

*Chapitre II***LES RESEAUX DE NEURONES ARTIFICIELS (RNAs)****II.1 Introduction**

Les Réseaux de Neurones Artificiels (RNAs) sont des outils puissants capables d'être utilisés dans presque tous les domaines scientifiques, et on peut citer : Le traitement du signal, Vision, parole, prévision, modélisation, aide à la décision,... Ils se caractérisent par une capacité d'adaptation et la possibilité d'intégration dans des systèmes matériels sous la forme de circuit intégré, ou logiciels sous forme de programme informatique implanté dans un ordinateur. Ce chapitre est dédié à l'étude des RNAs depuis leur première apparition jusqu'à nos jours, Un panorama historique est donné à cet effet, ainsi qu'une étude sur le principe de base utilisé pour les RNAs. Les différents types de réseaux (MLP, RBF) sont principalement décrits.

II.2 Historique

Le concept des réseaux de neurones n'est pas nouveau, l'idée était de concevoir un système ayant comme tâche de modéliser la biophysologie du cerveau. Cette modélisation tente d'expliquer comment le cerveau opère et fonctionne. Le but de la recherche sur les réseaux de neurones n'est pas de créer des machines qui traitent l'information plus rapidement que les calculateurs traditionnels, mais c'est de créer des machines qui se montrent supérieures dans les domaines où le cerveau humain dépasse ces calculateurs.

En 1943, Culloch et Pitts adoptèrent les affirmations de James (1890) et formalisèrent une description du neurone qui est l'élément fondamental de tous les réseaux. En 1949, Hebb introduit la notion de " plasticité synaptique ", c'est-à-dire, le mécanisme de modification progressive des couplages entre neurones responsables de changements permanents de leurs propriétés collectives, ce qu'on appelle " apprentissage ". En 1969, Minsky et Papert démontrèrent, dans leur ouvrage intitulé " Perceptrons ", un certain nombre de théorèmes sur les limitations d'un réseau mono-couche et conclurent que ces limitations se généralisent pour les réseaux multicouches. Ceci a poussé de nombreux chercheurs à abandonner cette voie pour se diriger vers l'intelligence artificielle qui semblait un domaine plus prometteur.

Dans le début des années 80, les travaux, de Rosenblatt sur le perceptron et ceux de Widrow et Hoff sur les algorithmes adaptatifs sont s'avèrent les plus importants.

Aujourd'hui, il y a deux groupes de chercheurs ; le premier est constitué des biologistes, physiciens et psychologues, dont le but est de développer un modèle neuronal qui imite d'une manière plus efficace (avec la précision voulue) le comportement du cerveau, quant au second groupe de chercheurs ; est un groupe d'ingénieurs qui étudient les architectures avec lesquelles les neurones peuvent être interconnectés pour former des réseaux dont les capacités de traitement seraient plus puissantes.

II.3 Système nerveux

Le cerveau humain, est le meilleur modèle de 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 (environ 10^{12} neurones), chacune reçoit et envoie des informations (1000 à 10000 synapse par neurone).

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

II.3.1 le corps cellulaire Parfois appelé soma, il contient le noyau de neurone et effectue les transformations biochimiques nécessaires à la synthèse des enzymes et d'autres molécules qui assurent la vie de neurone.

II.3.2 les dendrites Chaque neurone possède des dendrites, qui entourent le corps cellulaire. Parfois les dendrites sont si nombreuses que l'on parle alors de chevelure dendritique ou arborescente dendritique. Elles sont les récepteurs principaux de neurone pour capter les signaux qui lui parviennent (de l'extérieur vers le soma).

II.3.3 l'axone L'axone qui est à proprement parler la fibre nerveuse, sert de moyen de transport pour les signaux émis par le neurone. Les neurones sont connectés les uns aux autres suivant des répartitions spatiales complexes. Les connexions entre deux neurones se font en endroits appelés " synapse " où ils sont séparés par un petit espace synaptique de l'ordre du dizaine d'angströms (10^{-9} m) entre l'axone du neurone afférent et les dendrites du prochain neurone

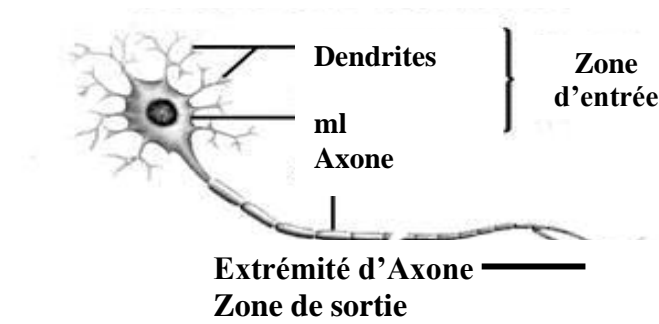


Figure (II.1) Le neurone biologique.

II.4 Fonctionnement des neurones

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

D'après le modèle classique, le soma effectue 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, il provoque à travers la membrane la libération d'un médiateur chimique celle-ci se diffuse jusqu'à la membrane du dendrite qui provoque la naissance d'un potentiel.

II.4 1 Le neurone formel :

Le neurone formel est le modèle mathématique du neurone biologique. Son travail consiste à faire une sommation pondérée de ses entrées provenant de l'extérieur ou de la sortie d'un autre neurone artificiel, le résultat obtenu est ensuite calculé en utilisant une fonction non linéaire, appelée fonction de seuil. La figure (II.2) montre le modèle d'un neurone artificiel.

L'élément de base d'un réseau de neurones est, bien entendu, le neurone artificiel. Un neurone contient deux éléments principaux:

- un ensemble de poids associés aux connexions du neurone,
- une fonction d'activation.

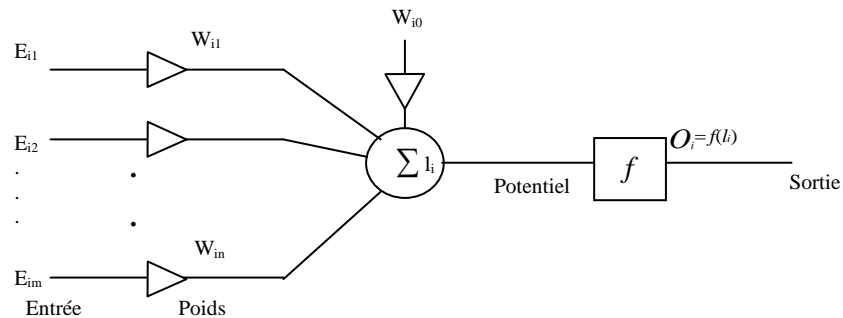


Figure (II.2) Le neurone formel

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 [16].

II.4.1 Fonction d'activation

C'est une fonction non linéaire appelée aussi fonction de seuil, elle présente la relation qui lie la fréquence moyenne des potentiels d'action (Y_j) limités en amplitude, au potentiel somatique (S_j). Le potentiel d'action (Y_j) sert ensuite à exciter les autres neurones qui lui sont connectés.

La transformation non linéaire appliquée au potentiel somatique, traduit deux caractéristiques importantes du neurone :

.l'effet du seuil.

.la saturation de la réponse au de là d'une certaine valeur du potentiel somatique.

.La fonction d'activation F est de nature très variée, elle peut être déterministe, continue, discontinue ou aléatoire.

La figure (II-3) donne quelques modèles des fonctions d'activation utilisées.

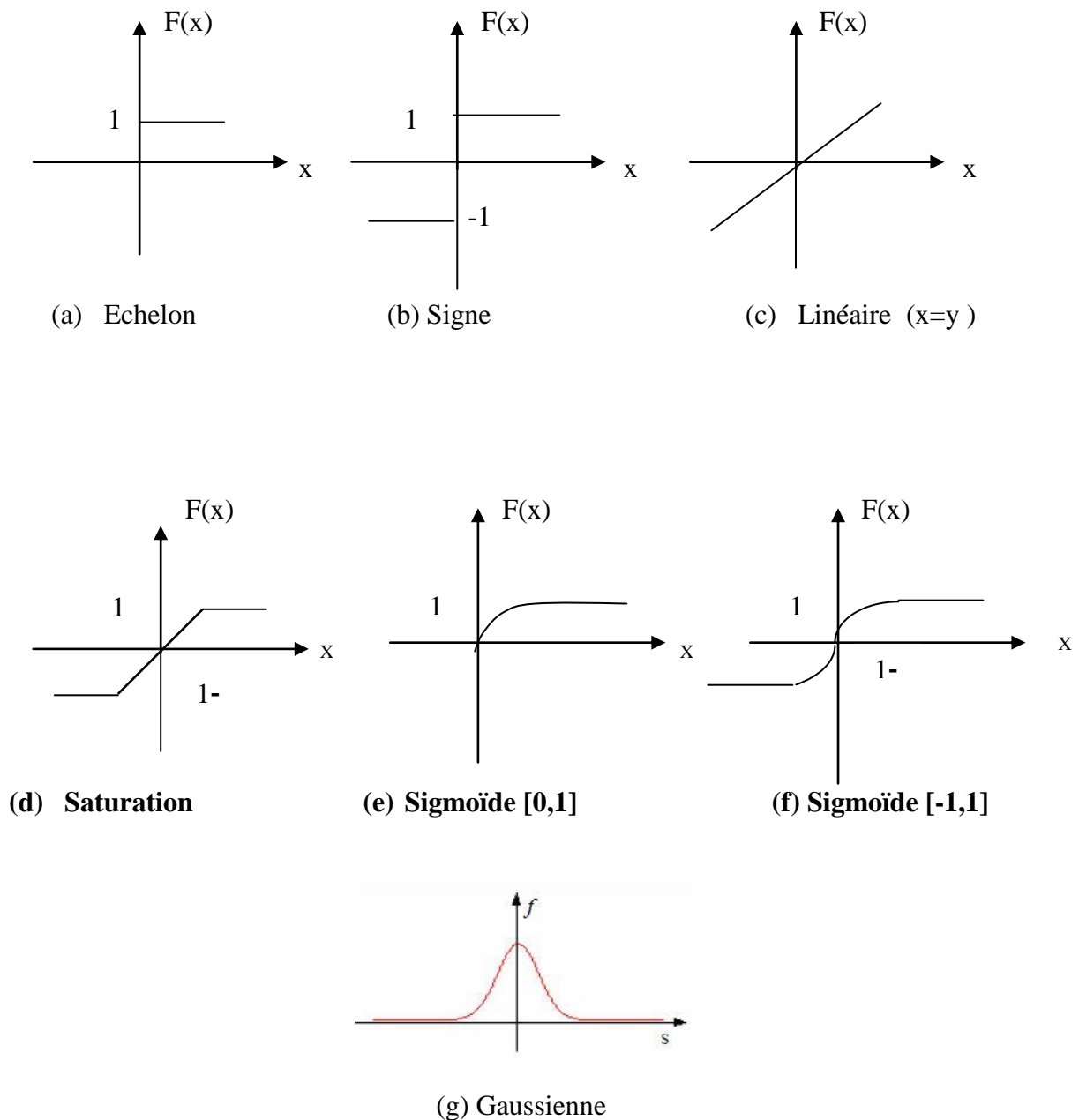


Figure (II.3) Les fonctions d'activation les plus utilisées

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

Dans de nombreux modèles formels des neurones, la fonction d'activation est une sigmoïde, la cause de ce choix vient du fait que la décision n'est pas tranchée (cas b) mais elle évolue graduellement entre -1 et +1.

II.5 Les réseaux de neurones

II.5.1 Définition

Le réseau est un graphe orienté et pondéré dont les nœuds sont des neurones formels, et les transitions sont des arcs pondérés appelés "liens synaptiques". On appelle "poids synaptiques" la pondérations d'un lien synaptique.

Les neurones sont reliés entre eux par une connexion. L'architecture du réseau peut varier selon la structure de ces connections. En général un réseau de neurones est défini comme étant l'association de plusieurs élément de traitement de l'information (neurones), interconnectés les un aux autres suivant des modèles qui peuvent être complexe

II.5.2 Propriété des réseaux de neurones

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

II.5.2.1 Le parallélisme : Cette notion se situe à la base de l'architecture des réseaux de neurones considérés comme ensembles d'entités élémentaires qui travaillent simultanément.

II.5.2.2 la résistance aux pannes : à cause de l'abondance des entrées et la structure du réseau. Les données bruitées ou les pannes locales dans certain nombre de ses éléments n'affectent pas ses fonctionnalités. Cette propriété résulte du fonctionnement collectif et simultané des neurones qui le composent.

II.5.2.3 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 systèmes dynamiques.

II.5.2.4 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 a appris.

II.6 L'utilisation des réseaux de neurones :

Les réseaux de neurones fournissent une technique pour obtenir la capacité traitement exigée en utilisant un grand nombre d'éléments de calcul simples fonctionnant en parallèle et c'est ce qui fait la puissance des réseaux de neurones, plusieurs défis et problèmes au quels ont fait face ingénieurs et scientifiques et dans les quelles les RNA ont apportées une avance considérable et des solutions fructueuses, et de ce succès les RNA sont devenus maintenant un vrais outil de résolution de problèmes ,dont on vas citer quelques uns : [17]

II.6.1 Classification :

La tâche de classification est d'assigner un modèle d'entrée (un signal vocal ou des caractères manuscrits) représenté par un vecteur de caractérisant une des classes préspecifiée. Des applications bien connues de la classification est la reconnaissance de caractères, reconnaissance de la parole, classification des signaux EEG classification des cellules sanguines, l'inspection des circuits imprimés.

II.6.2 Catégorisation (Clustering) :

Dans la catégorisation aussi connue comme la classification non supervisée, il n y a pas de modèles d'apprentissage avec bien connue. L'algorithme de catégorisation explore et teste la similarité entre les différents modèles qui lui sont présentés, et place les modèles similaires dans une même catégorie (cluster).

II.6.3 Approximation de fonctions :

Plusieurs problèmes scientifiques et d'ingénieries requièrent l'approximation. Ayant un ensemble de N paires d'apprentissage (entrée-sortie), générées par une fonction inconnue $M(x)$, la tâche d'approximation de fonction est de trouver une estimation, disons f , de la fonction inconnue M .

II.6.4 Prédiction :

Ayant un ensemble de N échantillons dans une séquence temporelle déterminé, $\{y(t_1), y(t_2), \dots, y(t_N) / t_1 < t_2 < \dots < t_N\}$, le but est de déterminer la valeur de $y(t_{N+1})$ à l'instant future t_{N+1} . La prédiction est très utilisée dans la bourse, le contrôle des procédés, prévisions météorologiques.

II.6.5 Optimisation :

Un problème d'optimisation peut généralement comprendre les éléments suivants :

.Un ensemble de variables indépendantes qui font référence à l'état du processus

- .Une fonction objective (fonction de coup/erreur) qui doit être optimisé.
- .Un ensemble de contraintes si elles existent bien sure.

Le but de l'optimisation est de trouver un état qui satisfait ces contraintes de telle façon que la fonction objective soit optimisée.

II.6.6 Mémoire associative :

La mémoire associative, peut être accédée par l'intermédiaire de sa contenance, et les données de cette mémoire peuvent être reconnues par cette dernière même si ils sont bruités ou seulement une partie d'un modèle d'entrée est disponible. E,g la récupération bibliographique des références d'une revue de l'information partielle, de ce fait la mémoire associative est très appréciée dans la construction des bases de données multimédia.

II.6.7 Contrôle :

Considérons un système définie par la paire $\{u(t), y(t)\}$ où $u(t)$ est le contrôle du système et $y(t)$ est la sortie résultante à l'instant t , (dans le contrôle adaptatif) le but est de générer une entrée de contrôle $u(t)$ de façon que le système suit la sortie désirée déterminée par le modèle de référence.[18]

II.7 Type d'architecture des réseaux de neurones artificiels

II.7.1 Les réseaux de neurones non bouclés

Un réseau est non bouclé, ou statique, si son graphe ne possède pas de cycle. Il réalise donc, de manière générale, une relation *algébrique* non linéaire entre ses entrées et ses sorties. Cette forme a l'avantage de faire apparaître les entrées effectives du réseau à chaque instant, et de faciliter l'apprentissage (car toutes les connexions sont de même type). Ses entrées sont ordonnées, et les connexions ne peuvent aller que d'une entrée à un neurone dont l'indice est supérieur, figure (II.4) Chaque neurone i du réseau calcule à l'instant k . [17]

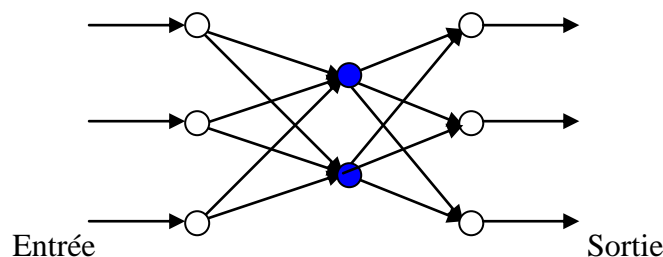


Figure (II.4) Forme d'un réseau non bouclé

II.7.2 Les réseaux de neurones bouclés

Un réseau est bouclé, ou dynamique, si son graphe possède au moins un cycle. Il constitue un filtre récuratif non linéaire à temps discret. En effet, tout réseau de neurones bouclé à temps discret d'ordre NS peut être représenté par un réseau dont la dynamique est décrite par NS équations aux différences couplées d'ordre 1, mettant en jeu NS variables d'état, et NI entrées externes, figure 2.5. Cette forme canonique n'est en général pas unique [18].

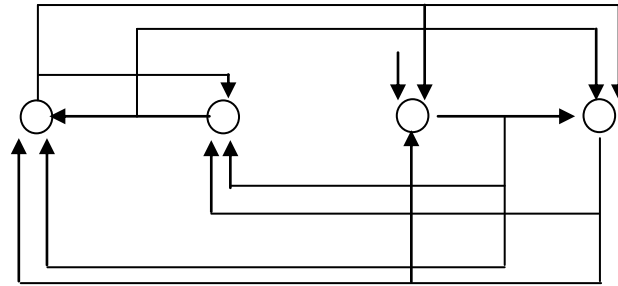


Figure (II.5) Forme d'un réseau bouclé

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 [17].

II.8 Quelque modèle des réseaux de neurones

II.8.1 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.

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 (II.6).

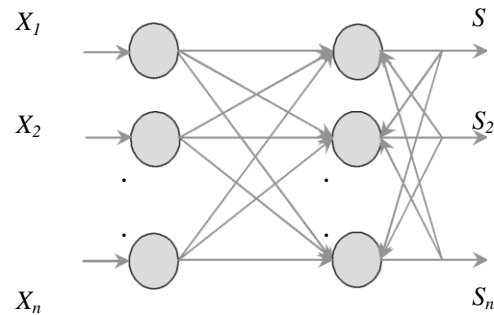


Figure (II.6) Le modèle de Kohonen

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.

II.8.2 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 [18].

II.8.2.1 L'architecture

Les modèles connexionnistes de Hopfield sont constitués de neurones formels de type McCulloch 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.

Formellement, comme la montre la figure (II.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 , $w_{ij} = w_{ji}$) [17].

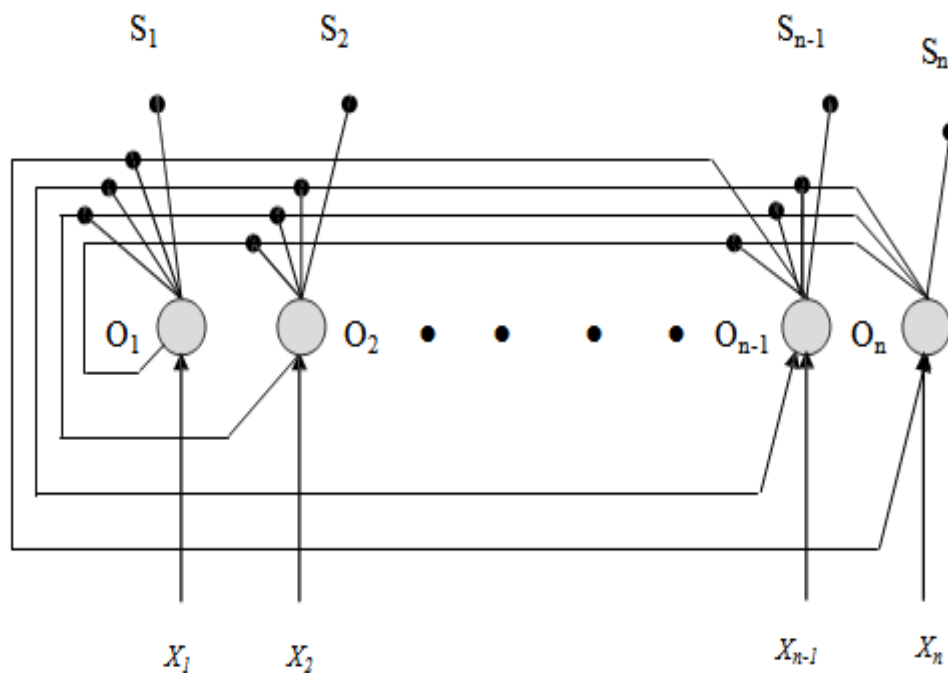


Figure (II.7) Réseau de Hopfield

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 [18].

II.8.3 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.

II.8.3.1 Structure du perceptron

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

II.8.3.2 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

II.8.3.3 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} ,

II.8.3.4 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 sil'on influe sur les connexions et la nature des fonctions f et h .

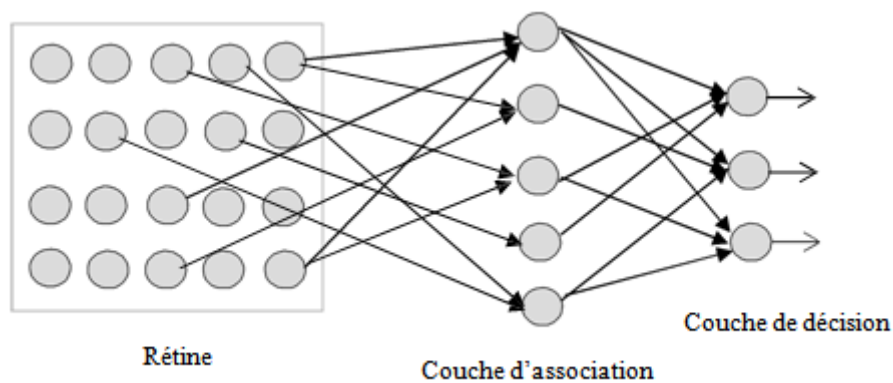


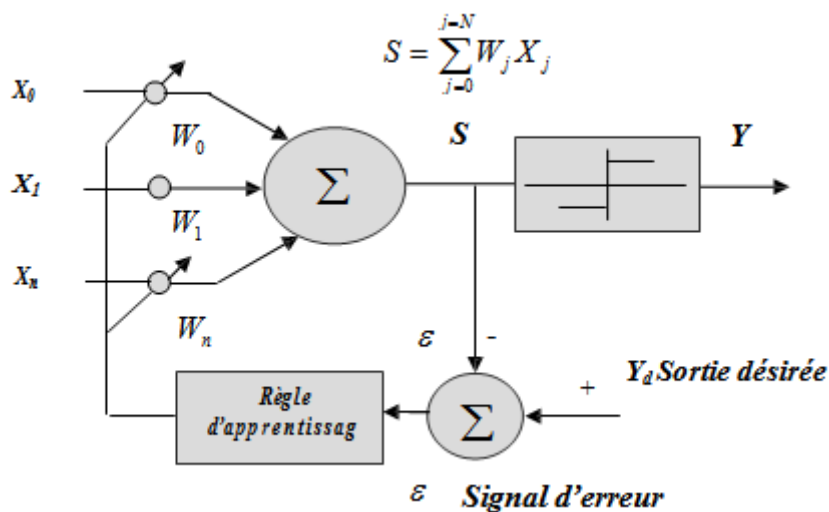
Figure (II.8) Schéma d'un Perceptron

II.8.3 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 (II.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 [18].



Figure(II.9) Schéma de principe de l'adalin

II.9 L'apprentissage dans les 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 lorsque 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 comportements [16].

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.

Le mécanisme d'apprentissage d'un réseau comprend la récurrence des phases suivantes

- Le réseau est stimulé par l'environnement,
- en réponse à cette stimulation, le réseau adapte son comportement,
- .Le réseau réagit alors différemment à l'environnement en fonction de la nouvelle expérience acquise consécutivement à la stimulation.

L'apprentissage des réseaux dépend des informations qui lui sont fournies à son entrée et à sa sortie. En tenant compte de la façon dont on peut lui faire fournir les informations et lui faire apprendre à les assimiler, c'est à dire le guider ou non durant son apprentissage. De la, il apparaît deux types d'apprentissages : l'apprentissage supervisé et l'apprentissage non supervisé .

Les algorithmes d'apprentissages donnent de meilleurs résultats lorsqu'on leur fournit des exemples multiples et variés ; ainsi le réseau peut assimiler toutes les connaissances. Ils existent différentes règles d'apprentissages parmi lesquelles on peut distinguer [16]:

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

II.9.1 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 et al, 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 [16].

L'apprentissage supervisé "Supervised Learning", repose sur le fait que les exemples sont des couples (Entrée, Sortie associée), figure (II.10) 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 [16]

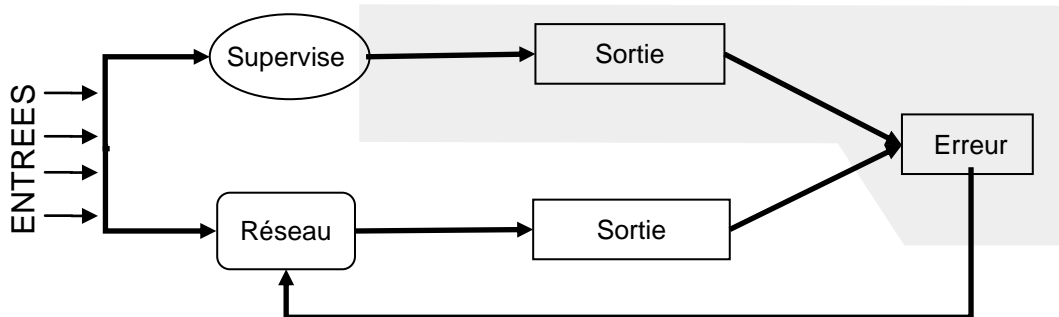


Figure (II.10) Illustration de l'apprentissage supervisé

II.9.2 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 mais qu'il s'auto adapte alors qu'il ne dispose que des valeurs (Entrée), figure (II.11) Remarquons cependant que les modèles d'apprentissage non supervisé nécessite avant la phase d'utilisation une étape de labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision.[16]

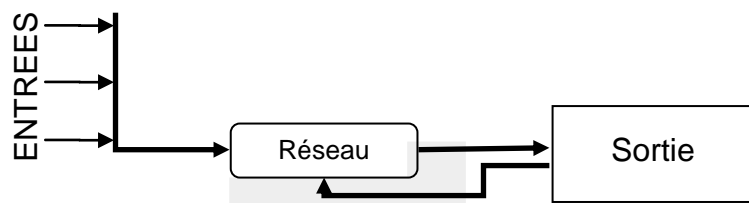


Figure (II.11) Illustration de l'apprentissage non supervisé

Tel que X et Y les vecteurs d'entrée et de sortie. On distingue des réseaux à deux couches tel que le perceptron et l'adaline (Adaptative Linear Neuron). 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.

II.10 Les réseaux de neurones à base radiale

Les réseaux de neurones à base radiale sont une classe particulière des réseaux de Neurones multicouches.

II.10.1 Présentation des réseaux RBF

L'idée générale des réseaux RBF dérive de la théorie d'approximation des fonctions, ces réseaux sont une architecture Feedforward puissante. Ce type de réseaux a été Introduit pour la première fois par Hardy, et la théorie correspondante a été développée par Powell, ensuite, ces réseaux ont pris le terme de réseaux de neurones grâce à Broomhead et Lowe. Sans oublier les oeuvres de MOODY et DARKEN (1989) d'une part, et de POGGIO et GIROSI (1990) d'autre part. La raison de son application vient du fait que le réseau utilise des fonctions gaussiennes standard qui sont à symétrie radiale, Son apprentissage est basé sur l'algorithme K-means et l'algorithme des moindres carrés [16].

Les réseaux de neurones RBFs, sont principalement utilisés pour résoudre des problèmes d'approximation de fonctions dans des espaces de grandes dimensions.

Ils sont lus adaptés, en raison d'apprentissage local. Ce type d'apprentissage peut rendre le processus d'entraînement bien plus rapide que dans le cas d'un MLP, qui apprend de façon globale.

II.10.2 Architecture générale d'un réseau RBF:

Pour des raisons de simplicité, on a décidé de faire une petite dualité entre le réseau RBF et le PMC, en précisant les ressemblances et les différences entre les deux types de réseaux. Ce choix est justifié par la popularité des PMCs et leur vaste utilisation dans les applications industrielles. Un réseau de neurone de type RBF est un PM spéciale, son architecture est identique à celle d'un PMC à une seule couche cachée donc on peut dire qu'il prend toutes les caractéristiques d'un PMC simple sauf qu'il diffère en quelques points nous citons quelques uns [16] :

-Le nombre des couches cachées :

Un réseau RBF ne peut contenir qu'une seule couche cachée, son architecture est fixée pour tous les problèmes à étudier.

-La fonction d'activation :

Ce sont les réseaux que l'on nomme aussi RBF ("Radial Basic Functions"). l'architecture est la même que pour les MLP cependant, les fonctions de base utilisées ici sont des fonctions Gaussiennes

-Les poids synaptiques :

Les poids entre la couche d'entrée et la couche cachée dans les modèles neuronaux de type RBF sont toujours d'une valeur d'unité, c'est-à-dire que l'information inscrite sur la couche d'entrée sera retransmise sans distorsion vers les neurones de la couche cachée. En ce qui concerne les ressemblances entre un réseau RBF et un PMC, on peut mentionner quelques points :

-La fonction de sortie :

Généralement une simple fonction linéaire qui renvoie une sommation pondérée des valeurs calculées par les neurones de la couche cachée. Bien sûr, ce n'est pas toujours le cas, parfois l'utilisation d'autres fonctions pourrait être plus adéquate dans un problème donné.

-Le sens des connexions :

Les connexions entre les couches suivent le même sens, on peut dire qu'elles ne sont pas récurrentes, et chaque neurone est entièrement connecté vers les neurones de la couche suivante.

-L'apprentissage :

Pour calculer les poids de la couche de sortie, on utilise un apprentissage supervisé pour les deux types de réseaux.

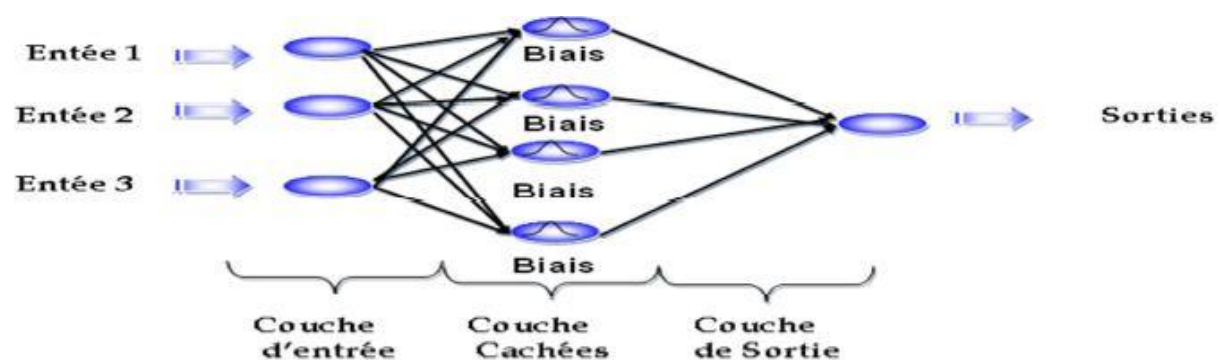


Figure (II.12) Architecture d'un Réseau de Neurone RBF

II.11 Le perceptron multi-couches

Comme nous l'avons déjà dit le cerveau humain est composée de milliers de neurones, alors il est évident qu'un simple neurone, ne peut rien faire à lui tout 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 [19].

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 (PMC 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.

II.11.1 Architecture du PMC

La sortie désirée de notre réseau est notée d_k . Le réseau possède N_0 entrée, $L-1$ couches cachées contenant chacune N_i neurones ($1 < i < L-1$) et une couche de sortie avec N_L neurones. Chaque neurone est caractérisé par un couple d'indice (j,k) , où j désigne le nombre de couches et k le nombre de neurones, figure (II.13). Le coefficient synaptique est désigné par w_{jki} où le troisième indice i indique le numéro des neurones transmetteurs. Le signal y_{jk} est la somme pondérée de toutes les entrées du neurone jk . x_{jk} est la sortie non linéaire du neurone jk et f est une fonction d'activation choisit comme suit : $f(y) = \frac{1 - \exp(-a \cdot y)}{1 + \exp(-a \cdot y)}$ où

'a' représente le seuil de la fonction d'activation.

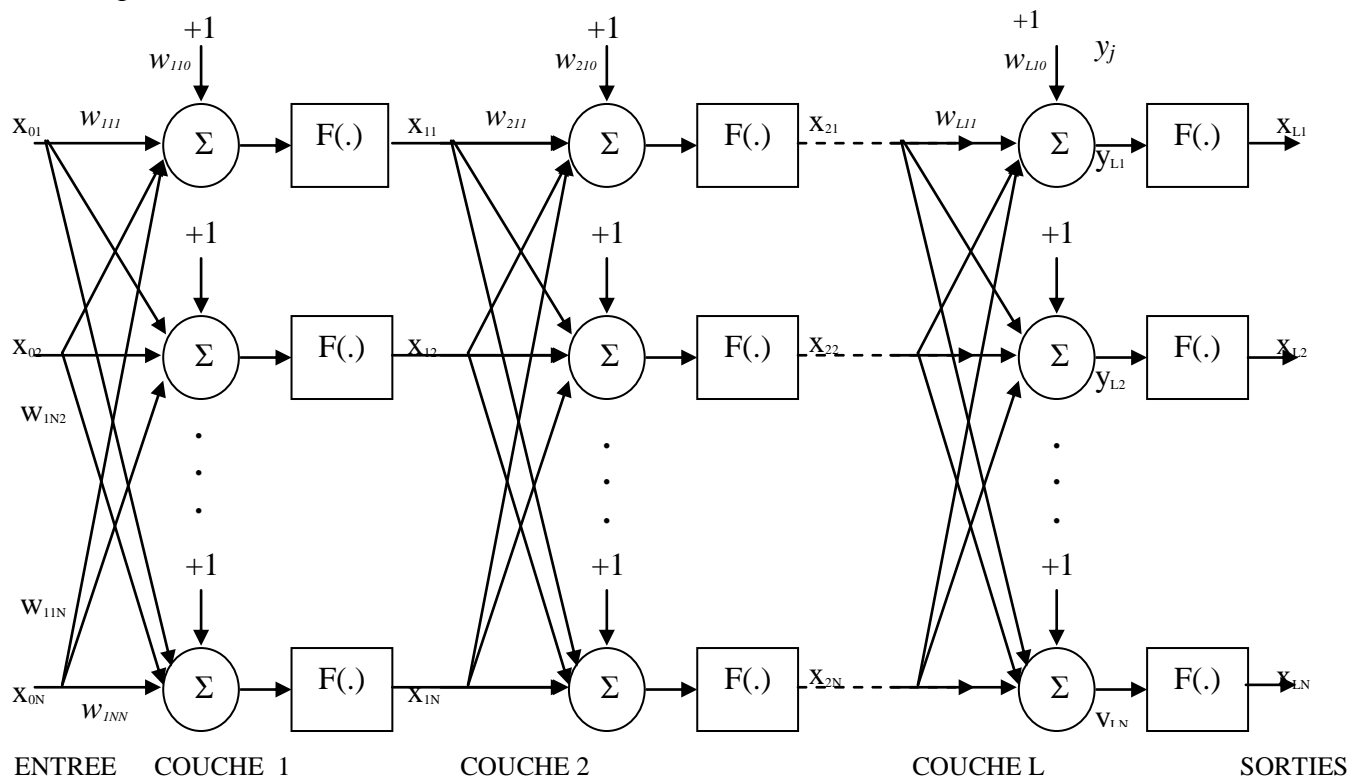


Figure (II.13) Architecture MLP

II.11.1 Algorithme de rétropropagation du gradient

L'algorithme de rétropropagation (Backpropagation / BP) est le plus utilisé dans l'apprentissage des réseaux MLP. Il consiste :

- 1- Initialiser les coefficients synaptiques avec des valeurs petites et aléatoires
- 2- Présenter au réseau les exemples à apprendre. Choisir un exemple et procéder à la propagation et la rétropropagation (étapes 3et4)
- 3- La propagation : l'activation des neurones se propage de la couche d'entrée à la couche de sortie.

Les activités internes du neurone k dans la couche j sont :

$$y_{jk} = \sum_{i=0}^{N_{j-1}} x_{j-1,i} w_{jki}$$

la sortie non linéaire est calculée d'après :

$$x_{jk} = f(y_{jk}) = \frac{1 - \exp(-a \cdot y_{jk})}{1 + \exp(-a \cdot y_{jk})}$$

- 4- rétropropager l'erreur : On calcule la dérivée de $f(y_{jk})$ en utilisant :

$$f'(y_{jk}) = \frac{2a[\exp(-a \cdot y_{jk})]}{[1 + \exp(-a \cdot y_{jk})]^2}$$

on calcule l'erreur à la couche de sortie (j=L) en évaluant la valeur suivante :

$$e_{Lk} = f'(y_{Lk}) \cdot (O_k - x_{Lk}) \text{ pour chaque neurone k, où } O_k \text{ est la } k^{\text{ème}} \text{ sortie désirée.}$$

Après cela, on calcule le signal d'erreur dans les couches cachées pour toute valeur de j=L-1 jusqu'à 1 par :

$$e_{jk} = f'(y_{jk}) \sum_{i=1}^{N_{j+1}} e_{j+1,k} w_{j+1,k,i}$$

- 5- adapter les coefficients synaptiques par :

$$w_{jki}^{(n+1)} = w_{jki}^{(n)} + \mu e_{jk}^{(n)} x_{j-1,i}^{(n)} + \alpha (w_{jki}^{(n)} - w_{jki}^{(n-1)})$$

μ : est le pas d'apprentissage

α : le momentum

- 6- présenter les paramètres pour une nouvelle itération jusqu'à que les coefficients synaptiques se stabilisent autour d'une valeur et l'erreur quadratique totale du réseau

$E = \sum_{p=1}^m \sum_{k=1}^{N_L} (O_k^p - x_{Lk}^p)^2$ soit inférieure à un certain seuil, m est le nombre totale des paramètres d'apprentissage. O_k^p et x_{Lk}^p représente respectivement la $k^{\text{ème}}$ sortie désirée et la $k^{\text{ème}}$ sortie du réseau pour la $p^{\text{ème}}$ paramètre.

En plus, il est possible d'arrêter l'apprentissage en fixant une limite au nombre d'itérations. Noter que l'ordre de présentation des exemples doit être aléatoire. Généralement le pas d'apprentissage et le momentum doivent être adaptés quand le nombre d'itération augmente.

II.11.2 Le minimum local

Le RNA 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, figure (II.14), 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 minimum locaux induisant ainsi notre réseau en erreur, 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. Des méthodes sont à appliquées afin d'éviter les minima locaux [20] :

- 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.
- Relancer l'apprentissage plusieurs fois en utilisant des poids initiaux différents.
- Ajouter du bruit aléatoire aux poids du réseau .

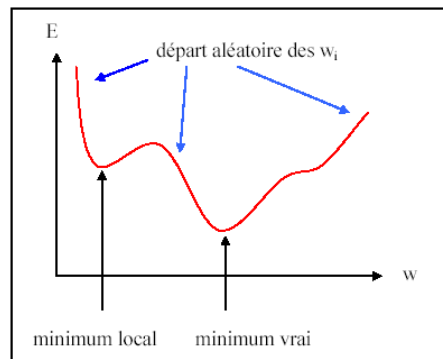


Figure (II.14) Courbe explicative du phénomène du minimum local

Une solution a été proposée par Matyas afin d'éviter ce problème, elle se base sur une solution stochastique qui est connue sous le nom de méthode d'optimisation aléatoire (ROM : Random Optimization Method). Une autre solution basé sur le concepts de la deuxième dérivé, elle consiste dans l'algorithme de Levenberg Marquardt, qui combine rapidité d'apprentissage et efficacité[20].

II.11.3 Algorithme de Levenberg Marquardt

La méthode de Levenberg-Marquardt (LM) [20] consiste à modifier les paramètres selon la relation suivante :

$$\theta_k = \theta_{k-1} [H_{k-1} + \lambda_{k-1} \cdot I]^{-1} \nabla J_{k-1} \quad \text{Avec } I = \text{Matrice Identité}$$

Cette méthode est particulièrement astucieuse car elle s'adapte d'elle-même à la forme de la fonction de coût. Elle effectue un compromis entre la direction du gradient et la direction donnée par la méthode de Newton. En effet, si λ_{k-1} est grand, on reconnaît la méthode du gradient (dans ce cas la valeur du pas est donnée par $1/\lambda_{k-1}$) et si λ_{k-1} est petit, la modification des paramètres correspond à celle de la méthode de Newton.

Cette méthode permet d'éviter les inconvénients du choix du pas et du nombres d'itérations, car elle choisit automatiquement un compromis entre la direction du gradient et la direction de Newton. Nous choisissons une valeur initiale de λ_0 (Bishop propose $\lambda_0 = 0.1$) qui est modifiée durant l'optimisation. A chaque itération, on calcule la fonction de coût $J(\theta)$ avec la valeur de λ précédente ; si la fonction de coût diminue, on effectue la modification des paramètres et on diminue λ (par exemple, divisé par 10) ; si la fonction de coût croît, on recherche à se rapprocher du gradient et on augmente λ (multiplie par 10) jusqu'à ce que le coût diminue. Cette méthode présente un intérêt pratique car elle peut être utilisée sans avoir à choisir le pas . Partant d'une valeur initiale pour p, la méthode de Levenberg-Marquardt se résume en :

1. Calculer $f^2(p)$;
2. Fixer λ à une valeur faible, par ex. 10^{-3} ;
3. Dédire δp_j du système linéaire, puis évaluer $f^2(p + \delta p)$;
4. Si $f^2(p + \delta p) > f^2(p)$, *accroître* λ d'un ordre de grandeur, par ex. $\times 10$, et retourner au point 3 ;
5. Si $f^2(p + \delta p) < f^2(p)$, *décroître* λ d'un ordre de grandeur, par ex. $\div 10$, remplacer p par $p + \delta p$ et retourner au point 3 ;
6. Arrêter le processus si le pas δp ne modifie f^2 que d'une quantité $\ll 1$.

II.12 Conclusion

Dans ce chapitre, nous avons fait un rappel sur les réseaux de neurones artificiels et présenté les différentes architectures des RNAs, ainsi que leurs méthodes d'apprentissage. Une étude a été faite sur le réseau MLP et RBF avec quelques algorithmes d'apprentissage (BP, ROM, LM). Il est clair que les RNAs sont connus pour leurs caractéristiques d'approximation universelle. Ainsi, ils sont bien placés pour approximer des fonctions non linéaires, ce qui correspond bien avec notre problématique, à savoir la prédiction de l'irradiation solaire par le biais des réseaux de neurones artificiels.

Afin de tester les performances des réseaux de neurones dans la prédiction météorologique nous allons utiliser le réseau RBF, qui fera l'objet du prochain chapitre.