

République Algérienne Démocratique et Populaire  
Ministre de l'enseignement Supérieur et de la Recherche Scientifique  
Université Mentouri - Constantine  
Faculté des Sciences de L'Ingénieur  
Département d'Electronique

N° d'ordre :

Série :

**Mémoire**

Présenté pour l'obtention du diplôme de Magister en Electronique

**Option : Contrôle**

**Thème :**

---

*Partitionnement neuronal et validité des classes*  
*Application à la segmentation d'images*

---

**Dirigé par :**  
*Dr. Fella Hachouf*

**Présenté par :**  
*Amel Boulemnadjel*

*Soutenu le :07.07.2009*

**Devant le jury :**

Prof. S. FILALI	Professeur, Université de Constantine	Président
Prof. M.KHAMADJA	Professeur, Université de Constantine	Examineur
Dr. S.CHIKHI	Maître de Conférences, Université de Constantine	Examineur
Dr. F.HACHOUF	Maître de Conférence, Université de Constantine	Rapporteur

# Remerciements

*Je remercie avant tout premièrement Dieu pour la volonté, la santé et la patience qu'il m'a donné durant toutes les années d'étude et de pouvoir parachever ce travail.*

*Mes remerciements les plus sincères sont adressés à mon encadreur Dr F. Hachouf qui m'a orienté d'une façon efficiente.*

*Je remercie les membres du jury qui ont accepté de juger ce travail.*

*- Prof. S. Filali Professeur, Université de Constantine*

*- Prof. M. Khamadja, Professeur, Université de Constantine.*

*- DR. S. Chikhi, Maître de conférence, Université de Constantine.*

*Je tiens également à remercier l'ensemble du groupe de centre d'imagerie médical*

*"EL RAYENNE" pour leur gentillesse et leur soutien.*

*Je veux également remercier docteur M. Bourahli le médecin radiologue pour l'aide et pour*

*l'interprétation des images utilisées.*

*Enfin, les derniers mots sont pour remercier toute ma famille et mes amies pour leur aide et*

*Leurs encouragements constants et pour leur soutien moral.*

# *Dédicaces*

*je dédie ce modeste travail en particulier a mes adorables parents :*

*A ma mère FATIHA qui m'a donné l'amour et l'affection*

*A mon père SALEH qui m'a donné le courage et la volonté.*

*Très spécialement à mes tantes HAKIMA & DJEMAA*

*A mes frères : TALLEL & HOUCIN*

*A mes sœurs : MERIEM & HADJER*

*A mes meilleurs amies RBIHA & SAFIA*

*A mes très chères amies AMINA, CHAHRAZED & INES*

*A mes cousines LOUBNA , SARAH & MERIEM*

*A mes petites anges : RIM, SOUNDOS & DERGHAM*

*A mes grands parents, mes oncles mes tantes et toute ma famille chacun avec son nom*

*Et sans oublier mon groupe de français "niveaux 9 à VLC" chacun avec son nom, La liste est très longue que même la taille d'un mémoire ne suffira pas donc a tous ceux ou celle avec qui j'ai partager un moment de ma vie.*

*Amel*

## **Résumé**

Dans ce travail, nous nous intéressons aux problèmes d'algorithmes neuronaux appliqués à la segmentation d'images via la classification non supervisée. Les cartes auto\_organisatrices de Kohonen et l'algorithme Fuzzy ART sont étudiés et améliorés. Nos investigations étaient orientées dans l'axe de la recherche du nombre de classes optimal, qui est le problème majeur de la classification non supervisée. Ce problème est résolu avec un processus d'évaluation des résultats des algorithmes de classification utilisant un indice de validité des classes. Différents indices sont testés et nous proposons un indice de validité qui offre beaucoup d'avantages. Ces indices de validité permettent d'identifier le nombre exact de classes qui donne une amélioration de la classification non supervisée.

## ملخص

في هذا العمل , كان اهتمامنا منصب حول مشاكل الخوارزميات العصبية المطبقة في تجزئة الصور فالتصنيف غير المراقب. البطاقات الذاتية الترتيب لكوهونان وخوارزم فيزي ارت هما الخوارزميان المستعملان و المحسنان في هذا العمل.

يتمحور هذا العمل أساسا حول البحث عن العدد المثالي للأقسام وهذا هو المشكل الرئيسي في طريقة التصنيف غير مراقب. هذا المشكل تم حله عن طريق آلية تسمح بتقييم نتائج التصنيف مستعملة مؤشر صحة الأقسام. عدة مؤشرات تم تجربتها في هذا العمل , أيضا المؤشر المقترح من طرفنا والذي سمح لنا بتقديم عدة تحسينات.

هذه المؤشرات سمحت لنا بتحديد العدد الصحيح للأقسام وبالتالي تحسين نوعية التصنيف غير المراقب.

## **Abstract**

In this work, we are interested in neuronal algorithms problems applied to the images segmentation denying the unsupervised classification. The self\_organizing map of Kohonen and Fuzzy ART algorithm are studied and improved.

Our investigations will be directed in the axis of the optimal search for the classes' number, which is the major problem of the unsupervised classification.

This problem can be resolved with an evaluation process of classification algorithms results using an index of classes' validity. Various indices are tested and we propose a validity index which offers many advantages.

These validity indices allow identifying the exact number of classes which gives an improvement of the unsupervised classification.

# Sommaire

<b>Introduction</b> .....	1
<b>Chapitre 1 Classification non supervisée</b> .....	3
1. Introduction .....	3
2. Qu'est-ce qu'une classification ? .....	3
3. Les différentes approches de la classification non supervisée .....	6
3.1 Classification par partitionnement .....	6
3.1.1 Partitionnement "doux" .....	6
3.1.2 Partitionnement "dur" .....	7
3.2 Classification hiérarchique .....	7
4. Connaissances du domaine et contraintes .....	9
4.1 Contraintes sur les groupes : forme et taille. ....	9
4.1.1 Contraintes de forme .....	9
4.1.2 Contraintes de taille.....	9
4.2 Contraintes sur les objets .....	10
4.2.1 Fusion et Exclusion .....	10
4.2.2 Etiquetage partiel.....	10
4.3 Contraintes sur les attributs .....	10
5. Nombre de classes .....	10
6. Qualité d'une partition .....	11
7. Distance et densité.....	11
8. Conclusion.....	12
<b>Chapitre 2 : Réseaux de neurones</b> .....	13
1. Introduction .....	13
2. présentation des réseaux de neurones .....	13
2.1 Définition .....	13
2.2 Historique .....	15
2.3 Structure du réseau .....	16
2.4 Fonctions de transfert.....	17

2.5 Différents types d'apprentissage.....	19
2.5.1 Apprentissage supervisé.....	19
2.5.2 Apprentissage non supervisé.....	19
2.6 Applications .....	19
2.7 Architecture.....	20
3. Carte auto organisatrices.....	20
3.1 K_means.....	20
3.2 Carte auto_ organisatrices de Kohonen (SOM) .....	21
3.2.1 Principe.....	22
3.2.2 Architecture.....	23
3.2.3 Règle d'apprentissage .....	24
3.2.4 Algorithme d'apprentissage .....	25
3.2.5 Propriétés de SOM .....	25
3.2.6 Classification de SOM .....	27
4. La Théorie de la résonance adaptative (ART) .....	33
4.1 L'approche Fuzzy ART .....	36
4.2 L'Algorithme : fuzzy ART.....	37
4.3 Les problèmes du fuzzy ART.....	38
4.4 Solutions proposées.....	38
5. Conclusion.....	39

**Chapitre 3 : Indices de validité des classes..... 40**

1. Introduction .....	40
2. Définition .....	40
3. Critère de validité.....	40
3.1 Critère de Davies_Bouldin.....	41
3.2 Critère de Dunn .....	42
3.3 Critère du C_index .....	43
3.4 La méthode de Validité Silhouette.....	43
3.5 Critère de compacité Wemmert et Gançarski .....	44
3.6 Critère basée sur la densité inter et intra classe (CDBw).....	44
3.7 Critère proposé .....	46
4. Conclusion.....	48

<b>Chapitre 4 : Résultats expérimentaux</b> .....	49
1. Introduction .....	49
2. L'imagerie médicale.....	49
3. Présentation des images médicales utilisées .....	49
4. Algorithme de K_means.....	51
5. La classification neuronale.....	52
5.1 Application de l'algorithme SOM.....	52
5.1.1 Visualisation de la carte .....	53
5.1.2 Classification de la carte .....	54
5.2 Application de l'algorithme Fuzzy ART.....	55
6. Validité des classes.....	57
7. Validité des classes par l'indice proposé.....	60
8. Conclusion.....	62
 <b>Conclusion Générale</b> .....	 63
<b>Bibliographie</b> .....	65

## Liste des figures

<b>Figure 2.1</b> : Schéma d'un neurone biologique .....	14
<b>Figure 2.2</b> : La comparaison entre un neurone biologique et un neurone artificiel. ....	14
<b>Figure 2.3</b> : Structure d'un neurone artificiel.....	17
<b>Figure 2.4</b> : La topologie des réseaux de neurones .....	20
<b>Figure 2.5</b> : Représentation de la topologie discrète d'une carte à deux dimensions constituées de 10 * 10 neurones.....	23
<b>Figure 2.6</b> : Architecture d'une carte topologique en 2-D .....	24
<b>Figure 2.7</b> : Relation entre la carte des caractéristique $\Phi$ et le vecteur poids $w_j$ du neurone gagnant $i(x)$ .....	26
<b>Figure 2.8</b> : Conservation de la topologie des données.....	27
<b>Figure 2.9</b> : SOM à deux niveaux.....	28
<b>Figure 2.10</b> : Regroupement des neurones de la carte par classification ascendante .....	29
hiérarchique	
<b>Figure 2.11</b> : Le neurone A à huit directions de voisinage.....	32
<b>Figure 2.12</b> : Architecture du réseau fuzzy ART .....	35
<b>Figure 3.1</b> : Les deux représentations possibles des classes .....	42
<b>Figure 3.2</b> : -a- et -b- montre deux exemples de classification.....	42
<b>Figure 3.3</b> : Exemple d'une classe $c_i$ .....	46
<b>Figure 4.1</b> : Le profil du sein droit .....	50
<b>Figure 4.2</b> : Coup axiale T1 du crâne (crâne 1) .....	50
<b>Figure 4.3</b> : Image du crâne (crâne 2) .....	51
<b>Figure 4.4</b> : Résultat de la classification par K_means .....	51
<b>Figure 4.5</b> : Résultat de la classification par K_means pour différentes valeurs de K.....	51
<b>Figure 4.6</b> : Etat des neurones .....	53
<b>Figure 4.7</b> : Classification par l'algorithme SOM directement.....	54
<b>Figure 4.8</b> : Classification par SOM à deux niveaux .....	55
<b>Figure 4.9</b> : Résultats obtenus par Fuzzy ART sans et avec modification.....	56
<b>Figure 4.10</b> : Image avant validité et image après validité.....	59
<b>Figure 4.11</b> : Histogramme des images : a, c: avant correction (SOM direct). b, d : après correction.....	61
<b>Figure 4.12</b> : Résultat de la classification. a, c image classifiées par l'algorithme de SOM direct b,d images après la correction de l'indice proposé .....	61

## Liste des tableaux

<b>Tableau 1.1</b> : Les différentes distances .....	8
<b>Tableau 2.1</b> : Fonctions de transfert $a = f(n)$ .....	18
<b>Tableau 2.2</b> : Algorithme ART1 et Fuzzy ART.....	35
<b>Tableau 4.1</b> : Les indice de Davies_Bouldin et Dunn.....	52
<b>Tableau 4.2</b> : Les valeurs de les indices CDbw et Davies_Bouldin pour chaque image .....	57
<b>Tableau 4.3</b> : Les valeurs de les indices CDbw et Davies_Bouldin pour l'algorithme de Fuzzy ART .....	58
<b>Tableau 4.4</b> : Les pourcentages des images classifiées.....	60

# Introduction

# Introduction

La classification non supervisée constitue une étape importante du traitement d'une image. Cette opération a pour but de séparer différentes zones homogènes d'une image afin d'organiser les objets en groupes (clusters) dont les éléments ont en commun diverses propriétés (intensité, couleur, texture, etc). On peut regrouper les méthodes de classification en deux catégories : la classification supervisée, qui s'opère à partir de la connaissance de chacune des classes définies par une approche probabiliste, et la classification non supervisée qui vise à séparer automatiquement l'image en clusters naturels, c'est-à-dire sans aucune connaissance préalable des classes. En classification non supervisée aucune information sur la classe d'un objet n'est fournie à la méthode (on dit que les objets ne sont pas étiquetés). Les méthodes de classification ont pour objectif d'organiser en classes homogènes les éléments d'un ensemble. Elles constituent donc une approche naturelle pour réaliser une segmentation d'images. La segmentation d'images permet d'apprécier les formes et de les interpréter de la manière la plus juste possible pour aboutir à une plus grande informativité de l'image sans altérer sa fidélité.

Différentes méthodes de classification ont été trouvées dans la littérature [1,2,3,4]. Nous nous sommes principalement intéressées aux méthodes de classification automatique non supervisées à l'aide des réseaux de neurones qui ont montré leur capacité à réaliser ce genre de tâche.

Les réseaux de neurones représentent un ensemble de neurones reliés entre eux par des synapses. Les neurones sont des unités nerveuses de base composées d'une cellule et d'un noyau. Les neurones ont la propriété de toutes les cellules vivantes. La particularité de ce réseau réside dans son autonomie de fonctionnement. En effet, l'homme n'agit pas directement sur ces réseaux pour superviser les tâches ou pour en contrôler le fonctionnement. C'est un réseau dit "intelligent" qui possède la propriété d'auto adaptation. Grâce à cette propriété, le réseau neuronal du cerveau humain est capable d'apprendre et de mémoriser des schémas de façon autonome, sans l'intervention d'un pilote. On appelle cela un réseau auto organisé. Il présente différents types d'architectures. Dans ce travail nous allons considérer deux types de réseau de neurone non supervisé, les cartes auto organisatrices de Kohonen (SOM) et les réseaux ART.

Les cartes auto-organisatrices de Kohonen sont utilisées à l'origine pour des applications de

classification; elles s'adaptent aux observations au moyen d'un algorithme d'apprentissage non supervisé, la particularité la plus importante des cartes auto-organisatrices est qu'elles rendent possible la comparaison des groupements qui ont été réalisés directement à partir des données. Les observations proches dans l'espace de définition sont associées à des points proches dans la carte.

Les réseaux ART ("Adaptive Résonance Théory") sont des réseaux à apprentissage par compétition, issus des travaux de Stephen Grossberg et Gail Carpenter. Ils sont basés sur une théorie d'inspiration biologique assez complexe. Le réseau ART comme les cartes auto-organisatrices de Kohonen appartient à la famille des réseaux de neurone effectuant une classification non supervisée et dont les poids des interconnexions codent les prototypes des classes. Chaque classe est ainsi définie par son prototype et sa sélectivité. Elles se manifestent par différentes implantations spécifiques dont celles nommées «ART1», «ART2», «ART3», «fuzzy ART», «ARTmap», «fuzzy ARTmap», etc. Dans ce mémoire nous allons nous concentrer sur une quatrième architecture, également non supervisée, à savoir fuzzy ART.

Le problème majeur de la classification non supervisée est la détermination du nombre de classes optimale pour avoir une bonne qualité de classification. Ce problème peut être résolu avec un processus d'évaluation des résultats des algorithmes de classification appelé indice de validité des classes. Nous nous proposons d'étudier les avantages et les inconvénients de différents indices proposés dans la littérature pour définir d'éventuels meilleurs critères pour la classification non supervisée basé sur le principe neuronal.

Ce mémoire est organisé de la manière suivante

Le chapitre un consiste à définir la classification non supervisée ainsi que les différentes méthodes de classification existantes dans la littérature.

Le deuxième chapitre présentera les réseaux de neurone et une description détaillée des différentes méthodes appliquées et proposées à la classification non supervisées.

Le chapitre trois va présenter les différents critères de validité ainsi que la solution que nous proposons pour résoudre ce type de problèmes.

Le chapitre quatre est consacré à la présentation des résultats obtenus en appliquant les différents indices pour l'obtention du nombre de classes optimal dans les réseaux présentés dans le deuxième chapitre.

Ce mémoire se termine par une conclusion générale.

# **Chapitre 1:**

## **Classification non supervisée**

## 1. Introduction

La classification est une notion qui intervient fréquemment dans la vie courante. En effet, il est souhaitable de regrouper les éléments d'un ensemble hétérogène en un nombre restreint de classes, les plus homogènes possibles. Son application a joué un rôle très important pour résoudre plusieurs problèmes en reconnaissance des formes, imagerie, segmentation d'images couleur, data mining, et dans différents domaines comme la médecine, la psychologie, la biologie, etc.....

## 2. Qu'est-ce qu'une classification ?

Classer un ensemble d'objets, c'est attribuer à chacun une classe ou une « catégorie ». Cette tâche est appelée « classification ». Un algorithme qui réalise automatiquement une classification est appelé classifieur.

Les statisticiens appellent aussi « classification » la tâche qui consiste à regrouper des données qui se ressemblent dans des classes qui ne sont pas définies à l'avance [5].

Le concept de classification est étroitement lié à la notion de partition d'un ensemble fini. Cette opération a pour but de rassembler des pixels entre eux suivant des critères pré-définis et de séparer différentes zones homogènes d'une image, afin d'organiser les objets en groupes (clusters) dont les membres ont en commun diverses propriétés (intensité, couleur, texture, etc). On peut regrouper les méthodes de classification en deux catégories: classification supervisée et classification non supervisée.

Nous nous intéressons dans ce travail seulement à la classification non supervisée qu'on peut définir comme suivant :

La classification non supervisée ou classification automatique - clustering - est une étape importante de l'analyse de données. Elle consiste à identifier des groupes d'objets ou d'individus similaires - clusters - à partir d'un ensemble de données sans en connaître au préalable la structure. Elle ne doit pas être confondue avec la classification supervisée ou classement - classification - qui consiste à déterminer les règles qui permettent de séparer un ensemble d'individus en classes connues à priori.

### 2.1. Quelques définitions

Etant donné un ensemble  $\mathbf{O} = \{\mathbf{o}_1, \dots, \mathbf{o}_N\}$  de  $N$  objets correspondant chacun à un point d'un espace métrique à  $M$  dimensions dont les coordonnées sont notées par le vecteur  $\mathbf{x}_i = (x_{i1}, \dots, x_{iM})$  pour l'objet  $\mathbf{o}_i$ , on peut définir les notions suivantes :

**a. Les distances**

► **Mesure de dissimilarité** : On appelle indice ou mesure de dissimilarité sur un ensemble  $\mathbf{O}$  une application  $\mathbf{d} : \mathbf{O} \times \mathbf{O} \rightarrow \mathbf{R}^+$  qui vérifie les propriétés suivantes pour tout couple  $(x, y) \in \mathbf{O} \times \mathbf{O}$  :

$$d(x, y) = d(y, x) \text{ (symétrie)} \dots \dots \dots (1.1)$$

$$d(x, y) \geq 0 \dots \dots \dots (1.2)$$

$$d(x, y) = 0 \Leftrightarrow x = y \text{ (séparabilité)} \dots \dots \dots (1.3)$$

► **Métrie** : On appelle métrie sur un ensemble  $\mathbf{O}$ , une application  $\mathbf{d} : \mathbf{O} \times \mathbf{O} \rightarrow \mathbf{R}^+$  qui vérifie les propriétés suivantes pour tout couple  $(x, y) \in \mathbf{O} \times \mathbf{O}$  :

$$d(x, y) = d(y, x) \text{ (symétrie)}$$

$$d(x, y) = 0 \Leftrightarrow x = y \text{ (séparabilité)}$$

$$d(x, y) \leq d(x, z) + d(z, y) \text{ (inégalité triangulaire)} \dots \dots \dots (1.4)$$

► **Ultramétrie** : On appelle ultramétrie sur un ensemble  $\mathbf{O}$ , une application  $\mathbf{d} : \mathbf{O} \times \mathbf{O} \rightarrow \mathbf{R}^+$  qui vérifie les propriétés suivantes pour tout couple  $(x, y) \in \mathbf{O} \times \mathbf{O}$  :

$$d(x, y) = d(y, x) \text{ (symétrie)}$$

$$d(x, y) = 0 \Leftrightarrow x = y \text{ (séparabilité)}$$

$$d(x, y) \leq \max \{d(x, z), d(z, y)\} \text{ (inégalité ultramétrique)} \dots \dots (1.5)$$

où  $x, y$  et  $z$  sont des vecteurs.

L'indice de dissimilarité le plus communément utilisé est la métrie de Minkowski :

$$d_r(o_i, o_j) = d_r(x_i, x_j) = \left( \sum_{k=1}^M q_k |x_{ik} - x_{jk}|^r \right)^{\frac{1}{r}} \dots \dots \dots (1.6)$$

$q_k$  est un facteur de pondération. Suivant la valeur de  $r$  ( $r \geq 1$ ) on obtient les mesures suivantes :

- $r = 1$  : distance de Manhattan ;
- $r = 2$  : distance Euclidienne ;
- $r = \infty$  :  $d_\infty(x_i, x_j) = \max_{1 \leq k \leq M} |x_{ik} - x_{jk}|$ .

Ces mesures sont souvent utilisées pour des données numériques. Dans le cas de données symboliques, d'autres distances doivent être utilisées. La plus connue étant la distance de

Hamming, à l'origine définie pour mesurer la distance entre des chaînes binaires, et qui correspond au nombre de bits différents dans les deux chaînes pour une même position [6].

**b. L'erreur quadratique**

L'erreur quadratique est un critère des plus courants parmi les critères utilisés pour le partitionnement.

Considérons qu'une partition de  $\mathbf{O}$  a été obtenue sous la forme de  $K$  classes  $c_1, \dots, c_K$  composées respectivement de  $|c_1|, \dots, |c_K|$  objets. Comme chaque objet appartient à une et une seule classe on a :

$$\sum_{i=1}^k |c_i| = N \dots\dots\dots(1.7)$$

Le centre de gravité  $g_i$  de la classe  $c_i$  est donné par :

$$g_i = \frac{1}{|c_i|} \sum_{j=1}^{|c_i|} x_j^{(i)} \dots\dots\dots(1.8)$$

Où  $x_j^{(i)}$  est le  $j^{\text{ème}}$  point de la classe  $c_i$ . L'erreur quadratique  $\epsilon_i^2$  sur la classe  $c_i$  est

$$\epsilon_i^2 = \sum_{j=1}^{|c_i|} d^2(x_j^{(i)}, g_i) \dots\dots\dots(1.9)$$

L'erreur quadratique (aussi appelée inertie intraclasse :  $I_w$ ) de la partition est alors :

$$I_w = E_q^2 = \sum_{j=1}^k \epsilon_j^2 \dots\dots\dots(1.10)$$

Bien évidemment, ce critère ne peut servir à comparer des partitions ayant un nombre de classe différent. En effet, si  $K = N$  alors  $E_q = 0$  car  $x_i = g_i \forall i \in \{1, \dots, N\}$  et l'erreur quadratique est minimale.

Le vecteur moyen  $g$  de l'ensemble  $\mathbf{O}$  est donné par :

$$g = \frac{1}{N} \sum_{i=1}^k |c_i| g_i \dots\dots\dots(1.11)$$

Enfin, l'inertie interclasse, notée  $I_B$ , est donnée par :

$$I_B = \sum_{i=1}^k |c_i| d^2(g_i, g) \dots\dots\dots(1.12)$$

L'inertie totale  $I$  :

$$I = I_B + I_w \dots\dots\dots(1.13)$$

Est stable pour un nombre de classes  $K$  fixé.

### 3. Les différentes approches de la classification non supervisée

Devant un problème défini de façon aussi imparfaite, il était naturel de voir apparaître un grand nombre de techniques, souvent de nature heuristique. On peut aujourd'hui les regrouper en deux grandes familles : la classification par partitionnement, et la classification hiérarchique.

#### 3.1 Classification par partitionnement

##### 3.1.1 Partitionnement "doux"

L'idée selon laquelle chacune des classes réelles, sous-jacentes, occupe une région limitée de l'espace peut paraître irréaliste. En particulier, l'analyse discriminante nous a habitués à penser en termes de classes ayant des distributions multinormales, et donc se chevauchant nécessairement. Il est donc naturel de considérer la possibilité que les classes empiètent les unes sur les autres.

##### ► Mélanges de modèles

Cette approche reprend l'idée selon laquelle les classes ont des distributions multinormales. Pour un nombre supposé de classes  $K$  donné, on considèrera donc que l'ensemble des données est en fait un échantillon tiré de  $K$  distributions multinormales, chacune affectée d'une probabilité a priori. On cherche alors les centres et matrices de covariances de ces distributions, ainsi que les probabilités a priori, de façon à maximiser la vraisemblance des données, ce qui se fait par l'algorithme classique dit "EM" (Expectation Maximization) introduit par Dempster et al. en 1977 [7]. Il consiste à itérer les deux phases : estimation et maximisation jusqu'à ce que l'amélioration de la log vraisemblance du modèle soit inférieure à un seuil  $\varepsilon > 0$ .

##### ► Classification floue

Dans les mélanges de modèles, les classes se chevauchent mais chaque observation appartient à une classe et une seule (bien que la détermination de cette classe ne puisse être faite que de façon probabiliste). En classification floue, à l'inverse, on reconnaît qu'une observation peut effectivement appartenir simultanément à plusieurs classes à des degrés divers dont la somme est égale à 1. Cette attitude reflète le fait que dans de nombreuses situations réelles, les définitions des classes (supervisées ou non) ne rendent pas obligatoire l'appartenance à une classe et une seule (exemple des classes : "Chaud" et "Très chaud"). La classification floue affecte alors à chaque observation des degrés d'appartenance aux

diverses classes d'une façon cohérente avec la répartition géométrique des données. Pour chaque observation, la somme des degrés d'appartenance à chacune des classes est égale à 1

### 3.1.2 Partitionnement "dur"

L'idée générale est de découper l'espace des observations en un certain nombre de régions disjointes, définies par des frontières, et de décréter que toutes les observations situées dans une même région de l'espace appartiennent à une même classe.

Chaque classe est représentée par un "prototype", observation virtuelle sensée être la plus représentative de la population de la classe. Le prototype d'une classe sera le plus souvent le barycentre des observations de la classe.

Ces prototypes sont positionnés de façon itérative dans les zones à forte densité, et les observations sont affectées aux classes sur la base d'un critère de proximité aux différents prototypes.

Il existe de nombreuses techniques de classification par partitionnement, la plus connue étant *K*-means (ou "*K*-moyennes") et ses variantes (voir les détails dans le chapitre suivant). Notez qu'un type très particulier de réseaux de neurones, les cartes de Kohonen (ou "SOM") peut être perçu comme une technique de partitionnement puisque cherchant à donner, dans la mesure du possible, une représentation plane des classes qui respecte leurs positionnements relatifs dans l'espace des données. Parmi les méthodes basées sur SOM est SOM à deux niveaux que nous développerons au chapitre suivant.

## 3.2. Classification hiérarchique

La classification hiérarchique est une famille de techniques qui génèrent des suites de partitions emboîtées les unes dans les autres, et allant depuis la partition triviale à une seule classe (contenant toutes les observations) jusqu'à la partition triviale où chaque observation est une classe. Entre ces deux extrêmes figurent de nombreuses partitions plus réalistes entre lesquelles l'analyste devra choisir. On distingue deux types d'approches de classification hiérarchique : les méthodes descendantes – *divisives* - et les méthodes ascendantes *agglomératives*.

### ► Méthodes descendantes

Elles considèrent l'ensemble des observations  $\mathbf{O}$  et procèdent par division successive jusqu'à obtenir une partition formée de singletons. Nous ne détaillerons pas d'avantage ces méthodes qui sont trop coûteuses pour être utilisées sur les volumes de données manipulés aujourd'hui. En effet, la division d'une partie à  $N$  éléments nécessite l'évaluation des  $(2^{N-1} - 1)$

divisions possibles.

► **Méthodes ascendantes**

Elles commencent avec la partition de l'ensemble  $O$  la plus fine et procèdent ensuite par fusion progressive des parties jusqu'à obtention de la partition la plus grossière. On obtient ainsi un arbre binaire dont la racine correspond à la partition ne comportant qu'une seule partie et dont les feuilles s'identifient aux différents singletons. Les différents noeuds intermédiaires correspondent à la fusion de deux parties.

La Classification Ascendante Hiérarchique (CAH) est sans doute la méthode la plus largement utilisée de cette catégorie. Différents types de distances inter et intra classes ont été utilisées et on peut les résumer dans le tableau ci-dessous :

Définition	Distance inter classes
Chaînage simple	$d_s = \min_{i,j} \{ \ x_i - x_j\  \}$
Chaînage complet	$d_{co} = \max_{i,j} \{ \ x_i - x_j\  \}$
Chaînage centre	$d_{ce} = \ c_k - c_l\ $
Chaînage moyen	$d_a = \frac{\sum_{i,j} \ x_i - x_j\ }{N_i N_j}$
Définition	Distance intra classes
Distance Moyenne	$s_a = \frac{\sum_{i,i'} \ x_i - x_{i'}\ }{N_k (N_k - 1)}$
Distance au plus proche voisin	$s_{nn} = \frac{\sum_i \min_{i'} \{ \ x_i - x_{i'}\  \}}{N_k}$
Distance centre	$s_c = \frac{\sum_i \ x_i - c_k\ }{N_k}$

**Tableau 1.1** : Les différentes distances.

Où  $x_i$  : un élément de la classe  $i$

$N_i$  : le nombre des éléments associés à la classe  $i$ .

$$c_k = \frac{1}{k} \sum_{x_i \in Q_k} x_i \quad / \quad Q \text{ représente une classe.}$$

## **4. Connaissances du domaine et contraintes**

La classification non supervisée permet de former des groupes d'objets susceptibles d'être intéressants pour l'utilisateur. Notons qu'il est fréquemment possible de construire différentes partitions d'un même ensemble d'objets et en absence d'informations complémentaires, le choix de l'une ou l'autre est nécessairement arbitraire. La prise en compte des attentes de l'utilisateur est donc un facteur de succès déterminant de l'application de ce type de méthodes. Nous rappelons dans cette section différentes approches proposées à cet effet ; elles procèdent par introduction de contraintes qui portent sur les groupes, sur les objets ou encore sur les attributs.

### **4.1 Contraintes sur les groupes : forme et taille**

#### **4.1.1 Contraintes de forme**

La forme des groupes est très souvent imposée par le choix de l'algorithme et de la mesure de (dis) similarité. Ainsi, l'algorithme des K-moyennes utilisant une distance euclidienne a tendance à former des groupes hyper-sphériques. Plus généralement, les modèles de mélange permettent d'imposer la forme du nuage de points des différentes sous population en contraignant les paramètres des différentes lois ; dans le cas d'un mélange de lois normales, il est commun d'imposer à la matrice de covariance d'être diagonale : les groupes formés sont ainsi hyper-ellipsodaux.

Dans certaines applications, les données revêtent un caractère spatial et il est parfois nécessaire d'obtenir des groupes contigus. La dimension spatiale peut être utilisée soit en ajoutant des variables de position que l'on traite ensuite comme les autres descripteurs, soit en utilisant une phase d'extraction de l'information spatiale en remplaçant par exemple la valeur d'un attribut par sa moyenne dans le voisinage (au sens spatial) de l'objet.

#### **4.1.2 Contraintes de taille**

Il est parfois souhaitable d'obtenir des groupes de taille plus ou moins homogène et il est possible d'introduire des contraintes sur la taille des clusters. Il est relativement simple d'imposer une contrainte de ce type à une hiérarchie de partitions, cela revient en effet à définir une hauteur maximale de coupure à chaque branche du dendrogramme. Pour finir, dans le contexte des modèles de mélanges une telle contrainte peut être appliquée en ajoutant un terme de régularisation qui fixe une probabilité à priori maximale qu'aucune classe ne peut dépasser.

## 4.2 Contraintes sur les objets

### 4.2.1 Fusion et Exclusion

Du point de vue de l'utilisateur, un moyen simple de préciser la partition qu'il attend consiste à indiquer quels sont les objets qui doivent être regroupés et quels sont ceux qui doivent s'exclure mutuellement.

Ainsi, lors d'un apprentissage actif, l'utilisateur peut affiner progressivement le résultat d'une classification automatique en précisant progressivement les regroupements ou séparations d'objets qu'il considère comme des anomalies.

### 4.2.2 Etiquetage partiel

Lorsqu'une partition d'un sous-ensemble des objets est connue, il est possible d'appliquer les approches décrites ci-dessus. L'algorithme des K-moyennes est appliqué normalement sans tenir compte des étiquettes connues.

## 4.3 Contraintes sur les attributs

Il est parfois souhaitable d'obtenir des groupes dans lesquels les valeurs prises par un attribut restent dans un intervalle de faible amplitude. C'est dans ce cadre que [7] donne des propositions de deux versions de la CAH utilisant l'indice du saut maximum comme critère d'agrégation. La première version, *Constrained Clustering with Complete-Link*, qui procède aux regroupements des objets en respectant la contrainte, est sensible à l'ordre des regroupements et laisse généralement de côté certains objets qui ne peuvent être ajoutés à aucun groupe sans violer la contrainte. Une deuxième version, *Progressive Constraint Relaxation Technique*, est proposée pour corriger ce problème. La contrainte imposée à l'intervalle de valeur dépend alors du niveau auquel le regroupement intervient dans la hiérarchie ; elle est relâchée progressivement.

## 5. Nombre de classes

Toutes les méthodes de classification non supervisées permettent à l'analyste de choisir le nombre de classes de la partition finale (niveau de la coupure de l'arbre en classification hiérarchique, nombre de prototypes en *K-means* etc...).

Cette possibilité contient des avantages et aussi des inconvénients. Si la population contient effectivement des clusters bien identifiables, mais si le choix du nombre de classes finales ne correspond pas au nombre de clusters, alors la partition :

- Soit regroupera à tort des clusters séparés.
- Soit découpera en plusieurs classes des clusters homogènes.

La recherche du nombre approprié de classes est toujours une phase indispensable dans la construction d'une classification de données, mais elle est longue et souvent ambiguë. Il n'y a pas de formule permettant de calculer ce nombre à partir des données. La recherche se fait par essais et erreurs. Une même technique est utilisée à plusieurs reprises avec un nombre croissant de classes, et pour chaque nouvelle partition obtenue, on calcule la valeur d'un critère de qualité. Le nombre de classes retenu est celui qui conduit à la meilleure valeur de ce critère.

## 6. Qualité d'une partition

Un algorithme de classification non supervisée produira toujours des classes. Mais ces classes reflètent-elles la structure des données ? Chaque classe abrite-t-elle un cluster dense et ses environs immédiats, comme le souhaite l'analyste ?

Il y a deux raisons pour lesquelles une typologie peut être non satisfaisante :

- Les données ne sont que faiblement structurées. Elles ne contiennent pas de cluster bien identifiable, et sont plus ou moins uniformément réparties.
- Il existe des clusters bien identifiables, mais l'analyse n'a pas réussi à les mettre en évidence.

Il est donc indispensable de pouvoir quantifier la qualité d'une partition, c'est à dire d'apprécier le fait que la partition correspond à des clusters présents dans les données. Il existe de nombreux critères de qualité d'une partition, ce qui signifie qu'aucun n'est universellement satisfaisant.

## 7. Distance et densité

Les classes rencontrées dans les applications ont souvent des distributions unimodales : à partir d'un noyau central, la densité des observations décroît de façon monotone dans toutes les directions de l'espace. Beaucoup de techniques de classification non supervisée s'appuient sur cette image et portent une attention particulière aux ensembles d'observations ayant entre elles de faibles distances (régions de forte densité). Elles le font de diverses façons :

- En utilisant les distances entre observations pour construire les classes (p. ex. méthodes hiérarchiques).
- En reconnaissant que les zones peuplées mais de faible inertie autour de leurs barycentres sont des zones de forte densité (*K-means*).
- En faisant de l'estimation de densité de façon paramétrique (modèles de mélanges) ou non paramétriques (méthodes basées sur l'estimation de densité par *K-proches*

Voisins).

## **8. Conclusion**

Nous avons expliqué dans ce chapitre les notions de la classification non supervisée ainsi leurs méthodes classiques. Dans ce mémoire nous nous intéressons aux méthodes de classification non supervisée neuronales qu'ont été bien expliquées dans le chapitre suivant.

# **Chapitre 2:**

# **Réseaux de neurones**

## 1. Introduction

La classification est un problème sur lequel les réseaux de neurones donnent de bons résultats et s'appliquent avec succès. Les réseaux de neurones à apprentissage non supervisé peuvent réaliser ce genre de tâches.

Les réseaux de neurones, fabriqués de structures cellulaires artificielles, constituent une approche permettant d'aborder sous des angles nouveaux les problèmes de perception, de mémoire, d'apprentissage et de raisonnement. Ils s'avèrent aussi des alternatives très prometteuses pour contourner certaines des limitations des ordinateurs classiques. Grâce à leur traitement parallèle de l'information et à leurs mécanismes inspirés des cellules nerveuses (neurones), ils infèrent des propriétés émergentes permettant de solutionner des problèmes jadis qualifiés de complexes.

Ce chapitre présente une brève généralité sur les réseaux de neurones ainsi que les architectures des deux types de réseaux de neurones à apprentissage non supervisé pour la classification.

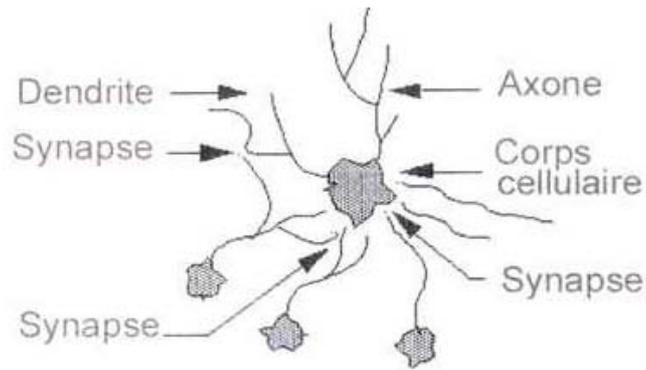
## 2. Présentation des réseaux de neurones

### 2.1 Définition :

Un réseau de neurone artificiel est un modèle de calcul dont l'inspiration originelle était un modèle biologique, c'est-à-dire le modèle du cerveau nerveux humain. Le cerveau humain est composé d'un grand nombre des cellules nerveuses. La cellule nerveuse humaine contient des neurones.

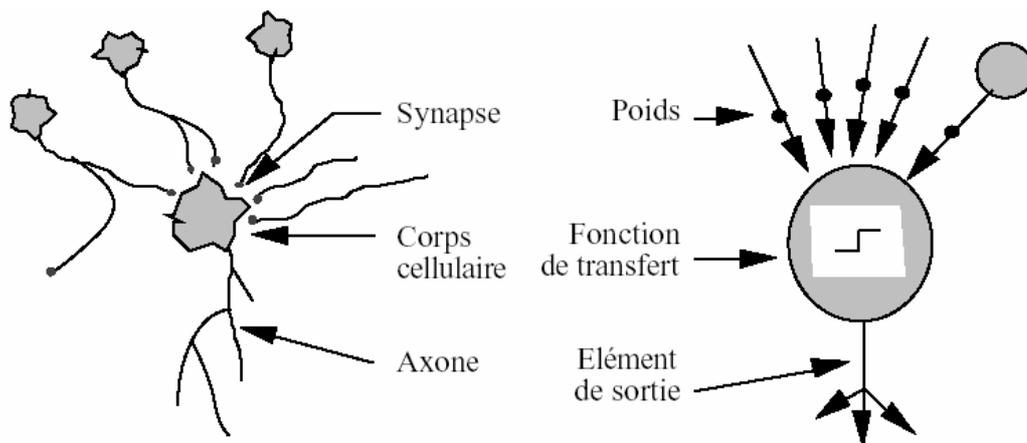
#### **Qu'est-ce qu'un neurone artificiel ?**

Les neurones sont des unités nerveuses de base composées d'une cellule et d'un noyau. Les neurones ont la propriété de toutes les cellules vivantes. Les neurones se distinguent des autres cellules car ils ont des axones. Les axones sont des câbles, des fibres nerveuses qui conduisent l'influx nerveux. Chaque neurone est relié aux autres neurones par des connexions nerveuses : les synapses. L'ensemble des neurones et des connexions constitue le réseau.(figure 2.1)



**Figure 2.1** : Schéma d'un neurone biologique.

Les neurones formels sont utilisés en informatique dans le domaine de l'intelligence artificielle pour modéliser et résoudre des problèmes complexes. (La figure 2.2) montre une comparaison entre le neurone biologique et le neurone artificiel. Dans la cellule nerveuse humaine, la synapse correspond au poids d'un neurone artificiel, le corps cellulaire à une fonction de transfert et l'axone à un élément de sortie.



**Figure 2.2** : La comparaison entre un neurone biologique et un neurone artificiel.

Donc un réseau de neurones artificiels est un ensemble de processeurs élémentaires, les neurones, reliés entre eux et qui échangent des informations par ces liaisons. L'interaction entre ces différents éléments donne au réseau son comportement global. On appelle « état » du neurone la valeur de sortie de ce neurone. Un neurone calcule son état à partir des informations qui lui parviennent de l'extérieur du réseau ou bien de l'état des neurones auxquels il est connecté.

Les réseaux de neurones constituent des modèles paramétrés dont les paramètres sont estimés

au cours d'une phase dite « d'apprentissage » à partir d'un ensemble d'observations statistiquement représentatives du problème à traiter (base d'apprentissage); ces modèles permettent certaines fonctions du cerveau, comme la mémorisation associative l'apprentissage par l'exemple, le travail en parallèle, etc. Cependant le neurone formel ne possède pas toutes les capacités des neurones biologiques, comme le partage de synapses, l'activation membranaire ou la structuration prénatale des neurones. Ces modèles prennent en compte quelques grands principes :

- **parallélisme** : les neurones sont des entités réalisant des fonctions très simples, mais très fortement interconnectées ce qui rend le traitement du signal massivement parallèle.
- **poids synaptiques** : les connections entre neurones ont des poids variables, ce qui rend les neurones plus ou moins influents sur d'autres neurones.
- **apprentissage** : on appelle « apprentissage » des réseaux de neurones qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui lui est affectée.

## 2.2 Historique :

L'histoire des réseaux de neurones est tissée à travers des découvertes conceptuelles et des développements technologiques survenus à diverses époques.

Brièvement, les premières recherches remontent à la fin du 19<sup>e</sup> et au début du 20<sup>e</sup> siècle. Ils consistent en de travaux multidisciplinaires en physique, en psychologie et en neurophysiologie par des scientifiques tels Hermann von Helmholtz, Ernst Mach et Ivan Pavlov [8]. A cette époque, il s'agissait de théories plutôt générales sans modèle mathématique précis d'un neurone. On s'entend pour dire que la naissance du domaine des réseaux de neurones artificiels remonte aux années 1940 avec les travaux de Warren McCulloch et Walter Pitts [9] qui ont montré qu'avec de tels réseaux, on pouvait, en principe, calculer n'importe quelle fonction arithmétique ou logique. Vers la fin des années 1940, Donald Hebb a ensuite proposé une théorie fondamentale pour l'apprentissage. Plus tard, en 1949, Hebb [10] propose une formulation du mécanisme d'apprentissage, sous la forme d'une règle de modification des connexions synaptiques (règle de Hebb).

Le premier réseau de neurones artificiels apparaît à la fin des années 1950, grâce aux travaux de Rosenblatt [11] qui conçoit le fameux Perceptron. C'est le premier système artificiel

capable d'apprendre par expérience, y compris lorsque son instructeur commet quelques erreurs (ce en quoi il diffère nettement d'un système d'apprentissage logique formel).

Malgré tout l'enthousiasme que soulève le travail de Rosenblatt dans le début des années 60 la fin de cette décennie sera marquée en 1969, par une critique violente du Perceptron par Minsky et Papert. Ils montrent dans un livre (« Perceptrons ») toutes les limites de ce modèle, et soulèvent particulièrement l'incapacité du Perceptron à résoudre les problèmes non linéairement séparables, tels que le célèbre problème du XOR (OU exclusif). Il s'en suivra alors, face à la déception, une période noire d'une quinzaine d'années dans le domaine des réseaux de neurones artificiels.

Heureusement, certains chercheurs ont persévéré en développant de nouvelles architectures et de nouveaux algorithmes plus puissants. En 1972, Teuvo Kohonen et James Anderson ont développé indépendamment et simultanément de nouveaux réseaux pouvant servir de mémoires associatives, également, Stephen Grossberg a investigué ce qu'on appelle les réseaux auto organisés[8].

Dans les années 1980, ces problèmes ont été résolus par l'invention de l'algorithme de rétro propagation des erreurs. Cette algorithme est la réponse aux critiques de Minsky et Papert formulées à la fin des années 1960. C'est ce nouveau développement, généralement attribué à David Rumelhart et James McClelland, mais aussi découvert plus ou moins en même temps par Paul Werbos et par Yann Le Cun, qui a littéralement ressuscité le domaine des réseaux de neurones. Depuis ce temps, c'est un domaine où bouillonnent constamment de nouvelles théories, de nouvelles structures et de nouveaux algorithmes.

### 2.3 Structure du réseau

Un réseau de neurone est en général composé d'une succession de couches dont chacune prend ses entrées sur les sorties de la précédente. Chaque couche (i) est composée de  $N_i$  neurones, prenant leurs entrées sur les  $N_{i-1}$  neurones de la couche précédente.

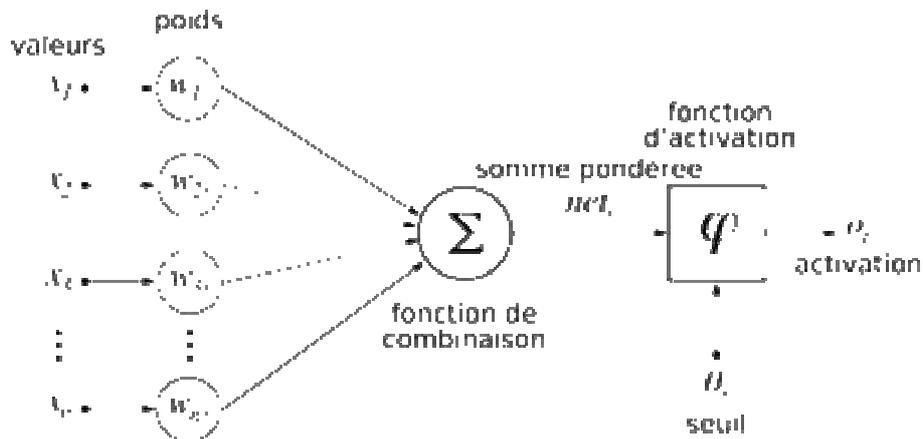
En général le système de neurone possède trois types des neurones :

- les neurones d'entrée qui reçoivent les données.
- les neurones de sortie qui envoient les données par la sortie du système .
- les neurones cachés, dont les signaux d'entrée et de sortie demeurent dans le système.

Dans les réseaux de neurones artificiels on utilise une modélisation mathématique du neurone biologique, appelé neurone formel. Chaque neurone reçoit des valeurs d'entrées (ou des signaux d'entrée) par son neurone voisin et les utilise pour calculer le signal de sortie qui se

propage par les autres neurones. En concordance avec ce processus on doit ajuster les poids des connexions synaptiques.

Sur la figure 2.3 on peut voir la structure d'un neurone artificiel. Le neurone calcule la somme de ses entrées puis cette valeur passe à travers la fonction d'activation pour produire sa sortie.



**Figure 2.3** : Structure d'un neurone artificiel.

## 2.4 Fonctions de transfert

La fonction de transfert génère alors la sortie grâce à ce potentiel. Cette fonction de transfert est très importante, et détermine le fonctionnement du neurone et du réseau. Elle peut prendre de nombreuses formes, peut être linéaire, binaire, saturée ou sigmoïdale.

Le neurone se décompose en deux parties :

- évaluation de la stimulation reçue : fonction d'entrée.
- évaluation de l'état de sa sortie (son état interne) : fonction de transfert (ou fonction d'activation).

Le tableau 2.1 récapitule les différents types de fonctions de transfert, les plus utilisées avec leurs équations mathématiques et leurs dérivées.

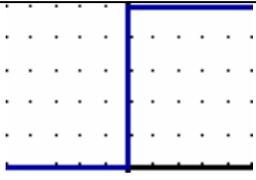
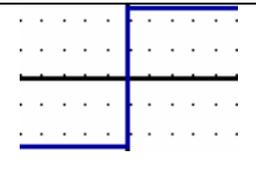
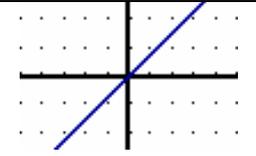
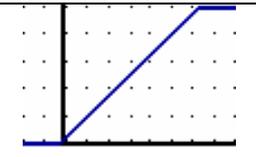
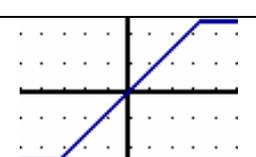
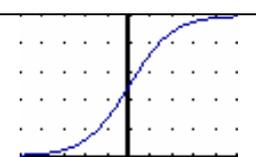
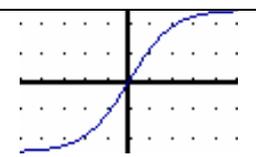
Catégorie	Type	Equation	Allure	Dérivée
Seuil	<b>Binaire (fonction de Heaviside)</b>	$f(x) = 1$ si $x > 0$ $f(x) = 0$ si $x \leq 0$		-
	<b>Signe</b>	$f(x) = 1$ si $x > 0$ $f(x) = -1$ si $x \leq 0$		-
Linéaire	<b>Identité</b>	$f(x) = x$		$f'(x) = 1$
	<b>Saturé positif</b>	$f(x,k) = 0$ si $x < 0$ $f(x,k) = 1$ si $x \geq 1/k$ $f(x,k) = k \cdot x$ sinon		$f(x,k) = 0$ si $x < 0$ $f(x,k) = 0$ si $x \geq 1/k$ $f(x,k) = k$ sinon
	<b>Saturé symétrique</b>	$f(x,k) = -1$ si $x < -1/k$ $f(x,k) = 1$ si $x \geq 1/k$ $f(x,k) = k \cdot x$ sinon		$f(x,k) = 0$ si $x < -1/k$ $f(x,k) = 0$ si $x \geq 1/k$ $f(x,k) = k$ sinon
Sigmoide	<b>Positive (type logistique)</b>	$f(x, k) = \frac{1}{1 + e^{-k \cdot x}}$		$f'(x, k) = \frac{k}{2 + e^{-k \cdot x} + e^{k \cdot x}}$
	<b>Symétrique (type tanh)</b>	$f(x, k) = \frac{2}{1 + e^{-k \cdot x}} - 1$		$f'(x, k) = \frac{2k}{2 + e^{-k \cdot x} + e^{k \cdot x}}$

Tableau 2.1 – Fonctions de transfert a = f(n).

## 2.5 Différents types d'apprentissage

Une caractéristique des réseaux de neurones est leur capacité à apprendre (par exemple à reconnaître une lettre, un son...). Mais cette connaissance n'est pas acquise dès le départ. La plupart des réseaux de neurones apprennent par l'exemple (de la même manière qu'un enfant apprend à reconnaître un chien à partir d'exemples de chiens qu'il a vu). Ils ont donc une capacité à classer, généraliser, à mémoriser et aussi à oublier. Un réseau de neurones reconnaîtra d'autant plus facilement un objet qu'il l'aura " vu " souvent. Les réseaux de neurones se divisent en deux principales classes, les réseaux à apprentissage supervisés (supervised learning) et les réseaux à apprentissage non supervisés (unsupervised learning).

**2.5.1 Apprentissage supervisé :** Pour les réseaux à apprentissage supervisé (Perceptron, Adaline, etc. ), on présente au réseau des entrées, et au même temps les sorties que l'on désirerait pour cette entrée. Par exemple on lui présente en entrée une lettre " a " manuscrite et en sortie un code correspondant à la lettre " a ". Le réseau doit alors se reconfigurer, c'est-à-dire calculer ses poids afin que la sortie qu'il donne corresponde bien à la sortie désirée d'une autre façon : on fournit une série d'exemples  $x$  et de résultats  $y$ , l'algorithme doit trouver  $W$  tel que  $y = F_W(x)$ .

**2.5.2 Apprentissage non supervisé :** Pour les réseaux à apprentissage non supervisé (Hopfield, Kohonen, etc.), on présente une entrée au réseau et on le laisse évoluer librement jusqu'à ce qu'il se stabilise.

On fournit seulement des exemples  $x$  à l'algorithme il doit trouver  $W$  tel que les  $x$  soient correctement groupés selon  $F_W$ .

## 2.6 Applications

Les réseaux de neurones servent aujourd'hui à toutes sortes d'applications dans divers domaines. Par exemple, on a développé un auto-pilote pour avion, ou encore un système de guidage pour automobile. On a conçu des systèmes de lecture automatique de chèques bancaires et d'adresses postales. On produit des systèmes de traitement du signal pour différentes applications militaires, un système pour la synthèse de la parole, des réseaux sont aussi utilisés pour bâtir des systèmes de vision par ordinateur, pour faire des prévisions sur les marchés monétaires, pour évaluer le risque financier ou en assurance, pour différents processus manufacturiers, pour le diagnostic médical, pour l'exploration pétrolière ou gazière en robotique, en télécommunication. Parmi les différentes applications réalisées à l'aide des réseaux de neurones et sur tout les carte auto-organisatrices et l'architecture ART, un assez

grand nombre sont des tâches de classification. Comme nous venons de l'écrire dans les sections suivantes.

## 2.7 Architecture

On appelle topologie des réseaux la façon dont les neurones sont connectés entre eux à travers les différentes couches. En général on peut distinguer deux grands groupes des réseaux de neurones selon leur topologie : **les réseaux à couches** et **les réseaux récurrents** (cf. figure 2.4).

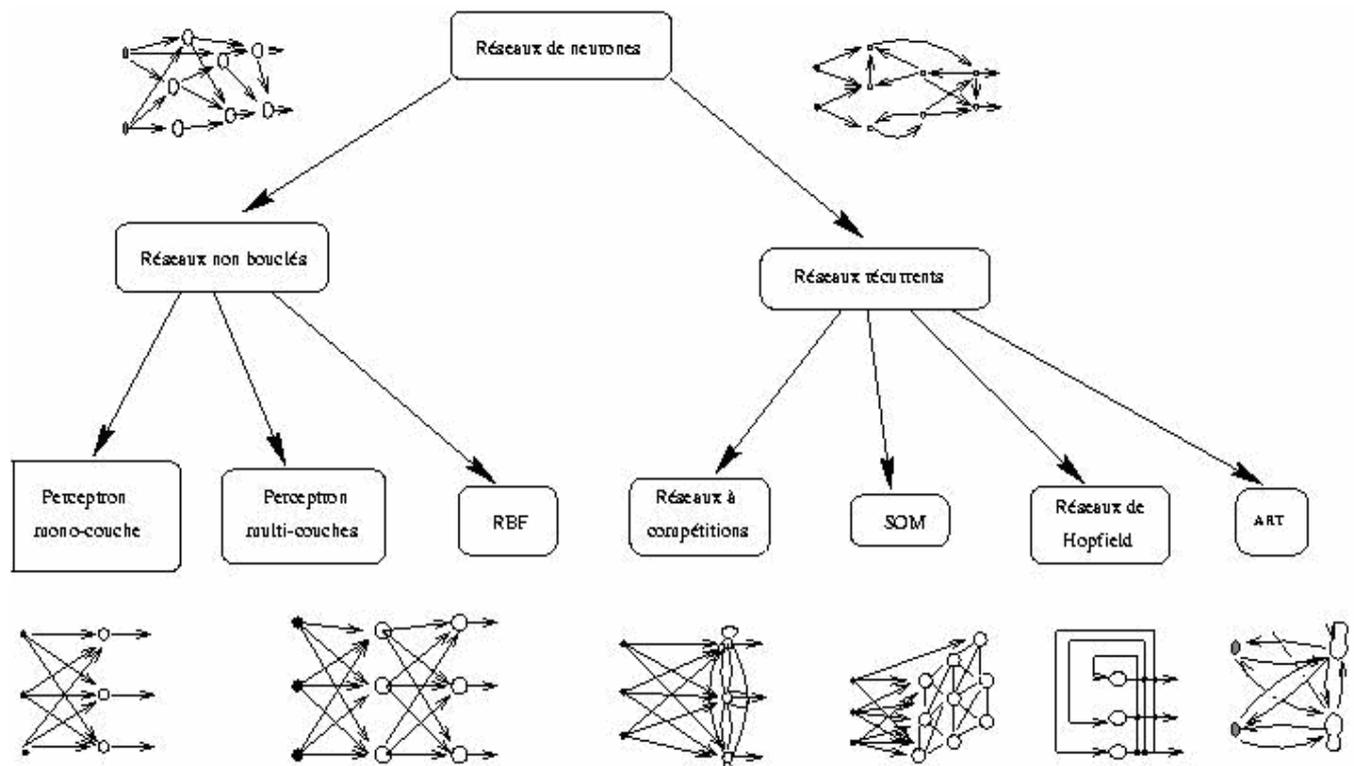


Figure 2.4 : La topologie des réseaux de neurones.

Nous nous intéressons aux deux types de réseaux qui ont été utilisés pour la classification non supervisée.

### 3. Carte auto organisatrice:

#### 3.1 K\_means

L'algorithme des K\_means est un algorithme de classification automatique très ancien qui est à l'origine des cartes auto\_organisatrices [5].

La méthode des k-means est un outil de classification classique qui permet de répartir un ensemble de données en  $k$  classes homogènes. La plupart des images (photos, dessins

vectoriels 2D, synthèses 3D, ...) vérifient localement des propriétés d'homogénéité notamment en terme d'intensité lumineuse. L'algorithme des *k-means* permet donc d'apporter une solution à la segmentation d'images.

### Description de l'algorithme

L'algorithme *k-means* est l'algorithme de clustering le plus connu et le plus utilisé, du fait de sa simplicité de mise en œuvre. Il partitionne les données d'une image en  $k$  classes. L'algorithme renvoie une partition des données, dans laquelle les objets à l'intérieur de chaque cluster sont aussi proches que possible les uns des autres et aussi loin que possible des objets des autres clusters. Chaque cluster de la partition est défini par ses objets et son centroïde. Le *k-means* est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centroïdes conditionne le résultat final de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme. *K-means* change les objets de la classe jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de classe compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur  $K$  du nombre des classes. Sur le plan algorithmique, l'algorithme des *K-means* se résume de la manière suivante :

---

#### Algorithme

---

- 1- Phase d'initialisation** :  $t = 0$  , choisir aléatoirement la position initiale des  $K$  cluster, fixer le nombre maximal d'itérations  $N$
  - 2- Phase d'affectation** : (Ré-) Affecter les objets à un cluster suivant un critère de minimisation des distances (généralement selon une mesure de distance euclidienne).
  - 3- répéter** la phase 2 jusqu'à ce que l'on atteigne  $k > N$  ou une stabilisation.
- 

### 3.2 Carte auto\_organisatrices de Kohonen (SOM)

Les cartes de Kohonen (self-organizing maps ou SOM) dans la littérature [12] peuvent être définies comme un algorithme de classification non supervisé issu du domaine des réseaux de neurones artificiels.

L'algorithme des "cartes auto-organisées", ou "cartes topologiques" a été introduit par Kohonen au début des années 80 pour modéliser un phénomène, couramment observé dans le cerveau : la formation de "cartes".

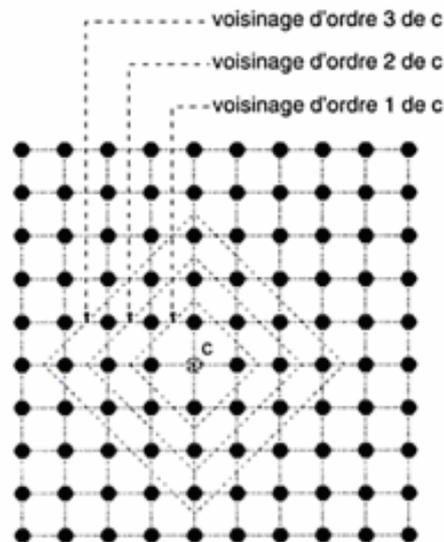
### 3.2.1 Principe :

Dans le cortex cérébral, on peut remarquer une organisation en régions qui correspondent à différentes modalités sensorielles : pour chaque région corticale, la structure topologique est la même que la structure topologique du capteur correspondant. On a ainsi des cartes rétino-topiques, somato-sensorielles, etc. Ces cartes se distinguent par la propriété commune suivante : pour un espace de données fixé, par exemple les signaux lumineux sur la rétine, la carte corticale est une représentation à deux dimensions telle que des données “voisines” aient des représentations voisines donc, un certain nombre des fonctions sensorielles sont dépendantes de la réalisation d’applications qui conservent la topologie entre l’espace des données (sur les capteurs) et l’espace des représentations (dans la zone corticale correspondante).

Du point de vue informatique, on peut traduire cette propriété de la façon suivante : supposons que l’on dispose de données que l’on désire classifier. On cherche un mode de représentation tel que des données voisines soient classées dans la même classe ou dans des classes voisines. L’algorithme proposé par Kohonen produit un réseau qui a cette propriété [12]. Cette algorithme projette l’espace des données  $X$  sur un espace discret de faible dimension. Cet espace, qu’on appelle la « carte », sera noté  $A$  dans la suite de ce travail. L’ensemble  $A$  est constitué par un ensemble de neurones interconnectés, les liens entre neurones se faisant par l’intermédiaire d’une structure de graphe non orienté. La structure de graphe induit une distance discrète  $\delta$  sur  $A$  : pour toute paire de neurones  $(c,r)$  de cette « carte »,  $\delta(c,r)$  est définie comme étant la longueur du plus court chemin entre le neurone  $c$  et  $r$ . Cette distance permet de définir la notion de voisinage d’ordre  $d$  de  $c$ .

La notion de distance entre neurones ou de voisinage découle alors directement de cette structure graphique, et définit la topologie discrète de la carte.

La figure 2.5 présente l’ensemble de cette notion de distance et de voisinage pour une carte topologique constituée par un treillis à deux dimensions. Chaque point de cette figure représente un neurone  $c$ . La distance  $\delta$  entre deux neurones est définie sur le maillage. Cette figure permet de définir le voisinage d’ordre  $d$  (ici  $d = 1, 2$  ou  $3$ ), qui représente l’ensemble des neurones dont la distance au neurone  $c$  est inférieure ou égale à  $d$ .



**Figure 2.5 :** Représentation de la topologie discrète d'une carte à deux dimensions constituées de  $10 * 10$  neurones.

### 3.2.2 Architecture

Les algorithmes d'apprentissage permettent de déterminer l'ensemble de tous les vecteurs de poids  $W$  d'une carte auto organisatrice. On peut représenter l'ensemble constitué par la carte et les vecteurs du poids sous la forme d'un réseau de neurone constitué de deux couches figure (2.6) :

La couche d'entrée : sert à la présentation des observation à classer ; les états de tous ses 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 (topologique) : est formé du treillis des neurones qui forme la carte. Chaque neurone  $i$  de la couche topologique est totalement connecté aux neurones de la couche d'entrée. Le vecteur poids  $w = \{w_{1i}, \dots, w_{ni}\}$  de ces connexions forme le référent ou le prototype associé au neurones. Il est de la même dimension que les formes d'entrées. La carte auto-organisatrice permettent de faire une représentation simplifiée des données décrites dans un espace de grande dimension  $X$ . Cette représentation simplifiée se fait dans un espace discret  $A$  de petite dimension qui est définit par une structure de graphe. Le problème est donc de choisir une structure de carte bien adaptée au problème particulier qui est traité. L'espace des neurone  $c$  de la carte sont liés par deux application : une application de  $A$  dans  $X$  qui permet d'affecter un neurone  $c$  de la carte un référent  $w_c$  qui est une observation «moyenne», et une fonction  $\Phi$  qui permet d'associer à tout vecteur de  $X$  un neurone  $c$  de la carte qui est son représentant. Ces deux application doivent vérifier la propriété de conservation de topologie.

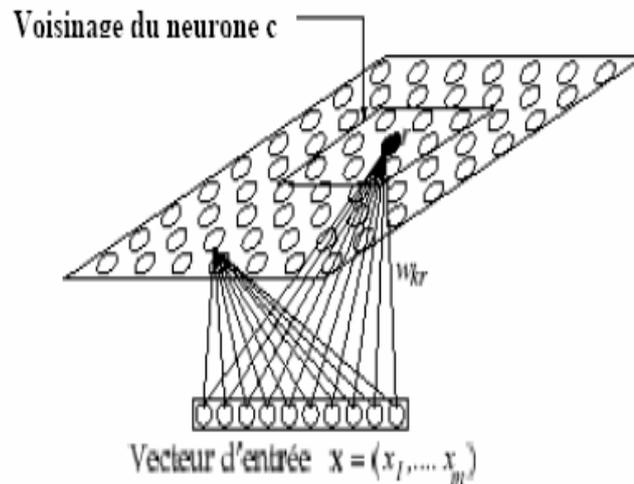


Figure 2.6: Architecture d'une carte topologique en 2-D.

**3.2.3 Règle d'apprentissage :**

Une étape d'apprentissage s'effectue en choisissant aléatoirement un vecteur de l'ensemble d'apprentissage X. Ce vecteur est ensuite comparé à chacun des vecteurs poids.

Pendant la phase d'apprentissage le processus d'auto-organisation permet de concentrer l'adaptation des poids des connexions essentiellement sur la région de la carte la plus 'active'.

Le neurone gagnant  $i(x)$  est ensuite identifié à l'aide d'une mesure de similarité habituellement définie comme étant la distance euclidienne entre les deux vecteurs :

$$i(x) = \arg \min \|x - w_j\| \quad j = 1, 2, \dots, N, \dots \dots \dots (2.1)$$

Où  $\| \cdot \|$  est la norme euclidienne telle que définie par :

$$\|x\| = \sqrt{\sum_{j=1}^p x_j^2} \dots \dots \dots (2.2)$$

Une fois trouvé le vecteur de poids gagnant, le réseau s'ajuste de manière à nouveau représenter ce vecteur d'apprentissage. Le résultat final peut être visualisé par un filet s'ajustant au nuage des données d'entrée de façon à ce que chaque vecteur poids approxime la densité de probabilité d'une région de l'ensemble d'apprentissage [13]. Pour un neurone  $j$  appartenant au voisinage topologique variable du neurone gagnant  $i(x)$  décrit une rotation des vecteurs vers le vecteur d'entrée et s'énonce comme suit :

$$w_i(n+1) = w_i(n) + \alpha(n)h_{bi}(x - w_i(n)) \dots \dots \dots (2.3)$$

Où  $\alpha(n)$  taux d'apprentissage

$h_{bi}(n)$  est la fonction de voisinage :

$$h_{bi}(n) = \exp\left(-\frac{\|r_b - r_i\|^2}{2\sigma^2(n)}\right) \dots\dots\dots(2.4)$$

Où  $\|r_b - r_i\|$  est la distance entre le neurone b et celui gagnant i(x)

Sachant que  $\alpha(n)$  et le taux d'adaptation  $\sigma(n)$  sont décroissants en fonction du temps.

### 3.2.4 Algorithme d'apprentissage

---

#### Algorithme

---

##### Etape1 . Initialisation :

Choix aléatoire des vecteurs poids initiaux  $w_j(0)$   $j=1 \dots\dots\dots N$  où N est le nombre des neurones dans la structure et est égal à la dimension des vecteurs d'apprentissage. Il est préférable d'avoir de petites valeurs pour les poids.

##### Etape2. Présentation :

Tirage aléatoire d'une entrée x dans la base d'apprentissage afin de la présenter au réseau.

##### Etape3. Similarité :

Recherche du neurone gagnant i(x) en utilisant le critère de la distance euclidienne minimale (neurone plus proche).

$$i(x) = \arg \min \|x - w_j\|$$

##### Etape4. Modification du poids :

Ajuster les poids des neurones appartenant au voisinage topologique utilisant :

$$w_i(n+1) = w_i(n) + \alpha(n)h_{bi}(x - w_i(n))$$

##### Etape5. Continuation :

Retour à l'étape 2 si le changement des poids demeure non négligeable.

---

### 3.2.5 Propriétés des SOM :

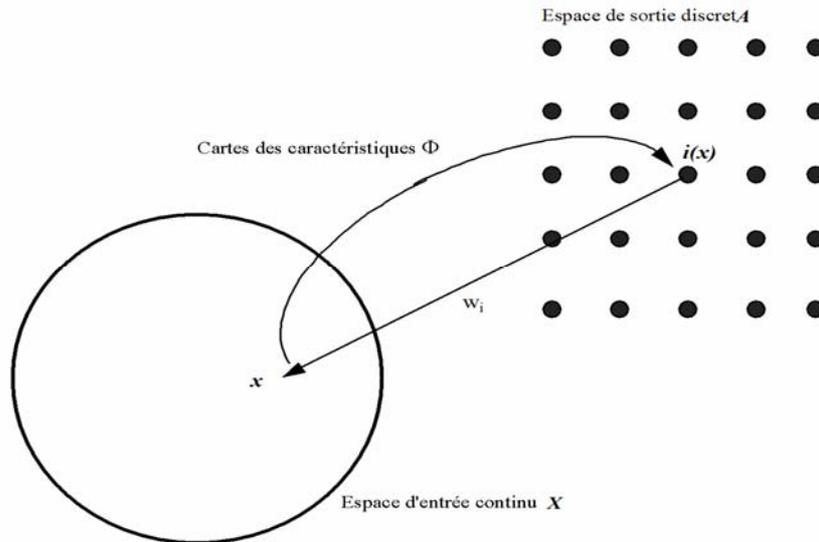
Une fois que l'algorithme a convergé, le SOM ou carte de kohonen démontre des caractéristiques statistiques importantes. Afin de démontrer ces propriétés, considérons un ensemble d'entrée continu X dont la topologie est décrite par une relation métrique des vecteurs  $x \in X$ . L'ensemble A est un ensemble discret de sorties dont la topologie a été

déterminée par l'apprentissage des vecteurs poids.

Dénotons aussi  $\Phi$  une transformation non\_linéaire que nous appellerons carte de caractéristique qui cartographie l'espace d'entrée  $X$  sur l'espace de sortie  $A$  tel que

$$\Phi = X \Rightarrow A \dots\dots\dots(2.5)$$

Le poids  $w_j$  associé au neurone gagnant  $i(x)$  est considéré comme un pointeur de ce neurone vers l'espace  $X$  comme il est présenté par la figure 2.7



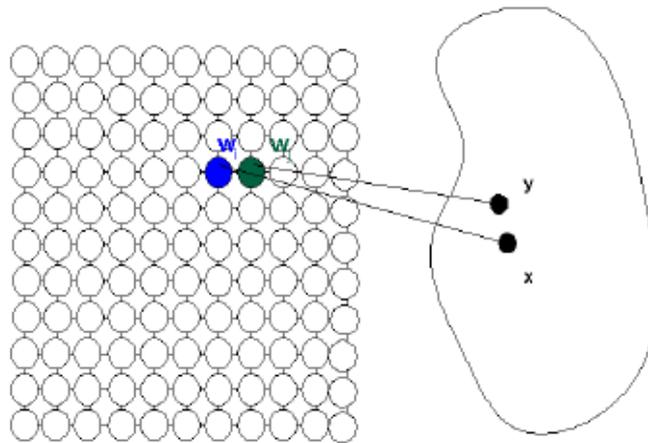
**Figure 2.7:** Relation entre la carte des caractéristique  $\Phi$  et le vecteur poids  $w_j$  du neurone gagnant  $i(x)$ .

**a. Approximation de l'espace d'entrée :** La carte des caractéristique  $\Phi$  représenté par les vecteur poids  $w_j \ j=1\dots\dots N$  constitue une bonne approximation de l'espace de départ  $X$ . Le but principal d'une telle transformation est de faire l'approximation d'un grand ensemble de vecteurs  $x \in X$  par une série plus petite de prototypes  $w_j \in A$ .

L'ensemble des vecteurs dont nous connaissons à priori la distribution permet de mieux visualiser la capacité de représentation lors de l'application des cartes sur des distributions plus complexes. Si on a une distribution dans  $R^2$  il est approprié de représenter ces points sur une carte 2D de Kohonen. Cette carte représente ultimement l'approximation de l'ensemble  $X$  par un autre plus petit  $A$  qui contient les neurones de coordonnées  $w_j$ . Les résultats de l'apprentissage de la carte de Kohonen est un ensemble représentatif de l'ensemble d'apprentissage en l'occurrence.

**b. Appariement de la densité d'une distribution** : La densité de la répartition des neurones dans l'espace de sortie reflète les variations statistiques de l'espace d'entrée : les régions de l'espace d'entrée desquelles les vecteurs d'entrée sont tirés avec une grande probabilité sont cartographiées sur de plus grandes régions dans l'espace de sortie. Ceci permet en général d'avoir une meilleure résolution que pour les régions où la probabilité de tirer un vecteur est plus faible. De façon générale, l'algorithme SOM tend à sur-représenter les régions de faibles probabilités et à sous-représenter les régions de grandes probabilités.

**c. Conservation de l'ordre topologique** : La propriété de l'ordre topologique est une conséquence directe de l'équation de mise à jour des vecteurs poids qui force entre autre, le neurone gagnant  $i(x)$  et tous les neurones appartenant à son voisinage topologique de se déplacer vers le vecteur d'entrée  $x$ . La proximité topologique des neurones est conservée après convergence, c'est-à-dire que les vecteurs semblables de l'ensemble de départ  $X$  sont cartographiés sur des neurones se trouvant près les uns des autres. Deux neurones voisins dans  $A$  (distance discrète) doivent représenter deux ensembles d'observations voisines dans l'espace de donnée.

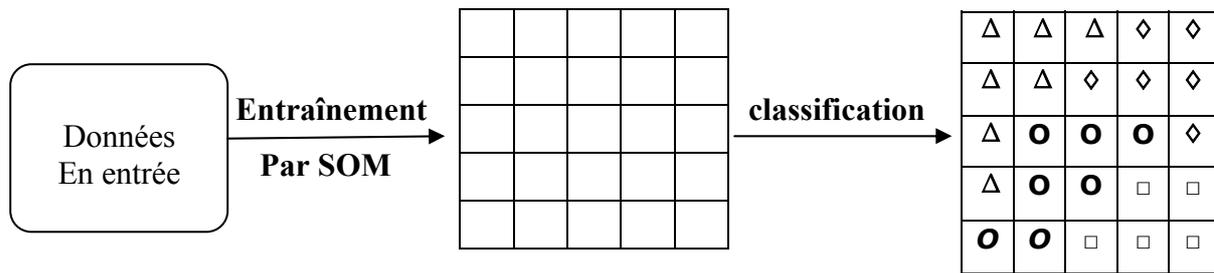


**Figure 2.8:** Conservation de la topologie des données

### 3.2.6 Classification de SOM

Parmi les différentes applications basées sur les cartes organisatrices [14,15], un assez grand nombre sont des tâches de classification. Ces cartes ne permettent pas de résoudre directement ce type de problèmes. Le résultat de l'apprentissage non supervisé permet d'affecter une observation à un sous-ensemble d'une partition indépendamment de toute notion de classe. Pour cette raison une méthode à deux niveaux a été étudiée et proposée dans

ce travail [13,16] (cf figure 2.9) où les différents symboles présentent les différentes classes.



**Figure 2.9** : SOM à deux niveaux

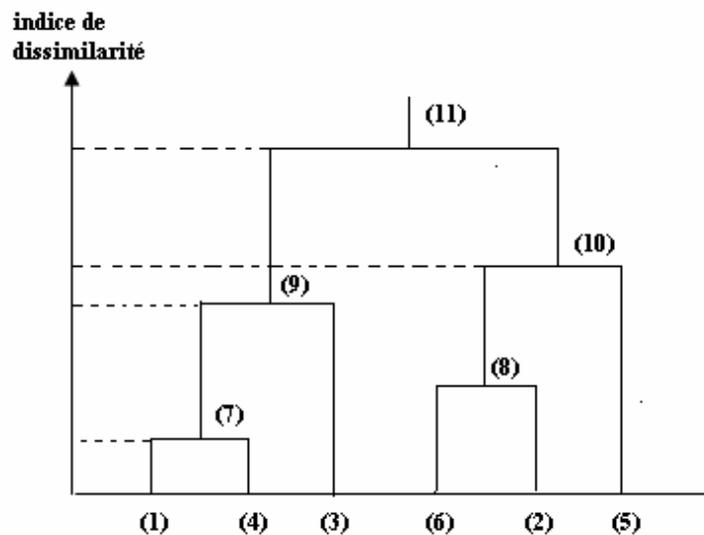
### Description de la méthode

Cette méthode est basée sur les cartes auto-organisatrices de Kohonen. L'idée est de partager la classification en deux niveaux.

Le premier niveau consiste à entraîner les données en utilisant l'algorithme SOM. Initialement, un certain nombre de vecteurs sont associés à chaque noeud au hasard. Pendant le processus d'apprentissage les vecteurs s'ajustent progressivement pour couvrir l'espace de façon à réduire la complexité. Une fois l'algorithme a convergé on obtient un certain nombre de groupes d'objets ordonnancés. L'ensemble des neurones de la carte représente donc une quantification vectorielle de l'ensemble X obtenue par l'analyse de l'ensemble d'apprentissage.

Le deuxième niveau consiste à classifier les groupes d'objets obtenus dans l'étape précédente. On trouve plusieurs approches pour la classification des différents sous ensembles d'observation. Une méthode progressive de la carte de Kohonen a été proposée [13] à travers le K\_means. Pour déterminer le nombre exact des classes, l'utilisation d'un critère qui valide le nombre de classes en l'occurrence l'indice proposé par Davies\_Bouldin dans [17] a résolu ce problème. On attribue un indice de validité pour mesurer la qualité de la classification. La meilleure classification possédera le plus petit indice de validité.

La classification ascendante hiérarchique qui est une méthode de classification automatique permet aussi d'effectuer la seconde phase nécessaire à l'élaboration du classifieur en regroupement « au mieux » (figure 2.10) les neurones où les feuilles de l'arborescence représentent les neurones (ici six neurones). L'axe des ordonnées donne pour chaque regroupement ou palier agrégatif, l'indice d'agrégation pour similarité choisie.



**Figure 2.10** : Regroupement des neurones de la carte par classification ascendante hiérarchique.

Il s'agit d'une méthode qui calcule une hiérarchie de partitions, chaque partition permettant de regrouper d'une 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 composée de l'ensemble des singletons (les neurones). La classification hiérarchique utilise cette partition initiale et procède à des regroupement 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, les deux sous-ensembles de neurones les plus semblables au sens de la mesure choisie.

On peut résumer ces étapes dans l'algorithme suivant :

### Algorithme

#### 1.Phase d'initialisation :

- chaque poids  $w$  est placé dans sa propre classe.  
Si le nombre des poids égale a  $M$  alors  
Le nombre des classes est égale a  $M$ .
- Calcul la distance entre tous les poids / « le chaînage complet a été utilisé » /

#### 2- Répéter

- Sélection dans  $M$  des deux classes les plus proche  $C_i$  et  $C_j$ .
- fusion de  $C_i$  et  $C_j$  dans une seule classe  $C_g$

- Mise à jour de M en calculant la ressemblance entre  $C_g$  jusqu'à la fusion des deux dernières classes.

La qualité de l'auto-organisation n'est pas toujours de bonne qualité. La prise en considération des propriétés des algorithmes des cartes auto-organisatrices impose que l'on émette l'hypothèse suivante :

La quantification des données est de bonne qualité : chaque vecteur représente bien l'ensemble des observations qui lui sont affectées. Mais si la qualité de cette carte n'est pas bonne, on obtient des vecteurs poids ne représentant aucune observation. Pour filtrer ces neurones Sitao et al [16] ont proposé un pré-traitement avant la classification hiérarchique qu'on peut résumer dans l'algorithme suivant.

### Pré-traitement avant la classification

Avant d'écrire l'algorithme il faut déterminer les notations suivantes :

$x$  : vecteur d'entrée.

$n_j$  : nombre d'éléments associés à la classe  $j$

$c$  : nombre des classes

### Algorithme :

1. Calcul de la distance euclidienne entre une entrée  $x_i$  et tous les poids  $w_j$  et association de chaque point au poids le plus proche.
2. Calcul des valeurs moyennes  $m_j$  de chaque classe  $j$

$$m_j = \frac{1}{n_j} \sum_{x_j \in c_j} x_j \dots \dots \dots (2.6)$$

3. Calcul de la déviation distance  $dev_j$  pour chaque neurone

$$dev_j = \|w_j - m_j\| \dots \dots \dots (2.7)$$

4. Calcul de la déviation moyenne  $mean\_dev_j$  pour toutes les classes

$$mean\_dev_j = \frac{1}{n_j} (|x_{1j} - m_j| + |x_{2j} - m_j| + \dots + |x_{n_j} - m_j|) \dots \dots \dots (2.8)$$

Calcul de la moyenne de toutes les déviations moyennes des classes :

$$mean\_dev = \frac{1}{c} \sum_{k=1}^c mean\_dev_k \dots \dots \dots (2.9)$$

5. Calcul de la déviation standard  $std\_dev_j$  pour chaque classe  $j$

$$std\_dev_j = \sqrt{\sum_{k=1}^{n_j} (x_k - m_j)^2 / (n_j - 1)} \dots\dots\dots(2.10)$$

Calcul de la déviation standard moyenne

$$std\_dev = \sqrt{\sum_{k=1}^c \|std\_dev_k\|^2 / c} \dots\dots\dots(2.11)$$

**Si**  $dev_j > mean\_dev + std\_dev$  **alors**

- Eliminer le neurone  $j$  :  $w(j)=0$
- Remplacer l'entrée  $x$  associée au poids  $w_j$  par l'entrée voisine.

**Fin si**

6. Calcul de la distance  $\|x_i - w_j\|$

**Si**  $\|x_i - w_j\| > mean\_dev_j + std\_dev_j$  **alors**

Remplacer  $x_i$  par l'entrée voisine

**Fin si**

7. Calcul de la moyenne  $mean\_num$  :

$$mean\_num = \frac{1}{c} \sum_{k=1}^c \|n_k\| \dots\dots\dots(2.12)$$

Où  $\|n_k\| = card(c_k)$

8. Calcul de la déviation standard  $std\_num$  :

$$std\_num = \sqrt{\sum_{k=1}^c \|n_k\|^2 / c} \dots\dots\dots(2.13)$$

**Si**  $n_j < mean\_num - std\_num$  **alors**

- Eliminer le neurone  $j$  :  $w(j)=0$
- Remplacer l'entrée  $x$  associée de poids  $w_j$  par l'entrée voisine.

**Fin si**

### Le principe de fusionnement

Cette étape est basée sur le principe de validité des classes que nous développons au chapitre suivant. Elle utilise un indice composé de densités intra et inter classes CD<sub>bw</sub> acronyme de "Composing Density Between and With Clusters". Initialement l'indice global est calculé par toutes les classes puis par chaque couple de classe (i,j). Les exemples de classes qui correspondent à la valeur minimale de cet indice entre classes sont pris si les deux classes de couple sont voisines c-à-d un pixel de la classe i est connexe à un pixel de la classe j au sens du 8\_voisinage (voir figure 2.11). Dans ces conditions les deux classes i et j sont fusionnées en une seule classe. L'indice global est recalculé pour toutes les classes. Si la valeur de cet indice globale est supérieure à celle calculée initialement, l'indice pour chaque couple de classe est recalculé et le fusionnement est effectué. Si la valeur de l'indice globale est inférieure à sa valeur dans une étape précédente le fusionnement est arrêté et le nombre de classes est déterminé.

Pour bien identifier le nombre des classes nous proposons d'ajouter l'indice de Davies\_Bouldin. Nous le calculons après chaque opération de fusionnement et le nombre final des classes possédera le plus petit indice de validité.

	B	C	D	
	E	A	F	
	G	H	I	

**Figure 2-11** : Le neurone A a huit directions de voisinage.

On peut résumer les étapes de cette méthode comme suivant :

---

#### Algorithme général

---

**Etape 1** : Entraîner les données par l'algorithme de SOM

**Etape 2** : Pré traitement pour éliminer les neurones qui ne représentent aucune observation.

**Etape 3** : Classification des neurones les neurones de la carte obtenue dans l'étape 1 par une méthode de Classification Ascendante Hiérarchique.

---

---

**Etape 4 :** Fusionnement à l'aide d'un indice de validation qui est basée sur la densité inter et intra classe .

**Etape 5 :** Définir le nombre exact des classes.

---

#### 4. La Théorie de la résonance adaptative ART:

Le principe général du modèle ART conçu par Carpenter et Grossberg [18] est de créer un système d'apprentissage autonome capable de concilier adaptation et stabilité (compromis à effectuer entre plasticité et rigidité synaptique). Le problème d'un système trop plastique est qu'il peut apprendre des informations même si elles ne sont pas pertinentes ou oublier des informations apprises. A l'inverse, un système trop rigide empêche tout apprentissage. Les réseaux ART sont des réseaux à compétition. Les valeurs de poids ne sont adaptées que s'il y a une approximation suffisante entre l'entrée du réseau et le prototype déjà connu par le réseau. Alors il existe une résonance. Sinon, l'entrée du système est très éloignée du prototype. Alors une nouvelle classe est créée. ART présente de plus l'avantage de ne pas nécessiter un nombre préalablement fixé des classes, celles-ci étant créées dynamiquement en fonction de la structure de l'ensemble des données présentées. Le nombre total de classes obtenu dépendra de la sélectivité de ces classes. En effet, le degré de nouveauté d'une donnée est mesuré par rapport à un paramètre de seuil fixé par l'opérateur. C'est ce paramètre, appelé vigilance par Carpenter et Grossberg, qui contrôle la sélectivité des classes. Le mode d'apprentissage peut être supervisé ou non. Les architectures ART se manifestent par différentes implantations spécifiques dont celles nommées ART1, ART2, ART3, fuzzy ART, ARTmap, fuzzy ARTmap, où ART-1 est le premier modèle de la série ART, il effectue une classification non supervisée de données binaires. Les autres membres de cette famille ne sont pas si limités. Le but le plus connu de ce réseau est de modéliser la reconnaissance d'un objet perçu, ainsi que de stocker une information. Mais ces réseaux sont très sensibles aux bruits qui apparaissent sur les données. Pour y échapper, on doit mettre en place une couche de filtrage. On peut résumer le fonctionnement de ce modèle dans l'algorithme suivant :

---

#### Algorithme

---

**Etape 1. Normalisation des entrées :**

$$I = \frac{a}{|a|} \text{ ou } I = (a, a^c) \text{ avec } a_i^c = 1 - a_i \text{ (compliment à 2)}$$

**Etape 2. Sélection de classe:**

---

a. Pour toute entrée  $I$  et nœud  $j$ , la fonction d'activation ou de choix s'écrit :

$$T_k(I) = \frac{|I \cap w_j|}{\alpha + |w_j|} \dots\dots\dots(2.14)$$

Où  $w$  vecteur poids associé au neurone  $j$

$\cap$  le Et booléen ( l'intersection ).

b. La classe choisie est celle correspondant au nœud  $j$  tel que  $T_j = \max(T_k)$ .

### étape 3. Résonance ou teste :

Le critère de comparaison entre l'exemple présenté et le prototype de la classe sélectionnée est :

$$M = \frac{|I \cap w_j|}{|I|} \dots\dots\dots(2.15)$$

le paramètre de vigilance est  $\rho$  ;  $\rho \in [0,1]$

**Si**  $M \succ \rho$  **alors**

aller à l'étape 4 (modifier les poids)

**Si non** le neurone gagnant inhibé

S'il y a encore des neurones non inhibés sur la couche cachée alors retour à l'étape 2.b

### Etape 4. Apprentissage :

Les vecteurs de poids  $w_j$  sont remis à jour selon l'équation :

$$w_j^{new} = \beta(I \cap w_j^{old}) + (1 - \beta)w_j^{old} \dots\dots\dots(2.16)$$

**Fin si**

Dans ce travail, nous allons nous concentrer sur une architecture, non supervisée, à savoir fuzzy ART qui présente de plus l'avantage de ne pas nécessiter un nombre préalablement fixé de classe, celle-ci étant créée dynamiquement en fonction de la structure de l'ensemble des données présentées.

**Fuzzy ART** : est un algorithme de classification qui utilise un apprentissage non supervisé. C'est un réseau compétitif à deux couches de neurones[19]. La première couche, notée  $E_c$ , sert à coder les stimuli d'entrée avec un encodage dit «complémentaire». La deuxième couche est une couche compétitive semblable à celle de Kohonen. Cependant, tout comme les autres architectures ART, le fuzzy ART incorpore un mécanisme de rétroaction permettant de

stabiliser les prototypes appris dans les vecteurs de poids qui relient les deux couches. Ce mécanisme dit de «résonance» est contrôlé par un paramètre qui permet de réinitialiser au besoin la couche compétitive. Pour chaque stimulus d'entrée, les sorties  $a^2$  du réseau spécifient une catégorie parmi S.



Figure 2.12 : Architecture du réseau fuzzy ART

Où  $E_c$  : La première couche de réseau.

$a$  : Les sortie du réseau

S : L'ensemble des neurones

Le module fuzzy ART est obtenu en remplaçant l'opérateur d'intersection de type booléen du module ART1 par un opérateur ET de type flou. Le Tableau 2.2 présente les différences au niveau algorithmique entre l'algorithme ART1 et fuzzy ART.

ART1	fuzzy ART
<b>Fonction de choix</b>	
$T_j = \frac{ I \cap w_j }{\alpha +  w_j }$	$T_j = \frac{ I \wedge w_j }{\alpha +  w_j }$
<b>Fonction de vigilance</b>	
$\frac{ I \cap w }{ I } \geq \rho$	$\frac{ I \wedge w }{ I } \geq \rho$
<b>Fonction d'apprentissage</b>	
$w_j^{new} = I \cap w_j^{old}$	$w_j^{new} = I \wedge w_j^{old}$
$\cap =$ ET booléen (intersection)	$\wedge =$ ET flou (minimum)

Tableau 2.2 : Algorithme ART1 et Fuzzy ART

#### 4.1 L'approche Fuzzy ART:

Le fuzzy ART est un réseau constructif où de nouveaux neurones sont alloués au fil de l'apprentissage. Généralement, on fixe au départ un nombre maximum de neurones  $S$ , ce qui fixe également un nombre maximum de catégories de stimuli. Initialement, aucun neurone n'est actif, le premier stimulus d'apprentissage activera le neurone associé à la première catégorie. L'allocation subséquente de nouvelles catégories dépendra à la fois des stimuli et des paramètres de l'algorithme.

Les entrées d'un fuzzy ART doivent être des ensembles flous définis sur un référentiel discret. Soit des entrées  $x$  sont des vecteurs de dimension  $n$ . Ces vecteurs sont tout d'abord normalisés grâce à un codage complémentaire avant d'être présentés à la couche d'entrée :

$$I = (x, x^c) \quad \forall i \in \{1, \dots, n\} \quad \dots \dots \dots (2.17)$$

$$x_i^c = (1 - x_i)$$

La couche de sortie est constituée d'un ensemble de neurones  $y_j$  associés à un vecteur poids  $w_j$ . Ces neurones correspondent aux différentes classes créées. L'opérateur **ET** de la logique floue ( $(a \wedge b) = \min(a; b)$ ).

On calcule la fonction d'activation  $T_j$

$$T_k(I) = \frac{I \wedge w_k}{\alpha + |w_j|}$$

$$T_j = \max(T_k) \quad j = 1 : n \quad \dots \dots \dots (2.18)$$

$$activation(y_k) = \begin{cases} 1 & \text{si } k = j \\ 0 & \text{sinon} \end{cases} \quad \dots \dots \dots (2.19)$$

Si plusieurs neurones ont un degré  $T_k$  égal, le neurone dont l'indice est le plus faible est le neurone gagnant.

L'apprentissage du réseau est réglé par la fonction  $M$  de la couche cachée.

$$M = \frac{|I \wedge w_j|}{|I|} \quad \dots \dots \dots (2.20)$$

Ce terme  $M$  indique la similarité de l'entrée  $x$  avec la classe  $y_j$  activée. Plus  $M$  est proche de 1 et plus  $x$  est semblable au représentant  $w_j$  de la classe et plus  $x$  est adapté à la classe. L'apprentissage s'effectue en ajustant le poids  $w_j$  dans le sens de la nouvelle entrée selon la relation :

$$w_j^{new} = \beta \min(I, w_j^{old}) + (1 - \beta)w_j^{old} \quad \dots \dots \dots (2.21)$$

Si la fonction  $M$  est inférieure à un seuil fixé  $\rho$  appelée vigilance, on considère qu'aucune classe n'est adaptée à la nouvelle entrée. Une nouvelle classe est créée centrée sur  $x$ . Ce paramètre permet de régler le degré de généralité des classes et donc la finesse de description de l'espace de points.

Le réseau fuzzy ART définit trois paramètres  $\alpha$ ,  $\eta$  et  $\rho$

$$\alpha > 0, \quad 0 < \eta \leq 1, \quad 0 < \rho < 1$$

Où  $\alpha$  représente un paramètre de sélection,  $\eta$  le taux d'apprentissage et  $\rho$  le taux de vigilance.

#### 4.2. Algorithme : fuzzy ART [20]

---

##### Début

---

**Etape 1.** initialisation des poids  $w=[w_1 \dots \dots w_n]$ ,

Fixer  $\alpha$ ,  $\eta$  et  $\rho$

$t=1$

**Etape 2.** pour  $t \leq t_{\max}$  faire

1. Choisir aléatoirement un ensemble flou  $x(t)$  parmi la base d'apprentissage
2. Effectuer l'encodage complémentaire.

$$I(t)=[x(t), x^c(t)] \text{ où } x^c(t)=1-x(t)$$

3. Calcule la fonction d'activation

$$A(I^t, W_j^t) = \frac{|\min(I^t, W_j^t)|}{\alpha + |W_j^t|} = \frac{\sum_{i=1}^n \min(I_i^t, W_{ji}^t)}{\alpha + \sum_{i=1}^n W_{ji}^t} \dots \dots \dots (2.22)$$

- a. Définir le neurone gagnant  $j$  ayant la plus grand valeur de la fonction (2.22)
- b. Activer la sortie  $y_j$ .

4. Evaluation de la compatibilité de l'entrée  $x$  avec la classe  $y_j$  activée utilisant (2.23)

$$M(W_j^t, I^t) = \frac{|\min(I^t, W_j^t)|}{|I^t|} = \frac{\sum_{i=1}^n \min(I_i^t, W_{ji}^t)}{\sum_{i=1}^n I_i^t} \dots \dots \dots (2.23)$$

**Si  $M \geq \rho$  alors**

- adapter les prototypes suivant la règle (2.24)
  - sauter à une nouvelle entrée.
-

$$R(I^t, W_j^t) = \beta \min(I^t, W_j^t) + (1 - \beta)W_j^t \dots\dots\dots (2.24)$$

**Si non**

- bloquer temporairement le neurone gagnant j pour qu'il ne puisse gagner et retourner à l'étape 3.a.

**Fin si**

t=t+1

---

**Fin**

---

### 4.3 Les problèmes de fuzzy ART :

- a. Le nombre des classes obtenu reste inférieur ou égale au nombre des entrées. Alors s'il existe des régions non représentées par des entrées, elles ne sont pas classées et elles seront introduites dans les autres classes, ce qui conduit à une erreur de classification.
- b. L'ordre des entrées influe directement sur le résultat final par exemple si nous n'avons pas une grande différence entre l'entrée numéro un et numéro deux elles seront considérées comme une seule classe.
- c. Le problème majeur de l'algorithme fuzzy ART est le problème de la prolifération des catégories. Ce problème apparaît quand le réseau va créer des neurones inutilisables ou des classes plus que le nombre exacte ou désirée. Ce problème survient quand les entrées sont mal arrangées. Le réseau va sélectionner par exemple l'entrée numéro deux et va activer la catégories deux alors que le choix de l'entrée une est réduit le nombre de catégories. Aussi ce problème est du au choix de la fonction.

### 4.4 Solutions proposés

Pour résoudre les problème - a- et - c - aussi nous avons proposé d'utiliser un vecteur d'entrée contenant les valeurs comprise entre 0 et 1 avec un écart type très faible. Donc il est assuré que toutes les données seront représentées. Ainsi le problème de la manque de données est éliminée. Comme il est montré dans la section ci\_ dessus l'ordre des entrées provoque le problème de prolifération ce qui influence les résultats finaux. Alors l'ordre des vecteurs d'entrées sont changés de façons à ce que la distance entre chaque deux entrées successive soit maximale.

Pour résoudre le problème - c- le coefficient de similarité de Jaccard [21] remplace la fonction du choix de l'algorithme Fuzzy ART. il est données par

$$T_j = \frac{|I \wedge w_j|}{|w_j| \vee |I|} \dots\dots\dots(2.25)$$

Où  $\vee$  présente le minimum entre le poids  $w$  et l'entrée  $I$ . Nous changeons seulement la fonction de choix dans notre modification au contraire de modification de référence [21] qui a changé aussi les règles d'apprentissages.

Ce coefficient permet d'identifier un nombre minimal de classes. comme l'explique l'exemple suivant :

Le vecteur d'entrée  $I = [ 0 \quad 0 \quad 1 \quad 0.5 \quad 0.4 \quad 0.5 ]$

Le vecteur du poids est donné comme suivant :

$$W1 = [ 0.01 \quad 0.05 \quad 1 \quad 0.5 \quad 0.01 \quad 0.5 ]$$

$$W2 = [ 0.01 \quad 0.01 \quad 0.01 \quad 0.50 \quad 0.4 \quad 0.01 ]$$

Les différents valeurs obtenues par la fonction de choix de l'algorithme Fuzzy ART et le coefficient de similarité de Jaccard sont représentés ci-dessous avec le paramètre de choix  $\alpha = 0.01$  et le paramètre de vigilance  $\rho = 0.3$ .

1- fonction de choix

$$T_j = \frac{|I \wedge w_j|}{\alpha + |w_j|} \quad T_1 = 0.966 \quad , \quad T_2 = 0.968$$

2- le coefficient de similarité de Jaccard

$$T_j = \frac{|I \wedge w_j|}{|w_j| + |I|} \quad , \quad T_1 = 0.838 \quad T_2 = 0.383.$$

D'après la fonction du choix le deuxième entrée est sélectionnée bien que la première entrée est similaire au premier vecteur du poids que la deuxième entrée. Mais d'après le coefficient de similarité de Jaccard sélectionne la première entrée.

## 5. Conclusion

Les réseaux de neurones sont donc une alternative qui peut être très efficace pour les problèmes que les algorithmes classiques ne peuvent résoudre. Grâce à leur capacité de **classification** et de **généralisation**, les réseaux de neurones sont généralement utilisés dans des problèmes de nature statistique, tels que la classification automatique et ils ont montré leur capacité de réalisé ce genre de tâche.

# Chapitre 3:

## Indices de validité des classes

## 1. Introduction

L'objectif d'une tâche de classification non supervisée consiste à proposer une partition des objets en  $k$  sous ensembles où le paramètre  $k$  est le nombre de regroupements attendus par l'utilisateur. Une variation de cette tâche est de ne pas utiliser le nombre attendu de regroupements comme une donnée du problème. Dans ce cas, l'algorithme construit plusieurs partitions candidates et choisit la meilleure. La meilleure partition est celle qui optimise un critère de qualité des partitions.

## 2. Définition

Les résultats de la classification obtenus dépendent fortement du nombre de classes fixé. Il est donc primordial de choisir le nombre exact de classes pour espérer avoir une bonne qualité de classification. Ceci n'est pas toujours simple, surtout en présence de chevauchement. Plusieurs approches ont été proposées sur ce sujet pour différentes applications [22.] Cependant, pour les mêmes données, on peut obtenir des résultats différents selon le nombre de classes  $k$  fixé par l'utilisateur. Pour des classes bien séparées, les algorithmes de classification retrouvent généralement le même nombre de clusters. Le problème se pose dans le cas de chevauchement de classes : rares sont les algorithmes qui arrivent à détecter le nombre réel de classes, et ils deviennent invalides pour un degré de chevauchement relativement fort. Le processus d'évaluation des résultats des algorithmes de classification est appelé indice de validité des clusters.

## 3. Critères de validité.

Dans le contexte de la classification automatique, il est naturel de s'interroger sur la validité de la partition obtenue. Les groupes découverts correspondent-ils à nos connaissances à priori ? Correspondent-ils vraiment à l'ensemble d'objets dont on dispose ? De deux classifications, laquelle est la plus pertinente ? Ces différentes questions permettent de distinguer trois catégories de critères : les critères externes, les critères internes et les critères relatifs.

- **Les critères externes** permettent de répondre à la première question et de mesurer l'adéquation entre une partition et les connaissances à priori dont on dispose
- **Les critères internes** quantifient l'adéquation entre une partition et l'idée subjective que l'on se fait d'une "bonne" classification. Ainsi, les propriétés les plus communément recherchées sont la compacité et la séparabilité des groupes découverts.
- **Les critères relatifs** s'intéressent à la troisième et dernière question et à défaut de

- donner une appréciation absolue de la validité d'une partition, ils permettent d'ordonner plusieurs classifications et d'en choisir "une meilleure".

Les deux premiers critères sont basés sur des méthodes statistiques et demandent beaucoup de temps de calcul [23]. Beaucoup des techniques sont basées sur le critère relatif [22,24]. On trouve des indices basés sur le nombre des classes et où la donnée elle-même est basée sur le principe du maximum d'entropie. L'indice proposé par l'article [25] nommé VMEP est déduit d'une combinaison originale entre des méthodes d'analyse des données et le critère du maximum d'entropie. Ses performances sont montrées à travers un ensemble d'exemples simulés et réels. La procédure est complètement automatique dans le sens qu'elle ne nécessite aucun paramètre de réglage. Dans ce travail nous nous intéressons qu'aux indices les plus utilisés dans la littérature.

### 3.1 Critère de Davies-Bouldin

L'indice de Davies-Bouldin [17] tient compte à la fois de la compacité et de la séparabilité des groupes. La valeur de cet indice est d'autant plus faible que les groupes sont compacts et bien séparés. Il est particulièrement bien adapté pour une utilisation avec la méthode des K-moyennes.

L'expression de cet indice est la suivante:

$$DB = \sum_{k=1}^c \max \left\{ \frac{s_c(Q_k) + s_c(Q_l)}{d_{ce}(Q_k, Q_l)} \right\} \dots\dots\dots(3.1)$$

Où  $s_c(c)$  est la distance moyenne entre un objet du groupe  $c_k$  et son centre.

$$s_c = \frac{\sum_i \|x_i - c_k\|}{N_k} \dots\dots\dots(3.2)$$

$N_k$ : le nombre des éléments dans la classe  $k$

$d_{ce}$ : la distance qui sépare les centres des groupes  $c_k$  et  $c_l$

$$d_{ce} = \|c_k - c_l\| \dots\dots\dots(3.3)$$

si  $s_{c1} + s_{c2} < d_{ce}(c_1, c_2)$  donc le couple  $(c_1, c_2)$  est bien séparé le cas (a) figure(3.1-a-)

si  $s_{c1} + s_{c2} > d_{ce}(c_1, c_2)$  il faut fusionner le couple  $(c_1, c_2)$  le cas (b) figure (3.1-b-).

La figure 3.1 illustre les deux cas .

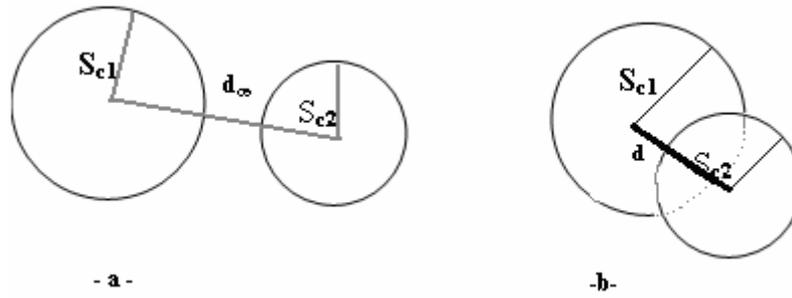


Figure 3.1 : Les deux représentations possibles des classes.

**3.2 Critère de Dunn :**

L'indice de Dunn tient compte à la fois de la compacité et de la séparabilité des groupes : la valeur de cet indice est plus élevée que les groupes sont compacts et bien séparés.

$C = \{c_1, c_2, \dots, c_n\}$  où n est le nombre des classes. L'indice de Dunn prend la forme suivante [26] :

$$I(c) = \frac{\min_{i \neq j} \{\delta(c_i, c_j)\}}{\max_{1 \leq l \leq k} \{\Delta(c_l)\}} \dots \dots \dots (3.4)$$

Où  $\delta(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(x, y) \dots \dots \dots (3.5)$

$$\Delta(c_i) = \max(\max_{x, y \in c_i} d(x, y)) \dots \dots \dots (3.6)$$

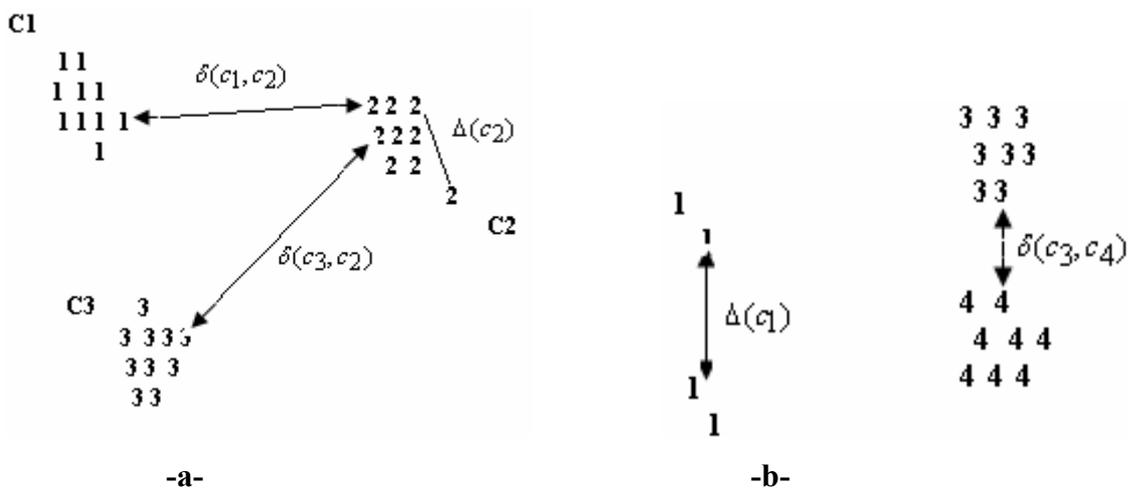


Figure 3.2 : -a- et -b- montrent deux exemples de classification

D'après l'expression (3.4) on peut remarquer que si la distance inter\_classes  $\delta$  est plus large que la distance intra\_classe  $\Delta$  donc la valeur de cet indice va augmenter et pour que nous ayons une meilleure classification il faut que les classes soient bien séparées et les objets d'une même classe sont bien compacts.

Les deux exemples suivants montrent initialement l'application de ce critère. La figure (3.2 a) présente une meilleure classification où les classes sont bien éloignées l'une des autres et elles sont compactes ce qui donne une valeur élevée de cet indice. Au contraire dans la figure (3.2 b) où la distance  $\Delta$  est plus large que cette distance  $\delta$  ce qui donne une faible valeur de cet indice, on conclut que la valeur de cet indice peut prendre une faible valeur si les données sont bruitées. Si la valeur de l'indice de Dunn est élevée on peut dire qu'on a une meilleure classification.

### 3.3 Critère du C\_index

Cet index est défini comme suit: [27]

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}} \dots\dots\dots(3.7)$$

Où  $S$  est la somme des distances, sur toutes les paires d'objets de la même classe. Soit  $L$  le nombre de ces couples,  $S_{\min}$  est la somme des  $L$  distances les plus petites,  $S_{\max}$  est la somme des  $L$  distances les plus larges de tous les couples. Une faible valeur de  $C$  indique un bon regroupement.

### 3.4 La Méthode de validité Silhouette

La technique de validation Silhouette [28] calcule la largeur de la silhouette de chaque échantillon, la largeur moyenne de la silhouette de chaque classe et de la largeur moyenne de la silhouette pour toutes les données. En utilisant cette approche, chaque groupe peut être représentée par ce qu'on appelle la silhouette, qui est basée sur la comparaison de sa compacité et de sa séparation. La largeur moyenne de la silhouette peut être appliquée à l'évaluation de la validité et le clustering pourrait également être utilisé pour décider combien le nombre des classes sélectionnées est bon.

Pour construire les silhouettes  $S(i)$ , la formule suivante est utilisée:

$$S(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}} \dots\dots\dots(3.8)$$

Où  $a(i)$  la différence moyenne entre le  $i$ -objet et tous les autres objets dans la même classe et  $b(i)$  le minimum de la différence moyenne du  $i$ -objet et tous les objets de la classe la plus proche.

La valeur de  $S(i)$  est limitée entre 1 et -1 ( $-1 \leq S(i) \leq 1$ ). Si la valeur  $S$  est proche de 1, cela signifie que l'échantillon est «bien groupé» et il a été attribué à un groupe tout à fait approprié. Si la silhouette est égale à zéro, cela signifie que l'échantillon peut être affecté à un autre groupe plus proche aussi, et l'échantillon se trouve également bien loin de ces deux groupes. Si la valeur de la silhouette est proche de -1, cela signifie que l'échantillon est "mal classé" et ne fait que quelque part entre les deux groupes. La moyenne globale de la largeur de la silhouette est simplement la moyenne de la  $S(i)$  pour tous les objets de l'ensemble de données. La plus grande valeur de la silhouette indique le meilleur regroupement. Par conséquent, le nombre des classes ayant largeur moyenne maximale de la silhouette est considéré comme le nombre optimal de classe.

### 3.5 Critère de compacité Wemmert et Gançarski

L'indice Wemmert et Gançarski considèrent à la fois la compacité et la séparabilité des groupes et s'appuient sur le rapport entre deux distances [7] : la distance d'un objet au centre de son groupe et la distance minimale au centre d'un autre groupe. Il se définit ainsi pour un groupe :

$$I_{WG}(C_i) = \max \left\{ 0 ; 1 - \frac{1}{N_i} \sum_{x \in C_i} \frac{\|x - w_i\|}{\min_{j \neq i} \{ \|x - w_j\| \}} \right\} \dots\dots\dots(3.9)$$

et la valeur de cet indice pour une partition correspond à la moyenne pondérée de l'indice de chacun des groupes :

$$I_{WG} = \frac{1}{N} \sum_{i=1}^K N_i x I_{WG}(C_k) \dots\dots\dots(3.10)$$

### 3.6 Critère basée sur la densité inter et intra classe ( CDbw )

Cet indice est basé comme tous les autres indices sur deux aspects. Le premier est la compacité des classes et le deuxième est la séparation des classes. La densité des classes a été ajoutée par cet indice. La densité inter et intra classe est calculée pour bien évaluer la compacité et la séparabilité des classes. La notation détaillée de cet indice est la suivante [16]:

La déviation standard est donnée :

$$stdev(i) = \sqrt{\sum_{k=1}^{n_i} (x_k - m_i)^2 / (n_i - 1)} \dots\dots\dots (3.11)$$

Où  $n_i$  le nombre des éléments associés à la classe  $i$

$m$  : la moyenne de classe

La moyenne de la déviation standard :

$$stdev = \sqrt{\sum_{i=1}^c \|stdev(i)\|^2 / c} \dots\dots\dots (3.12)$$

La densité intra classe est calculer par :

$$int\ ra\_den(c) = \frac{1}{c} \sum_{i=1}^c \sum_{j=1}^{r_i} density(w_{ij}), \quad c > 1. \dots\dots\dots (3.13)$$

$$density(w_{ij}) = \sum_{l=1}^{n_i} f(x_l, w_{ij})$$

Où  $x_i$  est un élément de la classe  $i$

$w$  : la représentation de la classe  $i$

$$f(x_i, w_{ij}) = \begin{cases} 1 & \text{si } \|x_l - w_{ij}\| \leq stdev \\ 0 & \text{autre cas} \end{cases} \dots\dots\dots (3.14)$$

La densité Inter\_classe est donnée :

$$inter\_dens(c) = \sum_{i=1}^c \sum_{\substack{j=1 \\ j \neq i}}^c \frac{\|clos\_rep(i) - clos\_rep(j)\|}{\|stdev(i)\| + \|stdev(j)\|} x density(u_{ij}), \quad c > 1; \dots\dots (3.15)$$

$$density(u_{ij}) = \sum_{k=1}^{n_i+n_j} f(x_k, u_{ij})$$

$$f(x_k, u_{ij}) = \begin{cases} 1 & \text{si } \|x_k - u_{ij}\| \leq (\|stdev(i)\| + \|stdev(j)\|) / 2 \\ 0 & \text{autre cas} \end{cases} \dots\dots (3.16)$$

Où **clos\_rep** : l'objet de la classe  $i$  le plus proche à les autres classes

Et  $u_{ij}$  est le milieu entre les deux points  $clos\_rep(i)$  et  $clos\_rep(j)$

La définition de la valeur de séparation des classes est :

$$sep(c) = \frac{\sum_{i=1}^c \sum_{\substack{j=1 \\ j \neq i}}^c \|close\_rep(i) - close\_rep(j)\|}{1 + inter\_den(c)} \quad c > 1 \dots\dots\dots(3.17)$$

Donc l'indice global est donné :

$$CDbw(c) = intra\_den(c) \times sep(c) \dots\dots\dots(3.18)$$

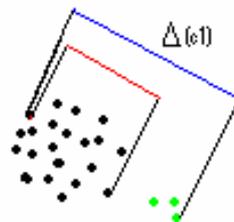
Nous utilisons dans ce mémoire les indices de Davies\_Bouldin, Dunn et CDbw.

**3.7 Critère proposé:**

A partir de l'application de ces indices on peut extraire les inconvénients suivants :

L'indice de Dunn devient prohibitif dès qu'on manipule de grands ensembles d'objets.

L'absence d'une notion de densité va provoquer des erreurs sur les évaluations de qualité de la classification. Comme le montre l'exemple suivant :



**Figure 3.3 :** Exemple d'une classe c<sub>i</sub>

Il faut remarquer que l'objet présenté en vert est plus éloigné des autres bien qu'il appartienne à la même classe. L'indice de Dunn est basé sur le calcul de la distance inter et intra classe comme le montre l'équation ( 3.4). La distance Δ calculée par l'expression (3.6) qui représente le diamètre de la classe (ligne bleu) alors que la vraie distance est représentée par la ligne rouge. Donc ce problème donne une fausse imagination sur la forme de la classe et ceci est du à la négligence de la densité.

L'indice qui est basé sur la densité inter et intra classe CDbw a pu corriger ce problème mais la complexité des calculs peuvent produire des erreurs.

Quelques autres indices ne peuvent pas sélectionner la meilleure classification qu'après plusieurs partitions. Les temps de calcul augmentent de façon considérable.

Pour ces raisons nous proposons un indice basé sur la densité des classes beaucoup moins

complexe que les précédents. La description de cet indice est la suivante :

Initialement la densité de toutes les classes ainsi une valeur de pourcentage pour évaluer la qualité de la classification sont calculées. Le nombre maximal des classes **nm<sub>x</sub>** est le résultat de la classification des données dans une première phase par l'algorithme des cartes auto\_organisatrices de Kohonen (SOM). Ce nombre permet de sélectionner un seuil SL qui présente la densité minimale d'une classe. Ce seuil va partager les classes en deux catégories : une catégorie constitue les classes valides et l'autre les classes refusées.

Les poids représentant les classes refusées sont éliminés ( $w=0$ ), La classification est refaite tous ces objets.

Cet indice offre plusieurs avantages :

1. Il permet d'évaluer la qualité de la classification facilement.
2. Il permet de corriger les partitions à la première passe. Il n'est pas nécessaire de recalculer les partitions, ce qui rend l'algorithme beaucoup plus rapide.
3. Les calculs sont faciles et simples.
4. Une étape de filtrage est introduite automatiquement dans l'algorithme de cet indice.

Nous pouvons résumer ces étapes dans l'algorithme suivant :

---

### Algorithme

---

#### 1. Etape d'initialisation

- Choisir un vecteur d'apprentissage pour entraîner l'algorithme SOM
- Faire la classification utilisant ce vecteur.
- $nm_x$  = le nombre maximale des classes obtenus
- Calculer la densité  $N(i)$  pour chaque cluster  $C_i$
- Calculer le minimum des cardinaux de nombre maximal des classes SL

$$SL = \min (\text{card} (N_i (nm_x))) \dots \dots \dots (3.19)$$

- Calculer la somme des densités :

$$\text{Som} = \sum_{i=1}^{nm_x} N_i \dots \dots \dots (3.20)$$


---

et

$$\text{Total} = \sum_{i=1}^n N_i \dots\dots\dots(3.21)$$

n : nombre des classes .

2. Etape d'évaluation : calculer le pourcentage pour évaluer la qualité de classification.

$$\text{Pourcentage} = (\text{Som} / \text{Total}) * 100 \dots\dots\dots(3.22)$$

3. Etape de validité

Pour i=1 :n faire

Si  $N(i) < SL$  alors

$W(i)=0$

Fin si

Fin pour

4. Etape de classification : Refaire la classification avec les nouveaux poids  $w_i$  .

#### 4. Conclusion

Les indices de validités sont des éléments très importants dans les problèmes de la classification non supervisée qui sont pu déterminer le nombre exacte des classes et bien sur d'améliorer la qualité de la classification.

## **Chapitre 4:**

# **Résultats expérimentaux**

## 1. Introduction

La segmentation est une étape indispensable dans de nombreuses chaînes de traitement de plusieurs domaines fondamentaux de la recherche clinique. La qualité de l'interprétation d'une image dépend fortement de celle de la segmentation qui est une étape de base du traitement d'une image. Parmi les images qu'on a utilisées dans ce mémoire les images médicales. L'information apportée par l'imagerie médicale est d'un apport considérable en matière de diagnostic.

Dans ce chapitre nous appliquons les méthodes étudiées dans les chapitres précédents. Cependant avant d'appliquer ces méthodes nous présentons une brève description des images médicales utilisées.

## 2. L'imagerie médicale

L'imagerie médicale macroscopique regroupe un ensemble de techniques reposant sur l'utilisation d'un phénomène physique et permettant de visualiser une partie du corps humain ou d'un organe et d'en conserver une image, dans l'objectif de réaliser un diagnostic, de guider un geste thérapeutique ou de suivre à moyen terme les résultats d'un traitement.

On peut classer les modalités d'imagerie médicale par agent physique, par type d'images réalisées, par nuisance ou par leur utilisation médicale. Les agents physiques peuvent être des rayonnements de photons, des champs magnétiques ou des ondes ultrasonores. Les images peuvent être des images de projection planes, des images de coupe appelées images tomographiques, et des séquences temporelles de ces types d'images. Nous nous intéressons à des images mamographiques et des images tomographiques.

### ► Mammographie

Une mammographie (ou mastographie) est un examen radiologique du ou des seins. Il permet d'obtenir des images de l'intérieur du sein à l'aide de rayons X et de détecter ainsi certaines anomalies. Dans la mammographie numérique, les images du sein obtenues par rayons X sont acquises numériquement et peuvent être visualisées sur un moniteur (image vidéo) ou sur papier par suite de leur impression au laser.

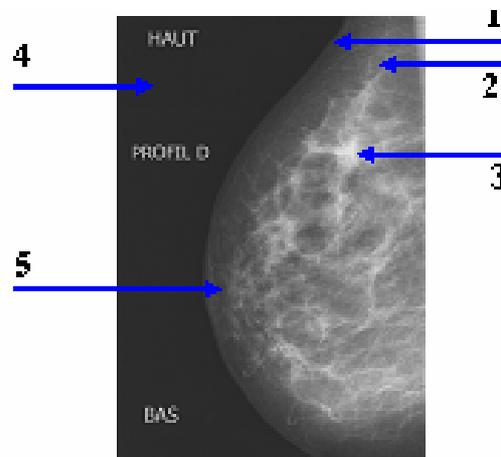
### ► La tomodensitométrie (TDM)

Ou scanographie, permet d'obtenir des coupes du corps. Un couple faisceau plat /détecteur linéaire numérique tourne autour de l'axe du corps et fournit une collection d'images autour du sujet. Chaque pixel contient le coefficient d'atténuation local aux rayons X.

## 3. Présentation des images médicales utilisées

Nous présentons quelque information sur le diagnostic de ces images médicales nécessaire pour savoir la qualité de la classification.

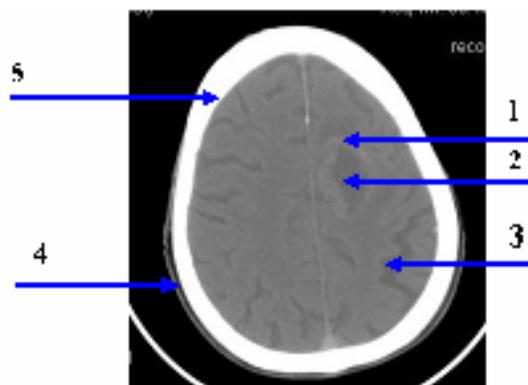
La figure 4.1 est un profil du sein à partir d'un examen mamographique qui présente au niveau du quadrant supérieur externe une opacité à contours irrégulier spéculés semblant contenir des micro calcifications fines et poudreuses.



**Figure 4.1** : Le profil du sein droit.

1. la peau 2. la graisse 3. l'opacité 4. le fond 5. le mamelon .

La figure 4.2 est une coupe axiale du crâne à partir d'un examen scanner qui contient une néoformation hypodense prenant le contraste en anneau irrégulier grossissement ovalaire mesurant environ 33 mm étendu sur 45mm entourée d'une zone hypodense correspondant à l'oedème péri-lésionnel. Cette masse de siège pariétal gauche est une tumeur grade III.



**Figure 4.2:** Coup axiale T1 du crâne (crâne 1)

1. La zone hypodense 2. La tumeur 3. Sillon central 4. La peau 5. L'os.

La figure 4.3 présente l'examen TDM du ce crâne en faveur d'une volumineuse masse fronto-temporo pariétale droite avec début d'engagement sous falcique évoquant une tumeur gliale.

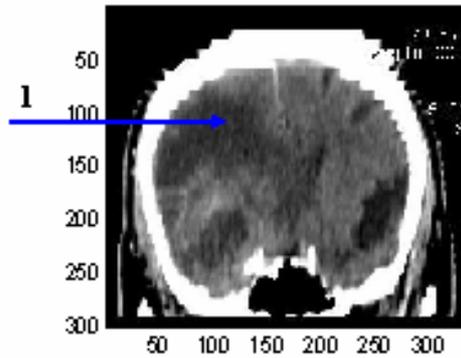


Figure 4.3 : Image du crâne (crâne 2).

1. Masse fronto-temporo pariétale.

**4. L’algorithme K\_means :**

Nous avons choisi de débiter par l’algorithme *k*-means car c’est le classificateur non supervisé le plus simple et le plus utilisé.

Les différentes images obtenues par l’implémentation de cet algorithme sont données par les figures 4.4 et 4.5

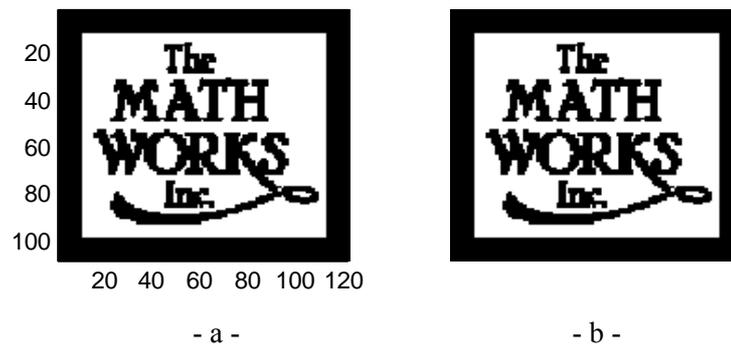


Figure 4.4: Résultat de la classification par K\_means pour K=2, a : image originale. b :image classifiée

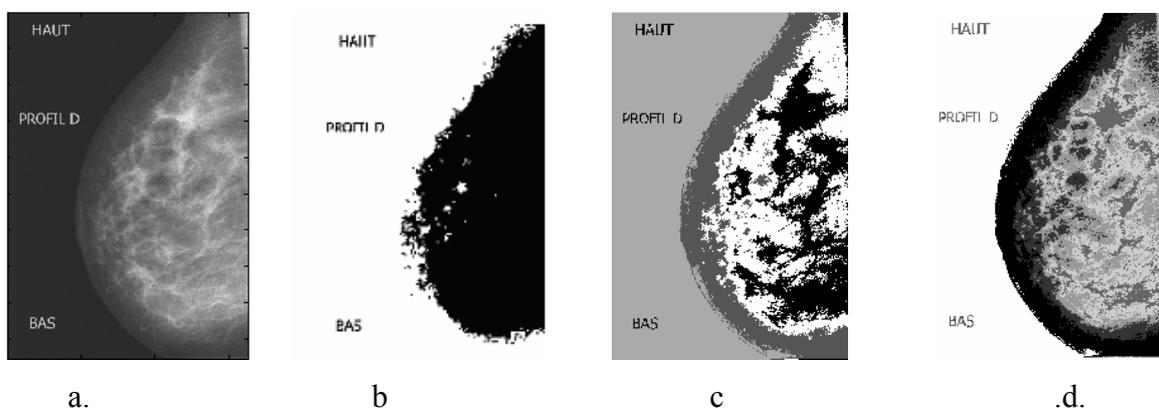


Figure 4.5 : Résultat de la classification par K\_means pour différentes valeurs de K. a : l’image originale. b, c, d :images classifiées respectivement pour K=2, 4, 6.

Pour obtenir une bonne classification il faut connaître le nombre exact des classes. sur certaines images il est facile de trouver ce nombre (exemple de la figure 4.4) où il est clair que le nombre de classes est égale à 2. Ce qui donne de meilleurs résultats. Cependant, souvent il est très difficile de sélectionner le nombre exact des classes c'est le cas des images médicales ou les images qui présentent un degré de chevauchement élevé. Plusieurs partitions ont été faites avec différentes valeurs de K en calculant les indices de validité Davies\_Bouldin et Dunn pour sélectionner le nombre exacte de classes. Les résultats relatifs à la figure (4.5) sont présentés dans le tableau suivant :

Nombre de classes k	2	3	4	5	6
Davies_Bouldin	0.413	0.382	<u>0.354</u>	0.3910	0.433
Dunn	0.0063	0.0081	0.0088	0.010	0.011

6

**Tableau 4.1:** Les indices de Davies\_Bouldin et Dunn.

D'après ce tableau la valeur optimale de classes est celle qui correspond à la valeur minimale de l'indice de Davies\_Bouldin qui est égale 0.354. Donc la meilleure partition est pour K=4. La figure (4.5.c) confirme ce calcul par ce que c'est le meilleur résultat parmi les autres. La valeur de l'indice de Dunn augmente avec le nombre de classe par ce que quand le nombre des classes augment la taille des classes sont diminuées et la valeur maximale intra classe diminue. Donc la valeur de l'indice augment. Cet indice ne donne aucune information sur la qualité de la classification (non représentatif).

## 5. La classification neuronale

Nous avons appliqué deux types de réseaux de neurones pour la classification non supervisée: les cartes de kohonen et le réseau Fuzzy ART. Initialement les versions originales ont été appliquées puis les algorithmes seront modifiés par la suite.

### 5.1 Application d'algorithme SOM

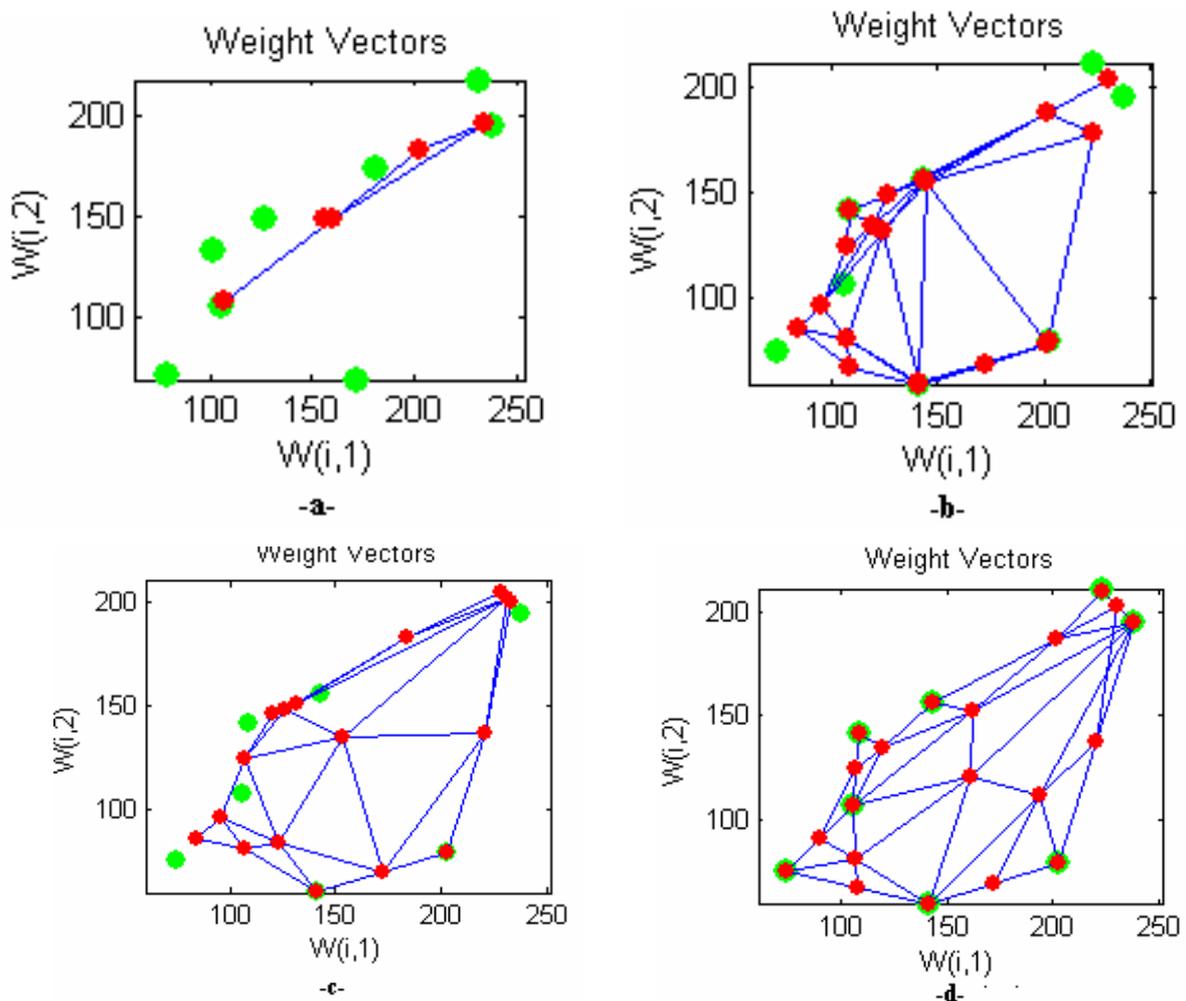
Tout processus de classification commence par une étape d'acquisition des observations qui consiste à déterminer les attributs caractérisant aux mieux les objets.

Dans ce travail, l'échantillon d'observations est constitué des pixels d'une image en niveaux de gris. Nous proposons de choisir l'échantillon d'observation par l'algorithme K\_means qui va calculer

les centres des classes selon le nombre de partition  $K$ , pour opérer en mode totalement non supervisé.

### 5.1.1 Visualisation de la carte

La figure (4.6) montre le déroulement de l'apprentissage. Il faut remarquer que les neurones convergent vers les composantes du vecteur d'apprentissage à chaque itération jusqu'à ce que tous les points de ce vecteur s'accordent avec les neurones de la carte. La visualisation de la carte de Kohonen est un support privilégié pour l'analyse averti qui peut l'exploiter pour prendre une décision concernant la qualité de la carte si elle est proche aux points du vecteur d'apprentissage qu'on a choisi ou non.

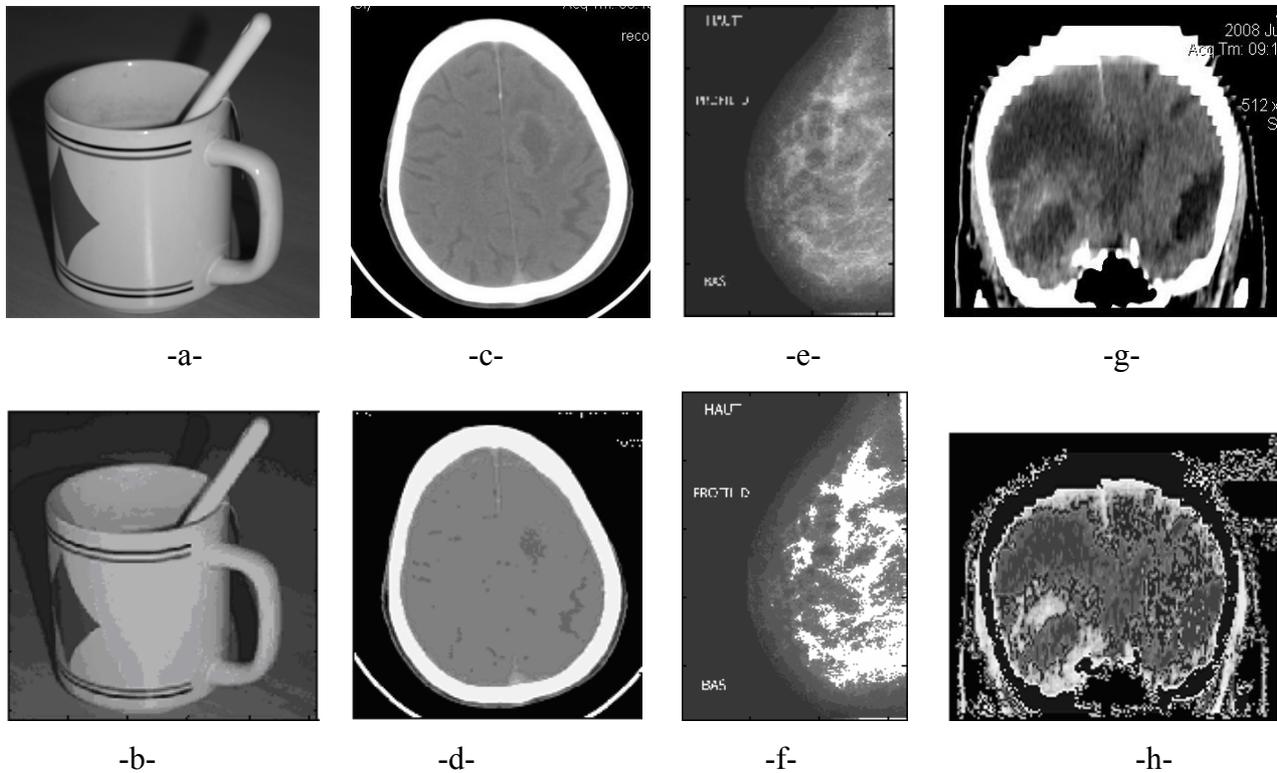


**Figure 4.6 :** Etat des neurones après a : 10, b : 500, c : 1000, d : 1500 itérations.

Les points rouges représentent les neurones de la carte et les points verts sont les points du vecteur d'apprentissage.

### 5.1.2. Classification de la carte

Initialement l'algorithme de SOM est appliqué directement pour la classification des images. Et les résultats sont illustrés sur la figure (4.7).



**Figure 4.7 :** Classification par l'algorithme SOM directement. a, c, e, g : les images originales.  
b, d, f, h : respectivement les résultats de la classification.

Les résultats de la classification sont acceptables, pour l'image (4.7.b) elle présente toutes les régions de façon claire et nette sauf les ombres de la tasse et de la cuillère qui sont presque fusionnées avec le fond et la couleur de la tasse qui n'est pas homogène.

L'image (4.7.d) représente la tumeur mais la zone hypodense n'est pas complètement détectée ainsi les sillons centraux ne sont pas séparés des autres composantes.

L'image (4.7.f) ne représente pas clairement les composantes du sein et c'est très difficile de séparer la peau et la graisse ainsi l'opacité qui prend une fausse forme.

L'image (4.7.h) est mal classifiée. Les classes ne sont pas bien séparées et elle ne peut donner aucune information sur le diagnostic.

Le deuxième niveau de la classification est basé sur la carte de Kohonen SOM.

A la convergence nous avons obtenu une carte qui constitue une bonne approximation de l'espace de départ. Nous avons appliqué la méthode SOM à deux niveaux qui a été exposée au chapitre deux.

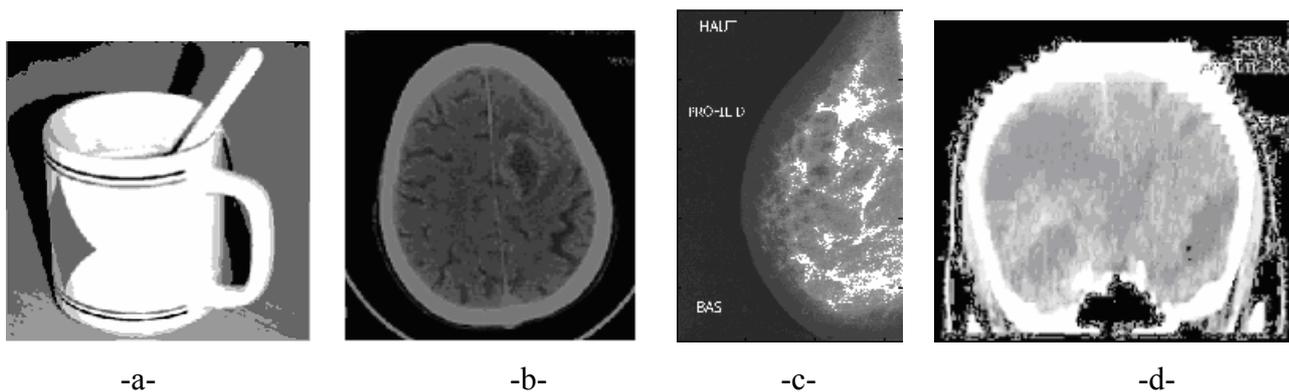
La classification ascendante hiérarchique a été appliquée pour regrouper d'une manière différente les

neurones de la carte. la distance du chaînage complet a été choisie.

La classification hiérarchique utilise cette partition initiale et procède à des regroupements successifs en fusionnant à chaque itération deux sous ensembles de neurones.

Avant de commencer la classification une étape de prétraitement a été ajoutée pour filtrer les neurones qui ne représentent aucune observation.

La classification est refaite avec les nouveaux poids qui représente les neurones obtenus par la classification hiérarchique. Les résultats obtenus illustrés par la figure (4.8).



**Figure 4.8** : Classification par SOM à deux niveaux

Une grande amélioration a été apportée par cette méthode. La comparaison entre les résultats de la figure (4.7) et ces résultats confirme cette amélioration.

Cette méthode a corrigé les défauts de la première méthode comme l'image (4.8.a) qui représente cette fois ci de façons plus distincte les classes à l'opposé de l'image précédente (figure 4.7.b). Et l'amélioration est remarquable sur l'image (4.8.b) où la tumeur est bien détectée et entourée complètement par la zone hypodense. Elle est continue. Aussi le sillon central est séparé des autres classes. L'image (4.8.c) montre l'amélioration de cette méthode où les classes sont bien détectées notamment l'opacité qui a pris sa vraie forme au contraire de l'image précédente quand l'opacité prend une espace très large. L'image de face de ce sein confirme ces résultats. Malgré le mauvais traitement de l'image originale par le manipulateur ou le technicien radiologue notre classification a bien détecté cette opacité.

Pour l'image (4.8.d) il est très facile de constater la différence avec la figure (4.7.h). Il est clair que les classes sont bien séparées.

## 5.2 Application de l'algorithme Fuzzy ART

La deuxième méthode utilisée dans ce travail est l'algorithme de Fuzzy ART. Initialement l'algorithme Fuzzy ART a été appliqué sans modification. Les résultats obtenus ne sont pas acceptables (voir figure 4.9 b) où l'image mamographique ne représente que la graisse et une petite

zone de l'opacité et l'image du crâne ne représente pas la tumeur complètement et elle est discontinue.

La première modification sur cet algorithme concerne les entrées du réseau celles-ci sont remplacées par un autre vecteur dont les valeurs sont prises entre 0 et 1 avec un écart type de 0.05.

Une modification a été introduit par le référence [20] consiste à remplacer la fonction de choix

$$T_j = \frac{|I \wedge w_j|}{\alpha + |w_j|} \text{ par la fonction } T_j = \frac{[|I \wedge w_j|]^2}{|I| \cdot |w_j|}$$

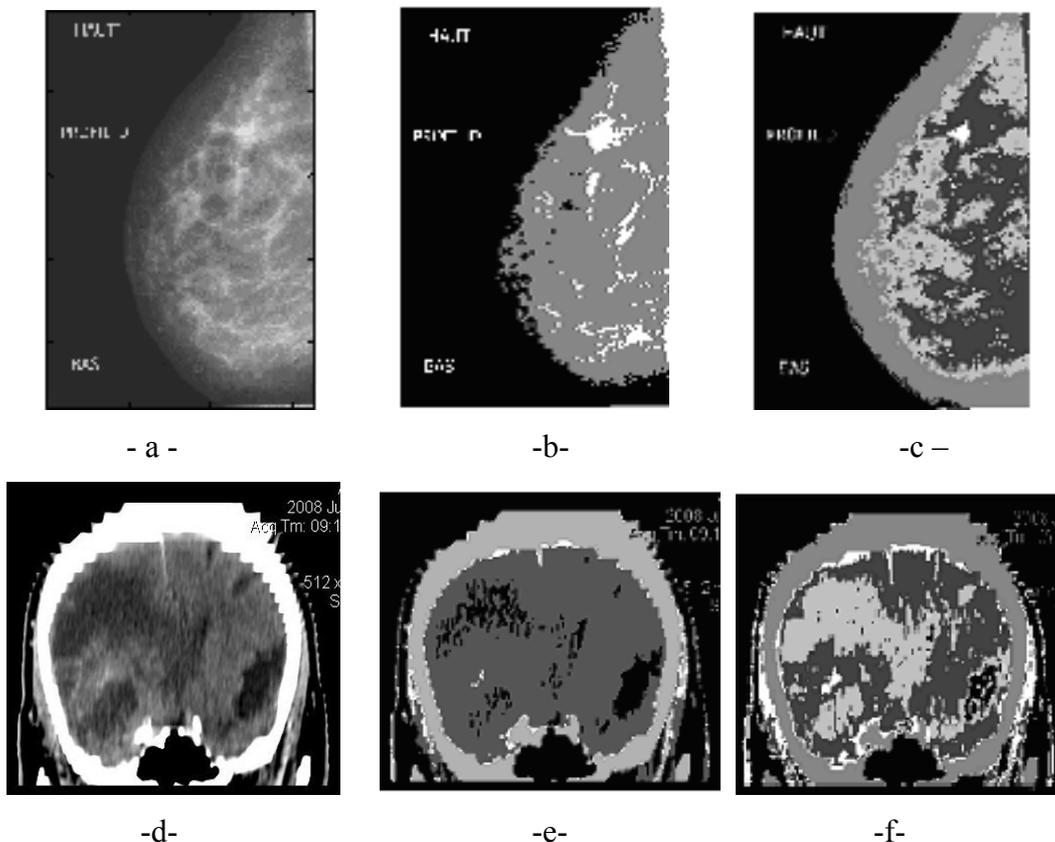
de test pour rendre le programme plus rapide. Nous avons testé les deux versions. Les résultats obtenus sont identiques tant sur les temps d'exécution que sur la qualité des images.

La deuxième modification sur la fonction du choix consiste à la remplacer par le coefficient de

similarité de Jaccard donné par :

$$T_j = \frac{|I \wedge w_j|}{|w_j| \vee |I|}$$

Les différentes images obtenues par l'implémentation de ces algorithmes sont données par la figure(4.9)



**Figure 4.9 :** Résultats obtenus par Fuzzy ART. a et d : Images originales , b , e :sans modification c,f : résultats obtenus par Fuzzy ART modifié utilisant le coefficient de similarité de Jaccard.

Les résultats obtenus par l'algorithme modifié utilisant le coefficient de similarité de Jaccard montre son efficacité. L'image (4.9.c) présente toutes les composantes du sein qui ne sont pas détecté dans l'image (4.9.b) surtout la peau. La détection de la tumeur dans l'image (4.9.f) est complète et elle prend une forme semblable à l'image originale.

## 6. Validité des classes

Malgré toutes ces améliorations l'observateur humain reste limité et il ne peut pas évaluer la qualité de la classification surtout pour définir le nombre exact des classes. Pour ces raisons une étape de validation a été introduite pour aider à fixer le nombre optimal de classes.

Les calculs des indices sont effectués pour chaque méthode et pour chaque partition. Une étape de fusionnement est aussi ajoutée pour corriger quelque faute de la classification. Cette méthode consiste à fusionner les deux classes qui sont voisines et qui sont semblables au sens de la distance mesurée. Les résultats de ces indices sont montrés sur les tableaux 4.2 et 4.3 :

Images		Algorithmes		SOM direct	SOM à deux niveau	Après 1 <sup>er</sup> fusionnement	Après le 2eme fusionnement
		CDbw	Davies_Bouldin				
La tasse	CDbw	2.612	<a href="#">304.57</a>	217.0	/		
	Davies_Bouldin	0.497	<a href="#">0.424</a>	0.542	/		
Le crâne 1	CDbw	437.88	<a href="#">1.04*10<sup>3</sup></a>	1.01*10 <sup>3</sup>	/		
	Davies_Bouldin	0.555	<a href="#">0.545</a>	0.634	/		
Le crâne 2	CDbw	245.98	986.71	<a href="#">1.11*10<sup>3</sup></a>	711.74		
	Davies_Bouldin	0.488	0.481	<a href="#">0.454</a>	0.565		
Le sein	CDbw	10.966	318.260 (6)	<a href="#">1226.2</a>	230.2519		
	Davies_Bouldin	0.423	0.501	<a href="#">0.422</a>	0.431		

**Tableau 4.2** : Les valeurs des indices CDbw et Davies\_Bouldin pour chaque image.

Algorithmes		Fuzzy ART original	Fuzzy ART modifié
Le sein	CDbw	24.8236	<a href="#">56.7475</a>
	Davies_Bouldin	0.3604	<a href="#">0.3416</a>
Le crâne 2	CDbw	92.6650	<a href="#">94.1026</a>
	Davies_Bouldin	0.3889	<a href="#">0.3070</a>

**Tableau 4.3 :** Les valeurs des indices CDbw et Davies\_Bouldin pour l'algorithme de Fuzzy ART

La valeur maximale de CDbw et la valeur minimale de l'indice de Davies\_Bouldin indiquent que le nombre des classes correspondant est le meilleur. L'analyse de ce tableau permet de sélectionner le nombre optimal de classes de chaque image :

La tasse : les deux indices de CDbw et Davies\_Bouldin donnent des résultats semblables concernant le nombre de classes qui est égale à 5 relatif à la valeur maximale de CDbw et à la valeur minimale de Davies\_Bouldin.

Le crâne 2 : les deux indices montrent que la meilleure partition est obtenue par le premier fusionnement. Donc il n'est pas nécessaire de continuer le fusionnement par ce que la valeur de l'indice CDbw pour le deuxième fusionnement est inférieure à la valeur précédente. Pour bien comprendre ce mécanisme voici l'exemple suivant :

$$\mathbf{M} = \begin{matrix} & 5.89 \cdot 10^6 & 1.82 \cdot 10^6 & 0.877 \cdot 10^6 & 1.05 \cdot 10^6 & 0.0001 \cdot 10^6 \\ \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 1.815 \cdot 10^6 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 2.0478 \cdot 10^6 \\ 11.67 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 1.476 \cdot 10^6 \\ 6.382 \\ \underline{5.203} \\ 0 \end{matrix} & \begin{matrix} 1.994 \cdot 10^6 \\ 41.753 \\ 7.342 \\ 15.476 \end{matrix} \end{matrix}$$

Où  $\mathbf{M}$  est la matrice qui contient les valeurs de l'indice CDbw pour tous les couples de classes. Le nombre total des classes est égale à 6.

Le mécanisme de fusionnement consiste initialement à chercher la valeur minimale entre un couple de classes (dans l'exemple la valeur minimale est égale à 5.203) donc le couple correspondant à

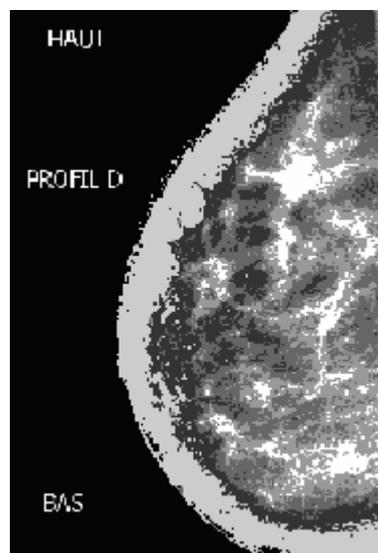
cette valeur est constitué des classes 5 et 4. Le deuxième test est effectué pour confirmer si ce deux classes sont voisines ou non. Si oui les deux classes seront fusionnées dans une seule classe. puis l'indice CDwb est recalculé et donnent une valeur égale a 711.74 inférieure à la valeur précédente ( $1.11 \cdot 10^3$ ). Donc le fusionnement est arrêté et la valeur optimale des classes est égale à 6.

L'analyse du tableau 4.3 montre que l'algorithme Fuzzy ART modifié est meilleur que l'algorithme Fuzzy ART original où les valeurs maximales de l'indice de CDwb et la valeur minimale de l'indice de Davies et Bouldin sont données par l'algorithme Fuzzy ART modifié.

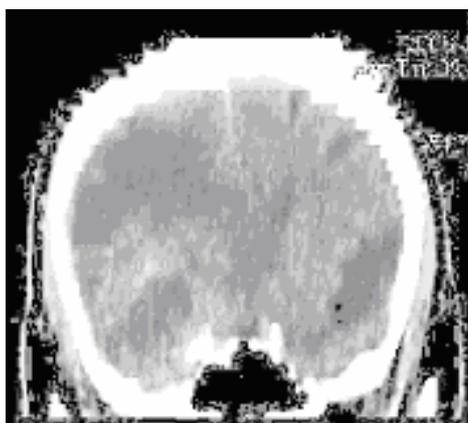
Les images de la figure (4.10) confirment les résultats numériques . Nous ne donnons que les images qui sont changées par l'étape de fusionnement.



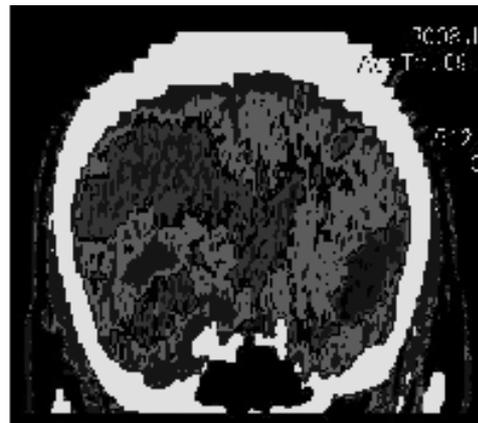
-a-



-b-



-c-



-d-

**Figure 4.10 :** -a et c image avant validité -b- et d image après validité.

Comme le montre la figure (4.10), les images après validité sont bien classifiées, les contours sont bien délimités.

**Remarque :**

L'étape de fusionnement a été faite seulement pour les résultats obtenus par l'algorithme de SOM à deux niveau par ce que dans l'algorithme du Fuzzy ART modifié le problème de prolifération des classes a été résolu. Donc cette étape n'est plus nécessaire.

**7. Validité des classes par l'indice proposé**

Cet indice est appliqué pour les méthodes de classification neuronal surtout pour l'algorithme de classification SOM .

Cet indice est basé sur la densité qui a été calculée pour toutes les classes. Le nombre maximal des classes qui est le résultat de la classification dans une première phase permet de trouver le seuil qui va partager les classes en deux catégories : la catégorie des classes valides et l'autre les classes refusées.

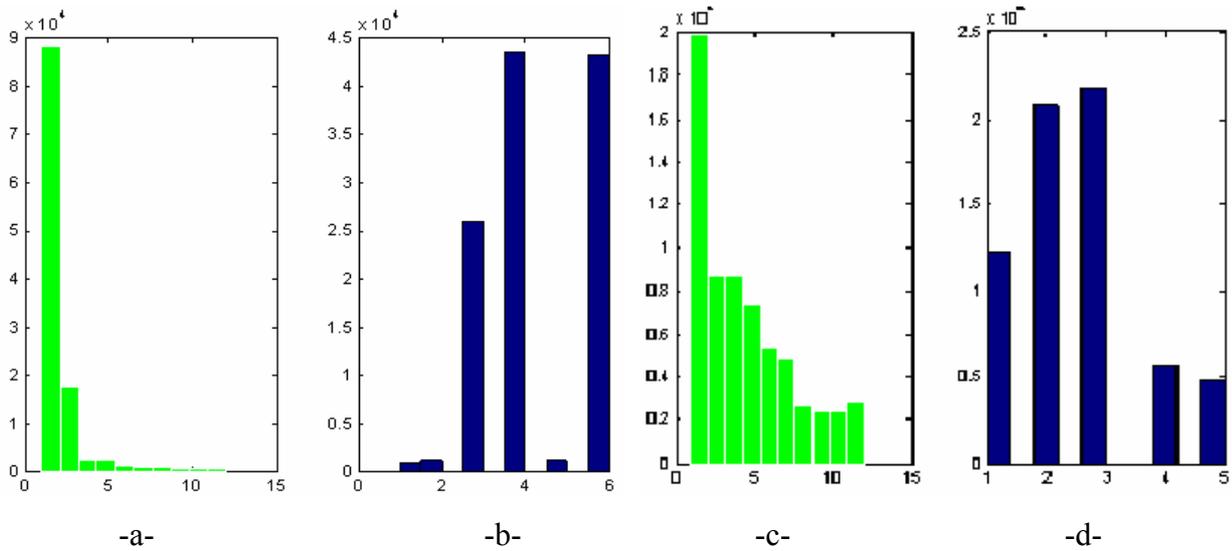
La valeur de pourcentage de classes valides permet d'évaluer les résultats de la classification qui sont illustrés sur la figure (4.7).

<b>Images</b>	<b>Les pourcentages</b>
La tasse	68.333 %
Le crâne 1	96.797 %
Le crâne 2	72.970 %
Le sein	85.214 %

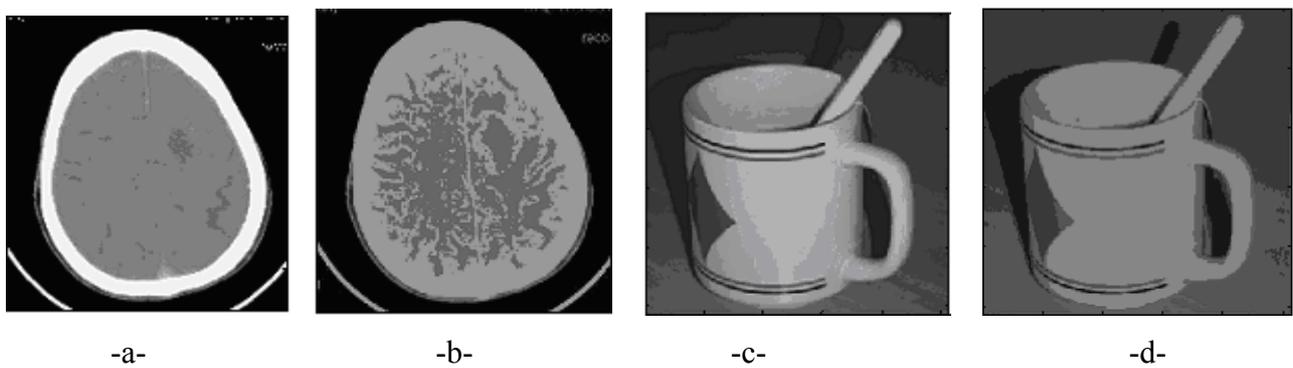
**Tableau 4.4 : Les pourcentages des images classifiées.**

Différentes valeurs de pourcentage pour chaque image sont donnés dans le tableau 4.4. Ces valeurs donnent un regard général sur les éléments qui sont bien classés. Pour l'image classifiée de la tasse par exemple elle présente 68.333 % des classes valides et les 31.667% sont des classes refusées qui nécessitent de refaire sa classification.

Les histogrammes des différentes images sont donnés sur la figure 4.11.



**Figure 4.11** : Histogramme des images : a, c: avant correction (SOM direct). b, d : après correction



**Figure 4.12** : Résultat de la classification. a, c image classifiées par l'algorithme de SOM direct  
b,d images après la correction de l'indice proposé

D'après l'histogrammes - a - le nombre de classes obtenu par l'algorithme SOM direct est égale a 10 avec un taux de bonne classification égale à 96.797%. Les 4% présentent les classes refusées à cause de leur petite tailles par rapport aux autres classes comme il est montré par l'histogramme (4.11.a). Après l'application de notre indice, le nombre de classes a pris la valeur 6 respectant le nombre d'objets présents dans l'image (4.12.b). La comparaison entre les deux images obtenues par SOM direct et la correction par notre indice confirme l'amélioration apportée par notre indice où les composantes qui ne sont pas présentes à la première image sont bien détectées dans la deuxième. Ainsi la tumeur et la zone hypodense ainsi que les sillons centraux sont bien séparés des autres classes, sauf l'os qui est fusionné aux autres classes. Ceci est confirmé par le médecin radiologue qui affirme qu'il n'est pas possible d'avoir toutes les composantes du crâne sur une seule

image. Dans le système du scanner le contraste de l'image est réglé selon les besoins. Pour ces arguments l'image apporte toutes les informations nécessaires pour faire le bon diagnostic.

Pour l'image de la tasse le nombre de classe après notre correction est égale à 5 (comme il est montré par l'histogramme 4.11.d) qui confirme les résultats précédentes obtenus par les calculs des indices CDbw et Davies\_Bouldin.

## **8. Conclusion**

La classification non supervisée est réalisée à l'aide des méthodes basées sur les réseaux de neurones comme les cartes auto\_organisatrices de Kohonen (SOM) et Fuzzy ART qui ont montrés leur efficacité.

Le problème majeur de la classification non supervisée est la détermination de nombre de classes. Ce problème a été résolu à l'aide de différents indices de validité.

# Conclusion générale

## Conclusion Générale

La classification non supervisée permet de former des groupes d'objets susceptibles d'être intéressants pour l'utilisateur. Notons qu'il est fréquemment possible de construire différentes partitions d'un même ensemble d'objets et en absence d'informations complémentaires, le choix de l'une ou l'autre est nécessairement arbitraire.

Dans ce travail l'algorithme de K\_means et la classification neuronale sont appliqués. Deux réseaux de neurone ont été choisis pour la classification non supervisée à savoir les cartes auto organisatrices de kohonen (SOM) et les réseaux ART qui sont des réseaux à apprentissage non supervisé. Les méthodes basées sur ces réseaux de neurones sont SOM à deux niveaux et fuzzy ART avec et sans modification. Les différentes méthodes appliquées ont donné une bonne classification.

L'algorithme des k\_means nécessite une connaissance préalable du nombre de classe à déterminer, ce qui rend impossible une éventuelle automatisation du processus et l'initialisation des centres est responsable de la qualité des classes obtenues. En effet, des centres mal initialisés (c à d concentrés dans la même zone) entraînent souvent une mauvaise classification où la lenteur d'exécution de l'algorithme. L'indice de Davies\_Bouldin a été introduit pour sélectionner la meilleur partition. Et aussi elle ne permet pas une bonne classification des régions qui ne sont pas clairement séparées. A l'opposé les cartes d'auto-organisatrices de Kohonen permettent une bonne classification même pour des images qui ont des degrés de chevauchement élevés.

Il apparaît que l'algorithme de k\_means n'est pas le plus adapté à ce type des images. Mais si l'on tient à l'utiliser a cause de simplicité il vaut mieux associer un indice de validité pour définir le nombre optimal de classes pour les images qui présente un chevauchement relativement fort.

Nous avons conclu que le choix du vecteur de l'ensemble d'apprentissage est très important dans l'étape d'apprentissage. SOM est une plat forme efficace pour la visualisation des données de grande dimension. Nous l'avons utilisé comme un premier niveau dans un processus de classification pour plusieurs avantages : d'abord la réduction du coût informatique et en second lieu la représentation visuelle des données tout en conservant leur topologie. La méthode SOM à deux niveaux permet une bonne classification même pour des images qui ont des degrés de chevauchement élevé.

L'algorithme de classification Fuzzy ART est la méthode la plus rapide parmi les autres méthodes de la classification.

Les différentes méthodes de classification ont permis de construire différentes partitions d'un même ensemble d'objets et les indices de validité peuvent être aidés dans ce sens. L'indice proposé donne de bons résultats concernant sa capacité de corriger les partitions obtenues ainsi que la réduction du coût informatique évitant la répétition des partitions à chaque fois.

Les réseaux de neurones sont donc très efficaces pour les problèmes de classification de différents types d'images et surtout les images médicales.

# Bibliographie

## Bibliographie

- [1] B. Stein et M. Busch “ Density-based Cluster Algorithms in Low-dimensional and High-dimensional Applications “ *Fachberichte Informatik*, pp 45-56, 2005.
- [2] M. Talibi-Alaoui, R. Touahni et A. Sbihi “ Classification des images couleurs par association des transformations morphologiques aux cartes de Kohonen “ *Cari-Hammamat*, pp 83-90, 2004.
- [3] K. Lung Wu , M-Shen Yang “ Mean shift-based clustering ” *Pattern Recognition*, vol40 pp 3035-3052, 2007
- [4] N. Ouarab, Y. Smara, J-P. Rassin “ Utilisation de méthodes de classification hiérarchique pour une classification supervisée d'images satellitaires” *Dix-septième colloque GRETSI, Vannes*, pp 13-17 ,1999.
- [5] G. Dreyfus et al “ *Réseaux de neurones méthodologie et applications* “ éditions EYROLLES. 2004
- [6] N. Monmarché “ Algorithmes de fourmis artificielles : applications à la classification et à l'optimisation “ Université François Rabelais Tours , 2000.
- [7] S. Guérif “ Réduction de dimension en Apprentissage Numérique Non Supervisé “ Thèse , université Paris 13 , 2006.
- [8] M. Parizeau “ *réseaux de neurones* “ *GIF-21140 et GIF-64326*,. Université LAVAL, 2004.
- [9] G. Burel “ Réseaux de neurones en traitement d'images des models théorique aux application industrielles “ Thèse doctorat de l'université de Bretagne occidentale,1991
- [10] D.O. Hebb ” The organization of behaviour “ *J. WILEY and Sons*, 1949.
- [11] F. Rosenblatt “ Principles of neurodynamics “ *Spartan, New York*, 1962.
- [12] T. Kohonen “ Self-Organizing Maps “ *Springer Series in Information Sciences*, Vol. 30, 2nd edition, 1997.
- [13] J.Vesanto, E Alhoniemi “clustering of the self-organizing map “ *IEEE trans. neural Neutwork* , vol 11 N°3, pp 586-600, 2000
- [14] H.Bélangier, “ Réseaux de Kohonen pour la ditection des contours d'objets dans une image à niveaux de gris “ thèse Montréal, 1998.
- [15] J. Vesanto & Mika Sulkava “ Distance matrix based clustering of the Self- Organizing Map” *Computer Science*, vol 2415, pp 951-956, 2002

- [16] S. Wu & T.W.S Chow, " clustering of self-organising map using a clustering validity index on inter-cluster and intra-cluster density " *pattern recognition*, vol 37 ,pp 175-188 2004.
- [17] D. L. Davies and D. W. Bouldin, " A cluster separation measure " *IEEE Trans. Patt. Anal Machine Intell.*, vol. PAMI-1, pp. 224–227, 1979
- [18] S. Grossberg " Adaptive pattern classification and universal recoding. I: Parallel development and coding of neural feature detectors " *Biological Cybernetics*, N° 23 pp 121-134, 1976.
- [19] Gail A et al " Fuzzy ART: Fast Stable Learning and categorization of Analog Patterns by an Adaptive Resonance System " *Neural Networks*, Vol. 4, pp 759-771, 1991.
- [20] L.cinique,G.Foresti,L.Lombardi " A clustering fuzzy approach for image segmentation " *Pattern Recognition*, vol 37 pp 1797-1807. 2004.
- [21] R. Ozdemir, G.gencyilmaz T.Aktin " the modifier fuzzy art and a two-stage clustering approach to cell design " *Information science* vol 177 pp 5219-5236, 2007
- [22] B. Stein, S.Meyer zu, E.Wißbrock " On Cluster Validity and the Information Need of Users " *ACTA Press*, pp 216-221, 2003.
- [23] O. Ammor, A. Lachkar, N. Rais & K, Slaoui " New efficient approach to determine the optimal number of clusters in overlapping cases" *IEEE* pp 26-31, 2006.
- [24] Ch.Yu Yen, K. J. Cios " Image recognition system based on novel measures of image similarity and cluster validity " *Neurocomputing*, 2008.
- [25] O. Ammor , N. Rais & K. Slaoui. " Détermination du nombre optimal de classes présentant un fort degré de chevauchement " *REVUE MODULAD* N° 37,pp 32-45, 2007
- [26] J.C. Dunn " Well separated clusters and optimal fuzzy partitions ». *J. Cybern.* Vol 4, pp 95-104 , 1974.
- [27] L. Hubert, J. Schultz. " Quadratic assignment as a general data-analysis strategy " *British Journal of Mathematical and Statistical Psychologie* vol 29. pp 190-241, 1976
- [28] P.J. Rousseeuw " Silhouettes: a graphical aid to the interpretation and validation of cluster analysis " *Journal of Computational and Applied Mathematics*. Vol 20. pp 53-65. 1987.