

République Algérienne Démocratique et Populaire  
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR

ÉCOLE NATIONALE POLYTECHNIQUE  
DÉPARTEMENT GÉNIE ÉLECTRIQUE  
SPÉCIALITÉ AUTOMATIQUE

**MEMOIRE DE FIN D'ETUDES**

*En vue de l'obtention du diplôme  
d'Ingénieur d'Etat en Automatique*

**THEME**

**RECONNAISSANCE DE FORMES  
PAR RESEAUX DE NEURONES  
APPLICATION : SIGNATURES**

Présenté par :  
Mr MAZOUZ.Abdelsamed  
Mr BOUAOULI.Nabil

Dirigé par :  
Mr SOUAMI

*Année 1995 - 1996*



# REMERCIEMENTS

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

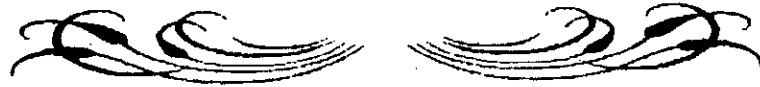
Au terme de cette étude, il nous est agréable d'adresser nos vifs remerciements à notre promoteur Mr M.C. SOUAMI pour sa précieuse assistance et ses conseils.

Nous remercions aussi Mr F. BOUDJEMAA pour son aide.

Que Mr LAKAF MADJID et SEDDOUD NABIL trouvent ici notre marque de reconnaissance pour leurs précieuses suggestions.

Et à tous ceux qui ont contribué de prêt ou de loin à l'élaboration de ce travail, merci.

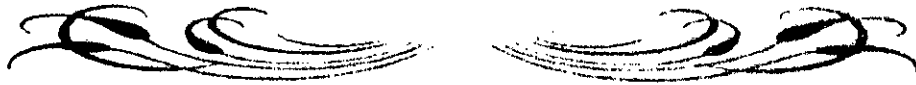
## DEDICACE



*À la mémoire de notre cher et regretté ami et  
frère que Dieu l'accueil en son vaste paradis*

. EL MEZOUAR MISSOUM,

# DEDICACE



المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique

*Je dédie ce travail :*

*A mes très chers parents,*

*A mes frères et mes soeurs,*

*A ma chère grand mère et ma tante,*

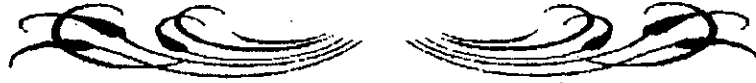
*Aux familles AMAROUCHENE et  
BERRAMOUNE.*

*Aux membres de "SABIR" et à  
DJAMALDINE.*

**NABIL**

# DEDICACE

المدرسة الوطنية المتعددة التقنيات  
المكتبة — BIBLIOTHEQUE  
Ecole Nationale Polytechnique



*A mes très chers parents, Merci pour tout.*

*A mes frères et soeurs*

*A mes belles soeurs (Fouzia et Samia) et mon beau frère  
(Karim).*

*A mes neveux et à mes nièces et au dernier né de la famille  
Merqam.*

*A ma très chère tante Daouia.*

*A ma chère Souhila.*

*Ainsi qu'à mes amis (Nino, Samir, Lamouri, Youcef,*

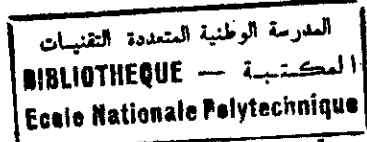
*Tarik, Fouad, etc...) et amies (Nabiha, Fayza, Asma,  
etc...)*

*A tous je dédie ce modeste travail.*

ABDELSAMED

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# SOMMAIRE



INTRODUCTION GENERALE:.....	1
CHAPITRE 1 :: IMAGERIE	
<b>1. INTRODUCTION :</b> -----	<b>4</b>
<b>2. DEFINITIONS :</b> -----	<b>5</b>
A. IMAGE ANALOGIQUE :-----	5
B. PIXEL :-----	5
C. IMAGE NUMERIQUE BINAIRE :-----	6
D. IMAGE NUMERIQUE MULTI NIVEAUX :-----	6
E. BRUIT :-----	6
F. HISTOGRAMME :-----	6
G. BINARISATION :-----	7
H. FILTRAGE :-----	8
I. SUIVI DE CONTOUR:-----	8
J. AMELIORATION DE CONTRASTE:-----	9
K. RECADRAGE DYNAMIQUE-----	10
<b>3. INTRODUCTION :</b> -----	<b>12</b>
<b>4. ACQUISITION :</b> -----	<b>04</b>
A. LES CAMERAS LINEAIRES :-----	12
B. LES CAPTEURS MATRICIELS :-----	13
C. LES SCANNERS :-----	13
<b>5. PRE-TRAITEMENT :</b> -----	<b>14</b>
<b>6. RESTITUTION D'IMAGE :</b> -----	<b>14</b>
<b>7. STOCKAGE :</b> -----	<b>14</b>



## CHAPITRE II : RECONNAISSANCE DE FORMES

<b>1. GENERALITES:</b>	17
<b>2. INTRODUCTION</b>	19
A. HISTORIQUE	19
<b>3. LES DEUX GRANDES CONTROVERSEES:</b>	21
A. CONTROVERSE COGNITIVE / COMPORTEMENTALISTE	21
B. CONTROVERSE APPRENTISSAGE / REPRESENTATION	22
<b>4. RECONNAISSANCE DE L'ECRIURE:</b>	24
<b>5. LES DIFFERENTES METHODES DE CLASSIFICATION:</b>	24

## CHAPITRE III : RESEAUX DE NEURONES

<b>1. INTRODUCTION:</b>	27
<b>2. LES NEURONES VIVANTS:</b>	28
A. LE TRAITEMENT DE L'INFORMATION PAR LES DENDRITES :	32
B. LE TRAITEMENT DE L'INFORMATION PAR L'AXONE :	35
<b>3. LES NEURONES FORMELS:</b>	36
A. UNE CARICATURE: L'AUTOMATE A SEUIL	36
<b>4. LE RESEAU DE NEURONES:</b>	38
A. ARCHITECTURE D'UN RESEAU NEUROMIMETIQUE	40
B. LE CHOIX DES POIDS SYNAPTIQUES :	42
C. LES ENTREES ET LES SORTIES DU RESEAU :	42
D. APPLICATIONS DES RESEAUX NEUROMIMETIQUES :	43
E. UN MODELE PLUS GENERAL :	45
F. LA FONCTION D'ACTIVATION D'UN NEURONE FORMEL :	48
1. <i>Caractéristiques des fonctions d'activation :</i>	48
2. <i>Exemples de fonctions d'activation :</i>	50
G. APPRENTISSAGE :	51
1. <i>La procédure d'apprentissage :</i>	51

2. Les différents types d'apprentissage : -----	52
1( Apprentissage supervisé-----	52
2( Apprentissage non supervisé-----	52

#### CHAPITRE IV : L'ALGORITHME DE RETROPROPAGATION DU GRADIENT.

1. INTRODUCTION -----	54
2. REGLE DE WIDROW-HOFF -----	55
3. ASPECTS PRATIQUES DE L'ALGORITHME -----	61
A. LISSAGE DE LA REGLE D'ADAPTATION : -----	61
B. VALEURS INITIALES DES POIDS:-----	61
C. CHOIX DU PAS D'ADAPTATION: -----	62
D. TEST D'ARRET-----	62
E. CHOIX DU RESEAU:-----	63

#### CHAPITRE V : RESEAU A ARCHITECTURE EVOLUTIVE.

1. INTRODUCTION -----	65
2. ALGORITHME A PLACEMENT DE CELLULES (RCE ALGORITHM)-	66
A. ARCHITECTURE DU RESEAU : -----	66
B. APPRENTISSAGE : -----	67
C. PARTICULARITE DE L'ALGORITHME : -----	71
1. INFLUENCE DE L'ORDRE DES PROTOTYPES APPRIS :-----	72

#### CHAPITRE VI: APPLICATIONS, RESULTATS ET COMMENTAIRES.

1. INTRODUCTION :-----	74
2. POURQUOI LES MOMENTS INVARIANTS :-----	74
3. CHOIX DES BASES D'APPRENTISSAGE ET DE TEST : -----	76
4. UTILISATION DE LA METHODE DE BACK-PROPAGATION-----	78
A. DESCRIPTION DU RESEAU-----	78

	المدرسة الوطنية المتعددة التقنيات <b>BIBLIOTHEQUE — المكتبة</b> <b>Ecole Nationale Polytechnique</b>	
B. CHOIX DES ENTREES -----		78
C. APPRENTISSAGE ET RESULTATS -----		79
1. Résultats avec les entrées normalisées $\phi'_i$ : -----		79
1( Résultats des tests:-----		79
D. INTERPRETATION DES RESULTATS -----		82
E. LES RESULTATS DES TESTS -----		84
<b>5. INTERPRETATION DES RESULTATS -----</b>		<b>87</b>
A. APPRENTISSAGE DU RCE-----		88
B. RESULTATS ET COMMENTAIRE:-----		89
CONCLUSION GENERALE.....		92
BIBLIOGRAPHIE		
ANNEXE -A-		

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

# INTRODUCTION GENERALE

## Introduction générale

Les défis que se lance l'homme, depuis la nuit des temps, pour satisfaire ses désirs, stimulés par la simple curiosité, à copier la nature ne cessent de le guider vers de nouvelles conquêtes de la science. L'histoire a vu l'évolution de l'intelligence humaine depuis l'époque où il utilisait les pierres pour fabriquer des armes et des outils de travail jusqu'à notre époque où le voyage dans l'espace relève du trivial et où l'homme cette fois essaye de se copier en imaginant, élaborant et perfectionnant des systèmes (machines) intelligents. Grâce aux développements que connaît la médecine dans le domaine de la neurobiologie, les chercheurs ont réussi à donner naissance à une branche de la science appelée intelligence artificielle.

En essayant de l'appliquer dans diverses technologies de pointe, ils ont réalisé rapidement l'utilité et l'intérêt à porter envers elle.

C'est pendant les années 70 que des chercheurs comme *Andersson*, *Kohonen* et bien d'autres ont travaillé sur les réseaux de neurones, apprenant progressivement à les formaliser en combinant les méthodes de l'algèbre linéaires et celles des techniques non linéaires. Il est vrai que le brio des théories de *Hopfield*, épaulé par un engouement des physiciens à partir de 1982, a été le moteur d'un éveil mondial sur les neurones artificiels.

### *Pourquoi des neurones artificiels?*

Capter une image, la numériser, la segmenter en éléments de contours, détecter un objet mobile, le reconnaître quelle que soit sa position et estimer sa profondeur. Capter le son d'une voix au milieu d'un bruit ambiant et reconnaître les mots qui sont prononcés. Capter et analyser la multitude des signaux issus d'un processus industriel et en déduire si tout est conforme ou si une avarie se prépare. Voici quelques problèmes pourtant courants dans les sciences de l'ingénieur mais dont les solutions, encore incomplètes, impliquent de multiples efforts de recherche dans la communauté scientifique.

Malgré la constante augmentation de puissance des calculateurs, malgré les approches théoriques de plus en plus sophistiquées, ils n'arrivent pas à exécuter des tâches relevant du traitement, en temps réel, de très grands flots de données souvent multidimensionnelles.

Or, chez les êtres vivants, un visage est reconnu en quelques dixièmes de seconde, une voix est reconnue au milieu d'un bruit ambiant et le discours est perçu. Pour une machine, réaliser de telles tâches serait une véritable prouesse, et nous, nous les réalisons sans même y penser; c'est là que réside l'attrait des neurones artificiels.

Dans ce présent travail, on va essayer d'utiliser des réseaux de neurones pour tenter de reconnaître les signatures manuscrites.

Le travail est organisé en six chapitres. Dans les deux premiers, on trouvera les informations essentielles sur la notion d'image et les techniques de reconnaissance de formes. Le troisième sera le sujet d'une revue générale sur les réseaux neuromimétiques. Deux philosophies différentes concernant ces derniers seront abordés aux chapitres 4 et 5, et on verra leurs applications dans le sixième chapitre.

**IMAGERIE**

---

**CHAPITRE I****IMAGERIE****1. INTRODUCTION :[5]**

Avec les sons, les images constituent l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec ses semblables. Les efforts récents élargissent même cette communication aux machines. Le traitement d'image est l'ensemble des méthodes qui cherchent à rendre cette opération possible, plus simple, plus agréable et plus efficace, c'est une aide importante par l'homme qui y trouve, en plus, une motivation dans la satisfaction de ces curiosités et des besoins qui en découlent.

Les premiers traitements d'images par moyens techniques remontent au 18<sup>ème</sup> siècle, à l'époque des pionniers de la photographie. La comparaison entre les premiers résultats de noircissement du nitrate d'argent par la lumière et ceux que l'on obtient aujourd'hui avec les films et les appareils commercialisables montre le progrès accompli.

L'électronique a fait son entrée dans le domaine de l'imagerie vers la fin du 19<sup>ème</sup> siècle grâce à *Nipkow*, qui a permis en 1884 la transcription d'une image en signal électrique grâce à son disque [N PR].

D'améliorations en perfectionnements, on est arrivé à la télévision vers 1941. Les techniques photographiques, optiques et électroniques analogiques constituent ce qui a été appelé "traitement analogique d'images".



## 2. DEFINITIONS :[3] [9]

### a. IMAGE ANALOGIQUE :

On peut associer à une scène visuelle, ou à son image dans le plan focale d'un système optique, une fonction à quatre variables  $f(x,y,t,\lambda)$ . Cette fonction décrit l'énergie lumineuse présentée en un point de coordonnées  $(x,y)$  à un instant  $t$  et sur la longueur d'onde  $\lambda$ .

Dans la plupart des cas la représentation de l'image se fait plus simplement en considérant :

- Qu'il n'y a pas d'évolution temporelle des grandeurs étudiées et des caractéristiques et que la longueur d'onde est unique dans ces conditions, la grandeur  $f(x,y,t,\lambda)$  s'écrit  $f(x,y)$  possédant les propriétés suivantes :

$$f(x,y) = \begin{cases} \geq 0 & \text{dans le domaine } D \\ = 0 & \text{ailleurs.} \end{cases}$$

$D$  est un rectangle de dimension  $I_x$  et  $I_y$ .

La fonction  $f(x,y)$  associée à l'image physique en un point donné a une valeur qu'on appelle intensité de l'image en ce point.

### b. PIXEL :

Le pixel correspond au rayonnement réfléchi par cet élément d'image, il constitue le plus petit élément de l'image et possède un niveau de gris.

### **c. IMAGE NUMERIQUE BINAIRE :**

On appelle image binaire une image pour laquelle les éléments de la matrice  $f(x,y)$  ont pour valeur  $I_1$  ou  $I_2$  tels que  $(0 \leq I_1 \leq 63$  et  $0 \leq I_2 \leq 63)$ . Elle est souvent utilisée en Reconnaissance de forme.

### **d. IMAGE NUMERIQUE MULTI NIVEAUX :**

Une image numérique est une matrice  $f(x,y)$  à deux dimensions dont les éléments représentent l'intensité lumineuse d'un faisceau de lumière mono-chromatique.

La valeur d'un élément  $f(x,y)$  représente le niveau de gris du point de coordonnées  $(x,y)$ . Cette valeur peut prendre 64 niveaux (de 0 à 63).

0 : correspond au blanc

63 : correspond au noir

### **e. BRUIT :**

Le bruit est caractérisé par des pixels isolés dans des régions homogènes. Ce dernier affecte de brusques variations de niveaux de gris.

### **f. HISTOGRAMME :**

Il nous permet de connaître la fréquence d'apparition de chaque niveau de gris dans l'image les niveaux de gris de 0 à 63 sont représentés en abscisse, le nombre de pixels affectés par chaque niveau de gris est affiché en ordonnée.

L'histogramme nous permet aussi de donner un grand nombre d'informations sur la distribution des niveaux de gris dans le cas d'une image trop claire ou trop foncée comme l'illustre la (fig 1.1).

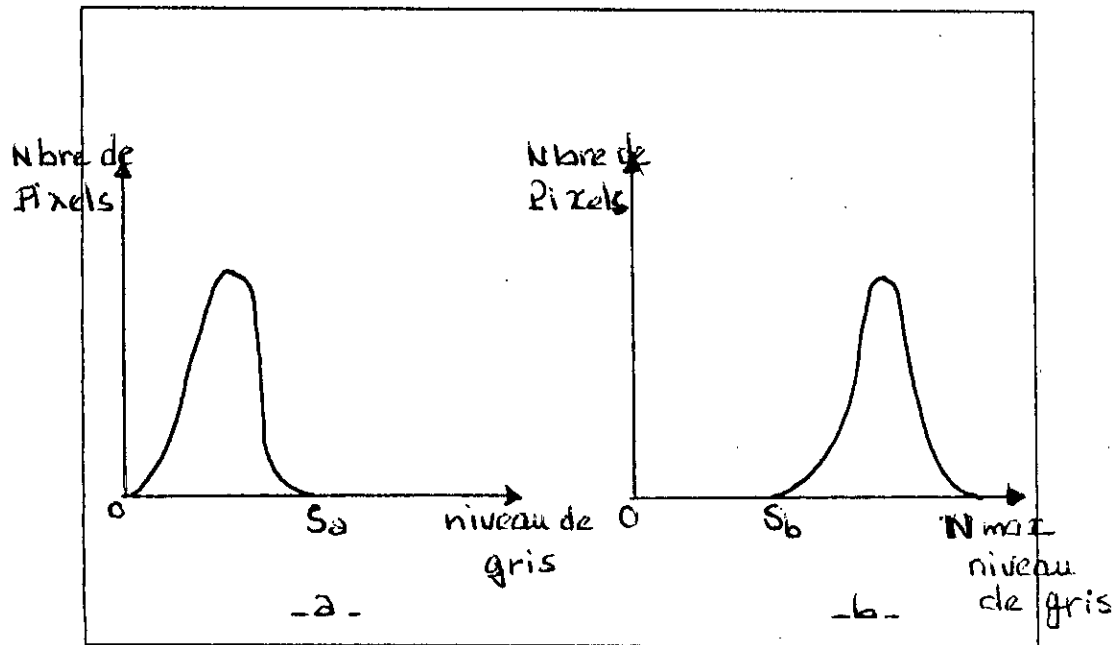


fig 1.1

### g. BINARISATION :

la binarisation d'une image présente plusieurs avantages l'un des plus importants est certainement le faible espace mémoire nécessité par une image binaire et la simplicité des opérateurs qui lui sont associés. En effet, l'image ne possédant que deux niveaux: noir et blanc; il suffit d'un bit pour coder un pixel, ce qui donne d'une taille mémoire, pour l'image, six fois moins élevée qu'une image codée sur 64 niveaux. La binarisation peut être aussi considérée comme la première étape d'isolement des objets par rapport au fond.

La binarisation (ou le seuillage) peut être formulée de la manière suivante: Soient  $f(x,y)$  l'intensité lumineuse (niveau de gris) et  $f'(x,y)$  la nouvelle intensité lumineuse alors :

$$f(x,y) = \begin{cases} \text{Noir si } f(x,y) > \zeta \\ \text{Blanc si } f(x,y) \leq \zeta \end{cases}$$

ou  $\zeta$  est la valeur du seuil choisi, comme l'illustre la fig 1.2.

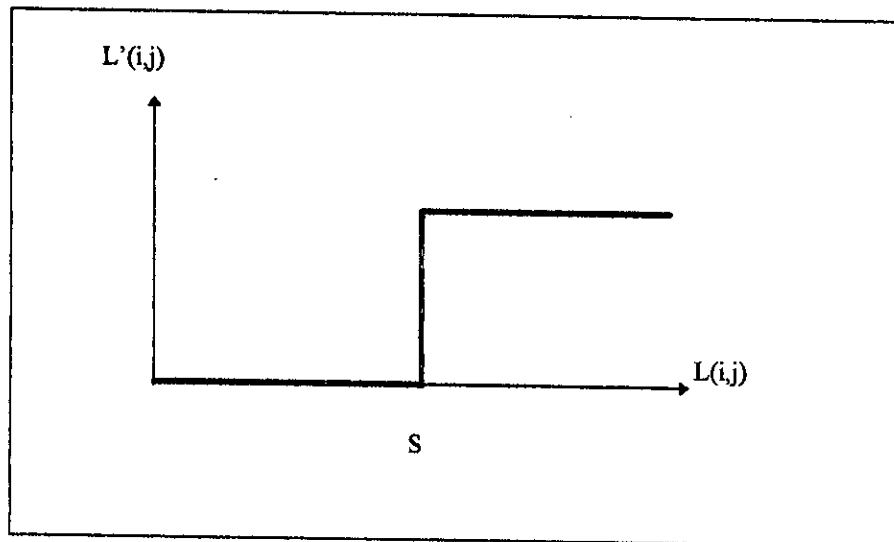


fig 1.2

### **h. FILTRAGE :**

Nous avons défini le bruit dans une image comme étant un phénomène de variation brusque d'un pixel isolé par rapport à ses voisins. Le filtrage nous permet alors d'éliminer les effets du bruit.

C'est une transformation localisée qui détermine la nouvelle valeur d'un point de l'image de coordonnées  $(x,y)$  en fonction de ses points voisins de l'image originale.

Les points voisins sont contenus dans une fenêtre rectangulaire de dimension  $M \times N$  construite autour du point considéré. L'opérateur correspondant est un masque de même dimension.

### **i. SUIVI DE CONTOUR:**

Le contour d'objet est la limite séparant deux régions d'intensité lumineuse différentes, c'est à dire les discontinuités ou les variations brusque de l'intensité lumineuse.

Les formes seront essentiellement définies à l'aide de leurs frontières par exemple on va parler de l'Algorithme de *CHOTTERA* et *SHRIDHAR*. Il est capable de

détecter les contours de tous les éléments de l'image. Notons donc un pixel frontière est un point image "blanc" qui a au moins un voisin "noir".

- balayer l'image jusqu'à la rencontre du premier point contour  $(x_1, y_1)$ , le point précédent est noté  $(x_2, y_2)$ .

- A partir du point  $(x_2, y_2)$ , en tournant autour du point  $(x_1, y_1)$  de façon à balayer ses voisins dans le sens horaire.

- Evaluer les coordonnées du  $K^{\text{eme}}$  voisin.

- Si le  $K^{\text{eme}}$  voisin est un point "noir", il sera le point suivant du contour et on définit  $(x_1, y_1)$  en ce point;  $(x_3, y_3)$  sera le  $(K-1)^{\text{eme}}$  point du voisinage de  $(x_1, y_1)$ , aller à (2).

- Si le  $K^{\text{eme}}$  voisin est un point "blanc", prendre  $K=K+1$  et aller à (3).

Continuer jusqu'à la rencontre du premier point contour détecté.

On peut avoir perte d'une partie du contour de l'objet (discontinuité du contour), il convient à ce stade de définir correctement l'amélioration de contraste.

### **J. AMELIORATION DE CONTRASTE:**

Elle consiste à améliorer l'aspect visuel d'une image donc transformer le niveau de gris du pixel  $f(x,y)$  en un niveau de  $f'(x,y)$ , c'est à dire que la transformation du niveau de gris d'un pixel est indépendante des pixels voisins.

Si  $S_1$  et  $S_2$  deux seuils différents de  $f(x,y)$  l'intensité lumineuse d'un point de l'image de coordonnées  $(x,y)$ .

$$f(x,y) < S_1 \quad f'(x,y) = K_1 X f(x,y)$$

$$S_1 < f(x,y) < S_2 \quad f'(x,y) = K_2 f(x,y) + (k_1 - k_2) S_1.$$

$$f(x,y) > S_2 \quad f'(x,y) = K_3 f(x,y) + (k_1 - k_2) S_1 + (k_2 - k_3) S_2.?$$

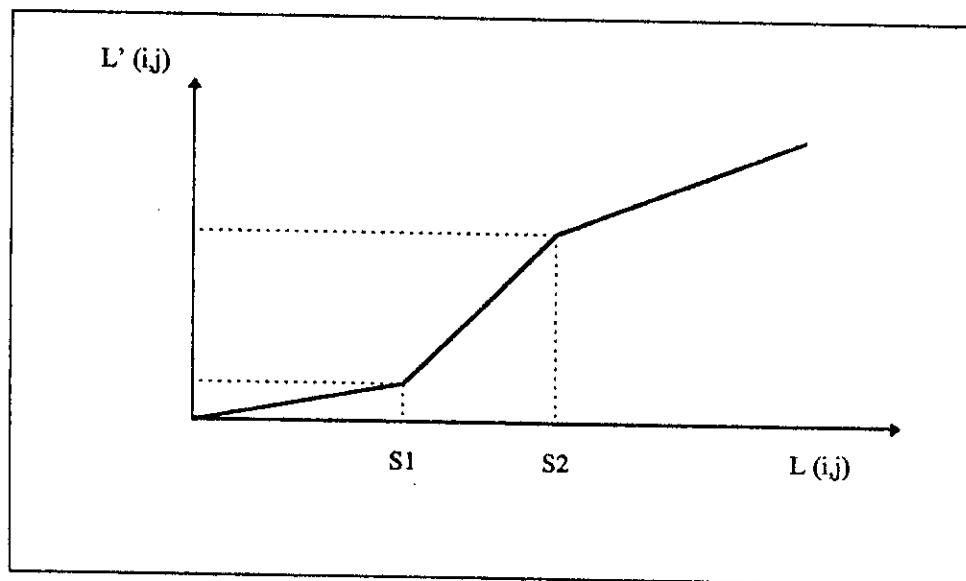


fig : Amélioration de contraste.

$K_1$  ,  $K_2$  et  $K_3$  sont des constantes avec:

$K_1 < 1$  ;  $K_2 > 1$  et  $K_3 < 1$ .

Deux cas sont à envisager:

\*  $k_1 = k_3 = 0$  et  $k_2 > 1$  appelé recadrage de dynamique.

\*  $k_1 = k_3 = 0$  ;  $k_2 \rightarrow \infty$  et  $S_1 = S_2$ . appelé seuillage.

### **k. RECADRAGE DYNAMIQUE**

Les images sont initialement trop claires, trop foncés ou bien peu contrasté, cela est dû au fait que les niveaux de gris sont regroupés vers le bas de l'échelle, vers le haut ou bien dans un intervalle étroit très visible sur l'histogramme, le but est alors de redistribuer le niveau de gris de l'image afin de leur faire occuper toute la bande de valeur possibles.

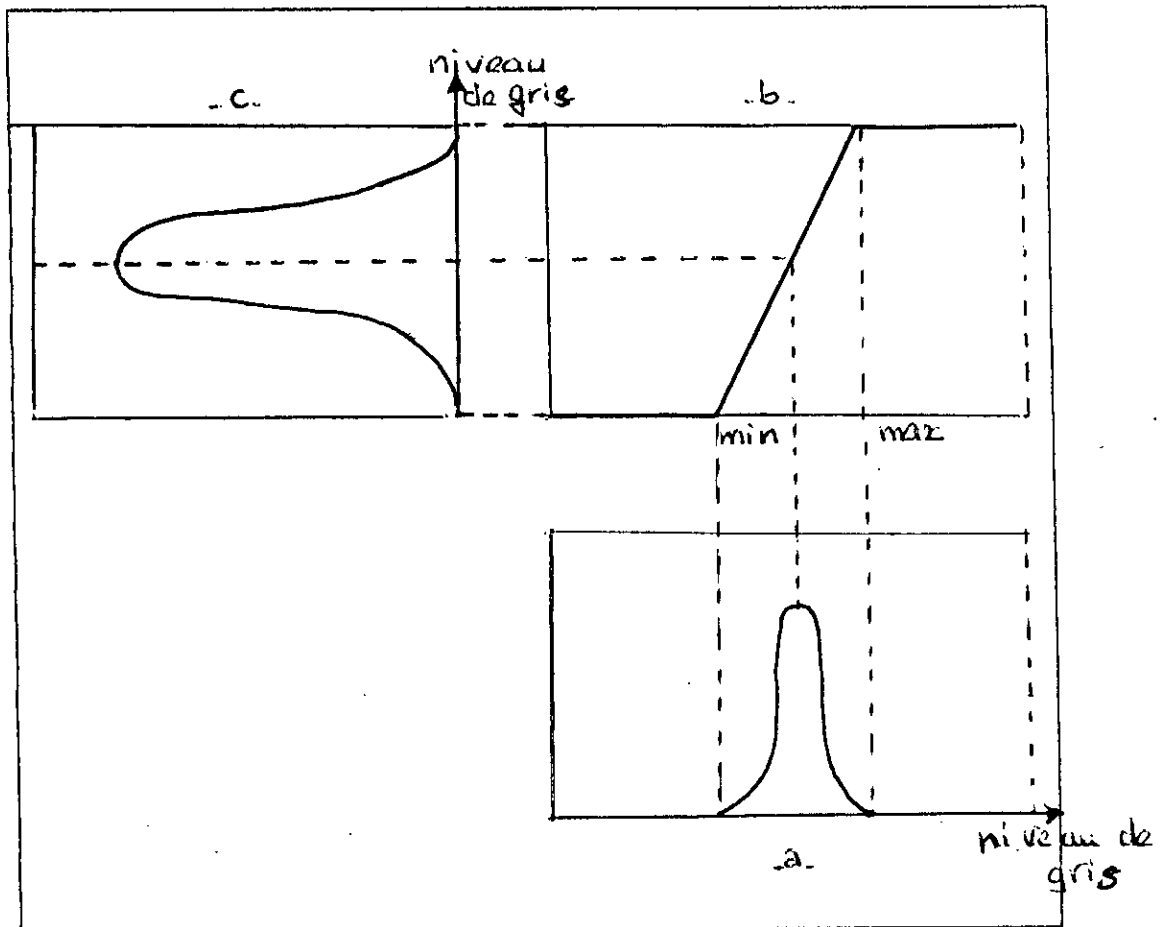


fig: Principe de recadrage de dynamique.

- a- Histogramme de l'image originale
- b- Modification de contraste avec saturation.
- c- Histogramme recadré.

---

## AQUISITION ET RESTITUTION [3] [5]

### 3. INTRODUCTION :

Tout système de traitement d'images peut être considéré comme succession de trois étapes : acquisition, traitement, affichage.

Ces étapes sont communes aussi à tous les systèmes de traitement de l'information.

On se rend compte aujourd'hui qu'il est absolument nécessaire de tenir compte de la nature des données images dans la conception des systèmes d'acquisition.

### 4. ACQUISITION :

Avant d'analyser les différents traitements envisagés pour le traitement des données nous ferons quelques remarques sur l'acquisition de ces données.

Dans ces conditions les systèmes d'acquisition actuels les plus courants sont donc des scanners, des caméras linéaires ou des capteurs matriciels.

#### a. *LES CAMERAS LINEAIRES :*

Contiennent des capteurs linéaires qui eux mêmes sont composés d'une rangée d'éléments photo sensibles, de deux registres analogiques et d'un amplificateur de sortie. Leur excellente résolution (de 256 à 2048 éléments par ligne) leurs faible bruit et leurs petites dimensions leurs ont longtemps gagné la faveur des utilisateurs.



**b. LES CAPTEURS MATRICIELS :**

Sont analogues aux capteurs linéaires à la différence près que les photos-éléments sont organisés en matrices et que les registres de transport sont disposés entre les colonnes des photos-éléments.

**c. LES SCANNERS :**

Sont les plus utilisés. Le terme répandu d'abord dans le domaine médical, est aujourd'hui utilisé pour désigner des produits fort dissemblables. Le mot qui vient de l'anglais "to scan", (balayer), signifie le balayage d'un document (image ou texte) par un système optique. L'opération est rendue possible soit par le déplacement du document lui-même soit par celui du faisceau de lumière.

S'il s'agit d'application Bureautique de P.A.O, les scanners actuels ont une résolution qui est au minimum de 300 dpi (dpi - dot per inch), c'est à dire le nombre de points par pouce, (1 inch = 2,54cm) et peut atteindre couramment 800 dpi. Ils permettent en outre la saisie d'images en niveau de gris (16 au 256 niveaux) voire en couleur.

Toute fois les scanners sont relativement lents il faut souvent des temps de calcul de l'ordre de la minute pour faire une acquisition. Cette relative lenteur est liée à la quantité d'informations (plusieurs Mega-octets de données) à transférer entre le scanner et le système de traitement.

## **5. PRE-TRAITEMENT :**

Lorsque l'acquisition est réalisée, la plupart des systèmes comportent une étape de pré-traitement. Généralement, ces pré-traitements ne sont pas spécifiques de la reconnaissance, mais sont des pré-traitements classiques en traitement d'images. Si l'acquisition a été effectuée en niveau de gris voire en couleur, la binarisation sera souvent l'un de ces pré-traitements. Toutefois il est à remarquer que certains chercheurs ne passent pas nécessairement par cette étape intermédiaire et extraient directement les primitives utiles à la reconnaissance à partir de l'image en niveau de gris.

## **6. RESTITUTION D'IMAGE :**

Les équipements de restitution d'images sont des systèmes qui permettent de reproduire une image sur écran, papier ou film. Il faut distinguer deux sortes de systèmes : ceux qui reproduisent des images à deux niveaux (noir et blanc par exemple) et ceux qui reproduisent des variations de gris, généralement en utilise (imprimante graphique) écran binaire, moniteur de télévision ou imprimante laser.

## **7. STOCKAGE :**

Le traitement d'images possède un certain nombre de caractéristiques qui le rend particulier en comparaison avec les autres types de traitement d'information. Une caractéristique très importante tient au fait que les images contiennent une grande quantité d'informations. Une seule image ordinaire noir et blanc de taille 512 x 512 contient un quart de Mbytes de données. Une séquence d'images contenant 25 de ces images par seconde sur une période de 10 secondes nous fournit un volume de données

dépassant 60 Mbytes. Ceci a des conséquences non désirables autant pour le stockage que pour le traitement.

Aucun ordinateur conventionnel ne pourra manipuler de tels volumes de données.

En vue d'augmenter le débit des ordinateurs de traitement d'images, il y a en principe deux mécanismes qui peuvent être utilisés :

- \* Augmentation de la fréquence d'horloge des circuits numérique
- \* Augmentation du parallélisme dans le traitement.

RECONNAISSANCE  
DE FORMES

CHAPITRE II**RECONNAISSANCE DE FORMES****1. GENERALITES: [9]**

La reconnaissance de formes (R.F.) est la première étape d'un long processus de compréhension de notre univers, en l'occurrence dans le cadre général de la communication homme-machine. Etant placée en amont de ce processus, la R.F. se doit de résoudre les premiers problèmes liés au codage des formes, à leur paramétrisation et leur discrimination. Cette tâche est souvent difficile pour plusieurs raisons. D'abord les formes appartiennent à un monde physique dont la transcription numérique est très complexe à cause de l'absence de capteurs adaptés à toutes les situations. Ensuite, la nature des formes et leur apparence varient d'un échantillon à l'autre (même au sein de même famille), ce qui multiplie les dimensions de l'espace de représentation et rallonge les temps de décision.

Il est pourtant essentiel que l'information transmise à l'ordinateur soit la plus adéquate possible vis à vis de la forme à traiter, et ceci n'est pas à priori une tâche triviale, il s'agit en effet de passer d'un objet dans un monde physique continu à une forme numérique dans un monde discret.

Ce passage, même s'il se fait par la voie de capteurs spécialisés, impose le plus souvent des simplifications de la représentation, donc un appauvrissement de l'information et une certaine « mise à plat » des propriétés de la forme.

Une des tâches importantes de la R.F. est donc de bien caractériser la forme à partir de sa seule transcription numérique c'est à dire de lui trouver une description qui la singularise mais aussi qui la rapproche de ses formes voisines et qui l'éloigne de ses « fausses » semblables. Pour arriver à réaliser finement cette distinction, la R.F. élargit la caractérisation à l'ensemble des échantillons d'une même famille de formes et juge ainsi globalement de la qualité de la description. Cette tâche, qui s'appelle communément apprentissage, constitue des groupements de formes ayant les mêmes caractéristiques.

Ainsi, plus les caractéristiques sont robustes et différentes d'un ensemble de formes à l'autre, plus il est facile de séparer les formes en familles homogènes et plus il est automatique d'associer une forme à une famille.

En revanche plus les propriétés sont redondantes, contradictoires et incomplètes, plus les répartitions des familles sont approximatives et plus il est difficile de décider. On voit là le compromis difficile que doit faire la R.F. entre la redondance et l'altération des données des capteurs, d'une part, et la fiabilité de sa modélisation, de l'autre. Ce compromis ne peut être réalisé sans l'aide de techniques de traitement souples et robustes à la fois; ce qui a incité les investigateurs dans ce domaine, depuis plus d'un demi-siècle, à développer et d'améliorer des techniques pour qu'elles approchent le mieux possible la réalité physique.

## 2. INTRODUCTION [5]

### a. . Historique

Les travaux sur la reconnaissance optique de textes ont débuté dès 1870 avec *C.R.Carley*, de Boston, qui invente "le scanner rétinien". C'était en fait un système de transmission d'images qui utilise une mosaïque de cellules photoélectriques pour reconnaître des lettres. D'autres dispositifs ont été mis au point par la suite. En 1890, *Nipkov* invente le scanner séquentiel. En 1912, *Fourrier D'Albe* crée "L'optophone", et en 1926, *Thomas* construit un procédé appelé "Tactile Relief Device". Au début des années cinquante, la reconnaissance de formes s'est développé de manière significative notamment grâce à *David Sheperd*, fondateur de la société *Intelligent Machine Research & Co.* qui peut être considéré comme le pionnier du développement des équipements d'OCR pour des applications commerciales. Avec l'apparition des premiers ordinateurs dans les années soixante, les chercheurs ont voulu associer la résolution des problèmes de la reconnaissance de l'écrit et de la parole au développement de l'informatique. Ils présageaient alors l'apparition rapide d'un ordinateur doté de facultés sensorielles. Les premiers systèmes lourds et entièrement câblés ont laissé peu à peu la place à d'autres beaucoup plus légers, flexibles et plus faciles à mettre en oeuvre (logique programmée).

Trente ans plus tard, alors que la micro-informatique a pris un essor considérable, surtout à partir des années 80, la reconnaissance de texte reste un chantier ouvert où des équipes de chercheurs à travers le monde entier tentent de trouver des solutions efficaces qui aboutiraient à un système de lecture automatique, à l'image de l'homme, répondant au besoin considérable en traitement automatique de documents.

Depuis une dizaine d'années, l'imagination des chercheurs semble avoir été véritablement libérée. De nombreuses techniques nouvelles ont été publiées dans lesquelles la recherche de taux de lecture élevés prime désormais sur la simplicité

des algorithmes. Les progrès récents en terme d'algorithmique et surtout technologique (rapidité et puissance de calcul) sont donc à l'origine de cette évolution considérable.

La reconnaissance de textes dactylographiés ou imprimés, est le domaine dans lequel les progrès les plus notables ont été obtenus, débouchant sur des systèmes fiables, commercialisés et éprouvés. Actuellement les systèmes les plus performants sont capables de reconnaître plusieurs fontes (polices de caractères) différentes avec des taux de reconnaissance relativement élevés quand la qualité du document est assez bonne.

Un autre aspect important de la R.F. est la décision. Décider revient à émettre un avis sur la forme en entrée et souvent à lui attribuer une "étiquette" ou un nom correspondant à celui d'une famille de la base d'apprentissage. Dans ce cas, on cherche parmi les familles celle qui "maximise" une fonction de correspondance, de ressemblance, de similitude ou de corrélation entre sa description et celle de la forme entrée. Afin de limiter le nombre de comparaisons et de simplifier l'étude de la ressemblance, on définit quelques représentants ou prototypes qui serviront d'uniques références pour les comparaisons. Les comparaisons sont opérées sur les caractéristiques, d'où, encore une fois, l'importance de la caractérisation dans la mise en oeuvre des règles de décision.

Enfin, l'interprétation est une tâche difficile et une même forme ne peut être reconnue de manière "sûre" qu'après l'analyse de son contexte. C'est le cas par exemple en lecture optique de formules, où un même trait horizontal est interprété comme un signe de soustraction ou un trait de fraction selon qu'il sépare horizontalement ou verticalement deux expressions arithmétiques. En vision, une même colonne de briques est reconnue comme une cheminée ou comme un poteau selon qu'elle se trouve placée sur le toit de la maison ou devant sa façade.



### 3. LES DEUX GRANDES CONTROVERSES:

Nous aborderons dans ce qui suit, très brièvement, deux thèmes quasi philosophiques assimilables à des controverses.

Sans espérer être complet, il nous a semblé en effet important de parler un peu de deux grandes interrogations qui animent la R.F. sous l'éclairage des sciences cognitives, c'est-à-dire des sciences qui s'interrogent sur les modalités, les objets et les structures de la connaissance (chez l'être humain).

Comment "fonctionne" le cerveau, la mémoire, l'apprentissage, le raisonnement? Existe-t-il un "langage" interne universel pour la pensée non verbale? Comment s'élabore la perception (visuelle ou auditive)?

Des chercheurs en R.F. se posent aussi ces questions, et surtout, sont convaincus qu'il faut se les poser pour progresser vraiment dans la résolution pratique des problèmes les plus difficiles de la R.F. Concevoir (même implicitement) la R.F. uniquement comme un ensemble de recettes est une erreur.

#### **a. CONTROVERSE COGNITIVE / COMPORTEMENTALISTE**

La résolution d'un problème donné en R.F. peut-il ou non se ramener à un ensemble fini de règles explicites, à une suite de morceaux de connaissances verbalisables et formalisables? .

Faut-il plutôt s'inspirer de nos connaissances sur la perception humaine et en particulier sur le fonctionnement "inconscient" du cerveau pour concevoir des systèmes de R.F. d'inspiration "cybernétique"?

Pour illustrer cette controverse, suivant *D.R.Hofstadter*, on peut poser la question "faussement naïve" suivante: qu'est ce qui définit la lettre "A"? Répondre n'est pas si simple si on songe à toutes les polices de caractères plus ou moins fantaisistes utilisées par les graphistes; pourtant nous savons reconnaître un "A" "sans réfléchir" et sans nous soucier du type de caractère employé. Comment est-ce possible?

Le cognitiviste va tenter de trouver les règles, aussi nombreuses soient-elles, aussi complexe que puisse être leur enchevêtrement, qui déterminent notre décision de reconnaître ou pas un "A" dans une forme. Le comportementaliste lui ne cherchera pas à analyser le concept de "A" (son essence) mais tentera de collecter un maximum de prototypes différents de "A"; ensuite, en utilisant une boîte noire (un réseau connexionniste par exemple), il essaiera de reproduire, d'imiter la fonction "reconnaissance du A" et de l'étendre à des exemples nouveaux. Est-ce une simple extrapolation ou bien une extraction de régularités?

En caricaturant, disons que le cognitiviste (le chercheur en IA?) cherche des règles alors que le comportementaliste (le statisticien, le connexionniste) recherche des régularités.

### **b. CONTROVERSE APPRENTISSAGE / REPRESENTATION**

Etant donné un ensemble de formes à reconnaître dont certaines ont été étiquetées, faut-il chercher un codage, une représentation, très sophistiquée de ces formes (en utilisant empiriquement l'ensemble d'apprentissage mais des connaissances explicites) ou plutôt concevoir des techniques d'apprentissage extrêmement efficaces permettant d'obtenir de bonnes performances sans un codage

aussi spécifique? Autrement dit, quel doit être le poids respectif de la représentation des formes et du système. Cette controverse correspond, suivant *E. Bienenstock*, à un dilemme bien connu en statistiques, celui du biais et de la variance.

La recherche de "bonnes" représentations permet de réduire le nombre de dimensions des vecteurs à classer. Cette réduction suppose cependant des connaissances à priori spécifiques sur les propriétés d'invariance des classes.

Nous savons par exemple que l'épaisseur du trait ne joue pas de rôle dans la reconnaissance d'une lettre; cela nous incitera peut être à squelettiser l'image pour diminuer le nombre de points à coder.

Inversement, en faisant porter notre effort sur la classification en raffinant le plus possible les techniques de comparaisons des représentations (par des métriques très habiles ou des fonctions discriminantes subtiles), nous essayons d'être moins spécifiques, d'avoir moins d'à priori, au risque d'être trop dépendant des prototypes que nous utiliserons.

## **4. Reconnaissance de l'écriture:**

Le but de la reconnaissance de l'écriture est de transformer un texte écrit en une représentation compréhensible par une machine et facilement reproductible par un traitement de texte. Cette tâche n'est pas triviale car les mots possèdent une infinité de représentations dues au fait qu'il existe de nombreuses polices de caractères par les textes dactylographiés, avec de nombreux styles (gras, italique, souligné, ombré, etc...) et des mises en page différentes et complexes. La reconnaissance de l'écriture apparaît donc quelque peu semblable à la reconnaissance vocale, en ce sens qu'il existe une multitude de façons d'écrire un mot, ce qui rend cette dernière tâche tout aussi difficile.

La reconnaissance de l'écriture, élargie depuis peu à l'analyse des documents, devient de jour en jour un thème de recherche à part entière. En parallèle avec les avancées sur l'écriture manuscrite dans le domaine de la lecture, de la segmentation, de l'identification, de la vérification et de l'authentification, la recherche sur l'imprimé s'est élargie au document composite où le caractère n'est pas la seule composante de l'image.

## **5. Les différentes méthodes de classification:**

Il existe plusieurs méthodes de classification. Nous dressons, ici un aperçu de celles qui sont le plus rencontrées dans la littérature et qui auront un lien, de loin ou de prêt, avec ce qui va être traité comme approche pour essayer de faire une classification de formes.

Devant la grande variété des travaux scientifiques effectués sur le sujet, il est nécessaire de classifier les techniques par types d'approches:

- - Méthodes analytiques
- - Méthodes par extraction de caractéristiques
- - Méthodes par analyse structurelle de formes.

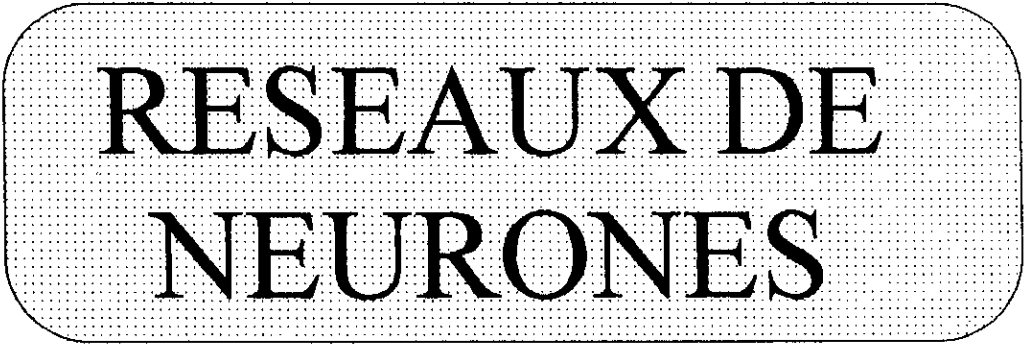
Avec toutes ces méthodes, deux types d'erreurs peuvent se produire:

- Une mauvaise reconnaissance, appelée substitution, quand le programme donne une mauvaise réponse, croyant donner la bonne.
- L'absence de reconnaissance quand le programme indique qu'il n'a pas reconnu le texte.

Remarque: les substitutions sont des erreurs plus graves pour l'utilisateur, car elles sont difficiles à localiser, alors que les non reconnaissances sont marquées et donc faciles à corriger (par des signes non significatifs).

Les méthodes classiques utilisées pour la R.F. sont basées, surtout, sur la classification suivant les paramètres extraits de l'image. Ces méthodes classiques ont l'inconvénient d'être lentes généralement, nécessitent souvent des pré-traitements de l'image tels que la squelettisation qui est peu fiable et cela est due au bruits ou à d'autres facteurs tels que l'écriture manuscrite. On comprend alors la nécessité d'introduire des R.N., non seulement parcequ'ils sont d'excellents approximateurs de fonctions, mais aussi parcequ'avec un choix adéquat de l'architecture des réseaux, ils permettent d'assurer au mieux une bonne séparabilité.

Un cas, très particulier de manuscrits; intéressant à traiter et nous rendant perspicace à la fois par sa difficulté relative dûe au caractère aléatoire dont parviennent les données (formes à classer) et par la manière d'élaborer une approche qui garantie la classification, est la signature. Ce sera la préoccupation majeure de ce travail.



RESEAUX DE  
NEURONES

CHAPITRE III**RESEAUX DE NEURONES****1. Introduction: [1]**

Une des plus importantes révolutions dans le monde de l'informatique depuis l'introduction des ordinateurs numériques s'appelle les réseaux de neurones. Ce sont des architectures parallèles capables d'apprendre sans aucune programmation simplement par apprentissage, ceci en s'inspirant de neuro biologie. Les R.N. regroupés sont aujourd'hui un certain nombre de modèles dont l'intention est d'imiter certaines fonctions du cerveau humain.

Tout cela à commencé en 1943 où *Mc-Culloch* et *PITTS* étudièrent un ensemble de neurones formels interconnectés en montrant leurs capacités à imiter quelques fonctions logiques.

En 1949, *Hebb* montrera l'importance qu'on doit au processus d'apprentissage ou couplage synaptique. Après c'est en 1958 que *ROSENBLATT* décrivit le premier modèle opérationnel de réseaux de neurones en se basant sur les idées de *Mc Culloch*, *Pitts* et *Hebb*. Ce qui à donné beaucoup d'espoir jusqu'à ce qu'en 1969 deux mathématiciens, *Minsky* et *PAPERT* démontrèrent les limites mathématique de ce modèle (le perceptron). Les chercheurs se désintéressent alors des réseaux de neurones jusqu'en 1982 lorsque *Hopfield* montre l'analogie de ces derniers avec certains systèmes physiques.

Les R.N. ont des propriétés fascinantes comme le parallélisme, la mémoire distribuée, la capacité de généralisation et la capacité d'adaptation. Toutes ces propriétés proviennent directement de la modélisation des propriétés du système nerveux de l'homme. Mais ça n'empêche que le R.N. à des limites surtout d'ordre technique avec des difficultés existantes pour l'utilisation du parallélisme.

Car le plupart des R.N. sont simulés sur des machines séquentielles. Ce qui entraîne un temps de calcul important; de plus, l'implantation des réseaux sur circuit n'en n'est encore qu'à ses débuts, et seul ce passage permettra l'exploitation réelle de leur parallélisme.

Face à ces problèmes, les principales applications proposées sont restées dans des dimensions modestes parmi les domaines d'applications.

- Reconnaissance de formes
- Synthèse vocale
- Analyse financière
- Diagnostic médical
- Traitement de parole
- Aide à la décision

Alors pour résoudre un problème, il faut d'abord définir les objectifs et choisir le type de réseaux.

## 2. LES NEURONES VIVANTS:

Le neurone est constitué d'un corps cellulaire (soma), d'où émergent deux types de prolongements : l'axone et les dendrites. L'axone est unique. Il est très fin, d'un diamètre de quelques millièmes de millimètre; sa longueur varie beaucoup suivant les neurones, entre quelques millimètres et un mètre. C'est un câble de transmission entre le corps cellulaire du neurone et sa cible, qui est soit un autre neurone, soit un muscle, soit une glande. Ce câble véhicule l'information nerveuse sous forme de potentiels électriques: Les potentiels d'action. Tout le long de son trajet, l'axone émet des branches collatérales qui, elles mêmes, se ramifient à leurs extrémités en fins prolongements et se terminent par un petit renflement appelé *bouton terminal* (Figure 3.1).



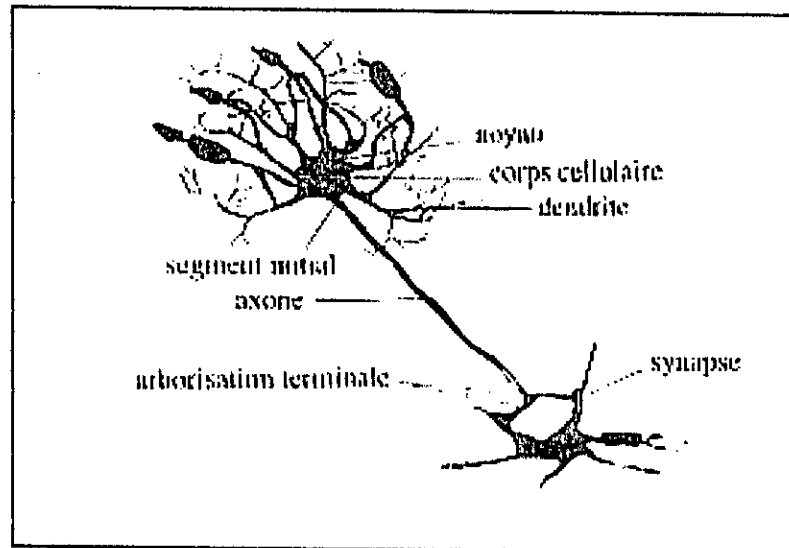


Fig : 3.1

La terminaison de l'axone est séparée de la cible par une étroite fente dont la largeur est de l'ordre d'un millionième de mètre. Ce lieu est la *synapse*, d'un mot *grec* qui signifie point de jonction. Les synapses, reproduites des milliards de fois dans le système nerveux central des mammifères, sont les lieux stratégiques où s'effectue la communication entre les neurones (*Figure 3.2*).

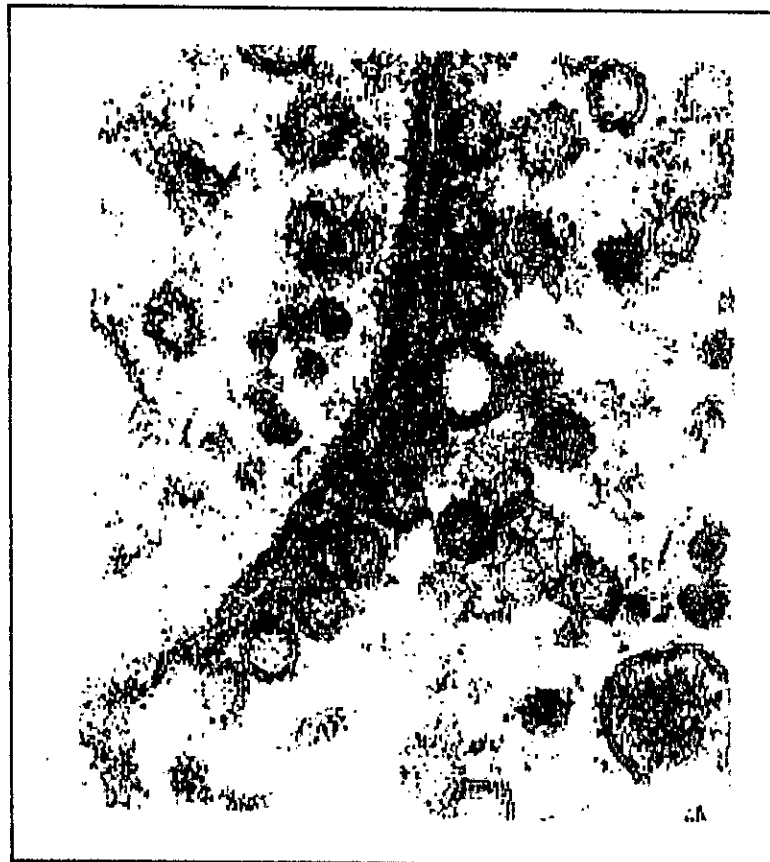


Fig: 3.2 : Cliché en microscopie électronique par transmission d'une synapse (grossissement 300.000).

Au niveau de la synapse chimique, l'arrivée des potentiels d'action provoque la libération d'une substance chimique, le neurotransmetteur, qui traversant la fente synaptique, agit sur la membrane du neurone cible. Il en résulte un changement local de la perméabilité aux ions dans la membrane du neurone cible, produisant un courant électrique : le courant synaptique. Il existe un autre type de synapse : la synapse électrique mettant en jeu un mode de transmission électrique des potentiels.

Les *dendrites*, dont le nombre est variable, quittent le soma par un tronc de gros diamètre, appelé "primaire", qui se divise en dendrites secondaires et en dendrites d'ordre supérieur jusqu'aux branches les plus éloignées et les plus fines. Cette arborescence permet de caractériser le neurone.

L'arborisation dendritique est l'un des traits évolutifs les plus frappants des architectures de cerveaux (*Figure 3.3*). Alors que la taille des corps cellulaires varie peu au cours de l'évolution, c'est à dire lorsque l'on s'élève dans l'échelle des vertèbres (phylogenèse), celle des arborisations dendritiques augmente et leur forme se complique. Par exemple, on constate que le volume du cerveau humain est beaucoup plus important que celui de la grenouille, qu'il contient un beaucoup plus grand nombre de neurones et donc d'interconnexions. Mais une observation plus fine montre aussi que la variété des formes des arborisations dendritiques et leur complexité structurale croît de manière fantastique. La différence entre le cerveau de la grenouille et celui de l'homme ne réside pas seulement dans le nombre d'éléments simples, mais surtout dans l'extraordinaire diversité de la forme des dendrites.

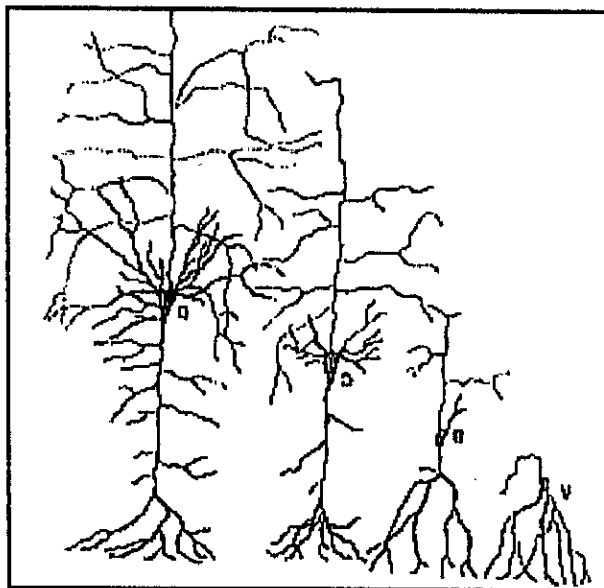


Figure 3.3 : Exemple d'évolution de neurone : les cellules "pyramidales" chez la grenouille (A), le lézard (B), le rat (C), l'homme (D)

La forme du neurone, élément de base constitutif des réseaux, détermine l'architecture générale des cerveaux. C'est au niveau des dendrites que la majorité des synapses contactent le neurone. La surface membranaire des dendrites représente 95% de la surface de réception du neurone, alors que celle du corps n'en représente que 5%.

Les techniques d'électrophysiologie permettent d'introduire une microélectrode dans le corps cellulaire du neurone, à l'aide d'un micromanipulateur. Ce microcapteur permet de mesurer les potentiels électriques produits par les synapses actives qui atteignent les dendrites et le corps cellulaire du neurone. Une sommation algébrique des potentiels synaptiques excitateurs inhibiteurs se produit au niveau du corps cellulaire. Dans le cas où le résultat de la sommation est un potentiel qui atteint le seuil critique d'excitabilité du neurone ( $-10\text{mV}$ ), un potentiel d'action est produit à l'endroit du neurone où l'axone quitte le corps cellulaire. Ce potentiel se propage vers les terminaisons axoniques. Mais la microélectrode, dont la pointe est située dans le corps cellulaire, ne voit que la résultante de tous les phénomènes synaptiques se produisant à un instant donné dans les différentes régions des dendrites. Elle ignore donc tout de la répartition spatiale de ces événements sur le neurone, et elle ne permet pas de décrire comment les courants synaptiques circulent dans l'arborisation. En d'autres termes, elle ne fournit aucune information sur la manière dont l'arborisation

dendritique traite le signal nerveux. Cette technique d'enregistrement intracellulaire ne permet pas d'élucider clairement le rôle des dendrites.

#### **a. Le traitement de l'information par les dendrites :**

Une autre démarche tente de prendre en considération l'arborisation dendritique d'un neurone. Elle consiste à assimiler les dendrites à des câbles électriques, et à leur appliquer la théorie mathématique des câbles, utilisée, par exemple, pour prédire la déformation des signaux dans les câbles sous-marin.

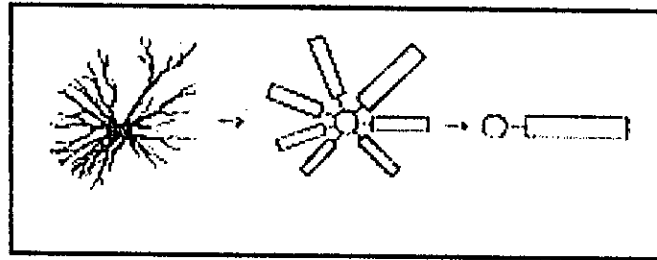
Cette approche implique une connaissance très précise de la géométrie des dendrites du neurone dans lequel les mesures électriques ont été réalisées *in vivo*.

Mais, pendant très longtemps, il a été impossible de connaître la géométrie dans l'espace d'une arborisation dendritique. L'absence de données quantitatives a conduit à développer un modèle mathématique du neurone dans lequel chaque dendrite est représentée par des cylindres équivalents, réduits eux-mêmes à un seul cylindre. Cette méthode permet de prédire l'amplitude des courants produits en différents sites de l'arborisation, lorsqu'il atteignent le corps cellulaire.

Les contraintes inhérentes au modèle conduisent à représenter l'arborisation dendritique par un seul cylindre. Cette géométrie simplifiée n'a plus rien à voir avec celle du neurone vivant (*Figure 3.1*). Pourtant, ce mode de raisonnement a fait progresser le concept du neurone en démontrant que c'est au niveau des dendrites et non du corps cellulaire que s'effectue l'essentiel du traitement de l'information. La recherche d'outil plus sophistiqués qui permettraient de mieux tenir compte à la fois de la géométrie réelle du neurone et des enregistrements électrophysiologiques de son activité, a abouti à la découverte d'un colorant permettant de marquer sélectivement un seul neurone dans le cerveaux. La substance, mise en solution, est chargée électriquement.

Introduite dans la microélectrode d'enregistrement, elle peut être injectée dans le neurone par électrophorèse, c'est-à-dire en faisant passer un courant électrique à travers

la microélectrode. Elle diffuse dans tous les prolongements dendritiques et dans l'axone du neurone. Il suffit alors de passer par toutes les étapes histologiques des coupes de cerveau et de pratiquer une réaction chimique simple pour que le neurone apparaisse, unique, coloré en brun. Ce progrès technique ouvre la voie d'un des domaines les plus fascinants de la neurobiologie actuelle : Il permet de ne plus dissocier le fonctionnement du neurone de sa forme et de son ultrastructure.



*Figure 3.4 : L'arborisation dendritique est mobilisée par un ensemble de cylindres conducteurs de dimensions différentes, puis est ramenée à un seul cylindre équivalent.*

Pour comprendre comment les dendrites d'un neurone unique traitent l'information, il faut connaître leur géométrie tridimensionnelle de manière détaillée et quantitative. Depuis 1965, une série de travaux, combinant les techniques modernes de la microscopie, de la métrologie et des ordinateurs, a développé des systèmes de reconstitution tridimensionnelle de neurones colorés intracellulairement, et dont la fonction a été analysée au préalable par les méthodes de l'électrophysiologie intracellulaire. Il est désormais possible d'élaborer un modèle de la circulation des courants dans une arborisation dendritique, en tenant compte de sa géométrie tridimensionnelle. On constate que, même dans les conditions les plus simple, cette circulation est beaucoup plus complexe qu'on ne l'imaginait.

On découvre que chaque dendrite du neurone véhicule les courants synaptiques selon sa propre géométrie. Si l'on imagine deux synapses injectant le même courant en deux sites équidistants du corps cellulaire, mais sur deux dendrites différentes, on peut prévoir qu'elles n'auront pas la même efficacité pour exciter le neurone. Les efficacité peuvent varier de 1 à 100. Grâce à des simulations sur ordinateur à partir de la description géométrique des dendrites, on découvre que, dans certaines dendrites, il y

a, dans les branches les plus éloignées du corps cellulaire, des régions au sein desquelles la circulation des courants est très facile. Cela signifie qu'entre les petites branches de l'arborisation terminale d'une dendrite, l'atténuation électrique est très faible. Ces zones particulières peuvent exister à l'extrémité de chaque dendrite et sont de ce fait, électriquement indépendantes les unes des autres.

On peut montrer par le calcul que, dans ces régions, les synapses peuvent interagir entre elles, que leurs effets ne sont pas simplement additionnés et que l'opération ressemble à une multiplication. "Ces calculs locaux" donnent un résultat, qui seul est communiqué au corps cellulaire du neurone par l'intermédiaire de la branche dendritique principale.

La seule façon de vérifier ces hypothèses et de valider les prédictions faites sur ordinateur, est de réaliser des enregistrements simultanés en de multiples points de l'arborisation dendritique d'un même neurone. Cette prouesse technique semble désormais possible grâce aux progrès récents réalisés dans le domaine des capteurs optiques. La miniaturisation de cellules photoélectriques permet d'en disposer un grand nombre sur des microcircuits pour constituer un capteur capable d'enregistrer simultanément des émissions de lumière ponctuelles. Il est possible, par l'intermédiaire d'un microscope, de placer dans le "champ de vision" de ce capteur un neurone contenant une substance dont l'opacité change en fonction de la concentration ionique dans le milieu. Lorsque le neurone fonctionne, on peut voir des variations locales de lumière en de multiples points de l'arborisation dendritique. Des expériences très récentes, réalisées *in vitro*, indiquent que de telles expériences sont possibles.

Par rapport au neurone classique décrit par Eccles, qui ressemblait au modèle de neurone formel proposé par McCulloch et Pitts, décrit un modèle plus complexe a été suggéré. Chaque dendrite du neurone, de par sa géométrie, est un canal spécifique de transfert d'information. Dans les régions éloignées de l'arborisation, des sous-unités indépendantes effectuent des opérations non linéaires, dont seuls les résultats sont transmis au sommateur, avec des efficacité différentes qui dépendent des propriétés de chaque canal dendritique principal. Chaque neurone peut être considéré comme un

assemblage de processeurs effectuant des opérations non linéaires en parallèle. Ses performances de calcul sont intimement liées à la complexité de sa forme.

### ***b. Le traitement de l'information par l'axone :***

Lorsque le résultat des opérations complexes effectuées par les dendrites atteint des valeurs supérieures au seuil d'excitabilité du neurone, il y a émission de potentiels d'action dans l'axone. Ces signaux se suivent dans le temps, avec une certaine fréquence. La géométrie de l'axone intervient également dans la façon dont le message circule dans les terminaisons axoniques, et peut en modifier la fréquence et le mode de distribution aux différentes cibles. Par exemple, des variations de diamètre le long de l'axone, ainsi qu'au niveau de ses points de branchement, augmentent ou diminuent la vitesse de propagation des potentiels d'action. La propagation peut même être interrompue. Les bifurcations jouent le rôle de "filtres" qui modifient la fréquence des potentiels. Lorsqu'un axone de gros diamètre se divise en deux branches de diamètres inégaux, les deux branches filles ont toutes les chances de véhiculer des messages différents.

Ainsi, les informations qui atteignent les cibles touchées par les différentes branches du même axone peuvent être différentes entre elles et surtout ne pas refléter celles qui sont émises au niveau du corps cellulaire. Il faut donc tenir compte du fait que l'axone n'est pas une simple ligne de transmission, mais qu'il peut, jusqu'à ses extrémités, modifier le message émis par le corps cellulaire.

Pour être moins schématique, une description fonctionnelle des neurones vivants devrait mentionner les faits suivants:

\* Un neurone peut ne pas avoir d'axone, mais seulement des éléments courts qui reçoivent, traitent et transmettent l'information. Ce type de neurone existe dans la rétine et le bulbe olfactif, mais n'a pas été mis en évidence de façon certaine dans d'autres parties du système nerveux central;

- \* Les axones peuvent établir des contacts synaptiques avec d'autres axones; par exemple, dans le cortex cérébral il existe des synapses sur le segment initial des axones de certains neurones;
- \* Les dendrites peuvent établir des synapses avec d'autres dendrites: elles constitueraient des éléments de sortie du neurone; certains axones peuvent produire un potentiel gradué et ne pas propager de potentiels d'action; les exemples en sont toutefois rares.

### 3. LES NEURONES FORMELS: [2] [4]

#### a. Une caricature: l'automate à seuil

Le neurone formel de McCulloch et Pitts est un automate élémentaire à deux états: INACTIF ou ACTIF (émission d'un *spike* le long de l'axone), ce qu'on peut formaliser en décrivant l'état (ou activité) par une variable binaire:

$V=0,1$ :

$V=0$  (repos) ou  $1$  (émission d'un spike)

On peut tout aussi bien choisir d'utiliser une variable:

$S=+1$  ou  $S=-1$

$S=1$  (repos) ou  $+1$  (émission d'un spike)

Le premier choix est plus naturel pour l'interprétation en terme de neurobiologie, le second est plus naturel dans l'interprétation en terme de systèmes magnétiques. Evidemment les deux choix sont équivalents; on passe de L'un à L'autre par un simple changement de variable:

$$S = 2V - 1$$



L'état du neurone dépend d'un certain ensemble de  $N$  signaux d'entrée,  $(V_j, j = 1, \dots, N)$  (Figure 2.5). Chaque  $V_j$  est soit l'activité d'un autre neurone, soit une valeur imposée de l'extérieur. Dans ce dernier cas l'unité  $j$  est par exemple un capteur, une cellule sensorielle; un ensemble de  $N$  capteurs peut alors représenter une rétine. Dans le langage du traitement de l'image, chaque unité  $j$  est un "pixel", et, dans le cas le plus simple d'une image en noir et blanc, chacun est soit dans l'état 1 (noir) soit dans l'état 0 (blanc). Chaque lien entre une unité d'entrée  $j$  et le neurone formel représente un contact synaptique entre l'axone issu du neurone  $j$  et l'arbre dendritique du neurone considéré. Il est caractérisé par un nombre  $J_j$ , appelé "efficacité synaptique", ou "poids" ou encore "couplage". L'efficacité synaptique est positive ou négative suivant que la synapse est excitatrice ou inhibitrice.

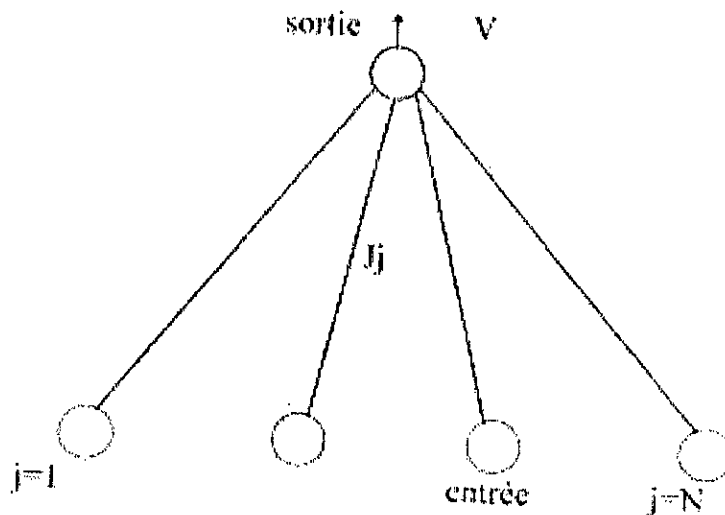


Figure 2.5 : Le neurone formel

Le neurone calcule son "potentiel post-synaptique"  $h$ , défini comme la somme des signaux d'entrée pondérés par le poids correspondant :

$$h = \sum_j J_j V_j$$

Si ce potentiel est plus grand qu'un certain seuil  $h^S$ , le neurone "décharge":

$$V = 1 \text{ si } h > h^S;$$

$$V = 0 \text{ sinon.}$$

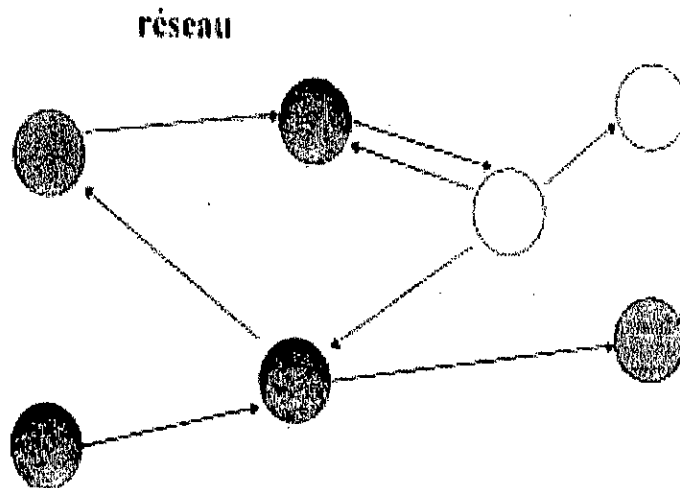
Ceci peut aussi s'écrire, en utilisant la variable  $S$ :

$$S = \text{sign}(h - h^S)$$

(où  $\text{sign}(x)$  désigne le signe de l'argument  $x$ ). Voyons maintenant comment assembler un grand nombre de ces automates.

#### 4. Le réseau de neurones: [2] [4] [1]

Un réseau de neurones est un graphe *orienté et pondéré*. Les noeuds de ce graphe sont des automates simples nommés neurones formels (ou *unités connexionnistes*) et dotés d'un état interne, l'activation, par lequel ils influencent les autres neurones du réseau. Cette activité se propage dans le graphe le long d'arcs pondérés appelés liens synaptiques; par extension, on appelle poids synaptique la pondération d'un lien synaptique. La règle qui détermine l'activation d'un neurone en fonction de l'influence de ses pairs est appelée règle ou *fonction d'activation*.



**FIGURE 2.6 :** *Un réseau neuromimétiques est un graphe dirigé et pondéré dont les noeuds sont des automates simples appelés neurones.*

L'état du réseau entier est composé de l'activation de ses neurones constitutifs. Si l'on suppose que les  $N$  neurones d'un réseau sont indexés, cet état global peut être supprimé sous la forme d'un vecteur d'activation, dont les composantes sont les valeurs d'activation des neurones individuels:

$$A = (a_1, a_2, \dots, a_N)$$

De mêmes, les liens synaptiques du réseau sont utilement décrits par la matrice des poids synaptiques, matrice carrée dans laquelle s'inscrivent les poids des liens :

$$W = \begin{bmatrix} W_{21} & \dots & W_{1n} \\ \vdots & & \vdots \\ W_{n1} & \dots & W_{nn} \end{bmatrix}$$

En général, un neurone n'est relié qu'à un sous-ensemble des autres neurones. En effet, la conductivité d'un réseau est rarement complète, mais obéit plutôt à des contraintes imposées lors de sa construction. Les liens manquants sont alors représentés dans la matrice des poids par une entrée nulle; quand les contraintes de conductivité sont sévères, la matrice des poids résultante est creuse. Ainsi, les caractéristiques de la matrice résument bien la structure de conductivité du réseau. On nommera *architecture* la structure (nombre et type des neurones, conductivité) du réseau.

#### a. *Architecture d'un réseau neuromimétique*

Il est clair que l'architecture d'un réseau joue un rôle important dans son comportement. De fait, les différents modèles neuromimétiques se distinguent souvent d'abord par les contraintes de conductivité qu'ils supposent. Parmi celles-ci, citons:

- L'existence ou l'absence de boucles ou de chemins circulaires dans le graphe (nous appellerons *récurrents* les réseaux pouvant comporter des boucles). Cette contrainte est particulièrement importante pour les modèles neuromimétiques. En effet, si l'on imagine l'activation du réseau se propageant le long des liens synaptiques, il est clair que celle-ci peut effectuer dans un réseau récurrent des cycles et des parcours élaborés qui ne sont pas possibles dans des réseaux non récurrents. La présence de boucles implique donc généralement une différence dans la complexité des comportements dynamique dont sera capable le réseau, et par conséquent dans la difficulté théorique de son analyse. En particulier, la matrice de poids d'un réseau non récurrent peut toujours être mise sous forme triangulaire:

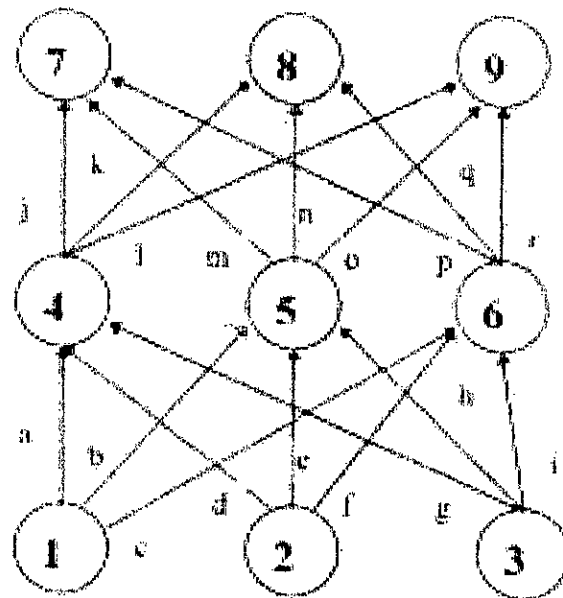


FIGURE 2.7 : Un réseau non récurrent a toujours une matrice de poids triangulaire.

- Une structuration du réseau en *couches de neurones*. Associée à cette contrainte est l'idée qu'on peut caractériser de façon différente les contraintes de connectivité intra-couches et inter couches. Ainsi, selon le modèle, les connexions entre neurones de la même couche seront interdites ou au contraire présentes sous différentes formes:

- De même, la connectivité inter-couches peut-être complète (où tous les neurones de la première couche sont reliés à tous les neurones de la seconde - c'est le cas le plus fréquent ), *injective* (chaque neurone d'une couche ne partage un lien qu'avec un seul neurone de l'autre), ou même *probabilité* (c'est-à-dire distribuée selon une probabilité de distribution donnée, souvent gaussienne) :

- Une contrainte de symétrie dans les connexions. Cette contrainte sévère est caractéristique d'une classe de modèle importantes. Elle contraint à tel point le comportement du réseau qu'une analyse théorique complète a dû être effectuée.

- Ainsi, il est possible de caractériser l'architecture des différents modèles selon le type de contraintes de connectivité qu'ils imposent à leurs réseaux.

### **b. Le choix des Poids synaptiques :**

Une fois l'architecture du réseau déterminée, il reste encore à choisir les valeurs des poids synaptiques des connexions que cette architecture a prescrites.

Souvent, ce choix est implicite dans les contraintes de connectivité choisies; ainsi, les neurones des modèles compétitifs doivent interagir par des liens essentiellement inhibiteurs. Dans d'autres cas, le réseau est suffisamment simple pour les déterminer directement, selon l'effet désiré. Cependant, les cas simples sont l'exception plutôt que la règle : un réseau neuromimétique comporte en général un grand nombre de connexions, qui ont de surcroît un rapport fort complexe avec le comportement du réseau.

De fait, la plupart des modèles de neurones disposent de mécanismes capables de modifier leurs poids synaptiques automatiquement; ils sont dotés de *règles d'apprentissage*. Avec la fonction d'activation et les contraintes de connectivité. Cette règle, ainsi que le choix de fonctions d'activation et de contraintes de connectivité, constituent les principaux éléments qui définissent un modèle neuromimétique.

Il en résulte que le réseau possède deux dynamiques: celle de l'activité de ses neurones et celle des poids de ses liens. En général, le changement d'activation du réseau est beaucoup plus rapide que l'évolution de ses poids synaptiques, et on suppose que la dynamique de l'activation est indépendante de la dynamique des poids.

### **c. Les entrées et les sorties du réseau :**

Tel que nous avons décrit jusqu'ici, le réseau neuromimétique n'a qu'un intérêt purement formel. En effet, tout l'intérêt d'un réseau réside en sa capacité de résoudre des problèmes pratiques. Pour l'utiliser, il est donc nécessaire de communiquer au réseau les données du problème en question, et de lui soutirer le résultat de ses calculs.

Répartissons donc les neurones du réseau en trois ensembles E, S (qui ne sont pas nécessairement disjoints) et C :

- *L'ensemble E des neurones d'entrée.* Nommons *neurones d'entrée* les neurones du réseau qui perçoivent les données du problème.

- *L'ensemble S des neurones de sortie.* Choisissons un sous-ensemble de neurones dont l'activation sera interprétée comme la réponse du réseau : ce sont *les neurones de sortie*.

- *L'ensemble C des neurones cachés.* Nommons enfin *neurones cachés* ceux qui ne sont ni à l'entrée ni à la sortie du réseau. Les neurones cachés n'ont aucun lien direct avec l'utilisateur et doivent donc agir par l'intermédiaire d'autres neurones .

A partir de ces définitions, il est possible de décrire les entrées et sorties du réseau. En effet, on supposera que les données sont communiquées au réseau sous la forme *d'un patron d'entrée I*, dont les composants sont les valeurs  $i_k$  scalaires communiquées a chaque neurones d'entrée :

$$I = (i_{j_1}, i_{j_2}, \dots, i_{j_s}), j_k \in E$$

Le résultat des calculs du réseau est constitué de l'activation de ses neurones de sortie. Collectivement, ces activation constituent le *vecteur de sortie* du réseau  $A_s$  :

$$A_s = (a_{i_1}, a_{i_2}, \dots, a_{i_s}), i_k \in S$$

#### **d. Applications des réseaux neuromimétiques :**

Nous avons vu que la recherche connexionniste a pris une ampleur considérable dans la dernière décennie, à tel point qu'il est aujourd'hui illusoire d'espérer dominer le sujet entier. Outre les travaux propres a réseaux neuromimétiques, un grand nombre d'auteurs utilisent le connexionnisme pour résoudre des problèmes d'autres domaines, autant appliqués que théoriques.

Ainsi, une grande partie des travaux connexionnistes peuvent être qualifiés « d'applicatifs », car ils ont comme souci premier la résolution efficace d'un problème concret. Les applications les mieux résolues par des méthodes connexionnistes à l'heure actuelle sont des tâches dites « de bas niveau », où il n'est en générale pas nécessaire d'effectuer des calculs séquentiels suivis. Dans ces domaines, les possibilités *d'apprentissage automatique*, de généralisation, et de traitement l'information *incomplètes ou bruitées* sont souvent citées comme les avantages prépondérants du connexionnisme. Il en est ainsi par exemple de :

- *La reconnaissance des formes*. Dans ce type de problème, les données d'entrée représentent l'information recueillie par un ou plusieurs capteurs (caméra, sonar, micro). Le but est ici de reconnaître le ou les objets perçus par le capteur .

- *La classification*. Ce type d'application ressemble beaucoup à la reconnaissance de forme, en ce que son but est d'associer à une entrée donnée l'étiquette d'une forme connue. La différence réside souvent dans la source des données (qui peut être quelconque). La similitude entre plusieurs types de réseaux et des techniques de classification et d'analyse de données est bien connue; en effet, cette correspondance est tellement forte que plusieurs auteurs considèrent les réseaux connexionnistes comme des classifieurs élaborés.

- *La transformation*. Plusieurs applications ont été proposées où le réseau doit apprendre une transformation mathématique entre l'entrée et la sortie. En traitement d'image, par exemple, il existe des modèles connexionnistes capables de calculer les contours d'objets, la correspondance temporelle d'objets en mouvement et la compréhension de données.

- *La prédiction et le contrôle de processus*. Le problème de la *prédiction* consiste à estimer l'état futur d'un processus à partir d'un historique de son comportement et des variables environnementales attenantes. *En contrôle*, il s'agit non seulement d'estimer passivement l'état futur, mais aussi d'intervenir de façon à atteindre un but donné. Dans ce cadre, un domaine où les réseaux ont un succès incontestable est le domaine de la finance. Dans un contexte plus industriel, les



applications sont diverses; citons la prédiction de séries temporelles par exemple, la consommation électrique ou le contrôle automatique de processus dynamiques.

Parallèlement à ces travaux d'application, plusieurs auteurs proposent des *modèles scientifiques* fondés sur des réseaux de neurones. Outre les modèles neurobiologiques, on trouve des réseaux de neurones en particulier dans les «sciences cognitives», disciplines qui ont comme but l'étude de la cognition humaine. Les modèles connexionnistes en psychologie sont nombreux, et ont une influence importante, notamment en démontrant la vraisemblance d'un traitement parallèle de l'information. C'est le cas, par exemple, des modèles du lexique, qui ont été beaucoup influencés par le connexionnisme. En *linguistique*, plusieurs modèles ont été proposés, bien qu'ils n'occupent pas une place aussi importante que les modèles psychologiques. Citons par exemple certains modèles connexionnistes dans l'étude de la sémantique, de la syntaxe et de la phonétique, ainsi que dans les nouvelles écoles de linguistiques dites «cognitives».

Si les mécanismes propres aux modèles connexionnistes sont des outils explicatifs puissants, leur intégration dans un modèle cognitif ne se fait cependant pas sans difficultés. De fait, les interprétations prêtées aux «neurones» de ces modèles sont souvent difficilement conciliables avec une quelconque réalité.

#### **e. Un modèle Plus général :**

Si les propositions de McCulloch et Pitts sur le raisonnement humain nous semblent aujourd'hui quelque peu dépassées, leur modèle théorique du neurone demeure en revanche très influent. En effet, L'ensemble des travaux connexionnistes lui sont (directement ou indirectement) apparentés. C'est notamment le modèle de McCulloch et Pitts qui démontra L'intérêt computations de cette propriété qu'ont les cellules nerveuses d'émettre et de recevoir des impulsions. Cependant, plusieurs différences importantes distinguent les modèles plus récents :

- On tient généralement compte du fait que l'influence d'un neurone sur un autre n'est pas toujours la même : elle varie de neurone en neurone, et peut changer dans le temps (notamment par apprentissage). De la neurobiologie, l'on sait que l'amplitude des impulsions nerveuses est (à peu de choses près) constante, et que c'est au niveau de la synapse que se détermine l'importance de l'impulsion sur le neurone récepteur. Ce fait est représenté dans le modèle par la *pondération* que l'on attribue à chaque lien synaptique.

-Souvent, l'activation du neurone n'est pas limitée aux seules valeurs 0 et 1, mais prend plutôt une valeur continue. De plus, la fonction d'activation est en général plus élaborée qu'une simple fonction seuil. Dans la mesure où l'analogie avec les neurones biologiques est encore possible dans ces modèles plus récents, la valeur d'activation n'est plus interprétée comme l'émission d'une impulsion unique, mais plutôt comme la *fréquence instantanée* d'émission des impulsions nerveuses par la cellule.

- Pour pouvoir rendre compte de tous ces facteurs, il nous faut un modèle du neurone plus élaboré.

Le modèle général comprend:

- Un ensemble *d'influences* provenant d'autres neurones par l'entremise de *liens synaptiques pondérés*.

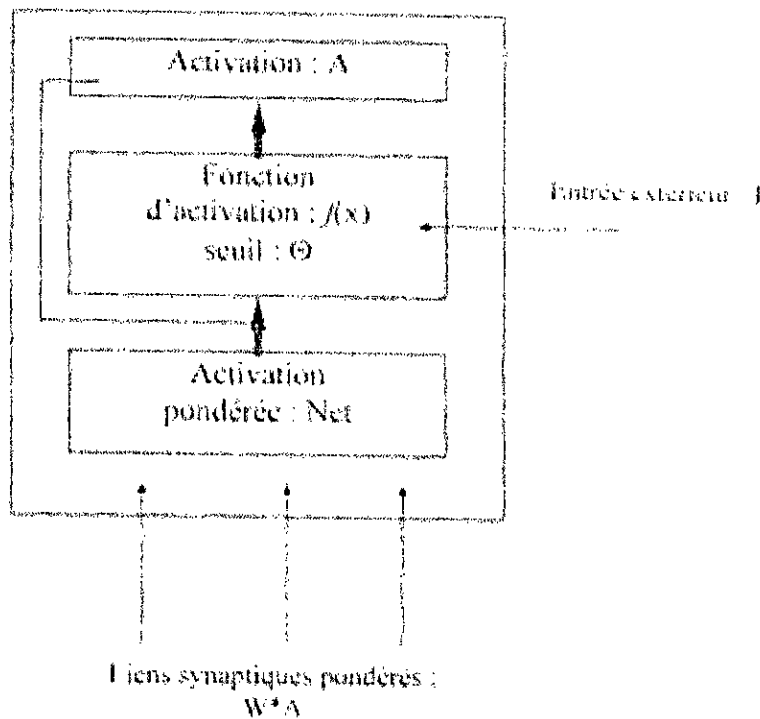
- Une étape interne de calcul où ces influences sont transformées en une valeur unique, *l'activation pondérée*.

- Un seuil d'activation  $\Theta$  du neurone, valeur à comparer à l'activation pondérée et en dessous duquel le neurone ne répondra pas (ou peu).

- Eventuellement, une *entrée extérieure*, permettant d'exprimer des informations aux neurones d'entrée du réseau.

- Enfin, une fonction qui détermine l'activation réelle du neurone à partir de son activité pondérée, de son seuil et de son entrée extérieure. C'est la *fonction d'activation*  $f(x)$ . Certains auteurs ajoutent à ce modèle une étape de calcul supplémentaire : la *fonction de transfert*  $g(x)$ , déterminerait une *valeur de sortie* du neurone.

FIGURE 3.8 Modèle général du neurone



Ce modèle s'exprime mathématiquement par les relations suivantes:

$$a_i^{t+1} = f_i(a_i^{t+1}, \text{Net}_i^t, \Theta_i, I_i^t)$$

où:

- $a_i^t$  est l'activation du neurone  $i$  au temps  $t$ ,
- $\text{Net}_i^t$  est son activation pondérée,
- $\Theta_i$  est son seuil,
- $I_i^t$  est la valeur de l'entrée extérieure,
- et  $f_i$  est sa fonction d'activation.

En général,  $\text{Net}_i$  est calculée par une simple somme pondérée des influences d'entrée des neurones :

$$\text{Net}_i = \sum w_{ij} * a_j^t$$

### **f. La fonction d'activation d'un neurone formel :**

A la lumière des sections précédentes, il est clair que la fonction d'activation joue un rôle prépondérant dans le comportement du neurone, et de là, du réseau entier.

Vu son importance, on se doute bien que plus d'une fonction a été proposée dans la littérature. Cependant, bien que toute fonction scalaire puisse en principe être candidate à l'emploi dans un réseau neuromimétique, relativement peu d'entre elles sont employées en pratique.

Dans cette section, nous relèverons les caractéristiques importantes de ces fonctions, puis regarderons de plus près quelques fonctions d'activations les plus connues.

#### **1. Caractéristiques des fonctions d'activation :**

Les fonctions d'activation utilisées dans les modèles connexionnistes d'aujourd'hui sont variées - certains modèles emploient même plusieurs fonctions différentes dans un même réseau selon le rôle du neurone. Malgré cette diversité, il est toutefois possible de caractériser les fonctions d'activation selon trois axes :

- *Fonctions à valeurs continues ou fonctions à valeurs discrètes.* Plusieurs modèles préconisent des fonctions d'activations discrètes bivaluées ( $[0,1]$  ou  $[-1, 1]$ ) ou à petit nombre de valeurs (c'est notamment le cas du modèle de McCulloch et de Pitts vu précédemment). Cette restriction est utile à plusieurs titres : les modèles qui les emploient sont souvent plus simples et se prêtent donc mieux à une analyse théorique. De plus, ils sont plus adaptés à une réalisation en « dur », par exemple sur cartes VLSI. Enfin, d'un point de vue neurobiologie, l'activation ou inactivation d'un neurone formel s'interprète naturellement en termes d'émission d'une impulsion nerveuse.

En revanche, la majorité des modèles utilisés aujourd'hui préfèrent employer des fonctions d'activation *continues*, qui permettent de communiquer et de traiter plus d'informations à la fois dans un seul neurone. Ceci a pour conséquence d'augmenter la

puissance de calcul des réseaux. De plus la dérivabilité des fonctions continues est souvent fort utile.

- *Fonctions déterministes ou fonctions stochastiques*: Une fonction stochastique est une fonction dont le résultat est en partie aléatoire : elle peut pour une même entrée répondre de plusieurs façon différentes selon une distribution de probabilité donnée. Bien que la majorité des réseaux neuromimétiques soient équipés de fonctions déterministes , il existe plusieurs modèles stochastiques.

En neurobiologie, il est reconnu que la réaction d'un neurone à une impulsion dépend en grande partie de la quantité de neurotransmetteur émis dans la terminaison axonale, et par le nombre de récepteur affectés à la réception. Ces valeurs ne sont pas constantes, mais peuvent varier d'impulsions en impulsions - en fait, le neurone biologique est souvent considéré par les neurobiologie une machine non déterministe. Cette caractéristique a de fortes implications pour un modèle qui tenterait d'expliquer le traitement de l'information dans le cerveau. Par exemple, elle implique une limite à la précision des calculs effectués par un seul neurone. Pour ceux qui recherchent une certaine vraisemblance neurobiologique, ceci est un argument en faveur des fonctions stochastiques. Neurobiologie mise à part, la variabilité d'une fonction stochastique a plusieurs effets utiles sur le calcul effectué par le réseau entier.

- *Fonction à mémoire ou fonction sans mémoire*. Une fonction peut ou non tenir compte de ses valeurs passées dans le calcul de ses valeurs futures.

Indépendamment de ces axes, il est possible de dégager un ensemble de caractéristiques que l'on retrouve souvent dans les modèles neuromimétiques:

- *monotonie* :  $f$  est en général monotone croissante.
- *sueillage* :  $f$  peut posséder une valeur en deçà de laquelle sa valeur est négligeable. Cet effet de seuillage confère au neurone une certaine résistance au bruit.
- *saturation* : de même  $f$  peut posséder une valeur maximale au-dessus de laquelle sa valeur de réponse est essentiellement fixe. Ceci permet d'éviter de propager des grandes valeurs dans le réseau .

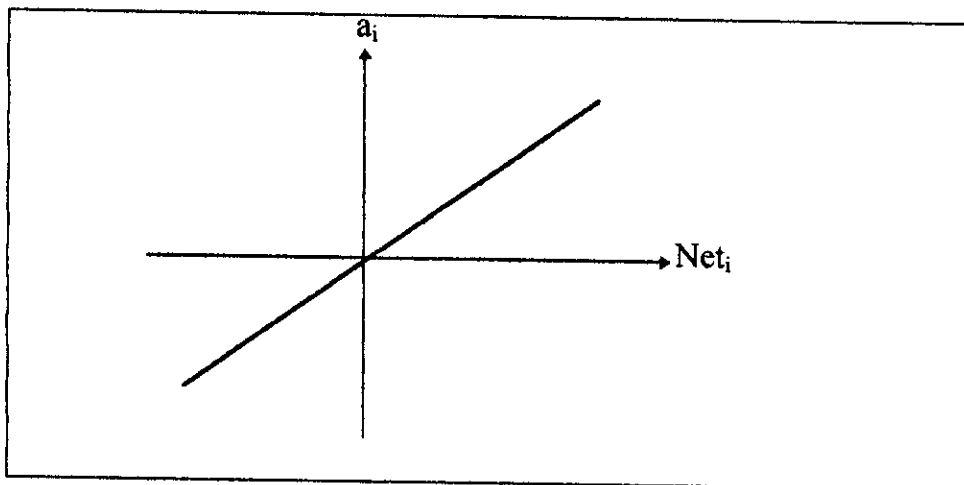
- *dérivabilité* :  $f$  est souvent dérivable, ce qui est nécessaire dans certains modèles, notamment lors de la définition des fonctions d'apprentissage.

On appelle « *squashing* » une fonction monotone croissante et bornée. Une fonction sigmoïde est par définition une fonction « *squashing* » continue.

## 2. Exemples de fonctions d'activation :

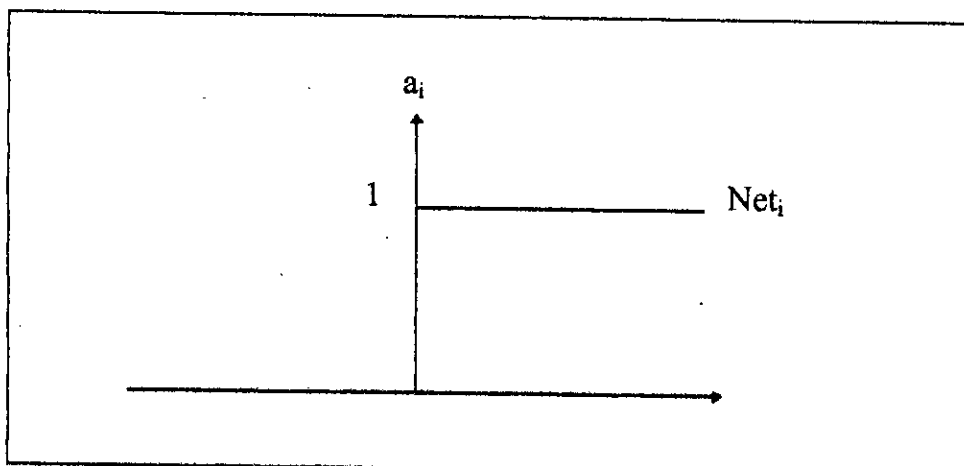
Fonction linéaire

$$a_i = \lambda \text{Net}_i$$



La fonction d'activation la plus simple qui soit. Mais cette simplicité induit des limitations au niveau des capacités de calcul.

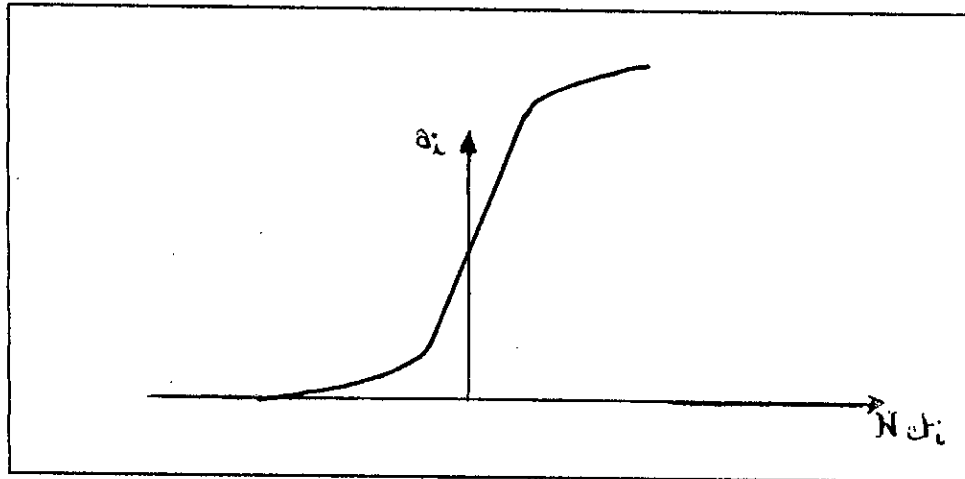
*Fonction seuil*



Le seuillage introduit une non linéarité dans le comportement du neurone. Cependant, il limite les réponses possibles à deux valeurs.

*Fonction sigmoïde exponentielle*

$$a_i = (1 + e^{-\text{Net}_i})^{-1}$$



La fonction sigmoïde a l'avantage d'être continue et d'avoir une dérivée simple à calculer.

### **g. Apprentissage :**

#### **1. La procédure d'apprentissage :**

L'apprentissage d'un réseau se fait généralement dans le contexte d'une tâche ou d'un comportement à apprendre. Les informations à traiter sont codées sous la forme d'un vecteur appelé *patron d'entrée*, qui est communiqué aux neurones d'entrée du réseau. La réponse du réseau s'interprète à partir de la valeur d'activation de ses neurones de sortie, dont le vecteur s'appelle *patron de sortie*. Lors d'un apprentissage supervisé, on dispose aussi du comportement de référence que doit apprendre le réseau, exprimé sous la forme de *patron de sortie*.

En général, l'apprentissage se fait sur une période relativement longue, durant laquelle les patrons d'entrée (et éventuellement de sorties désirées )

## 2. Les différents types d'apprentissage :

### 1) *Apprentissage supervisé*

Ici, il s'agit de deviner une règle qu'illustrent les exemples d'une base d'apprentissage. Prenons un problème de classification, par exemple la reconnaissance de chiffres manuscrits. On constitue une base d'exemples en digitalisant une série d'images de chiffres. Si chaque image contient N pixels binaires on va construire un réseau ayant N unités d'entrée, et par exemple 10 unités de sortie. On souhaite que, si une image représentant le chiffre K est présentée au réseau, alors la K<sup>i</sup>ème unité de sortie s'allume et les autres unités restent inactives. Pour une architecture donnée, l'apprentissage va donc consister à modifier les poids (les efficacités synaptiques) de façon à ce que le réseau ait le bon comportement pour tous les exemples de la base d'apprentissage manuscrit.

### 2) *Apprentissage non supervisé*

Dans certains cas la base d'apprentissage contient un ensemble de formes à classer, mais les classes ne sont pas définies à l'avance. On va alors demander au réseau de regrouper les exemples en classes en fonction d'un critère de similarité choisi *a priori*. En général, ce résultat est obtenu sans qu'il soit nécessaire de comparer directement les exemples entre eux. En pratique, on présente successivement les exemples sur la couche d'entrée, et on modifie les couplages par une règle choisie à l'avance. On parle d'apprentissage non supervisé. On emploie aussi le terme d' *auto-organisation*: partant d'un état initial quelconque, le réseau se structure "de lui même". De tels processus semblent être à l'oeuvre dans la formation du système visuel ou la constitution des cartes mentales, par exemple.



### e. Choix du réseau:

De nombreux travaux théoriques ont abordé l'approximation des fonctions continues par un réseau de neurones, ce qui a poussé à aborder le problème du choix de l'architecture du réseau.

On va se baser sur le théorème de *Kolmogorov* et celui de *Store-Weierstrass* qui fournissent des théorèmes d'existence d'un réseau multicouches.

- *Kolmogorov* prévoit que toute fonction continue de  $[0,1]^p$  dans  $[0,1]^n$  est approchable par un réseau multicouche constitué d'une première couche de  $p$  (neurones) d'une seconde de  $(2n+1)$  neurones et d'une couche de sortie de  $n$  neurones, chaque couche étant totalement connectée à la suivante.

-Alors que *Store-Weierstrass* montrent qu'un réseau de neurones possédant une fonction non linéaire dite saturante peut approximer toute fonction continue de  $[0,1]^p$  dans  $[0,1]^n$ .

Remarque: en pratique il ne faut pas se limiter systématiquement à 1 couche cachée. Certains problèmes peuvent être résolus par un petit réseau à 4 couches ( 2 couches cachées ) alors qu'ils font appel à une infinité de neurones avec un réseau à 3 couches.

ALGORITHME DE  
RETROPROPAGATION DU  
GRADIENT

## CHAPITRE IV

# L'ALGORITHME DE RETROPROPAGATION DU GRADIENT

## 1. Introduction

Les réseaux de neurones à rétropropagation ont été découverts indépendamment par plusieurs chercheurs de disciplines assez différentes. La première approche du gradient décroissant pour l'entraînement des réseaux de neurones multicouches est de source mathématique qui a été développée par *Amari* (1967) qui a introduit une seule couche de neurones cachés pour réaliser une classification non linéaire. L'approche de *Amari* était dans le bon sens mais cela n'empêche que c'était une description non complète de la manière de développer une architecture multicouche.

*Bryson & Ho* en 1969 ont développé un algorithme très similaire jusqu'à ce que *Werbos* en 1974 a découvert, en travaillant sur sa thèse de doctorat en statistiques, l'algorithme de rétro-propagation et plusieurs variantes-appelé l'algorithme du retour dynamique ( the algorithm dynamic feed-back ).

*Parker* en 1982 l'a appelé l'algorithme de la logique d'apprentissage (Algorithm learning logic). Il a même fait des pas préliminaires pour breveter son travail tout en faisant sa graduation à l'université de *Stanford*.

En 1986, *Rumelhart, Hinton et Williams*- groupe d'informaticiens exploite le pouvoir et le potentiel de la rétro-propagation ce qui a poussé la communauté scientifique à s'y intéresser de plus en plus ce qui a fait de lui l'algorithme connexioniste le plus utilisé qu'on désigne par « back propagation » basé sur la règle de *Windrow-Hoff* pour un réseau multicouche. Il constitue même l'une des méthodes utilisées pour régler le problème de séparabilité linéaire.

## 2. REGLE DE WIDOW-HOFF [4] [2] [1]

La règle de widow-hoff proposé en 1960 est intéressante à plusieurs titres par exemple elle peut s'obtenir en posant le problème de minimisation quadratique.

On considère un réseau constitué de  $n$  neurones recevant des vecteurs à  $p$  composantes les  $p$  entrées  $X_K$  du réseau sont distribuée sur tous les neurones. Les neurones de cette couches, distribuerons ces valeurs à la couche caché, ce qui donne

$$\text{net } p_j^h = \sum_{i=1}^N W_{ji}^h x_{p_i} + \theta_j^h \quad (1)$$

$W_{ji}^h$  : les poids des connexion entre la couche d'entrée et la couche cachée.

$\theta_j^h$  : est le biais

La sortie des neurones de la couche cachée est donnée par l'équation.

$$i_{pj} = f_j^h(\text{net}_{pj}^h) \quad (2)$$

La fonction  $f_j^h(x)$  est la fonction d'activation que nous avons vue dans le chapitre précédent, généralement on utilise la fonction sigmoïde  $f(x) = (1+\exp(-x))^{-1}$  et la fonction tangente hyperbolique  $f(x) = \text{th}(\alpha x)$ .

Les entrées de la couche de sortie est donnée par  $\text{net}_{ph}^0 = \sum_{j=1}^i w_{kj}^0 i_{pj} + \theta_k^0 \quad (3)$

$w_{kj}^0$  : les poids entre la couche cachée et la couche de sortie.

Les sorties de la couche de sortie sont:

$$O_{ph} = f_h^0(\text{net}_{ph}^0) \quad (4)$$

Comme on peut le voir dans la figure (4.1).

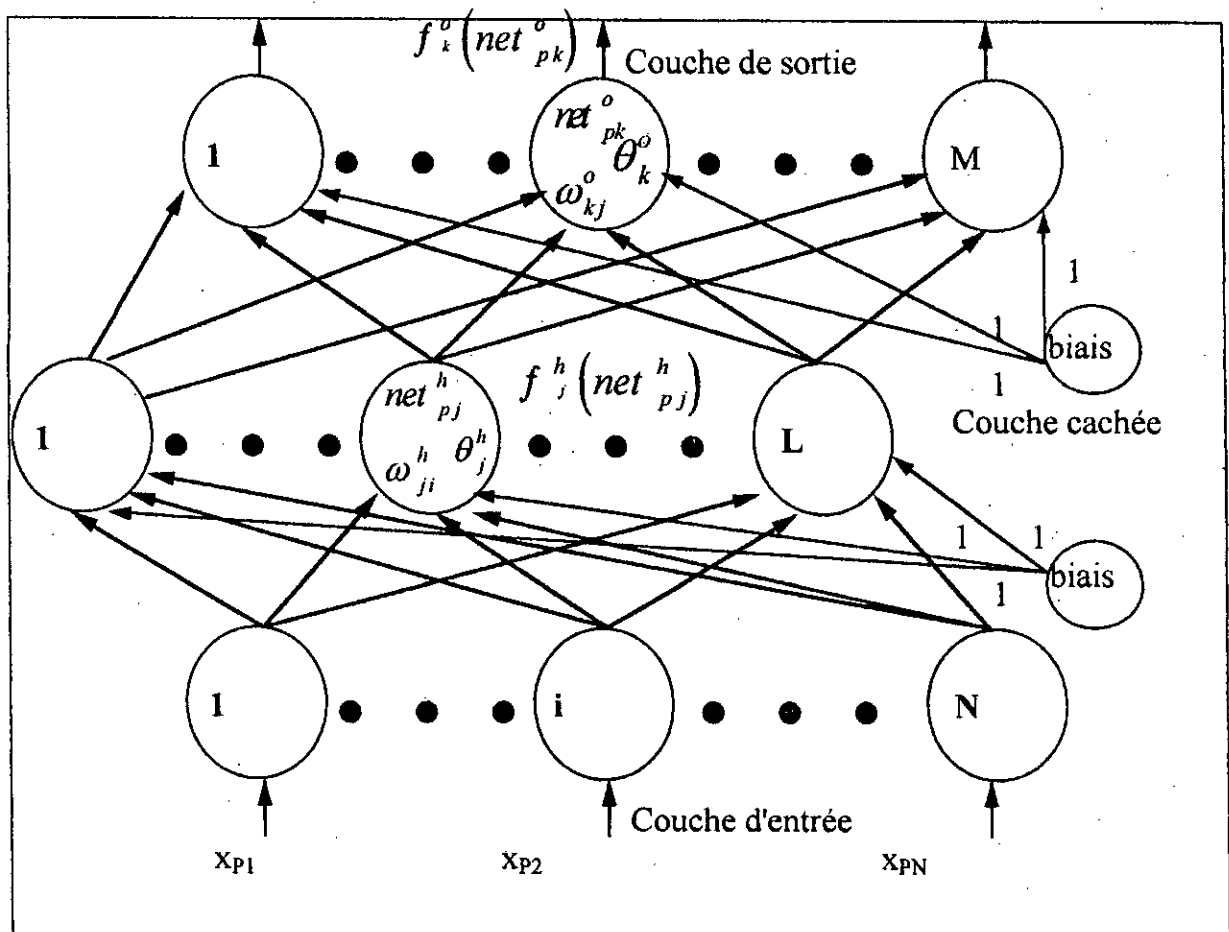


Figure 4.1: RESEAU DE RETROPROPAGATION A 03 COUCHES

On peut associer la différence entre les sorties désirées ( $y_{ph}$ ) et les sorties reçues ( $O_{ph}$ ) l'erreur quadratique.

$$E_p = \frac{1}{2} \sum_{h=1}^n (y_{ph} - O_{ph})^2 \quad (5)$$

$\frac{\partial E_p}{\partial \omega_{kj}}$  ce qui va nous permettre de calculer le gradient par rapport aux poids, ce qui nous emmène aux adaptations des poids de la couche cachée et les poids de la couche de sortie.

$$\delta_{ph}^o = (y_{ph} - O_{ph}) f_h^{o'}(net_{ph}^o)$$

On remplace dans (8) on obtient:

$$w_{hj}^o(t+1) = w_{hj}^o(t) + \eta \delta_{ph}^o i_{pj} \quad (10)$$

**\* Adaptation des poids de la couche cachée:**

Comme précédemment, on va reprendre les mêmes étapes sauf que cette fois ci, le calcul du gradient se fait par rapport aux poids de la couche cachée, pour cela reprenons l'erreur quadratique.

$$E_p = \frac{1}{2} \sum_{h=1}^n (y_{ph} - O_{ph})^2$$

d'après (4):

$$E_p = \frac{1}{2} \sum_h (y_{ph} - f_h^o(net_{ph}^o))^2$$

d'après (3):

$$E_p = \frac{1}{2} \sum_h \left( y_{ph} - f_h^o \left( \sum_j w_{hj}^o i_{pj} + \theta_h^o \right) \right)^2$$

le calcul du gradient devient:

$$\frac{\partial E_p}{\partial w_{ji}^h} = \frac{1}{2} \sum_n \frac{\partial}{\partial w_{ji}^h} (y_{ph} - O_{ph})^2$$

$$\frac{\partial E_p}{\partial w_{ji}^h} = - \sum_h (y_{ph} - O_{ph}) \frac{\partial O_{ph}}{\partial (net_{ph}^o)} \frac{\partial (net_{ph}^o)}{\partial i_{pj}} \frac{\partial i_{pj}}{\partial (net_{pj}^h)} \frac{\partial (net_{pj}^h)}{\partial w_{ji}^h}$$

ce qui va donner:

$$\frac{\partial E_p}{\partial w_{ji}^h} = - \sum_h (y_{ph} - O_{ph}) f_h^{\circ}(\text{net}_{ph}^{\circ}) w_{hj}^{\circ} f_j^h(\text{net}_{pj}^h) h_{pj} \quad (11)$$

d'après (11) on peut écrire:

$$\Delta_p w_{ji}^h = \eta f_j^{h'}(\text{net}_{pj}^h) x_{pi} \sum_k \delta_{ph}^{\circ} w_{hj}^{\circ} \quad (12)$$

pour condenser l'équation (12) on pose:

$$\delta_{pj}^h = f_j^{h'}(\text{net}_{pj}^h) \sum_h \delta_{ph}^{\circ} w_{hj}^{\circ} \quad (13)$$

on obtient enfin:

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta \delta_{pj}^h x_i \quad (14)$$

**\* Résumé de l'Algorithme:**

- 1) Appliquer le vecteur d'entrée  $X_p = (X_{p1}, X_{p2}, \dots, X_{pN})^t$  à la couche d'entrée.
- 2) Calculer les entrées totales de chaque neurone de la couche cachée:

$$\text{net}_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h$$

- 3) Calculer les sorties de la couche cachée:

$$i_{pj} = f_j^h(\text{net}_{pj}^h)$$

- 4) Passer à la couche de sortie, calculer les entrées totales de chaque neurone:

$$\text{net}_{ph}^0 = \sum_{j=1}^i w_{kj}^0 i_{pj} + \theta_k^0$$

- 5) Calculer les sorties:

$$O_{ph} = f_h^0(\text{net}_{ph}^0)$$

- 6) Calculer l'erreur de la couche de sortie:

$$\delta_{ph}^0 = (y_{ph} - O_{ph}) f_h^0(\text{net}_{ph}^0)$$

- 7) Calculer l'erreur de la couche cachée:

$$\delta_{pj}^h = f_j^h(\text{net}_{pj}^h) \sum_h \delta_{ph}^0 w_{hj}^0$$

- 8) Adaptation des poids de la couche de sortie:

$$w_{kj}^0(t+1) = w_{kj}^0(t) + \eta \delta_{pk}^0 i_{pj}$$

- 9) Adaptation des poids de la couche cachée:

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta \delta_{pj}^h x_i$$

- 10) Calculer l'erreur:

$$E_p = \frac{1}{2} \sum_{k=1}^m (O_{pk} - y_{pk})^2$$



### 3. Aspects pratiques de l'algorithme [2] [4]

On a constaté qu'il est nécessaire de parler de l'aspect pratique de l'algorithme à cause de sa nécessité et de son importance, en effet, c'est un algorithme de type gradient stochastique, fondé sur la minimisation d'une erreur instantanée et non pas d'une erreur calculée sur toute la base d'apprentissage.

#### a. Lissage de la règle d'adaptation :

La vitesse de convergence est assez lente et nécessite un pas d'adaptation petit pour éviter l'instabilité.

L'amélioration la plus couramment utilisée consiste à ajouter un terme de filtrage sur les incréments d'adaptation, ce terme est souvent appelé « Momentum » dans la littérature, en fait l'adjonction de ce terme correspond à la minimisation du critère E.

On arrive à la règle d'apprentissage bien connue:

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta \delta_{pk} i_{pj} + \alpha \Delta_p w_{kj}^o(t-1)$$

avec ...  $\alpha$  inférieur et voisin de 1.

#### b. Valeurs initiales des poids:

Les valeurs initiales des poids doivent être différentes de zéro, si c'était le cas, les incréments d'adaptation des poids sont nuls, en pratique, on choisit aléatoirement les valeurs initiales des poids selon une distribution uniforme dans l'intervalle  $[-\alpha_1, \alpha_2]$  tel que le choix des bornes dépendant de la dynamique des signaux d'entrée.

La solution la plus simple consiste alors à choisir les poids centrés, on a alors  $\alpha_1 = \alpha_2 = M$

d'après *Bunel* on a:

$$M = \sqrt{\frac{3\sigma_p^2}{\sum E[x_j^2]}}$$

où :

$$\sigma_p^2 = \sum_{j=1}^n E[w_{ij}^2] E[x_j^2]$$

### c. choix du pas d'adaptation:

Plusieurs solutions ont été proposées par de nombreux auteurs. d'après *Freeman* on choisit un pas d'adaptation entre [ 0.05, 0.25 ], mais le problème qui se pose chaque fois c'est le rapport vitesse complexité qui n'est pas toujours convaincant. l'une des solutions par exemple c'est celle de *Bunel* qui consiste à adapter le pas afin d'avoir une décroissance régulière de l'erreur quadratique moyenne (et non pas des paramètres

### d. test d'arrêt

En pratique il faut calculer un indice de convergence car on ne peut assurer la convergence de l'algorithme pour un nombre fini et petit d'itérations.

L'un des tests d'arrêt consiste à fixer un nombre d'itérations; alors qu'un autre cherche à arrêter l'algorithme si l'erreur  $E$  est minimale, (c'est à dire  $E$  est inférieur à un seuil  $E_{\min}$ ).

Enfin, il faut rappeler que le critère minimisé présente des minimums locaux dans lesquels l'algorithme peut converger, et qui ne sont pas de bonnes solutions.

### e. **Choix du réseau:**

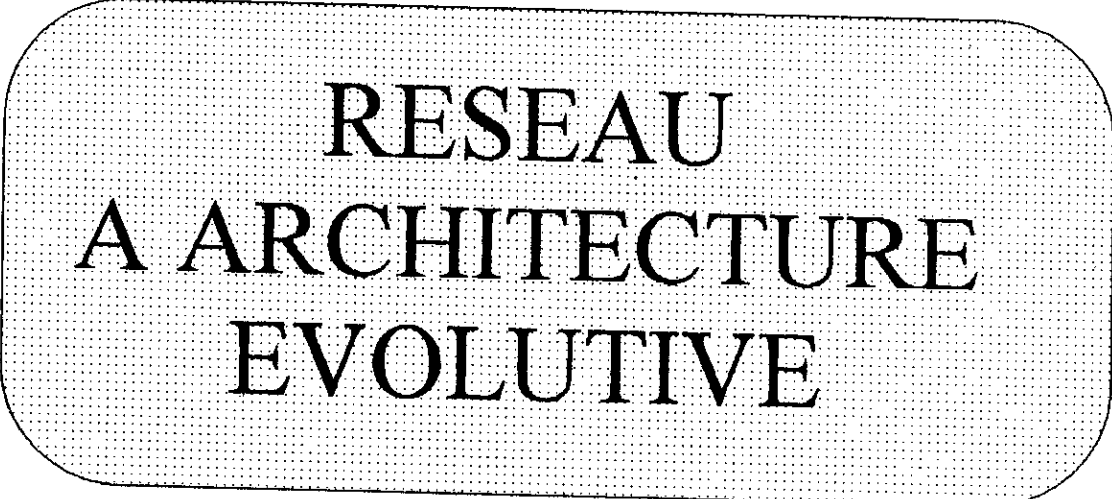
De nombreux travaux théoriques ont abordé l'approximation des fonctions continues par un réseau de neurones, ce qui a poussé à aborder le problème du choix de l'architecture du réseau.

On va se baser sur le théorème de *Kolmogorov* et celui de *Store-Weierstrass* qui fournissent des théorèmes d'existence d'un réseau multicouches.

- *Kolmogorov* prévoit que toute fonction continue de  $[0,1]^p$  dans  $[0,1]^n$  est approchable par un réseau multicouche constitué d'une première couche de  $p$  (neurones) d'une seconde de  $(2n+1)$  neurones et d'une couche de sortie de  $n$  neurones, chaque couche étant totalement connectée à la suivante.

-Alors que *Store-Weierstrass* montrent qu'un réseau de neurones possédant une fonction non linéaire dite saturante peut approximer toute fonction continue de  $[0,1]^p$  dans  $[0,1]^n$ .

Remarque: en pratique il ne faut pas se limiter systématiquement à 1 couche cachée. Certains problèmes peuvent être résolus par un petit réseau à 4 couches ( 2 couches cachées ) alors qu'il faut appeler à une infinité de neurones avec un réseau à 3 couches.



RESEAU  
A ARCHITECTURE  
EVOLUTIVE

CHAPITRE V**RESEAUX A ARCHITECTURE EVOLUTIVE [4]****1. INTRODUCTION**

Les propriétés d'approximateurs des perceptrons multicouches, quoique très intéressantes, ont une portée limitée en raison de l'indétermination qui subsiste sur le nombre de couches et le nombre de neurones par couche. Cette indétermination est d'autant plus grave qu'une sur-paramétrisation entraîne le phénomène de sur-apprentissage. L'idée séduisante d'ajuster, pendant l'apprentissage, la taille du réseau afin que sa complexité soit adaptée au problème à résoudre a conduit au développement de procédures d'élagage.

Dans ce chapitre on verra une approche basée sur ces principes appliquée aux perceptrons multicouches où on utilisera, pour le faire une méthode basée sur l'algorithme de *Reilley, Cooper et Erlbaum* (RCE) l'un des premiers modèles à architecture évolutive qui date de 1982.

L'idée de ces architectures incrémentales est assez naturelle compte tenu des problèmes de choix d'une architecture adaptée au problème à traiter.

On peut citer plusieurs méthodes qui traitent ce problème. a titre d'exemple, les 2 premières approches le "upstart algorithm" et le "Tiling algorithm".

Ce dernier, consiste, contrairement au premier, à ajouter des couches et non plus des neurones. Mais l'inconvénient de ces méthodes, c'est qu'elles sont très gourmandes en nombre de cellules, et ont tendance à apprendre par coeur, ce qui n'autorise pas d'avoir une bonne généralisation.

## 2. ALGORITHME A PLACEMENT DE CELLULES (RCE ALGORITHM)

### a. ARCHITECTURE DU RESEAU :

Les exemples de la base d'apprentissage sont des paires constituées de vecteurs d'entrée  $X^*$  à  $P$  composantes, et d'un vecteur classe  $Y^*$ . Le réseau est constitué de trois groupes de cellules (Fig 5.1) :

Les cellules d'entrée  $F$  qui reçoivent les entrées et les diffusent sans modification vers la couche suivante .

Les cellules de codage  $G$  qui codent les prototypes présentés par une affectation adéquate des coefficients des connexions entre les cellules  $F$  et  $G$ .

Initialement (avant apprentissage), le réseau ne comporte que des cellules  $F$  . Le principe de l'apprentissage incrémental consiste à ajouter des cellules  $G$  et  $H$  au cours de l'apprentissage et en cas de besoin uniquement.

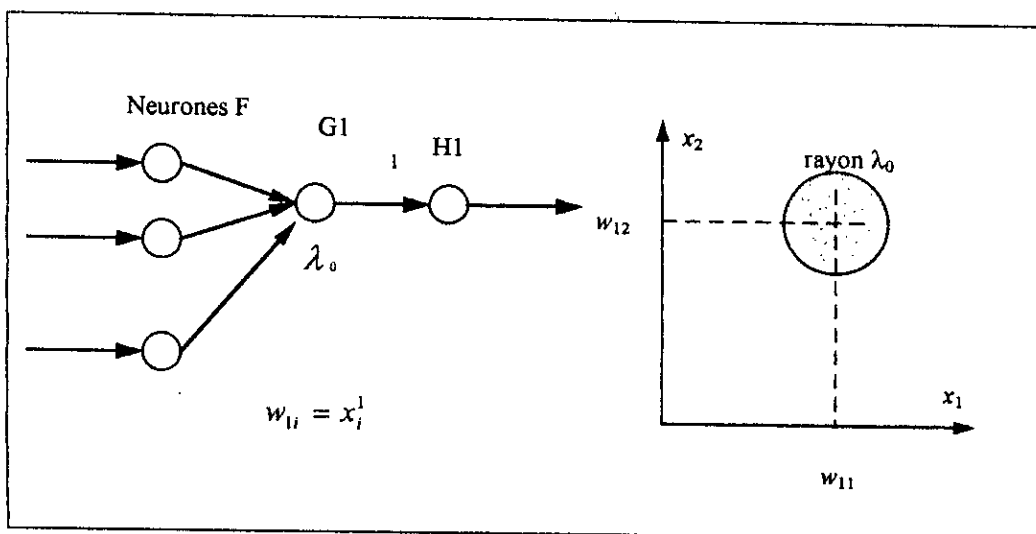


Fig 5.1. Architecture du réseau et zone d'influence dans le cas de vecteurs de dimensions 2.

### b. APPRENTISSAGE :

On présente le premier vecteur  $X^1$  qui appartient à la classe 1; puisque ce vecteur n'est pas codé et que la classe 1 n'est pas représentée, on doit créer une cellule de codage  $GI$  et une cellule de décision  $HI$  (Fig 5.1).

La cellule de décision de codage  $GI$  possède  $P$  poids  $W_{1i}$  (autant que  $X^1$  a de composantes) et l'on attribue :

$$W_{1i} = X_i^1 \quad (1)$$

Dans cet algorithme, un exemple de la base de données est appris par placement d'un neurone prototype. En posant  $W_1$  le vecteur poids de la cellule  $GI$ , la règle d'apprentissage s'écrit de façon vectorielle :

$$W_1 = X_1 \quad (2)$$

De plus, la cellule  $GI$  est caractérisée par le paramètre  $\lambda_0$ , qui est appelé "rayon d'influence". Si on présente un vecteur  $X$  à l'entrée du réseau, la sortie de la cellule  $GI$  vaut:

$$\begin{cases} 1, \text{ si } \text{dist}(X, W_1) \leq \lambda_0, \\ 0 \text{ sinon} \end{cases} \quad (3)$$

Dans notre cas, on utilisera la distance euclidienne, la sortie de  $GI$  vaut simplement :

$$\begin{cases} 1, \text{ si } \sum (x_i - w_{1i})^2 \leq \lambda_0, \\ 0 \text{ sinon} \end{cases} \quad (4)$$

La cellule  $HI$  associée à la classe 1, est connectée à la sortie de la cellule  $GI$ , par un poids de 1. la sortie d'un neurone de type  $H$  vaut 1 si la somme de ces entrées est supérieure ou égale à 1, et 0 dans le cas contraire. Lorsque la sortie de  $HI$  vaut 1 en réponse à un vecteur  $X$ , on décide que  $X$  appartient de la classe 1. De cette façon, le vecteur  $X^1$  est effectivement associé à la classe 1, de même que tous les vecteurs  $X$  dont la distance avec  $X^1$  est inférieure  $\lambda_0$ .

Présentons maintenant un deuxième vecteur  $X^2$ . Quatre situations peuvent se présenter

- 1)- Le vecteur  $X^2$ , qui appartient à la classe 1, active à 1 la sortie de  $G1$ . La sortie de  $H1$  vaut donc 1. La classification est correcte, on ne modifie pas le réseau.
- 2)- Le vecteur  $X^2$  appartient à la classe 1; mais n'active pas cellule  $G1$ . La sortie de  $H1$  est égale à 0. Le nouveau vecteur n'a pas été reconnu. Il faut rajouter une cellule de codage  $G2$  dont les poids avec les cellules de la couche d'entrée sont (en notation vectorielle) :

$$W_2 = X^2 \quad (5)$$

Le rayon d'influence associé à  $G2$  est  $\lambda_0$ . Sa sortie est connectée à  $H1$  avec un poids de 1 (fig 5. 2).

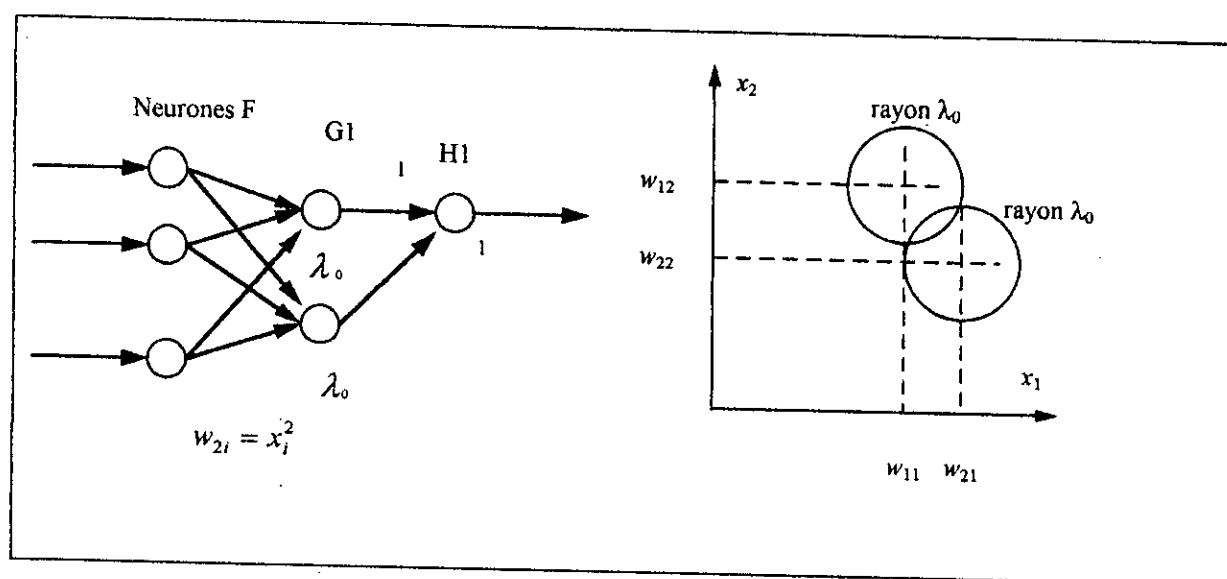


fig 5.2 : création d'une nouvelle cellule de codage et sa zone d'influence dans le cas de données à 2 dimensions.

- 3)- Le vecteur  $X^2$  appartient à la classe 2, la sortie de  $H1$  vaut 0. Le réseau ne sait pas classer ce vecteur. Il faut créer une cellule  $G2$  et une cellule  $H2$  (fig 5.3). les poids des connexions de  $G2$  avec les neurones  $F$  sont réglés selon la relation (n°5). Le rayon de



la zone d'influence de la cellule G2 vaut  $\lambda_0$ . sa sortie est reliée à H2 avec un poids de 1.

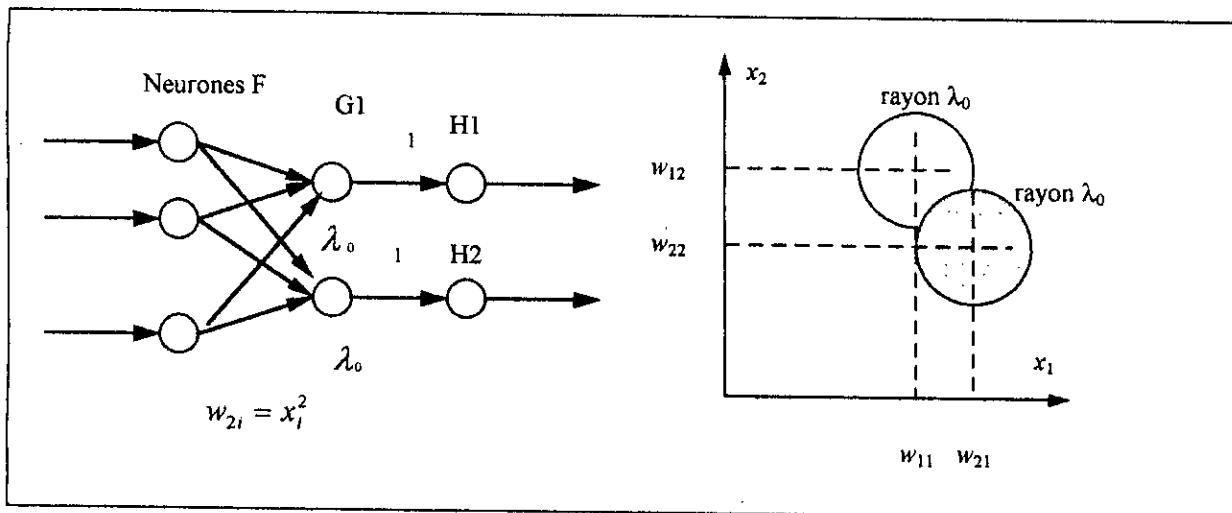


Fig 5.3 : Création d'une nouvelle classe par ajout de 2 cellules, G2 et H2.

4)- Le vecteur  $X^2$  appartient à la classe 2, mais active la sortie de G1 et celle de H1. Il est reconnu comme appartenant à la classe 1 : il y a confusion. Il faut donc coder  $X^2$  avec une cellule G2 et introduire H2 pour représenter la nouvelle classe. Mais il faut aussi limiter les rayons d'influence pour éviter la confusion (fig 5.4) :  $X^2$  ne devra plus activer ni G1 ni H1 après cette opération.

On crée une cellule G2 dont les poids sont réglés comme précédemment.

Les rayons d'influence de G1 et G2 (cellules de codages entraînant la confusion) sont réduits à :

$$\lambda_i < \text{dist}(X^2, W_i) \tag{6}$$

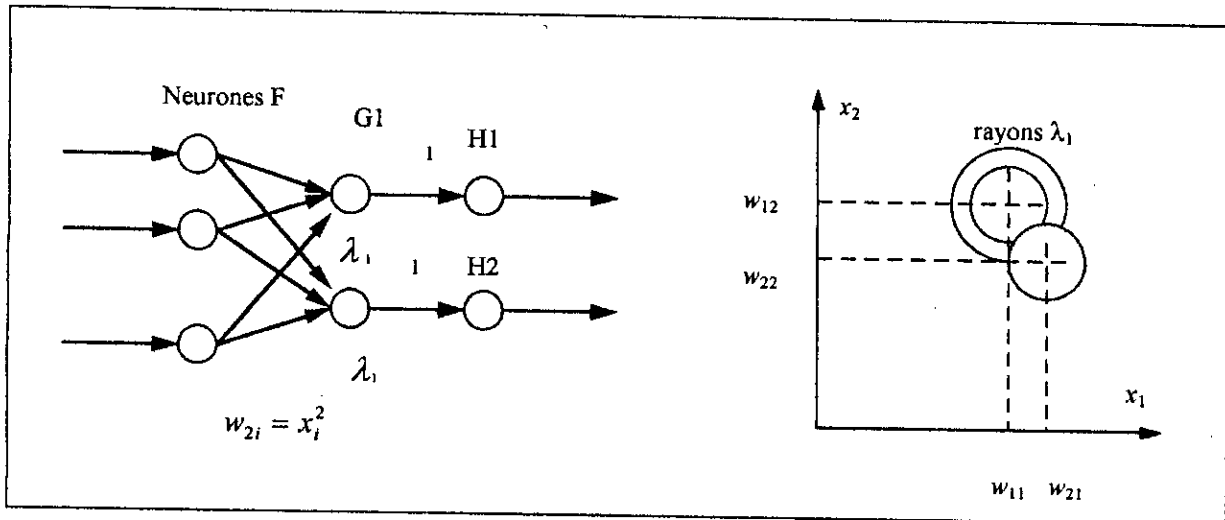


Fig 5.4 : Création d'une nouvelle cellule et réduction des rayons d'influence en cas de confusion

En cours d'apprentissage, dans le cas de confusion entre plusieurs cellules de codage, on doit ajuster le rayon d'influence de chaque prototype  $k$  qui entraîne une confusion avec le nouveau vecteur  $X^n$ , et du nouveau prototype  $n$ , selon les relations.

$$\lambda_k < \text{dist}(X^n, W_k), \text{ pour le prototype } k, \quad (n^\circ 7)$$

$$\lambda_n < \text{Min}_k [\text{dist}(X^n, W_k)], \text{ pour le nouveau prototype } n.$$

L'apprentissage se poursuit sur les vecteurs de la base d'apprentissage selon la même procédure. La présentation de la base d'apprentissage doit être répétée plusieurs fois, jusqu'à ce que qu'elle n'entraîne aucune modification : ni création de cellules, ni réduction de rayons d'influence.

En effet, chaque correction de confusion peut entraîner, lors d'une nouvelle présentation de la base d'apprentissage, la création de nouvelles cellules. Par exemple, soit un vecteur  $X^k$  se trouvant dans le rayon d'influence d'un prototype  $X^j$ , de sa classe, la classe  $X^k$  n'est pas directement codé. Si, ultérieurement par suite d'une confusion, on réduit le rayon d'influence de  $X^j$ , le vecteur  $X^k$  peut se trouver hors du nouveau rayon d'influence de  $X^j$ .

Dans le cas d'une distance euclidienne, cet algorithme modélise les classes elles-mêmes par la réunion d'hyperphères (cercle dans le cas de données en dimension 2, illustré

dans les figures 5.1 et 5.4) et non les frontières entre les classes comme dans les réseaux de type perception multicouches.

En pratique, les prototypes mémorisés (neurones de type G) dépendent de l'ordre des recteurs appris. L'apprentissage est rapide : le codage d'un prototype (et des prototypes lui ressemblant, situés dans son domaine d'influence) se fait en une fois. L'ajustement des rayons d'influence des neurones est effectué en un nombre réduit d'étapes jusqu'à ce qu'il n'y ait plus de confusion.

Un problème, relativement difficile, est celui du choix du paramètre initial : le rayon initial  $\lambda_0$ . Il est clair que la valeur de ce paramètre influe sur la résolution des classes et aussi sur le nombre de passes de la base d'apprentissage avant la convergence.

En pratique, on choisit  $\lambda_0$ , afin que le rayon initial des hypervolumes soit de l'ordre de 30% de l'espace occupé par les données.

Dans le cas de classes sans recouvrement, cet algorithme est très efficace. Cependant, dès que les classes se recouvrent, l'apprentissage ne peut pas être optimale (au sens de Bayes), car cet algorithme apprend par coeur des exemples de la base d'apprentissage. Autour de chaque point  $X^k$  de la base, l'algorithme crée une zone associée à la classe  $X^k$ , même si  $X^k$  se trouve dans une région où la probabilité d'appartenance à sa classe n'est pas la plus grande.

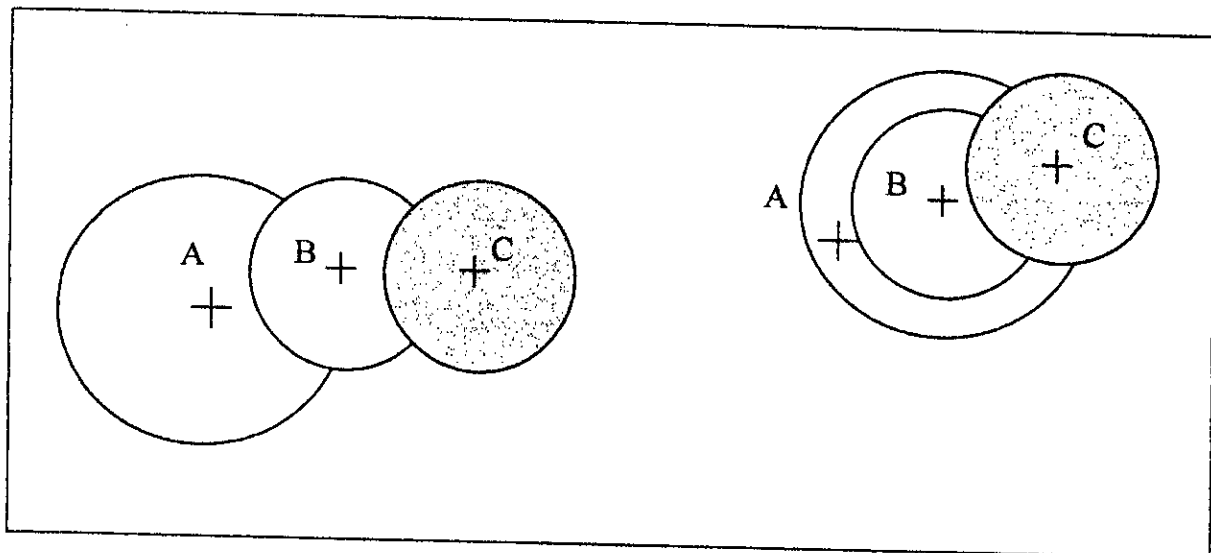
### **c. PARTICULARITE DE L'ALGORITHME :**

Cet algorithme est, comme la plupart des méthodes constructives, très "glouton". De même, le réseau tend à apprendre par coeur, ce qui nuit à une bonne généralisation du réseau.

L'algorithme du RCE étant très connu et utilisé, nous insistons dans ce paragraphe sur un problème particulier.

### 1. INFLUENCE DE L'ORDRE DES PROTOTYPES APPRIS :

Le nombre de prototypes créés dépend de l'ordre de présentation des vecteurs à apprendre. Examinons la *fig 5.5*, dans laquelle A et B sont deux points de la classe 1, et C un point de la classe 2.



*Fig 5.5. : Influence de l'ordre des exemples*

Si on apprend dans l'ordre (B,C,A), la confusion entre B et C oblige à réduire les rayons, et la présentation du prototype A génère une troisième cellule de codage G. En revanche, si l'on apprend la séquence (B,A,C), la présentation de A ne génère pas de cellule de codage car A se trouve dans le rayon d'influence de B. Après la présentation de C, le rayon de B est réduit, et le prototype A n'est finalement plus codé (*fig 5.5*). on verra ultérieurement l'illustration de cette méthode qui sera appliquée a un cas réel pour essayer d'en tirer plus d'informations sur cette dernière afin d'évaluer son applicabilité a la reconnaissance de signatures.

**APPLICATIONS,  
RESULTATS ET  
COMMENTAIRES**

CHAPITRE VI**APPLICATIONS, RESULTATS ET COMMENTAIRES****1. Introduction :**

Le besoin de vérifier l'identité d'un individu dans notre société a existé et existe toujours. Un des moyens usités à cette fin, qui est, sans doute, le plus répandu dans le monde, est la signature. C'est cette dernière qui est utilisée dans les transactions bancaires, et, en général, dans toutes les relations administratives.

En effet, c'est la procédure la plus révélatrice de l'identité, car elle présente l'avantage que ne possédant pas simultanément les autres méthodes d'identification (mot de passe, codes numériques, cartes, clés,...).

C'est l'avantage de représenter les falsifications; chose qui n'est pas offerte par les méthodes suscitées.

**2. Pourquoi les moments invariants : [5] [6] [7]**

Afin d'extraire les information utiles pour l'études, les signataires sont numérisées "scannées" avec une résolution de 150 dpi, elles sont sorties en fichier image TIFF.

Utiliser directement la matrice image contenue dans le fichier TIFF serait du "Don Quichottisme", car l'image contient un nombre  $n$  pixels, et utiliser des réseaux de neurones avec une couche d'entrée de  $n$  neurones demanderait des temps de calculs énormes.

Il existe un moyen intrinsèque donnant les caractéristiques d'une image, même si cette dernière subit des transformations (non des déformations). Ce sont "les moment invariants".

Ainsi, on utilisera que 7 nombres (moments) pour représenter une signature, plutôt que d'avoir à manipuler des vecteurs de  $n$  composantes. Les moments invariants sont des fonctions non linéaires qui prennent en compte l'organisation interne de la forme du caractère, ils sont invariants à la translation au changement d'échelle et à la rotation de l'objet d'une image donnée.

Un moment élémentaire  $M_{pq}$ , de l'image  $f$  du caractère est défini par :

$$M_{pq} = \sum_{i=1}^n \sum_{j=1}^m f(i,j) i^p j^q \quad (1)$$

où  $N$  et  $M$  représentent le nombre de pixels de l'image  $N \times M$ .

On utilise souvent les sept premiers moments correspondant aux valeurs  $p \leq 3$  et  $q \leq 3$ .

Le moment d'ordre 0,  $M_{00}$  représente la surface de l'image, définissent le centre de gravité  $(\bar{X}, \bar{Y})$  de la surface précédente.

$$\bar{X} = \frac{M_{10}}{M_{00}}, \quad \bar{Y} = \frac{M_{01}}{M_{00}} \quad (2)$$

Pour obtenir l'invariance de l'objet en translation, on définit les moments centrés d'ordre  $(p,q)$  qui sont définis par :

$$\mu_{pq} = \sum_{i=1}^n \sum_{j=1}^m (i - \bar{X})^p (j - \bar{Y})^q f(i,j) \quad (3)$$

Les moment centrés d'ordre 2,  $\mu_{20}$ ,  $\mu_{02}$ ,  $\mu_{11}$  permettent de trouver les axes principaux d'inertie et donc les allongements et les orientations de l'image.

Pour l'invariance en changement d'échelle, on prendra les moments normalisés:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \text{avec } \gamma = \left[ \frac{(p+q)}{2} \right] + 1 \quad (5)$$

Enfin, pour obtenir l'invariance en translation, rotation et changement d'échelle, nous utilisons la série de fonctions non linéaires définies sur  $\eta_{p,q}$  :

$$\phi_1 = \eta_{2,0} + \eta_{0,2} .$$

$$\phi_2 = (\eta_{2,0} - \eta_{0,2})^2 + 4 \eta_{1,1}^2 .$$

$$\phi_3 = (\eta_{3,0} - 3\eta_{1,2})^2 + (3\eta_{2,1} - \eta_{0,3})^2 .$$

$$\phi_4 = (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{2,1} + \eta_{0,3})^2 .$$

$$\phi_5 = (\eta_{3,0} - 3\eta_{1,2}) (\eta_{3,0} + \eta_{1,2}) [(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2] + (3\eta_{2,1} - \eta_{0,3}) \cdot (\eta_{2,1} + \eta_{0,3}) [3 \cdot (\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] .$$

$$\phi_6 = (\eta_{2,0} - \eta_{0,2}) [(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] + 4 \eta_{1,1} \cdot (\eta_{3,0} + \eta_{1,2}) (\eta_{2,1} + \eta_{0,3})$$

$$\phi_7 = (3\eta_{2,1} - \eta_{0,3}) (\eta_{3,0} + \eta_{1,2}) [(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2] + (3\eta_{1,2} - \eta_{3,0}) \cdot (\eta_{2,1} + \eta_{0,3}) [3 \cdot (\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] .$$

### 3. Choix des bases d'apprentissage et de test :

Les signatures ont été choisies de manière à ce qu'elles représentent pour le mieux la classe à laquelle elles appartiennent. On a opté pour 3 signatures par classe, ceci concerne la base d'apprentissage. Pour ce qui en est de la base de test, là on a plutôt choisi des signatures assez différentes de celles apprises, pour évaluer les capacités des réseaux utilisés pour reconnaître ou non les stimuli qui leurs seront présentés.

Les signatures sont dans l'annexe A.

**Remarque** : Les signatures sont de tailles différentes; et cela n'influe aucunement sur la nature des résultats car les moments ne sont pas affectés par le changement d'échelle.



## 4. UTILISATION DE LA METHODE DE BACK-PROPAGATION

### a. DESCRIPTION DU RESEAU

Les paramètres du réseau de BACK-PROPAGATION pour la reconnaissance de chaque signature sont données comme suit:

Nombre de couches	3
Nombre de neurones de la couche d'entrée	3 X 7
Nombre de neurones de la couche cachée	3
Nombre de neurones de la couche de sortie	1
Entrée totale	Somme des entrées pondérées par les poids des connexions
Fonction d'activation	tangente hyperbolique

TABLEAU 6.1

### b. CHOIX DES ENTREES

Au début on a fait un seul test qui applique a l'entrée du réseau les valeurs de  $\phi_i$  normalisées.

$$\phi_i = \frac{\text{Log}|\phi_i|}{\left[ \sum (\text{Log}^2|\phi_i|) \right]^{1/2}} \quad (6)$$

Après on a vu qu'il était préférable de refaire le travail avec les entrées réelles  $\phi_i$  (moments invariants).

**c. APPRENTISSAGE ET RESULTATS**

On a pris comme pas d'adaptation  $\eta = 0,01$  pour tous les essais et comme test d'arrêt la minimisation de l'erreur quadratique.

**1. Résultats avec les entrées normalisées  $\phi'_i$ :**

CLASSE 1:

base d'apprentissage	Sig 1	Sig 2	Sig 3
erreur quadratique	$1,391901 \cdot 10^{-5}$	$1,681768 \cdot 10^{-5}$	$1,916609 \cdot 10^{-5}$

TABLEAU 5.2

CLASSE 2:

base d'apprentissage	Sign 1	Sign 2	Sign 3
erreur quadratique	$1,145014 \cdot 10^{-3}$	$1,888334 \cdot 10^{-3}$	$1,032644 \cdot 10^{-3}$

TABLEAU 5.3

CLASSE 3:

base d'apprentissage	Siga 1	Siga 2	Siga 3
erreur quadratique	$1,387111 \cdot 10^{-3}$	$1,391479 \cdot 10^{-3}$	$1,388082 \cdot 10^{-3}$

TABLEAU 5.4

L'apprentissage de chaque série a été effectué entre 60 minutes et 180 minutes; au total entre 20.000 et 60.000 itérations.

**1) Résultats des tests:**

On a fait une désignation qui caractérise chaque nature de signature, qu'on prendra comme légende au cours des tests:

O : Signature de la même classe.

N: Signature différente de la classe.

M: Signature incomplète.

## CLASSE 1:

Signature	Désignation	Erreur quadratique
Sigat 1	O	$1,702803 \cdot 10^{-3}$
Sigat 2	M	$1,760338 \cdot 10^{-3}$
Siga 2	N	$1,869434 \cdot 10^{-3}$
Sign 3	N	$1,678807 \cdot 10^{-3}$
Sigat 5	M	$1,714958 \cdot 10^{-3}$
Sigat 6	M	$1,702858 \cdot 10^{-3}$
Sig 1	O	$1,391868 \cdot 10^{-3}$
Siga 8	N	$1,307829 \cdot 10^{-3}$

TABLEAU 6.5

## CLASSE 2:

Signature	Désignation	Erreur quadratique
Siga 1	N	$1,697497 \cdot 10^{-3}$
Sig 2	N	$2,816475 \cdot 10^{-4}$
Sign 1a	O	$1,430275 \cdot 10^{-3}$
Sign 1b	O	$1,262359 \cdot 10^{-3}$
Siga 7	N	$8,080914 \cdot 10^{-4}$
Sign 1	O	$1,142403 \cdot 10^{-3}$

TABLEAU 6.6

## CLASSE 3:

Signature	Désignation	Erreur quadratique
Siga 4	M	$1,384459 \cdot 10^{-3}$
Siga 5	O	$1,385654 \cdot 10^{-3}$
Siga 6	O	$1,384758 \cdot 10^{-3}$
Sigat 3	M	$1,381896 \cdot 10^{-3}$
Siga 8	N	$1,37739 \cdot 10^{-3}$
Sign 1	N	$1,382107 \cdot 10^{-3}$
Sig 1	N	$1,380927 \cdot 10^{-3}$
Siga 1	O	$1,387002 \cdot 10^{-3}$

TABLEAU 6...6

**d. INTERPRETATION DES RESULTATS**

Comme on peut le constater les erreurs quadratiques sont très proches alors qu'elles devraient être différentes car le test s'est fait sur des signatures différentes par exemple:

Dans la classe 3, il faut commencer l'analyse dès le troisième nombre après la virgule;  
 Dans la classe 2 le réseau accepte Siga 1 alors qu'il fallait la rejeter;  
 Dans la classe 1 toutes les erreurs quadratiques sont de même ordre ce qui rend notre analyse impossible.

L'utilisation des moments normalisés comme base d'apprentissage a fait que les moments de toutes les signatures soient approchés (Annexe A), ce qui prouve la nécessité de faire un bon choix de la nature des entrées.

**Résultats avec les entrées réelles****CLASSE 1**

Base d'apprentissage	Sig 1	Sig 2	Sig 3
Erreur quadratique	$5,51583 \cdot 10^{-6}$	$2,044986 \cdot 10^{-14}$	$1,23503 \cdot 10^{-2}$

TABLEAU 1.8

**CLASSE 2**

Base d'apprentissage	sign 1	sign 2	Sign 3
Erreur quadratique	$7,500248 \cdot 10^{-2}$	$4,496712 \cdot 10^{-2}$	$5,009312 \cdot 10^{-2}$

TABLEAU 1.9

**CLASSE 3**

Base d'apprentissage	Siga 1	Siga 2	Siga 3
Erreur quadratique	$5,039218 \cdot 10^{-7}$	$8,868978 \cdot 10^{-8}$	$2,329764 \cdot 10^{-7}$

TABLEAU 1.10

**REMARQUE:**

L'apprentissage s'est fait de la même manière que précédemment.

**e. LES RESULTATS DES TESTS**

CLASSE 1:

Signature	désignation	Erreur quadratique
Sigat 1	O	$1,212560 \cdot 10^{-1}$
Sigat 2	M	$1,934499 \cdot 10^{-2}$
Siga 2	N	$1,228161 \cdot 10^{-2}$
Sign 3	N	$4,501268 \cdot 10^{-1}$
Sigat 5	M	$8,625630 \cdot 10^{-2}$
Sigat 6	M	$1,21256 \cdot 10^{-1}$
Sig 1	O	$5,511573 \cdot 10^{-6}$
Siga 8	N	$1,604957 \cdot 10^{-13}$
Siga 7	N	$4,312747 \cdot 10^{-2}$

TABLEAU 6...11

CLASSE 2:

Signature	désignation	Erreur quadratique
Siga 1	N	$1,893434 \cdot 10^{-22}$
Sign 2	O	$4,496339 \cdot 10^{-2}$
Sign 1a	O	$4,086131 \cdot 10^{-2}$
Sign 1b	O	$4,751780 \cdot 10^{-2}$
Siga 7	N	$1,150159 \cdot 10^{-25}$
Sign 1	O	$3,814753 \cdot 10^{-2}$
Sign 3	O	$5,009112 \cdot 10^{-2}$
Siga 2	N	$5,511573 \cdot 10^{-6}$
Sigat 2	N	$5,546678 \cdot 10^{-32}$
Sigat 5	N	$1,306232 \cdot 10^{-19}$
Sigat 6	N	$2,496656 \cdot 10^{-16}$

TABLEAU 6.12



## CLASSE 3:

Signature	désignation	Erreur quadratique
Sigat 1	N	$9,408614 \cdot 10^{-4}$
Siga 5	O	$2,257608 \cdot 10^{-8}$
Siga 6	O	$8,613623 \cdot 10^{-8}$
Sigat 3	M	$1,044810 \cdot 10^{-7}$
Sign 2	N	$3,375049 \cdot 10^{-1}$
Siga 7	N	$1,113101 \cdot 10^{-5}$
Siga 3	O	$2,329777 \cdot 10^{-7}$
Siga 2	O	$8,869045 \cdot 10^{-8}$
Sign 1	N	$3,296274 \cdot 10^{-1}$
Sig 1	N	$2,480833 \cdot 10^{-1}$
Siga 1	O	$5,039204 \cdot 10^{-7}$

TABLEAU 6...13

## 5. INTERPRETATION DES RESULTATS

On peut déjà constater dans la classe 1 que les erreurs quadratiques de la base d'apprentissage sont très différentes d'ordre  $10^{-2}$ ,  $10^{-14}$ ,  $10^{-6}$  ce qui prouve que l'on a pris une mauvaise base cela est dû à la mauvaise prise des signatures.

On peut le remarquer sans faire d'apprentissage à travers les valeurs des moments : qui sont complètement différents même les résultats du test prouvent cela. Tableau 6.11.

Les tests sur les classes 2,3 ont donné de bons résultats mais cela n'empêche pas de remarquer qu'il est important que le test d'arrêt qui minimise l'erreur quadratique soit très petit.

L'erreur quadratique de la classe 2 est de l'ordre de  $10^{-2}$ ; le réseau accepte les signatures dont l'erreur est de même ordre et rejette celles qui sont différentes mais le mauvais choix du test d'arrêt fait que les signatures rejetées sont d'un ordre moins grand. *Tableau 6.12.*

Alors que pour la classe 3 on a pris un test d'arrêt qui minimise l'erreur quadratique beaucoup plus précis d'ordre  $10^{-7}$ ,  $10^{-8}$ , le réseau accepte les signatures dont l'erreur est de même ordre sous réserve de signatures incomplètes sigat 3; et rejette les signatures dont l'erreur quadratique est d'ordre supérieur. *Tableau 6.13.*

### COMMENTAIRE/

- L'utilisation des moments normalisés ( $\phi_i$ ) comme entrée a fait que le réseau de rétropropagation ne répond pas aux exigences attendues. Pour corriger ça, on les a remplacé par les moments invariants ( $\phi_i$ ) qui ont donné de bons résultats.

- La mauvaise prise des signatures (coupure) a fait que la classe 1 ait une mauvaise base d'apprentissage ce qui inclue de mauvais résultats, cela prouve l'influence de la nature de la base d'apprentissage sur le réseau..

- Il faut choisir un test d'arrêt qui minimise l'erreur quadratique très petit comme on l'a remarqué pour la classe 2 et la classe 3.

### a. Apprentissage du RCE

Les signatures étant représentées par des vecteurs de 7 composantes chacun, alors notre réseau initial ne comporte que les 7 neurones de la couche d'entrée. Les couches cachées et de sortie ne comportent aucun neurone. Les exemples d'apprentissage sont présentés un par un en entrée et le réseau se construit au fur et à mesure.

Comme on l'a précisé au chapitre V, l'ordre de présentation des exemples d'apprentissage a une influence sur la taille du réseau (voir Tableau 6.14).

Ordre des exemples	Réseau final
sig1-sig2-sig3	7 cellules f
sign1-sign2-sign3	9 cellules g
sigal-siga2-siga3	3cellules h
sig1-sig2-sig3	7 cellules f
sign3-sign2-sign1	8 cellules g
sigal-siga2-siga3	3 cellules h

TABLEAU 6.14

L'apprentissage est effectué sur un P.C équipé d'un processeur Pentium-75mhz. Les programmes sont écrits en langage C++(voir annexe B). Le temps d'apprentissage est très court (de l'ordre de 4 secondes).

Le temps de réponse du réseau est de l'ordre de la seconde.

Les résultats du test sont présentés dans le tableau 6.15 suivant les valeurs de  $\lambda$  (rayon d'influence initial), on présente pour chaque signature de la base de test, les classes activées.

**b. Résultats et commentaire:**

Classes	1	2		3						
$\lambda$	Sigat 1	Sign 1a	Sign 1b	sigat 5	sigat 6	sigat 2	sigat 3	sigat 4	sigat 7	sigat 8
0,04	H <sub>1</sub>	H <sub>2</sub> ,H <sub>3</sub>	H <sub>2</sub> ,H <sub>2</sub> ,H 3	H <sub>1</sub> ,H <sub>3</sub>	H <sub>2</sub> ,H <sub>3</sub>	rejetée	H <sub>1</sub> ,H <sub>2</sub> ,H 3	H <sub>1</sub> ,H <sub>2</sub> ,H 3	rejetée	rejetée
0,01	H <sub>1</sub>	H <sub>2</sub> ,H <sub>3</sub>	H <sub>2</sub> ,H <sub>2</sub> ,H 3	H <sub>1</sub> ,H <sub>3</sub>	H <sub>2</sub> ,H <sub>3</sub>	rejetée	H <sub>1</sub> ,H <sub>2</sub> ,H 3	H <sub>1</sub> ,H <sub>2</sub> ,H 3	rejetée	rejetée
0,005	H <sub>1</sub>	H <sub>2</sub> ,H <sub>3</sub>	H <sub>2</sub> ,H <sub>2</sub> ,H 3	H <sub>1</sub> ,H <sub>3</sub>	H <sub>2</sub> ,H <sub>3</sub>	rejetée	H <sub>1</sub> ,H <sub>2</sub> ,H 3	H <sub>1</sub> ,H <sub>2</sub> ,H 3	rejetée	rejetée
0,003	H <sub>1</sub>	H <sub>2</sub> ,H <sub>3</sub>	H <sub>2</sub> ,H <sub>2</sub> ,H 3	H <sub>3</sub>	H <sub>2</sub> ,H <sub>3</sub>	rejetée	H <sub>1</sub> ,H <sub>2</sub> ,H 3	H <sub>2</sub> ,H <sub>3</sub> ,H <sub>3</sub>	rejetée	rejetée
0,002	rejetée	H <sub>2</sub> ,H <sub>3</sub>	H <sub>2</sub> ,H <sub>2</sub> ,H 3	H <sub>3</sub>	H <sub>2</sub> ,H <sub>3</sub>	rejetée	H <sub>1</sub> ,H <sub>2</sub> ,H 3	H <sub>2</sub> ,H <sub>3</sub> ,H <sub>3</sub>	rejetée	rejetée

TABLEAU 6.15

D'après le tableau ci dessous, on constate la forte confusion qui existe entre les classes 2 et 3; par contre celle qui existe entre la classe 1 et 3, peut être surmonter par un ajustement du paramètre  $\lambda$ , a condition de ne pas entraîner le rejet de signatures telle que sigat 1.

Les signatures sigat 7 et sigat 8 n'appartiennent a aucune classe de la base d'apprentissage; elles sont carrément rejetées par le réseau. La signature sigat 2 est une version d'une des signatures de la classe 1, mais incomplète et comme on le constate le réseau la rejette. Pour ce qui est de sigat 3 et sigat 4; ce sont des tentatives de reproduction on remarque que le réseau ne parvient pas à les classer.

En effet, l'inconvénient de cette méthode c'est que les résultats ne sont pas très satisfaisants dès qu'il s'agit de recouvrement entre les classes. Il pourrait être appliqué avec succès pour une classification sans recouvrement de classes. Une manière

d'augmenter la résolution du réseau serait d'augmenter le nombre d'exemples d'apprentissage par classe.

**CONCLUSION  
GENERALE**

## CONCLUSION GENERALE

L'objectif principal de ce travail, était d'évaluer l'aptitude des réseaux de neurones à reconnaître des signatures. Deux heuristiques ont été utilisées à cette fin.

La première est l'application de l'Algorithme de rétropropagation de gradient a un perceptron multi-couches fixe. La seconde, quant à elle, est basée sur la principe des perceptrons à architecture évolutive.

Nous avons vu que la deuxième présentait par rapport à la première l'avantage d'être rapide, de ne pas nécessiter le choix de poids initiaux et de n'utiliser qu'un seul paramètre.

Cependant son efficacité se dégrade pour des classes qui se recouvrent. La nécessité d'avoir plus d'informations (élargir la base d'apprentissage) a été souligné aussi, sans oublier le choix de la nature des entrées.

La reconnaissance des signatures est un domaine vierge; les chemins à suivre sont très nombreux et le manque d'informations (de techniques) risquerait de dérouter celui qui s'aventure sans, au préalable, avoir étudié à fond le problème, elle n'est que le début d'une génération de recherches qui parviendront un jour, peut être, à apporter plus de bases théoriques afin de développer des techniques puissantes pour la reconnaissance de textes manuscrits.

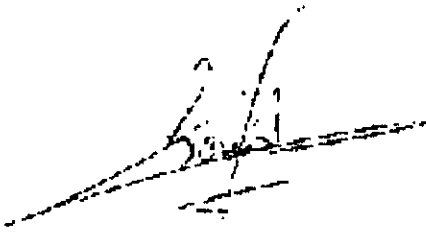
# BIBLIOGRAPHIE



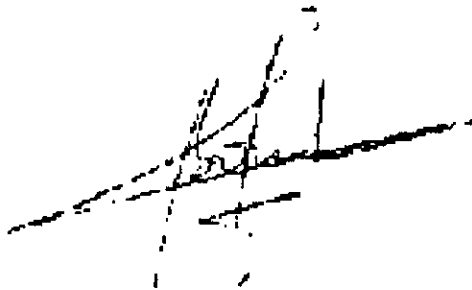
**BIBLIOGRAPHIE**

- [1]. E.DAVALO, P.NAÏM, "Des réseaux de neurones" Eyrolles, 1990.
- [2]. J.A.FREEMAN, D.M.SKAPURA, "Neural Networks. Algorithms, Applications and programming techniques "Addison-Wesley. Publishing Company, 1992.
- [3]. GONZALES, "Digital Image processing".
- [4]. J.HERAULT, C.JUTTEN, "Réseaux neuronaux et traitement du signal". Traité des nouvelles technologies. HERMES, 1994.
- [5]. M.KUNT, G.GRANLUND, M.KOCHER, "Traitement numérique des images". Collection électricité. Trait de l'information, Vol 02, Press Polytechniques Romandes, 1994.
- [6]. N.KHORISSI, A.GUESSOUM, H.MELIANI, "Simulation d'un réseau de neurones appliqué à la reconnaissance des caractères arabes", Proceedings du premier Séminaire national, Blida, Alger, Déc 1992.
- [7]. C.H.TCH, R.T.CHIN, "On image analysis by the méthodes of moments", IEEE Trans, Patern anal, Machine intell, Vol 10, July 1988.
- [8]. A.KHOTANZAD, J.H.LU, "Distorsion invariant character recognition by a MLP and backpropagation learning", in Prol IEEE 2nd. Int.conf.neural networks (SanDiègo, CA), pp625-632, July 1988.
- [9]. L.MICLET, "Méthodes structurelles pour la reconnaissance des formes". Eyrolles, 1990.

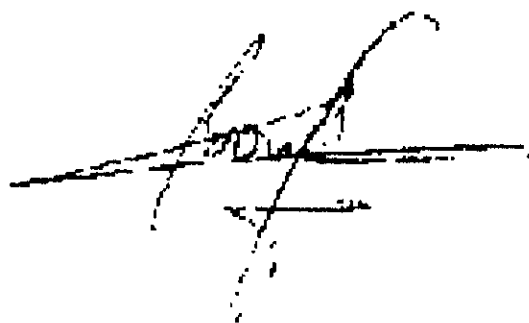
ANNEXE -A-

**Sig 1**

moments reels	moments normalisees
$4.417176 \cdot 10^{-2}$	$-8.334589 \cdot 10^{-2}$
$1.942185 \cdot 10^{-3}$	$-1.668147 \cdot 10^{-1}$
$8.584355 \cdot 10^{-5}$	$-2.501439 \cdot 10^{-1}$
$8.706023 \cdot 10^{-5}$	$-2.497679 \cdot 10^{-1}$
$7.526333 \cdot 10^{-9}$	$-4.997238 \cdot 10^{-1}$
$3.835920 \cdot 10^{-6}$	$-3.331811 \cdot 10^{-1}$
$-5.534475 \cdot 10^{-12}$	$-6.924860 \cdot 10^{-1}$

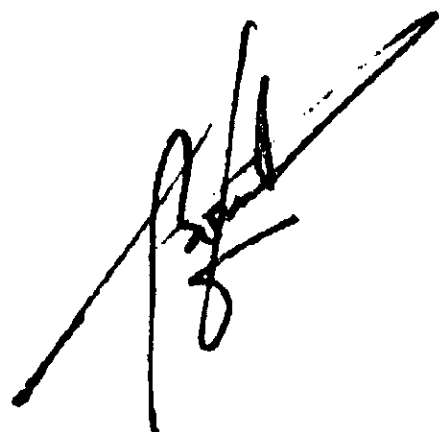
**Sig 3**

moments reels	moments normalisees
$1.643466 \cdot 10^{-2}$	$-7.360101 \cdot 10^{-2}$
$1.703221 \cdot 10^{-4}$	$-1.554625 \cdot 10^{-1}$
$8.071147 \cdot 10^{-7}$	$-2.513428 \cdot 10^{-1}$
$1.001644 \cdot 10^{-7}$	$-2.887251 \cdot 10^{-1}$
$-3.424049 \cdot 10^{-10}$	$-5.967095 \cdot 10^{-1}$
$-2.765451 \cdot 10^{-10}$	$-3.942833 \cdot 10^{-1}$
$-2.827328 \cdot 10^{-14}$	$-5.588894 \cdot 10^{-1}$



Sig 2

moments reels	moments normalisees
$1.138906 \cdot 10^{-1}$	$-1.112764 \cdot 10^{-1}$
$1.289491 \cdot 10^{-2}$	$-2.228545 \cdot 10^{-1}$
$1.097335 \cdot 10^{-2}$	$-2.311195 \cdot 10^{-1}$
$4.607183 \cdot 10^{-3}$	$-2.755710 \cdot 10^{-1}$
$3.091334 \cdot 10^{-5}$	$-5.318856 \cdot 10^{-1}$
$2.369903 \cdot 10^{-4}$	$-4.275590 \cdot 10^{-1}$
$-1.083877 \cdot 10^{-5}$	$-5.855672 \cdot 10^{-1}$



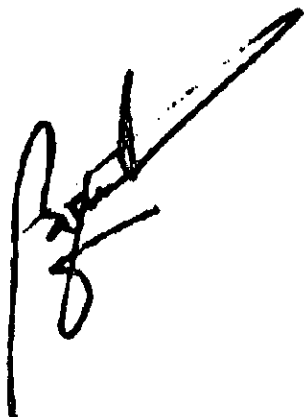
Sigat 1

moments reels	moments normalisees
$7.414774 \cdot 10^{-3}$	$-8.288145 \cdot 10^{-2}$
$2.030619 \cdot 10^{-5}$	$-1.825955 \cdot 10^{-1}$
$4.627866 \cdot 10^{-7}$	$-2.465007 \cdot 10^{-1}$
$3.424851 \cdot 10^{-8}$	$-2.905615 \cdot 10^{-1}$
$-4.284136 \cdot 10^{-15}$	$-5.591112 \cdot 10^{-1}$
$-1.404550 \cdot 10^{-10}$	$-3.833915 \cdot 10^{-1}$
$4.871357 \cdot 10^{-16}$	$-5.958536 \cdot 10^{-1}$



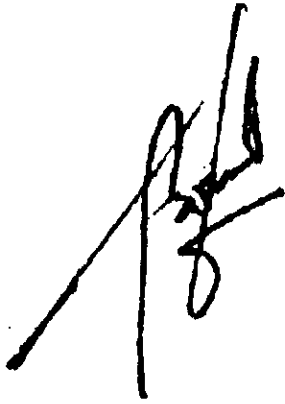
### Sigat 2

moments reels	moments normalises
$1.475537 \cdot 10^{-2}$	$-9.332556 \cdot 10^{-2}$
$1.306609 \cdot 10^{-4}$	$-1.979536 \cdot 10^{-1}$
$1.947962 \cdot 10^{-5}$	$-2.400822 \cdot 10^{-1}$
$1.911832 \cdot 10^{-6}$	$-2.914650 \cdot 10^{-1}$
$-1.123430 \cdot 10^{-11}$	$-5.580754 \cdot 10^{-1}$
$-1.952717 \cdot 10^{-8}$	$-3.929333 \cdot 10^{-1}$
$3.148473 \cdot 10^{-12}$	$-5.862327 \cdot 10^{-1}$



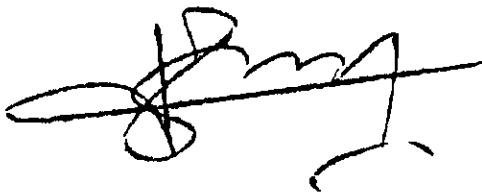
### Sigat 5

moments reels	moments normalisees
$8.911126 \cdot 10^{-3}$	$-7.559416 \cdot 10^{-2}$
$6.655282 \cdot 10^{-5}$	$-1.540165 \cdot 10^{-1}$
$1.015594 \cdot 10^{-7}$	$-2.578701 \cdot 10^{-1}$
$9.189625 \cdot 10^{-9}$	$-2.963452 \cdot 10^{-1}$
$-2.562475 \cdot 10^{-16}$	$-5.749149 \cdot 10^{-1}$
$-6.994428 \cdot 10^{-11}$	$-3.744645 \cdot 10^{-1}$
$1.146868 \cdot 10^{-16}$	$-5.877893 \cdot 10^{-1}$



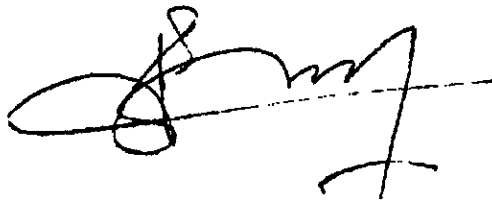
### Sigat 6

moments reels	moments normalises
$7.414774 \cdot 10^{-3}$	$-8.288145 \cdot 10^{-2}$
$2.030619 \cdot 10^{-5}$	$-1.825955 \cdot 10^{-1}$
$4.627866 \cdot 10^{-7}$	$-2.465007 \cdot 10^{-1}$
$3.424851 \cdot 10^{-8}$	$-2.905015 \cdot 10^{-1}$
$-4.284136 \cdot 10^{-15}$	$-5.591112 \cdot 10^{-1}$
$-1.404550 \cdot 10^{-10}$	$-3.833915 \cdot 10^{-1}$
$4.871357 \cdot 10^{-16}$	$-5.958536 \cdot 10^{-1}$

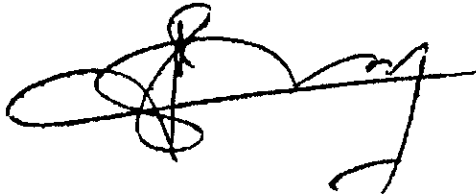


### Siga 1


moments reels	moments normalises
$1.031645 \cdot 10^{-2}$	$-8.058773 \cdot 10^{-2}$
$1.725686 \cdot 10^{-6}$	$-2.337967 \cdot 10^{-1}$
$2.278374 \cdot 10^{-7}$	$-2.694699 \cdot 10^{-1}$
$1.261696 \cdot 10^{-7}$	$-2.798826 \cdot 10^{-1}$
$1.370291 \cdot 10^{14}$	$-5.624061 \cdot 10^{-1}$
$1.495304 \cdot 10^{-10}$	$-3.985947 \cdot 10^{-1}$
$-1.642661 \cdot 10^{-140}$	$-5.592120 \cdot 10^{-1}$


**Siga 2**

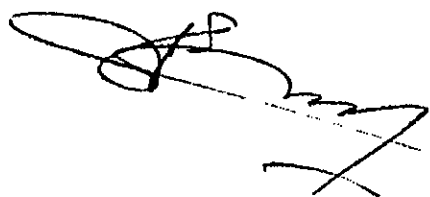
moments reels	moments normalises
$1.643264 \cdot 10^{-2}$	$-7.140077 \cdot 10^{-2}$
$1.454506 \cdot 10^{-4}$	$-1.535539 \cdot 10^{-1}$
$5.403929 \cdot 10^{-7}$	$-2.507937 \cdot 10^{-1}$
$3.591308 \cdot 10^{-8}$	$-2.979111 \cdot 10^{-1}$
$1.597844 \cdot 10^{-15}$	$-5.920996 \cdot 10^{-1}$
$4.28325 \cdot 10^{-10}$	$-3.748816 \cdot 10^{-1}$
$-4.741022 \cdot 10^{-15}$	$-5.731984 \cdot 10^{-1}$


**Siga 3**

moments reels	moments normalises
$1.552412 \cdot 10^{-2}$	$-7.715446 \cdot 10^{-2}$
$8.810527 \cdot 10^{-3}$	$-1.729477 \cdot 10^{-1}$
$6.207765 \cdot 10^{-7}$	$-2.647344 \cdot 10^{-1}$
$1.833658 \cdot 10^{-7}$	$-2.873228 \cdot 10^{-1}$
$4.936499 \cdot 10^{-14}$	$-5.675323 \cdot 10^{-1}$
$3.809999 \cdot 10^{10}$	$-4.017283 \cdot 10^{-1}$
$-3.728779 \cdot 10^{-14}$	$-5.727294 \cdot 10^{-1}$

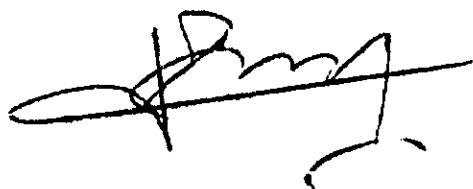

**Siga 5**

moments reels	moments normalises
$6.532960 \cdot 10^{-3}$	$-1.013872 \cdot 10^{-1}$
$2.243129 \cdot 10^{-5}$	$-2.157381 \cdot 10^{-1}$
$3.421203 \cdot 10^{-6}$	$-2.536350 \cdot 10^{-1}$
$7.040046 \cdot 10^{-7}$	$-2.854960 \cdot 10^{-1}$
$6.246262 \cdot 10^{-13}$	$-5.663298 \cdot 10^{-1}$
$1.866325 \cdot 10^{-9}$	$-4.050595 \cdot 10^{-1}$
$-8.964207 \cdot 10^{-13}$	$-5.590495 \cdot 10^{-1}$


**Siga 6**

moments reels	moments normalises
$1.645902 \cdot 10^{-2}$	$-8.82385 \cdot 10^{-2}$
$1.490224 \cdot 10^{-4}$	$-1.893178 \cdot 10^{-1}$
$3.286398 \cdot 10^{-6}$	$-2.712702 \cdot 10^{-1}$
$2.075215 \cdot 10^{-6}$	$-2.811476 \cdot 10^{-1}$
$5.308376 \cdot 10^{-12}$	$-5.577610 \cdot 10^{-1}$
$2.530837 \cdot 10^{-8}$	$-3.758276 \cdot 10^{-1}$
$-1.041758 \cdot 10^{-12}$	$-5.927879 \cdot 10^{-1}$




**Sigat 3**

moments reels	moments normalises
$1.633511 \cdot 10^{-2}$	$-8.678733 \cdot 10^{-2}$
$3.841573 \cdot 10^{-5}$	$-2.144571 \cdot 10^{-1}$
$5.959566 \cdot 10^{-6}$	$-2.537639 \cdot 10^{-1}$
$1.249423 \cdot 10^{-6}$	$-2.867184 \cdot 10^{-1}$
$2.175852 \cdot 10^{-12}$	$-5.664327 \cdot 10^{-1}$
$4.971572 \cdot 10^{-9}$	$-4.032951 \cdot 10^{-1}$
$-2.624752 \cdot 10^{-12}$	$-5.624763 \cdot 10^{-1}$


**Sign 1**

moments reels	moments normalises
$7.421136 \cdot 10^{-4}$	$-7.668997 \cdot 10^{-2}$
$1.290289 \cdot 10^{-8}$	$-1.933297 \cdot 10^{-1}$
$4.196798 \cdot 10^{-11}$	$-2.542932 \cdot 10^{-1}$
$1.345561 \cdot 10^{-12}$	$-2.909044 \cdot 10^{-1}$
$-9.876489 \cdot 10^{-24}$	$-5.637535 \cdot 10^{-1}$
$-1.045946 \cdot 10^{-16}$	$-3.916062 \cdot 10^{-1}$
$-2.167194 \cdot 10^{-24}$	$-5.798952 \cdot 10^{-1}$

**Sign 2**

moments reels	moments normalises
$7.031999 \cdot 10^{-4}$	$-7.765548 \cdot 10^{-2}$
$1.078731 \cdot 10^{-8}$	$-1.962269 \cdot 10^{-1}$
$3.904462 \cdot 10^{-11}$	$-2.563566 \cdot 10^{-1}$
$2.407450 \cdot 10^{-12}$	$-2.861587 \cdot 10^{-1}$
$-2.110443 \cdot 10^{-24}$	$-5.831234 \cdot 10^{-1}$
$-8.111890 \cdot 10^{-17}$	$-3.963134 \cdot 10^{-1}$
$-2.324523 \cdot 10^{-23}$	$-5.574602 \cdot 10^{-1}$

**Sign 3**

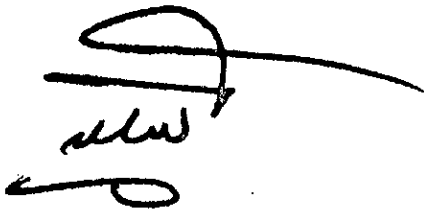
moments reels	moments normalises
$6.773265 \cdot 10^{-4}$	$-7.734497 \cdot 10^{-2}$
$1.273324 \cdot 10^{-8}$	$-1.926805 \cdot 10^{-1}$
$3.253260 \cdot 10^{-11}$	$-2.559538 \cdot 10^{-1}$
$2.483200 \cdot 10^{-12}$	$-2.832219 \cdot 10^{-1}$
$-2.231792 \cdot 10^{-23}$	$-5.528104 \cdot 10^{-1}$
$-2.796314 \cdot 10^{-16}$	$-3.795840 \cdot 10^{-1}$
$-2.314854 \cdot 10^{-25}$	$-6.012334 \cdot 10^{-1}$

**Sign 1.a**

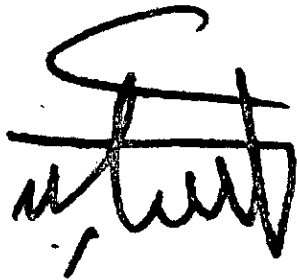
moments reels	moments normalises
$7.259140 \cdot 10^{-4}$	$-7.847200 \cdot 10^{-2}$
$8.946172 \cdot 10^{-9}$	$-2.011940 \cdot 10^{-1}$
$4.130066 \cdot 10^{-11}$	$-2.595817 \cdot 10^{-1}$
$2.713905 \cdot 10^{-12}$	$-2.891385 \cdot 10^{-1}$
$-1.325113 \cdot 10^{-23}$	$-5.719009 \cdot 10^{-1}$
$-1.528348 \cdot 10^{-16}$	$-3.953649 \cdot 10^{-1}$
$-2.549418 \cdot 10^{-23}$	$-5.647967 \cdot 10^{-1}$

**Sign 1.b**

moments reels	moments normalises
$6.899910 \cdot 10^{-4}$	$-7.979465 \cdot 10^{-2}$
$1.422601 \cdot 10^{-8}$	$-1.980737 \cdot 10^{-1}$
$4.425288 \cdot 10^{-11}$	$-2.613596 \cdot 10^{-1}$
$3.296239 \cdot 10^{-12}$	$-2.898309 \cdot 10^{-1}$
$-2.814687 \cdot 10^{-23}$	$-5.692268 \cdot 10^{-1}$
$-2.822515 \cdot 10^{-16}$	$-3.924968 \cdot 10^{-1}$
$-2.815384 \cdot 10^{-23}$	$-5.692241 \cdot 10^{-1}$


**Siga 7**

moments reels	moments normalises
$1.172806 \cdot 10^{-2}$	$-9.026331 \cdot 10^{-2}$
$1.075698 \cdot 10^{-4}$	$-1.855177 \cdot 10^{-1}$
$1.978290 \cdot 10^{-6}$	$-2.666474 \cdot 10^{-1}$
$7.740078 \cdot 10^{-7}$	$-2.857000 \cdot 10^{-1}$
$9.029111 \cdot 10^{-13}$	$-5.630713 \cdot 10^{-1}$
$6.533315 \cdot 10^{-9}$	$-3.826410 \cdot 10^{-1}$
$3.195032 \cdot 10^{-13}$	$-5.841634 \cdot 10^{-1}$


**Siga 8**

moments reels	moments normalises
$1.067243 \cdot 10^{-1}$	$-1.03127 \cdot 10^{-1}$
$1.11411 \cdot 10^{-2}$	$-2.072727 \cdot 10^{-1}$
$9.724454 \cdot 10^{-3}$	$-2.135408 \cdot 10^{-1}$
$9.731075 \cdot 10^{-3}$	$-2.135094 \cdot 10^{-1}$
$9.466157 \cdot 10^{-5}$	$-4.270346 \cdot 10^{-1}$
$1.027124 \cdot 10^{-3}$	$-3.171460 \cdot 10^{-1}$
$-7.444058 \cdot 10^{-8}$	$-7.564898 \cdot 10^{-1}$