

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERY de Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique

Département d'informatique



# ***MEMOIRE***

## ***DE FIN D'ETUDE***

*En vue de l'obtention d'un diplôme de Master Académique en Informatique.  
Option : Système Informatique.*

---

*Classification des images avec les réseaux de neurones  
Convolutionnels*

---

*Dirigé par :*

 *Madame FELLAG.*

*Réalisé par :*

 *Mr BELHADJER Hakim.*  
 *Mr SAROUER Brahim.*

*Promotion 2017-2018*

## **Sommaire**

### **CHAPITRE 1**

<b>I- Introduction :</b> .....	<b>9</b>
<b>II. Extraction de connaissances à partir de données (le processus d'ECD) :</b> ..	<b>9</b>
II.1 Les étapes du processus ECD : .....	10
II.1.1 L'acquisition de données : .....	10
II.1.2 Le prétraitement : .....	10
II.1.3 Le data mining : .....	11
II.1.4 L'évaluation et la présentation : .....	11
<b>III- Les domaines d'application :</b> .....	<b>11</b>
III.1- Secteur bancaire : .....	11
III.2- Secteur des assurances : .....	11
III.3- Secteur médical : .....	12
III.4- Le commerce : .....	12
III.5- Secteurs divers : .....	12
<b>IV- Data mining sur quels types de données :</b> .....	<b>12</b>
IV.1- Les fichiers plats : .....	12
IV.2- Les bases de données relationnelles : .....	12
IV.3- Les bases de données transactionnelles : .....	13
IV.4- Les bases de données multimédia : .....	13
IV.5- La base de données spatiale : .....	13
IV.6- Les bases de données de série temporelle : .....	13
<b>V- Tâches de data mining :</b> .....	<b>14</b>
V.1- La classification : .....	14
V.2- L'estimation : .....	14
V.3- La prédiction : .....	14
V.4- Les règles d'association : .....	14
V.5- La segmentation : .....	14
V.6- La description : .....	15
<b>VI- Les techniques de Data Mining :</b> .....	<b>15</b>
VI.1- Analyse du panier de la ménagère : .....	15
VI.2- Le raisonnement basé sur la mémoire : .....	15

VI.3- La détection automatique de clusters : .....	15
VI.4- L'analyse des liens : .....	16
VI.5- Les arbres de décision : .....	16
VI.6- Les réseaux de neurones : .....	16
VI.7- Les algorithmes génétiques : .....	16
VI.8- Les agents intelligents ou knowbot : .....	17
<b>Conclusion : .....</b>	<b>17</b>
<b>I- Introduction.....</b>	<b>19</b>
<b>II- Apprentissage automatique : .....</b>	<b>19</b>
II.1    Apprentissage Supervisé : .....	19
II.2    Apprentissage Non-Supervisé : .....	19
<b>III-La classification : .....</b>	<b>20</b>
III.1- Introduction : .....	20
III.2- Définition : .....	20
III.3- Les étapes d'une classification : .....	20
III.4- Mesure de la similarité : .....	20
III.4.1 :- Fonctions de similarité.....	21
III.5- Les méthodes de la classification : .....	22
III.5.1- Méthodes non supervisé : .....	23
III.5.1.1- Définition : .....	23
III.5.1.2- Principe : .....	23
III.5.2- Méthodes supervisée (Classification) : .....	24
III.5.2.1- Définition : .....	24
III.5.2.2- Principe : .....	24
III.5.2.3    Quelques algorithmes de la classification supervisée : .....	24
III.5.2.3.1    Arbre de décision : .....	24
III.5.2.3.2    Structure d'un arbre de décision : .....	25
III.5.2.3.3    Algorithme de construction d'un arbre de décision : .....	26
III.5.2.3.4    Les k plus proches voisins (K-PPV) : .....	27
III.5.2.3.5    La classification bayésienne : .....	28
III.5.2.3.6    Les réseaux de neurones : .....	29
<b>Conclusion : .....</b>	<b>30</b>
<b>I- Introduction : .....</b>	<b>32</b>

<b>II- Les réseaux de neurones artificiels (RNA) :</b>	<b>32</b>
II.1- Comportement d'un réseau de neurones artificiel :	33
II.1.1- La somme pondérée :	33
II.1.2- La fonction d'activation :	33
II.2- Classification des réseaux de neurones artificiels :	36
II.2.1- Les réseaux non bouclé (statique ou feed forward) :	36
II.2.1.1- Le perceptron mono-couche :	36
II.2.1.2- Perceptron multicouche :	37
II.2.1.3- Les réseaux de neurones RBF :	38
II.2.3- Les réseaux de neurones récurrents :	39
<b>III-La classification d'image avec les réseaux de neurones convolutifs :</b>	<b>40</b>
III.1- Introduction :	40
III.2- L'infographie et les images :	40
III.2.1- Définition d'image :	40
III.2.2- Caractéristiques d'une image :	40
III.2.3- Différents types d'images :	41
III.2.4- Codage d'images numériques :	41
III.2.5- La numérisation :	41
<b>IV. Traitement numérique des images :</b>	<b>42</b>
IV.1 Filtrage Numérique :	42
IV.1.1 Filtres linéaires :	42
IV.1.2 Filtres non linéaire :	43
<b>V Les réseaux de neurones convolutionnels :</b>	<b>45</b>
V.1. Définition :	45
V.2. Architecture de réseaux de neurones convolutionnels :	45
V.3. Fonctionnement d'un réseau de neurones convolutifs :	47
V.3.1 Couche convolution :	47
V.3.2 Unités linéaires rectifiées (ReLu) :	48
V.3.3 La couche de pooling :	49
V.3.4 Couche entièrement connectée :	50
V.3.5 L'entraînement d'un réseau de neurone convolutionnelle :	51
V.4 Sur-apprentissage et sous-apprentissage :	51
V.5 Indicateurs de performance d'un classifieur :	52

<b>Conclusion</b> .....	<b>55</b>
<b>I- Introduction</b> .....	<b>57</b>
<b>II- Description du notre système</b> .....	<b>57</b>
<b>III-Présentation des outils utilisés dans l'implémentation</b> .....	<b>59</b>
<b>IV. Création d'un réseau de neurone convolutifs avec Keras</b> .....	<b>60</b>
IV.1 Création d'une base de données image (images de train et images de test) .....	60
IV.2 Création d'un modèle .....	61
IV.3 Compilation et évaluation de modèle .....	61
<b>IV- Architecture de notre réseau</b> .....	<b>62</b>
V.1 -Architecture du premier modèle .....	62
V.2- Architecture du deuxième modèle .....	63
V.3- Architecture du troisième modèle .....	64
<b>V- Analyse et discussion</b> .....	<b>64</b>
VI.1- Le graphe de précision et le graphe d'erreur .....	64
VI.2- La matrice de confusion .....	68
VI.3- Tableau de comparaison de résultat obtenu .....	72
<b>VII. Code source de troisième modèle</b> .....	<b>73</b>
<b>Conclusion</b> .....	<b>76</b>
<b>Conclusion générale</b> .....	<b>77</b>
<b>Bibliographie</b> .....	<b>78</b>
<b>Annexe</b> .....	<b>79</b>

## *Listes des figures*

Figure 1 : Les différentes phases de l'ECD .....	09
Figure 2 : Structure d'un arbre de décision.....	25
Figure 3 : Exemple d'un arbre de décision .....	25
Figure 4 : K plus proches voisins .....	28
Figure 5 : La structure d'un neurone artificiel.....	32
Figure 6 : Structure d'un neurone artificiel pour le neurone amont d'indice i.....	32
Figure 7 : Fonction linéaire .....	33
Figure 8 : Fonction sigmoïde .....	34
Figure 9 : Fonction Heaviside.....	35
Figure 10 : Fonction signe.....	35
Figure 11 : Fonction linéaire à seuil.....	35
Figure 12 : Classification des réseaux de neurones .....	36
Figure 13 : Perceptron simple. ....	37
Figure 14 : Perceptron multicouche. ....	38
Figure 15 : Réseaux de neurones récurrents .....	39
Figure 16 : Codage sur 3 octets d'un morceau d'image couleur. ....	42
Figure 17 : Principe du filtre médian. ....	44
Figure 18 : Principe du filtre maximum.....	44
Figure 19 : Principe du filtre minimum. ....	44
Figure 20 : Opération d'une convolution sur image de 5*5 pixels.. ....	46
Figure 21 : Architecture d'un réseau de neurones convolutionnels. ....	47
Figure 22 : Etape de convolution .....	48
Figure 23 : Fonction Relu. ....	49
Figure 24 : Max-Pooling avec un filtre 2x2 et un pas de 2.....	49
Figure 25: Mise en commun appliquée aux cartes de fonctions rectifiées. ....	50
Figure 26 : La matrice de confusion d'un classifieur .....	52
Figure 27 : Le graphe de décision d'un réseau de neurone convolutifs .....	53
Figure 28 : Le graphe d'erreur d'un réseau de neurone convolutifs .....	53
Figure 29 : Architecture du système.....	58
Figure 30 : La liste d'image .....	60
Figure 31 : Architecture du premier modèle .....	62
Figure 32 : Architecture de deuxième modèle .....	63
Figure 33 : La courbe de décision du premier modèle.....	64
Figure 34 : La courbe d'erreur du premier modèle.....	65
Figure 35 : La courbe de décision du deuxième modèle.....	65
Figure 36 : La courbe d'erreur du deuxième modèle.....	66
Figure 37 : La courbe de précision du troisième modèle .....	66
Figure 38 : La courbe d'erreur du troisième modèle .....	67
Figure 39 : Matrice de confusion de premier modèle .....	68
Figure 40 : Le nombre d'images bien et mal classé.....	68
Figure 41 : La matrice de confusion de deuxième modèle.....	69

<b>Figure 42 : Nombre d'images bien et mal classé .....</b>	<b>70</b>
<b>Figure 43 : La matrice de confusion de troisième modèle .....</b>	<b>71</b>
<b>Figure 44 : Nombre d'images bien et mal classé .....</b>	<b>71</b>
<b>Figure 45 : Tableau de comparaison de résultat .....</b>	<b>72</b>

## **Introduction générale :**

Le data mining est l'ensemble des outils qui permettent l'extraction d'informations à partir de grand volume de données, ces informations sont utiles à la décision.

Ces outils ne se limitent pas au traitement des données structurées, ils offrent également des moyens d'extraire des informations à partir des images, car avec l'émergence des réseaux sociaux et des objets connectés, les bases de données sont inondées d'images.

Il existe différentes techniques d'extraire des connaissances à partir des images, mais la plus utilisée et la plus efficace est l'approche neuronale, qui fait partie des méthodes de classification.

Les réseaux de neurones convolutifs réalisent des résultats impressionnants de classification d'images et la reconnaissance des formes, c'est pourquoi ils font l'objet de beaucoup de recherches actuellement.

Notre travail, intitulé « Classification d'images avec les réseaux de neurones convolutifs », consiste à la création d'un classifieur d'images en utilisant la technologie « Tensorflow », qui est développé et rendu open source par Google.

Dans cette optique, nous avons divisés notre travail en quatre parties, les trois premières développent l'aspect théorique, la dernière sera dédié à la réalisation.

Dans la première partie intitulée « Data mining », nous avons présenté l'approche du Data mining et les techniques qui utilise pour extraire des connaissances à partir des données.

Dans la deuxième partie, nous mettons en évidence la tâche la classification en expliquant les différents types de classification existant, les différentes étapes à suivre pour faire la classification.

La troisième partie est consacrée à l'étude des réseaux de neurones, la classification d'images avec les réseaux de neurones convolutifs et leur fonctionnement.

La quatrième et dernière partie est dédiée à l'implémentation de notre application, en expliquant les différentes architectures des réseaux utilisés ainsi que les outils de réalisation.

**Premier chapitre :**

# **Data mining (fouille de données)**

## I- Introduction :

Le volume des données circulant sur le web (Réseaux sociaux, objets connectés) ou stockées par les entreprises est en croissance continue.

Afin de pouvoir exploiter cette richesse, il est nécessaire d'extraire des connaissances à partir de très grand volume de données (d'informations).

Le domaine ayant pour but de résoudre cette problématique est la science de données (data science). La science de données permet l'extraction de connaissances à partir de tout type de données, le data science regroupe plusieurs disciplines comme la statistique, les mathématiques, l'intelligence artificielle et la fouille de données (data mining).

Le data mining offre des méthodes et des outils d'analyse très utiles pour l'extraction de connaissances.

L'extraction de connaissances ne se fait pas uniquement à partir de données sous forme de tableau numérique, mais aussi à partir des images qui inondent aujourd'hui les bases de données (Réseaux sociaux, imagerie médicale, images satellite, images télescope).

## II. Extraction de connaissances à partir de données (le processus d'ECD) :

L'extraction de connaissances à partir de données est un processus homme-machine itératif et interactif d'analyse d'un grand ensemble de données afin d'extraire des connaissances à partir de ces données et de prédire le comportement face aux nouvelles données.

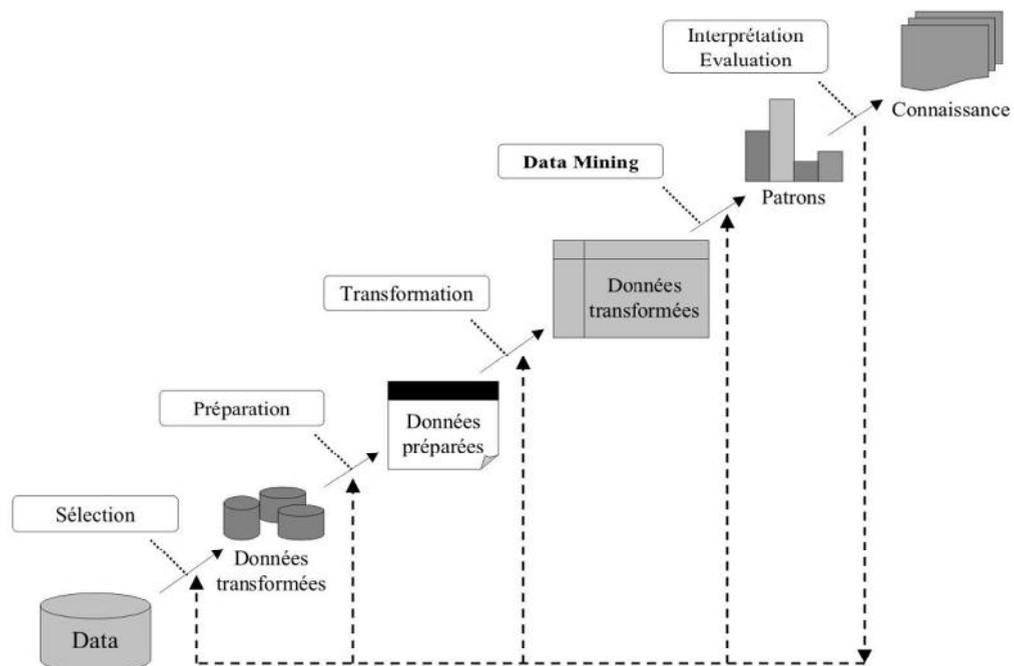


Figure 1 : Les différentes phases de l'ECD [1]

## II.1 Les étapes du processus ECD :

### II.1.1 L'acquisition de données :

En générale, les données peuvent être sur des sites différents de celui où s'effectue l'ECD, et peuvent être stockées sous différentes architectures : dans des bases de données relationnelles, dans des entrepôts de données, sur le web ou dans des banques de données spécialisées (images, bibliothèques ou librairies numériques, bases de données génomiques...), peuvent être structurées ou non selon des types différents : images, sons, séquences vidéo, données tabulaires, ou textuelles. [2]

### II.1.2 Le prétraitement :

Le prétraitement des données est une étape cruciale dans le processus de découverte de connaissances à partir de grandes bases de données. En effet, il permet d'améliorer la qualité des données soumises par la suite aux algorithmes de data mining. Le prétraitement des données est constitué de différentes étapes successives. [3]

Dû à la grande taille des bases de données actuelles, les données brutes sont généralement de faible qualité. Elles peuvent être incomplètes (valeurs manquantes ou agrégées), bruitées (valeurs erronées ou aberrantes) ou incohérentes (divergence entre attributs). L'application d'algorithmes de data mining sur de telles données complexifie l'apprentissage et nuit à la performance ainsi qu'à la fiabilité du modèle :

- Nettoyage
- Intégration
- Transformation
- Réduction
- Discrétisation

Les techniques de prétraitement de données offrent les possibilités suivantes :

- Identifier et traiter les données manquantes par remplacement (ignorer le tuple, compléter les données, utiliser la valeur la plus probable).
- Identifier les données aberrantes et éliminer le bruit des données par lissage, clustering hiérarchique ou régression.
- Identifier les données redondantes par une analyse de corrélation.
- Résumer les données à différents niveaux de granularité.
- Réduire les données par sélection d'attributs.
- Normaliser les données.

### **II.1.3 Le data mining :**

La fouille de données constitue véritablement le cœur du processus d'ECD, elle est souvent difficile à mettre en œuvre et coûteuse. Elle fait le choix des techniques et algorithmes appropriés correspondants aux tâches à effectuer.

Il faudra faire des compromis selon les besoins dégagés et les caractéristiques communes des outils, car il n'existe pas de meilleure technique de fouille, à tout jeu de données et à tout problème correspond donc une ou plusieurs méthodes, le choix se fera en fonction de la tâche à résoudre, de la nature des données ou encore de l'environnement de l'entreprise. Il est souhaitable de mettre en œuvre différentes techniques afin de les comparer et d'en retenir une ou plusieurs combinées.

### **II.1.4 L'évaluation et la présentation :**

La fouille de données pourrait extraire des connaissances utiles et intéressantes, il pourrait aussi générer des résultats triviaux voire insignifiants. C'est pourquoi une phase post-fouille est nécessaire.

Cette phase, dite d'évaluation ou de validation, a pour objectif de mesurer l'intérêt des modèles extraits. Deux approches sont communément utilisées dans la validation. Ces approches dépendent de la nature de la tâche à mener et du problème considéré. C'est à l'expert (d'un domaine donné) de vérifier le modèle réalisé.

## **III- Les domaines d'application :**

### **III.1- Secteur bancaire :**

Se concentre sur les aspects comportementaux des clients avec l'analyse des transactions et des données externes de géolocalisation de ces clients. Les banques sont amenées à déployer des services plus significatifs qui répondent mieux aux besoins des clients et cela en optimisant le temps des traitements, détecter et caractériser les fraudes et améliorer la relation client.

### **III.2- Secteur des assurances :**

Les compagnies d'assurance perdent beaucoup d'argent chaque année à cause des demandes d'indemnisation frauduleuses, le data mining notamment l'analyse prédictive, constituent des outils puissances pour la détection et l'anticipation des fraudes.

### III.3- Secteur médical :

Le data mining est utilisé aussi dans le domaine de la santé, exemple, déterminer des segments des patients susceptibles d'être soumis à des thérapies déterminés, chaque segment regroupant tous les patients réagissant identiquement.

Mettre en évidence des facteurs de risque ou de rémission dans certains médicaments, choisir le traitement approprié.

### III.4- Le commerce :

C'est un secteur où le data mining est très répandu, il permet de connaître l'emplacement géographique des clients pour le géomarketing<sup>(1)</sup>, segmenter les clients sur tout pour faire des promotions, assurer la fidélité des clients.

### III.5- Secteurs divers :

Dans le secteur industriel, contrôle de qualité en recherchant des facteurs expliquant les défauts de production, prévision du trafic routier pour rechercher des causes des accidents.

## IV- Data mining sur quels types de données :

Le data mining n'est pas spécifique à un type de médias ou de données, il est applicable à n'importe quelle type d'information. Le data mining est utilisé et étudié dans les bases de données relationnelles et les bases, données orientés objet, les data Warehouse (**Voir Annexe**), les bases de données transactionnelles, les bases de données avancées comme les bases de données spatiales, les bases de données multimédia, les bases de données de série temporelle, les bases de données textuelles et même les fichiers plats.

### IV.1- Les fichiers plats :

Les fichiers plats sont actuellement la source de données la plus commune pour les algorithmes de data mining, les fichiers plats sont des fichiers de données simples dans le format texte ou binaire connue par l'algorithme de data mining qui va être appliqué.

### IV.2- Les bases de données relationnelles :

Les algorithmes de data mining appliqués aux bases de données relationnelles sont plus polyvalents que les algorithmes spécifiques fait pour les fichiers plats, puisqu'ils peuvent profiter de la structure inhérente aux bases de données relationnelles. Le Data Mining peut

---

(1) : Le géomarketing est la branche du marketing qui consiste à analyser le comportement des individus économiques en tenant compte des notions d'espaces.

profiter du SQL pour la sélection, la transformation et la consolidation, il passe au-delà de ce que le SQL pourrait fournir, comme la prévision, la comparaison.

#### **IV.3- Les bases de données transactionnelles :**

En général, une base de données transactionnelle est un fichier où chaque ligne représente une transaction. Une transaction contient un identifiant unique de transaction et une liste d'items composant la transaction (les achats d'un client lors d'une visite). Les bases de données transactionnelles peuvent contenir d'autres informations telles que la date de la transaction, l'identifiant du consommateur, l'identifiant de la personne qui a vendu, etc....

#### **IV.4- Les bases de données multimédia :**

Les bases de données multimédia comportent des documents sonores, des vidéos, des images et des médias en textes et audio. Le multimédia est caractérisé par sa haute dimension ce qui rend le datamining sur ce type de données très difficile. Le data mining sur les supports des multimédias requiert exige la vision par ordinateur, l'infographie, l'interprétation des images.

#### **IV.5- La base de données spatiale :**

Ce sont des bases de données qui contiennent des informations géographiques comme les cartes et les positionnements mondiaux ou régionaux. De telles bases de données présentent de nouveaux défis aux algorithmes de data mining.

#### **IV.6- Les bases de données de série temporelle :**

Les bases de données de séries temporelles contiennent des données relatives au temps, comme les données du marché boursier ou les activités enregistrées. Ces bases de données ont couramment un flux continu de nouvelles données entrantes, qui oblige parfois, l'analyse en temps réel. Le data mining pour ce genre de bases de données est généralement l'étude des tendances et des corrélations entre les évolutions des différentes variables, aussi bien que la prédiction des tendances et des mouvements des variables par rapport au temps.

## **V- Tâches de data mining :**

Afin de pouvoir extraire des données nécessaires pour l'ECD, différentes méthodes sont proposées par le processus data mining citons :

### **V.1- La classification :**

La classification consiste à examiner des caractéristiques d'un élément nouvellement présenté, afin de l'affecter à une classe d'un ensemble prédéfini. Est donc l'opération qui permet de placer chaque élément dans une classe, parmi plusieurs classes prédéfinies.

### **V.2- L'estimation :**

Elle consiste à estimer la valeur d'un champ à partir des caractéristiques d'un objet. Le champ à estimer est un champ à valeurs continues. L'estimation peut être utilisée dans un but de classification. Il suffit d'attribuer une classe particulière pour un intervalle de valeurs du champ estimé.

### **V.3- La prédiction :**

Cela consiste à estimer une valeur future. En général, les valeurs connues sont historisées. On cherche à prédire la valeur future d'un champ. Cette tâche est proche des précédentes. Les méthodes de classification et d'estimation peuvent être utilisées en prédiction.

### **V.4- Les règles d'association :**

Cette tâche, plus connue comme l'analyse du panier de la ménagère, consiste à déterminer les variables qui sont associées. L'exemple type est la détermination des articles (le pain et le lait, la tomate, les carottes et les oignons) qui se retrouvent ensemble sur un même ticket de supermarché. Cette tâche peut être effectuée pour identifier des opportunités de vente croisées et concevoir des groupements attractifs de produits.

### **V.5- La segmentation :**

Consiste à former des groupes (clusters) homogènes à l'intérieur d'une population. Pour cette tâche, il n'y a pas de classe à expliquer ou de valeur à prédire définie à priori, il s'agit de créer des groupes homogènes dans la population (l'ensemble des enregistrements). Il appartient ensuite à un expert du domaine de déterminer l'intérêt et la signification des groupes ainsi constitués.

Cette tâche est souvent effectuée avant les précédentes pour construire des groupes sur lesquels on applique des tâches de classification ou d'estimation.

## **V.6- La description :**

C'est souvent l'une des premières tâches demandée à un outil de Data Mining. On lui demande de décrire les données d'une base complexe. Cela engendre souvent une exploitation supplémentaire en vue de fournir des explications. La technique la plus appropriée à la description est l'analyse du panier de la ménagère.

## **VI- Les techniques de Data Mining :**

Les techniques de Data Mining représentent une partie très importante dans la tâche de ce dernier, Nous allons citer quelques une afin de donner une description générale sans entrer dans les détails. [4]

### **VI.1- Analyse du panier de la ménagère :**

L'analyse du panier de la ménagère est un moyen de trouver les groupes d'articles qui vont ensemble lors d'une transaction. C'est une technique de découverte de connaissances non dirigée (de type analyse de clusters). Les règles générées sont simples, faciles à comprendre et assorties d'une probabilité, ce qui en fait un outil agréable et directement exploitable par l'utilisateur métier. [5]

**Exemple :** Le client qui achète de la peinture achète un pinceau, le client qui achète un téléviseur achète un magnétoscope sous 5 ans.

### **VI.2- Le raisonnement basé sur la mémoire :**

Le raisonnement basé sur la mémoire (RBM), est une technique de prédiction et de classification utilisée dans le cadre de la découverte de connaissances dirigée. Elle peut être également utilisée pour l'estimation. Pour chaque nouvelle instance présentée, le système recherche le(s) voisin(s) le(s) plus proche(s) et procède ainsi à l'affectation ou estimation. L'avantage du RBM est qu'il est facile à mettre en œuvre, très stable et supporte tout type de données. [6]

### **VI.3- La détection automatique de clusters :**

La détection automatique de clusters est une technique de découverte de connaissances non dirigée (ou apprentissage sans supervision). Elle consiste à regrouper les enregistrements en fonction de leurs similitudes. Chaque groupe représente un cluster. C'est une excellente technique pour démarrer un projet d'analyse ou de data mining. Les groupes de similitudes permettront de mieux comprendre les données et d'imaginer comment les utiliser au mieux.

#### **VI.4- L'analyse des liens :**

L'analyse des liens est une technique de description qui s'inspire et repose sur la théorie des graphes. Elle consiste à relier des entités entre elles (clients, entreprises, ...) par des liens. A chaque lien est affecté un poids, défini par l'analyse, qui quantifie la force de cette relation.

Cette technique peut être utilisée pour la prédiction ou la classification mais généralement une simple observation du graphe permet de mener à bien l'analyse.

#### **VI.5- Les arbres de décision :**

Les arbres de décision sont utilisés dans le cadre de la découverte de connaissances dirigée. Ce sont des outils très puissants principalement utilisés pour la classification ou l'estimation. Le principe de fonctionnement est le suivant : pour expliquer une variable, le système recherche le critère le plus déterminant et découpe la population en sous populations possédant la même entité de ce critère. Chaque sous population est ensuite analysée comme la population initiale. Le modèle rendu est facile à comprendre et les règles trouvées sont très explicites.

#### **VI.6- Les réseaux de neurones :**

Les réseaux de neurones représentent la technique de Data Mining la plus utilisée. C'est une transposition simplifiée des neurones du cerveau humain. Dans leur variante la plus courante, les réseaux de neurones apprennent sur une population d'origine puis sont capables d'exprimer des résultats sur des données inconnues. Ils sont utilisés dans la prédiction et la classification dans le cadre de découverte de connaissances dirigée. Cependant, on leur reproche souvent d'être une "boîte noire" : il est difficile de savoir comment les résultats sont produits, ce qui rend les explications délicates, même si les résultats sont bons.

#### **VI.7- Les algorithmes génétiques :**

Les algorithmes génétiques sont utilisés dans la découverte de connaissances dirigée. Ils permettent de résoudre des problèmes divers, notamment d'optimisation, d'affectation ou de prédiction. Leur fonctionnement s'apparente à celui du génome humain. Le principe de fonctionnement est le suivant : les données sont converties en chaînes binaires (comme les chaînes d'ADN - acide désoxyribo nucléique). Celles-ci se combinent par sélection, croisement ou mutation et donnent ainsi une nouvelle chaîne qui est évaluée. En fonction du résultat, les chaînes les plus faibles cèdent leur place aux plus fortes. Cette technique est

particulièrement intéressante pour résoudre des problèmes d'affectation ou des problèmes sur lesquels on peut poser une fonction d'évaluation car elle peut trouver des solutions optimisées parfois inexistantes dans les données d'origine.

#### **VI.8- Les agents intelligents ou knowbot :**

Les agents intelligents ou Knowbot sont des entités logicielles autonomes dont les plus récentes versions s'intègrent tout à fait dans le processus de data mining. Certains d'entre eux, les plus élaborés, sont capables de suivre et mémoriser les mouvements, visites et achats sur Internet et permettent d'élaborer des profils d'utilisateurs pour leur faire des offres commerciales "un à un (one to one)". [4]

#### **Conclusion :**

Dans cette première partie nous avons présenté le data mining, sa définition, ces sources de données, les techniques et algorithmes utilisés pour l'extraction d'informations. Nous avons donné quelques domaines d'utilisation de data mining.

Ces données sont utilisées par des systèmes dits d'apprentissage et aussi de classification pour prédire des valeurs et grouper les objets en classes.

Dans notre travail, nous nous intéressons à la classification automatique, et cela fera l'objet de la deuxième partie.

**Deuxième chapitre :**

# **L'apprentissage et la classification**

## I- Introduction

Comme on l'a déjà vu dans la première partie, d'énormes quantités de données sont récupérées et stockées dans des bases de données dans divers domaines. Ces données, après avoir été traitées par les techniques de data mining, l'apprentissage automatique (artificiel) et le système de classification, permettent de concevoir des algorithmes de prédiction. Ces modèles seront capables de prédire la nature (forme, apparence, texture, couleur, ...) et le comportement (fonction, fonctionnement, variables, variations,...).

## II- Apprentissage automatique :

L'apprentissage automatique concerne la conception et l'analyse, le développement et l'implémentation des méthodes permettant à une machine (au sens large) d'évaluer par un processus systématique ainsi que de remplir des tâches difficiles ou problématique par des moyens algorithmiques plus classiques, d'autre terme l'apprentissage c'est le processus de construire un modèle général à partir de données (observations) particulières du monde réel. Ainsi l'apprentissage automatique a pour objectif d'offrir des techniques pour développer une intelligence artificielle. Existe plusieurs types d'apprentissage, et parmi eux l'apprentissage supervisé et l'apprentissage non-supervisé.

### ➤ Exemple :

- La reconnaissance des caractères manuscrite
- Prédire un comportement face à une nouvelle donnée.

### II.1 Apprentissage Supervisé :

Les techniques d'apprentissage supervisé permettent de construire des modèles à partir d'exemples d'apprentissage ou d'entraînement dont on connaît le comportement ou la réponse. Ces modèles peuvent, ensuite, être utilisés dans différentes applications, telles que la prédiction ou la classification.

### II.2 Apprentissage Non-Supervisé :

Il vise à concevoir un modèle structurant l'information. La différence est que les comportements (ou catégories ou encore les classes) des données d'apprentissage ne sont pas connus, c'est ce que l'on cherche à trouver.

### **III- La classification :**

#### **III.1- Introduction :**

Les approches de l'apprentissage automatique sont développées, nous allons citer quelques-unes des approches utilisées dans l'apprentissage automatique.

La classification dans le domaine informatique est appliquée et utilisée sur les données numérique telle que (points, tableaux, images etc.) dans cette partie nous allons dans ce principe et nous représentons tout d'abord ce que c'est la classification, ses méthodes et techniques ainsi que le domaine d'application etc...

#### **III.2- Définition :**

La classification est une discipline utilisée dans nombreux domaine, cette dernière est souvent connue sur des termes varie (classification, segmentation, clustering) et pour attribuer une définition au terme classification il faudrait d'abord définir ses racines, ça vient de verbe « classer » qui désigne plus d'une action qu'un domaine.

En mathématique on appelle la classification la catégorisation d'objet, elle consiste à attribuer une classe à chaque objet ou individu à classer en se basant sur les données d'apprentissage, elle fait couramment appelle aux méthodes d'apprentissage.

#### **III.3- Les étapes d'une classification :**

- Choix des données.
- Calcul la similarité entre  $n$  individus à partir des données initiales.
- Choix d'un algorithme de classification et l'exécution.
- L'interprétation du résultat.
  - évaluation de la qualité de la classification
  - description des classes obtenues

#### **III.4- Mesure de la similarité :**

Pour comparer l'homogénéité ou la ressemblance, la similarité entre deux objets (points, images, classes, phonème...), il faut pouvoir mesurer la similarité (ou la dissimilarité) entre eux.

Tout système ayant pour but d'analyser ou d'organiser automatiquement un ensemble de données ou de connaissances doit utiliser, sous une forme ou une autre, un opérateur de

similarité dont le but est d'établir les ressemblances ou les relations qui existent entre les informations manipulées.

Il est à noter qu'il y a deux concepts pour exprimer la notion de proximité entre les objets à classer.

### III.4.1 :- Fonctions de similarité

- **La distance euclidienne :** (aussi appelée la distance à vol d'oiseau). Un rapport de clusters analyses en psychologie de la santé a conclu que la mesure de la distance la plus courante, dans les études publiées dans ce domaine de recherche est la distance euclidienne ou la distance au carré euclidienne.

$$d^2(x_1, x_2) = \sum_i (x_{1i} - x_{2i})^2 = (x_1 - x_2)(x_1 - x_2)'$$

- **La distance de Manhattan :** (appelée aussi taxi-distance)

$$d^2(x_1, x_2) = \sum_i |x_{1i} - x_{2i}|$$

- **La distance de Mahalanobis :** Corrige les données pour les différentes échelles et des corrélations dans les variables, l'angle entre deux vecteurs peut être utilisé comme mesure de distance quand le regroupement des données est de hautes dimensions. Voir l'espace produit scalaire.

$$d^2(x_1, x_2) = (x_1 - x_2)' C^{-1} (x_1 - x_2)$$

(C: Covariance)

- **La distance de Sebestyen :**

$$d^2(x_1, x_2) = (x_1 - x_2)' W (x_1 - x_2)$$

(W : Matrice diagonale de pondération)

- **La distance de Hamming :** Mesure le nombre minimum de substitutions nécessaires pour changer un membre dans un autre. Elle permet ainsi, de quantifier la différence entre deux séquences de symboles, généralement utilisée dans le cas des valeurs discrètes (vecteurs).

$$d(a, b) = \sum_{i=0}^{n-1} (a_i \neq b_i)$$

**Exemple :** Considérons les suites binaires suivantes :

$$a = (0001111), b = (1101011) \text{ alors } d = 1+1+0+0+1+0+0$$

La distance entre  $a$  et  $b$  est 3, car 3 bits différents.

$$X^2(x, y) = \sum_{i=0}^n x_i^2 = \frac{(x_i - y_i)^2}{x_i + y_i}$$

Cependant que pour les données de grandes dimensions, il y a une distance spécifique.

- **La métrique Minkowski :** Pour les données dimensionnelles, c'est la mesure populaire.

$$d_p(x_i, x_j) = \left( \sum_{k=1}^d |x_{ik} - x_{jk}|^p \right)^{\frac{1}{p}}$$

Où  $d$  est la dimensionnalité des données. La distance euclidienne est un cas particulier où  $p = 2$ , alors que Manhattan  $p = 1$ .

Néanmoins, il n'existe pas de directives générales théoriques pour la sélection d'une mesure à une application donnée. Une autre question, est de savoir comment mesurer la distance entre 2 classes  $D$  ( $C1, C2$ ), pour cela il y a certaines fonctions qui permettent de mesurer cette distance comme :

- **Diamètre maximum :**

$$\text{Max}(d(i, j), i \in C_1, j \in C_2)$$

- **Distance moyenne :**

$$\sum_{i,j} \frac{d(i, j)}{n_1 n_2}$$

- **Distance des centres de gravité :**

$$d(u_1, u_2)$$

### III.5- Les méthodes de la classification :

Il existe différents types de méthodes de classification qui s'appuient sur les méthodes supervisées (classement) et non supervisées (clustering).

### III.5.1- Méthodes non supervisé :

#### III.5.1.1- Définition :

Le clustering aussi connu sous nom (segmentation) est un regroupement en classes homogènes qui consiste à regrouper un nuage de points d'un espace quelconque en un ensemble de groupe appelés cluster d'autres termes le clustering consiste à construire une collection d'objets similaire au sein d'un même groupe et dissimilaire quand il appartient aux groupes différents.

#### III.5.1.2- Principe :

Contrairement à la classification (méthodes supervisées), on ne possède pas des connaissances a priori sur les classes prédéfinies des éléments. La division des objets dans les différents groupes (clusters) est réalisée en se basant sur le calcul de similarité entre les éléments. L'objectif des méthodes du clustering est de grouper des éléments proches dans un même groupe de manière à ce que deux données d'un même groupe soient le plus similaires possible et que deux éléments de deux groupes différents soient le plus dissemblables possible. [7]

Mathématiquement, on a un ensemble  $X$  de  $N$  données décrites chacune par leurs  $P$  attributs. Donc le clustering consiste à créer une partition ou une décomposition de cet ensemble en sous parties (clusters) telle que :

- Les données appartenant au même groupe se ressemblent,
- Les données appartenant à deux groupes différents soient peu ressemblantes.

**Il existe plusieurs types de clustering (Voir Annexe).**

### III.5.2- Méthodes supervisée (Classification) :

#### III.5.2.1- Définition :

La classification est une méthode supervisée qui consiste à définir une fonction qui permet de classer les objets dans des classes à partir des données quantitatives et qualitatives caractérisant ces objets.

#### III.5.2.2- Principe :

Le principe général de ce classifieur consiste à classifier un ensemble de données constitué de  $n$  objet, sachant que ces  $n$  objets sont « étiquetés ». Ce système de classification va être conçue en basant sur les exemples de superviseure (l'ensemble d'apprentissage ou pour tout exemple on connaît a priori sa classe), c'est-à-dire on cherche à prédire si un élément «  $x_i$  » de la base de données, décrit par un ensemble de descripteur « Appartient ou non à une classe  $c_j$  » parmi  $N$  classes et pour le faire, on a un ensemble d'apprentissage décrit par :

$$A = (x_1, c_1), (x_2, c_2), (x_3, c_3) \dots (x_i, c_j) \quad x_i \in R, c_j \in C$$

Donc pour chaque objet  $x_i$  de l'ensemble de données, on peut connaître sa classe a priori  $c_j$ . La classification supervisée tente de chercher, à partir des données de  $A$ , une fonction de décision  $f$  qui va associer à tout nouveau élément  $x_i$  de test une classe  $c_j$ , puis on compare ce que nous a donné cette fonction avec la classe connue a priori de cet élément, de sorte à minimiser les mauvais classements ( $f(x_i) \neq c_j$ ).

Donc l'objectif est de chercher à prédire la classe de chaque nouvelle donnée en utilisant les données de la base d'apprentissage.

### III.5.2.3 Quelques algorithmes de la classification supervisée :

#### III.5.2.3.1 Arbre de décision :

- Définition :

Les arbres de décisions permettent de classifier une population d'individus selon les valeurs de leurs attributs. C'est une représentation graphique de la procédure de classification où : Une feuille indique une classe, Un nœud spécifie un test que doit subir un certain attribut, Chaque branche correspond à une valeur possible de l'attribut.

Pour classifier un nouvel objet, on suit le chemin partant de la racine à une feuille en effectuant les différents tests d'attributs à chaque nœud.

### III.5.2.3.2 Structure d'un arbre de décision :

Chaque nœud interne d'un arbre de décision permet de répartir les éléments à classifier de façon homogène entre ses différents fils en portant sur une variable discriminante de ces éléments.

Les branches qui représentent les liaisons entre un nœud et ses fils sont les valeurs discriminantes de la variable du nœud. Et enfin, les feuilles sont les valeurs qui représentent les résultats de la prédiction des données à classifier.

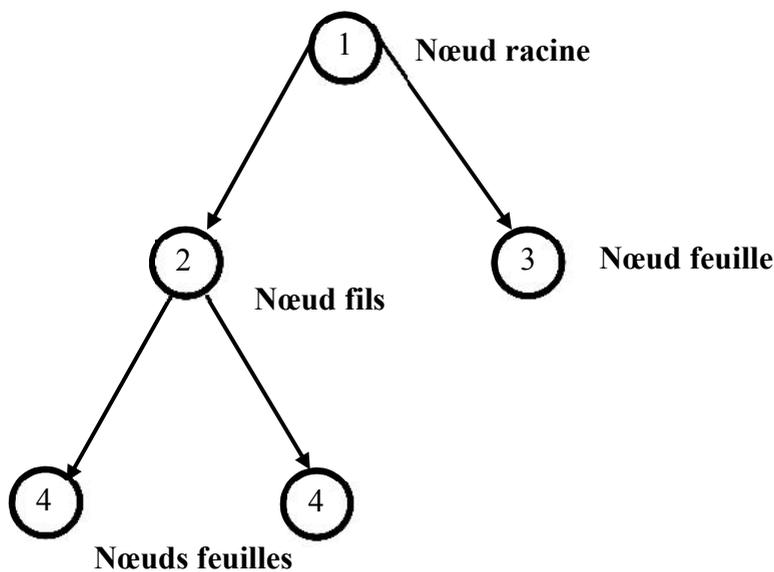


Figure 2 : Structure d'un arbre de décision.

Exemple :

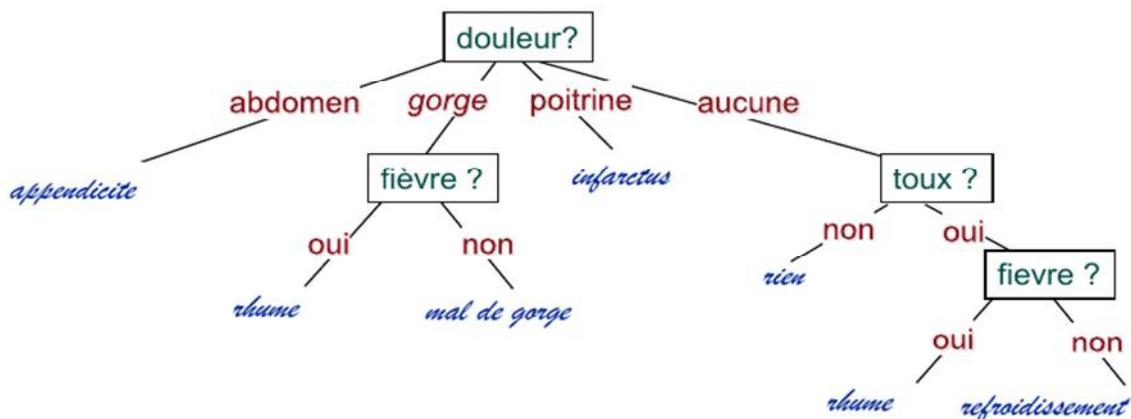


Figure 3 : exemple d'un arbre de décision

### III.5.2.3.3 Algorithme de construction d'un arbre de décision :

Existe plusieurs algorithmes pour construire des arbres de décision, citons :

- **CART (Leo Breiman 1984) :**

Cette méthode permet d'inférer des arbres de décision binaires (un nœud ne peut avoir que 2 fils), donc, tous les tests étiquetant les nœuds de décision sont binaires. Le langage de représentation est constitué d'un certain nombre d'attributs. Ces attributs peuvent être binaires, qualitatifs (à valeurs dans un ensemble fini de modalités) ou continus (à valeurs réelles). Le nombre de tests à explorer va dépendre de la nature des attributs. A un attribut binaire correspond un test binaire. A un attribut qualitatif ayant  $n$  modalités, on peut associer autant de tests qu'il y a de partitions en deux classes, soit  $2^{n-1}$  tests binaires possibles. Enfin, dans le cas d'attributs continus, il y a une infinité de tests envisageables. Dans ce cas, on découpe l'ensemble des valeurs possibles en segments, ce découpage peut être fait par un expert ou fait de façon automatique.

- **C 5.0 (Ross Quinlan 1998) :**

Cet algorithme fonctionne en cherchant à maximiser le gain d'information réalisé en affectant chaque individu à une branche de l'arbre. *C 5.0* présente la particularité de pouvoir, à chaque étape, séparer la population en plus de deux sous-populations : il n'est pas binaire. Cela vient de son traitement des variables catégoriques qui, au niveau d'un nœud-père, donnent naissance à un nœud-fils par modalité. [1]

- **CHAID (Chi-squared Automatic Interaction detector) : (ID3)**

C'est l'une des méthodes d'arbre de classification les plus anciennes. Cette méthode permet de construire des arbres de décisions non-binaires (des arbres de décisions dans lesquels nous pouvons avoir plus de deux branches connectées à un même nœud (nœud racine)), en utilisant un algorithme assez simple qui est particulièrement bien adapté à l'analyse des fichiers de données les plus volumineux. (...).

Il est utile lorsque on recherche des modèles dans des ensembles de données avec beaucoup de variables catégorielles et constitue un moyen pratique de récapituler les données car les relations peuvent être facilement visualisées.

### III.5.2.3.4 Les k plus proches voisins (K-PPV) :

L'algorithme des K-Nearest Neighbors (KNN) (K plus proches voisins) est un algorithme de classification supervisé. Chaque observation de l'ensemble d'apprentissage est représentée par un point dans un espace à n dimensions ou n est le nombre de variables prédictives. Pour prédire la classe d'une observation, on cherche les  $k$  points les plus proches de cet exemple. La classe de la variable cible, est celle qui est la plus représentée parmi les  $k$  plus proches voisins. Il existe des variantes de l'algorithme ou on pondère les  $k$  observations en fonction de leur distance à l'exemple dont on veut classer, les observations les plus éloignées de notre exemple seront considérées comme moins importantes.

#### a) L'algorithme :

*Soient  $\{(X_1, q_1), (X_2, q_2), \dots, (X_n, q_n)\}$  données, et  $\{Y, ?\}$  observé*

*{     initialiser les k plus proches voisins avec une distance saturée;*

*Pour (chaque élément  $X_i$  de l'échantillon*

*attention !!     {     Calculer la distance  $d(X_i, Y)$ ;*

*Insérer (si nécessaire)  $X_i$  dans les k plus proches voisins;*

*}*

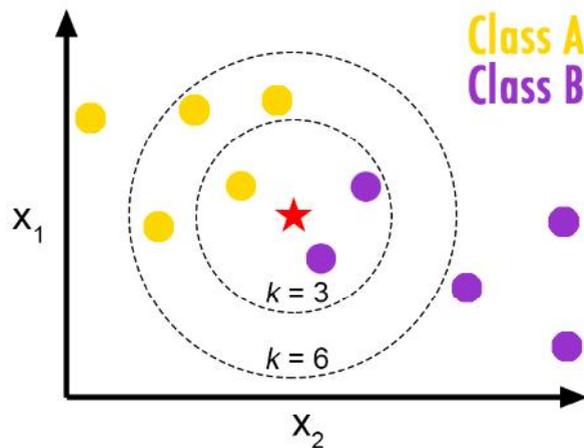
*Déterminer la classe C la plus représentée dans les k plus proches voisins;*

*Si C est suffisamment représentée et si  $d(Y, \ll C \gg) < d_{max}$*

*Alors le résultat est acceptable*

*Sinon Y n'est pas classable.*

*}*



**Figure 4 – Pour  $k = 3$  la classe majoritaire du point central est la classe B, mais si on change la valeur du voisinage  $k = 6$  la classe majoritaire devient la classe A**

### III.5.2.3.5 La classification bayésienne :

- **Définition :**

Un classifieur probabiliste linéaire simple basée sur le théorème de Bayes qui suppose que les descripteurs (attributs) qui décrivent les objets de l'ensemble d'apprentissage sont indépendants.

- **Principe :**

La classification bayésienne a pour objectif de chercher à classer un nouveau objet «  $X_{new}$  » non étiqueté en se basant sur l'ensemble d'apprentissage «  $A$  » qui a pour chaque objet sa classe à priori (objet étiqueté), donc le classifieur bayésien va choisir la classe majoritaire qui a la plus grande probabilité «  $C_k$  » parmi les classes définies dans l'ensemble d'apprentissage, on parle de règle MAP (maximum a posteriori).

-Premièrement il faut estimer les probabilités  $P(C_k)$  et  $p(X_{new}/C_k)$  à partir de l'ensemble d'apprentissage «  $A$  », et pour estimer les probabilités a priori des classes  $P(C_k)$  on se base sur la règle suivante :

$$P(C_k) = \frac{\text{Nombre d'expression d'apprentissage dans la classe } C_k}{\text{Le nombre totale de documents dans l'ensemble d'apprentissage}}$$

-ensuite il faut estimer les valeurs de  $P(X_{new}/C_k)$ , puisque les descripteurs (attributs) de «  $X_{new}$  » sont indépendants, alors on aura grâce aux théories d'indépendance bayésienne entre les variables.

$$P(X_{\text{new}}|C_k) = P(f_1|C_k)P(f_2|C_k)\dots p(f_n|C_k)$$

Où les «  $f_i$  » sont les attributs qui décrivent l'ensemble des données, sachant que

$$P(C, f_1, \dots, f_n) = P(C) \prod P(f_i|C)$$

Et pour estimer les paramètres d'une loi de probabilité relative à une caractéristique précise, il est nécessaire de présupposer le type de la loi en question.

#### **III.5.2.3.6 Les réseaux de neurones :**

Ensemble des fonctions mathématiques interconnectés permettant la résolution de problèmes complexes, tels que la reconnaissance des formes ou le traitement du langage naturel, grâce à l'ajustement des coefficients de pondération dans une phase d'apprentissage. Les réseaux de neurones seront détaillés dans le prochain chapitre.

## **Conclusion :**

Dans cette deuxième partie nous avons vu l'apprentissage automatique et la classification, nous avons donné quelques algorithmes des méthodes supervisées et non supervisées les plus utilisées, nous avons donné une petite définition des réseaux de neurones, cette approche qui fera l'objet de la troisième partie.

**Troisième chapitre :**

# Les réseaux de neurones convolutifs

## I- Introduction :

Les réseaux de neurones artificiels, est l'un des meilleurs outils utilisés aujourd'hui dans le domaine de l'apprentissage et la classification automatiques. Dans cette partie, nous allons présenter les réseaux de neurones convolutionnels et comment faire la classification d'images avec ces réseaux. Avant de voir les réseaux convolutionnels, nous allons citer les différents réseaux de neurones artificielles

## II- Les réseaux de neurones artificiels (RNA) :

Les réseaux de neurones artificielles sont des réseaux fortement connectés des processeurs élémentaire fonctionne en parallèle (neurones), chaque neurone artificielle est un processeur élémentaire, il reçoit un nombre de variable d'entrées en provenance de neurones amonts, à chacune de ces entrées est associées un poids  $W$  représentatifs de la force de connexion, chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit, qui se ramifie ensuite pour alimenter un nombre variable de neurones aval. A chaque connexion est associé un poids comme montre la figure 5. [8]

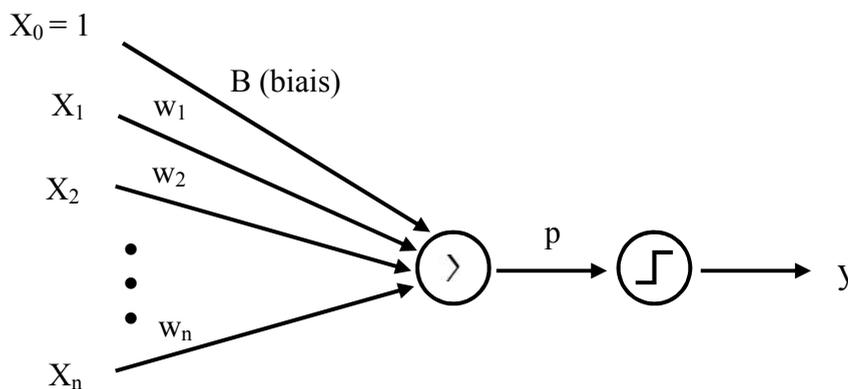


Figure 5 : La structure d'un neurone artificiel.

La figure 6 donne les notations que nous utilisons dans cette partie

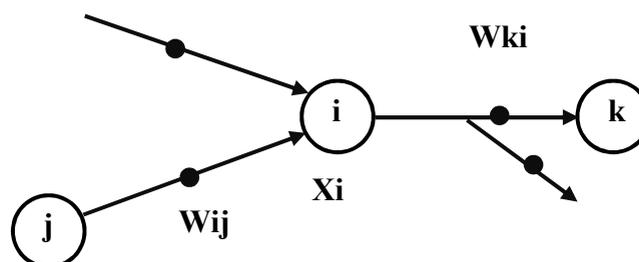


Figure 6 : structure d'un neurone artificiel pour le neurone amont d'indice  $i$ , les entrées de celui-ci sont de poids  $w_{ij}$  alors que les connexions aval sont de poids  $w_{ki}$ .

- **Les poids et connexion :**

Une connexion entre deux neurones  $i$  et  $j$  indique la possibilité d'une relation physique entre deux neurones et la valeur numérique associée à cette connexion appelée poids de connexion qui peut prendre des valeurs discrètes dans  $Z$  ou bien continue dans  $R$ , l'information qui traverse la connexion sera affectée par la valeur du poids correspondante,

Une connexion avec un poids  $W_{ij} = 0$  est équivalente à l'absence de connexion.

- **Le voisinage**

Le voisinage d'un neurone est l'ensemble des neurones connectés à ce neurone. Le voisinage d'ordre  $n$  d'un neurone  $i$ , les  $n$  neurones connectés à ce dernier et les connexions entre les neurones ont un sens.

## II.1- Comportement d'un réseau de neurones artificiel :

Le réseau de neurone artificiel fonctionne en deux phases :

### II.1.1- La somme pondérée :

La première phase consiste à calculer la somme pondérée des entrées ( $a$ ) selon l'expression suivante :

$$p = \sum (x_i \cdot w_i)$$

### II.1.2- La fonction d'activation :

La fonction d'activation (ou fonction de transfert) sert à convertir le résultat de la somme pondérée des entrées d'un neurone en une valeur de sortie.

Le biais  $b$  joue un rôle de seuil, quand le résultat de la somme pondérée dépasse ce seuil, l'argument de la fonction de transfert devient positif ou nul, dans le cas contraire il est considéré négatif, finalement si le résultat de la somme pondérée est :

1 en dessous du seuil : le neurone est considéré comme non-actif

2 aux alentours du seuil : le neurone est considéré en phase de transition

3 au-dessus du seuil : le neurone est considéré comme actif

Il y a plusieurs types de fonctions de transfert qui peuvent être utilisés dans les RNA, les fonctions d'activation souvent utilisées sont représentées ci-dessous : [9]

- **La fonction linéaire :**

La fonction est définie :

$$y = \Phi(p) = p$$

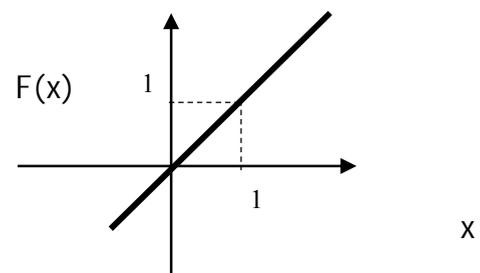
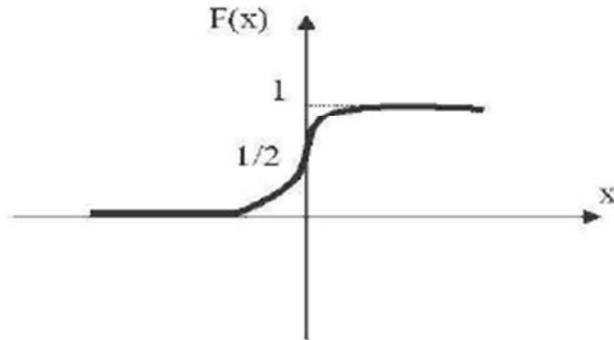


Figure 7 : fonction linéaire

- **La fonction non linéaire sigmoïde :**

La fonction sigmoïde est définie par :

$$F(x) = \frac{1}{1+e^{-x}}$$



**Figure 8 : Fonction sigmoïde.**

- **La fonction à seuil**

On peut citer comme exemple la fonction Heaviside et la fonction Signe qui sont définies respectivement comme suit :

- **Fonction Heaviside :**

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

- **Fonction Signe :**

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

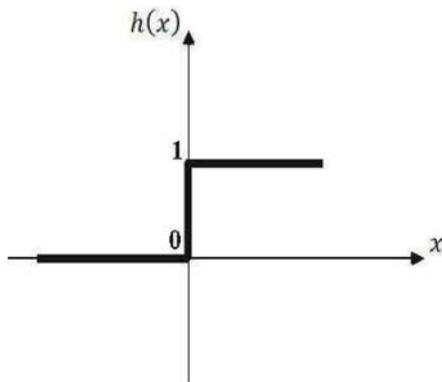


Figure 9 : Fonction Heaviside

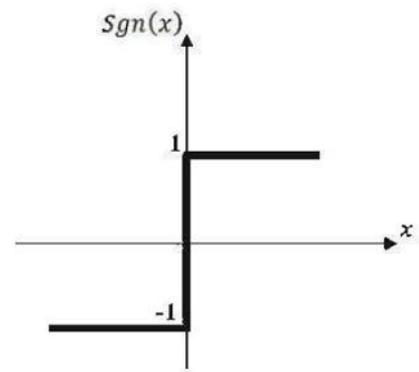


Figure 10 : Fonction Signe

### La fonction linéaire à seuil

On peut la définir comme suit :

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

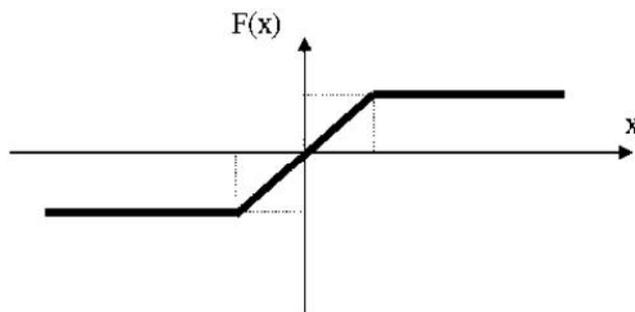


Figure 11 : Fonction linéaire à seuil.

- **La fonction SoftMax :**

Cette fonction calcule la probabilité qu'une image appartient à une classe déjà étiquetée.

L'avantage d'utiliser la fonction softmax est que la gamme sera entre 0 et 1, et la somme de toutes les probabilités sera égale à 1. Si la fonction softmax utilisée pour le modèle multi-classification renvoie les probabilités de chaque classe.

$$F(x_i) = \frac{\text{Exp}(x_i) \quad i = 0, 1, \dots, k}{\sum_{j=0}^k \text{Exp}(x_j)}$$

Avec :  $x_i$  le vecteur d'entrée de l'image à classifier.

$x_j$  : vecteur de classe  $j$ .

## II.2- Classification des réseaux de neurones artificiels :

La classification d'un réseau de neurone artificiel se fait à partir de sa topologie de connexion, nous distinguons deux grandes catégories de ces derniers, un réseau de neurone non bouclé (statique ou feed forward) et un réseau de neurone bouclé (dynamique, feed back ou récurrents) comme il est illustré dans la figure suivante :

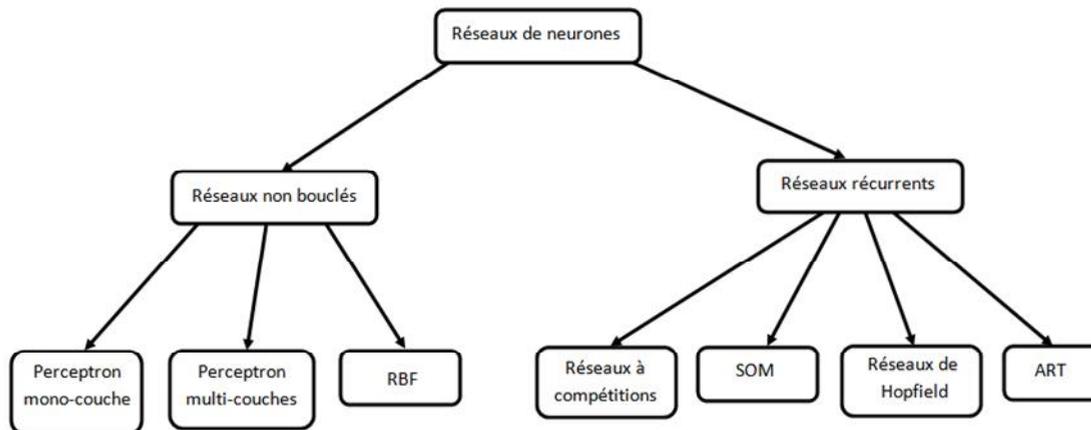


Figure 12 : Classification des réseaux de neurones.

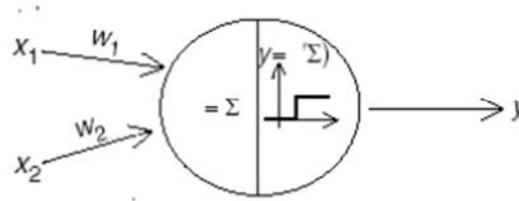
### II.2.1- Les réseaux non bouclé (statique ou feed forward) :

Dans les réseaux statiques la contrainte de temps n'est pas un paramètre significatif contrairement aux réseaux récurrents, en d'autre terme la modification d'entrée n'entraîne que la modification stable de la sortie, mais elle n'entraîne pas le retour de l'information de cette entrée.

Les réseaux statiques (FEED FORWARD) sont des réseaux à couches, généralement sont constitués de plusieurs couches (couche d'entrée et la couche de sortie et entre de ces deux couche on trouve aux moins une couche intermédiaire composée de nombreux éléments de traitements non linéaire appelée couche caché). Les données d'entrées se propagent de la premier couche vers la couche de sortie en passant par la ou les couches cachés, ainsi qu'il n'y pas de communication entre les unités de la même couche, et les liens dirigés connectant ces couches appelés les interconnexions. Il existe différents types réseaux de neurones non-bouclés et parmi eux :

#### II.2.1.1- Le perceptron mono-couche :

Le perceptron mono-couche est le premier modèle des réseaux de neurones, inventé en 1957 par Frank Rosenblat, peut être vue comme le type de réseau de neurone le plus simple puisqu'il ne se compose que d'une couche d'entrée et d'une couche de sortie et ne contient aucun cycle, donc le perceptron mono-couche n'a qu'une seule sortie à laquelle toutes les entrées sont connectées. Les entrées et les sorties sont booléennes en d'autre terme le perceptron mono-couche est constitué d'un seul neurone qui agit par la fonction d'activation comme il est illustré dans la figure suivante : [17]



**Figure 13 : Perceptron simple.**

Le neurone reçoit les deux entrées multipliées par les poids  $w_1$  et  $w_2$  respectivement qui donnent la valeur d'entrée  $a$ , où  $a = w_1x_1 + w_2x_2$  puis l'activation de la fonction à seuil de l'entrée  $a$  donnée comme suit.

$$f(a) = \begin{cases} 1 & \text{si } a \geq 0 \\ 0 & \text{si } a < 0 \end{cases}$$

Dans ce cas, le seuil est 0, Si  $a = 0$  alors  $w_1x_1 + w_2x_2 = 0$  Donc

$$x_1 = \left( \frac{-w_2}{w_1} \right) x_2$$

Cette équation présente la frontière de décision. On remarque qu'elle est une frontière linéaire.

### II.2.1.2- Perceptron multicouche :

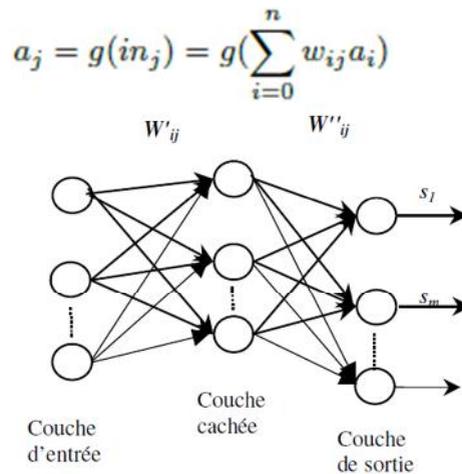
Multi Layer Perceptron (MLP) (perceptron multicouches) est un classifieur de type réseaux de neurones qui sont organisés en plusieurs couches, chaque couche étant formée d'un ou plusieurs neurones formels. Utilisé dans un cas d'apprentissage supervisé, il utilise l'algorithme de rétropropagation de gradient

Voici les caractéristiques d'un perceptron multicouche :

- Il comporte une seule couche d'entrée et une seule couche de sortie.
- il peut comporter une ou plusieurs couches cachées.
- Chaque neurone est relié uniquement à tous les neurones de la couche suivante.
- Chaque lien de la couche  $i$  vers la couche suivante  $j$  sert à propager l'activation  $a_i$  de  $i$  jusqu'à  $j$  et qui possède un poids  $w_{ij}$  qui détermine l'intensité du signal de la connexion. Chaque unité de la couche  $j$  calcul la somme pondérée de ses entrées : [17]

$$in_j = \sum_{i=0}^n w_{ij} a_i$$

Puis leur applique une fonction d'activation :



**Figure 14 : Perceptron multicouche.**

L'algorithme d'apprentissage utilisé dans un perceptron multicouche consiste à présenter au réseau des entrées et on lui demande de modifier sa pondération de façon à retrouver la sortie correspondante. D'abord on propage vers l'avant les entrées jusqu'à obtenir une sortie calculée par le réseau puis on compare cette sortie avec la sortie désirée, enfin on modifie les poids de telle sorte qu'à la prochaine itération, l'erreur commise entre les sortie réelles et les sortie désirées soit minimisée. On répète ce processus jusqu'à ce qu'on obtienne une erreur de sortie négligeable.

### II.2.1.3 Les réseaux de neurones RBF :

Les réseaux de neurones RBF (Radial Basis Function), ont une architecture feedforward puissante. Ces réseaux font partie des réseaux de neurones supervisés, ce dernier est constitué de trois couche, une couche d'entrée qui retransmet les entrées sans distorsion et une couche cachée, qui contient les neurones RBF et une couche de sortie dont les neurones. Sont généralement animés par une fonction d'activation linéaire ainsi que chaque couche est complètement connectés vers la suivant et il n'y a pas de connexion à l'intérieur d'une même couche. [12]

- **Fonctionnement :**

**a) La couche d'entrée :**

La couche d'entrée c'est un vecteur à  $n$  dimensions que on essaie de classifier. Le vecteur d'entrée est montré à chacun des neurones RBF.

**b) La couche cachée (Contient les neurones RBF) :**

Chaque neurone RBF stocke un vecteur « prototype » qui n'est que l'un des vecteurs de l'ensemble d'apprentissage. Chaque neurone RBF compare le vecteur d'entrée à son prototype et affiche une valeur entre 0 et 1 qui est une mesure de similarité. Si l'entrée est égale au prototype, alors la sortie de ce neurone RBF sera 1. Lorsque la distance entre l'entrée et le

prototype augmente, la réponse tombe exponentiellement vers 0. La forme de la réponse du neurone RBF est une courbe.

La valeur de réponse du neurone est également appelée sa valeur "activation".

Le vecteur prototype est aussi souvent appelé «centre» du neurone, puisque c'est la valeur au centre de la courbe en cloche.

### c) La couche de sortie :

La sortie du réseau consiste en un ensemble de nœuds, un par catégorie que nous essayons de classer. Chaque nœud de sortie calcule une sorte de score pour la catégorie associée. Généralement, une décision de classification est prise en attribuant l'entrée à la catégorie ayant le score le plus élevé.

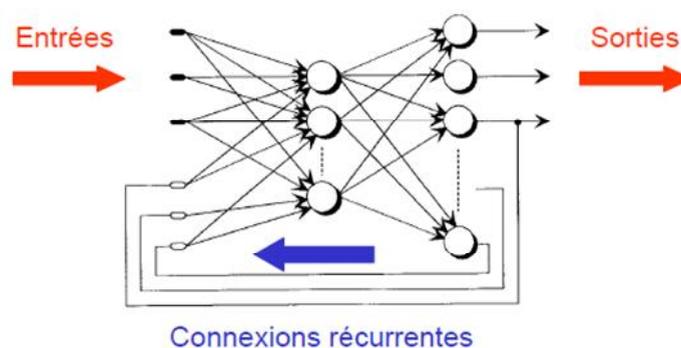
Le score est calculé en prenant une somme pondérée des valeurs d'activation de chaque neurone RBF. Par somme pondérée, nous voulons dire qu'un nœud de sortie associe une valeur de poids à chacun des neurones RBF, et multiplie l'activation du neurone par ce poids avant de l'ajouter à la réponse totale.

Comme chaque nœud de sortie calcule le score pour une catégorie différente, chaque nœud de sortie a son propre ensemble de poids. Le nœud de sortie donnera typiquement un poids positif aux neurones RBF qui appartiennent à sa catégorie, et un poids négatif aux autres.

### II.2.3- Les réseaux de neurones récurrents :

Le réseau de neurones récurrent est un réseau dans son graphe de connexion contient au moins un cycle, donc l'information se propage dans les deux sens, y compris des couches profondes aux premières couches, ils sont plus proches du vrai fonctionnement du système nerveux, qui n'est pas à sens unique, ces réseaux possèdent des connexions récurrentes au sens où elles conservent des informations en mémoire car ils peuvent prendre en compte à un instant  $t$  un certain nombre d'états passés.

**Exemple :** ART, Hopfield, Kohonen (Cartes auto-organisatrices 'Self-Organizing Maps')



**Figure 15 : Réseaux de neurones récurrents.**

### III- La classification d'image avec les réseaux de neurones convolutifs :

#### III.1- Introduction :

Les réseaux de neurones convolutionnels sont les systèmes les plus adaptés à la classification des images, c'est pourquoi nous allons tout d'abord expliquer et définir les caractéristiques des images.

#### III.2- L'infographie et les images :

##### III.2.1- Définition d'image :

Une image est une représentation planaire d'une scène ou d'un objet situé en général dans un espace tridimensionnel, elle est issue du contact des rayons lumineux provenant des objets formants la scène avec un capteur (caméra, scanner, rayons X, ...). Il ne s'agit en réalité que d'une représentation spatiale de la lumière. Est composée de points appelés : **Pixels**.

**Pixel** : Est l'unité de base permettant de mesurer la définition d'une image. Le mot pixel vient de la locution anglaise « *picture element* », qui signifie « *élément d'image* ».

##### III.2.2- Caractéristiques d'une image :

- **Définition :**

C'est le nombre de pixels qui composent cette image en hauteur et en largeur.

- **Résolution :**

C'est le nombre de pixels par unité de longueur qui est en général le pouce (1 pouce = 2,54cm). On parle alors de pixel/pouce **PPP** (point par pouces) ou **DPI** (dots per inch) en anglais, parfois en point par cm.

**Exemple :**

Une image de 1500 colonnes sur 1000 lignes, est d'une définition de :

$1500 * 1000 = 1\ 500\ 000$  pixels.

- **Taille d'une image :**

La taille d'une image est le produit de sa définition par le nombre d'octets par pixel.

**Exemple :**

Pour une image RGB (rouge, bleu, gré) de 1500 colonnes et de 1000 lignes, sa taille est de l'ordre de :

$1500 * 1000 * 3 = 4\ 500\ 000$  octets.

Plus le nombre de pixels qui représente l'image est grande, plus la qualité est meilleure.

### III.2.3- Différents types d'images :

#### a) Image RVB :

Pour représenter la couleur d'un pixel, il faut donner trois nombres, qui correspondent au dosage de trois couleurs de base : Rouge, Vert, Bleu. On peut ainsi représenter une image couleur par trois matrices chacune correspondant à une couleur de base.

#### b) Image d'intensités :

C'est une matrice dans laquelle chaque élément est un réel compris entre 0 (noir) et 1 (blanc). On parle aussi d'image en niveaux de gris, car les valeurs comprises entre 0 et 1 représentent les différents niveaux de gris.

#### c) Image binaire :

Une image binaire est une image pour laquelle chaque pixel ne peut avoir pour valeur que 0 ou 1. La manipulation de telles images regorge d'outils spécialisés ainsi que de théories mathématiques.

### III.2.4- Codage d'images numériques :

L'image numérique se représente sous la forme d'un tableau à deux dimensions contenant des valeurs entières pour les images en niveaux de gris ou des triplets de valeurs entières pour les images couleurs.

La valeur stockée dans une case est codée sur un certain nombre de bits déterminant la couleur ou l'intensité du pixel, on l'appelle profondeur de codage (parfois profondeur de couleur).

Il existe plusieurs systèmes de codage, on cite par exemple :

**RGB (RVB) :** Basé sur un mélange additif (combinaison de rayons lumineux) de trois couleurs primaires Rouge, Vert, Bleu).

**CMYK (CMJN) :** Basé sur un mélange soustractif (combinaison de pigments colorés) de trois couleurs (Cyan, Magenta, Jaune) et du noir.

**HLS (TLS) :** Basé sur la perception physiologique de la couleur par l'œil humain (Teinte, Luminance, Saturation).

### III.2.5 La numérisation :

La numérisation est l'étape de passage de l'état d'image physique, à la représentation de chacun de ces pixels en valeur numérique codée en binaire.

Pour chaque pixel, on code en binaire chacune des 3 valeurs des composantes R, V, B de la couleur du pixel. Le code binaire de l'image est obtenu en indiquant successivement pour chaque pixel le code binaire des 3 composantes. Si on code chaque composante sur 8 bits, chaque pixel sera donc représenté par 24 bits (exemple en **figure 16**).

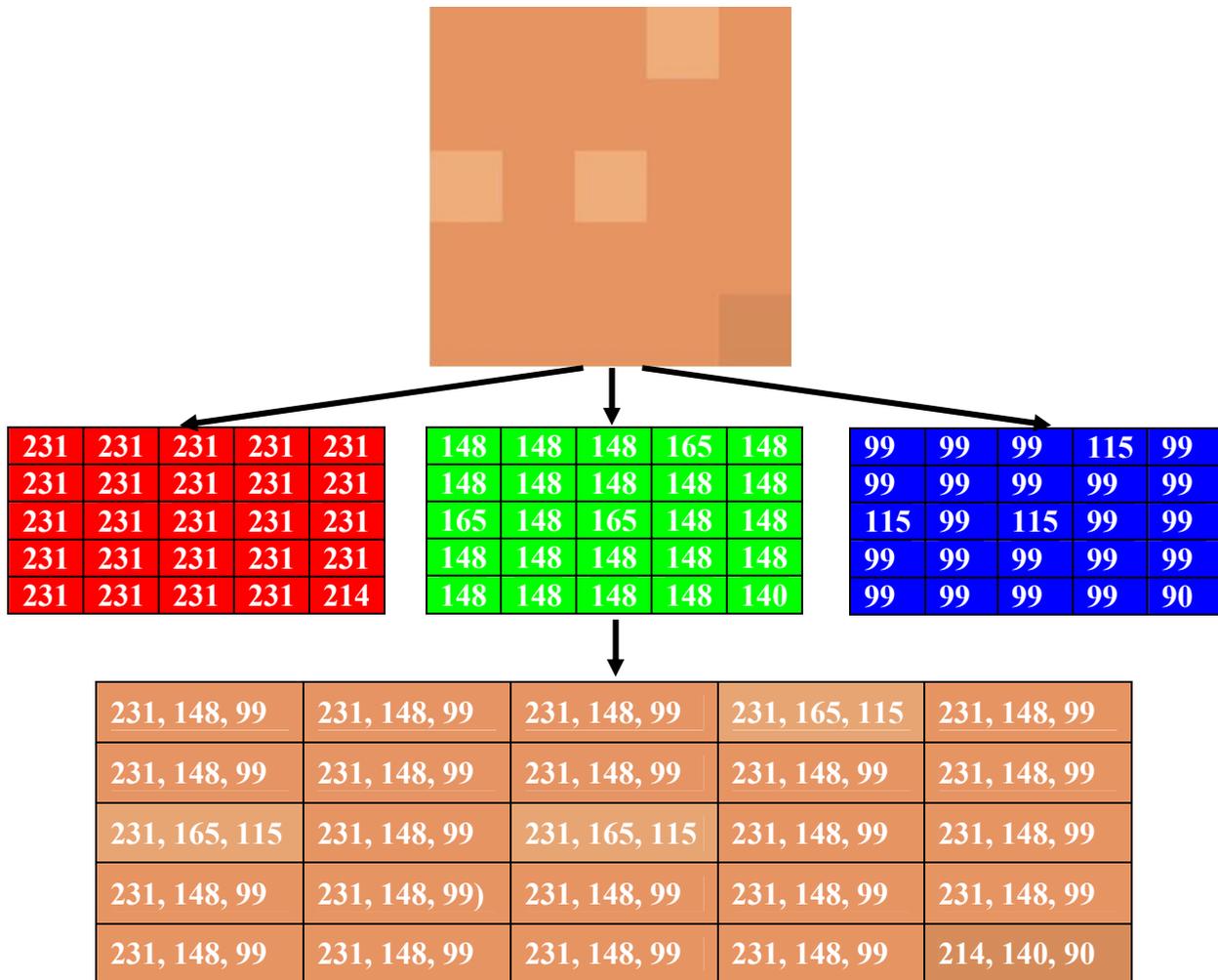


Figure 16 : Codage sur 3 octets d'un morceau d'image couleur.

#### IV. Traitement numérique des images :

Les techniques de traitement sont destinées à l'exploitation des informations contenues dans les images, ceci dans le but d'améliorer la qualité des images et de les rendre plus facilement interprétables, en d'autres termes elles permettent d'augmenter la qualité visuelle de l'image.

##### IV.1 Filtrage Numérique :

Les images numériques telles qu'elles sont acquises, sont très souvent inexploitable pour le traitement d'images. Elles contiennent des signaux bruités. Pour remédier à cela, différents prétraitements pour l'amélioration ou la correction sont effectués. [10]

On peut scinder les filtres en deux grandes catégories :

##### IV.1.1 Filtres linéaires :

Les filtres linéaires transforment un ensemble de données d'entrée en un ensemble de données de sortie par une convolution bidimensionnelle qui est une opération mathématique.

Ils permettent de supprimer le bruit dans l'image. Chaque filtre a une taille  $N \times N$  avec  $N$  impair. [2]

Les filtres linéaires les plus connus sont les filtres passe-bas, passe-haut.

#### A. Filtre passe-bas (lissage) :

Ce filtre n'affecte pas les composantes de basse fréquence dans les données d'une image, mais doit atténuer les composantes de haute fréquence. L'opération de lissage est souvent utilisée pour atténuer le bruit et les irrégularités de l'image. Elle peut être répétée plusieurs fois, ce qui crée un effet de flou. En pratique, il faut choisir un compromis entre l'atténuation du bruit et la conservation des détails et contours significatifs. [11]

$$1/9 * \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

**Masque du filtre**

#### B. Filtre Passe-haut (Accentuation) :

Le renforcement des contours et leur extraction s'obtiennent dans le domaine fréquentiel par l'application d'un filtre passe-haut. Le filtre digital passe-haut a les caractéristiques inverses du filtre passe-bas. Ce filtre n'affecte pas les composantes de haute fréquence d'un signal, mais doit atténuer les composantes de basse fréquence. [11]

$$H = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

**Masque du filtre**

#### IV.1.2 Filtres non linéaire :

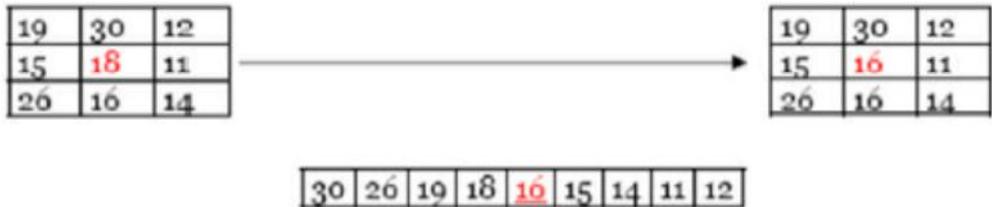
Ils sont conçus pour régler les problèmes des filtres linéaires, Leur principe est le même que celui des filtres linéaires, il s'agit toujours de remplacer la valeur de chaque pixel par la valeur d'une fonction calculée dans son voisinage. La différence majeure, est que cette fonction n'est plus linéaire mais une fonction quelconque (elle peut inclure des opérateurs de comparaisons ou de classification). Les filtres non linéaires les plus connus sont : [11]

##### A. Filtre médian :

Ce filtre est très utilisé pour éliminer le bruit sur une image qui peut être de différentes origines (poussières, petits nuages, baisse momentanée de l'intensité électrique sur les capteurs, ...). L'avantage de ce filtre réside dans le fait qu'il conserve les contours alors que les autres types de filtres ont tendance à les adoucir. [11]

L'algorithme de filtre médian est le suivant :

1. Trier les valeurs par ordre croissant.
2. Remplacer la valeur du pixel centrale par la valeur située au milieu de la triée.
3. Répéter cette opération pour tous les pixels de l'image.



**Figure 17 : Principe du filtre médian.**

### B. Filtre maximum :

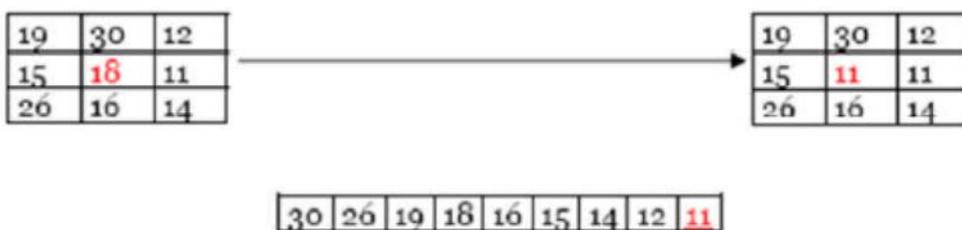
On applique le même traitement que celui du filtre médian mais la valeur du pixel du centre comme la montre la figure 18, va être changée par le maximum.



**Figure 18 : Principe du filtre maximum.**

### C. Filtre minimum :

On applique le même traitement que celui du filtre maximum mais, cette fois, la valeur du pixel du centre comme la montre la figure 19 va être remplacée par le minimum. [11]



**Figure 19 : Principe du filtre minimum.**

L'objectif principal du filtrage est d'améliorer la qualité visuelle de l'image (ex : atténuer et/ou éliminer le bruit) ou extraire des attributs de l'image (ex : contours), en modifiant la valeur de niveau du gris d'un pixel en fonction de la valeur de ses voisins, les éléments significatifs de l'image doivent être préservés au mieux.

## **V Les réseaux de neurones convolutifs :**

### **V.1. Définition :**

Les réseaux de neurones convolutifs ou (CNN pour Convolutional Neural Network en anglais), sont des réseaux de neurones multicouches spécialisés dans la classification des images et la reconnaissance de formes.

Ils comportent deux parties distinctes, première partie c'est la partie convolutive, son architecture repose sur des couches de convolution alternant avec des couches d'agrégation (pooling). L'image en entrée est fournie sous forme de matrice de pixels, elle passe par une succession de filtres au niveau de chaque couche de convolution, créant des nouvelles images appelées cartes de convolutions. La couche d'agrégation (pooling) réduit la taille de l'image par une fonction de max pooling.

Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code CNN.

Ce code CNN en sortie de la partie convolutive est ensuite branché en entrée d'une deuxième partie, constituée de couches entièrement connectées (perceptron convolutifs multicouche). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image.

La sortie est une dernière couche comportant un neurone par catégorie. Les valeurs numériques obtenues sont généralement normalisées entre 0 et 1, de somme 1, pour produire une distribution de probabilité sur les catégories.

### **V.2. Architecture de réseaux de neurones convolutifs :**

#### **a) Couches convolutives :**

Reconnaître des objets dans des images revient à comparer l'image pixel par pixel, pour trouver des caractéristiques.

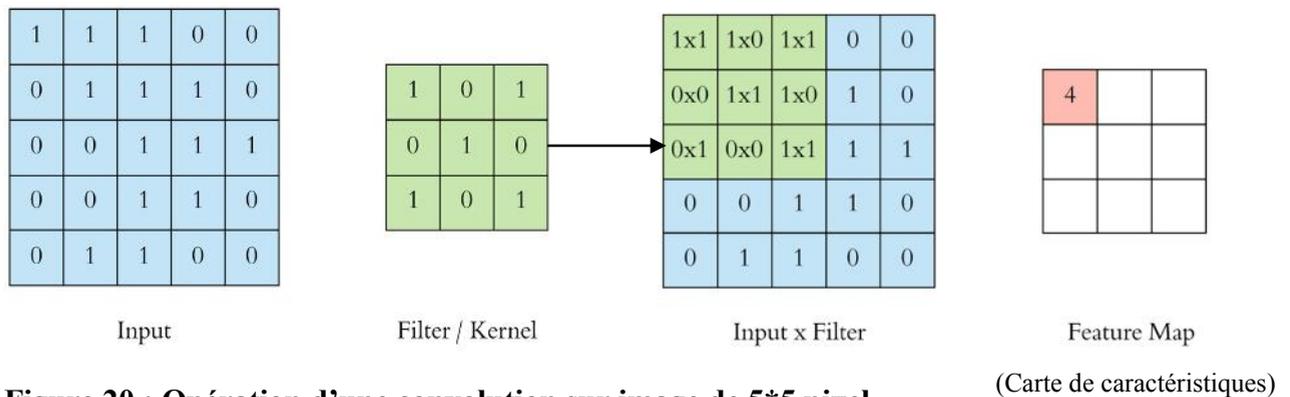
Bien qu'efficaces pour le traitement d'images, les réseaux de neurones multicouches, ont beaucoup de mal à gérer des images de grande taille, ce qui est dû à la croissance exponentielle du nombre de connexions avec la taille de l'image.

Par exemple une image de taille 200 x 200 x 3 (200 de large, 200 de haut, 3 canaux de couleur), conduirait ainsi à traiter 120 000 entrées par neurone ce qui, multiplié par le nombre de neurones, devient énorme.

Une couche de convolution permet de reconnaître des formes sur une image, généralement avec des filtres de taille 2 x 2 ou 3 x 3 pixels, ce qui limite le nombre d'entrées.

Pour chaque filtre, et pour tout carré de la taille du filtre sur la matrice d'entrée, les cases correspondantes de la matrice de sorties sont égales à la somme de toutes les valeurs du filtre, multipliées respectivement par ceux des carrés d'entrée.

Ainsi la taille de la matrice de sortie est égale à celle de l'entrée moins deux lignes et deux colonnes.



**Figure 20 : Opération d'une convolution sur image de 5\*5 pixel.**

### b) Couches de pooling :

Sont placées entre deux couches de convolution, ces couches abaissent les données de l'image extraites par les couches convolutionnelles pour réduire la dimension de la carte des caractéristiques afin de diminuer le temps de traitement.

Une couche de pooling permet de réduire la taille de la matrice d'entrée, d'augmenter la vitesse tout en gardant les informations les plus importantes.

Il existe deux types de pooling :

- **Le Maximum Pooling :**

Divise la taille de l'entrée par 2 en prenant le maximum des carrés de taille 2 x 2 pixels.

- **L'Average Pooling :**

Divise la taille de l'entrée par 2 en calculant la moyenne des carrés de taille 2 x 2 pixels.

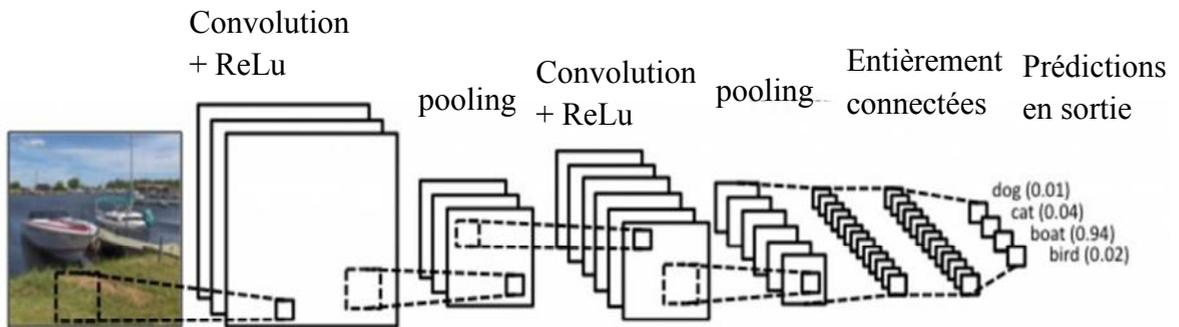
Le maximum pooling est le plus utilisé surtout pour la reconnaissance des objets de grandes tailles et complexes.

### c) Couches entièrement connectées :

La couche entièrement connectée est un perceptron multicouche qui utilise une fonction d'activation softmax dans la couche de sortie. Le terme «entièrement connecté» implique que chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante.

La sortie des couches convolutionnelles et de regroupement représente les fonctionnalités de haut niveau de l'image d'entrée.

La couche entièrement connectée a pour le but d'utiliser ces fonctionnalités pour classer l'image d'entrée dans différentes classes en fonction du jeu de données d'apprentissage.



**Figure 21 : Architecture d'un réseau de neurones convolutifs.**

Par exemple, ce réseau pourrait retourner 4 probabilités en sorties, chacune correspond à ce que l'image passé en entrée soit un bateau, un oiseau, un chien ou un chat.

Un CNN est composée d'une pile de modules convolutifs qui effectuent une extraction de fonctionnalités.

Chaque module se compose d'une couche convolutionnelle suivie d'une couche pooling, le dernier module convolutif est suivi d'une ou plusieurs couches entièrement connectées qui effectuent la classification.

La couche complètement connectée final dans un CNN contient un seul nœud pour chaque classe cible dans le modèle (toutes les classes possibles que le modèle peut prédire), avec une fonction d'activation de softmax pour générer une valeur entre 0 – 1 pour chaque nœud (la somme de toutes ces valeurs de softmax est égales à 1).

Donc ces couches entièrement connectées effectuent une classification sur les caractéristiques extraites par les couches convolutifs et diminuées par les couches pooling.

### V.3. Fonctionnement d'un réseau de neurones convolutifs :

#### V.3.1 Couche convolution :

La tâche de la couche de convolution est de chercher les caractéristiques présentes dans l'image, pour réaliser cette tâche on applique des filtres à l'image. Un filtre est une matrice souvent carré (2 x 2 ou 3 x 3) qu'on va balayer sur l'ensemble d'image, en sortie on obtient une image de taille réduite qu'on appelle « cartes de convolution ». Dans la couche de convolution on applique un ensemble  $K$  de filtres, on obtient alors une pile de  $K$  activations maps (cartes de convolution).

Chaque filtre détecte une caractéristique particulière à chaque emplacement de l'entrée, la sortie  $Y_j^i$  ( $j \in \{1 .. K\}$ ) de la couche  $C$  se compose de  $K$  caractéristiques. Chaque carte de convolution  $M_j^i$  est le résultat d'une somme de convolution des cartes de la couche précédente  $M_j^{i-1}$  par son noyau de convolution respectif. Un biais  $b_j^i$  est ensuite ajouté et le résultat est passé à une fonction de transfert non-linéaire, alors  $Y_j^i$  est calculé : [12]

$$Y_j^i = b_j^i + \sum_{n=1}^N Y_n^{i-1} \cdot K_n^i$$

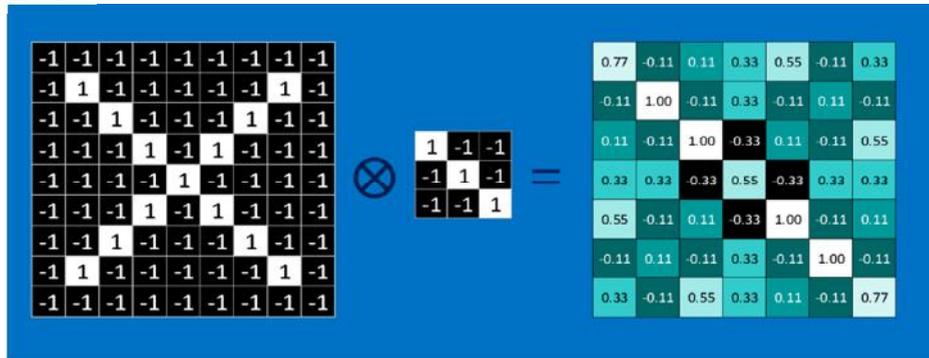


Figure 22 : Etape de convolution.

### V.3.2 Unités linéaires rectifiées (ReLU) :

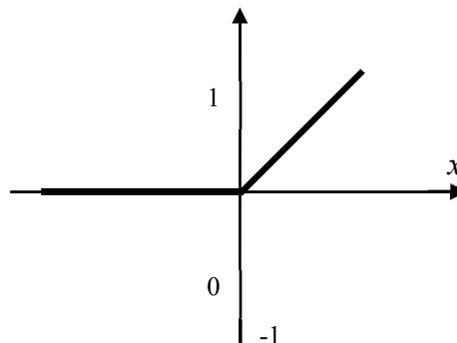
Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie.

La fonction ReLU (abréviation de : **Unités Rectifié linéaires**) :

$$Y(x) = \max(0, x)$$

Cette fonction force les neurones à retourner des valeurs positives.

$$F(x) = \max(0, x)$$



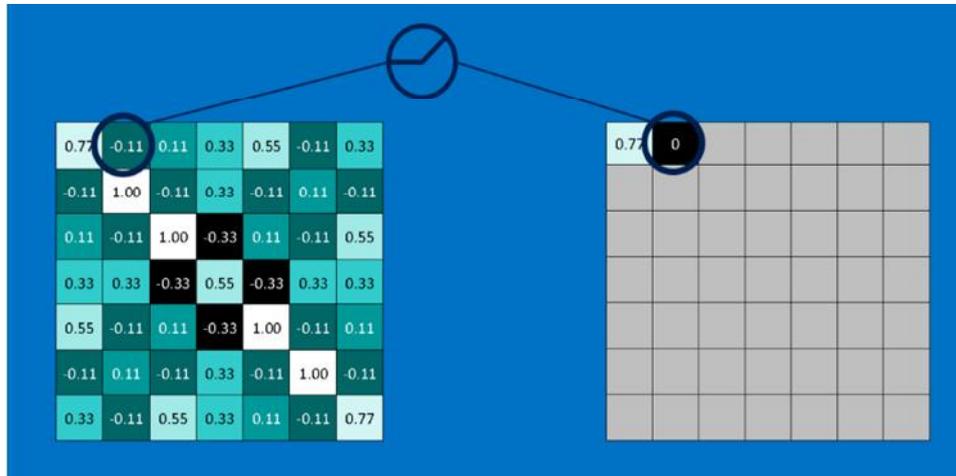


Figure 23 : Fonction Relu.

### V.3.3 La couche de pooling :

La couche de pooling (également appelée sous-échantillonnage) réduit la dimensionnalité de chaque carte d'entités mais conserve les informations les plus importantes. La mise en commun spatiale peut être de différents types : Max, Moyenne, Somme etc.

Dans le cas d'un pool maximum, nous définissons un voisinage spatial (par exemple, une fenêtre  $2 \times 2$ ) et prenons l'élément le plus grand de la carte d'entités rectifiées dans cette fenêtre. Au lieu de prendre l'élément le plus grand, nous pourrions également prendre la moyenne (moyenne Pooling) ou la somme de tous les éléments dans cette fenêtre.

La figure montre un exemple d'opération de regroupement maximal sur une carte d'entité rectifiée (obtenue après convolution + operation ReLU).

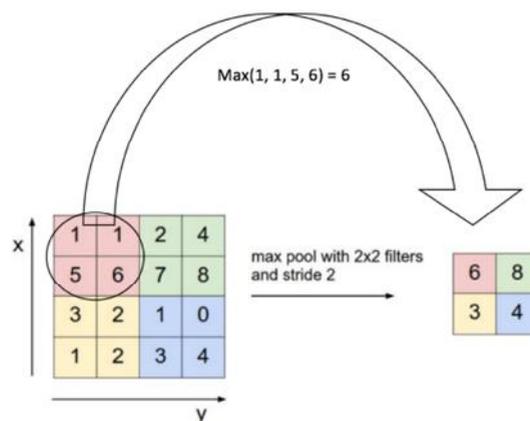
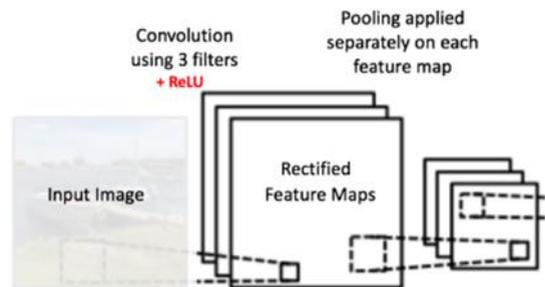


Figure 24 : Max-Pooling avec un filtre 2x2 et un pas de 2.

Nous faisons glisser notre fenêtre de  $2 \times 2$  par 2 cellules et prenons la valeur maximale dans chaque région. Comme le montre la figure 24, cela réduit la dimensionnalité de notre carte de caractéristiques.

Dans le réseau représenté sur la **Figure 25**, l'opération de regroupement est appliquée séparément à chaque carte de caractéristique (notez que, grâce à cela, nous obtenons trois cartes de sortie à partir de trois cartes d'entrée)



**Figure 25 : Mise en commun appliquée aux cartes de fonctions rectifiées.**

La fonction de Pooling consiste à réduire progressivement la taille spatiale de la représentation d'entrée. En particulier, la mise en commun

- Rend les représentations d'entrée (dimension caractéristique) plus petites et plus faciles à gérer.
- Réduit le nombre de paramètres et de calculs dans le réseau.
- Rend le réseau invariant aux petites transformations, distorsions et translations dans l'image d'entrée (une petite distorsion en entrée ne changera pas la sortie de Pooling - puisque nous prenons la valeur maximum / moyenne dans un voisinage local).
- Nous aide à arriver à une représentation invariable presque à l'échelle de notre image (le terme exact est "équivalent"). Ceci est très puissant car nous pouvons détecter des objets dans une image, peu importe où ils se trouvent.

#### V.3.4 Couche entièrement connectée :

Les couches totalement connectées dans un réseau convolutif sont pratiquement un perceptron multicouche (généralement deux à trois couches), qui vise à cartographier le volume d'activation de la combinaison des différentes couches précédentes en une distribution de probabilité de classe.

La couche de sortie du perceptron multicouche aura  $M^{l-1}$  sorties, c'est-à-dire des neurones de sortie où  $l$  indique le nombre de couches dans le perceptron multicouche.

La principale différence par rapport à un perceptron multicouche standard est la couche d'entrée où, au lieu d'un vecteur, on prend un volume d'activation en entrée.

La couche entièrement connectée est définie comme :

$$Y_i^l = f(Z_i^l) \quad \text{avec} \quad Z_i^l = \sum_{j=1}^{m^{l-1}} W_{i,j}^l y_j^{l-1}$$

Le but de la structure complète entièrement connectée est de régler les paramètres de poids  $W_{i,j}^l$ .

### V.3.5 L'entraînement d'un réseau de neurone convolutionnelle :

L'entraînement d'un CNN consiste à déterminer et à calculer empiriquement la valeur de chacun de ses poids. Le principe est le suivant : le CNN traite une image (de la base de données d'entraînement) et en sortie il fait une prédiction, c'est-à-dire qu'il dit à quelle classe il pense que cette image appartient. Sachant qu'on connaît préalablement la classe de chacune des images d'entraînement, on peut vérifier si ce résultat est correct. En fonction de la véracité de ce résultat, on met à jour tous les poids du CNN selon un algorithme qui s'appelle la rétropropagation du gradient de l'erreur.

Lors de la phase d'entraînement du modèle, le processus expliqué ci-dessus est répété plusieurs fois et avec la totalité des images de la base de données d'entraînement. Le but étant que le modèle classifie au mieux ces données.

Lorsque le modèle a fini de mettre à jour ses poids, on évalue le modèle en lui présentant la base de données de validation.

Il classe toutes ces images (qui sont des images que le modèle n'a jamais vues) et on calcule son taux de bonne classification, c'est ce qu'on appelle la précision du modèle. [12]

### V.4 Sur-apprentissage et sous-apprentissage :

A la fin du processus d'apprentissage trois cas de figure peuvent se présenter :

- Le modèle est aussi performant sur les données d'entraînement (images sur lesquelles il s'entraîne) que sur les données de validation (images qu'il n'a jamais vues), cela signifie que le modèle a très bien fait son travail et qu'il reconnaît aussi bien les images qu'il connaît que celles qu'il n'a jamais vues.
- Le modèle reconnaît très bien les images d'entraînement et moins bien celles de validation. Le modèle aura une faible capacité prédictive, il n'arrive pas à généraliser. On parle alors de sur-entraînement. Dans ce cas-là on peut ajouter davantage d'images pour pallier ce problème.
- Le modèle ne reconnaît pas très bien les images d'entraînements et pas très bien non plus les images de validation. On parle alors de sous-apprentissage. Dans ce cas-là ajouter plus d'images ne servira à rien, c'est généralement le modèle choisie qui ne convient pas, il faudrait utiliser un modèle plus complexe. [15]

## V.5 Indicateurs de performance d'un classifieur :

- **Matrice de confusion**

Prenons l'exemple d'un classifieur binaire, c'est-à-dire, qui prédit 2 classes notées classe 0 et classe 1.

Pour mesurer les performances de ce classifieur, il est d'usage de distinguer 4 types d'éléments classés pour la classe voulue :

**1-Vrai positif VP.** Élément de la classe 1 correctement prédit

**2-Vrai négatif VN.** Élément de la classe 0 correctement prédit

**3-Faux positif FP.** Élément de la classe 1 mal prédit

**4-Faux négatif FN.** Élément de la classe 0 mal prédit

Ces informations peuvent être rassemblés et visualisés sous forme de tableau dans une matrice de confusion. Dans le cas d'un classifieur binaire, on obtient :

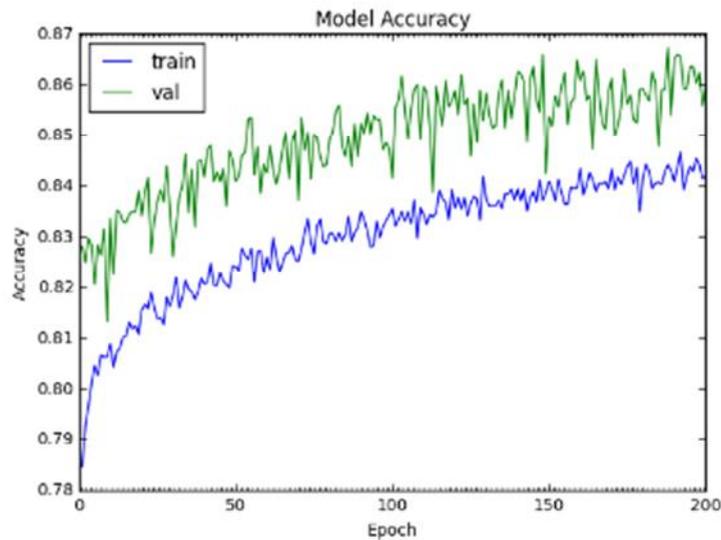
		Classe prédite	
		Classe 0	Classe 1
Classe réelle	Classe 0	VN	FN
	Classe 1	FP	VP

**Figure 26 : La matrice de confusion d'un classifieur.**

En particulier, si la matrice de confusion est diagonale, le classifieur est parfait. [8]

- **Le graphe de précision**

Un autre indicateur important pour visualiser la performance d'un CNN est le graphe de précision, le graphe de précision est une représentation de la précision de modèle en fonction de nombre d'itération utilisé pour l'apprentissage et le test, ce dernier comporte deux courbe, une courbe pour les données d'entraînement et une autre pour les données de test

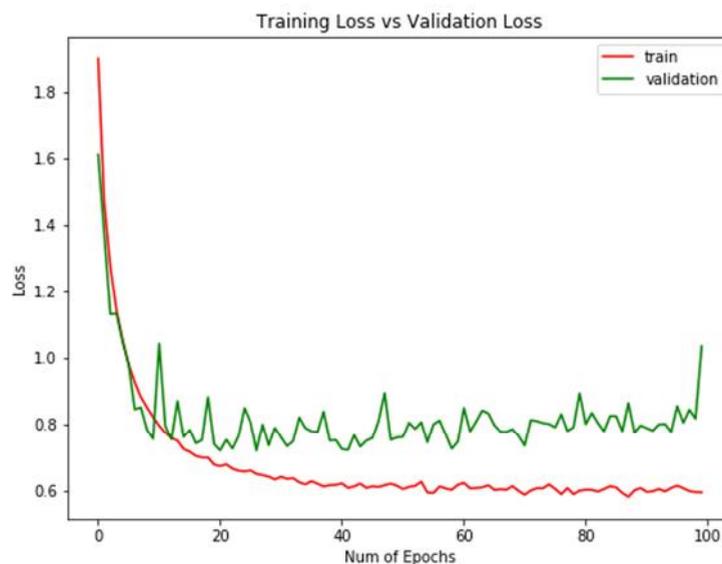


**Figure 27 : Le graphe de précision d'un réseau de neurone convolutifs.**

En particulier, si les deux courbes atteignent une précision = 100% la courbe, le classifieur est parfait

- **Le graphe d'erreur**

Le graphe d'erreur est une représentation de taux de dégradation d'erreur du modèle en fonction de nombre d'itération utilisé pour l'apprentissage et le test, ce dernier comporte deux courbe, une courbe pour les données d'entraînement et une autre pour les données de test



**Figure 28 : Le graphe d'erreur d'un réseau de neurone convolutifs.**

En particulier, Si l'erreur de deux courbes diminue, le classifieur est parfait.

- **Indicateurs de base**

Il est possible de calculer plusieurs indicateurs résumant la matrice de confusion. Par exemple si nous souhaitons rendre compte de la qualité de la prédiction sur la classe 1, on définit :

**Précision.** Proportion d'éléments bien classés pour une classe donnée :

$$Precision_{de\ la\ classe\ 1} = \frac{VP}{VP + FP}$$

**Rappel.** Proportion d'éléments bien classés par rapport au nombre d'éléments de la classe à prédire :

$$Rappel_{de\ la\ classe\ 1} = \frac{VP}{VP + FN}$$

**F-mesure.** Mesure de compromis entre précision et rappel :

$$F - mesure_{de\ la\ classe\ 1} = \frac{2 * (Précision * Rappel)}{Précision + Rappel}$$

Il est possible de calculer tous ces indicateurs pour chaque classe. La moyenne sur chaque classe de ces indicateurs donne des indicateurs globaux sur la qualité du classifieur.

$$Precision = \frac{1}{k} \sum_{i=1}^k \frac{VP_i}{VP_i + FP_i}$$

$$Rappel = \frac{1}{k} \sum_{i=1}^k \frac{VP_i}{VP_i + FN_i}$$

$$F - mesure = \frac{2 * (Précision * Rappel)}{Précision + Rappel}$$

### **Conclusion :**

Dans cette partie, nous avons défini les réseaux de neurones et nous avons détaillé les réseaux de neurones convolutionnels (architecture et fonctionnement). Dans la prochaine partie, nous allons présenter le système de classification d'images, la description des couches et leur fonctionnement.

**Quatrième chapitre :**

# Réalisation

## I- Introduction :

Après avoir présenté dans l'état de l'art l'apprentissage et les réseaux de neurone convolutifs. Dans cette partie nous allons détailler le classifieur qu'on a conçu en utilisant les réseaux de neurones convolutifs. Nous avons conçu un classifieur en utilisant trois modèles qui diffèrent les uns des autres par le nombre de couches de convolutions, le nombre de couches de pooling et le nombre d'itérations.

Nous avons comparé les différents classifieurs en utilisant les courbes de précisions, les courbes d'erreurs et les matrices de confusions.

## II- Description du notre système :

Le classifieur est un système de classification d'images supervisé conçu avec les réseaux de neurones convolutifs.

Nous avons pris 9000 images sur google, nous les avons soumis sur plusieurs traitement avant de les injecter dans notre réseau. Nous avons traité images en utilisant la bibliothèque *Opencv* qui permet de normaliser<sup>(2)</sup>, de redimensionner les images et de définir le nombre de canal pour chaque image. Ensuite nous avons utilisé la bibliothèque *Numpy* pour créer des vecteurs multi dimensions pour charger toutes les images traitées afin de définir une classe (label) pour chaque une de ces images.

Pour la création du réseau à convolution, nous avons opté pour le framework *keras*.

*Keras* nous a permis de créer des tensors (tableau à 4 dimensions), chaque tensors représenté par la taille de batch, la taille de l'image, sa hauteur, la largeur de l'image et le nombre de canal de chaque image.

Ensuite nous allons présenter les résultats de chaque modèle en donnant les graphes de précisions, les graphes d'erreur ainsi que la matrice de confusion. Nous allons décrire les paramètres ou les factures essentiels à modifier dans un modèle, pour améliorer ces performances.

---

(2) : divisé ces valeurs sur 255.



### III- Présentation des outils utilisés dans l'implémentation

- **Keras**

Keras est une bibliothèque de réseaux neuronaux de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow ou Theano . Il a été développé dans le but de permettre une expérimentation rapide

Utilisez Keras si vous avez besoin d'une bibliothèque d'apprentissage en profondeur qui :

-Permet un prototypage facile et rapide (grâce à la modularité totale, au minimalisme et à l'extensibilité).

-Prend en charge les réseaux convolutionnels et les réseaux récurrents, ainsi que les combinaisons des deux.

-Prend en charge les schémas de connectivité arbitraires (y compris les formations à entrées multiples et à sorties multiples).

-Fonctionne de manière transparente sur le processeur et le processeur graphique

- **Langage de programmation python**

Le langage de programmation Python a été créé par Guido van Rossum en 1990 et est rendu disponible sous licence libre. Son développement est aujourd'hui assuré par la **Python Software Foundation**, fondée en 2001. Il s'agit d'un langage interprété fonctionnant sur la plupart des plateformes informatiques (notamment Linux, Windows et macOS). Il est également très apprécié des pédagogues qui le considèrent comme étant un bon langage pour s'initier aux concepts de base de la programmation.

Depuis sa création, le succès du langage n'a cessé de croître, jusqu'à le porter parmi les dix langages de programmation les plus populaires. Il s'agit d'un langage de haut niveau, à usage général. Les dernières versions stables du langage sont la 2.7.12 (25 juin 2016) et la 3.5.2 (27 juin 2016). Ces deux versions, communément appelées Python 2 et Python 3, continuent toutes les deux d'exister, essentiellement pour des raisons de compatibilité. Dans ce travail, nous utiliserons la version 3.

Python est utilisé dans de nombreux projets, de types différents. Il est notamment utilisé comme langage de script pour des applications web. Il est également très présent dans des applications de calculs scientifiques, et est intégré dans de nombreux logiciels de modélisation, comme langage de script. Enfin, il est aussi utilisé dans le développement de jeux vidéo ou dans l'écriture de tâches dans le domaine de l'intelligence artificielle.

- **OpenCV**

OpenCV (pour Open Computer Vision) est une bibliothèque graphique libre, initialement développée par Intel, spécialisée dans le traitement d'images en temps réel. La société de robotique Willow Garage et la société ItSeez se sont succédé au support de cette

bibliothèque. Depuis 2016 et le rachat de ItSeez par Intel, le support est de nouveau assuré par Intel.

- **Configuration utilisé dans l'implémentation :**

Pc portable (Lenovo core i5)

Ram de taille 4 GO

Carte graphique NVidia Gforce GT777

Disque dure de taille 500 GO

Système d'exploitation : Windows 7 64 bits

#### IV. Création d'un réseau de neurone convolutifs avec Keras :

Avant de présenter l'architecture de trois modèle que nous allons créer, nous allons présenter les démarches que nous avons suivies afin de créer ces trois modèles d'une manière générale

##### IV.1 Création d'une base de données image (images de train et images de test) :

Dans notre projet nous avons utilisé 9000 images récoltées de Google (3000 image pour les chats, 3000 image pour les chiens, 3000 image pour les chevaux)



**Figure 30 : Liste d'images.**

Avant d'injecter ces images à notre réseau de convolution plusieurs traitements sont appliqués sur ces images afin de construire une base de données image qui contient des images pour l'entraînement et des images pour le test :

Définir le chemin vers les données images :

Redimensionner :

- Une longueur de 128 pixels

- Une largeur de 128 pixels

-Nombre de canaux =1 (des images au niveau de gris)

-Normaliser l'ensemble d'image on divise chaque image sur 255

Diviser l'ensemble de données pour avoir un ensemble d'entraînement pour entraîner le réseau et un ensemble image de test pour tester ou évaluer le réseau.

Après avoir divisé les 9000 images, nous obtenons 7200 images pour l'entraînement et 1800 images pour le test.

## IV.2 Création d'un modèle :

Avant de créer le modèle avec Keras, il faut d'abord importer toutes les bibliothèques nécessaires pour la formation de modèle, Ainsi que l'instance de toutes les couches.

```
import keras
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Activation, Flatten
from keras.layers.convolutional import Convolution2D, MaxPooling2D
from keras.optimizers import SGD, RMSprop, adam
```

Voici l'implémentation d'un modèle avec Keras.

```
model = Sequential()
```

Et l'empilement des couches se fait avec .add()

```
model.add(Convolution2D(32, 3, 3, border_mode='same', activation='relu'))
#model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

## IV.3 Compilation et évaluation de modèle :

Pour compiler un modèle avec Keras il faut sélectionner la fonction d'erreur ainsi que l'optimiseur utilisé pour l'apprentissage.

```
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```

Après la compilation de modèle en lance l'apprentissage, en spécifiant le nombre d'itération.

```
hist = model.fit(X_train, y_train, batch_size=16, nb_epoch=num_epoch, verbose=1, validation_split=0.2)
```

Après l'exécution de modèle, on l'évalue sur la base de test avec la commande.

```
score = model.evaluate(X_test, y_test, verbose=0)
```

## IV-Architecture de notre réseau :

Après que nous avons présenté comment créer un modèle avec Keras, dans cette section nous allons définir l'architecture de trois modèles créés ainsi que les résultats obtenus après l'exécution.

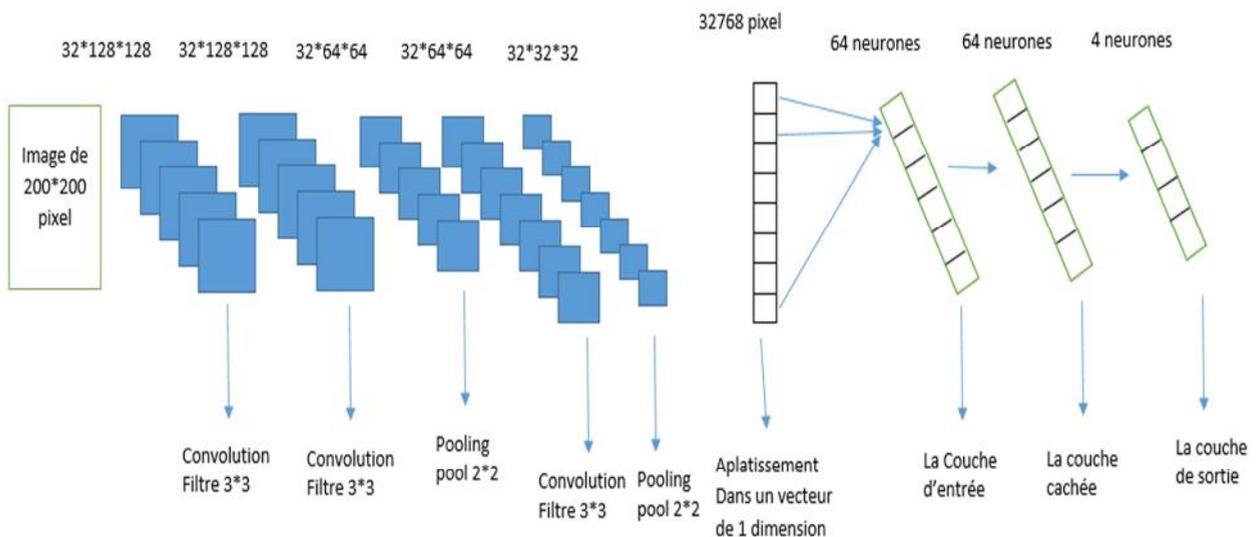
### V.1 -Architecture du premier modèle :

Comme nous avons décrit précédemment chaque modèle se compose de deux parties, une partie convolutionnels et une partie classifieur (perceptron).

- **La partie convolutionnels :**

Le premier modèle présenté dans **la figure 30**, est composé de trois couches de convolution et deux couches de maxpooling et trois couches entièrement connectées (dense) et le nombre d'itérations fixé à 15.

L'image en entrée est de taille 128\*128, l'image passe d'abord à la première couche de convolution. Cette couche est composée de 32 filtres de taille 3\*3, Chacune de nos couches de convolution est suivie d'une fonction d'activation ReLU. Cette fonction force les neurones à retourner des valeurs positives, après cette convolution 32 cartes de caractéristique de taille 126\*126 seront créés.



**Figure 31 : Architecture du premier modèle.**

Les 32 cartes de caractéristique qui sont obtenus auparavant, sont données en entrée de la deuxième couche de convolution qui est composée aussi de 32 filtres, une fonction d'activation RELU est appliquée sur cette couche, ensuite on applique Maxpooling pour réduire la taille de l'image ainsi la quantité de paramètres et de calcul. À la sortie de cette couche, nous aurons 32 cartes de caractéristique de taille 62\*62.

On répète la même chose avec les couches de convolutions trois, cette couche est composée de 32 filtres, la fonction d'activation ReLU est appliquée toujours sur chaque convolution.

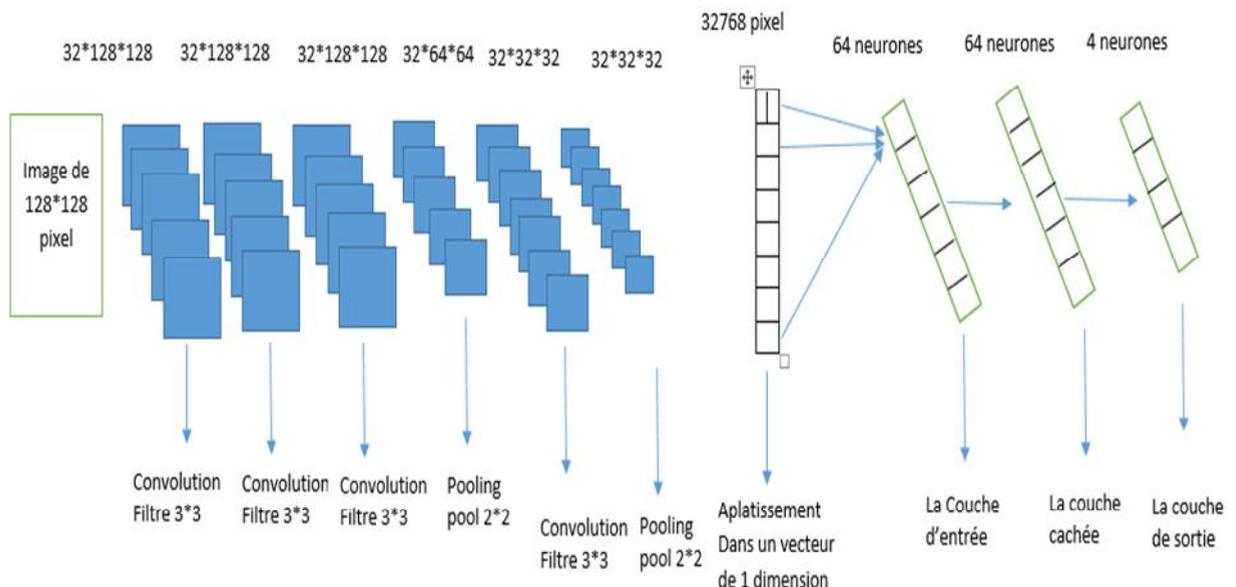
Une couche de Maxpooling est appliquée après la couche de convolution trois. À la sortie de cette couche, nous aurons 32 cartes de caractéristique de taille 30\*30. Le vecteur de caractéristiques issu des convolutions a une dimension de 9000(32\*30\*30).

Et pour connecter le résultat de la partie convolution avec la partie (classifieur) nous avons aplatie ce résultat dans un vecteur du caractéristiques de la dimension 32768 (32\*32\*32) qui sera l'entrée de la premier couche entièrement connectée de perceptron.

- **La partie classifieur**

La partie classifieur de premier modèle est composée de trois couches, une couche d'entrée de 64 neurones avec une fonction d'activation 'RELU 'et une couche caché de 64 neurones avec une fonction d'activation 'RELU' et une couche de sortie avec une fonction de décision 'SOFTMAX'.

**V.2- Architecture du deuxième modèle**



**Figure 32 : Architecture de deuxième modèle.**

- **la partie convolutionnelle du deuxième modèle :**

Dans la partie convolutionnelle du deuxième modèle, nous avons utilisé la même architecture de premier modèle et nous avons ajouté une couche de convolution ce qui permet d'extraire plus des caractéristiques d'une image avant de les transmettre à la partie classifieur et nous avons gardé le même nombre d'itération que le premier modèle.

- **la partie classifieur**

La partie dense de deuxième modèle est la même avec le premier modèle, nous avons utilisé un perceptron de trois couches.

### V.3- Architecture du troisième modèle :

L'architecture du troisième modèle ressemble à l'architecture du deuxième modèle mais nous avons augmenté le nombre d'itérations.

## V- Analyse et discussion

Dans cette section nous allons présenter les résultats obtenus pour les trois modèles.

### VI.1- Le graphe de précision et le graphe d'erreur

Nous commençons par les graphes de précision et les graphes d'erreur de trois modèles créés qui sont illustré dans les figures ci-dessous :

- **résultat du premier modèle**

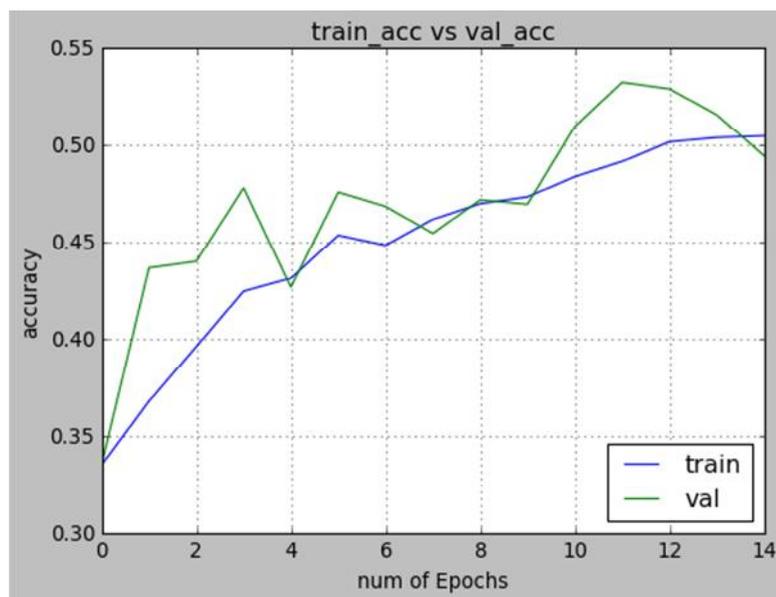
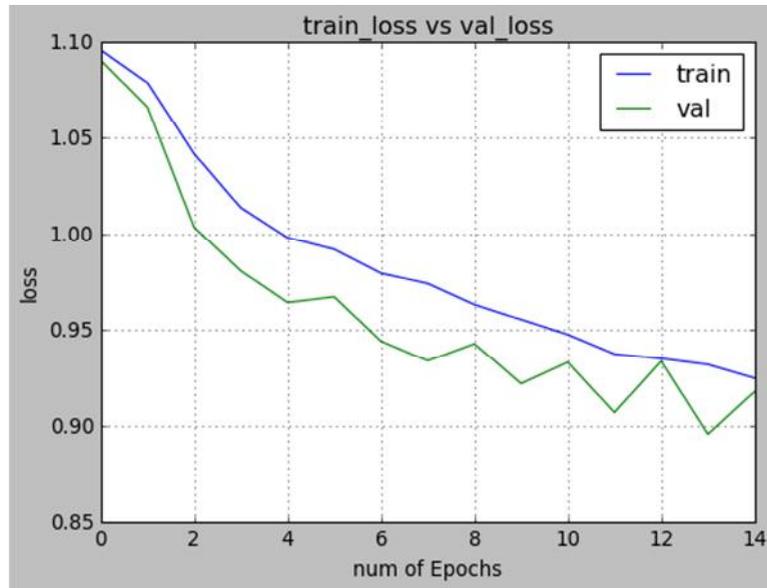


Figure 33 : Le graphe de décision du premier modèle.

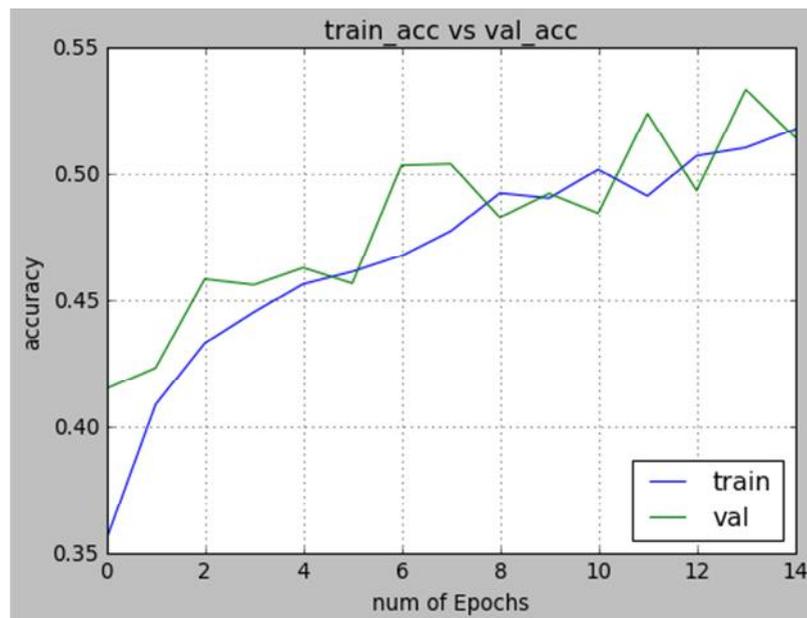


**Figure 34 : Le graphe d'erreur du premier modèle.**

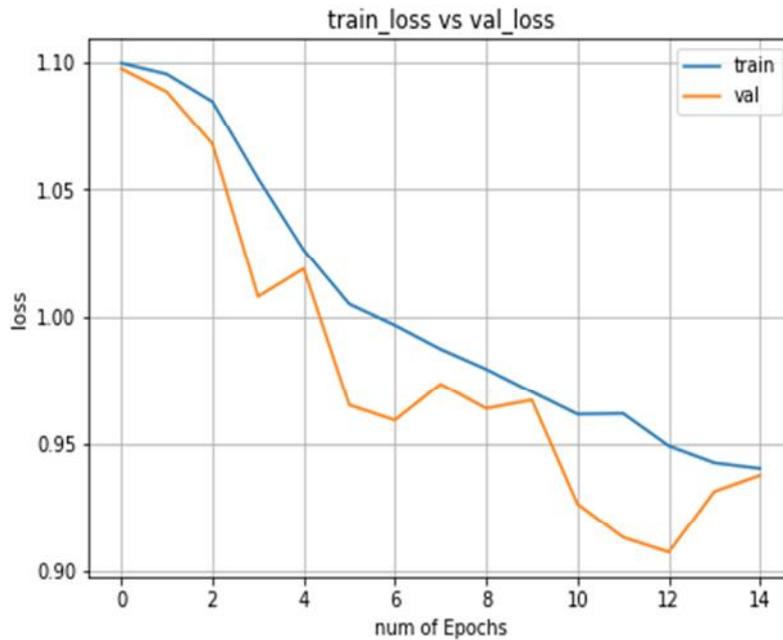
D'après la figure 33, nous constatons que la précision de notre modèle lors de l'entraînement et de test augmente à chaque itération, ceci explique qu'à chaque itération le modèle apprend plus d'information sur les images, pour mieux les classer et améliorer la précision ainsi que la performance de modèle.

D'autre part la figure 34, montre que l'erreur de classification diminue à chaque itération.

- **le résultat du deuxième modèle**



**Figure 35 : Le graphe de précision du deuxième modèle.**

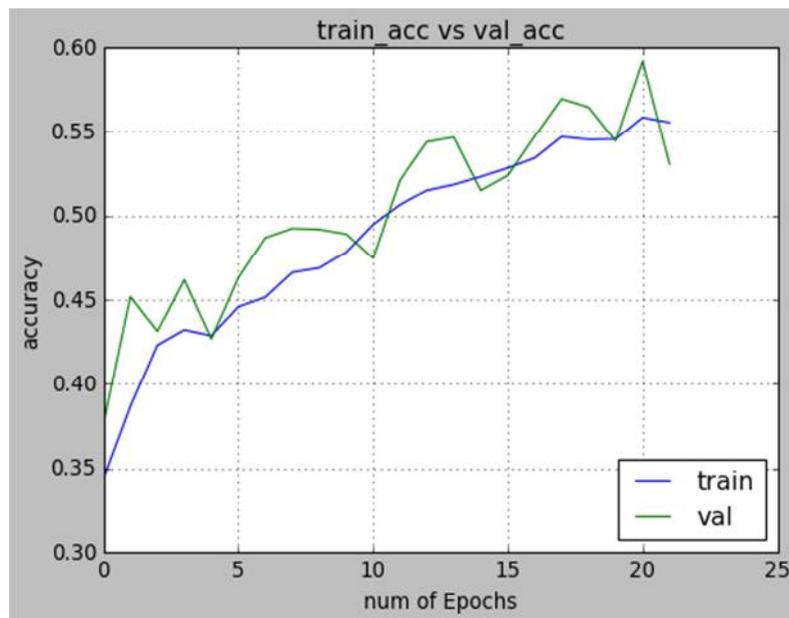


**Figure 36 : Le graphe d'erreur du deuxième modèle.**

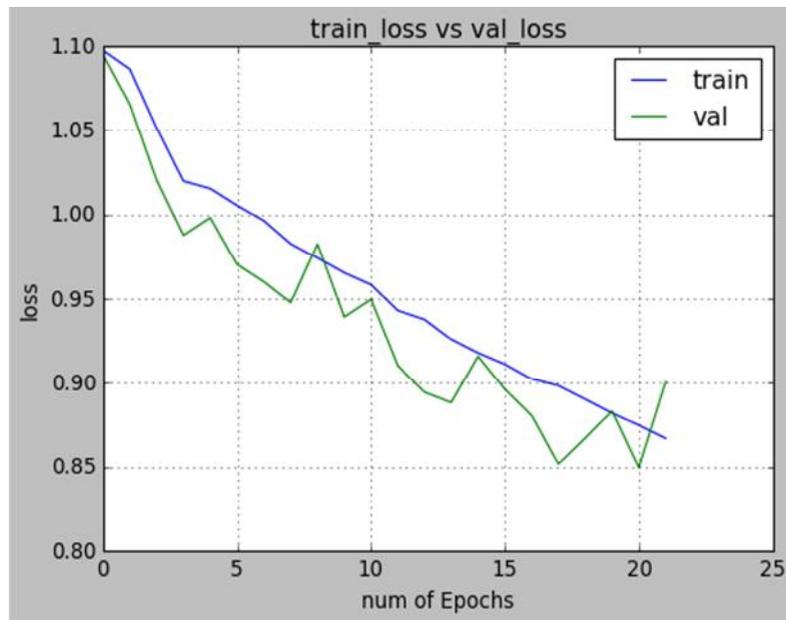
La figure 35, nous constatons que la précision de notre modèle lors de l'entraînement et de teste augmente chaque itération et la précision du deuxième modèle a augmenté par rapport au premier modèle.

La figure 36, montre que l'erreur de classification diminue avec le nombre d'itération, et le taux d'erreur du deuxième modèle est inférieur à celui du premier modèle.

- **Le résultat du troisième modèle**



**Figure 37 : La courbe de précision du troisième modèle.**



**Figure 38 : La courbe d'erreur du troisième modèle.**

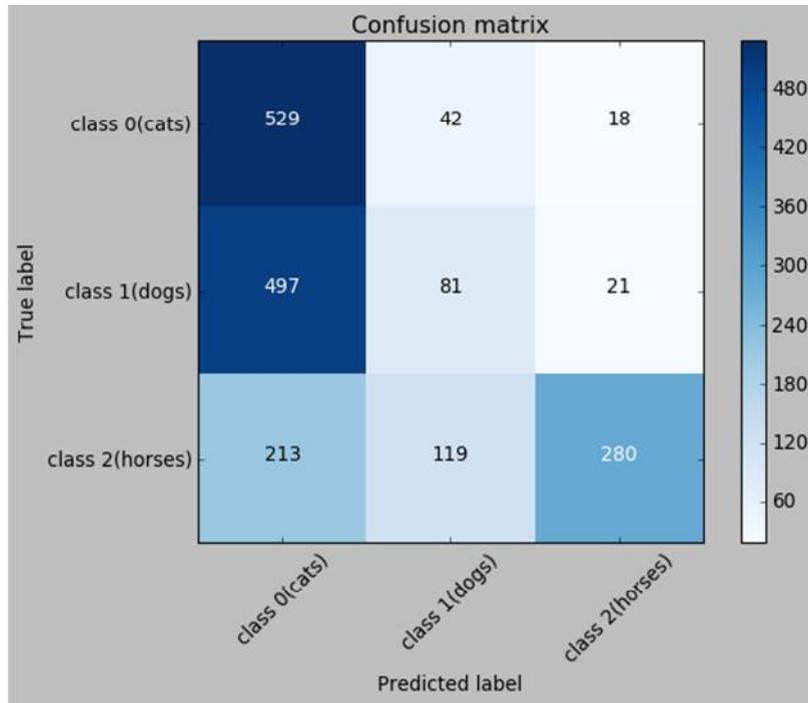
D'après la figure 37, nous constatons que la précision de modèle lors de l'entraînement et de teste augmente à chaque itération, et la précision du troisième modèle a augmenté par rapport au deuxième modèle.

La figure 38, montre que l'erreur de classification diminue avec le nombre d'itération, et le taux d'erreur du troisième modèle est inférieur au taux d'erreur du deuxième modèle.

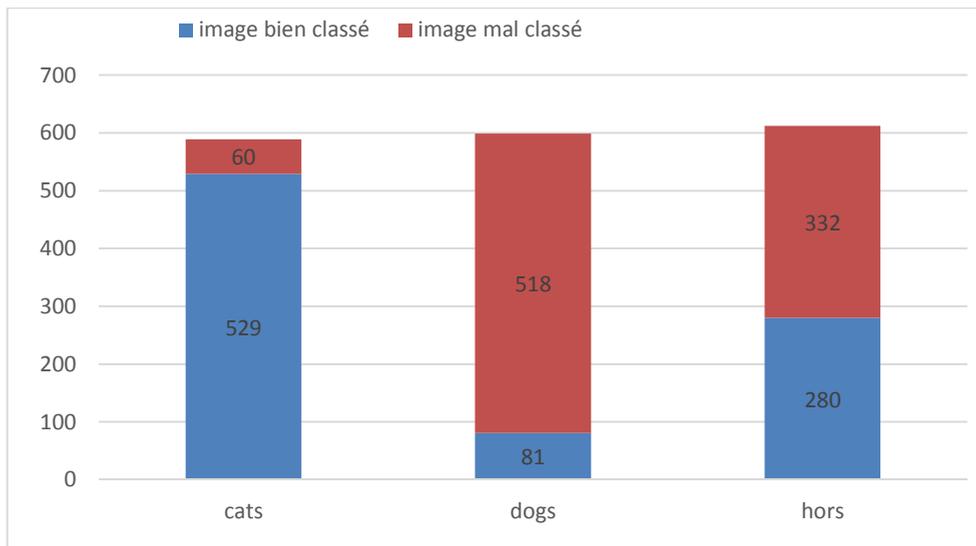
Donc pour améliorer la précision et minimiser l'erreur de classification dans le modèle, on augmente le nombre d'itérations.

**VI.2- La matrice de confusion**

- premier modèle



**Figure 39 : Matrice de confusion du premier modèle.**



**Figure 40 : Le nombre d’images bien et mal classé.**

La matrice d confusion de **la figure 39** résume les prédictions des images de test sur les trois classes :

Pour les cats : 529 images classées dans cats, 42 images classées dans dogs, 18 images classées dans horses.

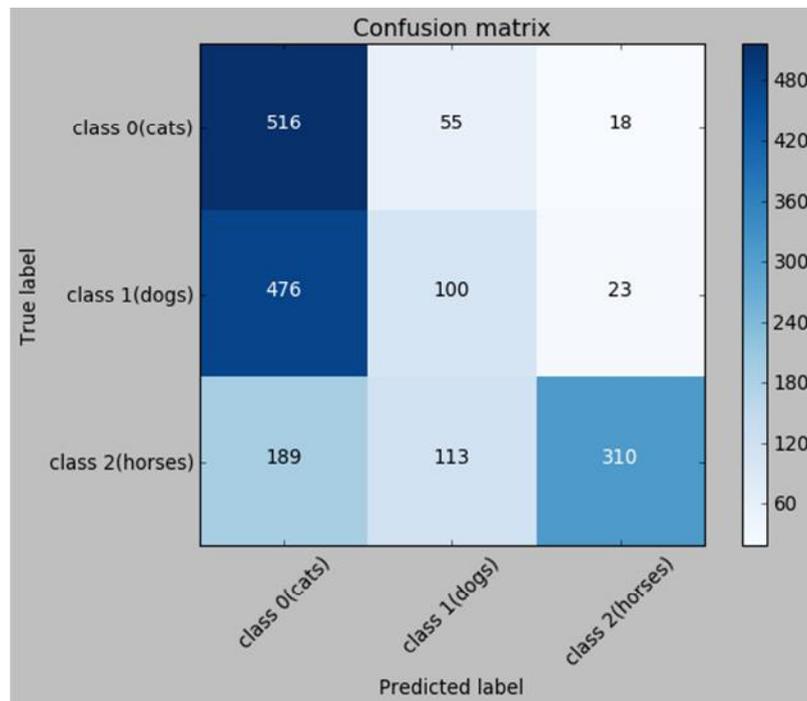
Pour les dogs : 497 images classées dans cats, 81 images classées dans dogs, 21 images classées dans horses.

Pour les horses : 213 images classées dans cats, 119 images classées dans dogs, 280 images classées dans horses.

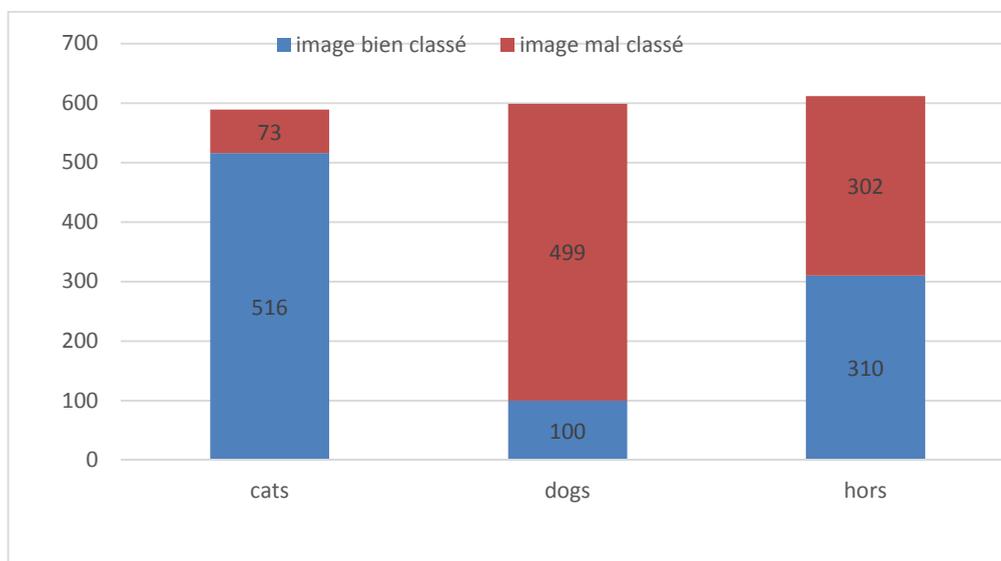
D'après **la figure 40** nous remarquons que la totalité des images mal classées est de 910 images, des images bien classées est de 890.

La matrice de confusion permet d'évaluer la performance de notre modèle, puisqu'elle reflète les métriques du Vrai positif, Vrai négatif, Faux positif et Faux négatif.

- **deuxième modèle**



**Figure 41 : La matrice de confusion du deuxième modèle.**



**Figure 42 : Nombre d'images bien et mal classé.**

La matrice de confusion **de la figure 41** résume la prédiction des images de test sur les trois classes :

Pour les cats : 916 images classées dans cats, 55 images classées dans dogs, 18 images classées dans horses.

Pour les dogs : 476 images classées dans cats, 100 images classées dans dogs, 31 images classées dans horses.

Pour les horses : 189 images classées dans cats, 113 images classées dans dogs, 310 images classées dans horses

D'après **la figure 42** nous remarquons que la totalité des images mal classées est de 874 images, et les images bien classées est de 926 et nous remarquons aussi que le deuxième modèle est mieux que premier modèle.

- troisième modèle

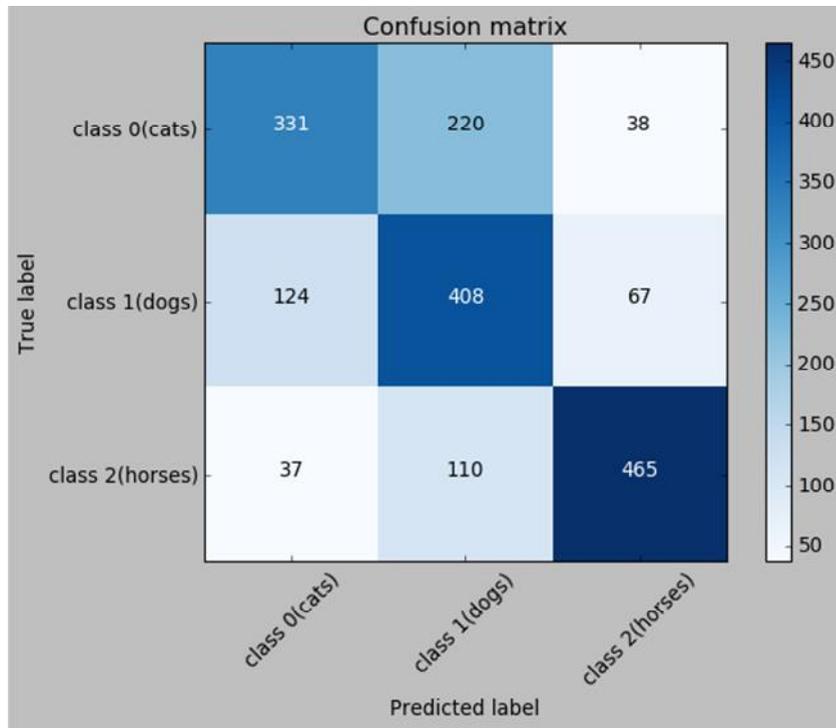


Figure 43 : La matrice de confusion du troisième modèle.

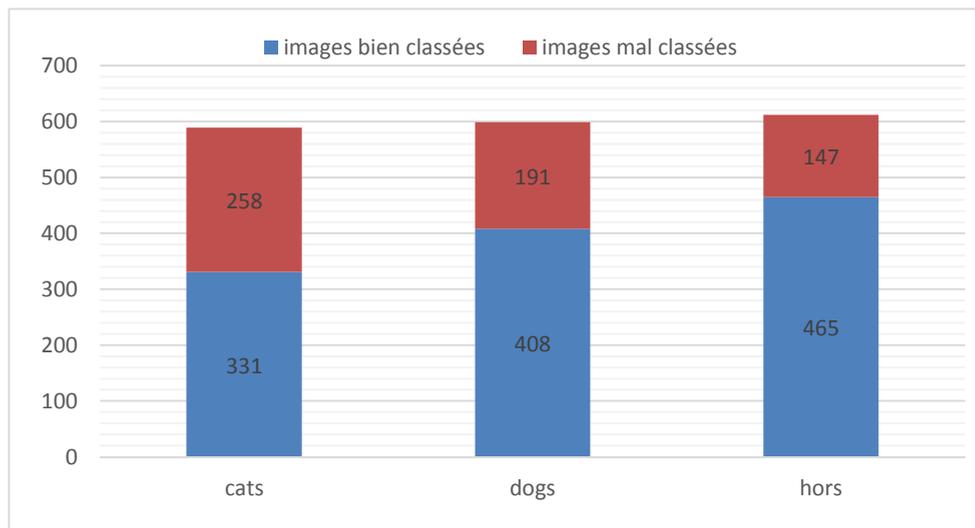


Figure 44 : Nombre d’images bien et mal classées.

La matrice de confusion de **la figure 43** résume les prédictions des images de test sur les trois classes :

Pour les cats : 331 images classées dans cats, 220 images classées dans dogs, 38 images classées dans horses.

Pour les dogs : 124 images classées dans cats, 408 images classées dans dogs, 67 images classées dans horses.

Pour les horses : 37 images classées dans cats, 110 images classées dans dogs, 465 images classées dans horses.

D'après la **figure 44** nous remarquons que la totalité des images mal classées est de 596 images, des images bien classées est de 1204 et nous remarquons aussi que le troisième modèle est mieux classé les images par rapport au deuxième modèle.

### VI.3- Tableau de comparaison de résultat obtenu :

modèle	Architecture utilisé			Nombre d'époque	Précision de la base train	Précision de la base de test	erreur	Temps d'exécution
	Nbr couche de convolution	Nbr de couche pooling	Nombre de couche dense					
<b>Premier modèle</b>	3	2	3	15	50.5%	48%	0.93	3225 Seconde
<b>Deuxième modèle</b>	4	2	3	15	52.44%	51,40%	0.927	4900 Seconde
<b>Troisième modèle</b>	4	2	3	20	57%	53,43%	0.87	7300 Seconde

**Figure 45 : Tableau de comparaison de résultat.**

Le tableau de la figure 45 résume les résultats obtenus pour les trois modèles créés. Chaque modèle est représenté par son architecture ainsi que le nombre d'itération utilisée et la précision lors de l'entraînement et de test ainsi l'erreur et le temps d'exécution nécessaire pour l'entraînement et le test. Le temps d'exécution est trop coûteux. Ceci revient à la grande dimension de la base ce qui nécessite l'utilisation d'un GPU<sup>(3)</sup> au lieu d'un CPU<sup>(4)</sup>. D'après les résultats décrits dans ce tableau, nous constatons que les résultats obtenus par le deuxième modèle (la précision de la base d'entraînement et la précision de la base de test et l'erreur) est meilleur que les résultats de premier vu qu'il possède de couche de convolutions. D'autre part, nous constatons que le troisième modèle a présenté les meilleurs résultats trouvés par rapport aux autres modèles. Le nombre d'itérations et de couches de convolution reflètent ces

(3) : Processeur graphique ou GPU (en anglais Graphics Processing Unit).

(4) : Un processeur (en anglais central processing unit, CPU).

bons résultats, cependant le temps d'exécution était très couteux (à cause du nombre d'itération utilisées).

D'une manière générale, le nombre de couche de convolution et le nombre d'itérations sont des facteurs primordiaux pour améliorer la performance d'un réseau de neurone convolutifs en termes de précision et d'erreur.

## VII. Code source de troisième modèle :

- 1- `nbr_epoch=1` #Nombre d'époque (nombre d'itération)
- 2- `border mode='sam` #le pas de convolution
- 3- `input_shape=img_BDD[0].sh` #L'entrée est une image de taille 200\*200 pixel
- 4- `pool_size=(2, 2)` #la taille de pool 2\*2 pixel
- 5- `nbr_classe=3` #nombre de classe est 4
- 6- `X_train.shap` #nombre d'image d'apprentissage est 7200 images
- 7- `X_test.shap` # nombre d'image de test est 1800 images
- 8- `model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=input_shape, activation='relu')`

Cette commande permet de créer 32 cartes de caractéristique on utilise des 32 filtres de taille 3\*3 pixel et un mode de bordure 'same' et une fonction d'activation 'RELU'

- 9- `model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=input_shape, activation='relu')`

Cette commande permet de créer une deuxième couche de convolution avec les mêmes caractéristiques que la première couche de convolution exécution.

10- `model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=input_shape, activation='relu'))`

Cette commande permet de créer une troisième couche de convolution avec les mêmes caractéristiques que la première couche de convolution exécutée.

11- `model.add(MaxPooling2D(pool_size=(2, 2)))`

Cette commande est utilisée pour réduire la taille d'une image ainsi la taille d'image d'entrée sera divisée par 2.

12- `model.add(Dropout(0.25))`

Pour ne pas tomber dans le problème de sur-apprentissage, il faut utiliser dropout elle est très efficace pour les réseaux de neurones pour le régulariser.

13- `model.add(Convolution2D(32, 3, 3, border_mode='same', input_shape=input_shape, activation='relu'))`

Cette commande permet de créer une troisième couche de convolution avec les mêmes caractéristiques que la première couche de convolution exécutée.

14- `model.add(MaxPooling2D(pool_size=(2, 2)))`

Cette commande est utilisée pour réduire la taille d'une image ainsi la taille d'image d'entrée sera divisée par 2.

15- `model.add(Dropout(0.25))`

Utiliser le dropout pour une autre fois pour régulariser le réseau.

16- `model.add(Flatten())`

Cette commande permet d'aplatir la matrice obtenue par la couche de pooling à un vecteur de 1 dimension puis connecter avec la première couche de perceptron pour commencer la classification.

17- `model.add(Dense(64), activation='relu')`

Cette commande permet de créer la première couche d'entrée de perceptron composée de 64 neurones et une fonction 'RELU' comme fonction d'activation.

17- `model.add(Dropout(0.5))`

Cette commande permet d'appliquer le dropout à la première couche de perceptron.

18- `model.add(Dense(64),`

Cette commande permet de créer une couche cachée composée de 64 neurones avec une fonction 'RELU' comme fonction d'activation.

19- `model.add(Dropout(0.5))`

Cette commande permet d'appliquer le dropout à la couche cachée de perceptron.

20- `model.add(Dense(nbr_classe), activation='softmax')`

Cette commande permet de créer la couche de sortie de perceptron composée de 4 neurones (nombre de classe utilisé) et une fonction de décision 'SOFTMAX' pour calculer la probabilité de chaque classe.

21- `model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['accuracy'])`

Cette commande permet de compiler le modèle créé et elle prend trois paramètres la fonction d'erreur qui permet de tracer le graphe d'erreur et l'optimiseur utilisé pour minimiser l'erreur de classification est le 'SGD' (Stochastic Gradient Descent) et le troisième paramètre c'est la métrique pour tracer le graphe de la précision.

22- `hist = model.fit(X_train, y_train, batch_size=16, nb_epoch=nbr_epoch, verbose=1, validation_data=(X_test, y_test))`

Cette commande permet de lancer l'apprentissage de modèle avec la base de train (X\_train, y\_train).

23- `score = model.evaluate(X_test, y_test, verbose=0)`

Cette commande permet d'évaluer le modèle avec la base de test (X\_test, y\_test).

## **Conclusion**

Dans cette partie nous avons décrit la démarche suivie pour construire un modèle de classification d'images le plus précis en se basant sur les réseaux de neurones convolutifs et l'apprentissage automatique, et pour cela nous avons défini trois modèles et chaque modèle sa propre architecture et ces propres paramètres, qui nous permettent d'obtenir différents résultats. La comparaison de ces résultats obtenus à partir de ces trois modèles, nous a aidés à obtenir l'architecture et les paramètres les plus appropriés pour construire un modèle de classification puissant et performant en termes de précision et d'erreur.

## **Conclusion générale**

Dans ce projet nous avons discuté des notions fondamentales des réseaux de neurones en générale et des réseaux de neurones convolutionnels en particulier et la contribution de ces derniers pour les taches de classification d'image.

Nous avons introduit ces réseaux de neurones convolutionnels en présentant les différents types de couches utilisées dans la classification : la couche convolutionnelle, la couche de rectification, la couche de pooling et la couche entièrement connectée.

Les paramètres du réseau sont difficiles à définir a priori. C'est pour cette raison que nous avons défini différents modèles avec des architectures différentes afin d'obtenir des meilleurs résultats en terme de précision et d'erreur.

Nous avons rencontré quelques problèmes dans la phase d'implémentation, l'utilisation d'un CPU a fait que le temps d'exécution était trop couteux. Afin de régler ce problème on doit utiliser des réseaux de neurones convolutionnels plus profonds déployé sur un GPU au lieu d'un CPU sur des bases plus importantes.

## **Bibliographie :**

[1] : François-Xavier Jollois. Contribution de la classification automatique à la Fouille de Données. PhD thesis, Université de Metz, 2003.

[2] : Djamel Abdelkader ZIGHED, Ricco RAKOTOMALALA, Extraction de connaissances à partir de données (ECD).

[3] : Sandrine Curtet. Modules de prétraitement de données dans le cadre du Data Mining, Haute école d'ingénierie et de Gestion du Canton de Vaud.

[4] : BENAMAR HOUMADI, étude exploratoire d'outils pour le data mining, université du québec.

[5] : Séraphin LOHAMBA OMATOKO, université Notre Dame de kasayi, Analyse et détection de l'attrition dans une entreprise.

[6] : Nazih Selmoune, Data mining.

[7] : Guillaume Cleuziou, Une méthode de classification non-supervisée pour l'apprentissage de règles et la recherche d'information.

[8] : Anne-sophie Bellanger-dujardin, contribution à l'études de structures neuronales pour la classification des signatures.

[9] : Andrei Doncescu, Les réseaux de neurones artificiels.

[10] : C.Houassine, segmentation d'images par une approche biomimétique hybride. Universite m'hamed bougara- boumerdes.

[11] : M .T.Chikh, amélioration des images par un modèle de réseau de neurones (comparaison avec les filtres de base), université abou-bakr belkaid tlemcen.2011

[12] : Statistica, les réseaux de neurones.

[13] : OpenClassrooms, sur les réseaux de neurones convolutionnelles.

[14] : GAMBRIGGE SPARK, Deep learning for complete beginners: convolutional neural networks with keras.

[15] : Sebastien Gambs, survol de l'apprentissage machine.

[16] : A.cornuéjols LRI, évaluation de l'apprentissage.

[17] : M. Gregory Gelly Réseaux de neurones récurrents pour le traitement automatique de la parole.

## Annexe :

### 1. Système décisionnel :

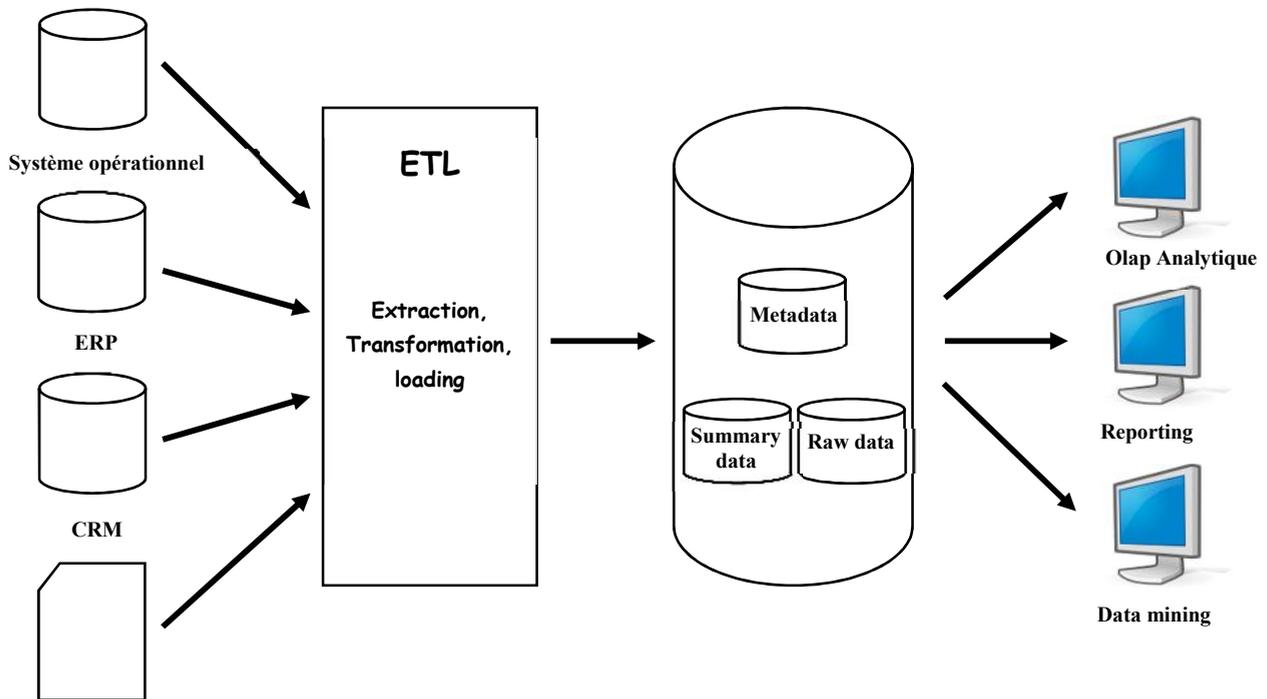


Figure A.1: schema d'un Data Warehouse

#### Orienté sujet :

Un data warehouse rassemble et organise des données associées aux différentes structures fonctionnelles de l'entreprise, pertinentes pour un sujet nécessaire aux besoins d'analyse.

#### Intégré :

Assurer la cohérence des données proviennent de plusieurs sources différentes afin de les assembler correctement dans le data warehouse.

#### Non volatiles :

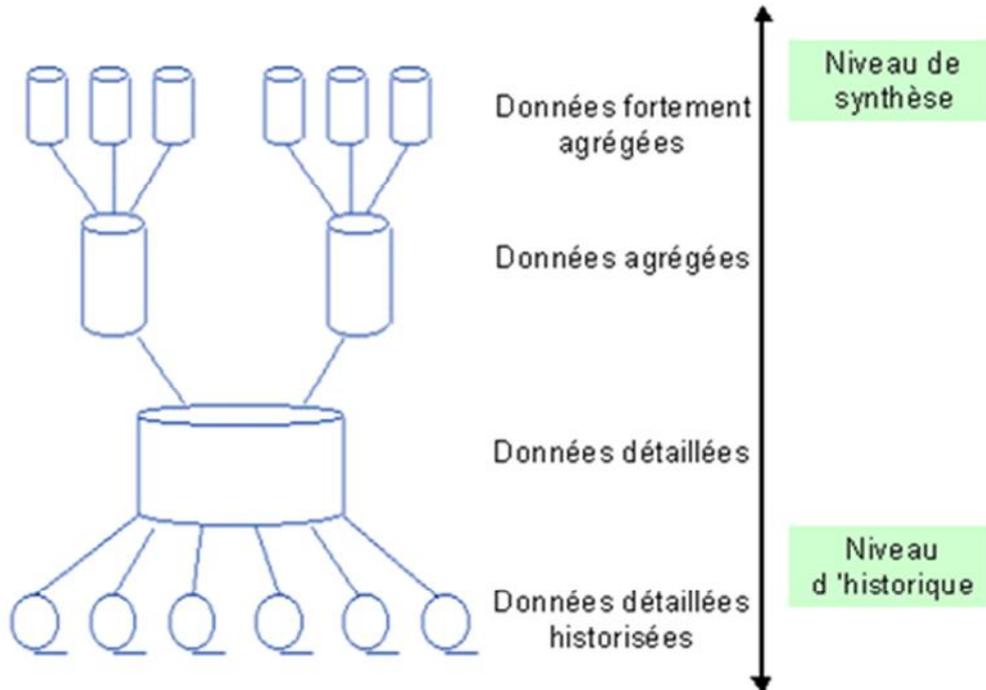
Les données sont essentiellement utilisées en interrogation (consultation) et ne peuvent pas être modifiées (sauf certains cas de rafraîchissement).

#### historisée :

Les données sont archivées et datées, pour pouvoir visualiser leurs évaluations dans le temps.

## 2. Structure de données d'un Data Warehouse :

Le Data Warehouse a une structure bien définie, selon différents niveaux d'agrégation et de détail des données.



**Figure A.2 : Structure des données d'un Data Warehouse.**

**Données détaillées :** Ce sont les données qui reflètent les événements les plus récents, fréquemment consultées, généralement volumineuses car elles sont d'un niveau détaillé.

**Données détaillées archivées :** Anciennes données rarement sollicitées, généralement stockées dans un disque de stockage de masse, peu coûteux, à un même niveau de détail que les données détaillées.

**Données agrégées :** Données agrégées à partir des données détaillées.

**Données fortement agrégées :** Données agrégées à partir des données détaillées, à un niveau d'agrégation plus élevé que les données agrégées.

### 3. Domaine de l'apprentissage automatique :

L'apprentissage automatique est un champ d'études de la fouille de données.

- La fouille de donnée (data mining) : Est le processus d'extraction des connaissances, il consiste à sélectionner les données à étudier à partir des bases de données (hétérogène ou homogène) à fin de l'utiliser en apprentissage pour construire un modèle.

#### Exemple :

Trouver une prescription pour un malade à travers de ces fichiers médicaux antérieurs.

- l'intelligence artificielle : La vision par ordinateur, la robotique, l'analyse et la compréhension des images, la reconnaissance des formes.

#### Exemple :

Système de vidéo surveillance pour la détection des intrus.

Logicielle biométrique de reconnaissance de visage et d'empreintes digitales.

### 4. Données d'apprentissage :

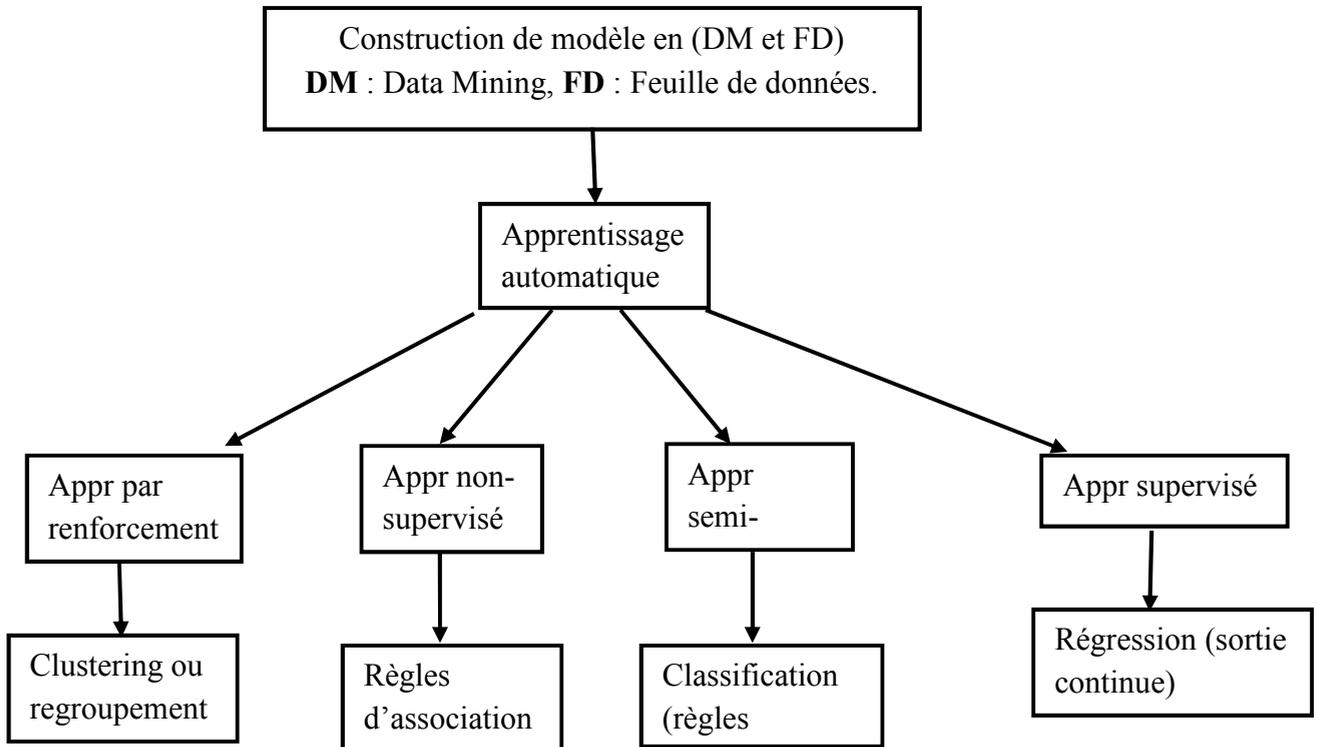
Les données d'apprentissages sont souvent regroupées :

- L'ensemble d'apprentissage ou l'ensemble d'entraînement : Les données initiales utilisées pour construire ou extraire un modèle (image, attribut, bd),
- L'ensemble de test : est constitué des données sur lesquelles sera appliqué le modèle d'apprentissage (pour tester et corriger l'algorithme),
- L'ensemble de validation : est utiliser lors de l'apprentissage à fin d'intégrer et valider le modèle obtenue.

### 5. Type d'apprentissage :

Une fois les données acquises du terrain ou du réel (images, vidéos, fichiers BD) sous forme de données brutes, il faut les analyser et en extraire de l'information sous forme de nouvelles connaissances, de descripteurs ou de vecteur de caractéristiques.

Les différentes techniques de construction de modèles, elles sont récapitulées sur le schéma de la Figure suivant :



**Figure A.3 : Types d'apprentissage.**

### **Apprentissage par Renforcement :**

Les données en entrée sont les mêmes que pour l'apprentissage supervisé, cependant l'apprentissage est guidé par l'environnement sous la forme de récompenses ou de pénalités données en fonction de l'erreur commise lors de l'apprentissage.

### **Apprentissage Semi-Supervisé :**

Les données d'entrée sont constituées d'exemples étiquetés et non étiquetés. Ce qui peut être très utile quand on a deux types de données, car cela permet de ne pas en laisser de côté et d'utiliser toute l'information.

## 6. Les types de cluster :

On trouve qu'il existe différents types de clustering et parmi eux :

A/ le clustering hiérarchique : d'agglomération («bottom-up»)

B/ le clustering non-hiérarchique : de division («top-down»)

**-Le clustering hiérarchique :** Consiste à créer une arborescence de groupes.

-la racine de l'arbre est formée par le cluster  $X$  contenant l'ensemble des objets

-chaque nœud de l'arbre constitue un cluster  $c_l$  qui appartient à  $X$

-les feuilles de l'arbre correspondent aux objets

-l'union des objets contenues dans les fils donné correspond aux objets présents dans ce nœud.

**Le clustering non-hiérarchique :** Dans le clustering non-hiérarchique on décompose l'ensemble d'individus en  $K$  groupes, les algorithmes de ce type peuvent aussi être utilisés comme algorithmes de division dans le clustering hiérarchique.

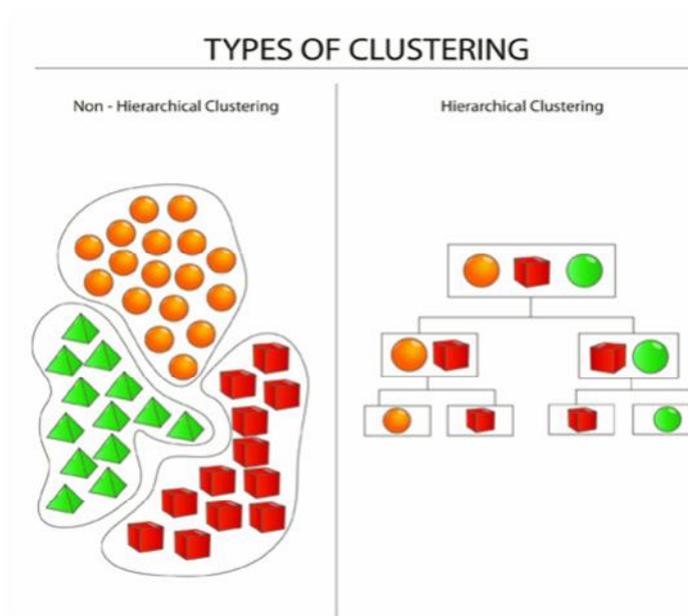


Figure A.4 : Les deux types de clustering non-hiérarchique/hiérarchique.

## 7. Algorithme de clustering :

Dans cette partie nous allons présenter quelque algorithme utilisés dans la classification non supervisé et chacun de ces algorithmes appartient à l'un des types de clustering énumérés ci-dessus et parmi ces algorithmes :

### K-means

#### L'algorithme de k-means :

L'algorithme k-means est utilisé dans la classification non supervisée lorsque les données d'apprentissage ne sont pas étiquetées (les données d'entraînement ne sont pas catégorisées), l'objectif de cet algorithme est de regrouper les données en  $k$  cluster et le nombre de groupes représentés par la variable  $K$ . L'algorithme fonctionne itérativement pour affecter chaque point de données à l'un des  $K$  groupes

L'idée principale de l'algorithme k-means consiste à :

- définir Les centroïdes des groupes  $K$ , qui peuvent être utilisés pour étiqueter de nouvelles données
- Etiquettes pour les données d'apprentissage (chaque point de données est affecté à un seul cluster).

Lorsqu'aucun point n'est pas en attente, la première étape est terminée et un groupage précoce est fait. À ce point nous avons besoin de recalculer les  $k$  nouveaux centroïdes mi des groupes issus de l'étape précédente qui vont remplacer les  $c_i$  ( $m_j$  est le centre de gravité de la classe  $S_j$ , calculé en utilisant les nouvelles classes obtenues). Après, on réitère Le processus jusqu'à atteindre un état de stabilité où aucune amélioration n'est possible, nous pouvons constater que les  $k$  centroïdes changent leur localisation par étape jusqu'à plus de changements sont effectués. En d'autres termes les centroïdes ne bougent plus.

#### Algorithme :

*Choisir  $k$  moyennes  $c_1, c_2, \dots, c_k$  initiales (par exp au hasard)*

*1. Répéter :*

*affectation de chaque point à son cluster le plus proche :*

$$S(t) i = \arg \min_j \|x_i - m_j(t)\| \quad \text{for all } i = 1, \dots, n \quad (2.1)$$

*mettre à jour la moyenne de chaque cluster*

$$m_j(t+1) = \frac{1}{|S(t) i|} \sum_{i \in S(t) i} x_i \quad (2.2)$$

*Jusqu'à : atteindre la convergence quand il n'y a plus de changement.*

*Fin.*

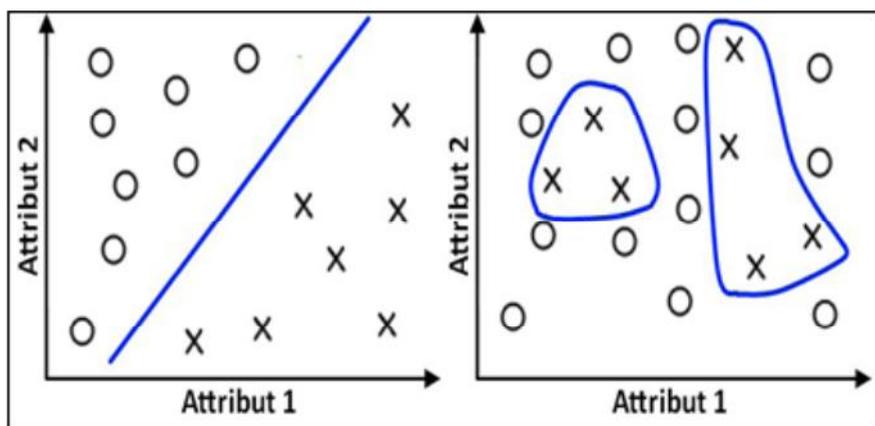
## Problème Linéaire et Non-Linéaire

Les méthodes de classification supervisée peuvent être basées sur

- des hypothèses probabilistes (cas du classifieur naïf bayésien)
- des notions de proximité (exemple,  $k$  plus proches voisins)
- des recherches dans des espaces d'hypothèses (exemple, arbres de décisions), En fonction du problème, il faut pouvoir choisir le classifieur approprié, c'est-à-dire celui qui sera à même de séparer au mieux les données d'apprentissage.

On dit qu'un problème est linéairement séparable si les exemples de classes différentes sont complètement séparables par un hyperplan (appelé hyperplan séparateur, ou séparatrice). Ce genre de problème se résout par des classifieurs assez simples, qui ont pour but de trouver l'équation de l'hyperplan séparateur.

Mais, le problème peut également être non séparable de manière linéaire comme illustré dans la figure A.5. Dans ce cas, il faut utiliser d'autres types de classifieurs, souvent plus longs à paramétrer, mais qui obtiennent des résultats plus précis.



**Figure A.5 A Gauche : Problème linéairement séparable (Frontière linéaire)  
A Droite : Problème non linéairement séparable.**

## Classifieurs à mémoire

L'intérêt de ces classifieurs est qu'ils ne nécessitent aucune phase d'apprentissage ou d'entraînement. Ainsi, ils permettent de déduire directement la classe d'un nouvel exemple à partir de l'ensemble d'apprentissage.