République Algérienne Démocratique 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

Domaine :*Sciences et Technologie*
Filière : *Automatique*
Spécialité : *Automatique*

Présenté par :
MEZAACHE Ahmed

Classification Des Données Basée Sur Les Réseaux De Neurones Récurrents (RNN)

Soutenue publiquement le 23/06/2024, devant les jurys composés de :

Mr. Debeche Mehdi	MAA Université de Guelma Encadreur
Pr. Babouri Abdesselam	Professeur Université de Guelma Examinateur principal
Dr. Boucerredj Leila	MCA Université de Guelma Examinateur

Année Universitaire : 2023/2024

Remerciement :

Je remercie Dieu Tout-Puissant de nous avoir donné la santé et le bien-être. Désir de commencer et de terminer cette lettre.
Tout d'abord, je tenais à remercier notre encadreur **Mr.Debeche mehdi** pour son Aide et encadrement, je le remercions aussi pour la qualité exceptionnelle de son encadrement Sa patience, sa précision et sa disponibilité lors de la préparation de mon mémoire.
Je tiens aussi à le remercier pour son aide pratique, son soutien moral et son encouragement.
Je remercie également tous nos enseignants pour leur générosité et leur grande patience Ils ont pu démontrer malgré leurs charges académiques professionnels.

Résumé :

Dans ce mémoire, nous explorons l'efficacité des réseaux de neurones récurrents (RNN) dans le cadre de la classification de données. Nous examinons comment les RNN peuvent être utilisés pour capturer les motifs complexes présents dans les séquences de données. En particulier, nous nous intéressons à leur application dans la classification des données des chiffres manuscrits, où la structure séquentielle des données peut jouer un rôle crucial dans la prise de décision.

Summary:

In this dissertation, we explore the effectiveness of recurrent neural networks (RNNs) in data classification. We examine how RNNs can be used to capture complex patterns present in data sequences. In particular, we focus on their application in the classification of handwritten cipher data, where the sequential structure of the data can play a crucial role in decision-making.

ملخص

في هذه الأطروحة، نستكشف فعالية الشبكات العصبية المتكررة (**RNNs**) في تصنيف البيانات. ندرس كيف يمكن استخدام الشبكات العصبية المتكررة لالتقاط الأنماط المعقدة الموجودة في تسلسل البيانات. وعلى وجه الخصوص، نحن مهتمون بتطبيقها في تصنيف البيانات المشفرة المكتوبة بخط اليد، حيث يمكن أن تلعب البنية المتسلسلة للبيانات دورًا حاسمًا في اتخاذ القرار.

Sommaire

Chapitre III

Les réseaux de neurones

Chapitre IV

Implémentation

Liste des figures

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Liste des tableaux

Chapitre II

Chapitre III

Chapitre IV

Liste des acronymes utilises dans le mémoire:

RNN : **R**éseau de **N**eurones **R**écurrent
LSTM : **L**ong **S**hort-**T**erm**M**emory
MPC : **M**ultilayer **P**erceptron
CNN : **C**onvolutional **N**eural **N**etwork
K-NN : **K**- **N**earest **N**eighbors.
SVM : **S**upport **V**ector**M**achine
RN : **R**éseaux de **N**eurones
AI : **A**rtificielle **I**ntelligence
ML : **M**achine **l**earning
GPU : **G**raphics **P**rocessing**U**nit
ADAM : **A**daptative **M**oment **E**stimation

Introduction générale

La classification est une tâche fondamentale dans le domaine de l'apprentissage automatique, visant à attribuer des étiquettes ou des catégories à des données en fonction de certaines caractéristiques ou des attributs. Elle trouve une application dans de nombreux domaines, notamment la reconnaissance de la parole, la détection d'objets, la traduction automatique, la classification d'image et bien plus encore. L'évolution de l'apprentissage automatique et des techniques de classification a considérablement amélioré la capacité des systèmes informatiques à traiter et à comprendre des données complexes.

Dans de nombreux cas, les données peuvent être séquentielles, c'est-à-dire qu'elles sont organisées dans une séquence temporelle ou spatiale. Les approches traditionnelles de classification peuvent être inefficaces pour traiter ces données, car elles ne prennent pas en compte la structure séquentielle. Les réseaux de neurones récurrents (RNN) ont émergé comme une solution puissante pour la classification de données séquentielles en exploitant la dépendance temporelle ou spatiale des données.

Bien que les RNN ne soient pas habituellement utilisés pour la classification, ils sont largement appliqués dans ce domaine en raison de leur capacité à modéliser des séquences de données et à capturer les dépendances à long terme dans ces séquences. Cette capacité en fait des modèles puissants pour la classification de données séquentielles telles que la reconnaissance de la parole, la traduction automatique, la génération de texte, et bien d'autres. Dans le contexte de la classification des chiffres manuscrits, les RNN peuvent être utilisés pour traiter des séquences d'images représentant des chiffres écrits à la main, en capturant les relations spatiales entre les pixels dans chaque image ainsi que les relations temporelles entre les images dans la séquence. Cette approche permet aux RNN de découvrir des motifs complexes et des caractéristiques discriminantes dans les données, conduisant à une classification précise des chiffres manuscrits. En résumé, bien que les RNN ne soient pas exclusivement dédiés à la classification, leur capacité à modéliser des données séquentielles en fait des outils puissants pour cette tâche, y compris dans le domaine de la classification des chiffres manuscrits.

Présentation de la Structure du Mémoire :

Notre mémoire se compose de quatre chapitres principaux, suivant une introduction générale sur la tâche de classification et le type particulier d'algorithme de classification représenté par les réseaux de neurones récurrents.

Le premier chapitre est consacré aux généralités sur les bases de données, en mettant l'accent sur le type de base de données utilisé spécifiquement pour la tâche de classification.

Dans le deuxième chapitre, nous abordons le principe et les différents types de classification, en fournissant des exemples d'algorithmes pour chaque type.

Le troisième chapitre traite du principe général des réseaux de neurones artificiels, de leurs architectures, avec une attention particulière portée aux réseaux de neurones récurrents.

Enfin, le quatrième chapitre propose une implémentation comparative de la performance de trois architectures de réseaux de neurones récurrents : LSTM, BLSTM et GRU, appliquées à la classification de la base de données de chiffres manuscrits.

Nous conclurons par une synthèse générale de l'ensemble des chapitres précédemment abordés

Chapitre I

Généralités Sur Les Bases De Données

1 Généralités Sur Les Bases De Données:

1.1 Introduction :

Les bases de données sont des outils essentiels en informatique pour stocker, organiser et gérer des données de manière structurée. Elles permettent de stocker des informations de manière efficace, d'accéder rapidement aux données, de les mettre à jour et de les interroger. Les bases de données sont utilisées dans de nombreux domaines tels que les applications web, les systèmes d'information, la gestion des entreprises, et bien d'autres. Elles offrent une solution centralisée pour gérer des volumes importants de données de manière cohérente et sécurisée. En résumé, les bases de données sont des éléments fondamentaux de l'informatique moderne, facilitant le stockage et la manipulation efficace des données pour répondre aux besoins des utilisateurs et des applications.

Dans ce chapitre nous présenterons les généralités des bases données, cette
présentation nous permettra d'avoir une idée sur la définition de base des données,ainsi que les critères d'une base de données , son rôle, ses types,L' importance de base de donnée et les systèmes de gestion de bases de données, pour l'exploiter dans les chapitres qui suits.

1.2 Définition d'une base de données:

Une base de données est une collection organisée d'informations structurées, généralement stockées électroniquement dans un système informatique.

Dans les opérations aujourd'hui, les données que contiennent les bases de données les plus courantes sont généralement modelées en lignes et en colonnes, dans une série de tables, pour assurer l'efficacité du traitement et de l'interrogation des données. Les données peuvent être facilement consultées, gérées, modifiées, mises à jour, contrôlées et organisées. La plupart des bases de données utilisent le langage SQL pour l'écriture et l'interrogation des données.

De plus, la base de données est une organisation cohérente (c'est-à-dire qu'elle satisfait à tous les Contraintes d'intégrité) sur les données persistantes (en supposant la gestion de la mémoire auxiliaire et des caches) sont accessibles aux utilisateurs simultanés (gestion des transactions),

Contraintes d'intégrité) sur les données persistantes (en supposant la gestion de la mémoire auxiliaire et des caches) sont accessibles aux utilisateurs simultanés (gestion des transactions),

fournissant Indépendance physique (entre application et description des données)
et optimisation automatique (le rôle de l'optimiseur) écrite dans une requête de

requête Langage déclaratif de haut niveau (SQL)[1].

1.3 critère d' une base de données:

La base de données doit remplir les conditions suivantes:

exhaustivité : tous les enseignements pertinents pour l'application en question sont présents dans cette base de données

Non-redondance des données : Non-duplication ou duplication multiple des données .

LA Structure : C'est l'adaptation de la façon dont les données sont stockées pour le traitement.

La structure que devrait avoir une base est liée au développement de la technologie[2].

1.4 Types d'objets de base de données:

Il existe quatre types différents d'objets de base de données qui aident les utilisateurs à compiler, saisir, stocker et analyser des données dans divers formats :

1. Tables
2. Requêtes
3. Formulaires
4. Rapports

1.5 Rôle d'une Base de données:

Contrairement aux approches classiques, la création d'une base de données qui soitpartagée par plusieurs utilisateurs est le reflet d'une évolution dans la gestion de l'entreprise.

Son rôle est de rendre possible :

- La centralisation de l'information : l'information n'est plus éparpillée dans
Fichiers à différents endroits.
- L'intégration (tout ce qui se fait dans un service est visible par d'autres services)
- La diffusion de l'information archivée (si l'information est disponible à un seul endroit,elle est facile à diffuser).

Ceci a pour avantages :

- D'améliorer la cohérence de l'information (une seule valeur pour une même information)
- De réduire les redondances (une même information n'est stockée si possible qu'une seule fois).

- De réduire les efforts de saisie et de mise à jour des informations (i.e. Une information qui doit être stockée une seule fois ne sera saisie qu'une seule fois. De même que sa mise à jour ne se fera qu'une seule fois)[2].

1.6 l'importance des bases de données :

Les bases de données sont utilisées pour stocker de grandes quantités de données collectées de manière organisée et les rendre facilement accessibles aux utilisateurs autorisés.

Chacune des entreprises utilise des bases de données différentes selon la nature de ses données.

Les bases de données sont importantes pour la croissance des entreprises à bien des égards:

- Permet à une entreprise de prendre des décisions commerciales éclairées.
- Donnez aux différents services commerciaux un accès instantané aux commerciales critiques qu'ils peuvent utiliser pour comprendre les modèles de données, créer des rapports et prédire les tendances futures.
- Fournit des applications opportunes et personnalisées basées sur les données et des analyses détaillées.
- Souvent les données sont mappées des bases de données hiérarchiquesutilisées par les systèmes hérités aux bases de données relationnelles utilisées dans les entrepôts de données.Aide à analyser et à aggraver les données commerciales.
- Collectez et stockez des données clients cruciaux à partir de différentes applications.
- Aide à analyser et à aggraver les données commerciales[1].

1.7 Système de Gestion de Base de Données :

La gestion et l'accès à la base de données sont assurés par un ensemble de
procédures Constitution d'un système de gestion de base de données (SGBD). Ce dernier est un logiciel qui permet de manipuler les données BBD. Les opérations, c'est-à-dire la sélection et l'affichage des informations obtenues à partir de cette base de données, la modification, l'ajout ou la suppression de données, invoquent le concept de transition.

Lors du traitement des transactions, le SGBD classique fonctionne selon des contraintes dites ACID, ce qui signifie « atomique », « cohérence », « isolation » et « durabilité ». Si la transaction satisfait à ces quatre contraintes, la transaction s'exécutera avec succès.

Ainsi, la notion de base de données est souvent associée à la notion de réseau informatique.[2].

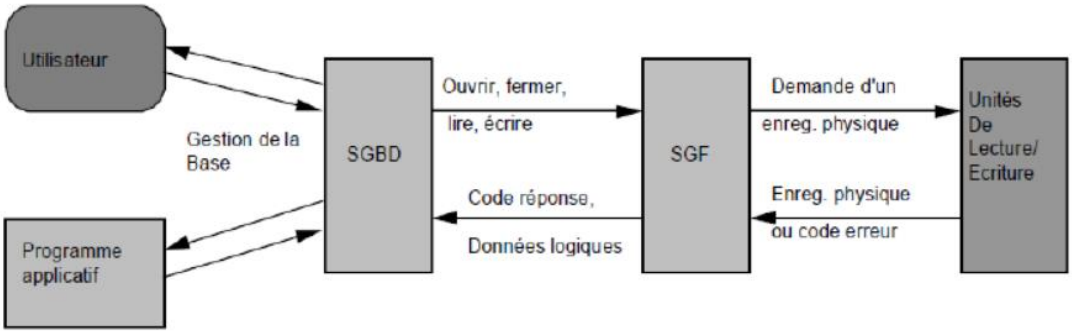


Figure 1.1 : SGBD système de gestion de bases de données

1.8 Types de bases de données:

Il existe de nombreux types de bases de données.La meilleure base de données pour une organisation particulière dépend de la manière dont l'organisation souhaite utiliser les données.

1. **Bases de données relationnelles**:Une base de données relationnelle, en anglais:Relational Data Base Management System RDBMS est une base de données où l'information est organisée dans des tableaux à deux dimensions appelés des relations ou tables, selon le modèle introduit par Edgar F. Codd en 1960. Selon ce modèle relationnel, une base de données consiste en une ou plusieurs relations. Les lignes de ces relations sont appelées des enregistrements. Les colonnes sont appelées des attributs.
2. **Bases de données orientées documents** : Les bases de données orientées documents sont flexibles et permettent de manipuler des données semi-structurées et non-structurées, mais sacrifient la conformité ACID pour la flexibilité.
3. **Bases de données en colonnes** : Les bases de données en colonnes sont plus adaptées à la gestion des données structurées et semi-structurées que les autres systèmes non-relationnels.
4. **Moteurs de recherche** : Les moteurs de recherche, comme Elasticsearch, permettent de stocker et de chercher des données de manière efficace, souvent utilisés pour les applications de recherche et de filtrage.
5. **Bases de données clé-valeur** : Les bases de données clé-valeur stockent des données sous forme de paires clé-valeur, permettant une manipulation simple et rapide des données.

6. **Bases de données distribuées** : Les bases de données distribuées permettent de stocker et de gérer des données sur plusieurs serveurs, offrant une grande scalabilité et une haute disponibilité.
7. **Entrepôts de données** : Les entrepôts de données stockent des données de manière centralisée, permettant une gestion efficace et une sécurité renforcée.
8. **Bases de données orientées objet** : Les bases de données orientées objet stockent des objets et leurs relations, permettant une manipulation plus facile des données complexes.
9. **Bases de données hiérarchiques** : Les bases de données hiérarchiques sont des systèmes de gestion de bases de données qui stockent des informations groupées dans des enregistrements, chaque enregistrement comporte des champs, et les enregistrements sont reliés entre eux de manière hiérarchique.
10. **Bases de données réseau** : Les bases de données réseau sont semblables au modèle hiérarchique, mais permettent des relations n-n (plusieurs parents / plusieurs enfants).[3]

1.8.1 Type de base de données utilisées dans la tâche de classification :

Dans la plupart des cas, les bases de données utilisées dans les tâches de classification sont des bases de données relationnelles. Les bases de données relationnelles sont le type le plus courant de système de gestion de base de données (SGBD) utilisé dans de nombreuses applications, y compris celles liées à l'apprentissage automatique et à la classification.

Dans une base de données relationnelle :

1. Les données sont organisées en tables.
2. Chaque table est composée de lignes et de colonnes.
3. Chaque ligne représente un enregistrement unique.
4. Chaque colonne représente un attribut ou un champ spécifique des données.
5. Les tables peuvent être liées les unes aux autres par des clés primaires et des clés étrangères pour établir des relations entre les enregistrements.

Ces caractéristiques font des bases de données relationnelles un choix approprié pour stocker des ensembles de données utilisés dans les tâches de classification. Par exemple, dans une tâche de classification d'images médicales, une base de données relationnelle peut contenir une table où chaque ligne représente une image médicale et chaque colonne représente des caractéristiques de l'image telles que les dimensions, la résolution, les attributs extraits, et une autre table qui lie chaque image à sa catégorie ou à son étiquette de classe.

Ainsi, les bases de données relationnelles fournissent une structure organisée et efficace pour stocker et manipuler les données nécessaires aux tâches de classification, ce qui les rend largement utilisées dans ce contexte.[1]

1.9 CONCLUSION:

Dans ce chapitre, nous avons exploré les fondements des bases de données, Nous avons vu que les bases de données constituent des outils essentiels pour stocker, organiser et manipuler efficacement de vastes quantités de données. Leur rôle dans de nombreux domaines, de la gestion d'entreprise à la recherche scientifique, est indéniable. Le choix du type de base de données et du système de gestion adapté dépend des besoins spécifiques de chaque projet.

Dans le deuxième chapitre, nous aborderons le sujet de la classification de ses types, et des critères de performances d'un classifieur.

Chapitre II
Classification

2 Généralités sur la classification :

2.1. INTRODUCTION:

« 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 couramment utilisées en apprentissageautomatique,possède un large éventail d'applications, notamment la reconnaissance de formes, la détection Spam, classification des clients, diagnostic médical et classification des images.

Dans les domaines de l'apprentissage automatique et des statistiques, la classification fait référence à Le problème consiste à attribuer de nouvelles observations à l'une des nombreuses classes ou sous-populations prédéfinies.

Cette décision est basée sur l'ensemble de données suivant formation, les observations sont connues pour appartenir à des catégories spécifiques.

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

2.2 DÉFINITION :

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 semblables 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 classe. 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 [3].

En mathématiques, la classification est la classification des objets. il comprend l'allocation la catégorie de chaque objet ou individu à classer en fonction des données de formation.

Il est important de noter qu'il ne faut pas confondre entre ces deux termes : « classification » et « classement », au fait le mot classification en anglais signifie une chose, alors que le même mot en français a une autre signification (utilité).

Dans un classement on affecte les objets à des groupes préétablis, c'est le but de l'analyse discriminante que de fixer des règles pour déterminer la classe des objets. La classification

est donc, en quelque sorte, le travail préliminaire au classement, à savoir la recherche des classes "naturelles" dans le domaine étudié, en anglais « Cluster Analysis ».
Cette collision entre les termes peut se résumer comme suite :

Français	**Anglais**
Classification	Clustering
Classement	Classification

Tableau. 2.1 : les noms attribués à la classification en Français/Englais.

D'une manière général en vertu de ces définitions, 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 caractérisent un domain particulier (animaux, plantes, malades, gènes,. . . etc.), où on les regroupe en plusieurs classes, [4].

2.3 Domaines d'application des algorithmes de classification :

La classification comme dit préalablement joue un rôle dans presque toutes les sciences et techniques qui font appel à la statistique multidimensionnelle. A titre d'exemple les sciences biologiques : botanique, zoologie, écologie, ... qui utilisent le terme taxonomie pour désigner l'art de la classification. Ainsi que les sciences de la terre et des eaux : géologie, pédologie, géographie, étude des pollutions, font grand usage de classifications.
Une autre forte utilité des techniques de classification dans les sciences de l'homme psychologie, sociologie, linguistique, archéologie, histoire, etc ... et sans oublier lesdernières emploient parfois les mots de "typologie" et "segmentation" pour désigner la classification, citons encore la médecine[Jamoulle, & al, 2000], l'économie,l'agronomie . . . etc. ! Dans toutes ces disciplines la classification peut être employée
comme un domaine particulier ; mais elle l'est souvent vue comme une méthode complémentaire à d'autres méthodes statistiques. Elle est très largement utilisée à l'interprétation des graphiques d'analyse factorielle, ou bien déterminer des groupes d'objets homogènes, préalablement à une régression linéaire multiple [5].
Voilà les quelques exemples de ses utilités :

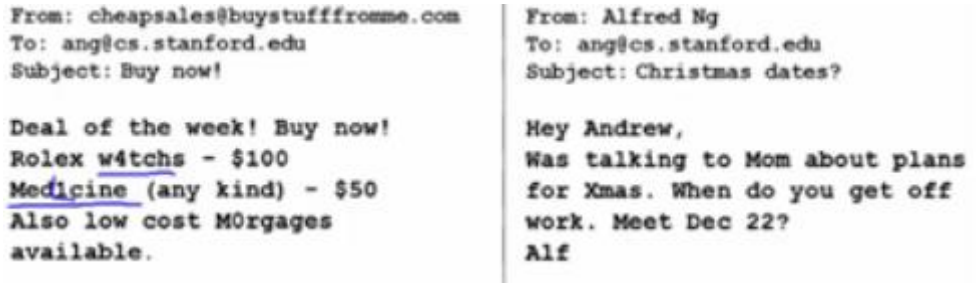
2.4 Exemples de problèmes de classification :

Ce sont les domaines que la classification pourrait viser et en même temps ils représentent

les différents types de données d'entrées des techniques de classification, ce qui est nécessaire d'en présenter avant d'aborder les méthodes classificatoires.

A/ Prédiction e-mail / Spam :

Comme le fait de différencier un E-mail valide d'un SPAM, d'ailleurs la classification est fortutile en ce qui concerne la catégorisation des documents sur internet quel que soit la nature du document (image , fichier , son . . .etc) [Michael & al, 2007] et elle peut même être utilisée pour classifier les document selon leur sens (le web sémantique et les moteurs de recherche où on associe des sens pour les termes et pour les classifier il faut développer un langage de traitement/classification sémantique par exemple à base d'ontologie.) ou tout simplement pour classifier les ouvrages dans le monde des bibliothèques et des archives(le système de classification de la Bibliothèque du Congrès LCC [5].



Spam **Non-spam**

Figure 2.1 : Représentation des emails spam et non-spam

B/ Reconnaissance de formes :

Généralement, il s'agit d'une question qui vise à reconnaître ou à identifier un modèle spécifique à partir de données brutes et à prendre une décision en fonction de la catégorie associée à ce modèle [Peter, 2001]. Ces modèles (formes) comprennent des images (visages, empreintes digitales, rayons X, EEG, etc.) et des sons (reconnaissance vocale).

Comme la reconnaissance des caractères manuscrits :

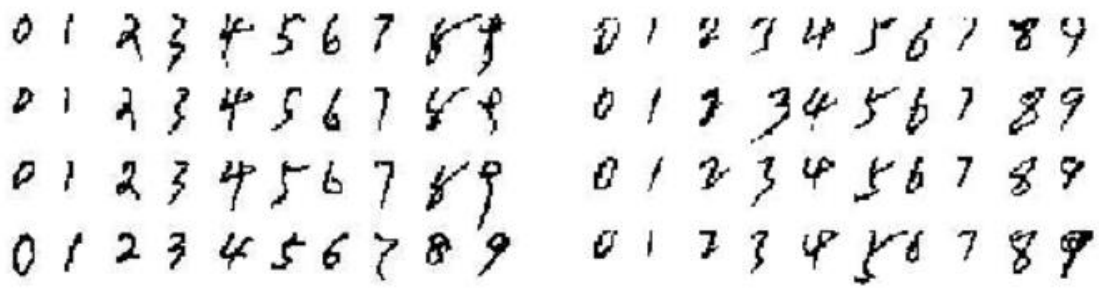


Figure2.2 : Reconnaissance des caractère manuscrits

Ou d'empreintes digitales :

Figure 2.3 : Reconnaissance d'empreintes digitales

Ou de Reconnaissance vocale :

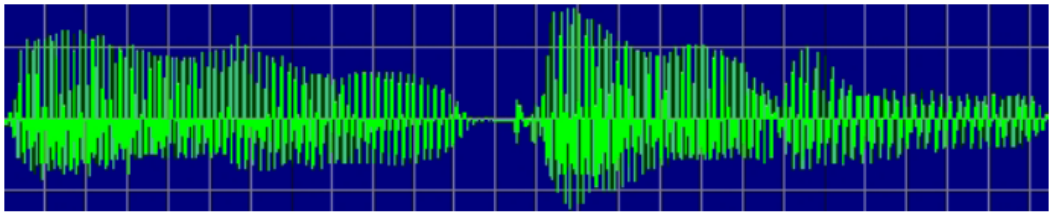


Figure2.4 : Reconnaissance vocale

C/ Des Tableaux :

comme les données bancaires : Ce type de données est représenté sous la forme d'un tableau ***N x M***. où ***N*** est le nombre d'exemples (individus, objets) et ***M*** est l'ensemble des descripteurs ou la qualité de ce qui est décrit (attributs).

Le champ (*i,j*) de cette table contient les informations relatives des éléments *j* sur *i*.

Le rôle de la classification dans ce cas est de juger le comportement d'une personne par rapport à ce qu'elle a appris des autres (base d'apprentissage)[6].

Taux de fraude
(Montant de la fraude, en millions d'euros)

Transactions nationales	2004	2005	2006	2007
Paiements	**0,036 % (81,2)**	**0,033 % (82,8)**	**0,035 % (92,3)**	**0,032 % (95,6)**
- dont paiements de proximité et sur automate	0,029 % (63,5)	0,025 % (59,2)	0,024 % (59,1)	0,017 % (45,4)
- dont paiements à distance	0,177 % (17,7)	0,196 % (23,6)	0,199 % (33,2)	0,236 % (50,1)
- dont par courrier / téléphone	nd	nd	0,194 % (19,8)	0,201 % (23,8)
- dont sur Internet	nd	nd	0,208 % (13,4)	0,281 % (26,4)
Retraits	**0,027 % (22,7)**	**0,017 % (15,0)**	**0,019 % (17,4)**	**0,020 % (19,0)**
Total	**0,033 % (103,9)**	**0,029 % (97,8)**	**0,031 % (109,6)**	**0,029 % (114,5)**

Source : Observatoire de la sécurité des cartes de paiement

Tableau2.2 :Données bancaires

2.5 Les étapes d'une classification :

1.Sélection des données à utiliser.

2.Calcul de la similarité entre les *n* individus en se basant sur les données initiales.

3.Choix d'un algorithme de classification et exécution de celui-ci.

4.Interprétation des résultats obtenus.

- ✓ Evaluation de la qualité de la classification,
- ✓ Description des classes obtenues.

2.6 Diffèrent type de classification :

2.6.1 Classification binaire:

La classification binaire consiste à catégoriser les données en deux classes ou catégories distinctes. C'est la forme de classification la plus simple, où le but est d'affecter chaque point de données à l'une des deux classes prédéfinies. Par exemple, déterminer si un e-mail est un spam ou non, classer une transaction comme frauduleuse ou légitime, ou prédire si un patient a ou non une certaine condition médicale.[7]

2.6.2 Classification multi classe :

La classification multi classe implique la catégorisation des données en plus de deux classes ou catégories. Dans ce type de classification, l'objectif est d'affecter chaque point de données à l'une de plusieurs classes prédéfinies. Par exemple, classer des images en différents types d'animaux (chat, chien, oiseau, etc.), reconnaître des chiffres manuscrits (0-9) ou identifier le genre d'une chanson (rock, pop, jazz, etc.). Les problèmes de classification multi classe peuvent être résolus à l'aide de divers algorithmes tels que les arbres de décision, la régression logistique, les machines à vecteurs de support ou les modèles d'apprentissage en profondeur comme les réseaux de neurones profond. Ces deux types de classification fournissent une base pour organiser et analyser les données en fonction de leurs catégories ou classes distinctes .[7]

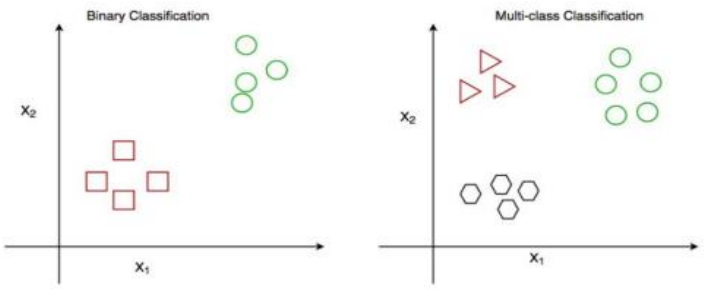


Figure 2.5 : Types de classification

2.7 MÉTHODES DE CLASSIFICATION :

Les méthodes de classification analysent les entrées (données : images, documents, état des patients, etc.) pour en dériver les sorties (classes, catégories, diagnostics, etc...) Autrement dit, nous caractérisons ces différentes données en les affectant à des classes. L'apprentissage automatique se propose de construire automatiquement une telle procédure. En effet, deux types de classification peuvent être distingués .

1. Supervisé (classement) : Est une méthode basée sur le concept d'apprentissage supervisé ou Observez le résultat selon lequel les catégories sont connues et prédéfinies.

2. Non supervisé (classification) :est basé sur le concept d'apprentissage non supervisé, où le résultat est non observé, classe inconnue.

Il existe cependant d'autres types de classifications basées sur d'autres types de méthodes d'apprentissage, comme l'apprentissage semi-supervisé et l'apprentissage par renforcement. En fait, il existe deux types d'apprentissage : « supervisé » et « non supervisé » car l'apprentissage semi-supervisé peut traiter de grandes quantités de données sans avoir à tout étiqueter et offre les avantages susmentionnés. L'apprentissage par renforcement est largement utilisé dans les exemples d'apprentissage interactif.

2.8 Approche Paramétrique versus non-paramétrique :

1- Non paramétrique :

Les approches dites non paramétriques (classification hiérarchique, méthode des centres mobiles) basée sur l'hypothèse : plus deux individus sont proches, plus ils ont de chances de faire partie de la même classe, en plus ce que distingue cette approche est qu'on ne fait aucune hypothèses sur le modèle que suivent les données, C'est le cas des plus proches voisins (K-PPV), donc il suffit de trouver les propriétés de convergence quand le nombre de données est grand.

2- Paramétrique :« Probabilistes »

La seconde grande famille des méthodes de classification, ce sont les approches probabilistes, utilisent une hypothèse sur la distribution des individus à classifier ,c'est-à-dire, on suppose que l'on connaît la forme du modèle qui a généré les données . Par exemple, on peut considérer que les individus de chacune des classes suivent une loi normale. Le problème qui se pose, est de savoir déterminer ou estimer les paramètres des lois (moyenne, variance) et à quelle classe les individus ont le plus de chances d'appartenir à partir de l'ensemble d'apprentissage.

Les paramètres d'une loi peuvent être déterminés de maintes façons, C'est le cas par exemple des classifications bayésiennes ou encore l'algorithme espérance-maximisation.[7]

2.9 La classification supervisée :

2.9.1 Définition :

La classification supervisée est une technique d'apprentissage automatique (Machine Learning) qui vise à développer des règles pour classer des objets dans des classes prédéfinies à partir de données étiquetées. Voici une définition concise de la classification supervisée :

La classification supervisée consiste à définir des règles permettant de classer des objets dans des classes prédéterminées en se basant sur des données annotées. Elle est caractérisée par l'utilisation de couples input-output (variables explicatives et variables à expliquer) pour entraîner un modèle qui associera une nouvelle variable d'entrée non étiquetée à une variable de sortie.[8]

La classification supervisée est utilisée pour résoudre un large éventail de problèmes pratiques, tels que la détection de défauts, la lutte contre la fraude, le tri automatique, la reconnaissance d'images, et bien d'autres.

La classification supervisée contraste avec la classification non supervisée, également appelée clustering, qui ne nécessite pas de données étiquetées et vise à regrouper les données selon leurs ressemblances.[9]

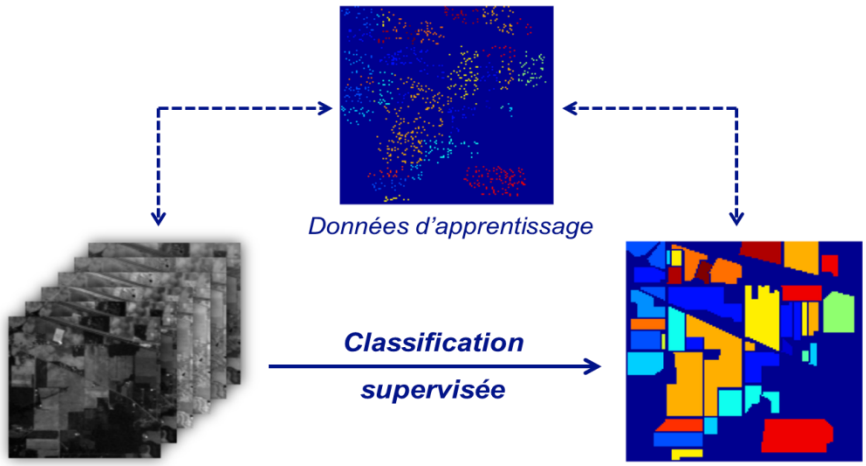


Figure 2.6 :Classification supervisé

2.8.2 Exemple :

On pourrait donner l'exemple le plus connu : problèmes d'aide ou diagnostic médical, où les superviseurs sont généralement les médecins afin de noter la classe des objets de l'ensemble

d'apprentissage à partir des remarques constatées. Ou bien l'exemple d'un tableau où le dernier descripteur (Jouer) représente la classes des exemples.

Numéro	Ensoleillement	Température (°F)	Humidité (%)	Vent	Jouer
1	soleil	75	70	oui	oui
2	soleil	80	90	oui	non
3	soleil	85	85	non	non
4	soleil	72	95	non	non
5	soleil	69	70	non	oui
6	couvert	72	90	oui	oui

TABLEAU. 2.3 : Base d'exemples "jouer un match" pour la classification

Ce tableau contient les données de l'apprentissage pour la classification dont chaque instance est labélisée par "Oui" et "Non". Ce qui permet de construire un modèle de classification permettant de prédire si ça marche pour jouer un match ou non ? La déduction se fait par rapport à l'apprentissage sur le jeu de données. Dans cet exemple on parle de « classification binaire », car on classifie les données en deux classes (|C| =2), et idem pour « n-aire » si on classifie les données en n classes. Nous allons présenter quelques techniques classiques de classification supervisée : les algorithmes étudiés sont présentés dans un ordre de difficulté croissant : l'algorithme « k plus proches voisins », « la classification bayésienne » .

2.8.3 ALGORITHMES:

Les algorithmes de classification supervisée sont des outils de machine learning qui permettent de prédire une classe ou une étiquette pour des données nouvellement introduites, en se basant sur des données étiquetées d'entraînement. Voici quelques exemples de ces algorithmes :

1 Les K-Plus Proches Voisins (K-PPV) ou K-Nearest Neighbors (KNN):

L'algorithme des k plus proches voisins (KNN) est une méthode d'apprentissage supervisé utilisée en intelligence artificielle. Il s'agit d'un algorithme simple et populaire qui peut être utilisé pour la classification et la régression.

Le principe de l'algorithme est de trouver les k voisins les plus proches d'une nouvelle observation dans le jeu de données d'apprentissage, puis d'attribuer à cette nouvelle observation la classe majoritaire parmi ces voisins.

Le nombre k représente le nombre de voisins à prendre en compte pour la prédiction.

Pour utiliser l'algorithme k-NN, il est nécessaire d'avoir des données labellisées pour l'apprentissage. Une fois l'apprentissage réalisé, l'algorithme calcule la distance entre la nouvelle observation et chaque observation du jeu de données d'apprentissage, puis sélectionne les k observations les plus proches.

Enfin, l'algorithme attribue à la nouvelle observation la classe majoritaire parmi ses k voisins [10]

Pour bien illustrer cette définition on va se servir de l'exemple ci-dessous,

On a un jeu de données qui permet de classer des individus dans deux familles A et B.

On ajoute un individu en noir. On prend k = 3.

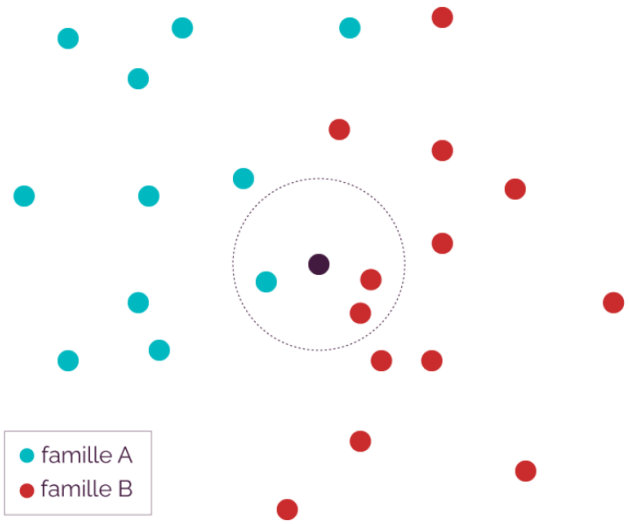


Figure 2.7 :Schéma d'une classification par la méthode KNN

En appliquant l'algorithme k-NN, l'individu fera partie de la famille B. Parmi ses 3 plus proches voisins, deux sont en effet rouges.

La distance euclidienne peut être calculée de la manière suivante :

$$|x_i - x_j| = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2} \quad \text{.....(2.1)}$$

où x_i et x_j sont des vecteurs avec p éléments. La distance euclidienne est un cas particulier de la distance L_q avec $q = 2$.

Ce qu'on peut remarquer sur cette méthode, c'est le coût de calcule qu'elle impose au fur et à mesure de ce processus de classification, car ce coût augmente avec chaque vecteur qu'on vient de classifier, plus on ajoute des nouveaux vecteurs déjà classés, plus que ça coûte augmente ce qui explique le temps d'exécution qu'elle prend pour classifier. en plus de la sensibilité de cet algorithme à l'initialisation des paramètres d'entrées (le choix de k , la distance utilisée ..) alors il faut que lors de la sélection des paramètres d'entrées que ces derniers respectent certaines contraintes (comme que k ne soit pas un multiple du nombre de classes pour éviter une surreprésentation d'une classe par apport à une autre .malgré ces

points , k-ppv reste une des méthodes les plus utilisées grâce à sa simplicité et robustesse et son caractère de généralisation à partir d'un nombre éminent de données d'apprentissage .

2 La classification bayésienne :

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

l'ensemble d'apprentissage «A» est connu, et chaque objet est étiqueté par sa classe «C_k», l'objectif est de chercher à classer un nouvel objet «X_{new} » non encore étiqueté. Le Classifieur baysien va choisir la classe «C_k» qui a la plus grande probabilité, on parle de règle MAP (maximum apostériori) (2.2):

$$C_{MAP} = argmax\ c_k \in p(c_k/X_{new}) =$$

$$argmax\ c_k \in \frac{p(X_{new}/c_k)p(c_k)}{p(x_{new})} = argmax\ c_k \in p(X_{new}\ /\ c_k)p(c_k) \quad \text{.....(2.2)}$$

Taille (cm)	Poids (kg)	Pointure (cm)	Sexe
182	81.6	30	masculin
180	86.2	28	masculin
170	77.1	30	masculin
180	74.8	25	masculin
152	45.4	15	féminin
168	68.0	20	féminin

Taille (cm)	Poids (kg)	Pointure (cm)	Sexe
183	59	20	???

FIGURE 2.8: La classification d'un nouvel exemple selon naïf bayes

donc Il nous faut estimer les probabilités $p(C_k)$et $p(X_{new}/C_k)$à partir des données d'apprentissage. Les probabilités a priori des classes $p(C_k)$, peuvent être estimées facilement par :

$$P(C_K) = \frac{Nombre\ d'element\ d'apprentissage\ de\ la\ classe\ Ck}{Le\ nombre\ totale\ de\ \ l'ensemble\ d'apprentissage}$$

Maintenant pour estimer les valeurs de $p(X_{new}/C_k)$, puisque les descripteur (attributs) de «X_{new}» sont indépendants, alors on aura grâce aux théories d'indépendance bayésienne entre les variables (2.3) :

$p(X_{new}/C_k) = p(f_1/C_k)p(f_2/C_k) \dots \dots p(f_n/C_k)$ …………………................… (2.3)

Ou les «f_i» sont les attributs qui décrivent l'ensemble de données , sachant que :

$P(C, f_1, f_2, \dots, f_n) = P(C) \prod_{i=1}^{n} p(f_i/C)$......…………………………….(2.4)

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

Le Classifieur naïf de Bayes est très performant même avec peu de données, car il fait souvent de bonnes hypothèses sur la distribution des données avec un peu de données d'entraînement afin d'estimer les paramètres nécessaires à la classification (moyennes et variances) **[Rennie & al. , 2003]**, mais lorsque le nombre de descripteurs est grand, il est parfois impossible de construire ce modèle sur des tableaux de probabilités.[11]

3 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 .

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 supervisé, 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 neurones biologiques ou les entrées de neurone artificiel correspondent aux dendrites biologiques, et la sortie de neurone artificiel correspond à l'axone de neurone biologique.[12]

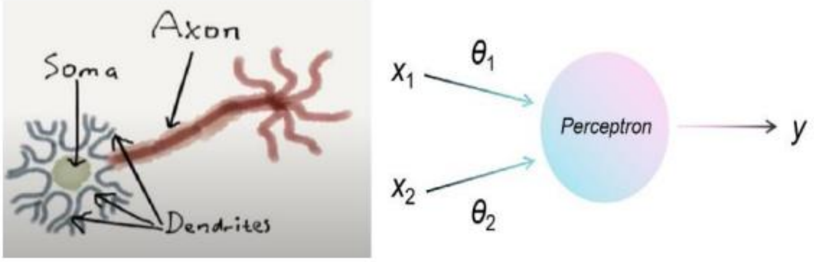


Figure 2.9: Neurone biologique et neurone artificiel

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

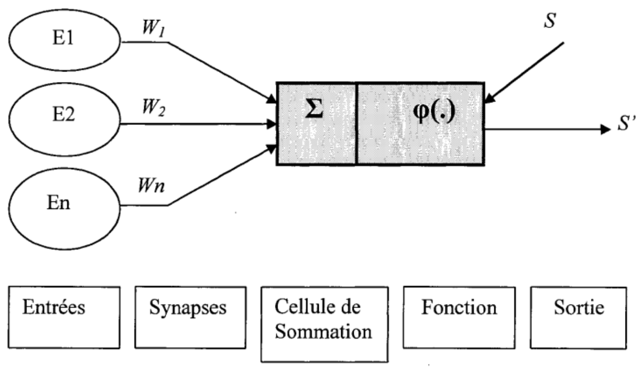


Figure 2.10:Représentation schématique du Perceptron

Les valeurs des entrées $El, . . . ,n$ représentent en général les attributs d'un objet à classer et les poids $Wl, . . . , Wn$ (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 :

$$S = f(x) = w.\, x + b \quad (2.5)$$

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

- Voyons comment fonctionne une classification linéaire avec un simple exemple

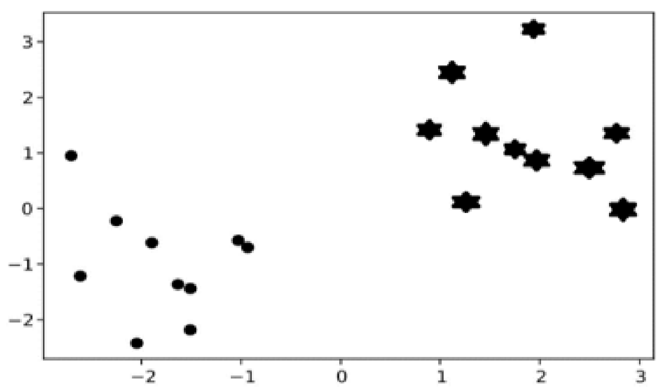


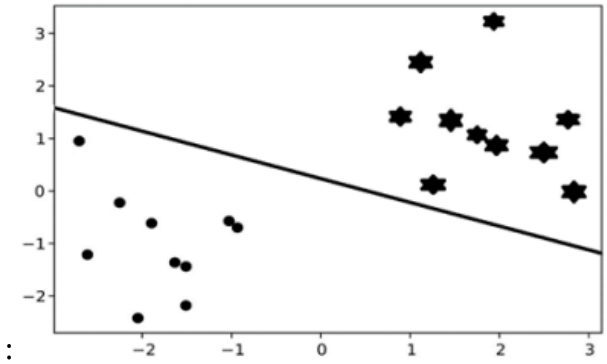
Figure 2.11: Données exemple

:

Nous voulons séparer ces données défini par des coordonnés (x_0, x_1)en deux classes. On appliquons*l'équationdeneurones* sur ces données après avoir calculé les paramètres adapté à ce problème $wetb$ nous obtiendrons les valeurs de sortie du perceptron:

$$S = \begin{cases} (+1)\ si\ w^T x + b\ \geq 0 \\ et \\ (-1)\ si\ w^T x + b\ < 0 \end{cases}$$

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



:

Figure 2.12: Classification linéaire par perceptron

4- Support Vector Machine (SVM) :

Les SVM (Support Vector Machines) sont une famille d'algorithmes d'apprentissage supervisé utilisés pour résoudre des problèmes de classification, de régression ou de détection d'anomalies Leur principe est de séparer les données en classes à l'aide d'une frontière aussi simple que possible, de telle façon que la distance entre les différents groupes de données et la frontière qui les sépare soit maximale. Les SVM sont qualifiés de "séparateurs à vaste marge", les "vecteurs de support" étant les données les plus proches de la frontière. Les algorithmes de SVM peuvent être adaptés à des problèmes de classification portant sur plus de 2 classes, et à des problèmes de régression.

Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables, les SVM utilisent une fonction noyau pour transformer l'espace de représentation des données d'entrée en un espace de plus grande dimension (possiblement de dimension infinie), dans lequel il est probable qu'il existe une séparation linéaire.[13]

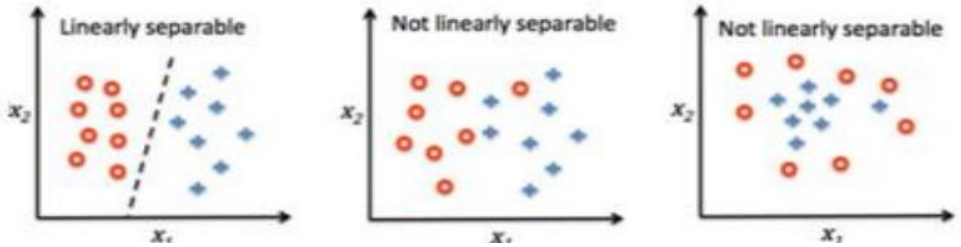


Figure 2.13 :Représentation des données linéairement séparables et non linéairement séparables

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

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

Si cette fonction est positive, le point x est classé en C1, et inversement, si elle est négative le point est classé en C 2. Se qui s'exprime mathématiquement sous la forme :

$$\begin{cases} f(x) \geq 0 \implies \text{x} \in \text{C1} \\ \qquad et \\ f(x) < 0 \implies \text{x} \in \text{C2} \end{cases}$$

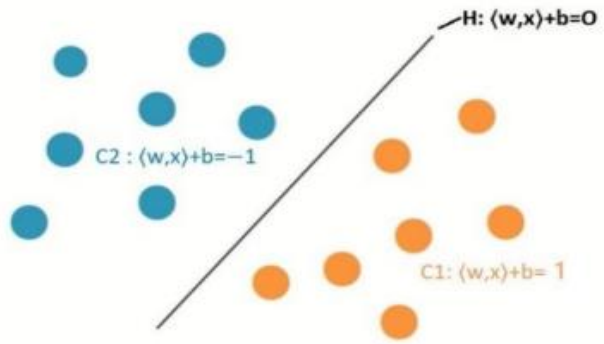


Figure 2.14: Séparateur SVM

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 donnés dans la catégorie correcte.

La marge se calcule avec la formule suivante : $Y = \frac{2}{|W|}$

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

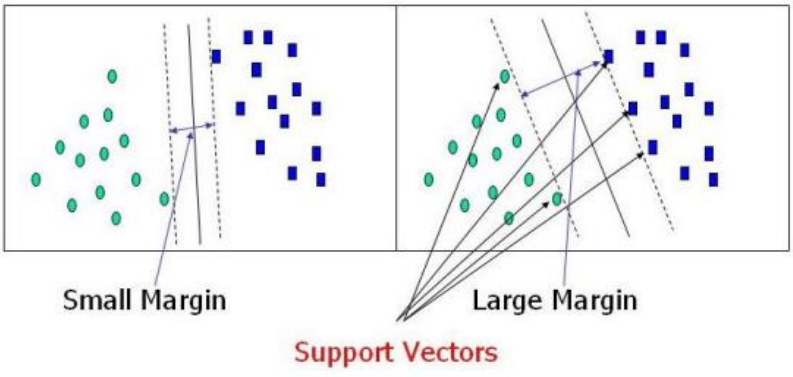


Figure 2.15: La marge séparatrice

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.

2.9 CLASSIFICATION NON SUPERVISE :

2.9.1 Définition:

La classification non supervisée, également désignée sous le terme français de "Clustering", est une technique d'apprentissage automatique (Machine Learning) ayant pour objectif de structurer des données en groupements homogènes (clusters) sans disposer de labels ni de classes prédéfinies. Plutôt que d'associer directement des entrées à des sorties particulières, la classification non supervisée met en évidence les schémas intrinsèques existants dans les données. Voici une définition concise de la classification non supervisée :

La classification non supervisée repose sur l'identification de similarités et de dissimilarités entre les éléments observés, aboutissant à la formation de groupes distincts. Ces groupes reflètent implicitement des tendances communes ou des comportements analogues au sein des données. Différemment de la classification supervisée, la classification non supervisée opère sans intervention extérieure pour attribuer initialement des classes aux données.[14]

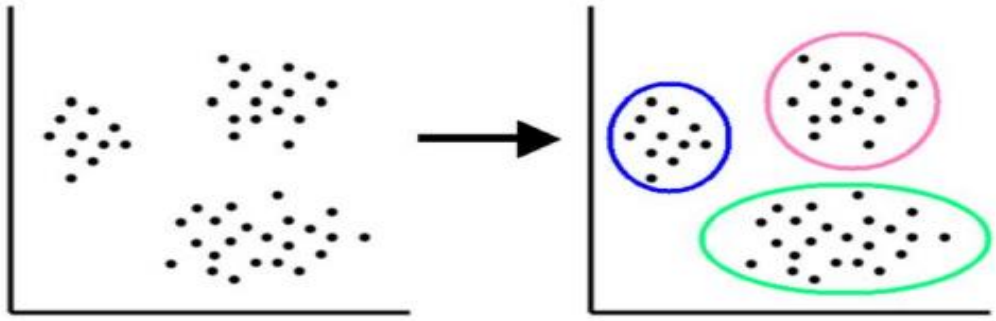


Figure 2.16: Classification non supervisé

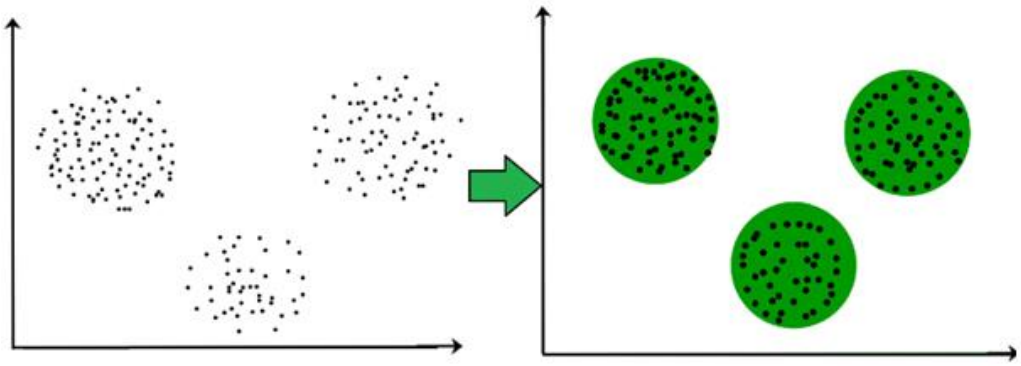


Figure 2.17 : Représentation d'un cluster

2.9.2 Exemple:

On utilise souvent ce type de classification en traitement d'images pour fixer les divers objets qu'elles contiennent (segmentation) : routes, villes, rues , des organes humaines (pour les images médicales) .

2.9.3 algorithme de classification non supervisée:

Les algorithmes de classification non supervisée sont utilisés pour regrouper des données sans étiquettes préalables en fonction de leurs similarités. Voici quelques exemples d'algorithmes de classification non supervisée :

1 K-moyennes (K-means) :

L'algorithme de k-means prend comme paramètre le nombre de clusters k et effectue un partitionnement d'un ensemble d'objets n en clusters k, avec l'objectif d'atteindre une grande similitude intra cluster et une faible similitude intercluster. La similitude entre les groupes est calculée par rapport à la valeur moyenne des objets au sein du groupe, appelé son centroïde [Han and Kamber, 2006].

L'algorithme commence par sélectionner aléatoirement des objets k, comme des centroïdes de groupe. Chacun des objets restants est affecté au cluster le plus proche en tenant compte de la distance entre l'objet et le centroïde du cluster, puis un nouveau centroïde pour les clusters est calculé. Le processus se répète jusqu'à la convergence. Le critère de convergence couramment utilisé est l'erreur carrée :

$$E = \sum_{i=1}^{k} \sum_{o \in C_i}^{K} |o - {m_i}^2| \quad \text{..(2.6)}$$

où E est la somme de l'erreur carré pour toutes les instances dans le jeu de données, o un objet représenté par un point dans l'espace et mi est la moyenne du groupe Ci. L'objectif de ce critère est de créer des groupes aussi compacts et séparés que possible [Han et Kamber, 2006].

La complexité temporelle de l'algorithme de k-means est O(lknm), où l représente le nombre d'itérations, k désigne le nombre de groupes, n est le nombre des objets et m'indique le numéro d'attributs. Nous devons préciser qu'il existe également des variations de l'algorithme classique, qui appliquent différentes techniques d'optimisation.[15]

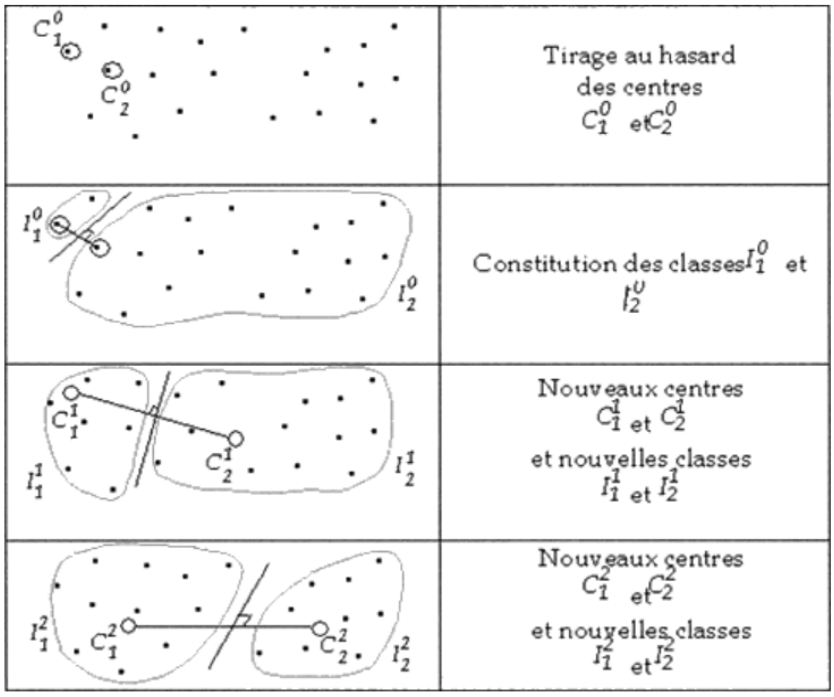


Figure 2.18:Exemple de l'algorithme K-means

Mathématiquement, l'algorithme des k-means fait généralement intervenir la distance euclidienne. Soient deux groupes d'éléments $p = (p_1, \dots, p_n)$, et $q = (q_1, \dots, q_n)$
alors la distance d entre les points p et q se calcule avec cette formule.

$$D(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \text{.....[2.7]}$$

$\boldsymbol{q_i}$:*Données*

$\boldsymbol{p_i}$:$Centredeclasse$

Et on affecte chaque individuq_i à la classe $c_p$pour laquelle la distance du centre est minimale :

$$C_i = k = \text{argmin}\|q_i - p_i\|^2 \text{ .....(2.8)}$$

$\|q_i - p_i\|^2$: $Distanceeuclidienneaucarré$

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

$$U_K = \frac{\sum_{i=1}^{N} 1\{C_i=k\}.q_i}{\sum_{i=1}^{N} 1\{C_i=k\}} \text{.....(2.9)}$$

$C_i = laclassedex_i$

En résumé, L'algorithme du k-means est un algorithme très utilisé en clustering. Il fonctionne généralement bien, il est rapide et relativement simple à comprendre. Il est non déterministe,

c'est-à-dire que les clusters obtenus peuvent changer légèrement si on relance l'algorithme plusieurs fois. Il a toutefois besoin qu'on lui spécifie le nombre de clusters à produire.

Pour choisir le nombre de clusters, on applique la méthode du "coude", et on cherche une "cassure" dans la courbe liant la variance intraclasse au nombre de clusters. [16]

La figure 2.18 suivante donne un aperçu de l'application des différentes étapes de l'algorithme des k-means pour le partitionnement (clustering) en 2 classes ($\boldsymbol{k} = \mathbf{2}$) :

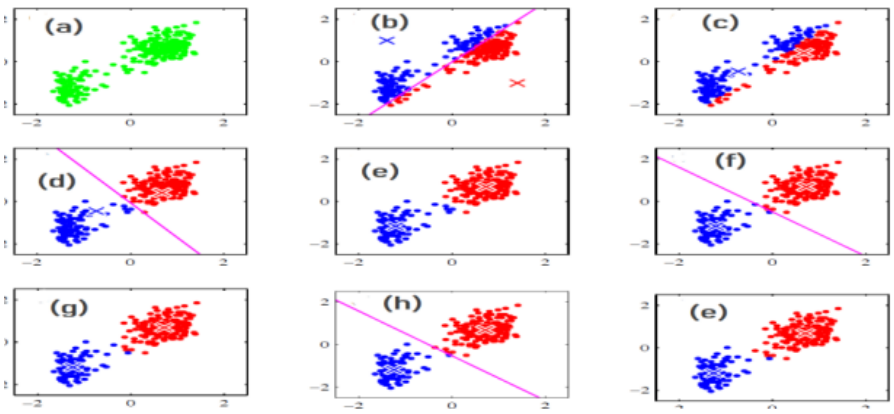


Figure 2.19: Aperçu de l'algorithme K-means

2 Algorithme Fuzzy C-Means :

Le Fuzzy c-means clustering (FCM) [Albayrak et Amasyali, 2003], [Jain et Dubes, 1988], ou fuzzy ISODATA, représente une méthode de cluster qui est différente de la cluster de k-mesures dures. FCM utilise l'idée de partitionnement flou, où une instance de données (objet) est attribuée à tous les groupes avec des degrés d'adhésion (variant de 0 à 1).

FCM utilise des ensembles flous dans le processus de regroupement, associant à chaque objet un degré d'appartenance à chaque groupe. Dénominateur par k le nombre souhaité de groupes, une matrice *U* est utilisée, où $U_{ij}(i \in \{1,2,,,,k\}, j \in \{1,2,,,,n\})$exprime le degré d'appartenance de l'objet j au cluster i, tel que :

$$\sum_{i=1}^{k} u_{ij} = 1, \forall j \in \{1,2,,,,,,,n\} \text{ .....(2.10)}$$

Grâce à un processus itératif, FCM met à jour les centroïdes des clusters et les degrés d'adhésion, afin de déplacer les moyens du cluster au bon endroit dans l'ensemble de données. La convergence de FCM vers la solution optimale n'est pas assurée, en raison de l'initialisation aléatoire des centroïdes initiaux, c'est-à-dire les valeurs initiales pour la matrice U. L'algorithme rapporte les valeurs finales de la matrice *U*. Compte tenu des degrés de membre définitifs donnés par la matrice *U*, un objet $O_j$est généralement attribué au groupe $i = argmax\ i = 1, k\ u_{ij}$ Ensuite, la formule suivante est utilisée :

$$k_i = \{j\ , j \in \{1,2,,,,,n\}, u_{ij} \geq u_{rj}\ , \forall r \in \{1,,,,,,n\}, r \neq i\}$$

afin d'obtenir les clusters dans les données après l'application de FCM. Le nombre de groupes formés par FCM est au plus k, car des groupes vides pourraient être obtenus.

La complexité du temps pour l'algorithme de regroupement FCM est O(nmk^2i), où $i$représente le nombre de passages de FCM sur l'ensemble de données, le nombre d'objets est exprimé par n, le nombre de groupes par k et le nombre des attributs par m.[17]

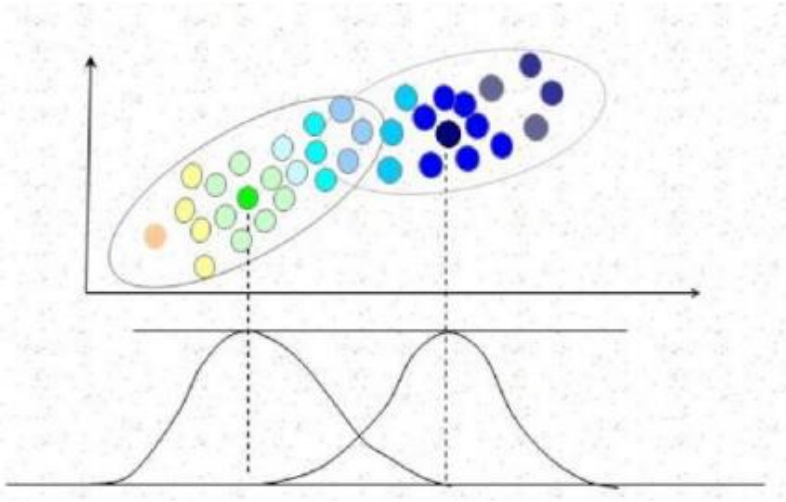


Figure 2.20 : Exemple des classes floues

3 Classification ascendante hiérarchique :

Les algorithmes hiérarchiques de regroupement effectuent un partitionnement d'objets dans un arbre de groupes. Selon la façon dont la décomposition hiérarchique est faite, du bas vers le haut en fusionnant ou du haut vers le bas en divisant, les algorithmes de regroupement hiérarchisés peuvent être classés en agglomérations ou divisibles.

La méthode de regroupement agglomératif, commence par une partition dans laquelle chaque objet est placé dans son propre singleton. Ensuite, la paire de clusters les plus proches est fusionnée en un seul, créant une partition et notamment d'une par le nombre de groupes. Cette étape se répète jusqu'à ce que tous les objets appartiennent à un seul groupe. La méthode de regroupement hiérarchique divisive effectue les mêmes étapes, mais dans l'ordre inverse : commence par une partition où tous les objets sont dans un seul groupe, et se termine lorsque tous les objets sont placés dans leur propre groupe.

Le processus de regroupement hiérarchique à une représentation semblable à un arbre, appelé dendrogramme, qui illustre comment les objets sont regroupés ensemble étape par étape.

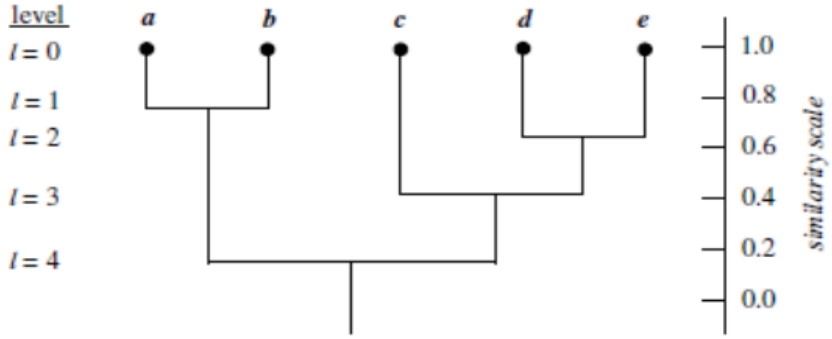


Figure 2.21: Dendrogramme pour le regroupement hiérarchique de cinq objets

Pour décider quels groupes fusionner, la distance entre les groupes est calculée.

Il existe plusieurs mesures utilisées dans la littérature pour la distance entre deux groupes, désignées par $d(C_i, C_j)$:

Distance minimale:[Han and Kamber, 2006] :$d_{min}(C_i, C_j) = \min d(o, o')\ , o \in C_j, o' \in C_j$

Distance maximale:[Han and Kamber, 2006] :$d_{max}(C_i, C_j) = \max d(o, o'), o \in C_i, o' \in C_j$

Moyenne distance: [Han and Kamber, 2006] : $d_{mean}(C_i, C_j) = d(m_i, m_j)$

où$d(o, o')$ exprime la distance entre deux objets o et o', m_i représente le nombre d'objets de C_i et m_i est la moyenne de C_i.

L'algorithme de regroupement qui mesure la distance minimale entre les regroupements est généralement appelé le plus proche voisin. En outre, si le processus de regroupement se termine lorsque la distance entre les regroupements les plus proches est supérieure à un seuil établi, il est appelé algorithme de liaison unique. L'algorithme de regroupement qui mesure la distance maximale entre les regroupements est généralement appelé le plus lointain voisin. Si le processus de regroupement se termine lorsque la distance entre les regroupements les plus proches est supérieure à un seuil établi, il est appelé algorithme de liaison complète.

Dans le contexte où les mesures de distance minimale et maximale sont sensibles aux extrémités et aux données bruyantes, représentant deux extrêmes, la distance moyenne ou moyenne est souvent choisie comme un compromis entre les deux. La distance moyenne est utilisée pour les données numériques, tandis que la moyenne peut également traiter des données catégoriques.

La complexité du temps d'un algorithme de regroupement hiérarchique dépend du type de liaison. Dans le cas d'un cluster de lien unique, la complexité est $O(n^2)$tandis que pour un lien complet, c'est $O(n^2)\log(n)$où n est le nombre d'objets.[18]

2.9.3 Exemples d'application :

La classification non supervisée, également connue sous le nom de clustering, est utilisée dans divers domaines pour regrouper des données sans l'aide de catégories prédéfinies. Voici quelques domaines d'application de la classification non supervisée :

•Biologie : L'élaboration de la taxonomie animale est un exemple d'application en biologie où la classification non supervisée est utilisée pour regrouper des espèces en fonction de leurs caractéristiques communes.

•Psychologie : La détermination des types de personnalités présents dans un groupe d'individus est un exemple d'application en psychologie où la classification non supervisée est utilisée pour identifier des profils similaires.

•Text Mining : Le partitionnement de courriels ou de textes en fonction du sujet traité est une application courante où la classification non supervisée est utilisée pour regrouper des documents similaires.

•Détection de fraudes : Identifier si une transaction bancaire est frauduleuse ou pas est un exemple d'application où la classification non supervisée peut être utilisée pour détecter des schémas anormaux.

•Reconnaissance d'images : Reconnaître des chiffres écrits à la main ou identifier le type de cancer dont souffre un patient sont des exemples d'applications en reconnaissance d'images où la classification non supervisée peut être utilisée pour regrouper des données similaires.[19]

2.10 Indicateurs de performance d'un modèle de classification :

1La matrice de confusion:

La matrice de confusion est un tableau de mesure de la performance des modèles de classification à deux classes ou plus. Dans le cas binaire, elle est composée de quatre valeurs représentant les différentes combinaisons de valeurs réelles et valeurs prédites. La matrice de confusion est indispensable pour définir les différentes métriques de classification telles que l'Accuracy, la Précision-Recall, etc.

Elle dépend des valeurs réelles de la variable cible, des probabilités d'un modèle prédictif, et d'un seuil de classification pour traduire ces probabilités en labels. Un grand nombre de seuils sont possibles pour un même modèle, ce qui rend la matrice de confusion "mouvante". Elle est utilisée pour évaluer la performance d'un modèle de classification et est un outil essentiel pour les data scientists et les non-data scientists qui cherchent à comprendre la performance des algorithmes.

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 positif sinon le teste est négatif.

True positifs (VP) : Est le nombre de personnes malade .

TrueNegative (TN) : Est le nombre de personnes qui ne sont pas malades.

False Positive (FP) : Est le nombre de personne qui ne sont pas malade mais ils sont classées comme malade

False Negative (FN) : Est le nombre de personnes qui sont malades mais qui sont classées comme pas malade.[20]

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

Table 2.4 : matrice de confusion

Un classifieur parfait correspond à une matrice diagonale.

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

- **Précision prédictions**: La précision mesure la proportion des prédictions positives correctes parmi l'ensemble des prédictions positives faites par le modèle. La formule de la précision est : [24].

  Précision = TP/ (TP + FP) … … … … …… … … . . … … ……. . (2.11)

- **Rappel (Recall) :** Le rappel mesure la proportion des vrais positifs correctement identifiés parmi tous les exemples réellement positifs.

  Rappel = TP/ (TP + FN)..... (2.12)

- **F-mesure :** La mesure F1 combine précision et rappel. Le résultat est la moyenne harmonique des deux valeurs ,il mesure la capacité d'un modèle à bien prédire les individus positifs, tant en termes de precision **(taux de prédictions positives correctes)** qu'en termes de recall **(taux de positifs correctement prédits)** il est calculé comme suit :

  F1 = 2 × (Précision × Rappel) / (Précision + Rappel)..... (2.13)

Prenons l'exemplede la matrice de confusion suivante représentant trois classe Water, Forest, Urban :

		Reference Data			
		Water	Forest	Urban	Total
Classified Data	Water	21	6	0	**27**
	Forest	5	31	1	**37**
	Urban	7	2	22	**31**
	Total	**33**	**39**	**23**	***95***

Voici les résultats ci ‘’**water**’’est utilisé comme réponse positive.

- **Précision =21 / (21+ 5+7) =0,6363**
- **Rappel = 21 / (21 + 6+0) = 0,7777**
- **F1 = 2 × (0,6363× 0,7777) / (0,6363 + 0,7777) = 0,699**

Comme nous pouvons le voir, la valeur F1 se situe entre les valeurs de précision et de rappel. Bien que la précision F1 ne soit pas aussi facile à comprendre, elle ajoute de la nuance au nombre de précision de base.

2.10.2 Mesures de précision:

Il existe de nombreuses façons différentes d'évaluer l'exactitude thématique d'une classification. La Matrice de confusion nous permet de calculer les mesures de précision suivantes :

1 Over all Accuracy (précision global):

La précision globale nous indique essentiellement, parmi tous les sites de référence, quelle proportion a été cartographiée correctement. La précision globale est généralement exprimée en pourcentage, une précision de 100 % étant une classification parfaite dans laquelle tous les sites de référence ont été correctement classés. La précision globale est la plus simple à calculer et à comprendre, mais elle ne fournit en fin de compte à l'utilisateur et au producteur de la carte que des informations de base sur la précision. C'est-à-dire la somme des vrais positifs et des vrais négatifs divisée par le nombre total d'individus testés.[21]

$$\text{OA} = \frac{\textit{Le nombre de sites correctement classés}}{\textit{Le nombre total de sites de référence}} \quad \ldots\ldots \quad (2.14)$$

Exemple basé sur la matrice de confusion précédente :

Nombre de sites correctement classés :**21+31+22 = 74**

Nombre total de sites de référence : **95**

Précision globale = 74/95 = 77,9 %

2 Le coefficient Kappa:

Le coefficient Kappa est généré à partir d'un test statistique pour évaluer l'exactitude d'une classification. Kappa évalue essentiellement les performances de la classification par rapport à une simple attribution aléatoire de valeurs, c'est-à-dire si la classification a fait mieux que le hasard. Le coefficient Kappa peut aller de -1 à 1. Une valeur de 0 indique que la classification n'est pas meilleure qu'une classification aléatoire. Un nombre négatif indique que la classification est nettement pire qu'aléatoire. Une valeur proche de 1 indique que la classification est nettement meilleure qu'aléatoire. Il convient de noter que le coefficient $Kappa$ doit être utilisé avec prudence et dans le contexte approprié. Il est recommandé de l'interpréter conjointement avec d'autres métriques d'évaluation pour obtenir une évaluation complète des performances du modèle de classification [21].

$$Kappa = \frac{N\sum m_{i,i} - \sum(G_i C_i)}{N^2 - \sum(G_i C_i)} \quad \ldots\ldots \quad (2.15)$$

- $\boldsymbol{i}$: Est le numéro de classe
- $\boldsymbol{N}$: Est le nombre total de valeurs classer
- $\boldsymbol{m_{i,i}}$: Est le nombre de valeurs appartenant à la classe de vérité $\boldsymbol{i}$qui ont également été classées dans la classe *i*
- $\boldsymbol{C_i}$: Est le nombre total de valeurs prédites appartenant à la classe *i*
- $\boldsymbol{G_i}$ **:** Est le nombre total de valeurs de vérité appartenant à la classe *i*

  D'après l'exemple de la matrice de confusion précédant, le coefficient kappa est de 0.6662

$$Kappa = \frac{95.(21+31+22) - (27*33 + 37*39 + 31*23)}{95^2 - (27*33 + 37*39 + 31*23)} = 0.6662$$

2.11 CONCLUSION :

Dans ce chapitre, nous avons discuté des concepts de classification, en explorant les divers types de classifications, ses méthodes, la conception de quelques algorithmes, domaine d'application de la classification et Indicateurs de performance .

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

Chapitre III
Les réseaux de neurones

3 Les réseaux de neurones:

3.1- Introduction:

Après un bref historique, le neurone artificiel est présenté avec les différentes fonctions d'activation. Le principe des réseaux est introduit avec les étapes d'apprentissage et d'inférence. Le perceptron monocouche avec des exemples d'implantation des fonctions logiques ET et OU, puis le perceptron multicouche sont introduits, avec les principes des algorithmes de rétropropagation du gradient pour la mise à jour des poids des neurones. Les caractéristiques des réseaux de neurones profonds sont présentées, notamment les réseaux convolutionnels (CNN) et des réseaux récurrents. L'implantation des réseaux de neurones est abordée : programmation à l'aide de bibliothèques comme Tensor Flow et supports matériels (processeurs neuronaux, opérateurs matériels spécialisés, formats de données réduits, etc.). Les caractéristiques permettant d'utiliser des réseaux de neurones pour l'embarqué sont présentées.

3.2- Chronologie :

Le champ des réseaux neuronaux va démarrer par la présentation en 1943 par W. MCCulloch et W. Pitts du neurone formel qui est une abstraction du neurone physiologique. Le retentissement va être énorme. Par cette présentation, ils veulent démontrer que le cerveau est équivalent à une machine de Turing, la pensée devient alors purement des mécanismes matériels et logiques. Il déclara en 1955 "Plus nous apprenons de choses au sujet des organismes, plus nous sommes amenés à conclure qu'ils ne sont pas simplement analogues aux machines, mais qu'ils sont machine." Mysterium Iniquitatis of Sinful Man Aspiring into the Place of God, repris in Embodiments of mind. La démonstration de McCulloch et Pitts sera un des facteurs importants de la création de la cybernétique.

En 1949, D. Hebb présente dans son ouvrage The Organization of Behavior une règle d'apprentissage. De nombreux modèles de réseaux aujourd'hui s'inspirent encore de la règle de Hebb.

En 1958, F. Rosenblatt développe le modèle du Perceptron. C'est un réseau de neurones inspiré du système visuel. Il possède deux couches de neurones : une couche de perception et une couche lié à la prise de décision. C'est le premier système artificiel capable d'apprendre par expérience .

Dans la même période, Le modèle de L'Adaline (ADAptiveLINearElement) a été présenté par B. Widrow, chercheur américain à Stanford. Ce modèle sera par la suite le modèle de base des réseaux multi-couches.

En 1969, M. Minsky et S. Papert publient une critique des propriétés du Perceptron. Cela va avoir une grande incidence sur la recherche dans ce domaine. Elle va fortement diminuer jusqu'en 1972, où T. Kohonen présente ses travaux sur les mémoires associatives. et propose des applications à la reconnaissance de formes.

C'est en 1982 que J. Hopfield présente son étude d'un réseau complètement rebouclé, dont il analyse la dynamique.

Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines (entre autres, vie artificielle et intelligence artificielle) à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques.[22]

3.3- Définition :

Un réseau de neurones artificiels est un système dont la conception est à l'origine inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques. Les réseaux de neurones artificiels sont des réseaux fortement connectés par des processeurs élémentaires fonctionnant en parallèle. Chaque processeur élémentaire (neurone artificiel) calcule une sortie unique sur la base des informations qu'ils reçoivent.[23]

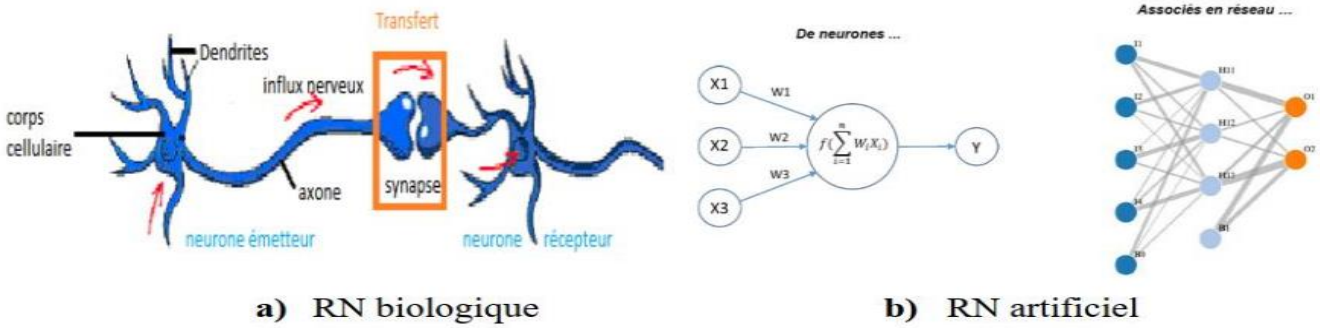


Figure 3.1 : Définition d'un réseau de neurones

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

Tableau 3.1 : Correspondance neurone biologique/neurone artificiel

3.4- Neurone artificiel :

Un réseau de neurones artificiels est un ensemble organisé de neurones interconnectés qui permet la résolution de problèmes complexes en intelligence artificielle. Ces réseaux sont caractérisés par un grand nombre de couches de neurones, dont les coefficients de pondération sont ajustés au cours d'une phase d'entraînement, ce qui est connu sous le nom d'apprentissage profond. Ils sont utilisés dans des domaines tels que la vision par ordinateur, le traitement du langage naturel et d'autres tâches nécessitant la modélisation de relations complexes entre les données. 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, comme montré dans l'équation suivante.[24]

$$y_j = \varphi(a_j) = \varphi(\sum_{i=0}^{n}(W_i * X_i) \ + \ b_j) \quad \text{.....(3.1)}$$

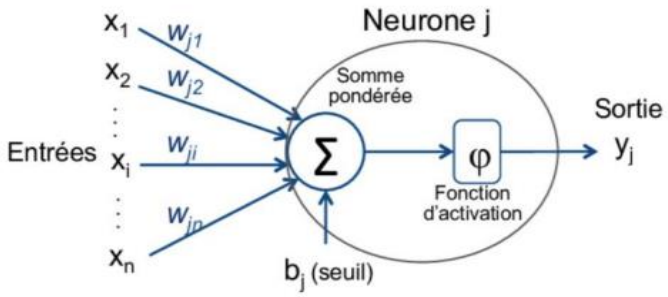


Figure 3.2 : Schéma d'un Neurone artificiel

3.5- Architecture d'un réseau de neurones artificiel

En général, un réseau de neurones est composé d'un ensemble de couches successives, où chaque couche reçoit ses entrées à partir des sorties de la couche précédente. Cela signifie que l'ensemble du réseau est entièrement connecté. Chaque couche est constituée de neurones qui ne sont pas connectés entre eux, mais qui reçoivent des informations numériques des neurones voisins. L'ensemble des couches comprend une couche d'entrée qui lit les valeurs d'entrée, une couche de sortie qui fournit les résultats du système, et entre ces deux se trouvent une ou plusieurs couches cachées qui participent au processus de transfert des informations [25].

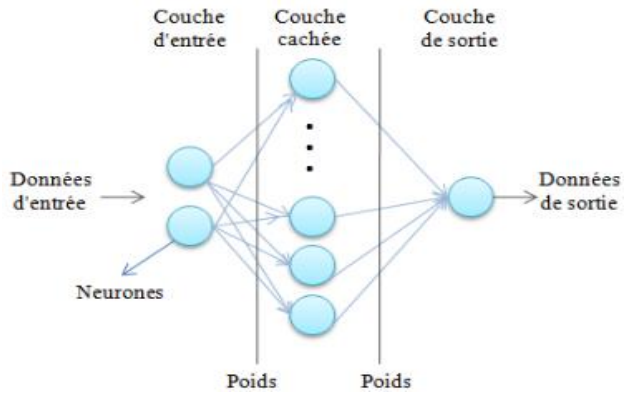


Figure 3.3: Architecture d'un réseau de neurones

3.6 -Types des réseaux de neurones :

Les types de réseaux de neurones comprennent une variété d'architectures spécialisées pour des tâches spécifiques en intelligence artificielle. Voici une liste des principaux types de réseaux de neurones :

1- Les réseaux de neurones à convolution(CNN/ConvNet) :

Les réseaux de neurones à convolution, également connus sous le nom de CNN (Convolutional Neural Networks), sont des architectures spécifiques de réseaux de neurones qui sont particulièrement efficaces pour l'analyse et la reconnaissance d'images. Ces réseaux sont conçus pour apprendre directement à partir des données et sont capables de détecter des motifs et des caractéristiques complexes dans les images pour effectuer des tâches de classification et de segmentation.

Les CNN peuvent être utilisés pour diverses tâches telles que la classification d'images, la détection d'objets, la segmentation d'images médicales, et bien d'autres applications où l'analyse d'images est nécessaire.[26]

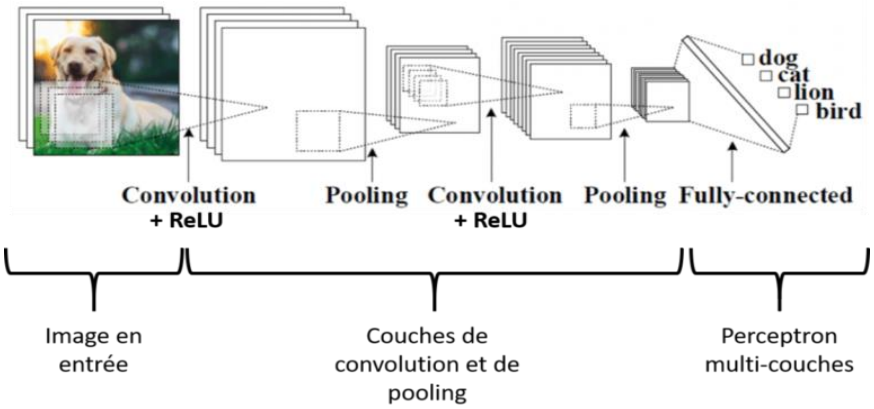


Figure 3.4: Architecture générale d'un réseau de neurones convolutif

1.2 -Fonctionnement des CNN:

Les réseaux de neurones convolutifs (CNN) sont des architectures spécifiques de réseaux de neurones conçues pour l'analyse et la reconnaissance d'images.Ils présentent quatre principaux types de couches

1.2.1 -Couche de convolution:

La couche de convolution est la composante clé des CNN, utilisant des filtres pour extraire des caractéristiques des images en entrée. Les filtres sont appliqués à l'image pour détecter des motifs visuels et générer des "featuremaps" qui capturent ces caractéristiques.

1.2.2- Couche d'activation ReLU :

Après la convolution, une fonction d'activation comme ReLU est souvent appliquée pour introduire de la non-linéarité dans le réseau.ReLU remplace les valeurs négatives par zéros, accélérant la convergence de l'apprentissage et aidant à extraire des caractéristiques pertinentes.

1.2.3- Couche de pooling:

Les couches de pooling sont utilisées pour réduire la dimension des "featuremaps" tout en préservant les caractéristiques importantes. Le max pooling et l'average pooling sont les types courants de pooling qui agrègent les informations en conservant les valeurs maximales ou moyennes.

1.2.4- Couche entièrement connectée (FC):

Les couches fully connected sont placées à la fin des CNN pour classifier les données en affectant des probabilités à chaque classe possible. Chaque neurone de la couche FC est connecté à tous les neurones de la couche précédente, permettant une classification précise des données.

La couche de convolution est la première couche d'un réseau convolutif. Bien qu'elle puisse être suivie de couches de convolution supplémentaires ou de couches de pooling, la couche entièrement connectée est la couche finale. Avec chaque couche, le CNN augmente sa complexité, identifiant de plus grandes portions de l'image. Les couches précédentes se concentrent sur des caractéristiques simples, telles que les couleurs et les bords. À mesure que les données image progressent dans les couches du CNN, celui-ci commence à reconnaître des éléments ou des formes plus importants de l'objet jusqu'à ce qu'il identifie enfin l'objet attendu.[27]

1.3- Architecture de réseaux de neurone convolutif :

L'architecture d'un réseau de neurones convolutif (CNN) est composée de différentes couches spécialisées qui travaillent ensemble pour extraire des caractéristiques des images en entrée. Voici un aperçu de l'architecture typique d'un CNN basé sur les sources fournies:

1.3.1 Couche de Convolution (CONV):

La couche de convolution est un élément essentiel des réseaux neuronaux convolutifs (CNN) et est généralement la première couche de ces réseaux.

Elle analyse les images en entrée et détecte des ensembles de caractéristiques à différents niveaux de complexité. Les couches de convolution sont formées de ce qu'on appelle des filtres. Les filtres sont des tableaux de valeurs appelées featuremaps. Chaque couche de convolution prend en entrée une image et produit une featuremap. Chaque featuremap est obtenue en appliquant le filtre à l'image. Par exemple, si l'image est de taille 5x5 et que le filtre est de taille 3x3, la featuremap sera de taille 3x3. La couche de convolution reçoit donc en entrée plusieurs images et calcule la convolution de chacune d'entre elles avec chaque filtre. Les filtres correspondent exactement aux features que l'on souhaite retrouver dans les images.[28]

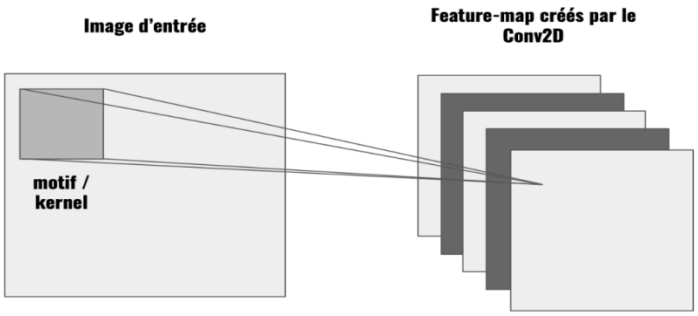


Figure 3.5: Couche de Convolution

1.3.2 Couche de Pooling (POOL):

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs featuremaps, et applique à chacune d'entre elles l'opération de pooling.

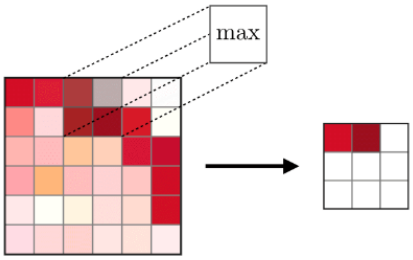
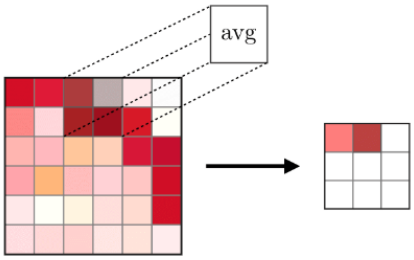
L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille

2×2pixels qui ne se chevauchent pas, ou des cellules de taille 3×3 pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc).

On obtient en sortie le même nombre de *featuremaps* qu'en entrée, mais celles-ci sont bien plus petites.La couche de *pooling* permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur apprentissage.

Les valeurs maximales sont repérées de manière moins précise dans les featuremaps obtenues après pooling que dans celles reçues en entrée – c'est en fait un grand avantage ! En effet, lorsqu'on veut reconnaître un chien par exemple, ses oreilles n'ont pas besoin d'être localisées le plus précisément possible : savoir qu'elles se situent à peu près à côté de la tête suffit !

Ainsi, la couche de *pooling* rend le réseau moins sensible à la position des *features* : le fait qu'une *feature* se situe un peu plus en haut ou en bas, ou même qu'elle ait une orientation légèrement différente ne devrait pas provoquer un changement radical dans la classification de l'image.[29]

Taper	Mise en commun maximale	Mise en commun moyenne
But	Chaque opération de pooling sélectionne la valeur maximale de la vue actuelle	Chaque opération de regroupement fait la moyenne des valeurs de la vue actuelle
Illustration		
commentaires	• Préserve les fonctionnalités détectées • Le plus couramment utilisé	• Carte des fonctionnalités de sous-échantillonnage • Utilisé dans LeNet

5

1.3.2.1 Types de pooling : Les types de pooling couramment utilisés dans les réseaux de neurones convolutifs (CNN) sont le max pooling et l'average pooling.

Max Pooling: Dans le max pooling, la valeur de sortie pour chaque région de pooling est simplement la valeur maximale des valeurs d'entrée dans cette région.

Average Pooling:En revanche, dans l'avertissement pooling, la valeur de sortie pour chaque région de pooling est la moyenne des valeurs d'entrée dans cette région.[29]

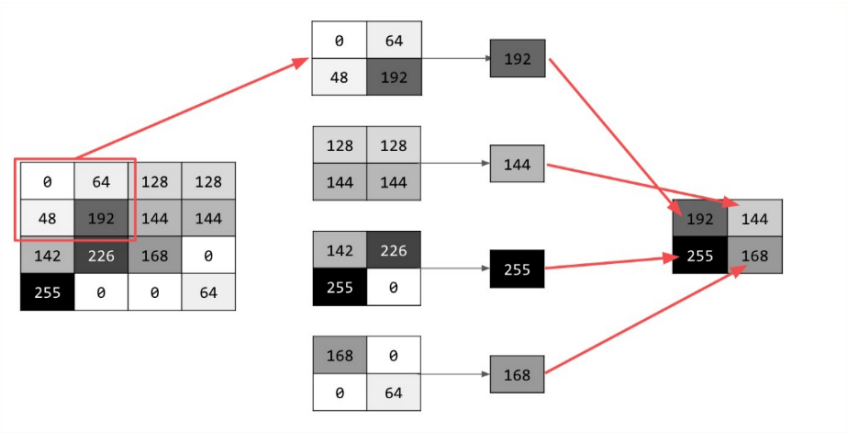


Figure 3.6: pooling maximal

1.3.3 Couche d'Activation ReLU:

La couche d'activation ReLU (RectifiedLinear Unit) est une fonction non linéaire couramment utilisée dans les réseaux de neurones, notamment dans les réseaux de neurones convolutifs (CNN).La fonction ReLU remplace toutes les valeurs négatives en entrée par des zéros, laissant les valeurs positives inchangées.

Cette fonction introduit une non-linéarité dans le réseau, ce qui est crucial pour permettre au réseau de modéliser des relations complexes et d'apprendre des représentations plus riches.

En plus de son rôle d'activation, ReLU aide à résoudre le problème de disparition du gradient lors de l'entraînement des réseaux profonds, ce qui contribue à une meilleure convergence du modèle. L'intérêt de ces couches d'activation est de rendre le modèle non linéaire et de ce fait plus complexe.

ReLU (*Rectified Linear Units*) désigne la fonction réelle non linéaire définie par :

$RelU(x) = max(0, x)$ Cette fonction force les neurones à retourner des valeurs positives [30]

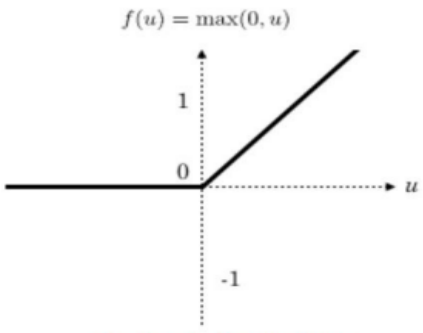


Figure 3.7:Fonction d'activation Relu

1.3.4 Couche FullyConnected (FC) :

La couche Fully Connected (FC) est une composante essentielle des réseaux de neurones, souvent située à la fin des architectures de Convolutional Neural Networks (CNN).La couche FC est entièrement connectée à tous les neurones de la couche précédente, ce qui signifie que chaque entrée est connectée à chaque sortie de la couche.Après avoir reçu un vecteur en entrée, la couche FC applique une combinaison linéaire suivie d'une fonction d'activation pour classifier les données en sortie.

La couche FC est cruciale pour la classification des données en attribuant des probabilités à chaque classe possible.Elle renvoie un vecteur de taille correspondant au nombre de classes, où chaque élément indique la probabilité pour l'entrée d'appartenir à une classe spécifique.[31]

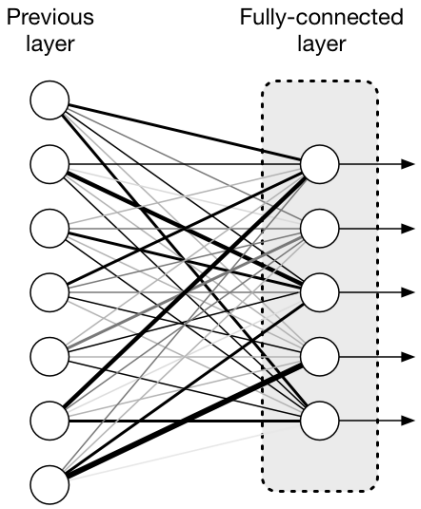


Figure 3.8: Une couche fully-connected (FC)

2 Réseaux de neurones récurrents:

Un réseau de neurones récurrent (RNN) est un type de réseau de neurones artificiel qui utilise des données séquentielles ou des données de séries temporelles. Ces algorithmes d'apprentissage en profondeur sont couramment utilisés pour des problèmes ordinaux ou temporels, tels que la traduction linguistique, le traitement du langage naturel, la reconnaissance vocale et le sous-titrage d'images ; ils sont incorporés dans des applications populaires telles que Siri, la recherche vocale et Google Translate. Comme les réseaux de neurones convolutifs (CNN) à propagation avant, les réseaux de neurones récurrents utilisent des données d'entraînement pour apprendre. Ils se distinguent par leur « mémoire » car ils prennent des informations d'entrées antérieures pour influencer l'entrée et la sortie en cours. Alors que les réseaux de neurones profonds traditionnels supposent que les entrées et les sorties sont indépendantes les unes des autres, la sortie des réseaux de neurones récurrents

dépend des éléments antérieurs au sein de la séquence. Alors que les événements futurs seraient également utiles pour déterminer la sortie d'une séquence donnée, les réseaux de neurones récurrents unidirectionnels ne peuvent pas rendre compte de ces événements dans leurs prédictions.[32]

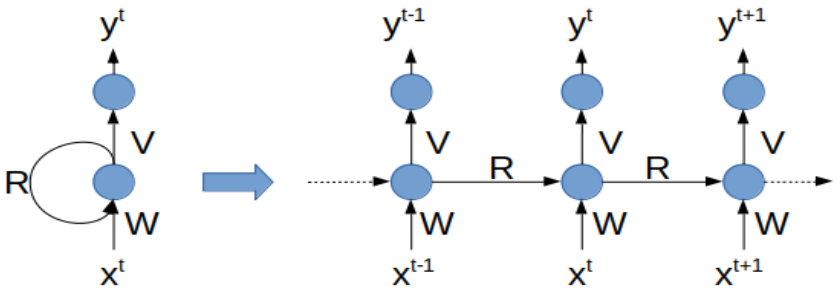


Figure 3.9 : (à gauche) Un RNN (à droite) Sa version déroulé Source

Pour nous aider à expliquer les RNN, prenons une expression idiomatique, telle que « ne pas être dans son assiette », qui est couramment utilisée lorsque quelqu'un est malade. Pour que cette expression ait du sens, il faut qu'elle soit exprimée dans cet ordre spécifique. Les réseaux récurrents ont donc besoin de justifier la position de chaque mot dans l'expression et ils utilisent ces informations pour prévoir le prochain mot dans la séquence.

Une autre particularité des réseaux récurrents est qu'ils partagent des paramètres à chaque couche du réseau. Alors que les réseaux de propagation vers l'avant ont des poids différents à chaque nœud, les réseaux de neurones récurrents partagent les mêmes paramètres de poids dans chaque couche du réseau. Autrement dit, ces poids sont toujours ajustés via des processus de rétropropagation et de descente de gradient pour faciliter l'apprentissage par renforcement.[33]

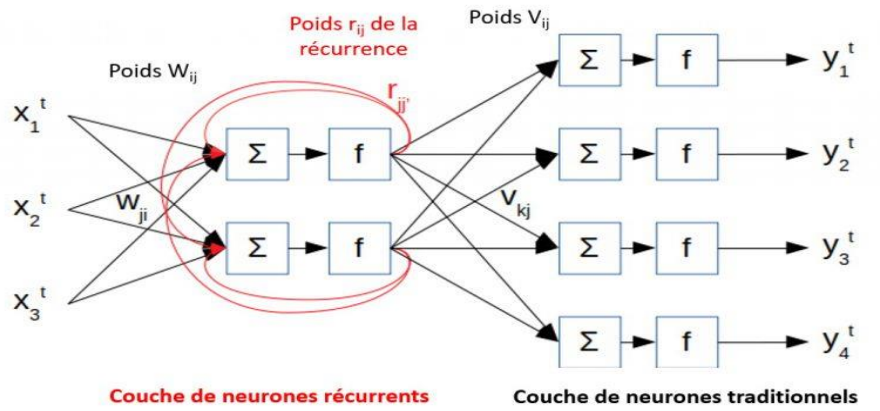


Figure 3.10:Couche de neurones récurrents devant une couche de neuronestraditionnels

2.1 Les types de réseaux de neurones récurrents :

Les types de réseaux de neurones récurrents (RNN) comprennent plusieurs variantes qui ont été développées pour améliorer les capacités des RNN traditionnels.

Pour traiter ce type de données, il existe trois grands types de réseaux de neurones récurrents :le RNN simple, le LSTM et le GRU.

- Le RNN simple est la forme la plus basique de RNN, ne possédant pas de portes pour contrôler le flux d'informations. Cependant, en pratique, les RNN simples ne sont généralement pas utilisés en raison de leurs limitations.
- Les LSTM sont une architecture populaire de RNN introduite pour résoudre le problème de la disparition du gradient. Les LSTM sont une variante avancée de RNN qui surmontent les limitations des RNN simples en introduisant des mécanismes de mémoire à court et long terme. Les LSTM utilisent des portes pour contrôler le flux d'informations et sont efficaces pour gérer des dépendances à long terme.
- Les GRU sont une autre variante de RNN qui simplifient l'architecture des LSTM en combinant les portes d'oubli et d'entrée en une seule porte d'update. Les GRU sont efficaces pour des tâches similaires aux LSTM mais avec une architecture plus simple.

Comme pour les réseaux de neurones traditionnels, les réseaux de neurones récurrents peuvent contenir plusieurs couches, ce qui leur permet de capturer davantage de non-linéarité parmi les données, mais augmente également le temps de calcul en phase d'apprentissage. On peut également combiner des couches récurrentes avec des couches classiques, telles que des couches denses (MLP) ou des couches de convolution (CNN).[34]

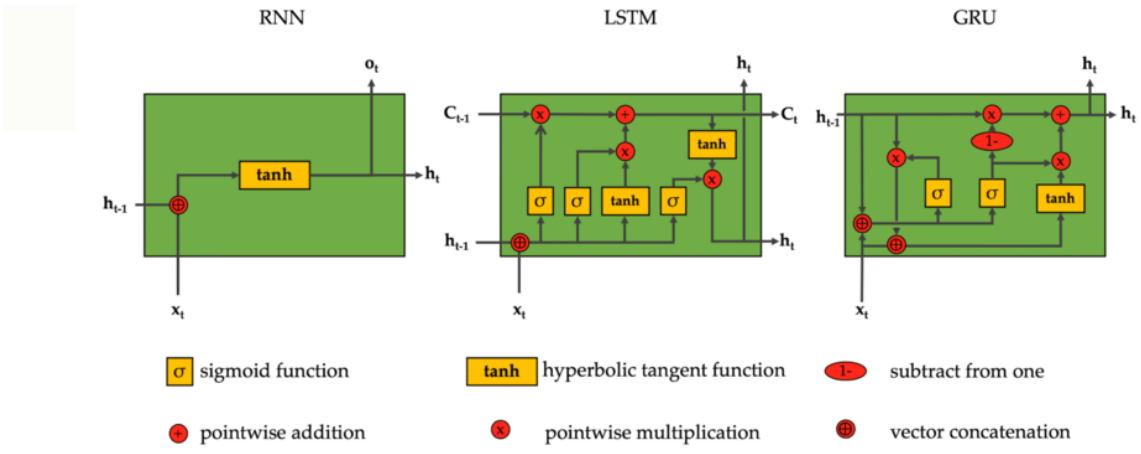


Figure 3.11 : Les types de réseaux de neurones récurrents

2.2 Architecture de RNN:

L'architecture d'un réseau neuronal récurrent (RNN) se compose généralement de plusieurs couches répétitives qui sont empilées les unes sur les autres. Comme mentionné précédemment, chaque couche récurrente peut être un simple RNN, LSTM ou un GRU.

L'architecture d'un RNN est constituée de :

2.2.1 Entrée (Input): Les données séquentielles sont introduites dans le réseau par l'intermédiaire de l'entrée. Chaque séquence de données est représentée par une série d'éléments (par exemple, des mots dans une phrase ou des instants temporels dans une série temporelle).

2.2.2 Couche récurrente (Récurrent Layer): Les informations séquentielles sont traitées dans la couche récurrente. Cette couche possède des connexions de rétroaction qui permettent aux informations de circuler d'une étape à l'autre dans la séquence. Cela permet au réseau de capturer les dépendances temporelles et de modéliser les relations complexes entre les éléments séquentiels.

2.2.3 Optionnel : Stacking de couches récurrentes (Stacking Récurrent Layers) : Il est possible d'empiler plusieurs couches récurrentes les unes sur les autres pour former un réseau de neurones récurrents profond. Chaque couche récurrente traite les informations provenant de la couche précédente, permettant ainsi au réseau de capturer des niveaux de représentation plus abstraits et complexes.

2.2.4 Couche de sortie (Output Layer): La couche de sortie génère les prédictions ou les sorties souhaitées en fonction des informations traitées par les couches récurrentes. La nature de la tâche détermine le type de couche de sortie utilisée. Par exemple, pour la classification, une couche de sortie dense avec une fonction d'activation appropriée peut être utilisée**.**[35]

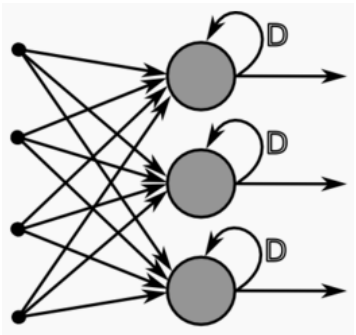


Figure 3.12: Recurrent Neural Network (RNN) Tutorial

2.3 Fonctionnement des RNN:

Les réseaux de neurones récurrents (RNN) fonctionnent sur deux principes clés selon les sources fournies :

1. **Astuce de la fenêtre glissante et Connexions Récurrentes:**

Les RNN reposent sur l'astuce de la fenêtre glissante pour traiter des signaux de taille variable. Ils utilisent des connexions récurrentes qui permettent d'analyser la partie passée du signal, offrant au réseau la capacité de "voir" la fenêtre correspondante à un instant donné et de se "souvenir" de sa décision à un instant précédent. Ces connexions récurrentes permettent au réseau de neurones de conserver une mémoire interne et de prendre en compte les dépendances temporelles dans les données.

2. **Modélisation des Dépendances Temporelles:**

   Les RNN sont conçus pour traiter des séquences de taille variable tout en modélisant les dépendances au sein de la séquence d'entrée. Ils peuvent être approximés par des réseaux non récurrents dépliés dans le temps, ce qui permet de visualiser leur fonctionnement de manière plus claire. Les RNN peuvent être utilisés pour des tâches telles que l'étiquetage de séquences, la classification de séquences et la génération de séquences, comme la prédiction de mots suivants dans un texte ou la classification de sentiments dans des avis en ligne.[35]

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

Un LSTM est un bloc fonctionnel spécial de réseaux neuronaux récurrents (RNN) doté d'une mémoire à court terme à long terme. Il s'agit d'une évolution des RNN et permet de résoudre le problème des gradients disparaissant, dans lequel les gradients de poids diminuent progressivement pendant l'entraînement et le réseau ne stocke donc plus d'informations utiles. Les cellules LSTM ont trois types de portes - une porte d'entrée, une porte de mémorisation et d'oubli et une porte de sortie.

Pour stocker les souvenirs des expériences passées. La mémoire à court terme est conservée longtemps et le comportement du réseau est encodé dans les poids. Les réseaux LSTM sont particulièrement adaptés pour faire des prédictions basées sur des données de séries temporelles, comme par exemple pour la reconnaissance de textes manuscrits et la reconnaissance vocale.[36]

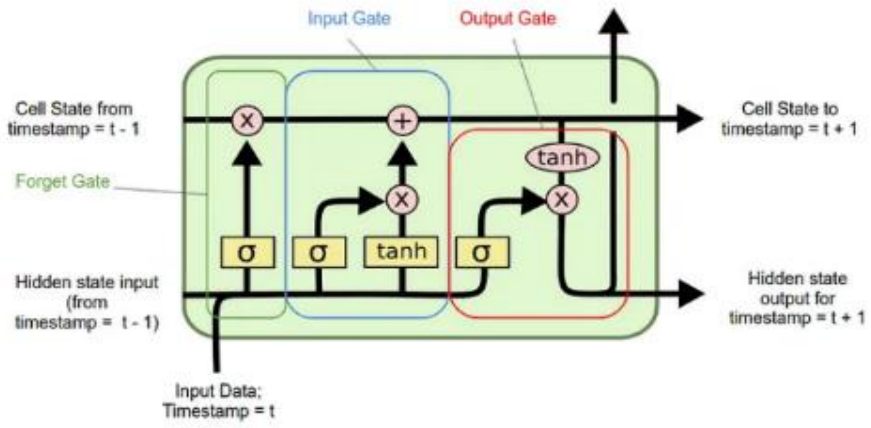


Figure 3.13 : Architecture de LSTM

Les portes d'un LSTM sont analogiques, sous la forme de sigmoïdes, ce qui signifie qu'elles vont de 0 à 1. Le fait qu'elles soient analogiques leur permet de faire une rétropropagation avec elles.

Les problèmes posés par la disparition des gradients sont résolus grâce à LSTM, car les gradients sont assez raides et l'entraînement est relativement court et la précision élevée.[36]

2.4.1 Fonctionnement de LSTM:

La cellule mémoire d'un LSTM est composée de plusieurs portes : une porte d'entrée, une porte de sortie et une porte d'oubli. Ces portes régulent le flux d'informations à l'intérieur de la cellule mémoire, permettant ainsi de contrôler les informations à retenir et celles à oublier. Cela donne aux LSTM la capacité de mémoriser des informations importantes sur de longues séquences et d'ignorer les éléments moins pertinents. h-t est l'état caché habituel des RNN mais dans les réseaux LSTM nous ajoutons un deuxième état appelé C_t. Ici, $h_t$représente la mémoire courte du neurone et $C_t$représente la mémoire à long terme.[36]

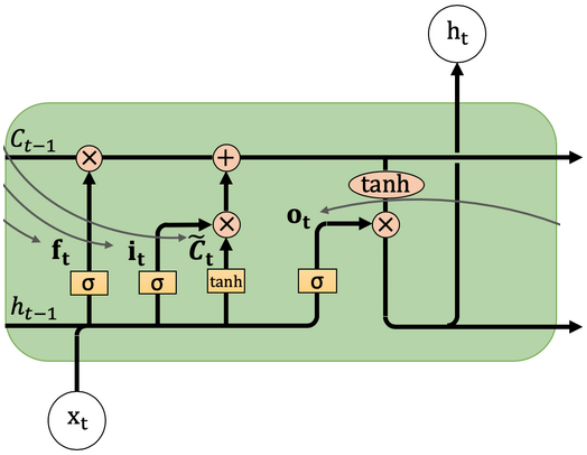


Figure 3.14 : Bloc de mémoire LSTM

La Porte entréei_t (3.2) détermine quelles information doivent être mises à jour dans la cellule mémoire. Elle prend en compte l entréeactuelx_t ainsi que l'état précédent de la cellule mémoire h_{t-1} et génère un vecteur d'activation qui représente les informations à ajouter à la cellule $\hat{C}_{t-1}$Cet ajout d'informations se traduit par une opération mathématique effectuée entre ce vecteur d'activation et l'état précédent $\hat{C}_{t-1}$(3.3)

$$i_t = \sigma(w_i.[h_{t-1}, x_t] + b_i) \quad (3.2)$$

$$\hat{C}_t = tanh(w_c, [h_{t-1}, x_t]) + b_i \quad (3.3)$$

La porte d'oubli f_t (3.4) permet au LSTM de supprimer les informations obsolètes de la cellule mémoire. Elle utilise à la fois l'entrée actuelle et l'état précédent pour générer un vecteur d'activation qui détermine quelles informations doivent être oubliées. Cela se traduit également par une opération mathématique effectuée entre ce deuxième vecteur d'activation et l'état précédent $\hat{C}_{t-1}$.C'est à partir des deux opérations precedents sur $\hat{C}_{t-1}$que l'on obtient l'état actual c_t (3.5).

$$f_t = \sigma\left(w_f.[h_{t-1}, x_t] + b_f\right) \quad (3.4)$$

$$c_t = f_t.C_{t-1} + i_t.\hat{C}_t \quad (3.5)$$

Enfin, la porte de sortie (3) (3.6) détermine la sortie du LSTM à un instant donné. Elle utilise l'entrée actuellex_t et l'état actuel de la cellule mémoire pour générer c_t un vecteur d'activation qui représente la sortie du LSTM. C'est ainsi que l'on obtient h_t(3.7) .

$$o_t = \sigma(w_o.[h_{t-1}, x_t] + b_o) \quad (3.6)$$

$$h_t = o_t.tanh(c_t) \quad (3.7)$$

où .et+ représentent la multiplication et l'addition point par point et b_i, b_f, et b_o représentent les Bais.

La combinaison de ces trois portes permet au réseau LSTM de gérer efficacement les dépendances à long terme. Lors de la rétropropagation du gradient, les LSTM peuvent maintenir un flux d'informations constant à travers le temps, évitant ainsi le problème du « vanishing gradient » et permettant un apprentissage plus stable et plus précis.

2.5 GatedRecurrent Unit (GRU):

L'unité récurrente fermée GRU (GatedRecurrent Unit) a été introduite en 2014 par Cho et Al [37] pour résoudre le problème de disparition du gradient rencontré par les réseaux récurrents classiques mais aussi pour proposer une architecture avec moins de paramètres à entrainer par rapport à une LSTM. À l'instar de LSTM, l'unité GRU est l'élément de base d'une architecture GRU. Une passe avant de l'unité GRU est modélisé par les équations :

$$z_t = \sigma(W_z.x_t + U_Z.H_{t-1} + b_z) \quad \text{.....(3.8)}$$

$$r_t = \sigma(W_r.x_t + U_R.H_{t-1} + b_r) \quad \text{.....(3.9)}$$

$$H_t = tanh(W_H.x_t + U_H(r_t.H_{t-1}) + b_h) \quad \text{.....(3.10)}$$

$$H_t = (1 - z_t).H_{t-1} + z_t.H_t \quad \text{.....(3.11)}$$

où σ est la fonction sigmoïde, z_t est le vecteur d'activation de la porte de mise à jour, $r_t$le vecteur d'activation de la porte de réinitialisation, het le vecteur candidat et $h_t$est le vecteur output de l'unité GRU. W et U sont des poids, b est le vecteur biais (poids et biais sont à entraîner durant le processus d'apprentissage) et le symbole * pour le produit de Hadamard.[37]

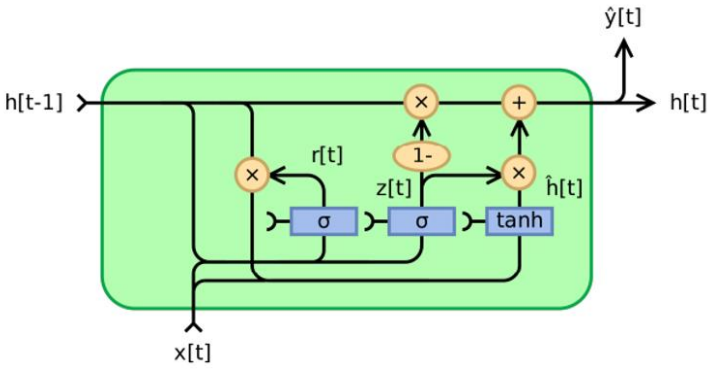


Figure 3.15 : Unité de base GRU.

2.6 LSTM bidirectionnel (BiLSTM):

Le BiLSTM est un modèle de réseau de neurones récurrent(RNN) qui consiste en deux couches LSTM : l'une traitant l'entrée dans le sens normal et l'autre dans le sens inverse. Cette approche permet au modèle d'avoir accès à l'information à la fois du passé et du futur pour chaque élément de la séquence, améliorant ainsi le contexte disponible pour l'algorithme. En traitant les données dans les deux directions, le BiLSTM est capable de mieux comprendre les relations entre les séquences, par exemple en connaissant les mots qui précèdent et suivent un mot dans une phrase.[38]

L'architecture du LSTM bidirectionnel comprend deux LSTM unidirectionnels qui traitent la séquence dans les sens avant et arrière.

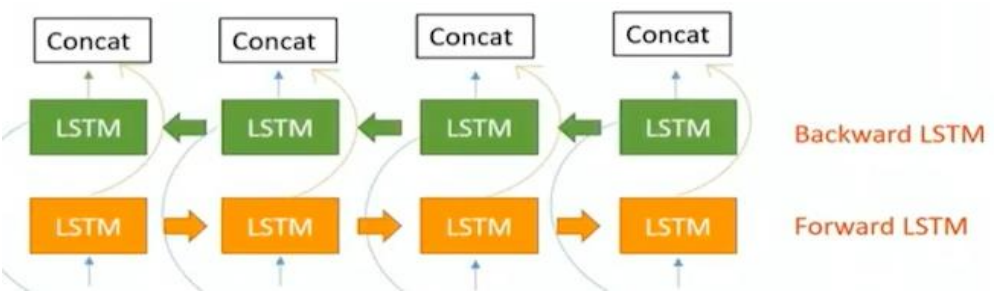


Figure 3.16 : Architecture de couche LSTM bidirectionnelle

1. **LSTM avant :**

$\overrightarrow{h_t} = Forward\ LSTM(x_t, h_{t-1})$……(3.12)

2. **LSTM arrière :**

$\overleftarrow{h_t} = Backward\ LSTM(x_t, h_{t+1})$……(3.13)

3. **Combinaison des états cachés :**

$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$……(3.14)

[·;·] représente la concaténation des deux états cachés (avant et arrière).

En combinant les informations des deux directions, le Bi-LSTM est capable de mieux comprendre le contexte global de la séquence d'entrée, ce qui le rend particulièrement efficace pour des tâches de traitement du langage naturel et autres applications séquentielles.[38]

3.7 Perceptron multicouches (MLP) :

Les réseaux de neurones sont construits en couches. Chaque couche est composée de neurones et chaque neurone possède des poids d'entrée. Les neurones sont représentés par des nœuds et les poids par des arcs. Un réseau de neurones est toujours composé au minimum d'une couche d'entrée puis d'une ou plusieurs couche(s) de neurones appelée(s) couche(s) cachée(s) et enfin d'une couche de sortie. Il n'y a pas d'architecture type d'un réseau de neurones en termes de nombres de couches et de neurones car celle-ci dépend de la tâche à résoudre, des données disponibles et des contraintes d'utilisation (espace mémoire, temps). L'architecture d'un réseau de neurones est un méta-paramètre.[39]

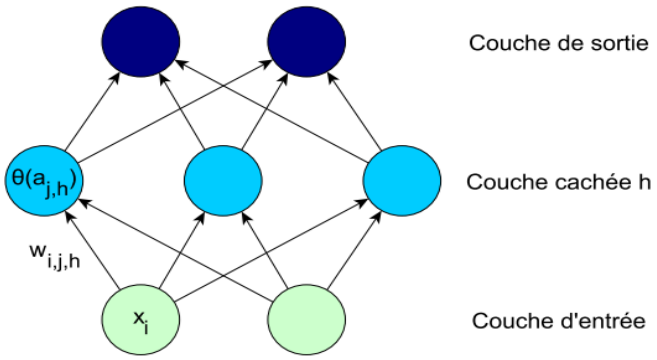


Figure 3.17 : Perceptron multi-couches

3.7.2 Architecture du MLP :

Le perceptron multicouche (PMC), également connu sous le nom de Multi-Layered Perceptron (MLP), est un type de réseau de neurones composé de plusieurs couches, dont au moins une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie.

Voici les différente coches d'unMLP:

La couche d'entrée :N'est pas une couche apprise et ne possède donc pas de neurones à proprement parler. Les valeurs de la couche d'entrée sont directement celles du vecteur de données d'entrée X et c'est pourquoi leur nombre est toujours égal au nombre d'éléments du vecteur d'entrée. Il est donc nécessaire pour utiliser un MLP d'avoir des vecteurs d'entrée de taille fixe. La couche d'entrée est toujours la première couche ou la couche la plus basse d'un réseau de neurones (h = 0) et elle est suivie de couches cachées (h).

Lescouches cachées : sont les couches contenant des neurones ayant des poids. Les poids des neurones des couches cachées sont les paramètres d'un réseau de neurones qu'il faut optimiser par l'apprentissage. Les valeurs de sortie ou valeurs d'activation des neurones sont calculées par produit scalaire des vecteurs de poids de chaque neurone et du vecteur d'entrée qui est le vecteur des valeurs de sortie des neurones de la couche cachée précédente $(h-1)$. Un neurone possède un biais qui est un poids dont l'entrée vaut toujours 1 et qui permet de modifier le seuil d'activation du neurone. Avant de définir l'opération élémentaire d'un neurone d'une couche cachée, nous définissons les termes qui vont suivre. $w_{i,j,h}$est le poids entre le neurone j de la couche cachée h et le neurone i de la couche précédente $h-1$. $i$et$j$ sont respectivement le nombre de neurones de la couche $h-1$et $x_i$est la valeur du neurone i de la couche précédente. $\beta_{j,h}$est le biais du neurone j de la couche cachée h.

Le calcul de la valeur d'activation $a_{j,h}$ du $j^{ème}$neurone de la couche cachée est présenté dans l'équation 3.15.

$a_{j,h} = \sum_{i=1}^{I} w_{i,j,h} x_i + \beta_{j,h}$ ……(3.15)

L'étape suivante consiste à calculer (3.16) la fonction d'activation des neurones.

$b_{j,h} = \theta_h(a_{j,h})$……...(3.16)

La fonction θ_h est généralement une tangente hyperbolique :

$\tanh(x) = \frac{e^{2x-1}}{e^{2x+1}}$……(3.17)

Ou une sigmoïde logistique :

$\sigma(x) = \frac{1}{1+e^{-x}}$…… (3.18)

La couche de sortie : Ces neurones représentent les valeurs attendues afin de résoudre la tâche comme par exemple des probabilités de classes pour une tâche de classification ou des valeurs réelles dans le cadre d'une régression. Les neurones de la couche de sortie se calculent de la même manière que ceux d'une couche cachée sauf qu'on n'y applique généralement pas la même fonction d'activation. Celle-ci dépend de la tâche que l'on cherche à réaliser : classification ou régression. Pour une tâche de classification, une fonction d'activation softmax est généralement utilisée. Le nombre de neurones de sorties est égal aux nombres de classes. L'équation de la fonction Softmax est définie par l'équation (3.19)oùc_k est la k ième classe, y_k la probabilité de sortie de la k ième classe, x_k l'activation du k ième neurone.

$p(c_k|X) = y_k = \frac{e^{x_k}}{\sum_{l=1}^{K} e^{x_l}}$ $pour\ l = 1 \dots K$ ……(3.19)

Dans le cadre d'une tâche de régression, la sortie correspond aux valeurs à prédire, comme par exemple des coordonnées dans une image. Il n'est alors pas nécessaire d'utiliser une fonction d'activation. Pour la régression, le nombre de sorties est égal au nombre de valeurs à estimer.[39]

3.7.3 Apprentissage supervisé d'un réseau de neurones :

Pour l'apprentissage supervisé d'un réseau de neurones, la difficulté est de trouver les bons poids permettant de résoudre la tâche. Il faut donc apprendre les poids du réseau de telle sorte que la sortie Y d'un réseau auquel on a présenté la donnée X donne la sortie Z attendue ou vérité terrain. C'est un problème d'optimisation où l'on cherche à faire correspondre au mieux la sortie Y à la sortie attendue Z. Par exemple, pour un problème de classification d'images de chiffres, on utiliserait un réseau à 10 sorties représentant les 10 chiffres. Pour apprendre ce réseau il faut des images de chiffres (X) et leur étiquette associée (Z) qui est un chiffre de 0 à 9. Z est alors un vecteur de taille 10 dont toutes les valeurs sont à 0 sauf celle

du chiffre attendu qui est à 1. La fonction de coût à optimiser est choisie pour rendre compte de l'aptitude du réseau à prendre les bonnes décisions.

3.7.4 Rétropropagation du gradient de l'erreur :

La rétropropagation du gradient, est une méthode fondée sur la descente de gradient. La descente de gradient est une méthode d'optimisation itérative permettant de trouver un minimum local d'une fonction. Cette méthode calcule le gradient (direction de la pente) en un point de la fonction à optimiser et modifier les paramètres θ de la fonction de façon à diminuer l'erreur. La modification des poids doit s'opposer à l'augmentation de l'erreur, elle est donc de sens opposé au gradient. Puis, on recommence à partir du nouveau point jusqu'à atteindre le minimum local. Pour une fonction E paramétrée par θ, la descente de gradient ayant un pas de descente λ est donnée par l'équation 3.21 :

$$\theta = \theta - \lambda \frac{\partial E}{\partial \theta} \quad (3.21)$$

Dans le cadre de l'apprentissage supervisé d'un réseau de neurones, seule l'évaluation du critère de la couche de sortie est possible. Pour un réseau de neurones, le problème est d'évaluer le critère d'optimisation pour les couches cachées. La descente de gradient à travers les couches d'un réseau de neurones est possible grâce à la règle de dérivation en chaîne. La règle de dérivation en chaîne permet de transmettre le coût à travers les couches cachées. La fonction de coût *E*(y, z) étant dérivable, on en déduit la dérivée partielle de l'erreur E par rapport à la sortie y

$$\frac{\partial E}{\partial Y} \quad (3.22)$$

Grâce à la règle de dérivation en chaîne, le calcul de la dérivée de l'erreur par rapport aux poids de la couche de sortie n s'exprime alors par l'équation 3.23 :

$$\frac{\partial E}{\partial W_{i,n}} = \frac{\partial E}{\partial Y} \frac{\partial Y}{\partial W_{i,n}} \quad (3.23)$$

Il n'est pas possible de calculer directement $\frac{\partial E}{\partial w_{i,n-1}}$, la couche $h = n - 1$n'étant pas directement reliée à la sortie. Il faut alors appliquer de manière récursive la règle de dérivation en chaîne afin de rétro propager l'erreur pouvant alors être rétro propagée jusqu'à la couche d'entrée:

$$\frac{\partial W}{\partial W_{i,0}} = \frac{\partial E}{\partial Y} \frac{\partial Y}{\partial w_{i,n}} \frac{\partial w_{i,n}}{\partial w_{i,n-1}} \dots \frac{\partial w_{i,1}}{\partial w_{i,0}} \quad (3.24)$$

Grâce à la règle de dérivation en chaîne, il est donc possible de transmettre le gradient de l'erreur aux poids de la couche de sortie, puis de le transférer aux couches cachées. C'est ce

processus que l'on appelle la rétropropagation du gradient. Le gradient de l'erreur est donc diffusé à partir de la couche de sortie à travers toutes les couches cachées grâce à la règle de dérivation en chaîne. Cette règle permet de calculer le gradient à soustraire à un poids afin de le mettre à jour. Le calcul du gradient des poids $w_{i,h}$ du neurone i de la couche h s'exprime alors comme suit :

$$\delta_{i,h} = \frac{\partial E}{\partial w_{i,h}} = \frac{\partial E}{\partial Y}\frac{\partial Y}{\partial w_{i,h}} \quad (3.25)$$

On itère alors la couche de sortie h = n jusqu'à la couche d'entrée h = 0 en suivant la règle de dérivation en chaîne afin de pouvoir rétro propager l'erreur sur l'ensemble du réseau. La rétropropagation du gradient ne peut donc se faire que séquentiellement en partant de la couche de sortie jusqu'à la couche d'entrée.

La mise à jour du gradient dans la rétropropagation est contrôlée par un taux d'apprentissage λ qui régule la descente de gradient. Le taux d'apprentissage est un pas de descente qui pourrait être calculé exactement par un algorithme d'ordre 2 (méthode de Newton). Cependant, c'est rarement le cas en pratique car le coût est en O(n^4) contre O(n^2) pour une rétropropagation sans calcul du pas de descente à chaque exemple, où n est le nombre de paramètres du réseau. La mise à jour d'un poids s'exprime alors en fonction du taux λ par :

$$w_{i,h} = w_{i,h} - \lambda\frac{\partial E}{\partial w_{i,h}} = w_{i,h} - \lambda\delta_{i,h} \quad (3.26)$$

Plus ce taux est grand, plus la descente est rapide ; plus il est petit, plus la descente est lente. Un taux d'apprentissage trop élevé ne permet généralement pas de converger vers l'optimum global. La direction du gradient induira un changement régulier de pentes de descente permettant de sortir des minimums locaux mais ne permettant pas d'y descendre. Un taux d'apprentissage trop faible conduit en général à une convergence lente et qui peut être bloquée dans un minimum local.

On peut stopper la descente de gradient de plusieurs manières. Une condition d'arrêt facilement évaluable est de déterminer un nombre d'itérations, c'est-à-dire le nombre de fois où toutes les données d'apprentissage ont été propagées en avant et rétro propagées.Pour contrôler ce sur-apprentissage une des techniques est d'utiliser une base de validation qui permet d'évaluer la valeur du critère pour des données inconnues. Une condition d'arrêt fréquemment utilisée consiste à arrêter lorsque la valeur du critère sur la validation augmente ou n'évolue plus.

Afin d'apprendre un réseau de neurones, il faut appliquer la descente de gradient pour tous les exemples X et renouveler l'opération jusqu'à atteindre la condition d'arrêt. La descente de

gradient peut nécessiter un grand nombre de calculs qui croit exponentiellement avec le nombre de neurones lorsqu'on calcule le pas de descente de gradient (complexité en O(n^4)). Il faut donc utiliser une méthode dont les temps de calculs permettent un apprentissage dans un temps réaliste. Pour y parvenir, l'algorithme de descente de gradient le plus efficace et rapide est la descente de gradient stochastique [Bottou, 1998, Bottou, 2012]. Nous détaillons cet algorithme ci-après dans l'algorithme 1. La descente de gradient utilise un pas fixe non optimal permettant des calculs de gradients plus rapides en O(n^2) et mélange les données ce qui permet une bonne convergence du réseau vers l'optimum. [40]

Algorithme 1: Algorithme de descente de gradient stochastique.

Input : network, X, Z, λ
Output: network
// Initialisation du réseau
1 network.initializeWeights() ;
2 **while** network has not converged **do**
3 X shuffled, Z shuffled = shuffle(X,Z);
4 **foreach** x, z in Xshuffled,Zshuffled **do**
5 y ← network.compute(x) ;
6 e ← E(y, z) ;
7 **foreach**$W_{i,j}$ in network **do**
8 $W_{i,j} = W_{i,j} - \lambda \frac{\partial E}{\partial W_{i,j}}$;
9 TestingNetworkConvergence();
10 **return** network

L'un des inconvénients les plus importants de la SGD est le taux d'apprentissage fixe. En effet, s'il est trop petit l'apprentissage est piégé dans un minimum local et s'il est trop grand l'apprentissage ne convergera pas. Généralement, il est intéressant de commencer par un taux d'apprentissage plus grand pour finir par un taux plus petit. Il existe des améliorations ou variantes à la descente du gradient stochastique.

Une technique consiste à introduire un moment (momentum) [Qian, 1999] dans le calcul du gradient lors de la descente. Ce moment garde l'inertie en ajoutant les informations des changements précédents lors de la descente de gradient, c'est-à-dire qu'à l'exemple t, on prend en compte le gradient de l'exemple précédent t − 1. Cette méthode permet de sortir des minimums locaux et d'accélérer l'optimisation du réseau. La mise à jour du gradient avec un moment η se traduit par l'équation 3.22.

$$w_{i,h}^{t} = w_{i,h}^{t-1} - \lambda\left(\frac{\partial E}{\partial w_{i,h}^{t}} + \eta\frac{\partial E}{\partial w_{i,h}^{t-1}}\right) \quad (3.27)$$

Une autre méthode AdaGrad [Duchi et al., 2011] utilise un algorithme de gradient adaptatif. Cette méthode permet de diminuer le taux d'apprentissage des poids qui ont souvent été mis à jour et d'augmenter celui de ceux qui ne l'ont pas été. Pour ce faire, le carré des gradients G est conservé et on obtient l'équation de G :

$$G_{i,h}^{t} = \sum_{\pi=1}^{t}\left(\delta_{i,h}^{t}\right)^{2} \quad (3.28)$$

Et la mise à jour des poids se fait alors comme suit :

$$W_{i,h}^{t+1} = w_{i,h}^{t} - \frac{\lambda}{\sqrt{G_{i,h}^{t}}}\frac{\partial E}{\partial W_{i,h}^{t}} \quad (3.29)$$

AdaDelta [Zeiler, 2012] et RMSProp [Tieleman and Hinton, 2012] sont également des méthodes permettant d'adapter le gradient en fonction de chaque paramètre. RMS Prop est identique à Adelta mais avec γ = 0.9. L'idée est de diviser le taux d'apprentissage par une moyenne E des magnitudes des valeurs de gradients récentes d'équation :

$$E_{i,h}^{t} = \gamma E_{i,h}^{t-1} + (1-\gamma)\left(\delta_{i,h}^{t}\right)^{2} \quad (3.30)$$

où γ est un facteur d'oubli. La mise à jour des poids se fait alors suivant l'équation :

$$W_{i,h}^{t+1} = w_{i,h}^{t} - \frac{\lambda}{\sqrt{E_{i,h}^{t}}}\frac{\partial E}{\partial W_{i,h}^{t}} \quad (3.31)$$

Il existe d'autres méthodes comme Adam [Kingma and Ba, 2014] Il s'agit d'une extension de la descente de gradient stochastique qui utilise les moyennes courantes des gradients et des seconds moments des gradients pour calculer des taux d'apprentissage adaptatifs pour chaque paramètre du modèle Adam combine les avantages des algorithmes AdaGrad et RMSProp, offrant ainsi une méthode efficace pour ajuster les paramètres du modèle et améliorer sa convergence[40].

Son itération correspond au schéma générique de (3.28) avec

$$m_k = \frac{(1-\beta 1)\sum_{j=0}^{K}\beta_1^{K-j} g_j}{\left(1-\beta_1^{K+1}\right)} \quad (3.32)$$

et

$$v_k = \sqrt{\frac{(1-\beta 2)\sum_{j=0}^{K}\beta_2^{K-j} g_j \odot g_j}{\left(1-\beta_2^{k+1}\right)}} \quad (3.33)$$

m_k : Est la moyenne mobile exponentielle des gradients.

v_k: Est la moyenne mobile exponentielle des carrés des gradients.

g_j : Est le gradient à l'itération.

Ces deux équations permettent de calculer les moments du premier et second ordre des gradients, qui sont ensuite utilisés pour mettre à jour les paramètres du modèle de manière adaptative, améliorant ainsi la convergence de l'algorithme d'optimisation.

4 CONCLUSION:

Nous avons consacré ce chapitre à l'introduction du concept de réseaux neuronaux convolutifs et à leur rôle dans la reconnaissance d'images. Nous avons également abordé les méthodes de classification et les types de couches des réseaux neuronaux, ainsi que l'architecture générale des réseaux neuronaux convolutifs.

Nous allons détailler le fonctionnement des réseaux de neurone et en particulier le PMC.

Dans le dernier chapitre, nous explorons la diversité des architectures de RNN (LSTM, BLSTM, GRU) et l'application d'un réseau de neurones récurrent à un cas concret celui de la base MNIST d'image de chiffres Manuscrit afin de les reconnaître et de les classer.

Chapitre IV
Implémentation

1-Introduction :

Dans ce chapitre nous explorons la diversité des architectures de RNN (LSTM, BLSTM, GRU) disponibles soulève la question de savoir quelles sont les plus efficaces pour la classification des chiffres manuscrits. Cette comparaison revêt une grande importance car elle permet d'identifier les modèles RNN les plus performants et de guider le choix des architectures appropriées pour cette tâche critique.

2 Description de la base de données MNIST :

La base de données MNIST (Modified National Institute of Standards and Technology Database) est l'une des bases de données les plus utilisées en apprentissage automatique et en vision par ordinateur. Elle est utilisée pour entraîner et tester des algorithmes de reconnaissance de chiffres manuscrits.

La base de données MNIST que nous avons utilisé contient un ensemble de 5 000 images d'apprentissage et un ensemble de test de 5 000 images, de diffèrent chiffre de 0 à 9. Chaque image dans la base de données MNIST est en niveaux de gris, avec une résolution de 28x28 pixels. Cette basse est organier sous forme de 10 lot contenant 1000 images représentant différente image d'un seul chiffre dont voici (Figure 3.1) quelque exemple d'image appartenant à notre base de données :

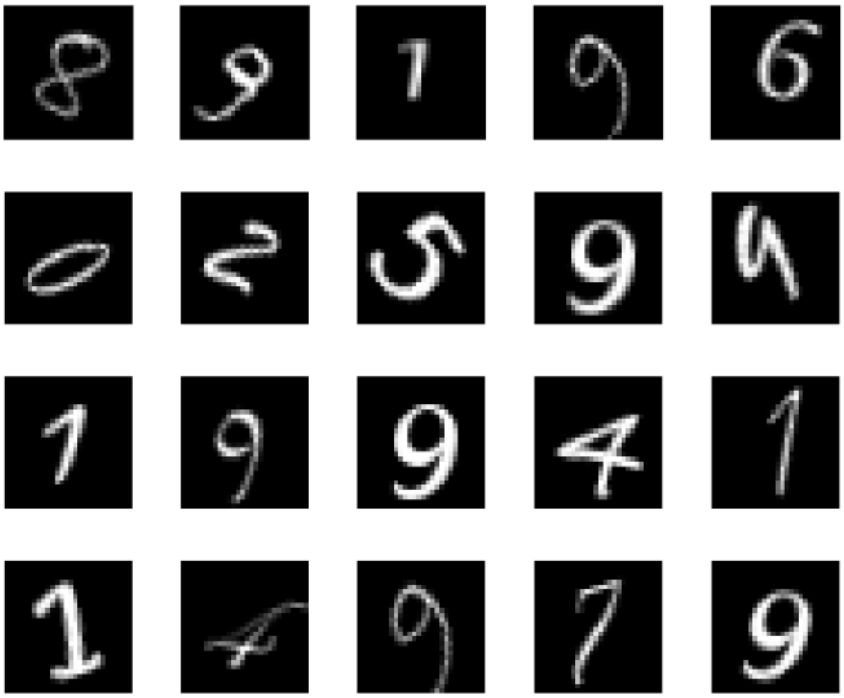


Figure 4.1 : Exemple d'image de chiffres appartenant à notre base de données

3 Description du logiciel utilisé :

Nous utilisons le logiciel MATLAB qui est un langage de programmation de quatrième génération et un environnement d'analyse numérique. MATLAB permet de faire du calcul matriciel, de développer et d'exécuter des algorithmes, de créer des interfaces utilisateur (IU) et de visualiser des données. Il est utilisé dans de nombreux domaines, tels que le traitement des images et des signaux, les communications, les systèmes de contrôle du secteur industriel, la conception de réseaux intelligents, la robotique et la finance computationnelle et la classification. [45]

4 Schéma général d'apprentissage :

L'organigramme figure (4.2) ci-dessous montre les étapes principales de l'algorithme d'apprentissage et de test pour les trois modèles.

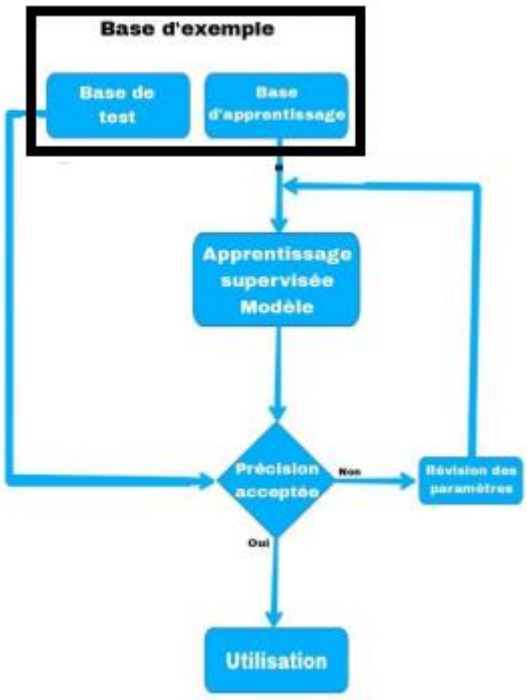


Figure 4.2 :Schéma général d'apprentissage

5 Optimisateur et paramètres utilisés pour l'apprentissage des trois RNN:

Nous utilisant la méthode stochastique de descente de gradient basée sur l'estimation adaptative des valeurs de premier ordre d'adam (Adaptative Moment Estimation) pour une taille de mini lot (MiniBatchSize) de 128, un taux d'apprentissage (InitialLearnRate) de 0.001, et un Seuil de gradient (GradientThreshold) de valeur 5, pour une durée de1950 = (39 mini lot *50 époques) itérations pour l'apprentissage des trois réseaux RNN

6 Architecture des trois RNN utilisées :

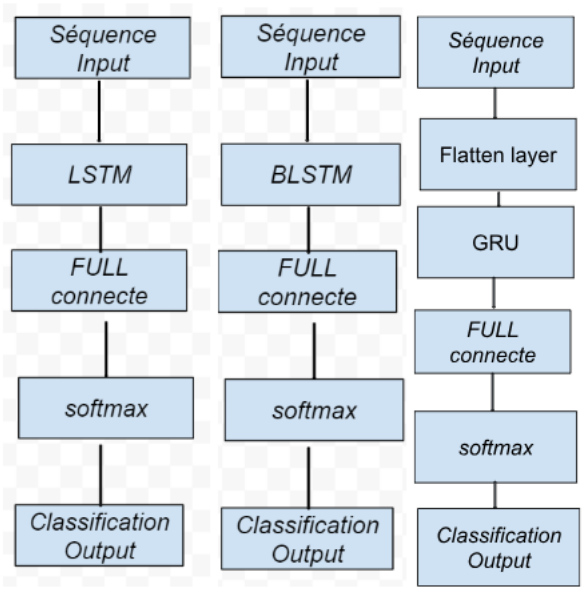


Figure 4.3 :Architecture des trois RNN (LSTM, BLSTM, GRU)

7 Méthodologie de Comparaison

Nous utilisons la base de données MNIST pour évaluer les performances des différents modèles RNN (LSTM, BLSTM, GRU). Nous mesurons la précision de classification sur un ensemble de tests distincts en termes de précisions, de recall , de précision global OA, de précisions moyenne AA, et de coefficient Kappa . Nous évaluons également les performances en termes de vitesse de convergence et de complexité du modèle.**7.1 Calcul de précision :**

7.1.1 Rappel de la définition de la précision :

La **précision** pour une classe *i* est la proportion des éléments correctement classés comme appartenant à la classe *i* par rapport à tous les éléments que le modèle a classés comme appartenant à cette classe *i*.

Précision	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6	Classe 7	Classe 8	Classe 9	Classe 10
LSTM	99.21	99.00	**99.00**	**98.80**	98.41	98.61	**99.59**	99.40	**99.40**	98.81
BLSTM	**99.80**	**99.39**	98.61	97.64	**99.80**	**99.40**	99.40	**99.80**	98.61	**99.80**
GRU	98.61	99.19	98.38	95.35	98.81	97.79	98.58	98.80	98.80	98.21

Tableau 4.1 : La précision pour une classe *i*

D'après les données du tableau (4.1), nous pouvons tirer les conclusions suivantes concernant la performance des trois modèles de réseaux neuronaux (LSTM, BLSTM et GRU) en termes de précision pour chaque classe .

7.1.2 Observations générales

1. LSTM : Le modèle LSTM présente une très haute précision pour toutes les classes, avec des valeurs oscillant principalement entre 98.41% et 99.59%. Les classes 7 et 8 sont particulièrement bien classées, avec des précisions respectives de 99.59% et 99.40%.
2. BLSTM : Le modèle BLSTM montre la meilleure performance globale parmi les trois modèles, avec des précisions allant de 97.64% à 99.80%. Il se distingue surtout dans les classes 1, 5, 8 et 10, où il atteint une précision de 99.80%.
3. GRU : Le modèle GRU a des performances légèrement inférieures par rapport aux deux autres modèles, avec des précisions variant de 95.35% à 99.19%. La classe 4 présente la plus basse précision de 95.35%, ce qui indique une certaine difficulté de ce modèle à bien classifier cette classe par rapport aux autres.

7.1.3 Comparaison entre les 3 modèles en termes de précision :

1. **Performance globale** : Le modèle BLSTM semble être le plus performant avec des précisions élevées dans presque toutes les classes. Il surpasse les modèles LSTM et GRU dans plusieurs classes, notamment les classes 1, 2, 5, 8 et 10, où il atteint ou dépasse 99.80%.
2. **Classes spécifiques** :
   - **Classe 1** : BLSTM a la précision la plus élevée (99.80%), suivi par LSTM (99.21%) et GRU (98.61%).
   - **Classe 2** : BLSTM (99.39%) et GRU (99.19%) surpassent légèrement LSTM (99.00%).
   - **Classe 4** : BLSTM (97.64%) a la meilleure précision, bien que toutes les précisions soient relativement inférieures à celles des autres classes.
   - **Classe 5** : BLSTM (99.80%) se démarque nettement par rapport aux autres modèles.
   - **Classe 7** : LSTM a la meilleure précision (99.59%), mais BLSTM (99.40%) n'est pas loin derrière.
   - **Classe 10** : BLSTM (99.80%) se distingue à nouveau comme le modèle le plus précis.

7.1.4 Conclusion :

Le modèle BLSTM (Bidirectional LSTM) se montre globalement supérieur aux modèles LSTM et GRU en termes de précision pour la majorité des classes. Sa capacité à capturer les dépendances bidirectionnelles semble lui donner un avantage distinct dans la classification précise des éléments. Bien que LSTM et GRU soient également performants, le BLSTM est clairement le modèle à privilégier pour obtenir la meilleure précision dans ce contexte particulier.

7.2 Calcul du Recall:

7.2.1 Rappel de la définition du recall :

Le **recall** (ou rappel en français) pour une classe *i* mesure la portion d'éléments correctement classés comme appartenant à la classe *i* par rapport à l'ensemble des éléments qui appartiennent réellement à cette classe *i*.

Recall	LSTM	BLSTM	GRU
Classe1	**100**	99.60	99.00
Classe2	**98.80**	98.40	97.40
Classe3	99.00	**99.20**	97.00
Classe4	98.60	**99.40**	98.40
Classe5	99.20	**99.80**	99.40
Classe6	99.40	**99.60**	97.40
Classe7	98.20	**99.00**	97.20
Classe8	98.80	**99.00**	98.80
Classe9	98.80	99.00	**99.20**
Classe10	**99.40**	99.20	98.60

Tableau 4.2 : Le Recall pour une classe *i*

D'après les données du tableau 4.2 sur le Rappel (Recall) pour chaque classe et chaque modèle (LSTM, BLSTM et GRU), nous pouvons tirer les conclusions suivantes concernant la performance des trois modèles en termes de rappel :

7.2.2 Observations générales :

1. **LSTM** : Le modèle LSTM affiche un rappel très élevé pour la plupart des classes, atteignant même 100% pour la classe 1. Ses valeurs de rappel oscillent entre 98.20% et 100%, ce qui montre une bonne capacité à identifier les éléments pertinents dans chaque classe.
2. **BLSTM** : Le modèle BLSTM présente une performance de rappel très solide également, avec des valeurs allant de 98.40% à 99.80%. Il excelle particulièrement dans les classes 5 et 6, avec des rappels respectifs de 99.80% et 99.60%.
3. **GRU** : Le modèle GRU a un rappel légèrement inférieur comparé aux deux autres modèles, avec des valeurs variant de 97.00% à 99.40%. Néanmoins, il reste performant dans certaines classes, comme la classe 9 où il obtient 99.20%.

7.2.3 Comparaison entre les modèles en termes de rappel :

1. **Performance globale** : Le modèle BLSTM se montre légèrement supérieur en termes de rappel, avec des valeurs élevées dans presque toutes les classes. LSTM suit de près, tandis que GRU est légèrement en retrait mais reste compétitif.
2. **Classes spécifiques** :
   - **Classe 1** : LSTM se distingue avec un rappel parfait de 100%, surpassant BLSTM (99.60%) et GRU (99.00%).
   - **Classe 2** : LSTM (98.80%) a un léger avantage sur BLSTM (98.40%) et GRU (97.40%).
   - **Classe 4** : BLSTM a le meilleur rappel (99.40%), devançant LSTM (98.60%) et GRU (98.40%).
   - **Classe 5** : BLSTM (99.80%) se démarque, suivi de près par GRU (99.40%) et LSTM (99.20%).
   - **Classe 7** : BLSTM a une meilleure performance (99.00%) par rapport à LSTM (98.20%) et GRU (97.20%).
   - **Classe 9** : GRU affiche un excellent rappel (99.20%), légèrement supérieur à BLSTM et LSTM (tous deux à 99.00%).

7.2.4 Conclusion :

En termes de rappel, le modèle BLSTM (Bidirectional LSTM) montre une légère supériorité par rapport aux modèles LSTM et GRU. Il démontre une capacité constante à identifier correctement les éléments pertinents dans presque toutes les classes. Le modèle LSTM suit de très près, avec des performances exceptionnelles, notamment un rappel parfait pour la classe 1. Le modèle GRU, bien qu'ayant des performances légèrement inférieures, reste compétitif et performant dans certaines classes spécifiques.

Dans l'ensemble, pour des tâches nécessitant un rappel élevé, le modèle BLSTM serait le choix privilégié, bien que LSTM soit également très performant et pourrait être utilisé efficacement dans des contextes similaires.

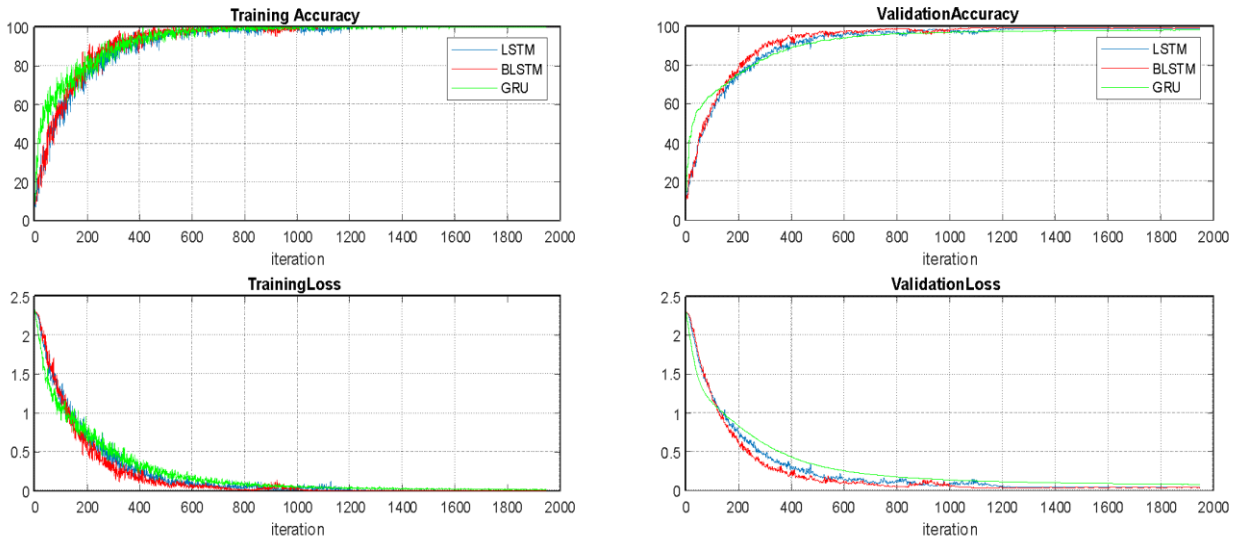
7.3 Calcul des métriques de précisions pour les trois modèles :

	Métriques de précisions pour la validation		
Modèles	**OA**	**AA**	**Kappa x 100**
LSTM	99.02	99.02	98.02
BLSTM	**99.22**	**99.22**	98.02
GRU	98.24	98.25	**98.04**

Tableau 4.3 : Les métriques de précisions pour les trois modèles

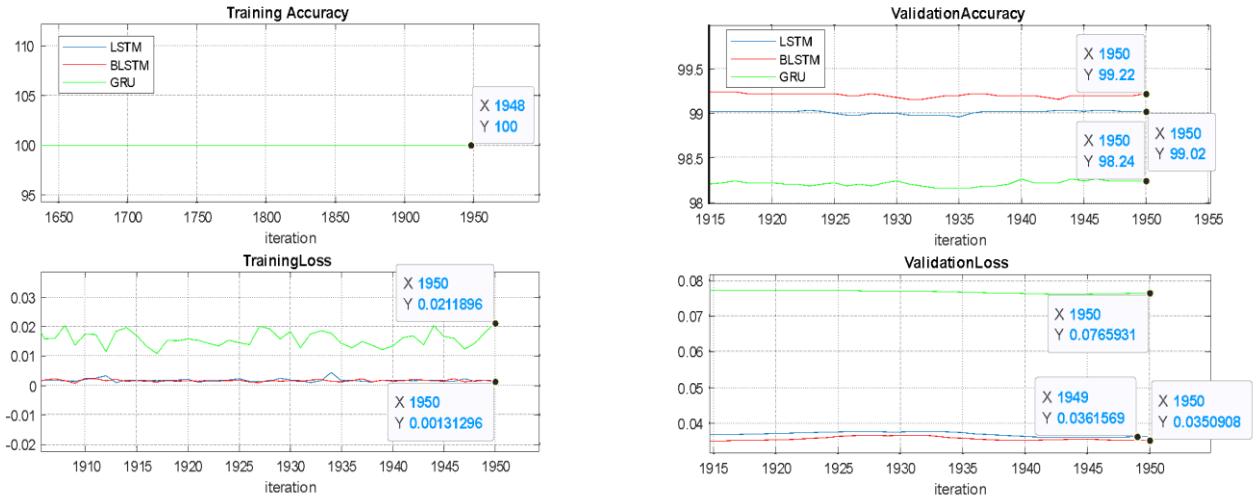
En examinant les valeurs de l'overallaccuracy (OA), de l'averageaccuracy (AA) et du coefficient kappa pour chaque modèle tab 4.3 , voici ce que nous observons :

- **OverallAccuracy (OA)** : BLSTM (99.22%) a la plus haute précision, suivie de près par LSTM (99.02%), et GRU 98.24%) comme l'atteste la Figure(4.3.1) et (4.3.2)



(a1)(b1)

Figure(4.3.1) : Progression (a1) de l'apprentissage, (b1) de la validation des trois réseaux RNN en fonction des itération



(a2) (b2)

Figure(4.3.2) : Zoom de la Progression (a2) de l'apprentissage, (b2) de la validation des trois réseaux RNN en fonction des itérations

- **Average Accuracy (AA)** : BLSTM (99.22%) et LSTM (99.02%) ont une précision moyenne égale, tandis que GRU (98.25%) est légèrement inférieur.
- **Kappa x 100** : Le coefficient kappa pour GRU est en fait de 98.04%, légèrement supérieur à celui des deux autres modèles qui est de 98.02%.

Ainsi, en se basant sur l'overall accuracy (OA) et l'average accuracy (AA), le modèle BLSTM semble être le plus performant, suivi de près par LSTM, et enfin GRU. Le modèle GRU, bien qu'il ait une précision légèrement inférieure, reste compétitif et peut être considéré comme performant également.

En conclusion, le modèle BLSTM semble être le meilleur choix en termes de précision globale dans ce cas.

7.3.1 Conclusion générale en fonction de toutes les métriques :

En considérant la précision, le rappel, l'OA et l'AA, et le coefficient Kappa , le modèle BLSTM émerge effectivement comme le plus performant parmi les trois modèles LSTM, BLSTM et GRU.

- Rappel et précision par classe :
- BLSTM : Affiche des performances impressionnantes avec des précisions et des rappels élevés dans la plupart des classes, témoignant de sa capacité à classifier avec précision les exemples positifs et négatifs de chaque classe.
- Accuracy globale :
- BLSTM : Présente la plus haute précision globale (OA) parmi les trois modèles, avec un OA de 99.22%, indiquant une excellente performance globale.

- Average Accuracy :
- BLSTM : Affiche également la plus haute précision moyenne (AA), ce qui signifie qu'il a une performance généralement élevée sur toutes les classes.
- Cohérence des prédictions :
- BLSTM : Bien que son coefficient kappa soit égal à celui de LSTM (98.02), cela indique une cohérence similaire dans ses prédictions par rapport aux vraies étiquettes.

7.3.2 Conclusion

En combinant tous ces aspects, le modèle BLSTM se distingue comme le plus performant en termes de précision, de rappel, d'accuracy globale (OA) et d'average accuracy (AA). Cela en fait le choix privilégié pour notre tâche de classification.

8 Comparaison entre les trois réseaux de neurones récurrents (RNN) LSTM, BLSTM et GRU en termes de complexité et temps d'exécution :

Critère	LSTM	BLSTM	GRU
Architecture	Architecture complexe avec des cellules LSTM qui ont des portes d'entrée, de sortie et d'oubli.	Deux LSTM en parallèle, un traitant la séquence de manière chronologique et l'autre en sens inverse.	Architecture simplifiée avec des portes de mise à jour et de réinitialisation.
Temps de Calcul	Plus lent en raison de sa complexité.(6 min 48 sec) avec GPU	Plus lent en raison de sa complexité supplémentaire. (7 min 15 sec) avec GPU	Plus rapide à entraîner et à exécuter en raison de sa simplicité.(2min 56 sec) avec GPU

Tableau 4.4: Comparaison entre les trois réseaux de neurones récurrents (RNN)

8.1 Exemple de classification par le LSTM :

La figure 4.4 ci-dessous représente les différentes couches utilisées pour la classification de l'image d'un chiffre écrit manuellement après avoir été transformé en séquence de 28 lignes chacune de 28 pixels.

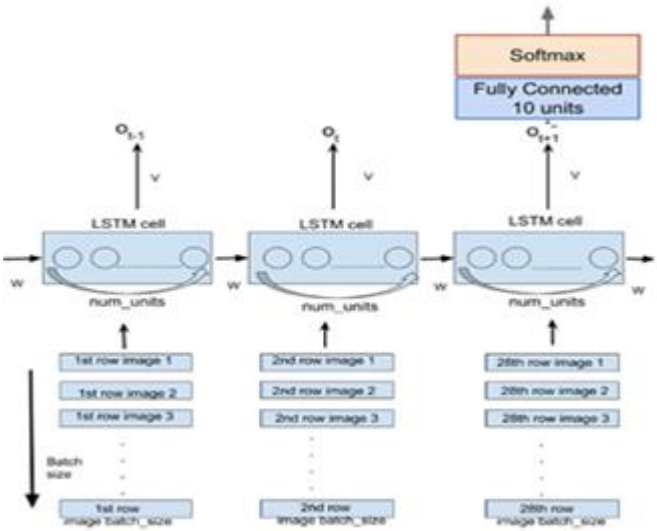


Figure 4.4 : Les différentes couches de la classification de l'image d'un chiffre manuscrit

8.2 Exemple de classification du chiffre zéro par les trois RNN :

8.2.1 Classification du chiffre zéro par le LSTM :

Image du chiffre zéro de taille 28×28 pixels

Transformation en séquence

Séquence de 28 lignes de 28 pixels {L1,1…L1,28} …….{L28,1……..L28,28}

Calcul de la sortie de la couche cachée de la 28 iéme cellule LSTM

h_last = [0.4324 -0.1115 0.9305 -0.9003 -0.8243 0.9016 -0.8706 0.7626 -0.8749 -0.7084
0.3617 0.7817 0.8811 -0.9197 -0.5597 -0.5519 -0.5424 0.9451 -0.7711 -0.7476 -0.8553
-0.9652 0.8975 0.4329 -0.6862 0.6229 0.7165 -0.0816 0.5233 -0.2164 -0.8765 0.4138
-0.5970 0.1770 -0.2064 0.6136 -0.9347 -0.9132 -0.6982 -0.9129 0.4990 0.3991 -0.2003
-0.9795 0.1180 -0.8453 0.9161 -0.9414 0.8365 0.7099 -0.8823 -0.8561 0.9726 0.9436
-0.8377 -0.9328 0.8880 0.5434 0.3537 -0.8144 -0.9065 0.8901 -0.8882 -0.4598 0.8899
-0.9860 0.9324 0.8982 0.8221 0.7217 -0.9679 0.2030 0.9568 -0.8053 -0.4630 -0.4481
0.9859 -0.7324 0.9634 -0.8289 0.8529 0.9298 -0.5049 -0.9236 -0.9528 0.7206 -0.9173
0.2774 0.8426 -0.7046 0.9281 -0.9340 -0.7045 -0.7980 -0.1347 0.6941 -0.9486 -0.8783
0.0562 0.6545]^T

Calcul des sorties de la couche entièrement connectée

Fc = [10.4308 -4.3243 2.4357 1.1423 -5.7359 -0.9724 1.2885 -4.8836 -0.8074 -3.1387]^T

Transformation des sorties de la couche entièrement connectée en probabilités

Softmax = [0.9994 0.0000 0.0003 0.0001 0.0000 0.0000 0.0001 0.0000 0.0000 0.0000]^T

8.2.2 Classification du chiffre zéro par le BLSTM:

Image du chiffre zéro de taille 28×28 pixels 

Transformation en séquence

Séquence de 28 lignes de 28 pixels {L1,1...L1,28} .....{L28,1.....L28,28}

Calcul de la sortie de la couche cachée de la 28 iéme cellule BLSTM

h_last_BLSTM = $[\overrightarrow{h_t}; \overleftarrow{h_t}]$ 200×1 vecteur colonne

$\overrightarrow{h_t}$= [0.7588 -0.9573 -0.9151 0.9717 -0.7401 -0.8668 -0.9561 0.5615 0.9579 -0.4119
-0.8664 -0.9206 -0.8790 -0.9709 -0.9767 0.7916 0.8165 0.6426 0.8700 0.9834 -0.8638
-0.7949 -0.7421 -0.9828 -0.9232 0.8616 -0.8838 -0.7897 -0.8539 -0.9417 0.4063 -0.9012
0.3861 -0.8531 -0.6146 0.9664 0.6273 0.2274 -0.9650 -0.6561 0.5994 -0.7660 0.9638
0.9059 -0.7017 -0.9011 -0.0543 0.5800 -0.3375 0.7238 0.1904 -0.3765 0.4994 -0.6175
0.9916 -0.8094 -0.9290 -0.9396 -0.9453 -0.6722 0.9497 -0.9787 -0.9539 0.8271 -0.8782
-0.8840 0.8376 0.9620 0.9600 -0.9512 0.9737 0.9165 -0.8411 -0.1525 -0.6515 -0.8513
-0.5501 0.9414 -0.9244 -0.8895 0.7707 0.5213 -0.9190 0.9202 0.7563 0.0607 -0.9292
-0.7978 -0.9805 0.6754 -0.9353 -0.9482 0.8974 -0.9610 0.9756 -0.9427 0.9563 0.9861
-0.9225 0.9489]^T 100×1 vecteur colonne

$\overleftarrow{h_t}$ = [-0.9296 0.9609 -0.2700 0.9166 0.9886 0.7877 -0.9761 -0.9388 0.9811 -0.9860
0.9055 -0.9555 -0.7626 0.7589 0.9480 -0.8624 0.9777 0.9454 -0.9795 -0.0341 0.9946
-0.8633 0.9583 -0.5727 -0.9502 0.9308 0.9918 0.9880 0.9388 -0.8696 -0.9010 -0.2251
0.9711 0.6023 -0.9165 -0.3699 0.2133 0.9691 -0.9616 0.9787 0.8079 0.9322 0.6507
0.9838 -0.9644 -0.9823 0.8915 -0.9729 -0.9801 -0.8937 0.6300 0.9208 -0.9798 0.7491
0.9730 -0.9270 0.9557 -0.8376 0.9293 -0.7315 -0.4754 -0.9581 0.9382 -0.9393 -0.9900
0.9895 -0.9788 -0.0354 -0.9112 0.9974 0.8203 0.8150 -0.9046 0.9578 -0.8932 -0.9301
0.9035 0.8369 0.9606 -0.9696 -0.1475 0.7344 0.8311 -0.9826 -0.9234 -0.1587 0.9629
-0.9775 0.8650 -0.3447 -0.9651 0.7211 0.8635 -0.3603 0.9602 -0.9895 -0.9141 -0.9271
-0.9790 0.9274]^T 100×1 vecteur colonne

Calcul des sorties de la couche entièrement connectée

10×1 vecteur colonne

Fc = [9.7875 -4.8720 3.8130 -0.4623 -6.8982 0.1495 0.7185 -5.6990 -1.4082 -0.9446]^T

Transformation des sorties de la couche entièrement connectée en probabilités

10×1 vecteur colonne

Softmax = [0.9997 0.0000 0.0000 0.0000 0.0000 0.0003 0.0000 0.0000 0.0000 0.0000]^T

8.2.3 Classification du chiffre zéro par le GRU:

Image du chiffre zéro de taille 28×28 pixels 

Transformation en séquence aplatissement de l'image

Ligne de Séquence de 784= 28 x 28 pixels {L1.....,L784}

Calcul de la sortie de la couche cachée de la cellule GRU

h_last_GRU =

[0.6099	-0.8176	0.2174	-0.1262	-0.6361	-0.4732	0.1568	0.7350	-0.1304	0.6360	-0.8523
0.5733	0.8247	-0.8403	-0.3713	0.8067	-0.4509	0.4799	-0.3510	0.4803	0.3378	-0.0193
-0.3832	-0.5198	0.8464	-0.8591	-0.6998	0.3908	0.4819	0.6825	-0.5090	0.8219	-0.8973
-0.5652	-0.2318	-0.8892	0.9497	0.5886	-0.5349	0.0907	0.6953	-0.3383	0.5051	-0.1781
0.6610	-0.6201	0.3157	0.3731	0.8319	-0.1410	0.2221	-0.7953	-0.1522	0.0083	-0.3936
-0.1985	0.4617	0.7156	0.6801	0.4378	-0.4871	-0.3000	0.7175	0.7074	-0.1506	-0.2040
-0.2136	-0.8163	0.6443	-0.7975	-0.8940	-0.4786	-0.4769	0.7003	-0.4226	-0.6411	-0.2543
0.3185	-0.2362	0.8383	0.0696	-0.1869	0.0495	0.7855	-0.5326	0.0434	0.8020	-0.3393
0.6997	-0.4004	-0.4403	-0.1328	0.7197	-0.8644	0.7627	-0.8467	0.7122	0.8476	0.4784
-0.6128]^T	100×1 vecteur colonne									

Calcul des sorties de la couche entièrement connectée

10×1 vecteur colonne

Fc = [9.7993 -2.7466 4.5490 0.6770 -7.9776 -1.8801 3.7481 -5.3756 0.3978 -1.3230]^T

10×1 vecteur colonne

Transformation des sorties de la couche entièrement connectée en probabilités

Softmax =[0.9922 0.0000 0.0052 0.0001 0.0000 0.0000 0.0023 0.0000 0.0001 0.0000]^T

8.3Comparaison entre les probabilités de sorties de chaque classe (Softmax) pour les trois RNN :

Sortie dela coucheSoftmax	Prob classe1	Prob classe2	Prob classe3	Prob classe4	Prob classe5	Prob classe6	Prob classe7	Prob classe8	Prob classe9	Prob classe10
LSTM	0.9994	0.0000	0.0003	0.0001	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000
BLSTM	0.9997	0.0000	0.0001	0.0000	0.0000	0.0003	0.0000	0.0000	0.0000	0.0000
GRU	0.9922	0.0000	0.0052	0.0001	0.0000	0.0000	0.0023	0.0000	0.0001	0.0000

Tableau4.5 :La probabilité de chaque classe pour les trois RNN

8.4 CONCLUSION :

D'après les résultats contenu dans le tableau (4.4) le classifieur Softmax a assigné la plus grande probabilité qui est de 0.9994 pour l'LSTM, 0.9997 pour le BLSTM , et 0.9922 pour le GRU à la première sortie qui correspond effectivement au chiffre zéro ce qui démontre que es trois RNN ont bien appris la relation entre les lignes (séquences) des différentes images appartenant à une même classe représentant le même chiffre et à des classes différentes.

Conclusion général

En conclusion, nous avons exploré les fondements des bases de données, Nous avons vu que les bases de données constituent des outils essentiels pour stocker, organiser et manipuler efficacement de vastes quantités de données. Leur rôle dans de nombreux domaines, de la gestion d'entreprise à la recherche scientifique, est indéniable. Le choix du type de base de données et du système de gestion adapté dépend des besoins spécifiques de chaque projet.

Nous avons également parcouru un large éventail de concepts liés à la classification, en examinant en détail les différents types de classifications, les méthodes qui leur sont associées, la conception de quelques algorithmes clés, les domaines d'application de la classification et les indicateurs de performance pertinents pour évaluer les classificateurs.

Nous avons consacré à l'introduction du concept de réseaux neuronaux convolutifs et à leur rôle dans la reconnaissance d'images. Nous avons également abordé les méthodes de classification et les types de couches des réseaux neuronaux, ainsi que l'architecture générale des réseaux neuronaux convolutifs. Nous avons détailler le fonctionnement des réseaux de neurone et en particulier le PMC.

Après avoir examiné en détail les architectures de réseaux de neurones récurrents (RNN) telles que LSTM, BLSTM et GRU dans le contexte de la classification des chiffres manuscrits, il est clair que chacune de ces architectures présente des avantages et des inconvénients distincts. Les LSTM offrent une capacité de mémorisation à long terme, tandis que les BLSTM permettent de capturer les dépendances temporelles dans les deux directions. Les GRU, quant à eux, sont plus simples en termes de structure et peuvent être plus rapides à entraîner.

Cependant, il est difficile de déterminer de manière catégorique qu'elle architecture est la plus efficace pour la classification des chiffres manuscrits, car cela dépend fortement des caractéristiques spécifiques du jeu de données et des contraintes de calcul. Dans certains cas, les LSTM peuvent surpasser les autres architectures en raison de leur capacité à gérer les séquences longues, tandis que dans d'autres cas, les GRU peuvent offrir des performances similaires avec une complexité moindre.

En résumé, ces chapitres soulignent que la comparaison revêt une grande importance car elle permet d'identifier les modèles RNN les plus performants et de guider le choix des architectures appropriées pour cette tâche critique.

Liste Des Références

[1] GUIBERT, Olivier. "Cours de base de données." Département Informatique, Institut Universitaire de Technologie, Université Bordeaux 1, Janvier 2018.

[2] "Conception et réalisation d'une base de données pour la gestion de facturation." Présenté le 04 Mai 2011.

[3] EL AKADI, Ali. "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." Thèse de Doctorat en Informatique et Télécommunications, Université Mohammed V - Agdal Faculté des Sciences Rabat, Mars 2012.

[4] Bouguelia, Mohamed-Rafik. "Classification et apprentissage actif à partir d'un flux de données évolutif en présence d'étiquetage incertain." Thèse de Doctorat, Université de Lorraine, Janvier 2016.

[5] MEGHAZI ABDELKADER, KHIAR MUSTAPHA." Traitement des requêtes OLAP lourde sur cloud." Mémoire de fin d'etude, 2021-2022, dirigé par Mr CHAALAL HICHAM, Université ibn khaldoun Tiaret, Faculté de Mathematiques et Informatique, Département D'Informatique.

[6] CLÉZIOU, Guillaume. "Une méthode de classification non supervisée pour l'apprentissage de règles et la recherche d'information." Thèse de Doctorat, Université d'Orléans, 2004.

[7] Mr BELHADJER Hakim, Mr SAROUEL Brahim. "Classification des images avec les réseaux de neurones Convolutionnels." Mémoire de fin d'étude, 2017-2018, Dirigé par Madame Fellag, Université Mouloud MAMMERI de Tizi-Ouzou, Faculté de Génie Electrique et d'Informatique, Département d'informatique.

[8] Koudri Mohamed. "Mémoire." Université de Tlemcen. Disponible sur : dspace.univ-tlemcen.dz/bitstream/112/1045/4/Memoire.pdf

[9] "Getting started with classification." GeeksforGeeks, [Last Updated : 24 Jan, 2024]. Disponible sur: https://www.geeksforgeeks.org/getting-started-with-classification/

[10] ARLOT, Sylvain. "Classification supervisée: des algorithmes et leur calibration * automatique." Cours de 3ème année, École Centrale de Paris, Mars 2009.

[11] LAOUAMER, Lamri. "Approche exploratoire sur la classification appliquée aux images." Mémoire, Université du Québec, Avril 2006.

[12] CHESNEAU, Christophe. "Cours Éléments de Classification." Master 1 MIASHS,

Université de Caen.

[13] Akbani, R., Kwek, S., &Japkowicz, N. (2004). "Applying support vector Machine to imbalanced Datasets." Pages 35-50.

[14] CLÉZIOU, Guillaume. "Une méthode de classification non supervisée pour l'apprentissage de règles et la recherche d'information." Thèse de Doctorat, Université d'Orléans, 2004.

[15] ALI, LABIAD. "Sélection des mots clés basée sur la classification et l'extraction des règles d'association." Mémoire, Juin 2017.

[16] "K-Means: Comment ça marche." Blent AI, [date non spécifiée]. Disponible sur: https://blent.ai/blog/a/k-means-comment-ca-marche

[17] Anil K.Jain et Richard C.Dubes, "Algortithms of clustring data",Michigan state university (1988).

[18] ARLOT, Sylvain. "Cours de classification." Université Paris-Saclay, [date non spécifiée].

Disponible sur: https://www.imo.universite-paris-saclay.fr/~sylvain.arlot/enseign/2009Centrale/cours-classif.pdf

[19] SERIDI, Merwan."Application des réseaux de neurones pour la classification des données." Dirigé par Mr Debeche Mehdi,Université 08 mai 1945 Guelma, Octobre 2020.

[20] "Matrice de confusion." Data Scientist, [date non spécifiée]. Disponible sur: datascientest.com/matrice-de-confusion

[21] "Accuracy Metrics." Humboldt State University, Disponible sur: gsp.humboldt.edu/olm /Courses/GSP_216/lessons/accuracy/metrics.html

[22] TSCHIRHART, Fabien. "Réseau de Neurones Formels Appliqués à l'Intelligence et au Jeu." Mémoire de recherche, École Supérieure de Génie Informatique, Paris, 2009.

[23] DREYFUS, G. "Les Réseaux de Neurones." Mécanique Industrielle et Matériaux, No.51,

Septembre 1998.

[24] "Neurone formel." Wikipedia, [date de visite: 10/04/2020].Disponible sur: https://fr.wikipedia.org/wiki/Neurone_formel

[25] "Réseaux de neurones automatisés." StatSoft, [date de visite: 18/08/2020].

Disponible sur: http://www.statsoft.fr/concepts-statistiques/reseaux-de-neurones-automatises/reseaux-de-neurones-automatises.htm#fonctions

[26] MAZIR, Melissa et AMRIOU, Hanane. "Convolutional Neural Network pour la

Détection du Port du Masque." Spécialité: Réseaux et Télécoms, 2021/2022.

[27] "Focus: Réseaux de neurones convolutifs." Pensée Artificielle. Disponible sur: https://penseeartificielle.fr/focus-reseau-neurones-convolutifs/#4_Le_flattening _ou_mise_a_pla

[28] LILIA, D.R. "La Détection de la colère chez le conducteur en utilisant le Deep Learning." Algérie/BISKRA, 2020.

[29] PHUNG, Van Hiep et RHEE, EunJoo. "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

[40] MALKI, Narimene. "Classification automatique des textes par les réseaux de neurones à convolution." Année académique 2018/2019.

[31] HABBA, Abdelaziz et ISHAK, Omar. "La classification des images satellitaires par l'apprentissage profond (deeplearning)." Encadré par Mr. OUAHAB Abdelwhab, Année Universitaire 2018/2019.

[32] "Réseaux de neurones récurrents." Data Analytics Post, [date non spécifiée]. Disponible sur: dataanalyticspost.com/Lexique/reseaux-de-neurones-recurrents/

[33] "Long short-term memory." Wikipedia, Disponible sur: en.wikipedia.org /wiki /Long_short-term_memory

[34] TSCHIRHART, Fabien. "Réseau de Neurones Formels Appliqués à l'Intelligence et au Jeu." Mémoire de recherche, sous la direction de M. Alain Lioret, École Supérieure de Génie Informatique, Paris, 2009.

[35] CHRAIBI KAADOUD, Ikram. "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.

[36] GERS, Felix et al. "Learning Precise Timing with LSTM Recurrent Networks." Journal of Machine Learning Research, vol. 3, 2002.

[37] RIVALS, I. "Modélisation et commande par réseaux de neurones : application au pilotage d'un véhicule autonome." Thèse de Doctorat, Université Pierre et Marie Curie, 1995.

[38] DONG, X. et al. "Transfer bi-directional LSTM RNN for nam edentity recognition in Chinese electronic medical records." In 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom).

[39] TSENG, Sy et LU, Wen-Kung. "The system for appraisal of vehicle accident based on radial basis function neural networks." DOI: 10.1109/ICNC.2011.6022220, 2011, pp. *

869–872.

[40] MOLLER, M.F. "A scaled conjugate gradient algorithm for fast supervised learning." Neural Networks, 6, 1993, pp. 525– [41] M. F. Moller, “A scaled conjugate gradient algorithm for fast supervised learning”, Neural Networks, 6, 1993, pp. 525.

[41] LOUNIS, Katia et MOUSSI, Dahbia. "La Classification d’images d’insectes ravageurs en utilisant le Deep Learning." Encadré par Mme AOUDJIT, Co-promotrice: Mme AIT ISSAD, Année académique 2019-2020.

[42] Jacques Prado ." introduction a matlab" , validé le 19 novembre 2019