5/98

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

D.E.R de Genie Electrique & Informatique
Filiere : Electronique

PROJET DE FIN D'ETUDES

Pour l'obtention du diplôme d'ingenieur d'etat en Electronique

RECONNAISSANCE DE FORMES (CARACTERES ET CHIFFRES ARABES IMPRIMES) PAR RESEAUX CONNEXIONISTES (Etape d'apprentissage)

Proposé par :
M[r] L. SAADAOUI

Dirigé par
M[me] L. HAMAMI
M[elle] C. FIALA

Réalisé par :
M[elle] ZEGHOUANE MOUNIRA
M[elle] BENZAÄMIA HAFIDA

***Promotion** : Juin 1998*

E.N.P 10, Avenue Hacen Badi El-Harrach - ALGER

DEDICACES

A mes très chers parents, pour leurs sacrifices, encouragements et leur présence a tout moment.

A mes deux chers sœurs : Djahida et Ibtissem.

A mes chers frères et surtout Abderaouf.

A mes tantes et oncles.

A toute la famille Benzaamia, plus spécialement Benzaamia Abd elkader.

A tous ceux et celles qui me sont chers.

A mes amies :Mounira , Nadia, Samira, Faiza, Malika,..

Je dédie ce modeste travail.

Hafida

Dedicaces

Aux yeux de mes yeux :

Ma Mère et Mon père.

A mes deux frères et surtout notre benjamin Samir.

A mes sœurs.

A mes chers Youcef et abdel Raouf.

A mon beau frère Salah.

A ma belle sœur Souad.

A mes amies respectivement Hafida, Nadia, Channez, Louiza , Rafika.....

A tous ceux à qui je tiens et que j'ai omis à citer.

A toute la communauté musulmane.

Mounira

Remerciements

Nous remercions Dieu de nous avoir permis de mener à bien de notre entreprise. Que son nom soit Glorifié.

Notre remerciement va ensuite à Melle C. Fiala qui a placé sa confiance en nous et en capacités et ce jusqu'au bout.

Un grand remerciement aussi pour Mme Hammami et Mr Saadaoui qui nous ont guidé et encouragé tout le long de notre travail.

Nous tenons à remercier Mr Zekrini pour le soutien qu'il nous a apporté afin de mener à bien ce projet.

Enfin, que toutes les personnes ayant contribué, de prés ou de loin, à la réalisation de ce projet trouvent ici, l'expression de notre profonde gratitude.

Sommaire

ملخص

كانت الشبكات العصبية تشكل عند ظهورها، محاولة لتجسيد خصائص العقل البشري، لغرض الاستفادة منها. حاليا اصبحت هده الاخيرة، تمثل مجموعة من العوامل الغير خطية.

تجميع هده العوامل في شكل شبكات ، يمكن ان تكون لها فعالية عند استعمالها في مختلف الميادين ونذكر على سبيل المثال معالجة الاشارات ، التصميم و التحكم الآلي للجمل.

الهدف من هدا العمل هو ابراز التطبيق المباشر للشبكات العصبية دات التمرن المشرف عليه ، للتعرف على الاشكال(الاحرف العربية المطبوعة و الاعداد العربية).

خوازمية التمرن المشرف عليه تكون عن طريق الانتشار العكسي للمشتق المعتمدة على مبدا ضبط معاملات التعديل لغرض تصغير القيمة التربيعية للخطا على مجمل عينات الاختبار ، ونظرا لحدود هده الخوارزمية ؛استعملنا بنية النوافد بنفس المبدا ، لاجتناب تشبع الاعصاب الا صطناعية وتقليص وقت التمرن.

اخيرا مرحلة التعرف تكون عن طريق اختبار ؛لاخد القرار.

Résumé:

Les réseaux de neurones constituaient lors de leur création, une tentative visant à mimer le cerveau biologique afin de benéficier de ses caractéristiques trés interessantes. Actuellement, ces derniers désignent un ensemble d'opérateurs non-linéaires qui rassemblés aux réseaux, peuvent étre efficacement utilisés dans plusieurs opérations relevant de différentes disciplines.
Nous citons le traitement de signal, la modélisation et la commande des systèmes.

L'objectif de notre travail est de présenter une application directe des réseaux de neurones à apprentissage supervisé appliqué à la reconnaissance de formes(caractéres et chiffres arabes imprimés).

L'algorithme d'apprentissage est la rétropropagation du gradient basé sur l'ajustement des coefficients de pendération afin de minimiser la valeur quadratique de l'erreur sur l'ensemble d'entraitement. A cause des limitations de la structure multicouche, nous avons adopté à la structure en fenètres pour éviter la saturation des neurones et réduire le temps d'apprentissage.
L'étape de reconnaissance se réalise par test et une prise de décision.

Abstract:

The objective of our work is to present a direct application of neuronals networks Applied on the shapes recognition(arabic printed characters and numbers).

Supervised apprentice ship algorithm is the backpropagation based on adjustement of weights of the network in a way to minimize the error of training data. Because limitations of this algorithm, we use windows structure to avoid neural saturation and to reduce apprentice ship time.

Finally, the recognition step consists of the decision after test.

INTRODUCTION GENERALE

Introduction générale :

Le grand nombre de données, leur variabilité, la nécessité de traitement en temps réel, et le fait que les problèmes posés ne répondent à aucun modèle physique clair, laisse parfois l'ingénieur démuni devant des taches de reconnaissance, de caractérisation ou de prise de décisions. Parmi ces opérations, on trouve la modélisation de systèmes, le contrôle de processus, le filtrage ou la reconnaissance de signaux.

*F*ace au développement récent de la biologie moderne et des neurosciences, la recherche ne peut rester indifférente aux multiples retombées que les développements ont engendrées. L'une des évolutions les plus marquée, qui a été engendrée, et celle de l'introduction des réseaux de neurones dans le domaine des sciences de l'ingénieur.

*E*n effet, l'idée de chercher dans la structure du système nerveux une inspiration pour la conception est ancienne. Tout à commencé en 1943 puis elle est tombée dans l'oubli[7]. Dans les années quatre-vingts, le domaine explosait alors très rapidement, le développement des calculateurs puissants et de la technologie *VLSI* (Very large Scale Integration) sont les causes principales qui ont rendu possibles les diverses applications des réseaux de neurones. De plus, le développement d'algorithmes, performant l'apprentissage de ces réseaux, leur a ouvert de nouvelles perspectives d'utilisation.

*L*a rétro-propagation 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 fonction dérivable[5].

*N*otre application est une simulation de la reconnaissance par réseaux connexionnistes, qui ont été choisis à cause de leurs avantages comme le parallélisme,la mémoire distribuée et ce que peut fournir une architecture bien déterminée du réseau comme résultats concernant la capacité de généralisation et la traitement en temps réel de l'image qui se présente à chaque test.

Le présent travail porte sur une application directe des réseaux de neurones. Les différentes structures des réseaux de neurones seront présentées ainsi que l'algorithme d'apprentissage supervisé dit de ***rétro-propagation du gradient «backpropagationn ».***

Cette étude peut être divisée en 4 chapitres :

*D*ans le premier chapitre, nous introduisons quelques généralités sur le système OCR (reconnaissance optique des caractères), le système d'acquisition, le processus et les différentes méthodes de reconnaissance.

*L*e second chapitre sera consacré aux concepts fondamentaux sur les réseaux de neurones, ainsi que l'algorithme d'apprentissage choisi et les différentes structures.

*L*e troisième et quatrième, porteront sur l'application et les résultats obtenus.

A la fin, et à la lumière des résultats obtenus dans nos différents entraînements, nous présenterons les conclusions et perspectives.

CHAPITRE 1

LE SYSTEME OCR

« OPTICAL CHARACTER RECOGNITION »

INTRODUCTION :

La reconnaissance optique des caractères ou OCR(Optical Character recognition) est l'un des travaux de recherche menés depuis quelques décennies dont le but est de faciliter la communication homme-machine.

En particulier l'introduction de texte écrit sur un support papier dans la mémoire de l'ordinateur, sous forme d'une suite de nombres ASCII peut grandement contribuer à cet objectif. Les principales applications auxquelles l'OCR est classiquement attaché sont les suivantes :

l'environnement bancaire :

Elle permet la saisie automatique des chèques pour la vérification de la somme d'argent donnée en chiffres et en toutes lettres et la vérification des signatures. Actuellement les seules applications utilisées avec succès sont celles qui font le relevé de chèques.

la bureautique :

Les archives «papier » arrivent à saturation !
L'utilisation des lecteurs en bureautique est destinée essentiellement à la saisie et à l'archivage de documents.

la poste :

Les lecteurs sont utilisés pour faire le tri automatique des courriers. Ils lisent l'adresse, localisent et identifient le code postal.

la lecture pour les non voyants :

Les lecteurs peuvent aider les non voyants à la compréhension des textes. Ils sont équipés d'un haut-parleur avec un synthétiseur de parole, ils sont limités à l'utilisation dans des associations de non voyants.

I- Généralités :

I-1-Historique :

L'origine de l'OCR remonte aux années 1900 au cours desquelles on inventa le Scanner à balayage pour la télévision et les machines à lire[14].
Tuyrin développa alors la première application d'aide aux handicapés visuels.

C'est en 1940 qu'apparaissent les premières applications informatiques de l'OCR. Ainsi en 1975, les Japonais utilisaient couramment des lecteurs qui déchiffraient le code postal inscrit à la main ou tapé à la machine.

Pour le monde arabe, la plupart des chercheurs ont utilisé la tablette graphique comme moyen d'acquisition car elle présente l'avantage de restituer fidèlement le sens du tracé.

Parmi ces travaux, on note le projet IRAC (interactive recognition arabic character)réalisé à l'université de Nancy par le professeur Adnan Amin . Ce projet a été le premier à avoir abordé sérieusement le problème de la reconnaissance de l'écriture arabe(imprimée et manuscrite).

Ainsi en 1991, le docteur Mohammed Bassam El-kurdy a conçu un système OCR pouvant reconnaître un texte imprimé en utilisant la segmentation en caractères.

En Algérie, les travaux les plus importants au niveau de l'ENP et ceux réalisé par Melle Ait Daouad en 1997 avec son système de reconnaissance de caractères arabes multifontes, multi tailles et les travaux dirigés par Mme Hamami sur la reconnaissance des caractères par les méthodes structurelles et statistiques.

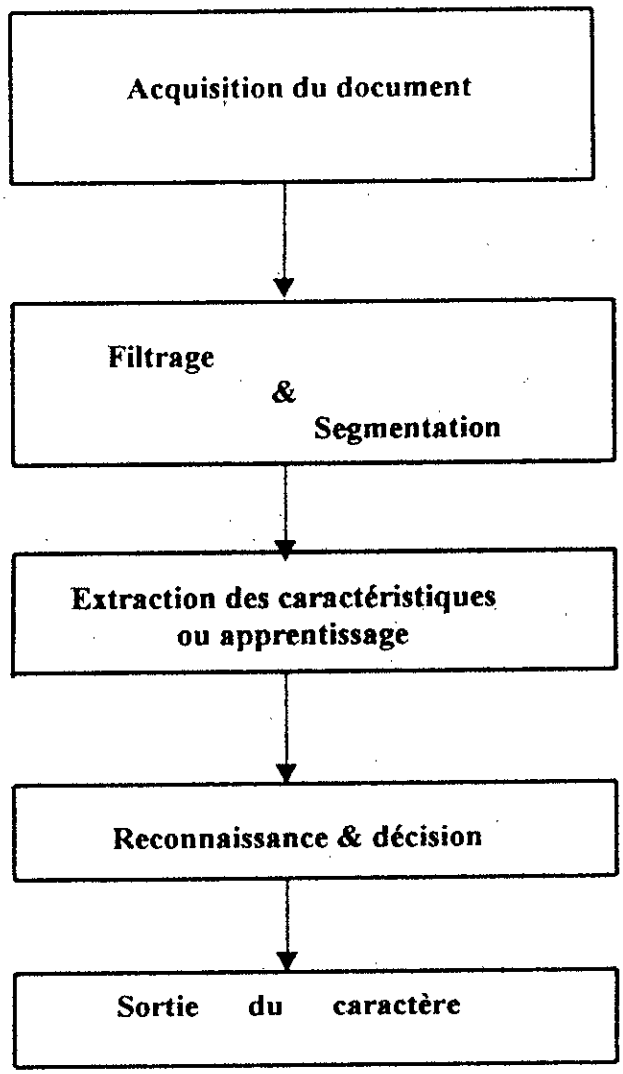
I-2. l'OCR (Reconnaissance Optique des Caractères) :

Elle consiste à récupérer automatiquement des textes imprimés ou manuscrits en évitant la phase pénible de la saisie par clavier.

La transformation d'un texte imprimé en code **ASCII** se déroule en deux phases :

La première se résume à numériser la page entière et la deuxième phase consiste à partir d'un fichier graphique (**BMP, PCX, . . .**) à distinguer des formes que l'on identifie comme des lettres de l'alphabet.

Un système OCR est décrit par le schéma suivant : (figurel -1)



Figurel-1 : ***DIAGRAMME FONCTIONNEL D'UN SYSTEME OCR***

Par la suite nous allons rappeler le rôle de chaque étape dans l'élaboration du processus complet de la reconnaissance, puis nous reviendrons dans les chapitres qui suivent plus en détail sur la fonction de certaines d'entre elles.

I-3. Définitions

I-3-1. Présentation de l'écriture arabe :

L'écriture arabe a connu plusieurs styles qui se développèrent à travers les siècles. Parmi ces styles une écriture angulaire de style Koufique(de la ville de Koufa en Irak) a servi au début de l'Islam à copier le Coran. Et vue sa structure complexe, elle est devenue par la suite une écriture décorative. Une autre structure plus simple appelée le « Naskhi »est venue ensuite pour remplacer rapidement la première. Cette écriture est la plus utilisée de nos jours par les imprimeries, les machines à écrire et les logiciels de traitement de texte[1].

Caractéristiques des caractères arabes :

L'alphabet arabe comprend 28 lettres plus el-hamza qui a un rôle très important dans la phonétique et le lemalif qui est en réalité un caractère composé du lem et du alif. Contrairement aux caractères latins, l'Arabe n'a pas de voyelles mais plutôt des signes de ponctuation qui peuvent être placés au-dessous ou bien au-dessus du caractère[16].
Chaque caractère peut s'écrire selon deux à quatre formes différentes en fonction de son contexte dans le mot :

- initiale : au début du mot.
- Médiane : au milieu du mot.
- Finale : à la fin du mot et lié à la lettre précédente.
- Isolé : non lié.

I-3-2. FONTE :

Une fonte est un ensemble de caractères d'une police particulière à une taille particulière.

I-3-3. Police :

Une police indique le style ou le graphisme des caractères et leurs attributs(italique, gras, souligné, ombré...).

I.4-Les différents aspects de l'OCR :

I.4-1-Reconnaissance monofonte, multifonte et omnifonte :

Pour un texte imprimé, un système est dit monofonte s'il ne traite qu'une seule fonte à la fois. Dans ce cas, l'apprentissage est facile et simple à réaliser et le taux de reconnaissance est très élevé.
Un système est dit multifonte s'il est capable de reconnaître un mélange de quelques fontes, parmi un ensemble, préalablement apprises. L'apprentissage doit gérer les ambiguïtés dues à la ressemblance des caractères des différentes fontes. Enfin, un système omnifonte est un système qui permet de reconnaître toute fonte sans l'avoir absolument apprise[2].

I.4- 2-Reconnaissance on-line (en ligne ou en temps réel) :

La reconnaissance on-line est une reconnaissance dynamique qui se déroule pendant l'écriture. Dans ce cas, on utilise la tablette graphique pour l'acquisition du document qui est un capteur sur lequel on écrit à l'aide d'un stylo spécial.
Les mouvements du stylo sont enregistrés magnétiquement ou électrostatiquement et transmis à l'ordinateur sous forme d'une succession de points. [2]

I. 4- 3-Reconnaissance off-line (hors ligne ou en différé) :

La reconnaissance démarre après l'acquisition du document entré. Ce mode permet l'acquisition d'un nombre important de documents et nécessite un traitement complexe pour retrouver l'ordre de l'écriture[2].

II- Acquisition de l'image et prétraitement :

La première étape d'un système OCR consiste à digitaliser l'écriture et à la présenter au système sous une forme lisible avec un minimum de bruit ou de déformations.
A ce stade de traitement, l'acquisition se déroule en deux étapes distinctes :

- La première concerne l'acquisition physique de l'image grâce au capteur.
- La seconde consiste à la récupération et la lecture du fichier image selon le format délivré, en vue de son exploitation.

II-1-Dispositifs d'acquisition :

Le Scanner est un organe périphérique de l'ordinateur permettant la saisie d'informations existantes sur un support papier. Il reçoit la lumière réfléchie par l'image fortement éclairée par des tubes lumineux, ces rayonnements sont convertis en ondes électriques puis un convertisseur transforme l'information électrique en une information numérique. Finalement, la numérisation d'une image est une opération qui dépend de plusieurs systèmes qui travaillent en cascade :

- Le capteur d'image proprement dit.
- Le convertisseur analogique numérique.

Ces systèmes constituent la chaîne de numérisation d'une image qui vont de l'image observée jusqu'à la mémoire de l'ordinateur.
Il existe différents types de scanners :

II.1-1-Scanner à main :

C'est un Scanner qu'il faut déplacer à la main sur le document à numériser. Sa forme est proche de celle d'une souris.

II.1-2-Scanner de table :

- ***Scanner à défilement :***

Le document est entraîné à travers une fonte pour passer devant la barrette CCD(charge coupled device). Ce type de Scanner n'accepte que les feuilles volantes, il oblige donc à utiliser des photocopies de documents volumineux.

- ***Scanner à plat :***

Le document repasse sur une vitre et n'est pas déplacé par l'appareil. C'est la barrette CCD qui se déplace le long de la page.

II-2-REPRESENTATION D'UNE IMAGE :

II-2-1-Définition d'un pixel(picture element):

Le pixel constitue le plus petit élément de l'image, il posséde un niveau de gris qui correspond au rayonnement réfléchi par cet élément[13].

II-2-2-Image analogique:

On associe à une scéne visuelle ou à son image dans le plan focal d'un système optique, une fonction à quatre variables L(I, J, t, λ) qui décrit l'énérgie lumineuse présentée par un point de coordonnées (I, J) à un instant t et sur la longueur d'onde λ.

Dans de nombreux cas, on considére:

- qu'il n'y a pas d'évolution temporelle des grandeurs et des caractéristiques du système.
- que la radiation est monochromatique, c'est à dire, que la longueur d'onde de la radiation est unique.

La grandeur propriétés données p L(I, J, t, λ) devient L(I, J) pour une longueur d'onde λ possédant les propriètès données par l'équation (2-1)[13].

$$L(I,J)=\begin{cases} >=0 \quad \text{dans un domaine D,} \\ \\ 0 \text{ ailleurs.} \end{cases} \qquad \ldots\ldots\ldots\ldots\ldots\ldots(2\text{-}1)$$

II-2-3-Image multi- niveau :

Appelée aussi ***image numérique***, l'image multi niveau est une matrice L(I, J) à deux dimensions dont chaque élément représente le niveau de gris du pixel de coordonnées (I, J)ce niveau de gris peut prendre des valeurs entières comprises entre 0 et 2^n,ou n est le nombre de bits du quantificateur analogique/ numérique ou numérique / analogique, la valeur minimale zéro(0) correspond au blanc et la valeur maximale 2^n correspond au noir[13].

II-2-4-Image binaire :

C'est l'image pour laquelle les éléments de la matrice L(X, Y) ont pour valeur : [7]

$$L(X,Y)=\begin{cases} 1 & \text{si } S_1 \le L(X,Y) \le S_2. \\ 0 & \text{sinon} \end{cases}$$

Avec :

0< S1<256 ; 0< S2<256 ; 256 : niveau de gris maximal.

II-3-Description du format Tiff :

Introduction :

Il existe au moins trois différents formats du bit image : Pcx, Bmp et Tiff.
Pcx est représenté par PC Paint Brush de Zsoft et Bmp est le format utilisé par Paint Brush de Windows. La spécification Tiff, utilisée par les fichiers Tiff a été développée à partir de 1986 par Aldus Corp et un groupe d'autres compagnies principalement Scanner Manifacturies telles que DCST Corp, Hewlett, Packard CO , Microsoft Corp, Microtek International, New Image Technology et Xerox Imaging Systems / Data copy [11].

II.3-1-Fichier Tiff :

La grande force du format Tiff est sa flexibilité. Les images peuvent être en couleurs, en échelle de gris, d'une taille aussi grande que voulue.
Un seul fichier peut même contenir plusieurs images(ou la même à des résolutions différentes). Cette flexibilité est le résultat de l'utilisation systématique de pointeurs au sein de fichier. Ainsi un fichier Tiff ne présente pas une structure figée mais de manière identique à celle des répertoires d'un disque, une organisation qui est elle-même décrite au sein du fichier.

II.3-2-Exploitation du fichier Tiff :

Tout type de fichier informatique à des règles qui gouvernent sa structure ; pour pouvoir exploiter ce type de fichier il faut connaître ses règles.
Le plus simple des fichiers Tiff (Tagged Image File Format) est constitué de 3 principales parties comme il est précisé dans la figureI-2.

a)- première partie : l'entête du fichier :

Composée de huit octets, ils contiennent 3 informations :

- Les deux premiers octets 0 et 1contiennent le code ASCII du caractère(I) pour Intel ou du(M) POUR Motorola.
- Les deux octets suivants 2 et 3 contiennent le numéro de la version (42 pour une compatibilité éventuelle).
- Les derniers quatre octets 4, 5, 6 et 7 contiennent un pointeur vers la 2éme partie, c'est à dire, le 1er IFD.

b) - la deuxième partie : l'IFD (Image File Directory)

C'est le répertoire du fichier, il contient des informations sur l'image comme : les dimensions, les couleurs, compression etc.....
Au début d'un IFD, on trouve le nombre de tags ou champs que contient cet IFD, écrit sur 2 octets, juste après ces deux octets commencent les tags.
Un tag est un groupe de 12 octets qui représente une information. Ces octets sont divisés en 4 parties :

• Les octets 0 et 1 désignent le type de l'information représentée par le tag(la largeur de l'image, sa hauteur, le nombre de bits par pixel, le format de compression, l'échelle de gris, etc.....).

Exemple :

- Si on trouve la valeur(256) alors ce champ contient la largeur de l'image.
- Si on trouve la valeur(258) alors ce champ représente le nombre de bits par pixel.

• Les octets 2 et 3 contiennent l'information sur le type de la donnée que représente le tag(type du point de vue informatique).

On peut trouver

1 : La donnée est de type ***byte***
2 : La donnée est en code ASCII.
3 : La donnée est de type ***short***(16 bits non signés).
4 : La donnée est de type ***long***(32 bits non signés).
5 : La donnée est de type ***rationnel.***

• Les octets 4, 5, 6 et 7 contiennent le nombre de données, s'il s'agit par exemple de la longueur de l'image, ce nombre est égal à **1** ; si la donnée représente une chaîne de caractères alors le nombre de données sera la longueur de cette chaîne.

• Les derniers octets 8, 9, 10 et 11contiennent la donnée elle-même. Si la donnée ne tient pas sur 4 octets alors ces 4 octets contiennent un pointeur vers la donnée(bien évidemment en dehors de l'IFD).

Un IFD se termine par 4 octets qui contiennent un pointeur vers un deuxième IFD d'une autre image dans le même fichier, c'est l'avantage du format ***Tiff,*** un seul fichier peut contenir plusieurs images.
Il faut noter qu'un IFD peut contenir jusqu'à 45 tags, ce nombre dépend de la richesse de l'image en informations, l'ordre des tags n'est pas important.

c) - Troisième partie : le Bit Map

C'est l'image elle-même, toutes les informations concernant l'image se trouvent dans la partie précédente : (figure I-2)
Il reste à noter que chaque octet de cette partie peut représenter M pixels avec :

$$\mathbf{M = 8/\ Bpp} \quad \text{.....(2-3)}$$

Où **Bpp** est le nombre de bits par pixel.

- Si **Bpp** = 4 alors **M=2** c'est à dire 16 couleurs.
- Si **Bpp = 8** alors **M**=1 c'est à dire 256 couleurs.
- Si **Bpp =1** alors **M=8** c'est à dire 2 couleurs(image binaire).

Les informations les plus importantes pour nous, pour exploiter un fichier Tiff sont les suivantes :

- Les dimensions(largeur, hauteur) de l'image.
- Le nombre de bits par pixel.
- Le pointeur vers **Bit Map.**

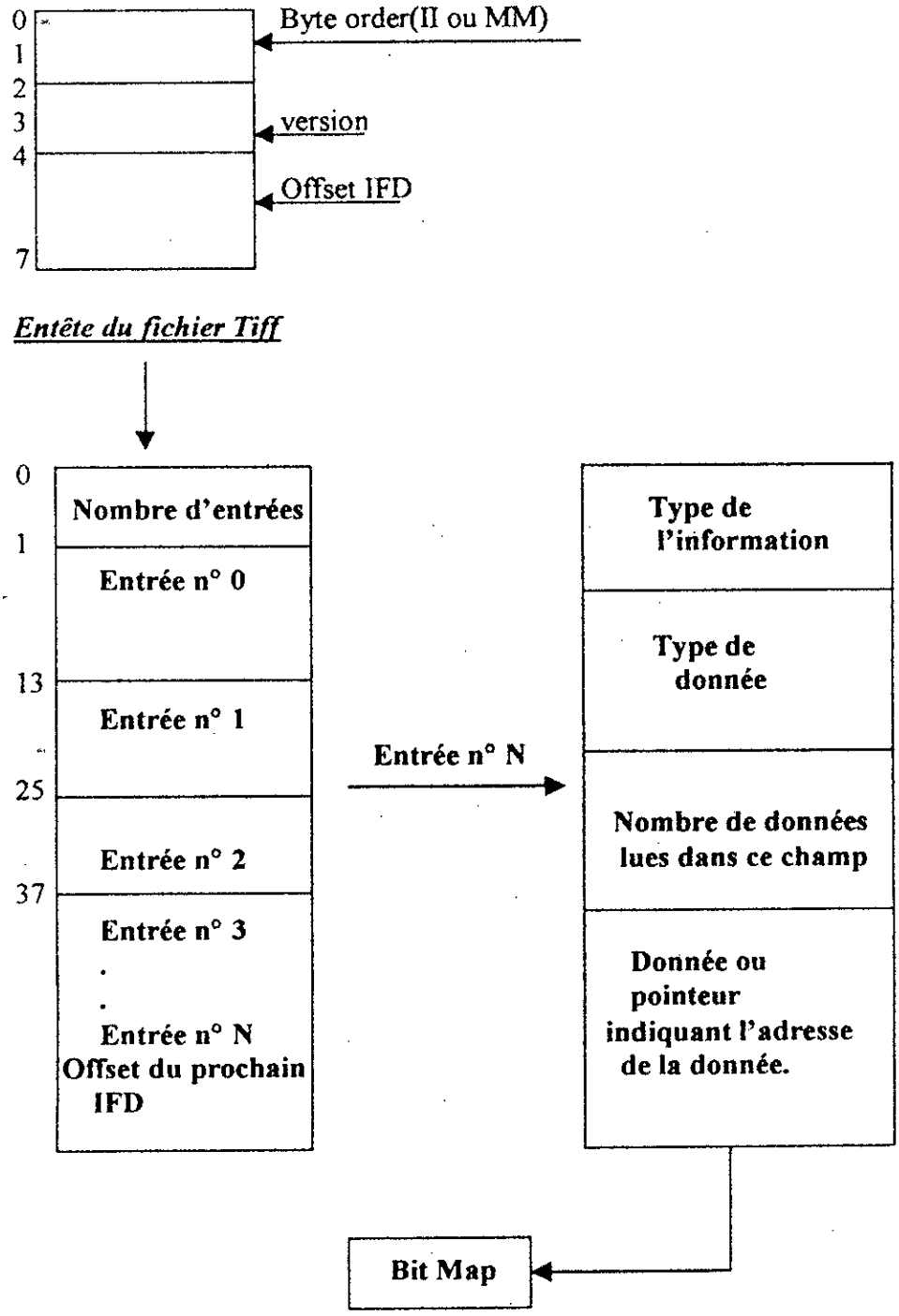


Figure I-2 : ***Structure du fichier Tiff***

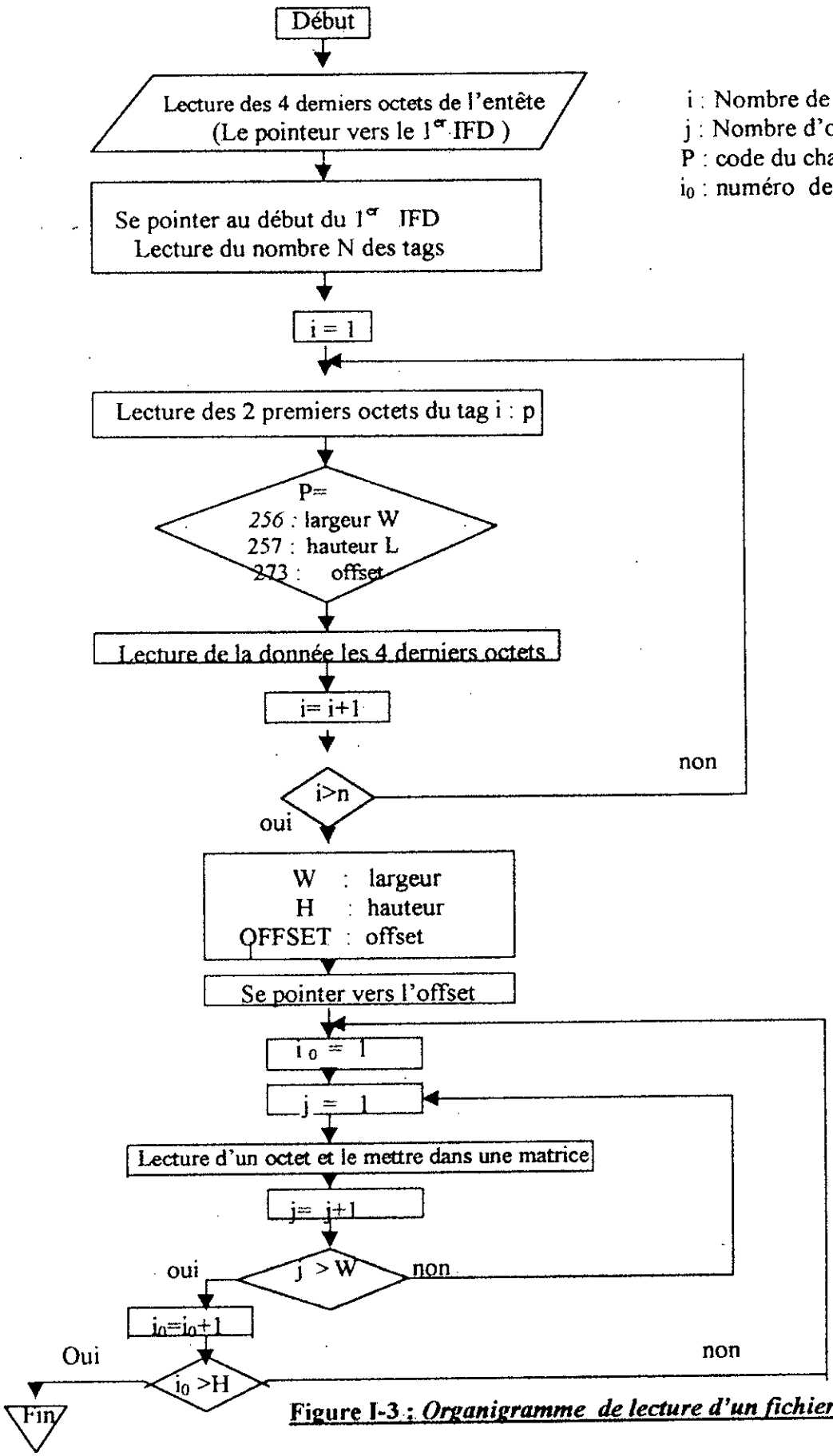


Figure I-3 : *Organigramme de lecture d'un fichier Tiff*

II-4-Prétraitement : (voir figureI-1)

La conversion numérique se fait d'un monde physique continu vers un monde discret. Celui ci est appelé l'espace de répartition. L'étape de prétraitement vise la réduction de la dimension de cet espace via la sélection des informations nécessaires à l'application.

Cette sélection doit passer par l'élimination du bruit(opération de filtrage) par la correction des erreurs(en s'aidant des opérateurs morphologiques : dilation et érosion) et éventuellement par la réduction des données.

II-4-1 Suppression du bruit :

Le but de cette étape est de débarrasser les données image du bruit de l'acquisition et de ne garder que l'information significative de la forme représentée.
Le bruit peut être du aux capteurs, mais aussi aux conditions de prise de vue(éclairage, positionnement, ...). Parmi les méthodes de suppression du bruit citons le filtrage.

Il existe plusieurs types de filtres dont le moyen et le médian.

II-4-1-1 Le filtre moyen :

Un pixel est remplacé par la valeur moyenne de ses pixels voisins. C'est un filtrage spatial passe bas qui supprime le bruit lorsque celui ci est aléatoire et cohérent avec l'image, mais à l'inconvénient de réduire la résolution spatiale de l'image.

II-4-1-2 Le filtre médian :

Ce filtre n'est pas le résultat d'une combinaison linéaire de pixels ; un pixel est remplacé par la valeur médiane de ses pixels voisins.

II-1-3 Le seuillage (binarisation) :

Cette fonction consiste à examiner les pixels et à ne retenir que ceux dont la valeur est comprise entre deux seuils de données. Ceci a pour but de sélectionner des plages de niveaux de gris correspondant à des régions d'intérêt dans la scène étudiée.

III. la segmentation :

La segmentation est une des étapes les plus importantes dans la chaîne de reconnaissance des caractères. Elle consiste à sélectionner l'information nécessaire à l'application. Toutes les opérations de reconnaissance s'appuient sur ses résultats d'où l'importance accordée à cette étape.

Les techniques de segmentation employées dans le domaine de la reconnaissance des caractères tentent de localiser les caractères dans le mot en indiquant--leur limite: gauche et droite.

Dans le cas du texte arabe, une telle opération présente quelques difficultés dues aux propriétés contextuelles propres à l'écriture arabe. Cette difficulté a souvent conduit les chercheurs à éviter la segmentation et à tenter la reconnaissance directement sur la totalité de la chaîne.

L'étape de segmentation peut être subdivisée en 3 sous étapes :

- La segmentation horizontale : on procède ici à la localisation des lignes de texte.
- La segmentation verticale : chaque ligne de texte obtenue dans la première étape est traitée pour donner les différentes parties connexes la constituant.
- La segmentation en caractères : cette étape vise l'obtention des caractères à partir des parties connexes préalablement trouvées.

IV. L'apprentissage :

Son rôle est d'éclairer la décision à l'aide de connaissance à priori sur les formes à partir de critères spécifiques aux formes. L'apprentissage tente de définir des modèles de référence ou de caractériser des «classes» de décision. Il permet ainsi de dicter au système l'algorithme de décision le plus adéquat vis à vis des règles de modélisation choisies.

V. La décision :

C'est l'étape de reconnaissance proprement dite de l'apprentissage réalisé. La méthode de décision est souvent exhibée par l'apprentissage. Ce qui veut dire que les critères utilisés pour la comparaison sont les mêmes que ceux utilisées pour l'apprentissage.

En effet, il est évident que le choix de critères différents pour la décision n'assurent pas l'uniformité de la description, et ne peut pas conduire avec certitude à des résultats cohérents.

Parmi les techniques utilisées, certaines sont fondées sur la notion de proximité et nécessitent le calcul d'une distance. La réponse de la décision peut être, selon le cas, le nom de la forme en cas de bonne connaissance, plusieurs noms en cas d'ambiguïté ou bien le rejet de la forme en cas d'incompatibilité de description avec les formes apprises de référence.

VI. Méthodes utilisées pour Les OCR :

Les méthodes les plus utilisées en reconnaissance de formes sont :

- Méthodes statistiques.
- Méthodes structurelles.

Il existe une autre méthode assez proche de la méthode statistique : la méthode connexioniste.

VI.1-Méthodes structurelles :

Les méthodes structurelles s'intéressent à la structure des formes et à leur description en terme d'assemblage de sous formes primitives. On distingue deux types de structures :

1. Les structures de graphe utilisées sont soit comme modèle descriptif soit comme outil de mise en correspondance entre un modèle et une forme.
2. Les structures syntaxiques : qui sont inspirées de la théorie des languesformelles. Une forme est analysée syntaxiquement comme une phrase d'un langage à l'aide d'une grammaire appropriée[2].

VI.1.1-Structure de graphes :

Un graphe est un outil mathématique permettant de décrire des relations dans un ensemble d'objets. Il permet de modéliser une forme.
Le problème de la décision se résoud alors par des algorithmes de comparaison de graphes. Il peut être également utilisé comme outil de mise en correspondance entre la forme à reconnaître et un modèle prédéfini[2].

VI.1.2- Structure syntaxique :

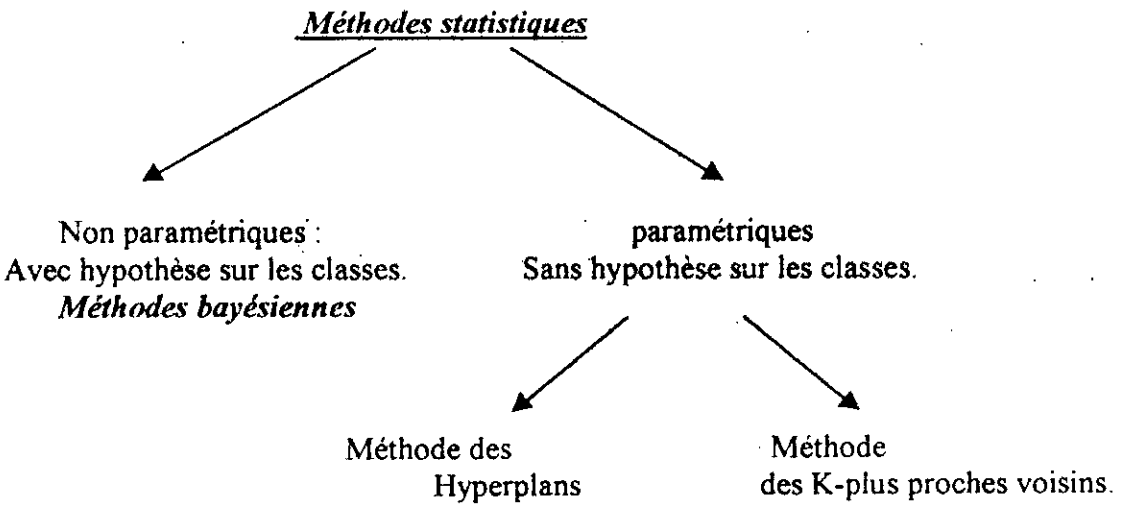
Dans les approches syntaxiques, on cherche et on exploite les règles de construction des formes à partir de leurs composantes d'où l'analogie entre l'analyse grammaticale et syntaxique.

En effet, à partir d'un ensemble d'éléments primitifs de formes et d'un ensemble de règles de combinaison de ces composants. On constitue d'autres composants intermédiaires qui seront combinés à leur tour avec d'autres éléments.

La robustesse des méthodes syntaxiques dépend beaucoup du choix des primitives et des relations qui les lient[2].

VI.3-Méthodes statistiques :

Les méthodes statistiques sont fondées sur une caractérisation statistique des paramètres des formes étudiées. Ces méthodes permettent de prendre une décision de classification d'une forme inconnue suivant un critère de «probabilité maximale d'appartenance à une classe ». Elles peuvent être résumées par l'arbre suivant :



Dans les méthodes paramétriques, on cherche à définir les frontières des classes dans l'espace de représentation, de façon à pouvoir classer les points inconnus par une série de tests simples. Par contre dans les méthodes non paramétriques on donne un modèle de la distribution de chaque classe et on cherche la classe à laquelle le point a la probabilité la plus grande d'appartenir.

Le premier cas consiste à définir des hyperplans qui séparent au mieux les classes d'apprentissage. La décision se réduit alors à une série de produits scalaires.

Une autre méthode non paramétrique très utilisée est celle de la décision par le plus proche voisin ; on attribue au point inconnu la classe de son plus proche voisin de l'ensemble d'apprentissage.

Dans le cas paramétrique, on cherche à minimiser l'erreur moyenne de mauvaise classification grâce aux hypothèses sur la nature statistique des classes. Si l'ensemble d'apprentissage se prête à ces hypothèses, on est dans une situation optimale pour la décision. La difficulté est bien souvent due à la répartition des points dans chaque classe qui n'est pas assez régulière ou bien alors les points ne sont pas assez nombreux. Pour que la précision des calculs soit proche de celle que donne la théorie. Ce type de méthodes est rapide et donne de bons résultats[9].

VI.2-Méthodes connexionnistes :

Elles relèvent du champ général des réseaux de neurones formels qui intéressent à la fois les neurobiologistes, les physiciens et les informaticiens. Elles consistent à appliquer des concepts des réseaux connexionnistes à la reconnaissance des formes.
Ces méthodes se rapprochent des méthodes statistiques. En effet, un réseau de neurones fournit une estimation de la ressemblance d'une forme donnée avec la forme apprise par le réseau[9]

Conclusion :

Les méthodes statistiques et structurelles ont fait l'objet de plusieurs travaux comme [16,1, 20].

L'apparition des réseaux de neurones a conduit les chercheurs à les appliquer à la reconnaissance des formes et en particulier aux caractères pour faire une comparaison avec les résultats obtenus par les méthodes précédentes

Pour les deux raisons citées plus haut, pour lesquels les méthodes classiques n'avaient pu répondre comme le temps d'apprentissage et le taux de reconnaissance, nous avons choisi la méthode connexioniste pour la reconnaissance des caractères imprimés.

CHAPITRE 2

LES RESEAUX DE NEURONES

<< NEURAL NETWORKS>>

INTRODUCTION :

Le cerveau est le meilleur modèle de machine, incroyablement rapide et surtout doué d'une incomparable capacité d'auto-adaptation, son comportement global est beaucoup plus mystérieux que le comportement de ses cellules de base. Il est formé d'environ 1000 milliards de neurones, et la plupart des neurones reçoivent et envoient des signaux à des milliers d'autres cellules.

En effet, les réseaux de neurones sont essentiellement utilisés pour résoudre des problèmes de classification, de reconnaissance de formes ou de traitement du signal.

Ce second chapitre a pour but d'introduire le modèle du neurone formel, ainsi que les différents types d'architectures des réseaux de neurones et leurs algorithmes d'apprentissage.

I-GENERALITES :

I-1-le neurone biologique :

Le neurone ou cellule nerveuse est l'unité élémentaire du traitement de l'information dans le cerveau.

Il est composé d'un corps cellulaire, contenant une 'machine métabolique' qui le fait vivre appelé *: **soma***, et de plusieurs prolongements très fins appelés : ***dendrites*** et d'un prolongement un peut plus épais : ***l'axone*** (figure 2.1)

Les neurones 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 ses signaux.
- Engendrer un influx nerveux.
- Le conduire.
- Le transmettre à un autre neurone capable de le recevoir.

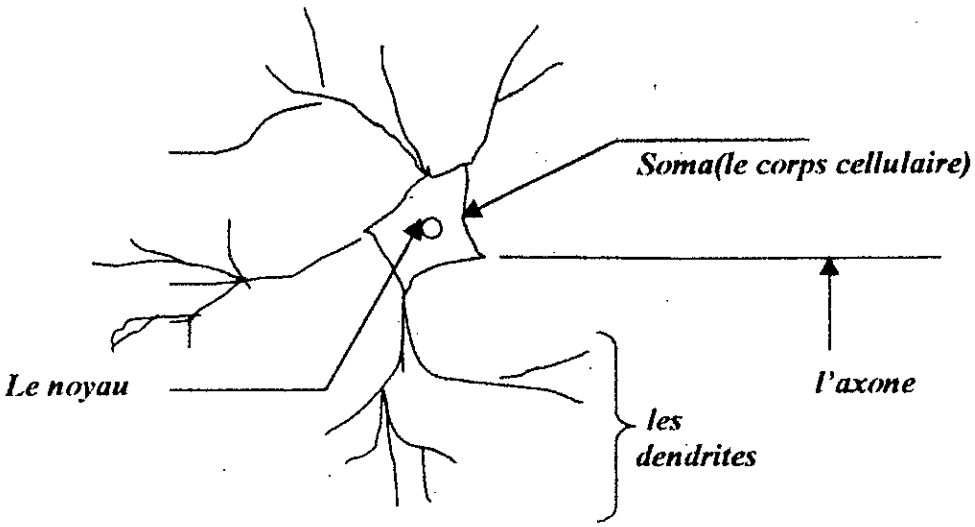


Figure 2-1 : la cellule nerveuse (le neurone biologique)

A l'intérieur des neurones, les signaux sont chimiques ou électriques, à l'extérieur, les communications sont assurées par des substances chimiques. Les connexions entre deux neurones se font en des endroits appelés synapses ou ils sont séparés par un petit espace synaptique de l'ordre d'un centième de microns. (figure 2.2)

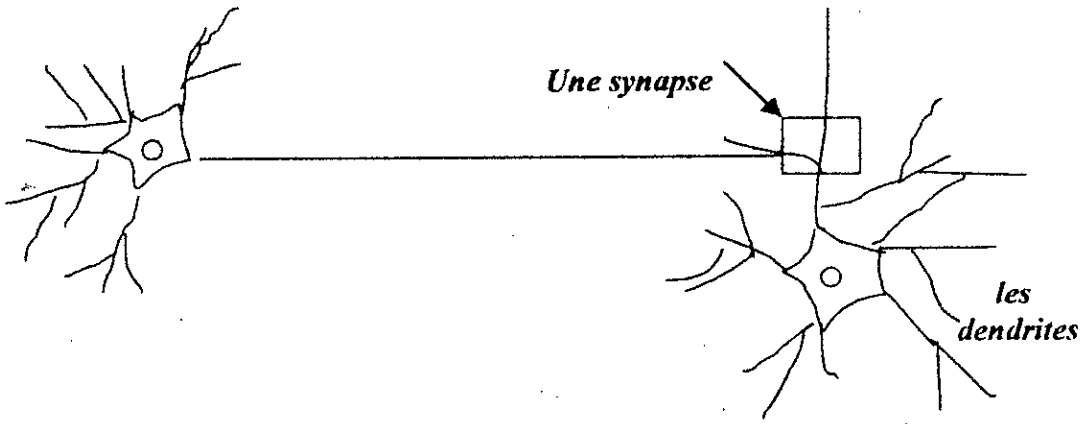


Figure 2-2 : schéma d'une synapse

En effet, un modèle de neurone, dans sa forme la plus simple, peut être considéré comme une unité de sommation et de seuillage. Elle calcule une somme algébrique des signaux excitateurs et inhibiteurs envoyés par les cellules adjacentes, et émettra un potentiel d'action.

Lorsque cette somme est supérieure à un certain seuil, il reste ainsi inactif tant que son seuil interne n'est pas dépassé.

Le regroupement de plusieurs neurones selon une architecture est en réalité trop complexe, forme ce qu'on appelle :un réseau de neurones biologiques.

I-2-Définition des réseaux de neurones :

Un réseau de neurone est crée par l'interconnexion de plusieurs neurones selon une architecture bien définie. Il possède trois caractéristiques principales :

- Il est composé d'un nombre très large d'éléments de traitement très simples.
- Chaque neurone est connecté avec plusieurs éléments voisins.
- Le fonctionnement du réseau est déterminé par les modifications des synapses pendant la phase d'apprentissage.

Hiecht Nielson a proposé la définition suivante :

« ***Un réseau de neurones est une structure de traitement parallèle et distribué d'informations comportant plusieurs éléments de traitement (neurones) , qui peuvent posséder des mémoires locales et exécuter des opérations de traitement d'informations locales, interconnectés les un aux autres avec des canaux des signaux unidirectionnels. Chaque élément de traitement à une sortie unique branchée à plusieurs connexions collatérales qui transmettent le même signal : le signal de sortie du neurone. Tout le traitement effectué dans un élément de traitement doit être complètement local, c'est à dire il ne dépend que des valeurs courantes des signaux d'entrée et des données stockées dans sa mémoire locale*** ». [5]

I-3-le neurone formel :

Le modèle que nous allons proposer fut le premier à être élaboré en 1943 par Mc.culloch et pitts.

Le neurone formel est un processeur élémentaire qui calcule la somme pondérée des potentiels d'actions qui lui parviennent(chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qui l'a émis), puis s'active suivant la valeur de cette sommation pondérée.

Si cette somme dépasse un certain seuil, le neurone est activé et transmet un potentiel d'action. Si le neurone n'est pas activé, il ne transmis rien (figure 2-3).

En effet, on peut modéliser ce neurone par deux opérateurs :

- Un opérateur de sommation : il calcule un 'potentiel' P égal à la somme de ses entrées, pondérées par des coefficients appelés poids ou coefficients synaptiques(W).
- Un opérateur calculant l'état de la sortie 'Y' du neurone en fonction de son potentiel 'P'.

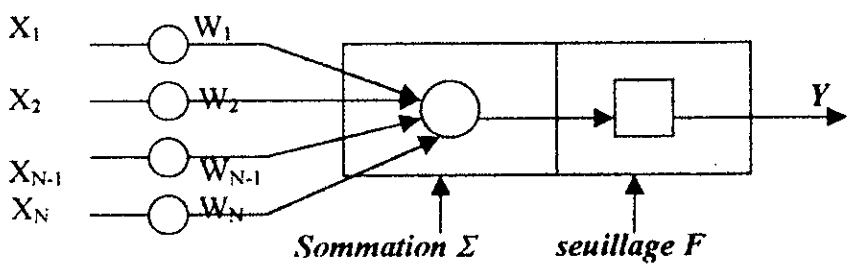


Figure 2-3 : Modèle du neurone Mc.culloch et pitts

On note : $(X_i)_{i=1,....N}$: Les entrées du neurone formel.

Y: sa sortie.

β :seuil de la fonction d'activation.

W_i: les paramètres de pondération (poids synaptiques).

F : la fonction de seuillage (voir figure 2-3).

Les entrées x_i sont des sorties d'autres neurones ou encore des entrées extérieures. Les calculs du potentiel P et de la sortie Y s'expriment par les relations suivantes :

$$P=\sum_{i=1}^{N} W_i X_i$$

Et

$$Y=F(P)$$

Avec $\begin{cases} F(P)=0, \text{ si } P<\beta \\ F(P)=1, \text{ si } P>=\beta \end{cases}$

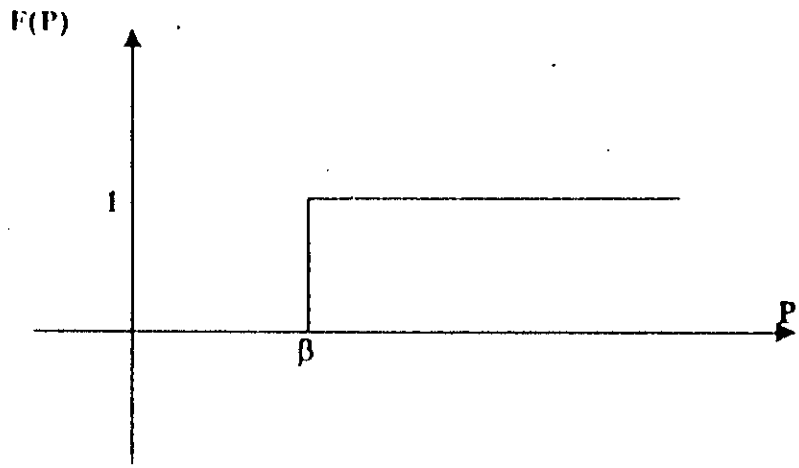


Figure 2-4 : fonction de seuillage du neurone

Modélisation générale :

D'une façon plus générale, on peut définir un neurone par les éléments suivants : (figure 2-5)

- La nature de ses entrées. Elles peuvent être binaires(0,1), bipolaire(1, -1) ou réelles.
- La fonction d'entée totale qui définit le prétraitement effectué sur les entrées.
- La fonction d'activation(ou d'état) qui détermine l'état interne du neurone en fonction de la valeur de sa fonction d'entrée.
- La fonction de sortie qui calcule la sortie du neurone en fonction de son état d'activation.

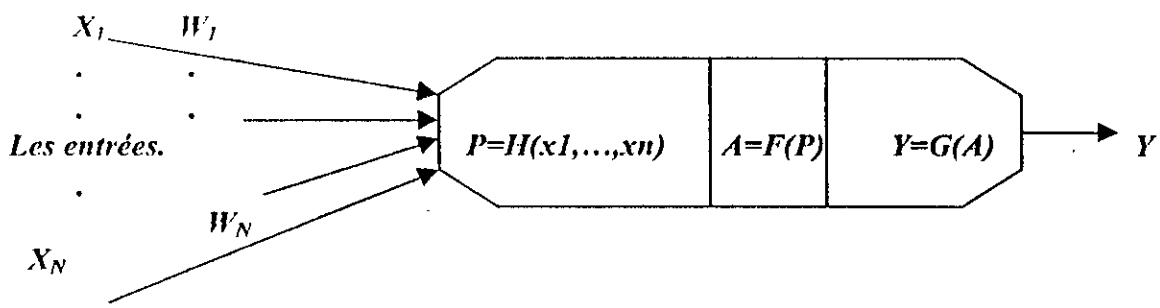


Figure 2-5 : Modélisation générale d'un neurone

On adopte les notations suivantes :

Xi (i=1,...,N) : les entrées du neurone.

Wi (i=1,...,N) : les poids synaptiques du neurone.

H: la fonction d'entrée totale ou 'potentiel' du neurone.

F : la fonction d'activation.

G : la fonction de sortie.

Et

$P=H(X_1,...X_N)$: entrée totale du neurone ou potentiel du neurone.

A=F(P) : état du neurone.

Y=G(A) : sortie du neurone.

La fonction d'entrée totale H, peut être ***linéaire*** ($H(X_1.....X_N) = \Sigma W_iX_i$)

Ou bien ***affine*** ($H(X_1,....,X_N)=\Sigma W_iX_i+b$).Le dernier cas et le plus fréquent.

Si la fonction H est affine, le terme constant b peut être entré sous le signe somme. Il correspond à la contribution d'une entrée fictive X_0 de valeur constante (+1), connectée par un poids W_0 qui est précisément ce terme constant.

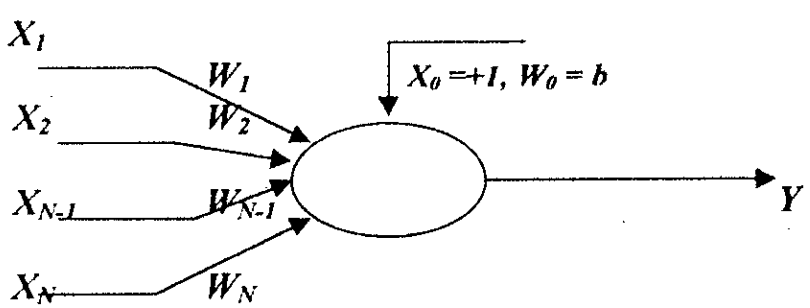


Figure 2-6 : neurone formel avec biais

Dans ce cas, $H(X_1, ...,X_N) = \Sigma W_i X_i$; ($X_0 =1, W_0 =b$), b est appelé biais du neurone. Par la suite, nous supposerons que cette entrée X_0 existe toujours.

Les non-linéarités, appelées aussi fonctions d'activation, les plus utilisées dans les réseaux de neurones sont représentés dans le tableau (2-7).

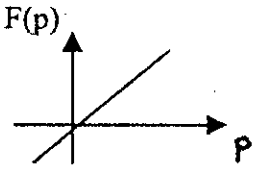
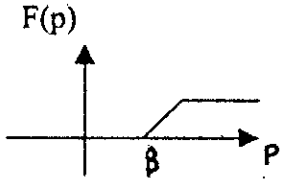
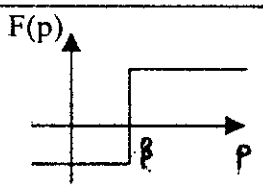
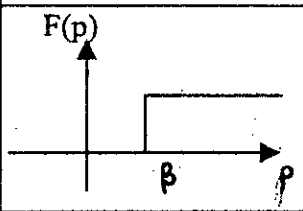
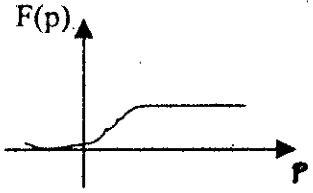
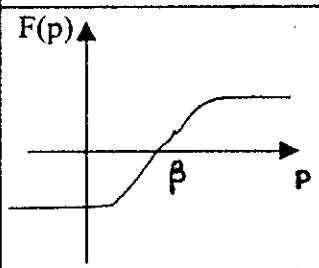
type	Représentation graphique	Représentation mathématique
Fonction *Linéaire*		$F(p) = p$
Fonction linéaire avec un seuil β		$F(P) = \begin{cases} 0 \text{ si } p \le \beta \\ ap \text{ si } \beta \le p \le \dfrac{A_{max}}{a} \\ A_{max} \text{ si } p > \dfrac{A_{max}}{a} \end{cases}$
Fonction signe avec seuil		$F(P) = Sgn(p - \beta)$
Fonction Echelon		$F(P) = U(p - \beta)$
La sigmoïde]0,1[		$F(P) = \dfrac{\exp\left(\frac{(p-\beta)}{T}\right)}{\exp\left(\frac{(p-\beta)}{T}\right)+1} \in]0\ ,\ 1[$
La sigmoïde]-1,1[		$F(P) = \dfrac{\exp\left(\frac{(p-\beta)}{T}\right)-1}{\exp\left(\frac{(p-\beta)}{T}\right)+1} \in]-1\ ,\ 1[$

Tableau 2.1: *quelques exemples de fonctions d'activation*

En général, la fonction de sortie est considérée comme la fonction identité

Alors : Y= F(P) = A

Par la suite, on confondra toujours activation et sortie du neurone.

I-4-Architectures des réseaux de neurones :

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

- Les réseaux statiques (non-récurrents).
- Les réseaux dynamiques (récurrents).

I-4-1-Les réseaux statiques :

Dans un réseau 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 du meme neurone .

Dans ce cas, la sortie du réseau est obtenue directement après l'application du signal d'entrée(autant de propagation près), l'information circule dans une indirection, de l'entrée vers la sortie.

La plupart des réseaux de neurones statiques utilisés, sont organisés en plusieurs couches de neurones appelés : ***réseau multicouches*** ou ***percéptrons multicouche***

Un réseau multicouche comporte :

- Une couche d'entrée.
- Une ou plusieurs couches cachées.
- Une couche de sortie.

Dans un tel réseau, la sortie de chaque neurone d'une couche l est connectée à l'entrée de chaque neurone de la couche suivante(l+1) (figure 2-8)

L'architecture d'un tel réseau est donnée par la figure(2-8) :

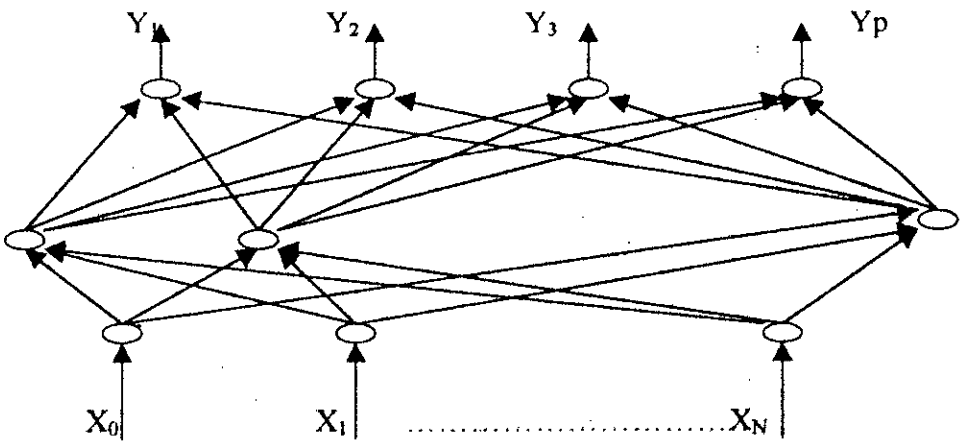


Figure 2-8 : Architecture d'un réseau de neurones statique(multi couches)

Réseaux structurés en fenêtres

Une autre structure est très utilisée dans la reconnaissance des caractères, est la structure en fenêtres.

Cette dernière ne diffère de la structure en multicouches que par ces connexions.

L'idée de base est de subdiviser l'image en plusieurs fenêtres(par exemple fenêtre 4*4) généralement identiques. Les neurones de chaque fenêtre sont connectés à un seul neurone de la couche suivante. (figure 2-8)

On peut continuer avec plusieurs autres couches successives, constitués sur le meme principe. [6]

Notons que pour ce type de structure, les calculs de sorties se font de la meme manière que ceux de la structure multicouche. Aussi les algorithmes d 'apprentissage s'appliquent de la meme façon.

Dans ce type de structure, le nombre de connexion est considérablement réduit, ce qui réduit d'autant les temps de calcul.

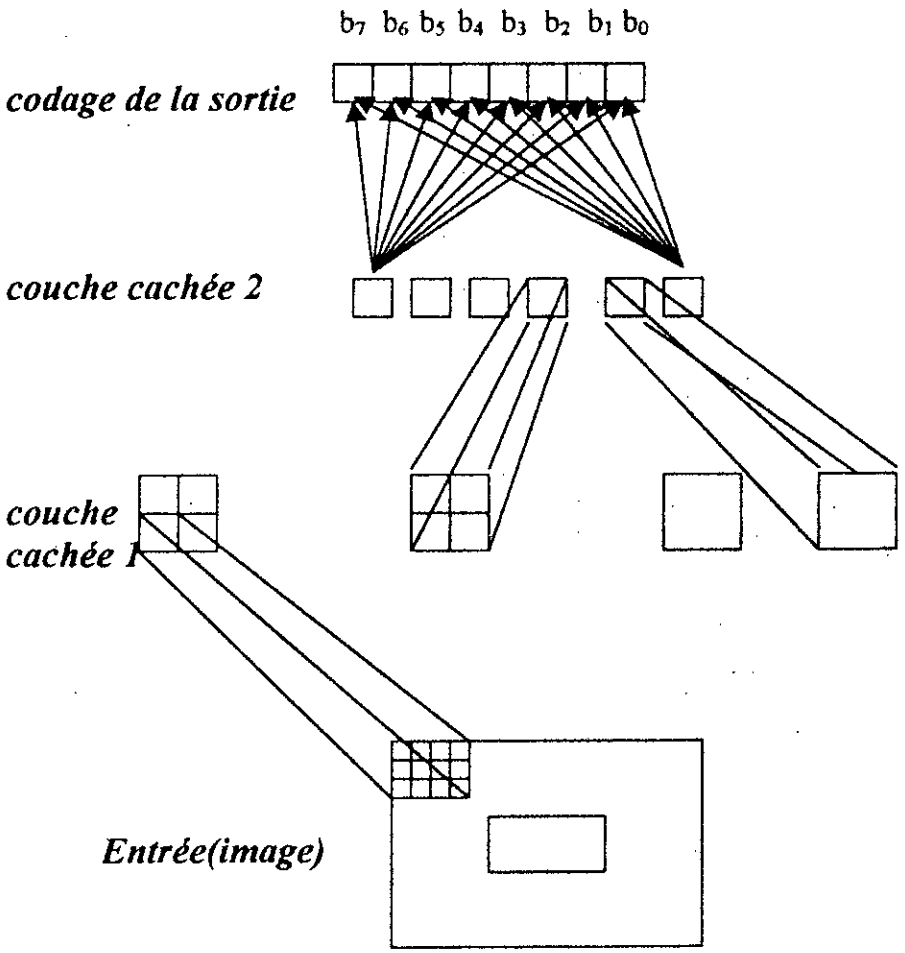


Figure 2-9 : Réseau structuré en fenêtres

I-4-2-Les réseaux dynamiques :

Contrairement aux réseaux statiques, les réseaux dynamiques contiennent dans leur architecture un bouclage(figure 2-10), ils sont organisés de telle sorte que chacun 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 état précédants.

On peut citer quelques exemples tel que : modèle de ***Hopfield, Kohonnen*** et ***Hamming.***

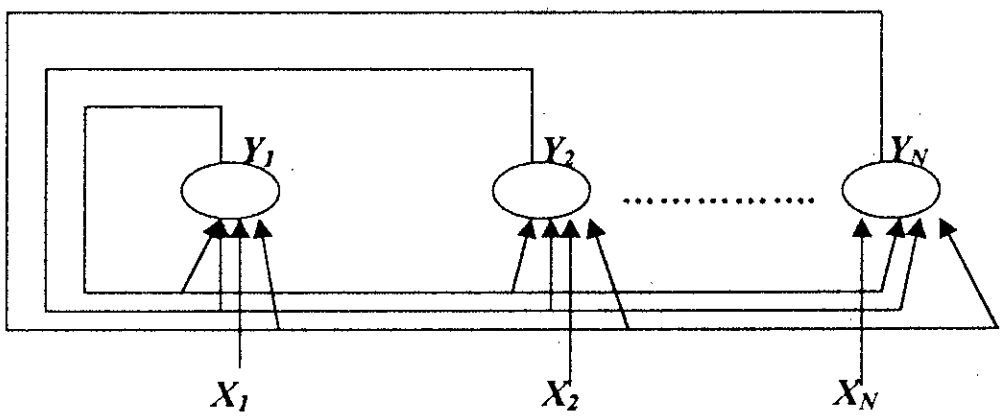


Figure 2-10 : Exemples de réseaux dynamiques (modèle de Hopfield)

I-5-Apprentissage :

Toute l'information que possède un réseau de neurones est représentée dans les poids de connexions entre les neurones. La phase d'apprentissage est l'une des propriétés les plus importantes des réseaux de neurones. Au cours de cette phase, les coefficients synaptiques ou poids sont ajustés de telle sorte que le réseau remplisse une tache définie par les exemples.

C'est en 1943 que ***Hebb*** a proposé le premier énoncé d'une règle d'apprentissage régissant sur la plasticité synaptique.

I-5-1-Règle de Hebb :

Le point fondamental de et énoncé est que le renforcement synaptique intervient lorsqu'il y 'a activité conjointe du neurone pré-synaptique(cause)et du neurone post-synaptique(effet). Chaque neurone présente deux états possibles : actif ou inactif.

L'efficacité de la synapse augmente, d'après Hebb, si les deux élément sont actifs simultanément.

Le tableau ci-dessous schématise cette règle :

Neurone pré-synaptique	*Neurone post-synaptique*	*Efficacité synaptique*
actif	actif	Renforcement
inactif	actif	Pas de modification
actif	inactif	Pas de modification
inactif	inactif	Pas de modification

Tableau 2-11 : Règle de plasticité de Hebb

On remarque que la règle de Hebb prévoit exclusivement le renforcement des efficacités synaptiques : la synapse ne peut que se renforcer. Ce qui conduirait, en l'absence de phénomènes limitants, à la saturation des réseaux.

Tout algorithme d'apprentissage se base sur la règle de ***Hebb***.

Les règles d'apprentissage peuvent êtres divisées en deux catégories/

- ***Apprentissage supervisé.***
- ***Apprentissage non supervisé.***

I-5-2-Apprentissage supervisé :

Dans un apprentissage supervisé, un «professeur » présente au réseau l'entrée, l'entrée est la sortie désirée (réponse) correspondante. Lors de l'apprentissage, les coefficients synaptiques(les poids) sont ajustés de manière à minimiser un critère qui est la mesure de la qualité de la réponse du réseau, à l'ensemble des couples choisis à l'apprentissage.

En phase d 'utilisation, les performances du réseau sont évaluées à l'aide d'un ensemble d'exemples (de même nature que l'ensemble d'apprentissage) dit ***ensemble de test.***

I-5-3-Apprentissage non-supervisé :

Pour ce type d'apprentissage, l'adaptation des poids, n'est pas basée sur la comparaison avec une certaine sortie désirée. Dans ce cas, le réseau organise lui-même les entrées qui lui sont présentées et sans qu'on lui fournisse d'autres éléments de réponse.

Cette propriété est appelée ***propriété d'auto-organisation.***

Ce type d'apprentissage possède souvent une souplesse ou une facilité dans les calculs en comparaison avec l'apprentissage supervisé.

II-Algorithmes d'apprentissage :

II-1-règle de Windrow-Hoff :

L'apprentissage du perceptron (réseau mono-couche) est un apprentissage supervisé qui se fait par correction d'erreur. On utilise pour cela la règle de Windrow-Hoff. C'est une méthode de minimisation de l'erreur par descente de gradient.

On considère un réseau constitué de N neurones recevant les valeurs à K composantes(figure 2-12)

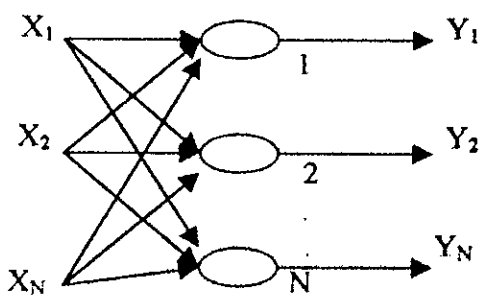


Figure 2-12 : réseau mono-couche

Les k entrées Xi du réseau sont distribuées dans tous les neurones du réseau. La sortie du neurone i vaut :

$$Y_i = F(p_i) = F(\sum_{j=1}^{K} W_{ij} X_j + b_i) \qquad \ldots\ldots \text{(II-1)}$$

Ou b_i est le biais du neurone i.

A un vecteur d'entrée X, on veut associer un vecteur de sortie désiré Yd. Si les poids W_{ij} ont des valeurs quelconques, le vecteur de sortie Y observé est différent de Yd.

On associe à cette différence l'erreur quadratique :

$$E = \frac{1}{2}\sum_{j=1}^{N}(Yd_j - Y_j)^2 \qquad \text{....(II-2)}$$

On calcule le gradient de cette erreur par rapport à Wik :

$$\frac{\partial E}{\partial W_{ik}} = \sum_{i=1}^{N}(Yd_i - Y_i)\ \frac{\partial(Yd_i - Y_i)}{\partial W_{ik}} \qquad \text{....(II-3)}$$

En remplaçant Yj par son expression, on trouve :

$$\frac{\partial E}{\partial W_{ik}} = -(Yd_i - Y_i)X_k F'(P_i) \qquad \text{....(II-4)}$$

ou F' est la dérivée de F.

En posant :

δi=(Ydi-Yi), le gradient de l'erreur s'écrit :

$$\frac{\partial E}{\partial W_{ik}} = -\delta_i X_k F'\left[\sum_{j=1}^{K} W_{ij}X_j + b_i\right] \qquad \text{.....(II.5)}$$

pour diminuer l'erreur, pour l'exemple x, il faut calculer un ΔWik dans le sens opposé à ce gradient, soit donc :

$$\Delta W_{ik} = \eta\delta_i X_k f'(p_i) \quad k=1;\ .....;K\ ,\ i=1;.....;N \qquad \text{....(II-6)}$$

et

$$W_{ik}(t+1)=W_{ik}(t)+\Delta W_{ik}$$

Notons que le coefficient d'apprentissage η 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'après un très grand nombre de pas et on risque même de tomber dans un minimum local dont il est impossible de sortie. Trop grand, il peut conduire à s'éloigner du minimum d'erreur recherché.[3]

L'apprentissage prend fin selon la décision de l'utilisateur, après un nombre fixe d'itérations ou lorsqu'on a atteint une erreur désirée sur les sorties de réseau.

Le perceptron a cependant permis de montrer qu'un réseau de neurones formels pouvait réaliser certains des fonctions du cerveau, même si cela est resté d'un niveau très limité.

II-2-La rétro-propagation du gradient :

Les limitations du perceptron ont été levées par la découverte de l'algorithme de la rétropropagation qui a permis de résoudre le problème de calcul des erreurs associées aux neurones cachés. [6]

II-2-1-Equation 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'autres part. On considère dans ce qui suit les réseaux non récurrents.

Considérons le réseau à trois couches de la figure(2-13)

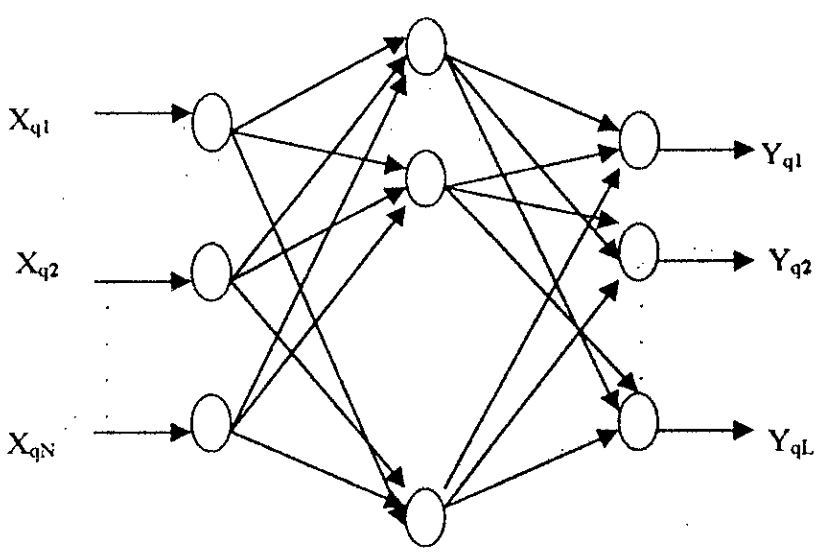


Figure 2-13 : réseau de neurones multicouches

On considère que la couche d'entrée ne fait que transmettre les entrées : le réseau ne comporte que deux couches réelles de traitement, la couche interne et la couche de sortie.

On adopte les notations suivantes :

- $X_q=(X_{q1},\ldots,X_{qN})$: vecteur d'entrée ou q est l'indice de l'exemple(q=1,....,NE) ou NE représente le nombre d'exemples de la base d'apprentissage.
- $Y_q=(Y_{q1},\ldots,Y_{qL})$ vecteur de sortie du réseau ou réponse à l'entrée Xq.
- $Yd_q=(Yd_{q1},\ldots,Yd_{qL})$: vecteur de sortie désiré pour Xq.
- $S_q=(S_{q1},\ldots\ldots,S_{qM})$: vecteur de sortie des M neurones de la couche cachée .
- W_{ji}^{h} : poids synaptiques entre le $i^{ième}$ neurone de la couche cachée et le $j^{ième}$ neurone de la couche cachée.
- W_{kj}^{0} : poids synaptiques entre le $j^{ième}$ neurone de la couche cachée et le $k^{ième}$ neurone de la couche de sortie.
- b_j^h : biais du neurone j de la couche cachée.
- b^o_k : biais du neurone k de la couche de sortie.
- P_{qj}^{h} : l'entrée totale du neurone j de la couche cachée, pour l'exemple q.
- P_{qk}^{o} l'entrée totale du neurone k de la couche de sortie, pour l'exemple q.
- F^h : fonction d'activation de la couche cachée.
- F^o : fonction d'activation de la couche de sortie.

Avec i=1,.....N ; j=1,.....,M ; k=1,..... ,L.

II-2-2-Algorithmes de rétropropagation :

L'objectif de la méthode de rétro-propagation est d'adapter les paramètres (W_{ji}^{h}, W_{kj}^{o}) de façon à minimiser l'erreur quadratique sur l'entraînement ou d'apprentissage.

L'algorithme d'apprentissage par rétropropagation du gradient est alors :

1-choix de la taille du réseau. initialisation aléatoire des poids et des neurones à des petites valeurs.

2-on applique le vecteur Xq à la couche d'entrée. Xq est choisit aléatoirement dans la base d'apprentissage.

3-on calcule l'entrée totale de chaque neurone de la couche cachée.

$$P_{qj}^{h} = \sum_{i=1}^{N} W_{ji}^{h} X_{qi} + b_j^h \qquad \ldots\ldots\ldots \text{(II-7)}$$

4-on calcule la sortie de la couche cachée :

$$S_{qj} = F^h(P^h_{qj}) \qquad \text{....(II-8)}$$

5- on calcule l'entrée totale des neurones de la couche de sortie :

$$P^o_{qk} = \sum_{j=1}^{M} W^o_{kj} S_{qj} + b^o_k \qquad \text{....(II-9)}$$

6-calcul de la sortie du réseau :

$$Y_{qk} = f^o(P^o_{qk}) \qquad \text{....(II-10)}$$

7-on calcule sur la couche de sortie le terme :

$$\delta^o_{qk} = (yd_{qk} - Y_{qk}) F^o(p^o_{qk}) \qquad \text{...(II-11)}$$

ou

$(Y^d_{qk} - Y_{qk})$ est l'erreur commise par le neurone k.

8- calcul de l'erreur commise sur la couche cachée :

$$\delta^h_{qj} = f^{h'}(P^h_{qj}) \sum_{k=1}^{L} \delta^o_{qk} W^o_{kj} \qquad \text{....(II-12)}$$

9- mise à jour des poids par :

$$\begin{cases} W^h_{ji}(t+1) = W^h_{ji}(t) + \eta \delta^h_{qj} X_{qi} & \text{....(II - 13)} \\ W^0_{kj}(t+1) = W^0_{kj}(t) + \eta \delta^0_{qk} S_{qj} & \text{...(II - 14)} \end{cases}$$

ou η est le coefficient d'apprentissage ou le pas du gradient.

Et des biais par :

$$\begin{cases} b^h_j(t+1) = b^h_j(t) + \eta \delta^h_{qj} & \text{....(II - 15)} \\ b^0_k(t+1) = b^0_k(t) + \eta \delta^0_{qk} & \text{...(II - 14)} \end{cases}$$

10-si le test n'est pas vérifié, retourner à l'étape 2.
Le test d'arrêt est l'erreur quadratique totale qu'on veut atteindre :

$$E = 1/2 \sum_{q=1}^{nbe} E_q^2 \qquad \ldots\ldots \text{(II-17)}$$

Avec $$E_q = \sum_{k=1}^{L} (Yd_{qk} - Y_{qk})^2 \qquad \ldots\ldots \text{(II-18)}$$

ou E est l'erreur entre la sortie Yq du réseau et la sortie désirée Ydq. [3]

II-2-3-Les difficultés et les limites du modèle :

L'algorithme de rétropropagation, bien qu'il ait prouvé son efficacité, il présente un certain nombre de difficultés qui sont :

*a) **Architecture du réseau :*** il n'existe pas de résultat théorique, ni même de règle empirique satisfaisante, qui permette de dimensionner correctement un réseau en fonction du problème à résoudre. [6]

*b) **Convergence de l'algorithme :***

Le problème de minimisation que la rétropropagation tente de résoudre n'est pas simple. En effet, la surface de la fonction d'erreur présente des caractéristiques peu satisfaisantes pour effectuer une descente de gradient : minimums locaux qui empêchent la convergence de l'algorithme. Dans sa version complète, l'algorithme comporte un certain nombre de paramètres continues, qu'il est difficile de régler, comme par exemple le pas du gradient.
Il est clair que ce dernier paramètre a une importance comme dans toute descente de gradient.
S'il est trop faible, la convergence du réseau risque d'être très lente, s'il est trop h

c) ***Temps de calcul :***

L'algorithme de rétropropagation est très consommateur de temps de calcul Sur les problèmes de grande taille. [6]
Ce qu'on recherche dans un algorithme d'entraînement, c'est la stabilité du Processus et la rapidité de convergence avec une bonne précision.
Depuis l'apparition de la rétropropagation quelques variantes visant à Améliorer les performances de cet algorithme ont été mise en œuvre.

Nous présentons ainsi les plus importantes.

III-Techniques d'accélération de la rétropropagation :

Plusieurs approches ont été proposées pour remédier aux problèmes cités Dans le paragraphe précédent.

- 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 à chaque fois que l'erreur se stabilise à un niveau inacceptable. Cette technique s'appelle la technique de création dynamique de neurone. (DNC : Dynamic Node Creation). [5]

- Choix d'un taux d'apprentissage décroissant avec l'évolution de l'apprentissage Le mieux est de choisir un taux adaptatif.

- L'utilisation du momentum est une méthode très importante et efficace. Actuellement, la rétropropagation est pratiquement toujours utilisée avec le momentum . Nous avons mentionné précédemment que la vitesse de convergence dépend du pas du gradient. Sa valeur est généralement choisie expérimentalement. Si η est trop petit, la convergence est lente mais la direction de descente est optimale. Si η est trop grand, la convergence est rapide mais la précision est médiocre, un phénomène d'oscillation intervient dés qu'on approche du minimum[5].

  L'utilisation du momentum permet de faire sortir l'erreur des minimums locaux, afin de chercher d'autres optimums, ce qui donne beaucoup de chance d'aboutir à un minimum global.

III-1-Aspects pratiques de l'algorithme :

L'algorithme proposé précédemment est rarement utilisé tel quel en pratique. En effet, c'est un algorithme de type gradient stochastique : on doit présenter les exemples dans un ordre aléatoire, et la mise à jour des poids a lieu à la suite de la présentation de chaque exemple.

On minimise donc une erreur instantanée, ce qui conduit à une vitesse d'apprentissage assez lente.

Pour remédier à ce problème, on accumule les erreurs de chaque exemple et on met les poids à jour après la présentation de tous les exemples de la base d'apprentissage : c'est l'algorithme du gradient global. L'algorithme modifié devient alors :

1-initialisation aléatoire des poids et des biais des neurones à des petites valeurs (entre –0.5 et 0.5).

2- q=1 ;

3- on applique le vecteur d'entrée Xq ;

4- calcul de l'entrée totale de chaque neurone de la couche cachée ;

$$P_{q_j}^h = \sum_{i=1}^{N} W_{ji}^h . X_{q_j} + b_j^h \qquad ...(II\text{-}19)$$

5- calcul de la sortie de la couche cachée :

$$S_{q_j} = F^h(P_{q_j}^h) \qquad ...(II\text{-}20)$$

6- calcul de la sortie des neurones de la couche de sortie :

$$P_{q_k}^0 = \sum_{j=1}^{M} W_{k_j}^0 . S_{q_j} + b_k^0 \qquad ...(II\text{-}21)$$

$$Y_{q_k}^0 = F^0(P_{q_k}^0) \qquad ...(II\text{-}22)$$

7- On calcule sur la couche de sortie le terme :

$$\delta_{q_k}^0 = (Yd_{q_k} - Y_{q_k})\left(F^0\right)'(P_{q_k}^0) \qquad ...(II\text{-}23)$$

8- calcul de l'erreur de la couche cachée :

$$\delta_{q_j}^h = \left(F^h\right)' \sum_{k=1}^{L} (\delta_{q_k}^0 . W_{k_j}^0) \qquad ...(II\text{-}24)$$

9- si le q<NE, alors q=q+1 et retourner à l'étape 3

10- mise à jour des poids et des biais :
pour la couche cachée.

$$\begin{cases} W_{ji}^h(t+1) = W_{ji}^h(t) + \eta . \sum_{q=1}^{N} (\delta_{q_j}^h . X_{qi}) \\ b_j^h(t+1) = b_j^h(t) + \eta . \sum_{q=1}^{N} \delta_{q_j}^h \end{cases} \qquad ....(II\text{-}25)$$

Pour la couche de sortie.

$$\begin{cases} W_{kj}^{\circ}(t+1) = W_{kj}^{\circ}(t) + \eta \sum_{q=1}^{nE} (\delta_{qk}^{\circ} S_{qj}) \\ b_k^{\circ}(t+1) = b_k^{\circ}(t) + \eta \sum_{q=1}^{nE} \delta_{qk}^{\circ} \end{cases} \quad \ldots.(II\text{-}26)$$

11- Si le test d'arrêt n'est pas vérifié, retourner à l'étape 2.

Cet algorithmes peut être appliqué à un réseau comportant un nombre quelconque de couches Cachées.[3]

III-2-Coefficient d'apprentissage :

Le coefficient d'apprentissage η ne doit pas être trop grand sinon il entraînerait Des oscillations De l'erreur autour d'un minimum qu'on ne pourra pas atteindre et si η est trop petit, le temps d'apprentissage sera très grand.

Une solution à ce problème consiste à utiliser un coefficient d'apprentissage adaptatif :

⇨Si l'erreur à l'itération (t) dépasse l'erreur à l'itération (t-1) d'un rapport c fixe (appelé rapport d'erreur), alors η est diminué par multiplication par une constante inférieure à 1.

⇨ Si l'erreur à l'itération (t) dépasse l'erreur à l'itération (t-1), alors η est augmenté par multiplication par une constante supérieure à 1 :

- Si erreur(t) > c* erreur(t-1) alors $\eta=\eta*(\eta\_déc)$ avec $\eta\_déc < 1$.
- Si erreur(t) <= c* erreur(t-1) alors $\eta=\eta*(\eta\_inc)$ avec $\eta\_inc > 1$.

Cette procédure accélère fortement le temps d'apprentissage du réseau. [10].

III-3-Momentum :

Un moyen efficace pour accélérer l'apprentissage et aussi pouvoir sortir des minimaux locaux de la surface d'erreur est d'utiliser un terme 'd'inertie' dans la correction des poids dans lequel, on tient compte de la correction des poids à l'étape précédante.

On a vu que la correction des poids se faisait par a relation :

$$W_{ij}(t+1)=W_{ij}(t) + \Delta W_{ij}(t+1) \qquad \ldots\ldots(II\text{-}27)$$

Ou la formule de ΔW_{ij} (t+1) dépend de la couche sur laquelle porte les corrections. Avec la technique du momentum, ΔW_{ij} (t+1) dépend de la couche sur laquelle porte les connexions et la loi d'adaptation devient:

$$\Delta W_{ij}(t+1)= \eta\Delta W_{ij}(t) + \alpha\Delta W_{ij}(t-1) \qquad \ldots\ldots(II\text{-}28)$$

Le paramètre α est le momentum. Il est variable suivant l'évolution de l'erreur du réseau. Sa valeur est généralement prise entre 0.8 et 0.9.[10]

Conclusion :

L'étude des modèles des réseaux de neurones nous a permis de dégager les principales caractéristiques qui les différecient des méthodes classiques utilisées dans la résolution des problèmes posés par plusieurs domaines : l'intelligence artificielle, le traitement du signal, la Reconnaissance de formes....

Ces caractéristiques sont :

- Leur parallélisme intrinsèque.
- Leur capacité d'adaptation(capacité d'apprentissage).
- La mémoire distribuée.

La mémorisation de la fonction à réaliser par le réseau est distribuée sur plusieurs neurones, ce qui introduit une résistance au bruit car la perte d'un neurone ne correspond pas à la perte de la fonction mémorisée.

- Leur capacité de généralisation.[6]

La rétropropagation constitue un algorithme général d'apprentissage dans Les réseaux multicouches.

Cet algorithme donne de bons résultats dans de nombreuses applications pratiques. Il présente en outre des caractéristiques intéressantes telles que généralisation et élaboration de représentations internes.

CHAPITRE 3 :

TRAVAIL REALISE

INTRODUCTION :

Dans notre travail nous avons simulé un réseau de neurones multicouche pour la reconnaissance des caractères et chiffres arabes imprimés isolés et de même taille acquise sur PC sous fichier Tiff. La rétro- propagation du gradient est l'algorithme d'apprentissage supervisé utilisé.

Après avoir converti les images en des matrices de taille(32x32), l'algorithme d'apprentissage consiste à adapter les poids et le seuil du réseau pour que ses sorties soient suffisamment proches des sorties souhaitées pour l'ensemble d'apprentissage dit ensemble d'entraînement. Une fois l'apprentissage est réalisé, nous enregistrons le fichier comportant les valeurs des poids de pondérations qui représentent la mémoire distribuée du réseau.

Finalement, l'étape de décision ou de la reconnaissance proprement dite, consiste à conserver les mêmes paramètres du réseau(c'est à dire, nombre de couches et le nombre de neurones de chacune d'elles), et faire passer tous les exemples de test séparément et le calcul des sorties se fera de la même manière avec évidemment l'utilisation du fichier poids, qui est le résultat de l'apprentissage.

Les applications de cette technique ont connu un succès spectaculaire et ses performances étaient quelques fois surprenantes.

Parmi les travaux qui ont été réalisés au laboratoire de recherche à l'ENP ceux de Melle C. Fiala sur «la reconnaissance des chiffres par réseaux connexionnistes « et Mr A. BENKRID sur « implémentation des réseaux de neurones sur PC ».

I- Préparation de l'ensemble d'apprentissage :

I-1-les entrées :

L'utilisation des réseaux nécessite la normalisation des images pour minimiser les variations de taille en entrée du réseau, car celui-ci comprend un nombre fixe de neurones de la couche d'entrée. En effet, les caractères qui regroupent l'ensemble de test ou d'entraînement ont été scannés ou générés par « Paint Brush » dans Windows arabe.

La procédure est la suivante :

- Sélectionner l'outil «texte » avec une police de caractères « Arabic transparant » de dimensions (20x20)pixels par caractère.

- Convertir l'image en format *Tiff* par le logiciel « ***Microsoft Photo Editor*** » livré par « ***Microsoft Office 97*** » puis les localiser dans des fenêtres(32x32)pixels[voir Annexe].

L'opération qui suit celle de la normalisation a pour but la lecture du fichier *Tiff*. Elle est primordiale pour l'extraction des informations concernant l'image, telles que la longueur (l), la largeur(h) et l'offset. A la sortie, et du point de vue informatique, l'image est représentée par une matrice image[i, j] (i indiquant le nombre de lignes de l'image. J représente le nombre de colonne de l'image.

Dans notre application, chaque élément de la matrice image[i, j] est de type binaire et il correspond à un pixel de l'image.
Et comme il s'agit d'un apprentissage supervisé, l'ensemble d'apprentissage est formé de couples (entrée, sortie désirée) avec :

- L'entrée est la matrice image[i, j].
- La sortie désirée est le code de la sortie[voir tableau III-2].

I-2-Spécification des neurones :

Les réseaux utilisés ont la structure suivante :

- L'entrée est de type binaire.
- La fonction d 'entrée est linéaire.

La fonction de seuillage qui est une sigmoïde monotone non décroissante donnée par l'équation (III-1), cette dernière que nous avons déjà présentée dans le chapitre 2, a pour rôle de limiter l'activité du neurone tout en gardant sa continuité.

$$F(x)= 1/(1+exp(- a x)) \quad ....(III-1)$$

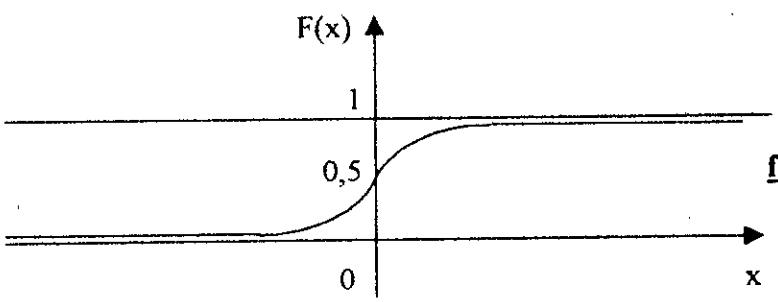


fig3-1 : *Fonction d'activation*

Par la suite, on confondra toujours activation et sortie du neurone.
Comme la fonction sigmoïde est limitée entre]0 , 1[, nous avons la possibilité de coder la sortie en binaire [voir tableau III-2].

caractére	code (38)
ا	000000
ب	000001
ت	000010
ث	000011
ج	000100
ح	000101
خ	000110
د	000111
ذ	001000
ر	001001
ز	001010
س	001011
ش	001100
ص	001101
ض	001110
ط	001111
ظ	010000
ع	010001
غ	010010
ف	010011
ق	010100
ك	010101
ل	010110
م	010111
ن	011000
هـ	011001
و	011010
ي	011011
0	011100
ا	011101
2	011110
3	011111
4	100000
5	100001
6	100010
7	100011
8	100100
9	100101

Tableau 3.2: codage de la sortie

I-3- Choix du langage de programmation (Visual C++ 4.0) :

Le développement de la micro informatique, fait apparaître le *Visual C++* pour simplifier la programmation par le C++. Il offre au programmeur les possibilités de travailler avec le «*Workspace* »et il dispose d'un compilateur rapide sophistiqué et d'un outil d'aide à la détection d'erreur «*Debbuger* ». En outre, il permet le développement des applications micro informatiques de n'importe quel domaine sur micro-ordinateur

II- Les différents types de structures étudiées :

Pour notre application, nous avons choisi au début la structure multicouche, ensuite et vue la complexité des connexions de cette architecture, nous avons simulé aussi un réseau ayant une structure en fenêtres.

II-1-Structure multicouche :

Le développement d'algorithme 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[5].
La structure multicouche choisie a les caractéristiques suivantes :

- Une couche d'entrée composée de 1024 neurones((32*32)pixels).
- Une ou deux couches cachées.
- Une couche de sortie de 6 neurones.

Les neurones de chaque couche sont connectés à tous les neurones de la couche suivante.
La structure du réseau à une couche cachée est représentée dans le tableau (3-3).

Réseau	Couche D'entrée	Couche Cachée	Couche de sortie
1	1024	200	6
2	1024	150	6
3	1024	100	6
4	1024	40	6

TABLEAU 3-3 : **Structure à une seule couche cachée.**

Et les dimensions des réseaux à deux couches cachées sont : (Tab3-4)

Réseau	Couche d'entrée	Couche cachée 1.	Couche cachée 2.	Couche de sortie
1	1024	70	40	6
2	1024	70	30	6
3	1024	40	20	6
4	1024	40	10	6

Tableau 3-4 : *RESEAU A DEUX COUCHES CACHEES.*

A cause des inconvénients de la structure multicouche qui se résument à un temps d'apprentissage énorme, ainsi que la convergence du réseau qui n'est assurée à cause du phénomène d'oscillations qui se présente pendant l'entraînement du réseau, ce ci conduit à la divergence de celui-ci.

On a recours à une autre structure dite : ***structure en fenêtres.***

II-2-Structure en fenêtres :

Cette architecture du réseau s'inspire de la multicouche. Elle réduit la taille du vecteur d'entrée, ceci entraînera une réduction considérable du temps d'apprentissage. Cette dernière se réalise, en subdivisant l'image de l'ensemble de test en plusieurs fenêtres et chacune d 'elles attaque un neurone de la couche suivante : (Fig. 3-5).

Nous avons appliqué cette structure uniquement à l'entrée du réseau, et les autres connexions se font de la même manière que celles de la multicouche.

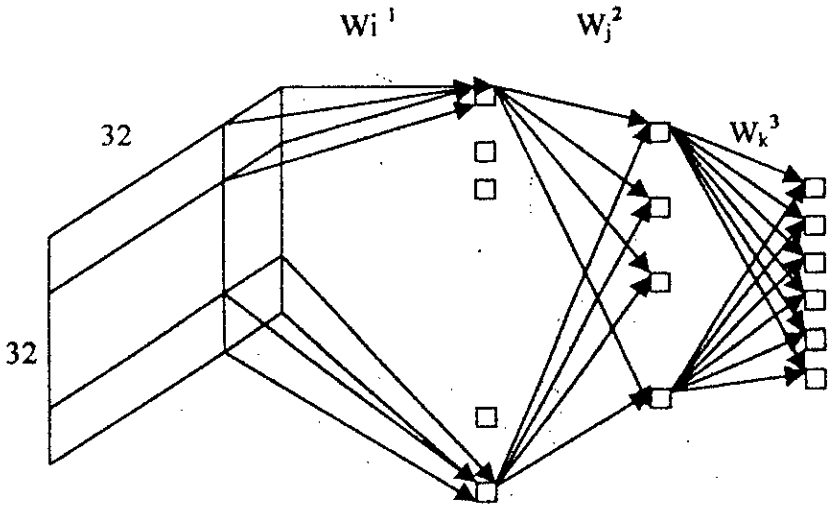


Figure 3-5 : ***Structure en fenêtres***

Les dimensions des réseaux structurés en fenêtres utilisés dans notre application sont dans le tableau (3-6) :

Réseau	Fenêtre	C-Entrée	C. cachée N°1	C. Cachée N°2	C. De sortie
1	4x4	1024	64	20	6
2	8x8	1024	16	8	6
3	8x4	1024	32	10	6

Tableau 3-6 : ***Structure en fenêtres***

III- Algorithme d'apprentissage :

Nous avons choisit la rétro-propagation comme la règle d'apprentissage supervisé dans les réseaux de notre application. Elle est généralement appliquée pour les réseaux multicouches, mais elle peut être appliquée à n'importe quelle architecture à fonctions dérivables.

III-1- Structure multicouche :

L'étape d'apprentissage nécessite un algorithme qui permet de diminuer l'erreur quadratique totale du réseau jusqu'à un seuil donné par l'utilisateur. Elle suit alors les étapes :

1. Initialiser les poids avec de petites valeurs aléatoires comprises entre] -0.5, 0.5[.
2. Présenter un exemple : l'exemple est la matrice [i, j] qui devait être converti en vecteur de dimension(l x h) et le code binaire de la sortie.

Puis calculer la sortie et la dérivée par rapport à chaque poids et ajuster les paramètres de pondérations par les équations(II-19 jusqu'à II-26).

3. Recommencer avec un autre exemple, et lorsque tous les exemples ont été présentés, tester si les sorties sont suffisamment proches des sorties désirées, sinon faire une nouvelle présentation de l'ensemble des exemples.

Sur la figure (3-7) est représentée une illustration graphique de l'algorithme de rétro-propagation utilisé pour la structure multicouche[5].

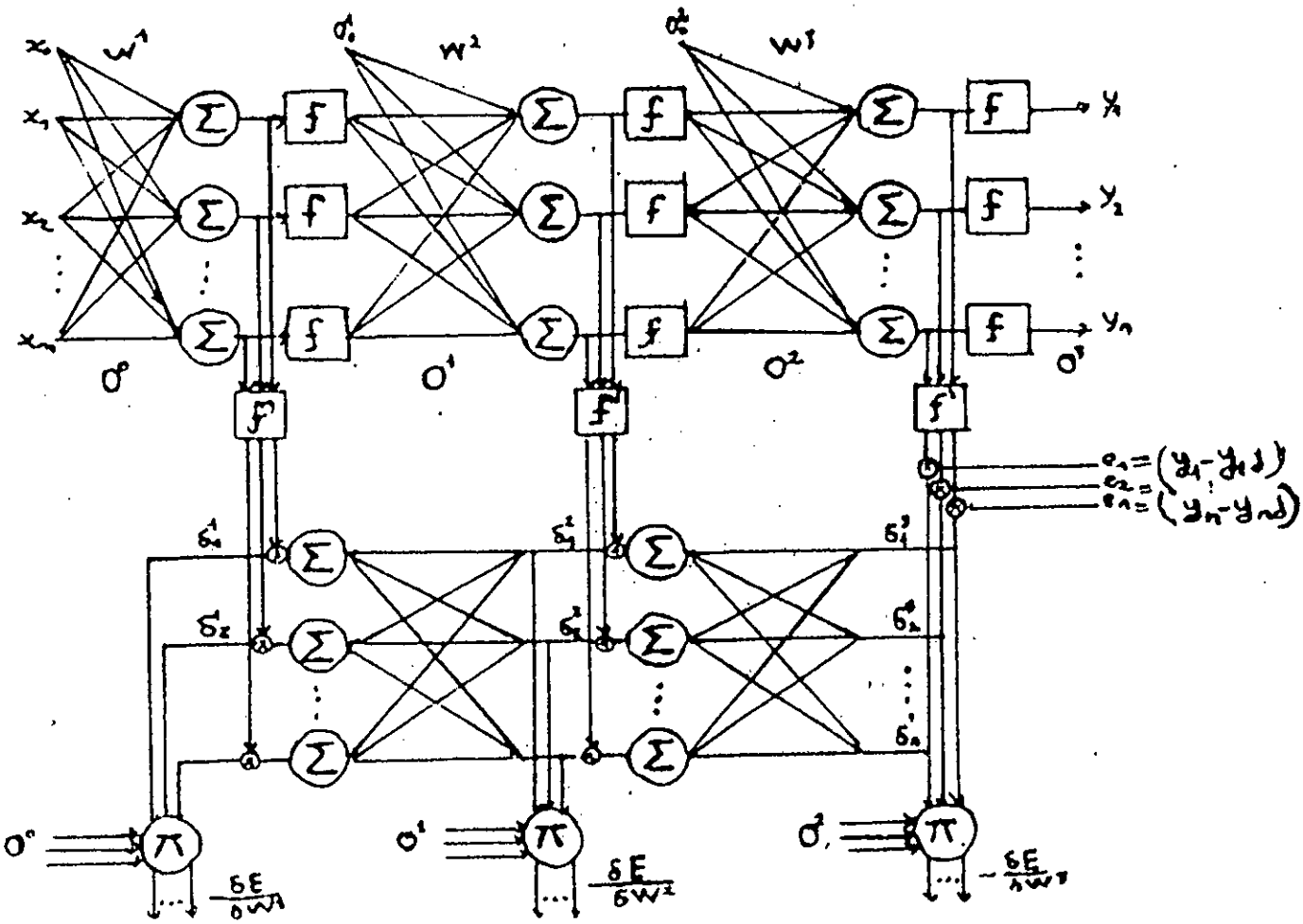


Fig. 3-7 :

Représentation graphique de la rétro-propagation.

III-2-Structure en fenêtres :

Dans cette structure, le nombre de connexions est considérablement réduit, ce qui réduit d'autant les temps de calcul.
La dimension du vecteur d'entrée n'est plus(32x32)pixels, on a la possibilité de subdiviser l'image en différentes fenêtres (8x8ou 4x8 ou 4x4). Chaque image est divisée en aires identiques(par exemple 8x8), cette dernière est directement connectée à un et un seul neurone de la couche cachée.

En se basant sur les mêmes calculs de la structure multicouche, nous sommes arrivés à élaborer un algorithme pour ce type de structure qui est le suivant :

1. Initialisation aléatoire des poids.
2. Choisir une fenêtre à l'entrée de taille(4x4, 8x8 ou 8x4).
3. q=1(exemple n°1)
4. f=1(fenêtre n°1).
5. Appliquer la vecteur d'entrée $X_q[f]$.
6. traitement de la première couche cachée(h_1) :
   a) Entrée totale :(t : la taille de la fenêtre)

$$P_{q_j}^{h_i}(f) = \sum_{i=1}^{t_1} W_{ji}^{h_1} X_{q_i}(f) \qquad \text{....(III-2)}$$

   b) Sortie :

$$S_{qj}^{h_j}(f) = F^{h_1}(P_{qj}^{h_1}) \qquad \text{.....(III-3)}$$

   c) Si f< (NF) ; alors f=f+1 et retourner à l'étape(5).
   (NF1 : nombre de fenêtres à l'entrée).

7. traitement de la deuxième couche cachée(h_2) :
   a) Entrée totale :

$$P_{qk}^{h_2} = \sum_{j=1}^{M} W_{kj}^{h_2} S_{qj}^{h_1} \qquad \text{....(III-4)}$$

   b) Sortie :

$$S_{qk}^{h_2} = F^{h_2}(P_{qk}^{h_2}) \qquad \text{....(III-5)}$$

8. calcul de la sortie des neurones de la couche de sortie :

$$P_{qk}^{o} = \sum_{k=1}^{L} W_{lk}^{o} S_{qk}^{h_2} \qquad \text{....(III-6)}$$

$$Y_{qk} = F^{o}(P_{qk}^{o}) \qquad \text{....(III-7)}$$

9. on calcul sur la couche de sortie le terme :

$$\delta^{o}_{qk} = (Yd_{qk} - Y_{qk}) F'^{o}(P^{o}_{qk}) \quad \text{....(III-8)}$$

10. calcul de l'erreur de la deuxième couche cachée :

$$\delta^{h_2}_{qp} = F'^{h_2}(P^{h_2}_{qk}) \sum_{k=1}^{L} \delta^{o}_{qk} W^{o}_{kp} \quad \text{....(III-9)}$$

11. calcul de l'erreur de la première couche cachée :

$$\delta^{h_1}_{qj} = F'^{h_1} \sum_{k=1}^{M} \delta^{h_2}_{qk} W^{h_2}_{kj} \quad \text{....(III-10)}$$

12. Si q< NE ; Alors q=q+1 et retourner à l'étape(3).

13. Mise à jour des poids

1ère couche cachée :

$$W^{h_1}_{ji}(t+1) = W^{h_1}_{ji}(t) + \eta \sum_{q=1}^{NE} \delta^{h_1}_{qj} X_{qi}(f) \quad \text{.....(III-11)}$$

2ème couche cachée :

$$W^{h_2}_{lp}(t+1) = W^{h_2}_{lp}(t) + \eta \sum_{q=1}^{NE} \delta^{h_2}_{ql} S^{h_1}_{qp} \quad \text{...(III-12)}$$

couche de sortie :

$$W^{o}_{kp}(t+1) = W^{o}_{kp}(t) + \eta \sum_{q=1}^{NE} \delta^{o}_{qk} S^{h_2}_{qp} \quad \text{...(III-13)}$$

14. Si le test d'arrêt (eq II-18) n'est pas vérifié ; Retourner à l'étape (3).

IV- Etape de reconnaissance sur des exemples d'entraînements :

Seules les images appartenant à l'ensemble d'apprentissage qui ont été testé.

La phase de reconnaissance pour notre application consiste à conserver le réseau utilisé dans la phase d'apprentissage, et faire passer un seul exemple de l'ensemble d'entraînement. Et en utilisant la mémoire du réseau, les différentes sorties des neurones se propagent d'une couche à une autre, une fois nous arriverons à la couche de sortie, nous la comparons d'une manière permanente, si elle se rapproche de la sortie imposée à l'étape d'apprentissage ou pas.

Pour cela, nous avons élaborer un algorithme simple plus le test de neurones de sorties :

1. ***Lire le caractère.***

2. ***Calcul de la sortie Yq [k]***
   Avec Yq[k] est le vecteur de sortie calculé par le réseau.

3. ***pour tout k(k=0,1,2,3,4,5)***
   Si $Y_q[k]=Yd[k]$
   Alors « le caractère est reconnu »
   Affichage du nom de caractère.

   Sinon
   « le caractère est mal reconnu »

4. ***fin.***

Conclusion :

Parmi les réseaux entraînés, seuls deux réseaux qui ont convergé. Ils ont les caractéristiques suivantes :

1. Réseau à structure multicouche avec une seule couche cachée ***1024-100-6.***
2. Réseau à structure multicouche en fenêtres ***1024-64-20-6.***

Les réseaux restants n'ont pas convergé à cause des problèmes d'oscillation qu'on a rencontré pendant les différents apprentissages des réseaux. Le phénomène d'oscillation est une parmi les limitations des algorithmes de rétro-propagation. Alors le choix approprié de la structure du réseau ainsi les dimensions, ont permis l'entraînement de deux réseaux :

- Le premier, celui de la structure multicouche, a présenté un temps d'apprentissage énorme et ceci sont du à la complexité des connexions et le temps de traitement.
- Contrairement au premier réseau, le second a appris tous les exemples avec précision et le temps d'apprentissage était très court.

Les résultats et les interprétations de cette application seront données dans le chapitre suivant.

CHAPITRE 4 :

RESULTATS & INTERPRETATIONS

INTRODUCTION :

Dans ce chapitre nous discuterons la convergence des réseaux multicouches, des réseaux en fenêtres et les contraintes qui se sont présentées.

Les deux réseaux qui ont convergé sont testés pour voir s'ils ont bien appris les exemples d'apprentissage. Ainsi nous donnerons le temps d'apprentissage de chaque réseau, l'erreur quadratique totale et le temps de reconnaissance.

1-Evolution de l'apprentissage :

L'étape d'apprentissage nécessite un algorithme qui permet de diminuer l'erreur quadratique totale du réseau jusqu'à un seuil donné .La classification se fait par le réseau lui même suivant les exemples d'apprentissage .

Ce que nous avons constaté au cours de l'apprentissage des différentes réseaux et que cet algorithme n'assure pas la convergence vers un minimum global. Le mauvais choix du pas du gradient risque de mener le réseau vers un minimum global ou vers la divergence ,une autre contrainte c'est présentée pendant l'étape d'entraînement est la saturation des neurones du au nombre élevé des connexions et les poids qui deviennent importants. Dans ce cas la fonction d'activation va prendre une valeur constante zéro ou un.

2-Réseaux convergeants :

2-1-Structure multicouche :

Durant notre travail, on a étudié la structure multicouche à une couche cachée et à deux couches cachées. Pour la structure multicouche, on a entraîné quatre réseaux à deux couches cachées dans le but d'obtenir des résultats satisfaisants .

Malgré qu'on ait modifié le nombre de neurones des couches cachées et le coefficient d'apprentissage η mais aucun des réseaux n'a convergé. Les neurones se saturent au début de l'entraînement.
Généralement pour faire sortir le réseau de l'état de saturation on diminue le seuil de la fonction d'activation ,une fois les réseaux sortes de leurs états de saturation ,un autre problème apparaît c'est le phénomène d'oscillation de l'erreur quadratique ensuite de divergence .On a adopté aux deux méthodes de modifications des poids une est celle faite à chaque présentation d'un exemple l'autre se fait après la présentation de tous les exemples . Pour remédier à cela, nous utilisons une seule couche cachée.

Pour la modification des poids , nous avons adopté deux méthodes :

- La première modifie les poids à chaque présentation d'un exemple.
- La seconde modifie les poids à la fin de la présentation de l'ensemble des exemples.

Parmi les quatre réseaux entraînés un seul réseau à une couche cachée a convergé dont la dimension est « 1024-100-6 » (ou 1024 correspond au nombre de neurones d'entrée, 100 celui des neurones de la couche cachée et 6 celui des neurones de sortie).
Ce réseau a convergé avec une l'erreur de 10^{-4} et un coefficient d'apprentissage égal à 0.4
le tableau (4-1) montre que le réseau a appris tous les exemples d'apprentissage .

Notons que la sortie du réseau codé sur 6 bits a été convertie en code décimal .Dans la structure convergente on a adopté une modification des poids après la présentation de tous les exemples au réseau .
Pour remédier aux problèmes de la saturation des neurones ,la divergence et la complexité nous avons adopté une autre structure qui est la structure en fenêtres.

2-2-structure en fenêtres :

Dans cette structure on a étudié trois réseaux à architectures différentes, on a modifié d'abord la taille de la fenêtre injectée à l'entrée(8x8, 4x8 ou 4x4), ainsi que le nombre de neurones de la deuxième couche.

Parmi les trois réseaux entraînés, un seul réseau a convergé ceci est du à la taille de la fenêtre : plus celle-ci est importante , plus le nombre de connexions augmente alors on risque de rencontrer le problème de saturation ou de divergence. Pour cela on a choisi la fenêtre (4x4) pour avoir une réduction suffisante de connexions pour les neurones de la première couche cachée et évidemment pour la deuxième couche cachée. La dimension du réseau devient :1024-64-20-6.
Chaque neurone de la première couche cachée n'est connecté qu'à 16 neurones de la couche d 'entrée. Le nombre de connexions est réduit de mille vingt quatre neurones à seize neurones. Ce réseau a convergé après sept mille quatre cent quatre vingt quatre(7424) itérations avec un coefficient d'apprentissage égal à 0.4.

Le tableau (4-2) montre que le réseau a appris tous les exemples d'apprentissage. Dans cette structure on a procédé à une modification des poids à chaque présentation d'un exemple et pour palier au phénomène d'oscillation de l'erreur, on a ajouté le terme du momentum à la modification des poids, sa valeur est égale à 0.8.

L'erreur donnée est la différence entre la sortie calculée par le réseau et celle souhaitée.
[voir les tableaux]

Tableau (IV-1) : *Résultats de l'Apprentisage du Réseau*

1024-100-6

Caractére à lire	Sortie du Réseau	Sortie désirée codé en décimal	Erreur
أ	0,287978	0	0,287978
ب	1,230025	1	0,230025
ت	2,182070	2	0,182070
ث	3,138961	3	0,138961
ج	4,178887	4	0,178887
ح	5,216202	5	0,216202
خ	6,152873	6	0,152873
د	7,240275	7	0,240275
ذ	8,161048	8	0,161648
ر	9,149929	9	0,149929
ز	10,161301	10	0,161301
س	11,080604	11	0,80604
ش	12,061436	12	0,061436
ص	13,0565656	13	0,056565
ض	14,12650	14	0,012650
ط	15,02217	15	0,022017
ظ	16,055641	16	0,05641
ع	17,873685	17	0,873685
غ	18?019308	18	0,019308
ف	18,980833	19	0,019308
ق	20,012415	20	0,012415
ك	21,014681	21	0,014681
ل	21,904432	22	0,95568
م	23,000845	23	0,000845
ن	23,935465	24	0,064535
ه	24,918097	25	0,085903
و	25,913801	26	0,086199
ي	26,915936	27	0,084064
0	27,989050	28	0,01095
1	28,881207	29	0,118793
2	29,839047	30	0,160953
3	30,929256	31	0,070744
4	31,832569	32	0,167431
5	32,929283	33	0,070717
6	33,870152	34	0,129848
7	34,881409	35	0,118591
8	35,845482	36	0,154518
9	36,818726	37	0,181274

Tableau (IV-2) : *Résultats de l'Apprentisage du Réseau*

1024-64-[illegible]-6

Caractére à lire	Sortie du Réseau	Sortie désirée codé en décimal	Erreur
ا	0,383316	0	0,383316
ب	1,050827	1	0,050827
ت	2,021560	2	0,021560
ث	3,000204	3	0,000204
ج	4,099595	4	0,099595
ح	5,227964	5	0,227964
خ	5,992884	6	0,00716
د	7,008384	7	0,008384
ذ	7,989383	8	0,010617
ر	9,022233	9	0,022233
ز	10,094985	10	0,094985
س	11,034391	11	0,034391
ش	11,987330	12	0,01267
ص	13,052964	13	0,052964
ض	13,981164	14	0,18836
ط	15,043406	15	0,043406
ظ	16,015152	16	0,015152
ع	16,966211	17	0,33789
غ	18,006739	18	0,006739
ف	18,993866	19	0,006134
ق	20,032755	20	0,032755
ك	21,006088	21	0,006088
ل	22,015402	22	0,015402
م	23,007204	23	0,007204
ن	23,994759	24	0,005241
ه	24,918097	25	0,068144
و	25,879112	26	0,120888
ي	26,995167	27	0,004833
0	27,990709	28	0,009291
1	28,928705	29	0,071295
2	29,986155	30	0,013845
3	30,976576	31	0,023424
4	31,863390	32	0,136610
5	32,813786	33	0,186214
6	33,896435	34	0,103565
7	34,950203	35	0,049797
8	35,826702	36	0,173298
9	36,907669	37	0,092331

3- *Taux de reconnaissance :*

Une fois testé pendant la phase de reconnaissance, on constate que le réseau a mémorisé tous les exemples d'apprentissage avec un taux de reconnaissance de 100% pour les deux réseaux :

1024-64-20-6 et 1024-100-6.

La base d'apprentissage utilisée contient 38 caractères (caractères arabes isolés imprimés et les chiffres de 0 à 9), la base se test est identique à celle de l'apprentissage.

4- *Temps d'apprentissage :*

Le temps d'apprentissage d'un réseau est lié étroitement au nombre d'itérations, aux dimensions des réseaux , au nombre d'exemples ainsi qu'au langage et moyens informatiques.

Dans le cas du réseau 1024-64-20-6 (structure en fenêtres), le temps nécessaire à la convergence du réseau est de 619sec(10mn 19sec).

Le temps d'apprentissage du second réseau 1024-100-6 atteint 16007sec(4h 26mn 47sec).

On remarque bien que le réseau à structure multicouche a consommé beaucoup de temps à cause de la complexité des connexions. Contrairement à la structure en fenêtres qui présente un temps d'apprentissage très faible.

5- *Temps de reconnaissance :*

Il est évident que le temps de reconnaissance des méthodes connexionnistes est plus faible que les méthodes statistiques ou structurelles. Il s'agit tout simplement dans notre cas de calculer la sortie du réseau et d'appliquer l'algorithme de décision développé dans le chapitre précédent.

Pour les deux réseaux convergeants, le temps de reconnaissance est très faible et le PC a affiché 0sec. (Rappelons que le temps d'apprentissage et de reconnaissance sont évalués sur un *pentium MMX 200*).

On peut conclure que le réseau est performant vu le faible temps de reconnaissance et l'erreur d'entrainement du réseau très faible(0.0001).

Conclusion :

Le choix approprié de la structure du réseau ainsi que ses dimensions ont permis d'atteindre l'objectif de notre travail.

On a constaté pendant l'entraînement du réseau à structure en fenêtres que le temps nécessaire pour apprendre les exemples de test était faible par rapport à celui du multicouche avec un taux de reconnaissance de 100%.

L'entrainement des réseaux multicouche choisis a connu deux contraintes :

- la saturation des neurones.
- La divergence.

Pour remédier à ces problèmes, nous avons choisi deux structures à dimensions différentes La phase de reconnaissance nous a permis de constater que tous les exemples ont été appris.

Finalement nous nous attachons au réseau le plus rapide du point du vue convergence en respectant toutes les conditions(taux de reconnaissance élevé et temps d'apprentissage très faible).

CONCLUSION GENERALE

CONCLUSION GENERALE :

Nous avons tenté dans ce travail de simuler la phase d'apprentissage d'un système OCR, des caractères et chiffres arabes isolés de même taille et de même fonte en utilisant des réseaux connexionnistes multicouche et en fenêtres, entraînés par l'algorithme de rétro-propagation avec et sans momentum.

Nous avons choisi ce moyen réseau connexioniste multicouche et en fenêtres vu le parallélisme de traitement de l'information et leur mémoire distribuée.

L'entraînement a été effectué avec un ensemble d'exemples large et bien choisi.

Les réseaux entraînés sont :

- 4 réseaux multicouche à 2 couches cachées dont les dimensions :
    - 1024-70-40-6
    - 1024-70-30-6
    - 1024-40-20-6
    - 1024-40-10-6

- 4 réseaux multicouche à une couche cachée dont les dimensions :
    - 1024-200-6
    - 1024-150-6
    - 1024-100-6
    - 1024- 40- 6

- 3 réseaux à structure en fenêtres dont les dimensions :
    - 1024-64-20-6
    - 1024-32-10-6
    - 1024-16-10-6

Parmi tous ces réseaux 2 réseaux ont convergé, le 1024-64-20-6 et 1024-100-6.

Pour les réseaux multicouches, cet algorithme bien qu'il ait prouvé son efficacité pratique dans le domaine de reconnaissance de formes et autres, présente quelques limitations qui ont empêché la convergence de quelques réseaux de notre application : Nous citons la saturation des neurones ainsi que le phénomène d'oscillations. Pour remédier à ces problèmes, nous avons utilisé la technique du momentum et la structure en fenêtres ; la première permet de faire sortir les poids des minimums locaux afin de chercher d'autres optimums, ce qui donne beaucoup de chances d'aboutir à un minimum global . La deuxième solution est de réduire considérablement le nombre de connexions donc le temps d'apprentissage et de reconnaissance .

Notre travail nous a permis d'élaborer le plan suivant pour ceux qui veulent continuer dans le domaine de reconnaissance des caractères par réseaux connexionnistes.

- Choisir la structure en fenêtres avec des fenêtres de faible taille pour réduire la complexité des connexions.
- Adapter l'algorithme de rétro-propagation avec momentum et bien contrôler le seuil pour avoir une grande probabilité de convergence.
- Faire la simulation sur un outil informatique puissant qui permet d'exploiter au maximum le parallélisme des réseaux.
- Elargir l'ensemble d'entraînement avec un bon choix d'exemples et de tests, pour pouvoir exploiter au maximum la capacité de généralisation.

Il faut ajouter que la taille de l'ensemble d'apprentissage et son utilisation comme ensemble de tests ne nous permettra de conclure que notre réseau a la capacité de généralisation.

Références bibliographiques :

[1] S. Ait- Daouad «***Réalisation d'un système de reconnaissance de caractères arabes multailles, multifontes*** », thèse de magister, ENP, 1997.

[2] A. Belaid & M. Belaid « ***Reconnaissance des formes*** » , Inter édition ,Paris, 1992.

[3] A. Benkrid, « ***Implémentation des réseaux de neurones sur un PC*** », Thèse d'ingeniorat , ENP, 1996.

[4] J. Boisgontier, « ***C et ses fichiers*** », édition P-S-I , Paris 1989.

[5] M. Boumahraz, « ***Identification et contrôle avec réseaux de neurones*** », Thèse de Magister, université de Sétif, 1995.

[6] E. Davalo & P. Naim, « ***Des réseaux de neurones*** », Edition Eyrolles 1993.

[7] C. Fiala, « ***Reconnaissance des chiffres par réseaux connexionnistes*** », Thèse d'ingéniorat, ENP, 1995.

[8] Y.A. Freeman & D.M. Skapura, « ***Neural networks algorithms, applications and programming technics*** », édition Adisou Wexley Publishing company Houston, 1991.

[9] J. P. Haton, « ***Intelligence artificielle et reconnaissance des formes*** », Techniques de l'ingénieur, traité informatique, H1900 , 15p, 1991.

[10] J. Hérault & C. Jutten, « ***Réseaux neuronaux et traitement du signal*** », édition Hermés, Paris 1994.

[11] A. Poor, « ***Looking at the Tiff specifications from the inside*** », PC magazine, décembre 1991.

[12] J.G. Postaire, « ***De l'image à la décision*** », édition Dunod informatique, Paris, 1987.

[13] L. Saadaoui, « ***Techniques de traitement numérique d'images en vue de la RF*** », Thèse de magister, ENP, 1991.

[14] M. Shunji, « ***Historical review of OCR research and development*** », IEEE, vol 80, n°7, pp1029-1057, juillet 1992.

[15] C. Souchard, « ***Les formes de fichiers images sur PC*** », PC expert pp141-144, Juillet , Août 92.

[16] B. Zekrini & A. Zerouati, « ***étape de décision et de reconnaissance des caractères arabes multifonte, multitaille, en utilisant une méthode structurelle*** », Thèse d'ingéniorat , ENP , 1995.

[17] B. Windrow, « ***30 years of adaptative neural networks, perceptron, madaline and backpropagation*** », IEEE , vol 73, n°9, pp1415-1441, Septembre1990.

[18] L. Hamami, « ***Système de reconnaissance de caractères arabes par la méthode structurelle en arbres*** »COMAE ,'96 Tlemcen.

[19] L. Hamami, « ***Recognition system of arabic characters printed*** » IASTED'97, Spain.

[20] S. Abdou et A. Merrouche, « ***Etude de la méthode des moments pour la reconnaissance des formes appliquée aux caractères arabes*** »,Thèse d'ingéniorat, ENP, 1991

ANNEXES

ا ب ت

ج ح خ

د ذ ر

ز س ش

ص ض ط

ظ ع غ

ف ق ك

ل م ن

ي	و	ه
2	1	0
5	4	3
8	7	6
		9