

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE DES MATHÉMATIQUES ET  
DE L'INFORMATIQUE



**DÉPARTEMENT D'INFORMATIQUE**

**MEMOIRE de fin d'études**

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

**Domaine : Mathématiques et Informatique**

**Filière : Informatique**

**Spécialité : Informatique Décisionnel et Optimisation**

**Thème**

*Développement d'un modèle ANN efficace  
pour détecter et classifier des patients diabétiques*

**Réalisé par :**  
**LAOUES Baya**

**Encadrant :**  
**KADRI Said**

**Soutenu publiquement le: / /2020 devant la commission du jury:**

.....  
.....  
.....

**Université de M'sila**  
**Université de M'sila**  
**Université de M'sila**

.....  
.....  
.....

**Promotion : 2019 /2020**



# Sommaire:

Liste des Figures .....	V
Liste des Tables .....	V
Dédicace: .....	VI
Remerciements:.....	VII
INTRODUCTION GÉNÉRALE .....	1
CHAPITRE I: LE DIABÈTE	
I.1. Introduction: .....	4
I.2. Le diabète gestationnel:.....	5
I.2.1. Définitions:.....	5
I.2.2. Facteurs de risque de diabète gestationnel:.....	5
I.2.3. Les symptômes :.....	6
I.2.4. Causes de diabète de grossesse :.....	6
I.2.5. Comment diagnostique un diabète gestationnel ?.....	6
I.2.6. Les complications de diabète gestationnel :.....	7
I.2.7. Les complications de diabète gestationnel :.....	7
CHAPITRE II: L'APPRENTISSAGE AUTOMATIQUE ET L'APPRENTISSAGE APPROFONDI	
II.1. introduction: .....	10
II.2. l'apprentissage automatique: .....	11
II.2.1.Types d'apprentissage automatique:.....	11
II.2.1.1. Apprentissage Supervisé: .....	12
II.2.1.2. Apprentissage Non Supervisé:.....	12
II.2.1.3. Apprentissage Par Renforcement: .....	12
II.2.2. Quelques algorithmes :.....	13
II.2.2.1. Les arbres de décisions: .....	13
II.2.2.2. Random forest : .....	14
II.2.2.3. Les machines à vecteur de support SVM:.....	15
II.2.2.4. Algorithme de Naïves Bayes:.....	15
II.2.2.5. Algorithme de K plus proche voisins: .....	15
II.3. L'apprentissage profond (approfondi) :.....	16
II.3.1. Histoire d'apprentissage approfondi : .....	16
II.3.2. Pourquoi l'apprentissage approfondi: .....	17
II.3.3. Les modèles d'apprentissage approfondi:.....	18
II.3.3.1. Les réseaux de neurones artificiels (ANN) : .....	18
II.3.3.2. Les réseaux de neurones Convolutifs (CNN) : .....	21

II.3.3.3. Les réseaux de neurones Récurrents (RNN) :	24
II.3.3.4. Les réseaux Long Short-Term Memory (LSTM) :	24
II.3.3.5. RBM:	25
II.3.3.6. DBM:	25
II.4. Conclusion:	25

### CHAPITRE III: ENVIRONNEMENT DEPROGRAMMATION

III.1. Introduction:	27
III.2. Présentation des outils:	27
III.2.1. Le hardware :	27
III.2.2. Les outils et les bibliothèques utilisés:	27
III.2.2.1. Python:	27
III.2.2.2. ANACONDA:	27
III.2.2.3. TensorFlow:	29
III.2.2.4. Keras:	29
III.2.2.5. Quelques Bibliothèques :	29
III.3. Base de données pima_diabetes :	31
III.4. Conclusion :	31

### CHAPITRE IV : IMPLÉMENTATION

IV.1. introduction:	33
IV.2. Problématique étudiée :	33
IV.3. Le modèle proposé :	33
IV.4. Évaluation :	34
IV.5. Résultats obtenus :	35
IV.6. Discussions :	40
IV.7. Quelques interfaces de l'application :	41
IV.8. Conclusion :	42
CONCLUSION GÉNÉRALE	43
Les notifications d'acceptation des papiers dans les conférences:	45
R ÉFÉRENCES BIBLIOGRAPHIQUES	46
Abstract:	50
Résumé:	50
ملخص:	50

## Liste des Figures

<i>Figure II.1: Intelligence artificielle, Machine Learning et Deep Learning</i> .....	10
<i>Figure II.2: Types d'apprentissage automatique</i> .....	13
<i>Figure II.3: Random Forest</i> .....	14
<i>Figure II.4: Le procédé du ML classique comparé à celui du Deep Learning</i> .....	18
<i>Figure II.5: Fonctions de seuil</i> .....	19
<i>Figure II.6: Fonctions linéaires</i> .....	19
<i>Figure II.7: Fonctions sigmoïdes</i> .....	20
<i>Figure II.8: Réseau mono-couche</i> .....	20
<i>Figure II.9: Réseau multi-couche</i> .....	21
<i>Figure II.10: Réseaux récurrents</i> .....	21
<i>Figure II.11: Réseau neuronal convolutif</i> .....	22
<i>Figure II.12: Architecture simple de CNN</i> .....	22
<i>Figure II.13: Max pooling</i> .....	23
<i>Figure II.14: Réseau de neurone récurrent</i> .....	24
<i>Figure IV.1: Architecture du modèle ANN proposé</i> .....	34
<i>Figure IV.2: Training loss Vs Validation loss du modèle ANN</i> .....	39
<i>Figure IV.3: Training Accuracy Vs Validation Accuracy du modèle ANN</i> .....	39
<i>Figure IV.4: Interface principale de l'application</i> .....	41
<i>Figure IV.5 : Exécution de premier exemple</i> .....	41
<i>Figure IV.6: Exécution de deuxième exemple</i> .....	42

## Liste des Tables

<i>Table II.1: Les étapes majeures du Deep Learning</i> .....	17
<i>Table II.2: Types d'architectures de réseaux</i> .....	21
<i>Table IV.1: Précision (accuracy) obtenue après l'application de différents algorithmes de ML</i> .....	37
<i>Table IV.2: Description du modèle proposé</i> .....	38
<i>Table IV.3: Les valeurs de perte et de précision obtenues lors de l'application du modèle proposé</i> .....	38

## *Dédicace:*

*Je dédie ce mémoire :*

*À ma très chère mère*

*Quoi que je fasse ou que je dise, je ne pourrai jamais te remercier  
comme il se doit. Ta présence à coté de moi a toujours été ma source de  
force*

*À mon très cher père*

*Tu as toujours été à mon coté pour me soutenir et m'encourager.  
Que ce travail traduit ma gratitude et mon affection.*

*À mon cher frère et mes très chères sœurs pour leur  
Patience, leur amour, leur soutien et leur  
Encouragement.*

*À tous ceux que j'aime.*

*À tous ceux qui ont toujours cru en moi et poussé à réussir, je vous  
dédie ce travail.*

## Remerciements:

*Avant tout, je remercie Dieu de m'avoir aidé à réaliser ce travail.*

*Je tiens à exprimer ma très profonde gratitude à Monsieur KADRI SAID, qui n'a épargné aucun effort pour me prendre en charge pour la réalisation de ce travail. Sa clairvoyance, sa générosité, sa gentillesse, ses connaissances, le temps qui m'a dispensé, et sa grande disponibilité dont il a fait preuve, m'ont énormément facilité ma tâche.*

*J'adresse mes très sincères remerciements à tous les membres du jury pour l'honneur qu'ils m'ont fait en acceptant de juger ce travail.*

## Liste des abbreviations

**IA:** Intelligence Artificielle.

**ML:** Machine Learning

**DL:** Deep Learning

**HGPO:** Hyperglycémie provoquée par voie orale

**ANN:** Artificial Neural Networks.

**CNN:** Convolutional Neural Networks

**ReLu:** Rectified Linear Unit

**LSTM:** Long Short-Term Memory

**RNN:** Recurrent Neural Network

**SVM:** Support Vector Machine

**NN:** Neural Network

**RAM:** Random Access Memory

**NB:** Naïve Bayésienne

**KNN:** K-Nearest Neighbors



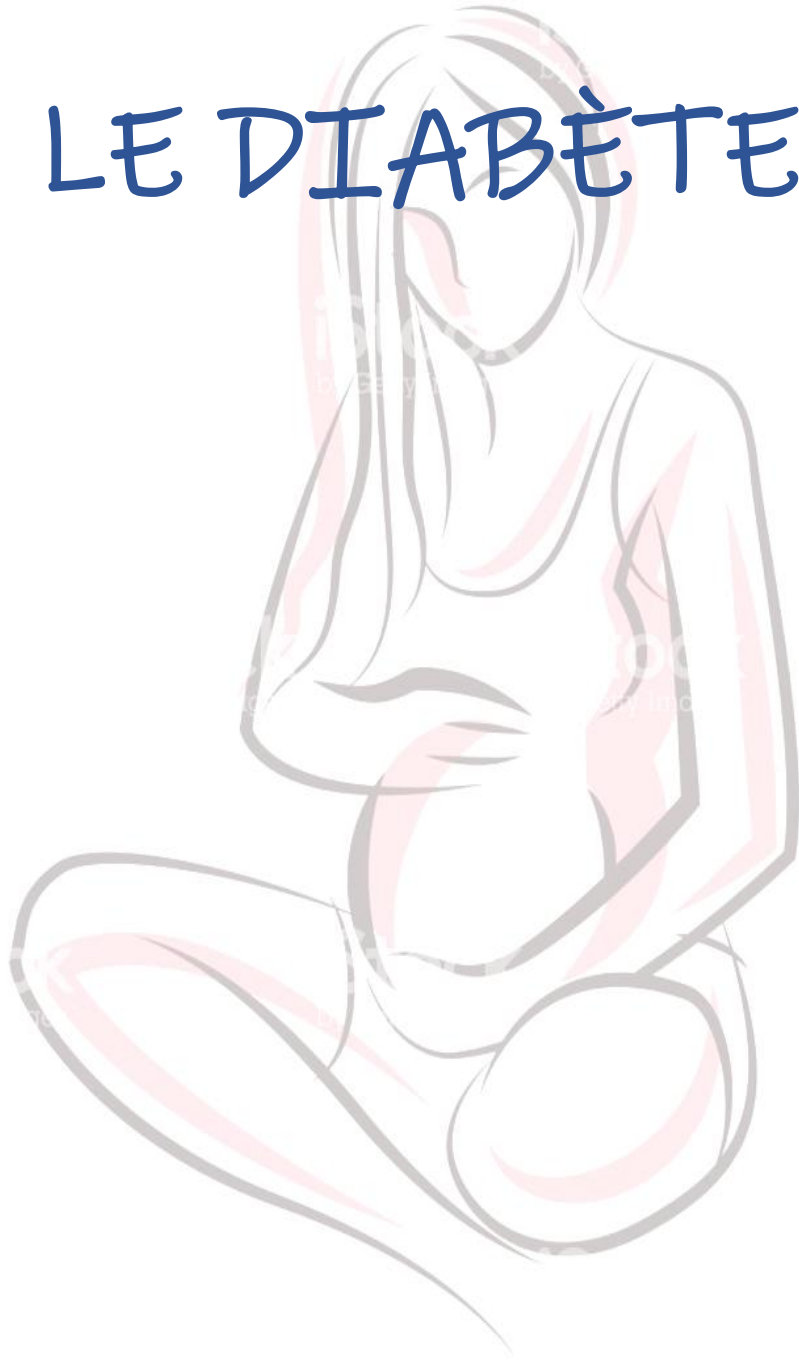
INTRODUCTION  
GÉNÉRALE

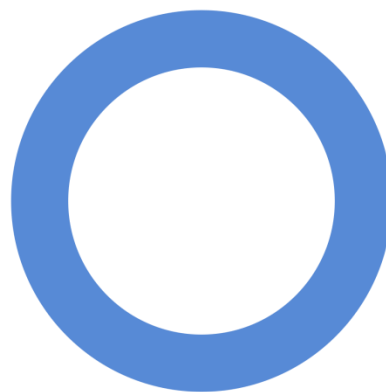
Le diabète est l'une des maladies chroniques qui provoquent une augmentation du taux de sucre dans le sang. De nombreuses complications graves peuvent se produire sur le corps du patient si le diabète n'est pas traité ou s'il n'est pas identifié. La meilleure façon pour identifier le diabète consiste à se rendre dans un centre de diagnostic du diabète ou à consulter un médecin spécialiste. L'évolution de l'apprentissage automatique (Machine Learning ML) et de l'apprentissage approfondi (Deep Learning DL) peut contribuer de manière significative à la résolution de ce problème critique.

La principale tâche à accomplir en utilisant ces approches est de concevoir un modèle prédictif qui aide à pronostiquer la probabilité de diabète chez les patients avec un maximum de précision. En termes simples, le domaine de l'apprentissage automatique est basé sur un ensemble de méthodes qui permettent à une machine d'apprendre des caractéristiques (modèles) significatives à partir de données sans l'interaction de l'homme. Au cours des dernières années, de nombreuses tâches d'IA comme la reconnaissance de formes, la classification d'images, la vision par ordinateur, etc. sont essentiellement basées sur des approches d'apprentissage automatique qui ont donné de très bons résultats. Plusieurs algorithmes ont été développés dans ce domaine, notamment : la régression logistique, les k plus proches voisins, Naïve Bayes, les arbres de décision, la machine à vecteurs de support, les réseaux neuronaux artificiels, etc. Les recherches sur les réseaux neuronaux artificiels RNA peuvent être considérées comme la plus ancienne discipline en ML et AI qui remonte à J. McCulloch et W. Pitts en 1943, mais ces recherches ont été rapidement interrompues en raison de leurs exigences élevées en termes de matériel, de logiciel et de temps d'exécution. De nombreuses années plus tard, elles ont été relancées en parallèle avec l'apparition d'un nouveau sous-domaine appelé apprentissage profond DL. Cette nouvelle approche simplifie considérablement le processus d'ingénierie des fonctionnalités dans de nombreux domaines vitaux tels que l'imagerie médicale.

# CHAPITRE I:

# LE DIABÈTE





### **I.1. Introduction:**

Depuis des années les maladies cardiaques ne sont pas devenues un signe de mort comme les gens les voyaient dans le passé parce qu'à nos jours le diabète est parmi les plus dangereuses maladies qui parfois provoquent d'autres maladies mortelles et graves tel que les maladies cardio vasculaire.

Le diabète est une maladie chronique caractérisé par une concentration élevée de sucre dans le sang, qui survient lorsque le pancréas ne secret assez d'insuline et dans certains types il y'a aucune production d'insuline.

Il y a deux grands types de diabète :

Type 1: c'est un type qui touchent certains personnes un peu prêt a 6 pourcent des diabétiques .C'est une maladie auto immune connu par le nom insulino-dépendant puisque il est caractérisé par l'absence totale de production d'insuline .Les causes de ce type sont toujours non précises mais certains scientifique pensent que ce diabète est une maladie génétique auto-immune tout se passe comme si l'organisme détruit les cellules bêta des ilots Langerhans qui secrètent d'insuline.

Type 2: appelé le diabète de la maturité représente environs 90 pourcent des cas de diabète se déclare généralement après l'age de 40 ans et aussi touche les enfants .Le diabète de ce type souvent considère comme une maladie lie au style de vie, il survient lorsque le pancréas continue de secréter de l'insuline mais l'organisme y est devenu résistant et cela logiquement traduit par une très mauvaise régulation de la glycémie.

## I.2. Le diabète gestationnel:

La présence d'un diabète au cours de la grossesse peut correspondre soit à un diabète préexistant et diagnostiqué avant la grossesse, soit à un diabète diagnostiqué au cours de la grossesse qui se nomme un diabète gestationnel.

### I.2.1. Définitions:

Selon l'OMS le diabète gestationnel est un trouble de la tolérance glucidique, conduisant à une hyperglycémie de sévérité variable, débutante ou diagnostiqué pour la première fois pendant la grossesse sous le terme diabète gestationnel.

Il est aussi appelé diabète de grossesse, c'est une pathologie qui touche seulement les femmes enceintes au cours de leur deuxième trimestre d'accouchement.

- Chiffre clé : une étude descriptive rétrospective qui concerne 262 femmes enceintes diabétiques hospitalisées dans le service de gynécologie obstétrique de l'EHS mère-enfant de Tlemcen du 1<sup>er</sup> janvier 2016 au 30 avril 2017. La collecte des données a été faite à partir des dossiers des patientes.
- Résultats : la fréquence du diabète chez les femmes enceintes est de 1.33 % dont 25.5 % DT1, 39.4 % DT2, 35,9% gestationnel. L'âge moyen est de 34 ans (extrêmes 20 à 48 ans) [1].

### I.2.2. Facteurs de risque de diabète gestationnel:

En particulier dans ce type les médecins se basent sur les facteurs de risque au lieu de se baser sur les symptômes ces facteurs sont :

- L'âge de la femme est de plus de 35 ans.
- Le sur poids et l'obésité.
- Le syndrome des ovaires poly kystiques.
- Des antécédents familiaux de diabète de type 2 ou personnel de diabète gestationnel.
- La macrosomie.
- Souffrir d'une décoloration de la peau souvent brunâtre au niveau du cou et sous les bras

### I.2.3. Les symptômes :

- Maux de tête.
- Soif exagérée.
- Polyurie et polydipsie.
- Vertige.

Généralement ces symptômes sont inaperçus ou plutôt la femme n'y prêt aucune attention car ils surviennent chez tous les femmes enceintes quel que soit elle est diabétique ou non.

### I.2.4. Causes de diabète de grossesse :

Durant la grossesse les besoins en insuline de la femme enceinte sont de 3 fois plus qu'en temps normale ; car le placenta sécrète des hormones anti insuline donc l'organisme doit contrebalancer entre les deux hormones, mais chez certaines femmes le pancréas ne parvient pas à sécréter assez d'insuline et cela entrainait une hyperglycémie et puis un diabète gestationnel.

### I.2.5. Comment diagnostique un diabète gestationnel ?

- **En premier lieu** et dans le premier trimestre ce type diagnostiqué par une glycémie à jeun.
- **En deuxième lieu** par mesurer des glycémies lors d'une HGPO (Hyperglycémie provoquée par voie orale). HGPO est réalisée entre la 24ème et la 28ème semaine de la grossesse. Elle consiste en l'absorption d'une quantité standard de glucose (75 g) ; le dépistage s'effectue par dosage de la glycémie 1 heure, puis 2 heures après l'ingestion.

Les valeurs seuils retenues par le Collège National des Gynécologues et Obstétriciens Français (CNGOF) sont les suivantes :

- Une glycémie à jeun avec un seuil fixé 0.92g /L (5.1mmol/l)
- Une glycémie après une heure avec un seuil fixé 1.80g/L (10mmol/l)
- Une glycémie après 2 heures avec un seuil fixé 1.53g/L (8.5mmol/l)

Donc s'il y'a une seule valeur anormale de glycémie égale ou supérieure aux seuils définis suffit à diagnostiquer un diabète gestationnel.

### I.2.6. Les complications de diabète gestationnel :

Le diabète en générale provoque d'autres dangereuses maladies, mais la différence dans ce cas c'est qu'il menace deux vies; la vie de la maman et des risques pour la vie de bébé.

#### a) Pour la maman :

- La complication la plus grave est la survenue d'une pré-éclampsie (ou toxémie gravidique) pouvant associer prise de poids, œdèmes et hypertension artérielle.
- Surplus de liquide amniotique donc un accouchement prématuré.
- Infection urinaire.
- Accouchement par césarienne.
- Risque de développer en diabète de type 2 après l'accouchement.

#### b) Pour le bébé :

- Détresse respiratoire et risque d'un manque de calcium.
- D'avoir un poids élevé que la moyenne (plus de 4 kg).
- D'être obèse plus tard et de présenter une intolérance au glucose au début de l'âge adulte.
- Risque d'être un diabétique au plus tard.

Toutes ces complications s'observent surtout lorsque le diabète gestationnel n'est pas bien contrôlé. Lorsque le diabète de grossesse est traité adéquatement, le risque de développer une de ces complications est grandement réduit et devient presque aussi faible que celui d'une femme non diabétique.

### I.2.7. Les complications de diabète gestationnel :

De plus en plus de preuves scientifiques appuient l'efficacité du traitement du diabète gestationnel à réduire les complications. Le traitement est adapté au cas par cas. Un lecteur de glycémie est indispensable pour vérifier et corriger au besoin la qualité du contrôle de la glycémie.

Quelques modifications au régime alimentaire et au mode de vie sont souvent suffisantes pour maintenir la glycémie à des taux acceptables, et pour que la mère et l'enfant se portent bien.

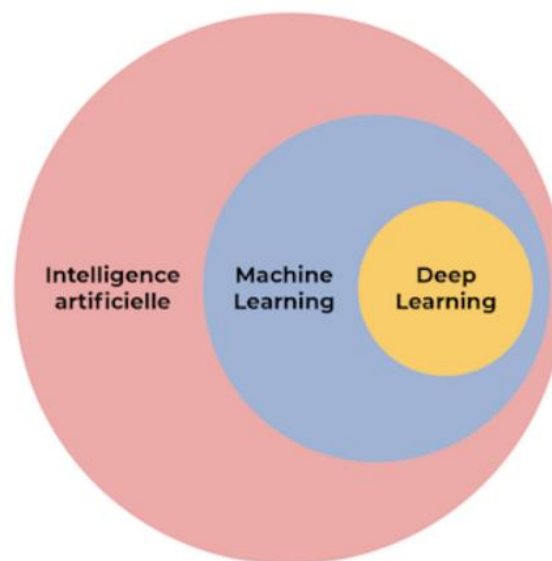
L'insuline est réservée aux femmes pour qui la modification de l'hygiène de vie ne permet pas à elle seule de contrôler la glycémie. La plupart des médicaments antidiabétiques sont contre-indiqués en cas de grossesse. Selon des études récentes, la metformine serait une solution de rechange efficace à l'insuline, et suffisamment sécuritaire pour la proposer aux femmes enceintes. Cependant, l'insuline demeure préférable aux médicaments antidiabétiques [2].



## II.1. introduction:

L'objectif de la recherche en intelligence artificielle (IA) est de doter un système informatique de capacités de réflexion similaires à celles des humains. Il y a donc un défi dans la compréhension du raisonnement humain, mais surtout dans la modélisation et dans la reproduction de celui-ci [3].

En raison de nombreuses réalisations, l'intelligence artificielle est devenue un sujet brûlant dans les médias et magazines scientifiques. De grandes entreprises telles que Google, Facebook, IBM, Microsoft et des constructeurs automobiles tels que Toyota, Volvo et Renault sont très actifs dans la recherche sur l'IA et prévoient d'y investir davantage à l'avenir. Plusieurs scientifiques spécialisés en IA dirigent désormais les laboratoires de recherche de ces grandes entreprises et bien d'autres. La recherche en intelligence artificielle a fait des progrès significatifs dans divers domaines au cours de la dernière décennie. L'avancement le plus célèbre est l'apprentissage automatique, notamment grâce au développement d'architectures d'apprentissage profond, des réseaux de neurones convolutifs multicouche dont l'apprentissage s'opère à partir de gros volumes de données sur des architectures de calcul intensif. Parmi les réalisations de l'apprentissage automatique, il convient de citer la résolution de jeux Atari (Bricks, Space invaders, etc.) par Google DeepMind, utilisant les pixels images affichés à l'écran comme données d'entrée afin de décider quelle action adopter pour atteindre le plus haut score possible à la fin de la partie [4].



**Figure II.1:** Intelligence artificielle, Machine Learning et Deep Learning

Dans ce chapitre, nous aborderons différents types d'apprentissage automatique et présenterons très brièvement leurs principes en détail, puis nous continuerons l'apprentissage en profondeur et examinerons leurs différents types.

## II.2. L'apprentissage automatique:

L'apprentissage automatique devient important dans toutes les disciplines. C'est aussi une branche de l'intelligence artificielle. L'apprentissage automatique peut signifier l'utilisation de machines (ordinateurs et logiciels) pour gagner du sens à partir des données. Cela peut également signifier donner aux machines la possibilité d'apprendre de leur environnement.

L'apprentissage automatique concerne l'exploration et le développement de modèles et d'algorithmes mathématiques pour apprendre des données telles que les documents, l'audio, les images, les films, le commerce électronique, l'énergie électrique, la médecine et la biologie. Son paradigme se concentre sur la classification des objectifs et consiste à modéliser une cartographie optimale entre le domaine de données et l'ensemble de connaissances et à développer des algorithmes d'apprentissage, ce qui nécessite un ensemble de données de formation (étiqueté), un ensemble de données de validation et un ensemble de données de test [5].

L'ensemble du processus d'apprentissage nécessite un ensemble de données comme suit :

- **Ensemble de données pour l'entraînement** : c'est la base de connaissance utilisée pour entraîner, notre l'algorithme d'apprentissage, pendant cette phase, les paramètres du modèle peuvent être réglés (ajustés) en fonction des performances obtenues.
- **Ensemble de données pour le test** : cela est utilisé juste pour évaluer les performances du modèle sur les données non-vues.

### II.2.1.Types d'apprentissage automatique:

Selon le type de données disponibles, l'apprentissage automatique peut être classé en apprentissage supervisé, apprentissage non supervisé et apprentissage par renforcement.

Supervisé et non supervisé sont principalement utilisés par beaucoup d'ingénieurs d'apprentissage automatique et de geeks de données, l'apprentissage par renforcement est vraiment puissant et complexe à appliquer pour les problèmes.

#### **II.2.1.1. Apprentissage Supervisé:**

L'apprentissage supervisé est le type d'apprentissage le plus simple. Il est basé sur un certain nombre d'exemples pré classifiés. Cette approche a pour objectif la conception d'un modèle reliant des données d'apprentissage à un ensemble de valeurs de sortie (un comportement) [6]. Nous donnons à l'machine différents paires d'entrées et de sorties (données de formation) . Nous formons d'abord le modèle avec beaucoup de données de formation, ensuite lorsque de nouvelles entrées seront présentées, vous aurez une sortie intelligente.

Dans un problème de régression, l'ensemble d'apprentissage est une paire formée par un objet et une valeur numérique associée. Il existe plusieurs algorithmes d'apprentissage supervisé qui ont été développés pour la classification et la régression. Parmi tous, les arbres de décision, les règles de décision, les réseaux de neurones et les réseaux bayésiens [7] [8] [9].

#### **II.2.1.2. Apprentissage Non Supervisé:**

Le deuxième type d'apprentissage automatique est appelée apprentissage non supervisé, dans ce cas, les données de formation n'incluent pas les cibles ici, donc nous ne disons pas au système où aller. Les données de formation ne sont pas structurées (contiennent des données bruyantes, des données inconnues, etc.) et le système doit se comprendre à partir des données que nous donnons, extrayez automatiquement la signification des données.

Les algorithmes d'apprentissage non supervisé sont particulièrement utilisés dans problèmes de clustering, dans lesquels, étant donné une collection d'objets, nous voulons être en mesure de comprendre et de montrer leurs relations. Une approche standard consiste à définir une mesure de similarité entre deux objets, puis à rechercher tout groupe d'objets plus similaires les uns aux autres, par rapport aux objets des autres clusters [7] [8] [9].

#### **II.2.1.3. Apprentissage Par Renforcement:**

L'apprentissage par renforcement est une approche de l'intelligence artificielle qui met l'accent sur l'apprentissage du système à travers ses interactions avec l'environnement. Avec l'apprentissage par renforcement, le système adapte ses

paramètres en fonction des réactions reçues de l'environnement, qui fournit ensuite un retour d'information sur les décisions prises. Par exemple, un système qui modélise un joueur d'échecs qui utilise le résultat des étapes précédentes pour améliorer ses performances, est un système qui apprend avec le renforcement. La recherche actuelle sur l'apprentissage avec renforcement est hautement interdisciplinaire et comprend des chercheurs spécialisés dans les algorithmes génétiques, les réseaux de neurones, la psychologie et les techniques de contrôle [7] [9].

La figure suivante résume les trois types d'apprentissage avec les problèmes connexes à résoudre.

Apprentissage supervisé	Apprentissage non supervisé	Apprentissage par renforcement
<ul style="list-style-type: none"> <li>. Classification</li> <li>. Régression</li> </ul>	<ul style="list-style-type: none"> <li>. Clustering</li> <li>. Segmentation</li> <li>. Réduction de dimension</li> </ul>	<ul style="list-style-type: none"> <li>. Processus de décision</li> <li>. Systèmes de recommandation</li> </ul>

**Figure II.2:** Types d'apprentissage automatique.

## II.2.2. Quelques algorithmes :

### II.2.2.1. Les arbres de décisions:

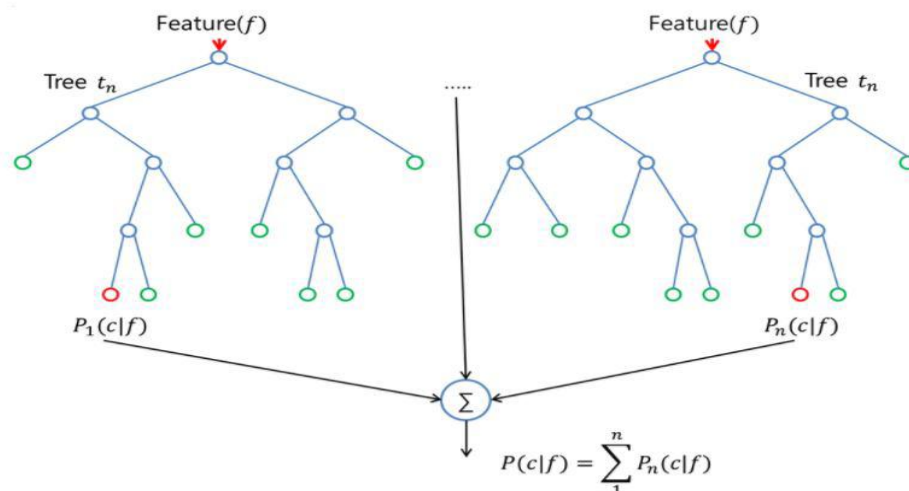
Les arbres de décision sont des modèles d'apprentissage automatique supervisés, pouvant être utilisés pour la classification que pour la régression.

L'arbre de décision représente une fonction qui prend les vecteurs d'attribut en entrée et renvoie une décision unique. L'entrée et la sortie peuvent être discrètes ou continues. L'arbre de décision prend des décisions en exécutant une séquence de tests. Chaque nœud interne de l'arbre correspond au test de la valeur d'attribut et la branche du nœud est la valeur d'attribut possible. Ensuite, la feuille de calcul détermine la catégorie de la variable cible, où observez la fin de la séquence de tests. La phase d'apprentissage consiste à trouver la séquence de tests correcte. Pour cela, il faut décider de maintenir de bonnes propriétés. Les bons attributs divisent les exemples en ensembles homogènes, c'est-à-dire qu'ils ne n'incluent que les observations appartenant à la même catégorie alors qu'un attribut inutile laissera les exemples avec presque la même proportion de valeur pour la variable cible. Ce dont

on a besoin c'est d'une mesure formelle de "bon" et "inutile". Pour cela, il existe des métriques standards homogénéisées avec lesquels on peut mesurer l'homogénéité d'un ensemble. Les plus connus sont l'indice de diversité de Gini et l'Entropie [10].

### II.2.2.2. Random forest :

La forêt aléatoire est un algorithme d'apprentissage supervisé. Comme tu vois à partir de son nom, il va créer une forêt et la randomiser. La "forêt" qu'il a construite est L'ensemble d'arbres de décision est principalement formé par la méthode de «bagging». L'idée générale de la méthode bagging n'est qu'une combinaison de modèles d'apprentissage améliorant l'effet global. En bref: Random Forest construit plusieurs arbres de décision, combiner pour obtenir des prévisions plus précises et stables. Un grand avantage de la forêt aléatoire est qu'elle peut être utilisée pour La classification et la régression constituent l'essentiel du système d'apprentissage courant automatique.



**Figure II.3:** Random Forest.

L'algorithme de forêt aléatoire ajoute un caractère aléatoire supplémentaire à son modèle à mesure que les arbres grandissent. Dans II ne recherchera pas la meilleure fonction lors de la division des nœuds, mais recherchera la meilleure fonction dans un sous-ensemble aléatoire de fonctions. Ce processus produit une grande diversité et généralement de meilleurs modèles peuvent être obtenus.

### II.2.2.3. Les machines à vecteur de support SVM:

La machine à vecteurs de support est un algorithme d'apprentissage automatique supervisé, les Support Vector Machine sont devenus des outils indispensables dans le domaine du Machine Learning en raison de leurs propriétés de généralisation et de leur capacité de régularisation. La machine à vecteurs de support peut être divisée en modèles linéaires et non linéaires. Il est appelé machine à vecteurs de support linéaire si le domaine de données peut être divisé linéairement (par exemple, ligne droite ou hyperplan) pour séparer les classes dans le domaine d'origine. Si le domaine de données ne peut pas être divisé linéairement, et s'il peut être transformé en un espace appelé espace de fonctionnalités où le domaine de données peut être divisé linéairement pour séparer les classes, alors il est appelé support non linéaire [11].

### II.2.2.4. Algorithme de Naïves Bayes:

La méthode de classification Naïve Bayésienne est un algorithme d'apprentissage supervisé qui permet de classer un ensemble d'observations selon des règles déterminées par l'algorithme lui-même. Cet outil de classification doit dans un premier temps être entraîné sur un jeu de données d'apprentissage qui montre la classe attendue en fonction des entrées. Pendant la phase d'apprentissage, l'algorithme élabore ses règles de classification sur ce jeu de données, pour les appliquer dans un second temps à la classification d'un jeu de données de prédiction. Le classificateur Naïves Bayes implique que les classes du jeu de données d'apprentissage soit connu et fournit, d'où le caractère supervisé de l'outil. Ces atouts les plus significatifs : on citera son apprentissage rapide qui ne nécessite pas un gros volume de données et son extrême rapidité d'exécution comparé à d'autres méthodes plus complexes [12].

### II.2.2.5. Algorithme de K plus proche voisins:

L'algorithme K-Nearest Neighbors (ou K-NN) est l'un des algorithmes de classification les plus simples et l'un des algorithmes d'apprentissage les plus utilisés. K-NN est un algorithme d'apprentissage paresseux, son but est d'utiliser une base de données dans laquelle les points de données sont séparés en plusieurs classes pour prédire la classification d'un nouveau point d'échantillonnage. Une technique est non paramétrique, cela signifie qu'elle ne fait aucune hypothèse sur la distribution de données. En d'autres termes, la structure du modèle est déterminée à partir des données [13].

### II.3. L'apprentissage profond (approfondi) :

L'apprentissage approfondi (deep learning) est un domaine de recherche sur l'apprentissage automatique basé sur un type particulier de mécanisme d'apprentissage. Il est caractérisé par l'effort de créer un modèle d'apprentissage à plusieurs niveaux, dans lequel les niveaux les plus profonds prennent en compte les résultats des niveaux précédents, les transformant et en faisant toujours plus d'abstraction. Cet aperçu des niveaux d'apprentissage est inspiré par la façon dont le cerveau traite l'information et apprend en réagissant aux stimuli externes. Chaque niveau d'apprentissage correspond, par hypothèse, à l'une des différentes zones qui composent le cortex cérébral [7].

Le Deep Learning est basé sur l'idée des réseaux de neurones artificielles et il est taillé pour gérer de larges quantités de données en ajoutant des couches au réseau. Un modèle de deep learning a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petit à petit à travers chaque couche avec une intervention humaine minimale [14] [15] [16] [17] [18] [19] [20] .

#### II.3.1. Histoire d'apprentissage approfondi :

Année	Contributeur	Contribution
300AC	Aristotle	Introduction de l'associationnisme, début de l'histoire des humains qui essayent de comprendre le cerveau
1873	Alexander Bain	Introduction du Neural Groupings comme les premiers modèles de réseaux de neurones
1943	McCulloch and Pitts	Introduction du McCulloch–Pitts (MCP) modèle considéré comme l'ancêtre des réseaux de neurones artificielles
1949	Donald Hebb	Considérer comme le père des réseaux de neurones, il introduit la règle d'apprentissage de Hebb qui servira de fondation pour les réseaux de neurones modernes
1958	Frank Rosenblatt	Introduction du premier perceptron
1974	Paul Werbos	Introduction de la retro propagation
1980	Teuvo Kohonen	Introduction des cartes auto organisatrices

1980	Kunihiko Fukushima	Introduction du Neocognitron, qui a inspiré les réseaux de neurone convolutif
1982	John Hopfield	Introduction des réseaux de Hopfield
1985	Hilton and Sejnowski	Introduction des machines de Boltzmann
1986	Paul Smolensky	Introduction de Harmonium, qui sera connu plus tard comme machines de Boltzmann restreintes
1986	Michael I. Jordan	Définition et introduction des réseaux de neurones récurrents
1990	Yann LeCun	Introduction de LeNet et montra les capacités des réseaux de neurones profond
1997	Schuster and Paliwal	Introduction des réseaux de neurones récurrents bidirectionnels
1997	Hochreiter and Schmidhuber	Introduction de LSTM, qui ont résolu le problème du vanishing gradient dans les réseaux de neurones récurrent
2006	Geoffrey Hinton	Introduction des Deep belief Network
2009	Salakhutdinov and Hinton	Introduction des Deep Boltzmann Machines
2012	Alex Krizhevsky	Introduction de AlexNet qui remporta le challenge ImageNet

**Table II.1:** Les étapes majeures du Deep Learning [21].

### II.3.2. Pourquoi l'apprentissage profond:

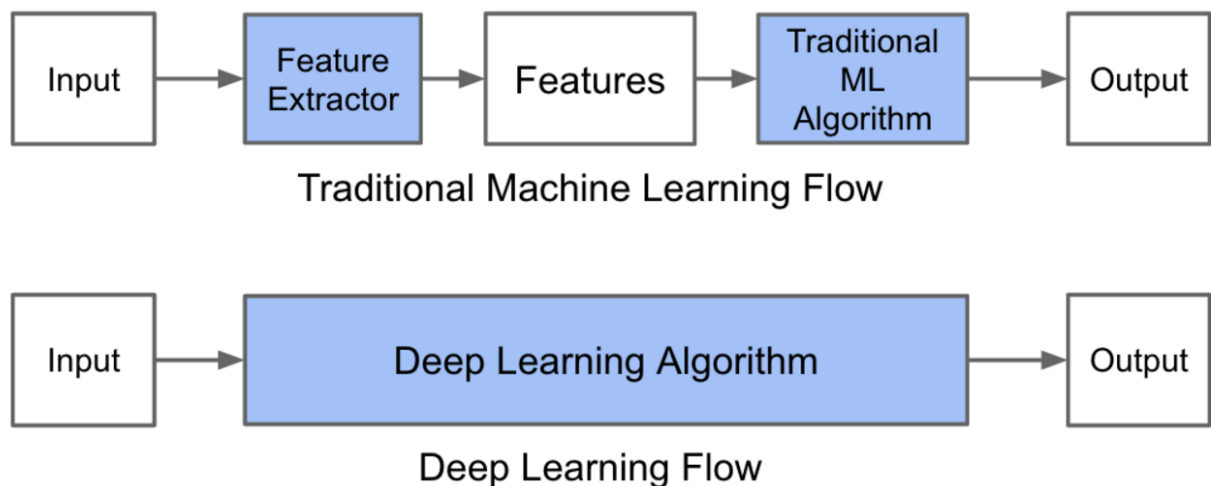
Les algorithmes de ML décrits déjà est applicable à divers problèmes. Cependant, ils n'ont pas réussi à résoudre certains problèmes d'IA tels que reconnaissance vocale et reconnaissance d'objets.

Le développement du DL est en partie dû à l'échec des algorithmes traditionnels d'IA. Mais ce n'est qu'après que de plus grandes quantités de données ne soit disponibles grâce notamment au Big Data et aux objets connectés et que les machines de calcul soient devenues plus puissantes qu'on a pu comprendre le potentiel réel d'apprentissage profond.

Une des grandes différences entre DL et ML est que la bonne adaptabilité, plus la quantité de données fournies est grande, meilleures sont les performances de l'algorithme d'apprentissage profond. Par contre plusieurs algorithmes de ML possèdent une borne supérieure à la quantité de données qu'ils peuvent recevoir des fois appelée "plateau de performance", les modèles de DL n'ont pas de telles

limitations (théoriquement) et ils sont même allés jusqu'à dépasser la performance humaine dans des domaines comme l'image processing.

Une autre différence s'agit d'étape d'extraction de caractéristiques. Dans les algorithmes de ML traditionnelles l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en Deep Learning cette étape est exécutée automatiquement par l'algorithme.



**Figure II.4:** Le procédé du ML classique comparé à celui du Deep Learning [22].

### II.3.3. Les modèles d'apprentissage profondi:

#### II.3.3.1. Les réseaux de neurones artificiels (ANN) :

Ce sont les plus simples et sont souvent utilisés car ils trient bien les informations. Le premier exemple d'un réseau neuronal a été appelé le perceptron, qui a été inventé par Frank Rosenblatt en 1957. Le perceptron est un réseau composé seulement d'une couche d'entrée et d'une couche de sortie. Dans le cas de classifications binaires, la couche de sortie n'a qu'un seul neurone ou unité. Le perceptron a semblé très prometteur dès le début, bien qu'il ait rapidement compris qu'il ne pouvait apprendre que des motifs linéairement séparables. Par exemple, Marvin Minsky et Seymour Papert ont montré qu'ils ne pouvaient pas apprendre la fonction logique XOR.

Dans ses représentations les plus basiques, les perceptrons ne sont que de simples représentations d'un neurone et son entrée (entrée qui peut être composée de plusieurs neurones). Etant donné les différentes entrées dans un neurone, nous

définissons une valeur d'activation par la formule ( $a(x) = \sum_i w_i x_i$ ) où  $x_i$  est la valeur pour le neurone d'entrée, tandis que  $w_i$  est la valeur de la connexion entre le neurone  $i$  et la sortie. Si la valeur d'activation, qui devrait être considérée comme l'état interne du neurone, est supérieure à un seuil fixe  $b$ , alors le neurone s'activera, c'est-à-dire qu'il se déclenchera, sinon ce ne sera pas le cas [23].

### II.3.3.1.1. Types de fonction d'activation :

Nous identifions 3 types de fonctions d'activation de base : fonctions de seuil ou fonctions Heaviside, fonctions linéaires par morceaux et fonctions sigmoïdes.

- **Les fonctions de seuil :**

$$\Phi(v) = 1 \text{ si } v \geq 0$$

$$\Phi(v) = 0 \text{ si } v < 0$$

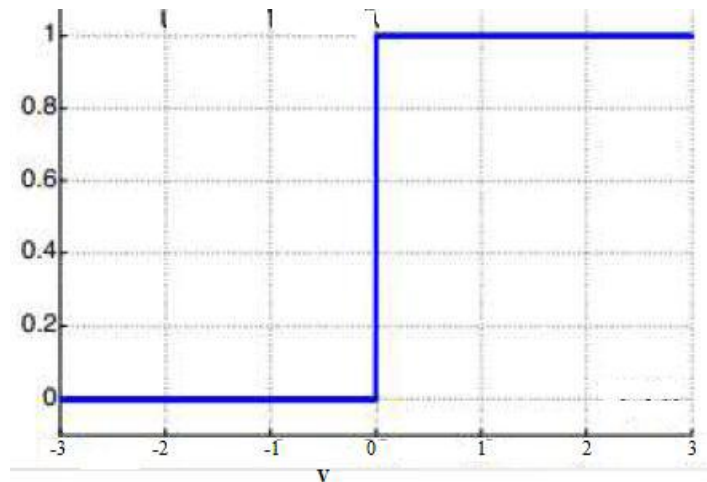


Figure II.5: Fonctions de seuil [9].

- **Les fonctions linéaires** sont les suivantes :

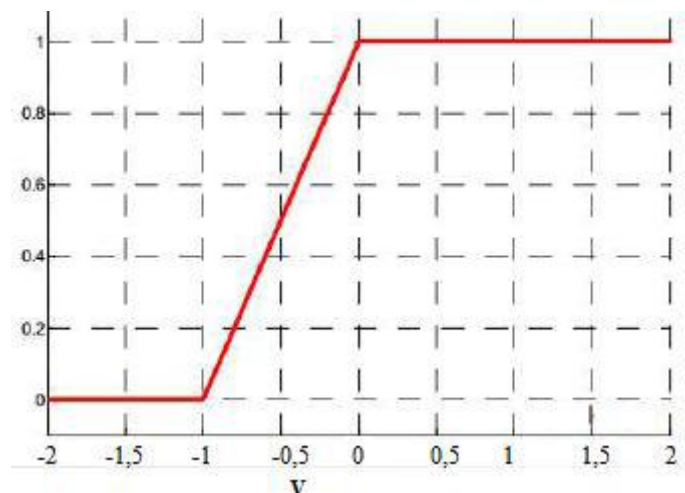


Figure II.6: Fonctions linéaires [9].

- **Les fonctions sigmoïdes** sont les fonctions les plus utilisées dans la création de réseaux neurones artificiels. C'est une fonction strictement croissante qui présente un équilibre entre le comportement linéaire et non linéaire. Par exemple :  $\varphi(v) = \frac{1}{1+e^{-av}}$  « a » est un paramètre qui indique la pente de la fonction [9] [24] [25] .

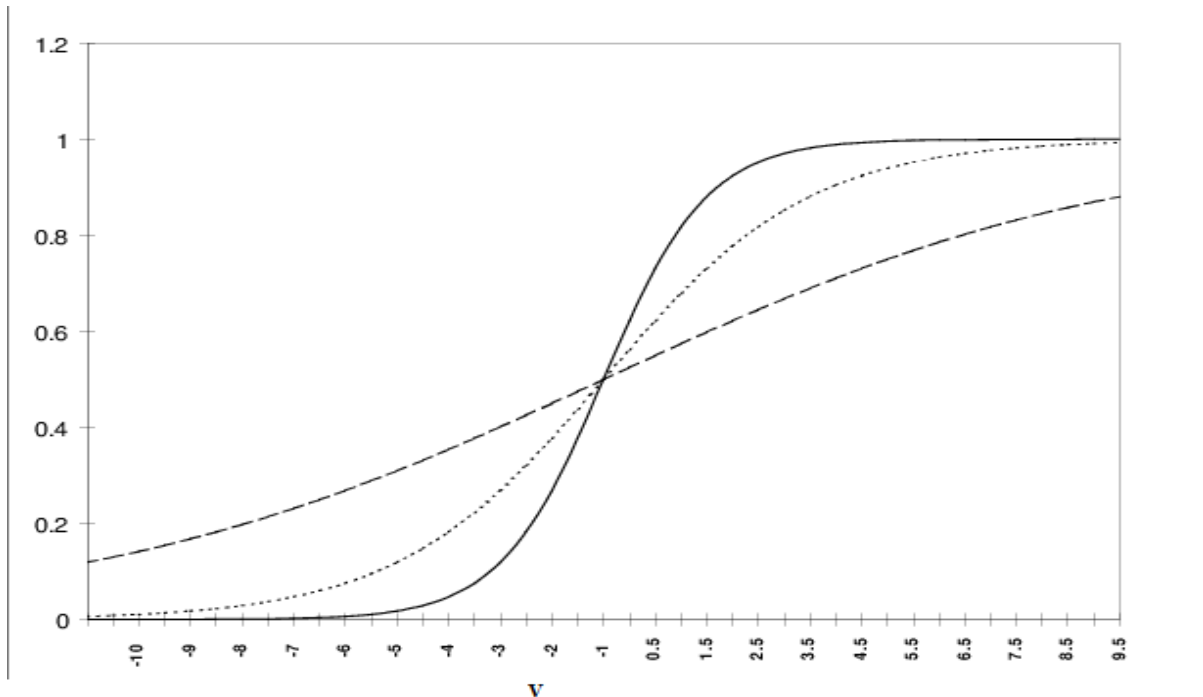
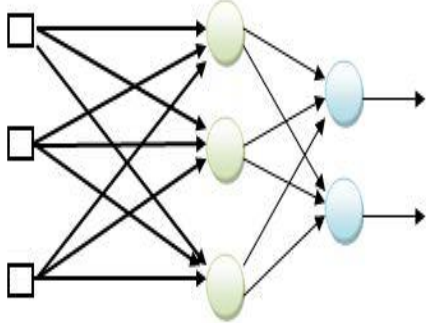
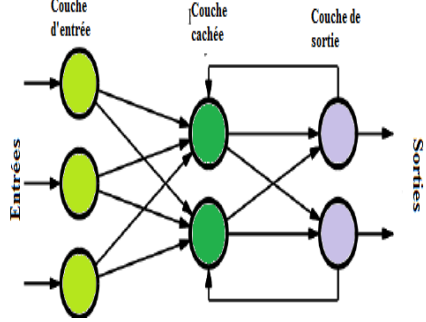


Figure II.7:Fonctions sigmoïdes [9].

### II.3.3.1.2. Types d'architectures de réseaux :

Type	Description	Illustration
<b>Réseaux à une couche (feed-forward)</b>	Dans cette forme simple de réseau en couches, nous avons des nœuds d'entrée et une couche de neurones (couche de sortie). Le signal se propage dans le réseau de manière linéaire, en commençant par la couche d'entrée et en se terminant par la couche de sortie. Il n'y a pas de connexions qui reviennent et pas de connexions transversales dans la couche de sortie [24].	<p><b>Figure II.8:</b>Réseau mono-couche .</p>

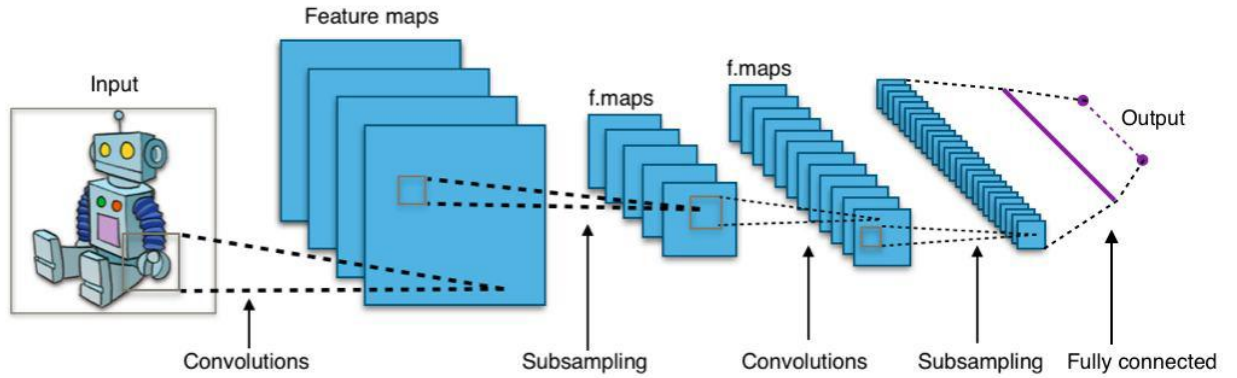
<p><b>Réseaux multi-couches (feed-forward)</b></p>	<p>Cette classe de réseaux feed-forward diffère de la précédente car elle a une ou plusieurs couches de neurones cachés (couches cachées) entre les couches d'entrée et de sortie. Chaque couche à des connexions entrantes de la couche précédente et sortantes dans la suivante, donc la propagation du signal se fait de façon linéaire sans cycles et sans connexions transversales. Ce type d'architecture fournit au réseau une perspective globale car il augmente les interactions entre les neurones [9] [24] [25].</p>	 <p><b>Figure II.9:</b>Réseau multi-couche.</p>
<p><b>Réseaux récurrents (feed-back)</b></p>	<p>Un réseau récurrent diffère des précédents par le fait qu'il est cyclique. La présence de cycles a un impact profond sur les capacités d'apprentissage du réseau et sur ses performances, notamment rendre le système dynamique [9] [24].</p>	 <p><b>Figure II.10:</b>Réseaux récurrents.</p>

**Table II.2:** types d'architectures de réseaux.

### II.3.3.2. Les réseaux de neurones Convolutifs (CNN) :

Les réseaux convolutifs ont été introduits pour la première fois par Fukushima [26], il a dérivé une architecture du réseau nerveux hiérarchique inspirée par le travail de recherche de Hubel [27]. Lecun [28], les a généralisés pour classer les chiffres avec succès et pour reconnaître les numéros de contrôle manuscrit par LeNet-5 qui est montré à (la figure II.11). .Ciresan [29] a utilisé les réseaux convolutifs et réalisé les meilleures performances dans la littérature pour la reconnaissance d'objets multiples pour des bases de données d'images : MNIST, NORB, CIFAR10 et l'ensemble de données ImageNet.

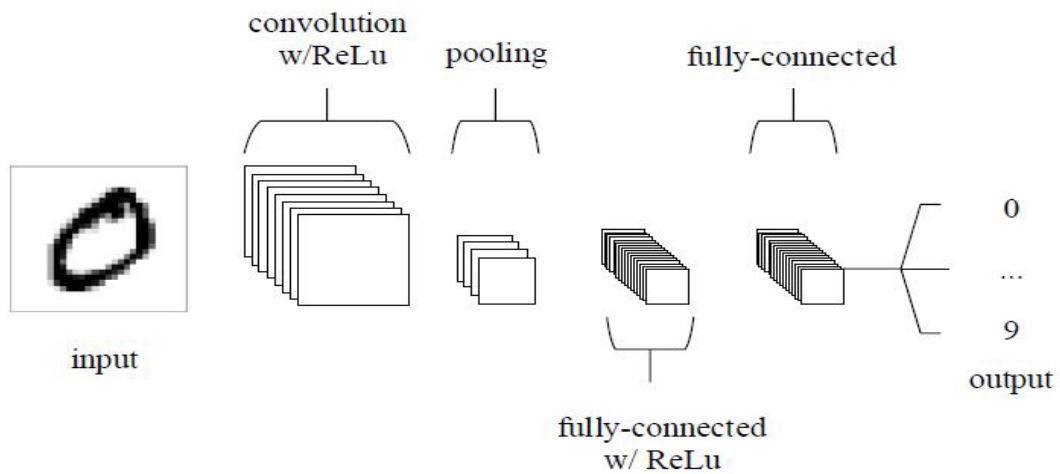
Les réseaux neuronaux convolutifs diffèrent des autres formes de réseaux neuronaux artificiels en ce sens. Qu'au lieu de se concentrer sur l'intégralité du domaine problématique, les connaissances sur le type spécifique d'entrées sont exploitées, cela permet à son tour de mettre en place une architecture réseau beaucoup plus performante [30] [31] [32].



**Figure II.11:** Réseau neuronal convolutif [33].

### II.3.3.2.1. Architecture globale de CNN :

La figure II.12 illustre une architecture CNN simplifiée pour la classification MNIST.



**Figure II.12:** Architecture simple de CNN [33].

#### II.3.3.2.1.1. Couche convolutive:

La couche convolutive forme le cœur du réseau convolutif. Ces couches se composent d'une grille rectangulaire de neurones avec un petit champ de réception qui s'étend jusqu'à la profondeur de tout le volume d'entrée. Donc, la couche convolutive est juste une convolution d'image de la couche précédente, où le poids spécifie le filtre de convolution.

La couche convolutive déterminera la sortie des neurones connectés à la région locale d'entrée en calculant le produit scalaire entre le poids et la région connectée au volume d'entrée. ReLu vise à appliquer des fonctions d'activation «élémentaire» telles que La fonction sigmoïde apparaît à la sortie de l'activation de la couche précédente [30] [31] [32].

#### II.3.3.2.1.2. Couche de pooling:

Après chaque couche convolutive, il peut y avoir une couche de regroupement (pooling). Sous cette couche échantillonner le long de la dimension spatiale d'une entrée donnée, ce qui réduira le nombre de paramètres dans cette activation. Il y a plusieurs façons de faire cette combinaison, comme prendre la valeur moyenne ou maximale, ou une combinaison linéaire prise par des neurones dans le bloc. Par exemple (la figure II.13) montre le max pooling sur une fenêtre 2x 2 [20] [21] [22].

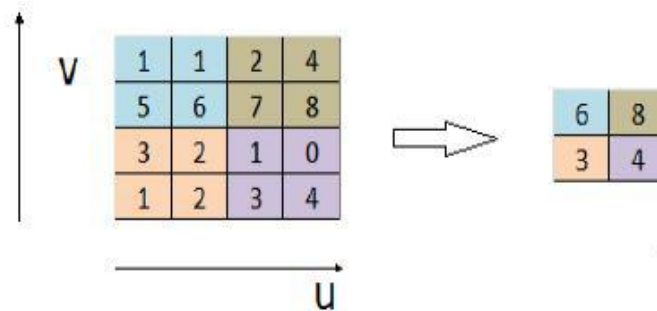


Figure II.13:Max pooling [33].

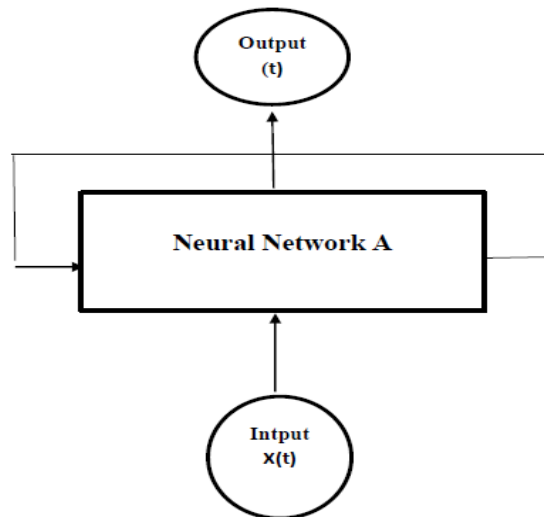
#### II.3.3.2.1.3. Couche totalement connectée:

Dans un réseau neuronal convolutif, chaque couche agit comme un filtre de détection s'il existe des caractéristiques spécifiques dans les données originales. Les premières couches du réseau détectent des caractéristiques qui peuvent être reconnues et interprétées facilement. Les couches suivantes sont de plus en plus détectées des caractéristiques plus abstraites. La dernière couche du réseau convolutif est puissant de faire une classification ultra-spécifique en combinant toutes les caractéristiques spécifiques détecté par la couche précédentes dans les données d'entrée.

Les couches totalement connectées font les mêmes tâches que celles des ANN standard et tenteront de produire des notes de classe à partir des activations, pour les utiliser pour la classification. Il est également suggéré d'utiliser ReLu entre ces couches pour améliorer les performances [30] [31] [32].

### II.3.3.3. Les réseaux de neurones Récurrents (RNN) :

Un réseau de neurones récurrent (Récurrent Neural Network - RNN) est un réseau de neurones dont le graphe de connexion contient au moins un cycle. Tout comme les systèmes de contrôle, les réseaux neurones récurrents fonctionnent selon le principe du bouclage et du chaînage, qui peut être élaboré comme suit:



**Figure II.14:** Réseau de neurone récurrent.

À partir de (la figure II.14) considérez que A est une petite partie du réseau de neurone. Ce bloc requiert des valeurs d'entrée pour démarrer les calculs et le traitement et produire des valeurs de sortie. Dans ce cas, les valeurs sont  $x(t)$  et  $h(t)$ . Le réseau de neurone A forme également une boucle, qui consiste essentiellement en plusieurs copies du même réseau et permet le partage des connaissances et des informations entre les étapes suivantes du réseau. Par conséquent cette caractéristique du partage des connaissances distingue les RNN des autres techniques de réseau de neurone. De plus, l'approche en chaîne des RNN leur permet de ressembler avec la séquence et la liste [34].

### II.3.3.4. Les réseaux Long Short-Term Memory (LSTM) :

Les réseaux Long Short-Term Memory généralement appelés simplement «LSTM» sont un type spécial de RNN, capable d'apprendre les dépendances à long terme. Ils ont été introduits par Hochreiter & Schmidhuber (1997).

**II.3.3.5. RBM:**

Inventée sous le nom d'Harmonium en 1986 par Maul Smolenski, c'est un réseau neurones artificiels utilisé pour avoir une estimation de la distribution d'un jeu de données [35].

**II.3.3.6. DBM:**

Est un modèle graphique génératif, ou bien une classe de réseaux de neurones profonds, composée de plusieurs couches de variables latentes ("unités cachées"), avec des connexions entre les couches mais pas entre chaque couche [36].

**II.4. Conclusion:**

Au cours de ce chapitre, j'ai présenté brièvement l'apprentissage automatique avec ses types, ses applications, ses différents algorithmes. Ensuite j'ai expliqué brièvement le principe de l'apprentissage profond, ainsi qu'un bref historique, ses différents modèles et ses avantages.



CHAPITRE III:  
ENVIRONNEMENT  
DE  
ANACONDA®  
PROGRAMMATION



### III.1. Introduction:

Après avoir étudié dans la deuxième partie l'IA, ML et DL, Nous allons présenter dans cette partie une étude approfondie sur l'environnement de programmation et les outils utilisés.

### III.2. Présentation des outils:

#### III.2.1. Le hardware :

L'apprentissage profond est un domaine avec de fortes exigences informatiques et une disponibilité des ressources (Surtout dans le GPU) dédié à cette tâche vont fondamentalement influencer sur l'expérience de l'utilisateur car sans ses ressources, apprendre de ses erreurs peut prendre beaucoup de temps.

Les expérimentations ont tous été effectuées sur une machine qui offre des performances acceptables dont les caractéristiques sont :

- ❖ Un PC portable HP i5-3600U CPU @ 2.40GHZ 2.50 GHZ
- ❖ RAM de taille 8 GO
- ❖ Disque dur de taille 500 GO
- ❖ Carte graphique Intel(R) HD Graphics 520
- ❖ Système d'exploitation 64 bits Windows 10

#### III.2.2. Les outils et les bibliothèques utilisés:

##### III.2.2.1. Python:

Python est un langage de programmation de haut niveau interprété (aucune étape de compilation) et orienté objet avec une sémantique dynamique. Un grand nombre de développeurs et de programmeurs ont une forte demande. Python est un langage simple, facile à apprendre et peut réduire considérablement les coûts de maintenance du code. Les bibliothèques (packages) Python encouragent la modularité et la réutilisabilité du code. Python et ses bibliothèques sont disponibles gratuitement (en code source ou en fichiers binaires) sur la plupart des plateformes et peuvent être redistribués gratuitement [37].

##### III.2.2.2. ANACONDA:

Anaconda est une distribution Python. A son installation, Anaconda installera Python ainsi qu'une multitude de packages. Cela nous évite de nous ruer dans les problèmes d'incompatibilités entre les différents packages.

### III.2.2.2.1. Installation d'Anaconda (la version utilisée):

#### A. Téléchargement :

- Chercher dans Google « Anaconda Python » → Python V 3.6 pour Windows 64 bits.

#### B. Installation :

- Installer le fichier d'installation Anaconda 64-bit pour Windows (Install for → option: Just for me)
- Ajouter le chemin d'installation à l'environnement des variables (Rubrique: PATH) pour faciliter l'accès.

#### C. Lancement et mise à jour d'anaconda:

- Après la fin de l'installation ouvrir Anaconda Prompt (C:\Users\...\Anaconda3).
- Ouvrez Anaconda Prompt en mode ADMINISTRATEUR. Ceci vous évitera des erreurs de droits en écritures par la suite lors de l'installation des bibliothèques / paquets. (anaconda prompt → bouton droit → menu → mode administrateur → C:WINDOWS\system32>).
- Modifier les packages d'anaconda en tapant les deux commandes :
  - >conda update conda.
  - >conda update -all.

#### D. Installation des accessoires nécessaires de mon travail :

- C:\>conda install jupyter → jupyter 1.0.
- C:\>conda install scipy → la bibliothèque des tableau Scipy.
- C:\>pip install sklearn → la bibliothèque de machine learning.
- C:\>pip install pandas → la bibliothèque de machine learning.
- C:\>pip install pandas-datareader → outils de lecture de bibliothèque ML pandas.
- C:\>pip install matplotlib → installer l'outil de traçage de courbes.
- C:\>pip install pillow → Outils complémentaire de ML.
- C:\>pip install requests → Outils complémentaire de ML.
- C:\>pip install h5py → Outils complémentaire de ML.
- C:\>pip install tensorflow/tensorflow ==1.2.1 → Installer la bibliothèque de machine learning et l'interface google tensorflow.
- C:\>pip install keras/keras==2.0.6 → Installer la bibliothèque de machine learning et l'interface keras.

### III.2.2.3. TensorFlow:

Nous avons utilisé dans notre travail TensorFlow qui est un framework de programmation pour le calcul numérique qui a été rendu open source par Google en Novembre 2015. Depuis son release, TensorFlow n'a cessé de gagner en popularité, pour devenir très rapidement l'un des frameworks les plus utilisés pour le Deep Learning, comme le montrent les dernières comparaisons suivantes, faites par François Chollet (auteur de la librairie Keras).

Bien qu'il ait initialement été développé pour optimiser les calculs numériques complexes, TensorFlow est aujourd'hui particulièrement utilisé pour le Deep Learning, et donc les réseaux de neurones. Son nom est notamment inspiré du fait que les opérations courantes sur des réseaux de neurones sont principalement faites via des tables de données multi-dimensionnelles, appelées Tenseurs (Tensor). Un Tensor à deux dimensions est l'équivalent d'une matrice.

Aujourd'hui, les principaux produits de Google sont basés sur TensorFlow: Gmail, Google Photos, Reconnaissance de voix, etc [38].

#### III.2.2.3.1. L'environnement tensorflow d'après anaconda:

1. Créer l'environnement de Tensorflow → `>conda create -n tensorflow`.
2. Activer l'environnement de Tensorflow créée → `> conda activate tensorflow`

### III.2.2.4. Keras:

Keras est une API de réseau neuronal avancée écrite en Python qui peut s'exécuter sur TensorFlow ou Theano. Son développement se concentre sur l'expérimentation rapide. Être capable de passer de l'idée au résultat en un minimum de temps est la clé d'une bonne recherche. Il fait partie des travaux de recherche du projet ONEIROS (Open Neuroelectronic Intelligent Robot Operating System) dont le principal auteur et mainteneur est l'ingénieur Google François Chollet [39].

### III.2.2.5. Quelques Bibliothèques :

1. **Numpy:** est une bibliothèque Python très populaire pour le traitement de matrices et de tableaux multidimensionnels, à l'aide d'une vaste collection de fonctions mathématiques de haut niveau. C'est très utile pour les calculs scientifiques fondamentaux en Machine Learning. Il est particulièrement utile pour l'algèbre linéaire, la transformée de Fourier et les capacités de nombres

- aléatoires. Les bibliothèques haut de gamme telles que TensorFlow utilisent Numpy en interne pour la manipulation de Tensors [40].
2. **Scipy:** est une bibliothèque très populaire parmi les passionnés d'apprentissage automatique car elle contient différents modules d'optimisation, d'algèbre linéaire, d'intégration et de statistiques. Il existe une différence entre la bibliothèque Scipy et la pile Scipy. Le SciPy est l'un des principaux packages qui composent la pile Scipy. Scipy est également très utile pour la manipulation d'images [40].
  3. **Scikit-learn:** est l'une des bibliothèques ML les plus populaires pour les algorithmes ML classiques. Il est construit sur deux bibliothèques Python de base, à savoir, Numpy et Scipy. Scikit-learn supporte la plupart des algorithmes d'apprentissage supervisé et non supervisé. Scikit-learn peut également être utilisé pour l'exploration et l'analyse de données, ce qui en fait un excellent outil pour les débutants en ML [40].
  4. **Pandas:** est une bibliothèque Python populaire pour l'analyse de données. Ce n'est pas directement lié à l'apprentissage automatique. Comme nous savons que le jeu de données doit être préparé avant la formation. Dans ce cas, les pandas sont pratiques car ils ont été développés spécifiquement pour l'extraction et la préparation de données. Il fournit des structures de données de haut niveau et de nombreux outils pour l'analyse des données. Il fournit de nombreuses méthodes intégrées pour tâtonner, combiner et filtrer les données [40].
  5. **Matplotlib :** est une bibliothèque Python très populaire pour la visualisation de données. Comme les pandas, il n'est pas directement lié à l'apprentissage automatique. Cela s'avère particulièrement utile lorsqu'un programmeur souhaite visualiser les modèles dans les données. C'est une bibliothèque de tracé 2D utilisée pour créer des graphiques et des tracés 2D. Un module appelé pyplot facilite le traçage des programmeurs car il offre des fonctionnalités permettant de contrôler les styles de trait, les propriétés de police, les axes de formatage, etc. Il fournit différents types de graphiques et de tracés pour la visualisation des données, l'affichage, l'histogramme, les graphiques d'erreur, les discussions en barres, etc [40].

6. **Pillow** : est une bibliothèque de traitement d'image, qui est un fork et successeur du projet PIL (Python Imaging Library). Elle est conçue de manière à offrir un accès rapide aux données contenues dans une image, et offre un support pour différents formats de fichiers tels que PPM, PNG, JPEG, GIF, TIFF et BMP [41].

### III.3. Base de données pima\_diabetes :

Nous avons utilisé une base de données constituée de 268 femmes diabétiques et 500 femmes non diabétiques. L'âge de ces femmes varie entre 21 et 81 ans et appartenant à la tribu des indiens Pima (près de Phoenix, Arizona).

Le but est de reconnaître la présence du diabète chez ces patients. Ces malades sont caractérisées par 8 attributs explicatifs :

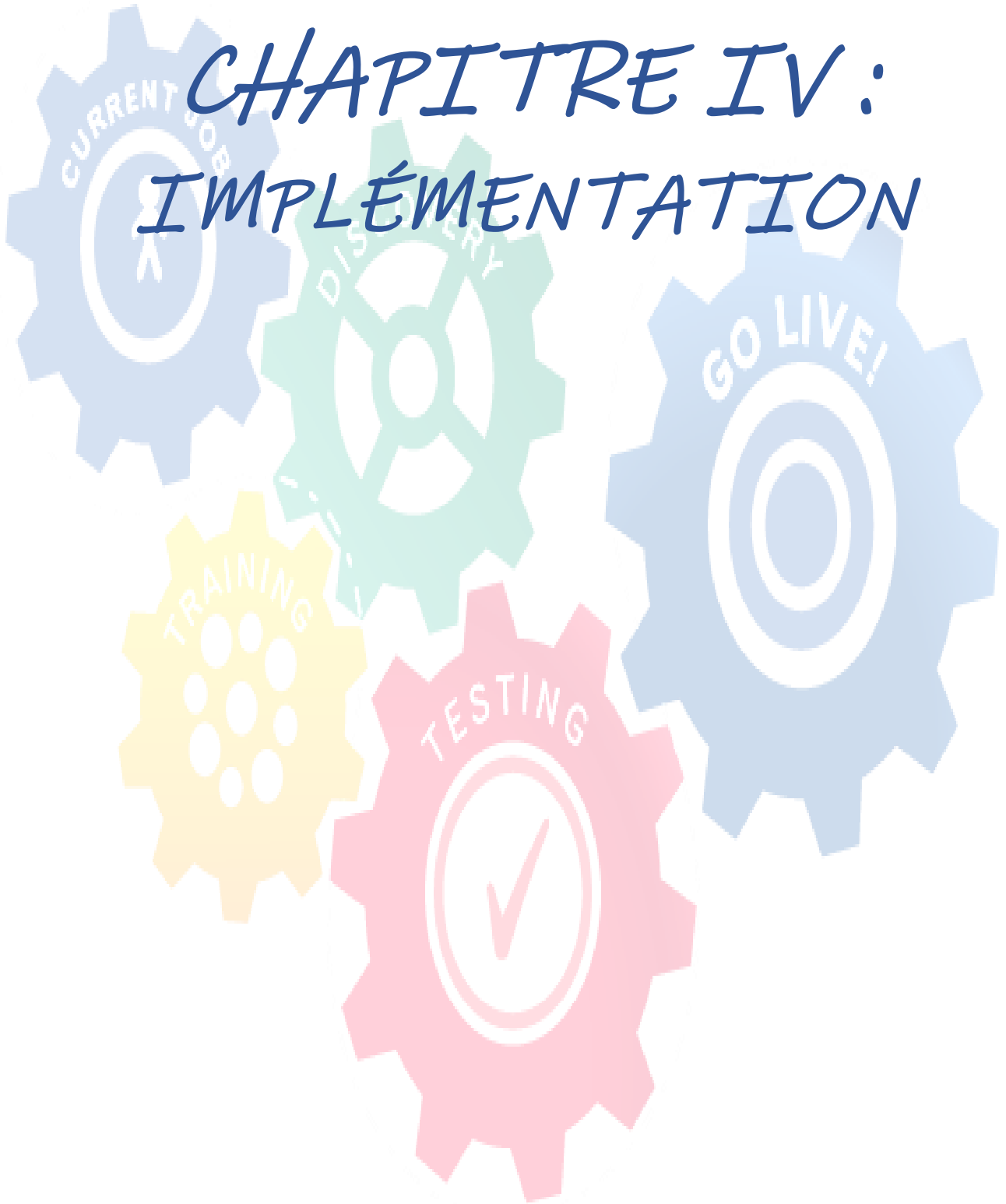
1. Nombre de grossesses (Ngross)
2. Concentration du glucose dans le plasma après un test de tolérance au glucose de 2 heures (mg/dl) (Gly)
3. Pression artérielle diastolique (mm Hg) (PAD)
4. Epaisseur de la peau au niveau du triceps (mm) (Epai)
5. Taux d'insuline au bout de 2 heures (mu U/ml) (INS)
6. Indice de masse corporelle (poids en kg/ m<sup>2</sup>) (IMC)
7. Fonction pédigrée du diabète (Ped)
8. Age (années)

La dernière colonne de l'ensemble de données indique si la personne a reçu un diagnostic de diabète ou. Les individus diabétiques appartiennent à la classe "1" (test positif), et les non diabétiques appartiennent à la classe "0" (test est négatif).

### III.4. Conclusion :

Dans ce chapitre nous avons décrit l'environnement de travail, notamment : les outils utilisés pour la réalisation de l'application, les bibliothèques de base, ainsi que la base de données utilisée exploitée.

# CHAPITRE IV : IMPLÉMENTATION



### IV.1. introduction:

Dans ce chapitre, on mettra en avant notre conception et on va montrer comment réaliser notre classifieur et créer notre modèle ANN.

### IV.2. Problématique étudiée :

Nous allons nous intéresser à la problématique de détection de diabète chez les patientes de notre base de données en se basant sur les valeurs des différents attributs cités auparavant.

### IV.3. Le modèle proposé :

Nous avons développé un classifieur binaire automatique qui permet d'identifier les personnes atteintes de diabète en fonction de certaines caractéristiques. Il s'agit donc d'une classification binaire en deux (02) classes données (apparition du diabète à 1 ou non à 0). Toutes les variables d'entrée qui décrivent chaque patient sont numériques. Nous notons également que pour construire le meilleur modèle prédictif, nous avons utilisé deux approches : l'approche classique ML et l'approche DL. Dans un premier temps, nous avons procédé à la classification en utilisant de nombreux algorithmes de ML, dont LR, LDA, KNN, CART, NB et SVM. Ensuite, nous avons proposé un modèle ANN composé de plusieurs couches simples : une couche d'entrée (100 neurones), quatre couches cachées (50, 30, 18,4 neurones), et enfin une couche de sortie (01 neurone). Comme outils de programmation, nous avons utilisé Python, Tensorflow et Keras qui sont les plus utilisés dans ce domaine. La figure IV.1 présente un diagramme détaillé du modèle ANN proposé afin d'améliorer la performance de la tâche de classification.

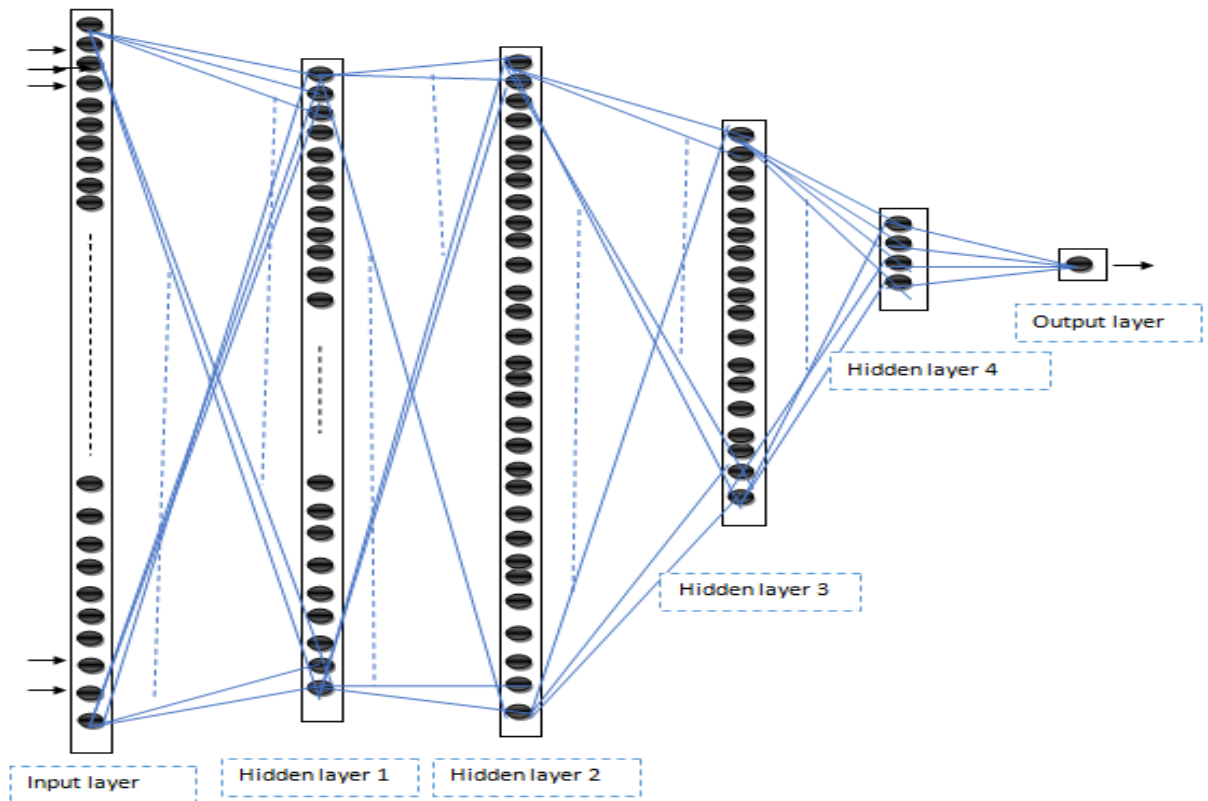


Figure IV.1: Architecture du modèle ANN proposé.

#### IV.4. Évaluation :

Afin de valider les différents algorithmes de ML, et d'obtenir le meilleur modèle, nous avons utilisé la méthode de validation croisée consistant à : diviser notre ensemble de données en 10 parties, s'entraîner sur 9 et tester sur 1 et répéter pour toutes les combinaisons de fractionnements train/test. Pour le modèle ANN, nous avons utilisé deux paramètres qui sont : la valeur de la perte (loss value) et la mesure de la (accuracy).

**1. Mesure de la précision (accuracy) :** il s'agit d'un rapport entre le nombre d'occurrences correctement prédites et le nombre total d'occurrences dans l'ensemble de données, multiplié par 100 pour obtenir un pourcentage (par exemple, précision de 90%).

**2. Valeur de perte (loss value):** utilisée pour optimiser un algorithme de ML ou un modèle de DL. Elle doit être calculée sur des ensembles de données de formation et de validation. Son interprétation simple est basée sur la performance de l'algorithme de ML ou du modèle construit de DL dans ces deux ensembles de

données. Elle donne la somme des erreurs commises pour chaque exemple dans l'ensemble d'apprentissage ou de validation.

#### IV.5. Résultats obtenus :

Pour construire le meilleur modèle prédictif, nous avons accompli deux tâches :

1. Appliquer de nombreux algorithmes ML, notamment : Régression logistique LR, Analyse discriminante linéaire LDA, KNN, Arbre de décision (variante CART), NB, Machine vecteur de support SVM. Pour ce faire, nous avons utilisé la bibliothèque Scikit-learn de python contenant les algorithmes d'apprentissage les plus connus.
2. Réalisation d'un modèle de réseau neurone (ANN) selon le processus suivant :
  - Nous avons proposé un modèle composé de six couches entièrement connectées, la première couche compte 100 neurones et attend 8 variables d'entrée. La deuxième couche cachée a 50 neurones, la troisième couche cachée a 30 neurones, la quatrième couche cachée a 18 neurones, la cinquième couche cachée a 4 neurones, et enfin, la couche de sortie a 1 neurone pour prédire la classe (apparition du diabète ou non).
  - Les six couches entièrement connectées sont définies à l'aide de la classe `Dense` de Keras. Nous pouvons spécifier le nombre de neurones dans la couche comme premier argument, la méthode d'initialisation comme second argument et spécifier la fonction d'activation en utilisant l'argument `activation`.
  - Nous initialisons les poids du réseau à un petit nombre aléatoire généré à partir d'une distribution uniforme ("uniforme"), dans ce cas entre 0 et 0,05 qui sont l'initialisation par défaut du poids uniforme dans Keras. Ou "normal" pour les petits nombres aléatoires générés à partir d'une distribution Gaussienne.
  - Nous utiliserons également la fonction d'activation ("Relu") sur les cinq premières couches et la fonction sigmoïde dans la couche de sortie.

- Nous utilisons un sigmoïde sur la couche de sortie pour nous assurer que la sortie de notre réseau est comprise entre 0 et 1 et qu'il est facile de la faire correspondre à une probabilité de classe 1 ou à une classification stricte de l'une ou l'autre classe avec un seuil par défaut de 0,5
- Nous compilons le modèle en utilisant les bibliothèques numériques efficaces de Keras sous les couvertures (the so-called backend) telles que TensorFlow. Le backend choisit automatiquement la meilleure façon de représenter le réseau pour la formation et la prévision de l'exécution sur votre matériel, tel que le CPU ou le GPU ou même distribué.
- Lors de la compilation, nous devons spécifier certaines propriétés supplémentaires requises lors de la formation du réseau. N'oubliez pas que l'entraînement d'un réseau consiste à trouver le meilleur ensemble de poids pour faire des prédictions pour ce problème.
- Nous devons spécifier la fonction de perte à utiliser pour évaluer un ensemble de poids, l'optimiseur utilisé pour rechercher différents poids pour le réseau et toute mesure optionnelle que nous souhaitons collecter et rapporter pendant l'entraînement. Ici, nous utiliserons la perte logarithmique, qui, pour un problème de classification binaire, est définie dans Keras comme "binary\_crossentropy".
- Nous utiliserons également l'algorithme efficace de descente de gradient "adam" pour la seule raison qu'il s'agit d'un défaut efficace.
- Enfin, comme il s'agit d'un problème de classification, nous collecterons et rapporterons la précision de la classification sous forme de métrique.
- Exécutez le modèle sur certaines données.
- Nous pouvons entraîner ou ajuster notre modèle sur nos données chargées en appelant la fonction d'ajustement (fit function) sur le modèle, le processus d'entraînement se déroulera pendant un nombre fixe d'itérations à travers l'ensemble de données appelé époques, que nous devons spécifier en utilisant l'argument n-

epochs. Nous pouvons également définir le nombre d'instances qui sont évaluées avant qu'une mise à jour du poids dans le réseau ne soit effectuée, appelée batch size et définie en utilisant l'argument `batch_size`. Pour ce problème, nous fonctionnerons pour un petit nombre d'itérations (350) et utiliserons une taille de lot relativement petite de 25. Celles-ci peuvent être choisies expérimentalement par tâtonnement.

- Nous avons formé notre réseau neuronal sur l'ensemble des données et évalué ses performances sur une partie du même ensemble de données (l'ensemble de données de test) en utilisant la fonction `evaluate()`. Cette fonction permet de générer une prédiction pour chaque paire d'entrée et de sortie et de collecter les scores, y compris la perte moyenne et toute mesure que vous avez configurée, telle que la précision.

Les tableaux suivants résument les résultats obtenus sur les algorithmes ML et le modèle ANN

Algorithme	Accuracy
LR	0.760224
LDK	0.763823
K NN	0.725499
CART	0.722232
NB	0.758590
SVM	0.644162

**Table IV.1:**Précision (accuracy) obtenue après l'application de différents algorithmes de ML.

Couche (type)	Forme de la sortie	Paramètres
Dense_1 (Dense)	(None, 100)	900
batch_normalization_1 (Batch	(None, 100)	400
Dense_2 (Dense)	(None, 50)	5050
batch_normalization_2 (Batch	(None, 50)	200
Dense_3 (Dense)	(None, 30)	1530
batch_normalization_3 (Batch	(None, 30)	120
Dense_4 (Dense)	(None, 18)	558
batch_normalization_4 (Batch	(None, 18)	72
Dense_5 (Dense)	(None, 4)	76
batch_normalization_5 (Batch	(None, 4)	16
Dense_6 (Dense)	(None, 1)	5
Total des paramètres	8,927	
Trainable parameters	8,523	
Non-trainable parameters	404	

**Table IV.2:** Description du model proposé.

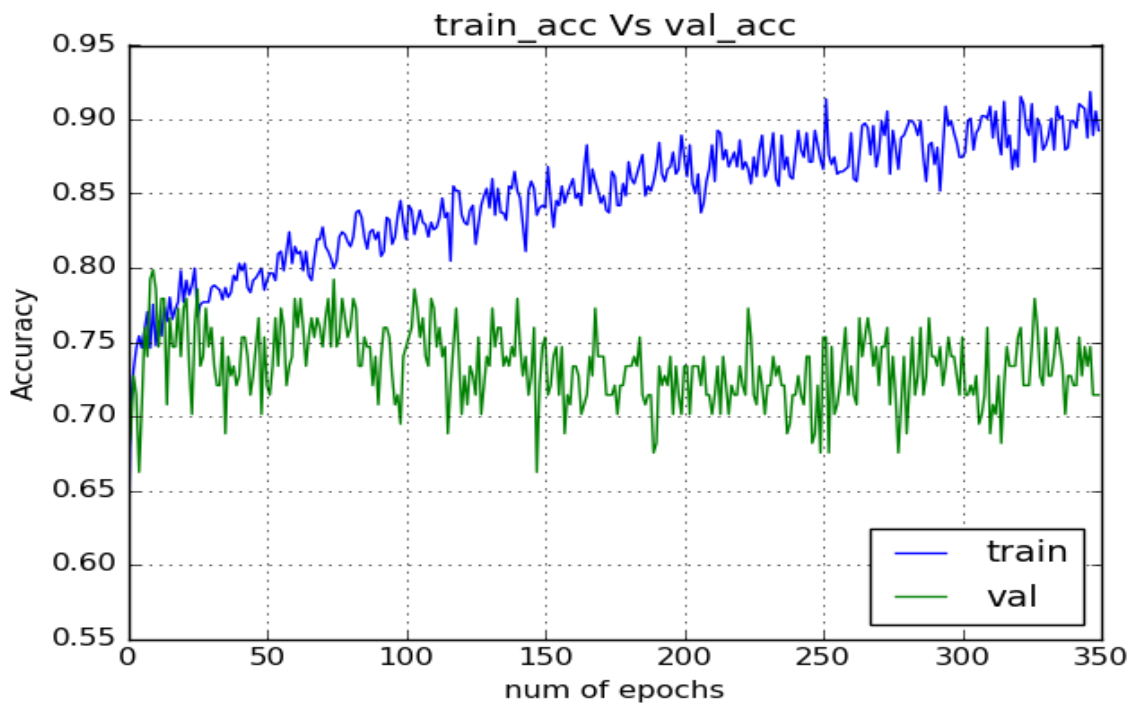
<b>ANN (100, 50, 30, 18, 4,1)</b>		
Training	Loss: 0.27	- acc: 0.8925
Test	Loss: 0.25	Accuracy: 91.41%

**Table IV.1:** les valeurs de perte et de précision obtenues lors de l'application du modèle proposé.

## Courbes



**Figure IV.2:** Training loss Vs Validation loss du modèle ANN.



**Figure IV.3:** Training Accuracy Vs Validation Accuracy du modèle ANN.

## IV.6. Discussions :

Le tableau Tab.IV.1 résume les résultats obtenus lors de l'application des différents algorithmes de ML, notamment LR, LDA, KNN, DT (CART), NB, SVM. Nous observons que LR, LDA, KNN, CART et NB donnent une précision de classification élevée ( $> 70\%$ ), tandis que SVM donne une valeur de précision relativement faible (64%) par rapport aux algorithmes précédents.

Le tableau Tab.IV.2 décrit l'architecture détaillée du modèle ANN proposé, qui est composé des couches suivantes : la première couche compte 100 neurones et attend 8 variables d'entrée. La deuxième couche cachée a 50 neurones, la troisième couche cachée a 30 neurones, la quatrième couche cachée a 18 neurones, la cinquième couche cachée a 4 neurones, et enfin, la couche de sortie a 1 neurone pour prédire la classe (apparition ou non du diabète).

Le tableau Tab.IV.3 présente les résultats obtenus lors de l'application du modèle ANN proposé sur l'ensemble de formation et l'ensemble de test. Deux mesures de performance sont considérées dans ce cas, la valeur de perte qui calcule la somme des erreurs après l'entraînement du modèle, et la valeur de précision qui donne le taux de correction. Il est clair que la valeur de perte est très faible par rapport à la précision qui est très élevée et dépend de la taille de l'ensemble utilisé. C'est la raison pour laquelle la précision de l'ensemble de formation est plus élevée que la précision de l'ensemble de test.

De la même manière, (la figure IV.2) montre l'évaluation de la perte de formation et de la perte de validation au fil du temps et en fonction du nombre d'époques. Elle commence très haut pour l'ensemble de formation et se termine très bas en raison du grand nombre d'échantillons, mais sa variation pour l'ensemble de validation n'est pas très rapide et semble relativement stable.

De même, (la figure IV.3) montre l'évolution de la précision de l'ensemble d'entraînement et de la précision de l'ensemble de validation en fonction du nombre d'époques. Contrairement à la valeur de perte, la précision commence très bas et finit très haut. Cette propriété est plus claire avec l'ensemble d'apprentissage en raison de sa grande taille.

Nous pouvons également voir sur les figures (figure IV.2 et figure IV.3) qui représentant successivement la valeur de perte et la valeur de précision que les deux valeurs de perte et de précision ne progressent pas continuellement au cours des

époques, mais augmentent et diminuent jusqu'à se stabiliser sur une valeur spécifique.

#### IV.7. Quelques interfaces de l'application :

The screenshot shows the 'Diabetes Prediction' application window. On the left, under 'Détails du patient', there is a form with the following fields: 'Le nom du patient:' (text input), 'Nombre de grossesses :', 'Concentration du glucose :', 'Pression artérielle diastolique:', 'Epaisseur de peau au triceps:', 'Taux d'insuline:', 'ndice de masse corporelle:', 'Fonction pédigrée du diabète:', and 'Age:'. Each field has a corresponding input box. At the bottom of the form are 'SUBMIT' and 'RESET' buttons. On the right, under 'Information du modele:', there is text stating: 'Ce modèle utilise classifieur binaire automatique. Accuracy du modèle: 91% PIMA Indians diabetes dataset est utilisé pour entraîner le modèle.' Below this, it says 'Remplissez les détails et appuyez sur Soumettre pour voir le résultat.'

Figure IV.4: Interface principale de l'application.

The screenshot shows the 'Diabetes Prediction' application window after a submission. The 'Détails du patient' form is now filled with the following values: 'Le nom du patient: Aya', 'Nombre de grossesses : 6', 'Concentration du glucose : 148', 'Pression artérielle diastolique: 72', 'Epaisseur de peau au triceps: 35', 'Taux d'insuline: 0', 'ndice de masse corporelle: 33.6', 'Fonction pédigrée du diabète: 0.627', and 'Age: 50'. The 'SUBMIT' and 'RESET' buttons are still present. On the right, under 'Rapport', the following information is displayed: 'Nom du patient: Aya', 'Nombre de grossesses: 6', 'Concentration de glucose plasmatique: 148', 'Pression artérielle diastolique : 72', 'Epaisseur de la peau au niveau du triceps: 35', 'Taux d'insuline au bout de 2 heures : 0', 'Indice de masse corporelle: 33.6', 'Fonction pédigrée du diabète :0.627'. Below this, a bold statement reads: 'Le diagnostic suggère que le patient ne souffre pas de diabète.' At the bottom, it repeats the model information: 'Ce modèle utilise classifieur binaire automatique. Accuracy du modèle: 91% PIMA Indians diabetes dataset est utilisé pour entraîner le modèle.'

Figure IV.5 : Exécution de premier exemple.

The screenshot shows a web application window titled "Diabetes Prediction". The interface is split into two main sections: "Détails du patient" (Patient Details) on the left and "Rapport" (Report) on the right. In the "Détails du patient" section, there are input fields for various medical parameters: "Le nom du patient:" (Selma), "Nombre de grossesses:" (3), "Concentration du glucose:" (78), "Pression artérielle diastolique:" (50), "Epaisseur de peau au triceps:" (32), "Taux d'insuline:" (88), "ndice de masse corporelle:" (31), "Fonction pédigrée du diabète:" (0.248), and "Age:" (26). Below these fields are "SUBMIT" and "RESET" buttons. The "Rapport" section displays the same data in a bold, black font, followed by a diagnostic conclusion: "Notre diagnostic suggère que le patient souffre de diabète. Veuillez visiter le médecin bientôt." Below the report, there is a small text block: "Ce modèle utilise classifieur binaire automatique. Accuracy du modèle: 91% PIMA Indians diabetes dataset est utilisé pour entraîner le modèle."

Paramètre	Valeur
Nom du patient	Selma
Nombre de grossesses	3
Concentration du glucose	78
Pression artérielle diastolique	50
Epaisseur de peau au triceps	32
Taux d'insuline	88
ndice de masse corporelle	31
Fonction pédigrée du diabète	0.248
Age	26

Figure IV.6: Exécution de deuxième exemple.

#### IV.8. Conclusion :

Nous avons présenté dans ce chapitre une approche de classification, pour cela on a utilisé deux approches : l'approche classique ML et l'approche DL. La comparaison des résultats trouvés a montré que le modèle ANN (approche DL) donnera les meilleurs résultats.



CONCLUSION  
GÉNÉRALE

Ces dernières années, la reconnaissance d'objets est essentiellement basée sur l'approche ML qui donne une performance élevée. Quelques années plus tard, des progrès importants ont été réalisés dans le domaine de la reconnaissance d'objets, notamment avec l'apparition d'un nouveau sous-domaine appelé apprentissage profond ou approfondi.

Il est principalement basé sur l'utilisation de nombreux réseaux neuronaux d'unités simples interconnectées afin d'extraire des modèles significatifs d'une grande quantité de données pour résoudre des problèmes complexes tels que : la classification d'images médicales, la détection de la fraude, la reconnaissance de caractères, etc.

Actuellement, nous sommes en mesure d'utiliser des ensembles de données plus importants pour apprendre des modèles puissants, et de meilleures techniques pour éviter le sur- et le sous-équipement. Jusqu'à nos jours, les résultats obtenus dans ce domaine de recherche sont très surprenants dans différents domaines. On parle de valeurs de précision très élevées qui dépassent souvent le seuil de 90%. Dans le présent document, nous avons effectué un travail de classification sur un ensemble de très connu (Pima-Indians-Diabetes). Dans un premier temps, nous avons utilisé de nombreux algorithmes ML, à savoir : LR, LDA, KNN, DT (CART), NB, SVM. Nous avons obtenu un bon résultat de précision. Dans un deuxième temps, nous avons construit un modèle ANN pour effectuer la même tâche de classification. La performance obtenue est très surprenante.

Dans la perspective de ce travail prometteur, nous proposons d'améliorer ces résultats en améliorant l'architecture du modèle ANN construit en changeant certains paramètres du modèle tels que : le nombre de couches, le nombre de neurones dans chaque couche, le nombre d'époques d'entraînement et la taille des lots de données.

## Les notifications d'acceptation des papiers dans les conférences:

---

1. Nadjat Bouafia, Said Gadri, Hassina Tahmi, Nourelhouda Adouane and Baya Laoues: Efficient Clothes Categorization Based on Deep Learning Predictive Model, 4th Edition of the National Study Days on Computer Sciences Research JERI'2020, Université de Sekikda, May 20, 2020.

“Dear Baya Laoues:  
We are pleased to inform you that your paper “Efficient Clothes Categorization Based on Deep Learning Predictive Model,” to JERI’2020 has been accepted for Poster Presentation. We look forward to seeing you in Saida!  
Best regards,  
JERI’2020 PC chairs “”
2. Hassina Tahmi, Said Kadri , Baya Laoues, Nour Elhouda Adouane and Nadjat Bouafia: Handwritten Digit Recognition: Developing an Efficient ML and DL Model to Recognize Handwritten Digits, Conférence Nationale sur l’Intelligence artificielle et les technologies de l’information CNIATI’20, Chadli Bendjedid University El Tarf, Algeria, 2020.

“Paper ID: 14

Author(s): TAHMI Hassina, Said GADRI, LAOUES Baya, ADOUANE Nour ElHouda, BOUAFIA Nadjat

Title: Handwritten Digit Recognition: Developing an Efficient ML and DL Model to Recognize Handwritten Digits

It is our pleasure to inform you that the paper referenced above has been accepted for Oral presentation at the CNIATI’2020, to be held in El-Tarf, Algeria 17 March 2020. “”
3. Nour Elhouda Adouane, Said Kadri , Hassina Tahmi, Baya Laoues and Nadjat Bouafia: An Efficient CNN Model to Detect Traffic Signs for the Use of Self-Driving Cars, Conférence Nationale sur l’Intelligence artificielle et les technologies de l’information CNIATI’20, Chadli Bendjedid University El Tarf, Algeria, 2020.

“Paper ID: 13

Author(s): ADOUANE Nour ElHouda, Said GADRI, TAHMI Hassina, LAOUES Baya, BOUAFIA Nadjat

Title: An Efficient CNN Model to Detect Traffic Signs for the Use of SelfDriving Cars

It is our pleasure to inform you that the paper referenced above has been accepted for Poster presentation at the CNIATI’2020, to be held in El-Tarf, Algeria 17 March 2020. “”
4. Nour Elhouda Adouane, Said Kadri , Hassina Tahmi, Baya Laoues and Nadjat Bouafia: An Efficient CNN Model to Detect Traffic Signs for the Use of Self-Driving Cars, 4th Edition of the National Study Days on Computer Sciences Research JERI'2020, Université de Sekikda, May 20, 2020.

“We are pleased to inform you that your paper “An Efficient CNN Model to Detect Traffic Signs for the Use of Self-Driving Cars” to JERI’2020 has been accepted for Oral Presentation. We look forward to seeing you in Saida!  
Best regards,  
JERI’2020 PC chairs “”

# RÉFÉRENCES BIBLIOGRAPHIQUES



## Bibliographie

- [1] B.Anfel , C.Khadidja, “le diabète au cour de la grosse”, Mémoire, 2016-2017.
- [2] Dr C.Arnaud, “diabète gestationnel”, septembre 2015.
- [3] C. Beulac, “Intelligence artificielle avec apprentissage automatique pour l'estimation de la position d'un agent mobile en utilisant les modèles de markov cachés”, Mémoire. Montréal (Québec, Canada), Université du Québec à Montréal, Maîtrise en mathématique 2015.
- [4] B.Braunschweig, “Intelligence artificielle - livre blanc”, pp. 1–21, 2016.
- [5] Shan Suthaharan , “I Machine Learning Models and Algorithms for Big Data Classification Thinking with Examples for Effective Learning 123”
- [6] A. Amraoui, W. Baghli, and B. Benmammam, “Amélioration de la fiabilité du lien sans fil pour un terminal radio cognitive mobile”, Les 12èmes Journées Doctorales en Informatique et Réseaux (JDIR'11), Nov 2011, Belfort-Montbéliard, France. 2011.
- [7] Z.Giancarlo, MD R.Karim, M.Ahmed, “Deep learning with tensorflow”,2017
- [8] D. Ala eddine, “Planification et optimisation de trajectoire d'un robot manipulateur à 6 ddl par des techniques neuro\_floues”, 2015
- [9] P.Marc, “ Réseaux de neurones”, 2004
- [10] Shannon and Weaver 1949
- [11] O. Kramer , “Machine Learning for Evolution Strategies” , 2016.
- [12] [En ligne]. Available: <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/> .
- [13] [En ligne]. Available: [https://etav.github.io/algorithms/knn\\_implementation\\_scikit-learn.html](https://etav.github.io/algorithms/knn_implementation_scikit-learn.html).
- [14] L. Deng, D. Yu, et al., “Deep learning : methods and applications,” Foundations and Trends R in Signal Processing, vol. 7, no. 3–4, pp. 197–387, 2014.
- [15] Y. Bengio et al., “Learning deep architectures for ai,” Foundations and trends R in Machine Learning,vol. 2, no. 1, pp. 1–127, 2009.
- [16] Y. Bengio, A. Courville, and P. Vincent, “Representation learning : A review and new perspectives,”IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.
- [17] J. Schmidhuber, “Deep learning in neural networks : An overview,” Neural networks, vol. 61, pp. 85–117, 2015
- [18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

- [19] I. Arel, D. C. Rose, and T. P. Karnowski, "Deep machine learning-a new frontier in artificial intelligence research [research frontier]," IEEE Computational Intelligence Magazine, vol. 5, no. 4, pp. 13–18, 2010.
- [20] J. Schmidhuber, "Deep learning.," Scholarpedia, vol. 10, no. 11, p. 32832, 2015.
- [21] H.Wang, B. Raj, and E. P. Xing, "On the origin of deep learning," arXiv preprint arXiv :1702.07800, 2017.
- [22] M.D.Youcef, "*Deep Learning pour la classification des images*", Mémoire, 2016-2017.
- [23] Z. Valentino, S.Gianmario, S. Daniel, R.Peter, "Python deep learning", 2017.
- [24] N. Pat, "Neural networks and deep learning : deep learning explained to your granny a visual introduction for beginners who want to make their own deep learning neural network", 2017
- [25] T.Claude," Les réseaux de neurones artificiels, introduction au connexionnisme", 1992
- [26] M FUKUSHIMA Kunihiko, MAYAKE Sei, "Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition",1982
- [27] D.H. HUBEL, T.N.WIESEL, "Ferrier lecture-functional architecture of macaque monkey visual cortex",1977
- [28] Y. LECUN, L. BOTTOU, Y. BENGIO, P.HAFFNER, "Gradient-based learning applied to document recognition",1998
- [29] Dan C. CIRESAN, U.MEIER, J.MASCI, Luca M.GAMBARDELLA, J.SCHIDHUBER, "Flexible, high performance convolutional neural networks for image classification", 2011
- [30] K. O'SHEA, R. NASH , "An introduction to convolutional neural networks", 2015
- [31] P.WANG, J.XU, B.XU, Ch.LIU, H. ZHANG, F. WANG , H. HAO, "Semantic clustering and convolutional neural networks for short text categorization", 2015
- [32] <http://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-tensorflow/>
- [33] O.Mehdi,K.Salim, " *Classification d'objets avec le Deep Learning*", mémoire, 2017/2018
- [34] J. Brownlee, "What Are Word Embeddings for Text?". October 11, 2017  
<https://machinelearningmastery.com/what-are-word-embeddings>
- [35] P.SMOLENSKY, "Information processing in dynamical systems: foundations of harmony theory".1986
- [36] Geoffrey E.HINTON , "Deep belief networks", 2009
- [37] <https://blog.engineering.publicissapient.fr/2017/03/01/tensorflow-deep-learning-episode-1-introduction/>.
- [38] <https://keras.io/>.

[39] <https://www.python.org/> .

[40] <http://www.euro-langues.org/meilleures-bibliotheques-python-pour-lapprentissage-automatique-apprendre-une-langue-etrangere/>.

[41] <https://he-arc.github.io/livre-python/pillow/index.html>.

## Abstract:

Diabetes is one of the chronic diseases which causes an increase in blood sugar. Many complications may occur if diabetes remains untreated or unidentified. The obvious method to identify diabetes is to visit a diagnostic center or consulting a doctor in the field. But with the rise in machine and deep learning approaches, we can solve easily this critical problem. In the present work, we have designed an intelligent predictive model which permits to detect diabetes in patients with maximum accuracy. For this purpose, we have used in a first stage machine learning algorithms, including: LR, LDA, KNN, CART, NB, and SVM. In a second stage, we have developed an Artificial Neural Networks model composed of many layers to improve detection accuracy. Finally, we established a comparison between ML and DL algorithms.

Keywords: Diabetes classification, machine learning, deep learning, Artificial neural networks.

---

## Résumé:

Le diabète est l'une des maladies chroniques qui provoquent une augmentation du taux de sucre dans le sang. De nombreuses complications peuvent survenir si le diabète reste non traité ou non identifié. La méthode évidente pour identifier le diabète est de se rendre dans un centre de diagnostic ou de consulter un médecin spécialiste. Mais avec l'essor des approches d'apprentissage automatique et approfondi, nous pouvons résoudre facilement ce problème critique. Dans le présent travail, nous avons conçu un modèle prédictif intelligent qui permet de détecter le diabète chez les patients avec un maximum de précision. À cette fin, nous avons utilisé dans un premier temps des algorithmes d'apprentissage automatique, notamment : LR, LDA, KNN, CART, NB et SVM. Dans un deuxième temps, nous avons développé un modèle Réseaux de Neurones Artificiel composé de plusieurs couches pour améliorer la précision de la détection. Enfin, nous avons établi une comparaison entre les algorithmes ML et DL.

Mots-clés : Classification du diabète, apprentissage machine, apprentissage profond, réseaux neuronaux artificiels.

---

## ملخص :

مرض السكري من الأمراض المزمنة التي تؤدي إلى ارتفاع نسبة السكر في الدم. قد تحدث العديد من المضاعفات إذا ظل مرض السكري دون علاج أو غير معروف. الطريقة الواضحة لتحديد مرض السكري هي زيارة مركز تشخيص أو استشارة طبيب مختص. ولكن مع ارتفاع أساليب التعلم الآلي والعميق ، يمكننا بسهولة حل هذه المشكلة الحرجة. في العمل الحالي ، قمنا بتصميم نموذج تنبؤي ذكي يسمح باكتشاف مرض السكري بأقصى قدر من الدقة. لهذا الغرض، استخدمنا في المرحلة الأولى خوارزميات للتعلم الآلي ، بما في ذلك: LR و LDA و KNN و CART و NB و SVM. في المرحلة الثانية، قمنا بتطوير نموذج RNA يتكون من عدة طبقات لتحسين دقة الكشف. أخيرًا، أجرينا مقارنة بين خوارزميات ML و DL.

الكلمات المفتاحية: تصنيف السكري ، التعلم الآلي ، التعلم العميق ، الشبكات العصبية الاصطناعية .