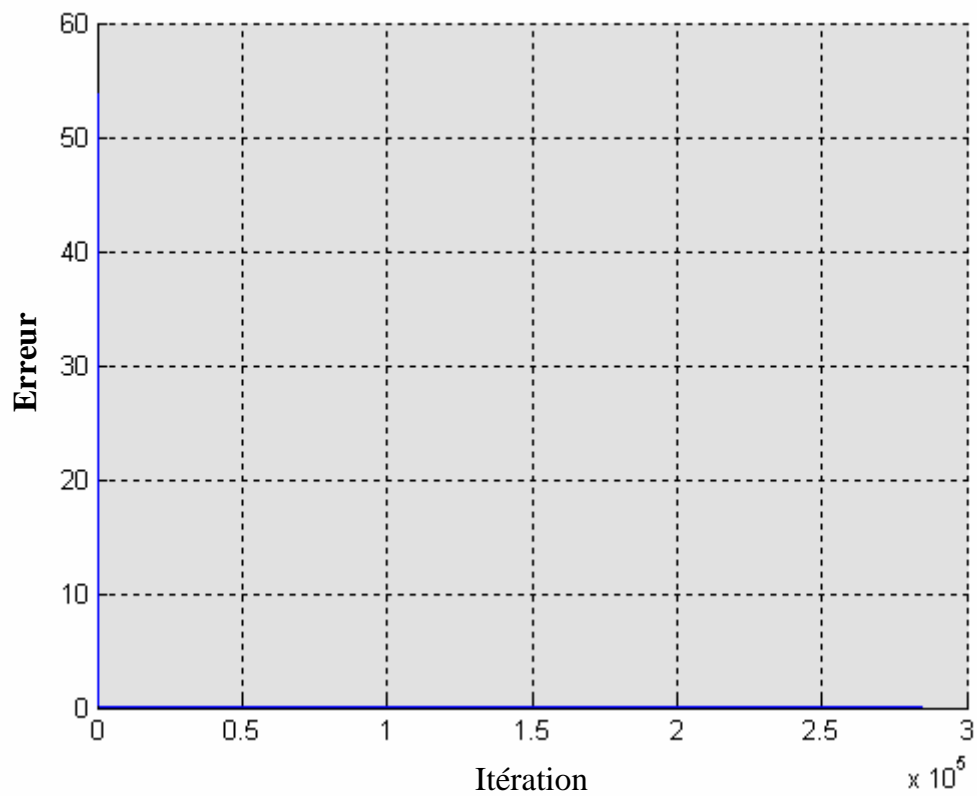


Figure (I.12) Représentation de l'erreur

On trouve que l'identification de la fonction  $\sin(x)$  donne des résultats formidables.

### **I.13 Domaines d'application des réseaux de neurones**

#### **I.13.1 Applications 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 et prototypes), développée par la société Nestor.

#### **I.13.2 Applications bancaires et financières**

Toutes les architectures et les algorithmes sont utilisés pour l'analyse de risque de prêts, la détection de fraude sur les cartes de crédit, l'évaluation de solvabilité d'entreprises, la prévision de cours de marchés de devise, la prédiction de cours de moyen terme de matières premières, etc. Modèles de rendement n'a décrit par exemple l'aide à la délivrance de prêts bancaires, qui, après analyse d'une quantité significative de dossiers (en principe autant de bons dossiers que de dossiers à problème), est capable de réduire la marge de risque du prêteur, lui assurant ainsi à terme un profit accru. Il est difficile de dresser un panorama significatif de réalisations, étant donné le peu d'informations disponibles dans la cas où un système serait véritablement opérationnel. Il faut noter cependant que des colloques scientifiques spécialisés se tiennent désormais sur ce sujet depuis peu d'années.

#### **I.13.3 Traitement de la parole**

C'est un domaine de grande activité et de grande diversité, dans lequel de nombreuses techniques non neuronales ont déjà éprouvées et mises en fonctionnement. Cependant, les difficultés sont nombreuses et conséquentes, et les réseaux de neurones peuvent contribuer à les surmonter. Mais il ne semble pas encore y avoir d'applications industrielles en fonctionnement autrement qu'à l'état de prototype, et qui concernent surtout la reconnaissance de mots isolés. Dans ce domaine, les réseaux de neurones

sont souvent utilisés conjointement à d'autres techniques comme les modèles de Markov cachés.

#### **I.13.4 Industries de consommation**

C'est surtout au Japon, où les consommateurs sont assez sensibles à la valeur ajoutée « technologique » dans un produit que les grandes compagnies ont fait un effort pour annoncer la présence de réseaux de neurones (souvent conjointement avec de la logique floue) dans des produits électroménagers ou audiovisuels. Cette présence est plus symbolique que réelle, ou bien se limite à certaines préprogrammations en usine. Il n'en reste pas moins que le marché grand public peut devenir fortement porteur si des applications décisives voient le jour (en terme de problèmes à résoudre comme en terme de matériels spécialisés, comme les circuits intégrés) [9].

#### **I.14 Conclusion**

Les réseaux de neurones présentés dans ce chapitre sont des systèmes paramétriques non linéaires qui relient des variables d'entrée à des variables de sortie. Les paramètres sont déterminés statistiquement lors d'une phase d'apprentissage à partir d'une base de données d'exemples, appelée base d'apprentissage.

En pratique, l'utilisation de la méthode neuronale pose certaines difficultés. La principale difficulté est l'optimisation de la phase d'apprentissage. Le choix de l'architecture adéquate ou la détermination du « pas d'apprentissage » se fait par essais successifs [10].