

وزارة التربية الوطنية
MINISTERE DE L'EDUCATION NATIONAL

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDE

SUJET

RECONNAISSANCE DES CHIFFRES

PAR

RESEAUX CONNECTIONISTES

Propose par :
Mr L.SAADAQUI

Etudie par :
C.FIALA

Dirige par :
Mr L.SAADAQUI
Mr A.NASRI

PROMOTION
1994-1995

ERRATUM

Page de garde lire "RECONNAISSANCE DES CHIFFRES PAR RESEAUX CONNEXIONISTES".

Sommaire: troisième page lire "IV RESULTATS ET INTERPRETATIONS".

Page 2: lire "Les performances obtenues par l'approche connexioniste, en reconnaissance des formes, sont particulièrement encourageantes ...".

Page 9: lire "... des règles de modélisation...".

Page 10: lire "...accompagnée d'un taux de reconnaissance ou d'un score de confiance".
 Lire "... Dans le paragraphe ...".

Page 13: lire "1.5 DIFFERENTES APPROCHES EN RECONNAISSANCE DES FORMES".

Page 21: la figure II.5 légendes des axes $f(E)$ et E .
 Lire "la fonction de sortie: prise, en général, fonction identité.".

Page 23: lire "... Les couches intermédiaires sont appelées couches cachées...".

Page 24: lire dans le tableau II.1 "prés-synaptique et post-synaptique".

L'équation II.1
$$w_{ij}(t + \partial t) = w_{ij}(t) + \mu A_i A_j$$

Page 28: lire "...des réseaux multicouches connexionistes, on propose ...".

Page 27: l'équation II.3
$$E = 0.5 \sum_{p=1}^P E_p$$

Page 37: lire "... codée sur cinq neurones...".
 Lire "... codée sur un neurone...".

Page 40: lire "... ainsi que le nombre de neurones de la couche de sortie...".

Annexe 6: les chiffres de la dernière page ne figure pas dans l'ensemble de test.

ANNEXE 2



GENERATION D'UN BRUIT BLANC GAUSSIEN DE MOYENNE b ET DE VARIANCE var [11]:

1^{er} cas : Soit $b=0$

soit le signal $\xi(t)$ suivant une loi normale $N(0, var)$:

$$\xi(t) = \sum_{n=1}^{\infty} (a_n \cos(n \omega t) + b_n \sin(n \omega t)) \quad 1$$

où a_n, b_n sont des variables aléatoires et $\omega = 2\pi/T$ (T période du signal, T tend vers l'infini).

on écrit (1) sous la forme:

$$\xi(t) = \sum_{n=1}^{\infty} [a_n \cos(n \omega - \omega_c + \omega_c)t + b_n \sin(n \omega - \omega_c + \omega_c)t] \quad 2$$

Avec ω_c quelconque on a:

$$\xi(t) = x(t) \cos(\omega_c t) - y(t) \sin(\omega_c t) \quad 3$$

$$x(t) = \sum_{n=1}^{\infty} [a_n \cos(n \omega - \omega_c)t + b_n \sin(n \omega - \omega_c)t] \quad 4$$

$$y(t) = \sum_{n=1}^{\infty} [a_n \cos(n\omega - \omega_c)t - b_n \sin(n\omega - \omega_c)t]$$

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

par application du théorème central limité, x et y suivent une loi normal $N(0, \text{var})$.

En posant:

$$x = R \cos \theta \quad \text{et} \quad y = R \sin \theta \quad 6$$

Donc, on obtient l'expression suivante:

$$\xi(t) = R(t) \cos(\omega_c t + \theta(t)) \quad 7$$

$R(t)$ et $\theta(t)$ sont des variables aléatoires qui suivent respectivement une loi de RAYEIGH et une loi uniforme [11].

$$f(r) = \left(\frac{r}{\text{var}}\right) \exp\left(-\frac{r^2}{2\text{var}}\right) \quad r \text{ positif}$$

8

$$g(\theta) = \frac{1}{2\pi} \quad 0 \leq \theta \leq 2\pi$$

La relation liant R et θ est donnée par:

$$R(t) = \sqrt{2\text{var} \ln \frac{2\pi}{\theta(t)}}$$

9

En posant $w_c=0$, on a enfin:

$$\xi(t) = R(t) \cos(2\pi U(t)) \quad \text{avec} \quad U(t) = \frac{\theta(t)}{2\pi} \quad 10$$

$U(t)$ suit bien une loi uniforme sur $[0,1]$.

On peut alors générer $x(t)$ à chaque instant connaissant $U(t)$.

Génération de $U(t)$:

La suite $U_L(t)$ définie par :

$$U_L(t) = \frac{U(t)}{2^n - 1}$$

11



Où $U(t+1) = k U(t) [2^n - 1]$, $U(0) = U_0$ impair suit une distribution uniforme sur $[0,1]$ pour $k=8m+3$ proche de $2^{(n-1)/2}$.

En prenant $n=16$ (nombre de bit du microprocesseur), on obtient $m=16$ et $k=131$, on choisit $U_0=12357$.

L'algorithme de génération récursive de $x(t)$ se présente alors comme suit :

$$t=0 \quad U_0 = 12357$$

tant que $t < t_{max}$

$$\text{faire } U(t) = (131 \times U(t)) \bmod (2^{15} - 1), \quad U_L(t) = U(t) / (2^n - 1)$$

$$R(t) = \sqrt{2 \ln\left(\frac{1}{U_L(t)}\right) \text{var}}$$

$$\xi(t) = R(t) \cos(2\pi U_L(t+1))$$

$$t = t + 1$$

stop

12

2^{ème} cas : Dans le cas où b est différent de 0, il suffit de l'ajouter à $x(t)$:

$$\xi(t) = R(t) \cos(2\pi U_L(t+1)) + b$$

13

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التربية الوطنية

MINISTERE DE L'EDUCATION NATIONAL

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDE

SUJET

RECONNAISSANCE DES CHIFFRES

PAR

RESEAUX CONNECTIONISTES

Propose par :
Mr .L.SAADAOUI

Etudie par :
C.FIALA

Dirige par :
Mr L.SAADAOUI
Mr .A.NASRI

PROMOTION
1994-1995

SOMMAIRE

INTRODUCTION GENERALE	2
I GENERALITES	
I.1 INTRODUCTION	5
I.2 DEFINITIONS	5
1.2.1 Pixel	5
1.2.2 Image analogique	5
1.2.3 Image multiniveau	6
1.2.4 Image binaire	6
1.2.5 Bruit	6
1.2.6 Histogramme	7
I.3 PROCESSUS DE RECONNAISSANCE	7
1.3.1 Le monde physique	8
1.3.2 Le codage	8
1.3.3 Le prétraitement	8
1.3.3.1 <i>La suppression de bruit</i>	8
1.3.3.2 <i>Le seuillage</i>	9
1.3.4 L'analyse	9
1.3.5 L'apprentissage	9
1.3.6 La décision	10
I.4 SYSTEME D'ACQUISITION ET DE TRAITEMENT D'IMAGE	10
1.4.1 Système d'acquisition et de numérisation d'image	10

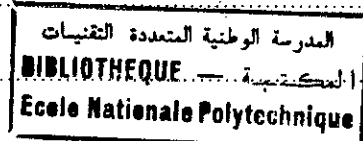
I.4.2 Système de visualisation	12
I.4.3 Calculateur	12
I.4.4 Mémoire de masse	13
I.5 DIFFERENTES APPROCHES EN RECONNAISSANCE DES FORMES.....	13
I.5.1 Approche statistique	13
I.5.2 Approche structurale	14
I.5.3 Approche connexioniste	15
I.6 CONCLUSION	15

II RESEAUX CONNEXIONISTES

II.1 INTRODUCTION	17
II.2 NEURONE BIOLOGIQUE	17
II.2.1 Les dendrites	18
II.2.2 Le corps cellulaire ou soma	18
II.2.3 L'axone	18
II.2.4 Les synapses	19
II.3 NEURONE FORMEL	19
II.4 MODELISATION GENERALE	21
II.5 RESEAUX CONNEXIONISTES	22
II.5.1 Dynamique des connexions	23
II.5.2 La règle de HEBB	24
II.6 STRUCTURE MULTICOUCHE	25
II.6.1 Apprentissage	26
II.6.2 Représentation des formes	27
II.7 STRUCTURE EN FENETRES	27

II.8 ALGORITHME D'APPRENTISSAGE: METHODE D'OPTIMISATION

ALEATOIRE MODIFIEE	28
II.9 CONCLUSION	29



III APPLICATIONS

III.1 INTRODUCTION	32
III.2 ACQUISITION DES EXEMPLES D'APPRENTISSAGE	32
III.2.1 Chiffres manuscrits	32
III.2.2 Chiffres générés	33
III.2.3 Format TIFF	33
III.2.4 Algorithme de lecture	36
III.3 CODAGE DE LA SORTIE	37
III.4 SPECIFICATIONS DES RESEAUX UTILISES	38
III.5 STRUCTURES ET DIMENSIONS DES RESEAUX UTILISES	38
III.5.1 Structure multicouche	38
III.5.2 Structure en fenêtres	39
III.6 ETAPE D'APPRENTISSAGE	42
III.7 ETAPE DE DECISION	42
III.8 CONCLUSION	43

IV RESULTATS ET INTERPRETATION

IV.1 INTRODUCTION	45
IV.2 EVOLUTION DE L'APPRENTISSAGE	45
IV.3 Réseaux convergents	46
IV.3.1 Réseau 150-10-5-1	47

IV.3.2 Réseau 256-8-4-1	47
IV.4 CAPACITE DE GENERALISATION	47
IV.5 TEMPS D'APPRENTISSAGE	50
IV.6 TEMPS DE RECONNAISSANCE	50
IV.7 LANGAGE ET MOYENS INFORMATIQUES	51
IV.8 CONCLUSION	51
CONCLUSION GENERALE	53

Bibliographie

Annexe 1

Annexe 2

Annexe 3

Annexe 4

Annexe 5

Annexe 6

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Le concept des réseaux connexionistes (qui est inspiré du fonctionnement de cerveau humain) pour le traitement de l'information est l'objet d'un intérêt croissant dans plusieurs domaines, comme celui de la parole et du traitement d'image. Ces réseaux sont constitués de processeurs élémentaires interconnectés appelés *neurones formels*. Les réseaux connexionistes sont caractérisés par les nombreux connexions qui interconnectent les neurones formels.

Le but principal, tenter d'imiter le fonctionnement du cerveau humain pour le traitement du signal et de l'information, est d'explorer une alternative aux architectures classiques d'ordinateurs au vu de la supériorité du cerveau pour la tâche de reconnaissance de formes.

Cette discipline, le connexionisme, a vu le jour dans les années quarantes, puis est tombée dans l'oubli dans les années soixantes pour reprendre vie au début des années quatre vints.

Durant ces dernières années, des développements théoriques significatifs ont posé les nouvelles bases fondamentales des réseaux connexionistes. Grâce aux récents progrès des systèmes informatiques, la puissance de calcul est devenue suffisante pour confirmer par simulation, les potentialités de nouveaux modèles connexionistes tel que le perceptron multicouche ou le modèle de KOHONEN [8].

Les performances sont particulièrement encourageantes en comparaison avec celles obtenues par des méthodes classiques [9]. Cela est essentiellement dû à la faculté d'apprentissage inhérente aux techniques neuronales, une faculté rarement partagée par des algorithmes classiques de traitement de données [9]. Cette propriété est fondamentale quand on connaît, par exemple, l'importance des efforts et le temps consacré au développement de la base de données d'un système expert à partir de l'expérience acquise par des experts humains .

L'objectif de notre travail est de présenter une application directe des réseaux connexionistes appliqués à la reconnaissance des formes: c'est la reconnaissance des chiffres. Cette application a été choisie pour voir de près les problèmes qui sont liés à la simulation d'une telle application et ce qu'on peut obtenir d'un réseau (entraîné pour la reconnaissance des chiffres) comme résultat concernant la capacité de généralisation et le traitement en temps réel de l'image.

Le premier chapitre portera sur quelques généralités sur l'image, le système d'acquisition, le processus de reconnaissance des formes, et les différentes approches en reconnaissance des formes.

Dans le deuxième chapitre, seront présentés les réseaux connexionistes et l'algorithme choisi pour l'apprentissage. Le troisième et le quatrième chapitre, eux porteront sur l'application et les résultats obtenus.

En annexe, nous présenterons la description de quelques algorithmes utilisés ainsi que l'ensemble de chiffres **générés et manuscrits**.

CHAPITRE 1

GENERALITES

1.1 INTRODUCTION

La reconnaissance de l'écriture est attachée à la reconnaissance des caractères manuscrits et imprimés. Parmi les applications de la reconnaissance d'écriture et plus spécialement les chiffres, nous signalons :

- le tri du courrier,
- la lecture des montants de chèques...etc.

Jusqu'à présent, un très grand nombre d'algorithmes a été établi dans le but de faire en temps réel une bonne reconnaissance des caractères.

Il est à noter que les chiffres à reconnaître sont des images binaires. Pour cela, dans ce chapitre, seront présentés, brièvement, quelques définitions sur l'image et son prétraitement, le processus de reconnaissance, le système d'acquisition et de traitement d'image, ainsi que les différentes méthodes sur lesquelles sont basés les différents algorithmes de reconnaissance existants.

1.2 DEFINITIONS SUR L'IMAGE

Par la suite seront données quelques définitions sur l'image.

1.2.1 Pixel

Le pixel constitue le plus petit élément de l'image. Il possède un niveau de gris qui correspond au rayonnement réfléchi par cet élément [7].

1.2.2 Image analogique [17]

On associe, à une scène visuelle, ou à son image dans le plan focal d'un système optique, une fonction à quatre variables $L(I, J, t, \lambda)$ qui décrit l'énergie 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 considère:

- 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 $L(I, J, t, \lambda)$ devient $L(I, J)$ pour une longueur d'onde λ et possédant les propriétés données par l'équation (1.1) [17]:

$$L(I, J) = \begin{cases} \geq 0 & \text{dans un domaine } D, \\ 0 & \text{ailleurs.} \end{cases} \quad (1.1)$$

où le domaine D est généralement une fenêtre rectangulaire.

1.2.3 Image multiniveau

Appelé aussi image numérique, l'image multiniveau 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 zéro (0) et 2^n , où 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 [17].

1.2.4 Image binaire

On appelle image binaire, l'image pour laquelle les éléments de la matrice $L(I, J)$ ont pour valeur l_1 ou l_2 ($0 < l_1 < N$ et $0 < l_2 < N$, où N est le niveau de gris maximal) [17].

1.2.5 Bruit

Le bruit est défini comme étant des pixels isolés dans des régions homogènes, donc ce bruit produit de brusques variations de niveau de gris [17].

I.2.6 Histogramme

L'histogramme d'une image donne le nombre de pixels affectés par chaque niveau de gris (0 à N). On peut exprimer ce nombre de pixels en pourcentage (figure I.1).

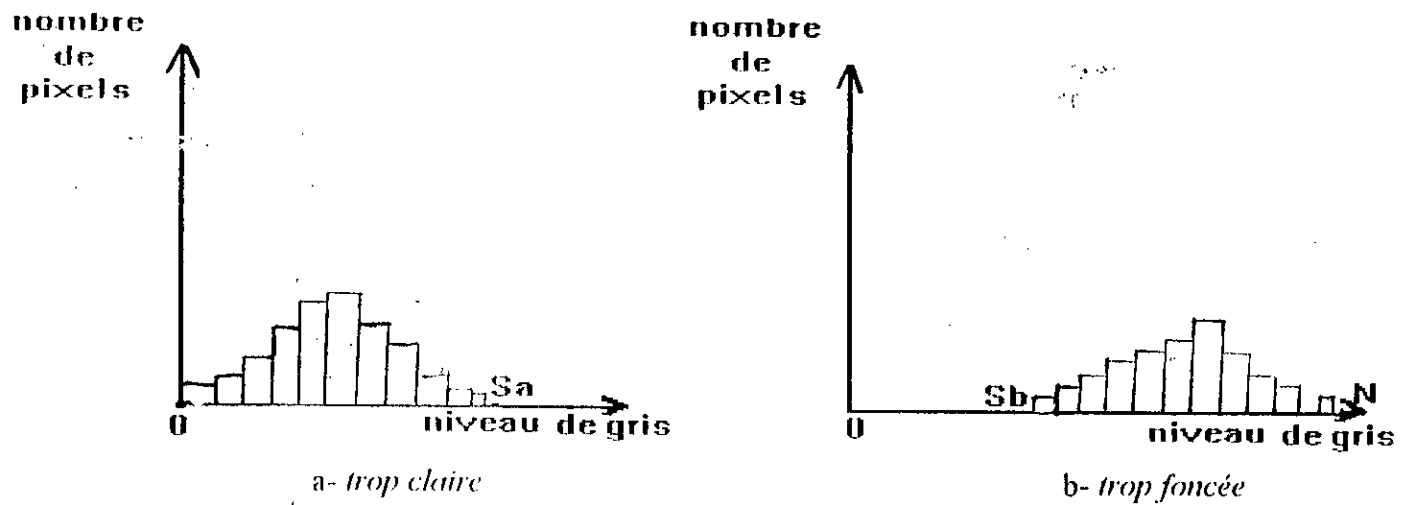


FIGURE I.1: histogrammes

I.3 PROCESSUS DE RECONNAISSANCE

La démarche classique suivie en reconnaissance des formes, consiste à opérer selon le schéma général de la figure I.2.

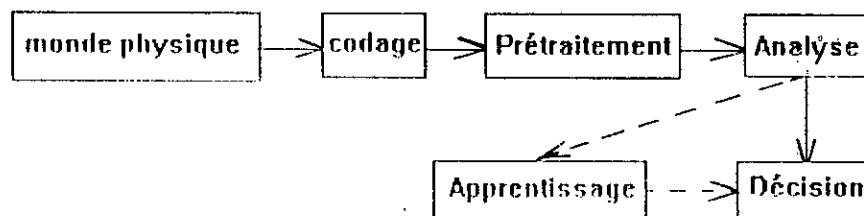


FIGURE I.2 : schéma général d'un système de reconnaissance des formes.

I.3.1 Le monde physique

Le monde physique est un espace analogique de dimensions infinies appelé espace de formes. Dans le cas de la reconnaissance des caractères, le passage de ce monde au monde discret (représentation numérique des caractères) se fait à l'aide d'un capteur doté d'un digitaliseur [4].

I.3.2 Le codage

C'est une opération de conversion du monde analogique vers le monde numérique. Ce dernier, appelé aussi espace de représentation, a une dimension trop importante même si elle est finie. La dimension de cet espace est choisie volontairement grande, de manière à pouvoir disposer d'un maximum d'information sur la forme à pouvoir y sélectionner des sous-ensembles pour de multiples usages [4].

I.3.3 Le prétraitement

L'étape de prétraitement consiste à sélectionner dans l'espace de représentation l'information nécessaire à l'application. Cette sélection passe souvent par l'élimination du bruit dû aux conditions d'acquisition, par la suppression de la redondance [4].

Le rôle du prétraitement est de « préparer » les données reçues du capteur à la phase suivante: l'analyse consacrée à l'extraction des paramètres. Cette phase n'est fiable que si les données du capteur sont dénuées du bruit et réduites à l'essentiel [4].

I.3.3.1 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 dû, bien sûr, aux capteurs, mais aussi aux conditions de prise de vue (éclairage, positionnement, ...etc), ainsi qu'à l'environnement de la scène (fond, composition de la forme, nature de la matière, ...etc) [4].

Le bruit s'insère dans le signal des images de différentes manières. Il peut être aléatoire, dans ce cas, il est inclus dans le domaine des fréquences spatiales de l'image et ne peut être supprimé qu'en faisant appel à des connaissances a priori sur l'image, ce qui cause souvent une perte sensible de la résolution spatiale. Si par contre, le bruit est périodique, il se trouve en dehors des informations utiles, sa suppression est souvent aisée par des techniques classiques de filtrage et ne cause aucune destruction. Parmi les méthodes de suppression on peut citer le lissage.

Le lissage

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 avec l'image. cependant, le lissage a l'inconvénient de réduire la résolution spatiale de l'image [4].

1.3.3.2 Le seuillage

Cette fonction consiste à examiner les pixels et à ne retenir que ceux dont la valeur est comprise entre deux seuils donnés. Ceci a pour but :

- de réduire l'espace mémoire utilisé pour stocker l'image,
- d'isoler l'objet du fond,

La procédure de seuillage, appelée binarisation, sera donnée dans l'annexe 1.

1.3.4 L'analyse

Lors de l'étape d'analyse, les techniques de traitement déterminent un certain nombre de caractéristiques ou de paramètres. Ces paramètres correspondent à des mesures de nature géométrique, topologique, ou statistique et servent comme seules données représentants la forme. Ainsi l'espace obtenu des paramètres a une dimension plus petite [4].

1.3.5 L'apprentissage

C'est une étape clé dans la chaîne de reconnaissance. Son rôle est d'éclairer la décision à l'aide de reconnaissance à priori sur la forme. A partir de critères spécifiques aux formes, l'apprentissage tente de définir des modèles de références ou de caractériser des « classes » de décision. Il permet ainsi de citer au système l'algorithme de décision le plus adéquat vis-à-vis des règles de modélisation choisie [4].

I-3-6 La décision

La décision ou classement est l'étape de reconnaissance proprement dite. Son rôle est d'identifier la forme test à partir de l'apprentissage réalisé. La méthode de décision est souvent « exhibée » par l'apprentissage, ce qui veut dire que le critère utilisé pour la comparaison est le même que celui utilisé pour l'apprentissage [4].

Parmi les techniques utilisées, certaines sont fondées sur la notion de proximité et nécessitent de calculer une distance ou une probabilité de ressemblance avec des modèles prédéfinis [4]. D'autres sont fondées sur l'analyse de la structure de la forme et essaient plutôt de vérifier une certaine cohérence dans la relation entre les différentes sous-structures [4]. La réponse de la décision peut être, selon le cas, le nom de la forme en cas de bonne reconnaissance, plusieurs noms en cas d'ambiguïté ou le rejet de la forme en cas d'incompatibilité de description avec les formes de référence. Dans les deux premiers cas, la réponse peut être accompagnée d'un taux ou d'un score de confiance.

Les applications aux quelles cette discipline est classiquement attachée sont :

- la reconnaissance de la parole,
- la reconnaissance d'écriture, ...etc.

I.4 SYSTEME D'ACQUISITION ET DE TRAITEMENT D'IMAGE

Le système d'acquisition et de traitement d'image est composé d'éléments suivants (figure I.3):

I.4.1 le système d'acquisition d'image

Son rôle est de transformer l'image analogique, obtenue par un capteur, en une image numérique. Parmi ces capteurs on cite: la tablette graphique, la camera, le scanner. Le paragraphe suivant sera expliquer le fonctionnement du scanner, car c'est le capteur utilisé dans notre travail [6].

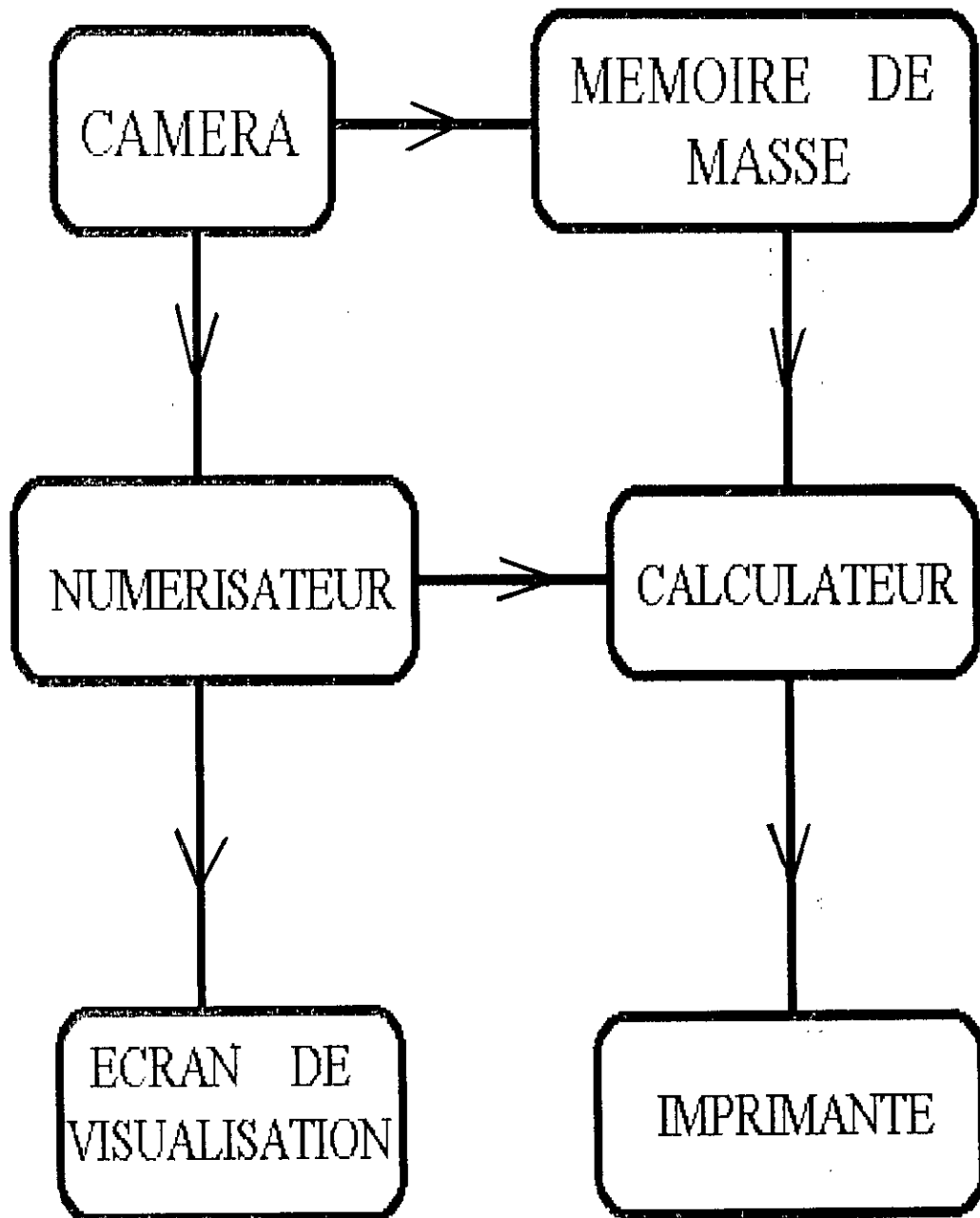


FIGURE II.3: configuration d'un système de traitement d'image

scanner

Il existe deux types de scanner: le scanner à main et scanner à table.

Le scanner comprend une source lumineuse constante. Les zones sombres du document à scanner, une fois éclairées, réfléchissent la lumière. Par l'intermédiaire d'un miroir et d'une lentille focalisatrice, des capteurs photosensibles (phototransistors à diamètre très petit ou cellules C.C.D: Charged Coupled Device) recueillent la luminosité réfléchie, et émettent une tension proportionnelle à celle-ci. L'image obtenue en sortie de ces différents capteurs est numérisée à l'aide d'un digitaliseur, et devient une image formée de pixels à plusieurs niveaux de gris [6].

L'opération de digitalisation nécessite une prise de décision sur la résolution de l'image, qui est liée à la taille de la mémoire disponible.

Soit une image $N \times N$ de pixels, où N est une puissance entière de 2. Cette image possède G niveaux de gris.

Avec:

$$N=2^n \quad \text{et} \quad G=2^m.$$

où m et n sont des entiers.

Pour stocker cette image on a besoin de $N \times N \times m$ bits. Donc, il faut bien choisir la résolution suivant la taille de la mémoire disponible.

I.4.2 le système de visualisation

Il s'agit de l'écran où pourra être reconstituée, ou visualisée l'image obtenue à partir des capteurs, ou de la mémoire du digitaliseur [6,7].

I.4.3 calculateur

C'est l'unité de traitement de l'information obtenue de l'image, et de la prise de décision appropriée vis à vis des résultats obtenus [6].

I.4.4 mémoire de masse

C'est la mémoire supplémentaire pour stocker les résultats d'un traitement. Il est nécessaire de disposer d'une mémoire de masse de grande taille. Par exemple une image de 256×256 pixels et de 64 niveaux de gris nécessite 393216 bits.

1.5 LES DIFFERENTES APPROCHES EN CONNAISSANCE DES FORMES

Il existe deux approches en reconnaissance des formes :

- l'approche statistique,
- l'approche structurelle.

En plus de ces deux approches, il existe une nouvelle approche: l'approche connexioniste.

I.5.1 Approche statistique

Le principe de l'approche statistique consiste à faire un certain nombre de mesures sur un objet à reconnaître et à étudier la répartition de ces mesures dans un espace métrique afin de trouver la plus forte probabilité d'appartenance de l'objet à une classe de formes.

Il existe deux grands types de méthodes dans cette approche: les méthodes paramétriques, où l'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; les méthodes dites non paramétriques ou bayésiennes, où l'on se donne un modèle de la distribution de chaque classe, et où l'on cherche la classe à la quelle le point a la probabilité la plus grande d'appartenir. Dans le premier cas, une tactique simple consiste à définir des hyperplans qui séparent au mieux les classes d'apprentissage. La décision se réduit alors à une série de produits scalaires. Ces méthodes sont dites de séparation linéaire. La détermination des équations de ces hyperplans a fait l'objet de nombreux travaux, et les probabilités théoriques d'erreur correspondantes ont été largement étudiées [12].

Une autre méthode non paramétrique très utilisée 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 que la répartition des points dans chaque classe n'est pas assez régulière, ou les points ne sont pas assez nombreux, pour que la précision des calculs soit proche de celle que donne la théorie. Ces réserves faites, ce type de méthode est rapide et donne de bons résultats [12].

1.5.2 Approches structurelles

Les approches statistiques et connexionistes ne permettent pas de prendre en compte l'information structurelle et contextuelle d'une forme. Les deux structures les plus courantes sont :

- les structures de graphes utilisées, soit comme modèle descriptif, soit comme outil de mise en correspondance entre un modèle et une forme [4].

- les structures syntaxiques qui sont inspirées de la théorie des langues formelles. Une forme est analysée syntaxiquement comme une phrase d'un langage à l'aide d'une grammaire appropriée [4].

1.5.2.1 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 résout 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 [4].

1.5.3.2 Structure syntaxique

Dans les approches syntaxiques, on cherche et on exploite les règles de construction des formes à partir de leurs composants. D'où l'analogie que l'on peut faire avec l'analyse grammaticale ou syntaxique. La tendance consiste à hiérarchiser la structure des formes. En effet, à partir d'un ensemble d'éléments primitifs de forme 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 [4].

1.5.3 Approches connexionistes

Les approches connexionistes consistent à appliquer le concept des réseaux connexionistes 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. Les deux structures les plus utilisées dans le domaine de la reconnaissance des caractères sont :

- la structure multicouche,
- la structure en fenêtres.

Ces deux structures seront détaillées dans le chapitre suivant.

1.6 CONCLUSION

Les méthodes statistiques et structurelles ont fait l'objet de plusieurs travaux:[1,2,6]. L'apparition des réseaux connexionistes a poussé les chercheurs à les appliquer à la reconnaissance des formes, en particulier les caractères pour comparaître les résultats obtenus et ceux obtenus par les méthodes classiques.

Pour les deux raisons citées plus haut, nous avons choisis le concept des réseaux connexionistes pour les appliquer à la reconnaissance des chiffres.

CHAPITRE 2

RESEAUX CONNEXIONISTES

II.1 INTRODUCTION

Actuellement, les réseaux connexionistes (réseaux de neurones) trouvent leur application dans le domaine du traitement de signal (image, parole, ..., etc), tant au niveau de prétraitement qu'au niveau de reconnaissance. Deux fonctions sont généralement présentes:

- Un prétraitement adéquat,
- Une classification optimale.

Avant d'aborder l'application concernant la reconnaissance des chiffres, en utilisant les réseaux connexionnistes, ce chapitre sera consacré aux définitions du neurone biologique et du neurone formel, aussi à la description détaillée de l'architecture en couches des réseaux connexionnistes, et à la fin on introduira l'algorithme d'apprentissage.

II.2 NEURONE BIOLOGIQUE

Les cellules nerveuses, appelées neurones, sont les éléments de base du système nerveux central. Celui-ci en posséderait environ cent milliards. Un neurone est une cellule constituée de trois parties (figure II.1).

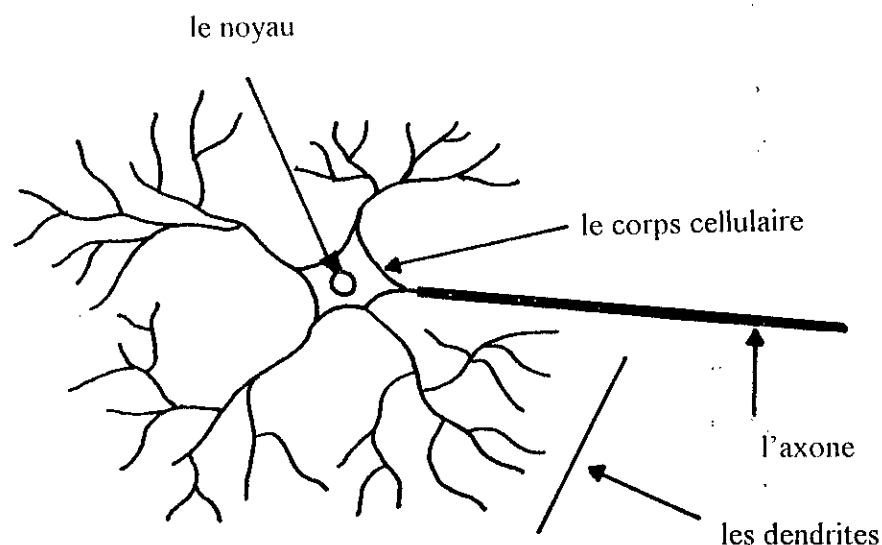


FIGURE II.1: *neurone biologique*

II.2.1 Les dendrites

Elles forment une espèce d'arborescence autour du corps cellulaire (soma). Elles sont les récepteurs principaux du neurone pour capter les signaux, venants d'autres cellules ou de l'extérieur

II.2.2 le corps cellulaire ou soma

Il reçoit les informations sous forme de signaux électriques collectées par les dendrites et les somme. Si le potentiel somatique (la somme obtenue) dépasse un certain seuil, il émet un signal bref (d'environ une milliseconde) qu'on appelle signal d'activation ou «spike».

II.2.2 L'axone

L'axone sert de moyen de transport, sans atténuation, pour les signaux émis par le soma .

Le schéma classique, résumant les principales fonctions d'un neurone, présenté par les biologistes est donné dans la figure II.2.

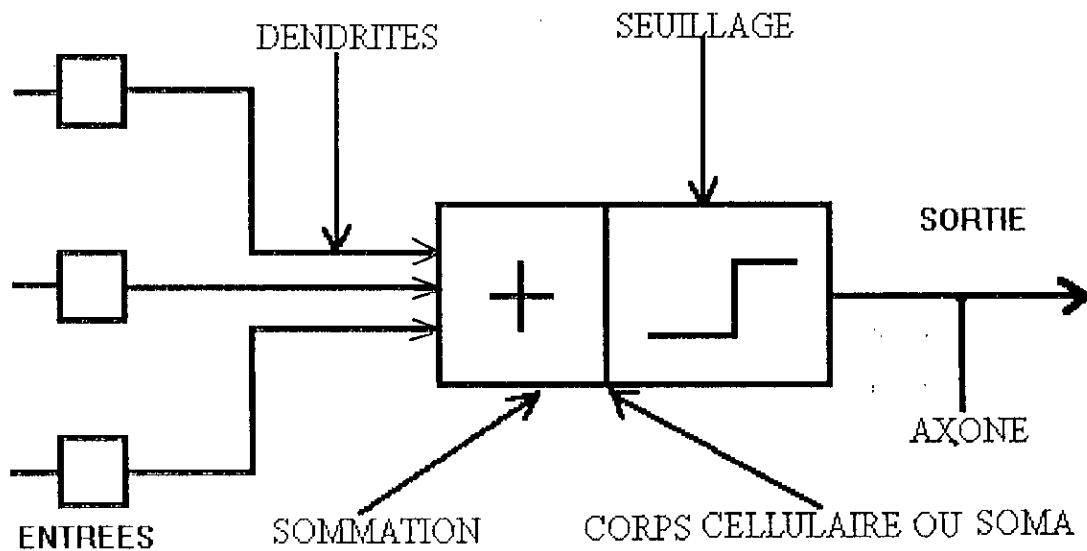


FIGURE II.2: schéma fonctionnelle d'un neurone biologique.

Pour former le système nerveux, les neurones sont connectés les uns aux autres suivant des répartitions spatiales complexes. La connexion entre de neurones se fait en des endroits appelés **synapses**.

II.1.4 Les synapses

Le rôle des synapses est fondamental pour permettre aux cellules nerveuses de communiquer entre elles (figure II.3).

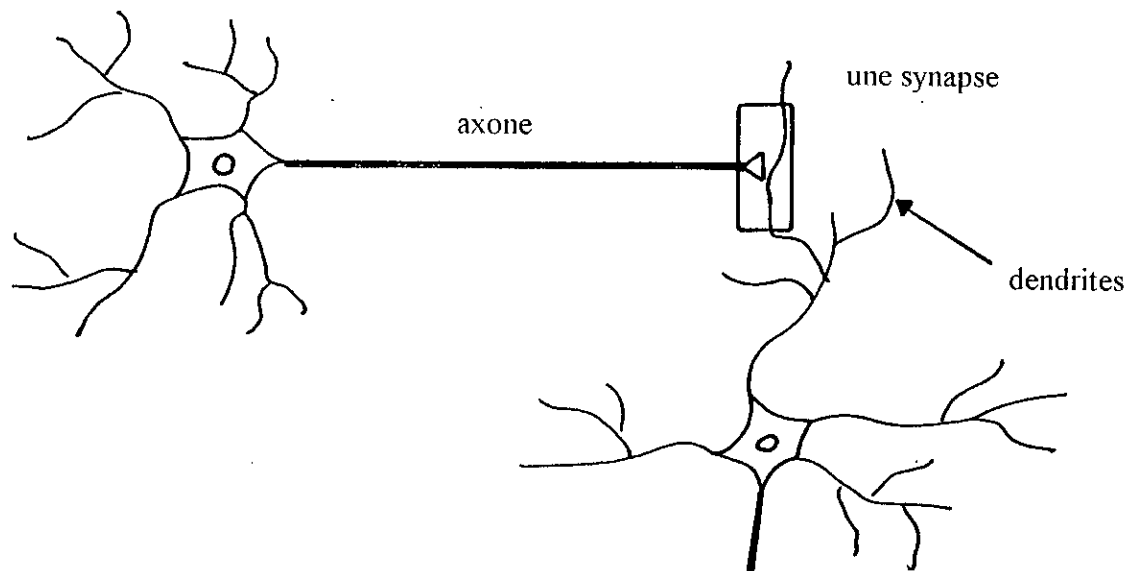


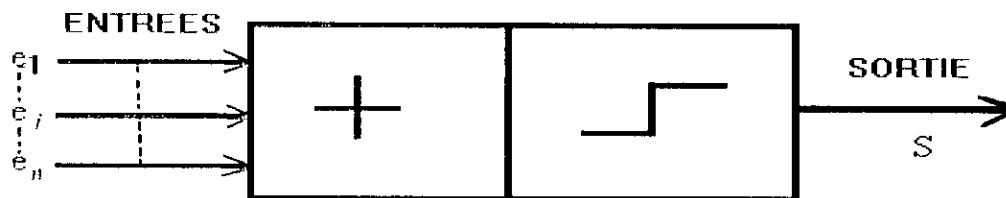
FIGURE II.3: connexion de deux neurones

Pour plus de détails sur les neurones biologiques, les références [5,8] présentent de façon très simple le fondement biologique et la modélisation des réseaux de neurones.

II.3 LE NEURONE FORMEL

En 1943, MC CEULLOCH et PITTS [5,8] ont étudié un ensemble de neurones formels interconnectés, et ont montré leurs capacités à calculer certaines fonctions logiques. Ensuite ils ont proposé la première modélisation d'un neurone.

Un neurone formel est un circuit de calcul (figure II.4), qui fait la somme pondérée des potentiels d'action parvenants d'autre neurones (chacun de ces potentiels est une valeur numérique qui présente l'état du neurone qui l'a émis), puis s'active suivant la valeur de cette somme pondérée. Si cette somme dépasse un certains seuil, le neurone s'activera et transmettra une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien [8].

FIGURE II.4: *neurone formel.*

Pour le reste (les sections II.4 et II.5) on désigne par:

- $(e_i)_{i=1,n}$: les entrées du neurone formel,
- $(w_i)_{i=1,n}$: les paramètres de pondération,
- E : l'entrée totale,
- h : la fonction d'entrée totalement,
- f : la fonction d'activation,
- β : le seuil du neurone,
- A : l'état du neurone,
- g : la fonction de sortie,

donc

$$E = h \left(\sum_{i=1}^n w_i \times e_i \right),$$

$$A = f (E),$$

$$S = g (A),$$

avec

$$f (E) = \begin{cases} 1 & \text{si } E \succ \beta \\ 0 & \text{si } E \prec \beta \end{cases}$$

La fonction f est illustrée dans la figure II.5.

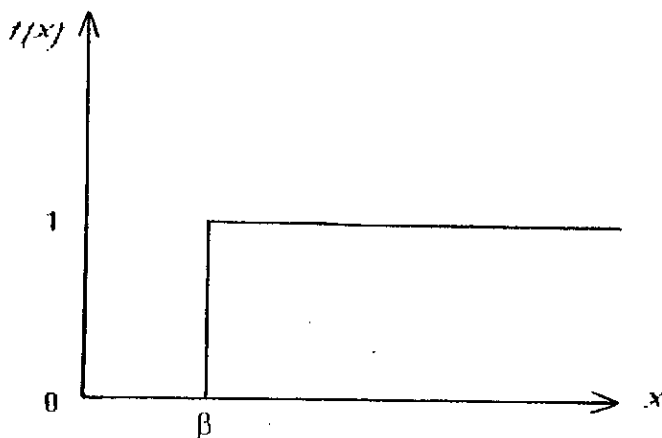


FIGURE II 5: fonction d'activation.

II.4 MODELISATION GENERALE

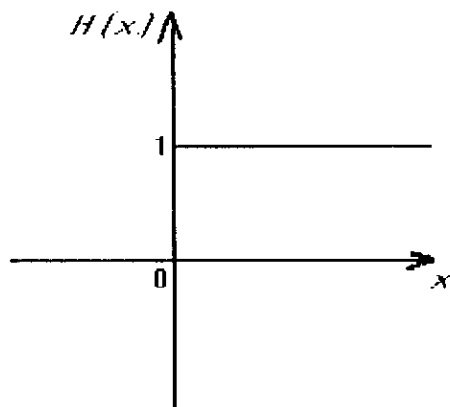
En général pour définir un neurone formel [8], il faut spécifier :

- la nature des entrées : binaires (0 ou 1, 1 ou 1), ou réelles ;
- la fonction d'entrée :

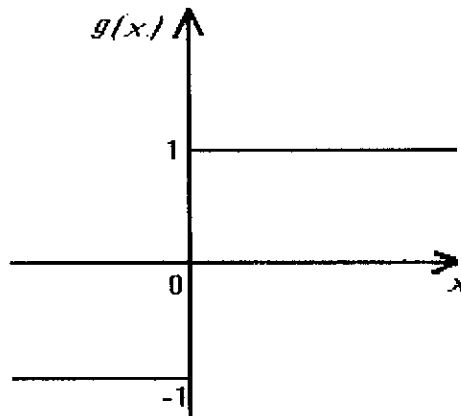
* linéaire $h (e_1, \dots, e_n) = \sum_{i=1}^n w_i \times e_i,$

** ou affine $h (e_1, \dots, e_n) = \sum_{i=1}^n w_i \times e_i - c.$

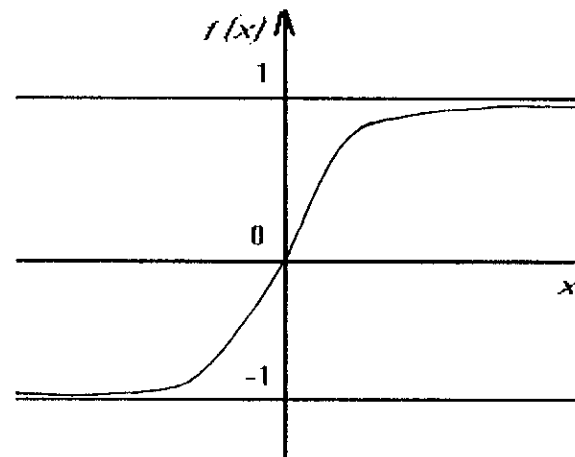
- la fonction d'activation : fonction binaire, HEAVISIDE ou sign, linéaire à seuils, ou fonction sigmoïde, (figure II.6)
- la fonction de sortie considérée, en générale comme la fonction identité.



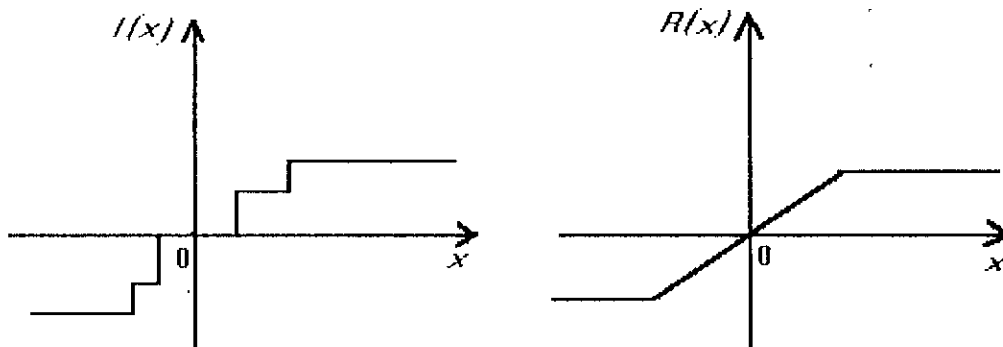
a: fonction de HEAVISIDE



b: fonction sign



c: fonction sigmoïde



d: fonction linéaire multiseuil

e: fonction linéaire à seuil

FIGURE II.6 fonctions d'activation

II.5 RESEAUX CONNEXIONISTES

L'association de plusieurs neurones connectés entre eux suivant différentes structures est appelée réseau connexioniste, vu le grand nombre de connexions qui caractérise ces réseaux. Ils sont souvent appelés réseaux de neurones [8.14].

La structure la plus simple est celle d'un réseau multicouche : les neurones appartenant à la même couche ne sont pas connectés entre eux; chacune des couches reçoit des signaux de la couche précédente, et transmet le résultat de ses traitements à la couche suivante.

La première couche qui reçoit les entrées du milieu extérieur, est appelée couche d'entrée, tandis que la dernière couche est appelée couche de sortie. Les couches intermédiaires sont appelées couches cachées (figure II.7).

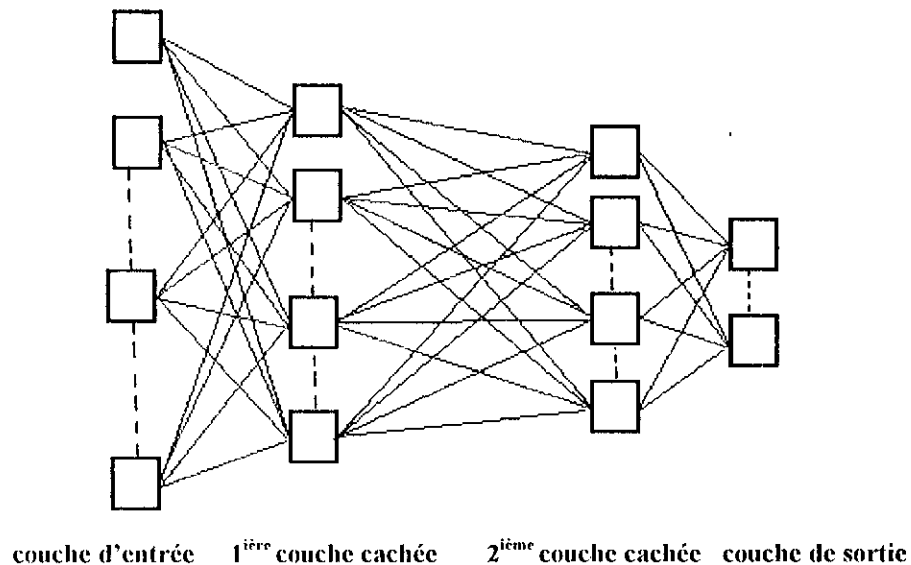


FIGURE II.7 réseau multicouche

II.5.1 Dynamique des connexions

Dans un réseau de neurones formels, les différents neurones sont reliés entre eux par des connexions (pour cela l'appellation réseau connexioniste) représentant les synapses du modèle biologique.

Pour modéliser le fonctionnement des synapses, on affecte, à chaque connexion entre deux neurones formels, un poids qui pondère le signal transmis.

La dynamique des connexions ne s'intéresse qu'aux réseaux dont les poids des connexions sont évolutifs. L'évolution du poids est liée à l'information que reçoit le neurone par la connexion relative à ce poids au cours de l'apprentissage [8].

II.5.2 La règle de HEBB

En 1949, HEBB [5.8] énonçait une règle qualitative, qui est le premier mécanisme d'évolution des synapses. Le point fondamentale de l'observation de HEBB est que le renforcement synaptique intervient lorsqu'il y a activité conjointe de neurone prés-synaptique (cause) et post-synaptique(effet) [5, 8]. Chaque neurone a deux états possibles : actif ou inactif.

Les quatre états conjoints de deux neurones connectés entre eux sont résumés dans le tableau II.1.

TABLEAU II.1: états conjoints de deux neurones connectés .

<i>état du neurone prés. synaptique</i>	<i>état du neurone post. synaptique</i>	<i>évolution de synapse</i>
<i>actif</i>	<i>actif</i>	<i>renforcement</i>
<i>inactif</i>	<i>actif</i>	<i>pas de modification</i>
<i>actif</i>	<i>inactif</i>	« « « «
<i>inactif</i>	<i>inactif</i>	« « « «

La modélisation de la règle de HEBB

soit $w_{ij}(t)$ le poids de la connexion entre le neurone j et le neurone i à l'instant t . Au cours de l'apprentissage on a:

$$w_{ij}(t + \partial t) = w_{ij}(t) + \mu A_i A_j \quad \text{avec } \mu > 0. \quad (II.1)$$

μ est un paramètre positif de l'intensité de l'apprentissage, et A_i, A_j sont respectivement les états neurones i, j . L'apprentissage n'est rien d'autre que la modifications des poids des connexions.

Il existe deux classes de règles d'apprentissage qui diffèrent par leur source d'inspiration:

- règles d'apprentissage à source biologique qui se base sur la règle de HEBB;

- règles d'apprentissage à source mathématique: pour cette classe le réseau est considéré comme une fonction de transfert des entrées présentées. Pour calculer les bons paramètres de cette fonction de transfert, des algorithmes, basés sur la minimisation de la fonction d'erreur quadratique, comme la descente du gradient [3,8] et la méthode d'optimisation aléatoire [3], sont élaborés.

II.6 STRUCTURE MULTICOUCHE

La structure proposée dans la section II.5 est utilisée pour associer une forme, un motif ou une représentation d'une situation quelconque, à une autre de nature similaire ou totalement différente [8].

Pour simplifier la compréhension du fonctionnement des réseaux multicouches, connexioniste ou propose le réseau monocouche de la figure II.8.

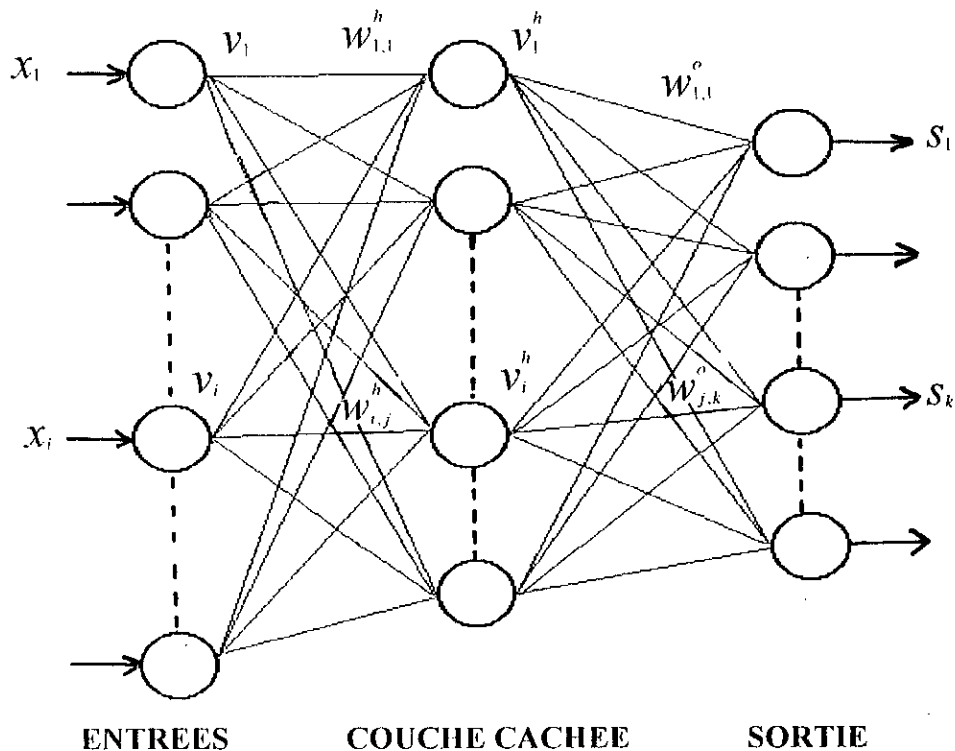


FIGURE II.8 réseau monocouche

Avec

$X = \{x_1, x_2, \dots, x_n\}$, vecteur d'entrée e.

$S = \{s_1, s_2, \dots, s_l\}$, vecteur de sorties calculées.

$V = \{v_1, v_2, \dots, v_n\}$, vecteur de sortie de la couche d'entrée e.

$V^h = \{v_1^h, v_2^h, \dots, v_m^h\}$, vecteur de sortie de la couche cachée

$Y = \{y_1, y_2, \dots, y_l\}$, vecteur de la sortie désirée e.

$(W_{i,j}^h)$ poids de la couche cachée e.

$(W_{i,k}^o)$ poids de la couche de sortie.

avec $f(x) = 1 / (1 + \exp(-a \times x))$.

où f est la fonction d'activation. Calculons les sorties de chaque couche:

$$v_i = f(x_i),$$

$$v_j^h = f\left(\sum_{i=0}^n W_{i,j}^h \times v_i\right),$$

$$s_k = f\left(\sum_{j=0}^m W_{j,k}^o \times v_j^h\right),$$

Cependant, il faut noter que dans les réseaux multicouches, l'apprentissage est supervisé.

II.6.1 L'apprentissage

La plupart des réseaux connexionistes proposés jusqu'à ce jour sont des réseaux susceptibles de subir un apprentissage. Au cours de cette phase d'apprentissage, les poids sont calculés de telle sorte que le réseau remplisse une tâche (classification, prédiction des signaux temporels, ...) définie par des exemples (et non par des règles), c'est à dire par un exemple de couple entrée-sortie désiré dit « ensemble de d'apprentissage » [15]. Le plus souvent, il s'agit d'un apprentissage supervisé: un « professeur » présente au réseau une donnée d'entrée et la sortie (réponse) souhaitée correspondante (ou valeur désirée). Par exemple, la donnée d'entrée peut être un chiffre, et la donnée de sortie le code binaire de ce chiffre. Les poids sont calculés de manière à minimiser un critère qui mesure la qualité des réponses du réseau aux exemples de l'ensemble d'apprentissage [15]. Un critère souvent utilisé est la somme des carrés des différences entre les sorties désirées et les sorties fournis par le réseau de tout l'ensemble d'apprentissage [5,8].

Soit $X = \{x^1, x^2, \dots, x^P\}$, l'ensemble des exemples d'apprentissage, où P est le nombre d'exemples.

L'erreur quadratique relative à l'exemple p est donnée par l'équation II.2:

$$E_p = \sum_{k=1}^l (s_k^p - y_k^p)^2, \quad (\text{II.2})$$

d'où l'erreur quadratique totale, relative à l'ensemble d'apprentissage, est donnée par l'équation (II.3):

$$E = \sum_{p=1}^P E_p, \quad (\text{II.3})$$

Cette erreur dépend des poids du réseau, et sa minimisation consiste à faire varier les poids jusqu'à atteindre le minimum globale de cette fonction d'erreur : c'est l'étape d'apprentissage.

Une fois l'apprentissage effectué, le réseau peut être utilisé pour la tâche prévue (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 ». Dans certains cas , la phase d'apprentissage et la phase d'utilisation sont confondues: le réseau apprend en permanence et de manière analogue aux systèmes adaptatifs de traitement de signal [15].

II.6.2 Représentation des formes

La représentation des exemples se fait d'une manière générale par des vecteurs de nombre réels. Le choix d'une représentation pour un classifieur est difficile. Un système de reconnaissance de forme est en général constitué d'une cascade de modules dont le classifieur est le dernier. La difficulté de la tâche qui lui est demandée pour satisfaire les performances du cahier des charges dépend essentiellement de la représentation choisie [15]. Les modules précédents (acquisition, filtrage, numérisation, ..., etc) sont autant de transformations des formes à classer qui doivent être conçues spécifiquement pour aboutir à une représentation adéquate au classifieur choisi [15].

II.7 STRUCTURE EN FENETRES

Une autre structure est très utilisée dans la reconnaissance des caractères est la structure en fenêtres [4.8]. Cette structure ne diffère de la structure en multicouche que par les connexions. La couche

d'entrée est connectée à la première couche cachée, où chaque neurone de cette couche cachée reçoit ses entrées non pas de l'ensemble de la couche précédente mais seulement d'un petit voisinage d'un neurone (par exemple une fenêtre 3×3). On peut continuer avec plusieurs autres couches successives, constituées sur le même principe. Dans cette structure, le nombre de connexions est considérablement réduit, ce qui réduit d'autant les temps de calcul (figure II.9).

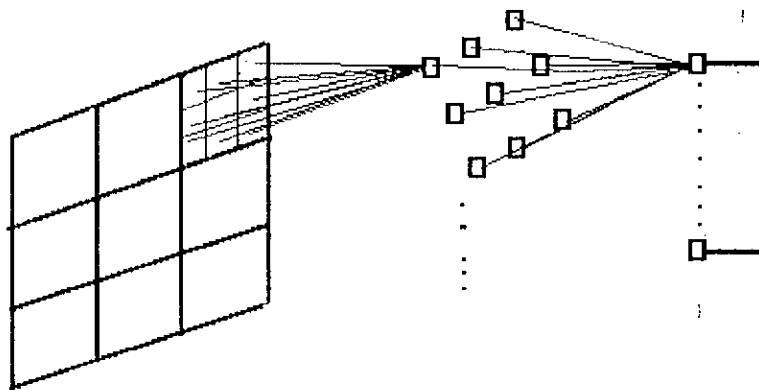


FIGURE II.9: réseau structuré en fenêtres.

Dans cette structure les calculs de sorties se font de la même manière que ceux de la structure multicouche. Aussi les algorithmes d'apprentissage s'appliquent de la même manière.

II.8 ALGORITHME D'APPRENTISSAGE: méthode d'Optimisation Aléatoire Modifiée (M.O.A)

Plusieurs algorithmes d'apprentissage ont été établis, en s'inspirant de la minimisation de la fonction d'erreur. Le plus connu de ces algorithmes est *la rétropropagation* [8,10,14]. Cependant, la convergence vers un minimum global n'est pas assurée par cet algorithme qui utilise la descente du gradient pour minimiser l'erreur totale. Aussi, il a une autre contrainte c'est le problème des minimas locaux [8,10,14].

Pour cette raison, dans ce travail, nous avons utilisé l'algorithme découlant de la méthode d'optimisation aléatoire (SOLIS-WETS 1981) élaboré par NORIO BABA à l'université de TOKUSHIMA en 1988 appelé méthode d'optimisation aléatoire modifiée. Cet algorithme assure la convergence vers un minimum global avec une probabilité égale à un [3].

Soit $f(x)$ la fonction à minimiser où $x \in X$ (X le domaine de recherche), $\xi(k)$ le vecteur aléatoire gaussien de variance constante et moyenne $b(k)$ variable à chaque itération k (voir génération de bruit blanc gaussien dan l'annexe 2), et M le nombre d'itérations.

Algorithme

- I initialiser $x(0)$, $b(k) = 0$,
- II si $x \in X$ aller à III,
sinon stop.
- III si $f(x(k) + \xi(k)) < f(x(k))$ alors :
 $x(k+1) = x(k) + \xi(k)$,
 et $b(k+1) = 0.2b(k) + 0.4\xi(k)$.
 sinon
 si $f(x(k) + \xi(k)) \geq f(x(k))$ et $f(x(k) - \xi(k)) < f(x(k))$ alors:
 $x(k+1) = x(k) - \xi(k)$,
 et $b(k+1) = b(k) - 0.4\xi(k)$.
 sinon $x(k+1) = x(k)$, et $b(k+1) = 0.5b(k)$.
- IV $k = k + 1$,
 si $k > M$ stop
 sinon aller vers II.

Dans les réseaux connexionistes, la fonction $f(x)$ est remplacée par la fonction d'erreur quadratique totale du réseau, et x par le vecteur poids (organigramme dans l'annexe 4). Dans les réseaux connexionistes, il n'existe pas de lois qui permettent de déterminer le nombre de couches cachées, ou le nombre de neurones par couche cachée. Cela est laissé à l'essai .

II.9 CONCLUSION

Les algorithmes de minimisation de fonction ont été adaptés aux réseaux connexionistes pour obtenir des algorithmes d'apprentissage. Il s'agit de minimiser la fonction d'erreur quadratique totale, jusqu'au minimum global de cette fonction. Le plus utilisé de ces algorithmes d'apprentissage pour les réseaux multicouches est la rétropropagation que nous avons évité d'utiliser dans notre travail vu ses inconvénients. Par contre, l'algorithme de l'optimisation aléatoire modifiée garantit la minimisation jusqu'au minimum global, lorsque le réseau ne se sature pas.

Les deux structures présentés dans ce chapitre seront utilisés par la suite dans l'application des réseaux connexionistes à la reconnaissance des chiffres.

CHAPITRE 3

APPLICATIONS

III.1 INTRODUCTION

Le travail réalisé, dans ce projet, est une application directe des réseaux connexionistes appliqués à la reconnaissance des caractères: la reconnaissance des chiffres. Il s'agit de l'étape d'apprentissage et de décision. Pour cela les étapes qui précèdent cette étape ne seront pas développées dans notre travail. Dans cette application nous n'utiliserons que les images des chiffres en noir et blanc, c'est à dire que leurs matrices représentatives ne possèdent que des zéros (0) et des uns (1).

Avant de passer à l'étape d'apprentissage, on parlera d'abord de l'acquisitions des chiffres manuscrits, de la génération des exemples d'apprentissage, du codage de la sortie, des structures des réseaux et enfin de l'étape d'apprentissage et de décision.

III.2 ACQUISITION DES EXEMPLES D'ENTRAINEMENT

Dans cette application, les exemples d'apprentissage, qui sont des chiffres de zéro (0) à neuf (9), ont été générés sur micro-ordinateur. Une fois l'apprentissage achevé, la capacité de généralisation des réseaux sera testée par un ensemble de chiffres manuscrits.

III.2.1 Chiffres manuscrits

Les chiffres manuscrits utilisés pour le test sont des chiffres écrits par trois personnes sur feuilles blanches, ensuite scanner par un scanner de type « HP SCAN JET IIp » (HPC 1790 A) avec une résolution optique de trois cents dpi, et, des résolutions horizontale et verticale de soixante dix huit dpi. Cette résolution (78 dpi) permet d'avoir des chiffres localisés dans des fenêtres de 16×16 pixels.

Une fois les chiffres scannés, l'opération suivante c'est de localiser les chiffres dans des fenêtres de 16×16 pixels, pour l'étape de test. Ces chiffres sont donnés dans l'annexe 6.

II.2.2 Chiffres générés

L'utilisation du logiciel « paint-brush » sous *WINDOWS* a permis la génération de soixante dix chiffres, grâce à la variation de style d'écriture dans ce logiciel. Trente chiffres sont centrés dans des fenêtre de 15×10 pixels, tandis que quarante sont centrés dans des fenêtres de 16×16 pixels. Ces chiffres seront donnés dans l'annexe 5.

Il reste à mentionner que tous les images des chiffres sont sauvegardées dans des fichiers de format TIFF. Les spécifications de cette format seront données dans le paragraphe suivant.

III.2.3 Format TIFF (Tagged Image File Format) [16]

Le grand avantage du format TIFF est sa flexibilité : les images peuvent être en couleur, ou en échelle de gris, et avec une taille aussi grande que voulue. Un seul fichier TIFF peut contenir plusieurs images. Cette flexibilité est le résultat de l'utilisation systématique des pointeurs au sein du fichier TIFF, qui présente une organisation identique à celle des répertoires d'un disque. Cette organisation est elle même décrite au sein du fichier.

Sur les huit premiers octets du fichier TIFF (figure III.1), on peut lire:

- la spécification du format utilisé, sur deux octets(II pour Intel, MM pour Motorola).
- le numéro de version du format TIFF, sur deux octets,
- un pointeur, calculé en octets, vers le premier répertoire.

et chaque répertoire est organisé en champs.

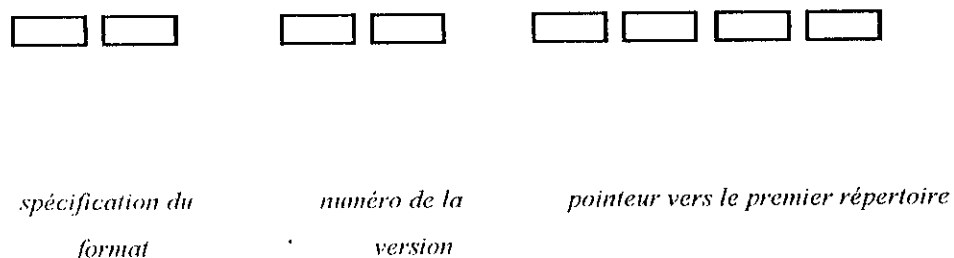


FIGURE III.1 : l'entête du fichier TIFF

Le fichier TIFF est organisé en répertoire, et chaque répertoire est organisé en champs (figure III .2).

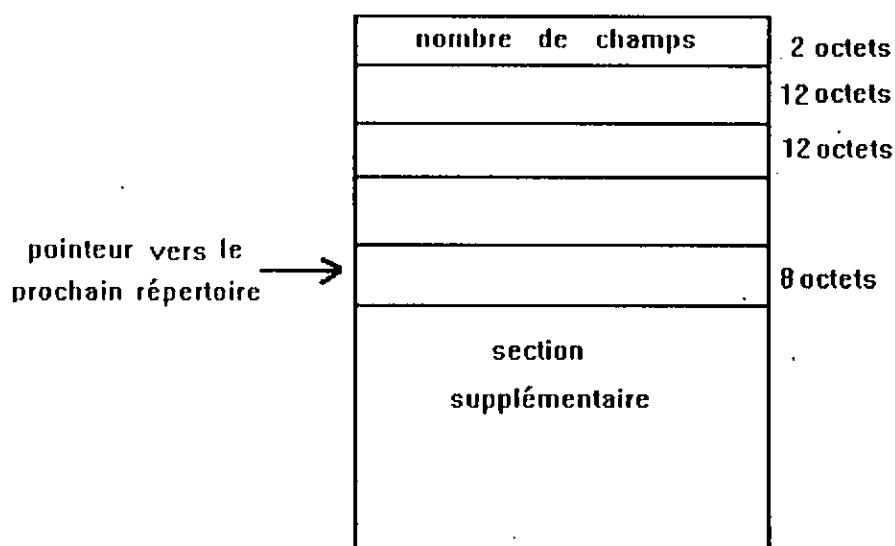


FIGURE III.2: organisation d'un répertoire

A l'entête du répertoire on peut lire sur deux octets le nombre de champs du répertoire (les « tages », ou étiquette), où chaque champs est codé sur douze octets. A la fin du répertoire se trouve un pointeur vers le répertoire suivant; entre eux, il peut exister une section réservée aux informations qui ne tiennent pas sur les douze octets alloués aux champs. Dans ce cas l'information contenue dans le champs est un pointeur sur cette section supplémentaire.

Chaque champ contient une information, sur l'image, codé sur douze octets. Les deux premiers octets correspondent au type d'information (largeur d'image, nombre de bits par pixel, résolutions,etc). Chacune de ces informations est repérée par une valeur appropriée. Les deux octets suivants indiquent le type des données:

- octet (8 bits) représenté par un (1),
- code ASCII (8 bits) représenté par deux (2),

- short (16 bits) représenté par trois (3),
- long (32 bits) représenté par quatre (4),
- Rational, rapport de deux longs, représenté par cinq (5).

Les quatre octets suivants indiquent le nombre de données. Enfin les quatre derniers octets indiquent la donnée elle-même ou bien (si cette donnée ne tient pas sur ces quatre octets) un pointeur sur la section supplémentaire où cette information est stockée.

Le logiciel pilotant le « HP SCAN JETIIp », utilise le format TIFF possédant les spécifications suivantes:

- 254: nouveau ou ancien fichier,
- 256: largeur de l'image,
- 257: hauteur de l'image,
- 258: nombre de bits par pixel,
- 259: présence ou absent de compression,
- 273: pointeur vers l'offset,
- 277: nombre d'échantillonnages (la valeur une (1) représente une image en noir et blanc ou en échelle de gris),
- 278: nombre de rangs par bande,
- 279: nombre de bytes par bande,
- 282: résolution horizontale,
- 283: résolution verticale,
- 296: unité de résolution (2 pour dpi).

L'offset est un pointeur vers le bit-map de l'image. Les spécifications des images des chiffres utilisées sont données dans l'annexe 3.

III.2.4 Algorithme de lecture des images

Compte tenu des informations obtenues sur l'image, l'algorithme de lecture suivant peut être appliqué pour lire l'image en binaire:

soit M et N la hauteur et la largeur de l'image en pixel.

I) Lire M et N .

II) Si N est un multiple de 8 aller vers III.

Sinon

1) $m =$ reste de division de N par 8.

2) $n =$ partie entière inférieure de $(N/8) + 1$.

3) $i = 0$.

4) Lire 1 octet.

5) $i = i + 1$.

6) Si $i > n \times M$ stop.

Sinon lire m premiers bits de l'octet suivant.

7) $i = i + 1$.

8) Si $i > n \times M$ stop.

Sinon aller vers 4).

III)

1) Lire 1 octet.

2) $i = 0, n = N/8$.

3) $i = i + 1$.

4) Si $i > n \times M$ stop.

Sinon aller vers 1).

IV) Fin.

III.3 CODAGE DE LA SORTIE

Deux codes sont choisis pour représenter la sortie désirée

1^{er} code: la sortie est codée sur 5 bits (tableau III .1).

TABLEAU III 1 : sortie codée sur 5 bits.

<i>chiffres</i>	<i>code</i>
0	0000 0
1	1000 0
2	0100 0
3	0010 0
4	0001 0
5	0000 1
6	1000 1
7	0100 1
8	0010 1
9	0001 1

2^{ème} code: la sortie est codé sur 1 bits (tableau III.2).

TABLEAU III 2: sortie codée sur 1 bit.

<i>chiffre</i>	<i>code</i>
0	0.0
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
6	0.6
7	0.7
8	0.8
9	0.9

La fonction d'activation choisie (des neurones) a une valeur comprise entre zéro (0) et un (1). A cet effet, la valeur de la sortie désirée de chaque neurone doit être comprise entre zéro (0) et un (1).

III.4 SPECIFICATIONS DES RESEAUX UTILISES

Les réseaux utilisés ont les spécifications suivantes:

- les entrées sont des zéros (0) ou des uns (1).
- la fonction d'entrée est linéaire.
- la fonction d'activation est une sigmoïde. elle est donnée par l'équation III .1 :

$$f(x) = 1 / (1 + \exp(-a \times x)) \quad (\text{III .1})$$

où a est un facteur multiplicatif positif non nul (figure III .3). Le choix de cette fonction permet d'éviter les états de saturation des réseaux.

- la fonction de sortie est l'identité [8].

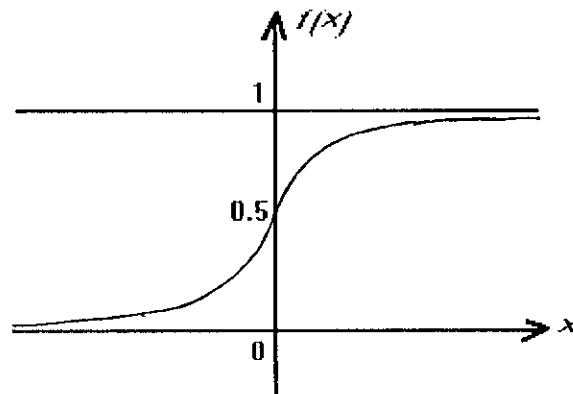


FIGURE III .3: fonction d'activation.

III. 5 STRUCTURES ET DIMENSIONS DES RESEAUX UTILISES

Deux structures différentes sont utilisées pour construire des réseaux pour atteindre l'objectif de notre travail.

III.5.1 Structure multicouche

La structure multicouche est choisie pour construire des réseaux de différentes dimensions. L'entraînement de ces derniers est fait par les 30 images de 15×10 pixels des chiffres générées. Ces réseaux sont constitués :

- d'une couche d'entrée de cent cinquante neurones,
- d'une ou de deux couches cachées,
- d'une couche de sortie d'un ou de cinq neurones.

Les neurones de chaque couche sont connectés à tous les neurones de la couche suivante (figure III.4).

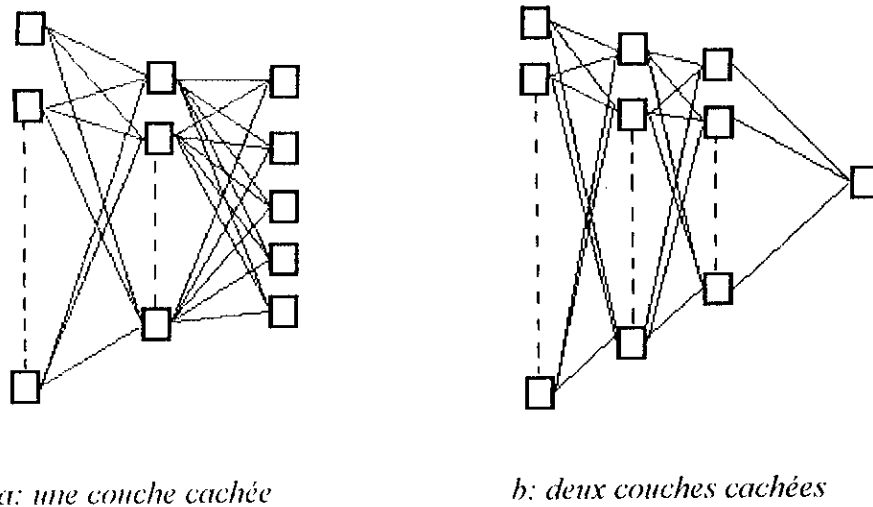


FIGURE III.4: *structure multicouche.*

Les dimensions des réseaux, à une couche cachée sont résumées dans le tableau III.3, et les dimensions des réseaux, à deux couches cachées sont résumées dans le tableau III.4.

TABLEAU III .3: dimensions des réseaux à une couche cachée.

<i>réseau</i>	<i>couche d'entrée</i>	<i>couche cachée</i>	<i>couche de sortie</i>
1	150	40	5
2	150	30	5
3	150	20	5
4	150	10	5

TABLEAU III .4: dimensions des réseaux à deux couches internes.

<i>réseaux</i>	<i>couche d'entrée</i>	<i>1^{ère} couche cachée</i>	<i>2^{ème} couche cachée</i>	<i>couche de sortie</i>
1	150	10	7	5
2	150	10	5	5
3	150	7	7	5
4	150	7	6	5
5	150	7	7	1
6	150	7	6	1
7	150	7	5	1

Le passage des réseaux à une couche cachées aux réseaux à ceux de deux couches cachées, est fait pour diminuer le nombre de connexions de la couche de sortie, ainsi que le nombre de couche de sortie passe de cinq à un neurone

Tous les réseaux cités ci-dessus se retrouvent dans des états de saturation. Pour dépasser cette contrainte nous avons utilisé une autre structure où le nombre de connexions est considérablement réduit.

III.5.2 Structure en fenêtres

Dans cette structure, seules les connexions entre la couche d'entrée et la première couche cachée sont semblables à celles existantes dans la structure en fenêtres. Les connexions entre les autres couches sont semblables à celles d'une structure multicouche (Figure III.5).

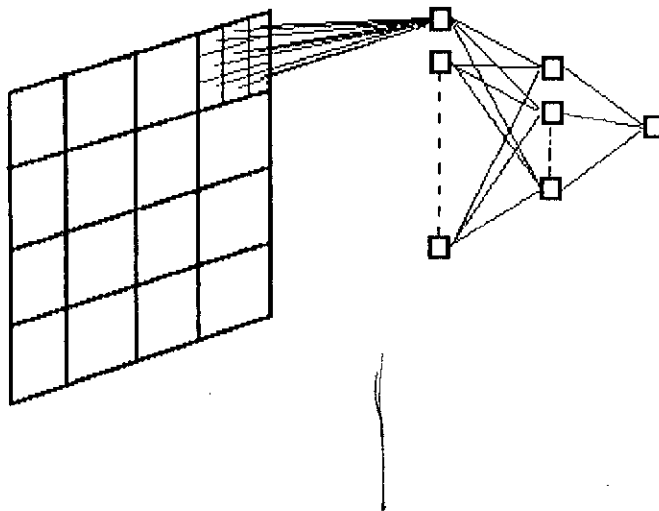


FIGURE III .5: structure en fenêtres

Pour les images de 15×10 pixels, un seul réseau à deux couches cachées sera entraîné. La fenêtre utilisée pour la décomposition de l'image est de 3×5 pixels. Chaque neurone de la première couche cachée n'est connecté qu'à quinze neurones de la couche d'entrée. Le nombre de connexions est réduit de cent cinquante neurones à quinze neurones. Ce réseau a convergé après cent quatre vingt dix huit milles itérations.

Après ce résultat, nous avons augmenté les dimensions des images chiffres de 15×10 pixels à 16×16 pixels. Cette fois deux réseaux à deux cents cinquante six neurones dans la couche d'entrée sont entraînés. L'un utilise une fenêtre de 4×4 pixels et l'autre utilise une fenêtre de 4×8 pixels. Le premier réseau tombe dans un état de saturation, tandis que l'autre converge après cent trente sept milles itérations. Les dimensions des trois réseaux entraînés sont résumées dans le tableau III .5.

TABLEAU III .5: dimensions des réseaux aux couche d'entrée à 256 neurones.

<i>réseau</i>	<i>fenêtre</i>	<i>couche d'entrée</i>	<i>1^{ère} couche cachée</i>	<i>2^{ème} couche cachée</i>	<i>couche de sortie</i>
1	3×5	150	10	5	1
2	4×4	256	16	8	1
3	4×8	256	8	4	1

III.6 ETAPE D'APPRENTISSAGE

Dans les réseaux connexionistes, on ne peut pas analyser ni spécifier explicitement la fonction effectuée. Il faut donc adopter une méthode d'apprentissage qui permette à un réseau donné de trouver automatiquement cette fonction à partir des exemples d'apprentissage. Donc, aucune règle ne sera spécifier pour effectuer le classement pendant l'apprentissage [4]. L'algorithme utilisé pour l'apprentissage est le M.O.A. Il n'est pas spécifique à cette application. Au cours de cette étape, les images chiffres sont présentées à l'entrée du réseau accompagné de leurs sorties appropriées: c'est un apprentissage supervisé.

III.7 ETAPE DE DECISION

Une fois l'apprentissage terminé, on passe à l'étape de reconnaissance (c'est tester et prendre une décision). Pour cela, il faut établir un algorithme de décision pour la sortie calculée par le réseau pour une image chiffre lu. Dans l'étape de reconnaissance, on conserve le même réseau qui a été

entraîné en attribuant à chaque connexion son poids obtenu à la fin de l'apprentissage. Les neurones conservent les mêmes spécifications que celle de l'étape d'apprentissage.

Une fois la sortie calculée pour n'importe quel chiffre, on passe à l'étape de décision. Trois cas possibles se présentent :

- la reconnaissance du chiffre,
- l'erreur ou la confusion de reconnaissance du chiffre,
- le rejet du chiffre.

Algorithme de décision

- I) Lire le chiffre.
- II) Calcul de sortie S par le réseau,
- III) Si $(S \geq 9.5)$ alors le chiffre est rejeté.
Sinon:
 - 1) $S' =$ partie entière inférieure de S .
 - 2) $t = S - S'$.
 - 3) Si $(0 \leq t < 0.5)$ alors le chiffre lu est S' .
Sinon le chiffre lu est $S' + 1$.
- IV) Stop.

III.8 CONCLUSION

La structure multicouche n'a pas abouti à un résultat pour cette application. Par contre deux réseaux de structure en fenêtres ont convergé après plus de cent vingt milles itérations. Le taux de reconnaissance des chiffres servants d'exemples d'apprentissage est de 100%. Le deuxième réseau de 256-8-4-1 est testé par un ensemble de chiffres manuscrits. Les résultats et les interprétations, de cette application, seront donnés dans le chapitre suivant.

CHAPITRE 4

RESULTATS ET INTERPRETATIONS

IV.1 INTRODUCTION

Dans ce chapitre, nous discuterons la contrainte qui s'est opposée à la convergence des réseaux multicouches et le réseau en fenêtre (256-16-8-1).

Les deux réseaux qui ont convergé sont testés pour voir :

- s'ils ont bien appris les exemples d'apprentissage,
- si le réseau 256-8-4-1 a la capacité de généralisation.

Aussi, nous donnerons le temps d'apprentissage de chaque réseau, l'erreur quadratique totale et le temps de reconnaissance.

IV.2 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 automatiquement par le réseau lui-même suivant les exemples d'apprentissage.

L'algorithme choisi est la méthode d'optimisation aléatoire modifiée (M.O.A)[3]. Cet algorithme garantit la convergence vers un minimum global avec une probabilité égale à un, tel qu'il a été présenté en [3].

Ce que nous avons constaté au cours de l'apprentissage des différents réseaux est que cet algorithme n'assure la convergence vers un minimum global que si les réseaux ne tombent pas dans un état de saturation, ce qui est le cas des réseaux multicouches et des deux réseaux en fenêtres: 256-16-8-1; 256-8-4-1. L'allure de la descente de l'erreur est donnée dans la figure IV .1.

Etat de saturation

La contrainte majeure que rencontre l'évolution des poids au cours de l'apprentissage est bien la saturation des réseaux entraînés. La fonction d'activation choisie est une fonction sigmoïde (équation IV.1)

$$f(x) = 1 / (1 + \exp(-a \times x)) \quad (\text{IV.1})$$

avec a un réel positif non nul.

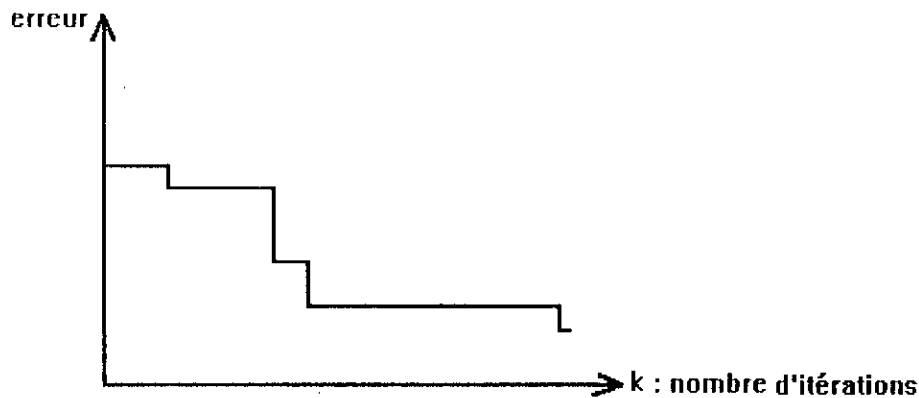


FIGURE IV.1: *descente en escalier de l'erreur quadratique totale*

La variable de cette fonction est une somme pondérée par les poids des connexions. Si au cours de l'apprentissage ces poids prennent des valeurs très élevées, l'exponentielle tendra vers l'infini ou vers zéro.

En poursuivant l'évolution des états des neurones au cours de l'apprentissage, nous avons remarqué que ces neurones se saturent pour les poids $(w_{ij} + \xi_{ij})$ et les poids $(w_{ij} - \xi_{ij})$, et les erreurs qui leurs correspondent sont supérieures à l'erreur correspondante aux poids (w_{ij}) .

L'état de saturation est dû au nombre élevé des connexions, si les poids deviennent importants. Dans ce cas la fonction d'activation va prendre une valeur constante: zéro ou un. C'est le cas des réseaux multicouches et les réseaux en fenêtres: 256-16-8-1; 256-8-4-1.

Généralement pour faire sortir le réseau de l'état de saturation, on diminue le facteur multiplicatif α de la fonction d'activation. Cette solution n'était pas efficace pour les réseaux multicouches, ce-ci nous a poussé à utiliser la structure en fenêtres pour diminuer le nombre de connexions.

Le réseau 256-16-8-1 est tombé dans un état de saturation, par contre, le réseau 256-8-4-1 s'est débloqué de son état de saturation en diminuant le facteur α .

IV.3 RESEAUX CONVERGEANTS

Parmi tous les réseaux entraînés seuls deux réseaux en fenêtres ont convergé vers les erreurs quadratiques voulues.

IV.3.1 Réseau 150-10-5-1

Ce réseau a convergé avec une erreur de $5 \cdot 10^{-4}$ après cent quatre vingt dix huit itérations. Les résultats du tableau IV .1 montrent que le réseau a appris tous les exemples d'apprentissage. Chaque chiffre est représenté par trois exemples différents.

IV.3.2 Réseau 256-8-4-1

Ce réseau a convergé avec une erreur de $1.12 \cdot 10^{-4}$ après cent trente sept milles itérations. Les résultats du tableau IV .2 montrent que le réseau a appris tous les exemples d'apprentissage. Chaque chiffre est représenté par quatre exemples différents.

La valeur de la sortie du réseau est inférieure à un. Pour faire la reconnaissance il faut multiplier cette sortie par dix. L'erreur donnée est la différence entre le chiffre à lire et la valeur de la sortie du réseau multiplier par dix.

IV.4 CAPACITE DE GENERALISATION

La capacité de généralisation est testée seulement pour le réseau 256-8-4-1 par l'ensemble de test, qui est constitué de 40 chiffres manuscrits en trois écritures différentes. Les résultats du test se résume comme suit:

- six chiffres rejetés,
- six chiffres reconnus,
- vingt huit chiffres: erreur de reconnaissance.

Nous constatons que le taux de reconnaissance est faible (15%) pour cet ensemble de test. Pour mieux tester ce réseau et évaluer sa capacité, il faut élargir l'ensemble de test. Si le taux reste toujours faible, il faut continuer l'apprentissage avec un très grand nombre d'exemples pour chaque chiffres avec différentes écritures.

TABLEAU IV .1: résultats de l'apprentissage du réseau 150-10-5-1.

<i>chiffre à lire</i>	<i>sortie du réseau</i>	<i>erreur</i>	<i>chiffre lu</i>
0	0.054853	0.0548530	0
0	0.165879	0.165879	0
0	0.174119	0.174119	0
1	0.828743	0.171257	1
1	1.050955	0.050955	1
1	1.068219	0.068219	1
2	1.987337	0.012663	2
2	2.005362	0.005362	2
2	1.971373	0.028627	2
3	3.00103	0.001030	3
3	2.995576	0.004424	3
3	3.001898	0.001898	3
4	3.999303	0.000697	4
4	3.995644	0.004356	4
4	4.002456	0.002456	4
5	5.012104	0.012104	5
5	4.999605	0.000395	5
5	5.001042	0.001042	5
6	5.998654	0.001346	6
6	5.995497	0.004503	6
6	6.003089	0.003089	6
7	7.007247	0.007247	7
7	6.997076	0.002924	7
7	6.976553	0.023447	7
8	7.981131	0.018869	8
8	7.989831	0.010169	8
8	8.001482	0.001482	8
9	9.000625	0.000625	9
9	9.032480	0.032480	9
9	9.011181	0.011181	9

TABLEAU IV.2: résultats de l'apprentissage du réseau 256-8-4-1 .

<i>chiffre à lire</i>	<i>sortie du réseau</i>	<i>erreur</i>	<i>chiffre lu</i>
0	0.074994	0.074994	0
0	0.080125	0.080125	0
0	0.072841	0.072841	0
0	0.074334	0.074334	0
1	0.999631	0.000369	1
1	1.000925	0.000925	1
1	1.001465	0.001465	1
1	1.000502	0.000502	1
2	1.999593	0.000407	2
2	1.995880	0.000412	2
2	1.997607	0.002393	2
2	1.999818	0.000182	2
3	2.998004	0.001996	3
3	2.997840	0.002160	3
3	3.000350	0.003500	3
4	3.999824	0.000176	4
4	3.999691	0.000309	4
4	3.999365	0.000121	4
4	4.000121	0.000121	4
5	4.992347	0.007653	5
5	4.999753	0.002052	5
5	5.008335	0.008335	5
5	4.997948	0.002052	5
6	6.002421	0.002421	6
6	5.998411	0.001589	6
6	6.001823	0.001823	6
6	6.000044	0.000044	6
7	6.998313	0.001687	7
7	6.999848	0.000152	7
7	7.001381	0.001381	7
7	6.999590	0.000410	7
8	8.001276	0.001276	8
8	8.000302	0.000302	8
8	8.001755	0.001755	8
8	7.996046	0.003954	8
9	9.000514	0.000514	9
9	9.003078	0.003078	9
9	9.002000	0.002000	9
9	8.996542	0.003458	9

IV.5 TEMPS D'APPRENTISSAGE

Le temps d'apprentissage d'un réseau est lié directement au nombre d'itérations, aux dimensions du réseaux et au nombre d'exemples.

Dans le cas du réseau 150 - 10 - 5 - 1 le temps nécessaire à la convergence du réseau (temps d'apprentissage) était : 16 heures, 56 minutes et 27 secondes. Dans le cas du réseau 256 - 8 - 4 - 1 le temps nécessaire à la convergence du réseau était : 14 heures, 47 minutes et 30 secondes. On remarque bien que le temps nécessaire à la convergence du premier réseau est supérieur à celui du second réseau. En effet, aucune loi bien déterminé ne définit la relation entre le temps d'apprentissage et les paramètres auxquels il est lié .

IV.6 TEMPS DE RECONNAISSANCE

Il est évident que le temps de reconnaissance des méthodes connexionistes est plus faible que celui des méthodes statistiques ou culturelles . Il s'agit ici tout simplement de calculer la sortie d'un réseau et de faire trois tests (algorithme de décision de la section III .7). Par contre dans les autres méthodes, la reconnaissance est une série de calcul ou d'extraction de paramètres suivie d'une série de tests.

Donc, dans le cas des réseaux counexionistes, le temps de reconnaissance se réduit au temps de calcul de la sortie: pour les réseaux 256 - 8 - 4 - 1 et 150 - 10 - 5 - 1, le temps de reconnaissance est très faible:

150 - 10 - 5 - 1 : temps de reconnaissance est 0.030 secondes,

256 - 8 - 4 - 1 : temps de reconnaissance est 0.054 secondes.

Il faut bien noter que les temps d'apprentissage et de reconnaissance sont évalués surr un PC 486.

IV.7 LANGAGE ET MOYEN INFORMATIQUE

La programmation a été faite en turbo C⁺⁺. Ce langage nous a permis d'exploiter les quatre Méga octets de RAM disponible sur le PC 486 utilisé. Avec des petites modifications, notre programme a été aussi exécuté sur le vax 750.

IV.8 CONCLUSION

Le choix approprié de la structure du réseau, ainsi que ses dimensions ont permis l'entraînement de deux réseaux et l'atteinte de l'objectif de notre travail. Il faut aussi noter que le temps de reconnaissance est très faible par rapport à d'autres méthodes. L'inconvénient majeur de cette méthode est bien la grande consommation en temps d'apprentissage.

CONCLUSION GENERALE

CONCLUSION GENERALE

Dans cette application, nous avons essayé de simuler, par programmation en langage C sur PC et sur VAX, un réseau capable de reconnaître les chiffres.

L'objectif de cette application est d'entraîner un réseau pour la reconnaissance d'un ensemble d'exemples de chiffres.

Les deux réseaux connexionistes entraînés ont donné un taux de reconnaissance très élevé pour les exemples d'apprentissage. Par contre ce taux est faible (15 %) pour la base de test, formée de quarante chiffres manuscrits. Pour augmenter ce taux, il faut bien continuer l'apprentissage avec un nombre très élevé d'exemples, par chiffres manuscrits, de différentes écritures, pour ne pas donner une représentation trop particulière au réseau. Il est à noter que l'ensemble de tests n'était pas assez large pour conclure sur le taux de reconnaissance des chiffres manuscrits. Mais vu que les exemples d'apprentissage n'ont pas dépassé quarante, le réseau obtenu a une représentation particulière.

L'utilisation de la structure multicouche n'a pas donnée de résultats pour les raisons déjà citées dans le chapitre précédent. Malgré ces résultats, nous ne pouvons pas confirmer que cette structure n'est pas valable pour une telle application, vu que le domaine des réseaux connexionistes est basé sur l'essai.

L'algorithme d'apprentissage utilisé nécessite très peu de paramètres à spécifier et assure la convergence mis à part le problème de saturation. L'inconvénient majeur de cet algorithme est sa grande consommation de temps de calcul.

Par contre, le temps de reconnaissance est très faible, autour de 0.054 secondes. A cet effet, nous pouvons dire que la reconnaissance se fait en temps réel.

Aussi l'utilisation des réseaux connexionistes, mis à part l'avantage d'autoclassification pendant l'apprentissage, économise le temps consacré à l'établissement d'une base de données, et permet la reconnaissance en temps réel. Ce temps peut être encore amélioré, car les réseaux connexionistes doivent être considérés comme un circuit intégré (processeur parallèle) et non pas comme un programme.

En fin, avec des moyens appropriés: scanner, micro assez puissant possédant une RAM supérieure à 4 Mega octets, cette application peut être pousser en plus. C'est à dire entraîner des réseaux qui font la reconnaissance des chiffres à plusieurs niveaux de gris.

BIBLIOGRAPHIE

- [1] S.Abdou et A.Merrouche, *Étude de la méthode des moments pour la reconnaissance des formes appliquée aux caractères arabes*. Projet de fin d'étude, ENP, Alger, 1991.
- [2] F.Aït Boudaoud et M.Belhandouz, *Méthode des arbres pour la reconnaissance des formes appliquée aux caractères de l'alphabet arabe*. Projet de fin d'étude, ENP, Alger, 1992.
- [3] N.BABA, "A new approach for finding the global minimum of error function of neural networks," *Neural networks*, vol.10, pp.367-373, 1989.
- [4] A.Belaïd et Y.Belaïd, *Reconnaissance des formes: méthodes et applications*. Ed . interédition, , 1992.
- [5] F.Blayo, "Réseaux de neurones formels supervisés," *Neuro sciences et sciences de l'ingénieur*, Isère, 1991.
- [6] N.Benaïfa, *Étude de la méthode structurelle pour la reconnaissance des formes appliquée aux caractères arabes manuscrits*. Projet de fin d'étude, ENP, Alger, 1993.
- [7] D.Clyssen, D.Lobstein et J.Zeitoun, *Les nouvelles images.Introduction à l'image informatique*. Ed .dunond, Paris, 1987.
- [8] E.Davalo et P.Naïm, *Des réseaux de neurones*. Ed . Eyrolles, Paris, 1990.
- [9] M.Duranton, "Les applications industrielles des réseaux de neurones III," *Neuro sciences et sciences de l'ingénieur*, Isère, 1991.
- [10] J.A.Freeman et D.M.Skapura, *Neural networks. Algorithms, applications and programmingtechnics*. Ed . Adison Wesley publishing compagny, Houston, 1991.
- [11] M.Kunt, *Traitement numérique des signaux*.Ed . Dunond, Paris, 1981.
- [12] L.Miclet, *Méthodes structurelles pour la reconnaissance des formes*. Ed . Eyrolles, Paris, 1984.
- [13] M.Latrech et L.Saâdaoui, *Traitement et acquisition d'images: application à l'extaction des caractéristiques des images*. Projet de fin d'étude, ENP, Alger, 1987.
- [14] A .Lehr et B.Windrow, "30 years of adaptative neural networks. Perceptron, MADALINE, and Backpropagation," *IEEE*, vol.73, n° 9, pp.1415-1441, sept. 1990.
- [15] L.Pérsonaz, "Réseaux de neurones et reconnaissance des formes," *Neuro scinces et sciences de l'ingénieur*, Isère, 1991.
- [16] A .Poor, "Looking at the TIFF spécifications from the inside," *PC Magazine*, pp.371-376, déc. 1991.

[17] L. Saâdaoui, *Techniques et traitements numériques d'images en vue de la reconnaissance des formes*. Thèse de magister, ENP, Alger, 1991.

[18] D.J. Snouci-Briksi et Y.M. Yeddou, *Etude comparative entre le traitement d'images par les méthodes classiques et par les réseaux de neurones*. Projet de fin d'étude, ENP, Alger, 1994.

ANNEXE 1

ANNEXE 1

BINARISATION A DEUX SEUILS [4]

Soit l'image de la Figure 1, son histogramme est donné par la Figure 2.

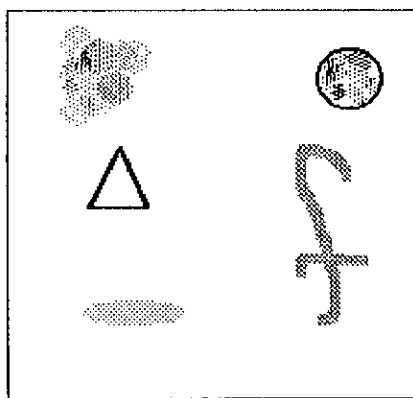


FIGURE 1 : image à binariser

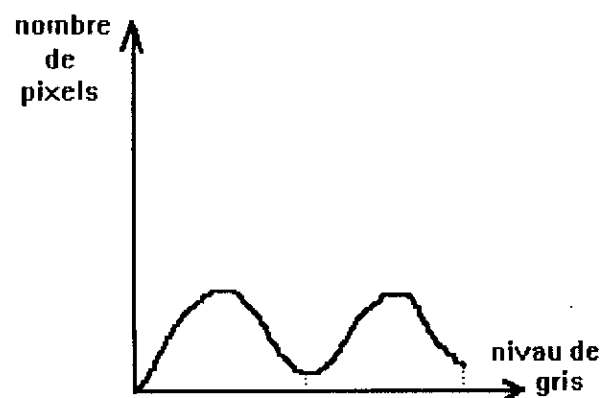


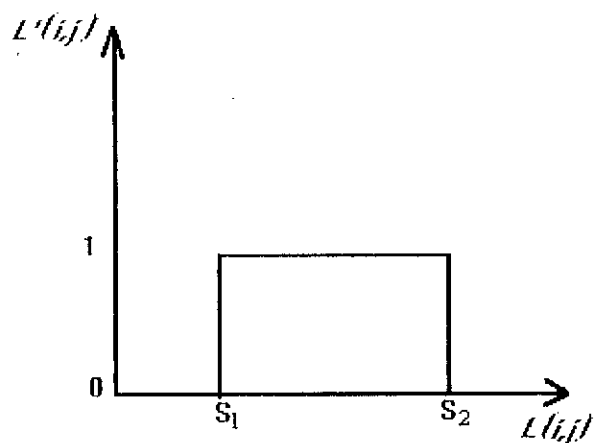
FIGURE 2 : histogramme

L'objet à extraire est localisé entre les deux minimums S_1 et S_2 . La binarisation peut être formulée comme suit:

Soit $L(i, j)$ l'intensité lumineuse et $L'(i, j)$ la nouvelle représentation de l'objet,

$$L'(i, j) = \begin{cases} \text{noir} & \text{pour } S_1 < L(i, j) < S_2, \\ \text{blanc} & \text{pour } L(i, j) > S_2 \text{ et } L(i, j) < S_1. \end{cases}$$

La nouvelle matrice $L'(i, j)$ ne contient que des zéros correspondant au fond, et des uns correspondant à l'objet (Figure 3).

FIGURE 3: *binarisation avec deux seuils*

Dans cette annexe seule la binarisation en utilisant l'histogramme de l'image a été présenté. D'autres méthodes pourront être utilisées.

ANNEXE 2

ANNEXE 2

GENERATION D'UN BRUIT BLANC GAUSSIEN DE MOYENNE b ET DE VARIANCE var [11]:

1^{er} cas : Soit $b=0$

soit le signal $\xi(t)$ suivant une loi normale $N(0, \text{var})$:

$$\xi(t) = \sum_{n=1}^{\infty} (a_n \cos(n\omega t) + b_n \sin(n\omega t)) \quad 1$$

où a_n, b_n sont des variables aléatoires et $\omega = 2\pi/T$ (T période du signal, T tend vers l'infini).

on écrit $T(1)$ sous la forme:

$$\xi(t) = \sum_{n=1}^{\infty} [a_n \cos(n\omega - \omega_c + \omega_c)t + b_n \sin(n\omega - \omega_c + \omega_c)t] \quad 2$$

Avec ω_c quelconque on a:

$$\xi(t) = x(t) \cos(\omega_c t) - y(t) \sin(\omega_c t) \quad 3$$

$$x(t) = \sum_{n=1}^{\infty} [a_n \cos(n\omega - \omega_c)t + b_n \sin(n\omega - \omega_c)t] \quad 4$$

$$y(t) = \sum_{n=1}^m [a_n \cos(n\omega - \omega_c)t - b_n \sin(n\omega - \omega_c)t] \quad 5$$

par application du théorème central limité, x et y suivent une loi normal $N(0, \text{var})$.

En posant:

$$x = R \cos \theta \quad \text{et} \quad y = R \sin \theta \quad 6$$

Donc, on obtient l'expression suivante:

$$\xi(t) = R(t) \cos(\omega_c t + \theta(t)) \quad 7$$

$R(t)$ et $\theta(t)$ sont des variables aléatoires qui suivent respectivement une loi de RAYEIGH et une loi uniforme [11].

$$f(r) = \left(\frac{r}{\text{var}}\right) \exp\left(-\frac{r^2}{2\text{var}}\right) \quad r \text{ positif}$$

$$g(\theta) = \frac{1}{2\pi} \quad 0 \leq \theta \leq 2\pi$$

8

La relation liant R et θ est donnée par:

$$R(t) = \sqrt{2\text{var} \ln \frac{2\pi}{\theta(t)}}$$

9

En posant $w_c=0$, on a enfin:

$$\xi(t) = R(t) \cos(2\pi u(t)) \quad \text{avec} \quad u(t) = \frac{\theta(t)}{2\pi}$$

10

$u(t)$ suit bien une loi uniforme sur $[0,1]$.

On peut alors générer $x(t)$ à chaque instant connaissant $u(t)$.

Génération de $u(t)$:

La suite $U_i(t)$ définie par :

$$U_i(t) = \frac{U(t)}{2^n - 1}$$

11

Où $U(t+1) = k U(t) [2^n - 1]$, $U(0) = U_0$ impair suit une distribution uniforme sur $[0,1]$ pour $k=8m+3$ proche de $2^{(n-1)/2}$.

En prenant $n=16$ (nombre de bit du u_p), on obtient $m=16$ et $k=131$, on choisit $U_0 = 12357$.

L'algorithme de génération récursive de $x(t)$ se présente alors comme suit :

$$t=0 \quad U_0 = 12357$$

tant que $t < t_{max}$

faire $U(t) = (131 \times U(t)) \bmod (2^{15})$, $U_i(t) = U(t) / (2^n - 1)$

$$R(t) = \sqrt{2 \ln\left(\frac{1}{U_{L,t}}\right) \text{var}}$$

$$\xi(t) = R(t) \cos(2\pi U_L(t))$$

$$t = t + 1$$

stop

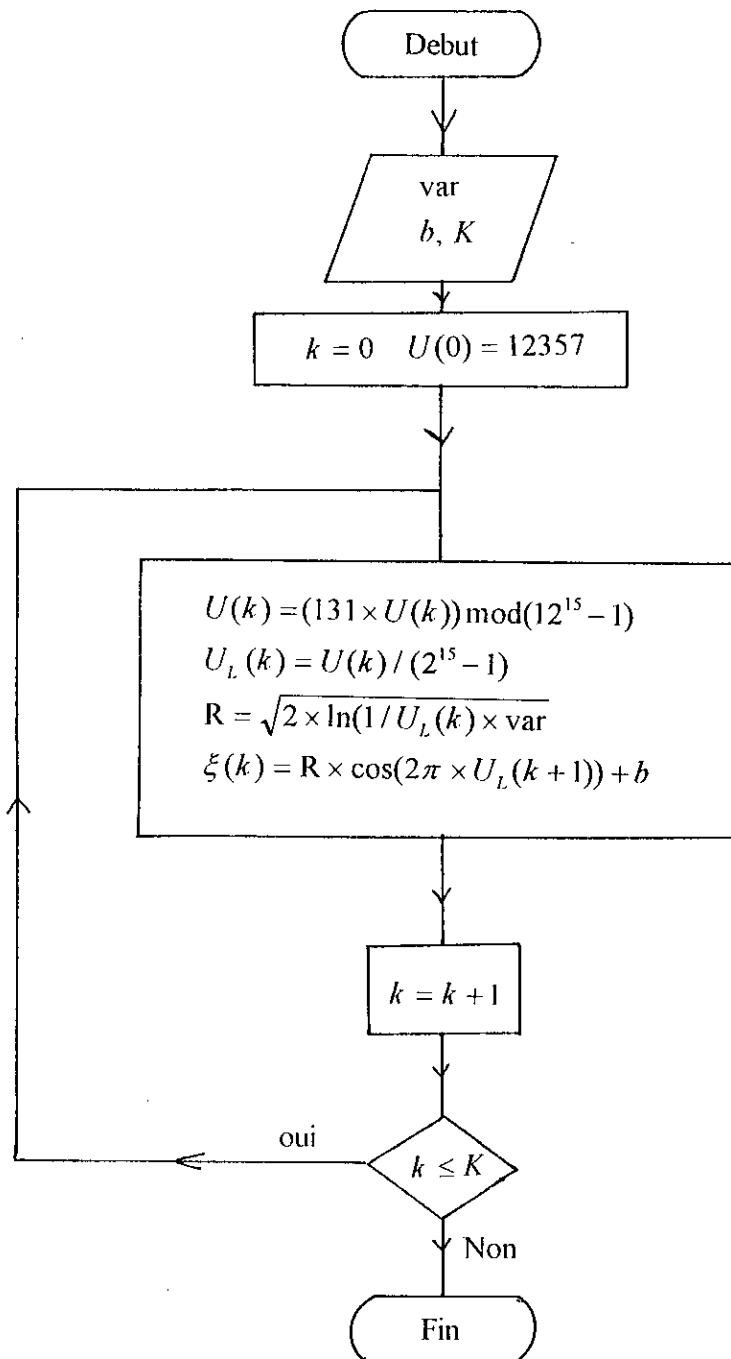
12

2^{ème} cas : Dans le cas où b est différent de 0, il suffit de l'ajouter à $x(t)$:

$$\xi(t) = R(t) \cos(2\pi u_i(t)) + b$$

13

Organigramme:



ANNEXE 3

ANNEXE 3

SPECIFICATIONS DU FORMAT TIFF UTILISE

SIGNIFICATION DE CHAQUE DONNEE LUE DANS LES CHAMPS DES FICHER TIFF DU LOGICIEL PILOTANT LE « HP SAN II P »

Les treize données lues sont :

Champ 1 = 254 - 4 - 1 - 0 :

4 = long,

1 = une donnée,

0 = fichier nouveau une seule image est sauvegardée.

Champ 2 = 256 - 3 - 1 - 10 :

3 = short,

1 = une donnée,

10 = 10 pixels en largeur.

Champ 3 = 257 - 3 - 1 - 15 :

3 = short,

1 = une donnée,

15 = 15 pixels.

Champ 4 = 258 - 3 - 1 - 1 :

3 = short,

1 = une donnée,

1 = un bit pour chaque pixel.

Champ 5 = 259 - 3 - 1 - 1 :

- 3 = short,
- 1 = une donnée,
- 1 = absence de compression.

Champ 6 = 262 - 3 - 1 - 1 :

- 3 = short,
- 1 = une donnée,
- 1 = le zéro est un noir et le un est un blanc.

Champ 7 = 273 - 4 - 2 - 186 :

- 4 = long,
- 2 = deux données,
- 186 = un pointeur vers l'offset.

Champ 8 = 277 - 3 - 1 - 1 :

- 3 = short,
- 1 = une donnée,
- 1 = image noir et blanc.

Champ 9 = 278 - 4 - 1 - 8 :

- 4 = long,
- 1 = une donnée,
- 8 = huit rangs par bande.

Champ 10 = 279 - 4 - 1 - 194

- 4 = long,
- 1 = une donnée,
- 194 = 194 byte par bande.

Champ 11 = 282 - 5 - 1 - 170 :

5 = rationnal,

1 = une donnée,

170 = 170 dpi (résolution horizontale).

Champ 12 = 283 - 5 - 1 - 178 :

5 = Rationnal,

1 = une donnée,

178 = 178 dpi (résolution verticale).

Champ 13 = 296 - 3 - 1 - 2 :

3 = short,

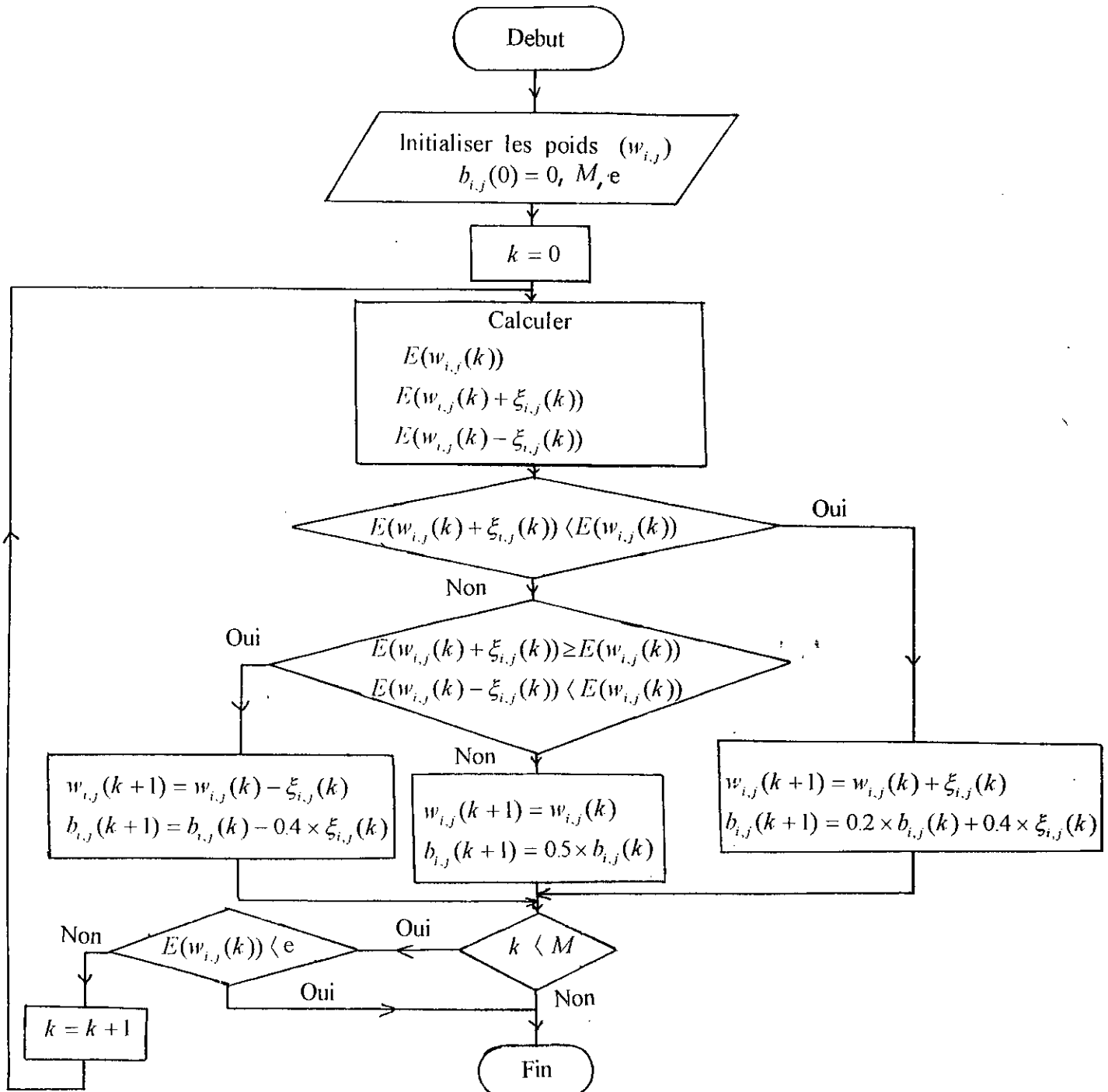
1 = une donnée,

2 = unité « dpi ».

ANNEXE 4

ANNEXE 4

ORGANIGRAMME DE LA METHODE M.O.A



ANNEXE 5

0

1

0

1

0

1

Chiffres générés sur PC utilisés pour l'apprentissage du réseau 150-10-5-1.

2

3

2

3

2

3

Chiffres générés sur PC utilisés pour l'apprentissage du réseau 150-10-5-1.



Chiffres générés sur PC utilisés pour l'apprentissage du réseau 150-10-5-1.
Chiffres générés sur PC utilisés pour l'apprentissage du réseau 150-10-5-1.

4

4

4

4

5

5

5

5

Chiffres générés sur PC utisés pour l'apprentissage du réseau 256-8-4-1.

8

8

8

8

9

9

9

9

Chiffres générés sur PC utilisés pour l'apprentissage du réseau 256-8-4-1.

ANNEXE 6

0

1

2

3

4

5

6

7

8

9

Chiffres manuscrits utilisés pour tester le réseau 256-8-4-1.

0

1

2

3

4

5

6

7

8

9

Chiffres manuscrits utilisés pour tester le réseau 256-8-4-1.

0

1

2

3

4

5

6

7

8

9

Chiffres manuscrits utilisés pour tester le réseau 256-8-4-1.

0

1

2

3

4

5

6

7

8

9

Chiffres manuscrits utilisés pour tester le réseau 256-8-4-1.

0

1

2

3

4

5

6

7

8

9

Chiffres manuscrits utilisés pour tester le réseau 256-8-4-1.