

**République Algérienne Democratique Et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**UNIVERSITEMOULOUDE MAMMERI DE TIZI-OUZOU**



**FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE**  
**DEPARTEMENT D'AUTOMATIQUE**

**Mémoire de Fin d'Etude**  
**DE MASTER 2 ACADEMIQUE**

Spécialité : .....

Filière : .....

*Présenté par*

**Nadia RAHMOUNI**

**Markia KHOUAS**

Mémoire dirigé par **Sadia ALKAMA**

**Thème**

**Segmentation Des Images Couleurs Par les Réseaux de Neurones SOM**

*Mémoire soutenu publiquement le ..... mois 2015 devant le jury composé de :*

**M Prénom NOM**

Grade, Lieu d'exercice, Président

**M Prénom NOM**

Grade, Lieu d'exercice, Rapporteur

**M Prénom NOM**

Grade, Lieu d'exercice, Examineur

**M Prénom NOM**

Grade, Lieu d'exercice, Examineur

**M Prénom NOM**

Grade, Lieu d'exercice, Examineur

# REMERCIEMENT

On adresse notre vif remerciement à notre promotrice madame ALKAMA SAADIA pour son aide précieuse et sa patience qui nous a accompagnés tout le long de cette thèse.

On remercie également les membres des jurys pour avoir accepté d'examiner notre travail

Que tous ceux et celles qui nous ont aidé et soutenu à mener à bien ce travail. Trouvent également ici l'expression de nos remerciements.

# Sommaire

<b>Introduction</b> .....	1
 <b>Chapitre I : Généralités sur les réseaux de neurones</b>	
I.1. Introduction.....	2
I.2. Historique .....	2
I.3. Le neurone biologique .....	3
I.4. Le neurone formel .....	5
I.5. Les réseau de neurone formel .....	7
I.6. Architecture d'un réseau de neurone formel .....	10
I.6.1. Les réseaux Feed Forward .....	11
I.6.2. Les Réseaux "FEED-BACK" .....	11
I.7. Avantages et application des réseaux de neurones .....	12
I.8. <b>Conclusion</b> .....	12
 <b>Chapitre II : Carte auto-organisatrice de Kohonen</b>	
II.1. Introduction .....	13
II.2. Notations .....	13
II.3. Principe de fonctionnement des cartes topologiques de Kohonen.....	14
II.4. Notion de voisinage et de distance .....	15
II.5. Algorithme d'apprentissage de Kohonen.....	17
I.6. La classification hiérarchique .....	21
II.6.1 Objectif et principe de classification ascendante hiérarchique .....	22
II.6.2 Algorithme de CAH ascendante hiérarchique .....	22
II.6.3 Avantage de la classification ascendante hiérarchique .....	22
II.7. Conclusion.....	23
 <b>Chapitre III : Technique adoptée, tests et résultats</b>	
III.1. Introduction.....	24
III.2. Description de la technique adoptée .....	24
III.3 Tests et résultats. ....	26

III.4. Conclusion .....	30
<b>Conclusion générale</b> .....	<b>31</b>
<b>Bibliographie</b>	

# INTRODUCTION GENERALE

# *Introduction générale*

Avec les progrès de l'électronique et de l'informatique, l'image est devenue un vecteur fondamental de l'information et de la communication, elle tient une place importante dans des secteurs divers tels que la médecine, la biologie, la télédétection ...

Pour utiliser au mieux les images qui nous parviennent de différents domaines, on a souvent recours à la méthode de segmentation. Cette dernière est une étape primordiale de traitement d'image puisque elle conditionne la qualité de l'interprétation puis la prise de décision effectuée sur l'image plus tard. En effet, il s'agit d'extraire une information structurée de haut niveau permettant de reconnaître et d'indexer les images.

Il n'existe pas de méthodes de segmentation universelles et le choix d'une technique est difficile puisqu'elle est liée, notamment à la nature de l'image (éclairage, texture...) et aux opérations en aval de la segmentation (compression), aux primitives à extraire (régions).

Dans notre thèse, on a abordé la segmentation d'image avec la différente finesse imposée par l'utilisateur qui juge que certaines zones en plus d'intérêt que d'autres comme ce fait le cas dans la médecine ou on a besoin d'examiner qu'une partie d'un scanner. La segmentation sera donc celle qui permettra d'arriver à une bonne interprétation.

Nous allons segmenter des images dont des pixels appartiennent à des classes inconnues, pour cela on utilisera les cartes topologiques auto-organisatrices, qui sont une famille des réseaux de neurones à apprentissage non supervisé, après organisation de la carte, on appliquera une classification hiérarchique ascendante afin de regrouper les neurones en nombres de classes souhaitées. L'affectation aux différentes classes se fera en suite par rapport aux valeurs de ces neurones. La phase finale permettra à l'utilisateur de choisir sur l'image des zones intéressantes.

Pour mener à bien notre travail, nous l'avons organisé en trois chapitres :

- Le premier chapitre, traitera des notions générales sur les réseaux de neurones.
- Le second chapitre, sera consacré à carte auto-organisatrice de Kohonen.
- Les techniques adoptées, tests effectués et les résultats obtenus ainsi que les différentes interprétations seront présentés dans le troisième chapitre.

Et enfin, nous terminerons notre travail par une conclusion générale.

# CHAPTER 1

## I.1 Introduction

Les vingt dernières années ont vu naître des applications basées sur une technologie appelée « réseaux de neurones artificiels ». Ces applications couvrent de nombreux domaines tels que la reconnaissance des formes, l'optimisation des tâches répétitives. Etc. Le terme réseaux de neurones artificiels décrit des familles d'algorithmes qui étaient à l'origine inspirés des réseaux de neurones humaines. Le cerveau est capable d'organiser ces neurones selon un assemblage complexe, de manière à pouvoir réaliser des tâches très élaborées. C'est la tentative d'imiter son comportement et ces performances qui a conduit à une modélisation mathématique du neurone ce qui à donner naissance aux réseaux de neurones artificiels. Ce chapitre a pour but de donner les notions de base pour la compréhension des réseaux de neurones.

## I.2 Historique

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit. [1].

la première modélisation d'un neurone formel fut proposé par W.Pitts et MC Culloch en 1943 en s'inspirant de leurs travaux sur les neurones biologiques.

En 1949, D.Hebb, publie un livre sur l'organisation du comportement et les relations existantes entre la physiologie et la psychologie. On lui doit la règle de Hebb toujours utilisée dans les techniques d'apprentissage des réseaux neuronaux.

En 1958, F.ROSENBLATT commence à travailler sur le perceptron, un des premiers modèles spécifiés de manière suffisamment complexe pour en espérer des résultats intéressants .

En 1960, B.RINDROW , chercheur Américain à Stanford , développe le modèle adaline (adaptative linear element). Ce réseau est la base des réseaux multicouches.

En 1969, M.MINSKY et S.PAPERT publient un ouvrage démontrant les limites du perceptron, puis ils étendent ces limitations à tous les modèles du réseau de neurones. Par conséquent, plusieurs recherches dans ce domaine sont abandonnées.

Beaucoup de chercheurs se tournent alors vers d'autres domaines plus prometteurs tels que l'intelligence artificielle, toutefois certains d'entre eux persistent et les réseaux de neurones demeurent dans l'ombre jusqu'en 1982, date où J.J.HOPFIELD, à qui on doit le renouveau d'intérêt pour les réseaux de neurones, présente un article clair et convaincant développant son modèle de réseau de Hopfield.

En 1985, est apparu l'apprentissage par l'algorithme de rétro propagation pour les réseaux multicouches.

Aujourd'hui, l'étude des réseaux de neurones est une voie prometteuse de l'intelligence artificielle, ils ont des applications dans plusieurs domaines tels que l'industrie, les finances et les télécommunications ... .

### I.3 Le neurone biologique

Les neurones biologiques sont des cellules nerveuses qui ont leurs propres caractéristiques. Ces cellules nerveuses sont responsables de la réception et la transmission des influx nerveux et forment pour cela de longues fibres reliées entre elles. Elles sont composées d'un corps cellulaire qui contient un noyau d'un axone et d'une ou plusieurs dendrites. Le neurone biologique est représenté par la figure I.1. [2].

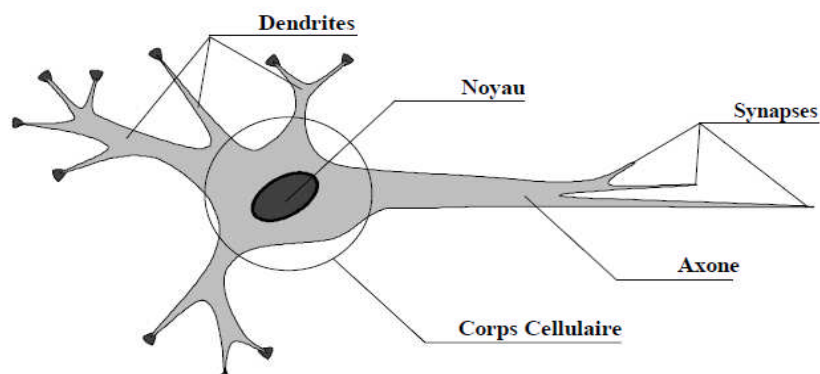


Figure I .1 Le neurone biologique.

On peut décomposer le neurone en quatre régions principales :

**Le corps cellulaire :** Il contient le noyau du neurone ainsi que la machine biochimique nécessaire à la synthèse des enzymes. Corps cellulaire de forme sphérique ou pyramidale contient aussi les autres molécules essentielles à la vie de la cellule. Sa taille est de quelques microns de diamètres.

**Les dendrites :** Ce sont les fines extensions tubulaires qui se ramifient autour du neurone et forment une sorte de vastes arborescences. Les signaux envoyés au neurone sont captés par les dendrites. Leur taille est de quelques dizaines de mètres.

**L'axone :** C'est le long de l'axone que les signaux partent du neurone. Contrairement aux dendrites qui se ramifient autour du neurone, l'axone est plus long et se ramifie à son extrémité ou il se connecte aux dendrites des autres neurones. Sa taille peut varier de quelques millimètres à plusieurs mètres.

**Le Synapse :** La connexion entre deux neurones se fait en des endroits appelés synapses. On peut les trouver entre deux dendrites, entre un axone et un corps cellulaire, ou entre deux axones. C'est le lieu où le signal électrique de l'impulsion nerveuse est converti en un signal biochimique. Le synapse est représenté par la figure I.2. [3]

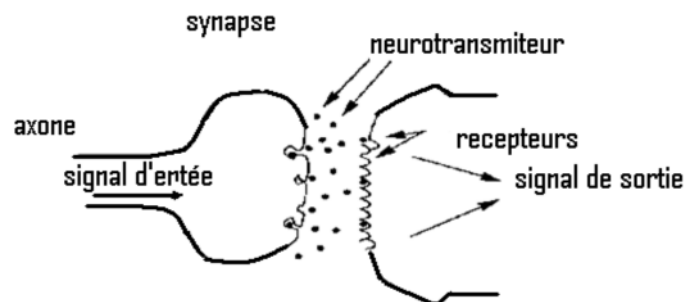


Figure I.2 Schéma d'une synapse

Du point de vue fonctionnel, il faut considérer le neurone comme une entité polarisée, c'est-à-dire que l'information ne se transmet que dans un seul sens : des dendrites vers l'axone. Le neurone va donc recevoir des informations, venant d'autres neurones, grâce à ces dendrites. Il va ensuite y avoir sommation, au niveau du corps cellulaire, de toutes ces informations et via un potentiel d'action (signal électrique) le résultat de l'analyse va transiter le long de l'axone jusqu'à la terminaison synaptiques.

A cet endroit, lors de l'arrivée du signal, des vésicules synaptiques vont venir fusionner avec la membrane cellulaire, ce qui va permettre la libération des neurotransmetteurs (médiateurs chimique) dans la fente synaptique. Le signal électrique ne pouvant pas passer la synapse (dans le cas d'une synapse chimique), les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre.

#### I.4 Le neurone formel

Le neurone formel est un automate caractérisé par un petit nombre de fonctions mathématiques. Il traite un signal d'entrée recueilli à travers ses connexions entrantes pour fournir un signal de sortie calculé par la fonction de transfert. Le neurone formel est représenté par la figure I.3.

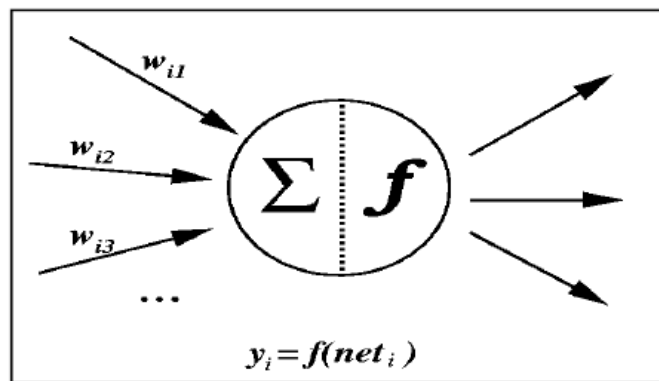


Figure I.3 Modèle d'un neurone formel

Les  $\{x_i\}$  sont les variables (ou entrées), les  $\{w_j\}$  sont des paramètres (ou poids).

Les entrées peuvent être booléennes, binaires (0,1), bipolaire (-1,1) ou réels. En règle générale, le calcul de la valeur de cette fonction peut se décomposer en deux étapes :

Une combinaison linéaire des entrées est formulée par l'équation(I.1) :

$$U = \sum_{i=1}^n w_j x_i - \theta \quad (I-1)$$

La sortie du neurone est formulée par l'équation(I.2) :

$$y = f(U) = f(\sum_{i=1}^n w_j x_i - \theta) \quad (I-2)$$

( $\theta$ ) est appelé biais, il peut être considéré comme la pondération de l'entrée 0 fixée à 1.

(U) est appelé potentiel du neurone.

La fonction (f) permet de définir l'état interne du neurone en fonction de son entrée totale.

Dans sa première version, le neurone formel était implémenté avec une fonction à seuil, mais de nombreuses versions existent. Ainsi le neurone de McCulloch et Pitts a été généralisé de différentes manières, en choisissant d'autres fonctions d'activations. [4].

La sortie du neurone dépend d'une fonction de transfert, dont les principales sont : fonction binaire à seuil ; fonction linéaire ; fonction linéaire à seuil ou multi seuil deux exemples sont donnés de la fonction binaire à seuil ci-dessous et sont représentées par la figure I.4 :

$$h(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{sinon} \end{cases}$$

$$\text{sgn}(x) = \begin{cases} -1 & \text{si } x < 0 \\ 1 & \text{sinon} \end{cases}$$

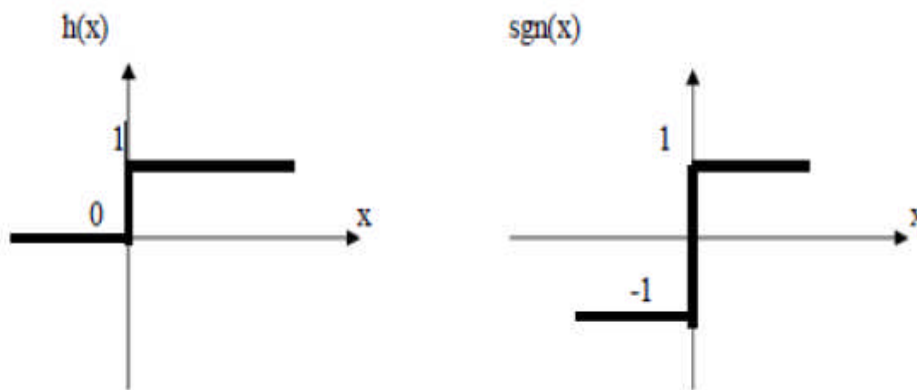


Figure I .4 fonction binaire à seuil

Le seuil introduit une non-linéarité dans le comportement du neurone, cependant il limite la gamme des réponses possibles à deux valeurs.

les fonction linéaire sont des fonctions d'activations simples, la fonction linéaire est définie par sa fonction :  $f(x)=x$  et représentée par la figure I.5.

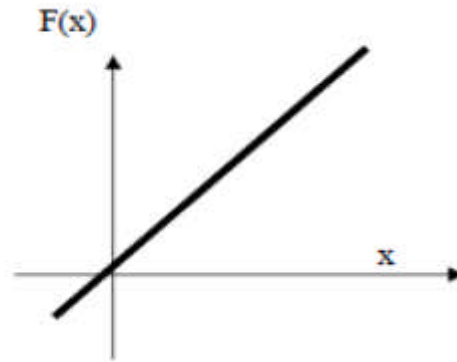


Figure I .5 fonction linéaire

La fonction linéaire à seuil ou multi seuil présente un compromis entre les fonctions linéaire et les fonctions seuil, entre leur deux barres de saturation, elles confèrent aux neurones une gamme de réponses possibles. En modulant la pente de la linéarité, on affecte la plage de réponse du neurone. [5].

Un exemple de fonction linéaire à seuil est donnée par l'équation ci-dessous et est représentée par la figure I.6.

$$F(x) = \begin{cases} x & x \in [u, v] \\ v & \text{si } x \geq v \\ u & \text{si } x \leq u \end{cases}$$

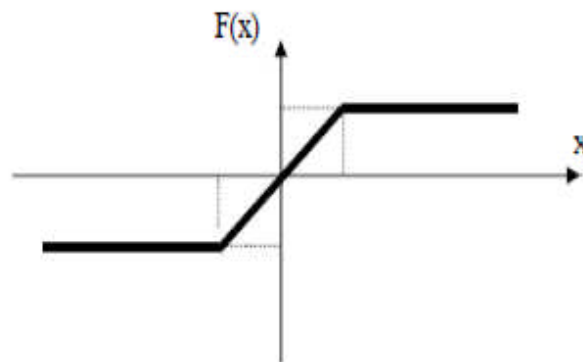


Figure I .6 fonction linéaire à seuil

## I.5 Les réseaux de neurone formel

Un RNA (réseaux de neurones artificiels) est un ensemble de neurones formels associés en couches (sous groupe) et fonctionnant en parallèle. Chaque

neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante comme l'indique la figure I.7.

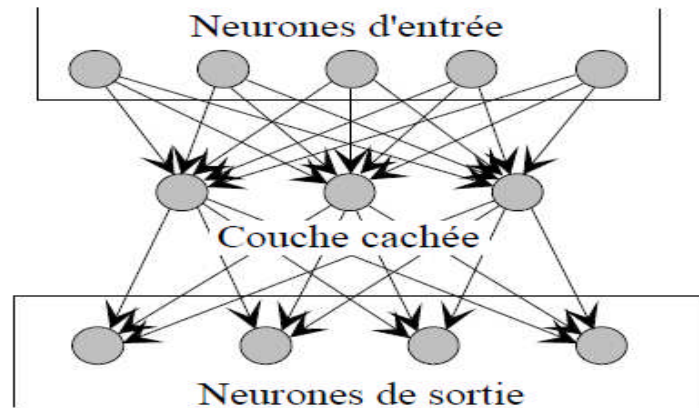


Figure I.7 Schéma d'un RNA

Pour un RNA, l'apprentissage peut être considéré comme le problème de la mise à jour des poids des connexions au sein du réseau, afin de réussir la tâche qui lui est demandée. L'apprentissage est la caractéristique principale des RNA et il peut se faire de différentes manières. Il existe 4 modes d'apprentissage des RNA qui sont :

**Mode supervisé :** Dans ce type d'apprentissage, le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie. Ainsi, le réseau va se modifier jusqu'à ce qu'il trouve la bonne sortie.

**Le renforcement :** Le renforcement est en fait une sorte d'apprentissage supervisé et certains le classe ailleurs. Dans cette approche le réseau doit apprendre la corrélation entrée/sortie via une estimation de son erreur, c'est-à-dire du rapport échec/succès. Le réseau va donc tendre à maximiser un index de performance qui lui est fourni, appelé signal de renforcement. Le système étant capable ici, de savoir si la réponse qu'il fournit est correct ou non, mais il ne connaît pas la bonne réponse.

**Le mode non-supervisé (ou auto organisationnel) :** Dans ce cas, l'apprentissage est basé sur des probabilités. Le réseau va se modifier en fonction des régularités statistiques de l'entrée et établir des catégories en attribuant et en optimisant une valeur de qualité, aux catégories reconnues.

**Le mode hybride :** Le mode hybride recouvre en fait les deux autres approches, puisque une partie des poids va être déterminée par apprentissage supervisé et l'autre partie apprentissage non- supervisé.

L'apprentissage se fait selon des règles bien définies, parmi les règles d'apprentissage on peut citer :

**Règle de correction d'erreurs :** Cette règle s'inscrit dans le paradigme d'apprentissage supervisé, c'est-à-dire dans le cas où l'on fournit au réseau une entrée et la sortie correspondante. Si on considère  $Y$  comme étant la sortie calculée par le réseau, et  $D$  la sortie désirée, le principe de cette règle est d'utiliser l'erreur  $(D-Y)$ , afin de modifier les connexions et de diminuer ainsi l'erreur globale du système. Le réseau va donc s'adapter jusqu'à ce que  $Y$  soit égal à  $D$ . Ce Principe est notamment utilisé dans le modèle du perceptron simple.

**Apprentissage de Boltzmann :** Les réseaux de Boltzmann sont des réseaux symétriques récurrents. Ils possèdent deux sous-groupes de cellules, le premier étant relié à l'environnement (cellules dites visibles) et le second ne l'étant pas (cellules dites cachées). Cette règle d'apprentissage est de type stochastique (c'est-à-dire qui relève partiellement du hasard) et elle consiste à ajuster les poids des connexions, de telle sorte que l'état des cellules visibles satisfasse une distribution probabiliste souhaitée.

**Règle de Hebb :** Cette règle, basée sur des données biologiques, modélise le fait que si des neurones, de part et d'autre d'une synapse, sont activés de façon synchrone et répétée, la force de la connexion synaptique va croître. Il est à noter ici que l'apprentissage est localisé, c'est à dire que la modification d'un poids synaptique  $w_{ij}$  ne dépend que de l'activation d'un neurone  $i$  et d'un autre neurone  $j$ .

**Règle d'apprentissage par compétitions :** La particularité de cette règle, c'est qu'ici l'apprentissage ne concerne qu'un seul neurone. Le principe de cet apprentissage est de regrouper les données en catégories. Les patrons similaires vont donc être rangés dans une même classe, en se basant sur les corrélations des données, et seront représentés par un seul neurone, on parle de « winner-take all ». Dans un réseau à compétition simple, chaque neurone de sortie est connecté aux neurones de la

couche d'entrée, aux autres cellules de la couche de sortie (connexions inhibitrices) et à elle-même (connexion excitatrice). La sortie va donc dépendre de la compétition entre les connexions inhibitrices et excitatrices.

## I.6 Architecture d'un réseau de neurone formel

Les RNA sont classées en deux grandes catégories : les réseaux Feed Forward ; les réseaux Feed Back.

### I.6.1 Les réseaux Feed Forward

Les réseaux Feed Forward sont appelés aussi « réseau de type perceptron ». Ce sont des réseaux dans lesquels l'information se propage de couche en couche sans retour en arrière possible. Parmi ces réseaux, on trouve :

- Le perceptron monocouche
- Le perceptron multicouche
- Les réseaux à fonction radiale

**a) Le perceptron monocouche :** Historiquement, le premier RNA, est le Perceptron de Rosenblatt. C'est un réseau simple, puisque il ne se compose que d'une couche d'entrée et d'une couche de sortie. IL a été conçu dans un but premier de reconnaissance des formes, cependant, il peut aussi être utilisé pour faire de la classification et pour résoudre des opérations logiques simples (telle "ET" ou "OU"). Sa principale limite est qu'il ne peut résoudre que des problèmes linéairement séparables. Il suit généralement un apprentissage supervisé selon la règle de correction de l'erreur.

**b) Le perceptron multicouche :** Est une extension du perceptron monocouche, avec une ou plusieurs couches cachées entre l'entrée et la sortie. Chaque neurone dans une couche est connecté à tous les neurones de la couche précédente et de la couche suivante et il n'y a pas de connexions entre les cellules d'une même couche. Les fonctions d'activation utilisées dans ce type de réseaux sont principalement les fonctions à seuil ou sigmoïdes. Il peut résoudre des problèmes non-linéairement séparables et des problèmes logiques plus compliqués. Il suit aussi un apprentissage supervisé selon la règle de correction de l'erreur.

**c) Les réseaux à fonction radiale :** Ce sont les réseaux que l'on nomme aussi RBF ("Radial Basic Functions"). Les fonctions de base utilisées ici sont des fonctions Gaussiennes. Les RBF seront donc employés dans les mêmes types de problèmes que les PMC à savoir, en classification et en approximation de fonctions, particulièrement. L'apprentissage le plus utilisé pour les RBF est le mode hybride et les règles sont soit, la règle de correction de l'erreur soit, la règle d'apprentissage par compétition.

### **I.6.2 Les Réseaux "FEED-BACK"**

Ils sont appelés aussi "réseaux récurrents", ce sont des réseaux dans lesquels il y a retour en arrière de l'information. Parmi ces réseaux, on peut citer :

- **Les réseaux de Hopfield ,**
- **Les cartes auto-organisatrices,**

#### **a) Les réseaux de Hopfield**

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

#### **b) Les cartes auto-organisatrices**

Les cartes topologiques ou cartes auto-organisatrices de réseau de neurones artificiels fondée sur la méthode d'apprentissage non-supervisé. On les désigne souvent par le terme self organisation map (SOM), ou cartes de KOHONEN du nom du statisticien ayant développé le concept en 1981. Elles sont utilisées pour cartographier un espace réel : étudier la répartition de données dans un espace à grande dimension. En pratique, cette cartographie peut servir à réaliser des tâches de discrétisation, quantification vectorielle ou classification.

### I.7 Avantages et application des réseaux de neurones

L'avantage des réseaux de neurones est qu'ils acceptent des données incomplètes, incertaines ou bruitées. Les réseaux de neurones offrent la capacité de représenter n'importe quelle dépendance fonctionnelle, c'est à dire que le réseau découvre la dépendance lui même.

Les principales applications des réseaux de neurones sont diverses, on retrouve essentiellement :

- **La reconnaissance des formes** : La reconnaissance des formes est un domaine privilégié d'application pour les réseaux de neurones. Elle désigne les processus qui traitent des informations afin de parvenir à leur classification.

- **Le traitement du signal** : dans ce domaine, les applications réalisées traitent essentiellement de la reconnaissance de signatures radar.

- **Le contrôle adaptatif** : Ce domaine recouvre, la robotique et le contrôle de qualité

- **La vision et la parole** : Des applications ont été développées aussi bien pour la compréhension de la parole que pour sa synthèse.

### I.8 Conclusion

Les réseaux de neurones formels, tels que nous les avons définis, possèdent une propriété remarquable qui est la phase d'apprentissage qui est à l'origine de leur intérêt dans des domaines très divers. Nous étudierons dans le chapitre suivant les cartes de Kohonen, qui est un type d'un réseau de neurone à apprentissage non supervisé.



# CHAPITRE 2

## II.1 Introduction

La classification par l'algorithme de Kohonen offre l'originalité de proposer une visualisation de la structure de voisinage entre les classes et la particularité de fournir une méthode de classification couplée avec une représentation appelée carte de Kohonen qui traduit la topologie entre les classes. Dans ce chapitre, nous présentons le principe des cartes auto-organisatrices ainsi que la méthode de classification hiérarchique ascendante qui permettra de regrouper et de ranger les neurones de la carte.

## II.2 Notations

Nous notons  $D$  l'ensemble qui représente l'espace des observations de dimension  $n$  tel que  $D \subset R^n$ . Soit le sous-ensemble  $A = \{z_i; i = 1, \dots, N\}$  de  $D$ , constitué de  $N$  individus qui formera l'ensemble d'apprentissage permettant d'estimer les différents paramètres du modèle.

Soit l'ensemble  $W = \{w_c; c = 1, \dots, p\}$  de  $p$  vecteurs de  $D$ . Ces vecteurs de dimension  $n$  constituent l'ensemble des référents.

Soit  $X$  une fonction d'affectation qui est une application de  $D$  dans l'ensemble des indices  $\{1, \dots, p\}$ . Cette fonction permet de réaliser une partition  $P = \{p_1, \dots, p_c, \dots, p_p\}$  de  $D$  en  $p$  sous-ensemble,  $p_c = \{z \in D / X(z) = c\}$ .

La figure II.1 montre le principe général de la modélisation. Une observation  $z$  est associée à un indice  $c$  choisi parmi  $p$  à l'aide de la fonction  $X$ ; cet indice permet de définir le référent  $w_c$ . On peut donc considérer que le vecteur référent  $w_c$  est un représentant de l'ensemble  $p_c$  dont il résume l'ensemble des observations. Les paramètres des différents modèles étant estimés à partir des observations de l'ensemble d'apprentissage  $A$ . La connaissance de l'ensemble des vecteurs référents  $w_c$  et de  $X$  détermine ce que l'on appelle une quantification vectorielle qui est utilisée pour affecter une observation  $z \in D$  à son référent  $w_{X(z)}$ ; tel que  $X(z)$  représente l'indice du référent auquel est associée l'observation  $z$ . La connaissance de la fonction

$X$  permet donc, au-delà de la quantification vectorielle, de définir la partition  $P$  de  $D$  en  $p$  sous-ensembles. [6]

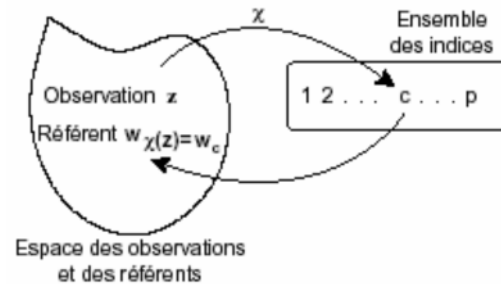


Figure II.1 Principe général de la modélisation

### II.3 Principe de fonctionnement des cartes topologiques de Kohonen

Une carte auto-organisée est un type de réseau de neurones artificiel compétitifs composé souvent de deux couches de neurones. La première couche sert uniquement à représenter les observations de l'espace des entrées. Chaque observation  $x$  est décrite par un vecteur à attributs. La seconde couche représentant la couche de compétition, est structurée en un treillis de neurones. L'algorithme SOFM (Self-Organizing Feature Maps) proposé par Kohonen [7] cherche à projeter l'espace des observations sur un espace discret de faible dimension. L'espace de projection est appelé carte. Cette carte est constituée d'un ensemble de neurones interconnectés, les liens entre ces neurones se faisant par l'intermédiaire d'une structure de graphe appelée les poids synaptiques. Cette structure de graphe induit à deux notions principales dans l'algorithme d'apprentissage des cartes de Kohonen, la distance et le voisinage entre les neurones.

On peut représenter l'ensemble constitué par la carte et les référents sous la forme d'un réseau de neurones (comme l'indique la figure II.2) constitué de deux couches :

- **La couche d'entrée** : sert à la présentation des observations à classer ; les états de tous ces neurones sont forcés aux valeurs des observations. Cette couche contient donc exactement  $n$  neurones ( $n$  étant la dimension de l'espace des observations).

- **La couche d'adaptation** : est formée du treillis des neurones qui forment la carte. La structure du réseau employé peut être soit fixée a priori, soit évoluer lors de

l'apprentissage. Les neurones utilisés à ce niveau sont des simples neurones « distances », chacun d'entre eux étant connecté à tous les éléments de la couche d'entrée. Soit  $\varphi$  la grille neuronale carrée d'une carte auto-organisatrice. Une carte de neurone assigne à chaque vecteur d'entrée  $z \in A$ , un neurone  $c \in \varphi$  désigné par son vecteur référent  $w_c$ , tel que le vecteur référent  $w_c$  est le plus proche de  $z$ . [6].

Kohonen a montré qu'il existe entre les neurones émetteurs et récepteurs un mécanisme d'interaction latérale qui dépend de la distance entre les neurones concernés. Ces cartes sont utilisées pour cartographier un espace réel, c'est-à-dire pour étudier la répartition de données dans un espace à grande dimension. En pratique, cette cartographie peut servir à réaliser des tâches de discrétisation, quantification, ou classification et regroupement.

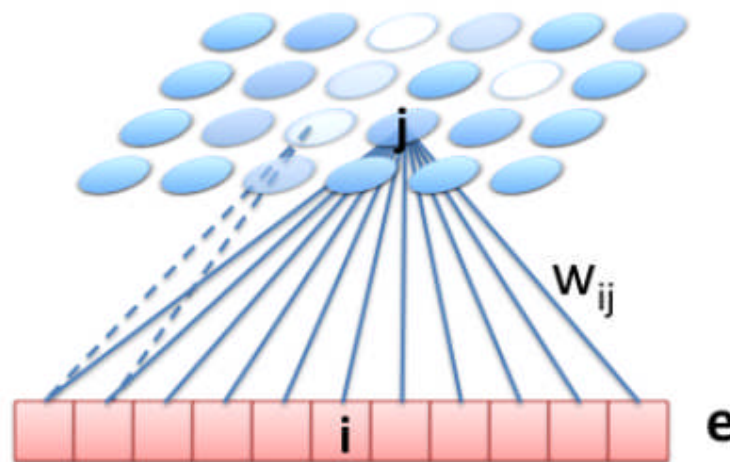


Figure II.2 projection des observations sur la carte topologique.

#### II.4 Notion de voisinage et de distance

La structure de graphe induit une distance  $\delta$  sur  $\varphi$ , pour toute paire de neurones  $(c, r)$  de cette carte,  $\delta(c, r)$  est définie comme étant la longueur du plus court chemin entre  $c$  et  $r$  sur le graphe  $\varphi$ . Pour chaque neurone  $c$ , cette distance discrète permet de définir la notion de voisinage d'ordre  $d$  de  $c$  :  $V_c(d) = \{r \in \varphi, \delta(c, r) \leq d\}$ .

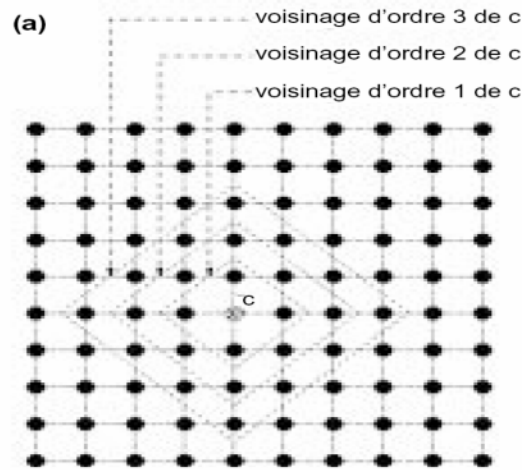


Figure II.3 Représentation de la topologie discrète d'une carte topologique à 2D.

La figure II.3 présente l'ensemble de ces notions de voisinage et de distance pour une carte topologique constituée par un treillis à deux dimensions.

Les distances  $\delta(c, r)$  qui lient le neurone  $c$  aux autres neurones ( $r$ ) de la carte permettent de faire varier l'influence relative des différents neurones, cette importance est quantifiée par  $K(\delta(c, r))$ . La fonction de voisinage  $K$  force les neurones qui se trouvent dans le voisinage de  $c$  à rapprocher leurs vecteurs référents du vecteur d'entrée  $Z$ . Moins un neurone est proche du vainqueur dans la grille, moins son déplacement est important.

La fonction de voisinage à seuil dans lequel les neurones du voisinage ont la même influence est donnée par la relation :

$$K(\delta) = \begin{cases} 1 & \text{si } \delta < 1 \\ 0 & \text{sinon} \end{cases} \quad (\text{II-1})$$

La fonction de voisinage de type gaussien dont l'influence entre deux neurones dépend de la distance entre ces neurones est tel que :

$$K(\delta) = \exp(-|\delta|) \quad (\text{II-2})$$

Afin de gérer la taille de voisinage, on utilise la famille de fonction  $K^T$  paramétrée par  $T$ .

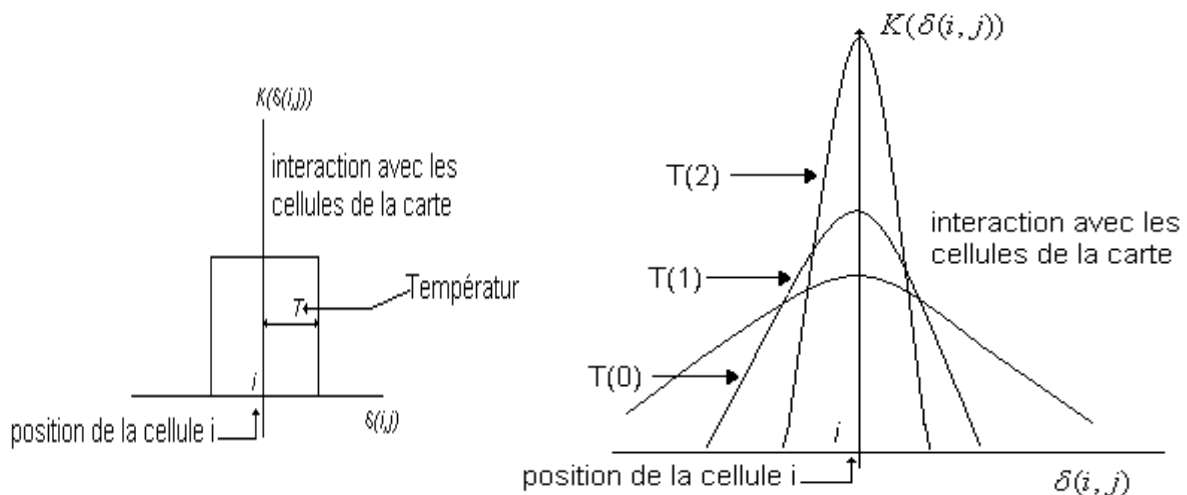
Les fonctions noyaux qui sont les plus utilisées dans la pratique sont :

- $K^T(\delta) = \begin{cases} 1 & \text{si } \delta < T \\ 0 & \text{sinon} \end{cases}$  (II-3)

- $K^T(\delta) = \exp(-\frac{|\delta|}{T})$  (II-4)

La fonction (II.3) est représenté en figure (II.4.a), elle se confond avec la fonction (II.1) pour T=1.

La fonction (II.4) est représenté sur la figure (II.4.b) pour des valeurs de températures différentes.



a) Fonction de voisinage à seuil

b) Fonction de voisinage de type gaussien

Figure II.4 : Exemples de fonctions de voisinage

## II.5 Algorithme d'apprentissage de Kohonen

L'idée de base de l'algorithme de Kohonen est de discrétiser un espace de grande dimension. Pour ce faire, un algorithme d'apprentissage est utilisé. Ce dernier, minimise une fonction coût  $J_{som}^T$ . Cette fonction cherche de manière itérative à trouver sur la carte les neurones les plus représentatifs d'un groupe de données de l'espace

d'entrée. Ainsi, chaque neurone représente une partie de l'espace des observations. Les algorithmes des cartes auto-organisatrices minimisent donc cette fonction qui est donnée par la formule suivante :

$$J_{som}^T(X, w) = \sum_{Z_i \in A} \sum_{c \in C} K^T(\delta(c, X(z_i))) \|z_i - w_c\|^2 \quad (\text{II.5})$$

La minimisation de cette fonction pour une valeur T fixée, est réalisée par itérations successives chacune se décomposant en deux phases. La première phase affecte l'ensemble des observations et la seconde minimise la valeur de la fonction coût associée à la partition. L'algorithme de Kohonen peut se résumer à ces étapes suivantes :

- **Phase d'initialisation**

Dans cette étape de l'algorithme, on choisit la taille et la structure de la carte), on initialise les poids synaptiques (généralement de manière aléatoire), et on fixe le nombre d'itérations.

- **Phase d'affectation**

Dans cette étape, on calcule la distance euclidienne de chacun des neurones par rapport à chaque vecteur de l'espace des observations en utilisant le critère de minimisation en distance euclidienne, et on affecte ce vecteur au neurone donnant une réponse maximale.

La fonction d'affectation utilisée par Kohonen est donnée par :

$$X(z_i) = \arg \min_c \|z_i - w_c\|^2 \quad (\text{II-6})$$

- **Phase de minimisation**

Cette phase minimise la fonction  $J_{som}^T(X, W)$  par rapport à l'ensemble des référents W. Cette minimisation est effectuée en gardant la fonction d'affectation X fixée et égale à la fonction calculée durant la phase précédente. La minimisation est donc obtenue pour la valeur qui annule la dérivée de  $J_{som}^T(X, W)$ . Dans la version stochastique de Kohonen, il n'est pas obligatoire de trouver le minimum global de  $J_{som}^T(X, W)$  pour X fixée, il suffit de faire décroître sa valeur. Ainsi à l'itération t et pour un neurone c le nouveau référent est donnée par :

$$w_c^t = w_c^{t-1} - \mu^t \frac{\partial J_{som}^T}{\partial w_c^{t-1}} \quad (\text{II-7})$$

$$\text{Avec, } \frac{\partial J_{som}^T}{\partial w_c} = 2 \sum_{z_i \in A} K^T(\delta(c, X(z_i)))(z_i - w_c) \quad (\text{II-8})$$

Cette méthode non adaptative suppose que l'on dispose de toutes les observations de l'ensemble d'apprentissage  $A$ . La contribution d'une seule observation  $z_i$  à la correction de  $w_c$  et représentée par le terme de la somme  $2K^T(\delta(c, X(z_i)))(z_i - w_c^{t-1})$ .

D'où les nouveaux référents sont donnés par le terme :

$$w_c^t = w_c^{t-1} - \mu^t K^t(\delta(c, X_t(z_i)))(w_c^{t-1} - z_i) \quad (\text{II-9})$$

Où  $\mu^t$  : représente le pas du gradient de l'itération  $t$ .

Les étapes de l'algorithme Kohonen sont détaillées dans l'algorithme (1) suivant [8] :

### Algorithme(1) : Cartes auto-organisées de Kohonen

#### 1. Première étape

- Initialisation aléatoire des poids pour la première itération
- choisir la structure et la taille de la carte et les  $p$  référents initiaux (en général, d'une manière aléatoire).
- fixer les valeurs de  $T_{\max}$  et  $T_{\min}$  et le nombre d'itération  $N_{iter}$  ; prendre  $t=0$ .

#### 2. Etape itérative $t$

L'ensemble des référents  $w^{t-1}$  de l'étape précédente étant connus :

- choisir une observation  $z_i$  (en général d'une manière aléatoire).
- calculer la nouvelle valeur de  $T$  en appliquant la formule :

$$T = T_{\max} * \left( \frac{T_{\min}}{T_{\max}} \right)^{\frac{t}{N_{iter}}}$$

Pour cette valeur du paramètre effectuer les deux phases suivantes :

- phase d'affectation : On suppose  $W^{t-1}$  connu ; on affecte l'observation  $z_i$  au neurone  $X_t(z_i)$  défini par la relation II.6
- phase de minimisation : Calcul de l'ensemble des nouveaux référents  $w^t$  par la relation (II.9).

**3. Répéter** l'étape itérative en faisant décroître la valeur du paramètre  $T$  jusqu'à ce que l'on atteigne  $t=N_{iter}$ .

L'algorithme détermine le centre d'activité comme étant le voisinage de la carte associé au neurone dont l'état ( $\|z_i - w_c\|$ ) est le plus petit. C'est l'utilisation de ce voisinage qui introduit les contraintes topologiques dans la représentation finale.

Dans la formule de modification des vecteurs référents, le pas du gradient  $\mu^t$  décroît avec les itérations. Au début de l'algorithme, la valeur de  $\mu^t$  est grande et la

décroissance de la fonction  $J_{som}^T$  n'est pas strictement assurée. Par la suite, le pas du gradient devient suffisamment petit : la modification des référents à chaque itération est petite.

Lorsque le paramètre  $T$  est petit, la modification de la relation (III.10) ne concerne qu'un ensemble réduit de neurones. La fonction de voisinage  $K^T(\delta(c,r))$  décroît lorsque la distance  $\delta(c,r)$  croît.

Les propriétés de l'algorithme de KOHONEN et de ses cartes de sortie permettent conjointement d'analyser des données qui ont une structure non linéaire et de réaliser des représentations graphiques qui mettent en évidence certaines de leurs caractéristiques. L'application la plus courante est la classification de l'espace d'entrée, où l'on définit une notion de voisinage entre les classes. Une fois la carte est organisée nous regroupons les neurones de la carte en utilisant une classification hiérarchique ascendante.

## II.6 La classification hiérarchique ascendante (CAH)

Il s'agit d'une méthode qui calcule une hiérarchie de partitions, chaque partition permet de regrouper de manière différente les neurones de la carte. Les différentes partitions de la hiérarchie sont déterminées d'une manière itérative, en commençant par la partition la plus fine qui est composé de l'ensemble des neurones.

La classification hiérarchique utilise cette partition initiale, et procède à des regroupements successifs en fusionnant à chaque itération deux sous ensembles de neurones. Le choix des deux sous ensembles qui vont fusionner à une étape donnée est effectué à l'aide d'une mesure de similitude, définie entre deux sous ensembles. On choisit, parmi tous les couples de sous ensembles qui constituent la partition à cette étape, les deux sous ensembles de neurones les plus semblables, au sens de la mesure choisie.

La classification hiérarchique permet d'engendrer un nombre variable de sous ensembles, car le processus de regroupement peut être arrêté à tout moment. Pour une mesure de similarité donnée le nombre d'éléments de la partition que l'on choisie dépend du nombre  $S$  de classe recherchées.

### II.6.1 Objectif et principe de classification ascendante hiérarchique

La classification ascendante hiérarchique (CAH) est une méthode de classification itérative dont le principe est simple.

1. On commence par calculer la dissimilarité entre les  $N$  objets.
2. Puis on regroupe les deux objets dont le regroupement minimise un critère d'agrégation donné, créant ainsi une classe comprenant ces deux objets.
3. On calcule ensuite la dissimilarité entre cette classe et les  $N-2$  autres objets en utilisant le critère d'agrégation. Puis on regroupe les deux objets ou classes d'objets dont le regroupement minimise le critère d'agrégation.
4. On continue ainsi jusqu'à ce que tous les objets soient regroupés.

### II.6.2 Algorithme de CAH

Les étapes de l'algorithme CHA se résume dans l'algorithme (2) suivant :

#### Algorithme (2) : Classification hiérarchique

1. **Initialisation.** Considérer la partition formée par les neurones ; chaque neurone est alors affecté à un sous ensemble distinct. Choisir le nombre  $K$  de groupement que l'on souhaite former.
2. Pour une partition donnée trouver les deux sous-ensembles les plus proches au sens du critère de similitude choisi, et les fusionner de manière à former un seul sous ensemble.
3. Si le nombre de groupement de la partition courante est inférieure à  $K$ , revenir à l'étape 2, sinon l'algorithme se termine.

### II.7.3 Avantage de la classification ascendante hiérarchique

La classification ascendante hiérarchique (CAH) est une méthode de classification qui présente les avantages suivants :

- On travaille à partir des dissimilarités entre les objets que l'on veut regrouper. On peut donc choisir un type de dissimilarité adapté au sujet étudié et à la nature des données.
- L'un des résultats est le dendrogramme voir figure II.5, qui permet de visualiser le regroupement progressif des données. On peut alors se faire une idée d'un nombre adéquat de classes dans lesquelles les données peuvent être regroupées.

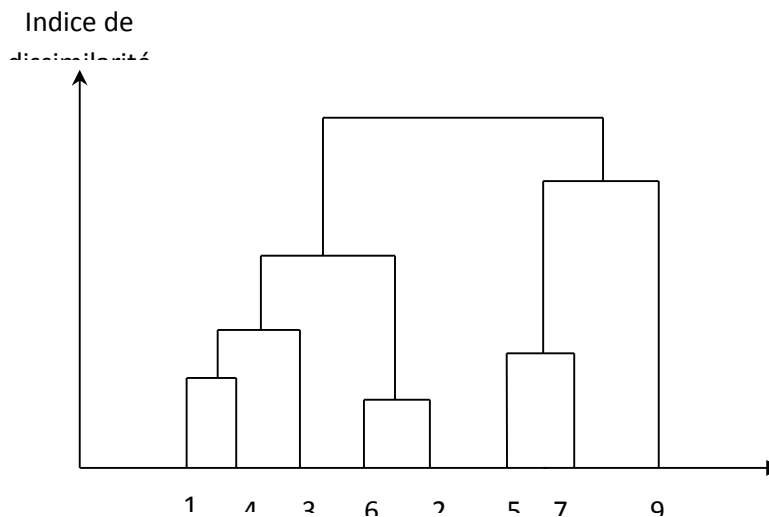


Figure II.5 Regroupement des neurones de la carte par classification ascendante hiérarchique

## II.8 Conclusion

Dans ce chapitre nous avons défini les cartes de Kohonen comme outil d'organisation de donnée,( nous utiliserons dans notre travail la segmentation d'images), puis nous avons en suite présenté la méthode de la classification ascendante hiérarchique qui sert à regrouper les données en classes homogènes, dans le chapitre suivant nous développerons un programme qui donnera la possibilité à l'utilisateur de donner le nombre de classes qu'il souhaite avoir dans chaque zone d'image.

# CHAPITRE 3

## II.I Introduction

Dans la pratique, on est souvent confronté à l'analyse d'images, dans but d'extraire de l'information contenue dans celle-ci. L'analyse de l'image fait souvent appel à la segmentation qui est une étape algorithmique fondamentale, où l'on va tenter d'associer à chaque pixel de l'image un label en s'appuyant sur l'information portée celui-ci, de délimiter dans l'image étudiée un ensemble de région pertinentes. Pour l'interprétation ou la modélisation de la scène perçue. Il s'agit donc d'une étape-clé.

Dans ce chapitre, notre but est la segmentation des images couleurs, pour cela nous avons opté pour une segmentation avec les réseaux de neurones SOM, car ces derniers nous permettent l'auto organisation de la carte topologique au fur à mesure que les données (image) sont introduites.

Par la suite on procédera, au regroupement des référents à fin d'obtenir le nombre de classes désirées, en appliquant la méthode de classification ascendante hiérarchique à laquelle nous allons ajouter la contrainte spatiale. Pour se faire, on commencera à implanter sous MATLAB, l'algorithme de KOHONEN avec les paramètres choisis à savoir : fonction de voisinage  $K^T$ , le pas du gradient  $\mu'$ , la taille de la carte topologique dans notre cas, elle est égale à  $10 \times 10$  et à la fin on implantera l'algorithme de la classification ascendante hiérarchique qui nécessite la connaissance du nombre de groupes  $nc$  à former dans le réseau. Les étapes essentielles sont les suivantes.

## III.2 Description de la technique adoptée

### 1) Organisation de la carte :

On commence à initialiser aléatoirement les poids de chaque référents, calculant ensuite la distance entre chaque référent, pour chaque pixel (image) on lui trouve le neurone gagnant, réajuster ensuite les poids des proches pour qu'elle ressemble encore plus à la cellule gagnante.

### 2) Regroupement

La procédure du regroupement consiste à trouver pour chaque neurone sa distance par rapport à ses quarts voisins, chercher la distance minimale entre deux

référents, moins un neurone est proche du vainqueur sur la grille, moins son déplacement est important.

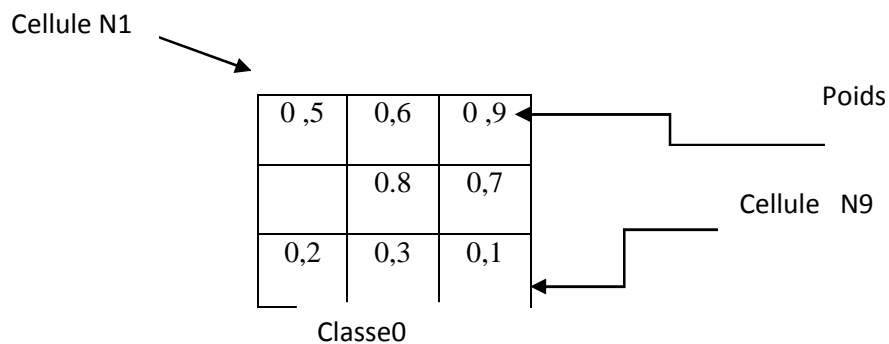
Trouver les neurones à fusionner avec ces deux référents, pour calculer à nouveau la valeur de leurs poids qui est la moyenne  $[\sum(\text{valeur des poids } i) / i]$ .

A la fin attribuer cette valeur trouvée à toutes les cellules fusionnées, répéter l'opération jusqu'à l'obtention du nombre de classes  $n_c$  désirées.

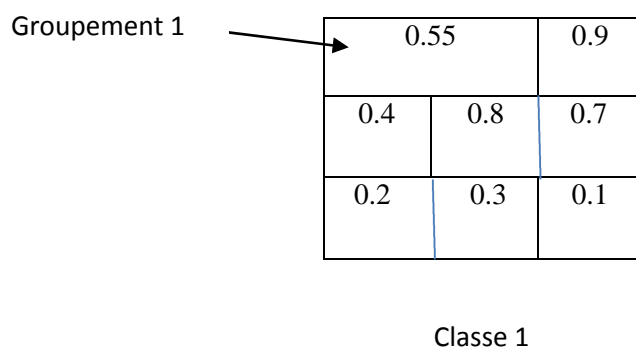
**3) Exemple d'application :**

Pour mieux comprendre, nous allons illustrer notre travail dans un exemple d'application sachant que on prendra la taille de la carte topologique 3\*3

On attribue des valeurs aléatoires normalisées pour les neurones comprises entre [0 1]

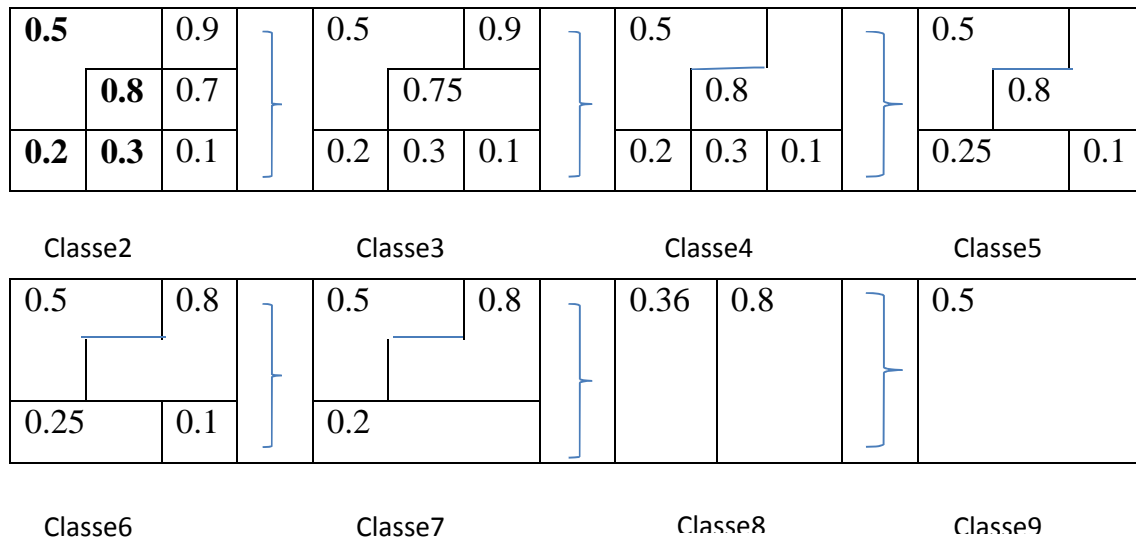


FigureIII.1 : Exemple de carte de Kohonen3\*3



FigureIII.2 Regroupement des référents voisins le N1 et le N2.

On continue à regrouper les référents à chaque fois que c'est possible, sachant que quand on fait un regroupement d'une cellule avec un regroupement déjà fait, il faut prendre les valeurs des poids initiaux.



FigureIII.3 Regroupement des référents dans différentes classe

**4) La Classification :**

Il vient ensuite la classification qui est la dernière étape de notre travail, elle consiste : pour n'importe qu'elle image couleur, en la parcourant , à chaque pixel de l'image , on doit comparer sa valeur à celle de chaque classe , si une classe est plus proche à la valeur du pixel , on attribuera cette valeur au pixel, selon le nombre de classes désirées à la fin , on aura une image segmentée classifiée.

**III.3 La Mise en pratique de la technique adoptée**

Afin de bien illustre la mise en pratique de la technique que nous avons développée, nous utiliserons l'image figure III.1, qui contient deux zones d'intérêt le drapeau et l'astronaute



Figure III.4 Image originale

En effet, l'utilisateur peut vouloir réaliser la segmentation seulement du drapeau et de l'astronaute et considérer le reste comme un fond.

Il sélectionne alors ces deux zones, voire figure III.2

En indiquant pour chaque zone le nombre de classe qu'il souhaite avoir.

Le nombre de classe maximum dans ce travail est fixé par la taille de la carte auto-organisatrice de kohonen, elle est dans ce cas fixée à 100.

Ainsi l'utilisateur aura à fixer les paramètres suivants :

- Nombre de zones à sélectionner (NS).
- Les précisions de chaque zone c'est-à-dire, le nombre de classes pour chaque zone  $P_i$  ( $i=1 \dots NS$ ).
- La précision des pixels qui n'ont pas été sélectionnés c'est-à-dire ceux qui seront considérés appartenir au fond : PF.

Dans l'exemple précédent en choisissant :

NS= 2,  $P_1= 40$ ,  $P_2 =50$  et PF= 1 ; on obtient le résultat de la figure III.3.

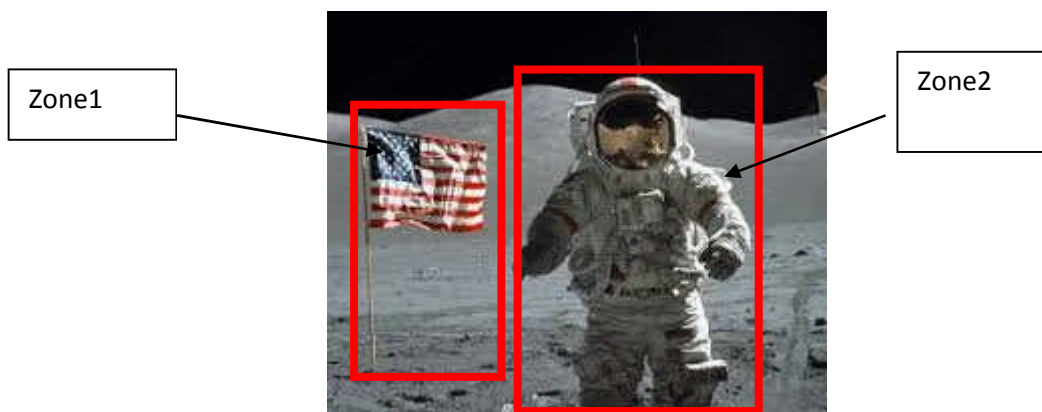


Figure III.5 Image avec les zones sélectionnées

Après segmentation, on aura le résultat suivant :

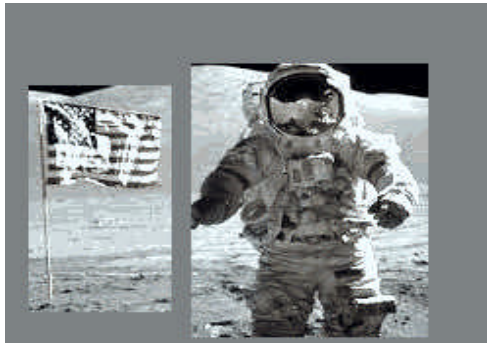


Figure III.6: Image segmentée

### III.4 Tests et Résultats

Nous allons appliquer le programme que nous avons réalisé sur plusieurs images de plusieurs domaines d'utilisation.

Les figures suivantes illustrent les résultats obtenus :

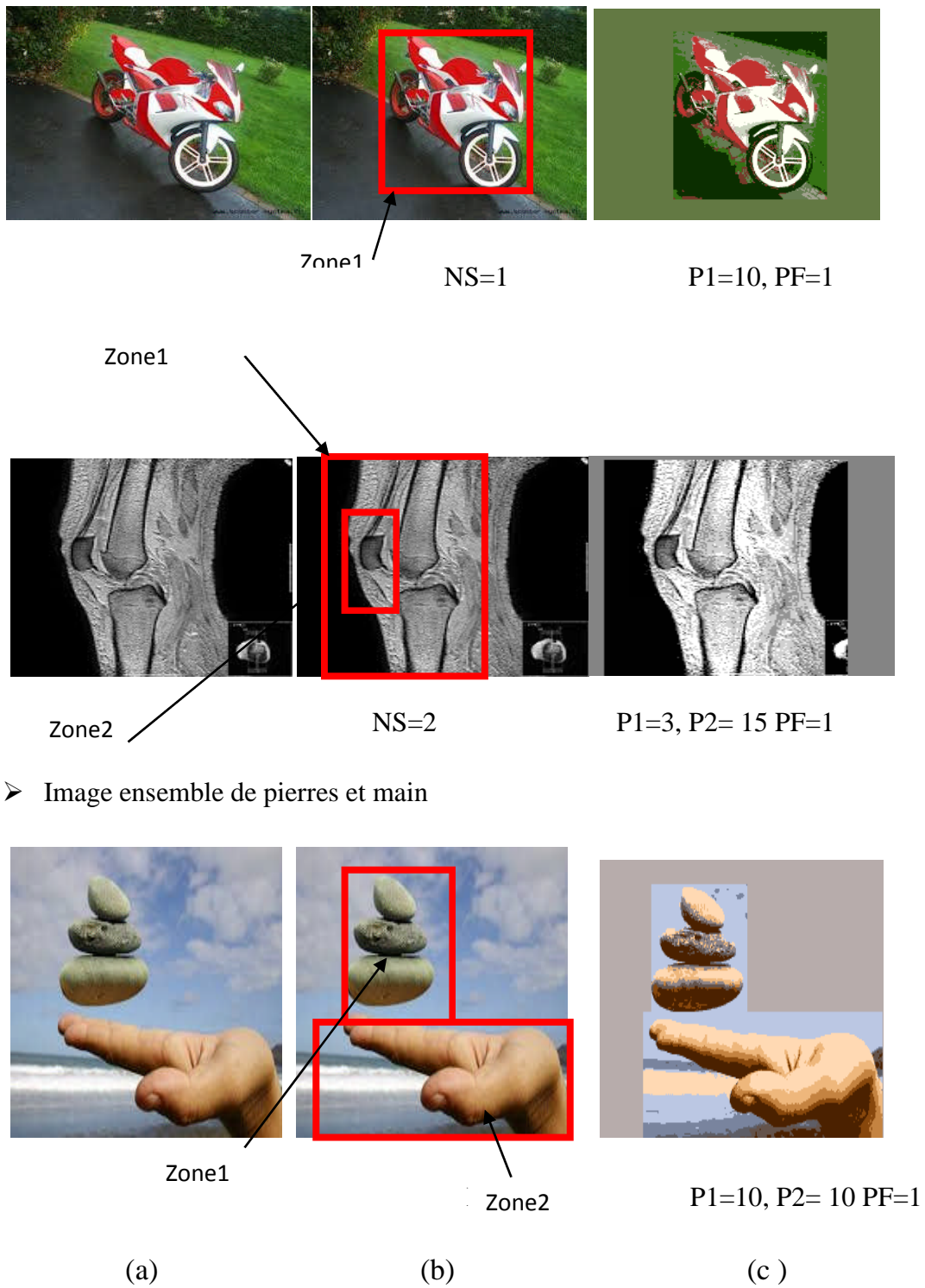


Figure III.7 : résultats et tests

Les images : 1) moto ; 2) Imagerie médicale ; 3) Ensemble de pierres et main .

a) Images initiale, b) Sélection de zones, c) Résultats de segmentation

Dans l'image moto (figure III.4.1.a), il est évident que l'unique objet de cette scène qui est en l'occurrence la moto, sera l'unique objet d'intérêt ( figure III.4.1.b ).

NS sera donc fixé à 1(NS=1).

L'utilisateur peut choisir le nombre de classe en lesquelles, il souhaite segmenter cette moto.

Dans cet exemple, le nombre est fixé à 10 (P1=10).

Comme tout le reste de la scène ne présente aucun intérêt pour l'utilisateur, le nombre de classe est alors fixé à 1 (PF=1).

Le résultat de la segmentation de l'image moto avec les valeurs des paramètres cités précédemment est donné sur la figure III.4.1.b.

Dans certains domaines l'utilisateur, tel que le domaine médical, il est souvent utile d'isoler une zone spécifique qui est généralement la zone malade pour pouvoir apporter un diagnostic.

Dans le cas de la figure genou (figure III.4.2.a) ; la zone à observer est la rotule. L'utilisateur peut choisir deux zones, une première zone qui sera le genou entier du patient en lui affectant un nombre de classe réduit P1=3 car le médecin n'a pas besoin de voir les détails et une deuxième zone qui sera la rotule en lui affectant un nombre de classe plus élevé P2=15,

car les détails seront important pour le diagnostic. Le reste de l'image ne présentant aucun intérêt, l'utilisateur fixera le nombre de classes à PF=1.

Le résultat de cette segmentation est affichée en figure III.4.2.c.

Dans le cas de la figure ensemble pierres et main (figure III.4.3.a). ; Les zones à observer sont les pierres et la main. L'utilisateur choisi d'affecter le nombre de classe aux pierres P1= 10 et le nombre de classe P2= 10 pour la main, pour les pixels sont intérêt, l'utilisateur fixera le nombre de classes à PF=1.

Le résultat de cette segmentation est affichée en figure III.4.3.c.

Le choix des zones et des nombres de classes est laissé à l'appréciation de l'utilisateur .

Les zones sont choisies par l'utilisateur selon les besoins et le traitement qu'il veut faire et le but à atteindre.

Il est évident, que à chaque fois que on attribue plus de classe à une zone sa segmentation sera plus fine, on aura plus de détail comme indique par exemple les cas de la figure III.5.



NS=1

P1= 2, PF=1

P1= 5, PF=2

P1= 50, PF=2

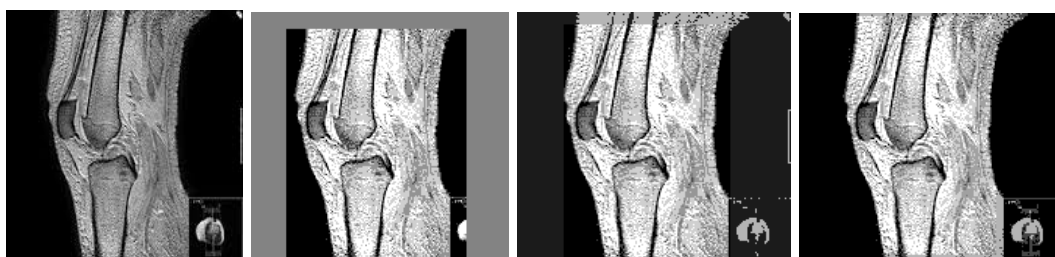


NS=2

P1= 3, P2= 3, PF=1

P1= 6, P2= 6, PF=3

P1= 20, P2= 20, PF=5



NS=2

P1= 5, P2= 6, PF=1

P1= 7, P2= 9, PF=2

P1= 9, P2= 10, PF=3

(a)

(b)

(c)

(d)

Figure III.8 : variation de nombre de classe affecté à chaque zone

(a) image avec zones sélectionnées.

b), c), et d) : résultats de la segmentation

**III .5 Conclusion**

Dans ce chapitre nous avons présenté différents résultats de l'application de notre programme sur différentes images et répondre aux exigences de l'utilisateur, la segmentation est réalisée sur plusieurs images avec des zones et des nombre de classes choisies par l'utilisateur.

On a effectué des tests sur une même image et voir l'influence de nombre de classe à chaque fois que cette dernière augmente la précision est plus importante.

# CONCLUSION GENERALE

# *Conclusion générale*

Après avoir étudié, dans le premier chapitre, les réseaux de neurones et leurs utilités dans différents domaines, dans le deuxième chapitre nous nous sommes intéressés plus particulièrement aux cartes auto-organisatrices qui est un type de réseaux de neurone non supervisé, nous avons inclus une explication détaillée du principe des cartes, de leurs algorithmes d'apprentissage et de leur rôle dans la classification des données,( nous utiliserons dans notre travail la segmentation des images). Puis nous avons implémenté cet algorithme sous MATLAB.

Les cartes de Kohonen permettent le classement des observations sur lesquels aucune information relative a leur classification n'est disponible. Une fois le réseau organisé (une fois l'apprentissage effectué), chaque observation est affectée aux différents neurones puis classée dans une zone après avoir trouver les valeurs des référents, ensuite faire un regroupement des neurones à l'aide la classification ascendante hiérarchique.

Dans le troisième chapitre, nous avons utilisé La méthode citée au paravent pour réaliser une segmentation qui permet de classer les pixels de images avec un nombre de classe spécifié par l'utilisateur, à savoir extraire dans une image des zones à caractéristiques importantes, en faisant varier le nombre de classe, on a pu constater que le résultat est meilleur

La segmentation réalisée, nous a permis d'initier au domaine du traitement d'image, au logiciel MATLAB, on a pu constater que on peut choisir une zone aléatoirement à main levée.

# Bibliographie

[1] [2] les reseaux de neurones artificiels introduction au connexionnisme cours, exercices et travaux pratiques claude touzet juillet 1992

[4] [5] : Mémoire de master ,en physique de la matière condensée et des semi-conducteurs

présentée par :*melle kadous djamila* : utilisation des réseaux de neurones comme outil du datamining :génération de modèle comportemental d'un processus physique à partir de données UNIVERSITE ABOU-BAKR BELKAID DE TLEMCEM

[6] [7] G. DREYFUS, J. M. MARTINEZ, M. SAMUELIDES, M. B. GORDON, F. BARDRAN, S. THIRIA et L. HERAULT, Réseaux de neurones : méthodologie et application, Edition EYROLLES, 2004

[8] Kohonen T. (1995). *Self-Organizing Maps*. Springer Series in Information Sciences Vol 30, Springer Verlag, Berlin.

