

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR

ET DE LA RECHERCHE SCIENTIFIQUE

جامعة محمد بوضياف - المسيلة



UNIVERSITE MOHAMED BOUDIAF - M'SILA



كلية الرياضيات و الاعلام الالي

FACULTE DE MATHEMATIQUE ET L'INFORMATIQUE

قسم الاعلام الالي

DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE MASTER

THEME

TECHNIQUE D'ANALYSE AI POUR LA
PREDICTION DU COVID 19

Dirigé par :

DEBBA ALI

MOUKHTARI RABEH

Réalisée par :

KADI BADREDDINE

BEN MERZOUG WIDAD

Année universitaire : 2021/2022

Remerciements

الحمد لله الذي وهب لنا التوفيق والسداد ومنحنا الثبات
وأعاننا على تمام هذه المذكرة الشكر الأول لمن كان لهما فضلا في
وجودي وفضل تربيتي تعليمي ووصولي لما أنا عليه شكرا أمي نبع
أماني شكرا أبي مسندي وعزوتي أطال وبارك في عمر كما شكرا
لنفسي التي لم تخذلني وقت العزم والتي ثابرت تعبت اجتهدت وكم
اجتازت من صعاب بجداره وامتياز شكرا, لكل شخص مره بحياتي
وكان جزءا منها من الطفولة حتى اليوم شكرا لوجودكم كنتم
وستبقون ذكرى جميلة مرت شكرا لكل من امن بقدراتي شكرا لكل
من دعمني لكل من كان سبب بسمه رسمت على وجهه شكرا
مدرستي متوسطتي ثانويتي وجامعتي.
ألف شكر كنتم حضنا اعتر بوجودي في يوم من الأيام شكرا
لكل أستاذ درسني وكان سببا في تنويري أنار الله بصيرتك

شكرا لكل صديق عرفته و قاسمت معه الأيام شكرا لصديقاتي
اللاتي جمعتنا الدراسة وكنا خير خليلات مؤنسات شكرا للقدر الذي
جمعني برفيق المذكرة شكرا لمشرفي الذي ساندنا طوال فتره
التحضير وألف حمد وشكر على كل باسمه ذكراكم خالدة وفقكم الله
فيما تتمنون وتطمحون

Dédicace

ما أجمل أن يجود المرء بأعلى ما لديه والأجمل أن يهدي
الغالي للأعلى هي هذه ثمره جهدي أنهيتها اليوم هي هديه
أهديها لمن غادرت دنياي لروح شاركتني كل المشوار بدعائها
جدتي و لوالدي أطال الله في عمرهما إلى سندي الصحيح
إخوتي رفقاء الدرب الذين لا يكلون
وأخيرا انتهت الحكاية ورفعت قبعتي مودعتا للسنين التي
مضت

Dedication

I dedicate this work to:

Many thanks to **my father** for his morel and financiering supporting during all my life.

To the strongest woman ever, the one who has always sacrificed herself to see me succeed, to **my Mom**.

To my pearls Imen, Radhia, Ikram and my little one Amira .

To Madjid and Sami thank you and you know the rest...

To "Floukies" hadji Madjid, Anis , "the shark" Imad, "white horse" Ameer ,
"scanndar" Ala,

" GUINIH " Achraf, Fares "L'ben", Saad ,Akram,Tadj and "ELEC_FAMILY"
thanks for your smiles and challenges.

To Youssef and Bilel thanks for your encouragements.

Thanks for my 2017/2022 promo and my profs.

To Baali , boulahya , walid,radwan,salah ,rachad ,yaakoub for good moments

To "El Gasfa" thanks for supporting me and give me happiness.

To my sister Bushra "soupapa" thanks sista .

Thanks Doudi you do your best.

Thanks you all for your support and for all the moments that we had together.

Résumé

Notre époque actuelle a vécu une crise épidémique qui a balayé toutes ses régions appelée « Covid 19 », qui a fait récemment des pertes humaines catastrophiques, et cette situation que nous traversons depuis longtemps 3Des années

Nous ont poussés à faire ce travail pour prédire si oui ou non l'infection à un taux élevé et une vitesse plus rapide que la vitesse de sa propagation en utilisant les symptômes obtenus grâce à l'analyse du sang et ceci en utilisant l'apprentissage en profondeur

ملخص

عاش عصرنا الحالي أزمة وبائية اجتاحت كل إرجائه سميت بالكوفيد 19 حصدت خسائر بشرية كارثية في الآونة الأخيرة وهذا الوضع الذي مررنا به لمدت 3 سنوات

دفعنا للقيام بهذا العمل للتنبؤ بالإصابة من عدمها به بدفقه عالية وسرعة أسرع من سرعة انتشاره باستخدام الأعراض المتحصل عنها من خلال تحليل الدم وهذا باستعمال التعلم بعمق

Abstract

Our current era has lived through an epidemic crisis that has swept all parts of it called “Covid 19”, which has claimed catastrophic human losses recently, and this situation that we have been through for 3 years

We were prompted to do this work to predict whether or not infection with it at a high rate and at a speed faster than the speed of its spread using the symptoms obtained through blood analysis and this using deep learning

Table des matières

Remerciements	2
Dédicace	3
Dédicace	4
Résumé	6
Introduction général	13
1 Problématique	
2 Objectifs	

Chapitre 01 : le covid-19 et ses symptômes

1 - Introduction	15
2 -Le coronavirus et le COVID-19?	
2.1 Qu'est-ce qu'un coronavirus ?.....	16
2.2 C'est quoi le Corvidé-19 ?	16
2.3 Comment se transmet le coronavirus ?	16
2.4 Étiquette et forme	17

2.5 Les symptômes de la maladie Covid-19	17
2.6 Les personnes les plus à risque.....	18
2.7 Les moyens de prévention pour éviter la propagation de l'épidémie.....	18
Entrée publique	
3- IA	
3.1 définition	19
3.2 Historique.....	20
3.3 applications de l'intelligence artificielle	21
3.4 Future de l'intelligence artificielle.....	21
4- ML	
4.1 Définition.....	21
4.2 Historique.....	21
4.3 Les 3 type de ML	22
4.4 Catégories de ML	25
4.4.1 Machines par séparation	26
A)-SVM	
B)-Les Réseaux de Neurons	
4.4.2 Machines par modélisation.....	26
5-DL	
5.1 Introduction.....	26
Conclusion.....	27

Chapitre 2 : Deep Learning 1

1 Introduction	29
2 Histoire de Deep Learning.....	29
2.1 Définition de DL	30

2.2 ML VS DL	30
2.3 Pour quoi le deep Learning.....	32
2.4 domaines d' applications du Deep Learning.....	32
2.5 Comment ça marche ?	33
2.6 Quels sont les réseaux de neurones pour l'apprentissage en profondeur?..	34
2.7 NN	33
2.8 Principe de fonctionnement	33
2.9 architectures DL	35
1 DNN.....	35
2 CNN.....	37
3 RNN	38
4 Réseaux LSTM.....	40
3 Les fonctions d'activation	40
➤ Sigmoidé	
➤ Soft max	
➤ Relu	
➤ Maxout	
Conclusion.....	42

Chapitre 3 : dataset

1-La base de données utilisée.....	42
2- Les paramètres des modèles de deep-Learning.....	44
Conclusion.....	45

Chapitre 4 : résultat et discussion

1-introduction.....	48
2-Environnement matériel.....	48

3- Environnement matériel (Machine utilisé)

3.1-langage de matériel (python)	49
3. 2 - Les outils et les librairies utilisés	50
3.2.1-numpy(python)	50
3.2.2-tensor flow(python)	50
3.2.3-keras.....	51
3.2.4-keras&tensorflow.....	52
3.2.5-conda.....	53
4-Les métriques utilisées	54
5-L'architecture de l'application	56
6-Notre travaille.....	58
Conclusion générale.....	72
Bibliographie.....	74

Liste de figure :

Figure	Titre	page
Figure n 1	Une image montrant la forme du covid19	17
Figure n 2	Une image montrant les 3 modes de ML	22
Figure n 3	la technique de classification est la régression	24
Figure n 4	chima de l'Apprentissage par renforcement	25
Figure n 5	La relation entre l'intelligence artificielle, le ML et le deep Learning	32
Figure n 6	Topologie de réseau de neurones profond	36
Figure n 7	DNN	37
Figure n 8	RNN	40
Figure n 9	Logo Python	46
Figure n 10	Logo numpy	47

Figure n 11,12	Logo keras Logo tensorflow	49
Figure n 13	Logo conda	51
Figure 14	Taux de classification illustré dans la matrice de confusion	36
Figure 15	Taux de classification illustré dans la matrice de confusion	36
Figure16	CNN	40

Liste des abréviations :

IA : intelligence artificielle

ML : machine learning

DL : deep learning

MERS : *Middle East respiratory syndrome*

GPS : général problème solver

FAIR : facebook artificial intelligence recherche

MMC : modèles de Markov cachés (Chaines de Markov cachées)

MMG : Modèle de mélange gaussien

ConvNet : CNN (Réseaux Neurones Convolutionnels)

RNN : réseaux de Neurones récurrents

MLP : multi layer perceptron

SVM : Machines à Vecteurs de Support

OS :System exploitation

OMS :organisation mondial de la santé

ADN :

Introduction Générale :

La rapidité d'affichage des résultats d'un test est la chose la plus importante que les gens recherchent dans divers domaines quotidiens, alors comment, s'il s'agit de leur santé physique, l'un des principaux défis en médecine est de détecter une maladie dans une courte période de temps à travers ses symptômes uniquement au lieu d'analyses chimiques et d'échantillons physiques, ce qui entraîne un coût élevé financièrement et même psychologiquement pour la maladie en raison de la longue attente. Comment si cette maladie est une épidémie mondiale si elle se jette sur une personne qui se propage en quelques jours et l'exemple le plus frappant que connaît notre monde actuel est le virus covid10

Maladie à coronavirus 2019 ou simplement COVID-19, également connue sous le nom de maladie respiratoire aiguë émergente liée au coronavirus. Le virus est très proche du virus du SRAS, qui a été détecté pour la première fois dans la ville chinoise de Wuhan en 2019. En raison de l'énorme économie chinoise, cela a entraîné un déplacement de sa population vers des pays du monde entier. En conséquence, des personnes dans le monde ont été infectées et des millions de personnes ont perdu la vie à cause de ce virus.

La nécessité de défier ce virus émergent commence par la facilité de détection de l'infection avec lui aux coûts et moyens les plus bas et dans les plus brefs délais avec le développement de l'AI dans divers domaines. Il a également contribué au domaine médical et dans ce mémoire. L'objectif principal de notre travail est de détecter COVID-19 sur symptômes uniquement utilisant les techniques DL.

Ce mémoire comprend trois chapitres : Dans le premier chapitre, nous avons parlé sur le corona et leurs symptômes avec une partie un peu générale sur le IA, ML et DL.

Dans le chapitre 2, nous présentons DL point par point. Dans le troisième chapitre, nous présentons les résultats de ce travail.

Chapitre 01

Le covid et ses symptômes

1 -Introduction

Tout le monde En 2019 est confronté à une urgence sanitaire en raison de l'émergence récente du coronavirus (COVID-19), À propos 196 pays sont touchés par cette vu ric tandis que les États-Unis d'Amérique, l'Italie, la Chine, l'Espagne, l'Iran et la France ont le plus grand nombre de cas actifs de Covid19, Les cas et les services médicaux et de santé sont confrontés à des retards dans la détection du COVID-19

Certain systèmes basés sur l IA sont conçus pour détecter automatiquement le covid-19 à l'aide de symptômes médicaux

2- COVID-19

2.1 Qu'est-ce qu'un coronavirus ?

Les coronavirus font partie d'une famille de virus variés susceptibles d'être à l'origine d'un large éventail de maladies. Leur nom signifie « **virus en couronne** » et vient du fait qu'ils possèdent tous un aspect en forme de couronne lorsqu'ils sont observés au microscope.

Ces virus à ARN peuvent infecter aussi bien l'homme que l'animal et possèdent un taux de mutation élevé à l'instar de celui de la **grippe** ou du virus **VIIH**. Les coronavirus sont aussi zoonotiques, c'est-à-dire qu'ils se transmettent de l'animal à l'homme. Des enquêtes détaillées ont révélé que le SARS-CoV se transmettait de la civette, un petit mammifère, à l'homme et le MERS-CoV du dromadaire à l'homme. On connaît plusieurs coronavirus qui circulent chez certains animaux, mais qui n'ont pas encore infecté l'homme.

Chez l'homme, les manifestations de l'infection vont du simple rhume à une infection pulmonaire sévère, responsable d'une détresse respiratoire aiguë. Les coronavirus ont été identifiés pour la première fois chez l'humain dans les années 1960.

Depuis trois coronavirus ont entraîné des épidémies graves :

- le SRAS-CoV (pour syndrome respiratoire aigu sévère), responsable d'une épidémie mondiale entre novembre 2002 et juillet 2003,
- le MERS-CoV (pour *Middle East respiratory syndrome*), découvert pour la première fois en 2012 au Moyen-Orient.
- le SRAS-CoV-2, à l'origine responsable de la première pandémie liée à un coronavirus en 2020. [23]

2.2 C'est quoi le Covid-19 ?

Le **Covid-19** est une maladie infectieuse causée par un nouveau coronavirus découvert dans la ville de Wuhan en Chine. En effet, l'Organisation mondiale de la santé (OMS) a été informée de plusieurs cas de pneumonies de cause inconnue dans cette région, le 31 décembre 2019.

Le **temps d'incubation**, durée entre l'exposition au virus et la manifestation des premiers symptômes, est estimé entre 2 et 14 jours après un contact à risque. Pendant cette période le sujet peut être contagieux. En moyenne, la maladie se déclare entre le 3^e et 7^e jour.

Comme pour beaucoup de maladies infectieuses, les personnes âgées ou présentant des pathologies chroniques présentent un risque plus élevé de morbidité. [23]

2.3 Comment se transmet le coronavirus ?

Les coronavirus se transmettent d'homme à homme lors de contacts rapprochés dans un espace confiné à moins d'un mètre et/ou pendant plus de 15 minutes.

Il existe deux modes de transmission :

- par **voie aérienne** : les postillons propagés lors d'une toux, d'un éternuement ou d'une discussion,
- par **contact physique** : comme une poignée de main ou un baiser. Ou par de biais de surfaces contaminées (poignées de porte, claviers, rampe d'escalier...). [23]

2.4 forme est Étiquette :

Le motte «**coronavirus**» est dérivé de (**latin: corona**) et (**grec:(κορώνη)** («**korónē**»), qui signifie couronne ou couronne), et signifie également couronne ou halo , Le nom fait référence à l'apparence caractéristique des virions (la forme infectieuse du virus) qui apparaît au microscope électronique, car ils ont un feuillage / villosités de grandes saillies de surface bulbeuses, les montrant comme une couronne de roi ou une couronne solaire (Figure1).

Cette formation se produit via les fibrinogènes de l'épine virale (S), qui sont des protéines qui remplissent la surface du virus et déterminent le tropisme d'un hôte

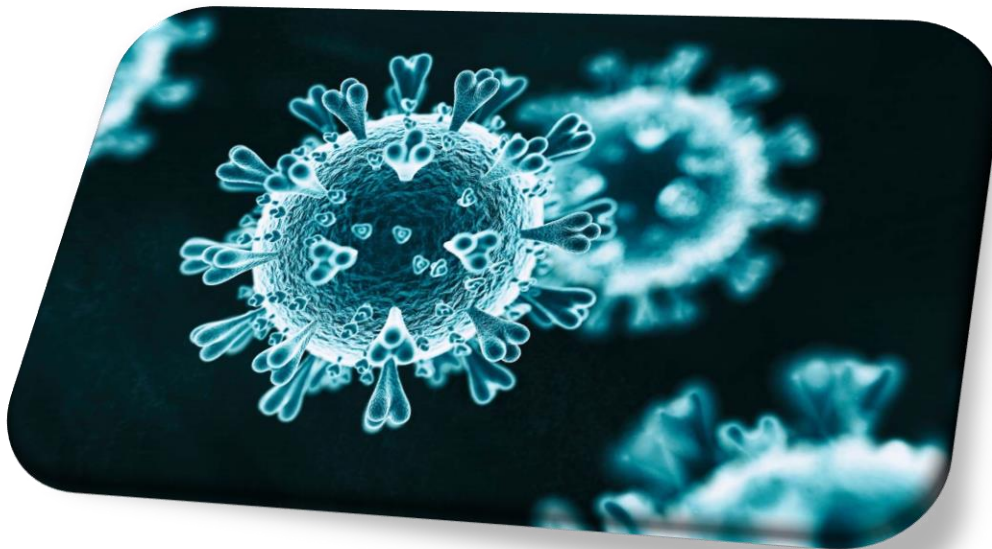


Figure 1: Une image montrant la forme du covid19[1]

2.5 - Les symptômes de Covid-19

Les signes courants d'infection sont :

- la fièvre
- la toux
- le nez qui coule
- une fatigue intense
- des douleurs musculaires inhabituelles,
- une sensation d'oppression ou d'essoufflement

- des difficultés respiratoires.

Dans les cas plus graves, l'infection peut provoquer une pneumonie, un syndrome respiratoire aigu sévère, une insuffisance rénale et même la mort [23]
Ces symptômes sont généralement bénins, Mais environ une personne sur six présente des symptômes plus sévères [22]

La maladie reste bénigne dans 80 % des cas et les chercheurs estiment le taux global de mortalité du virus à environ 3,2 %. Certains patients présentent également une forme asymptomatique du virus : ils sont porteurs du virus, mais ne présentent pas de symptômes. [23]

En cas de fièvre et de signes d'infection respiratoire :

- contactez le **Samu (15)** en faisant état de vos symptômes,
- ne vous rendez pas chez votre médecin traitant ou aux urgences, pour éviter toute potentielle contamination [23]

2.6 Les personnes les plus à risque

Les personnes les plus à risque sont les personnes âgées et présentant d'autres problèmes de santé comme l'hypertension artérielle, des problèmes cardiaques et du diabète. Le taux de mortalité augmente avec l'âge : il est de 0,2 % pour les personnes les plus jeunes (de 10 à 39 ans), mais atteint 14,8 % pour les personnes âgées de 80 ans et plus. Le taux de mortalité moyen est évalué entre 1 et 3 %. [22]

2.7 Les moyens de prévention pour éviter la propagation de l'épidémie

Selon l'OMS, les moyens de prévention efficaces pour ne pas contracter le Covid-19, mais aussi pour éviter sa propagation, sont :

- le lavage fréquent des mains au savon ou avec une solution hydroalcoolique
- éviter les contacts rapprochés, comme faire la bise ou serrer la main, avec des personnes qui toussent ou éternuent

- se couvrir la bouche avec le pli du coude, ou un mouchoir jetable, lors d'une toux ou d'un éternuement
- ne pas se toucher les yeux, le nez ou la bouche
- en cas de symptômes respiratoires et de fièvre, portez un masque et restez confinés pour ne pas contaminer votre entourage. Appelez votre médecin et suivez ses instructions. [22]

Entrée publique

● Les algorithmes

Un algorithme est une suite d'instructions permettant d'obtenir un résultat.

3- IA

3.1 Définition IA

C'est une discipline scientifique relative au traitement des connaissances et au raisonnement dans le but de permettre à une machine d'exécuter des fonctions normalement associées à l'être humain.

La norme ISO 2382-28 définit l'intelligence artificielle comme la « capacité d'une unité fonctionnelle à exécuter des fonctions généralement associées à l'intelligence humaine, telles que le raisonnement et l'apprentissage »

« La construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique » Marvin Minsky

Le but d'IA

in basic termes, the goal of using AI Is to maker computers think as humans do ,This may seem like something new, but the field was born in the 1950s

3.2 Historique de IA

Il est important de jeter un coup d'œil à l'histoire de l'AI pour pouvoir en apprendre davantage sur le cycle de vie de cette révolution :

- 1950 : Alan M. Turing, mathématicien et théoricien précurseur de l'informatique, lance le concept d'intelligence artificielle
- 1955-1956: Lancement du premier programme d'intelligence artificielle par Allen Newell, John C. Shaw et Herbert A. Simon, Logic Theorist
- 1957: Modélisation des jeux d'échec
- 1958: John McCarthy invente le Lisp (list processing), langage de programmation interactif (développement au MIT)
- 1958: Construction du premier réseau neuronal, le Perceptron, de Frank Rosenblatt, machine dite connexionniste
- 1959: Élaboration du premier GPS (général problem solver) -fin de la première période de l'intelligence artificielle.
- 1970: Néoconnexionnisme
- 1989: DeepThought, supercalculateur d'IBM, deux millions de coups par seconde
- 1990 - 1997: Développement de Deep Blue rebaptisé DeeperBlue: conception d'un système de 256 processeurs fonctionnant en parallèle, chaque processeur peut calculer environ trois millions de coups par seconde.
- 2009: Le MIT a lancé un projet visant à repenser la recherche en intelligence artificielle
- 2011: Watson, le superordinateur d'IBM remporte deux des trois manches du jeu télévisé Jeopardy! La performance a consisté pour cette intelligence artificielle à répondre à des questions de culture générale
- 2013: Humain Brain Project. Google ouvre un laboratoire de recherches dans les locaux de la NASA
- 2014: DeepKnowledge Ventures : nomme à son conseil d'administration VITAL, un algorithme capable d'élaborer ses décisions en analysant les bilans comptables des entreprises potentiellement intéressantes, le test clinique, la propriété intellectuelle et les précédents investissements

-2015: Facebook Artificial Intelligence Research (FAIR). Google rend sa technologie d'intelligence artificielle TensorFlow accessible à tous. Développement d'une crainte que l'intelligence artificielle dépasse à terme les performances de l'intelligence humaine

-2016: Amelia d'IPSoft un agent virtuel. Aussi, AlphaGo bat trois fois consécutives le champion du monde de jeu de go, Lee Se-Dol en cinq manches.

3.3 les applications de IA :

- | | |
|---|---|
| <ul style="list-style-type: none">-Les moteurs de recherche- Les moteurs de recommandation-La traduction automatique-Les assistants personnels <p>(Siri, Cortana, Google Now...)
vidéo</p> | <ul style="list-style-type: none">- Les agents conversationnels- Les véhicules autonomes- Les systèmes de navigation GPS- Les finances - La médecine
- Le cyber sécurité - Les jeux |
|---|---|

3.4 Future de IA:

au vu des recherches qui sont menées dans le monde entier, elle a encore de beaux jours devant elle, Les applications de IA pour le futur peuvent aller encore plus loin notamment dans le domaine de : robotique, bâtiments connectés, voitures autonomes et la médecine.

4-ML

4.1 définition

l'étude scientifique des algorithmes et des modèles statistiques que les ordinateurs utilisent pour accomplir une tâche sans instruction explicite, mais plutôt en s'appuyant sur des motifs et de l'inférence

L'apprentissage automatique est une technique dans laquelle vous entraînez le système à résoudre un problème au lieu de programmer explicitement les règles

4.2 historique

le Machine Learning (ML) a émergé dans la seconde moitié du XXème siècle du domaine de l'intelligence artificielle et correspond à l'élaboration

d'algorithmes capables d'accumuler de la connaissance et de l'intelligence à partir d'expériences, sans être humainement guidés au cours de leur apprentissage, ni explicitement programmés pour gérer telle ou telle expérience ou donnée spécifique [16]

4.3 les modes de ML : son 3 modes d` apprentissage

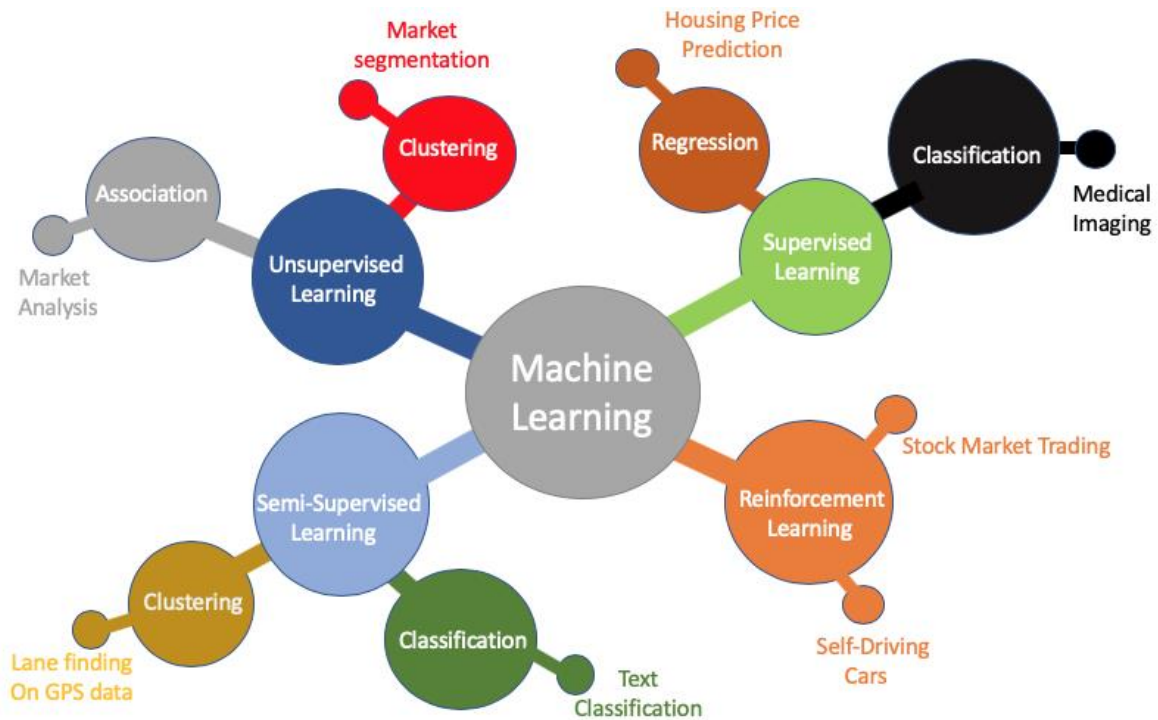


Figure2: Une image montrant les 3 modes de ML [21]

A)-Apprentissage supervisé:

C'est ce qu'on appelle l'apprentissage supervisé, car le processus d'un algorithme tiré de l'ensemble de données de formation (training set) peut être considéré comme un enseignant supervisant le processus d'apprentissage. Nous connaissons les réponses correctes, l'algorithme effectue des prédictions itératives sur les données d'apprentissage et est corrigé par l'enseignant. L'apprentissage s'arrête lorsque l'algorithme atteint un niveau de performance acceptable. [11]

La majorité des apprentissages automatiques utilisent un apprentissage supervisé :

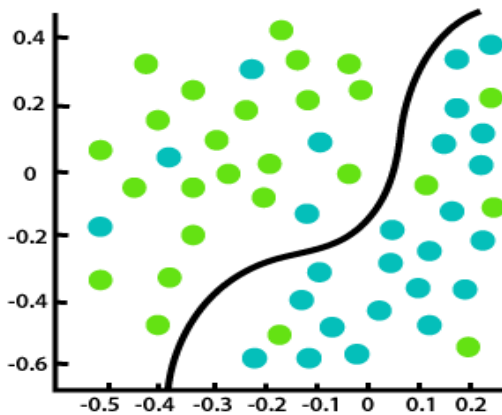
- consiste en des variables d'entrée (x) et une variable de sortie (Y)

● Vous utilisez un algorithme pour apprendre la fonction de mapping de l'entrée à la sortie

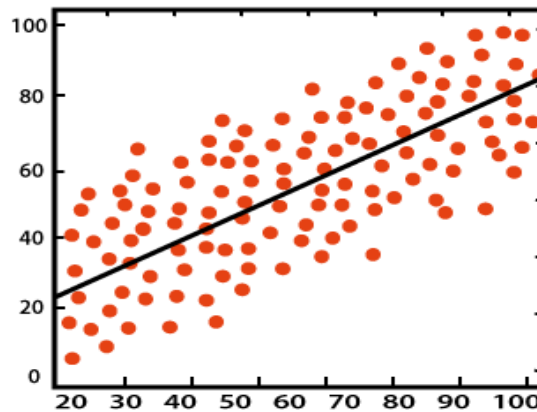
● $Y = f(X)$

● Le but est d'appréhender si bien la fonction mapping que, lorsque vous avez de nouvelles données d'entrée (x), vous pouvez prédire les variables de sortie (Y) pour ces données (apprendre une fonction de prédiction à partir d'exemples annotés)

Elle a 2 taches :



Classification



Regression

Figure3 : la technique de classification est la regression [17]

Avantages :

- Il faut beaucoup de données
- Les données doivent être annotées

inconvénients

- Très efficace quand on a suffisamment de données
- Plus performant que l'homme pour certains problèmes

B)-Apprentissage non supervisé :

On l'appelle apprentissage non supervisé car, contrairement à l'apprentissage supervisé ci-dessus, il n'y a pas de réponse correcte ni d'enseignant.

Les algorithmes sont laissés à leurs propres mécanismes pour découvrir et présenter la structure intéressante des données.

L'apprentissage non supervisé comprend deux catégories d'algorithmes :

*Algorithmes de regroupement * et d'association. [12]

C)- Apprentissage par renforcement

L'apprentissage par renforcement consiste, pour un agent autonome, à apprendre les actions à prendre, à partir d'expériences, de façon à optimiser une récompense au cours du temps.

➤ imiter la façon dont les animaux ou les humains apprennent

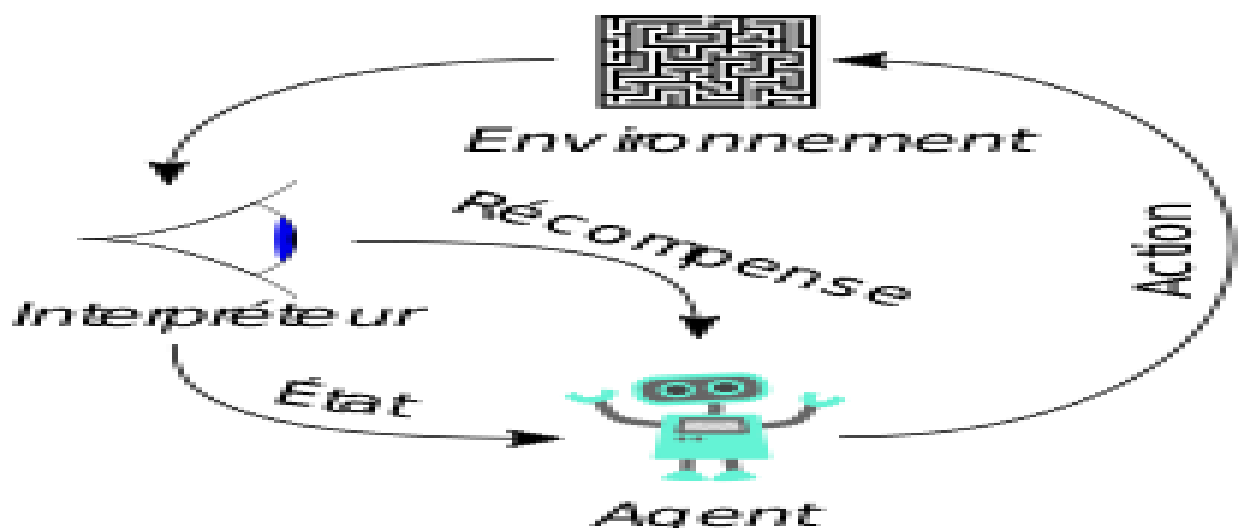


Figure4 :le chima de l' Apprentissage par renforcement [18]

4.4 categories de ML

4.4.1 Machines par séparation :

A)-SVM (Machines à Vecteurs de Support) :

est un algorithme d'apprentissage automatique supervisé relativement simple utilisé pour la classification et/ou la régression. Il est plus préféré pour la classification mais est parfois très utile pour la régression également.

Fondamentalement, SVM trouve un hyper-plan qui crée une frontière entre les

types de données. Dans l'espace à 2 dimensions, cet hyperplan n'est rien d'autre qu'une ligne.

dans SVM, nous traçons chaque élément de données dans l'ensemble de données dans un espace à N dimensions, où N est le nombre de caractéristiques/attributs dans les données. Ensuite, trouvez l'hyperplan optimal pour séparer les données. Donc, par là, vous devez avoir compris que, par nature, SVM ne peut effectuer qu'une classification binaire (c'est-à-dire choisir entre deux classes)[19]

B)-Les Réseaux de Neurones :

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)[20]

***types de réseaux neurones :** Il existe beaucoup de types chaque type étant développé pour un objectif particulier[5].

- Neurone Formel
- Neurones multicouche
- Neurones récurrents
- Réseaux de Hopfield
- ConvNet

4.4.2 Machines par modélisation

A)- MMG :

Un modèle de mélange gaussien (MMG) est une catégorie de modèle probabiliste qui stipule que tous les points de données générés sont dérivés d'un mélange de distributions gaussiennes finies sans paramètres connus. Les paramètres des modèles de mélange gaussien sont dérivés d'une estimation maximum a posteriori ou d'un algorithme itératif d'anticipation maximisation à partir d'un modèle antérieur bien entraîné. Les modèles de mélange gaussien sont très utiles pour modéliser des données, en particulier des données provenant de plusieurs groupes.

Elle utilisés dans les systèmes biométriques, où le modèle paramétrique aide à comprendre les caractéristiques ou les mesures associées à des caractéristiques telles que les caractéristiques spectrales du tractus vocal, sont également utilisés pour l'estimation de la densité et sont considérés comme les techniques les plus matures sur le plan statistique pour la classification.

B)- MMC :

sont des modèles statistiques, riches et largement utilisés en traitement du signal.

Ils sont développés par Andrew Markov (étudiant de Tchebychev), et ils sont premièrement, orientés vers des objectifs linguistiques dans des travaux de littérature Russe.

Ceux sont des outils efficaces en modélisation des données séquentielles ou 'time-series Data'.

Utilisés par la suite dans des problèmes de reconnaissance de la parole par Baker, leur théorie de base est introduite par Baum et ses collègues à la fin des années soixante est Actuellement, ces modèles sont, de plus en plus adoptés en reconnaissance automatique de la parole, pour l'analyse des séquences ADN, et dans des problèmes liés à l'écriture et le traitement de texte. Aussi, leur utilisation en visionique est vaste. Ils sont implémentés en segmentation d'images, reconnaissance des visages, interprétation des gestes, ainsi que la modélisation de l'arrière-plan et récemment en traitement vidéo

5-DL

5-1 introduction

les problèmes de prédiction deviennent plus difficiles lorsque vous utilisez différents types de données en entrée

Un autre nom pour les données d'entrée est fonctionnalité, et l'ingénierie des fonctionnalités est le processus d'extraction de fonctionnalités à partir de données brutes. Lorsque vous traitez différents types de données, vous devez trouver des moyens de représenter ces données afin d'en extraire des informations significatives.

Un exemple de technique d'ingénierie de caractéristiques est la lemmatisation, dans laquelle vous supprimez l'inflexion des mots dans une phrase.

Par exemple, les formes fléchies du verbe "regarder", comme "regarder", "regarder" et "regarder", seraient réduites à leur lemme, ou forme de base : "regarder"

si vous utilisez des tableaux pour stocker chaque mot d'un corpus, alors en appliquant la lemmatisation, vous vous retrouvez avec une matrice moins clairsemée. Cela peut augmenter les performances de certains algorithmes d'apprentissage automatique.

Conclusion

Le COVID-19 est difficile à combattre en raison de son comportement mystérieux et de sa biologie inconnue. Nous pouvons essayer des mesures de précaution et des leçons tirées d'autres flambées de santé publique comme le SRAS-CoV et Corona virus. Masquer la sensibilisation, la distanciation sociale, l'isolement, l'hygiène et la quarantaine peuvent réduire les opportunités de la propagation de l'épidémie.

Chapitre 2

Deep Learning

1 -Introduction

L'apprentissage en profondeur est une branche de l'apprentissage automatique qui repose entièrement sur des réseaux de neurones artificiels, car les réseaux de neurones simulent le cerveau humain, de sorte que DL est également une sorte de simulation du cerveau humain.

Dans l'apprentissage en profondeur, nous n'avons pas besoin de tout programmer explicitement. Le concept d'apprentissage en profondeur n'est pas nouveau. Il existe depuis plusieurs années. C'est un sujet à la mode ces jours-ci parce que nous n'avions pas beaucoup de puissance de traitement et beaucoup de données auparavant. Comme au cours des 20 dernières années, la puissance de traitement s'est considérablement améliorée et dans ce chapitre, nous allons d'abord introduire des concepts liés à DL

2 -Histoire de Deep Learning

Depuis 2012, les algorithmes à base de deep learning (apprentissage profond) semblent prêts à résoudre bien des problèmes : reconnaître des visages comme le propose DeepFace, vaincre des joueurs de go ou de poker ou bientôt permettre la conduite de voitures autonomes ou encore la recherche de cellules cancéreuses. Pourtant, les fondements de ces méthodes ne sont pas si récents : le deep learning a été formalisé en 2007 à partir des nouvelles architectures de réseaux de neurones dont les précurseurs sont McCulloch et Pitts en 1943.

Suivront de nombreux développements comme le perceptron, les réseaux de neurones convolutifs de Yann Le Cun et Yoshua Bengio en 1998 et les réseaux de neurones profonds qui en découlent en 2012 et ouvrent la voie à de nombreux champs d'application comme la vision, le traitement du langage ou la reconnaissance de la parole.

Pourquoi maintenant ?

parce que ces nouvelles techniques de machine learning profitent de données massives (big data) que l'on est désormais capable d'analyser ainsi que de capacités de calcul phénoménales notamment grâce aux processeurs graphiques. Preuve que chaque domaine irrigue les autres, c'est pour pouvoir utiliser les immenses promesses du deep learning que Google a mis au point les accélérateurs TPU. [3]

2.1 Définition de DL

L'apprentissage profond (« Deep Learning ») est un ensemble de techniques d'apprentissage automatique qui a permis des avancées importantes en intelligence. Dans l'apprentissage automatique, un programme analyse un ensemble de données afin de tirer des règles qui permettront de tirer des conclusions sur de nouvelles données. L'apprentissage profond est basé sur ce qui a été appelé, par analogie, des « réseaux de neurones artificiels », composés de milliers d'unités (les « neurones ») qui effectuent chacune de petites opérations simples.

Les résultats d'une première couche de « neurones » servent d'entrée aux calculs d'une deuxième couche et ainsi de suite. [5]

L'apprentissage en profondeur est une technique dans laquelle vous laissez le réseau de neurones déterminer par lui-même quelles fonctionnalités sont importantes au lieu d'appliquer des techniques d'ingénierie de fonctionnalités. Cela signifie qu'avec l'apprentissage en profondeur, vous pouvez contourner le processus d'ingénierie des fonctionnalités.

2.2 Machine learning vs Deep learning

Beaucoup de gens s'inquiètent de l'IA parce qu'ils ne peuvent pas faire confiance aux ordinateurs existants pour apprendre et prendre des décisions. L'apprentissage automatique ML et l'apprentissage profond DL sont deux des concepts les plus importants qui rendent l'intelligence artificielle possible. Les deux termes sont souvent confondus, même s'ils font référence à deux méthodes complètement différentes utilisées dans des domaines d'application différents.

Le ML et le DL font tous deux partie de l'intelligence artificielle, car les deux méthodes poussent les ordinateurs à prendre des décisions intelligentes. L'apprentissage en profondeur est une sous-catégorie de l'apprentissage automatique, comme le montre la figure ci-dessous.

L'utilisation de ML et DL est déjà dans de nombreux domaines et conditions où les machines peuvent prendre des décisions intelligentes similaires à celles prises par les humains, les deux techniques nécessitent certainement beaucoup de données, elles servent de base à l'apprentissage, et les similitudes s'arrêtent là [6]

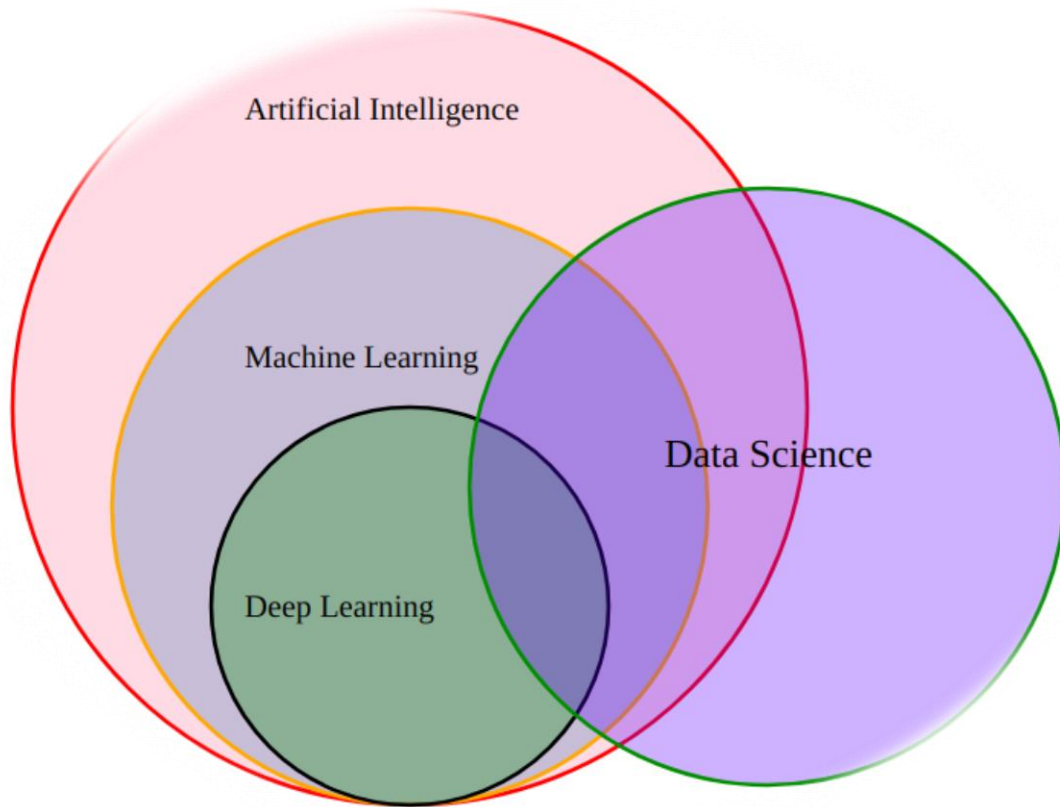


Figure5: La relation entre l'intelligence artificielle, le ML et le sdeep learning[4].

2.3 Pour quoi le deep learning

خوارزميات تعلم الآلة ليست مفيدة اثناء العمل مع البيانات ذات الابعاد العالية حيث يكون لدينا عدد كبير من المدخلات والمخرجات اما التعلم العميق فهو كفاءة عالية في هذه الحالة

2.4 Domaines d'application de DL

La technologie de DL est l'une des techniques les plus utilisées dans de nombreux domaines comme :

- 1- La reconnaissance faciale
- 2 -Le traitement automatique de langage naturel
- 3 -Voitures autonomes
- 4 -Recherche vocale et assistants à commande vocale
- 5 -Traduction automatique :
 - Traduction automatique de texte

- Traduction automatique d'images

6 -Recherche en marketing

2.5 Comment ça marche

Pour comprendre comment fonctionne le deep learning, nous allons utiliser un exemple concret de reconnaissance faciale.

par exemple

notre but soit qu'il reconnaisse une photo contenant un ballon . Pour pouvoir reconnaître un ballon , l'algorithme doit d'abord savoir distinguer tous les types de ballons existantes, mais aussi identifier précisément le ballon et le chemin il est autonome quel que soit l'angle auquel il se trouve

Pour ce faire, c'est simple : le réseau de neurones artificiels s'entraîne en analysant des milliers d'images de ballons et apprend à les reconnaître sur des photos d'autres objets , Ces données sont ensuite affectées à différentes informations, permettant à l'algorithme intelligent de déduire s'il y a des boules dans les images qu'il analyse Le réseau artificiel va également comparer cette réponse aux bonnes réponses indiquées par les humains

Si il a vu juste ,l'algorithme de reconnaissance garde cette réussite en mémoire et s'en resservira plus tard pour reconnaître des ballons

Contrairement , s'il s'est trompé : il en prend note et corrige son erreur de lui-même la fois suivante

C'est en répétant ce système d'entraînement des milliers de fois que le réseau de neurones finit par être capable de reconnaître un ballon dans toutes circonstances (avec un degré de réussite proportionnel à la durée d'entraînement du réseau et au nombre de couches qu'il possède)

Cette technique d'apprentissage est appelée **supervised learning** [8]

2.6 Quels sont les réseaux de neurones pour l'apprentissage en profondeur?

Ne pas avoir à s'occuper de l'ingénierie des fonctionnalités est une bonne chose car le processus devient plus difficile à mesure que les ensembles de données deviennent plus complexes.

Par exemple : comment extrayez-vous les données pour prédire l'humeur d'une personne à partir d'une photo de son visage ?

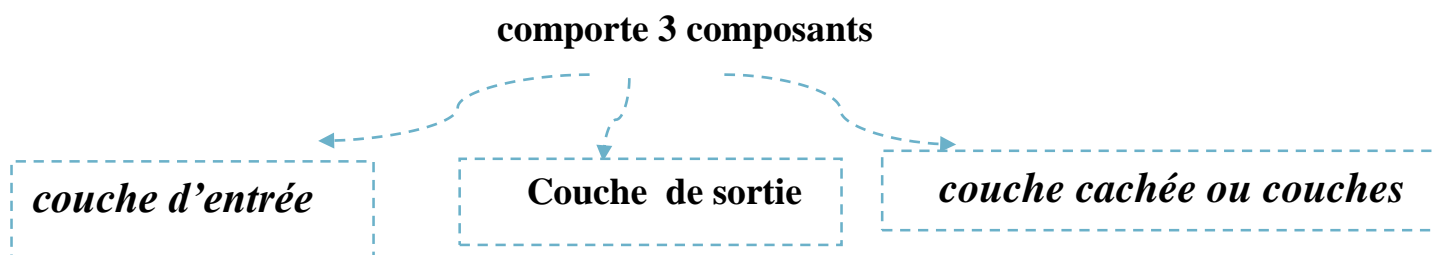


Avec les réseaux de neurones , vous n'avez pas à vous en soucier car les réseaux peuvent apprendre les fonctionnalités par eux-mêmes.

2.7 NN :

Un type d'algorithme d'apprentissage automatique avancé, connu sous le nom de réseaux de neurones artificiels, est à la base de la plupart des paradigmes d'apprentissage en profondeur, de sorte que l'apprentissage en profondeur peut parfois être appelé apprentissage neuronal profond ou réseau neuronal profond [13]

2.8 Principe de fonctionnement :



Le terme « profond » se rapporte généralement au nombre de couches cachées du réseau de neurones.

Les réseaux de neurones classiques ne comportent que 2 à 3 couches cachées, tandis que les réseaux profonds peuvent en compter jusqu'à 150 L'idée est d'utiliser la structure de couche de réseau neuronal en empilant plusieurs couches les unes sur les autres, de manière à faciliter le mécanisme de décomposition.

Par conséquent, chaque couche d'un réseau de neurones profonds (Deep Neural Networks DNN) fonctionne comme une seule transformation pour extraire davantage les données.

Le réseau de neurones le plus connu et le plus simple à comprendre est le réseau de neurones multicouches à anticipation. Il contient un calque d'entrée, un ou plusieurs calques masqués et un seul calque de sortie. Chaque couche peut avoir un nombre différent de neurones et chaque couche est entièrement connectée à la couche adjacente. [10]

Un réseau de neurones est défini comme un ensemble de nœuds (appelés neurones) connectés via des liaisons dirigées (flèche), chaque flèche représente une connexion entre la sortie d'un neurone et l'entrée d'un autre (les flèches entrantes étant les entrées du neurone et les flèches sortantes étant les sorties du neurone), Chaque flèche porte un poids (w), reflétant son importance, chaque nœud étant une unité de traitement qui exécute une fonction de nœud statique sur son signal entrant pour générer une sortie de nœud unique

Les valeurs d'entrée, ou en d'autres termes, nos données sous-jacentes, sont transmises via ce «réseau» de couches masquées jusqu'à ce qu'elles convergent vers la couche de sortie.

La couche en sortie correspond à notre prédiction : il peut s'agir d'un nœud si le modèle ne génère qu'un nombre ou de quelques nœuds s'il s'agit d'un problème de classification multi-classe.

La forme à l'intérieur des neurones dans les couches centrales représente une fonction d'activation (typiquement un $1 = (1 + e^{-x})$) qui est appliquée à la valeur du neurone avant de le transmettre à la sortie. [9]

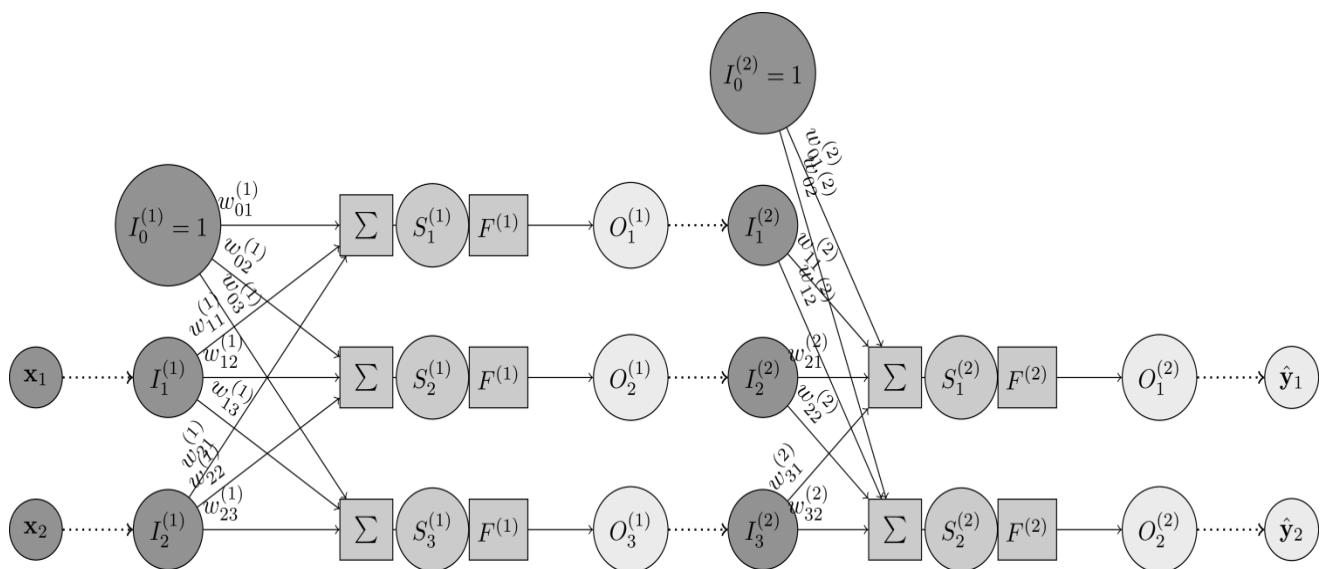


Figure6: Topologie de réseau de neurones profond. [0].

Les couches cachées d'un réseau de neurones apportent des modifications aux données pour éventuellement déterminer quelle est sa relation avec la variable cible.

Chaque nœud a un poids et multiplie sa valeur d'entrée par ce poids. Pour déterminer ce que devraient être ces petits poids, nous utilisons généralement un algorithme appelé Back propagation.

2.9 Architectures de réseaux de neurones profonds

nous examinerons les cinq plus importantes des architectures du deeplearning Dans cette section parce que Le nombre d'architectures et d'algorithmes utilisés dans l'apprentissage en profondeur est varié qui ont été l'une des méthodes les plus utilisées dans différentes domaine

donc, nous présenterons un bref aperçu des structures communes que l'on retrouve dans de nombreux réseaux profonds :

- Artificial Neural Networks (DNN)
- Convolution Neural Networks (CNN)
- Recurrent Neural Networks (RNN)

-DNN :

Les réseaux de neurones profonds (DNN) sont généralement des réseaux à alimentation directe (FFNN) dans lesquels les données circulent de la couche d'entrée à la couche de sortie sans revenir en arrière³ et les liens entre les couches sont à sens unique dans le sens aller et ils ne touchent jamais un nœud de nouveau

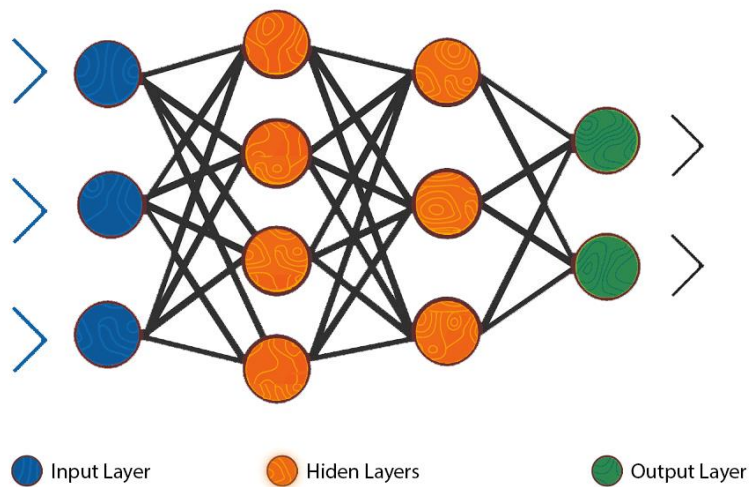


Figure 7:DNN [26].

Les sorties sont obtenues par apprentissage supervisé avec des ensembles de données de certaines informations basées sur «ce que nous voulons» par rétropropagation.

Comme vous allez au restaurant et que le chef vous donne une idée des ingrédients de votre repas.

Les FFNN fonctionnent de la même manière que vous aurez la saveur de ces ingrédients spécifiques en mangeant, mais juste après avoir terminé votre repas , vous oublierez ce que vous avez mangé. Si le chef vous donne à nouveau le repas avec les mêmes ingrédients, vous ne pouvez pas reconnaître les ingrédients,

Vous devez recommencer à zéro car vous n'en avez aucun souvenir. Mais le cerveau humain ne fonctionne pas comme ça [24]

Comment fonctionne un DNN :

En général, un Deep Neural Network DNN a une couche d'entrée, une couche de sortie et au moins une couche entre les deux. Plus le nombre de couches est élevé, plus un réseau est dit " profond ".

Chacune de ces couches effectue différents types de tri et de catégorisation spécifique dans un processus nommé " hiérarchie de caractéristique ".

Les différentes données entre la couche d'entrée et la couche de sortie doivent être traitées progressivement pour résoudre une tâche ou établir une prédiction. [25]

La première couche du réseau reçoit les données et exécute un calcul de fonction d'activation pour produire un résultat. Il peut s'agir par exemple d'une prédiction de probabilité

Ce résultat est transmis à la couche suivante de neurones. La connexion entre deux couches successives est associée à un " poids " , Ce poids définit l'influence des données sur le résultat produit par la couche suivante et éventuellement pour le résultat final.[25]

-CNN :

Next comes the Convolutional Neural Network (CNN ou ConvNet) qui est une classe de réseaux de neurones profonds qui est le plus souvent appliqué à l'analyse de l'imagerie visuelle. Leurs autres applications incluent la compréhension vidéo, la reconnaissance vocale et la compréhension du traitement du langage naturel.

En outre, LSTM combiné aux réseaux de neurones convolutifs (CNN) a amélioré le sous-titrage automatique des images comme ceux que l'on voit sur Facebook. Ainsi, vous pouvez voir que RNN ressemble plus à nous aider dans le traitement des données en prédisant notre prochaine étape alors que CNN nous aide dans l'analyse visuelle.[24]

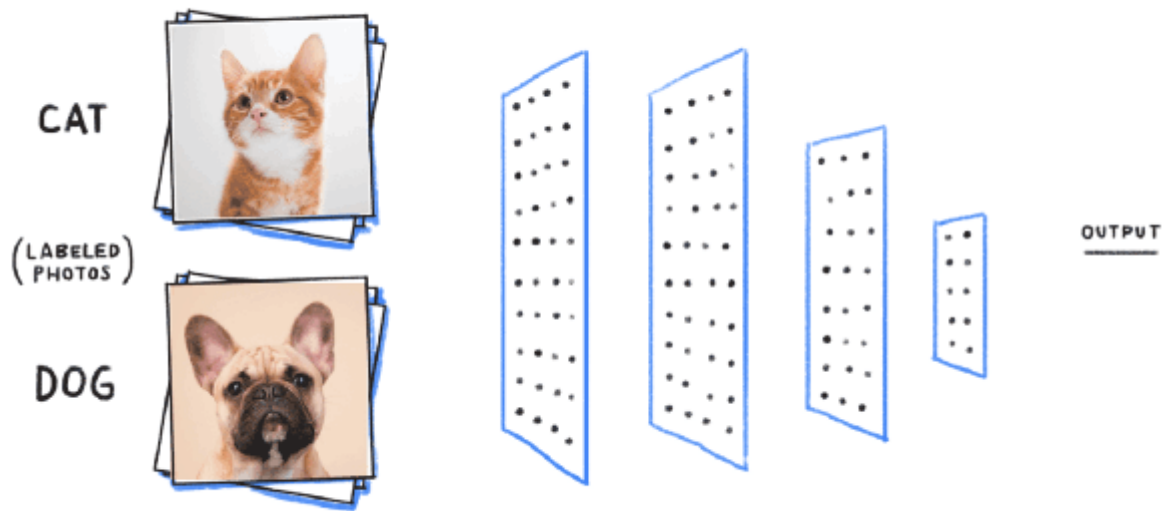


Figure 16 :CNN[26]

-RNN :

Un réseau neuronal récurrent (RNN) résout ce problème qui est un FFNN avec une torsion temporelle.

Cette réseau de neurones n'est pas sans état, a des connexions entre les passages et les connexions à travers le temps. Il s'agit d'une classe de réseaux de neurones artificiels où les connexions entre les nœuds forment un graphe orienté le long d'une séquence, comme des liens d'entités d'une couche aux couches précédentes, permettant aux informations de refluer dans les parties précédentes du réseau. Ainsi, chaque modèle dans les couches dépend du passé événements, permettant aux informations de persister.[24]

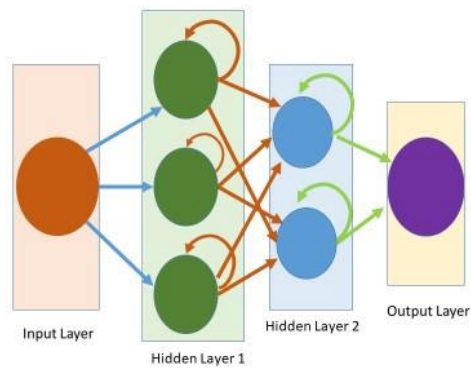


Figure8 :RNN[26]

De cette manière :: les RNN peuvent utiliser leur état interne (mémoire) pour traiter des séquences d'entrées.

Cela les rend applicables à des tâches telles que la reconnaissance d'écriture manuscrite non segmentée et connectée ou la reconnaissance vocale.

Mais ils ne fonctionnent pas seulement sur les informations que vous alimentez, mais également sur les informations connexes du passé, ce qui signifie que tout ce que vous alimentez et formez le réseau est important, comme le nourrir de « poulet », puis « œuf » peut donner un résultat différent par rapport à « œuf » puis "poulet".

Les RNN ont également des problèmes tels que la disparition (ou l'explosion) du gradient/problème de dépendance à long terme où les informations se perdent rapidement au fil du temps. En fait, c'est le poids qui se perd lorsqu'il atteint une valeur de 0 ou 1 000 000, pas le neurone. Mais dans ce cas, l'état précédent ne sera pas très informatif car c'est le poids qui stocke les informations du passé[24]

-LSTM :

Heureusement, des percées telles que la mémoire à long court terme (LSTM) n'ont pas ce problème ! Les LSTM sont un type spécial de RNN, capables d'apprendre des dépendances à long terme qui rendent RNN intelligent pour se souvenir des choses qui se sont produites dans le passé et trouver des modèles dans le temps pour donner un sens à ses prochaines suppositions.

Les LSTM ont battu des records d'amélioration de la traduction automatique, de la modélisation du langage et du traitement du langage multilingue.[24]

3- Fonction d'activation :

3.1 Définition :

Une fonction d'activation est une fonction mathématique utilisé sur un signal. Elle va reproduire le potentiel d'activation que l'on retrouve dans le domaine de la biologie du cerveau humain. Elle va permettre le passage d'information ou non de l'information si le seuil de stimulation est atteint. Concrètement, elle va avoir pour rôle de décider si on active ou non une réponse du neurone.[7]

La fonction d'activation sert avant tout à modifier de manière **non-linéaire** les données. Cette non-linéarité permet de modifier spatialement leur représentation. Dit simplement, la **fonction d'activation** permet de **changer notre manière de voir** une donnée.

Par exemple si on a comme donnée :

chaque semaine 50% des clients d'un magasin achètent des barres de chocolat; la fonction d'activation permettrait de changer la donnée en 50% des clients aiment le chocolat ou encore, 50% des clients prévoient d'acheter du chocolat chaque semaine.

Le changement de représentation peut, dans notre exemple, nous permettre d'ajuster la stratégie commercial, ou bien les stocks possédés, de notre magasin. Un modèle étant composé de multiples couches, et donc de multiples fonctions d'activation, des changements successifs et complexes de

représentation s'opèrent. Cela permet d'avoir un nouveau point de vue sur nos données que l'homme serait incapable d'avoir en peu de temps.

Il ne faut **pas confondre** fonction d'activation et loss function. La loss function est appliqué sur tous le modèle et.. est donc unique, elle permet de calculer la performance du modèle.

Au contraire la fonction d'activation est spécifique à chaque couche, elle permet de transformer les données.

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

Chaque neurone d'une couche va appliquer la fonction d'activation de la couche sur les données.

Cette transformation sera différente selon chaque neurone car chacun possède un poids différent.

À retenir : la loss function s'appliquant sur **tous le modèle**, elle comprend, elle contient la fonction d'activation. Donc lorsqu'on calcul le gradient (la dérivé de la loss function) on calcul aussi les dérivées des loss functions.[7]

Un neurone ne va faire qu'appliquer la fonction suivante :

$$X = (\text{entrée} * \text{poids}) + \text{biais}$$

(1)

Exemple pour elle :

- Sigmoïde
- Tangente hyperbolique
- ReLU (Unité linéaire rectifiée ("RectifiedLinear Unit"))
- Softmax
- Maxout

Conclusion :

Aujourd'hui DL au-delà des différentes méthodes ML en termes de performances et est largement utilisé pour une variété de tâches différentes. L'apprentissage en profondeur a amélioré la précision par rapport à d'autres approches de tâches telles que la traduction de la langue et la reconnaissance d'images.

Mais cela s'est / n'a pas pu arriver en quelques années, cela a pris des décennies[15].

Chapitre 3

dataset

1: La base de données utilisées :

On utilise un dataset extrait des bilans sanguins « FNS » de 600 patients. Les données

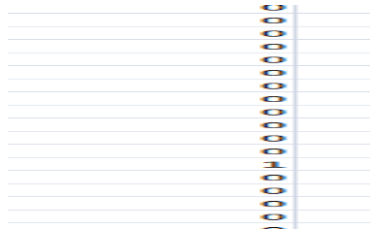
	A	B	C	D	E	F	G	H	I	J	K	L
1	Patient ID	Patient age	Hematocrit	Hemoglobin	Platelets	Red blood C	Lymphocyte	Leukocytes	Basophils	Eosinophils	Monocytes	Serum Gluco
2	126e9dd1395	17	0,23651545	-0,02234027	-0,51741302	0,10200419	0,31836575	-0,09461035	-0,22376651	1,48215818	0,35754666	-0,14064808
3	8bb9d64f021	1	-1,57168222	-0,774212	1,42966747	-0,85003501	-0,00573804	0,36455047	-0,22376651	1,01862502	0,06865151	-0,41320837
4	6c9d3323975	9	-0,74769306	-0,58624387	-0,42948034	-1,36131525	-1,11451376	-0,88492316	0,08169255	-0,66695017	1,27675891	0
5	d3ea751f3db	11	0,99183822	0,79218763	0,07299204	0,54276311	0,04543625	-0,21148765	-0,83468473	-0,70908952	-0,22024387	0
6	2c2eae16c12	9	0,19073813	-0,14765212	-0,66815472	-0,12719065	0,00279103	-1,13259172	0,3871516	-0,70908952	2,01212883	-0,83719093
7	fc41531ca4fa	13	1,0147264	0,85484386	-0,17824416	0,48987165	-0,73070687	-0,07513076	2,52536511	0,21797679	0,06865151	0
8	891d0f6449ff	14	0,7400642	0,85484386	0,36191365	0,43698105	-0,22749306	0,10575075	-0,52922559	-0,3719745	0,12117788	0
9	ebdd7c67fcb	9	-0,6790275	-0,71155578	0,95231873	-0,32112399	-0,87570053	-0,28662303	-0,22376651	-0,54053205	0,80402118	-0,41320837

M	N	O	P	Q	R	S	T
Neutrophils	Urea	Proteina C re	Creatinine	Potassium	Sodium	Alanine tran	Aspartate tra
-0,61908603	1,19805908	-0,14789495	2,08992839	-0,30578715	0,86251164	0	0
-0,12739536	-0,06730879	-0,28698576	-1,83862317	0,93002027	0,50313175	-0,58682126	-0,16219996
0,88057011	-0,81164283	0	-0,90817678	0,4356966	-0,21562791	-0,55925655	-0,5492875
0,26595679	0	-0,48767394	0	0	0	0	0
-0,42240995	-1,33267665	0	-0,90817678	-0,55294895	-0,57500774	0	0
0,65111488	-0,73720944	-0,4340246	-0,70141089	0	0	0	0
0,34790528	-0,1417422	-0,52940118	0,33241844	0	0	0	0

ces données présentent 2 classe :

Cas positif(1) : Les résultats des tests sanguins qui contiennent des valeurs indiquant La maladie

Cas négatif(0) : Les résultats des tests sanguins qui contiennent des valeurs no indiquant pas La maladie



Les données se présentent une seul répertoires, Les données d'apprentissage 70% et de test 30%. Et dans cett répertoire se trouve un sous-dossier pour chaque classe

2 : Les paramètres des modèles de deep-learning

✓ **Batch size** : La taille du lot est un terme utilisé dans l'apprentissage automatique et fait référence au nombre d'exemples de formation utilisés dans une itération.

La taille du lot peut être l'une des trois options suivantes: mode batch : où la taille du lot est égale à l'ensemble de données total, ce qui rend les valeurs d'itération et d'époque équivalentes.

✓ **Nombre d'Epoch** : Une époque est un terme utilisé dans l'apprentissage automatique et indique le nombre de passes de l'ensemble de données d'entraînement complet que l'algorithme d'apprentissage automatique a effectué.

Les ensembles de données sont généralement regroupés en lots (en particulier lorsque la quantité de données est très importante).

Certaines personnes utilisent le terme itération de manière vague et se réfèrent au passage d'un lot dans le modèle en tant qu'itération

✓ **Optimizer** : Les algorithmes d'optimisation sont utilisés dans l'apprentissage

automatique pour réduire une fonction appelée fonction de perte ou fonction d'erreur.

En minimisant la fonction de perte, les algorithmes d'optimisation peuvent atteindre

une différence minimale ou nulle entre la sortie réelle et prévue, ce qui rend notre modèle plus précis pour une tâche.

Il existe différents types d'algorithmes d'optimisation utilisés dans les réseaux de neurones : variantes de descente en pente (gradient descent) descente de gradient par lots descente de gradient stochastique descente de gradient mini-lot Gradient accéléré Nesterov (moment Nesterov) algorithmes avec des taux d'apprentissage adaptatifs AdaGrad AdaDelta RMSprop Adam

- ✓ **Loss function** : Une fonction de perte est une fonction mathématique couramment utilisée en statistique. Les fonctions de perte sont fréquemment utilisées pour créer des algorithmes d'apprentissage automatique.

La fonction de perte calcule l'erreur pour un seul exemple d'apprentissage contrairement à une fonction de coût, qui est la moyenne des fonctions de perte de chaque exemple dans l'ensemble de données d'apprentissage.

Les fonctions de perte sont utilisées dans l'apprentissage automatique pour mesurer la distance mathématique entre les valeurs prédites et les valeurs réelles

- ✓ **Activation function** : les fonctions d'activation effectuent une transformation sur une somme pondérée d'entrées plus des biais sur un neurone afin de calculer sa sortie.

En utilisant une analogie biologique, la fonction d'activation détermine la « cadence de tir » d'un neurone en réponse à une entrée ou à un stimulus. Ces fonctions introduisent des non-linéarités dans les réseaux de neurones leur permettant d'effectuer des tâches complexes telles que la reconnaissance d'images et le traitement du langage.

Sans fonctions d'activation non linéaires, les réseaux de neurones artificiels se comportent comme de simples modèles de régression linéaire.

Ces fonctions comprennent : fonction sigmoïde fonction d'unité linéaire rectifiée (ReLU) fonction tangente hyperbolique (Tanh).

Conclusion :

Dans ce chapitre nous avons vu les différents variations de paramètres

Chapitre 4

Résultat et discussion

1-Introduction

Après avoir présenté la théorie de l'apprentissage profond (DL) « deep Learning (1) » dans le chapitre précédent. Cette partie sera consacrée à la mise en œuvre d'un système pour la détection de Covid-19.

Pour cela, nous allons utiliser un groupe des résultats d'analyse sanguin sélectionnées à partir de la base de données qui contient une collection des données numériques(FNS) « numération formule sanguine (2)».

2- Environnement du travail :

Dans cette section .nous présenterons le matériel et le logiciel utilisés dans notre travail.

3 - Environnement matériel (Machine utilisé):

Afin de mettre en œuvre notre projet, nous avons utilisé un ensemble de matériel dont les caractéristiques sont les suivantes :

Un ordinateur ASUS avec les caractéristiques suivantes :

Processeur : Intel® *Core™*i7-7500U CPU 3.5GHz

Mémoire installé (RAM) :12Go

Type de système : système d'exploitation 64bits.

System exploitation (OS) : Microsoft Windows 10.

3.1- Langage de programmation(Python) :

Python est un langage de programmation interprété, interactif et orienté objet. Il intègre des modules, des exceptions, un typage dynamique, des types de données dynamiques de très haut niveau et des classes.

Il prend en charge plusieurs paradigmes de programmation au-delà de la programmation orientée objet, tels que la programmation procédurale et fonctionnelle. Python combine une puissance remarquable avec une syntaxe très claire.

Il a des interfaces avec de nombreux appels système et bibliothèques, ainsi qu'avec divers systèmes de fenêtres, et est extensible en C ou C++. Il est également utilisable comme langage d'extension pour les applications nécessitant une interface programmable. Enfin, Python est portable : il fonctionne sur de nombreuses variantes d'Unix, y compris Linux et MacOS, et sur Windows.



Figure 9: Logo Python

3. 2: Les outils et les bibliothèques utilisés :

3.2 .1- NumPy (Python) :

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 des tableaux masqués et des matrices) et un assortiment de routines pour des opérations rapides sur des tableaux, y compris mathématiques, logiques, manipulation de forme, tri, sélection, E/S, transformées de Fourier discrètes, algèbre linéaire de base, opérations statistiques de base, simulation aléatoire et bien plus encore.



Figure 10: Logo NumPy

3.2.2- TensorFlow (Python) :

TensorFlow est une bibliothèque open source pour le calcul numérique rapide.

Il a été créé et est maintenu par Google et publié sous la licence open source Apache 2.0. L'API est nominale pour le langage de programmation Python, bien qu'il existe un accès à l'API C++ sous-jacente.

Contrairement à d'autres bibliothèques numériques destinées à être utilisées dans Deep Learning comme Theano, TensorFlow a été conçu pour être utilisé à la fois dans la recherche et le développement et dans les systèmes de production,

notamment RankBrain dans la recherche Google et le projet amusant DeepDream.

Il peut fonctionner sur des systèmes à processeur unique, des GPU ainsi que des appareils mobiles et des systèmes distribués à grande échelle de centaines de machines.



Figure 11: Logo TensorFlow

3.2.3-Keras (Python) :

Keras est une API d'apprentissage en profondeur écrite en Python, exécutée sur la plate-forme d'apprentissage automatique TensorFlow. Il a été développé dans le but de permettre une expérimentation rapide. Pouvoir passer de l'idée au résultat le plus rapidement possible est la clé d'une bonne recherche.



Figure 12: Logo Keras

3.2 .4-Keras & TensorFlow :

TensorFlow 2 est une plate-forme d'apprentissage automatique open source de bout en bout. Vous pouvez le considérer comme une couche d'infrastructure pour la programmation différentiable. Il combine quatre capacités clés :

- Exécution efficace d'opérations de tenseur de bas niveau sur CPU, GPU ou TPU.
- Calcul du gradient d'expressions différentiables arbitraires.
- Mise à l'échelle du calcul sur de nombreux appareils, tels que des clusters de centaines de GPU.
- Exportation de programmes ("graphiques") vers des environnements d'exécution externes tels que des serveurs, des navigateurs, des appareils mobiles et intégrés.

Keras est l'API de haut niveau de TensorFlow 2 : une interface accessible et hautement productive pour résoudre les problèmes d'apprentissage automatique, en mettant l'accent sur l'apprentissage en profondeur moderne. Il fournit des

abstractions et des blocs de construction essentiels pour développer et diffuser des solutions d'apprentissage automatique avec une vitesse d'itération élevée.

Keras permet aux ingénieurs et aux chercheurs de tirer pleinement parti de l'évolutivité et des capacités multiplateformes de TensorFlow 2 : vous pouvez exécuter Keras sur TPU ou sur de grands clusters de GPU, et vous pouvez exporter vos modèles Keras pour qu'ils s'exécutent dans le navigateur ou sur un mobile. Appareil.

3.2.5- Conda(python)

Conda est un système de gestion de packages open source et un système de gestion d'environnement qui s'exécute sous Windows, macOS et Linux. Conda installe, exécute et met à jour rapidement les packages et leurs dépendances. Conda crée, enregistre, charge et bascule facilement entre les environnements sur votre ordinateur local. Il a été créé pour les programmes Python, mais il peut empaqueter et distribuer des logiciels pour n'importe quel langage.

Conda en tant que gestionnaire de packages vous aide à trouver et à installer des packages. Si vous avez besoin d'un package qui nécessite une version différente de Python, vous n'avez pas besoin de passer à un autre gestionnaire d'environnement, car conda est également un gestionnaire d'environnement. Avec seulement quelques commandes, vous pouvez configurer un environnement totalement séparé pour exécuter cette version différente de Python, tout en continuant à exécuter votre version habituelle de Python dans votre environnement normal.



Figure 13: Logo Conda

4-Les métriques utilisées :

1-La matrice de confusion : La matrice de confusion, appelée également matrice d'erreur, est un tableau qui présente différentes prévisions et résultats de tests, en les comparant avec des valeurs réelles. Ces matrices sont utilisées dans les statistiques, le data mining, les modèles d'apprentissage automatique et d'autres applications de l'intelligence artificielle. [36]

2-Taux de classification : Le taux de classification est le nombre de prédictions correctes faites par le modèle sur toutes les sortes de prédictions effectuées.

		ACTUAL	
		Positive	Negative
P R E D I C T E D	Positive	TP True Positive	FP False Positive
	Negative	FN False Negative	TN True Negative

Figure 14: Taux de classification illustré dans la matrice de confusion

Dans le numérateur, nos prédictions correctes (vrais positifs et vrais négatifs) (marquées en rouge dans la figure ci-dessus) et dans le dénominateur, sont le genre de toutes les prédictions faites par l’algorithme (correctes ou non).

3-Taux d’erreur L’erreur : dans les problèmes de classification est le nombre de prédictions incorrectes faites par le modèle sur tous les types de prédictions effectuées.

		ACTUAL	
		Positive	Negative
P R E D I C T E D	Positive	TP True Positive	FP False Positive
	Negative	FN False Negative	TN True Negative

Figure 15 : Taux d’erreur illustré dans la matrice de confusion

Dans le numérateur, nos prédictions incorrectes (faux positifs et faux négatifs) (marquées en rouge dans la figure ci-dessus) et dans le dénominateur, sont le genre de toutes les prédictions faites par l’algorithme (correctes ou non)

5-Métrique utilisée :

ACCURACY : Est une mesure pour évaluer les modèles de classification. De manière informelle accuracy est la fraction des prédictions que notre modèle a obtenues. Formellement, accuracy a ce qui suit :

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

(2)

Où TP = True Positives, TN = True Négatives , FP = False Positives, and FN = False Négatives.

✓ **True Positive (TP):** la prédiction ET la valeur réelle sont positives.

Exemple : Une personne malade et prévu malade.

✓ **True Négative (TN):** la prédiction ET la valeur réelle sont négatives.

Exemple : Une personne saine et prévu saine.

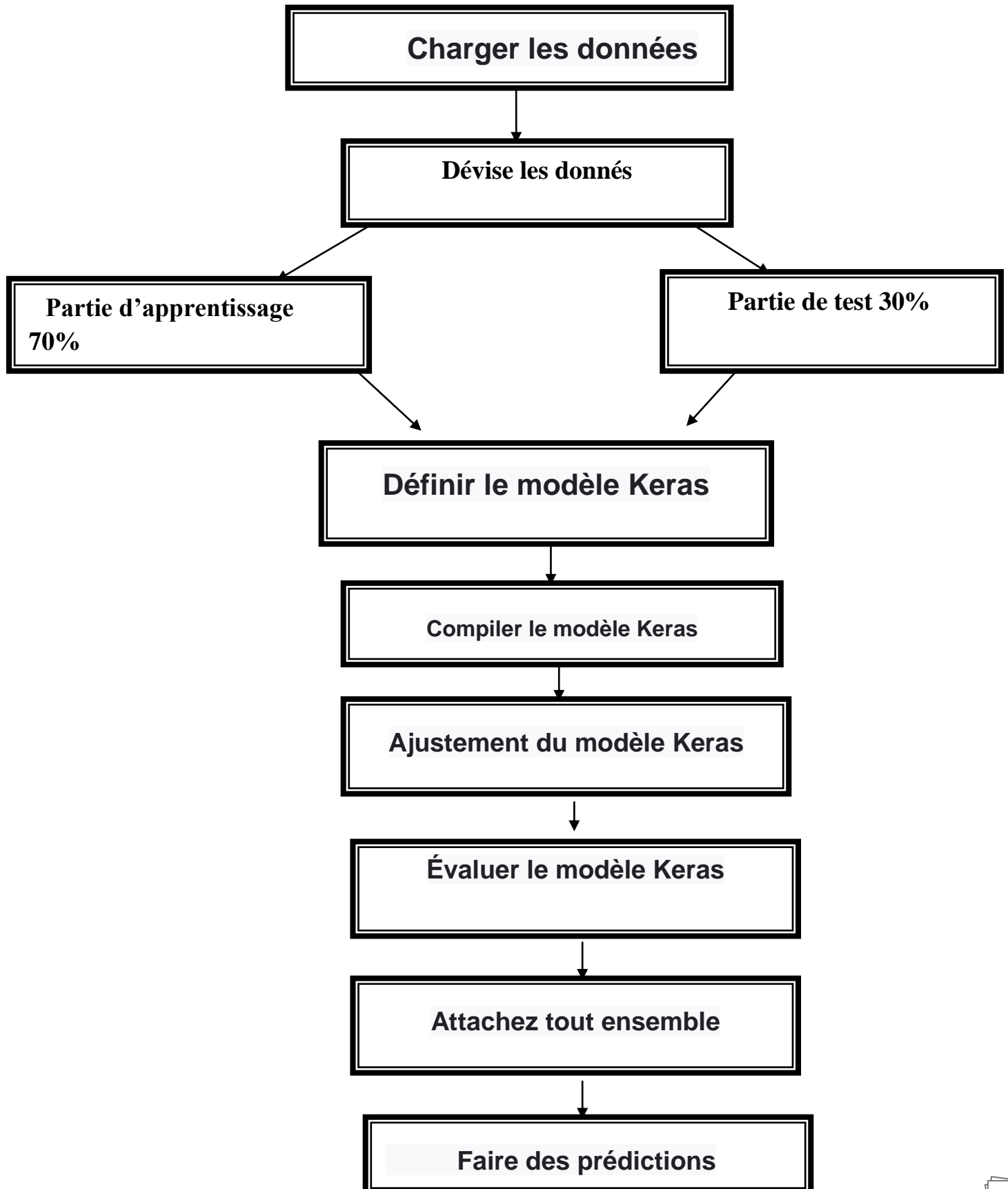
✓ **False Positive (FP):** la prédiction EST positive alors que la valeur réelle EST négative.

✓ **False Négative (FN):** la prédiction EST négative alors que la valeur réelle EST négative. [51]

L'architecture : Dans ce programme nous expliquerons comment créer notre modèle de réseau de neurones d'apprentissage profond en Python à l'aide de Keras

Nous avons suivi une série d'étapes

L'architecture générale de l'application :



Notre travail :

1- Charger les données :

La première étape consiste à définir les fonctions et les classes que nous avons l'intention d'utiliser dans ce programme.

Nous utiliserons :

La bibliothèque NumPy pour charger BDD

Utiliserons deux classes de la bibliothèque Keras pour définir notre modèle.

```
from numpy import loadtxt
from keras.models import Sequential
from keras.layers import Dense
import pandas as pd
```

Donc pouvons maintenant charger notre BDD

Dans ce programme Keras, nous allons utiliser les analyses de données du BDD. Il s'agit donc d'un problème de prédiction binaire (malade avec covid comme 1 ou non comme 0).

Toutes les variables d'entrée qui décrivent chaque patient sont numériques. Cela le rend facile à utiliser directement avec les réseaux de neurones qui attendent des valeurs d'entrée et de sortie numérique, et idéale pour notre premier réseau de neurones à Keras.

L'ensemble de données est disponible ici :

*Fichier CSV (.csv)

*Détails de l'ensemble de données

Après :

- ✓ Téléchargez la BDD au même emplacement de notre fichier python.
- ✓ Enregistrez-la avec



```
dataset = loadtxt('Balanced Dataset12-original.csv', delimiter=',')
```

- ✓ charger le fichier sous forme de matrice de nombres à l'aide de la

→ fonction **NumPy loadtxt()**

Divise la BDD en 2 parties

donne d'entrée (les analyses)

sortie

(0 ou 1 la dernière colonne)

Patient	age	quantile	Hematocrit	Hemoglobin	Platelets	Red blood Cells
0	17	0.236515	-0.022340	-0.517413	0.102004	-0.850035
1	1	-1.571682	-0.774212	1.429667	-1.361315	0.542763
2	9	-0.747693	-0.586244	-0.429480	-0.127191	0.384090
3	11	0.991538	0.792188	0.072992	0.542763	0.542763
4	9	0.190738	-0.147652	-0.668155	-0.127191	0.384090
...
595	19	0.190738	0.165628	-0.102873	0.384090	0.578024
596	19	-0.289922	-0.523588	0.663397	0.754327	0.613284
597	15	0.717175	1.105468	-0.492289	0.613284	0.613284
598	17	-3.242548	-2.779203	-1.773594	-3.318285	-3.318285
599	19	0.694287	0.541564	-0.906829	0.578024	0.578024

Lymphocytes	Leukocytes	Basophils	Eosinophils	Monocytes	
0	0.318366	-0.094610	-0.223767	1.482158	0.357547
1	-0.005738	0.364550	-0.223767	1.018625	0.068652
2	-1.114514	-0.884923	0.081693	-0.666950	1.276759
3	0.045436	-0.211488	-0.834685	-0.709090	-0.220244
4	0.002791	-1.132592	0.387152	-0.709090	2.012129
...
595	-1.583611	-0.328365	-0.834685	0.892207	1.066653
596	-1.532437	1.569499	0.081693	-0.540532	1.670707
597	0.002791	-0.550988	-0.223767	-0.709090	0.909074
598	-1.830953	-1.733675	-1.140144	-0.582671	1.381812
599	-0.295726	-1.288428	-1.140144	-0.835508	0.567652

Serum Glucose	Neutrophils	Urea	Proteina C reativa mg/dL	
0	-0.140648	-0.619086	1.198059	-0.147895
1	-0.413208	-0.127395	-0.067309	-0.286986
2	0.000000	0.880570	-0.811643	0.000000
3	0.000000	0.265957	0.000000	-0.487674
4	-0.837191	-0.422410	-1.332677	0.000000
...
605	2.433532	1.118221	0.370292	3.627427

Nous allons apprendre un modèle pour mapper des rangées de variables d'entrée (X) à une variable de sortie (y), que nous résumons souvent par $y = f(X)$

***Variables d'entrée (X) :**

Patient âge quantile	Leukocytes	Urea
Hématocrite	Leukocytes	Proteina C reativa mg/dL
Hémoglobine	Eosinophiles	Créatinine
Pl atelets	Monocytes	Potassium
Reed Blood Cellas	Sérum Glucose	Sodium
Lymphocytes	Neutrophils	Alanine transaminase
	Aspartatetransaminase	

***Variables de sortie (y) :**

0 → pas malade

1 → malade

- ✓ âpre la téléchargement de csv on divisez les colonnes de BDD a : entré est sortie
- ✓ stocké les données dans un tableau 2D où la première dimension est les lignes et la deuxième dimension est les colonnes,

Ex : [rangées, colonnes]

- ✓ divisez le tableau en deux tableaux en sélectionnant des sous-ensembles de colonnes à l'aide de l'opérateur de tranche **NumPy standard** ou ":"
Nous pouvons sélectionner les 18 premières colonnes de l'index 0 à l'index 17 via la tranche 0: 17.

On peut alors sélectionner la colonne de sortie (la 19ème variable) via l'index 18

```
[ ] dataset = loadtxt('Balanced_Dataset12-original.csv', delimiter=',')
X = dataset[:,0:18]
y = dataset[:,19]
```

2- Définir le modèle Keras :

Créer un modèle séquentiel et ajouter des calques un par un

ona :

- ✓ La première consiste à s'assurer que la couche en entrée contient le bon nombre d'entités en entrée, Cela peut être spécifié lors de la création de la première couche avec l'argument **input_dim** et en le définissant sur 8 pour les 8 variables d'entrée.
- ✓ le nombre de couches et leurs types

Les couches entièrement connectées sont définies à l'aide de la classe Dense.

Nous pouvons spécifier le nombre de neurones ou de nœuds dans la couche comme premier argument et spécifier la fonction d'activation à l'aide de l'argument d'activation.

Nous utiliserons la fonction d'activation d'unité linéaire rectifiée appelée ReLU sur les deux premières couches et la fonction sigmoïde dans la couche de sortie.

```
model = Sequential()  
model.add(Dense(12, input_dim=18, activation='relu'))  
model.add(Dense(18, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```

3- Compiler le modèle Keras :

Maintenant que le modèle est défini, nous pouvons le compiler.

La compilation du modèle utilise les bibliothèques numériques efficaces sous les couvertures (le soi-disant backend) telles que Theano ou TensorFlow.

Lors de la compilation, nous devons spécifier certaines propriétés supplémentaires requises lors de la formation du réseau. N'oubliez pas que la formation d'un réseau signifie trouver le meilleur ensemble de pondérations pour mapper les entrées aux sorties dans notre ensemble de données.

Nous devons spécifier la fonction de perte(los) à utiliser pour évaluer un ensemble de poids,

L'optimiseur est utilisé pour rechercher parmi différents poids pour le réseau et toutes les mesures facultatives que nous aimerions collecter et rapporter pendant la formation.

Dans ce cas, nous utiliserons l'entropie croisée comme argument de perte.

Cette perte est pour un problème de classification binaire et est définie dans Keras comme "binary_crossentropy".

Vous pouvez en savoir plus sur le choix des fonctions de perte en fonction de votre problème ici

Nous définirons l'optimiseur comme l'algorithme efficace de descente de gradient stochastique "Adam" Il s'agit d'une version populaire de la descente de gradient car elle s'ajuste automatiquement et donne de bons résultats dans un large éventail de problèmes

Enfin, comme il s'agit d'un problème de classification, nous collecterons et rapporterons la précision de la classification, définie via l'argument metrics.

```
[ ] model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

4- Ajustement du modèle Keras :

Il est maintenant temps d'exécuter le modèle sur certaines données.

Nous pouvons former ou ajuster notre modèle sur nos données chargées en appelant la **fonction fit ()** sur le modèle.



La formation se déroule sur des époques et chaque époque est divisée en lots.

Époque : un passage dans toutes les lignes de l'ensemble de données d'apprentissage.

Lot : un ou plusieurs échantillons pris en compte par le modèle au cours d'une époque avant la mise à jour des pondérations.

Une époque est composée d'un ou plusieurs lots, en fonction de la taille de lot choisie et le modèle est adapté à de nombreuses époques

Le processus de formation s'exécutera pendant un nombre fixe d'itérations à travers l'ensemble de données appelé epochs, que nous devons spécifier à l'aide de l'argument epochs.

Nous devons également définir le nombre de lignes de l'ensemble de données qui sont prises en compte avant que les poids du modèle ne soient mis à jour à chaque époque, appelé la taille du lot et défini à l'aide de l'argument batch_size.

Pour ce problème, nous allons exécuter un petit nombre d'époques (406) et utiliser une taille de lot relativement petite de 18

Ces configurations peuvent être choisies expérimentalement par essais et erreurs. Nous voulons former suffisamment le modèle pour qu'il apprenne un bon (ou assez bon) mappage des lignes de données d'entrée à la classification

sortie

```
model.fit(X, y, epochs=406, batch_size=18, verbose=0)
```

Le modèle contiendra toujours une certaine erreur, mais la quantité d'erreur se stabilisera après un certain point pour une configuration de modèle donnée ,C'est ce qu'on appelle la convergence du modèle.

5- Évaluer le modèle :

nous séparons les données en ensembles de données d'entraînement et de test pour l'entraînement et l'évaluation du modèle

nous évaluons l'ensemble de données d'entraînement du modèle à l'aide de la **fonction `évalue()`** et lui transmettons les mêmes entrées et sorties que celles utilisées pour entraîner le modèle.

Cela générera une prédiction pour chaque paire d'entrée et de sortie et collectera des scores, **y** compris la perte moyenne et toutes les mesures que vous avez configurées, telles que la précision.

La fonction `évalue()` —→ renverra une liste avec deux valeurs. Le premier sera la perte du modèle sur BDD et le second sera la précision du modèle sur BDD

Nous sommes uniquement intéressés par la précision des rapports, nous ignorerons donc la valeur de perte.

```
accuracy = model.evaluate(X, y)
print('Accuracy: %.2f' % (accuracy*100))
```

Résultat et discussion :

Dans le résultat, vous voyez un message pour chacune des 150 époques qui imprime la perte et la précision, suivi de l'évaluation finale du modèle formé sur l'ensemble de données de formation.

L'exécution sur mon poste de travail exécutant le processeur prend environ 10 secondes.

Idéalement, nous aimerions que la perte soit nulle et que la précision soit de 1,0 (par exemple 100%). Cela n'est pas possible pour des problèmes autres que de simples problèmes d'apprentissage automatique. Au lieu de cela, nous aurons toujours des erreurs dans notre modèle. L'objectif est de choisir une configuration de modèle et une configuration de formation qui permettent d'obtenir la perte la plus faible possible et la précision la plus élevée possible pour un ensemble de données donné.

```
1 ...
2 768/768 [=====] - 0s 63us/step - loss: 0.4817 - acc: 0.7708
3 Epoch 147/150
4 768/768 [=====] - 0s 63us/step - loss: 0.4764 - acc: 0.7747
5 Epoch 148/150
6 768/768 [=====] - 0s 63us/step - loss: 0.4737 - acc: 0.7682
7 Epoch 149/150
8 768/768 [=====] - 0s 64us/step - loss: 0.4730 - acc: 0.7747
9 Epoch 150/150
10 768/768 [=====] - 0s 63us/step - loss: 0.4754 - acc: 0.7799
11 768/768 [=====] - 0s 38us/step
```

```
DecisionTreeRegressor accuracy_score
84.52380952380952 %
```

Variation du nombre d'epoches

Epoche=30



Figure : Courbes accuracy, val accuracy, los, val los avec nombre d'epoches = 30

➤ on remarque du overfitting quand le nombre d'époche=10 les résultats s'améliorent après

Epoche=148

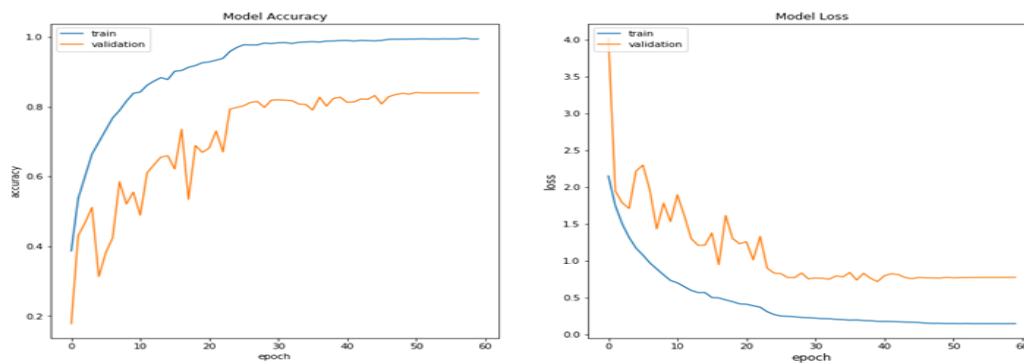


Figure : Courbes accuracy, val accuracy, los, val los avec nombre d'epoches = 148

➤ on remarque qu'il ya du overfitting au début de la courbe mais les résultats s'améliorent après

NB : Si essayez de l'exécuter dans un bloc-notes IPython ou Jupyter on obtenez une erreur.

La raison en est les barres de progression de sortie pendant la formation. Vous pouvez facilement les désactiver en définissant `verbose=0` dans l'appel aux **fonctions `fit ()` et `évaluation ()`**

```
model.fit(x, y, epochs=400, batch_size=18, verbose=0)
accuracy = model.evaluate(x, y, verbose=0)
print(accuracy)
```

Les résultats peuvent varier compte tenu de la nature stochastique de l'algorithme ou de la procédure d'évaluation, ou des différences de précision numérique

Donc l'exécution du programme plusieurs fois est très importante

Pour l'adapter

```
1 Accuracy: 75.00
2 Accuracy: 77.73
3 Accuracy: 77.60
4 Accuracy: 78.12
5 Accuracy: 76.17
```

```
DecisionTreeRegressor accuracy_score
84.52380952380952 %
```

```
DecisionTreeRegressor accuracy_score
85.0 %
```

Tous les scores de précision sont d'environ 77 % et la moyenne est de 85 %.

- **Faire des prédictions:**
- Nous pouvons adapter le programme ci-dessus et l'utiliser pour générer des prédictions sur l'ensemble de données d'entraînement, en prétendant qu'il s'agit d'un nouvel ensemble de données que nous n'avons jamais vu auparavant que nous concevons l'ensemble de données en 2 parties les 70 % les plus présents pour effectuer la formation et les 30 secondes présentes 30% pour faire des tests .

```
import numpy as np
mask = np.random.rand(len(df)) <= 0.7
training_data = df[mask]
testing_data = df[~mask]

print(f"No. of training examples: {training_data.shape[0]}")
print(f"No. of testing examples: {testing_data.shape[0]}")
```

```
No. of training examples: 422
No. of testing examples: 178
```

Faire des prédictions est aussi simple que d'appeler la fonction **prédicat ()** sur le modèle,

Sigmoïde → Nous utilisons une fonction d'activation sur la couche de sortie, de sorte que les prédictions seront une probabilité comprise entre 0 et 1

Nous pouvons facilement les convertir en une prédiction binaire nette pour cette tâche de classification en les arrondissant

```
predictions = model.predict(X)
rounded = [round(x[0]) for x in predictions]
print(predictions)
```

Alternativement, nous pouvons convertir la probabilité en 0 ou 1 pour prédire directement les classes nettes

```
predictions = model.predict(x)
rounded = [round(x[0]) for x in predictions]
print(predictions)
predictions = (model.predict(x) > 0.5).astype(int)
```

La partie Test de modèle :

L'exécution du programme n'affiche pas la barre de progression comme auparavant car nous avons défini l'argument verbeux sur 0.

Une fois le modèle ajusté, des prédictions sont effectuées pour tous La BDD, et les lignes d'entrée et la valeur de classe prévue sont imprimées et comparées à la valeur de classe attendue.

Nous pouvons voir que la plupart des lignes sont correctement prédites. En fait, nous nous attendrions à ce qu'environ 85 % des lignes soient correctement prédites sur la base de nos performances estimées du modèle

```
1 [6.0, 148.0, 72.0, 35.0, 0.0, 33.6, 0.627, 50.0] => 0 (expected 1)
2 [1.0, 85.0, 66.0, 29.0, 0.0, 26.6, 0.351, 31.0] => 0 (expected 0)
3 [8.0, 183.0, 64.0, 0.0, 0.0, 23.3, 0.672, 32.0] => 1 (expected 1)
4 [1.0, 89.0, 66.0, 23.0, 94.0, 28.1, 0.167, 21.0] => 0 (expected 0)
5 [0.0, 137.0, 40.0, 35.0, 168.0, 43.1, 2.288, 33.0] => 1 (expected 1)
```

- **Accuracy : 85%**

- **le nombre de prédictions correctes pour chaque classe :**

- infecter : 18

- pas infecter : 597

- **le nombre de prédictions incorrectes pour chaque classe :**

- infecter : 2

- pas infecter :1

Conclusion général

La science de l'apprentissage en profondeur a été témoin d'un avenir large et rapide dans le développement de diverses applications et dans une notre mémoire, il a développé une phrase d'instructions qui permettait de prédire l'infection au covid 19 de manière rapide et efficace

En utilisant les résultats des analyses de sang et en utilisant les derniers développements de la science des médias automatisés, l'apprentissage en profondeur, nous avons choisi le DNN comme méthode de classification.

Notre travail, c'est un modèle général qui peut s'adapter à n'importe quelle base de données, c'est-à-dire qu'il est multi-usage

Bibliographie :

- [1]. <https://www.letribunaldunet.fr/wp-content/uploads/2020/12/virus.jpg>
- [2] www.health.gov.au/sites/default/files/documents/2020/03/coronavirus-covid-19-frequently-asked-questions.pdf
- [3] H. LAROCHELLE AND Q. SHERBROOKE, Historique de travail, Études, 3, p. 2Y5
- [4] <https://user-images.githubusercontent.com/7995307/64065525-153d9580-cbfe-11e9-9236-d9ffc3f34a43.png>
- [5] D. LEARNING, Deep Learning, High-Dimensional Fuzzy Clustering, (2020).
- [6] <https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/deeplearningvs-machine-learning/> (02/07/2021)
- [7] inside-machinelearning.com/fonction-dactivation-comment-ca-marche-une-explication-simple/#:~:text=Smith%20on%20Unsplash-,Qu'est-ce%20qu'une%20fonction%20d'activation,manière%20de%20voir%20une%20donnée.
- [8] C. TOUZET, les réseaux de neurones artificiels, introduction au connexionnisme, Ec2, 1992
- [9] s. boubaya, d. berbit, et al., deep learning pour la segmentation d'images, phd thesis, university of m'sila, 2021
- [10] B. DYHIA AND I. THANINA, Classification d'images agricoles avec le Deep Learning., PhD thesis, Université Mouloud Mammeri, 2019
- [11] Traitement de l'image De la numérisation à l'archivage et la communication ; Coordonné par Olivier Héléon/Joël Chabriais/Bernard Gibaud/Denis Mariano
- [12] A. BUHOT, Etude de propriétés d'apprentissage supervisé et non supervisé par des méthodes de physique statistique, PhD thesis, Université Joseph-Fourier-Grenoble I, 1999
- [13] <https://actualiteinformatique.fr/intelligence-artificielle/definition-deeplearning>
(22/08/2021)
- [14] A. GRIDI AND T. GUERRAM, Un outil de deep learning pour les données textuelles, (2020). (02/07/2021)
- [15] Z.Sellami. 7 juillet 2019. Optimisation du CNN par l'algorithme Génétique pour la Reconnaissance de Visage. Mémoire de fin d'étude. Université Mohamed Khider de Biskra
- [16] <https://www.quantmetry.com/blog/une-petite-histoire-du-machine-learning/>
- [17] <https://static.javatpoint.com/tutorial/machine-learning/images/regression-vs-classification-in-machine-learning.png>

[18] https://fr.wikipedia.org/wiki/Apprentissage_par_reforcement#/media/Fichier:Reinforcement_learning_diagram_fr.svg

[19] <https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/>

[20] <https://www.juripredis.com/fr/blog/id-19-demystifier-le-machine-learning-partie-2-les-reseaux-de-neurones-artificiels>

[21] <https://medium.com/@varunachary96/an-overview-about-machine-learning-882c20df944d>

[22] <https://www.futura-sciences.com/sante/definitions/coronavirus-covid-19-18585/>

[23] <https://www.lumni.fr/article/coronavirus-definition-transmission-et-symptomes>

[24] <https://medium.com/@sprhlab/understanding-deep-learning-dnn-rnn-lstm-cnn-and-r-cnn-6602ed94dbff>

[0] https://www.google.com/url?sa=i&url=http%3A%2F%2Fcedric.cnam.fr%2F~thomen%2Fcours%2FUS330X%2Ftpbackprop.html&psig=AOvVaw1XXEiL5UN_Zl2pT88R6C5I&ust=1653368933114000&source=images&cd=vfe&ved=0CAwQjRxqFwoTCOiDkrbt9PcCFQAAAAAdAAAAABAO

[25] <https://datascientest.com/deep-neural-network>

[26] <https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/>