

<u>Université 20 Août 1955 – Skikda</u>		<u>جامعة 20 أوت 1955 - سكيكدة</u>
<u>Faculté des Sciences</u>		<u>كلية العلوم</u>
<u>Departement d'informatique</u>		<u>قسم الإعلام الآلي</u>

**Mémoire De fin d'étude en vue de l'obtention du Diplôme  
de Master en Informatique**

**Option : système d'information avancé et application**

**Sujet :**

**Conception et réalisation d'un système  
De classification de diabète**

**Réalisé par les étudiants :**

- Boumous Hadil
- Blada Rania

**Dirigé par :**

Dr. Bnoudina Lazher

Année Universitaire 2022-2023

---

# Remerciement

**En préambule à ce mémoire nous remerciant ALLAH qui nous a aidé et nous a donné la patience et le courage durant ces longues années d'étude.**

**Nous souhaitant adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.**

**Ces remerciements vont tout d'abord au corps professoral et administratif du département d'informatique pour la richesse et la qualité de leurs enseignements et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée.**

**Nous tenons à remercier sincèrement Monsieur Bnoudina Lazher, en tant qu'encadrant de mémoire, il a toujours montré l'écoute et la disponibilité tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'elle a bien voulu nous consacrer pour que ce mémoire voit le jour**

**On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience. Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragés au cours de la réalisation de ce mémoire. Merci à toutes et à tous.**

---

# Dédicaces

**A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,**

**A mes chères sœurs pour leurs encouragements permanents, et leur soutien moral,**

**A mon frère, mon fiancé pour leur appui et leur encouragement,**

**A toute ma famille pour leur soutien tout au long de mon parcours universitaire,**

**Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infailible,**

**Merci d'être toujours là pour moi.**

---

# Résumé

Aujourd'hui, le diabète est une maladie chronique répandue et l'une des principales causes de décès dans le monde. La détection précoce du diabète est cruciale pour un traitement rapide et pour stopper la progression de la maladie. Dans cette recherche, nous proposons une méthode de prédiction du diabète en utilisant une approche de classification. Notre modèle repose sur l'utilisation des couches cachées des réseaux de neurones profonds. Nous avons ajusté plusieurs paramètres et utilisé la fonction de perte «binary\_crossentropy» pour obtenir un modèle de prédiction de haute précision. Les résultats expérimentaux démontrent l'efficacité de notre modèle d'apprentissage en profondeur pour la prédiction du diabète, avec une précision supérieure lors de l'entraînement sur les ensembles de données des Indiens Pima.

# Abstract

Today, diabetes is a widespread chronic disease and one of the leading causes of death worldwide. Early detection of diabetes is crucial for prompt treatment and halting the progression of the disease.

In this research, we propose a method for diabetes prediction using a classification approach. Our model is based on the utilization of hidden layers in a deep neural networks. We fine-tuned several parameters and employed the binary\_crossentropy loss function to achieve a highly accurate prediction model.

The experimental results demonstrate the effectiveness of our deep learning model for diabetes prediction, with superior accuracy when trained on the Pima Indian diabetes dataset.

Compared to modern methods, our proposed model shows significant improvements in performance. These findings highlight the potential of our approach for accurate diabetes prediction and pave the way for future enhancements and applications in this field.

---

# Table des matières

Remerciement .....	2
Dédicaces .....	3
Résumé .....	4
Table des matières .....	5
Liste des tableaux .....	8
Table des figures .....	9
Introduction générale .....	10
Chapitre1 : Le diabète.....	12
1. Introduction .....	12
2. La définition du diabète .....	12
3. Les différents types du diabète .....	13
3.1. Diabète de type1 .....	13
3.2. Diabète de type 2 .....	14
3.3. Diabète gestationnel.....	15
4. Les conséquences habituelles du diabète .....	16
5. Le diagnostic médical du diabète .....	16
6. Les symptômes du diabète .....	17
7. La relation du diabète avec.....	17
7.1. L'hérédité .....	17
7.2. La Grossesse .....	18
7.3. L'alimentation.....	18
8. Conseil .....	18
9. Conclusion.....	18
Chapitre 02 : Les réseaux de neurones artificiels.....	19
1. Introduction .....	19
2. Le neurone.....	19
2.1. Le neurone biologique.....	19
2.2. Le neurone artificiel .....	21
2.3. Du neurone biologique au neurone artificiel .....	22
3. Les fonctions d'activation .....	23

---

4. Les règles d'apprentissage .....	25
4.1. La règle de Hebb.....	25
5. Réseau de neurones artificiels .....	26
5.1. Les différents types de réseaux de neurones artificiels.....	26
5.2. Apprentissage d'un réseau de neurone.....	28
6. Perceptron multi couche.....	30
7. L'algorithme d'apprentissage rétropropagation du gradient.....	31
7.1. Méthode du gradient .....	32
7.2. Rétro propagation des erreurs.....	32
8. Les réseaux de neurones artificiels et leurs applications .....	35
9. Applications dans diverses industries .....	36
9.1. En gestion et finance .....	36
9.2. Reconnaissance de formes.....	36
9.3. Dans divers domaine .....	36
10. Conclusion.....	37
Chapitre 03 : Analyse et conception.....	38
1. Introduction .....	38
2. Organigramme d'un système.....	38
2.1. Présentation d'organigramme du notre système.....	39
3. Modélisation UML .....	39
3.1. Définition d'UML.....	39
4. Les différents types de diagrammes UML .....	40
4.1. Les diagrammes de structure .....	40
4.2. Les diagrammes de comportement .....	40
5. Diagramme de cas d'utilisation .....	40
5.1. Diagramme de cas d'utilisation du notre système .....	41
6. Diagramme de flux de donnée.....	41
6.1. Définition.....	41
6.2. DFD du notre système.....	42
6.3. La description .....	42
7. Conclusion.....	43
Chapitre 04 : Réalisation.....	44
1. Introduction .....	44
2. Présentation des outils de développement.....	44
2.1. Python.....	44

---

2.2. Bibliothèques Python .....	45
2.3. Editeur de code .....	47
3. La base de donné .....	48
3.1. Description de la base de données .....	48
3.2. séparation ou la division de la base de donné.....	48
4. l'apprentissage du réseaux de neurones .....	50
5. Fonction de perte ou de coût .....	50
6. Optimiseur .....	51
7. Procédure de descente de gradient.....	51
8. Test.....	51
9. Implémentation et résultats obtenus .....	51
9.1. Description des différentes structures .....	51
9.2. Différentes Procédures .....	52
9.3. Résultats et discussion.....	54
10. Prédiction du diabète.....	55
11. Conclusion.....	56
Conclusion générale .....	57
Références.....	58

---

# Liste des tableaux

Tableau 1.1. Les fonctions d'activation .....	24
Tableau 1.2. Fonctions d'activation.....	25
Tableau 3: Réseaux de neurones en gestion et finance .....	36
Tableau 4: Réseaux de neurones dans divers domaine.....	37

---

# Table des figures

Figure 1: Cas non diabete .....	13
Figure 2: diabète type1 .....	13
Figure 3: diabète type2 .....	14
Figure 4: le schéma d'un neurone .....	20
Figure 5:Synape.....	20
Figure 6: schéma d'un neurone artificiel.....	21
Figure 7: Neurone Artificiel.....	22
Figure 8 : Les réseaux de neurones - comportement intelligent .....	23
Figure 9: Réseaux de neurones à propagation avant .....	27
Figure 10: Réseau de neurone récurrent .....	28
Figure 11: le classement hiérarchique des méthodes d'apprentissage .....	29
Figure 12: exemple d'un réseau de type perceptron multicouche.....	31
Figure 13: organigramme du notre système.....	39
Figure 14:diagramme de cas d'utilisation du système .....	41
Figure 15: Diagramme de flux de donnée .....	42
Figure 16: Logo du python .....	44
Figure 17: Bibliothèque du python.....	45
Figure 18: TensorFlow logo.....	45
Figure 19: NumPy logo .....	46
Figure 20: pandas logo .....	46
Figure 21: Sckikit-learn logo .....	47
Figure 22: La division de la base de donné .....	48
Figure 23: Les premiers lignes du dataset de test.....	49
Figure 24: Les premiers lignes du dataset d'enytrainement .....	49
Figure 25: Fonction d'apprentissage .....	50
Figure 26: Division du data set .....	51
Figure 27: Prétraitement des données.....	52
Figure 28: Fonction d'entrainement du notre réseaux de neurones .....	52
Figure 29: Entrainement .....	53
Figure 30: Validation.....	53
Figure 31: Visualisation des performances du modèle entraîné. ....	55
Figure 32: cas diabétique .....	56
Figure 33: cas non diabétique .....	56

# Introduction générale

Le diabète est une maladie chronique qui affecte des millions de personnes dans le monde entier. Il se caractérise par une incapacité du corps à réguler efficacement la glycémie, entraînant des niveaux élevés de sucre dans le sang. Cette condition peut avoir des conséquences graves sur la santé, notamment des complications cardiovasculaires, rénales, oculaires et neurologiques. Le diabète représente un fardeau majeur pour les systèmes de santé et nécessite une détection précoce et une gestion appropriée. Cependant, l'accès à des soins médicaux spécialisés peut être limité, en particulier dans les régions éloignées ou les pays en développement.

Face à cette problématique de l'absence de médecins, l'intelligence artificielle et l'apprentissage automatique offrent de nouvelles opportunités. Les systèmes de classification basés sur des réseaux de neurones peuvent jouer un rôle crucial dans le diagnostic du diabète, en permettant une évaluation précoce et précise de la maladie, même en l'absence d'un médecin.

L'objectif de ce projet de recherche est de développer un système automatisé de classification du diabète qui peut remplacer partiellement l'intervention d'un médecin. Ce système vise à fournir un outil fiable, précis et accessible pour le diagnostic précoce du diabète. En permettant aux individus de détecter rapidement et facilement la présence de la maladie, ce projet contribuera à améliorer les résultats de santé et à réduire la charge sur les ressources médicales.

Afin de mener à bien notre projet de recherche, nous suivrons le plan suivant :

- Chapitre 1 : consistera en la présentation du diabète, ses différents types, le diagnostic, ses symptômes, le diagnostic médical du diabète.
- Chapitre 2 : portera sur les réseaux de neurone artificiels, différents types d'apprentissage, tels que la rétropropagation et la règle de Hebb, et leur applications
- Chapitre 3 : concernant l'analyse et de conception du système portera sur la modélisation UML et les diagrammes de flux de données, qui permettront de décrire de manière précise le fonctionnement du système de classification.

- Chapitre 4 : nous présenterons l'implémentation du système de classification du diabète, en détaillant les outils de programmation utilisés, la base de données utilisée pour l'entraînement et les différentes fonctionnalités du système

# Chapitre 1 : Le diabète

## 1. Introduction

Le diabète est une affection qui perturbe l'utilisation efficace de l'énergie provenant des aliments consommés par le corps. Cette maladie survient lorsque le pancréas cesse de produire suffisamment d'insuline ou lorsque le corps développe une résistance à cette hormone. L'insuline joue un rôle crucial dans la régulation du taux de sucre dans le sang. L'hyperglycémie, une élévation de la concentration de sucre dans le sang, est un effet courant du diabète mal contrôlé, pouvant entraîner des complications graves affectant différents systèmes organiques, en particulier les nerfs et les vaisseaux sanguins. Le diabète est une maladie répandue, principalement chez la population âgée à l'échelle mondiale. Selon l'Organisation mondiale de la santé (OMS), plus de 463 millions de personnes sont touchées par le diabète dans le monde, avec environ 5 millions de diabétiques en Algérie en 2019. Dans ce chapitre, nous aborderons les aspects généraux de la maladie du diabète, y compris les différents types de diabète, les symptômes, les méthodes de diagnostic et les options de traitement.

## 2. La définition du diabète

Le diabète est une maladie métabolique caractérisée par une élévation anormale chronique de la glycémie, définie par le taux de sucre dans le sang. Cette augmentation de la glycémie est causée par un dysfonctionnement de la sécrétion ou de l'action de l'insuline, une hormone fabriquée par le pancréas. Elle peut provoquer à plus ou moins long terme des lésions de différents organes, comme les yeux, les reins, les nerfs et les vaisseaux sanguins. Près de 90% des diabétiques vivent pendant des années avec cette maladie sans le savoir car le diabète ne provoque en général pas de manifestations au début de son évolution [1].

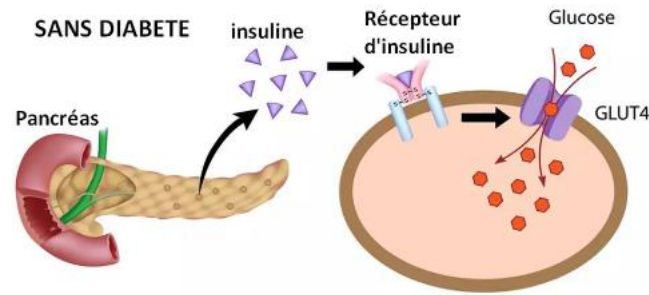


Figure 1: Cas non diabète

### 3. Les différents types du diabète

#### 3.1. Diabète de type 1

Le diabète de type 1, encore appelé diabète insulino-dépendant, concerne environ 6% des diabétiques. Au cours du diabète de type I, des cellules du pancréas sont détruites entraînant alors **une** diminution voire une absence totale de sécrétion d'insuline, le pancréas ne réussit plus à synthétiser suffisamment d'insuline. Le glucose n'arrive donc plus à pénétrer dans les cellules, augmentant ainsi son taux sanguin et provoquant une hyperglycémie. Le diabète de type 1 apparaît surtout chez l'enfant et le jeune adulte [1].

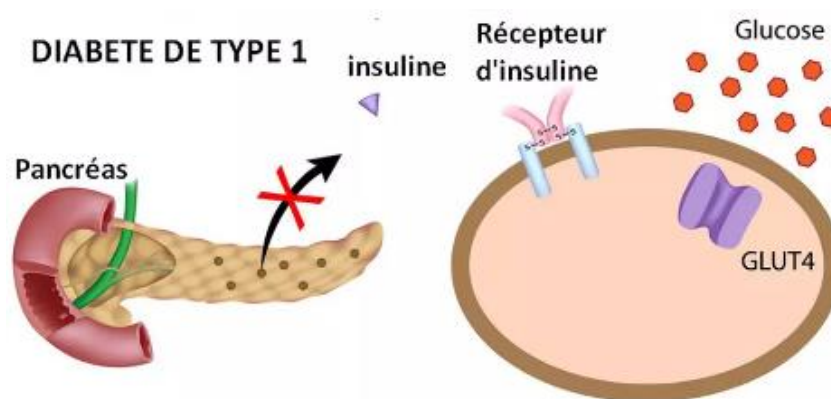


Figure 2: diabète type 1

#### 3.1.1. Les causes

Il n'est pas toujours possible de savoir quels sont les déclencheurs du diabète de type 1. La maladie peut être multifactorielle [2].

Des facteurs de risque ont été identifiés chez les personnes atteintes de diabète de type 1 :

- **L'hérédité** : une prédisposition génétique peut expliquer, en partie, le développement du diabète de type 1 ;

- **Une pathologie** : les maladies du pancréas peuvent déclencher un diabète de type 1 ;
- **Une infection par un virus ou une bactérie** : certaines études semblent démontrer que certaines infections pourraient entraîner des dysfonctionnements dans l'organisme, notamment un dérèglement du système immunitaire ;

### 3.1.2. Traitement

Le traitement du diabète de type 1 repose sur des injections sous-cutanées d'insuline, plusieurs fois par jour, pour compenser son défaut de production par l'organisme. On utilise aujourd'hui des analogues d'insuline humaine, produits par des bactéries génétiquement modifiées.

- Les analogues « rapides » possèdent une action quasiment immédiate et de courte durée, utile pour faire redescendre rapidement le taux de glucose en cas de prise alimentaire.
- Les analogues d'action ultra lente (insulines basales) sont actifs pendant environ 24 heures et assurent la présence permanente d'insuline dans le sang tout au long de la journée, comme chez un individu non diabétique.

Ces deux types d'analogues sont complémentaires [3].

### 3.2. Diabète de type 2

Le diabète de type 2 résulte d'une diminution des effets de l'insuline, on parle alors d'insulino-résistance. Si elle ne fonctionne pas bien, le sucre s'accumule dans le sang, la glycémie augmente, le diabète apparaît. Le diabète de type 2 touche plus de 90% des diabétiques, surtout chez l'adulte avec une survenue possible dès l'adolescence [1].

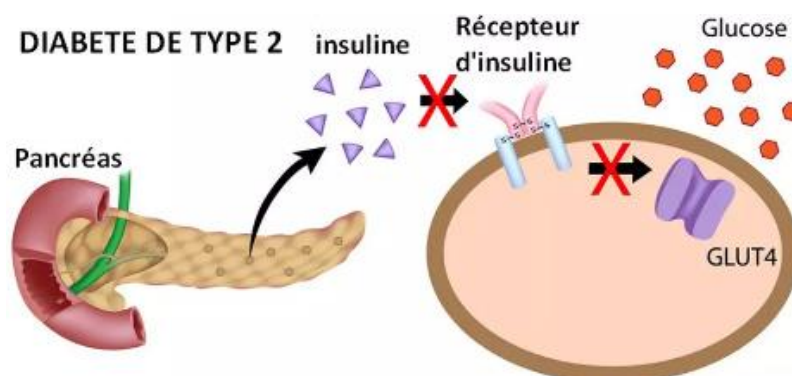


Figure 3: diabète type2

### 3.2.1. Les causes

Il existe un terrain génétique favorisant l'apparition du diabète de type 2. Toutefois, personne ne connaît la cause exacte de cette maladie. Ce trouble est plus fréquent chez les personnes présentant les caractéristiques suivantes [4] :

- âge supérieur à 45 ans ;
- surpoids prononcé ou obésité : la présence d'un syndrome métabolique multiplie le risque de diabète de type 2 par 12 et le risque de problèmes cardiovasculaires par 3 ;
- antécédents familiaux de diabète ;
- sédentarité.

### 3.2.2. Traitement

Le traitement de référence du diabète de type 2 est l'optimisation des habitudes de vie.

- une perte de poids si nécessaire,
- une activité physique régulière,
- et une alimentation équilibrée peuvent être suffisants pour contrôler la glycémie dans un premier temps[5].

## 3.3. Diabète gestationnel

Le diabète gestationnel est un trouble de la tolérance au sucre avec augmentation de la glycémie (quantité de sucre dans le sang) plus ou moins importante. Il est **diagnostiqué pour la première fois au cours de la grossesse**. Cette anomalie de la tolérance au sucre est transitoire : elle apparaît pendant la grossesse et disparaît après l'accouchement [6].

### 3.3.1. Les causes

La grossesse augmente les besoins de la mère en insuline (l'hormone produite par le pancréas pour diminuer la glycémie) d'un facteur de 2 à 3. Cette augmentation est liée à l'équilibre hormonal de la grossesse. Chez certaines femmes prédisposées, le pancréas n'est pas capable de répondre à cette augmentation du besoin en insuline et un diabète gestationnel s'installe.

Dans certains cas, une hyperglycémie sans symptôme était déjà présente avant le début de la grossesse et la découverte de l'hyperglycémie n'est que le révélateur d'un diabète de type 2 pré-existant méconnu [7].

### 3.3.2. Traitement

Le traitement du diabète gestationnel réduit les complications autour de la naissance et les risques de macrosomie fœtale (bébé de plus de 4 kg à la naissance à terme).

Il est généralement coordonné par le médecin gynécologue ou gynéco-obstétricien. Il peut être utile de recourir à un médecin endocrinologue et/ou à un(e) diététicien(ne).

Il est indispensable que la femme enceinte présentant un diabète de grossesse suive les recommandations de l'ensemble de l'équipe médicale.

Le traitement repose sur des conseils diététiques et une activité physique régulière et si nécessaire, l'insulinothérapie ( l'insuline est un biomédicament : on parle de biothérapie).

Une autosurveillance de la glycémie est conseillée en cas de prise en charge diététique et indispensable en cas d'insulinothérapie [8].

## 4. Les conséquences habituelles du diabète

Avec le temps, le diabète peut endommager le coeur, les vaisseaux sanguins, les yeux, les reins et les nerfs.

- Le diabète multiplie par 2 ou 3 le risque chez l'adulte de souffrir d'accidents cardiaques ou vasculaires cérébraux<sup>2</sup>.
- Associée à une diminution du débit sanguin, la neuropathie qui touche les pieds augmente la probabilité d'apparition d'ulcères des pieds, d'infection et, au bout du compte, d'amputation des membres.
- La rétinopathie diabétique est une cause importante de cécité et survient par suite des lésions des petits vaisseaux sanguins de la rétine qui s'accumulent avec le temps. 2,6% de la cécité dans le monde peut être attribuée au diabète<sup>3</sup>.
- Le diabète figure parmi les principales causes d'insuffisance rénale<sup>4</sup> [9].

## 5. Le diagnostic médical du diabète

Dans un laboratoire d'analyse médicale, il vous suffit de faire une **prise de sang à jeun** pour déterminer le taux de la glycémie (taux de sucre dans le sang).

Un diabète est avéré lorsque la glycémie à jeun est égale ou supérieure à 1.26 g/l, à deux reprises, ou égale ou supérieure à 2 g/l à n'importe quel moment de la journée.

Parfois un doute sur le diagnostic demeure, même après le second prélèvement, dans le cas par exemple où votre résultat est douteux ou contradictoire avec le premier. Dans ce cas, il est recommandé d'effectuer une épreuve d'hyperglycémie provoquée par voie orale (HGPO). Ce test consiste à ingérer une quantité de sucre déterminée par votre médecin, puis à mesurer les taux de glycémie à 1 heure, 2 heures ou 3 heures après l'ingestion [10].

## **6. Les symptômes du diabète**

Les symptômes les plus fréquents du diabète sont l'augmentation du volume et de la fréquence des urines, la soif intense, une faim exagérée ainsi que la perte de poids[11].

- Fatigue / Somnolence ;
- Nausées / Vomissements ;
- Bouche sèche ;
- Vision trouble ;
- Diminution de l'endurance physique ;
- Irritabilité ;
- Fourmillements / Engourdissements dans les mains et les pieds ;
- Dysfonction érectile ;
- Plaies qui guérissent mal ;
- Infections fréquentes et récurrentes.

## **7. La relation du diabète avec**

### **7.1. L'hérédité**

Le poids de l'hérédité diffère selon qu'il s'agit du diabète de type 1 ou du diabète de type 2. Lorsque l'un des deux parents est diabétique de type 2, le risque de transmission à la descendance est de l'ordre de 40 % et si les deux parents sont atteints, le risque grimpe à 70%. Dans le diabète de type 1, le risque se situe entre 4 et 8%, plus précisément 8 % si le père est diabétique, 4 % si c'est la mère (mais 30 % si les deux parents le sont). Il est donc utile de se construire un arbre généalogique pour repérer les membres de sa famille diabétiques et connaître son patrimoine génétique.[12]

## **7.2. La Grossesse**

La plupart des femmes atteintes de diabète gestationnel le développent parce qu'elles ne peuvent pas produire suffisamment d'insuline. L'insuline aide à contrôler le taux de sucre (glucose) dans le sang. Durant la grossesse, le corps a besoin de plus d'insuline, car le placenta produit une hormone qui rend le corps moins réceptif à l'insuline (affection appelée insulino-résistance). Cet effet est plus particulièrement perceptible tard dans la grossesse [13].

## **7.3. L'alimentation**

L'alimentation occupe un rôle important, au même titre que l'activité physique et les traitements médicamenteux. Les effets bénéfiques d'une alimentation équilibrée sur le glycémie et la santé en général sont scientifiquement prouvés. Manger équilibré est donc un bienfait indispensable[14].

## **8. Conseil**

- Une alimentation équilibrée est avant tout une alimentation variée
- Plus votre assiette ou votre plateau repas est coloré, plus il y a de chances qu'il soit équilibré ;
- 3 repas par jour, c'est essentiel! Et évitez de grignoter entre les repas ;
- En cas de petit creux, optez pour un fruit ou un yaourt ;
- Mangez lentement et prenez le temps de bien mastiquer, vous sentirez plus facilement la satiété ;
- Goûtez vos plats avant de les resaler ou d'ajouter du sucre ;
- Il n'y a pas d'aliment « mauvais », uniquement des mauvaises habitudes alimentaires [15].

## **9. Conclusion**

Dans ce chapitre, nous avons abordé la maladie du diabète, ses symptômes, ainsi que les méthodes de diagnostic et de traitement. De plus, nous avons présenté les trois principaux types de diabète, . Dans le prochain chapitre, nous décrirons les réseaux de neurones artificiels.

# Chapitre 02 : Les réseaux de neurones artificiels

## 1. Introduction

Dans ce chapitre, nous allons explorer le domaine des réseaux de neurones, qui est l'un des piliers de l'apprentissage automatique et de l'intelligence artificielle. Les réseaux de neurones sont des modèles informatiques inspirés du fonctionnement du cerveau humain et sont capables d'apprendre à partir de données pour effectuer des tâches complexes telles que la reconnaissance de motifs, la classification et la prédiction.

## 2. Le neurone

Un neurone est une cellule d'un système permettant la communication et le traitement de l'information [16].

### 2.1. Le neurone biologique

Un neurone est une cellule nerveuse constituant la base du système nerveux spécialisée dans le traitement des signaux électriques. C'est en 1881 que le terme de neurone (all. Neuron, n) fut intégré au vocabulaire médical par l'anatomiste allemand Heinrich Wilhelm Waldeyer. En biologie, le cerveau humain contient un grand nombre de neurones fortement interconnectés constituant des réseaux de neurones. Le neurone biologique est composé de trois parties principales[17] :

- Le corps cellulaire composé du centre de contrôle traitant les informations reçues par les dendrites ;
- Les dendrites sont les principaux fils conducteurs par lesquels transitent l'information venue de l'extérieur ;
- L'axone est fil conducteur qui conduit le signal de sortie du corps cellulaire vers d'autres neurones.

1: Récepteur

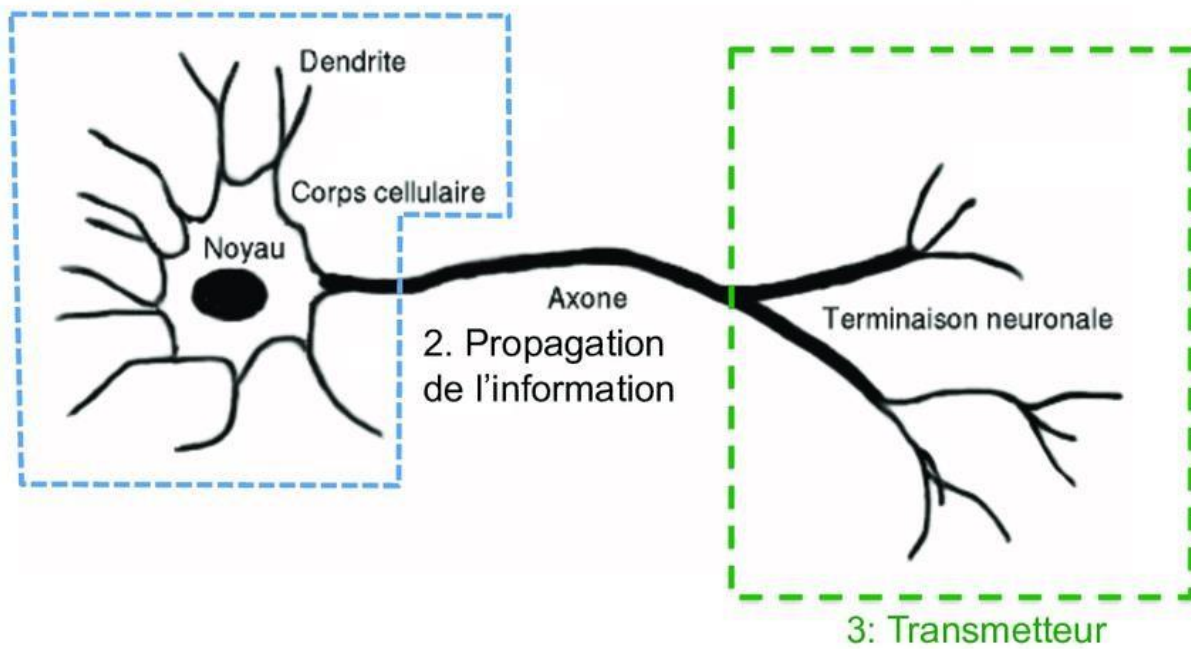


Figure 4: le schéma d'un neurone

Quant aux synapses, elles font effet de liaison et de pondération entre neurones et permettent donc aux neurones de communiquer entre eux.

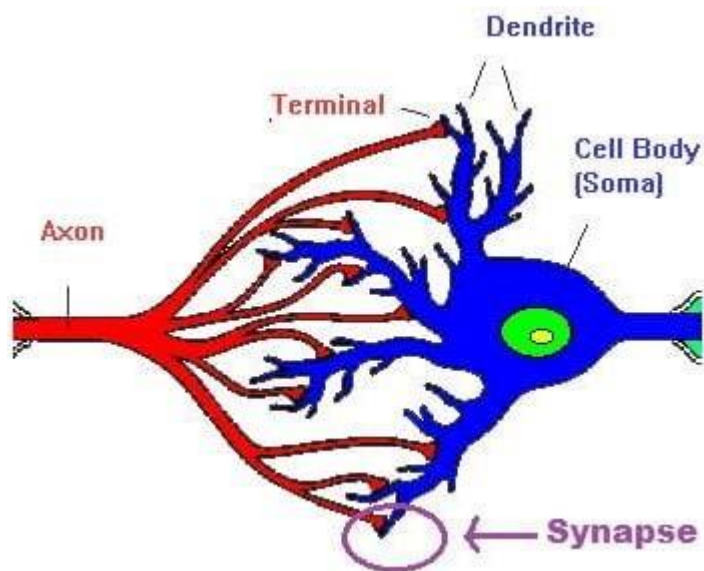


Figure 5: Synapse

Les neurones biologiques ont deux propriétés qui peuvent être appliquées au niveau des neurones artificiels

- L'excitabilité : c'est-à-dire la capacité de réagir sous l'effet de stimulations extérieures et de convertir celles-ci en impulsions
- La conductivité : c'est-à-dire la capacité de transmettre les impulsions nerveuses.

## 2.2. Le neurone artificiel

Le schéma montre la composition d'un neurone artificiel. Chaque neurone faisant partie du réseau est une machine élémentaire. Il reçoit un certain nombre de variables d'entrées provenant de neurones en amont. À chacune de ces entrées, est associé un poids qui représente la force de la connexion entre les deux neurones. Chaque neurone est doté d'une unique sortie qui se ramifie ensuite pour représenter les entrées qui alimenteront d'autres neurones en aval. Pour résumer, chaque neurone calcule une sortie unique en se basant sur les informations qui lui sont données[18].

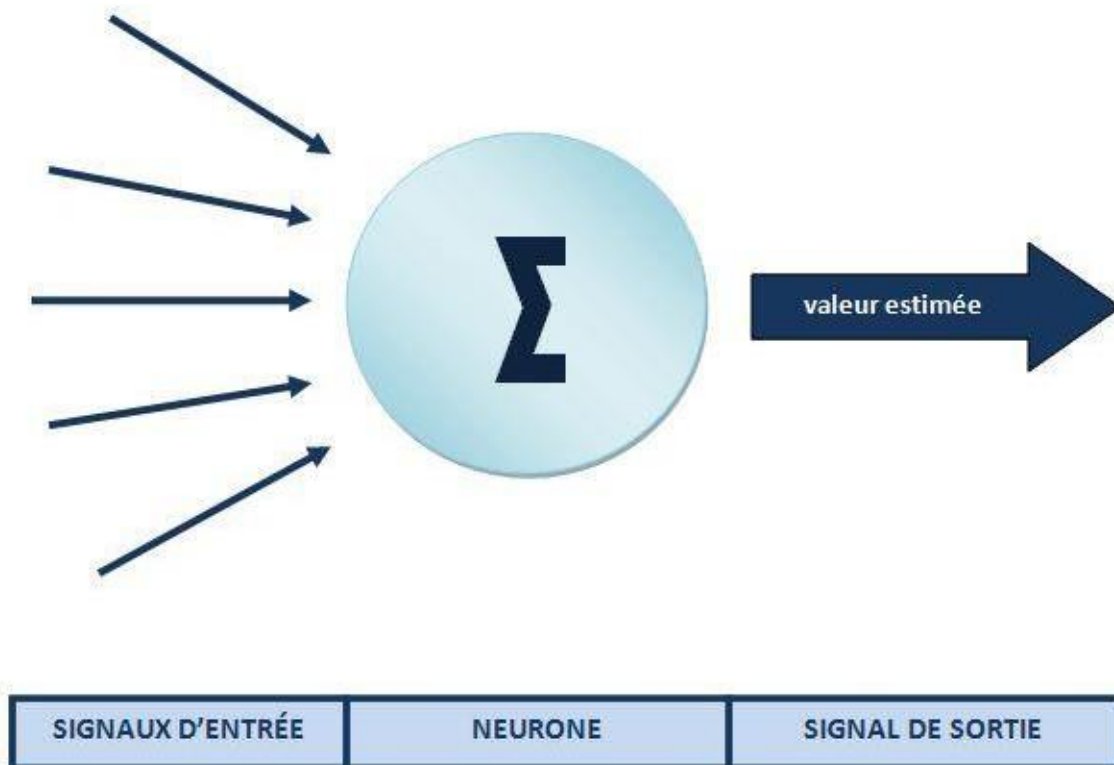


Figure 6: schéma d'un neurone artificiel

Le neurone en tant qu'unité élémentaire agit de la façon suivante :

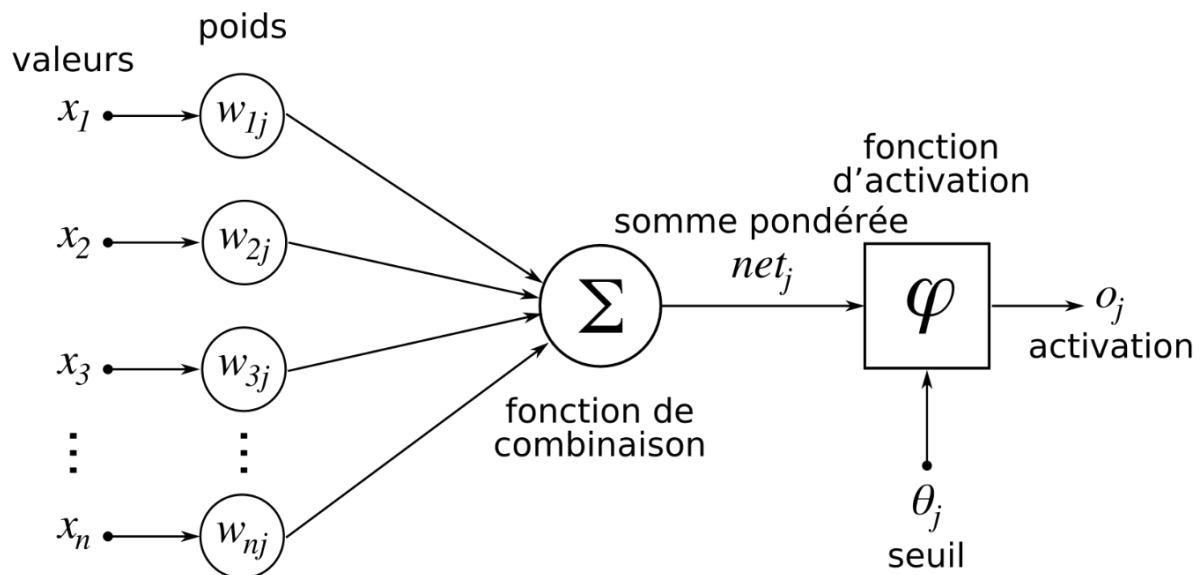


Figure 7: Neurone Artificiel

- 1ère phase : Le neurone fait le calcul de la somme pondérée des entrées (en fonction de la force des connexions). L'apprentissage ayant été réalisé auparavant, le poids des connexions est ici déjà déterminé et fixe.
- 2eme phase : À partir de la valeur obtenue, une fonction d'activation ou de transfert va calculer la valeur de l'état du neurone. Le neurone compare la somme pondérée des entrées à une valeur de seuil et fournit alors une réponse en sortie.

La majorité des fonctions sont continues et offrent donc une infinité de valeurs possibles comprises dans les intervalles  $[0; +1]$  ou  $[-1; +1]$ . Il existe différents types de fonctions de transfert pour le neurone artificiel[18] :

- Fonction à seuil ;
- Fonction linéaire par morceaux ;
- Fonction sigmoïde ;
- Fonction gaussienne.

### 2.3. Du neurone biologique au neurone artificiel

Les réseaux de neurones biologiques qui constituent le cerveau humain réalisent simplement de nombreuses applications telles que la reconnaissance de formes, le traitement du signal, la mémorisation, la généralisation, l'apprentissage par l'exemple, etc. Or ces applications sont actuellement à la limite de leurs possibilités actuelles, même en tenant compte de tous les efforts déployés en algorithmique ou en intelligence artificielle. C'est à partir de l'hypothèse que le comportement intelligent humain est le résultat de la structure et des éléments de bases du système nerveux central (que sont les neurones), que l'on a développé les réseaux de neurones artificiels. Cette inspiration à partir du modèle biologique provient du fait

que le cerveau humain est un système apprenant basé sur une structure contenant environ 100 milliards de neurones reliés entre eux par 10000 contacts synaptiques ce qui représente un million de milliards de synapses. Les réseaux de neurones artificiels sont donc un moyen de modéliser le mécanisme d'apprentissage et de traitement de l'information qui se produit dans le cerveau humain[18].

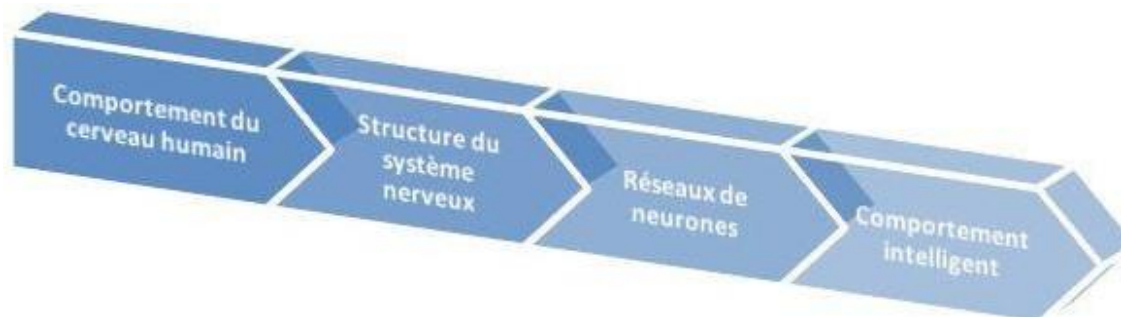


Figure 8 : Les réseaux de neurones - comportement intelligent

### 3. Les fonctions d'activation

La fonction d'activation sert avant tout à modifier de manière non-linéaire les données. Cette non-linéarité permet de modifier spatialement leur représentation.

Dit simplement, la fonction d'activation permet de changer notre manière de voir une donnée. Par exemple si on a comme donnée : chaque semaine 50% des clients d'un magasin achètent des barres de chocolat ; la fonction d'activation permettrait de changer la donnée en 50% des clients aiment le chocolat ou encore, 50% des clients prévoient d'acheter du chocolat chaque semaine.

Le changement de représentation peut, dans notre exemple, nous permettre d'ajuster la stratégie commerciale, ou bien les stocks possédés, de notre magasin.

Un modèle étant composé de multiples couches, et donc de multiples fonctions d'activation, des changements successifs et complexes de représentation s'opèrent. Cela permet d'avoir un nouveau point de vue sur nos données que l'homme serait incapable d'avoir en peu de temps. Il ne faut pas confondre fonction d'activation et loss function. La loss function est appliqué sur tous le modèle, elle permet de calculer la performance du modèle Au contraire la fonction d'activation est spécifique à chaque couche, elle permet de transformer les données.

La particularité de cette fonction d'activation est qu'elle est non-linéaire. Cette non-linéarité permet de changer la représentation des données, d'avoir une nouvelle approche sur ces données. Ce changement de représentation ne serait pas possible avec une transformation linéaire.

Chaque neurone d'une couche va appliquer la fonction d'activation de la couche sur les données. Cette transformation sera différente selon chaque neurone car chacun possède un poids différent. Dans ce tableau on va présenter les différentes fonctions d'activation [19].

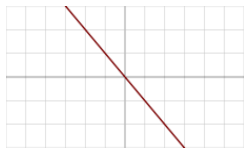
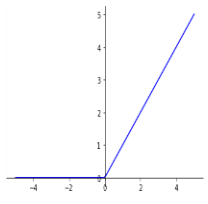
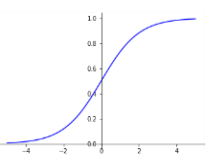
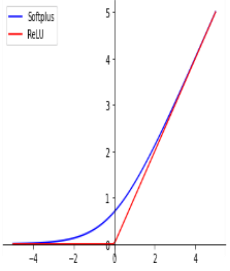
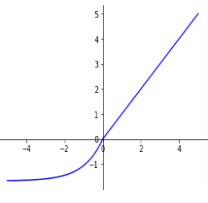
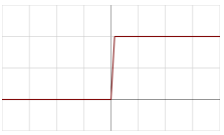
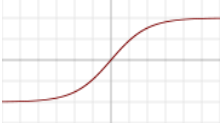
Nom	Graphe	Equation	Dérivé	Étendue
Identité Rampe		$f(x) = x$	$f'(x) = 1$	$\mathbb{R}$
Unité de rectification linéaire (ReLU)		$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$	$\mathbb{R}_+$
Logistique Sigmoïde		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$	$\{0; 1\}$
Unité de rectification linéaire douce (SoftPlus)		$f(x) = \ln(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$	$\mathbb{R}_+$
Unité exponentielle linéaire (ELU)		$f(x) = \begin{cases} a(e^x - 1) & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + a & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases}$	$[-a, +\infty[$
Marche		$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{si } x \neq 0 \\ ? & \text{si } x = 0 \end{cases}$	$\{0, 1\}$
Tangente hyperbolique		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$	$[-1, 1]$

Tableau 1.1. Les fonctions d'activation

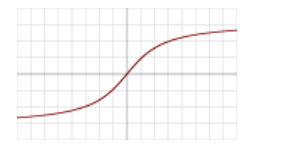
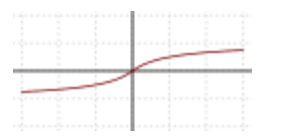
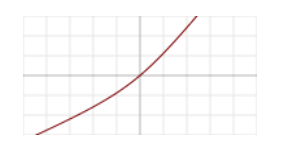
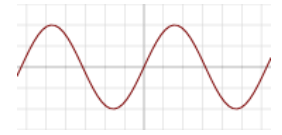
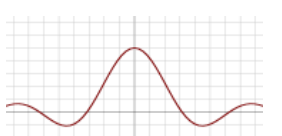
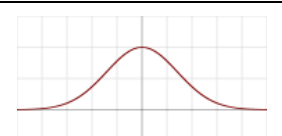
<b>Arc tangente</b>		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$	$[-\frac{\pi}{2}, \frac{\pi}{2}]$
<b>Signe doux</b>		$f(x) = \frac{x}{1 +  x }$	$f'(x) = \frac{1}{(1 +  x )^2}$	$[-1, 1]$
<b>Identité courbée</b>		$f(x) = \frac{\sqrt{x^2 + 1} - 1}{2} + x$	$f'(x) = \frac{x}{2\sqrt{x^2 + 1}} + 1$	$\mathbb{R}$
<b>Sinusoïde</b>		$f(x) = \sin(x)$	$f'(x) = \cos(x)$	$[-1, 1]$
<b>Sinus cardinal</b>		$f(x) = \begin{cases} \sin(x) & \text{si } x \neq 0 \\ x & \text{si } x = 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{si } x = 0 \\ \cos(x) - \frac{\sin(x)}{x^2} & \text{si } x \neq 0 \end{cases}$	$[-0, 217234\dots; 1]$
<b>Fonction gaussienne</b>		$f(x) = e^{-x^2}$	$f'(x) = -2xe^{-x^2}$	$]0, 1]$

Tableau 1.2. Fonctions d'activation

## 4. Les règles d'apprentissage

### 4.1. La règle de Hebb

C'est une technique d'apprentissage pour les réseaux de neurones permettant l'apprentissage par renforcement. Elle permet de modifier les poids des connexions entre les neurones. Cette technique s'appuie sur le postulat suivant [20]:

- Si un réseau effectue une action correcte, alors on conforte cette action. Cela est décliné de la manière suivante :
- Si la sortie de deux neurones connectés est similaire, les poids de leur connexion doit augmenter.
- Si la sortie de deux neurones connectés est différente, les poids de leur connexion doit décroître.

La variation introduite dans les poids des connexions entre le neurone a et le neurone b est formalisée par l'équation :

$$\Delta w_{ab} = \alpha \cdot y_a x_b$$

Où  $w_{ab}$  représente le poids reliant les neurones a et b,  $\Delta w_{ab}$  est la variation de  $w_{ab}$ ,  $y_a$  est la sortie du neurone a,  $x_b$  représente l'entrée du neurone b et  $\alpha$  est une constante modélisant la vitesse d'apprentissage.

L'application de la règle de Hebb est possible en assumant deux hypothèses[20] :

- Le robot explore son environnement et rencontre des situations pour lesquelles il obtient des entrées qui le permettent à son réseau de neurones d'estimer un ensemble de sorties;
- Le robot reçoit des informations lui indiquant si son comportement est bon ou non[20].

## 5. Réseau de neurones artificiels

Un réseau de neurones artificiels est un système de technologie de l'information basé sur le fonctionnement du cerveau humain, dont sont équipés les ordinateurs dotés de fonctions d'intelligence artificielle.

Les réseaux de neurones artificiels (Artificial neural networks en anglais) constituent une branche spécifique de la recherche en informatique et en neuro-informatique.

Il existe différents types de réseaux de neurones artificiels, offrant différentes possibilités de traitement de l'information.

Les réseaux de neurones artificiels permettent aux ordinateurs de résoudre des problèmes de façon autonome et renforcent leurs capacités d'une manière générale. Certains nécessitent une supervision initiale, en fonction de la méthode d'intelligence artificielle utilisée[21].

### 5.1. Les différents types de réseaux de neurones artificiels

On fait appel à différentes structures de réseaux de neurones artificiels en fonction de la méthode d'apprentissage utilisée et de l'objectif recherché.

#### 5.1.1. Perceptron

À l'origine, la forme la plus simple de réseau neuronal artificiel était composée d'un seul neurone modifié par des pondérations et doté d'une valeur seuil. Désormais, le terme « Perceptron » désigne aussi les réseaux à propagation avant à couche unique[21].

### 5.1.2. Réseaux de neurones à propagation avant

Un réseau de neurones artificiels à propagation avant ne peut transmettre l'information que dans un unique sens de traitement. Les réseaux peuvent être monocouches, c'est-à-dire constitués uniquement de couches d'entrée et de sortie, ou multicouches, c'est-à-dire disposant d'un certain nombre de couches cachées[21].

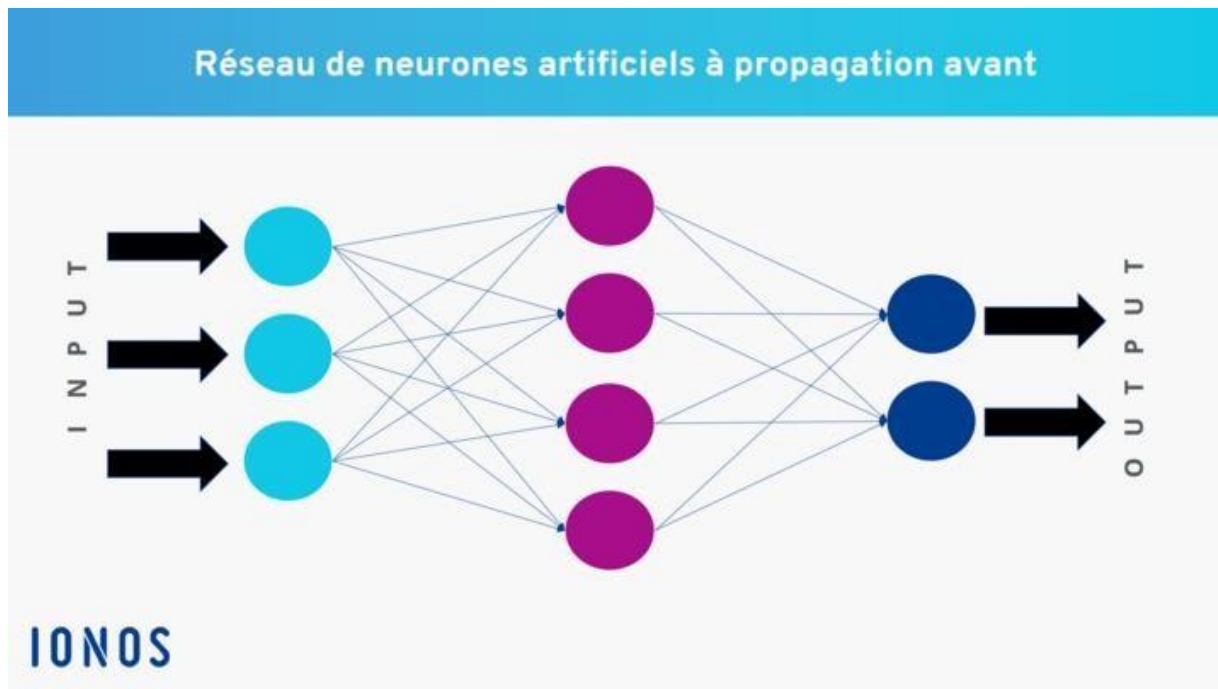


Figure 9: Réseaux de neurones à propagation avant

### 5.1.3. Réseaux de neurones récurrents

Dans les réseaux de neurones récurrents, il est possible de faire passer l'information dans des boucles de rétroaction, et ainsi de la faire revenir vers une couche précédente. Ces rétroactions permettent au système de se constituer une mémoire. Les réseaux de neurones récurrents sont utilisés par exemple en matière de reconnaissance vocale, de traduction et de reconnaissance d'écriture manuscrite[21].

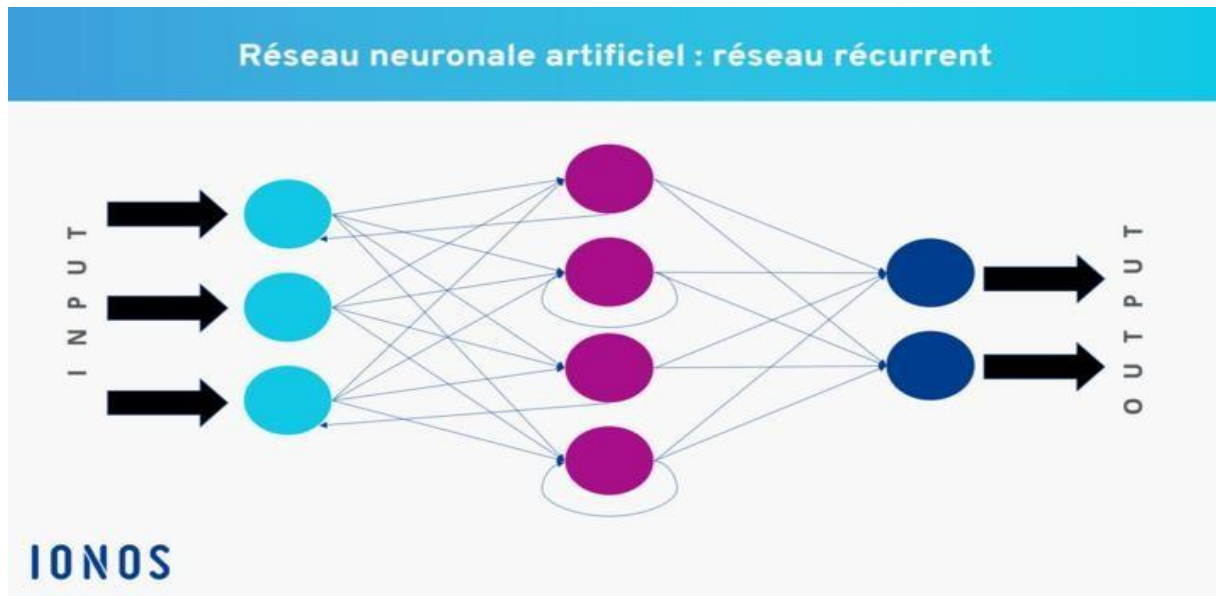


Figure 10: Réseau de neurone récurrent

#### 5.1.4. Réseaux de neurones convolutifs

Un réseau neuronal convolutif est un type de réseau multicouche. Il est composé d'un minimum de cinq couches. Il est procédé sur chacune de ces couches à une reconnaissance de motif. Le résultat obtenu sur chaque couche est transmis à la couche suivante. Ce type de réseau neuronal artificiel est utilisé en matière de reconnaissance d'images[21].

#### 5.2. Apprentissage d'un réseau de neurone

L'apprentissage est une étape de développement du réseau de neurones au cours de laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement souhaité.

L'ajustement des poids des liens entre neurones peut se faire selon diverses équations mathématiques, dont la plus connue est sans doute la loi de Hebb[22].

- Le choix de l'équation d'adaptation des poids dépend largement de la topologie du réseau de neurones utilisé.
- Il est également possible de faire évoluer l'architecture du réseau, c'est-à-dire le nombre et l'interconnexion des neurones, mais en utilisant d'autres types d'algorithmes d'apprentissage. Il existe plusieurs méthodes d'apprentissage dont les plus courantes sont :
  - L'apprentissage supervisé ;
  - L'apprentissage non supervisé.
  - L'apprentissage renforcé Ces trois types sont classés selon :

- (1) La présence ou l'absence d'un « enseignant » apprenant ;
- (2) L'information procurée par le système à étudier.

La figure suivante montre le classement hiérarchique des méthodes d'apprentissage sus citées.

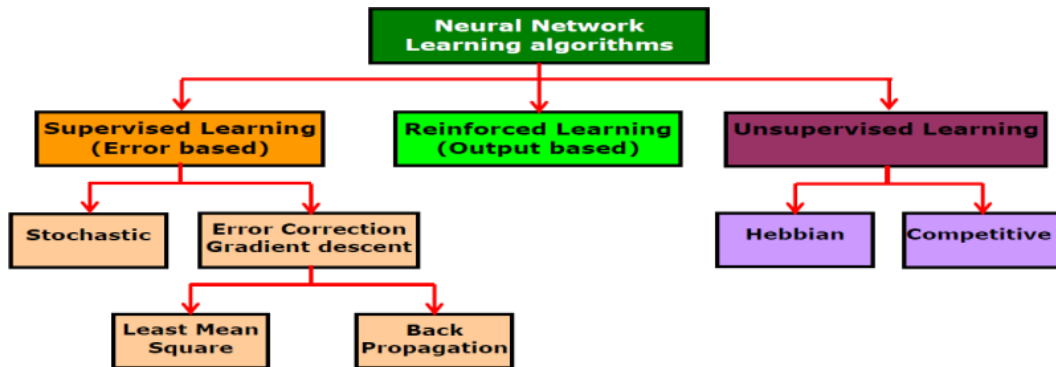


Figure 11: le classement hiérarchique des méthodes d'apprentissage

### 5.2.1. Apprentissage supervisé (La rétropropagation)

Cet algorithme d'apprentissage ne peut être utilisé que lorsque les combinaisons d'entrée sorties désirés sont connues[22].

1. L'enseignant (entrée /sortie) est présent durant le processus d'apprentissage et présente les résultats attendus.
2. Chaque modèle d'entrée est utilisé pour entraîner le réseau.
3. Le processus d'apprentissage est basé sur la comparaison entre la sortie obtenue par le réseau et la sortie désirée générant ainsi une erreur.
4. L'erreur est utilisée pour ajuster les paramètres du réseau (les poids), ce qui améliorera les performances.

**En résumé :** Un apprentissage est dit supervisé lorsqu'on force le réseau à converger vers un état final précis.

### 5.2.2. Apprentissage non-supervisé

Il n'y a pas de connaissances a priori des sorties désirées pour des entrées données. En fait, c'est de l'apprentissage par exploration où l'algorithme d'apprentissage ajuste les poids des liens entre neurones de façon à maximiser la qualité de classification des entrées. Les RNA

qui utilisent ce type d'apprentissage sont appelés «auto-organisatrice» où ce type d'apprentissage possède souvent une moindre complexité dans le calcul par rapport à l'apprentissage supervisé[22].

### **5.2.3. Apprentissage par renforcement**

Dans ce cas, bien que les sorties idéales ne soient pas connues directement, il y a un moyen quelconque de connaître si les sorties du RNA s'approchent ou s'éloignent du but visé. Ainsi, les poids sont ajustés de façons plus ou moins aléatoire et la modification est conservée si l'impact est positif ou rejetée sinon[22].

## **6. Perceptron multi couche**

Le perceptron multicouche est un réseau orienté de neurones artificiels organisé en couches et où l'information voyage dans un seul sens, de la couche d'entrée vers la couche de sortie. La figure 16 donne l'exemple d'un réseau contenant une couche d'entrée, deux couches cachées et une couche de sortie. La couche d'entrée représente toujours une couche virtuelle associée aux entrées du système. Elle ne contient aucun neurone. Les couches suivantes sont des couches de neurones. Dans l'exemple illustre, il y a 3 entrées, 4 neurones sur la première couche cachée, trois neurones sur la deuxième et quatre neurones sur la couche de sortie. Les sorties des neurones de la dernière couche correspondent toujours aux sorties du système. Dans le cas général, un perceptron multicouche peut posséder un nombre de couches quelconque et un nombre de neurones (ou d'entrées) par couche également quelconque[23].

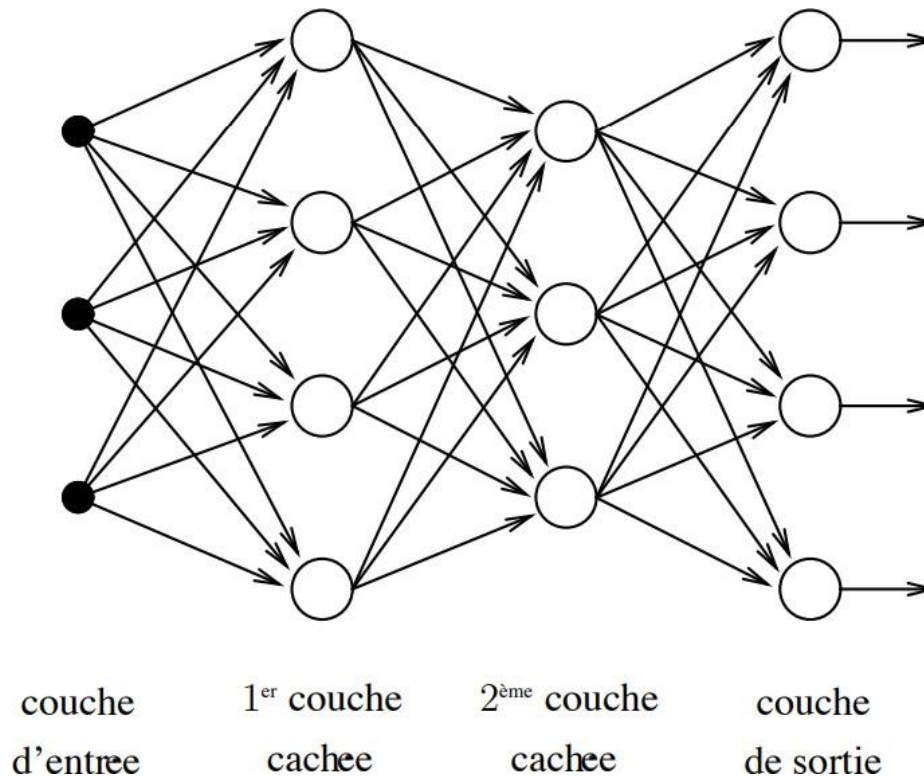


Figure 12: exemple d'un réseau de type perceptron multicouche

Les neurones sont reliés entre eux par des connexions pondérées. Ce sont les poids de ces connexions qui gouvernent le fonctionnement du réseau et “programment” une application de l’espace des entrées vers l’espace des sorties à l’aide d’une transformation non linéaire. La création d’un perceptron multicouche pour résoudre un problème donne donc par l’inférence de la meilleure application possible telle que définie par un ensemble de données d’apprentissage constituées de paires de vecteurs d’entrées et de sorties désirées. Cette inférence peut se faire, entre autres, par l’algorithme dit de rétropropagation[23].

## 7. L’algorithme d’apprentissage rétropropagation du gradient

Cet algorithme permet de réaliser un apprentissage du réseau de neurones. On cherche à obtenir du réseau une réponse préétablie comme étant correcte. On dispose d’une base de connaissance de type entrée (p) - sortie attendue (d). On compare ensuite la sortie (a) à la sortie attendue. On introduit une fonction  $e = d - a$  qui est la fonction d’erreur que l’on va chercher à minimiser en modifiant les poids du réseau. Une fois les poids définis par cet algorithme à partir d’exemples connus de l’utilisateur, on va chercher à extrapoler le réseau, en lui fournissant des entrées inconnues[24].

### 7.1. Méthode du gradient

On cherche une méthode itérative permettant de minimiser une fonction  $f$  (qui correspondra à l'erreur du réseau de neurones)[24]

$$f: \begin{cases} \mathbb{R}^n \rightarrow \mathbb{R} \\ x \rightarrow f(x) \end{cases}$$

On cherche à trouver un  $\Delta x$  tel que  $f(x + \Delta x) < f(x)$ . On applique la formule de Taylor à l'ordre 1 en considérant que  $\Delta x$  est "petit" :

$$f(x + \Delta x) - f(x) = \langle \nabla_x f | \Delta x \rangle + o(\|\Delta x\|)$$

En écrivant  $\Delta x = \eta y$ , avec  $\eta > 0$  avec  $\eta > 0$  un pas fixé arbitrairement, il nous suffit de trouver un vecteur  $y$  qui satisfasse l'équation :

$$\eta \langle \nabla_x f | y \rangle < 0$$

Qui équivaut à :

$$\langle \nabla_x f | y \rangle < 0 \quad (1)$$

Ainsi tout vecteur  $y$  respectant la relation 1 conviendra. On appelle ce vecteur la direction de descente. Afin de s'assurer que le produit scalaire est bien négatif, on choisit  $y = -\nabla_x f$ .

On obtient alors comme vecteur  $\Delta x$  :

$$\Delta x = -\eta \nabla_x f \quad (2)$$

Où  $\eta$  est un pas "petit", fixé arbitrairement.

En itérant le processus, on fait diminuer  $f(x)$ . Il a été démontré que dans le cas d'une fonction  $f$  fortement convexe et lipschitzienne la méthode du gradient conduit (au bout d'un nombre infini d'itération) à trouver le minimum global de la fonction, quel que soit la valeur du  $x$  initial. Bien que les fonctions considérées ici n'entrent presque jamais dans le cadre de ces hypothèses, on considérera qu'elles conviennent et on s'assurera de manière expérimentale (dans l'algorithme) que la méthode converge effectivement[24].

### 7.2. Rétro propagation des erreurs

On va chercher à minimiser la fonction d'erreur du réseau de neurones. Plus précisément, on va ici chercher à minimiser l'erreur de la couche  $k$  du perceptron[24].

### 7.2.1. Expression de la direction de descente

$m$  : Nombre de neurones sur la couche  $k$

$p$  : Nombre d'entrées de la couche  $k$

$d$  : Sortie attendue à la dernière couche du réseau

$a$  : Sortie de la dernière couche du réseau

$a^k \in \mathbb{R}^m$  : Sortie de la couche  $k$  du réseau

$W \in M_{m,p}(\mathbb{R})$  : Matrice des poids de la couche  $k$

$b \in \mathbb{R}^m$  : Vecteur biais de la couche  $k$

$f$  : Fonction d'activation de la couche  $k$ ,  $f: \mathbb{R} \rightarrow \mathbb{R}$

Par abus, si  $f: \mathbb{R} \rightarrow \mathbb{R}$ , on notera  $f \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$

D'après la construction du perceptron multicouche on a :

$$a^k = f(n^k(W, b))$$

Avec

$$n^k(W, b) = W a^{k-1} - b$$

On peut maintenant définir la fonction d'erreur que l'on va chercher à minimiser. On la choisit dépendant uniquement de  $W$  puisque qu'on ne s'intéresse ici qu'à modifier les poids de la couche.

$$f(W, b) = \langle d - a | d - a \rangle$$

D'après la méthode du gradient (section 0.1), une itération pour minimiser  $F$  sera de modifier  $(W, b)$  en  $(W, b) + (\Delta W, \Delta b)$ , avec  $(\Delta W, \Delta b) = -\eta \nabla F_W$  avec  $\eta > 0$  choisi arbitrairement.

On va maintenant chercher à modifier séparément chaque coefficient  $w_{i,j}$  de la matrice  $W$

Cela revient à modifier le poids de l'entrée  $j$  du neurone  $i$ .

On a:

$$\Delta w_{i,j} = -\eta \frac{\partial F}{\partial w_{i,j}}(w, b)$$

Pour calculer  $\frac{\partial F}{\partial w_{i,j}}$  on peut voir  $F$  comme une fonction de  $n^k$ , et  $n^k$  une fonction de  $W$  et  $b$ .

Cela se traduit par l'introduction d'une fonction  $C$  et de l'écriture de  $F$  comme une composée :

$$F = C \circ n^k$$

Ou  $C: \mathbb{R}^P \rightarrow \mathbb{R}$  et  $n^k : \begin{cases} M_{m,p}(\mathbb{R}) \times \mathbb{R}^p \rightarrow \mathbb{R}^P \\ W \mapsto Wa^{k-1} - b^k \end{cases}$

En dérivant cette composée, on a :

$$\frac{\partial F}{\partial w_{i,j}}(W, b) = \frac{\partial C}{\partial n_i^k} \frac{\partial n_i^k}{\partial w_{i,j}}(W, b)$$

Ou  $\frac{\partial C}{\partial n_i^k} = \frac{\partial C}{\partial n_i}(n^k(W, b))$

On va calculer séparément les deux facteurs.

$$\frac{\partial n_i^k}{\partial w_{i,j}}(W, b) = \frac{\partial}{\partial w_{i,j}} [Wa^{k-1} - b^k] = a_j^{k-1}$$

On note le second  $s_k^i = \frac{\partial C}{\partial n_i^k}$ , on le calculera dans la section 0.2.2. On a donc comme

expression de  $\partial w_{i,j}$

$$\partial w_{i,j} = \eta s_i^k a_j^{k-1} \quad (3)$$

En refaisant exactement les mêmes calculs en dérivant par rapport à  $b_i$  (en s'intéressant donc à la modification du biais), on obtient :

$$\Delta b_i = -\eta s_i^k \quad (4)$$

Pour résumer, en réécrivant les relations (3) et (4) sous forme matricielle, on a obtenu notre direction de descente[24] :

$$\Delta W = -\eta s^{k,t} (a^{k-1}) \quad (5)$$

$$\Delta b = \eta s^k \quad (6)$$

### 7.2.2. Calcul de la sensibilité $s^k$

Soit N le nombre de couches du réseau. La sensibilité de la dernière couche N est facile à exprimer :

$$s_i^N = \frac{\partial c}{\partial n_i^N} = \frac{\partial}{\partial n_i^N} [(d_i - f(n_i^N))^2] = -2(d_i - a_i) f'(n_i^N) \quad (7)$$

On peut alors exprimer C :

$$\bar{C}(n^k) = f^N(n^N \left( f^{N-1} \left( \dots f^{k+1} \left( n^{k+1} \left( f^k(n^k) \right) \dots \right) \right) \right) \right)$$

Donc en posant  $n^{k+1} : \left| \begin{array}{c} \mathbb{R}^P \rightarrow \mathbb{R}^{p'} \\ n^k \rightarrow W^{k+1} f^k(n^k) - b^{k+1} \end{array} \right.$ , on a  $D: \mathbb{R}^P \rightarrow \mathbb{R}^{p'}$

Telle que :  $C = D \circ n^{k+1}$

Donc

$$s_i^k = \frac{\partial C}{\partial n_i^k} = \frac{\partial D}{\partial n_k^{k+1}} \frac{\partial n_k^{k+1}}{\partial n_i^k} = s_j^{k+1} \frac{\partial n_k^{k+1}}{\partial n_i^k}$$

Reste à calculer le second facteur de cette expression :

$$\frac{\partial n_k^{k+1}}{\partial n_i^k} = \frac{\partial C}{\partial n_i^k} \left[ \sum_l w_{j,l}^{k+1} f^k(n_l^k) - b_j^{k+1} \right] = w_{j,l}^{k+1} f'_k(n_i^k)$$

On a donc une relation de récurrence entre  $s^k$  et  $s^{k+1}$  (d'où le terme "rétropropagation"), le premier terme  $s^N$  étant donné par l'équation (7)[24] :

$$s^k = f'_k(n^k) \cdot {}^t W^{k+1} s^{k+1} \quad (8)$$

$$s^N = -2 f'_N(n^N) (d - a) \quad (9)$$

## 8. Les réseaux de neurones artificiels et leurs applications

Aujourd'hui, les réseaux de neurones ont de nombreuses applications dans des domaines très variés[25] :

- traitement d'image : compression d'images, reconnaissance de caractères et de signatures, reconnaissance de formes et de motifs, chiffrement, classification;
- traitement du signal : traitement de la parole, identification de sources, filtrage, classification;

- traitement automatique des langues : segmentation en mots, représentation sémantique des mots (plongements lexicaux), étiquetage morpho-syntaxique, traduction automatique;
- contrôle : diagnostic de pannes, commande de processus, contrôle qualité, robotique;
- optimisation : allocation de ressources, planification, régulation de trafic, gestion, finance;
- simulation : simulation boîte noire, prévisions météorologiques ;
- classification d'espèces animales étant donnée une analyse ADN ;
- modélisation de l'apprentissage et perfectionnement des méthodes de l'enseignement;
- approximation d'une fonction inconnue ou modélisation d'une fonction connue mais complexe à calculer avec précision.

## 9. Applications dans diverses industries

### 9.1. En gestion et finance

Industrie	Application
Banque	Prêts et scoring
Cartes de crédit	Détection des fraudes
Finance	Analyse d'investissements et de fluctuations des taux de change
Assurance	Couverture assurantielle et estimation des réserves
Marketing	Ciblage des prospections, mesures et comparaisons des campagnes et des méthodes

Tableau 3: Réseaux de neurones en gestion et finance

### 9.2. Reconnaissance de formes

La reconnaissance de formes est utile pour comprendre le fonctionnement d'un réseau de neurones et sa capacité d'apprentissage. En premier lieu, on « fait apprendre » au réseau de neurones une forme, une image, une silhouette. On peut alors présenter au système une des images qu'il a apprises soit incomplète soit erronée, et celui-ci sera capable de reconnaître de quoi il s'agit[25].

### 9.3. Dans divers domaine

<b>Industrie</b>	<b>Application</b>
Médecine	Diagnostic médicaux
Pharmacie	Efficacité de nouveaux médicaments
Psychologie	Prévisions comportementales
Immobilier	Études de marchés
Recherche scientifique	Identification de spécimens, séquençages de protéines
Télécommunication	Détection des pannes de réseaux
Environnement	Prévisions de la qualité de l'air et de l'eau

Tableau 4: Réseaux de neurones dans divers domaine

## **10. Conclusion**

Dans ce chapitre, nous avons d'abord parlé les réseaux de neurones artificiels: définition et architecture ainsi que la fonction d'activation, d'apprentissage, la règle d'apprentissage rétropropagation et l'intelligence artificiel et leurs application.

# Chapitre 03 : Analyse et conception

## **1. Introduction**

Dans ce chapitre d'Analyse et Conception de notre système, nous allons aborder les aspects essentiels de la structuration et de la planification du système. Nous allons commencer par identifier l'organigramme du système, qui permettra d'avoir une vue globale des différentes parties et de leur interconnexion. Ensuite, nous examinerons le diagramme de cas d'utilisation, qui décrira les différentes actions que les utilisateurs pourront effectuer avec l'application. Enfin, nous créerons un diagramme de flux de données pour visualiser le mouvement des données à travers le système, en identifiant les entrées, les sorties et les transformations. Ces éléments d'analyse et de conception seront essentiels pour la compréhension de notre système et pour guider le processus de développement.

## **2. Organigramme d'un système**

Un organigramme présente la structure interne d'une organisation ou d'une entreprise ou d'un système[26].

## 2.1. Présentation d'organigramme du notre système

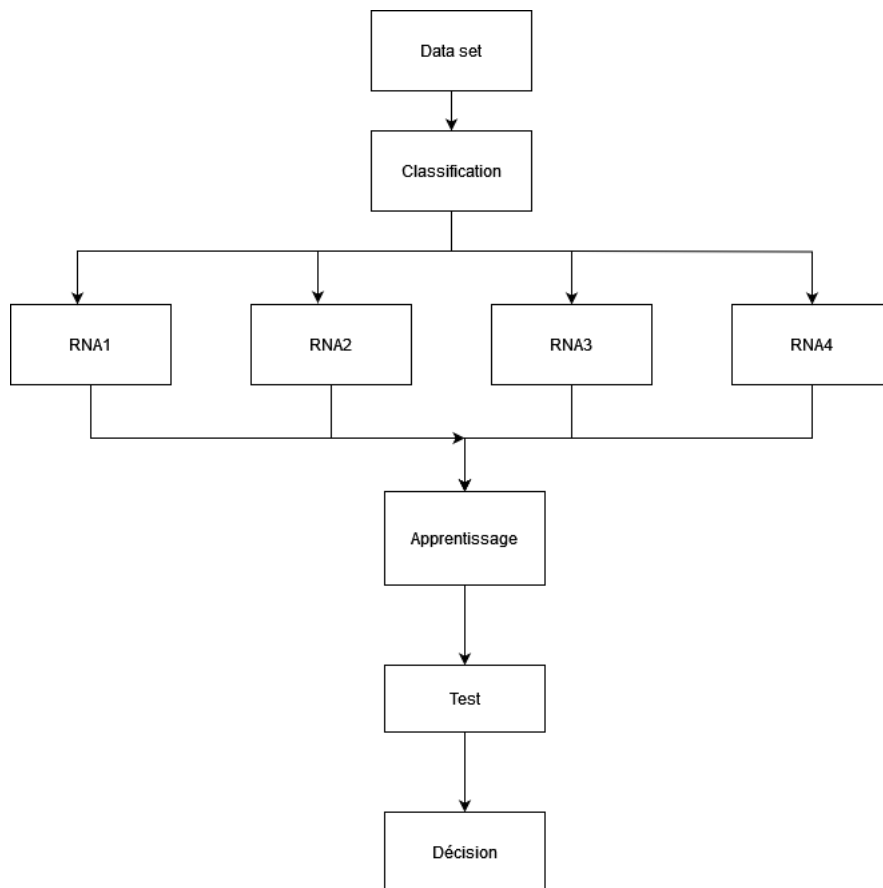


Figure 13: organigramme du notre système

## 3. Modélisation UML

### 3.1. Définition d'UML

- UML (*Unified Modeling Language* ou Langage de modélisation unifiée en français) est un langage graphique de modélisation informatique. Ce langage est désormais la référence en modélisation objet, ou programmation orientée objet[27].
- L'UML est constitué de diagrammes qui servent à visualiser et décrire la structure et le comportement des objets qui se trouvent dans un système. Il permet de présenter des systèmes logiciels complexes de manière plus simple et compréhensible qu'avec du code informatique. L'UML a des applications dans le développement logiciel, mais aussi dans l'industrie (pour modéliser les flux de processus par exemple), dans l'ingénierie ou le marketing[27].

## 4. Les différents types de diagrammes UML

L'UML définit 14 types de diagrammes divisés en deux catégories[28].

### 4.1. Les diagrammes de structure

représentent les éléments du système, leurs propriétés et leurs relations entre eux[28] :

- diagramme de classes ;
- diagramme d'objets ;
- diagramme de composants ;
- diagramme de structure composite ;
- diagramme d'ensemble ;
- diagramme de déploiement ;
- diagramme de profil.

### 4.2. Les diagrammes de comportement

représentent les processus et les interactions entre les objets[28] :

- diagramme de cas d'utilisation ;
- diagramme d'activité ;
- diagramme d'état-transition ;
- diagramme de séquence ;
- diagramme de communication ;
- diagramme de temps ;
- diagramme d'aperçu d'interaction.

## 5. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système.

Le diagramme de cas d'utilisation décrit les fonctions générales et la portée d'un système. Ce diagramme identifie également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans le diagramme de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne[30].

## 5.1. Diagramme de cas d'utilisation du notre système

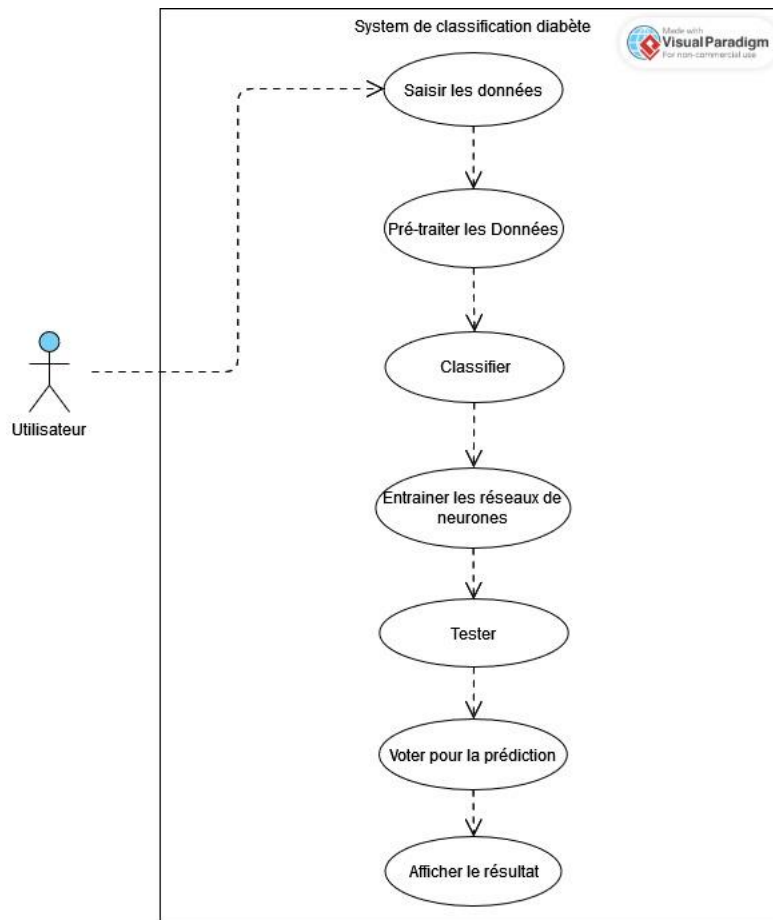


Figure 14:diagramme de cas d'utilisation du système

## 6. Diagramme de flux de donnée

### 6.1. Définition

Un diagramme de flux de données schématise le flux d'informations de n'importe quel processus ou système. Il utilise des symboles définis comme des rectangles, des cercles et des flèches, ainsi que des étiquettes aux libellés courts, pour représenter les entrées de données, les sorties, les points de stockage et les trajets entre chaque destination. Les organigrammes de données peuvent aller de vues d'ensemble de processus simples, voire dessinées à la main, à des diagrammes complexes sur plusieurs niveaux qui approfondissent progressivement le traitement des données. On peut les utiliser pour analyser un système existant ou en modéliser un nouveau. Comme tous les diagrammes et graphiques de qualité, un diagramme de flux de données peut souvent « dire » visuellement des choses qui seraient difficiles à expliquer avec des mots[31].

## 6.2. DFD du notre système

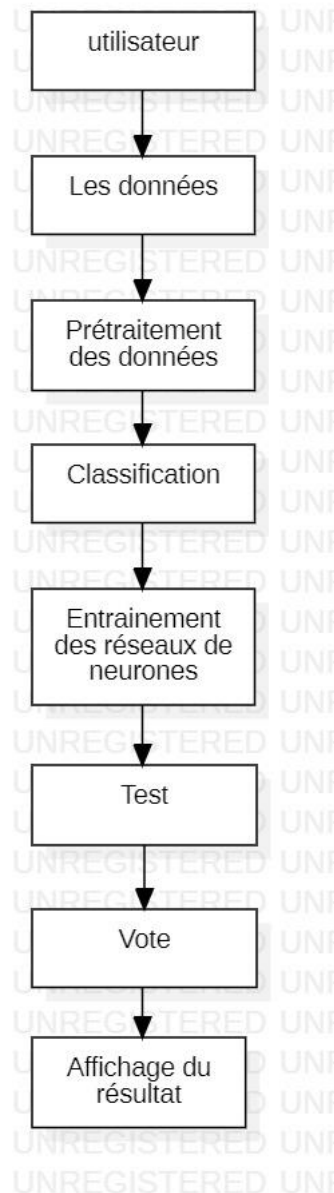


Figure 15: Diagramme de flux de donnée

Ce diagramme illustre comment les données circulent à travers les différentes étapes de l'application, en montrant les interactions entre l'utilisateur, les processus de prétraitement, de classification, de vote et d'affichage des résultats.

## 6.3. La description

- L'utilisateur saisit les données (par exemple, les valeurs des caractéristiques du patient) ;
- Les données saisies sont prétraitées ;

- Les données prétraitées sont utilisées pour la classification, où un modèle de classification prédit la classe (diabétique ou non diabétique) pour chaque exemple ;
- Les prédictions des différents modèles sont votées pour obtenir une prédiction finale ;
- Les résultats (prédiction diabétique ou non diabétique) sont affichés à l'utilisateur.

## **7. Conclusion**

Dans ce chapitre, nous avons identifié les principaux éléments qui structureront notre système. L'organigramme du système nous a donné une vue d'ensemble de son architecture, tandis que le diagramme de cas d'utilisation a défini les actions et fonctionnalités des utilisateurs. Le diagramme de flux de données a mis en évidence le mouvement des données à travers le système. Ces éléments fournissent une base solide pour le développement de l'application, en nous permettant de comprendre sa structure, ses fonctionnalités et le flux des données. Ils serviront de guide pour répondre efficacement aux besoins des utilisateurs lors de la phase de développement.

# Chapitre 04 : Réalisation

## 1. Introduction

Dans ce chapitre, nous aborderons l'implémentation de notre application , qui consiste à utiliser les réseaux de neurones pour la classification du diabète. Nous allons maintenant passer de la théorie à la pratique en mettant en œuvre les différentes étapes nécessaires pour construire notre application.

## 2. Présentation des outils de développement

### 2.1. Python

Python est un langage de programmation inventé par Guido van Rossum. La première version de python est sortie en 1991[32].

Python est un langage de programmation interprété, c'est-à-dire qu'il n'est pas nécessaire de le compiler avant de l'exécuter. Si vous avez déjà touché un peu à la programmation, vous verrez que ce langage possède une certaine poésie. Les programmeurs s'amuse souvent à trouver la manière la plus élégante et efficace d'écrire une suite d'instructions[32].



Figure 16: Logo du python

L'environnement python est riche en bibliothèques. Vous trouverez toujours des projets open source qui vous faciliteront la vie.

Python est également très populaire dans le domaine de l'IA en raison de ses bibliothèques de haute qualité et de sa puissance de calcul. Beaucoup de chercheurs et de développeurs utilisent Python pour implémenter et expérimenter avec de nouveaux algorithmes

d'apprentissage automatique et de deep learning, en particulier grâce aux frameworks tels que TensorFlow et PyTorch. En utilisant Python, il est possible de développer rapidement des modèles complexes et puissants, ce qui en fait un choix idéal pour les projets d'IA de toutes tailles.

## 2.2. Bibliothèques Python

Les bibliothèques ne sont rien d'autre que des collections de modules avec du code pré-écrit et peuvent être facilement importées et utilisées par les développeurs pour implémenter n'importe quelle fonctionnalité. Le ML nécessite un traitement de données régressif et continu, et Python répond à cette exigence en accédant aux données, en les manipulant et en les transformant[33].



Figure 17: Bibliothèque du python

### 2.2.1. TensorFlow

TensorFlow est une bibliothèque de calcul numérique open-source pour le machine learning basé sur les réseaux neuronaux. En 2015, l'équipe de recherche Google Brain l'a créée pour l'utiliser en interne dans les produits Google[33].



Figure 18: TensorFlow logo

### 2.2.2. NumPy

Python n'a pas été initialement développé comme un outil de calcul numérique. Cependant, l'arrivée de NumPy a été la clé de l'expansion des capacités de Python avec des fonctions mathématiques, sur la base desquelles des solutions de machine learning seraient construites[33].



Figure 19: NumPy logo

Elle traite l'algèbre linéaire, les calculs matriciels, la génération de nombres aléatoires, etc. qui aident les développeurs à créer des systèmes intelligents et réactifs.

Les fonctions mathématiques de haut niveau sont exécutées sur des tableaux, ce qui permet aux algorithmes Python de s'exécuter plus rapidement.

En cas de problème, les programmeurs peuvent se tourner vers la communauté NumPy et partager leurs expériences ou trouver une solution toute faite à leurs problèmes.

### 2.2.3. Pandas

Pandas est une bibliothèque Python de bas niveau construite sur NumPy. Pandas est surtout utilisé pour l'analyse et la manipulation des données, ainsi que pour les opérations de machine learning sous la forme de dataframes. En utilisant les dataframes, les développeurs peuvent facilement avoir une vue d'ensemble des données pour garantir une meilleure qualité du produit[33].



Figure 20: pandas logo

Les développeurs utilisent cette bibliothèque pour structurer, remodeler et filtrer de grands ensembles de données.

#### 2.2.4. Scikit-Learn

Scikit-learn, encore appelé *sklearn*, est la bibliothèque la plus puissante et la plus robuste pour le machine learning en Python. Elle fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression et le clustering via une interface cohérente en Python[34].



Figure 21: Scikit-learn logo

#### 2.2.5. Keras

Keras est une bibliothèque open source écrite en Python (sous licence MIT) basée principalement sur les travaux du développeur de Google François Chollet dans le cadre du projet ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). Une première version du logiciel multiplateforme a été publiée le 28 mars 2015. Le but de cette bibliothèque est de permettre la constitution rapide de réseaux neuronaux. Dans ce cadre, Keras ne fonctionne pas comme un framework propre mais comme une interface de programmation applicative (API) pour l'accès et la programmation de différents frameworks d'apprentissage automatique. Theano, Microsoft Cognitive Toolkit (anciennement CNTK) et TensorFlow font notamment partie des frameworks pris en charge par Keras[35].

### 2.3. Editeur de code

Colaboratory, souvent raccourci en "Colab", est un produit de Google Research. Colab permet à n'importe qui d'écrire et d'exécuter le code Python de son choix par le biais du navigateur. C'est un environnement particulièrement adapté au machine learning, à l'analyse de données et à l'éducation. En termes plus techniques, Colab est un service hébergé de notebooks Jupyter

qui ne nécessite aucune configuration et permet d'accéder sans frais à des ressources informatiques, dont des GPU[36].

### 3. La base de donnéé

On a utilisé une base de donnée banchmark qui a été téléchargé depuis le site <https://www.kaggle.com/search?q=diabete>, qui est: Pima. Le choix de cette base est justifié par les critères suivants :

- La taille de la base.
- Nombre des attributs.
- Nombre de classe.

#### 3.1. Description de la base de données

La base de données sur le diabète des Indiens Pima comprend 768 **cas**, dont 268 sont diabétiques et 500 ne sont pas diabétiques. Chaque cas se compose de 9 attributs, dont 8 représentent des facteurs de risque **9** représente la classe du patient 0 ou 1, 0 indiquant un patient non diabétique et 1 indiquant un patient diabétique.

#### 3.2. séparation ou la division de la base de donnéé

Les données d'ensemble sont séparées en deux parties : les entrées (X) et les sorties (Y). Les entrées (X) correspondent aux données d'entrée, tandis que les sorties (Y) représentent les résultats attendus. Pour la partie de test, elle représentera 30% de l'ensemble de données total, tandis que les 70% restants seront utilisés pour entraîner les réseaux neurones.

```
# Diviser les données en ensemble d'entraînement et ensemble de test
X = df.drop('Outcome', axis=1)
y = df['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figure 22: La division de la base de donnéé

Contenu du dataset de test :

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	98	58	33	190	34.0	0.430	43	0
2	112	75	32	0	35.7	0.148	21	0
2	108	64	0	0	30.8	0.158	21	0
8	107	80	0	0	24.6	0.856	34	0
7	136	90	0	0	29.9	0.210	50	0
6	103	72	32	190	37.7	0.324	55	0
1	71	48	18	76	20.4	0.323	22	0
0	117	0	0	0	33.8	0.932	44	0
4	154	72	29	126	31.3	0.338	37	0
5	147	78	0	0	33.7	0.218	65	0
10	111	70	27	0	27.5	0.141	40	1
7	179	95	31	0	34.2	0.164	60	0
4	148	60	27	318	30.9	0.150	29	1
5	96	74	18	67	33.6	0.997	43	0
2	88	58	26	16	28.4	0.766	22	0
1	125	50	40	167	33.3	0.962	28	1
3	84	72	32	0	37.2	0.267	28	0
5	86	68	28	71	30.2	0.364	24	0
4	183	0	0	0	28.4	0.212	36	1
0	140	65	26	130	42.6	0.431	24	1
5	104	74	0	0	28.8	0.153	48	0
1	88	78	29	76	32.0	0.365	29	0
13	106	72	54	0	36.6	0.178	45	0
1	96	64	27	87	33.2	0.289	21	0
10	129	76	28	122	35.9	0.280	39	0
7	184	84	33	0	35.5	0.355	41	1
6	109	60	27	0	25.0	0.206	27	0
1	100	74	12	46	19.5	0.149	28	0
2	102	86	36	120	45.5	0.127	23	1
3	89	74	16	85	30.4	0.551	38	0
0	162	76	56	100	53.2	0.759	25	1

Figure 23: Les premiers lignes du dataset de test

Contenu du nouvel ensemble d'entraînement équilibré :

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	0	100	70	26	50	30.800000	21	0
1	1	80	74	11	60	30.000000	22	0
2	1	84	64	23	115	36.900000	28	0
3	6	194	78	0	0	23.500000	59	1
4	6	123	72	45	230	33.600000	34	0
5	3	78	50	32	88	31.000000	26	1
6	3	106	72	0	0	25.800000	27	0
7	13	129	0	30	0	39.900000	44	1
8	4	129	86	20	270	35.100000	23	0
9	3	61	82	28	0	34.400000	46	0
10	2	81	72	15	76	30.100000	25	0
11	0	102	75	23	0	0.000000	21	0
12	3	99	54	19	86	25.600000	24	0
13	0	107	62	30	74	36.600000	25	1
14	3	170	64	37	225	34.500000	30	1
15	11	85	74	0	0	30.100000	35	0
16	0	102	78	40	90	34.500000	24	0
17	5	97	76	27	0	35.600000	52	1
18	2	127	58	24	275	27.700000	25	0
19	7	133	84	0	0	40.200000	37	0
20	2	94	76	18	66	31.600000	23	0
21	3	83	58	31	18	34.300000	25	0
22	0	123	88	37	0	35.200000	29	0
23	7	187	68	39	304	37.700000	41	1
24	13	152	90	33	29	26.800000	43	1
25	2	129	74	26	205	33.200000	25	0
26	1	106	76	0	0	37.500000	26	0
27	3	111	58	31	44	29.500000	22	0
28	0	111	65	0	0	24.600000	31	0
29	11	143	94	33	146	36.600000	51	1

Figure 24: Les premiers lignes du dataset d'entraînement

## 4. l'apprentissage du réseaux de neurones

Dans notre cas, l'apprentissage est supervisé c'est-à-dire que les réseaux s'adaptent par comparaison entre les résultats qu'ils ont calculés en fonction des entrées fournies, et la réponse attendue en sortie.

```
history = model.fit(X_train, y_train, epochs=250, batch_size=32)
ensemble.append(model)
histories.append(history)
```

Figure 25: Fonction d'apprentissage

Pour ce type d'apprentissage, nous avons utilisé une fonction d'activation de type sigmoïde pour la couche de sortie et une fonction d'activation de type relu pour la couche d'entrée ainsi que pour les trois couches cachées.

Le modèle réseau neuronal est créé de manière séquentielle. Des couches denses (entièrement connectées) sont ajoutées une par une en spécifiant la fonction d'activation Sigmoïde.

Sigmoïde est utilisé à la place de relu car relu donne 1 ou 0, et la sortie doit être une probabilité et sigmoïde donnera un flottant entre 0 et 1, ce qui sera la probabilité prédite d'un diagnostic de diabète positif.

Des réseaux de neurones utilisent la fonction de perte "binary\_crossentropy" pour calculer la perte lors de l'entraînement. et les optimiseurs de gradient d'Adam (optimisés) pour améliorer la précision métrique.

- La fonction d'activation relu a été utilisée pour tous les nœuds à l'exception de ceux de la couche de sortie, qui donne 1 ou 0.
- La couche de sortie utilise l'activation sigmoïde pour donner une valeur comprise entre 0 et 1, ce qui serait la chance prévue de diagnostic.

## 5. Fonction de perte ou de coût

On a utilisé La fonction de perte "binary\_crossentropy" est couramment utilisée pour les problèmes de classification binaire où la sortie du modèle est une probabilité entre 0 et 1. Cette fonction calcule la perte entre les probabilités prédites par le modèle et les étiquettes de classe réelles.

## 6. Optimiseur

L'optimiseur "adam" utilise une variante de la descente de gradient stochastique (SGD) adaptative avec des taux d'apprentissage adaptatifs pour chaque paramètre du modèle. Il est couramment utilisé en raison de sa rapidité de convergence et de son bon comportement dans de nombreux problèmes d'apprentissage en profondeur.

## 7. Procédure de descente de gradient

La procédure de descente de gradient avec l'optimiseur Adam fonctionne en ajustant itérativement les poids et les biais du modèle en utilisant les gradients calculés à partir de la fonction de perte. Adam adapte les taux d'apprentissage de manière adaptative pour chaque paramètre du modèle, ce qui permet d'obtenir une convergence plus rapide et une meilleure performance.

## 8. Test

On a initialement divisé nos données en ensembles d'apprentissage (70 %) et de test (30 %) pour diviser l'ensemble d'apprentissage en ensembles d'apprentissage (70%) et de validation (30 %). Cela nous a permis d'obtenir un ensemble de données distinct pour évaluer les performances du modèle pendant l'entraînement.

```
# Diviser les données en ensemble d'entraînement et ensemble de test
X = df.drop('Outcome', axis=1)
y = df['Outcome']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Figure 26: Division du data set

## 9. Implémentation et résultats obtenus

### 9.1. Description des différentes structures

Nos réseaux de neurones sont implémentés avec plusieurs couches, chacune composée de plusieurs neurones. Dans notre code, nous avons utilisé une couche d'entrée suivie de trois couches cachées, et enfin une couche de sortie. Cette architecture en couches nous permet de capturer les informations et les patterns complexes des données d'entrée, et d'effectuer la classification ou la prédiction souhaitée à travers la couche de sortie.

## 9.2. Différentes Procédures

1. Prétraitement des données : nous avons utilisé le sur-échantillonnage SMOTE pour équilibrer les classes en générant des échantillons synthétiques de la classe minoritaire. Ensuite, nous avons effectué la standardisation des fonctionnalités en utilisant le StandardScaler pour mettre à l'échelle les caractéristiques et les rendre comparables en termes d'échelle. Ces étapes sont importantes pour garantir des résultats cohérents et améliorer les performances de notre modèle de classification.

```
# Effectuer le sur-échantillonnage SMOTE sur l'ensemble d'entraînement
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)

# Standardiser les fonctionnalités
scaler = StandardScaler()
X_train_smote_scaled = scaler.fit_transform(X_train_smote)
X_test_scaled = scaler.transform(X_test)
# Effectuer le sur-échantillonnage SMOTE sur l'ensemble d'entraînement
smote = SMOTE(random_state=42)
X_train_smote, y_train_smote = smote.fit_resample(X_train, y_train)
```

Figure 27: Prétraitement des données

2. Entraînement : nous avons utilisé un ensemble de quatre réseaux de neurones pour mon apprentissage. Chaque réseau était composé d'une couche d'entrée avec huit neurones pour représenter nos caractéristiques, suivi de trois couches cachées avec 16 neurones chacune. Enfin, nous avons ajouté une couche de sortie avec un seul neurone qui utilisait une fonction d'activation sigmoid pour prédire la sortie.

```
for _ in range(4):
    model = keras.Sequential([
        layers.Dense(8, input_shape=(8,)), activation='relu'), # Input layer with 8 neurons
        layers.Dense(16, activation='relu'), # First hidden layer with 16 neurons
        layers.Dense(16, activation='relu'), # Second hidden layer with 16 neurons
        layers.Dense(16, activation='relu'), # Second hidden layer with 16 neurons

        layers.Dense(1, activation='sigmoid') # Output layer with sigmoid activation
    ])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=250, batch_size=32)
ensemble.append(model)
histories.append(history)
```

Figure 28: Fonction d'entraînement du notre réseaux de neurones

```
Epoch 1/250
19/19 [=====] - 4s 7ms/step - loss: 0.6771 - accuracy: 0.6535
Epoch 2/250
19/19 [=====] - 0s 8ms/step - loss: 0.6565 - accuracy: 0.6552
Epoch 3/250
19/19 [=====] - 0s 5ms/step - loss: 0.6315 - accuracy: 0.6552
Epoch 4/250
19/19 [=====] - 0s 2ms/step - loss: 0.6010 - accuracy: 0.6552
Epoch 5/250
19/19 [=====] - 0s 2ms/step - loss: 0.5648 - accuracy: 0.6672
Epoch 6/250
19/19 [=====] - 0s 2ms/step - loss: 0.5313 - accuracy: 0.7067
Epoch 7/250
19/19 [=====] - 0s 2ms/step - loss: 0.5050 - accuracy: 0.7461
Epoch 8/250
19/19 [=====] - 0s 2ms/step - loss: 0.4909 - accuracy: 0.7427
Epoch 9/250
19/19 [=====] - 0s 2ms/step - loss: 0.4828 - accuracy: 0.7547
Epoch 10/250
19/19 [=====] - 0s 3ms/step - loss: 0.4776 - accuracy: 0.7616
```

Figure 29: Entraînement

3. Validation : Après chaque phase d'entraînement, nous avons évalué la performance de nos modèle en utilisant un ensemble de validation distinct. Cela nous permis de surveiller la capacité de notre modèle à généraliser et à éviter le surajustement. Nous avons utilisé des mesures telles que la précision et le rappel pour évaluer la qualité de mes prédictions.

```
# Effectuer la prédiction sur la nouvelle donnée avec chaque modèle de l'ensemble
predictions = []
for model in ensemble:
    prediction = np.round(model.predict(X_test_scaled))
    predictions.append(prediction)

# Calculer le vote majoritaire
final_prediction = np.round(np.mean(predictions, axis=0))
```

Figure 30: Validation

4. Affichage des résultats : nous avons affiché les résultats de chaque étape de mon code pour mieux comprendre la progression de l'apprentissage. Nous avons tracé les courbes de perte (loss) et d'exactitude (accuracy) pour chaque réseau de neurones, ce qui nous a permis d'analyser la performance du modèle au fil du temps.
5. Décision finale : Une fois que notre quatre réseaux de neurones ont été entraînés, nous avons utilisé un mécanisme de vote pour prendre la décision finale concernant la

condition du patient (diabétique ou non). Nous avons comparé les prédictions des quatre réseaux et nous avons pris la décision en fonction de la majorité des votes.

### **9.3. Résultats et discussion**

Nous avons utilisé la fonction fit pour entraîner notre réseaux de neurones sur les données d'apprentissage, permettant ainsi au modèle d'apprendre à partir de ces données spécifiques. En utilisant un certain nombre d'epochs, nous avons itéré sur le processus d'entraînement pour ajuster progressivement les poids du modèle et améliorer ses performances.

Lors de chaque epoch, nous avons effectué une passe avant (forward pass) où les données de test ont été alimentées dans le modèle, et les prédictions ont été comparées avec les valeurs réelles. En utilisant la fonction de perte binaire, nous avons calculé l'erreur et utilisé l'optimiseur Adam pour ajuster les poids du modèle lors de la rétropropagation (backward pass).

Après chaque epoch, nous avons évalué la performance du modèle sur les données de validation, ce qui nous a permis de surveiller l'exactitude et d'autres métriques pour évaluer sa capacité à généraliser aux données non vues.

Une fois l'entraînement terminé, nous avons évalué le modèle final sur les données de test pour obtenir sa précision finale. Cette précision représente la capacité du modèle à classer correctement les exemples de test en fonction de leurs étiquettes réelles.

En ce qui concerne les performances du modèle d'ensemble, nous avons évalué sa précision sur les données d'apprentissage. Nous avons constaté que le modèle d'ensemble a obtenu une précision élevée de 94% sur les données d'apprentissage, démontrant ainsi sa capacité à bien s'adapter aux exemples d'apprentissage et à les classer avec précision. Cette précision élevée sur les données d'apprentissage indique que le modèle est capable de capturer les motifs et les relations présents dans ces données.

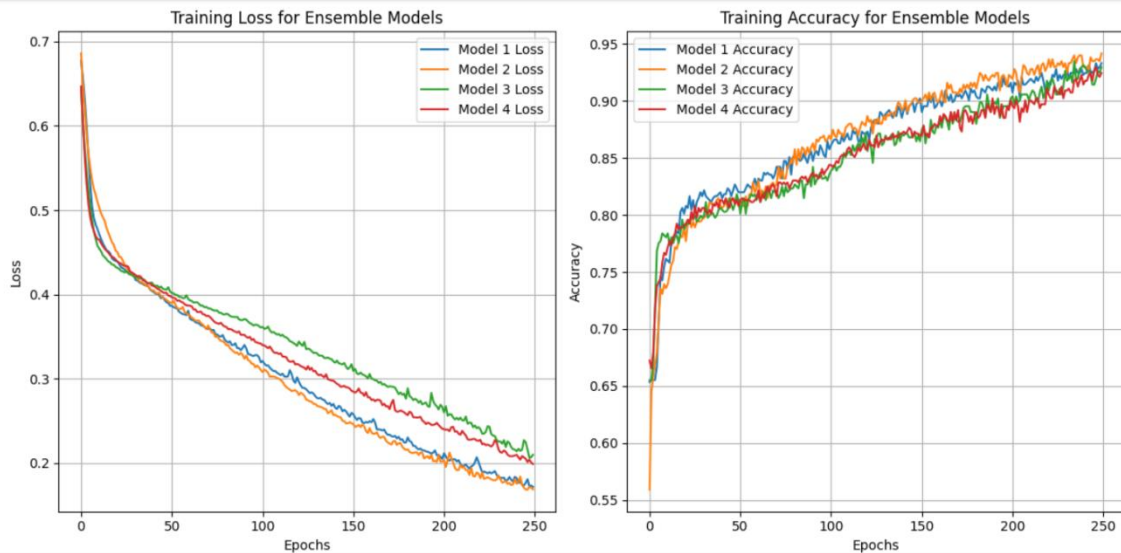


Figure 31: Visualisation des performances du modèle entraîné.

Le training loss représente la quantité d'erreur du modèle sur l'ensemble d'entraînement au fur et à mesure de l'apprentissage. Plus le training loss diminue, meilleure est la capacité du modèle à ajuster ses prédictions aux données d'entraînement.

Le training accuracy quant à lui indique la proportion d'exemples correctement classés par le modèle sur l'ensemble d'entraînement. Une augmentation du training accuracy signifie que le modèle apprend à mieux généraliser et à classer avec précision les données d'entraînement.

## 10. Prédiction du diabète

Après avoir entraîné notre modèle et l'avoir enregistré, nous pouvons le charger pour effectuer des prédictions sur le diabète en utilisant les données fournies par l'utilisateur. Cette approche facilite l'utilisation du modèle dans des situations réelles et permet aux utilisateurs d'obtenir des informations précieuses sur leur état.

```
1/1 [=====] - 0s 21ms/step
1/1 [=====] - 0s 22ms/step
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning:
  warnings.warn(
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 22ms/step
Prediction of Model 1: [[0.89455205]]
Prediction of Model 2: [[0.98175657]]
Prediction of Model 3: [[0.7947694]]
Prediction of Model 4: [[0.88018346]]
```

```
▶ if final_prediction == 1:
    print("Ensemble Prediction: The patient is diabetic")
else:
    print("Ensemble Prediction: The patient is not diabetic")
```

```
Ensemble Prediction: The patient is diabetic
```

Figure 32: cas diabétique

```
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 21ms/step
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarnin
  warnings.warn(
1/1 [=====] - 0s 24ms/step
1/1 [=====] - 0s 23ms/step
Prediction of Model 1: [[0.04639066]]
Prediction of Model 2: [[0.00728023]]
Prediction of Model 3: [[0.00022518]]
Prediction of Model 4: [[0.27767974]]
```

```
▶ if final_prediction == 1:
    print("Ensemble Prediction: The patient is diabetic")
else:
    print("Ensemble Prediction: The patient is not diabetic")
```

```
Ensemble Prediction: The patient is not diabetic
```

Figure 33: cas non diabétique

## 11. Conclusion

Notre système de classification du diabète offre à l'utilisateur la possibilité de choisir et d'entraîner différentes structures pour la prédiction. Nous avons suivi un processus méthodique comprenant la sélection des structures, le prétraitement des données, l'entraînement des modèles et l'évaluation de leurs performances. Cela nous a permis de développer un système fonctionnel pour la prédiction du diabète, Cette application ouvre des perspectives prometteuses dans le domaine de la santé et de l'aide à la décision médicale.

# Conclusion générale

Notre travail de recherche visait à développer un système de classification pour le diagnostic du diabète, en remplacement partiel du médecin. À travers ce projet, nous avons exploré les concepts fondamentaux des réseaux de neurones et de l'apprentissage automatique, ainsi que les différents types de diabète, leurs causes, leurs symptômes et leurs méthodes de traitement. Nous avons conçu et implémenté un système utilisant des techniques de prétraitement des données, d'apprentissage des réseaux de neurones et de classification pour prédire le statut diabétique des patients.

L'utilisation de techniques de classification dans le domaine médical présente un fort potentiel pour améliorer les diagnostics et réduire la dépendance à l'égard des médecins. Cependant, il est important de souligner que notre système de classification ne remplace pas totalement l'expertise médicale. Il doit être considéré comme un outil d'aide à la décision, fournissant des informations supplémentaires aux professionnels de la santé pour faciliter le processus de diagnostic.

Notre travail représente une étape importante vers l'utilisation de techniques de classification automatisée pour le diagnostic du diabète. Il ouvre la voie à de futures recherches et applications dans le domaine de l'intelligence artificielle en médecine, en contribuant à l'amélioration des soins de santé et à la prise de décision clinique.

# Références

- [1] : Dr Anne-Christine Della Valle Mis à jour le 19/08/22 12:24  
Partager sur and Valle, D.A.-C.D. (2022) *Diabète : Définition, symptômes, Quel est le Taux normal ?*, *Journal des femmes*. Available at: <https://sante.journaldesfemmes.fr/fiches-maladies/2499922-diabete-type-1-2-definition-insipide-causes-symptomes-traitement-taux-normal-journee-mondiale/> (Accessed: 24 June 2023).
- [2] : *Diabète de Type 1 : causes* (no date) *Elsan*. Available at: <https://www.elsan.care/fr/pathologie-et-traitement/maladies-endocriniennes/diabete-type-1-causes-traitements> (Accessed: 24 June 2023).
- [3] : *Diabète de Type 1 · Inserm, LA science pour la santé* (no date) *Inserm*. Available at: <https://www.inserm.fr/dossier/diabete-type-1/> (Accessed: 24 June 2023).
- [4] : *Les causes et la prévention du diabète de type 2* (no date) *VIDAL*. Available at: <https://www.vidal.fr/maladies/metabolisme-diabete/diabete-type-2/causes-prevention.html> (Accessed: 24 June 2023).
- [5] : *Traitement du Diabète: Comment Soigner Le Diabète ?* (no date) *Traitement du diabète / Comment soigner le diabète ?* Available at: <https://www.federationdesdiabetiques.org/information/traitement-diabete/page-d-introduction> (Accessed: 24 June 2023).
- [6] : *Diabète gestationnel (Diabète de Grossesse) : Définition et conséquences* (no date) *Diabète gestationnel (diabète de grossesse) : définition et conséquences | ameli.fr | Assuré*. Available at: <https://www.ameli.fr/assure/sante/themes/diabete-gestationnel/definition-facteurs-risque-consequences> (Accessed: 24 June 2023).
- [7] : *Diabète gestationnel - symptômes, causes, traitements et prévention* (no date) *VIDAL*. Available at: <https://www.vidal.fr/maladies/metabolisme-diabete/diabete-gestationnel.html> (Accessed: 24 June 2023).
- [8] : *Diabète gestationnel : Quel traitement ?* (no date) *Diabète gestationnel : quel traitement ? | ameli.fr | Assuré*. Available at: <https://www.ameli.fr/assure/sante/themes/diabete-gestationnel/traitement-suivi-femme-enceinte-bebe> (Accessed: 24 June 2023).
- [9] : *Diabète* (no date) *OPS/OMS | Organisation panaméricaine de la santé*. Available at: <https://www.paho.org/fr/sujets/diabete> (Accessed: 24 June 2023).

- [10] : *Le Diabète : Les différents types* (no date) *Cerballiance*. Available at: <https://www.cerballiance.fr/fr/blog/bilans-standards/le-diabete-les-differents-types> (Accessed: 24 June 2023).
- [11] :
- [12] : *Diabète : Types de diabète: Qu'est-ce que le diabète ?* (no date) *Diabète : Types de diabète | Qu'est-ce que le diabète ?* Available at: <https://www.federationdesdiabetiques.org/information/diabete> (Accessed: 24 June 2023).
- [13] : Friel, L.A. (2023) *Diabète pendant la grossesse - problèmes de santé de la femme, Manuels MSD pour le grand public*. Available at: <https://www.msmanuals.com/fr/accueil/probl%C3%A8mes-de-sant%C3%A9-de-la-femme/grossesse-complic%C3%A9e-par-la-maladie/diab%C3%A8te-pendant-la-grossesse> (Accessed: 24 June 2023).
- [14] : *Diabète* (no date) *OPS/OMS | Organisation panaméricaine de la santé*. Available at: <https://www.paho.org/fr/sujets/diabete> (Accessed: 24 June 2023).
- [15] : *Diabète* (no date) *OPS/OMS | Organisation panaméricaine de la santé*. Available at: <https://www.paho.org/fr/sujets/diabete> (Accessed: 24 June 2023).
- [16] : *Réseaux de Neurones/Qu'est-CE Qu'un Neurone ?* - *wikiversité* (no date) - *Wikiversité*. Available at: [https://fr.wikiversity.org/wiki/R%C3%A9seaux\\_de\\_neurones/Qu%27est-ce\\_qu%27un\\_neurone\\_%3F](https://fr.wikiversity.org/wiki/R%C3%A9seaux_de_neurones/Qu%27est-ce_qu%27un_neurone_%3F) (Accessed: 25 June 2023).
- [17] : rédac, T. (2023) *Réseaux de Neurones - Deep Learning : Biologiques ou artificiels ?, Formation Data Science | DataScientest.com*. Available at: <https://datascientest.com/deep-learning-reseau-de-neurones-biologiques-ou-artificiels> (Accessed: 25 June 2023).
- [18] : *Réseaux de Neurones/Neurone artificiel* - *wikiversité* (no date) - *Wikiversité*. Available at: [https://fr.wikiversity.org/wiki/R%C3%A9seaux\\_de\\_neurones/Qu%27est-ce\\_qu%27un\\_neurone\\_%3F#Le\\_neurone\\_artificiel](https://fr.wikiversity.org/wiki/R%C3%A9seaux_de_neurones/Qu%27est-ce_qu%27un_neurone_%3F#Le_neurone_artificiel) (Accessed: 25 June 2023).
- [19] : *Fonction d'activation* (2023) *Wikipedia*. Available at: [https://fr.wikipedia.org/wiki/Fonction\\_d%27activation](https://fr.wikipedia.org/wiki/Fonction_d%27activation) (Accessed: 25 June 2023).
- [20] : (No date) *Apprentissageparrenforcement larègledhebb - hebergement.u-psud.fr*. Available at: [http://hebergement.u-psud.fr/sergio.rodriguez/lib/exe/fetch.php?media=wiki:5\\_apprentissage\\_reinforcement.pdf](http://hebergement.u-psud.fr/sergio.rodriguez/lib/exe/fetch.php?media=wiki:5_apprentissage_reinforcement.pdf) (Accessed: 24 June 2023).
- [21] : IONOS, L. éditoriale (2020b) *Réseau de Neurones Artificiels : Quelles sont leurs capacités ?*, *IONOS Digital Guide*. Available at: <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-quun-reseau-neuronal-artificiel/> (Accessed: 25 June 2023).

- [22] : *Staff: University of Batna 2* (no date) *Staff / UNIVERSITY OF BATNA 2*. Available at: <https://web.univ-batna2.dz/staff> (Accessed: 25 June 2023).
- [23] : (No date a) *Le Perceptron Multicouche et son algorithme de ... - reussirlem1info*. Available at: <https://reussirlem1info.files.wordpress.com/2012/05/mlp.pdf> (Accessed: 24 June 2023).
- [24] : (No date a) *Algorithme de rétropropagation - CNRS*. Available at: [https://emoutot.perso.math.cnrs.fr/static/etudes/MP/TIPE/preuve\\_grad.pdf](https://emoutot.perso.math.cnrs.fr/static/etudes/MP/TIPE/preuve_grad.pdf) (Accessed: 24 June 2023).
- [25] :
- [26] : *Organigramme : Définition, rôle et utilité* (no date) *Lucidchart*. Available at: <https://www.lucidchart.com/pages/fr/organigramme> (Accessed: 25 June 2023).
- [27] : Deluzarche, C. (2021) *Définition: UML: Futura Tech, Futura*. Available at: <https://www.futura-sciences.com/tech/definitions/informatique-uml-3979/> (Accessed: 25 June 2023).
- [28] : Deluzarche, C. (2021a) *Définition: UML: Futura Tech, Futura*. Available at: <https://www.futura-sciences.com/tech/definitions/informatique-uml-3979/> (Accessed: 25 June 2023).
- [29] : Deluzarche, C. (2021a) *Définition: UML: Futura Tech, Futura*. Available at: <https://www.futura-sciences.com/tech/definitions/informatique-uml-3979/> (Accessed: 25 June 2023).
- [30] : *Différents diagrammes UML - objectif et utilisation* (no date) *Edrawsoft*. Available at: <https://www.edrawsoft.com/fr/uml-introduction.html> (Accessed: 25 June 2023).
- [31] : *Qu'est-CE qu'un diagramme de flux de données ?* (no date) *Lucidchart*. Available at: <https://www.lucidchart.com/pages/fr/diagramme-de-flux-de-donnees> (Accessed: 25 June 2023).
- [32] : *Apprendre le Langage de programmation python* (no date) *Python language de programmation apprendre cours tutoriaux django informatique développement web internet script*. Available at: <https://python.doctor/> (Accessed: 25 June 2023).
- [33] : Mobiskill (2023) *Les Bibliothèque python à utiliser pour le machine learning.*, *Mobiskill*. Available at: <https://mobiskill.fr/blog/conseils-emploi-tech/les-bibliotheque-python-a-utiliser-pour-le-machine-learning/> (Accessed: 25 June 2023).
- [34] : JVC, J. (2022) *Scikit-Learn : Guide de Démarrage Rapide en machine learning avec python, Data Transition Numérique*. Available at: <https://www.data-transitionnumerique.com/scikit-learn-python/> (Accessed: 25 June 2023).

[35] : IONOS, L. éditoriale (2020) *Keras : Une Bibliothèque open source pour la Constitution de Réseaux neuronaux*, *IONOS Digital Guide*. Available at <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/quest-ce-que-keras/> (Accessed: 25 June 2023).

:

[36] : (No date) *Google colab*. Available at: <https://research.google.com/colaboratory/faq.html?hl=fr> (Accessed: 25 June 2023).

[37] :