الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITE BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــــــــــة

**Faculté : Sciences de l'ingéniorat**
**Département : Informatique**
**Domaine : Mathématique et Informatique**
**Filière : Informatique**
**Spécialité : Analyse et Gestion des données massives**

# Mémoire
# Présenté en vue de l'obtention du Diplôme de Master

## Thème :

# Réseau de Neurones Convolutionnel pour la Reconnaissance des Globules Blancs dans le Sang

**Présenté par :** *Abbaci Alaa Eddine*

**Encadrant :** *Mohamed Ben Ali Yamina* *Grade: Pr* *UBMA*

## Jury de Soutenance :

| | | | |
|---|---|---|---|
| Azizi Nabiha | Pr | UBMA | Président |
| Mohamed Ben Ali Yamina | Pr | UBMA | Encadrant |
| Zenakhra Djamel | Mr | UBMA | Examinateur |

**Année Universitaire : 2020/2021**

# Remerciement

Avant tout, je remercie ALLAH de m’avoir aidé à accomplir ce modeste travail.

Je voudrais témoigner ma reconnaissance sincère à notre encadreur docteur Madame Mohamed Ben Ali Yamina pour ses conseils et ses encouragements tout au long de ce projet.

Je remercie les examinateurs pour avoir accepté d’examiner ce travail.

Enfin, je ne saurais terminer ces remerciements sans y associer toute personne qui, de près ou de loin, m’a apporté son aide ou sa sympathie.

# Dédicaces

## À mes très chers parents

Avec un énorme plaisir, un cœur ouvert et une immense joie, que je dédie mon travail à mes très chère, respectueux et magnifiques parents qui m'ont soutenus tout au long de ma vie, dont leurs mérites, leurs sacrifices, leurs qualités humaines m'ont permis de vivre ce jour : les mots me manquent pour exprimer toute la reconnaissance, la fierté et le profond amour que je vous porte pour les sacrifices qu'ils ont consenti pour ma réussite, qu'ils trouvent ici le témoignage de mon attachement ma reconnaissance, gratitude et respect, que Dieu leur préservent bonne santé et longue vie. Tous mes sentiments de reconnaissance pour vous.

## À mes frères et mes très chères sœurs

J'espère atteint le seuil de vos espérances. Que ce travail soit l'expression de ma profonde affection. Khairou, Mouheb, Hocine, S et A je vous remercie pour le soutien moral et l'encouragement que vous m'avez accordé. Je vous souhaite tout le bonheur que vous méritez. En leur souhaitant un brillant avenir.

## À mon plus beau cadeau de ma vie ROSA

## À toute ma famille et à tous ceux que j'aime et qui m'aiment.

## À tous mes amis et collègues

## À tous les étudiants de la promotion de 2020/2021 en Gestion et Analyse de Données Massives

# Table des matières

# Table de figure

# Résume

Les globules blancs, également connus sous le nom de leucocytes, jouent un rôle important dans le corps humain. En augmentant l'immunité en combattant contre les maladies infectieuses. La classification des globules blancs, joue un rôle important dans la détection d'une maladie chez un individu. La classification peut également aider à l'identification de maladies comme les infections, les allergies, l'anémie, la leucémie, le cancer, le syndrome d'immunodéficience acquise (SIDA), etc., qui sont dues à des anomalies du système immunitaire.

Cette classification aidera l'hématologue à distinguer le type de globules blancs présents dans le corps humain et à trouver l'origine des causes des maladies. Actuellement, il y a une grande quantité de recherches en cours dans ce domaine. Considérant un énorme potentiel dans l'importance de la classification des globules blancs, nous allons utiliser une technique d'apprentissage profond de convolution (CNN) qui peut classer les images de WBC dans ses sous-types à savoir les neutrophiles, les éosinophiles, les lymphocytes et les monocytes.

# Abstract

White Blood Cells also known as leukocytes plays an important role in the human body by increasing the immunity by fighting against infectious diseases. The classification of White Blood Cells, plays an important role in detection of a disease in an individual. The classification can also assist with the identification of diseases like infections, allergies, anemia, leukemia, cancer, Acquired Immune Deficiency Syndrome (AIDS), etc. that are caused due to anomalies in the immune system.

This classification will assist the hematologist distinguish the type of White Blood Cells present in human body and find the root cause of diseases. Currently there are a large amount of research going on in this field. Considering a huge potential in the significance of classification of WBCs, we will be using a deep learning technique Convolution Neural Networks (CNN) which can classify the images of WBCs into its subtypes namely, Neutrophil, Eosinophil, Lymphocyte and Monocyte.

## Mots clés :

CNN · Blood cell classification · Basophils · Eosinophil · Monocytes · Lymphocytes · Neutrophils · Tensorflow · Keras · Softmax function · Relu functi

# Introduction Générale

## Contexte de recherche

Avec les progrès de la technologie, étant donné le nombre de données dans le domaine médical augmente jour par jour, ce qui a entraîné une grande quantité de données structurées, non structurées et inutilisées, ainsi, un grand nombre de techniques pour explorer et structurer ces données sont apparues au cours des dernières années. Parmi ces techniques, l'apprentissage automatique et les réseaux de neurones convolutionnel (CNN).

## Problématique

Dans le domaine médical, l'un des défis à relever est d'identifier et de déterminer le nombre de globules blancs. La raison principale est l'abondance de globules rouges. Chez un adulte en bonne santé, les globules blancs représentent environ 1 % du volume sanguin total. En raison de cette faible proportion de globules blancs dans le sang, la reconnaissance des globules blancs devient un défi, ce qui complique encore plus la tâche du médecin.

Les GB est un défi, ce qui rend encore plus difficile la classification les sous-types de globules blancs.

Le changement du nombre de n'importe quel sous-type de classification signifie qu'il y a un problème dans le corps et que le corps est en train de le résoudre problème dans l'organisme et que celui-ci réagit à un type d'agent pathogène. La suite pronostic de la maladie peut être déterminée par le type de maladie et peut aider à prescrire un traitement pour cette affection spécifique à la prescription d'un traitement pour cette maladie spécifique.

La méthode traditionnelle de classification des types de globules blancs comprend l'étude des lames de sang au microscope optique et électronique.

Les cellules sanguines sont colorées avant d'être étudiées sous un microscope pour une identification parfaite, les pathologistes doivent rechercher la forme du noyau et comparer la taille par rapport aux GR. Comme il s'agit d'un processus manuel, il est sujet à des erreurs et prend du temps. Le plus gros inconvénient de la classification manuelle est l'aspect d'erreur humaine associé au balayage mécanique de la lame de verre et le commerce de l'information. Le balayage mécanique de la lame de verre et le compromis entre la résolution de l'image et le champ de vision microscopique (FOV).

Pour surmonter ces inconvénients, il est nécessaire besoin d'une méthode automatisée pour classer les globules blancs en utilisant des images de cellules sanguines colorées.

## Objectif

Ce mémoire vise à développer un modèle d'apprentissage profond pour résoudre le problème de la classification des globules blancs, qui est l'un des problèmes les plus complexes dans le diagnostic sanguin. Un cadre basé sur CNN est construit pour classer automatiquement les images de cellules sanguines en sous-types de cellules.

Des expériences sont menées sur un ensemble de données de 13k images de cellules sanguines avec leurs sous-types. Les résultats montrent que le modèle proposé fournit de meilleurs résultats en termes de paramètres d'évaluation.

## Organisation du mémoire :

Le mémoire est organisé selon le plan suivant :

Le premier chapitre nous introduisons des connaissances sur les globules sanguins et on bien focalisé sur les globules blancs.

Le deuxième chapitre consiste à présenter quelques concepts de base de l'apprentissage machine et les différents aspects ainsi que les différentes méthodes et algorithme d'apprentissage aussi on a parlé sur les réseaux de neurones.

Le troisième chapitre est consacré à la conception du notre système et on parle sur la base de données utilisée et les tests appliqués. Puis on détaille les résultats obtenus par le modèle réalisé et la présentation de notre application dans un quatrième chapitre qui concerne l'implémentation.

Enfin, on clôture avec une conclusion générale et des perspectives.

# Chapitre I : Les cellules sanguines

Dans cette section, je donnerai un aperçu général, définition et le rôle des cellules sanguines

Il existe trois grandes catégories de cellules sanguines :

## 1.1 Les Globules Rouges :

Les globules rouges sont des cellules anucléées dont le constituant essentiel est une hémoprotéine de liaison de l'oxygène : l'hémoglobine (environ 14,5 g / 100 ml). Le rôle principal de ces cellules est d'assurer le transport de l'oxygène et du gaz carbonique entre les alvéoles pulmonaires et les tissus

Aspect en microscopie optique

Il s'agit d'une cellule de 5 à 7 µ de diamètre d'aspect homogène, coloré en orangé au May Grünwald Giemsa. Son épaisseur est de 1,8 µm. Son volume moyen est de 90 fentolitres (µm3).

Le nombre de globules rouges est d'environ 5 tera/l (millions/mm3), taux un peu plus élevé chez l'homme que chez la femme (5,7 et 4,5 tera/l) [MAT 1].

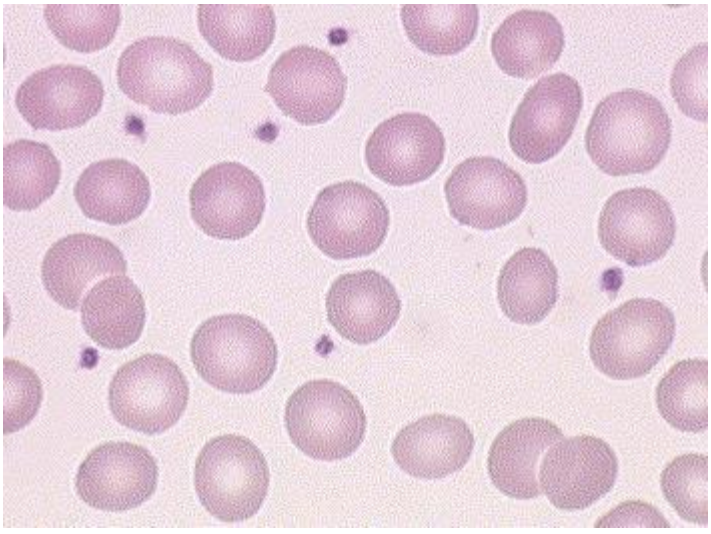

Figure 1 : Aspect en microscopie optique [WEB 1]

Ce sont des cellules biconcaves, aplaties au centre ayant un aspect de disque. Elles ne possèdent ni mitochondrie, ni ribosome, ni REG.

La membrane plasmique de l'hématie est le siège des antigènes qui déterminent les groupes sanguins (Système ABO, système rhésus et autres systèmes érythrocytaires) qui sont des récepteurs portés par les molécules de glycophorine. Ces cellules ont une durée de vie de 120 jours. Leur production est de 200x109 nouvelles cellules par jour.

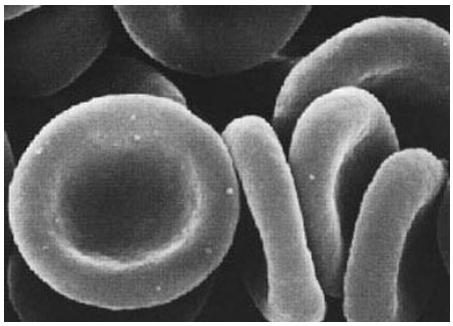

Figure 2 : Aspect en microscopie électronique à balayage [WEB 1]

## Structure moléculaire :

Leur cytosquelette est formé de deux chaînes polypeptidiques de spectrine sont reliées entre elles par de l'actine F, l'ensemble formant un réseau ancré à la membrane plasmique par des protéines associées : l'ankyrine, elle-même accrochée à une protéine transmembranaire : la protéine 3 (protéine la plus abondante : 25% de l'ensemble des protéines de membrane).

Les glycophorines - qui portent les antigènes des groupes sanguins - peuvent être liées à la protéine 4.1 (ou bande 4.1) elle-même fixée aux filaments d'actine.

Ce cytosquelette assure le maintien de la forme aplatie de la cellule et permet sa déformabilité notamment pour circuler dans les petits capillaires dont le diamètre ne dépasse pas 3 microns.

## Fonction des globules rouges :

Le transport de l'oxygène et du gaz carbonique se fait par l'intermédiaire de l'hémoglobine.
L'hémoglobine est formée de globine, protéine associée à quatre groupements hème.

Chaque hème associe un noyau porphyrique à un atome de fer ferreux.

On trouve également dans le sang circulant des réticulocytes, globules rouges jeunes possédant quelques mitochondries et des ribosomes (moins de 1% des globules rouges) [WEB 1].

## 1.2 Les Globules Blancs :

Les globules blancs constituent 1 % du sang humain chez un adulte en bonne santé. Ils sont présents dans tout l'organisme et chaque type de globules blancs à une certaine fonctionnalité dans le corps humain et sert à protéger le corps humain contre diverses infections et maladies. S'ils détectent l'un de ces éléments dans le sang, ils l'attaquent pour contrer tout dommage potentiel que ces éléments peuvent causer dans le corps. Le site structure du globule blanc, principalement, comprend un grand noyau lobé qui peut être utilisé pour distinguer un globule blanc des autres types de cellules sanguines. En dehors d'un noyau, les WBC sont constitués de cytoplasme et de paroi cellulaire [MAT 1].

Il existe cinq grandes catégories de GB dans le corps humain. Cependant, en raison des contraintes liées à l'ensemble des données, nous avons classé les données en quatre catégories : Basophiles (0,4 % environ)

(0,4 % environ), Éosinophiles (2,3 % environ), Monocytes (5,3 % environ),

Lymphocytes (30% environ) et Neutrophiles (62% environ) [LAF 2].

### 1.2.1 Les monocytes

Le nombre de monocytes présents dans un corps humain sain varie entre 6-9% du nombre total de globules blancs. La durée de vie d'un monocyte varie de quelques heures à une journée.

Les monocytes sont également responsables de la présentation des agents pathogènes aux cellules T afin qu'ils puissent être facilement tués. Afin de les tuer facilement et de réduire le temps de réponse des anticorps chez l'homme. Les monocytes peuvent être reconnus dans l'ensemble de données d'images BCCD en utilisant certaines caractéristiques. Le noyau est une cellule arrondie en forme de rein avec une couleur rouge peau et un peu de couleur violette sans aucun lobe [LAF 2].

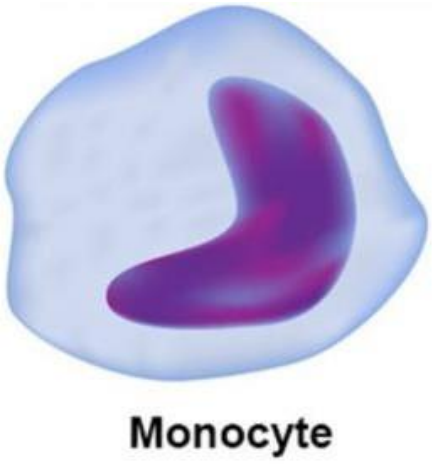


Figure 3 : Les Globules Blancs monocytes [WEB 2]

En microscopie optique, elles apparaissent arrondies, ayant un diamètre de 15 à 20µm. Le cytoplasme est gris bleuté (ciel d'orage) au MGG et a un aspect un peu granuleux. Il existe en périphérie des voiles cytoplasmiques, visibles en microscopie optique. Le noyau est central, en fer à cheval ou en E.

En microscopie électronique, la chromatine est fine, les organites bien développés et situés dans l'encoche du noyau. Il existe de nombreuses granulations azurophiles, de petite taille correspondant à des lysosomes. La membrane plasmique est irrégulière avec de nombreuse expansions et microvillosités. Les monocytes représentent 2 à 10 % de l'ensemble des globules blancs [HUT 4].

### 1.2.2 Les lymphocytes :

Le nombre de lymphocytes présents dans un corps humain sain représente environ 25 à 30 % du nombre total des globules blancs. Ces cellules sont plus présentes dans le système lymphatique que dans le sang. Les lymphocytes se composent de deux types de cellules, à savoir les lymphocytes B et les lymphocytes T. Ces cellules sont responsables de la destruction directe des cellules infectées par des virus dans le corps humain et de l'élimination des cellules cancéreuses. Les lymphocytes peuvent être facilement identifiés dans l'ensemble de données BCCD en regardant le noyau, qui est clairement rond, de couleur violette, et excentrique [ALB 3].


Figure 4 : Les Globules Blancs lymphocytes [WEB 2]

En microscopie optique, ce sont des cellules de petites tailles, environ 7 µm de diamètre avec un noyau occupant la quasi-totalité de la cellule. Leur forme est régulière et arrondie. Il existe une petite frange cytoplasmique périphérique d'aspect mauve au MGG. Le noyau est sphérique, dense.

En microscopie électronique à transmission, la chromatine est dense, il n'existe pas de nucléole. Le cytoplasme est pauvre en organites (quelques ribosomes et un ergoplasme réduit).

Tous les lymphocytes sont semblables sur le plan morphologiques mais il existe plusieurs groupes de lymphocytes mis en évidence par des marqueurs antigéniques de membrane : les lymphocytes B et les lymphocytes T, dont la maturation se fait au niveau du thymus. On décrit également un troisième groupe apparenté aux lymphocytes T : Les cellules NK ou Natural Killer. La population lymphocytaire sanguine comprend 8 à 12 % de lymphocytes B, 70 à 80 % de lymphocytes T et 5 à 15 % de cellules NK [HUT 4].

## Fonction des lymphocytes :

Ces cellules sont responsables des réponses spécifiques immunitaires.

Les lymphocytes B effectuent leur différenciation dans la moelle osseuse (organe lymphoïde primaire). Ils sont responsables de l'immunité humorale et peuvent fabriquer les anticorps ou immunoglobines après présentation de l'antigène par une cellule présentatrice d'antigène (macrophages, cellules folliculaires, cellules dentritiques).

Les lymphocytes B possèdent des immunoglobulines de membrane qui constituent le marqueur phénotypique de ces cellules. La fabrication des anticorps se fait au niveau des organes lymphoïdes secondaires où les lymphocytes se transforment en plasmocytes.

Les lymphocytes T acquièrent leur différenciation au niveau du thymus (organe lymphoïde primaire). Les lymphocytes T matures expriment le récepteur de membrane CD3. Parmi ces lymphocytes matures, on distingue plusieurs groupes caractérisés par la présence d'autres récepteurs de membrane :

Les CD4 ou T helpers qui reconnaissent l'antigène en association avec les molécules HLA de classe II (représentent environ la moitié des T)

Les CD8 ou T suppresseurs ou cytotoxiques qui reconnaissent l'antigène en association avec les molécules HLA de type I (de 20 à 30 % des T)

Les lymphocytes T participent à la réponse immunitaire humorale en stimulant ou en freinant la production d'anticorps par les lymphocytes B mais sont également impliqués dans l'immunité cellulaire et secrètent des cytokines ou lymphokines [HUT 4].

### 1.2.3 Les polynucléaires

Ce groupe de cellules possède des caractéristiques communes. Elles contiennent un noyau plurilobé. Les lobes sont reliés les uns aux autres par des ponts fins de chromatine. Dans le cytoplasme, il existe deux types de granulations : des granulations non spécifiques primaires, riches en hydrolases et en peroxydases, communes à l'ensemble des polynucléaires et des granulations secondaires spécifiques à chaque groupe ayant des propriétés tinctoriales différentes. Dans la cellule mature, les granulations non spécifiques diminuent.

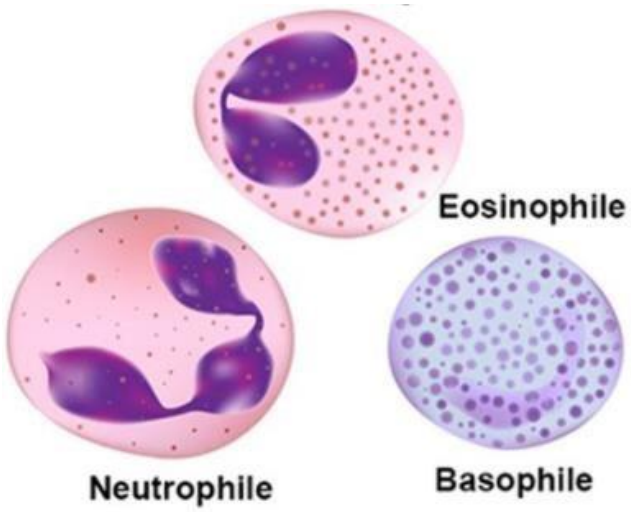


Figure 5 : Les Globules Blancs polynucléaires [WEB 2]

### 1.2.4 Neutrophiles

Ce sont les polynucléaires les plus nombreux 40 à 75 % de l'ensemble des globules blancs. Leur durée de vie est de l'ordre de 24 heures. Leurs granulations spécifiques sont neutrophiles.


Figure 6 : Les Globules Blancs Neutrophiles [WEB 2]

En microscopie optique, ce sont des cellules d'environ 12 µm de diamètre, le noyau est généralement trilobé mais le nombre de lobes varie de 2 à 5 lobes et est un indice de maturation de la cellule. La formule d'Arneth est la répartition des polynucléaires neutrophiles en fonction du nombre de lobes. Le cytoplasme apparaît clair, non colorable au MGG. En effet, les granulations azurophiles ne sont colorables que par la mise en évidence spécifique de la myélopéroxydase.

En microscopie électronique, le noyau à une chromatine dense, le cytoplasme contient deux types de granulations : les granulations non spécifiques ou primaires, azurophiles qui renferment une myélopéroxydase, des hydrolases acides et du lysosyme et des granulations spécifiques secondaires, neutrophiles, de petite taille (0,3 à 0,8 µm) éparses dans le cytoplasme. Ces granulations sont dépourvues d'enzymes lysosomiales et de péroxydases mais contiennent du lysosyme et de la collagénase. Il existe en périphérie de la cellule une bande riche en filaments d'actine [HUT 4].

#### La fonction de ces neutrophiles :

La défense non spécifique de l'organisme et notamment la lutte anti-bactérienne. Cette fonction est permise par les propriétés des neutrophiles :

Les phénomènes de diapédèse leur permettent de quitter le milieu sanguin en passant entre les cellules endothéliales. Ces phénomènes sont assurés grâce à des cytokines sécrétées sur le lieu de l'infection, notamment l'interleukine 8 (IL-8) qui active les polynucléaires neutrophiles et par les molécules d'adhésion qui apparaissent à la surface du polynucléaire et se lient à leur ligand spécifique situé sur les cellules endothéliales.

Le chimiotactisme les attire sur les lieux de l'inflammation : l'IL-8 secrété par les monocytes ainsi que certaines fractions du complément participent à ce chimiotactisme notamment en provoquant une réorientation du cytosquelette et des organites au sein de la cellule.

Les propriétés de la phagocytose lui permettent de détruire les agents étrangers notamment les bactéries. La phagocytose peut être facilitée par un phénomène d'opsonisation caractérisé par une liaison spécifique des lipopolysaccharides de certaines parois bactériennes ou avec des immunoglobulines qui se lient à leur récepteur situé sur la membrane du polynucléaire

L'action de la myélopéroxydase des granulations azurophiles lui confère une activité bactéricide, qui lui permet de détruire les bactéries phagocytées [ALB 3].

### 1.2.5 Eosinophiles

Ces cellules représentent 1 à 3 % des globules blancs. Elles ont une demi-vie dans le sang circulant de 4 à 5 heures puis passent dans les tissus (peau, poumon, tractus digestif) où elles restent 8 à 10 jours. La proportion d'éosinophiles dans les tissus est 100 fois plus importante que celle du sang.


Figure 7 : Les Globules Blancs Éosinophiles [WEB 2]

En microscopie optique, leur diamètre est de 10 à 14 µm, le noyau est généralement bi-lobé, le cytoplasme apparaît en orangé au MGG, d'aspect granuleux à cause de la présence des granulations spécifiques. Ces granulations sont volumineuses et acidophiles.

En microscopie électronique, les granulations spécifiques, éosinophiles sont volumineuses, de 0,5 à 1,5 µm de diamètre et contiennent une matrice granulaire au sein de laquelle se trouve une formation cristalloïde allongée.

Ces granulations contiennent une péroxydase (différente de la myélopéroxydase des neutrophiles) et des hydrolases acides.

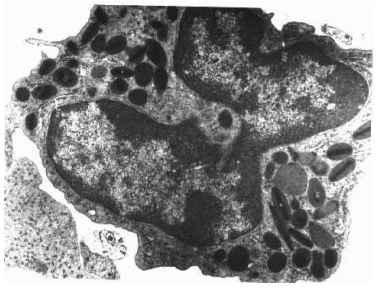

Figure 8 : Eosinophilies : microscopie électronique [WEB 1]

### Fonction des éosinophiles :

Ces cellules participent en synergie avec d'autres cellules, aux réactions d'hypersensibilité immédiate et retardée. Elles ont à des degrés moindres que les neutrophiles des propriétés de bactéricidie et de phagocytose. Elles interviennent essentiellement dans la destruction des parasites par l'intermédiaire de protéines de haut poids moléculaires (Eosinophil Cationic Protein - ECP et la Major Basic Protein - MBP) contenues dans les cristalloïdes des granulations. La membrane plasmique possède un récepteur pour les immunoglobulines de type lgE et pour l'histamine.

## 1.2.6 Basophiles

Ces cellules sont les moins nombreuses des polynucléaires, (0 à 1 % de l'ensemble des globules blancs). La durée de vie de ces cellules est de 3 à 4 jours.


Figure 9 : Les Globules Blancs Éosinophiles [WEB 2]

En microscopie optique, ces cellules ont un diamètre de 10 à 14 µm. Leur noyau est irrégulier. Il peut prendre un aspect de trèfle, qui est généralement masqué par les nombreuses granulations métachromatiques (prennent une coloration rouge avec les colorants acides comme le bleu de toluidine ou le bleu alcian) qui apparaissent pourpres au MGG.

En microscopie électronique, les granulations apparaissent homogènes, formées de petits grains denses entourés d'une membrane. Ces granulations basophiles contiennent de l'histamine.

### Rôle des basophiles :

C'est la cellule des manifestations allergiques de type immédiat.

La membrane plasmique des basophiles possède des récepteurs pour le fragment Fc des immunoglobulines de type IgE. De ca fait, les IgE fabriquées de façon spécifique contre un allergène sont fixées à la membrane des basophiles. Quand il y a à nouveau contact avec l'allergène, le pontage des IgE par l'allergène provoque la dégranulation des basophiles, responsable des manifestations allergiques. [WEB 1]

## 1.3 Les Plaquettes :

Leur durée de vie est de 8 à 12 jours.

En microscopie optique, les plaquettes sanguines ou thrombocytes sont des fragments cellulaires anuclées de 2 à 5 µm de diamètre. On distingue deux zones : le centre de la cellule (chronomère) contenant des granulations et la périphérie (hyalomère) plus homogène.

Figure 10 : Les plaquettes [WEB 1]

En microscopie électronique, elles apparaissent riches en granulations azurophiles denses aux électrns contenant de l'ADP, du glycogène. Leur cytosquelette est très développé avec notamment un faisceau marginal de microtubules circulaires et des microfilaments d'actine(thrombas thénine). Il existe également un réseau canalaire constitué par invagination de lamembrane plasmique augmentant ainsi la surface de la membrane.

### Fonction des plaquettes :

Elles jouent un rôle fondamental dans les phénomènes initiaux de coagulation. Le feuillet externe de la membrane plasmique contient un épais glycolemme riche en molécule d'adhésion qui sont exprimées quand la plaquette est activée. Elles adhèrent ainsi au collagène quand il y a effraction de l'endothélium. L'actine et le système de microtubules provoquent une adhésion des plaquettes entre elles. Le faisceau de microtubules en se dépolyomérisant en filaments participe à l'agrégation des plaquettes. La couronne d'actine périphérique permet également, en se contractant, l'extrusion du contenu des granulationspar le réseau canalaire, et provoque la synthèse de thromboxane à partir de l'acide arachidonique contenu dans les phospholipides des membranes plasmiques. Le thromboxane libéré a une action vasoconstrictrice. Les substances excrétées provoquent l'adhérence des autres plaquettes [WEB 1].

## 1.4 Conclusion

Dans ce chapitre, nous avons introduit quelques notions fondamentales sur les types des cellules sanguins, on a défini les caractéristiques de chaque cellule et le rôle de chaque une dans le corps humain.

# Chapitre II : Concepts de base de l'apprentissage machine et Les réseaux de neurones

Dans ce chapitre, je donnerai un aperçu général de l'apprentissage machine et la classification d'image et ces différentes méthodes ainsi que les champs de traitement et concepts appliqués dans ce mémoire.

## 2.1 Apprentissage automatique

L'apprentissage automatique, également appelé apprentissage machine ou apprentissage artificiel et en anglais machine learning, est une forme d'intelligence artificielle (IA) qui permet à un système d'apprendre à partir des données et non à l'aide d'une programmation explicite. Cependant, l'apprentissage automatique n'est pas un processus simple. Au fur et à mesure que les algorithmes ingèrent les données de formation, il devient possible de créer des modèles plus précis basés sur ces données. Un modèle de machine learning est le résultat généré lorsque vous entraînez votre algorithme d'apprentissage automatique avec des données. Après la formation, lorsque vous fournissez des données en entrée à un modèle, vous recevez un résultat en sortie. Par exemple, un algorithme prédictif crée un modèle prédictif. Ensuite, lorsque vous fournissez des données au modèle prédictif, vous recevez une prévision qui est déterminée par les données qui ont servi à former le modèle.

Des techniques d'apprentissage automatique sont nécessaires pour améliorer l'exactitude des modèles prédictifs. Selon la nature du problème métier traité, il existe différentes approches qui varient selon le type et le volume des données. Dans cette section, nous discutons des catégories de l'apprentissage automatique [WEB 25].

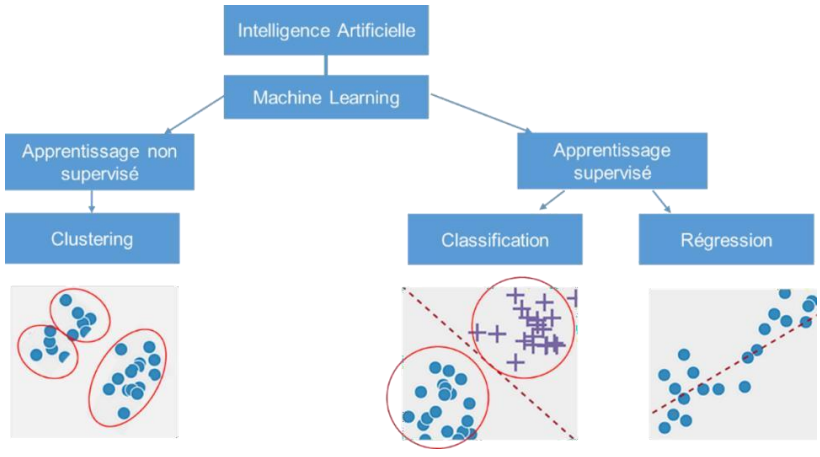


Figure 11 : Apprentissage-Supervisé-Non Supervisé [WEB 3]

### 2.1.1 Apprentissage supervisé

L'apprentissage supervisé, dans le contexte de l'intelligence artificielle (IA) et de l'apprentissage automatique, est un système qui fournit à la fois les données en entrée et les données attendues en sortie. Les données en entrée et en sortie sont étiquetées en vue de leur classification, afin d'établir une base d'apprentissage pour le traitement ultérieur des données.

Les systèmes d'apprentissage automatique supervisé alimentent les algorithmes d'apprentissage avec des quantités connues qui étayeront les futures décisions. Supervisé ou non supervisé, l'apprentissage automatique est utilisé notamment pour les chatbots, les véhicules autonomes, les programmes de reconnaissance faciale, les systèmes experts et les robots. Les systèmes d'apprentissage supervisé sont associés pour la plupart à une intelligence artificielle basée sur la récupération, mais ils peuvent aussi reposer sur un modèle d'apprentissage génératif.

Les données utilisées pour l'apprentissage supervisé sont une série d'exemples comprenant des paires composées de sujets en entrée et de sorties attendues (appelées également signaux de supervision). Prenons l'exemple d'un système d'apprentissage supervisé pour le traitement d'images dans lequel on introduit des photos de véhicules appartenant aux catégories voitures et camions. Après un temps d'observation suffisant, le système doit être capable de faire la distinction entre plusieurs images non étiquetées et de les catégoriser ; une fois cet objectif atteint, l'apprentissage peut être considéré comme terminé [WEB 26].

### 2.1.1 Apprentissage non supervisé

L'apprentissage non supervisé est utilisé lorsque le problème nécessite une quantité massive de données non étiquetées. Par exemple, les applications de réseaux sociaux, telles que Twitter, Instagram et Snapchat, exploitent toutes de très grandes quantités de données non étiquetées. Pour comprendre le sens de ces données, il est nécessaire d'utiliser des algorithmes qui classifient les données en fonction des tendances ou des clusters qu'ils décèlent. L'apprentissage non supervisé mène un processus itératif, analysant les données sans intervention humaine. Il est utilisé avec la technologie de détection de spam envoyé par e-mail. Les e-mails normaux et les spams comportent un nombre de variables beaucoup trop élevé pour qu'un analyste puisse étiqueter les e-mails indésirables envoyés en masse. En revanche, les discriminants d'apprentissage automatique, basés sur la mise en cluster et l'association, sont appliqués pour identifier les courriers électroniques non désirés [WEB 25].

### 2.1.3 Apprentissage par renforcement

L'apprentissage par renforcement (RL pour Reinforcement Learning) fait référence à une classe de problèmes d'apprentissage automatique, dont le but est d'apprendre, à partir d'expériences successives, ce qu'il convient de faire de façon à trouver la meilleure solution.

Dans un tel problème, on dit qu'un « agent » (l'algorithme, au sens du code et des variables qu'il utilise) interagit avec « l'environnement » pour trouver la solution optimale. L'apprentissage par renforcement diffère fondamentalement des problèmes supervisés et non supervisés par ce côté interactif et itératif : l'agent essaie plusieurs solutions (on parle « d'exploration »), observe la réaction de l'environnement et adapte son comportement (les variables) pour trouver la meilleure stratégie (il « exploite » le résultat de ses explorations). Un des concepts clés de ce type de problèmes est l'équilibre entre ces phases d'exploration et d'exploitation. Cette méthode est particulièrement adaptée aux problèmes nécessitant un compromis entre la quête de récompenses à court terme et celle de récompenses à long terme. Parmi les exemples de problèmes traités de cette façon, on peut évoquer : apprendre à un robot à marcher en terrain difficile, à conduire (cas de la voiture autonome) ou à accomplir une tâche spécifique (comme jouer au jeu de go), piloter un agent à travers un labyrinthe, etc. Les principales familles de problèmes d''apprentissage par renforcement sont les algorithmes de bandits, les problèmes de décisions (partiellement) markovien et les arbres de jeu [WEB 27].

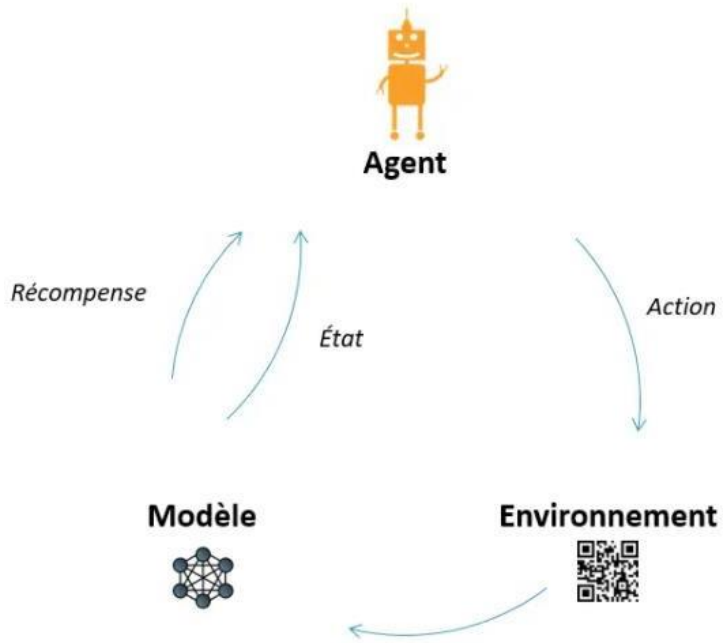


Figure 12 : Schéma d'apprentissage par renforcement [WEB 4]

### 2.1.4 Apprentissage profond

L'apprentissage profond est une méthode spécifique d'apprentissage automatique qui intègre des réseaux neuronaux en couches successives afin d'apprendre des données de manière itérative. L'apprentissage profond est particulièrement utile lorsque vous tentez de détecter des tendances à partir de données non structurées. Les réseaux neuronaux complexes d'apprentissage en profondeur sont conçus pour émuler le fonctionnement du cerveau humain, de sorte que les ordinateurs peuvent être entraînés pour faire face à des abstractions et des problèmes mal définis. La plupart des enfants de cinq ans distiguent facilement le visage de leur instituteur de celui de l'agent chargé de leur faire traverser le passage piéton. En revanche, l'ordinateur doit fournir un travail considérable pour identifier chaque visage. Les réseaux neuronaux et l'apprentissage en profondeur sont souvent utilisés dans les applications de reconnaissance d'image, de communication orale et de vision numérique [WEB 25].

## 2.2 Prétraitement de données

Le prétraitement des données est une technique d'exploration de données qui consiste à transformer des données brutes dans un format compréhensible par la machine. Les données réelles sont souvent incomplètes, incohérentes et sont susceptibles de contenir de nombreuses erreurs. Le prétraitement des données est une méthode destinée à résoudre ces problèmes. Le prétraitement des données prépare les données brutes à un traitement ultérieur.

Les données passent par une série d'étapes pendant le prétraitement :

- **Nettoyage des données :** les données sont nettoyées par des processus tels que le remplissage des valeurs manquantes, le lissage des données bruyantes ou la résolution des incohérences dans les données.
- **Intégration des données :** les données avec différentes représentations sont rassemblées et les conflits au sein des données sont résolus.
- **Transformation des données :** les données sont normalisées, agrégées et généralisées.
- **Réduction des données :** cette étape vise à présenter une représentation réduite des données dans un entrepôt de données.
- **Discrétisation des données :** implique la réduction d'un certain nombre de valeurs d'un attribut continu en divisant la plage d'intervalles d'attribut.

## 2.3 Classification

La classification consiste à attribuer de nouveaux échantillons de données à des catégories prédéfinies, sur la base d'échantillons d'apprentissage où la catégorie de chaque échantillon est déjà connue, la classification est donc une approche d'apprentissage supervisé. Pendant la phase d'apprentissage, le classifier apprend la relation entre les données d'entrée et la catégorie de sortie (étiquette) et est ensuite en mesure de prédire la catégorie de nouveaux échantillons de données qui n'ont pas de catégorie connue. Il y a différents types de problèmes de classification.

### 2.3.1 Classification Binaire (Binary Classification)

La classification binaire (ou la classification binomiale) est une transformation de données qui vise à répartir les membres d'un ensemble dans deux groupes disjoints selon que l'élément possède ou non une propriété / fonctionnalité donnée. Par exemple, un modèle d'apprentissage automatique qui classe les courriers électroniques en tant que «indésirable» ou «légitime») est une classification binaire. [WEB 5]

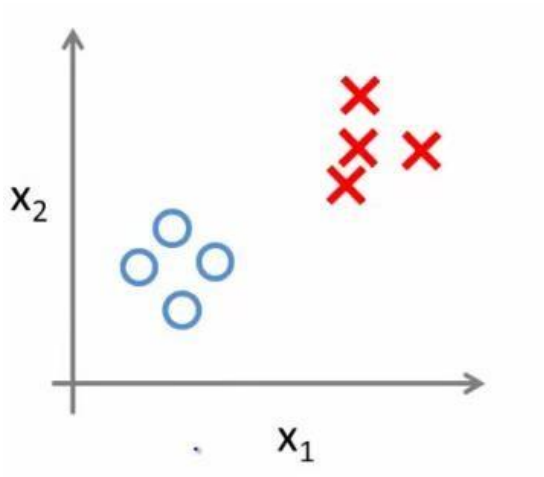


Figure 13 : Classification Binaire [WEB 6]

### 2.3.2 Classification multi-classes

Dans ce type de classification, les points sont classés en trois classes ou plus (Figure 14). Pour ce type de problèmes de classification, il existe de nombreux algorithmes comme k-NN, les arbres de décision et les réseaux de neurones pour n'en nommer que quelques-uns qui permettent naturellement la classification de plus de deux classes. La classification multi-classes peut être transformée en un problème de classification binaire en utilisant les deux approches suivantes. [WEB 6]

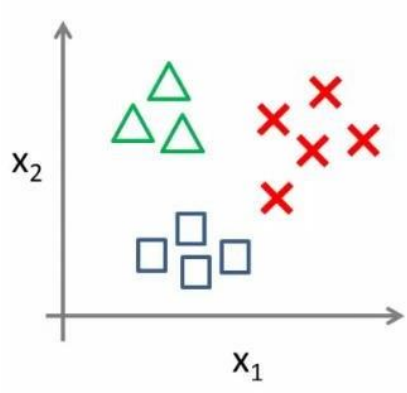


Figure 14 : Classification multi-classes [WEB 6]

### a) One Vs One

Dans cette approche, un classifieur binaire est formé pour chaque paire de classes. Pour un problème de classification avec N classes N * (N-1) / 2 classifieurs binaires sont entraînés. Au moment de la prédiction, un schéma de vote est appliqué, chaque classifier vote pour une classe et la classe avec le plus de votes serait la prédite. [WEB 6]

### b) One Vs Rest

Dans One Vs Rest (également appelé One Vs All), un classifieur binaire est formé pour chacune des classes et dans chaque processus de formation du classifieur. Les étiquettes des données appartenant à la classe cible représentent la classe positive et tous les autres points qui appartiennent au reste de la classe représentent la classe négative comme le montre la figure 6. Lors de la prédiction de la classe d'un nouvel échantillon, tous les classifiers formés sont invités à prédire la classe de l'échantillon et le classifier avec la probabilité la plus élevée détermine la classe. [WEB 6]

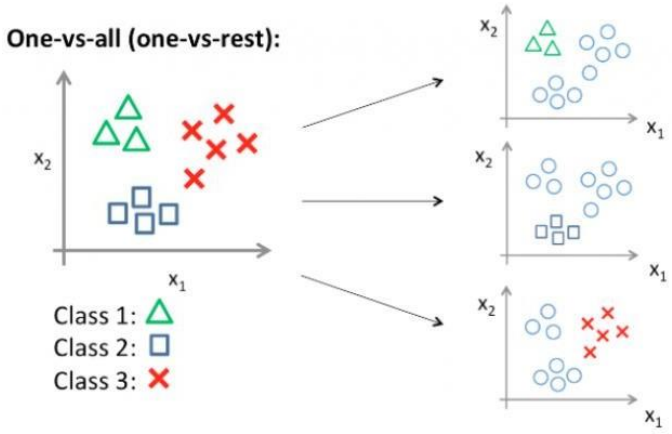


Figure 15 : One Vs Rest [WEB 6]

### 2.3.3 Classification multi-label

La classification multi-label est une extension de la classification traditionnelle dans laquelle les classes ne sont pas mutuellement exclusives, chaque individu pouvant appartenir à plusieurs classes simultanément. Ce type de classification est requis par un grand nombre d'applications actuelles telles que la classification d'images. [WEB 7]

| x | y |
|---|---|
| $X_1$ | $[t_2, t_5]$ |
| $X_2$ | $[t_1, t_2, t_3, t_4]$ |
| $X_3$ | $[t_3]$ |
| $X_3$ | $[t_2, t_4]$ |
| $X_3$ | $[t_1, t_3, t_4]$ |

Figure 16 : Classification multi-label [WEB 8]

## 2.4 Historique sur les réseaux de neurones

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

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

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

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

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

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

Aujourd'hui, les réseaux neuronaux sont utilisés dans de nombreux domaines (entre autres, vie artificielle et intelligence artificielle) à cause de leur propriété en particulier, leur capacité d'apprentissage, et qu'ils soient des systèmes dynamiques. [WEB 9]

## 2.5 Les Réseaux de Neurones Artificiel

Les réseaux de neurones, communément appelés des réseaux de neurones artificiels sont des imitations simples des fonctions d'un neurone dans le cerveau humain pour résoudre des problématiques d'apprentissage de la machine (Machine Learning).

Le neurone est une unité qui est exprimée généralement par une fonction sigmoïde.

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

Pourquoi recourir à des réseaux de neurones ? La réponse est plutôt simple dans le sens où les réseaux de neurones s'avèrent plus performants que les techniques de régressions pour des tâches de Machine Learning. [WEB 10]

Les domaines d'application des réseaux neuronaux sont souvent caractérisés par une relation entrée-sortie de la donnée d'information :

- La reconnaissance d'image
- Les classifications de textes ou d'images
- Identification d'objets
- Prédiction de données
- Filtrage d'un set de données

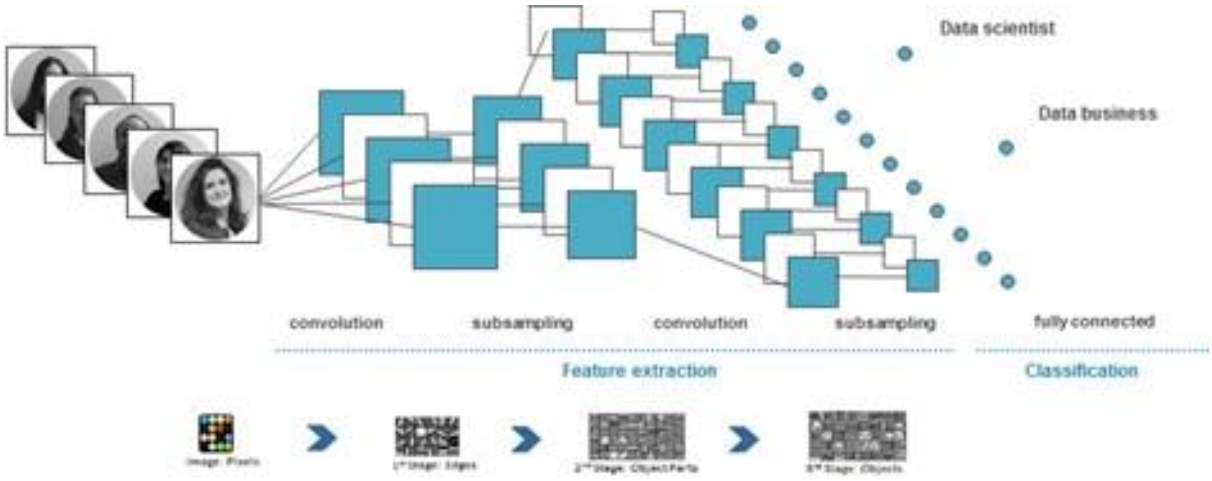


Figure 17 : Réseaux neuronaux classiques pour reconnaissance d'image. [WEB 10]

Avant de s'attaquer au fonctionnement précis des réseaux de neurones, on s'est dit qu'il serait intéressant de faire un parallèle avec les neurones biologiques.

## 2.6 Les Neurones Biologiques : Structure et rôle

Le système nerveux est composé de milliards de cellules : c'est un réseau de neurones biologiques. En effet, les neurones ne sont pas indépendants les uns des autres, ils établissent entre eux des liaisons et forment des réseaux plus ou moins complexes. [WEB 12] Le neurone biologique est composé de trois parties principales :

- 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.

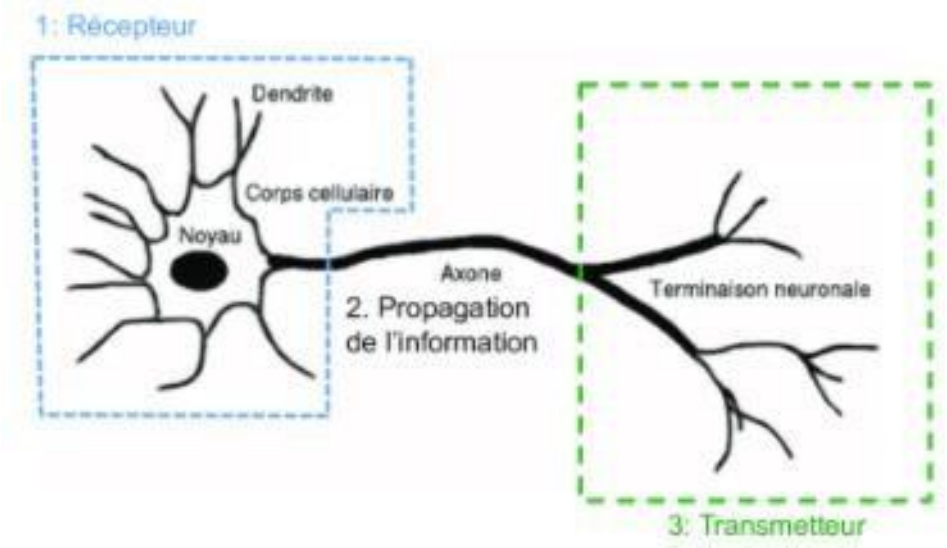


Figure 18 : Schéma d'un neurone [WEB 11]

Quant aux synapses, elles font effet de liaison et de pondération entre neurones et permettent donc aux neurones de communiquer entre eux. [WEB 12]

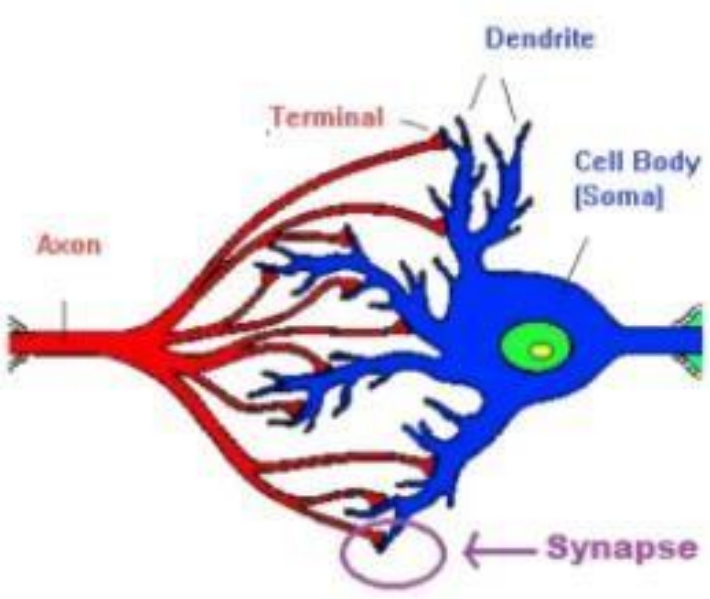


Figure 19 : The synapse between two neurons [WEB 13]

Les neurones biologiques disposent d'un centre de contrôle (appelé cellule somatique) faisant la somme des informations recueillies par les dendrites [WEB 12]. Ensuite, le centre de contrôle retourne un potentiel d'action en suivant les règles suivantes :

- Si la somme en entrée ne dépasse pas le seuil d'excitation : pas de message nerveux via l'axone.

- Si la somme en entrée dépasse le seuil d'excitation : un message nerveux est émis via l'axone (c'est l'idée, mais en réalité un peu plus compliqué).

## 2.7 Le lien entre les neurones biologiques et neurones artificiels

Procédons à une comparaison simple des étapes principales de l'algorithme du perceptron avec les éléments constitutifs des neurones biologiques [WEB 12]. Ce choix d'algorithme se justifie car il serapproche au mieux du fonctionnement des neurones biologiques :

- Les synapses/dendrites : pondération de chaque élément en entrée wi ● xi
- Corps cellulaires : application d'une fonction d'activation f à la somme des entrées pondérées
- Axone : sortie de notre modèle

$$Output = f(\sum_{i=1}^{n} w_i \cdot x_i) = f(< w|x >)$$

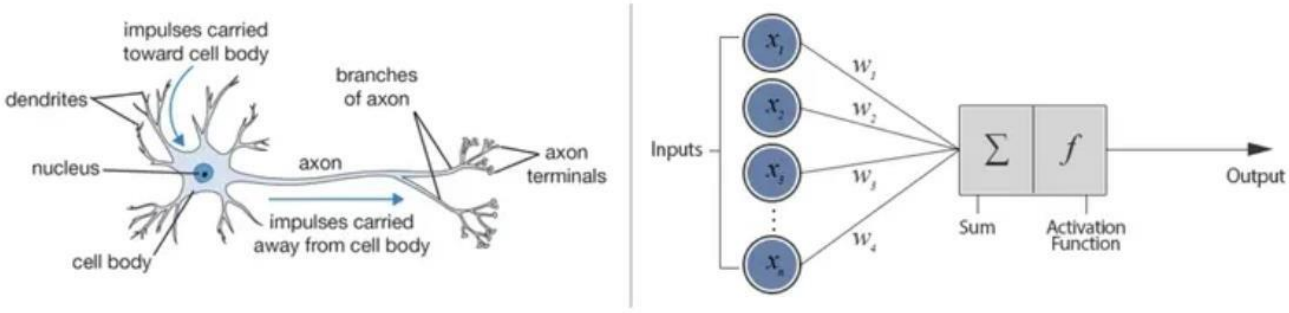


Figure 20 : Neurone biologique contre réseau neuronal artificiel [WEB 14]

Le vocabulaire spécifique à cet algorithme est le suivant :

- Le vecteur w porte le nom de vecteur de poids (qui s'ajuste lors de l'entraînement).
- Le vecteur x porte le nom de vecteur d'entrée.

f porte le nom de fonction d'activation. [WEB 12]

## 2.8 L'architecture D'un Réseau Neuronal

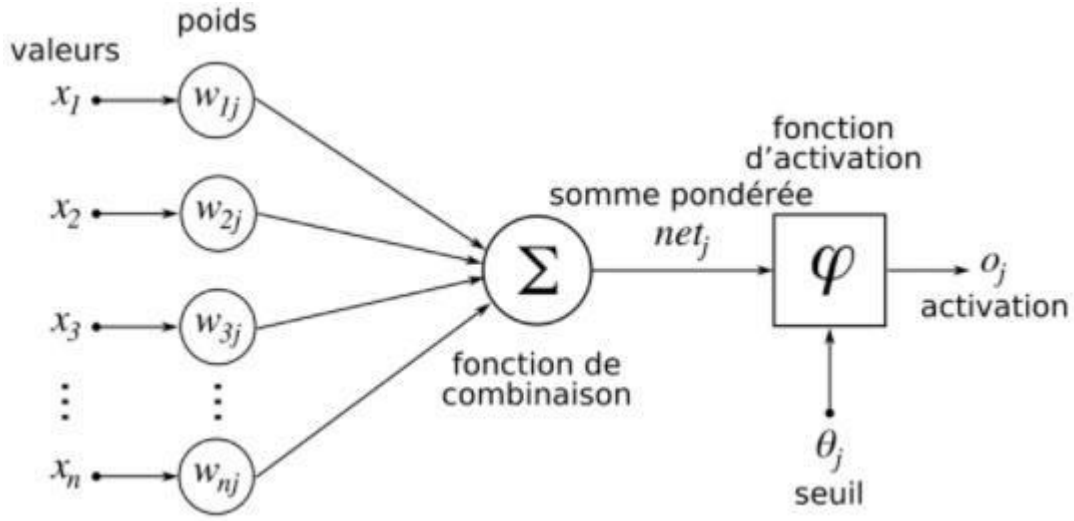


Figure 21 : L'architecture D'un Réseau Neuronal [WEB 10]

Un réseau de neurones peut prendre des formes différentes selon l'objet de la donnée qu'il traite et selon sa complexité et la méthode de traitement de la donnée.

Les architectures ont leurs forces et faiblesses et peuvent être combinées pour optimiser les résultats. Le choix de l'architecture s'avère ainsi crucial et il est déterminé principalement par l'objectif. [WEB 10]

Les architectures de réseaux neuronaux peuvent être divisées en 4 grandes familles :

- Réseaux de neurone convolutif (CNN)
- Réseaux de neurone récurrent (RNN)
- Réseaux de neurones à résonance
- Réseaux de neurones auto-organisés

## 2.9 Les types de réseaux neuronaux

### 2.9.1 Réseaux de neurones convolutif (CNN)

Les CNN désignent une sous-catégorie de réseaux de neurones et sont à ce jour un des modèles de classification d'images réputés être les plus performants.

Leur mode de fonctionnement est à première vue simple : l'utilisateur fournit en entrée une image sous la forme d'une matrice de pixels.
Celle-ci dispose de 3 dimensions :

- Deux dimensions pour une image en niveaux de gris.
- Une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert, Bleu).

Contrairement à un modèle MLP (Multi Layers Perceptron) classique qui ne contient qu'une partie classification [WEB 10], l'architecture du Convolutional Neural Network dispose en amont d'une partie convolutive et comporte par conséquent deux parties bien distinctes :

#### 1) Une partie convolutive :

Son objectif final est d'extraire des caractéristiques propres à chaque image en les compressant de façon à réduire leur taille initiale. En résumé, l'image fournie en entrée passe à travers une succession de filtres, créant par la même occasion de nouvelles images appelées cartent de convolutions. Enfin, les cartes de convolutions obtenues sont concaténées dans un vecteur de caractéristiques appelé code CNN.

La convolution est un outil mathématique simple qui est très largement utilisé pour le traitement d'image, ce qui explique que les réseaux de neurones à convolution soient particulièrement bien adaptés à la reconnaissance d'image.
La convolution agit comme un filtrage. On définit une taille de fenêtre qui va se balader à travers toute l'image (rappelez-vous qu'une image peut être vue comme étant un tableau). Au tout début de la convolution, la fenêtre sera positionnée tout en haut à gauche de l'image puis elle va se décaler d'un certain nombre de cases (c'est ce que l'on appelle le pas) vers la droite et lorsqu'elle arrivera au bout de l'image, elle se décalera d'un pas vers le bas ainsi de-suite jusqu'à ce que le filtre est parcourue la totalité de l'image :

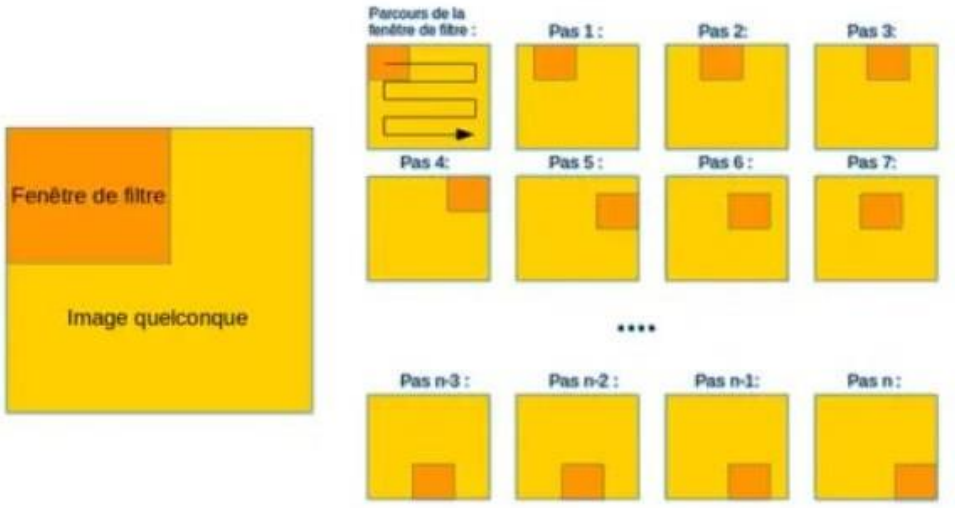


Figure 22 : Schéma du parcours de la fenêtre de filtre sur l'image [WEB 15]

## Exemple d'un filtre de convolution classique :

Lors de la partie convolutive d'un Convolutional Neural Network, l'image fournie en entrée passe à travers une succession de filtres de convolution. Par exemple, il existe des filtres de convolution fréquemment utilisés et permettant d'extraire des caractéristiques plus pertinentes que des pixels comme la détection des bords (filtre dérivateur) ou des formes géométriques. Le choix et l'application des filtres se fait automatiquement par le modèle. [WEB 16]

Parmi les filtres les plus connus, on retrouve notamment le filtre moyenneur (calcule pour chaque pixel la moyenne du pixel avec ses 8 proches voisins) ou encore le filtre gaussien permettant de réduire le bruit d'une image fournie en entrée :

Voici un exemple des effets de ces deux différents filtres sur une image comportant un bruit important (on peut penser à une photographie prise avec une faible luminosité par exemple). Toutefois, un des inconvénients de la réduction du bruit est qu'elle s'accompagne généralement d'une réduction de la netteté :

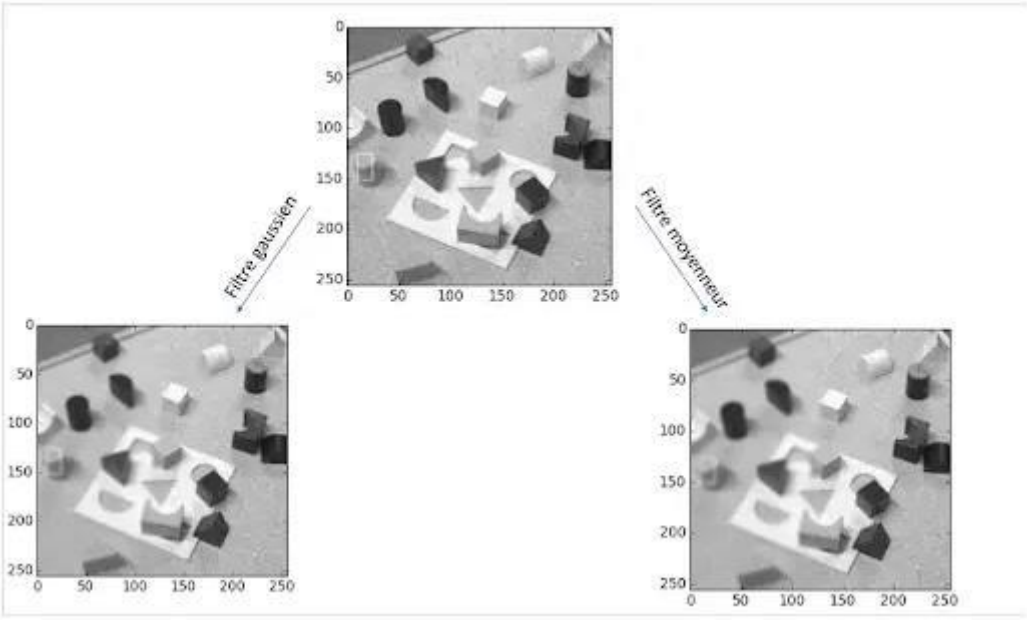


Figure 24 : Effet des filtres moyenneur et gaussien [WEB 16]

Outre sa fonction de filtrage, l'intérêt de la partie convolutive d'un CNN est qu'elle permet d'extraire des caractéristiques propres à chaque image en les compressant de façon à réduire leur taille initiale, via des méthodes de sous-échantillonnage tel que le Max-Pooling. [WEB 16]

## Méthode de sous échantillonnage : le Max-Pooling :

Le Max-Pooling est un processus de discrétisation basé sur des échantillons. Son objectif est de sous-échantillonner une représentation d'entrée (image, matrice de sortie de couche cachée, etc.) en réduisant sa dimension. De plus, son intérêt est qu'il réduit le coût de calcul en réduisant le nombre de paramètres à apprendre et fournit une invariance par petites translations (si une petite translation ne modifie pas le maximum de la région balayée, le maximum de chaque région restera le même et donc la nouvelle matrice créée restera identique). [WEB 17]

Pour rendre plus concret l'action du Max-Pooling, voici un exemple : imaginons que nous avons une matrice 4×4 représentant notre entrée initiale et un filtre d'une fenêtre de taille 2×2 que nous appliquerons sur notre entrée. Pour chacune des régions balayées par le filtre, le max-pooling prendra le maximum, créant ainsi par la même occasion une nouvelle matrice de sortie où chaque élément correspondra aux maximums de chaque région rencontrée.

Illustrons le processus :

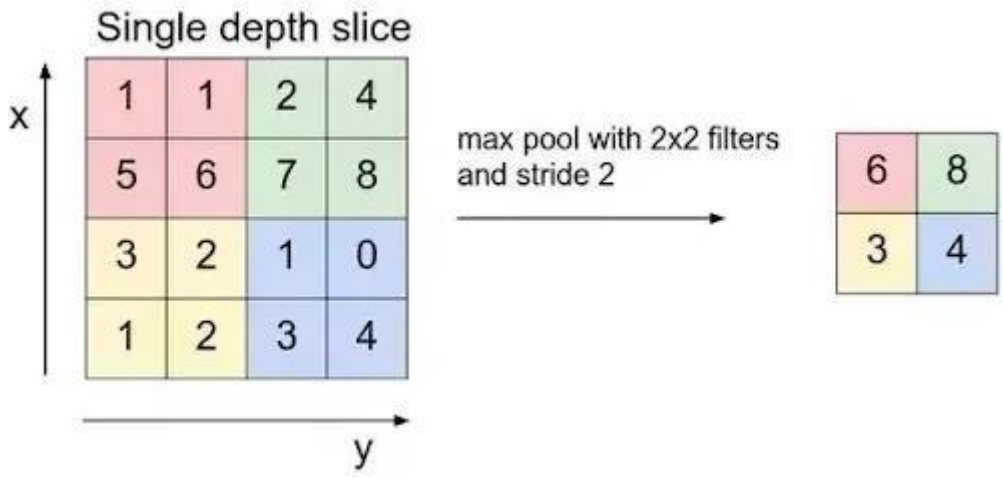


Figure 25 : Processus de Max-Pooling [WEB 17]

La fenêtre de filtre se déplace de deux pixels vers la droite (stride/pas = 2) et récupère à chaque pas "l'argmax" correspondant à la valeur la plus grande parmi les 4 valeurs de pixels.

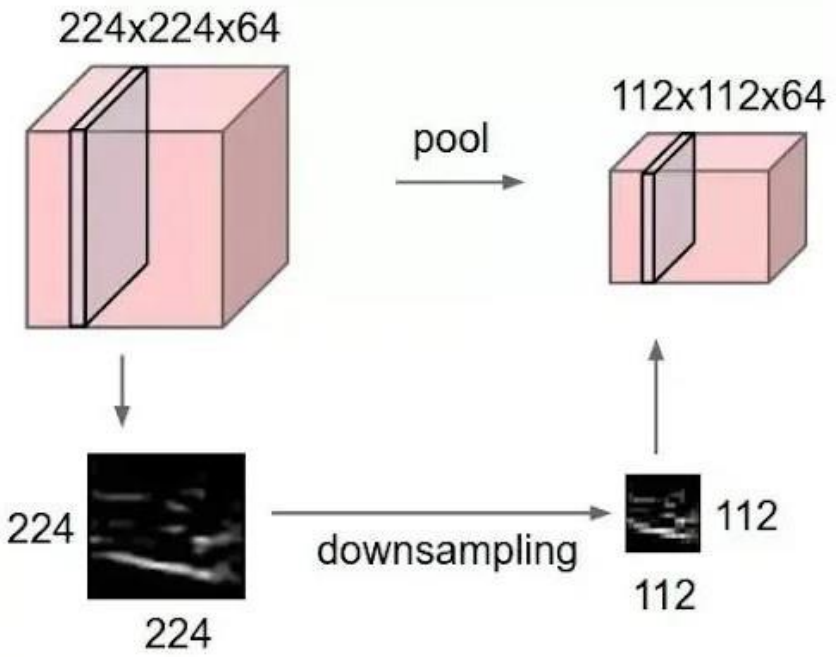


Figure 26 : Exemple d'effet du Max-Pooling [WEB 17]

On comprend mieux l'utilité de la partie convolutive d'un CNN : contrairement à un modèle MLP classique, l'ajout en amont de la partie convolutive permet d'obtenir en sortie une "carte de caractéristiques" ou "code CNN" (matrice de pixels située à droite dans l'exemple) dont les dimensions sont plus petites que celles de l'image initiale ce qui va avoir l'avantage de diminuer grandement le nombre de paramètres à calculer dans le modèle [WEB 17].

### 2) Une partie classification :

Le code CNN obtenu en sortie de la partie convolutive est fourni en entrée dans une deuxième partie, constituée de couches entièrement connectées appelées perceptron multicouche (MLP pour Multi Layers Perceptron). Le rôle de cette partie est de combiner les caractéristiques du code CNN afin de classer l'image.

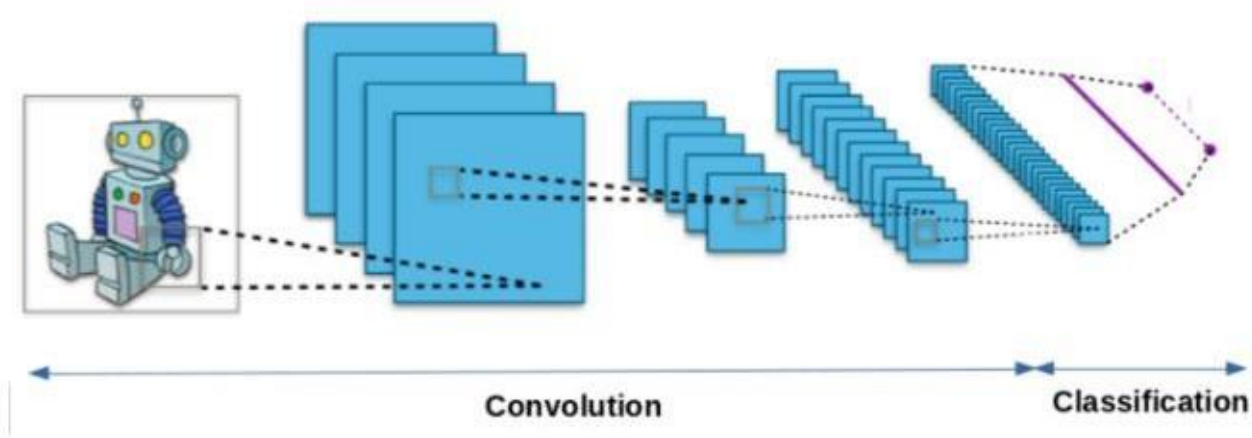


Figure 23 : Schéma représentant l'architecture d'un CNN [WEB 15]

1. Dans un premier temps, on définit la taille de la fenêtre de filtre située en haut à gauche.
2. La fenêtre de filtre, représentant la feature, se déplace progressivement de la gauche vers la droite d'un certain nombre de cases défini au préalable (le pas) jusqu'à arriver au bout de l'image.
3. À chaque portion d'image rencontrée, un calcul de convolution s'effectue permettant d'obtenir en sortie une carte d'activation ou feature map qui indique où localisées les features dans l'image : plus la feature map est élevée, plus la portion de l'image balayée ressemble à la feature.

## 2.9.2 Les réseaux de neurones récurrents (RNN)

Les Réseaux de Neurones récurrents traitent l'information en cycle. Ces cycles permettent au réseau de traiter l'information plusieurs fois en la renvoyant à chaque fois au sein du réseau.

La force des Réseaux de neurones récurrents réside dans leur capacité de prendre en compte des informations contextuelles suite à la récurrence du traitement de la même information. Cette dynamique auto-entretient le réseau.

Les Réseaux de neurones récurrents se composent d'une ou plusieurs couches. Le modèle de Hopfield (réseau temporel) est le réseau de neurones récurrent d'une seule couche le plus connu.

Les Réseaux de neurones récurrents à couches multiples revendiquent quant à eux la particularité de posséder des couples (entrée/sortie) comme les perceptrons entre lesquels la donnée véhicule à la fois en propagation en avant et en rétro propagation. [WEB 10]

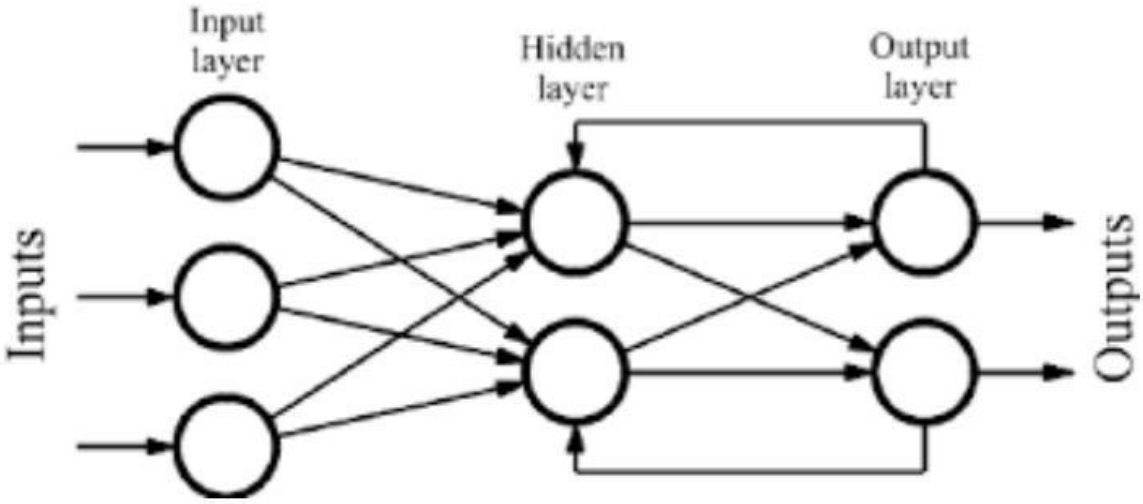


Figure 27 : Réseau de neurones récurrents [WEB 11]

## 2.9.3 Les réseaux de neurones à résonance

L'appellation du réseau neuronal fait encore une fois référence à son fonctionnement. En effet, au sein des réseaux de neurones à résonance, l'activation de tous les neurones est renvoyée à tous les autres neurones au sein du système. Ce renvoi provoque des oscillations, d'où la raison du terme résonance.

Il va sans dire que ces réseaux de neurones peuvent prendre différentes formes avec des degrés de complexité plutôt élevés. Pour aller plus loin, je vous invite à vous intéresser à la Mémoire Associative Bidirectionnel qui permet d'associer deux informations de natures différentes ou encore le modèle ART (Adaptative Resonance Theory) qui fait interagir une information contextuelle avec la connaissance que l'on a déjà pour identifier ou reconnaître des objets. [WEB 10]

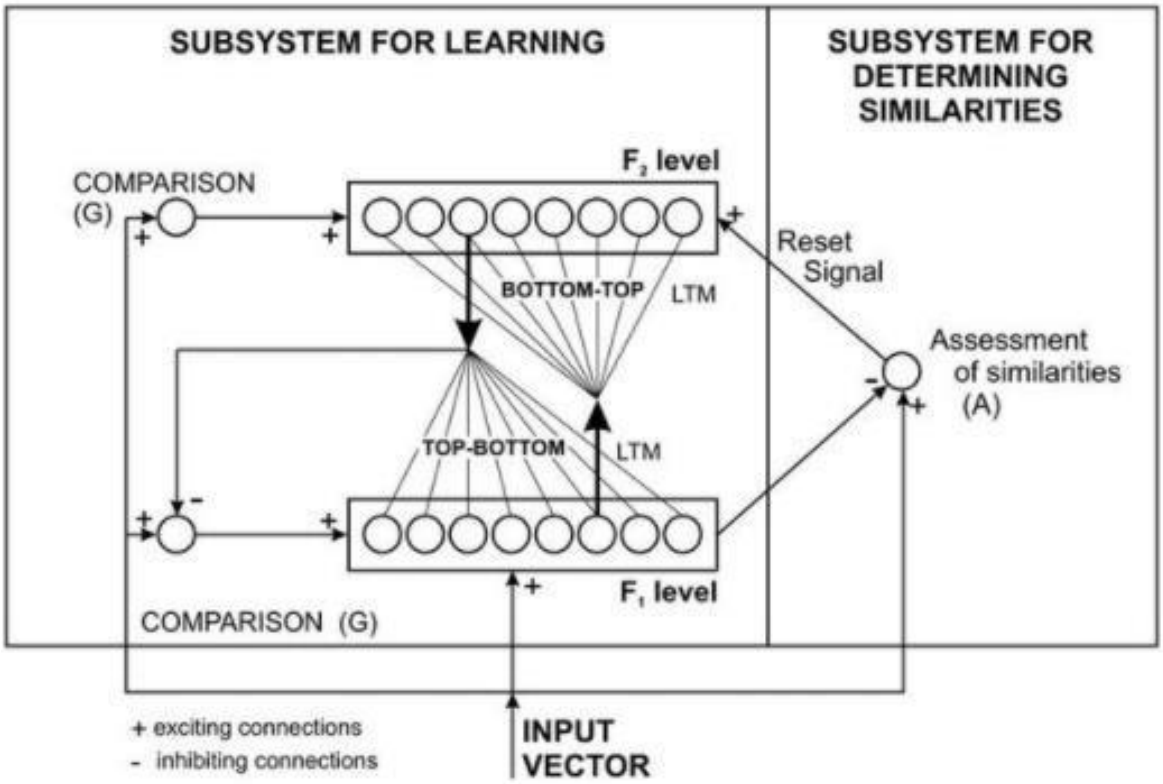


Figure 28 : Les réseaux de neurones à résonance [WEB 11]

## 2.9.4 Les réseaux de neurones auto-organisés

Les Réseaux de neurones auto-organisés sont surtout adaptés pour le traitement de d'informations spatiales. Par des méthodes d'apprentissage non-supervisé, les réseaux neuronaux auto-organisés sont capables d'étudier la répartition de données dans des grands espaces comme par exemple pour des problématiques de clusterisation ou de classifications. [WEB 10]

Le modèle le plus connu de ce type de réseaux de neurones est sans doute la carte auto-organisatrice de Kohonen :

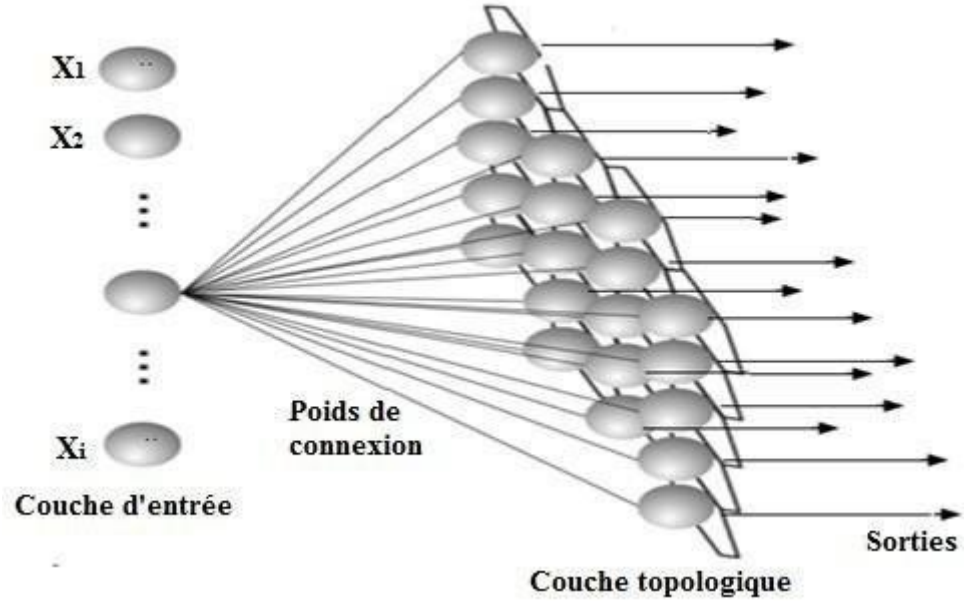


Figure 29 : Les réseaux de neurones auto-organisés [WEB 11]

## 2.10 Transfer Learning : définition

Le Transfer Learning, ou apprentissage par transfert en français, désigne l'ensemble des méthodes qui permettent de transférer les connaissances acquises à partir de la résolution de problèmes donnés pour traiter un autre problème.

Le Transfer Learning a connu un grand succès avec l'essor du Deep Learning. En effet, bien souvent, les modèles utilisés dans ce domaine nécessitent des temps de calcul élevés et des ressources importantes. Or, en utilisant des modèles pré-entraînés comme point de départ, le Transfer Learning permet de développer rapidement des modèles performants et résoudre efficacement des problèmes complexes en Computer Vision ou Natural Language Processing, NLP. [WEB 18]

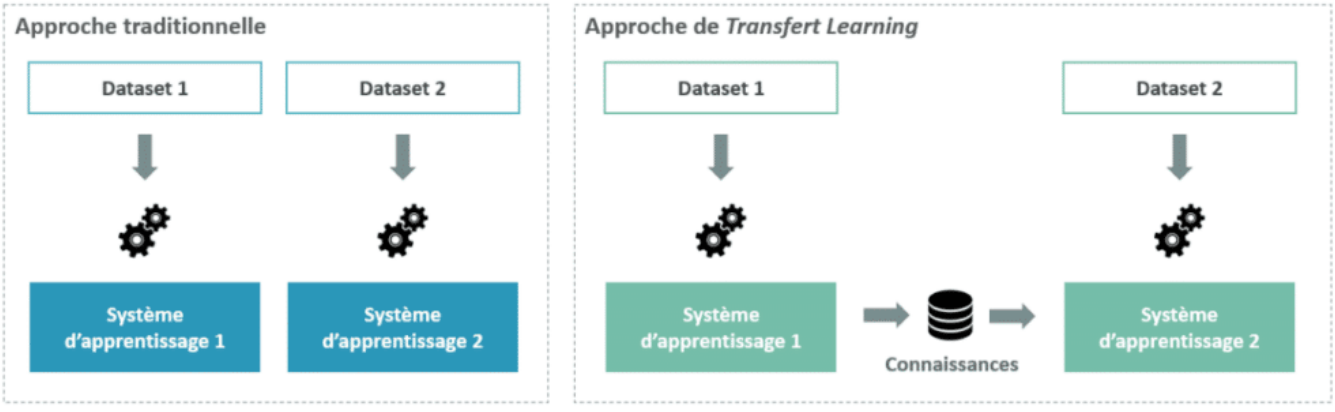


Figure 30 : Approche traditionnelle vs. Approche de Transfert Learning [WEB 18]

Intuitivement, le Transfer Learning s'inspire fortement du processus avec lequel nous apprenons.

Prenons l'exemple de quelqu'un qui maîtrise la guitare et souhaite apprendre à jouer au piano. Il pourra capitaliser sur ses connaissances en musique pour apprendre à jouer un nouvel instrument. De la même manière, un modèle de reconnaissance de voiture pourra être très rapidement réadapté à la reconnaissance de camions.

### 2.10.1 Les techniques de Transfer Learning

Le Transfer Learning repose sur une idée simple, celle de ré-exploiter les connaissances acquises dans d'autres configurations (sources) pour la résolution d'un problème particulier (cible). Dans ce contexte, on peut distinguer plusieurs approches selon ce que l'on souhaite transférer, quand et comment réaliser le transfert. Globalement, nous pouvons distinguer trois types de Transfer Learning :

#### 2.10.1.1 Apprentissage par transfert inductif, ou Inductive Transfer Learning

Dans cette configuration, les domaines source et cible sont les mêmes (même données), mais les tâches source et cible sont différentes mais proches. L'idée consiste alors à utiliser les modèles existants pour réduire de manière avantageuse le champ d'application des modèles possibles (biais de modèle) comme illustré dans le figure ci-dessous.

Illustration de l'apprentissage par transfert inductif, ou Inductive Transfer Learning

Par exemple, il est possible d'utiliser un modèle entraîné pour la détection d'animaux sur des images pour construire un modèle capable d'identifier des chiens.

#### 2.10.1.2 Apprentissage par transfert non supervisé, ou Unsupervised Transfer Learning :

Comme dans le cas de l'apprentissage par transfert inductif, les domaines source et cible sont similaires, mais les tâches sont différentes. Toutefois, les données des deux domaines ne sont pas labellisées.

Il est souvent plus facile d'obtenir de grandes quantités de données non labellisées, à partir de bases de données et de sources sur le web par exemple, que des données labellisées. C'est pourquoi l'idée

d‘utiliser l’apprentissage non supervisé en combinaison avec le Transfer Learning suscite un grand intérêt.

A titre d’exemple, le Self-taught clustering est une approche qui permet de réaliser le clustering de petites collections de données cibles non labellisées, avec l’aide d’une grande quantité de données sources non labellisées. Cette approche s’avère plus performante que les approches de pointe traditionnellement utilisées, lorsque les données cibles sont labellisées de manière non pertinente.

#### 2.10.1.3 Apprentissage par transfert transductif, ou Transductive Transfer Learning :

Dans cette configuration, les tâches sources et cibles sont similaires, mais les domaines correspondants sont différents soit en termes de données ou de distributions de probabilités marginales.

Par exemple, les modèles de NLP, tels que ceux utilisés pour l’étiquetage morpho-syntaxique de mots, Part-Of-Speech Tagger (POS Tagger) en anglais, sont généralement entraînés et testés sur des données d’actualité comme celles du Wall Street Journal. Ils peuvent être adaptés aux données issues des réseaux sociaux dont le contenu est différent mais proche de celles des journaux. [WEB 18]

## 2.11 Conclusion

Dans ce chapitre, nous avons passé en revue quelques méthodes et concepts liés à la classification. Ainsi nous avons listé de manière non-exhaustive les différentes sortes de classification. Ceci dit il existe plusieurs méthodes différentes de classification qui permettent de traiter les images et Dans la deuxième partie de ce chapitre, nous allons focaliser sur les réseaux de neurones. La philosophie de ce chapitre reste intacte, c’est à dire rendre le contenu accessible au plus grand nombre de personnes.

# Chapitre III : Conception

Les leucocytes, également appelés globules blancs, sont un groupe de cellules qui protègent l'organisme contre les infections, ce qui constitue une partie importante du système immunitaire. La classification des globules blancs est largement utilisée pour diagnostiquer diverses maladies, telles que le SIDA, la leucémie, le myélome et l'anémie. Cependant, les méthodes conventionnelles de classification des globules blancs prennent beaucoup de temps et sont sujettes à des erreurs.

Dans ce chapitre, nous allons faire la conception de notre système intelligent de classifications, l'un des réseaux neuronaux les plus populaires, le réseau neuronal convolutionnel (CNN), est sélectionné pour classifier et détecter les différents types de globules blancs, à savoir l'éosinophile, le lymphocyte, le monocyte et le neutrophile. Le CNN a été entraîné avec le jeu de données obtenu auprès de la plateforme Kaggle. Ensuite, des filtres gaussiens et médians ont été appliqués séparément aux images de la base de données. Les résultats obtenus après l'application des deux filtres aux images étaient meilleurs que les résultats obtenus avec les données originales. Les résultats de la recherche facilitent le diagnostic des maladies liées au sang.

## 3.1 Dataset

La classification s’applique à plusieurs types de données de plusieurs domaines (médicine, biologie, chimie, marketing etc.).

Le jeu de données obtenu auprès de la plateforme Kaggle contient 12 500 images augmentées de cellules sanguines au format JPEG. Avec les étiquettes de type de cellule qui les accompagnent. Les types de cellules sont Eosinophil, Lymphocyte,Monocyte et Neutrophile. Il y a environ 3 000 images pour chacun des quatre types de cellules. Ce jeu de données est accompagné d'un jeu de données supplémentaire contenant les 410 images originales pré-augmentées, ainsi que les 410 images originales. Il contient également 2500 images augmentées ainsi que quatre étiquettes de sous-type supplémentaires (JPEG+ CSV). Le site Il y a approximativement 2 000 images augmentées pour chacune des classes par rapport à 88, 33, 21 et 207 des images originales.

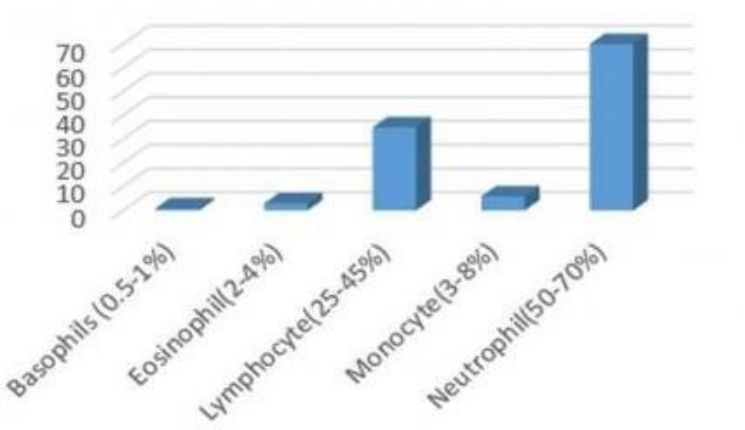


Figure 31 : Valeurs moyennes pour une numération normale de globules blancs chez l'adulte

Les images sanguines d'une personne jouent un rôle important dans le diagnostic et le contrôle de nombreuses maladies. Le corps humain adulte compte environ 4 000 à 1 000 000 de leucocytes. Un nombre de leucocytes en dehors de cette fourchette peut être le signe de diverse maladie. Les valeurs moyennes n des globules blancs d'une personne normale sont indiquées dans la figure 31.

## 3.2 Prétraitement des données

Puisque le jeu de données que nous avons est petit, nous avons décidé d'augmenter les images par rotation, réflexion autour de l'axe horizontal et décalage et un décalage horizontal et vertical. Nous devons nous assurer que le temps de calcul sans perdre trop de précision. Par conséquent, la taille de l'image d'entrée a été réduite à 80 60. Les transformations d'échelle ne sont pas appliquées car l'identification correcte l'identification correcte du type de cellule dépend de la taille des noyaux.

Les quatre types de cellules ont été transformés en un vecteur à 4 dimensions avec un codage à un coup, c'est-à-dire que toutes les composantes sont nulles, à l'exception de celle qui correspond à la classe appropriée.

Par exemple, le type de cellule 'NEUTROPHIL' peut être codé comme suit comme [1,0,0,0], et 'MONOCYTE' peut être codé comme [0,0,1,0]. Cette transformation nous permet d'utiliser la fonction de perte softmax, définie comme suit :

$$y_k = \frac{\exp(\phi_k)}{\sum_j^c \exp(\phi_j)},$$

## 3.3 Système proposé

Le but de ce travail est la proposition d'un système de classification de cellules sanguines blanches. Le système proposé a choisi de développer un site web qui détecte les globules blancs. Il dispose d'algorithmes et deux modèles pour reconnaître les sous types des cellules et le groupe de la cellule en utilisant le réseau neuronal convolutif.

Dans les parties suivantes, on présente l'architecture du système, les paramètres et les méthodes d'apprentissage utilisés.

## 3.4 Architecture du système

Cette section fournit une description détaillée de l'estimation proposée des globules blancs en utilisant des techniques de traitement d'image. L'image microscopique de l'échantillon de sang est prise comme entrée, les échantillons de sang contenant des GR, des GB et des plaquettes. La première étape consiste en un prétraitement au cours duquel les globules rouges et les plaquettes sont éliminés.

L'architecture de système comme suit :

### 3.4.1 L'étape de prétraitement comprend

a) Elimination des Globules Rouges :

- L'image microscopique de l'échantillon de sang est prise comme entrée pour l'élimination des GR.
- L'élimination des GB se fait par la valeur de l'intensité verte où l'intensité verte est plus élevée pour les RBC par rapport aux WBC.
- Si l'intensité verte est faible, le pixel devient noir et le pixel avec une intensité verte élevée devient blanc.

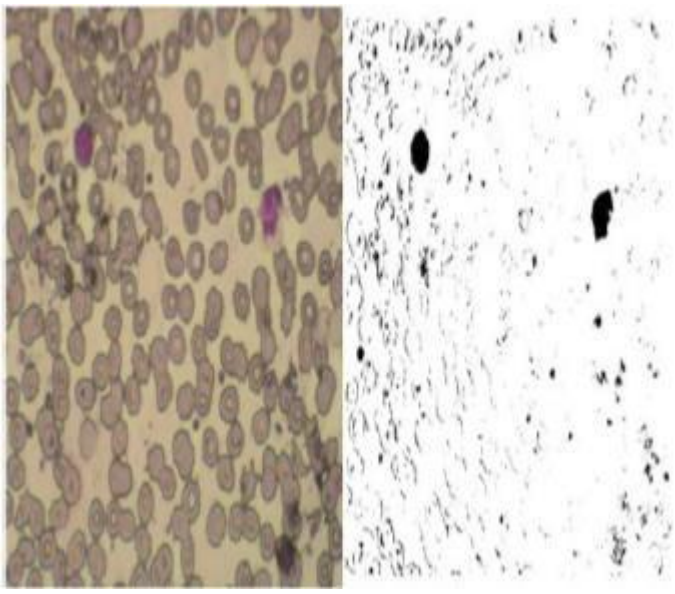

Figure 32 : Elimination des Globules Rouges

b) L'élimination des plaquettes :

- Après avoir retiré partiellement les globules rouges, les plaquettes ainsi que l'ensemble des globules rouges sont retirés en considérant chaque pixel et en comptant le pixel noir et le pixel blanc du pixel considéré.
- Si la valeur du pixel blanc est plus élevée, on considère qu'il s'agit d'un RBC et de plaquettes et ils sont éliminés en conséquence, car la zone de WBC est plus importante lorsque la valeur du pixel blanc est plus élevée.

### 3.4.2 Classification des globules blancs

La classification du WBC est effectuée par un réseau neuronal convolutif. Les 4 types d'images de WBC sont entraînés par les couches disponibles dans le CNN et les caractéristiques sont extraites en conséquence.

Le WBC localisé dans une fenêtre est donné comme entrée au CNN entraîné et le type particulier localisé est classé. Les couches utilisées dans le CNN sont :

- Couche de normalisation
- Couche d'entrée
- Couche de convolution
- Couche de MaxPooling
- Couche entièrement connectée (FC)
- Couche de classification

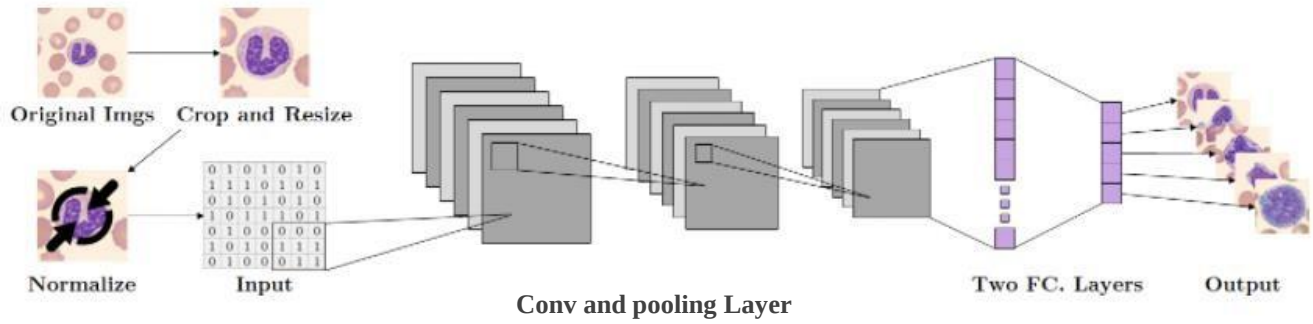


Figure 33 : Une vue d'ensemble du prétraitement et de l'architecture CNN proposée pour la classification des images WBC [WEB 19]

Le prétraitement consiste à recadrer, redimensionner et normaliser.

## 1. La couche de convolution

La couche de convolution est la composante clé du notre model CNN, et constitue toujours au moins leur première couche.

Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée. Pour cela, on réalise un filtrage par convolution : le principe est de faire "glisser" une fenêtre représentant la feature sur l'image, et de calculer le produit de convolution entre la feature et chaque portion de l'image balayée. Une feature est alors vue comme un filtre : les deux termes sont équivalents dans ce contexte.

La couche de convolution reçoit donc en entrée plusieurs images, et calcule la convolution de chacune d'entre elles avec chaque filtre. Les filtres correspondent exactement aux features que l'on souhaite retrouver dans les images.

On obtient pour chaque paire (image, filtre) une carte d'activation, ou feature map, qui nous indique où se situent les features dans l'image : plus la valeur est élevée, plus l'endroit correspondant dans l'image ressemble à la feature.

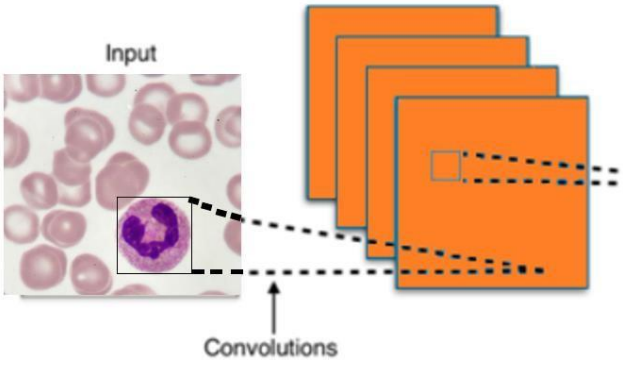


Figure 34 : La couche convolution [WEB 15]

## 2. La couche de Maxpooling

Cette couche est souvent placée entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling.

L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules adjacentes de taille 2 × 2 pixels qui ne se chevauchent pas, ou des cellules de taille 3 × 3 pixels, distantes les unes des autres d'un pas de 2 pixels.

On obtient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites.

La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage.

## 3. Couche entièrement connectée (FC)

La couche FC constitue toujours la dernière couche d'un réseau de neurones convolutif ou non – elle n'est donc pas caractéristique d'un CNN.

Ce type de couche reçoit un vecteur en entrée et produit un nouveau vecteur en sortie. Pour cela, elle applique une combinaison linéaire puis éventuellement une fonction d'activation aux valeurs reçues en entrée.

La couche FC détermine le lien entre la position des features dans l'image et une classe. En effet, le tableau en entrée étant le résultat de la couche précédente, il correspond à une carte d'activation pour une feature donnée : les valeurs élevées indiquent la localisation (plus ou moins précise selon le pooling) de cette feature dans l'image. Si la localisation d'une feature à un certain endroit de l'image est caractéristique d'une certaine classe, alors on accorde un poids important à la valeur correspondante dans le tableau.

Dans cette partie du notre model CNN deux fonctions activations sont utilisées : Softmax et Rectified Linear Unit (ReLu)

- Softmax : La fonction softmax est une fonction qui transforme un vecteur de K valeurs réelles en un vecteur de K valeurs réelles qui totalisent 1. Les valeurs d'entrée peuvent être positives, négatives, nulles ou supérieures à un, mais le softmax les transforme en valeurs entre 0 et 1, afin qu'ils puissent être interprétés comme des probabilités. Si l'une des entrées est petite ou négative, le softmax la transforme en une petite probabilité, et si une entrée est grande, alors elle la transforme en une grande probabilité, mais elle restera toujours entre 0 et 1.
- Rectified Linear Unit (ReLu) : Il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. La fonction ReLU (abréviation de Unités Rectifié linéaires) : F(x)=max(0,x) Cette fonction force les neurones à retourner des valeurs positives.

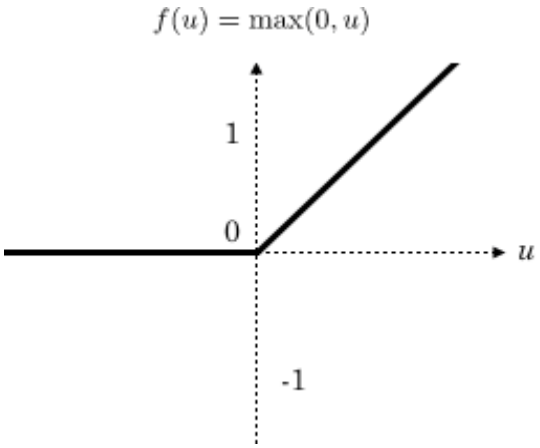


Figure 35 : Allure de la fonction ReLU [WEB 20]

## 4. La fonction de perte

C'est une fonction pour évaluer l'effet de l'algorithme d'apprentissage sur la modélisation de l'ensemble de données. Si la prédiction est complètement fausse, la fonction produira une grande valeur, si non, la valeur sera petite. Plus simplement, la fonction de perte permet de calculer l'erreur entre un ensemble de sorties prédites et leurs étiquettes. Il existe une multitude de fonctions de perte, chacune est utiliser selon le problème traité. Pour l'apprentissage du système proposé dans ce travail, on a utilisé la fonction «keras.losses.categorical_crossentropy».

Durant l'apprentissage du modèle, le but est de minimiser cette fonction d'erreur, et avoir des prédictions aussi correctes et optimisées que possible.

Dans la partie suivante, on va voir comment on peut arriver à cela.

## 5. Optimiseur

Les optimiseurs façonnent et moulent votre modèle dans sa forme la plus précise possible en utilisant les poids. La fonction de perte est le guide du terrain, indiquant à l'optimiseur quand il se déplace dans la bonne ou la mauvaise direction. Il existe plusieurs algorithmes d'optimisation, pour notre système on a utilisé l'optimiseur 'Adam'.

Adam est un algorithme basé sur un gradient de premier ordre de fonctions objectives stochastiques, basé sur des estimations adaptatives de moments d'ordre inférieur. Adam est l'un des derniers algorithmes d'optimisation de pointe utilisés par de nombreux praticiens de l'apprentissage automatique. Le premier moment normalisé par le second moment donne le sens de la mise à jour.

Règle de mise à jour Adam:

$$\theta_{n+1} = \theta_n - \frac{\alpha}{\sqrt{\hat{v}_n + \epsilon}} \hat{m}_n$$

## 6. L'apprentissage du modèle

Après avoir déterminé la topologie et défini divers paramètres CNN, l'apprentissage est effectué en se basant sur les données des cellules, le modèle doit apprendre à classer chaque sous type des globules blancs. Deux autres hyper-paramètres sont définis pour l'entraînement du modèle : le nombre

d’époques, et la taille du batch. Le nombre d'époques est un hyper-paramètre qui définit le nombre de fois que l'algorithme d'apprentissage fonctionnera sur l'ensemble de données d’entraînement.

La taille du batch est un hyper-paramètre spécifie le nombre d’échantillons à traiter avec la mise à jour de poids du modèle.

L’apprentissage du CNN proposé est établi à travers 15 époques avec un batch de taille 64.

Le modèle est ensuite validé avec un autre ensemble de données qui n’a pas été utilisé dans la phase d’apprentissage, pour voir s’il est arrivé à généraliser durant l’entraînement ou pas.

### 7. Application basée sur le Web

Une application Web a été développée pour démontrer la compatibilité du modèle sur le serveur. Cette application a été développée avec Gradio.

## 3.5 Conclusion

Dans ce chapitre nous avons détaillé l'architecture du notre système pour classifier les sous type des globules blancs. Nous avons commencé par la source de la base de données utilisée, Ensuite on a parlé sur l’architecture du modèle, en précisant aussi les paramètres d’apprentissage en entrée tels que les fonctions d’activation, le nombre d’époques d’entraînement, l’algorithme d’optimisation utilisé…etc.

# Chapitre IV : Implémentation

Dans ce dernier chapitre du mémoire nous présentons la phase d'implémentation. On présente le langage de programmation utilisés et les différentes bibliothèques nécessaires jusqu'à l'obtention des résultats à analyser. Nous nous attacherons dans ce chapitre à détailler les choix d'implémentation et les caractéristiques de chacune d'elles.

## 4.1 Outils utilisés

Pour l'implémentation nous avons utilisé Google colab afin d'écrire le programme en langage python.

### 4.1.1 Langage utilisé (Python)

Langage de programmation utilisé en machine learning et en data science, le langage Python s'impose également dans d'autres secteurs d'activité grâce à sa simplicité et sa compatibilité.

Le langage Python est un langage de programmation open source multi-plateformes et orienté objet. Grâce à des bibliothèques spécialisées, Python s'utilise pour de nombreuses situations comme le développement logiciel, l'analyse de données, ou la gestion d'infrastructures. Il n'est donc pas, comme le langage HTML par exemple, uniquement dédié à la programmation web.

Langage de programmation interprété, Python permet l'exécution du code sur n'importe quel ordinateur. Utilisable aussi bien par des programmeurs débutants qu'experts, Python permet de créer des programmes de manière simple et rapide.

Les principaux usages de Python

Python est principalement utilisé pour le scripting et l'automatisation de tâches simples mais fastidieuses, c'est-à-dire l'interaction avec les navigateurs web. Mais Python est aussi utilisé pour :

- programmer des applications.
- générer du code.
- créer des services web.
- faire de la métaprogrammation.

Langage principalement utilisé pour le machine learning et la data science, Python a fortement contribué à l'essor du big data. Grâce à ses nombreuses bibliothèques telles Panda, Bokeh, Numpy,

Scipy, Scrapy, Matpotlib, Scikit-Learn ou encore TensorFlow, Python offre une grande flexibilité dans les tâches à effectuer et une grande compatibilité quelle que soit la plateforme utilisée [WEB 21].


Figure 36 : Logo du langage Python

### 4.1.2 Google Colab

Google Colab ou Colaboratory est un service cloud fourni par Google (gratuit), basé sur Jupyter Notebook. Cette application web open source destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud, traiter et visualiser un ensemble de données, partager les documents contenant du code, etc.

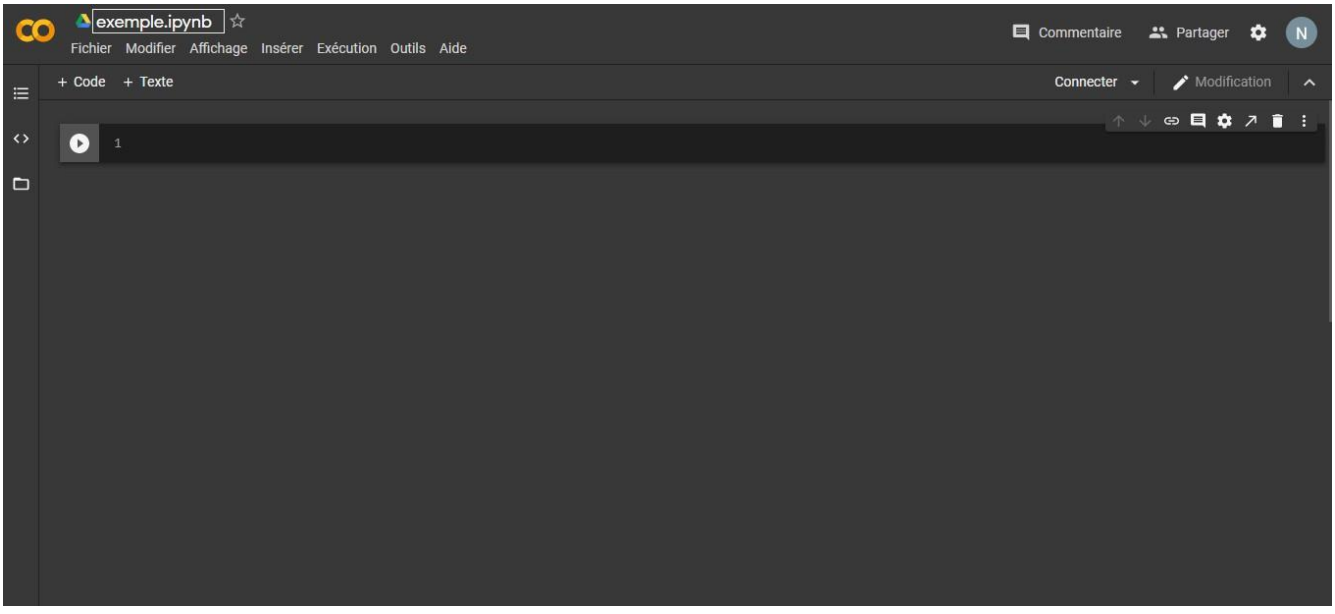


Figure 37 : Une interface de l'environnement de travail Google Colab

## 4.2 Étapes de l'implémentation et le code source

Les différentes étapes de l'implémentation sont résumées sur le Schéma suivant :

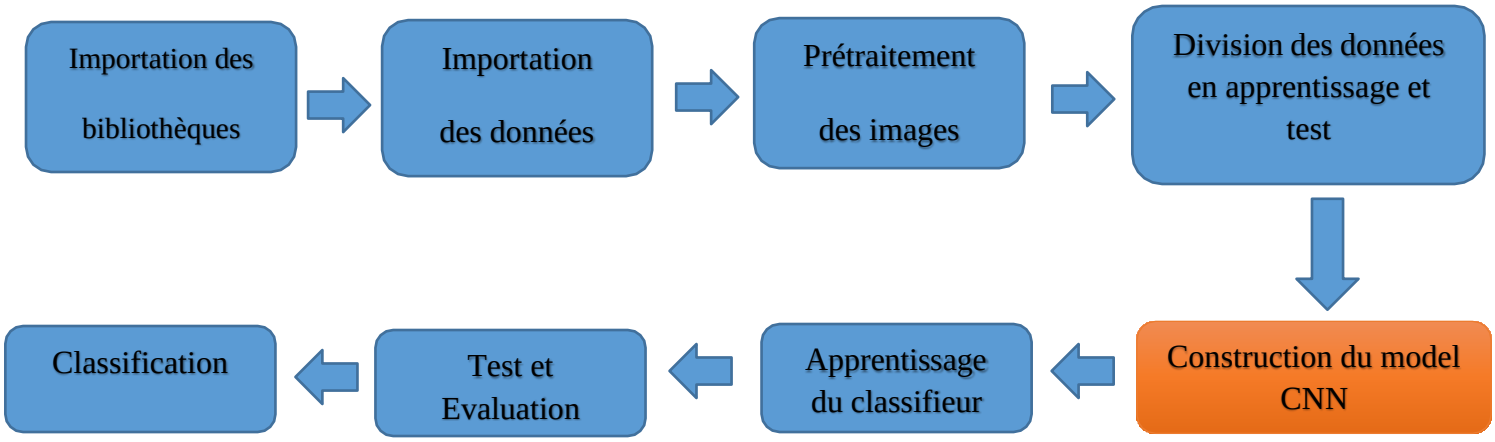


Figure 38 : Étapes de l'implémentation

### 4.2.1 Importation des bibliothèques nécessaires

La première partie du programme est l'importation des bibliothèques nécessaires pour mener à bien les différentes tâches. Pour ce faire nous avons eu recours à des bibliothèques principales ci-dessous.

#### a) NumPy

NumPy est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que les tableaux masqués et les matrices) et un assortiment de routines permettant d'effectuer des opérations rapides sur les tableaux, notamment des opérations mathématiques, logiques, de manipulation de formes, de tri, de sélection, d'E/S, de transformées de Fourier discrètes, d'algèbre linéaire de base, d'opérations statistiques de base, de simulation aléatoire et bien plus encore.

Au cœur du package NumPy se trouve l'objet ndarray. Il encapsule des tableaux à n dimensions de types de données homogènes, de nombreuses opérations étant effectuées en code compilé pour des raisons de performances. Il existe plusieurs différences importantes entre les tableaux NumPy et les tableaux Python standard :

- Les tableaux NumPy ont une taille fixe à la création, contrairement aux listes Python (qui peuvent croître dynamiquement). Changer la taille d'un tableau NumPy va créer un nouveau tableau et supprimer l'original.
- Les éléments d'un tableau NumPy doivent tous être du même type de données, et auront donc la même taille en mémoire. L'exception : on peut avoir des tableaux d'objets (Python, y compris NumPy), permettant ainsi des tableaux d'éléments de tailles différentes.
- Les tableaux NumPy facilitent les opérations mathématiques avancées et d'autres types d'opérations sur de grands nombres de données. En général, ces opérations sont exécutées plus efficacement et avec moins de code que ne le permettent les séquences intégrées de Python.
- Une pléthore croissante de paquets scientifiques et mathématiques basés sur Python utilisent les tableaux NumPy ; bien qu'ils prennent généralement en charge l'entrée de séquences Python, ils convertissent cette entrée en tableaux NumPy avant le traitement, et ils produisent souvent des tableaux NumPy. En d'autres termes, pour utiliser efficacement une grande partie (peut-être même la plupart) des logiciels scientifiques/mathématiques actuels basés sur Python, il ne suffit pas de savoir comment utiliser les types de séquences intégrés à Python - il faut aussi savoir comment utiliser les tableaux NumPy [WEB 22].


Figure 39 : Logo Numpy

## b) Keras

Keras est une bibliothèque open-source de composants de réseaux neuronaux écrits en Python. Keras est capable de fonctionner au-dessus de TensorFlow, Theano, PlaidML et autres. La bibliothèque a été développée pour être modulaire et conviviale, mais elle a initialement commencé dans le cadre d'un projet de recherche pour le système d'exploitation intelligent neuro-électronique ouvert ou ONEIROS. L'auteur principal de Keras est François Chollet, un ingénieur de Google qui a également écrit XCeption, un modèle de réseau neuronal profond. Bien que Keras ait été officiellement lancé, il n'a été intégré à la bibliothèque centrale TensorFlow de Google qu'en 2017. Un support supplémentaire a également été ajouté pour l'intégration de Keras avec Microsoft Cognitive Toolkit.

Composée d'une bibliothèque de composants d'apprentissage automatique couramment utilisés, notamment des objectifs, des fonctions d'activation et des optimiseurs, la plateforme open-source de Keras offre également un support pour les réseaux neuronaux récurrents et convolutifs. En outre, Keras propose le développement de plateformes mobiles pour les utilisateurs ayant l'intention de mettre en œuvre des modèles d'apprentissage profond sur les smartphones, tant iOS qu'Android. En 2018, la bibliothèque compte 22 % d'utilisation à travers de ses plus de 200 000 utilisateurs [WEB 23].

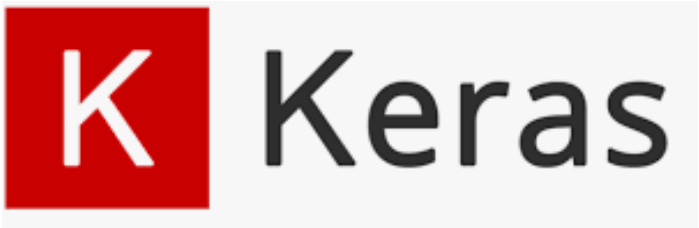


Figure 40 : Logo Keras

## c) Pandas

Pandas est un paquet Python open source qui est le plus largement utilisé pour la science des données, l'analyse des données et les tâches d'apprentissage automatique. Il est construit au-dessus d'un autre paquet nommé Numpy, qui fournit un support pour les tableaux multidimensionnels. Pandas est l'un des paquets de traitement de données les plus populaires et fonctionne bien avec de nombreux autres modules de science des données dans l'écosystème Python. Il est généralement inclus dans toutes les distributions Python, qu'il s'agisse de celles fournies avec votre système d'exploitation ou de distributions commerciales comme ActivePython d'ActiveState [WEB 24].

Pandas simplifie l'exécution d'un grand nombre de tâches répétitives et fastidieuses liées à l'exploitation des données, notamment :

- le nettoyage des données
- Remplissage de données
- Normalisation des données
- Fusions et jointures
- la visualisation des données
- Analyse statistique
- Inspection des données
- Chargement et sauvegarde des données
- Et bien d'autres choses encore


Figure 41 : Pandas Logo

## d) Gradio

Gradio est une bibliothèque python open-source qui vous permet de créer rapidement des parties d'interface utilisateur simples à utiliser et ajustables pour votre modèle ML, toute API ou toute capacité subjective en seulement quelques lignes de code. Il est plus facile de jouer avec vos modèles dans votre navigateur Web en faisant simplement glisser vos images, votre texte, l'enregistrement de votre propre voix, etc. et de voir le résultat en direct de manière interactive. Vous pouvez coordonner l'interface graphique directement dans votre carnet de notes Python, ou vous pouvez partager le lien avec n'importe qui.

Gradio aide à construire une interface graphique en ligne en quelques lignes de code, ce qui est pratique pour montrer des expositions de la présentation du modèle. Il est rapide, simple à mettre en place, prêt à être utilisé et partageable en tant que connexion publique que tout le monde peut obtenir pour exécuter à distance et en parallèle le modèle dans votre machine. Gradio fonctionne avec un large éventail de médias - texte, images, vidéo et son. En dehors des modèles ML, il peut très bien être utilisé comme des embeddings de code python [WEB 28].


Figure 42 : Gradio Logo

## 4.2.2 Importation des données

La première chose dont la machine a besoin pour réussir à apprendre à détecter le type de cellule sanguine qu'elle a trouvé, c'est de données.

Dans cette deuxième partie du code on a chargé toute les images des types de cellules : Eosinophil, Lymphocyte, Monocyte et Neutrophile et on a divisé les données en deux parties : une partie pour l'apprentissage et une partie pour le test de tel sorte que la partie test concerne 30% des données et donc 70% pour la partie d'apprentissage (train). Ainsi une fois la division faite, les parties x_train et x_test concerneront la colonne plot qui contient le résumé (prétraité) des cellules et les parties y_train, y_test, z_train et z_test concerneront les labels 1 et 2.

```
100%|██████████| 2499/2499 [00:12<00:00, 193.80it/s]
100%|██████████| 2497/2497 [00:13<00:00, 191.45it/s]
100%|██████████| 2483/2483 [00:13<00:00, 187.72it/s]
100%|██████████| 2488/2488 [00:13<00:00, 188.08it/s]
100%|██████████| 624/624 [00:03<00:00, 184.80it/s]
100%|██████████| 620/620 [00:03<00:00, 188.21it/s]
100%|██████████| 620/620 [00:03<00:00, 182.09it/s]
100%|██████████| 623/623 [00:03<00:00, 188.59it/s]
{1: 'NEUTROPHIL', 2: 'EOSINOPHIL', 3: 'MONOCYTE', 4: 'LYMPHOCYTE', 5: 'BASOPHIL'}
{0: 'Mononuclear', 1: 'Polynuclear'}
```

Figure 43 : Chargement de données

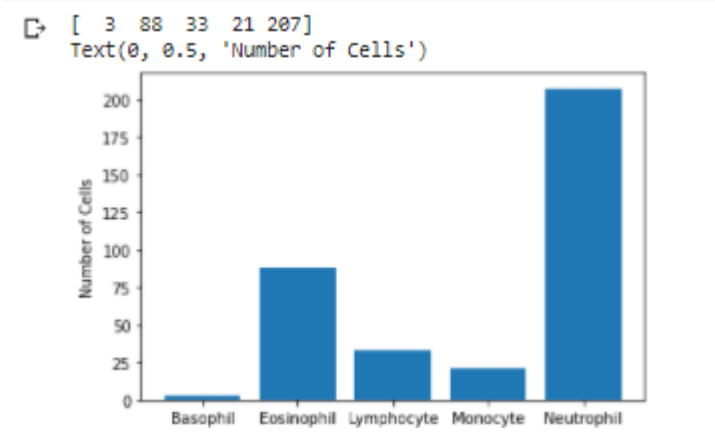


Figure 44 : nombre de cellules

Maintenant, nous devons faire un codage à chaud pour pouvoir calculer la perte pendant la propagation arrière.

Encodage à chaud : Il s'agit de transformer des mots/lettres (labels) en quelque chose que l'ordinateur peut comprendre comme [0,1].

```
from keras.utils.np_utils import to_categorical
y_trainHot = to_categorical(y_train, num_classes = 5)
y_testHot = to_categorical(y_test, num_classes = 5)
z_trainHot = to_categorical(z_train, num_classes = 2)
z_testHot = to_categorical(z_test, num_classes = 2)
print(labels)
print(labels2)
```

Figure 45 : Transformation des labels en langage machine [0,1]

## 4.2.3 Entrainement et la construction du model CNN

Un modèle CNN ou également connu sous le nom de réseau neuronal convolutif est un algorithme de Deep Learning qui utilise des filtres pour différencier une image d'une autre. La première couche de convolution suivie des couche Relu et MaxPooling et on a évaluez le modèle sur les données de test avant de former le modèle. Pour commencer, nous allons créer l'architecture principale du CNN :

```
def CNN(imgs,img_labels,test_imgs,test_labels,stride):
    epochs = 15
    num_classes = len(img_labels[0])
    batch_size = 64
    #Size of image
    img_rows,img_cols=imgs.shape[1],imgs.shape[2]
    input_shape = (img_rows, img_cols, 3)
    #Creating the model
    model = Sequential()
    #First convolution layer
    model.add(Conv2D(32, kernel_size=(3, 3),
                     activation='relu',
                     input_shape=input_shape,
                     strides=stride))
    #First maxpooling layer
    model.add(MaxPooling2D(pool_size=(2, 2)))
    #Second convolution layer
    model.add(Conv2D(64, (3, 3), activation='relu'))
    #Second maxpooling layer
    model.add(MaxPooling2D(pool_size=(2, 2)))
    #Third convolution layer
    model.add(Conv2D(128, (3, 3), activation='relu'))
    #Third maxpooling layer
    model.add(MaxPooling2D(pool_size=(2, 2)))
    #Convert the matrix to a fully connected layer
    model.add(Flatten())
    #Dense function to convert FCL to 128 values
    model.add(Dense(128, activation='relu'))
    #Final dense layer on which softmax function is performed
    model.add(Dense(num_classes, activation='softmax'))
    #Model parameters
    model.compile(loss=keras.losses.categorical_crossentropy,
                  optimizer='adam',
                  metrics=['accuracy'])
```

Figure 46 : Code source Entrainement de CNN

Les modèles doit être formé à l'aide d'un ensemble de données spécifique, puis testé par rapport à un ensemble de données différent. Les modèles sont entraînés pour 15 époques.

```
history = model.fit(imgs,img_labels,
                    shuffle = True,
                    epochs=epochs,
                    validation_data = (test_imgs, test_labels))
```

## Model 1 :

```
model = CNN(X_train,y_trainHot,X_test,y_testHot,1);
```

```
Epoch 1/15
156/156 [==============================] - 84s 530ms/step - loss: 1.3827 - accuracy: 0.2873 - val_loss: 1.2478 - val_accuracy: 0.3788
Epoch 2/15
156/156 [==============================] - 82s 527ms/step - loss: 1.0651 - accuracy: 0.5070 - val_loss: 0.9528 - val_accuracy: 0.5722
Epoch 3/15
156/156 [==============================] - 82s 527ms/step - loss: 0.7963 - accuracy: 0.6433 - val_loss: 0.6880 - val_accuracy: 0.6265
Epoch 4/15
156/156 [==============================] - 82s 525ms/step - loss: 0.6544 - accuracy: 0.7092 - val_loss: 0.7894 - val_accuracy: 0.6574
Epoch 5/15
156/156 [==============================] - 83s 533ms/step - loss: 0.5447 - accuracy: 0.7657 - val_loss: 0.5753 - val_accuracy: 0.7415
Epoch 6/15
156/156 [==============================] - 83s 530ms/step - loss: 0.4778 - accuracy: 0.7966 - val_loss: 0.5820 - val_accuracy: 0.7467
Epoch 7/15
156/156 [==============================] - 82s 529ms/step - loss: 0.4006 - accuracy: 0.8326 - val_loss: 0.5714 - val_accuracy: 0.7105
Epoch 8/15
156/156 [==============================] - 83s 532ms/step - loss: 0.3470 - accuracy: 0.8577 - val_loss: 0.5265 - val_accuracy: 0.7567
Epoch 9/15
156/156 [==============================] - 83s 531ms/step - loss: 0.2976 - accuracy: 0.8808 - val_loss: 0.5056 - val_accuracy: 0.7768
Epoch 10/15
156/156 [==============================] - 83s 530ms/step - loss: 0.2501 - accuracy: 0.8994 - val_loss: 0.5174 - val_accuracy: 0.7728
Epoch 11/15
156/156 [==============================] - 83s 531ms/step - loss: 0.2367 - accuracy: 0.9070 - val_loss: 0.6350 - val_accuracy: 0.7583
Epoch 12/15
156/156 [==============================] - 83s 531ms/step - loss: 0.2020 - accuracy: 0.9231 - val_loss: 0.5470 - val_accuracy: 0.7805
Epoch 13/15
```

Figure 47 : Code source construction du modèle CNN 1 (15 époques)

## Model 2 :

```
model = CNN(X_train,z_trainHot,X_test,z_testHot,2);
```

```
Epoch 1/15
156/156 [==============================] - 19s 117ms/step - loss: 0.6946 - accuracy: 0.4978 - val_loss: 0.6929 - val_accuracy: 0.4994
Epoch 2/15
156/156 [==============================] - 18s 115ms/step - loss: 0.6672 - accuracy: 0.5754 - val_loss: 0.5515 - val_accuracy: 0.7825
Epoch 3/15
156/156 [==============================] - 18s 115ms/step - loss: 0.4582 - accuracy: 0.7900 - val_loss: 0.5034 - val_accuracy: 0.7230
Epoch 4/15
156/156 [==============================] - 18s 116ms/step - loss: 0.3141 - accuracy: 0.8742 - val_loss: 0.2096 - val_accuracy: 0.9232
Epoch 5/15
156/156 [==============================] - 18s 117ms/step - loss: 0.2642 - accuracy: 0.8990 - val_loss: 0.2680 - val_accuracy: 0.9123
Epoch 6/15
156/156 [==============================] - 18s 118ms/step - loss: 0.1784 - accuracy: 0.9362 - val_loss: 0.1842 - val_accuracy: 0.9312
Epoch 7/15
156/156 [==============================] - 18s 118ms/step - loss: 0.1446 - accuracy: 0.9481 - val_loss: 0.2407 - val_accuracy: 0.9308
Epoch 8/15
156/156 [==============================] - 19s 119ms/step - loss: 0.1472 - accuracy: 0.9457 - val_loss: 0.1645 - val_accuracy: 0.9522
Epoch 9/15
156/156 [==============================] - 18s 118ms/step - loss: 0.1042 - accuracy: 0.9621 - val_loss: 0.2773 - val_accuracy: 0.9276
Epoch 10/15
156/156 [==============================] - 18s 117ms/step - loss: 0.0874 - accuracy: 0.9691 - val_loss: 0.2202 - val_accuracy: 0.9304
Epoch 11/15
156/156 [==============================] - 18s 117ms/step - loss: 0.0665 - accuracy: 0.9745 - val_loss: 0.2412 - val_accuracy: 0.9268
Epoch 12/15
156/156 [==============================] - 18s 117ms/step - loss: 0.0766 - accuracy: 0.9723 - val_loss: 0.3030 - val_accuracy: 0.9316
Epoch 13/15
```

Figure 48 : Code source construction du modèle CNN 2 (15 époques)

### 4.2.4 Discussion de résultats

Nous obtenons meilleurs résultats avec une précision de 90% pour le model 1 et 92% pour le model 2, Selon les tests effectués et les résultats obtenus, nous pouvons remarquer que le choix des paramètres dans un CNN est un critère sensible pour l'obtention des résultats satisfaisante. Ces résultats nous ont permis de rendre le model en production par la création d'un site web.

```
78/78 [==============================] - 5s 66ms/step - loss: 0.2262 - binary_accuracy: 0.9045

Keras CNN accuracy: 0.9045435786247253

The predicted class is :  1
The real class is :  1
```

```
78/78 [==============================] - 1s 19ms/step - loss: 0.3366 - accuracy: 0.9292

Keras CNN accuracy: 0.9292320013046265

The predicted class is :  1
The real class is :  1
Model: "sequential_1"
```

Figure 49 : Précision de résultat

Par conséquent, le modèle a une grande précision. La figure 50 et 51 montre l'écart entre l'apprentissage et la perte de validation correspondante pour l'ensemble de données. L'une des principales raisons d'atteindre cette précision réside dans MaxPooling. Il fournit les constantes de traduction initiales pour la représentation interne tout en réduisant le nombre de paramètres que le modèle doit apprendre. Ce processus d'estimation par échantillon réduit la représentation de l'entrée qui constitue l'image, en réduisant ses dimensions.

Le nombre de neurones a une valeur idéale de 64 ce qui n'est pas très élevé. Avoir un plus grand nombre de neurones et de filtres peut entraîner de moins bonnes performances. Les valeurs de filtre optimisées et la taille de la piscine permettent de filtrer la partie principale (la cellule) de l'image pour détecter leur type. Le système peut détecter efficacement les cellules.

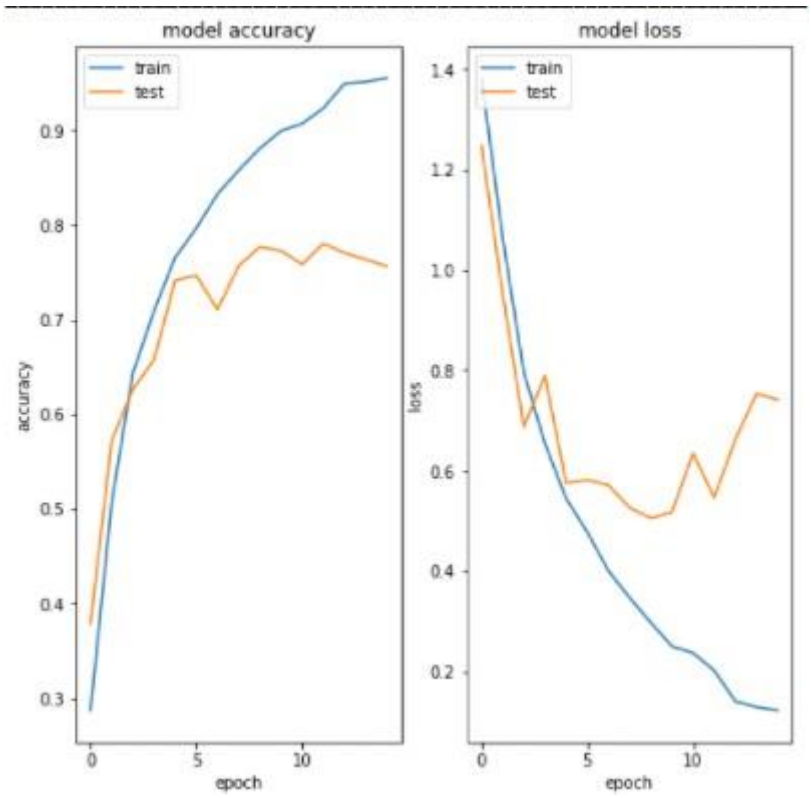


Figure 50 : Le model 1 Accuracy – Loss

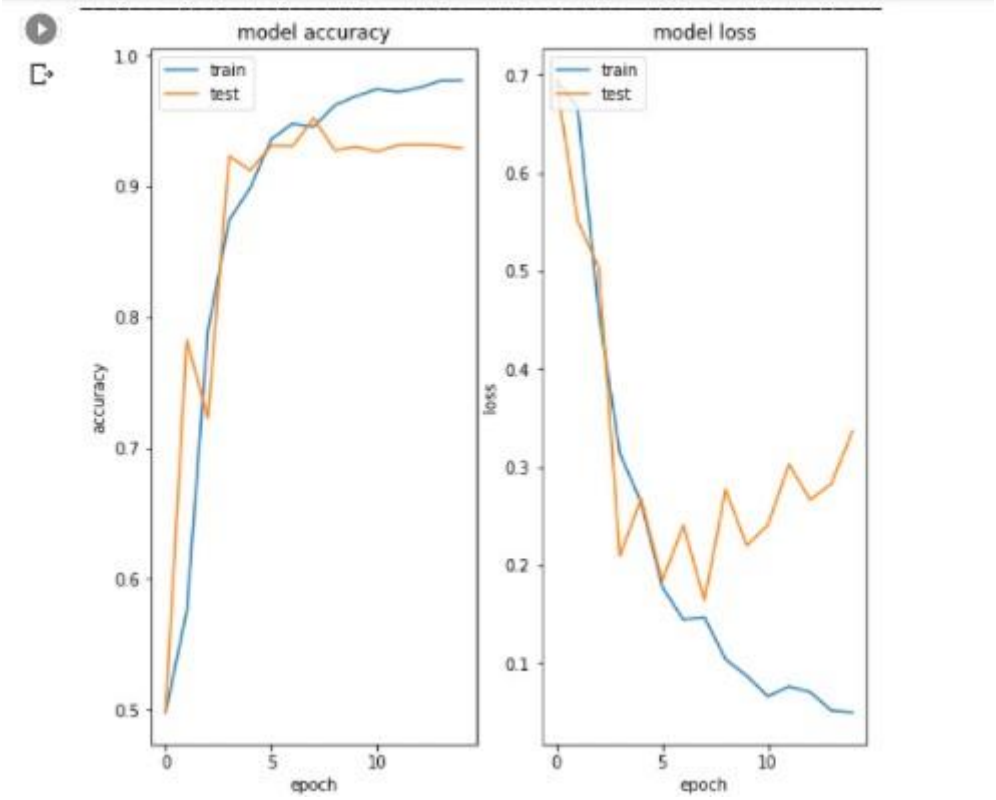


Figure 51 : Le model 2 Accuracy – Loss

### 4.2.5 Exemples de prédictions de labels

Les labels prédits :

```
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.models import load_model

LABELS = ["NEUTROPHIL","EOSINOPHIL","MONOCYTE","LYMPHOCYTE","BASOPHIL"]
LABELS2 = ["Mononuclear","Polynuclear"]

def prepare(filename):
  # load the image
  img = load_img(filename, target_size=(60, 80))
  # convert to array
  img = img_to_array(img)
  # reshape into a single sample with 3 channels
  img = img.reshape(1, 60, 80, 3)
  return img

# load an image and predict the class
def run_example():
  trainedModel = tf.keras.models.load_model("finale_model.h5")
  prediction = model.predict([prepare('Eosinophil.jpg')])
  print("La cellule de type:", LABELS [int (prediction[0][0])],"Sous le groupe :", LABELS2 [int (prediction[0][0])])

# entry point, run the example
run_example()
#["NEUTROPHIL","EOSINOPHIL","MONOCYTE","LYMPHOCYTE","BASOPHIL"]
#["Mononuclear","Polynuclear"]
```

```
La cellule de type: EOSINOPHIL Sous le groupe : Polynuclear
```

Figure 52 : Exemples de prédictions

## 4.3 Présentation de l'application

Nous allons maintenant présenter l'application de notre approche de classification des images.

### 4.3.1 Page d'accueille

La figure ci-dessous montre la page d'accueil de notre application :

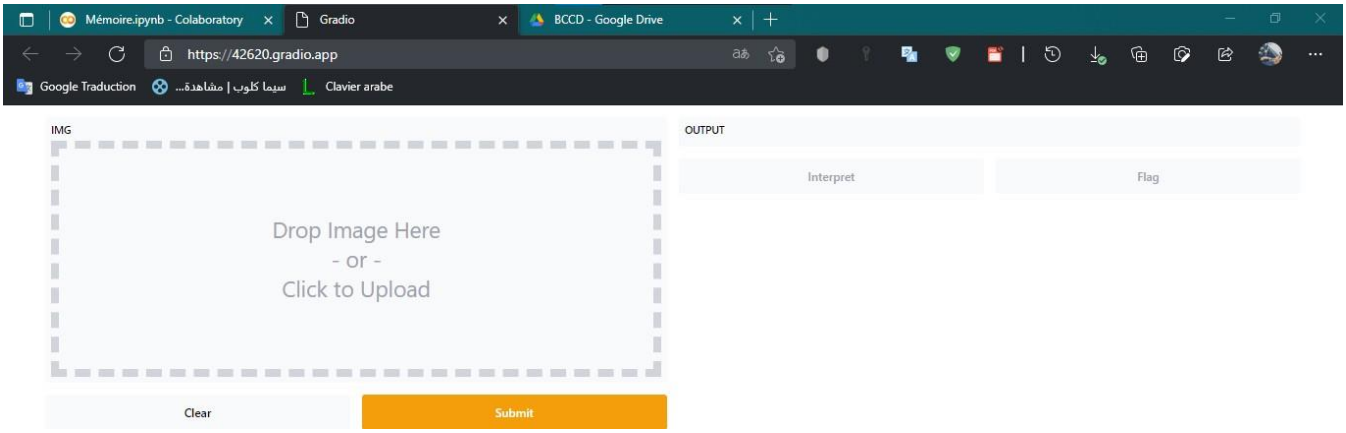


Figure 53 : Page d'accueille

## 4.3.2 Page de prédiction

Après d'importer une image, l'utilisateur envoyer une requête au serveur par cliquer sur bouton «Submit », le serveur répondre au utilisateur par une page web contienne tous les détails de prédiction, La figure ci-dessous montre la page de prédiction du notre application.

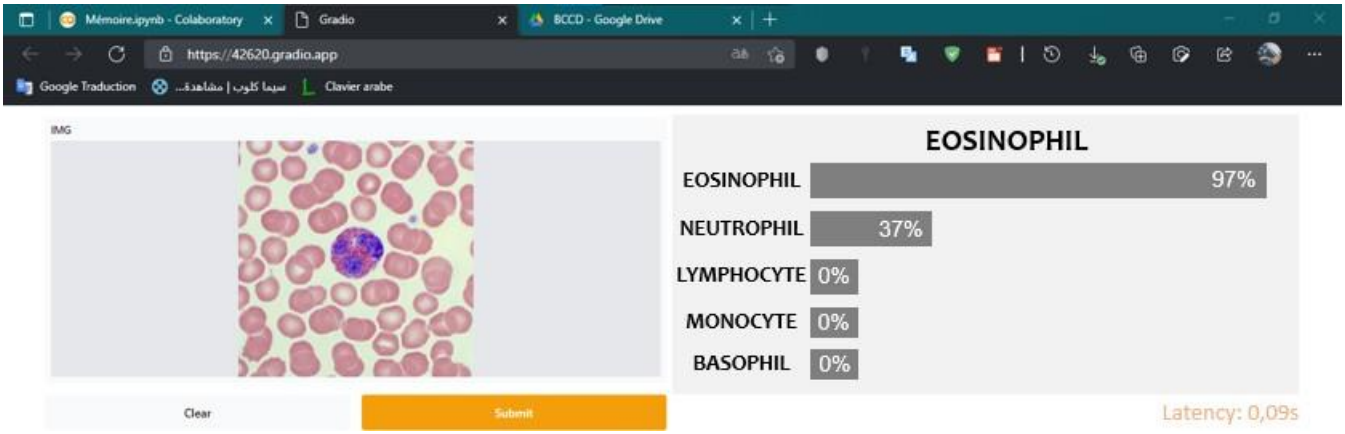


Figure 54 : Page de prédiction

## 4.4 Conclusion

Dans ce chapitre, on a vu le langage de programmation utilisé pour la réalisation du travail ainsi que l'environnement de travail et les outils. Ainsi nous avons présenté quelques bibliothèques qui ont été indispensables. Ensuite on a donné les mesures de performances obtenues par le CNN avec la signification de chacune de ces mesures (Taux de réussite, précision, spécificité …).

Enfin on a présenté notre application et leur fonctionnement.

# Conclusion générale

L’objectif principal de ce mémoire est aidé l'hématologue à classifier les globules blancs dans leurs sous-types à l'aide d'images microscopiques de cellules, en utilisant les techniques du réseau neuronal convolutionnel. L’entraînement du modèle a été effectué sur la base de données BCCD. Cette dernière est devisée en un ensemble d’apprentissage et un autre pour tester les performances du modèle.

Cette classification permet de distinguer les cellules et de vérifier le type de maladie dont souffre un patient. Les résultats obtenus à partir de cette expérience permettent d'identifier les images de manière robuste par rapport aux méthodes de laboratoire orthodoxes. Le bon niveau de précision est supérieur à 92% pour l'ensemble de test. Par conséquent, lorsque le modèle est formé avec des capacités de calcul élevées, un modèle parfait peut être formé et peut être appliqué dans l'analyse médicale et les applications traitant du nombre de globules blancs et des sous-types de globules blancs.

# Perspective

Malheureusement, le temps attribué à ce travail était très court, d’où il était difficile de fixer certains paramètres pour étudier d’autres approches et algorithmes. Nous proposons comme perspectives :

- Implémenter un autre classifieur CNN « MobileNet » pour avoir l’occasion de les comparer avec notre classifieur.
- Augmentée la base de données avec des nouvelles données pour améliorer les performances.
- Convertir notre modèle pour le déploiement en microscope optique et nous donne le résultat en temps réel.

# Références

## Bibiographie


- [MAT 1] Maton, A. : Human Biology and Health. Prentice Hall, Englewood Cliffs (1993)
- [LAF 2] LaFleur-Brooks, M. : Exploring Medical Language: A Student-Directed Approach, 7th edn, p. 398. Mosby Elsevier, St. Louis Missouri (2008). ISBN 978-0-323-04950-4
- [ALB 3] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P. : Molecular Biology of the Cell, p. 1367. Garland Science, New York (2002)
- [HUT 4] HUTIN Ed : Aspects cytologiques normaux et pathologiques des éléments du sang et des organes hématopoïétiques, Centre d'Arts Graphiques 1981


## Webographie


[WEB 1] : https:// Les cellules sanguines (cerimes.fr)

[WEB 2] : https:// Monocytose : définition - docteurclic.com

[WEB 3] : https:// IA & Marketing : la promesse de la data intelligence ! (Part 2) (fanvoice.com)

[WEB 4] : https:// Reinforcement Learning : Définition et domaines d'application (datascientest.com)

[WEB 5] : https://datafranca.org/wiki/Classification_binaire

[WEB 6] : https:// Logistic Regression pour Machine Learning - Une Introduction Simple (mrmint.fr)

[WEB 7] : https:// Méthodes d'apprentissage pour la classification multi label - Detail (utc.fr)

[WEB 8] : https:// Movie Genre Prediction Using Multi Label Classification (analyticsvidhya.com)

[WEB 9] : https:// Présentation & historique des réseaux neuronaux | Dossier (futura-sciences.com)

[WEB 10] : https:// Machine learning : comprendre les réseaux de neurones (juripredis.com)

[WEB 11] : https:// 1-Schéma d'un neurone. | Download Scientific Diagram (researchgate.net)

[WEB 12] : https:// Réseaux de neurones - Deep Learning : Biologiques ou artificiels ? (datascientest.com)

[WEB 13] : https:// What happens at the synapse between two neurons - Studdy

[WEB 14] : https:// (Tutorial) KERAS Tutorial: DEEP LEARNING in PYTHON - DataCamp

[WEB 15] : https:// Les réseaux de neurones convolutifs. (natural-solutions.eu)

[WEB 16] : https:// Convolutional neural network | Deep Learning | DataScientest

[WEB 17] : https:// CS231n Convolutional Neural Networks for Visual Recognition

[WEB 18] : https:// Transfer Learning | Formation Data Science | DataScientest.com

[WEB 19] : https:// W-Net: A CNN-based Architecture for White Blood Cells Image Classification | DeepAI

[WEB 20] : https:// Découvrez les différentes couches d'un CNN - Classez et segmentez des données visuelles - OpenClassrooms

[WEB 21] : https:// Définition | Python | Futura Tech (futura-sciences.com)

[WEB 22] : https:// What is NumPy? — NumPy v1.20 Manual

[WEB 23] : https:// Keras Definition | DeepAI

[WEB 24] : https:// What Is Pandas in Python? Everything You Need to Know (activestate.com)

[WEB 25] : https://Apprentissage automatique ou machine learning - France | IBM

[WEB 26] : https://Que signifie Apprentissage supervisé? - Definition IT de Whatis.fr (lemagit.fr)

[WEB 27] : https://Apprentissage par renforcement - Data Analytics Post

[WEB 28] : https://Gradio Library | Create Interface For Machine Learning Models with Gradio (analyticsvidhya.com)