

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université 8 Mai 1945 – Guelma

Faculté des Sciences et de la Technologie

Département de Génie Electrotechnique et Automatique

Réf:...../2020



MEMOIRE

Présenté pour l'obtention du **diplôme de MASTER Académique**

Domaine : Sciences et Technologie

Filière : Automatique

Spécialité : Automatique et Informatique Industrielle

Par : Bourougaa Nour El Islam et Seridi Merouane

Thème

Application des réseaux de neurone pour la classification des Données

Soutenu publiquement le 10 / 10 / 2020 devant le jury composé de :

| | | | |
|-----------------------|------------|--------------|-----------|
| M Moussaoui Abdelkrim | Professeur | Univ.Guelma | Président |
| M Debeche Mehdi | MAA | Univ. Guelma | Encadreur |
| M Sebbagh Abdennour | MCB | Univ. Guelma | Examineur |

Résumé :

L'objectif de ce mémoire, est d'explorer la technique de classification et les algorithmes fondamentaux largement utilisés par celle-ci, en se concentrons beaucoup plus sur les applications de cette technique et de son impact dans le monde actuel, ainsi qu'aux réseaux de neurones artificiels, leurs concepts de base et leurs apprentissages.

Notre approche pratique sera l'implémentation de l'algorithme de réseau de neurones artificiels pour classifier les fleurs d'Iris.

Mots clés : Intelligence artificielle , apprentissage automatique ,classification, réseaux de neurone , base de donnée Iris.

Summary :

The aim of this thesis is to explore the classification technique and the basic algorithms widely used by it, focusing more on the applications and impact of this technology in the current world, as well as on artificial neural networks, their basic concepts and their learning. Our practical approach would be to apply an artificial neural network algorithm to classify the iris flowers.

Keywords : Artificial intelligence, machine learning, classification, neuron networks , Iris database.

ملخص :

الهدف من هذه المذكرة هو استكشاف تقنية التصنيف والخوارزميات الأساسية المستخدمة على نطاق واسع من قبلها ، مع التركيز أكثر على تطبيقات هذه التقنية وتأثيرها في العالم الحالي ، وكذلك على الشبكات العصبية الاصطناعية ، مفاهيمها الأساسية وتعلمهما .

سيكون منهجنا العملي هو تطبيق خوارزمية الشبكة العصبية الاصطناعية لتصنيف زهور السوسن .

الكلمات الدالة :

الذكاء الاصطناعي ، التعلم الآلي ، التصنيف ، الشبكات العصبية ، قاعدة بيانات زهور السوسن . Iris

Remerciement

Au début, nous remercions Allah qui nous a aidés à accomplir ce travail, et qui a été avec nous à tous les moments de notre chemin d'étude. A nos chers parents pour leur patience, leur amour, leur soutien et leurs encouragements.

Nos sincères remerciements à tous les membres du jury qui nous ont fait l'honneur d'examiner ce travail.

*Un remerciement spécial à notre encadreur **Mr.Debeche Mehdi** pour sa patience, et ses conseils judicieux pour élaborer ce travail.*

Tables des matières

Introduction générale

| | | |
|----------|--|-----------|
| 0.1 | Intelligence artificiel et apprentissage automatique..... | 8 |
| 0.2 | Objectifs de mémoire..... | 10 |
| 0.3 | DESCRIPTION DES CHAPITRES | 10 |
| 1 | GENERALITES SUR LA CLASSIFICATION | 11 |
| 1.1.1 | INTRODUCTION | 12 |
| 1.2 | DEFINITION..... | 12 |
| 1.3 | DEFINITION FORMELLE..... | 13 |
| 1.4 | TERMINOLOGIE..... | 14 |
| 1.5 | EXEMPLES DE PROBLEMES DE CLASSIFICATION | 14 |
| 1.6 | DOMAINES D'APPLICATION | 17 |
| 1.7 | METHODES DE CLASSIFICATION..... | 17 |
| 1.7.1 | CLASSIFICATION NON SUPERVISE..... | 18 |
| 1.7.2 | Définition..... | 18 |
| 1.7.3 | EXEMPLE | 19 |
| 1.7.4 | ALGORITHMES..... | 20 |
| 1.7.5 | Exemples d'application | 24 |
| 1.8 | LA CLASSIFICATION SUPERVISEE | 24 |
| 1.8.1 | Définition | 25 |
| 1.8.2 | EXEMPLE | 25 |
| 1.8.3 | ALGORITHMES..... | 26 |
| 1.8.4 | EXEMPLE D'APPLICATION..... | 31 |
| 1.9 | INDICATEURS DE PERFORMANCE D'UN MODELE DE CLASSIFICATION | 32 |
| 1.9.1 | La matrice de confusion | 32 |
| 1.9.2 | La courbe ROC (receiver operating characteristic)..... | 33 |
| 1.10 | CONCLUSION | 34 |
| 2 | LES RESEAUX DE NEURONES..... | 35 |
| 2.1 | INTRODUCTION | 36 |
| 2.2 | CHRONOLOGIE | 36 |
| 2.3 | NEURONE BIOLOGIQUE | 37 |
| 2.3.1 | Le corps cellulaire | 37 |
| 2.3.2 | Les dendrites | 37 |
| 2.3.3 | L'axone..... | 38 |
| 2.3.4 | Les synapses..... | 38 |
| 2.4 | NEURONE ARTIFICIEL | 38 |
| 2.4.1 | Le poids | 39 |
| 2.4.2 | Le biais | 39 |
| 2.4.3 | La fonction d'activation | 40 |
| 2.5 | LES RESEAUX LES PLUS CELEBRES | 40 |
| 2.5.1 | Le perceptron | 41 |

| | | |
|----------|--|-----------|
| 2.5.2 | Algorithme du perceptron | 41 |
| 2.5.3 | Réseau ADALINE (Adaptive Linear Neuron)..... | 41 |
| 2.6 | LE PERCEPTRON MULTICOUCHE (MPC)..... | 42 |
| 2.7 | RESEAUX A FONCTIONS RADIALES DE BASE | 42 |
| 2.8 | RESEAUX DE NEURONES RECURRENTS..... | 43 |
| 2.8.1 | Les réseaux de mémoire à long terme à court terme (LSTM) | 44 |
| 2.9 | LES RESEAUX DE NEURONES A CONVOLUTION | 45 |
| 2.9.1 | Architecture d'un CNN | 45 |
| 2.10 | CONCLUSION | 46 |
| 3 | L'APPRENTISSAGE..... | 47 |
| 3.1 | INTRODUCTION | 48 |
| 3.2 | DEFINITION D'APPRENTISSAGE DU RESEAU DE NEURONE | 48 |
| 3.3 | STRATEGIE D'APPRENTISSAGE :..... | 49 |
| 3.4 | TYPES D'APPRENTISSAGE | 49 |
| 3.4.1 | L'apprentissage supervisé | 49 |
| 3.4.2 | L'apprentissage non-supervisé | 49 |
| 3.5 | RESEAU DE NEURONE APPLIQUE A NOTRE CAS (PERCEPTRON MULTICOUCHE) | 50 |
| 3.6 | APPRENDRE AVEC LE RETRO-PROPAGATION DU GRADIENT | 52 |
| 3.6.1 | Descente de gradient..... | 52 |
| 3.6.2 | Rétro propagation du gradient..... | 53 |
| 3.7 | CONCLUSION | 57 |
| 4 | APPLICATION..... | 58 |
| 4.1 | INTRODUCTION | 59 |
| 4.2 | LA DESCRIPTION DU PROBLEME | 59 |
| 4.3 | LOGICIELS | 61 |
| 4.4 | CONFIGURATION DES DONNEES | 61 |
| 4.4.1 | Meas | 61 |
| 4.4.2 | Species | 61 |
| 4.5 | LES ENSEMBLES DE CREATION D'UN MODELE DE CLASSIFICATION PAR UN RESEAU DE NEURONE | 62 |
| 4.5.1 | Training set | 62 |
| 4.5.2 | Validation set | 62 |
| 4.5.3 | Test set | 62 |
| 4.6 | RESULTATS EXPERIMENTALE ET DISCUSSION | 62 |
| 4.6.1 | La première proposition | 63 |
| 4.6.1.1 | La fonction de rétropropagation de Levenberg–Marquardt (trainlm) | 63 |
| 4.6.1.2 | La fonction de rétropropagation du Gradient conjugué mis à l'échelle (trainscg) | 63 |
| 4.6.1.3 | Simulation1 | 64 |
| 4.6.1.4 | Simulation 2 | 65 |
| 4.6.1.5 | Discussion et Comparaison des résultats | 66 |
| 4.6.2 | La deuxième proposition | 66 |
| 4.6.2.1 | Simulation 3 | 67 |
| 4.6.2.2 | Simulation 4 | 68 |
| 4.6.2.3 | Simulation 5 | 69 |
| 4.6.2.4 | Simulation 6 | 70 |
| 4.6.2.5 | Discussion et comparaison des résultats | 71 |
| 4.7 | CONCLUSION | 71 |
| | CONCLUSION GENERAL..... | 71 |
| | BIBLIOGRAPHIE | 71 |

Listes des figures

| | |
|--|----|
| Figure 0.1 - Apprentissage automatique et l'intelligence artificielle | 8 |
| Figure 0.2 - système d'apprentissage automatique | 9 |
| Figure 1.1 - Principe de classification [33]..... | 13 |
| Figure 1.2 - Représentation des emails spam et non-spam | 14 |
| Figure 1.3 - Reconnaissance des chiffres manuscrite | 15 |
| Figure 1.4 - Reconnaissance d'empreintes digitales | 15 |
| Figure 1.5 - Reconnaissance vocale..... | 15 |
| Figure 1.6 - Diagnostic médical automatique..... | 16 |
| Figure 1.7 - Classification à partir d'une image satellite | 16 |
| Figure 1.8 - Classification non supervisé [33]..... | 18 |
| Figure 1.9.a - Représentation d'un cluster | 19 |
| Figure 1.9.b - Image d'un cluster [35]..... | 19 |
| Figure 1.10 - Aperçu de l'algorithme K-means | 21 |
| Figure 1.11 - Exemple des classes floues..... | 22 |
| Figure 1.12 - Regroupement en hiérarchie | 24 |
| Figure 1.13 - Partitionnement d'une hiérarchie | 24 |
| Figure 1.14 - Classification non supervisé | 25 |
| Figure 1.15 - Schéma d'une classification par la méthode k plus proche voisin..... | 27 |
| Figure 1.16 - Représentation des données linéairement séparables et non linéairement séparables..... | 27 |
| Figure 1.17 - Séparateur SVM..... | 28 |
| Figure 1.18 - La marge séparatrice | 28 |
| Figure 1.19 - Hyperplans possibles..... | 29 |
| Figure 1.20 - Neurone biologique et neurone artificiel..... | 29 |
| Figure 1.21 - représentation schématique du Perceptron | 30 |
| Figure 1.22 - Données exemple..... | 30 |
| Figure 1.23 - Classification linéaire par perceptron..... | 31 |
| Figure 1.24 - La courbe ROC | 33 |

| | |
|---|----|
| Figure 2.1 - Neurone biologique [35] | 38 |
| Figure 2.2 - Schéma d'un Neurone artificiel | 39 |
| Figure 2.4 - Illustration d'un réseau de neurones à fonctions de base radiale (RBF) [37] | 43 |
| Figure 2.5 - Un exemple d'un réseau récurrent qui se déroule [38]..... | 43 |
| Figure 2.6 - Structure réseau de neurones récurrent | 44 |
| Figure 2.7 - Une chaîne de cellule LSTM [38]..... | 45 |
| Figure 2.8 - Architecture standard d'un réseau de neurone à convolution CNN [25]..... | 46 |
| | |
| Figure 3.1 - Représentation graphique d'un neurone simple | 50 |
| Figure 3.2 - Exemple d'un réseau de neurone multicouche..... | 51 |
| Figure 3.3 - Représentation de la minimisation de la fonction d'erreur par descente de gradient | 53 |
| Figure 3.4 - Schéma global de la méthode de rétro propagation [41] | 57 |
| | |
| Figure 4.1 - Les trois types de fleur iris [39] | 60 |
| Figure 4.2 - Nuage de points de la base de données iris [40]..... | 60 |
| Figure 4.4 - La matrice de confusion de la simulation | 64 |
| Figure 4.5 - La courbe ROC pour la simulation1 | 64 |
| Figure 4.6 - Représentation des classe réelle et prédite en 3D de la simulation1 | 64 |
| Figure 4.7 - La matrice de confusion de la simulation 2..... | 65 |
| Figure 4.8 - la courbe ROC pour la simulation 2..... | 65 |
| Figure 4.9 - Représentation des classes réelle et prédite en 3D de la simulation 2 | 65 |
| Figure 4.10 - La matrice de confusion de la simulation 3..... | 67 |
| Figure 4.11 - La courbe ROC pour la simulation 3..... | 67 |
| Figure 4.12 - Représentation des classes réelle et prédite en 3D de la simulation 3..... | 67 |
| Figure 4.13 - La matrice de confusion de la simulation 4..... | 68 |
| Figure 4.14 - La courbe ROC pour la simulation 4..... | 68 |
| Figure 4.15 - Représentation des classes réelle et prédite en 3D de la simulation 4..... | 68 |
| Figure 4.16 - La matrice de confusion de la simulation 5..... | 69 |
| Figure 4.17 - La courbe ROC pour la simulation 5..... | 69 |
| Figure 4.18 - Représentation des classes réelles et prédite en 3D de la simulation5..... | 69 |
| Figure 4.19 - La matrice de confusion de la simulation 6..... | 70 |
| Figure 4.20 - La courbe ROC pour la simulation 6..... | 70 |
| Figure 4.21 - Représentation des classe réelle et prédite en 3D de la simulation 6..... | 70 |

Liste des tables

| | |
|--|----|
| Table 1.1 Exemple des notes élèves | 19 |
| Table 1.2 Différents tailles de tige et de pétale des fleurs..... | 25 |
| Table 1.3 matrice de confusion | 32 |
| Table 2.1 Correspondance neurone biologique/neurone artificiel | 39 |
| Table 2.2 Différents types de fonctions d'activations..... | 40 |
| Table 4.1 Informations générales sur les caractéristiques d'entrée | 60 |
| Table 4.2 Tableau des résultats de la première proposition..... | 66 |
| Table 4.3 Tableau des résultats de la deuxième proposition..... | 71 |

Acronyme

ADALINE : Adaptive Linear Neuron.

MPC : Multilayer Perceptron.

RBF : Radial Basis Functions.

RNN : Recurrent Neural Network.

LSTM : Long Short Term Memory.

CNN : Convolutional Neural Network.

K-NN : Nearest Neighbors.

SVM : Support Vector Machine.

Introduction générale

INTRODUCTION GENERALE

0.1 Intelligence artificiel et apprentissage automatique :

Le domaine de l'intelligence artificielle connaît de nos jours une évolution majeure. Il est en train de bouleverser plusieurs domaines, tel que la santé, le transport, le multimédia, l'industrie automobile avec les voitures autonomes, l'industrie aérospatiale, l'industrie biomécanique, les secteurs de finance, d'économie et de marketing, l'éducation, et même le domaine de la justice est touché par l'intelligence artificiel ou un jour les logiciels ou les robots pourront remplacer les juges et les avocats.

Comme son nom l'indique l'intelligence artificielle est une intelligence acquise par des machines dans le but de reproduire les comportements des humains. L'apprentissage automatique (appelé Machine Learning en anglais), sous-ensemble de l'Intelligence Artificielle, son principal objectif est d'offrir des techniques pour développer cette intelligence.

L'apprentissage automatique est une science consiste à concevoir et développer des algorithmes permettant à donner à l'ordinateur ou à une machine la capacité d'apprendre à partir des données d'exécuter des taches complexe sans avoir été explicitement programmé. Ces algorithmes ont la particularité, au fil d'un processus appelé 'entraînement ou apprentissage d'ajuster leur modèle et leurs comportements en fonctions des résultats observé



Figure 0. 1 - Apprentissage automatique et l'intelligence artificielle

Une explication plus formelle de l'apprentissage automatique a été donné par Tom Mitchell [1] : "Un programme informatique est dit capable d'apprentissage à partir d'une expérience E dans le respect d'une classe de tâche T avec la mesure de performance P s'il accomplit la tâche T , mesurée par P , et améliorée par l'expérience E ". On peut décrire un système d'apprentissage automatique comme une boîte noire d'entrés et sortie schématisée dans la **figure 0 .2**, ou le système prend en entrés les données d'apprentissage X et renvoie une sortie intelligente Y .



Figure 0.2 - système d'apprentissage automatique

L'apprentissage automatique se divise en deux grandes familles : apprentissage supervisé et apprentissage non supervisé, dans le cas d'apprentissage supervisé, la sortie Y est fournie a priori c'est-à-dire que les données d'entrainements possèdent les entrées et les réponses désirées. Dans le cas de l'apprentissage non supervisé, les données d'entrainements ne possèdent pas les réponses désirées, la sortie n'est pas fournie.

Les méthodes d'apprentissage automatique sont utilisées pour une variété de tâches, nous mentionnons ci-dessous les plus importantes :

- La classification.
- La régression.
- Le clustering.
- Système de recommandation.
- Détection des anomalies.
- Estimation de densité.

Plusieurs algorithmes employés pour exécuter ces tâches, nous citons les plus utilisés et les plus célèbres :

- Les réseaux de neurones.
 - Les machines à vecteur de support (SVM),
 - Les arbres de décision,
 - K -plus proches voisins,
 - Algorithme de clustering hiérarchique
 - Les algorithmes de partitionnement
 - Les algorithmes de Naïves Bayes.
- Dans ce mémoire nous allons nous intéresser à la tâche de classification à travers l'utilisation de l'algorithme de Réseau de neurone artificiel.

0.2 Objectif de mémoire

L'objectif de ce mémoire, est d'explorer la technique de classification et les algorithmes fondamentaux largement utilisés par celle-ci, en se concentrons beaucoup plus sur les applications de cette technique et de son impact dans le monde actuel, ainsi qu'aux réseaux de neurones artificiels, leurs concepts de base et leurs apprentissages. Notre approche pratique sera l'implémentation de l'algorithme de réseau de neurones artificiels pour classifier les fleurs d'Iris.

0.3 Description des chapitres

Notre mémoire comporte les parties suivantes :

Le premier chapitre : débutera par une introduction générale citant les définitions et points vocabulaire, les méthodes de classification, ainsi que leurs diverses applications et leur impact sur plusieurs domaines.

Le deuxième chapitre : Sera consacré à la présentation des réseaux de neurones et de leurs différentes structures.

Le troisième chapitre : Sera entièrement dédiés à l'apprentissage des réseaux de neurone, nous expliquons la méthode de rétropropagation du gradient qui permet l'apprentissage d'un réseau de neurone multicouche.

Le quatrième chapitre : Sera dédiés à l'application consistant à classifier des fleurs d'iris par un réseau de neurone multicouche simulé par le logiciel Matlab.

1 Généralités sur la classification

1.1.1 INTRODUCTION

La classification est l'une des techniques les plus utilisées dans l'apprentissage automatique, avec un large éventail d'applications, y compris la reconnaissance de forme, la détection du spam, la catégorisation des clients, le diagnostic médical et la classification des images.

L'apprentissage automatique et en particulier la classification, a pour le but d'accomplir des tâches précises impliquant la prise de décision permettant la résolution automatique des problèmes complexes.

Par exemple, l'aide au diagnostic médical, est actuellement réalisé de manière automatique par des machines réussissant à lire et interpréter des images des patients sains et pathologiques.

Cette discipline de l'intelligence artificielle se placera comme un outil précieux dans de nombreux domaines scientifiques.

Nous présenterons dans ce chapitre tout d'abord ce que c'est la classification, ses méthodes, ses grandes approches, domaines d'applications, . . . etc.

1.2 DEFINITION

« Le seul moyen de faire une méthode instructive et naturelle, est de mettre ensemble les choses qui se ressemblent et de séparer celles qui diffèrent les unes des autres.

M. Georges Buffon, Histoire naturelle, 1749.

La classification est l'une des techniques les plus anciennes d'analyse et de traitement de données, Une classe est un ensemble d'éléments qui sont semblables entre eux et qui sont dissemblables à ceux d'autres classes. La classification repose sur des objets à classer. Les objets sont localisés dans un espace de variables (ont dit aussi attributs, caractéristiques ou critères). Il s'agit de les localiser dans un espace de classes. Ce problème n'a de sens que si on pose l'existence d'une correspondance entre ces deux espaces. Résoudre un problème de classification, c'est trouver une application de l'ensemble des objets à classer, décrits par les variables descriptives choisies, dans l'ensemble des classes. L'algorithme ou la procédure qui réalise cette application est appelé classifieur [2].

La figure (1.1) illustre le principe de la classification :

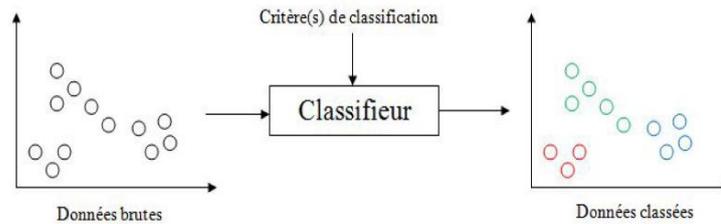


Figure 1. 1 - Principe de classification [33]

D'une manière générale, la classification se définit alors comme une méthode mathématique d'analyse de données, pour faciliter l'étude d'une population d'effectif important, généralement des bases d'observations caractérisant un domaine particulier (Animaux, plantes, malades, gènes, . . . etc.), où on les regroupe en plusieurs classes [3].

1.3 DEFINITION FORMELLE

Soit $x \in R^d$ un ensemble représentant un espace à d dimensions, appelé *Espace des instances*. La donnée $x \in X$ est appelée *une instance* et représente un point dans l'espace X . L'instance x est présentée sous forme d'un vecteur de taille d , $x = (x(1), \dots, x(d))$, où chaque composante $x(i) \in R$ est une valeur discrète ou continue. Soit Y un ensemble fini de classes où chaque classe $y \in Y$ est présentée sous forme d'une valeur discrète appelée *étiquette de classe* (un nom ou identifiant unique pour la classe). Le classifieur se présente alors sous forme d'une fonction de classification h (appelée aussi modèle de classification) permettant d'associer une donnée $x \in X$ à une étiquette de classe $y \in Y$

$$h : \begin{cases} X \rightarrow Y \\ x \rightarrow y = h(x) \end{cases}$$

Notez que pour le problème de *classification*, l'espace des réponses $Y \in \mathcal{N}$ est discret et fini, vu que chaque $y \in Y$ représente une classe. Lorsque Y est continu (c.à.d. $Y \in R$), on parle alors de problème de régression [4].

1.4 TERMINOLOGIE

- **Classification** : ou classification automatique, terme généralement employé par les auteurs français, il est employé dans un autre sens par les anglo-saxons qui disent « classification. » pour désigner la technique prédictive que les français appellent « classement ».

| FRANÇAIS | ENGLAIS |
|----------------|----------------|
| Classification | Clustering |
| Classement | Classification |

- **Segmentation** : terme employé en marketing.
- **Typologie**, ou **analyse typologique**.
- **Clustering** : terme anglo-saxon le plus courant.
- **Taxinomie** ou **Taxonomie** (Biologie ou zoologie).
- **Nosologie** (Médecine).

1.5 EXEMPLES DE PROBLEMES DE CLASSIFICATION

✓ EMAIL SPAM

Peut-être le type de classification que nous voyons tous les jours à chaque fois que nous ouvrons notre courrier électronique, le filtre anti spam, à chaque fois qu'un email arrive fait une prédiction, si c'est un spam ou non.

Cette prédiction doit être faite sur la base de texte d'email, l'expéditeur, l'adresse *IP* (*Internet Protocol*) de cet expéditeur ..etc, et à partir de ces informations le classificateur décide s'il s'agit de spam ou non .

From: cheapsales@buystufffromme.com
To: ang@cs.stanford.edu
Subject: Buy now!

Deal of the week! Buy now!
Rolex watches - \$100
Medicine (any kind) - \$50
Also low cost M0rgages available.

Spam

From: Alfred Ng
To: ang@cs.stanford.edu
Subject: Christmas dates?

Hey Andrew,
Was talking to Mom about plans for Xmas. When do you get off work. Meet Dec 22?
Alf

Non-spam

Figure 1.2 - Représentation des emails spam et non-spam

✓ **ROCONNAISSANCE DE FORMES :**

C'est un problème de classification très répandu , le programme prend en entrés des donnés (formes) qui peuvent s'agir d'une image (visage, empreinte digitale, chiffre manuscrite écrit par personneetc.) ou sonore (reconnaissance de paroles) ,et bien d'autres , puis il détermine en sortie la catégorie attribué à ces données .

Exemple :

la reconnaissance des chiffre manuscrites

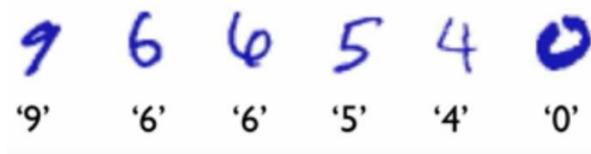


Figure 1.3 - Reconnaissance des chiffres manuscrite

Ou d'empreintes digitales :



Figure 1.4 - Reconnaissance d'empreintes digitales

Ou de Reconnaissance vocale :

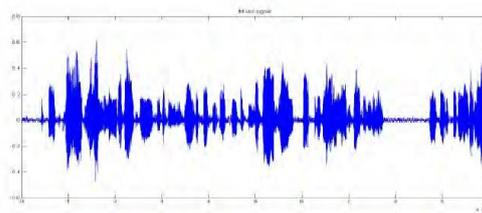


Figure 1.5 - Reconnaissance vocale

✓ **SYSTEME AIDE AU DIAGNOSTIC MEDICAL**

L'objectif est que le classifieur décide à partir de certaines entrées qui peuvent être une température de thermomètre, une radiographie, un électro-gramme ou des tests de laboratoire, si un patient présente une maladie ou non.

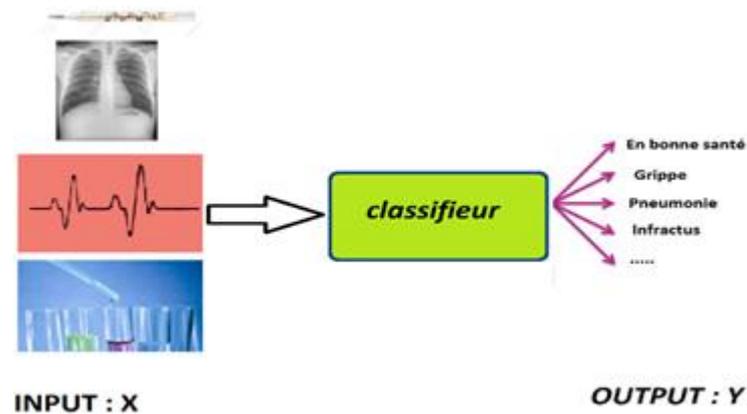


Figure 1.6 - Diagnostic médical automatique

✓ **Classification d'images satellitaires**

De nos jours, de grandes quantités d'images de télédétection haute résolution sont acquises quotidiennement, la classification des images satellites est requise pour de nombreuses applications telles que l'urbanisme, l'agriculture et la surveillance de l'environnement.

La figure suivante montre un système de classification satellitaire qui a pu identifier à partir des données satellites sur une surface donnée : des maisons, route, arbres, pelouse.

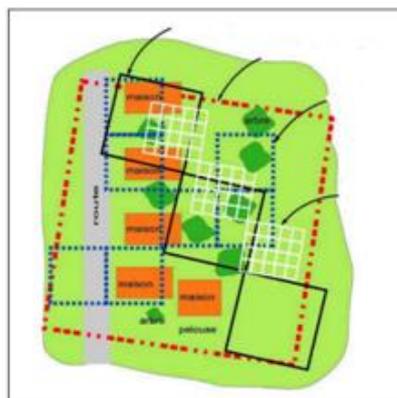


Figure 1.7 - Classification à partir d'une image satellite

1.6 DOMAINES D'APPLICATION

La classification a connu une croissance importante ces dernières années et désormais appliqué dans plusieurs domaines, de la reconnaissance de forme à la médecine en passant même par le domaine militaire.

Elle est largement utilisée en reconnaissance de formes : reconnaissance des caractères manuscrite comme la reconnaissance de code postale, les chèques et les cartes, reconnaissance de parole.... etc. On retrouve des applications en :

L'ingénierie : génie mécanique, génie électrique, l'industrie.

L'informatique : Classification des documents, classification des donnés, segmentation d'image, moteur de recherche... etc.

La médecine : Système d'aide au diagnostic, c'est l'une des plus importantes applications de la classification dans le domaine médical, un système qui apportera des informations au médecin afin de l'aider à sa décision.

Marketing : Par exemple, une entreprise peut collecter des informations sur ces clients afin de savoir leur satisfaction.

Multimédia : la classification des documents web afin d'offrir une organisation thématique de ces derniers, l'accessibilité ou non d'une page web pour enfant à partir de son contenu.

Autres domaines : comme **la géographie, l'astronomie** ou en **biologie**.

Cette diversité reflète le grand intérêt de la classification dans la recherche scientifique.

1.7 METHODES DE CLASSIFICATION

Les méthodes de classification analyse des entrés (des donnés : images, document, état de patient...) pour en déduire des sorties (classes, catégories, diagnostic...), autrement dit étiqueter ces différents donnés en leur associant une classe.

L'apprentissage automatique se propose de construire automatiquement une telle procédure. En effet, deux types de classification peuvent être distingués .

1. « Supervisé » qui sont des méthodes basées sur la notion d'apprentissage supervisé ou les sorties sont observées c'est-à-dire les classes sont connus et prédéfinis.
2. « Non supervisé » basée sur la notion d'apprentissage non supervisé ou les sorties sont non observé, les classe sont inconnus.

Cependant, il existe d'autres types de classification, citons l'apprentissage semi-supervisé ou les sorties ne sont que partiellement observées, et on peut dire que ce type de classification est un bon compromis entre les deux apprentissages supervisé et non supervisé.

1.7.1 CLASSIFICATION NON SUPERVISE

1.7.2 Définition

La classification non supervisée, comme son nom l'indique, consiste à apprendre sans superviseur. A partir d'une population, il s'agit d'extraire des classes ou groupes d'individus présentant des caractéristiques communes, le nombre et la définition des classes n'étant pas donné a priori [5] .

Ceci est fait en optimisant un critère visant à regrouper les individus dans des classes, chacune le plus homogène possible et, entre elles, les plus distinctes possible [6] .

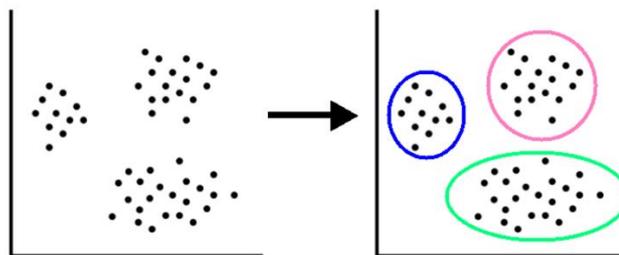


Figure 1.8 - Classification non supervisé [33]

La classification non supervisée ou Le *clustering* est un outil important pour l'analyse de données. Il vise à trouver les structures intrinsèques des données en les organisant en groupes homogènes et distincts (les *clusters*). Les objets dans un même *cluster* doivent être similaires entre eux et différents des objets des autres clusters [7].

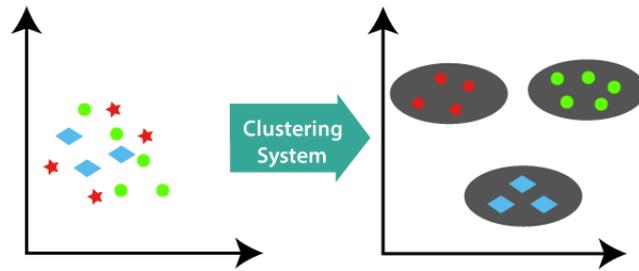


Figure 1.9.a - Représentation d'un cluster



Figure 1.10 - Image d'un cluster [35]

1.7.3 EXEMPLE :

Dans une classe, un professeur veut faire **2** groupes d'élèves de compétences semblables
 Les notes des élèves sont détaillées dans le tableau suivant :

| Matières Prés Nom | Maths | Physique | Histoire | Philosophie |
|----------------------|-----------|-----------|-----------|-------------|
| Karim | 20 | 20 | 0 | 0 |
| Mohamed | 7 | 7 | 13 | 13 |
| Khaled | 20 | 20 | 0 | 0 |
| Ines | 0 | 0 | 20 | 20 |
| Salim | 10 | 10 | 10 | 10 |
| Ikram | 4 | 4 | 16 | 16 |

Table 1. 1 - Exemple des notes élèves

Tous les élèves ont la moyenne **10/20**, mais en comparant les notes par matière, on peut les classer comme ceci :

Groupe 1: Karim, Khaled, Salim

Groupe 2: Mohamed, Ines, Ikram

Certes, Si on possède une classe de 30 élèves et on considère 10 matières, la classification serait impossible. C'est pourquoi des méthodes mathématiques ont été créées pour résoudre ces problèmes.

Donc, si les éléments sont sans étiquettes et si les classes sont inconnus, l'objectif est de regrouper en classes ces éléments non classés, on parle alors de classification non supervisée.

1.7.4 ALGORITHMES

Il existe une très large famille de méthodes dédiées à la classification non supervisée, ils se distinguent en deux grandes types : algorithmes de partitionnement tel que : K-moyennes, fuzzy C-means , K-médoïdes, .. etc, et algorithmes hiérarchique comme classification ascendante hiérarchique *CAH*, nous citons :

- **L'algorithme du K-moyennes (k-means) :**

C'est une méthode efficace qui permet de diviser un ensemble de données en **k** classes homogènes.[18]

- Dans un premier temps, l'algorithme choisit le centre des **k** clusters à **k** objets.
- Par la suite, on calcule la distance entre les objets et les **k** centres et on affecte les objets aux centres les plus proches.
- Ensuite, les centres sont redéfinis à partir des objets qui ont été affectés aux différents clusters.
- Puis, les objets sont assignés en fonction de leur distance aux nouveaux centres et ainsi de suite. L'algorithme se répète donc jusqu'à ce qu'il y ait convergence. [8]

Mathématiquement une fois le k centre de classe désigné $U_1, U_2, \dots, U_k \in R^p$, et pour chaque individu i , on calcule la distance avec la formule de distance euclidienne :

$$D(x_i, U_k) = \sqrt{\sum_{i=1}^n (x(i) - U_k)^2} \dots \dots \dots (1. 1)$$

$x(i)$: Donnés

U_k : Centre de classe

Et on affecte chaque individu $x(i)$ à la classe C_k pour laquelle la distance du centre est minimale :

$$C(i) = k = \operatorname{argmin} \|x(i) - U_k\|^2 \dots \dots \dots (1. 2)$$

$\|x(i) - U_k\|^2$: Distance euclidienne au carré

La redéfinition des centres (Barycentre) se calcule avec la formule suivante :

$$U_k = \frac{\sum_{i=1}^n 1_{\{C(i) = k\}} \cdot x(i)}{\sum_{i=1}^n 1_{\{C(i) = k\}}}$$

$C(i)$ = la classe de $x(i)$

D'une manière plus général on peut résumer l'algorithme des **k-means** comme suit :

- Initialisation : Choix du centre .
- Etape1 : Regroupement de point en fonction du centre .
- Etape2 : Redéfinition des nouveaux centres .
- Itération : On itère les étapes 1 et 2 jusqu'à convergence .

La **figure 1.11** suivante donne un aperçu de l'application des différentes étapes de l'algorithme des **k-means** pour le partitionnement (clustering) en 2 classes ($k = 2$) :

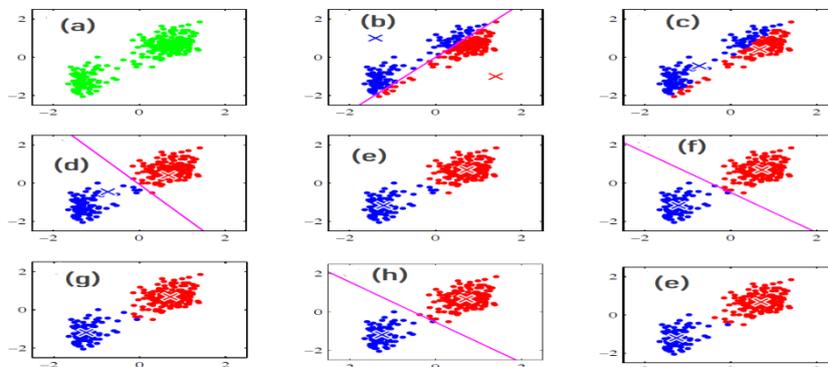


Figure 1.10 - Aperçu de l'algorithme K-means

Algorithme Fuzzy C-Means :

C'est un algorithme de classification flou très similaire à l'algorithme *K* means, les classes ne sont pas définies nettement et précisément. La figure suivante montre un exemple de cette classification.

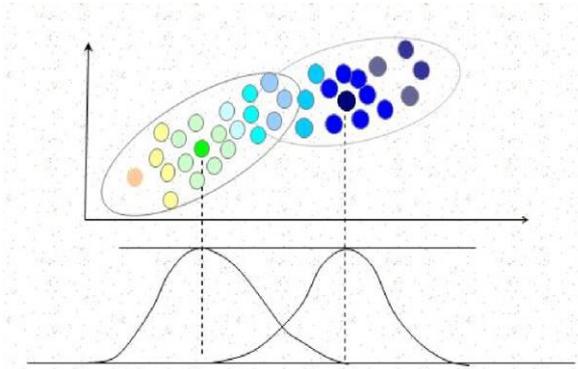


Figure 1.11 - Exemple des classes floues

Dans ce cas, les données ne sont plus assignées à un cluster unique, mais à plusieurs clusters avec un degré d'appartenance, cela est fait par un processus itératif basé sur la minimisation de la fonction objectif suivante :

$$J_m = \sum_{i=1}^N \sum_{j=1}^C U_{ij}^m \cdot \|X_i - C_j\|^2 \dots\dots\dots (1.3)$$

U_{ij} : Matrice d'appartenance

X_i : Donnés

C_j : Clusters.

L'algorithme de clustering flou peut être résumé comme suit :

Tous d'abord l'algorithme définit la matrice d'appartenance d'une façon aléatoire et calcule les centres de clusters initiaux par cette fonction :

$$v_{ij} = \frac{\sum_{k=1}^N (U_{ik})^2 \cdot X_{kj}}{\sum_{k=1}^m (U_{ik})^2} \dots\dots\dots (1.3)$$

v_{ij} : Le centroïde de cluster ij

U_{ik} : Le degré d'appartenance

Ensuite on calcule la distance entre chaque point et centroïdes avec la formulé de distance euclidienne :

$$d_{jk} = \sqrt{(X_k + X_i)^2 + (Y_k + Y_i)^2} \dots\dots\dots (1. 4)$$

Enfin on met à jour la matrice d'appartenance par la formule ci-dessous :

$$U_{ik}^{(r+1)} = \left[\sum_{j=1}^c \left(\frac{d_{ik}^{(r)}}{d_{jk}^{(r)}} \right)^2 \right]^{-1} \dots\dots\dots (1. 5)$$

U_{ik} : Matrice d'appartenance.

c : Nombre de clusters.

d_{jk} : Distance entre le cluster i et la donnée k .

En dernier lieu, on vérifie la convergence en calculant le maximum de la différence entre la matrice d'appartenance calculé et la matrice d'appartenance initiale.

$$\max_{ik} \left| U_{ik}^{(1)} - U_{jk}^{(0)} \right| \leq \varepsilon \dots\dots\dots (1. 6)$$

On recommence depuis la deuxième étape s'il n'y pas de convergence.

- **Classification ascendante hiérarchique :**

Le fonctionnement de cet algorithme consiste à mettre en évidence des liens hiérarchique entre individus ou groupe d'individus.

Au début de processus, l'algorithme parcourt l'ensemble des objets pour repérer les deux objets les plus proches c'est-à-dire qui ont une distance minimale ; les fissionner pour former un groupe. Puis, de nouveau parcourir l'ensemble des objets et groupe d'objets déjà constitué et assembler les deux groupes les plus proches et ainsi de suite jusqu'à qu'il ne reste plus qu'un seul groupe.

Le résultat de cet algorithme est une suite de partition hiérarchie représenté par un graphique nommé Dendrogrammes.

Pour obtenir une classification finale, on coupant le dendrogramme et on forme une partition en classe, ce qui nous permet d'extraire plusieurs partitionnement facilement.

La mesure de distance entre individu calculé par distance euclidienne, méthode couramment utilisée.

- Un exemple de regroupement en hiérarchie est représenté ci-dessous :

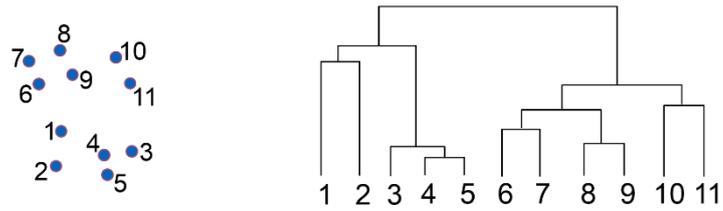


Figure 1.12 - Regroupement en hiérarchie

- Exemple d'obtention de plusieurs partitionnements à partir d'un dendrogramme :

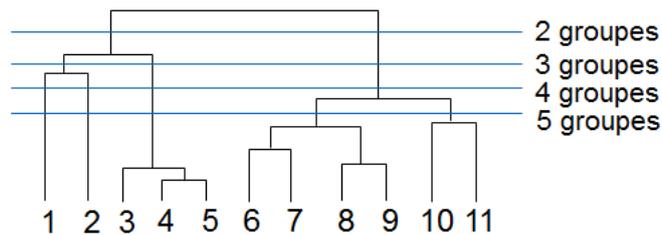


Figure 1.13 - Partitionnement d'une hiérarchie

1.7.5 Exemples d'application :

Quelques exemples d'applications de la classification non-supervisée sont présentés ci-dessous :

- Application 1 : En biologie, on veut regrouper les espèces suivant leurs caractéristiques et donc de leurs origines communes.
- Application 2 : En psychologie, on veut classer les individus selon leur type de Personnalités.
- Application 3 : En chimie, on veut classer des composés selon leurs propriétés.
- Application 4 : Dans l'industrie, on veut :
 - Analyser des résultats d'enquêtes.
 - Identifier les clients potentiels d'une entreprise.
 - Identifier les clients susceptibles de partir à la concurrence.
 - Déterminer des lieux de ventes (pose de distributeurs de billets. . .).
 - Analyser, identifier les risques (dégâts des eaux. . .),
 - Analyser des données textuelles [9] .

1.8 La classification supervisée :

1.8.1 Définition :

La classification supervisée en anglais classification, est une méthode supervisée consiste à définir une fonction qui permet de ranger des donnés dans des classes prédéterminées par un expert.

Selon [Govaert, 2003] : « la classification supervisée (appelée aussi classement ou classification inductive) a pour objectif « d’apprendre » par exemple. Elle cherche à expliquer et à prédire l’appartenance de documents a des classes connues a priori. Ainsi c’est l’ensemble des techniques qui visent à deviner l’appartenance d’un individu a une classe en s’aidant uniquement des valeurs qu’il prend. »

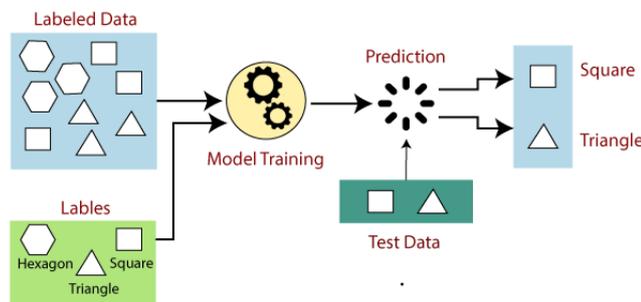


Figure 1.14 - Classification non supervisé

1.8.2 EXEMPLE

Nous allons illustrer le concept de la classification supervisé avec cet exemple.

Nous avons des différentes fleurs classées par la taille de pétale et la taille de la tige en mettant deux couleurs pour deux espèces de fleurs, vert et rouge.

Dans le tableau suivant, la taille de pétale en mm et la taille de la tige en centimètres.

| Taille de pétale | Taille de la tige | Couleur (Espèce) |
|------------------|-------------------|------------------|
| 19 | 44 | VERT |
| 27 | 37 | VERT |
| 18 | 45 | VERT |
| 14 | 49 | ROUGE |
| 16 | 58 | ROUGE |
| 15 | 56 | ROUGE |

Table 1. 2 - Déférents tailles de tige et de pétale des fleurs

D'autre part, une fleur a les caractéristiques suivantes :

| Taille de Pétale | Taille de la tige |
|------------------|-------------------|
| 15 | 37 |

A partir de ces caractéristiques, à quel couleur cette fleur pourrait appartenir ?

La réponse est difficile, c'est pourquoi des méthodes mathématiques ont été créées pour résoudre ces problèmes.

Donc, si les classes possibles sont connu, l'objectif est de chercher à prédire la classe de toute nouvelle donnée, on parle alors de classification supervisée.

1.8.3 ALGORITHMES

Pour résoudre les taches de la classification supervisé, plusieurs algorithmes sont possibles, nous mentionnons les plus employées :

(a) Les **K**- plus proches voisins (**Kppv**) :

C'est l'algorithme de classification le plus simple et largement utilisé, sur nommé « nearest neighbors (**K-NN**) » (plus proches voisins, en français).

L'idée principale de cette règle de classification est que l'étiquette d'un point $x \in X$ est correctement prédite par celle de ses voisins dans X . On obtient une règle de classification en posant que l'étiquette d'un point $x \in X$ est celle qui est majoritaire parmi les étiquettes de ses **k** plus proches voisins dans l'échantillon. Le nombre **k** de voisins considérés et la distance sur X utilisée pour définir la notion de « voisin dans X » sont à fixer librement par l'utilisateur [10].

Pour bien illustrer cette définition on va se servir de l'exemple ci-dessous, un jeu de données avec deux classes bleu et rouge et on veut classer le nouveau point vert, L'algorithme consiste à prendre les plus proches voisins pour déterminer à quelle classe appartient ce point.

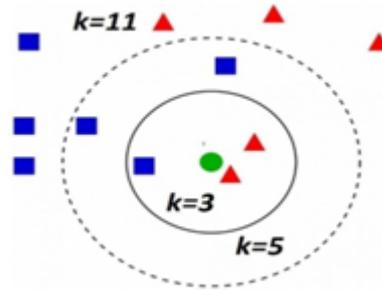


Figure 1.11 - Schéma d'une classification par la méthode k plus proche voisin

Disons que $K = 3$ (3-NN), par conséquent le point appartient à la couleur rouge car c'est la classe majoritaire parmi ses k plus proche voisin

Et si on prend $K = 5$, le point appartient à la couleur bleu.

Enfin avec $K = 11$, notre point appartient à la couleur bleu.

(b) Machines à Vecteurs de Support (SVM) : Le but des SVM est de trouver un séparateur entre deux classes qui soit au maximum éloigné de n'importe quel point des données d'entraînement. Si on arrive à trouver un séparateur linéaire c'est-à-dire qu'il existe un hyperplan séparateur alors le problème est dit linéairement séparable sinon il n'est pas linéairement séparable et il n'existe pas un hyperplan séparateur. [11]

Le schéma ci-dessous, représente les données qui sont linéairement séparables et ceux qui ne le sont pas.

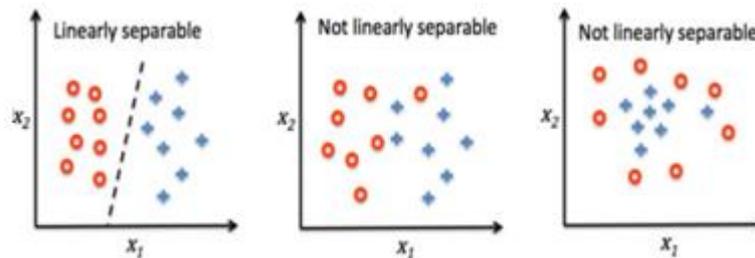


Figure 1.12 - Représentation des données linéairement séparables et non linéairement séparables

Plus formellement, considérons nos données $X \in \mathbb{R}^{n \times p}$ (n points, p dimensions) et les deux classes représentés par un vecteur $y \in \{C1, C2\}$.

L'équation de l'hyperplan séparateur est $w \cdot x + b = 0$ on l'utilise directement comme une fonction discriminante : $f(x) = w \cdot x + b$

Si cette fonction est positive, le point x est classé en **C1**, et inversement, si elle est négative le point est classé en **C2**.

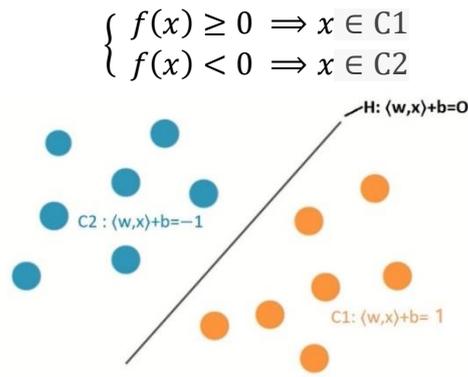


Figure 1.17 - Séparateur SVM

En fait , si les données sont linéairement séparable , il existe de nombreux hyperplans possibles qui pourraient être choisis , l'objectif est de trouver la droite qui possède la marge maximale, c'est-à-dire la distance maximale entre les points de données des deux classes, le SVM préfère donc entre ces droites celle qui a la plus grande marge, cela nous permet de mettre avec confiance un nouveau point de données dans la catégorie correcte.

La marge se calcule avec la formule suivante : $\gamma = \frac{2}{\|w\|}$.

L'idée est illustrée dans les schémas ci-dessous :

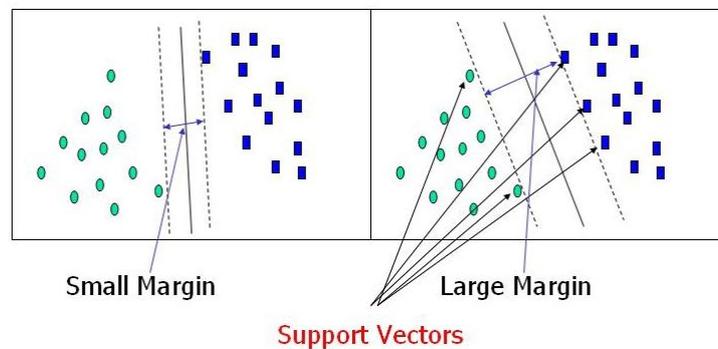


Figure 1.18 - La marge séparatrice

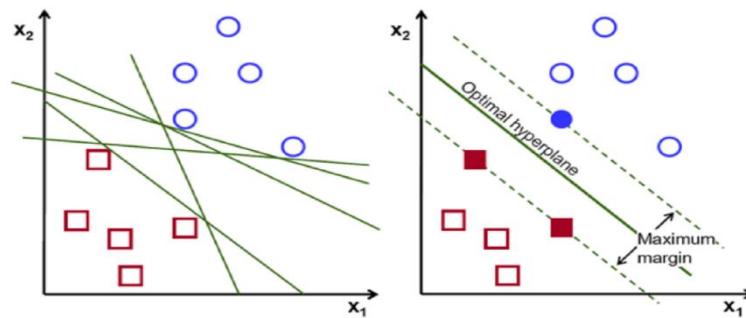


Figure 1.13 - Hyperplans possibles

Lorsque nos données ne sont pas linéairement séparables, on utilise l'astuce du noyau pour appliquer une transformation sur nos données pour pouvoir les séparer linéairement.

(c) Réseau de Neurone :

Les réseaux de neurones sont à l'origine d'une tentative de modélisation mathématique du Cerveau humain. Le principe général consiste à définir des unités simples appelées neurones, chacune étant capable de réaliser quelques calculs élémentaires sur des données numériques. On relie ensuite un nombre important de ces unités formant ainsi un outil de calcul puissant [12].

Afin de mieux comprendre la technique de classification par les réseaux de neurones qui fonctionnent tous avec la même manière, nous utilisons l'algorithme de base des réseaux de neurones de la classification supervisée, qui est le perceptron, dédié à la classification binaire, c'est-à-dire séparation linéaire des données en deux classes, et nous réalisons un exemple très simple rendant le principe très clair.

- Le perceptron appelé neurone artificiel ou neurone formel, est inspiré de neurone biologique ou les entrées de neurone artificiel correspondent aux dendrites biologiques, et la sortie de neurone artificiel correspond à l'axone de neurone biologique.

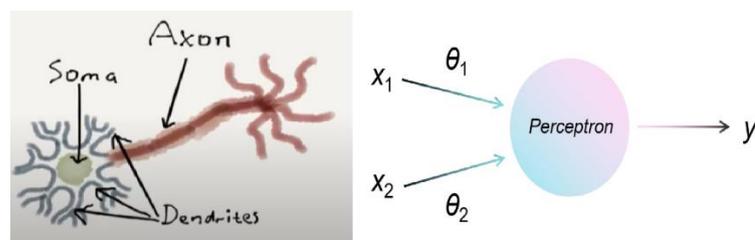


Figure 1.20 - Neurone biologique et neurone artificiel

- Un neurone (perceptron) est une unité de traitement de l'information, schématiquement il est représenté comme suit :

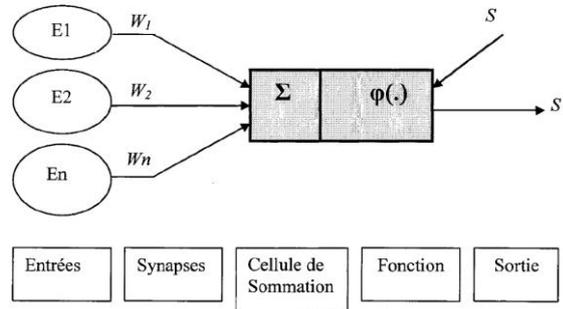


Figure 1.21 - représentation schématique du Perceptron

Les valeurs des entrées E_1, \dots, E_n représentent en général les attributs d'un objet à classer et les poids W_1, \dots, W_n (ou coefficients synaptiques) associés aux entrées sont des variables de la fonction score du poids, appelée aussi *fonction d'activation du neurone* (la fonction d'activation la plus utilisée est la somme pondérée des valeurs d'entrée). La valeur d'activation est ensuite passée comme argument à la fonction de sortie qui détermine la valeur de sortie du neurone S' .

L'entrée supplémentaire S sert à indiquer au neurone la valeur de sortie attendue pour qu'il puisse corriger ses coefficients synaptiques et s'approche de cette valeur [12].

La valeur de la sortie d'un neurone formel est calculée par l'équation suivante :

$$S = f(x) = w \cdot x + b$$

Où chaque entrée est multiplié par un coefficient correspondant au poids aléatoire, le tout est sommé et additionné à un biais (b) et passant par une fonction f . Puis ces coefficients sont modifiés progressivement durant le processus jusqu'à avoir une bonne séparation des données en deux classes.

- Voyons comment fonctionne une classification linéaire avec un simple exemple :

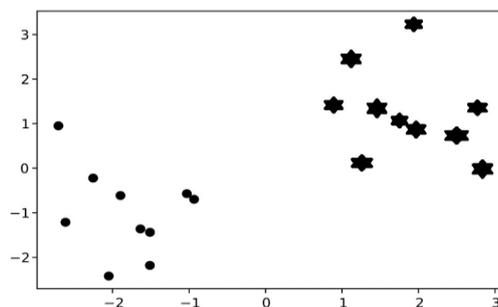


Figure 1.14 - Données exemple

Nous voulons séparer ces données défini par des coordonnées (x_0, x_1) en deux classes.

On appliquons *l'équation de neurones* sur ces données après avoir calculé les paramètres adapté à ce problème w et b (processus itératif que nous détailleront au **chapitre 2**), nous

obtiendrons les valeurs de sortie du perceptron $S = \begin{cases} 1 & \text{si } w^T \mathbf{x} + b \geq 0 \\ \text{et} \\ (-1) & \text{si } w^T \mathbf{x} + b < 0 \end{cases}$

Si on trace une droite avec les résultats négatives et positives obtenus, on obtient un hyper plan H qui divise nos données en deux sous-espaces comme le montre la figure suivante :

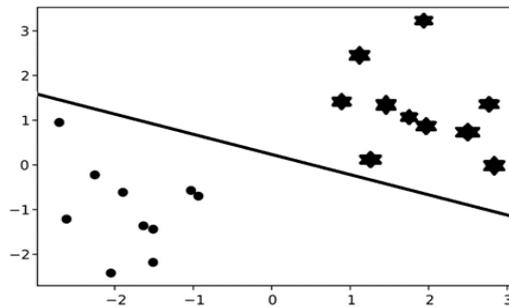


Figure 1.23 - Classification linéaire par perceptron

Nous voyons que le neurone a réussi à séparer les données en deux classes.

La notion de Réseau de Neurone sera reprise au chapitre 2 et détaillées au chapitre 3.

1.8.4 EXEMPLE D'APPLICATION :

Il existe une multitude de problèmes qui entrent dans le cadre de la classification supervisée, Quelques domaines d'application sont présentés ci-dessous :

- Application 1 : Un archéologue cherche à déterminer si des restes humains sont ceux d'un Homme ou d'une femme.
- Application 2 : Dans une banque, une commission de crédit doit décider, à partir de Paramètres financiers, si on accorde ou non un prêt à un particulier.
- Application 3 : Étant donné un ensemble de symptômes, un médecin doit poser un Diagnostic.
- Application 4 : Dans l'industrie, on veut :
 - Identifier des visages, des empreintes digitales,
 - Identifier des objets dans des séquences vidéo,
 - Rechercher des clients potentiels dans des bases de données,
 - Rapprocher un ou plusieurs mots de manière pertinente au texte le plus pertinent [13].

1.9 Indicateurs de performance d'un modèle de classification

1.9.1 La matrice de confusion :

La matrice de confusion est un outil puissant qui sert à évaluer la qualité d'un modèle de classification, l'avantage de cette matrice est qu'elle est simple à lire et à comprendre.

- Elle offre une visualisation rapide afin d'analyser les performances d'un modèle et d'identifier les tendances qui peuvent aider à modifier les paramètres [14].
- Dans une matrice de confusion les lignes correspondent aux classes réelles et les colonnes correspondent aux classes prédites.
- Prenons comme exemple un classificateur binaire, qui classe chaque personne selon son état de santé après un diagnostic c'est à dire. Si il est malade ou pas malade, si il est malade donc le teste est positive sinon le teste est négative.

Vrais positifs (VP) : Est le nombre de personne malade .

Vrais négatifs (VN) : Est le nombre de personne qui ne sont pas malade.

Faux négatifs(FN) : Est le nombre de personne qui ne sont pas malade mais ils sont classées comme malade

Faux positifs (FP) : Est le nombre de personne qui sont malade mais ils sont classées comme pas malade.

| | | Classe prédite | |
|-------------|------------|----------------|--------|
| | | Pas malade | Malade |
| Classe réel | Pas malade | VN | FP |
| | Malade | FN | VP |

Table 1.3 - matrice de confusion

Un classifieur parfait correspond a une matrice diagonale.

Il y a plusieurs paramètres qui résument la matrice de confusion qui sont :

- **Précision** : Proportion d'éléments bien classés pour une classe donnée elle est représentée par l'équation suivante

$$Précision_{de\ la\ classe\ 1} = \frac{VP}{VP + VN}$$

- **Rappel** : Proportion d'éléments bien classés par rapport au nombre d'élément de la classe à prédire elle est représentée par l'équation suivante

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

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

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

NB : La matrice de confusion peut être utiliser même pour la multi classification c a d pour un nombre de classe supérieur à deux.

1.9.2 La courbe ROC (receiver operating characteristic) :

La courbe ROC est un graphique représentant les performances d'un modèle de classification pour tous les seuils de classification. Cette courbe trace le taux de vrais positifs en fonction du taux de faux positifs [15]

La représentation de la courbe ROC offre aussi un indice de classification appelé AUC (Area under the curve) plus cette aire rapproche a un plus le modèle de classification est performant est performant.

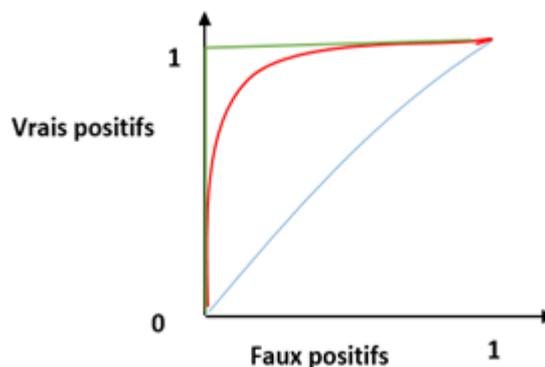


Figure 1.24 - La courbe ROC

- La courbe bleue correspond à un modèle de classification aléatoire.
- La courbe rouge correspond à un bon modèle de classification.
- La courbe verte correspond à modèle de classification parfait.

1.10 CONCLUSION :

Dans ce chapitre, nous avons présentés une généralité sur la classification, ses méthodes, la conception de quelques algorithmes, et quelque exemple et domaine d'application de la classification.

Dans le chapitre suivant nous allons-nous intéresser aux différentes structures de réseau de neurone artificiel ainsi qu'à leur principe de fonctionnement général .

Chapitre 2

2 Les réseaux de neurones

2.1 Introduction :

Le domaine de recherche des réseaux de neurones est très vaste, commencé dans le XXe siècle, inspiré de la biologie par un neurophysiologiste et développé par les mathématiciens et les informaticien dans le but d'atteindre le niveau de l'intelligence humaine , notamment au niveau de la perception , la mémorisation et même le raisonnement afin de nous facilite la résolution de déférent problème scientifique.

Les efforts qui ont été fait par les chercheurs ces dernières décennies rendent les réseaux de neurones un atout très puissant, de nos jours les réseaux de neurones occupent la grande partie de l'intelligence artificielle.

Dans ce chapitre on va aborder les réseaux de neurones, on donnant un aperçu général sur ce domaine, commençant par l'historique passant par les premières structures proposées tel que le perceptron, le réseau ADALINE, le perceptron multicouche en finalisant par les structures profondes telles que les réseaux de neurone récurrent, les LSTM et les réseaux de neurone à convolution CNN.

2.2 Chronologie :

L'étude de l'intelligence humaine a commencée avant des milliers d'années, avec l'essor massif dans le domaine de l'électronique programmable, les ambitions de réaliser une machine qui simule le système nerveux humain a augmenté, la première tentative vers les réseaux de neurones artificiels c'étaient en 1943, par le neurophysiologiste Warren Mc Culloch et le mathématicien Walter Pitts ont pu modéliser le tout premier réseau de neurones artificiels par circuits électriques.

En 1949, le psychologue Donald O. Hebb a présenté dans son livre "The Organization of Behavior" une règle d'apprentissage permettent de définir les coefficients synaptiques cette règle fait la suite des travaux des deux chercheurs Mc Culloch et pitts connue par "Règle de Hebb" cette règle est encore utilisée jusqu'à nos jour.

En 1957, l'apparition d'un nouveau concept connue par le perceptron inventé par Fank Rosenblatt dans le laboratoire d'aéronautique de l'université Cornell c'est le premier système apte à apprendre.

Plus de dix ans après exactement en 1969, les deux chercheurs Marvin Lee Minsky et Seymour Papert ont découvert des insuffisances dans le perceptron par exemple l'incapacité de représenter des problèmes essentielles comme le problème XOR et la séparation linéaire, Ces critiques ont conduit à une stagnation de la recherche dans ce domaine, jusqu' à l'arrivé de l'année 1972 le chercheur T.Kohonen à introduire un modèle d'une mémoire associative.

En 1982, le physicien John Joseph Hopfield donna un nouveau souffle au neuronal en publiant un article introduisant un nouveau modèle de réseau de neurones (complètement récurrent) [16]. Mais le problème de non linéarité reste toujours non résolu et, il faudra attendre 1984 pour obtenir un modèle de réseaux de neurones qualifié de multicouches et ne possédant pas les défauts démontrés par Minsky et Papert, ce modèle proposé par Paul J. Werbos et mit en œuvre par David Rumelhart en 1986 repose sur la rétro propagation du gradient de l'erreur dans des systèmes à plusieurs couches [17].

Depuis ; la recherche dans les réseaux de neurones se sont multipliés et l'application de cette technologie dans les différents domaines ne cesse d'augmenter tel que la classification, le contrôle l'optimisation, la reconnaissance de caractères et de signatures, la reconnaissance de formes et de motifs, l'industrie, les jeux vidéo, grâce à sa capacité d'apprentissage.

2.3 Neurone biologique :

Un neurone biologique est composé de quatre éléments principaux :

2.3.1 Le corps cellulaire :

C'est l'élément qui contient le noyau du neurone et l'organe biochimique nécessaire pour la production des enzymes, il est de forme pyramidale ou sphérique sa fonction est de faire la somme des influx entrant et envoie lui-même un influx lorsque la somme dépasse un seuil précis via l'axone.

2.3.2 Les dendrites :

Ils se prolongent du corps cellulaire, et ils se ramifient autour du neurone et forment une sorte de vaste arborescence leur fonction est de capter les signaux envoyées au neurone.

2.3.3 L'axone :

Il conduit les signaux émis par le corps cellulaire, sa taille peut varier de quelques millimètres jusqu' à plus d'un mètre, il peut se ramifié à son extrémité comme il peut se connecter aux dendrites des autres neurones.

2.3.4 Les synapses :

Ils sont les point de communication entre les neurones, en général entre l axone d'un neurone émetteur et les dendrites d'un neurone récepteur.

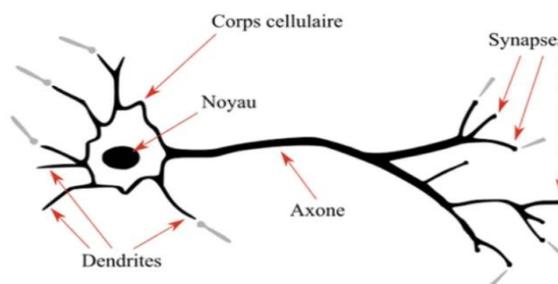


Figure 2.1 - Neurone biologique [35]

2.4 Neurone artificiel :

Appelé aussi neurone formel est un modèle mathématique inspiré d'un neurone biologique, le neurone artificiel possède plusieurs entrées qui correspond aux dendrites où il recueille l'information et une sortie qui correspond au cône d'émergence (point de départ de l'axone) d'où il envoie un signal électrique, Les actions excitatrices et inhibitrices des synapses sont représentées, la plupart du temps, par des coefficients numériques (les poids synaptiques) associés aux entrées. Les valeurs numériques de ces coefficients sont ajustées dans une phase d'apprentissage. Dans sa version la plus simple, un neurone formel calcule la somme pondérée des entrées reçues et ajoute un biais, puis applique à cette valeur une fonction d'activation, généralement non linéaire. La valeur finale obtenue est la sortie du neurone [18], comme montré dans l'équation suivante.

$$y_j = f(a_j) = \varphi(\sum_{i=0}^n (W_i * X_i) + b_j) \dots \dots \dots (2.1)$$

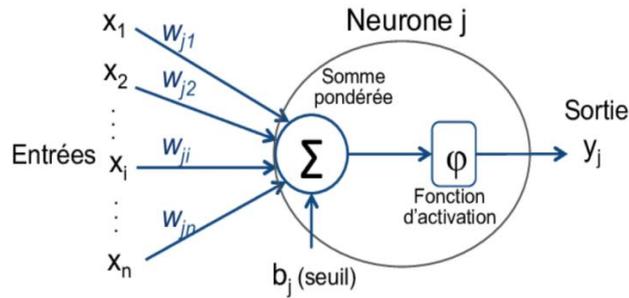


Figure 2.2 - Schéma d'un Neurone artificiel

| Neurone biologique | Neurone artificiel (formel) |
|--------------------|-----------------------------|
| Axones | Signal d'entrée |
| Synapses | Poids de la connexion |
| dendrites | Signal de sortie |

Table 2.1 - Correspondance neurone biologique/neurone artificiel

2.4.1 Le poids :

Le poids est le coefficient qui control le signal d'entrée (la force de connexion), en d'autres termes le poids décide l'influence de l'entrée sur la sortie.

2.4.2 Le biais :

Le biais se connecte aux neurones de la couche précédente par l'intermédiaire d'un poids, généralement appelé seuil [19], il est similaire à l'interception ajoutée dans une équation linéaire. C'est un paramètre supplémentaire dans le réseau neuronal qui est utilisé pour ajuster la sortie avec la somme pondérée des entrées du neurone. Ainsi, le biais est une constante qui aide le modèle de manière à ce qu'il s'adapte le mieux aux données.

2.4.3 La fonction d'activation :

La fonction d'activation est une fonction mathématique responsable du type de l'information émise par le neurone, elle intervient après le calcul de la somme pondérée et l'ajout du biais, elle possède un influe profond sur les performances du réseau. D' où il est préférable de bien choisir le type de cette fonction dans chaque couche du réseau, notons que les neurones d'entrée ne possède pas de fonction d'activation, ils utilisent la fonction identité.

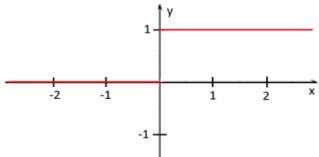
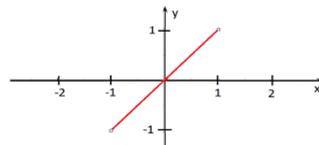
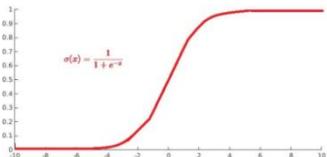
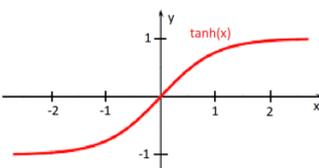
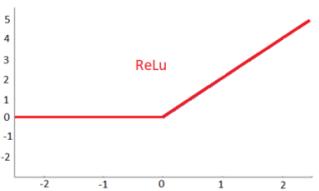
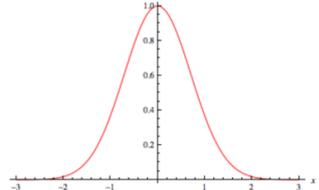
| Fonction | Equation | Représentation graphique |
|---|--|--|
| Heaviside (Echelon) | $f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$ |  |
| Rampe (Linéaire) | $f(x) = x$ |  |
| Sigmoïde | $\sigma(x) = \frac{1}{1 + e^{-x}}$ |  |
| Tangente hyperbolique | $f(x) = \tanh(x)$ |  |
| Unité de rectification linéaire (ReLU) | $f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$ |  |
| Gaussienne | $f(x) = e^{-x^2}$ |  |

Table 2.2 Différents types de fonctions d'activations

2.5 Les réseaux les plus célèbres :

2.5.1 Le perceptron :

Le perceptron (**Figure 2.2**) est inspiré du système visuelle humain basé sur les travaux de McCulloch et Pitts en 1943 et Donald Hebb 1949, c'est le premier réseau évolutif c'est à dire qui a la capacité d'apprendre, inventé par Frank Rosenblatt en 1957, il est utilisé pour la première fois dans le but de reconnaître les lettres de l'alphabet avec des cellules photovoltaïques comme capteurs, Le perceptron utilise la fonction de Heaviside (Echelon) (Tableau 2) comme fonction d'activation.

2.5.2 Algorithme du perceptron :

Entrée : Soit une liste de données d'apprentissage , $(x_1, y_1), \dots, (x_n, y_n)$ où $x_i \in R^d$ et $y_i \in \{-1, 1\}$, Le nombre d' iteration N .

Sortie : Le vecteur de pondération w .

1. Initialiser le vecteur $w \leftarrow 0$
2. **Pour** $i = 1$ à N **faire**
3. **Pour** chaque exemple $(x_i, y_i) \in S$ **faire**
4. Calculer la prédiction $\hat{y}_i = \text{signe}(\langle w, x_i \rangle)$
5. **Si** $\hat{y}_i \neq y_i$ **alors**
6. Ajuster $w \leftarrow (w + x_i)$ si y_i est positive, ou $w \leftarrow (w - x_i)$ si y_i est négative
7. **Fin si**
8. **Fin pour**
9. **Fin pour**

2.5.3 Réseau ADALINE (Adaptive Linear Neuron) :

Le réseau ADALINE est exactement identique au perceptron (**Figure 2.2**) sauf que le réseau ADALINE utilise une fonction d'activation linéaire (Rampe) (Tableau 2) au lieu de la fonction échelon utilisé sur le perceptron. Les réseaux de neurones de type ADALINE utilisent la méthode des moindres carrés, ce qui permet de réduire considérablement les parasites reçus en entrée ; ils sont d'ailleurs employés pour la réduction du bruit en télécommunication [20] .

Le réseau de type ADALINE fait la somme pondéré des entrées qu'il reçoit et ajoute un biais (seuil), et afin d'activer le neurone il utilise une fonction linéaire.

2.6 Le perceptron multicouche (PMC ou MPC) :

Le perceptron multicouche (PMC) appelé en anglais (MPC : Multilayer perceptron) est un type de réseau de neurone artificiel, il est composé au minimum de trois couche , une couche d'entrée , une autre au milieu dite couche cachée, et une dernière en sortie. On peut avoir plusieurs couche cachées, chaque couche est composée de plusieurs nombre de neurone où chaque neurone utilise une fonction d'activation non linéaire , l'information reçue dans la couche d'entrée se propage vers la couche de sortie dans une voie unidirectionnel appelé (feed forward), la couche de sortie représente la sortie du réseau neuronal .

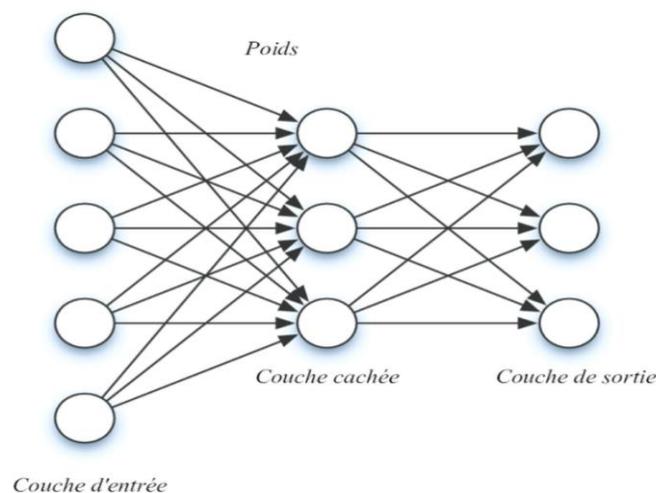


Figure 2.3 - Un perceptron multicouche contenant trois couches [36]

2.7 Réseaux à fonctions radiales de base :

Les réseaux à fonctions de base radiales (RBF : Radial Basis Functions) sont des réseaux de neurones qui utilisent des fonctions de base radiales comme fonction d'activation, les RBF sont usuellement composées de trois couches, les deux couche d'entrée et de sortie utilisent une fonction d'activation linéaire et la couche cachée utilise une fonction d'activation non linéaire. On trouve les RBF dans de nombreuses applications y compris l'approximation de fonctions, la prévision des séries chronologiques, et la classification.

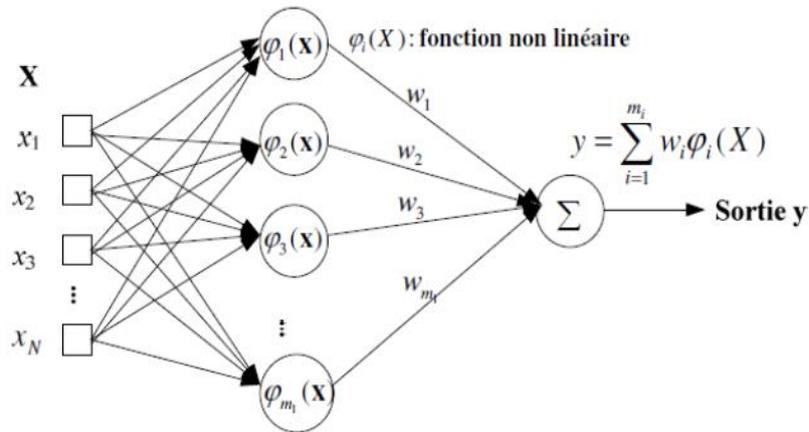


Figure 2.4 - Illustration d'un réseau de neurones à fonctions de base radiale (RBF) [37]

2.8 Réseaux de neurones récurrents :

Les réseaux de neurones récurrents (RNN pour Recurrent Neural Networks) sont des réseaux de neurones présentant des connexions récurrentes, c'est-à-dire qu'une sortie d'un neurone situé dans la couche de sortie ou la couche cachée peut représenter une entrée d'un neurone situé dans couche précédente, Ces neurones interconnectées interagissant non-linéairement [21], comme représenter dans (Figure 2. 5) ci-dessous.

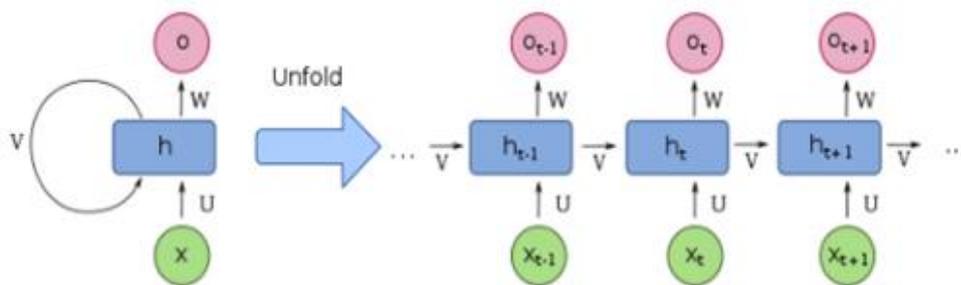


Figure 2.5 - Un exemple d'un réseau récurrent qui se déroule [38]

Les RNNs sont très similaire au fonctionnement du système nerveux, ces réseaux ont la capacité de mémoriser des informations ils peuvent prendre en considération à un instant, un nombre d'itérations passés. Pour cette raison , les RNNs sont particulièrement adaptés aux applications faisant intervenir le contexte, et plus particulièrement au traitement des séquences temporelles comme l'apprentissage , la reconnaissance de formes , la traduction automatique et la génération de signaux, c'est-à-dire quand les données forment une suite et ne sont pas indépendantes les unes des autres[22].

La technique d'apprentissage utilisé pour les RNNs est la (rétropropagation du gradient). Mais la technologie des RNNs présente des insuffisances notamment sur la capacité de mémorisation, il ne peut mémoriser que le passé dit proche, et l'information commence à disparaître au bout d'une cinquantaine d'itération, ce problème appelé la disparition du gradient rend l'entraînement d'un RNNs très compliqué ce n'est que récemment que des méthodes efficaces ont été mises au point comme les LSTM (Long Short Term Memory) à fin de rendre les RNNs plus efficace est simple à entrainer.

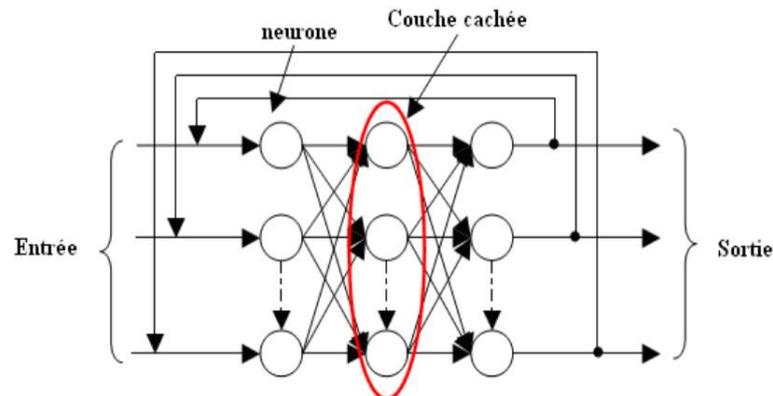


Figure 2.6 - Structure de réseau de neurones récurrent

2.8.1 Les réseaux de mémoire à long terme à court terme (LSTM) :

Les réseaux de mémoire à long terme à court terme appelés habituellement (LSTM : Long Short-Term Memory). Est un modèle d'un réseau de neurone récurrent le plus utilisé, le LSTM a été proposé en 1997 par Sepp Hochreiter et Jürgen Schmid huber ils ont introduit des cellules mémoires, une forme de mémoire intermédiaire permettant de stocker des informations importantes sur une période de temps plus longue que les RNN existants [23]. Pour le but de créer un modèle qui ne souffre pas du problème du débordement ou de l'évanouissement de l'erreur lors de l'apprentissage.

L'idée associée au LSTM est que chaque unité computationnelle est liée non seulement à un état caché h mais également à un état c de la cellule qui joue le rôle de mémoire. Le passage de c_{t-1} à c_t se fait par transfert à gain constant et égal à 1. De cette façon les erreurs se propagent aux pas antérieurs (jusqu'à 1 000 étapes dans le passé) sans phénomène de disparition de gradient [24].

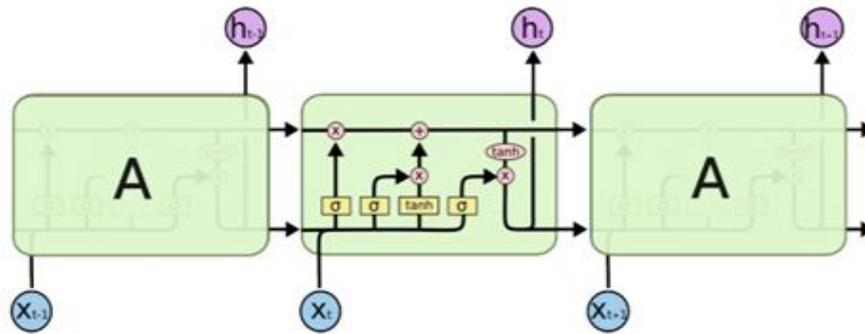


Figure 2.7 - Une chaîne de cellule LSTM [38]

2.9 Les réseaux de neurones à convolution :

Les réseaux de neurones à convolution appelé (CNN : Convolutional Neural Network) est un type de réseau de neurones artificiels acyclique (feed-forward) .

De nos jours cette technique est considérée comme étant la plus performante utilisée pour la classification des images, la détection d'objet et dans d'autres applications dédiées à l'images, proposée par Fukushima et le Cun ces neurones sont connectés entre eux d'une manière inspirée du cortex visuel des animaux [25].

2.9.1 Architecture d'un CNN :

Les CNN sont très différents des autres algorithmes de reconnaissance de formes, notamment les CNN combinent à la fois l'extraction et la classification des fonctionnalités , un CNN est composé de cinq différentes couches : une couche d'entrée, une couche à convolution, une couche pooling, une couche entièrement connectées, et la couche de sortie, comme représenté dans la (Figure 2. 8).

La couche d'entrée :

Cette couche est conçue pour fixer la taille de l'image que l'on cherche à classifier, et aussi la redimensionner si nécessaire.

La couche convolutive :

Celle-ci est conçue pour faire la convolution de l'image avec plusieurs noyaux appris en utilisant des poids partagés.

La couche pooling :

Ca fonction est de réduire la taille de l'image sans perte d'information.

La couche entièrement connectée :

La combinaison des caractéristiques extraites après l'opération pooling.

La couche de sortie :

Cette couche affiche le résultat de la classification.

Ces couches sont regroupées en deux parties bien distinctes, Une première partie dite la partie convolutive est composé de trois premières couches : la couche d'entrée, la couche à convolution, la couche pooling, cette partie joue le rôle d'un extracteur de caractéristiques, et une dernière appelée la partie de classification est composé des deux dernières couches : la couche entièrement connectée, et la couche de sortie, cette partie est conçue pour la classification à partir des valeurs numérique.

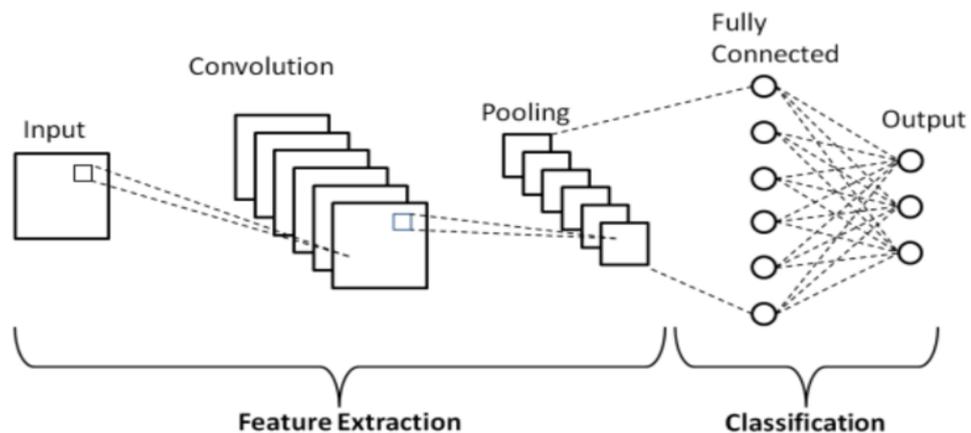


Figure 2.8 - Architecture standard d'un réseau de neurone à convolution CNN [25]

2.10 Conclusion :

Dans ce chapitre nous avons présenté les notions principales des réseaux de neurones, ainsi que le principe de fonctionnement de chaque type, en effet un réseau neuronal ne représente qu'un bloque constitué de plusieurs neurones interconnectées qui fonction sur des règles mathématique, et structuré d'une manière optimale qui permet d'aboutir au résultat (sortie) désirée (le comportement voulu).

- Dans le chapitre suivant nous allons détaillés le fonctionnement des réseaux de neurone et en particulier le PMC et cela en définissant la stratégie d'apprentissage qui est basé sur la méthode de rétropropagation de l'erreur.

Chapitre 3

3 *L'apprentissage*

3.1 Introduction :

Avant de pouvoir utiliser les capacités de classification et d'approximation de fonctions d'un réseau de neurones, il faut le construire, ceci se déroule en quatre temps [26]:

1. La construction de la structure du réseau.
2. La constitution d'une base de données de vecteurs représentant au mieux le domaine à modéliser. Celle-ci est scindée en deux parties : une partie servant à l'apprentissage du réseau (on parle de base d'apprentissage) et une autre partie servant aux tests de cet apprentissage (on parle de base de test).
3. Le paramétrage du réseau par apprentissage.

Au cours de l'apprentissage, les vecteurs de données de la base d'apprentissage sont présentés séquentiellement et plusieurs fois au réseau. Un algorithme d'apprentissage ajuste le poids du réseau afin que les vecteurs soient correctement appris. L'apprentissage se termine lorsque l'algorithme atteint un état stable.

4. La phase de reconnaissance qui consiste à présenter au réseau chacun des vecteurs de la base de test. La sortie correspondante est calculée en propageant les vecteurs à travers le réseau, La réponse du réseau est lue directement sur les unités de sortie et comparée à la réponse attendue. Une fois que le réseau présente des performances acceptables, il peut être utilisé pour répondre au besoin qui a été à l'origine de sa construction.

3.2 Définition de l'apprentissage du réseau de neurone :

L'apprentissage est une phase de développement d'un réseau de neurones durant la quelle une modification du comportement du réseau se fait jusqu'à l'obtention d'un comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement (sortie désirée). Durant cette phase les poids synaptiques seront modifiés, donc d'une manière plus simple, on peut dire que l'apprentissage est une action de mise à jour des poids des connexions dans le but d'atteindre la sortie du réseau pour les exemples présentés en entrée [27] .

3.3 Stratégie d'apprentissage :

En générale, l'apprentissage se réalise sur un intervalle long, pendant laquelle chaque prototype d'entrée (et éventuellement de sortie désirée), est présenté au réseau, plusieurs fois. L'apprentissage se fait en quatre étapes, ces étapes sont répétées jusqu'à la fin de l'apprentissage selon le schéma suivant [44] :

Etape 1 : Initialisation des poids du réseau a des petites valeurs aléatoires.

Etape 2 : Présentation du prototype d'entrée au RNA. Des valeurs de sortie correspondants sont ensuite calculées après propagation d'activation.

Etape 3 : Calcul de l'erreur : ce terme s'emploi plus volontiers pour un apprentissage supervise ou l'erreur tient compte de la différence entre l'activation des neurones de sortie et de la sortie désiré (lie au prototype d'entrée).

Etape 4 : Calcul du vecteur de correction : à partir des valeurs d'erreurs, on détermine alors la correction à apporter aux poids des connexions et au seuils des neurones .

3.4 Types d'apprentissage :

L'apprentissage est l'étape la plus importante dans le fonctionnement d'un réseau de neurones, c'est un processus dynamique et itératif permettant de modifier les paramètres d'un réseau en réaction avec le stimulus qu'il reçoit de son environnement.

Il existe plusieurs types d'apprentissage, nous présentons les deux plus importants et les plus utilisés [28]:

3.4.1 L'apprentissage supervisé :

Dans ce type, une information précise sur la sortie désirée est disponible. Le réseau apprend par présentation de pair d'entrée / sortie. Durant l'apprentissage, les valeurs de sorties désirées sont comparées à celles produites par le réseau, l'erreur résultante est alors utilisée pour l'ajustement des poids des connexions. La règle du delta en méthode rétropropagation est utilisée telle qu'elle dans les réseaux multicouches que nous détaillerons par la suite [29].

3.4.2 L'apprentissage non-supervisé :

Dans cet apprentissage, aucune information sur la sortie désirée du réseau n'est disponible. Ainsi, le réseau manipule des données qui lui sont présentées en entrée et cherche à extraire quelques propriétés qui formeront les sorties du réseau. L'extraction de ces propriétés dépend de la règle d'apprentissage utilisée dans le réseau [29].

3.5 Réseau de neurone appliqué à notre cas (perceptron multicouche MLP):

Le choix de réseau de neurone est lié principalement à la nature du problème étudié, dans notre cas, on va utiliser le Perceptron multicouche, qui est le réseau de neurone le plus connu et le plus utilisé pour réaliser les tâches de classification et de prédiction.

En effet, les réseaux monocouche (neurone formel) sont limité et ne peuvent pas résoudre des problèmes complexes, comme ceux des données non linéairement séparable, un seul neurone échoue. Pour surmonter ces limitations on utilise un réseau a multicouche ou la sortie d'un neurone est affecté à une ou plusieurs entrées d'autres neurones d'où le réseau possédera une ou plusieurs couches cachées, ce qui lui apportera une richesse et une puissance à ces capacités. Donc le problème de séparation linéaire est désormais résolu.

Pour son apprentissage, ce réseau utilise le modèle d'apprentissage supervisé par la méthode de rétro propagation de l'erreur.

Ces réseaux ont été utilisés pour des applications diverses : la reconnaissance de formes, l'aide à la décision, traitement d'images, traitement de signal ...etc.

Modèle mathématique de réseau utilisé :

Nous commençons tout d'abord par un bref aperçu sur la modélisation mathématique d'un neurone simple (neurone formel). La figure suivante montre sa représentation graphique :

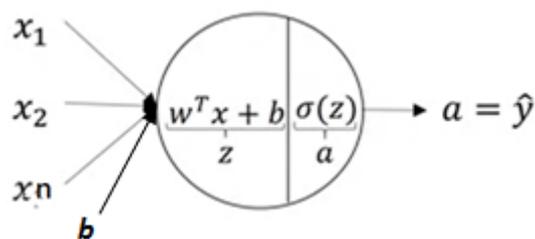


Figure 3.1 - Représentation graphique d'un neurone simple.

Ce neurone fait un calcul en deux étapes, la première est de calculer $z = wx + b$, la deuxième est de calculer l'estimer de la sortie désiré y donné par : $a = f(z) = \sigma(z)$ résultat finale du neurone noté également \hat{y} .

Où :

x : Caractéristiques d'entrées

w : Coefficient de poids

b : Biais

f ou σ : Fonction d'activation

Un réseau de neurone multicouche représenté par la figure 3.2, où les neurones sont groupés en couche, effectue les mêmes étapes de calcul pour chaque nœud dans la couche cachée de la même manière que pour un neurone formel.

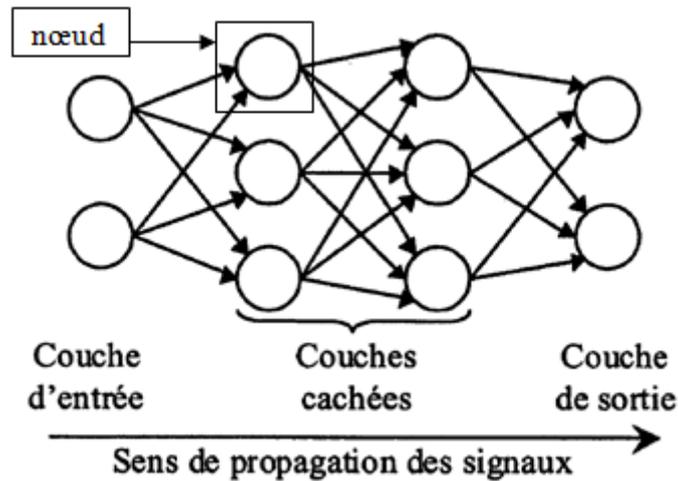


Figure 3.2 - Exemple d'un réseau de neurone multicouche.

A partir de la représentation de la figure 3.2, nous pouvons déterminer trois caractéristiques importantes d'un réseau multicouche :

1. Les neurones d'une même couche ne sont pas interconnectés c'est-à-dire le calcul de la sortie de neurone peut se faire seulement avec la sortie de la couche précédente.
2. Chaque neurone d'une couche donnée (c) est relié uniquement aux neurones de la couche précédente ($c-1$)
3. Une seule et unique fonction d'activation pour les neurones d'une couche donnée.

En tenant compte de l'équation du neurone formelle et suivant les caractéristiques précédemment décrites, nous pouvons obtenir les équations équivalentes pour une couche (c) sous la forme scalaire suivante :

$$a_s^c = f^c(b_s^c + \sum_{i=1}^{n_{c-1}} w_{si}^c a_i^{c-1}) \dots \dots \dots (3.1)$$

- a_s^c : La sortie de neurone s de la couche c
- f^c : Fonction d'activation de la couche c
- b_s^c : Coefficient du neurone s de la couche c
- w_{si}^c : Coefficient de pondérations de neurone s , de la couche c , d'entrée i
- a_i^{c-1} : L'entrée i de la couche c (ou sortie de neurone i de la couche $c - 1$)

On peut aussi écrire cette équation sous la forme matricielle suivante :

$$A^c = f^c(B^c + W^c a^{c-1}) \dots \dots \dots (3.2)$$

- W^c : matrice des coefficient de pondérations des entrées
- B^c : vecteur des coefficient ou biais

3.6 Apprendre avec le Retro-Propagation du gradient :

L'idée générale de l'apprentissage d'un réseau de neurone est d'ajuster les paramètres w (poids synaptique) et b (le biais) afin de trouver le minimum d'une fonction d'erreur globale (sortie désiré – sortie calculé) appelé la fonction de cout. Ce processus est itératif c'est-à-dire à chaque couple (entré, sortie) nous allons refaire ce calcul jusqu' à l'obtention d'une configuration optimisée de ces paramètres qui permettent d'aboutir de ce fait au comportement désiré.

Pour cela on va se servir de l'algorithme de rétro propagation du gradient désigné en anglais « Back propagation » qui est le plus utilisé pour l'apprentissage supervisé. Cette méthode consiste à calculer le gradient de l'erreur pour chaque neurone du réseau (nœud).

3.6.1 Descente de gradient :

L'algorithme de descente de gradient est un algorithme d'optimisation qui converge vers le minimum d'une fonction convexe c'est-à-dire une fonction qui converge vers un seul minima.

Après avoir choisi aléatoirement des valeurs initiales des poids de connexions $w_0, w_1 \dots w_j$, on applique à chacun de ces poids la formule de l'algorithme de descente de gradient pour les mettre à jour comme suit :

$$w_j^{(t+1)} = w_j^{(t)} - \eta \frac{\partial \text{Erreur}(d(x)_i - y_i)}{\partial w_j^{(t)}} \dots \dots \dots (3.3)$$

- *Erreur* $(d(x)_i^{(p)} - y_i^{(p)})$: fonction de cout local définie par :

$$E_p = \frac{1}{2} \sum_{i=1}^J \left(d(x)_i^{(p)} - y_i^{(p)} \right)^2 \dots \dots \dots (3.4)$$

Avec : $d(x)_i^{(p)}$ la p -ième sortie désiré, et $y_i^{(p)}$ la p -ième sortie calculé.

La fonction de cout global la plus utilisé et dont nous allons utiliser lors de notre

étude est la fonction d'erreur quadratique moyenne: $E_{moyenne} = \frac{1}{Nc} \sum_{p=1}^{Nc} E_p$ où Nc

est le nombre de couples donné en entrainement au réseau de neurones et E_p

représente la p -ième erreur d'apprentissage, dans la suite pour simplifier on va utiliser

l'écriture $(d(x)_i - y_i)$ au lieu de $(d(x)_i^{(p)} - y_i^{(p)})$

- η : l'hyper paramètre du réseau appelé vitesse d'apprentissage, est le pas entre deux itérations, le choix de sa valeur joue un rôle très important
 - ✓ Si η est grand on peut s'éloigner de notre objectif donc l'algorithme risque de ne pas converger vers une solution optimale.
 - ✓ Si η est trop petit, alors on risque de mettre un temps infini pour atteindre notre objectif.
- $\frac{\partial \text{Erreur}(d(x)_j - y_j)}{\partial w_j}$: Descente de gradient ou la dérivé partielle par rapport au poids synaptiques.

On itère ces calcul sur l'intégralité de nos données jusqu'à ce que la descente de gradient converge vers le minimum.

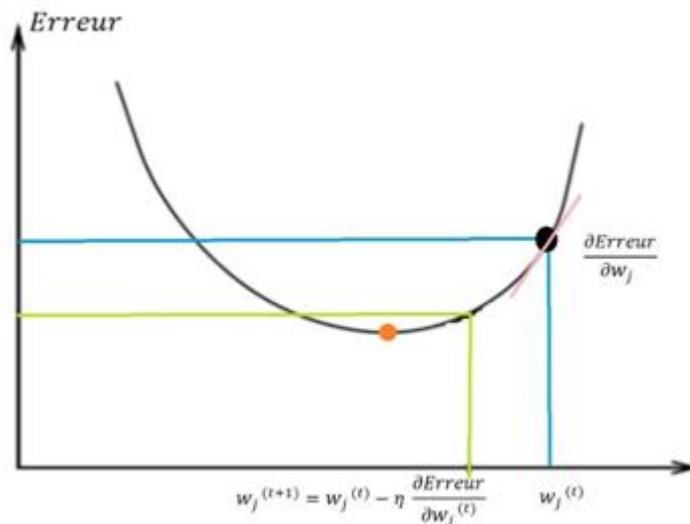


Figure 3.3 – Représentation de la minimisation de la fonction d'erreur par descente de gradient

- Cette règle est facilement applicable au cas d'un perceptron (neurone simple) mais pour un réseau multicouche les calculs de dérivés partielle peut paraître ardu, c'est pour ça on va utiliser la méthode de rétro propagation du gradient qui va permettre de faciliter grandement les calculs.

3.6.2 Rétro propagation du gradient :

Il s'agit de décomposer le dérivé partiel de l'erreur par la règle de dérivation en chaîne.

Prenons un réseau à une seule couche cachée, L'équation de la sortie de la couche caché est donné par :

$$z_m = h(r_m) = h(z_{om} + \sum_{n=1}^M x_n w_{nm}) \dots \dots \dots (3. 1)$$

Où :

h : fonction d'activation des neurones de la couche caché

w_{nm} : poids reliant n neurones d'entrée avec m neurone cachés

z_{om} : biais de la couche caché du neurone m

L'équation de la sortie du réseau est donnée par :

$$f_j = g(s_j) = g\left(f_{oj} + \sum_{j=1}^J z_m u_{mj}\right) \dots \dots \dots (3.6)$$

Où :

g : Fonction d'activation des neurones de sortie .

u_{mj} : Poids reliant m neurone caché avec j neurones de sortie .

f_{oj} : Biais de la couche de sortie du neurone j .

La mise à jour des poids et des biais de la caché et ceux de la couche de sortie est donné par :

$$u_{mj}^{(t+1)} = u_{mj}^{(t)} - \eta \frac{\partial \text{Erreur}}{\partial u_{mj}^{(t)}} \dots \dots \dots (3.7)$$

$$w_{nm}^{(t+1)} = w_{nm}^{(t)} - \eta \frac{\partial \text{Erreur}}{\partial w_{nm}^{(t)}} \dots \dots \dots (3.8)$$

$$z_{om}^{(t+1)} = z_{om}^{(t)} - \eta \frac{\partial \text{Erreur}}{\partial z_{om}^{(t)}} \dots \dots \dots (3.9)$$

$$f_{oj}^{(t+1)} = f_{oj}^{(t)} - \eta \frac{\partial \text{Erreur}}{\partial f_{oj}^{(t)}} \dots \dots \dots (3.10)$$

Le calcul des dérivés partiels sera par la méthode de dérivation en chaînes comme suit :

$$\frac{\partial \text{Erreur}}{\partial u_{mj}} = \frac{\partial \text{Erreur}}{\partial f_j} \frac{\partial f_j}{\partial s_j} \frac{\partial s_j}{\partial u_{mj}} \dots \dots \dots (3.11)$$

$$\frac{\partial \text{Erreur}}{\partial w_{mn}} = \frac{\partial \text{Erreur}}{\partial z_m} \frac{\partial z_m}{\partial r_m} \frac{\partial r_m}{\partial w_{mn}} \dots \dots \dots (3.12)$$

$$\frac{\partial \text{Erreur}}{\partial f_{oj}} = \frac{\partial \text{Erreur}}{\partial f_j} \frac{\partial f_j}{\partial s_j} \frac{\partial s_j}{\partial f_{oj}} = \frac{\partial \text{Erreur}}{\partial f_j} \frac{\partial f_j}{\partial s_j} \dots \dots \dots (3.13)$$

$$\frac{\partial \text{Erreur}}{\partial z_{om}} = \frac{\partial \text{Erreur}}{\partial z_m} \frac{\partial z_m}{\partial r_m} \frac{\partial r_m}{\partial z_{om}} = \frac{\partial \text{Erreur}}{\partial z_m} \frac{\partial z_m}{\partial r_m} \dots \dots \dots (3.14)$$

Puisque : $\frac{\partial s_m}{\partial f_{oj}} = \frac{\partial z_m}{\partial r_m} = 1$

Cas particulier de la fonction Sigmoid $\sigma \equiv g \equiv f$:

i. Calcul de $\frac{\partial \text{Erreur}}{\partial u_{mj}}$:

$$\text{On a : } \frac{\partial \text{Erreur}}{\partial u_{mj}} = \frac{\partial \text{Erreur}}{\partial f_j} \frac{\partial f_j}{\partial s_j} \frac{\partial s_j}{\partial u_{mj}} \dots\dots\dots (3.15)$$

Le terme $\frac{\partial \text{Erreur}}{\partial f_j}$ peut-être décomposé comme suit :

$$\frac{\partial \text{Erreur}}{\partial f_j} = \frac{\partial \text{Erreur}}{\partial (d(x)_j - y_j)} \frac{\partial (d(x)_j - y_j)}{\partial f_j} \dots\dots\dots (3.16)$$

$$\text{où : } \frac{\partial \text{Erreur}}{\partial (d(x)_j - y_j)} = \frac{\partial \left[\frac{1}{2} \sum_{i=1}^J (d(x)_i - y_i)^2 \right]}{\partial (d(x)_j - y_j)} \dots\dots\dots (3.17)$$

$$= \frac{1}{2} \frac{\partial (d(x)_j - y_j)^2}{\partial (d(x)_j - y_j)} = (d(x)_j - y_j) \dots\dots\dots (3.18)$$

$$\text{et } \frac{\partial (d(x)_j - y_j)}{\partial f_j} = \frac{\partial (d(x)_j - f_j)}{\partial f_j} = (-1) \dots\dots\dots (3.19)$$

$$\text{Donc : } \frac{\partial \text{Erreur}}{\partial f_j} = -(d(x)_j - y_j) \dots\dots\dots (3.20)$$

Calculons à présent le terme $\frac{\partial f_j}{\partial s_j}$:

$$\text{On a : } f_j = g(s_j) = g \left(f_{oj} + \sum_{j=1}^J z_m u_{mj} \right) \dots\dots\dots (3.21)$$

Avec g fonction d'activation de type **sigmoid** σ ayant la forme suivante :

$$f_j = \sigma(s_j) = \frac{1}{1 + e^{-s_j}} \dots\dots\dots (3.22)$$

$$\frac{\partial f_j}{\partial s_j} = \frac{\partial \left[\frac{1}{1 + e^{-s_j}} \right]}{\partial s_j} \dots\dots\dots (3.23)$$

$$= \frac{e^{-s_j}}{(1 + e^{-s_j})^2} \dots\dots\dots (3.24)$$

$$= \frac{1}{(1 + e^{-s_j})} \cdot \frac{[(1 + e^{-s_j}) - 1]}{(1 + e^{-s_j})} \dots\dots\dots (3.25)$$

$$\frac{\partial f_j}{\partial s_j} = f_j \cdot (1 - f_j) \dots\dots\dots (3.26)$$

Et finalement le terme :

$$\frac{\partial s_j}{\partial u_{mj}} = \frac{\partial \left[z_{om} + \sum_{j=1}^J z_m u_{mj} \right]}{\partial u_{mj}} = z_m \dots \dots \dots (3.2)$$

On obtient donc :

$$\frac{\partial \text{Erreur}}{\partial u_{mj}} = -(d(x)_j - y_j) \cdot f_j \cdot (1 - f_j) \cdot z_m \dots \dots \dots (3.3)$$

Et la formule de mise à jour de $u_{mj}^{(t+1)}$ peut être exprimée par :

$$u_{mj}^{(t+1)} = u_{mj}^{(t)} - \eta \cdot \delta_j \cdot z_m \dots \dots \dots (3.4)$$

Où : $\delta_j = -(d(x)_j - y_j) \cdot f_j \cdot (1 - f_j)$ représente le gradient local

ii. Calcul de $\frac{\partial \text{Erreur}}{\partial w_{mn}}$:

$$\text{On a : } \frac{\partial \text{Erreur}}{\partial w_{mn}} = \frac{\partial \text{Erreur}}{\partial z_m} \frac{\partial z_m}{\partial r_m} \frac{\partial r_m}{\partial w_{mn}} \dots \dots \dots (3.5)$$

Le 2ème et le 3ème sont semblables aux termes précédents pour la couche de sortie :

$$\frac{\partial z_m}{\partial r_m} = z_m \cdot (1 - z_m) \dots \dots \dots (3.6)$$

$$\frac{\partial r_m}{\partial w_{mn}} = x_n \dots \dots \dots (3.7)$$

Pour le terme $\frac{\partial \text{Erreur}}{\partial z_m}$ on aura :

$$\frac{\partial \text{Erreur}}{\partial z_m} = \sum_{j=1}^J \frac{\partial \text{Erreur}(d(x)_j - y_j)}{\partial z_m} = \sum_{j=1}^J \frac{\partial \text{Erreur}}{\partial s_j} \frac{\partial s_j}{\partial z_m} \dots \dots \dots (3.33)$$

On a : $\frac{\partial \text{Erreur}}{\partial s_j} = \frac{\partial \text{Erreur}}{\partial f_j} \frac{\partial f_j}{\partial s_j}$

Avec : $\frac{\partial \text{Erreur}}{\partial f_j} = -(d(x)_j - y_j)$ et $\frac{\partial f_j}{\partial s_j} = f_j \cdot (1 - f_j)$ calculée précédemment

on obtient : $\frac{\partial \text{Erreur}}{\partial s_j} = -(d(x)_j - y_j) \cdot [f_j \cdot (1 - f_j)] = \delta_j \dots \dots \dots (3.34)$

$$\text{Et finalement : } \frac{\partial s_j}{\partial z_m} = \frac{\partial \left[z_{om} + \sum_{j=1}^J z_m u_{mj} \right]}{\partial z_m} = u_{mj} \dots \dots \dots (3.35)$$

$$D'où : \frac{\partial \text{Erreur}}{\partial z_m} = \left(\sum_{j=1}^J \delta_j \cdot u_{mj} \right) \dots \dots \dots (3.36)$$

On obtient donc notre dérivé partielle :

$$\frac{\partial \text{Erreur}}{\partial w_{mn}} = \left(\sum_{j=1}^J \delta_j \cdot u_{mj} \right) \cdot z_m \cdot (1 - z_m) \cdot x_n \dots \dots \dots (3.37)$$

Le poids $w_{nm}^{(t+1)}$ sera mis à jour par la formule suivante :

$$w_{nm}^{(t+1)} = w_{nm}^{(t)} - \eta \cdot \delta k \cdot x_n \dots \dots \dots (3.38)$$

Où : $\delta k = \delta j \cdot u_{mj} \cdot z_m \cdot (1 - z_m)$ représente le gradient local.

iii. Calcul de $\frac{\partial Erreur}{\partial f_{oj}}$:

$$\text{On a : } \frac{\partial Erreur}{\partial f_{oj}} = \frac{\partial Erreur}{\partial f_j} \frac{\partial f_j}{\partial s_j} = -(d(x)_j - y_j) \cdot f_j \cdot (1 - f_j) = \delta j \dots \dots \dots (3.39)$$

$$\text{D'où la formule de mis jour du biais } f_{oj}^{(t+1)} \text{ suivante : } f_{oj}^{(t+1)} = f_{oj}^{(t)} - \eta \delta j \dots \dots (3.40)$$

iv. Calcul de $\frac{\partial Erreur}{\partial z_{om}}$:

$$\text{On a : } \frac{\partial Erreur}{\partial z_{om}} = \frac{\partial Erreur}{\partial z_m} \frac{\partial z_m}{\partial r_m} = \left(\sum_{j=1}^J \delta j \cdot u_{mj} \right) \cdot z_m \cdot (1 - z_m) = \delta k \dots \dots \dots (3.41)$$

$$\text{D'où la formule du biais } z_{om}^{(t+1)} \text{ suivante : } z_{om}^{(t+1)} = z_{om}^{(t)} - \eta \cdot \delta k \dots \dots \dots (3.42)$$

- Récapitulons le processus d'apprentissage d'un réseau de neurone avec la méthode de rétro propagation de gradient par le schéma global suivant :

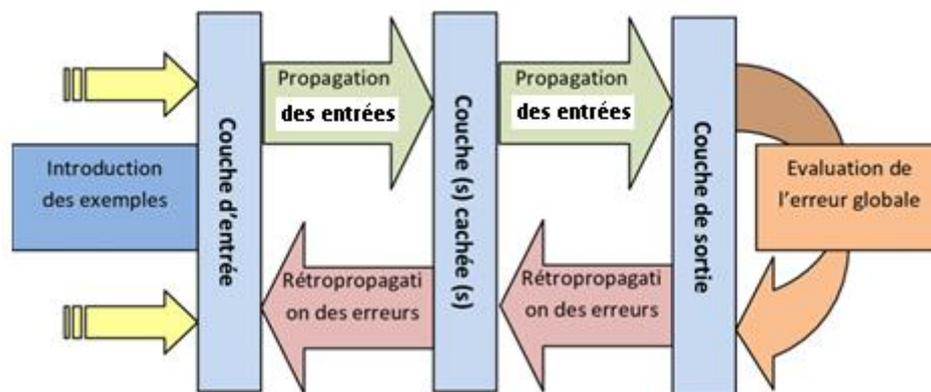


Figure 3.4 - Schéma global de la méthode de rétro propagation [41]

3.7 Conclusion :

Les réseaux de neurones sont considérés comme un outil très puissant dans l'apprentissage automatique, ils sont connus par leur puissance d'apprentissage. Dans ce chapitre nous avons présenté leur principe de fonctionnement, ainsi leur type d'apprentissage, et nous avons illustré la méthode de rétropropagation utilisé pour entrainer efficacement un réseau de neurone multicouche à travers la technique du gradient et la règle de chaines.

Chapitre 4

4 Application

4.1 Introduction :

Dans ce chapitre nous nous intéressons à la sélection des paramètres adéquats d'un perceptron multicouche via plusieurs simulations qui portent essentiellement sur la fonction d'apprentissage et la fonction d'activation à fin d'obtenir une classification optimale et précise des fleurs d'iris.

- Dans la première simulation on fera une étude comparative des performances de classification de fleurs d'iris entre deux méthodes de rétropropagation, celle dite de Levenberg-Marquardt et celle du gradient conjugué.
- Dans la deuxième simulation on effectuera une permutation de fonction d'activation sigmoïde et linéaire entre la couche cachée et celle de sortie, en évaluant les performances du perceptron multicouche pour obtenir la meilleure précision de classification possible.

4.2 La description du problème :

La base de données iris de Fisher, appelée en anglais iris flower data set ou Fisher's iris data, est présentée en 1936 par Ronald Fisher dans son article «The use of multiple measurements in taxonomic problems» est l'une des bases de données la plus populaire et la plus connue dans le domaine de la classification par les réseaux de neurones.

IRIS data set comprend trois classes, chaque classe contient 50 objets, chaque un de ces classes se réfère à un type de fleur d'iris. La liste des attributs présents dans la base IRIS peut être décrite comme catégorielle, nominale et continue.

Les experts ont mentionné qu'il n'y avait aucune valeur manquante dans aucun attribut de cet ensemble de données. L'ensemble de données est complet. La première des classes se distingue linéairement des deux autres, les deux secondes n'étant pas linéairement séparables l'une de l'autre, chaque fleur parmi les 150 est caractérisée par quatre attributs numériques figure comme suit :

- longueur sépale
- largeur sépale
- longueur des pétales
- largeur des pétales

On se rend compte que la largeur des pétales est toujours inférieure à la longueur des pétales et la largeur des sépales également inférieure à la longueur des sépales.

| Caractéristiques | La taille minimale de la caractéristiques | La taille maximale de la caractéristique |
|--------------------|---|--|
| Longuer de sépale | 4.3 | 7.9 |
| Largeur de sépale | 2.0 | 4.4 |
| Longueur de sépale | 1.0 | 6.9 |
| Longueur de pétale | 0.1 | 2.5 |

Table 4. 1 - Information générales sur les caractéristiques d'entrée



Figure 4.1 - Les trois types de fleur iris [39]

le cinquième attribut est l'attribut prédictif qui est l'attribut de classe qui signifie le nom de classe d'identification, dont chacun est l'un des suivants: IRIS Setosa, IRIS Versicolour ou IRIS Virginica, la figure cidesous reprénte une nuage de points qui dépent des quatre variables de la base de données iris .

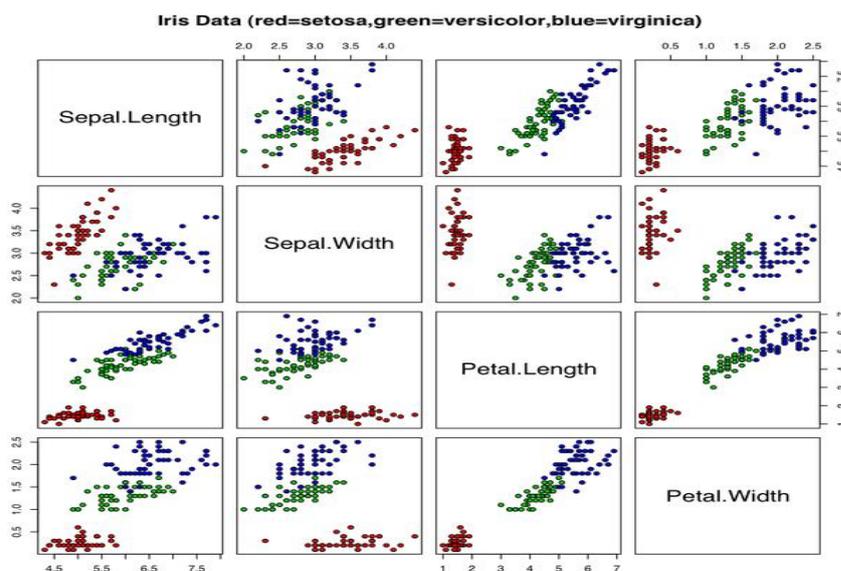


Figure 4.2 - Nuage de points de la base de données iris [40]

4.3 Logiciels :

Le matlab est un langage de programmation et de simulation de haut niveau créé par Cleve Moler à la fin des années 1970, et développé par la société the mathworks, sa première version était en 1984, il fournit des constructions qui permettent une programmation claire à petite et à grande échelle.

Le matlab, prend en charge de multiples paradigmes de programmation, y compris orientés objet, impératifs, fonctionnels et procéduraux, Il permet l'utilisation d'un ensemble de fonctions prédéfinies appelées boîtes à outils et des fonctions et programmes définis par l'utilisateur. MATLAB Neural Network Toolbox fournit des outils pour la conception, la mise en œuvre, la visualisation et la simulation de réseaux de neurones pour diverses applications.

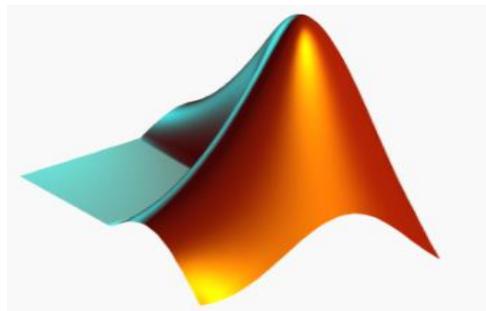


Figure 4.3 - logo du matlab

4.4 Configuration des données :

La base de données iris est sauvegardée dans le fichier matlab sous `irisdata.mat`, le chargement du fichier se fait par l'utilisation de la commande `load fisheriris`, qui permet de charger toutes les entrées et les sorties de la base de données comme suit :

4.4.1 Meas 150×4 : 150 lignes par 4 colonnes chaque ligne contient 4 attributs qui représentent les dimensions d'une fleur iris.

4.4.2 Species 150×1 : 150 lignes par une seule colonne divisé par trois classes de fleur identique, contient le nom de la fleur **iris setosa**, **iris versicolor** et **iris virginica**.

4.5 Les ensembles de création d'un modèle de classification par un réseau de neurone :

La création d'un modèle de classification par réseau de neurone provient généralement de plusieurs ensembles de données. En particulier, trois ensembles sont couramment recommandés, training set, validation set et test set, chaque un de ces ensembles joue un rôle important pour l'obtention d'un modèle de classification finale, apte à prédire des résultats les plus juste possible, c'est trois ensembles sont décrits comme suit :

4.5.1 Training set :

Celui-ci va absorber la grande quantité de données. En effet, c'est sur cette ensemble que le réseau va itérer pour adapter les paramètres des réseaux (poids et biais) et les ajuster au mieux, certaines règles préconisent qu'il soit composé de 80% des données .

4.5.2 Validation set :

Cette phase est appelée à la fin de chaque itération d'entraînement pour une seule fois, elle est composée de 10% à 15% des données, afin de nous permettre d'équilibrer le modèle.

c-a-d affiner les paramètres du modèle, lors de la phase d'ajustement de modèle sur l'ensemble de données d'entraînement

4.5.3 Test set :

Ce dernier est composé 10% à 15% des données, il a un rôle qui est différent des autres phases, puisque il ne servira pas à ajuster les données, En effet, il va avoir pour rôle d'évaluer le réseau sous sa forme finale, et de voir comment il arrive à prédire [30] .

4.6 Résultats expérimentale et discussion :

Toutes les simulations ont été réalisées grâce au logiciel MATLAB 2019 b, sur un PC de système d'exploitation Windows 10 (64 bits) avec un processeur de type i5-6200 CPU @ 2.30 GHz ayant 4 Go de RAM.

Le matlab contient des algorithmes d'entraînement et des fonctions d'activation prête à l'utilisation, ce qui à faciliter grandement notre tâche de classification.

Afin d'appliquer notre classifieur neuronal avec succès à notre base de données IRIS nous avons en premier lieu optés pour le choix d'un perceptron multicouche possédant quatre entrées, une seule couche caché contenant 4 neurones et une couche de sortie produisant un vecteur $3 * 1$ correspondant aux trois classes définies comme suit :

- Le vecteur de sortie de la classe **Setosa** représenté par : $[1, 0, 0]$
- Le vecteur de sortie de la classe **Versicolor** représenté par : $[0, 1, 0]$
- Le vecteur de sortie de la classe **Virginica** représenté par : $[0, 0, 1]$

Deuxièmement pour le processus d'apprentissage, les données ont été divisées en trois ensembles : le premier utilisé pour la phase d'entraînement (70%), le second pour la phase de validation (15%) et le dernier pour la phase de tests (15%), les paramètres d'entraînement de base du système sont le nombre maximum d'itérations qui est de 100, et le taux d'apprentissage qui est de 0.1.

Et afin d'augmenter la précision de notre classifieur nous avons fait deux propositions :

4.6.1 La première proposition :

Notre première proposition consiste à utiliser deux fonctions d'apprentissage différentes `trainlm` et `trainscg` afin de choisir la meilleure des deux en terme de performance c-a-d qui permet d'obtenir la meilleure précision, ces deux fonctions sont définies comme suit :

4.6.1.1 La fonction de rétropropagation de Levenberg–Marquardt (`trainlm`) :

Est une fonction d'entraînement basée sur l'algorithme de quasi newton elle est considérée comme la plus rapide dans les réseaux de taille moyenne, elle localise le minimum d'une fonction multivariée qui peut être exprimée comme la somme des carrés de fonctions réelles non linéaires. C'est une technique itérative qui fonctionne de telle manière que la fonction de performance sera toujours réduite à chaque itération de l'algorithme [31].

4.6.1.2 La fonction de rétropropagation du Gradient conjugué mis à l'échelle (`trainscg`) :

Appelée en anglais (`scaled conjugate gradient`), cette fonction ne nécessite pas de recherche de ligne à chaque étape d'itération comme les autres fonctions d'entraînement du gradient conjugué. Grâce à l'utilisation d'un mécanisme de mise à l'échelle de la taille, ce qui évite une recherche de ligne longue par itération d'apprentissage. Ce mécanisme rend l'algorithme plus rapide que tout autre algorithme du second ordre. La fonction `trainscg` nécessite plus d'itérations pour converger que les autres algorithmes de gradient conjugué, mais le nombre d'opérations à chaque itération est considérablement réduit car aucune recherche de ligne n'est effectuée [32].

4.6.1.3 Simulation 1 : entrainement de notre perceptron multicouche avec la fonction (trainlm), la couche cachée utilise la fonction sigmoïde et la couche de sortie utilise la fonction d'activation linéaire.

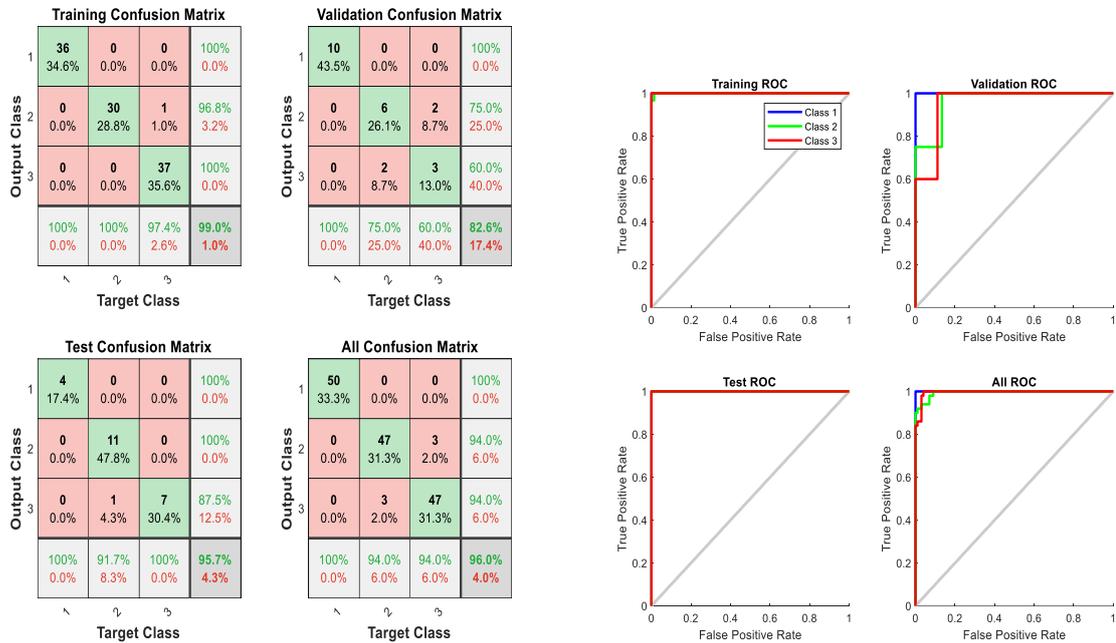


Figure 4.4-La matrice de confusion de la simulation 1. Figure 4.5-la courbe ROC pour la simulation1.

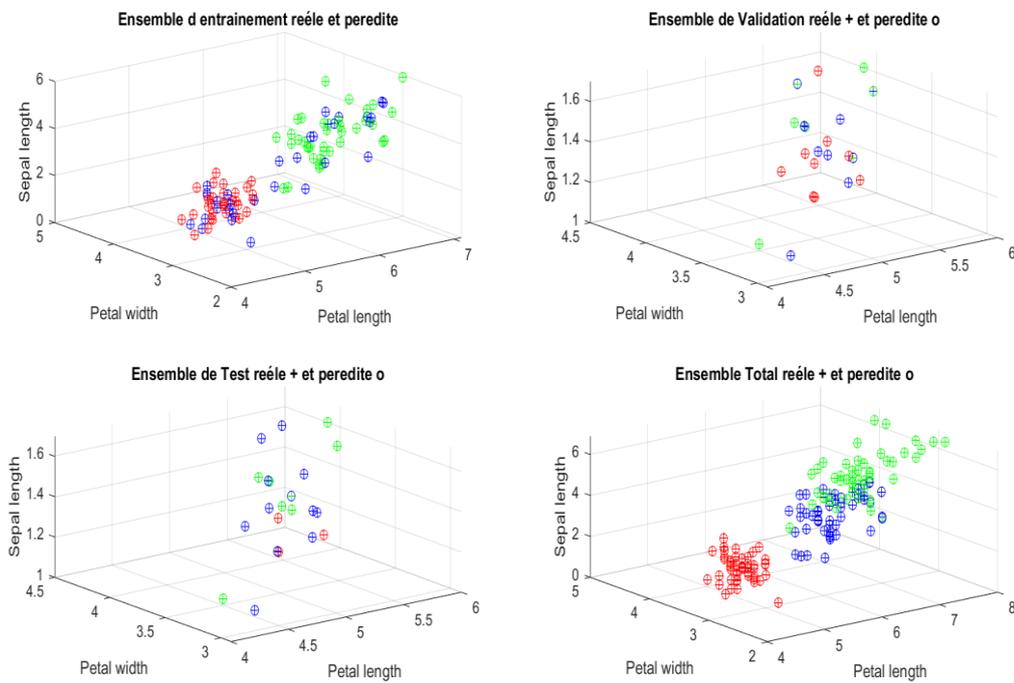


Figure 4.6 – Représentation des classe réelle et prédite en 3D de la simulation1.

4.6.1.4 Simulation 2 : entraînement d'un perceptron multicouche avec la fonction (traincsg), la couche cachée utilise la fonction sigmoïde et la couche de sortie utilise la fonction d'activation linéaire.

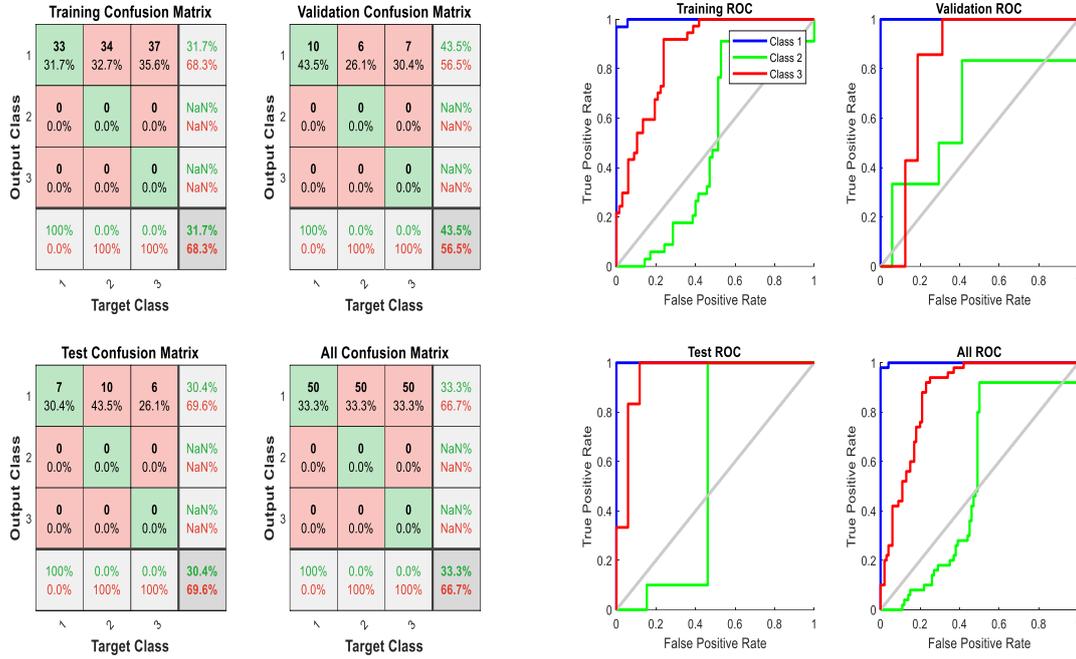


Figure 4.7-La matrice de confusion de la simulation 2 Figure 4.8-La courbe ROC de la simulation 2.

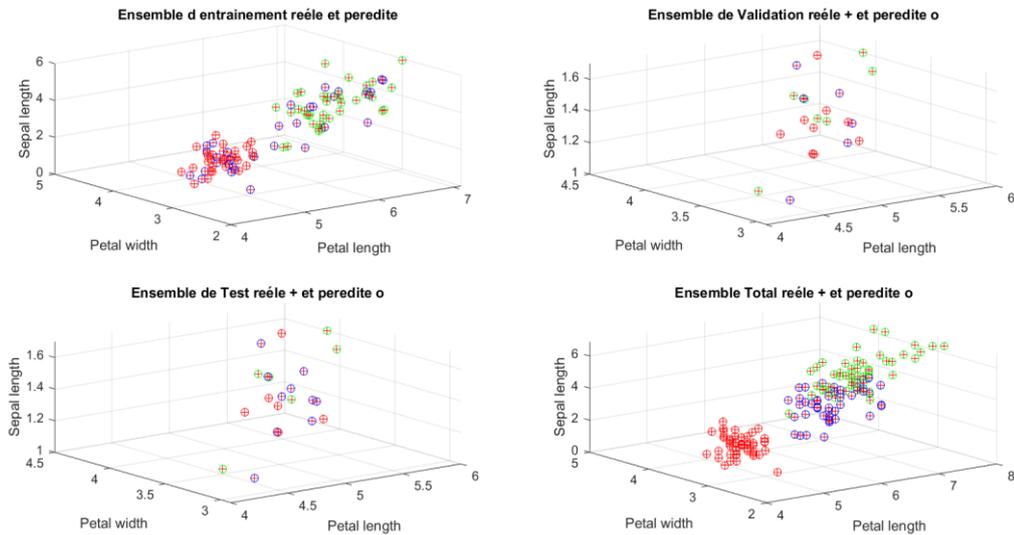


Figure 4.9 – Représentation des classe réelle et prédite en 3D de la simulation2.

| La fonction utilisée | Précision de classification de la phase d'entraînement | Précision de classification de la phase de validation | Précision de classification de la phase de test | Précision Totale du MLP |
|--|--|---|---|-------------------------|
| La fonction de levenberg-marquardt | 99% | 82.6% | 95.7% | 96% |
| La fonction de gradient conjugué mis à l'échelle | 31.7% | 43.5% | 30.4% | 33.3% |

Table 4. 2 -Tableau des résultats de la première proposition

4.6.1.5 Discussion et Comparaison des résultats :

Les résultats de la simulations mentionnés dans le tableau ci dessus obtenus à partir de la matrice de confusion montrent que la fonction de levenberg mrquardt est plus performante que la fonction de gradient conjugué mis à l'échelle dans les trois phases de classification de notre réseau (PMC), et plus précisément dans la phase de test où l'on remarque que la fonction de levenberg marquardt a atteint une précision de 95.7% contre 30.4% pour La fonction de gradient conjugué mis à l'échelle.

4.6.2 La deuxième proposition :

Cette proposition consiste à permuter les fonctions d'activation utilisée dans la couche cachée et la couche de sortie et voir qu'elle combinaison nous permet d'obtenir la meilleure précision. C'est se qu'on verra dans les simulations suivante :

4.6.2.1 **Simulation 3** : dans cette simulation on va utiliser la fonction linéaire pour les deux couches.

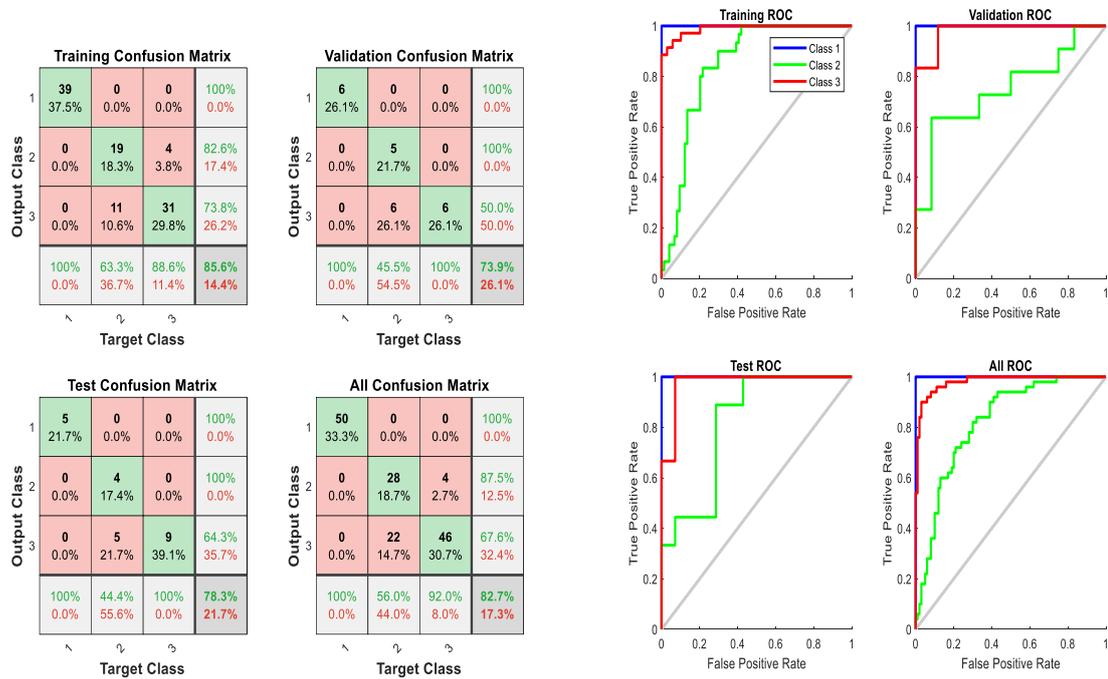


Figure 4.10- La matrice de confusion de la simulation 3. figure 4.11 –La courbe ROC de la simulation 3.

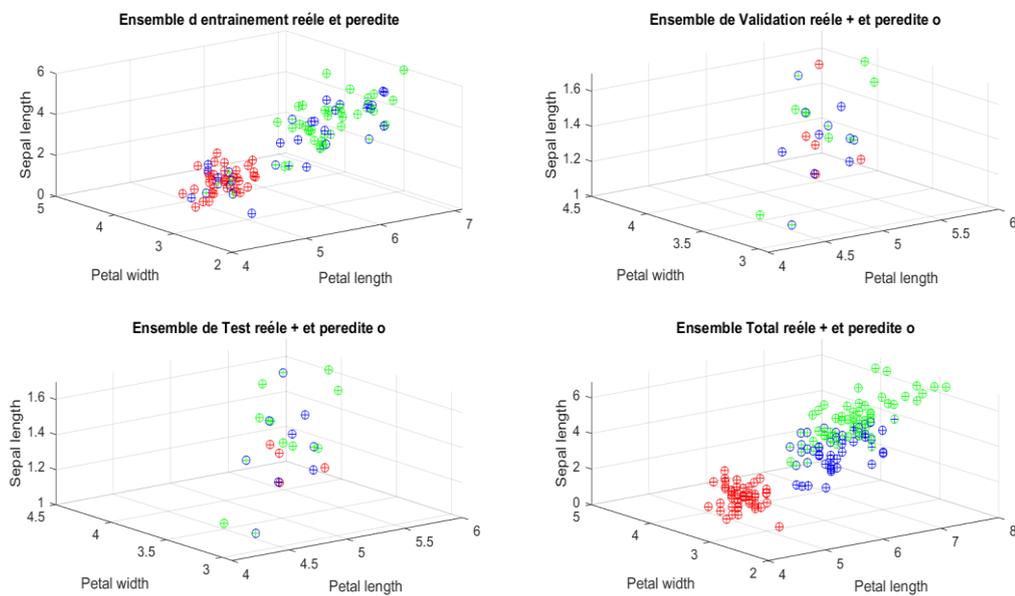


Figure 4.12 – Représentation des classe réelle et prédite en 3D de la simulation3.

4.6.2.2 Simulation 4 : dans cette simulation on va utiliser la fonction linéaire pour la couche caché et la fonction sigmoïde pour la couche de sortie.

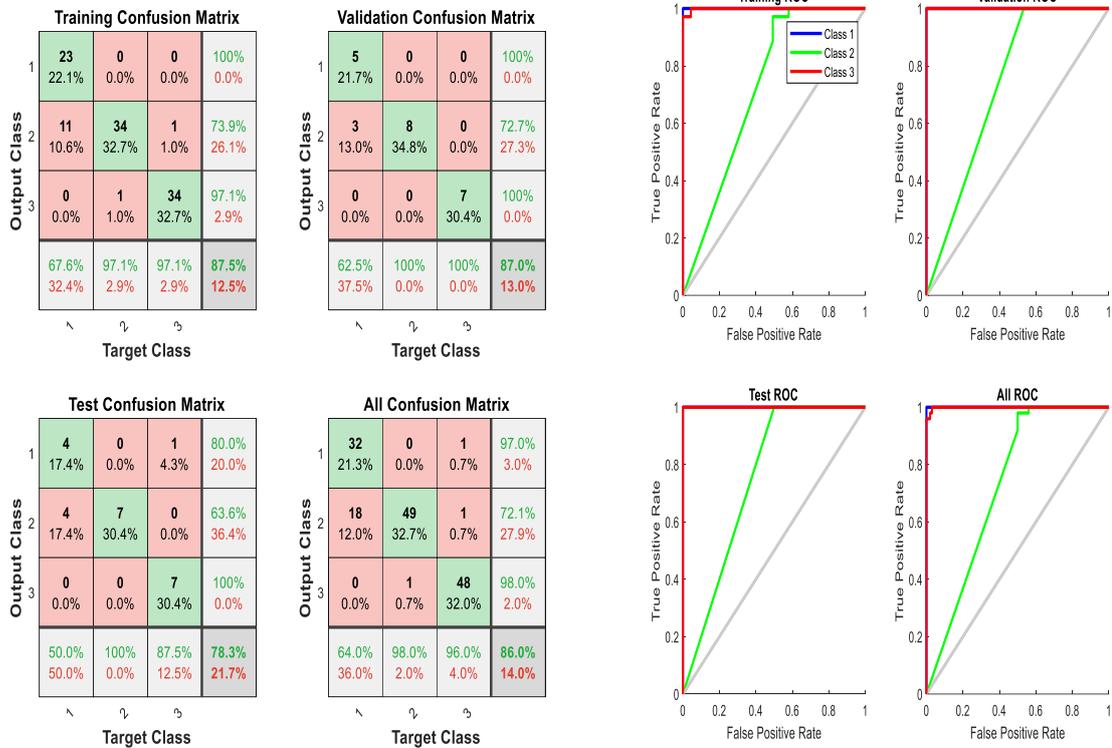


Figure 4.13- La matrice de confusion de la simulation 4. figure 4.14-La courbe ROC de la simulation 4.

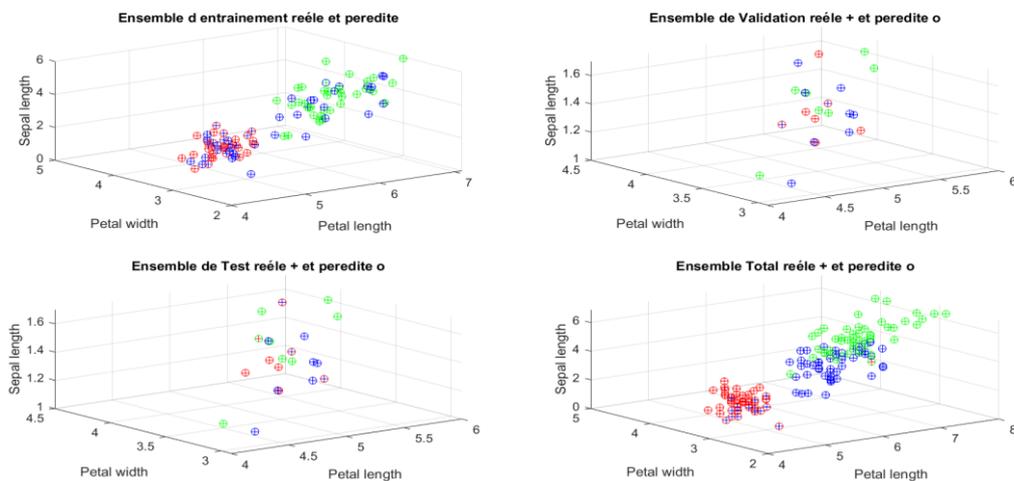


Figure 4.15 – Représentation des classes réelle et prédite en 3D de la simulation 4.

4.6.2.3 Simulation 5 : dans cette simulation on va utiliser successivement la fonction d'activation sigmoïde pour la couche caché et la fonction d'activation linéaire pour la couche de sortie.

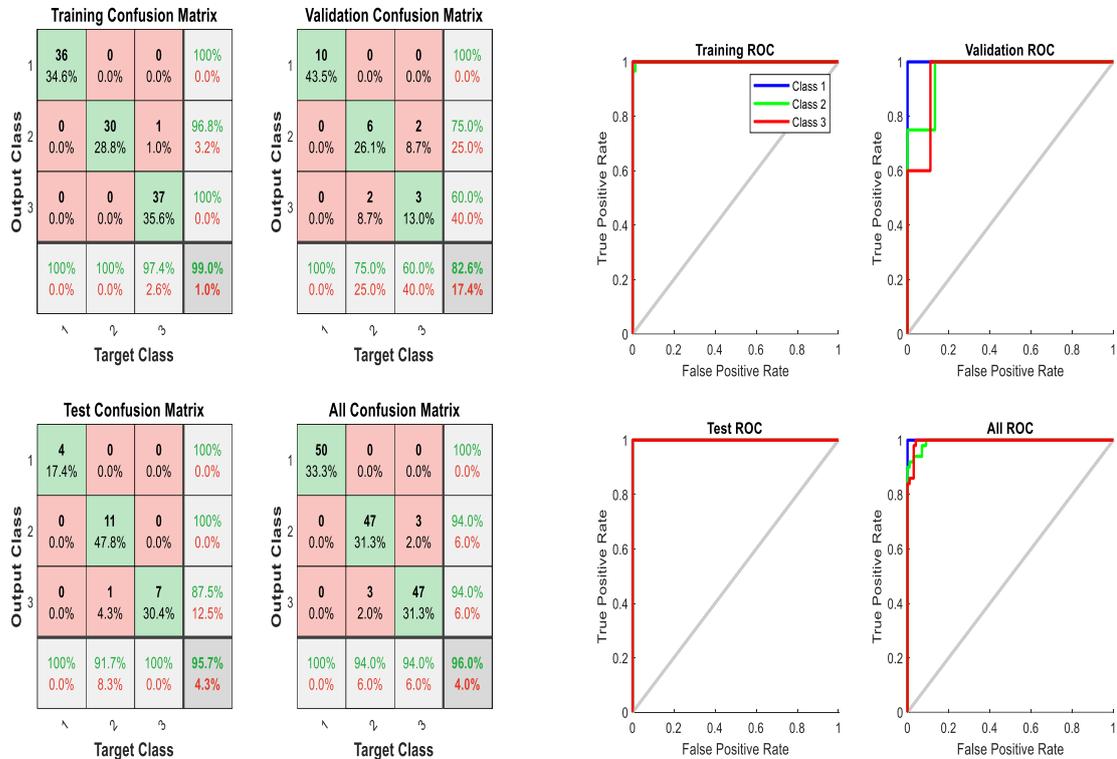


Figure 4.16- La matrice de confusion de la simulation 5. figure 4.17 –La courbe ROC de la simulation 5.

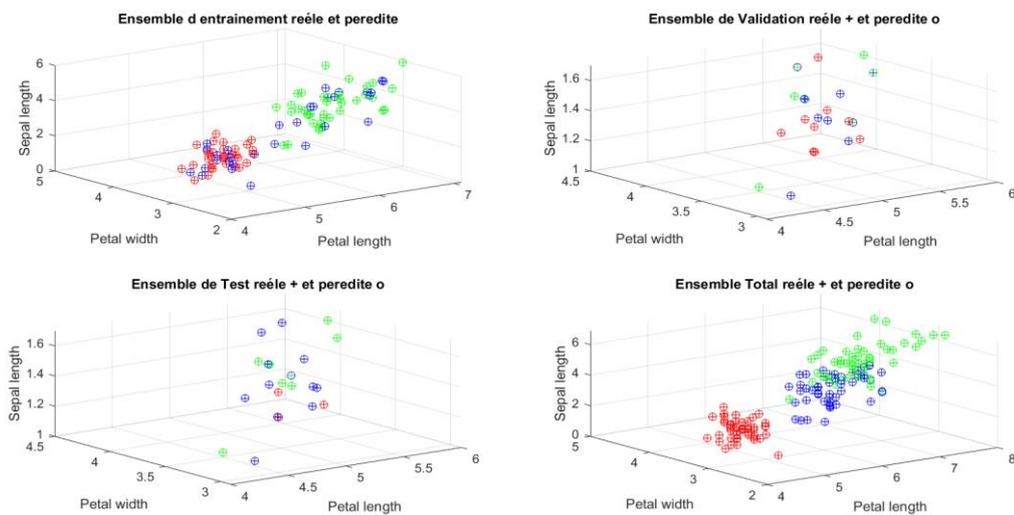


Figure 4.18 – Représentation des classe réelle et prédite en 3D de la simulation 5.

4.6.2.4 **Simulation 6** :dans cette simulation on va utiliser la fonction d'activation sigmoïde pour les deux couches.

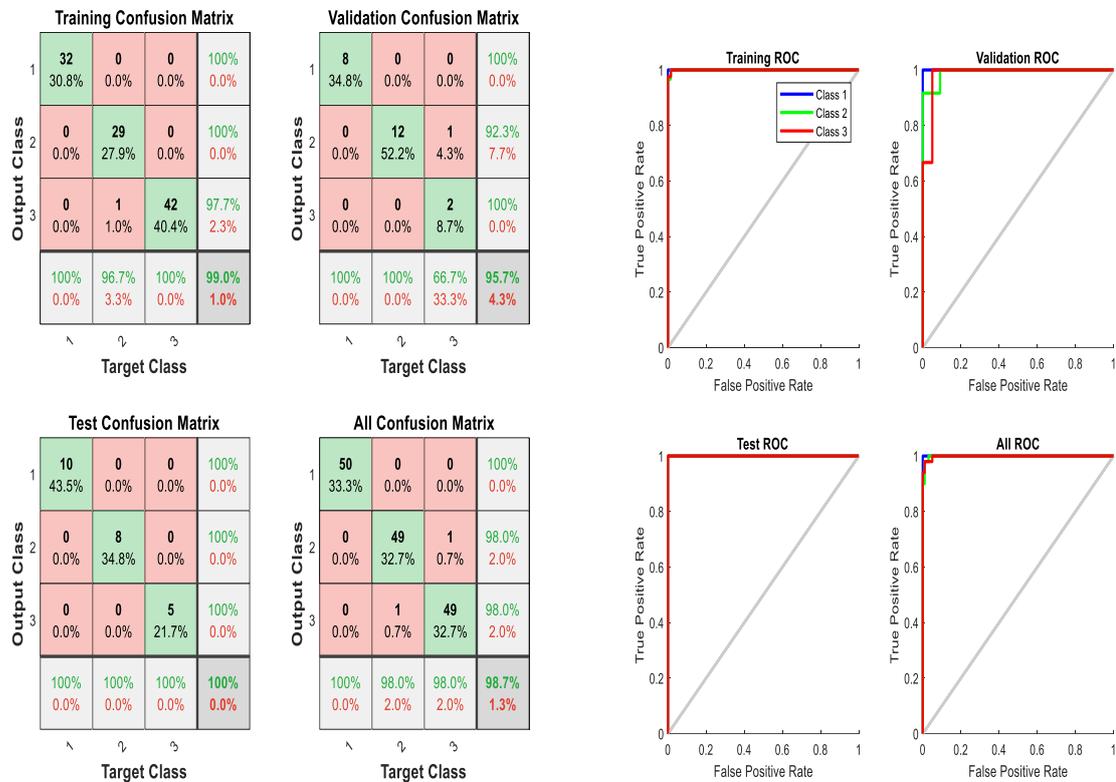


Figure 4.19-La matrice de confusion de la simulation 6 figure 4.20–La courbe ROC de la simulation 6.

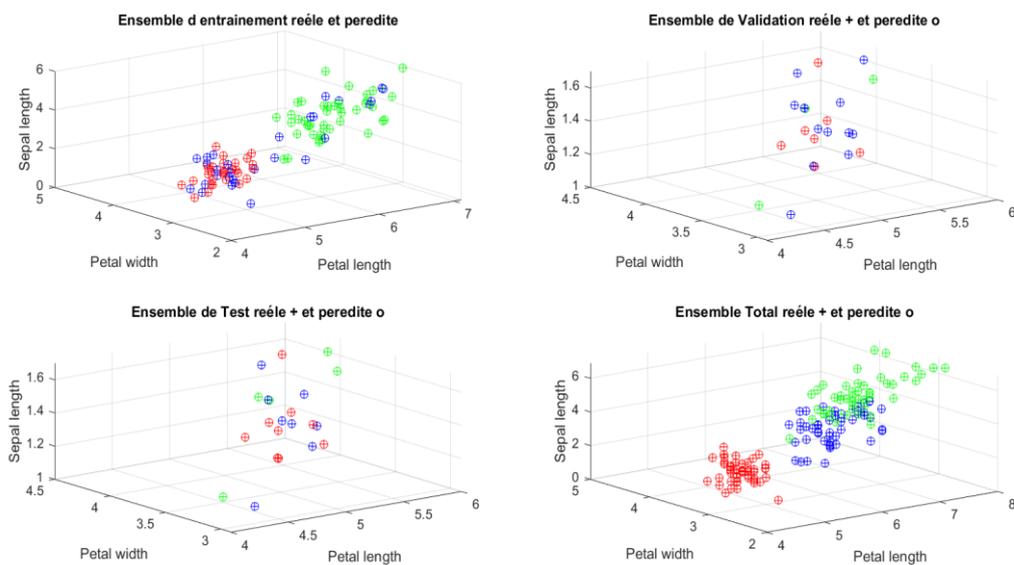


Figure 4.21 – Représentation des classe réelle et prédite en 3D de la simulation 6.

| Fonction d'activation de la couche cachée | Fonction d'activation de la couche de sortie | Précision de classification de la phase d'entraînement | Précision de classification de la phase de validation | Précision de classification de la phase de test | Précision Totale du MLP |
|---|--|--|---|---|-------------------------|
| linéaire | linéaire | 85.6% | 73.9% | 78.3% | 82.7% |
| linéaire | sigmoïde | 87.5% | 87.0% | 78.3% | 86.0% |
| sigmoïde | linéaire | 99.0% | 82.6% | 95.7% | 96% |
| sigmoïde | sigmoïde | 99% | 95.7% | 100% | 98% |

Table 4. 3 - Tableau des résultats de la deuxième proposition

4.6.2.5 Discussion et comparaison des résultats :

D'après le tableau on remarque que l'utilisation de la fonction sigmoïde dans au moins l'une des couches (cachée ou sortie) permettait d'améliorer la précision par rapport au choix linéaire pour les deux couches, et que la meilleure précision était obtenue pour le choix de la fonction sigmoïde pour les deux couches, ce qui montre l'efficacité de cette fonction d'activation dans ce genre de problème de classification.

4.7 Conclusion :

Dans ce chapitre nous avons présenté une approche de classification des fleurs iris par le perceptron multicouche, pour cela on a proposé deux choix, sur la base d'une étude comparative, afin d'augmenter la performance de notre perceptron multicouche en termes de précision.

Notre première proposition c'est portée sur le choix de la fonction d'entraînement de Levenberg marquardt qui montre de meilleure performance par rapport à la fonction du gradient conjugué en termes de précision, et pour la deuxième proposition, le choix de la fonction d'activation sigmoïde pour la couche cachée et la couche de sortie montre une meilleure précision par rapport aux autres choix de combinaison de fonction utilisés.

Conclusion générale

Dans ce mémoire nous avons eu l'occasion d'entamer le domaine de la classification par le biais de l'une des techniques occupant une grande partie de l'intelligence artificielle qui utilise les réseaux de neurone, et cela à travers l'application d'un perceptron multicouche pour la classification des fleurs iris.

- Notre tâche la plus délicate a été le choix de la combinaison adéquate des paramètres qui permettait d'optimiser le rendement de notre PMC c-à-d minimiser l'erreur tout en augmentant la précision

- Pour ce faire on a fait une première proposition de choix de la fonction d'apprentissage et une deuxième consistant à choisir la fonction d'activation sur la base d'une étude comparative. Cela nous a permis d'atteindre une précision globale de classification de 98% et une précision de classification pour l'ensemble de test de 100 % ce qui montre que le choix de nos paramètres et structure de réseaux de neurone a bien porté ses fruits.

- Pour finir, ce travail nous a permis de mettre en pratique nos connaissances sur les réseaux de neurone, et particulièrement le perceptron multicouche, d'autre part il nous a offert l'opportunité de se familiariser avec les différents outils du logiciel matlab.

- Ce travail n'est qu'une simple application sur une petite base de données de fleurs dans un vaste monde de classification et de réseaux de neurones, mais on peut toujours envisager d'aller plus loin dans ce domaine de recherche, telles que :

- 1) Utilisation des autres architectures des réseaux de neurones pour la classification comme CNN et RNN.
- 2) Cibler le domaine industriel pour l'implantation de ces programmes de classification par exemple les chaînes de production.

- Enfin, nous souhaitons que notre travail soit un apport pour les futurs étudiants qui cherchent à travailler dans ce domaine.

Bibliographie

- [1] Tom MITCHELL, Machine learning, McGraw-Hill Science/Engineering/Math, 1997
- [2] Ali EL AKADI. Contribution à la sélection de variables pertinentes en classification supervisée : application à la sélection des gènes pour les puces à ADN et des caractéristiques faciales. DOCTORAT en Informatique et Télécommunications, UNIVERSITÉ MOHAMMED V – AGDAL FACULTÉ DES SCIENCES Rabat, mars 2012.
- [3] MEMOIRE MASTER UNIVERSITE ABOU BAKR BELKAID (Information Non Disponible)
- [4] These, Classification et apprentissage actif à partir d'un flux de données évolutif en présence d'étiquetage incertain. Mohamed-Rafik Bouguelia. Université de Lorraine, janvier 2016.
- [5] Thèse. Une méthode de classification non supervisée pour l'apprentissage de règles et la recherche d'information. Guillaume Cléziou. Université d'Orléans 2004.
- [6] Faire attention aux faux amis français / anglais : discrimination / classification (supervisée) et classification / clustering (non-supervisée)]
- [7] [<http://www-ia.lip6.fr/~tollaris/ARTICLES/THESE/node7.html>]
- [8] MEMOIRE. SÉLECTION DES MOTS CLÉS BASÉE SUR LA CLASSIFICATION ET L'EXTRACTION DES RÈGLES D'ASSOCIATION. Par LABIAD ALI juin 2017
- [9] Christophe Chesneau cours Éléments de classification du Master1 MIASHS de l'université de Caen.
- [10] Classification supervisée : des algorithmes et leur calibration automatique .Sylvain Arlot. Cours de 3ème année École Centrale de Paris mars 2009.
- [11] Akbani, R. Kwak, S. and Japkowicz, N (2004). Applying support vector Machine to imbalanced Datasets. Page 35-50
- [12] Mémoire Approche Exploratoire Sur La Classification Appliquée Aux Images. Lamri Laouamer Université Du Québec, Avril 2006
- [13] Christophe Chesneau Cours Éléments De Classification Du Master1 MIASHS De L'université De Caen.
- [14] <https://www.lebigdata.fr/confusion-matrix-definition> (visité le 17 /09/2020)

- [15] <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=fr> (visité le 24/09/2020)
- [16] Hopfield, J. J. Proc. natn. Acad. Sci. U.S.A. 79, 2554–2558 (1982).
- [17] Fabien Tschirhart, « RESEAUX DE NEURONES FORMELS APPLIQUES A L'INTELLIGENCE ARTIFICIELLE ET AU JEU », mémoire de recherche, sous la direction de M. Alain Lioret, ECOLE SUPERIEURE DE GENIE INFORMATIQUE, paris, 2009, p.10
- [18] https://fr.wikipedia.org/wiki/Neurone_formel (visité au 10/04/2020)
- [19] <http://www.statsoft.fr/concepts-statistiques/reseaux-de-neurones-automatisees/reseaux-de-neurones-automatisees.htm#fonctions> (visité au 18/08/2020)
- [20] Adrien Verger (2009), Réseaux de neurones artificiels
- [21] <https://dataanalyticspost.com/Lexique/reseaux-de-neurones-recurrents/> (visité au 15/08/2020)
- [22] https://en.wikipedia.org/wiki/Long_short-term_memory (visité au 16/08/2020)
- [23] IkramChraibiKaadoud, «Apprentissage de séquences et extraction de règles de réseaux récurrents : application au traçage de schémas techniques », thèse de doctorat en informatique, sous la direction de Frédéric Alexandre, Université de Bordeaux, 2018, p. 71
- [24] Gers, Felix, Schraudolph, Nicol et Schmidhuber, Jürgen, « Learning Precise Timing with LSTM Recurrent Networks », Journal of Machine Learning Research, vol. 3, 2002, p. 115-143.
- [25] Van HiepPhung et and EunJoo Rhee « A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets », MDPI journal, 2019, 9, 4500; doi:10.3390/app9214500.
- [16] Hopfield, J. J. Proc. natn. Acad. Sci. U.S.A. 79, 2554–2558 (1982).
- [17] Fabien Tschirhart, « RESEAUX DE NEURONES FORMELS APPLIQUES A L'INTELLIGENCE ARTIFICIELLE ET AU JEU », mémoire de recherche, sous la direction de M. Alain Lioret, ECOLE SUPERIEURE DE GENIE INFORMATIQUE, paris, 2009, p.10
- [18] https://fr.wikipedia.org/wiki/Neurone_formel (visité au 10/04/2020)
- [19] <http://www.statsoft.fr/concepts-statistiques/reseaux-de-neurones-automatisees/reseaux-de-neurones-automatisees.htm#fonctions> (visité au 18/08/2020)
- [20] Adrien Verger (2009), Réseaux de neurones artificiels
- [21] <https://dataanalyticspost.com/Lexique/reseaux-de-neurones-recurrents/> (visité au 15/08/2020)
- [22] https://en.wikipedia.org/wiki/Long_short-term_memory (visité au 16/08/2020)
- [23] IkramChraibiKaadoud, «Apprentissage de séquences et extraction de règles de réseaux récurrents : application au traçage de schémas techniques », thèse de doctorat en informatique, sous la direction de Frédéric Alexandre, Université de Bordeaux, 2018, p. 71

- [24] Gers, Felix, Schraudolph, Nicol et Schmidhuber, Jürgen, « Learning Precise Timing with LSTM Recurrent Networks », *Journal of Machine Learning Research*, vol. 3, 2002, p. 115-143.
- [25] Van HiepPhung et and EunJoo Rhee « A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets», *MDPI journal*, 2019, 9, 4500; doi:10.3390/app9214500.
- [26] Modélisation par Réseaux de Neurones Artificiels (RNA) et commande Prédictive non linéaire d'une station de production d'eau froide.
- [27] G. Dreyfus, *Les Réseaux de Neurones*, Mécanique Industrielle et Matériaux, No.51, septembre 1998.
- [28] *Memoire Modélisation Avec Les Réseaux De Neurones D'un Canal U\B Dans Un Environnement Minier Souterrain. Mina Kalakh Mars 2013. Université Du Québec En Abitibi-Témiscamingue.*
- [29] *These Diagnostic des defaillances d'une turbine à gaz a base des reseaux de neurones artificiels pour l'amelioration de leur systeme de detection des vibrations. Mohamed BEN RAHMOUNE. Universite Ziane Achour de Djelfa .2017.*
- [30] <https://deeplearning.fr/cours-theoriques-machine-learning/decoupage-des-datasets/>
- [31] D.Pham, S. Sagiroglu, "Training multilayered perceptrons for pattern recognition: a comparative study of four training algorithms", *International Journal of Machine Tools and Manufacture*, vol.41, 2001, pp. 419–430.
- [32] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning", *Neural Networks*, 6, 1993, pp. 525–533.
- [33] https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788393485/16/ch16lvl1sec76/unsupervised-learning
- [34] <https://www.guru99.com/unsupervised-machine-learning.html>
- [35] deeplearning.fr/cours-theoriques-deep-learning/fonctionnement-du-neurone-artificiel/
- [36] https://www.researchgate.net/figure/Un-perceptron-multi-couches-contenant-3-couches-couche-dentree-de-5-cellules-une_fig3_312328378
- [37] <https://studylibfr.com/doc/2027238/laboratoire-4-%E2%80%93-classification--1-2--%E2%80%93-rbf> (visité au 10/08/2020)
- [38] <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (visité au 16/08/2020)
- [39] <https://medium.com/@akashkhurana55/iris-dataset-classification-f05fb601b5d6>
- [40] https://fr.wikipedia.org/wiki/Iris_de_Fisher
- [41] https://www.researchgate.net/figure/Apprentissage-des-reseaux-de-neurone-par-lalgorithme-de-retropropagation_fig34_324929383