الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Master

Filière Électronique

Spécialité Instrumentation

Présenté par

KRAMOU Rime & DJADI Aya

Détection d'activité vocale basée sur l'apprentissage profond

Proposé par : YKHLEF Farid

Année Universitaire 2022-2023

Remerciements

Nous tenons à exprimer nos profondes gratitudes envers Dieu qui nous a accordé l'aide la patience et le courage nécessaires pour mener à bien ce travail.

Nous tenons à remercier tout particulièrement notre promoteur Mr F.YAKHLEF

Pour ses conseils utiles, disponibilité et dévouement, et surtout de nous avoir accordé

Sa confiance pour mener à bien ce travail ; qu'il puisse trouver ici l'expression

De notre reconnaissance, de notre profond respect et de nos plus vifs remerciements.

Nous tenons à adresser nos plus vifs remerciements aux membres de jury

pour Avoir accepté de juger notre travail.

On tient à remercier nos familles respectives, surtout nos parents pour nous avoir

donnés tous les moyens nécessaires afin d'accomplir nos études.

Enfin, que tous ceux qui ont participé de près ou de loin au bon déroulement

de ce travail, trouvent ici l'expression de notre reconnaissance et de

nos remerciements les plus profonds.

À toutes et à tous Grand Merci

Dédicaces

C’est avec profonde gratitude et sincères mots, que je dédie ce modeste travail

de fin d’études à mes chers parents, qui ont sacrifié leur vie pour ma réussite

et éclairé le chemin par leurs conseils judicieux, J’espérais qu’un jour, je pourrais

rendre un peu de ce qu’ils ont Fait pour moi, Que Dieu leur Prête

bonheur et langue vit, je dédie ce travail à mes frères, ma famille, tous mes amis et tous mes professeurs qui nous enseigner

Rime kramou

C’est avec profonde gratitude et sincères mots, Je tiens à dédier ce modeste travail :

A l’homme de ma vie, mon exemple éternel, celui qui s’est toujours sacrifié

pour me voir réussir, que dieu te donne bonheur et longue vie. À toi Papa

A la lumière de mes jours, ma vie. Maman que dieu te accorde une bonne santé et longue vie.

A tous mes frères, mes sœurs et mes neveux.

A tous les membres de Ma famille.

A tous mes amis et tous mes professeurs.

Aya Djadi

ملخص : يعتبر الكشف عن النشاط الصوتي (DAV) أحد التقنيات الرئيسية للعديد من التطبيقات الصوتية. هذه طريقة مهمة في معالجة الكلام، حيث تكتشف وجود أو غياب الصوت. في السابق كان أداء vad يعتمد على طرق تعتمد على معالجة الإشارات، لكنه لم تقدم أداءً مرضيًا في البيئات عالية الضوضاء، لذلك أصبح التعلم العميق بديلاً. منذ ذلك الحين، اعتمدنا في الدراسة التجريبية على ثلاثة هياكل للتعلم العميق، وهي الشبكات العصبية التلافيفية (CNN) وشبكة DenseNet ، واستخدمنا أيضًا قواعد البيانات الثلاث للكلام والضوضاء، وهي LibriSpeech ،TidiGets و Chimie5 على التوالي. قمنا بقياس الدقة في بيئات الضوضاء المنخفضة بحساسيات مختلفة وحققنا دقة بنسبة 100٪.

كلمات المفاتيح: الشبكات العصبية التلافيفية، التعلم العميق، قاعدة البيانات

Résumé : La détection d'activité vocale (DAV) est considérée comme l'une des principales techniques pour de nombreuses applications vocales. C'est une méthode importante dans le traitement de la parole, car elle détecte la présence ou l'absence de la voix. Auparavant les performances de la DAV étaient basées sur des méthodes qui dépendent du traitement du signal, mais ne donnaient pas des performances satisfaisantes dans des environnements à bruit élevé, donc l'apprentissage profond est devenu une alternative. A partir de l'a, nous avons adoptés dans l'étude expérimentale sur trois structures pour l'apprentissage profond qui sont les réseaux de neurones Convolutifs (CNN) et un réseau DenseNet, et nous avons également utilisés les trois bases de données pour la parole et le bruit, qui sont LibriSpeech, TidiGets et Chimie5 successivement. Nous avons mesurés la précision dans des environnements à faible bruit avec diverses sensibilités et nous avons obtenus une précision de 100%.

Mots clés : Réseaux de neurones convolutifs, apprentissage profond, base de données.

Abstract: Voice activity detection (VAD) is considered one of the most important techniques for many speech applications. It is an important method in speech processing, as it detects the presence or absence of speech. Previously VAD performance was based on methods that depended on signal processing signal processing, but did not perform satisfactorily in high-noise environments, so deep learning became an alternative. A , we adopted in the experimental study three structures for deep learning deep learning, namely Convolutional Neural Networks (CNN) and a DenseNet network, and we also used the three databases for speech and noise, namely LibriSpeech, TidiGets and Chimie5 in succession. We measured accuracy in low-noise environments with various sensitivities and achieved 100% accuracy.

Keywords: Convolutional Neural Networks, deep learning, database.

Listes des acronymes et abréviations

CNN	:	CONVOLUTIONAL NEURAL NETWORKS
CPU	:	Central Processing Unit
dB	:	Décibels
DF	:	Data Farme
DL	:	Deep Learning
DVA	:	Détections D'Activité vocale
FIR	:	Finite Impulse Response
GPU	:	Graphic Processing Unit
IA	:	Intelligence Artificielle
LPC	:	Linear Predictive Coding
ML	:	Maching Learning
MLP	:	MULTILAYER PERCEPTRON
MAE	:	Mean Absolut error
MFCC	:	Mel Frequency Cepstral Coefficients
PLP	:	Perceptual Linear Predictive
PMC	:	Perceptron Multi-Couche
PS	:	Poids Synaptique
RAP	:	Reconnaissance automatique de la parole
RASTA-PLP	:	RelAtive SpacTrAl-PLP
RMSE	:	ROOT MEAN SQUARE ERROR
RSB	:	Rapport Signal sur Bruit
ReLU	:	Rectified Linear Unit
SGD	:	Stochastic Gradient Descent
STFT	:	Short-Time Fourier Transform
SGDM	:	Stochastic Gradient Descent Momentum
TFD	:	Transformée de Fourier Discrète
TPU	:	Tensor Processing Unit

Table des matières

Liste des figures

Introduction générale

Ces dernières années, la DAV a reçu beaucoup d'attention dans les domaines de la recherche scientifique et dans le domaine de la communication vocale en particulier, où les chercheurs et les experts ont travaillé sur son développement efficace, car c'est un facteur clé dans le processus de nombreuses applications vocales, y compris la découverte de parties du silence dans la parole et la détection à propos du bruit.

La DAV est une technologie en plein essor qui a le potentiel d'améliorer de nombreuses applications dans le domaine du traitement audio.

Ces derniers temps, la plupart des recherches se sont concentrées sur l'intelligence artificielle spécifiquement les méthodes d'apprentissage profond (Deep Learning) pour développer certaines technologies. L'intelligence artificielle a été développée pour simuler le comportement du cerveau humain. Les premières tentatives de modélisation du cerveau sont anciennes, même avant l'ère de l'informatique. Dans ce sens, les scientifiques ont pensé à essayer d'imiter le fonctionnement de l'esprit humain et ont découvert que le neurone est l'élément le plus important pour la formation et la collecte du cerveau. Dans cette perspective, des études ont commencé sur le mécanisme des réseaux neuronaux biologiques pour simuler leur travail sur un ordinateur pour résoudre des problèmes complexes. Par conséquent, des réseaux neuronaux artificiels ont été conçus.

L'apprentissage profond (DL) est un sous-ensemble de l'apprentissage automatique (ML) où ce dernier est l'un des domaines de l'intelligence artificielle. L'apprentissage profond est basé sur l'idée des réseaux neurones artificiels, donc ces concepts sont interconnectés même s'ils ne sont pas équivalents. Il dispose de plusieurs architectures parmi eux le réseau CNN, est un type des réseaux qui

permet d'extraire des caractéristiques, qui tire son travail du système visuel humain.

De nos jours, les recherches prennent la DAV comme un problème de classification binaire, car les réseaux de neurones convolutifs ont montré de bons résultats en tant qu'architecte qui gère efficacement les données séquentielles.

Afin de mettre en œuvre une détection d'activité vocale, dans notre projet on s'intéresse aux techniques du Deep Learning, plus spécialement les réseaux de neurones convolutifs. Dans cette optique, le réseau de type DenseNet[1] a été utilisé. Ce dernier est l'un des architectures d'un réseau de neurone convolutif, qui fonctionne mieux avec moins de complexité. Aussi il répond à une certaine architecture communément appelée réseaux de neurones densément connectés.

Dans la partie expérimentale, nous avons utilisés un ensemble de base du signal de la parole tel que : LibriSpeech, TidiGets et Chimie5. L'organisation de cette étude s'appuie sur 3 chapitres :

Le premier chapitre : Comprend tout ce qui concerne la DAV et son principe,

Le deuxième chapitre : introduit le mécanisme de l'IA (Maching Learning et Deep Learning), ensuite l'approche d'apprentissage profond (les réseaux de neurones convolutifs).

Le troisième chapitre : présente les expérimentations menées sur la DAV et nous discutons également des différents résultats obtenus.

Enfin, nous présentons une conclusion générale qui résume ce qui a été précédemment étudié, en plus de suggérer des perspectives d'avenir pour ces travaux.

[1] https://datascientest.com/reseaux-de-neurones-densenet

Chapitre 1 Notions de base sur la détection d'activité vocale

1.1 Introduction

Dans des environnements bruyants, l'être humain est confronté à de nombreux problèmes lorsqu'il parle avec d'autres, c'est pourquoi une technique de traitement de la parole a été découverte, qui est la détection d'activité vocale (DAV) à travers laquelle la présence ou l'absence de la voix humaine est déterminée.

Pour atteindre notre objectif, dans ce chapitre nous abordons brièvement quelques concepts de base, qui sont : la définition de la DAV, la parole, bruit et rapport signal sur bruit.

1.2 Détection d'activité vocale (DAV)

La DAV est considérée comme une technique de traitement de la parole, car elle traite le silence et le bruit ou les informations vocales sans rapport avec la voix humaine comme des zones non parole.

Cette technique distingue des zones parole et des zones non parole. Le codage de la parole et la reconnaissance vocale font partie des principales applications liées à DAV [1].

1.2.1 Principe et Fonctionnement

Comme nous l'avons vu précédemment, DAV fait la distinction entre deux zones du signe de la parole. Idéalement, DAV produit "1" s'il y a de voix (zone active), ou "0" s'il n'y a pas de voix (zone inactive), voir la figure 1.1 [2].

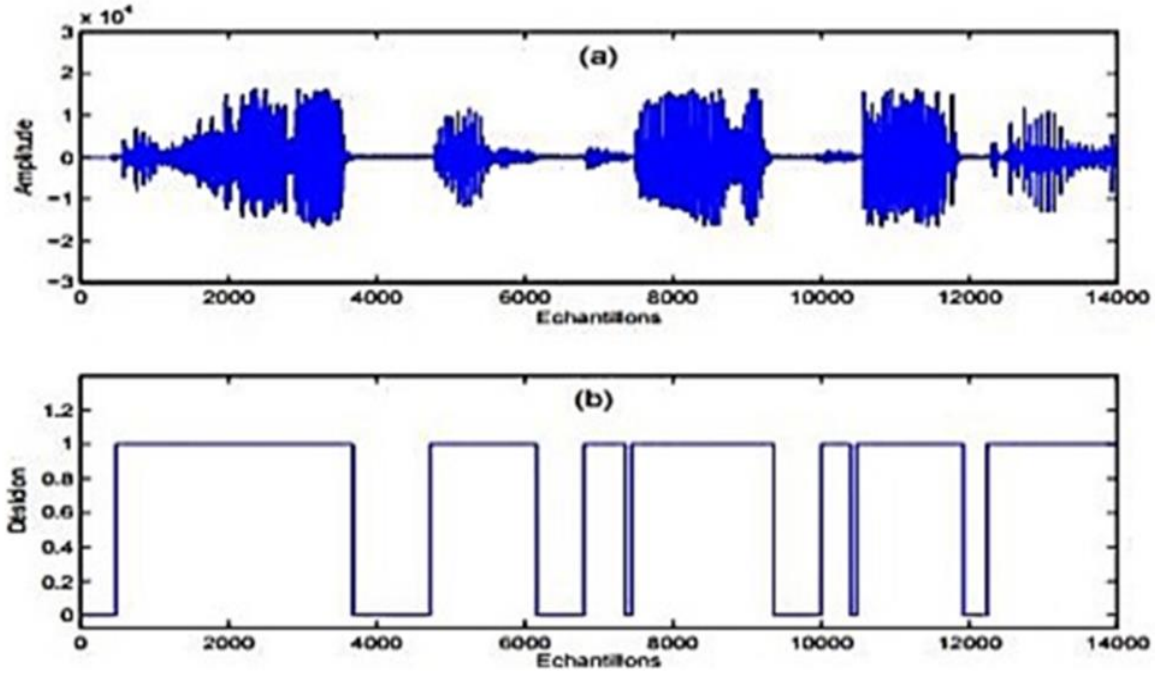


Figure 1-1 : exemple illustre le principe de la DAV

.

Le signal audio d'entrée est découpé en trame, par la plupart des méthodes DAV. La trame fait partie d'un signal de longueur fixe qui est ordonné en quelques millisecondes. Par conséquent, la DAV décide la trame qui contient de la parole ou qui ne contient pas de parole, alors il est possible de découper les trames avec ou sans chevauchement. La figure 1.2 clarifie les types de découpages sur un signal audio, où chaque découpage représente un pourcentage du chevauchement [3].

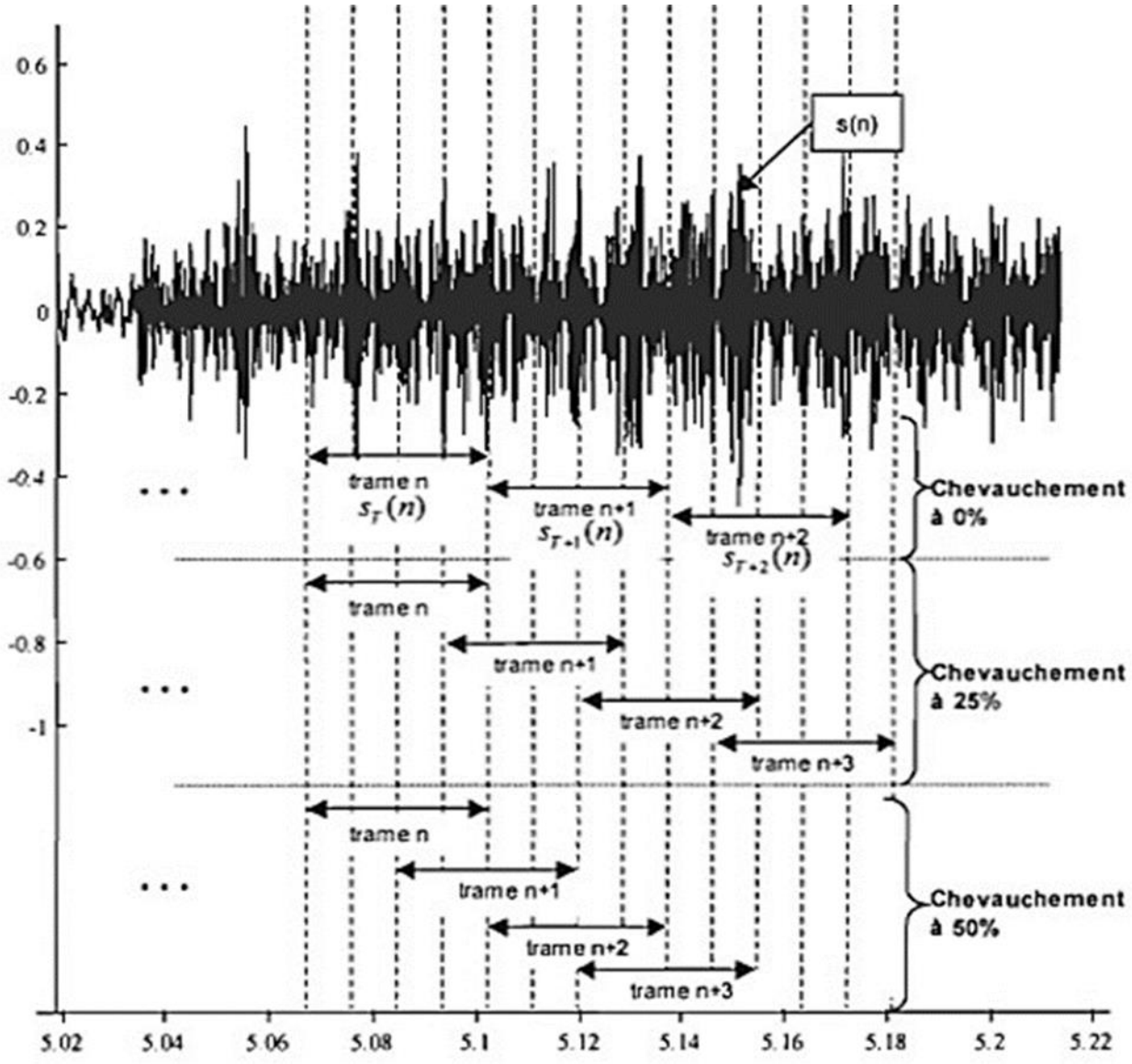


Figure 1-2 : Découpage du signal audio à trois taux de chevauchement différents

Généralement, chaque algorithme DAV suit le processus général qui décrite dans la figure 1.3 pour fournir une décision 0 ou 1 à partir d'un signal audio.

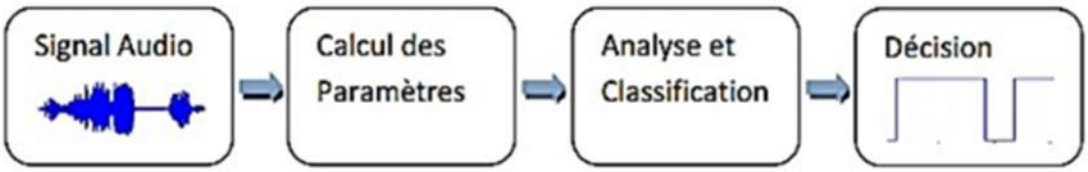


Figure 1-3 : Processus général d'un algorithme de la DAV.

La première étape du processus générale est le calcul des paramètres, où à partir d'une trame on peut extraire un ensemble de paramètres qui sont calculés soit dans le domaine temporel, soit dans le domaine spectral.

1.2.2 Domaine temporelle

Ce domaine comprend de nombreuses techniques d'analyse, notamment :

a. Taux de passage à zéro (ZCR : Zero Crossing Rate)

Dans un intervalle de temps ou un terme donné, le ZCR mesure le nombre de fois des changements de l'amplitude du signal. Alors, si les échantillons successifs ont des signes algébriques différents, un passage à zéro se produit selon le contexte des signaux à temps discret [4].

Le ZCR s'agit d'une simple mesure du bruit pour les signaux complexes. Aussi, il est utilisé pour faire une estimation approximative de la fréquence fondamentale pour les signaux à une seule voix [5].

b. Énergie à court terme

Représente l'enveloppe temporelle du signal, car elle exprime la valeur quadratique moyenne des valeurs de forme d'onde dans la trame de données. Grâce à la variation que cette énergie crée au fil du temps, cette variation peut devenir un indicateur fort du contenu du signal sous-jacent [6].

1.2.3 Domaine spectrale

Il existe de nombreuses techniques d'analyse dans ce domaine, mais nous fournirons une description de certain des techniques suivantes :

a. Transformée de Fourier discrète (TFD)

La TFD est une technique d'analyse, il est d´définit comme suite [7] :

$$S(K) = f\{s[n]\} = \sum_{n=0}^{N-1} s[n] e^{\frac{(-j2\pi nk)}{N}}$$

s[n] : Une séquence de temps discrète de N échantillons, avec n = 0,1,..., N-1.

K : Variable discrète de fréquence.

Le résultat de la TFD est un nombre complexe de longueur N. Si (K = 0), veux dire la fréquence nulle ou la composante continue du signal.

b. *Ondelettes*

C'est l'un des outils d'analyse du signal, l'analyse par ondelette est apparue au début des années 80. Cette analyse offre une large gamme de fonctions de base parmi lesquelles on peut choisir la plus appropriée pour une application donnée. La transformée en ondelettes offre la possibilité d'analyser un signal simultanément dans le domaine du temps et celui des fréquences, où il a été nommé «la technique d'analyse temps-fréquence » [8].

c. *MFCC (Mel Frequency Cepstral Coefficients)*

C'est une technique la plus couramment utilisée pour extraire des caractéristiques, car elle prend le domaine fréquentiel comme une base principale et fonctionne avec une approximation plus proche de l'audition humaine [9].

Le processus de cette technique illustre dans la figure 1.4

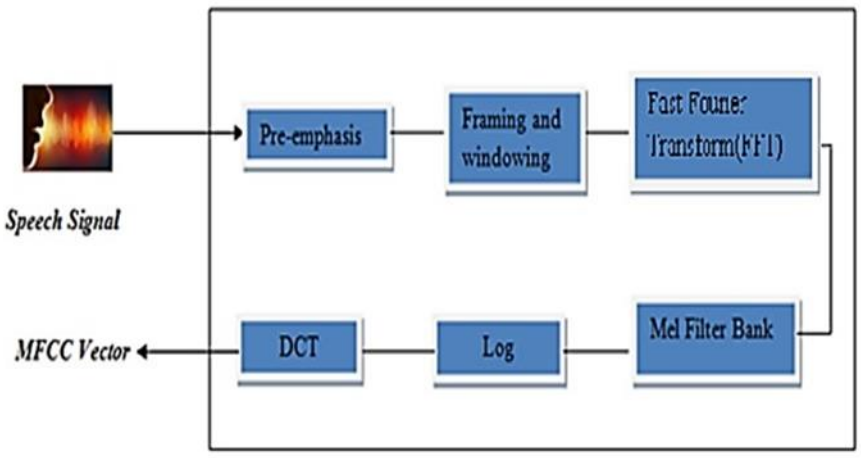


Figure 1-4 : Schéma fonctionnel pour l'extraction de caractéristiques MFCC.

Les étapes pour produire les caractéristiques MFCC sont décrites dans l'ordre suivant [10] :

1. Préaccentuation : A cette étape, le signal de parole augmente l'amplitude des bandes haute fréquence et aussi les amplitudes des bandes inférieures qui sont mis en œuvre par le filtre FIR (Finite Impulse Response) qui sont réduits.

2. Encadrement et fenêtrage : A cette étape, le signal de parole est divisé en un certain nombre de trames, où la taille de la trame est 25 ms et afin de réduire l'interruption du signal au début de chaque bord de la trame, une fenêtre de Hamming est appliquée.

3. Transformateur de Fourier rapide : La conversion dans le domaine temporel en domaine fréquentiel pour chaque trame ayant N échantillons.

4. Banque de filtres Mel : C'est une conversion d'une échelle linéaire à une échelle mel pour l'échelle de fréquence.

5. Logarithme : Cette étape connue sous le nom de spectre log mel, car le logarithme est pris pour la banque de filtres mel.

6. Transformée discrète en cosinus : A cette étape, une caractéristique MFCC est produite car une conversion en domaine de fréquence `a domaine de temps se faire pour L'échelle log mel.

d. LPC (Linear Predictive Coding)

Le codage prédictif linéaire est une technique d'analyse de la parole utilisée pour réduire la somme des différences quadratiques entre le signal de parole d'origine et le signal de parole estimé sur une période de temps limitée. Il est considéré comme une technique statique pouvant produire des paramètres d'ordre inférieur. LPC est utile pour encoder la qualité de parole avec un taux de bit faible [9] [11].

e. PLP (Perceptual Linear Predictive)

Le Prédictif linéaire perceptif est une technique basée sur le spectre à court terme de la parole. Il est similaire à l'analyse LPC et MFCC, car il est au but de d'écrire plus précisément la psychophysique de l'audition humaine dans le processus d'extraction des caractéristiques [11].

Le processus de cette méthode se résume à trois étapes de traitement consécutives [12] :

Dans la première étape, selon une échelle d'audition, le signal de parole est analysé pour obtenir un spectre.

Dans la deuxième étape, par interpolation et transformée de Fourier inverse, le spectre obtenu à partir de la première étape est modifié et le signal obtenu est également passé à travers un filtre afin de réduire les dimensions du spectre et d'augmenter la résolution fréquentielle.

Dans la troisième étape qui peut être supprimée. Par filtrage inverse et le passage dans le domaine fréquentiel et désaccentuation, le signal de parole peut être reconstruit.

f. RASTA-PLP

RASTA est une abréviation de RelAtive SpacTrAl, cette méthode a été développée en raison des limitations rencontrées par l'algorithme PLP, afin d'atténuer les effets des distorsions spectrales linéaires. Donc, le principe de la méthode RASTA-PLP est de remplacer le spectre à courte terme par un spectre estimé, où par passage à travers un filtre chaque canal fréquentiel est modifié. Lors de l'exécution de ce filtrage dans le domaine spectral logarithmique, il supprime les composantes spectrales fixes et de celui-ci les effets convolutifs du canal de communication sont également supprimés [12].

1.3 Signal de la parole

•***Parole*** : C'est la capacité d'exprimer et de percevoir des sons pour permettre aux ˆêtres humains de communiquer entre eux [13].

•***Son :*** Physiquement, c'est la propagation des changements de pression dans un milieu (gazeux, liquide, solide), formant des ondes, il est donc considéré comme une vibration mécanique. Ainsi, le son est un signal perçu par le sens de l'ouïe. Donc, le son peut être caractérisé par [14] :

1) Fréquence : C'est le nombre de changements de pression par seconde, il est estimé en hertz, car plus la valeur de fréquence est élevée, plus le son est élevé et vice versa.

2) Intensité : Représente la force du son, qui est mesurée en décibels, cette force correspond à la base des différences de pression et en fonction du milieu qui s'y répand.

3) Durée : Puisqu'un son est une onde qui se déplace dans le temps dans le milieu, donc la durée est estimée comme l'intervalle de temps entre deux évènements.

•***Signal :*** C'est l'énoncé de l'évolution temporelle ou spatiale d'un phénomène sous une forme physique.

1.3.1 Production du signal de la parole

La personne utilise ses systèmes respiratoire et digestif pour produire le signal de la parole, où les sons sont produits par l'air expiré des poumons. Après que l'air dans les poumons se soit déplacé de la trachée à travers le larynx, où il touche les cordes vocales, qui en elles-mêmes effectuent le processus d'ouverture et de fermeture de la glotte aux voies vocales s'étendant du pharynx aux lèvres, un signal acoustique est émis, comme illustré par la figure 1.5 [15].

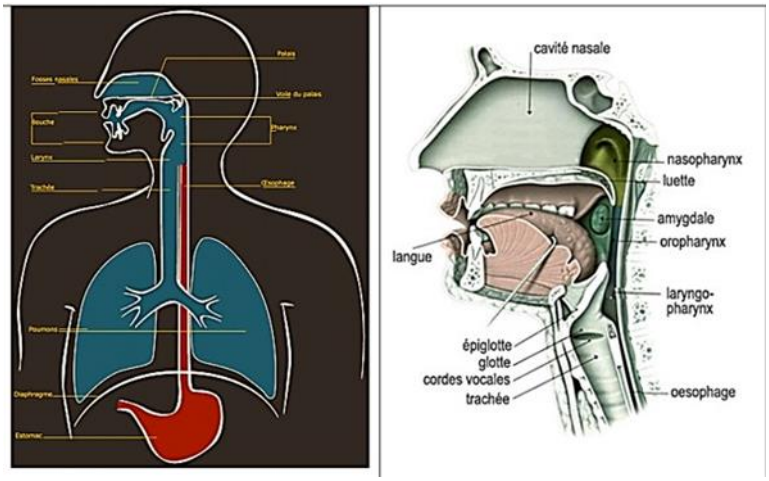


Figure 1-5 : Appareil phonatoire

Ce signal acoustique distinctif avec une énergie non stationnaire limitée et sa structure complexe atteignent l'oreille (Figure1.6) et exactement à l'oreille interne, où se trouve le nerf auditif, qui `a son tour transmet le message parlé au cerveau pour que ce dernier l'interprète [13].

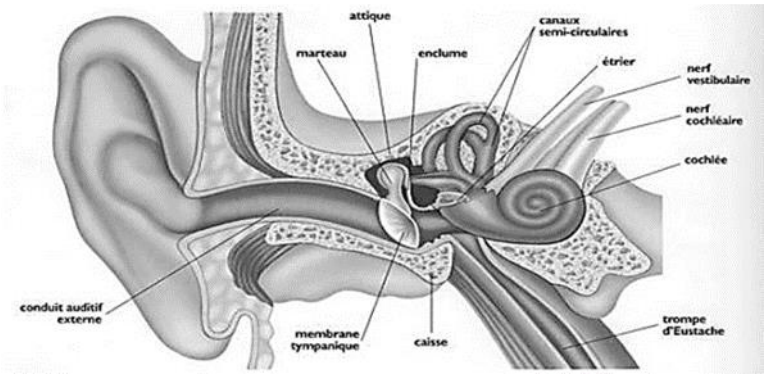


Figure 1-6 : Appareil phonatoire.

1.3.2 Caractéristiques du signal de la parole

Comme nous l'avons déjà dit à propos du signal vocal, il a une structure complexe, ou` il est parfois périodique et autre aléatoire, de sorte que les sons de la parole sont divisés en deux catégories, qui sont les suivantes [16] :

• ***Sons voisés*** : Ce sont des signaux semi-périodiques, résultant de vibrations périodiques des cordes vocales et également dus au conduit vocal et à sa configuration semi-stable, comme le montre la figure 1-7.

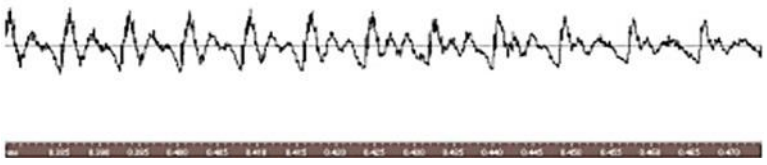


Figure 1-7 : Exemple de son voisé.

• ***Sons non voisés*** : Les cordes vocales ne vibrent pas car elles sont dans une position écartée. Donc, ces signaux ne sont pas de structure périodique, comme le montre la figure 1.8.

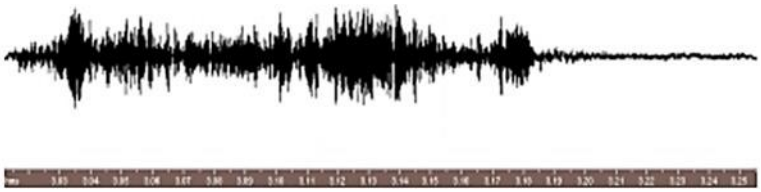


Figure 1-8 : Exemple de sons non voisé.

1.3.3 Paramètres du signal de la parole

De manière générale, le signal vocal est caractérisé par trois paramètres, qui sont les suivants [17] :

a) Fréquence fondamentale : Ou pitch pour les sons voisés, il montre la fréquence du cycle d'ouverture et de fermeture des cordes vocales.

b) Energie : Au début du larynx, l'intensité du son est liée à la pression de l'air et à partir de celle-ci l'énergie est représentée. En fonction du type de son, l'amplitude du signal de parole varie dans le temps. Expression de l'énergie selon d'une portion du signal de parole est comme suit :

$$E = \sum_{n=0}^{N-1} s^2(n)$$

S (n): Segment analysé.

N : Taille de la trame

c)Spectre : C'est l'intensité du son selon la fréquence, qui est obtenue au moyen d'une analyse de Fourier à court terme.

1.4 Bruit

1.4.1 Définition

Le bruit s'agit d'une perturbation basée sur la distorsion du message envoyé. Donc, il est difficile de percevoir et de comprendre les informations (parole), ce qui conduit à un changement de la qualité de la communication [17].

1.4.2 Types de bruit [18]

Bruit acoustique : A travers les mouvements des sources (trafic, pluie, ventilateurs, vent, voitures, etc.) ce bruit est généré.

Bruit blanc : Il a la même énergie pour toutes les fréquences, car ces fréquences composent ce bruit au même niveau statistique.

Bruit coloré : Il est caractérisé par une représentation spectrale, où le signal aléatoire est appelé bruit de coloré, le bruit rose et le bruit brun font partie des types de bruit coloré.

Bruit musical : Le but de l'utilisation des algorithmes de soustraction spectrale ou de filtrage Wiener (algorithmes d'atténuation spectrale à court terme) est de réduire le bruit. Ce qui se conduit à un bruit résiduel gênant ce qui représente le bruit musical.

Bruit ambiant : La plupart des sons ´émis par toutes les sources proches et ´éloignées composent ce bruit. Alors ce bruit représentant la somme du bruit spécial émis par la source et du bruit restant.

Bruit impulsif : C'est un bruit très gênant pour transmettre des données, car il apparaît sous la forme d'une tension gênante pendant une courte période de valeur élevée, ainsi la forme du signal reçu est modifiée à tout moment en raison de ce signal gênant.

1.4.3 Rapport signal sur bruit (RSB)

Le RSB mesure la quantité de bruit dans le signal utile. La qualité de transmission est qualifiée en raison du quotient de division de la puissance du signal utile PS par la puissance du signal de bruit PN [19].

Le RSB est exprimé en décibels dB, il est donné par la relation suivante :

$$RSB = 10log_{10}(\frac{P_S}{P_N})$$

1.5 Conclusion

Dans ce chapitre, nous avons donnés la définition et le principe de la DAV et nous avons donnés aussi une brève description sur des différents techniques d'extraction des caractéristiques d'un signal, ainsi nous avons introduits quelques notions de base sur le traitement de signal. Dans le chapitre suivant, nous présenterons une généralité sur l'IA (Deep Learning & Maching Learning) et aux réseaux de neurones convolutifs.

Chapitre 2 Généralités sur l'intelligence artificielle et l'apprentissage profond, (ConvNet ou CNN).

2.1 Introduction

L'apprentissage en profondeur (Deep Learning) est un nouveau domaine de recherche du ML, qui a été introduit dans le but de rapprocher le ML de son objectif principal : l'intelligence artificielle. Il concerne les algorithmes inspirés par la structure et le fonctionnement du cerveau. Ils peuvent apprendre plusieurs niveaux de représentation dans le but de modéliser des relations complexes entre les données.

Pour clarifier davantage, dans ce chapitre, nous traitons d'une étude semi-approfondie, nous étudions des concepts de base des réseaux de neurones qui nous amènent à présenter l'apprentissage profond et à expliquer son importance et les différentes architectures qui le composent.

2.2 Intelligence artificielle

Aussi connue sous l'abréviation « IA », fait partie du domaine informatique .elle consiste à doter un système informatique d'un certain degré d'intelligence afin qu'il puisse effectuer une ou plusieurs tâches de façon autonome et automatique en suivant un raisonnement logique.

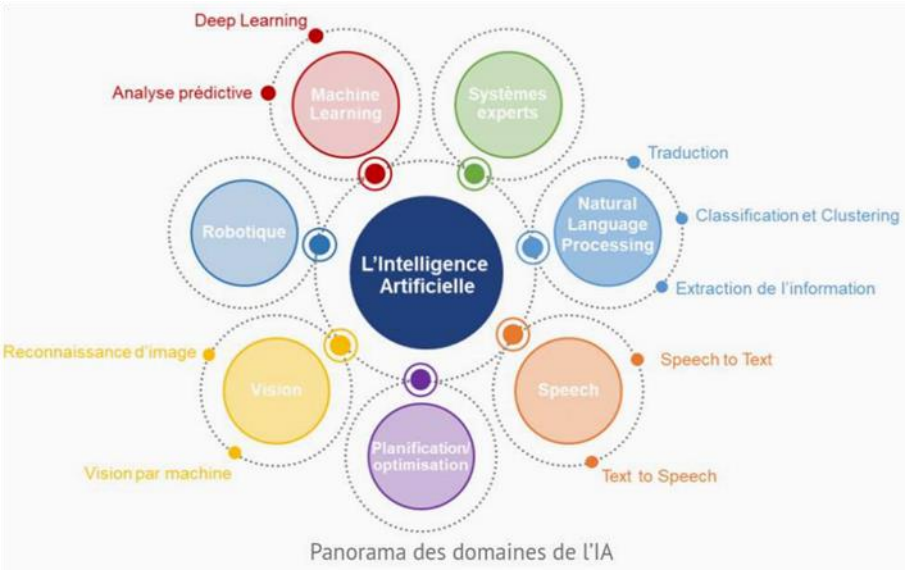


Figure 2-1 : Les domaines de l'intelligence artificielle

Il existe plusieurs domaines dans intelligence artificielle et parmi ces domaines :

2.2.1 Systèmes experts

Soit un logiciel capable de simuler le comportement d'un humain effectuant une tâche très précise. C'est un domaine où l'intelligence artificielle est incontestablement un succès, dû au caractère très précis de l'activité demandée à simuler.

2.2.2 Calcul formel (opposé au calcul numérique)

Traiter les expressions symboliques. Des logiciels sur le marché, comme Mathématique, Maple, etc., effectuent tous des calculs formels.

2.2.3 La Swarm Intelligence

La Swarm Intelligence ou intelligence distribuée est le comportement collectif de systèmes naturels ou artificiels décentralisés et auto-organisés. Plus précisément, il s'agit généralement d'un comportement collectif résultant des interactions locales entre plusieurs individus ou avec leur environnement. On trouve aussi parmi les domaines que nous avons cités le Machine Learning (ML).

2.3 Machine Learning (ML)

L'Apprentissage Automatique est un sous-domaine de l'Intelligence Artificielle (IA). C'est une technique de création d'un modèle en se reposant sur des données, les données peuvent être, entre autres, des images, du son ou du texte [20].

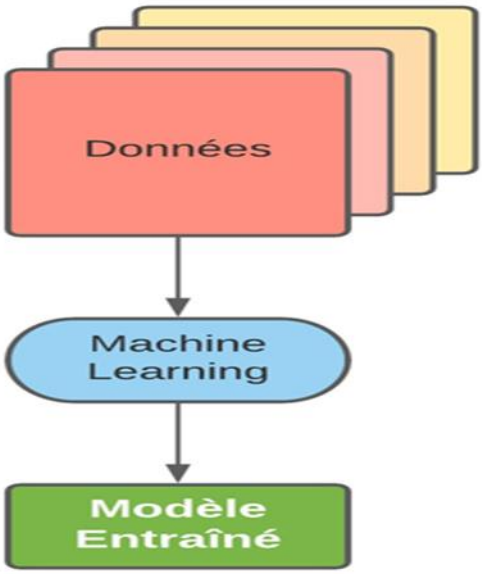


Figure 2-2 : Processus d'apprentissage en Machine Learning.

Il existe quatre familles de type d'approches utilisées en ML :

• Apprentissage basé sur l'information.

• Apprentissage basé sur la similarité.

• Apprentissage basé sur l'erreur.

• Apprentissage basé sur la probabilité.

On trouve trois types de l'apprentissage automatique :

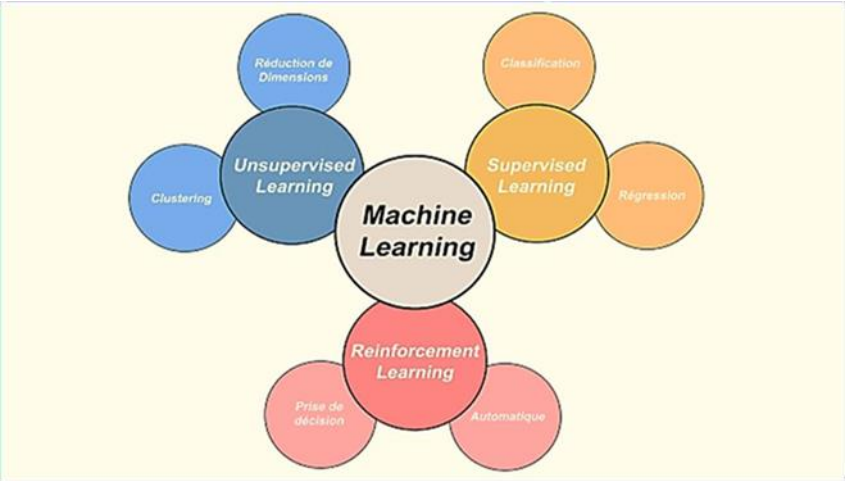


Figure 2-3 : Les types de l'apprentissage automatique.

2.3.1 Apprentissage supervisé

Pour cet apprentissage, nous avons des données en entrée (Features) et le résultat attendu (Label). Il nous permet de faire des prédications basées sur un modèle qui est obtenu partir de données d'historique et de l'algorithme choisi.

2.3.2 Apprentissage non-supervisé

Avec cet apprentissage on a toujours des features, mais pas de label, car nous n'essayons pas de prédire quoi que ce soit, il sert généralement à découvrir des structures et des modèles dans les données.

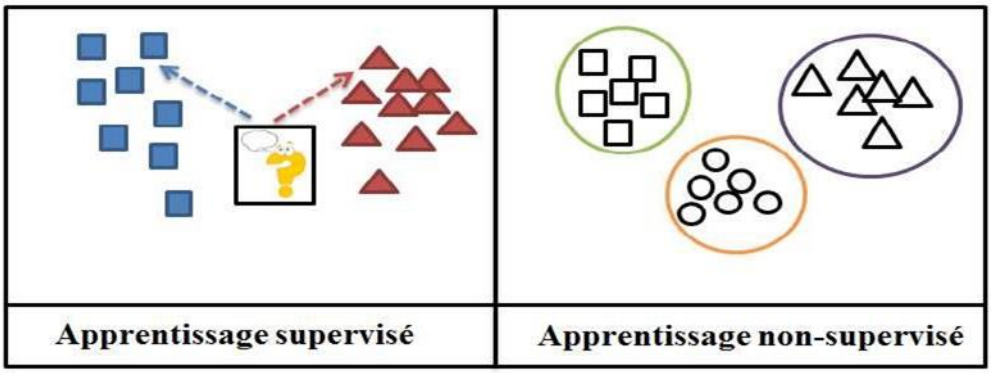


Figure 2-4 : L'apprentissage supervisé et non supervisé.

2.3.3 Apprentissage par renforcement

Dans ce type d'apprentissage, l'algorithme apprend le comportement à partir d'observations. L'effet de l'algorithme sur l'environnement crée une valeur de flux qui pilote l'algorithme d'apprentissage. Ce type est souvent utilisé en théorie, robots et véhicules autonomes.

2.4 Deep Learning (DL)

Le Deep Learning (DL) est une discipline qui apparaît au Machine Learning (ML), qui à son tour fait partie de l'Intelligence Artificielle (IA). Le DL consiste à modéliser et entraîner un réseau de neurones profond. Un réseau de neurones profond est composé d'une couche d'entrée où les données sont insérées dans le réseau. Une couche de sortie où les prédictions sont faites. Et finalement une ou plusieurs couches intermédiaires entre la couche d'entrée et la couche de sortie, appelées aussi couches cachées.

2.5 Neurone biologique

Le système nerveux central contient des cellules appelées neurones, dont le nombre dépasse 10^{12} neurones, ce qui signifie l´´équivalent d'un plus de 1000 milliards de neurones. Il se compose de trois composants de base, qui sont illustrés dans la figure (2.6).

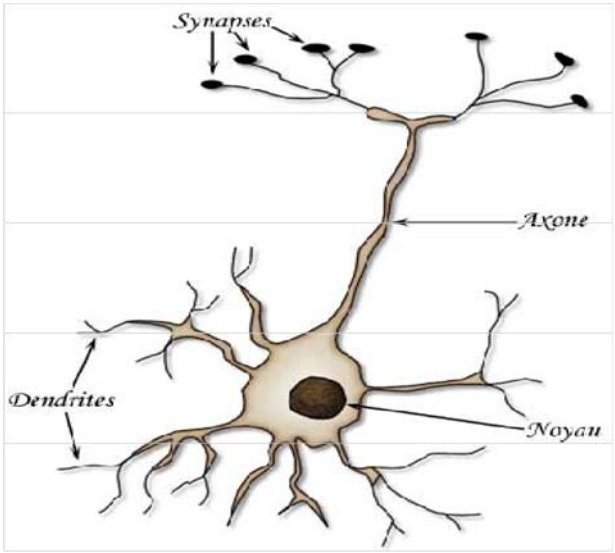


Figure 2-5 : Les éléments constituants le neurone biologique.

a) Synapses : Le corps cellulaire un noyau, qui est le principal responsable de la réalisation de transformations biochimiques qui aide à la synthèse de certains éléments qui comportent la vie du neurone.

b) Dendrites : Les dendrites permettent aux neurones de capter les signaux de l'extérieur.

c) Axone : L'axone se caractérise par sa longueur par rapport aux ramifications, il se ramifie à son extrémité et se connecte à d'autres neurones, cette structure particulière lui permet de transmettre des signaux émis par les neurones.

Pour transférer des informations, il existe une seule voie, qui va des dendrites aux axones. Le+ neurone reçoit les informations des autres neurones, à travers les dendrites, ces informations sont importées et consommées par le corps cellulaire.

2.6 Le perceptron

Le perceptron ou neurone artificiel est une conception artificielle inspirée du neurone biologique et mathématiquement est une fonction. L'analogie entre le neurone biologique et le neurone artificiel peut être décrite comme étant les dendrites sont les entrées Xi, l'axone est la sortie Y et les forces synaptiques sont les poids Wi.

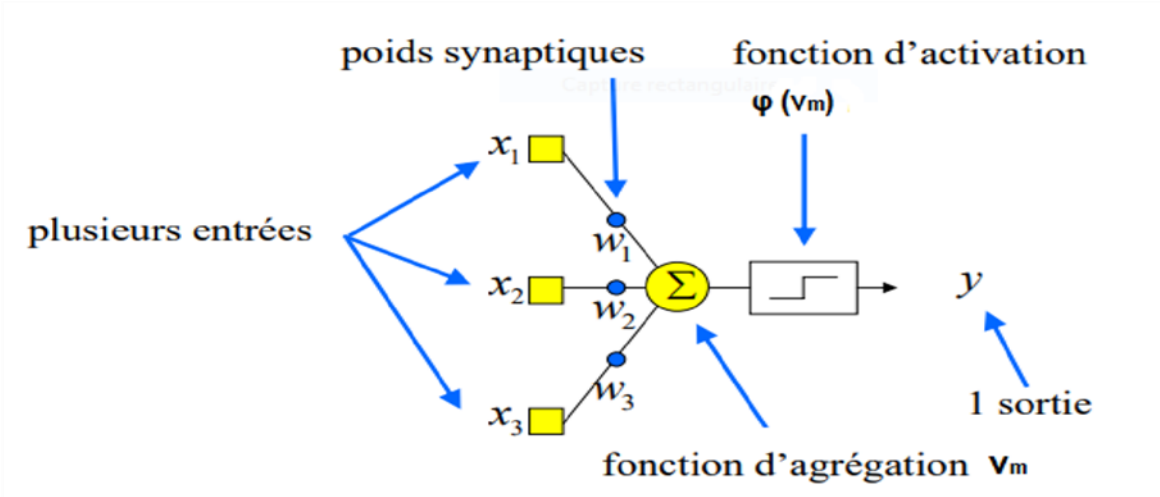


Figure 2-6 : Neurone Artificiel.

2.6.1 Le poids synaptique

Le poids synaptique (PS) représente le poids de la relation entre deux neurones connectés, autrement dit la probabilité d'obtenir une réponse de l'élément Post-Synaptique (fonction d'activation) en fonction de l'entrée, venant de l'élément Pré-Synaptique (poids synaptique). On le retrouve dans tous les réseaux de neurones modélisée et biologiques.

2.6.2 La fonction d'agrégation

En traitement des bases de données, la fonction d'agrégation ou la fonction de combinatoire est un opérateur permettant de réduire des groupes de lignes à une valeur calculée à partir de l'une des colonnes en jeu. Les fonctions les plus simples sont : la somme, la moyenne, le maximum et le minimum des valeurs.

2.6.3 La fonction d'activation

La fonction d'activation est une fonction mathématique qui simule l'activité et le comportement des neurones, elle est appliquée à un signal en sortie d'un neurone artificiel. Parmi les fonctions d'activation que nous utilisons :

a. Fonction sigmoïde

La fonction sigmoïde est l'une des fonctions d'activation non linéaires les plus importantes et les plus couramment utilisées, limitant les valeurs à varier entre 0

et 1 de sorte que si les valeurs sont des nombres positifs, le résultat est 1, et si les valeurs sont des nombres négatives ou nulles, le résultat à 0, comme illustré par la figure (2.8) [21].

La fonction sigmoïde est donnée par la relation suivante :

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

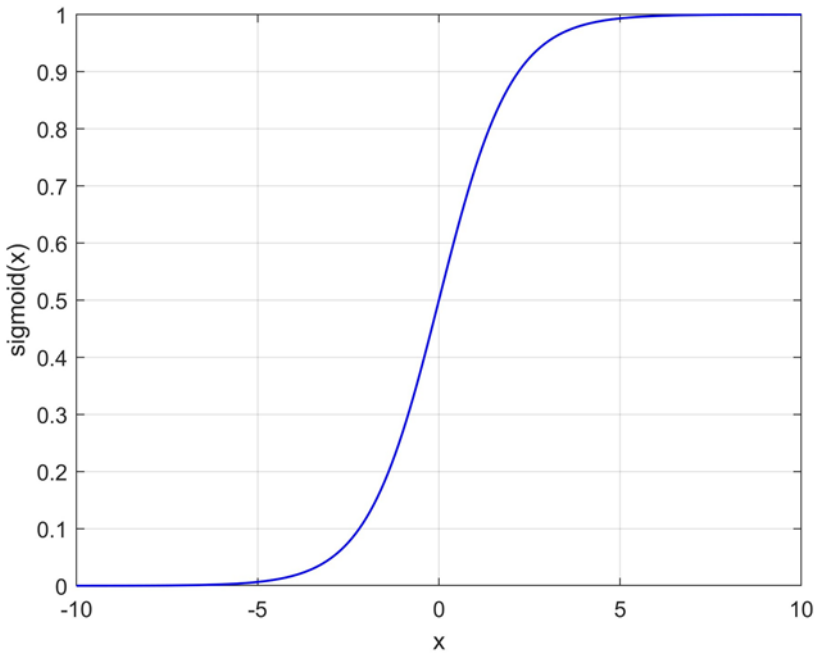


Figure 2-7 : la courbe de la fonction sigmoïde.

b. Fonction Tangente hyperbolique (tanh)

La fonction tangente hyperbolique est une version similaire `a la fonction sigmoïde sauf qu'au point de sortie est spécifié entre -1 et 1, comme montre la figure (2.9) [21].

La fonction tangente hyperbolique connue par la relation :

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

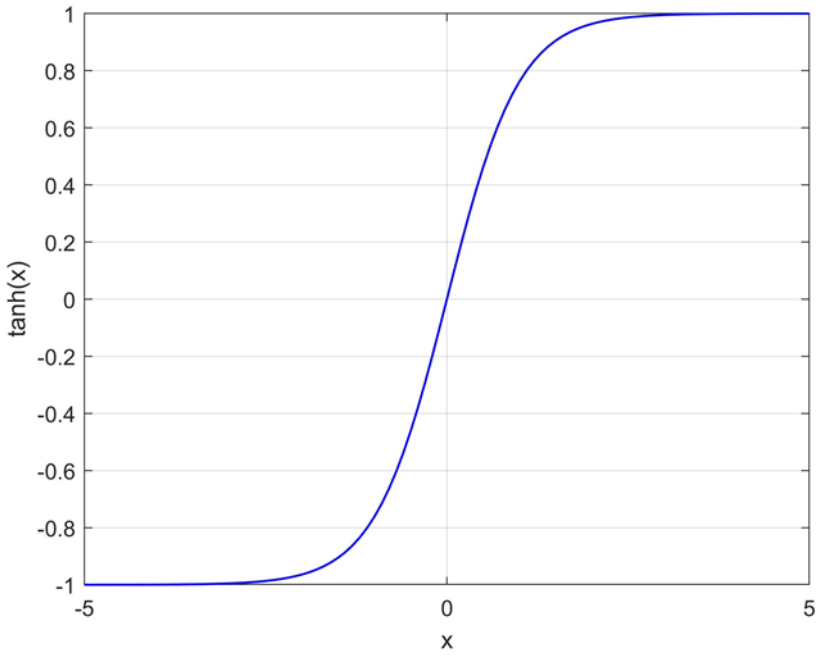


Figure 2-8 : la courbe de la fonction tangente hyperbolique.

c. Fonction Unité Linéaire Rectifiée (Rectified Linear Unit/ReLU)

Le résultat de cette fonction est 0 si les valeurs d'entrée sont négatives, au contraire les valeurs de sortie restent les mêmes que les valeurs d'entrée, comme illustrée par la figure (2.10) [21].

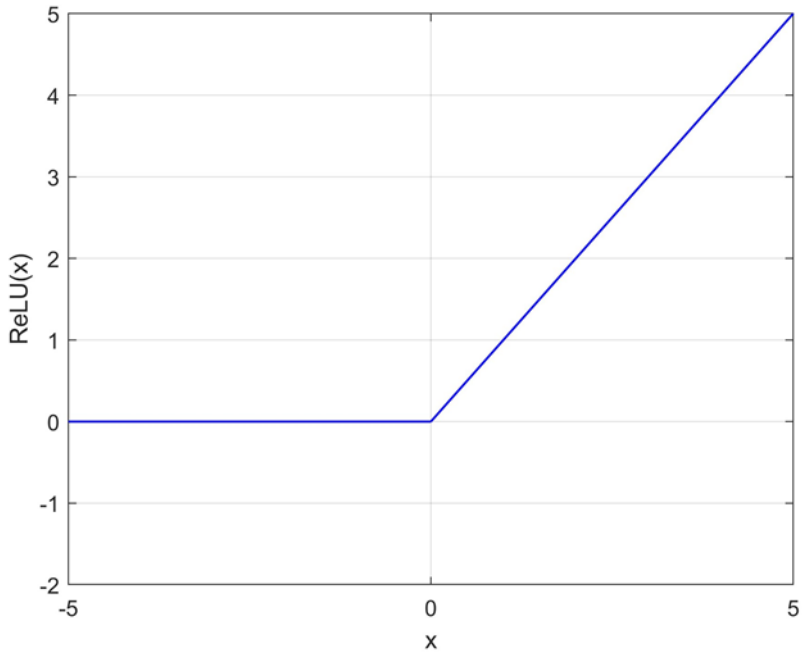


Figure 2-9 : la courbe de la fonction ReLU

La fonction ReLU est donnée par la relation mentionnée ci-dessous :

$$f(x) = max(0, x)$$

Où

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

d. Fonction softmax

La fonction softmax est une régression logistique dans un cas si nous voulons aborder plusieurs classes, elle est utilisée dans les problèmes de multi-classification [21].

Softmax est calculé avec la formule ci-dessous :

$$f(x_i) = \frac{e^{xi}}{\sum_{i=1}^{k}(e^{xe})}$$

Où K est le nombre des classes.

2.6.4 Perceptron simple

Le perceptron simple ou monocouche, a été développé par Rosenblatt, composé d'un neurone, comme illustré dans la figure, il est défini par une fonction d'activation comme suit [22] :

$$f(x) = \begin{cases} 1, si \ \ y > 0 \\ 0 \ (ou -1), si \ \ y \leq 0 \end{cases}$$

Ou y est le produit scalaire des entrées avec les poids, c'est—dire :

$$\sum_{i=1}^{n}(w_i \, x_i \mathrm{Y}=\overrightarrow{w} \bullet \vec{x}=)$$

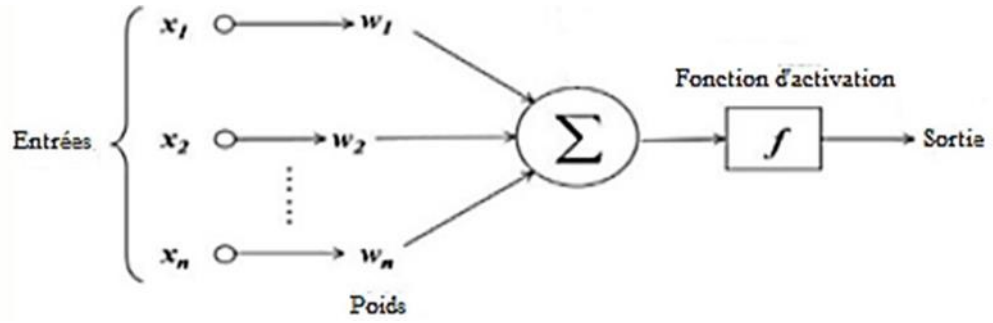


Figure 2-10 : Architecture d'un neurone simple.

2.6.5 Le Perceptron multicouches (PMC)

Les réseaux multicouches (MLP) sont aujourd'hui les modèles les plus employés. Ce dernier se compose d'une couche d'entrée, au moins une couche cachée en plus d'une couche de sortie, comme illustrée dans la figure (2.12).

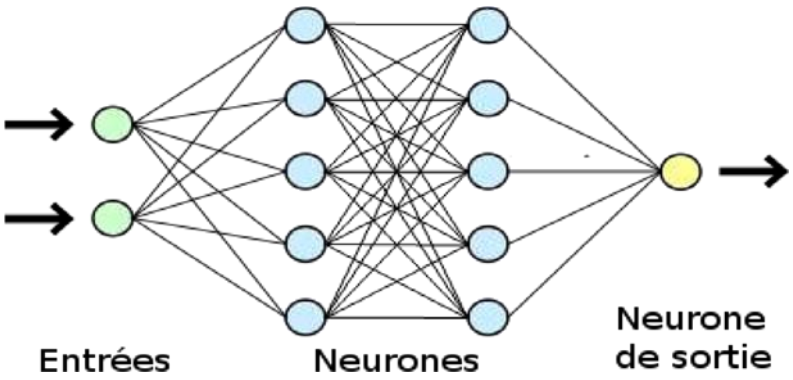


Figure 2-11 : Réseaux de neurone multicouches.

2.7 Propagation directe

Durant l'apprentissage, faire propager les données depuis l'entrée d'un réseau de neurones à sa sortie est appelé : "Propagation directe". Les sorties de chaque couche du réseau sont calculées en fonctions de ses poids et des sorties de la couche précédente et ce jusqu'à la dernière couche de sortie Propagation directe.

2.8 L'algorithme de rétro-propagation

Durant l'apprentissage d'un réseau de neurones, il est généralement nécessaire d'effectuer une opération appelée : "Backpropagation" ou "Delta Rule", en français

"Rétro-propagation". Celle-ci permet de mettre à jour à chaque itération les valeurs des poids connectés entre les nœuds (neurones). La valeur à ajouter ou à soustraire de la valeur existante d'un poids dépend de la valeur de l'erreur calculée lors de la propagation et du critère d'optimisation

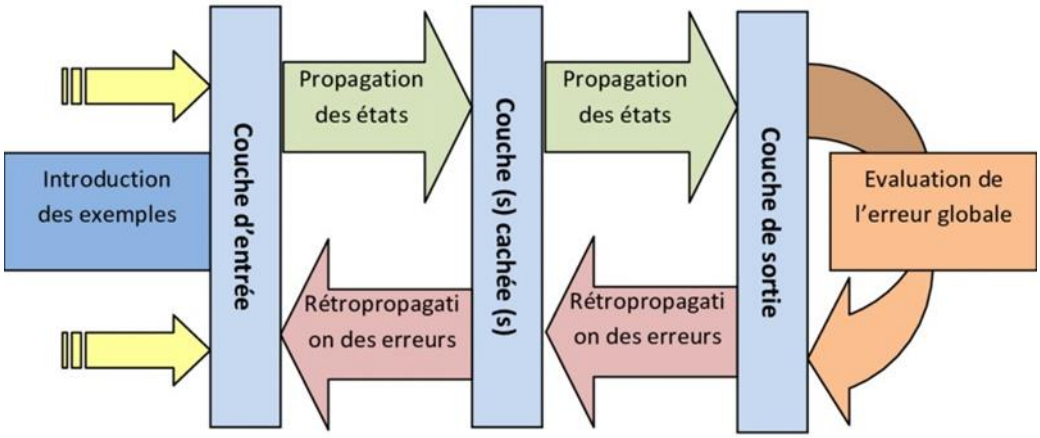


Figure 2-12 : apprentissage des réseaux de neurone par l'algorithme de rétropropagation

2.9 Réseaux de neurones convolutifs (ConvNet ou CNN)

Les réseaux de neurones convolutifs ou Convolutional Neural Networks (CNN) est le type de réseaux de neurones le plus utilisés en vision artificielle. Leur particularité réside dans le fait qu'ils utilisent des filtres de convolution afin d'extraire les caractéristiques d'une image. Les couches de convolution peuvent apparaître à n'importe quelle couche du réseau. Les CNN sont utilisés pour la reconnaissance et la classification, l'analyse vidéo, recherche de médicaments et aussi la segmentation.

Cette architecture est déterminée par la dimension de la couche d'entrée, le nombre, l'ordre et la nature des couches composant le réseau de neurones profond. Les CNN peuvent être implémentés suivant une architecture précise. La dimension dès la couche d'entrée, la succession, l'ordre et la nature des couches qui compose le réseau profond détermineront le type d'architecture. Chaque architecture est plus ou moins spécifique pour une application donnée.

2.9.1 Les couches d'un réseau de convolution

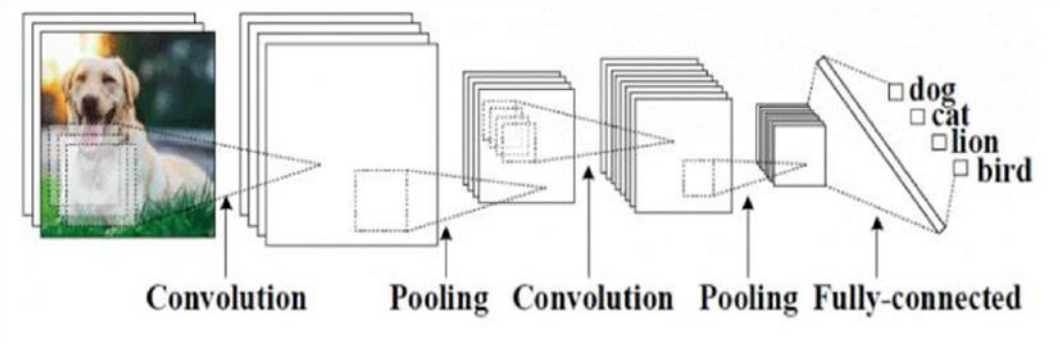


Figure 2-13 : les différentes couches d'un réseau convolutif.

a. Couche convolutive

La couche de convolution est la couche la plus importante des réseaux de neurone convolutif, Il est prévu d'appliquer un filtre de convolution à l'image pour découvrir les caractéristiques de l'image. Une image passe à travers une succession de filtres, créant de nouvelles images appelées cartes de caractéristiques ("Feature map output") selon un certain nombre de filtres ("Kernels"en anglais). Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. La sortie est une matrice, chacun de ses éléments est calculé à partir de la somme des multiplications des éléments du filtre par les éléments de la matrice d'entrée (souvent une image), chaque élément du résultat est trouvé en fonction de la position du filtre [23].

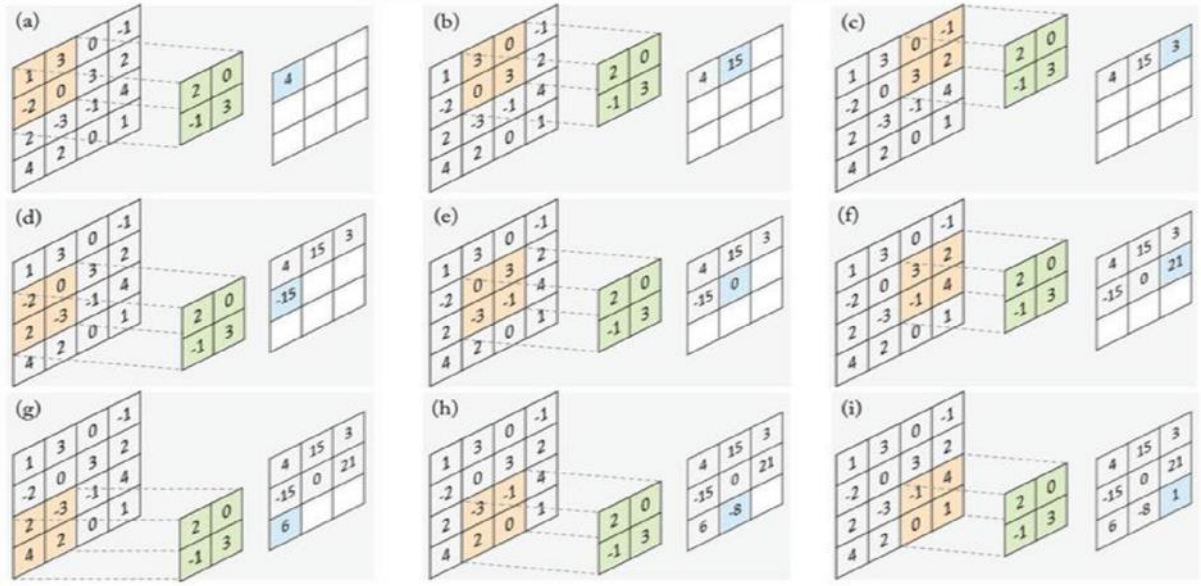


Figure 2-14 : L'opération de convolution avec filtre 2×2 stride 1.

Il consiste à diminuer le nombre d'échantillons à traiter (réduire la dimension de la matrice), elle consiste à faire déplacer à chaque étape le filtre d'un certain nombre de crans, horizontalement et verticalement, ce nombre est appelé Stride, l'augmentation de stride fait réduire la dimension de la matrice de sortie. Sur la Figure précédente le Stride égale à 1 parce que le filtre est glissé d'un seul cran à chaque étape.

La figure suivante nous montre la différence entre un Stride de 1 et un Stride de 2 :

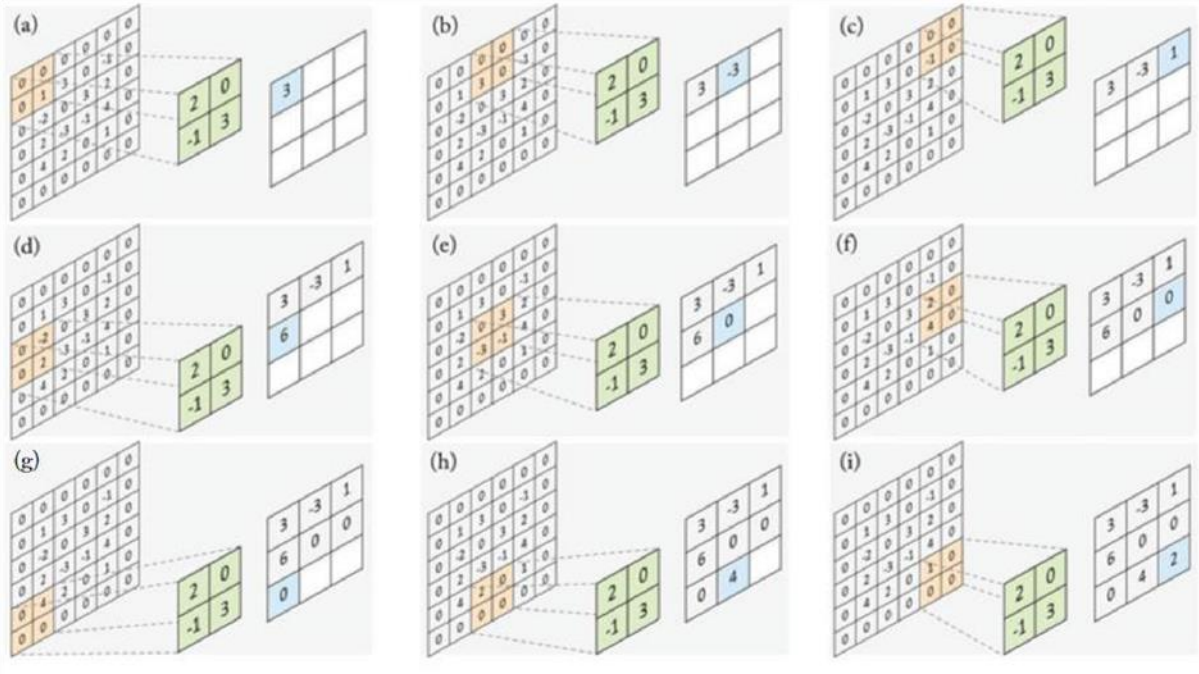


Figure 2-15 : L'opération de convolution avec filtre 2×2 stride 2.

b. La couche de normalisation

La couche de normalisation applique la normalisation à ces données. En particulier, cette couche assure que les données ont la même distribution. Nous pouvons imaginer normaliser nos données afin qu'elles tombent toutes sur une échelle de 0 à 1. En appliquant la normalisation à nos données après chaque niveau, le modèle peut apprendre plus facilement et avec moins d'erreurs [23].

c. La couche Pooling

La couche Pooling est une technique de sous-échantillonnage qui permet de réduire la taille de l'entrée. Semblablement à l'opération de convolution, le Stride s et la taille t de la région où le Pooling est opéré doivent être spécifiés. La sortie est calculée à partir du maximum ou de la moyenne de la région de Pooling.

Si la taie de l'entrée est w × h de la région de Pooling est t_w×t_h, le Stride est notée s, la taille de la matrice de sortie sera :

$$w\_Pooling=\left[\frac{w-t_w+s}{s}\right]$$

$$h\_Pooling=\left[\frac{h-t_h+s}{s}\right]$$

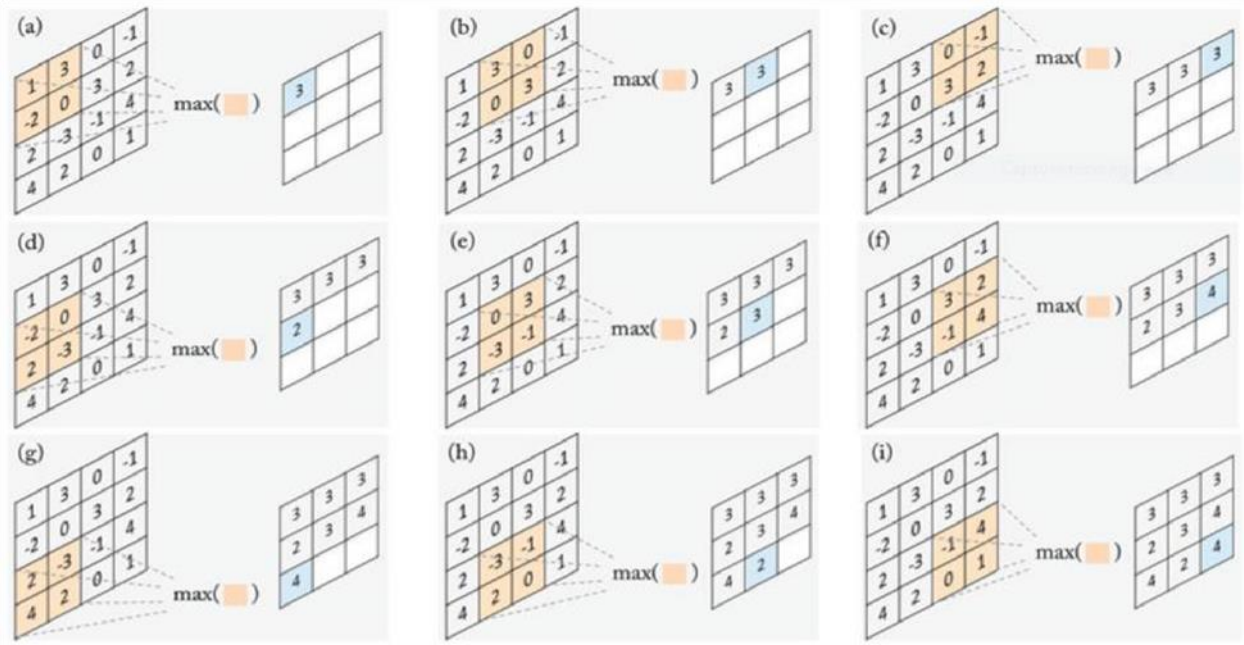


Figure 2-16 : L'opération de Max Pooling.

Le Max Pooling

Le Max Pooling calcule les éléments de la sortie en retenant uniquement la valeur maximale de la région de Pooling.

Le Mean Pooling

Le Mean Pooling est l'opération qui permet de calculer chaque élément de la sortie par rapport via une région adjacente de l'entrée.

d. La couche Fully Connected

La couche "Fully Connected" ou entièrement connectée est une couche qui se compose d'un empilement de neurones, chacun est alimenté par des entrées provenant des sorties de la couche précédente et sa sortie est connectée à tous les neurones de la couche suivante :

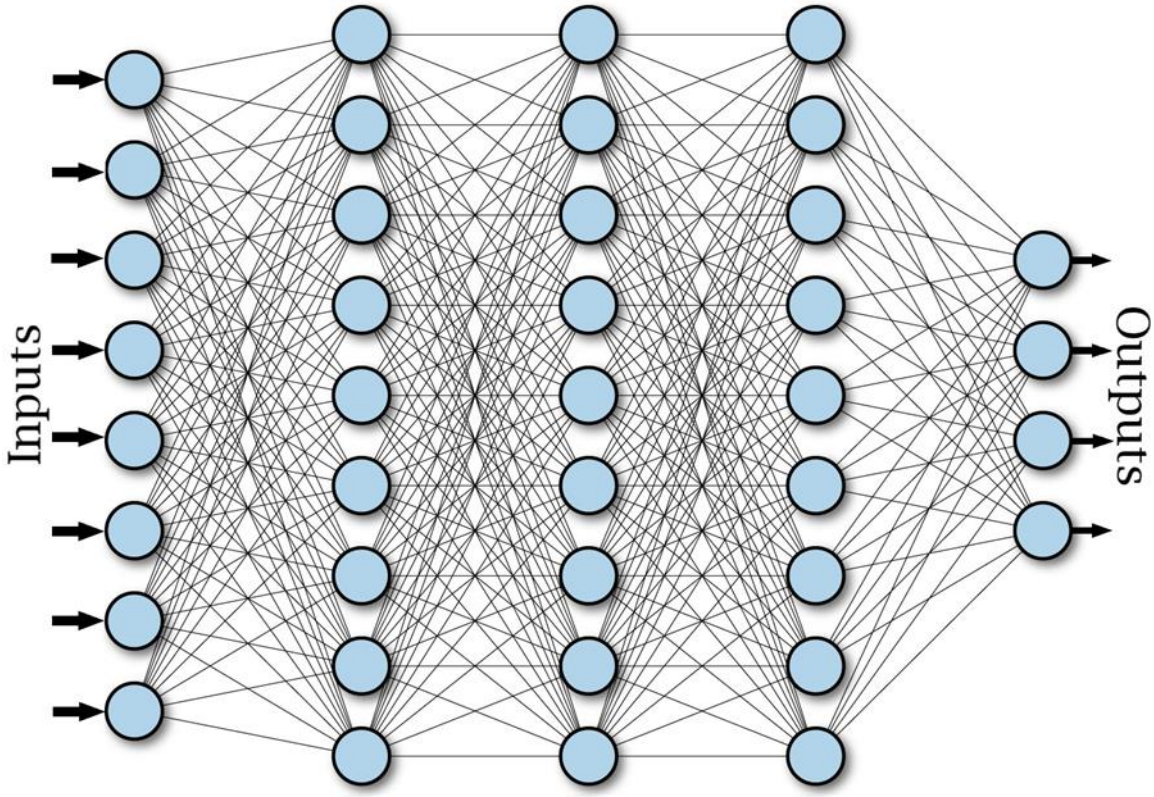


Figure 2-17 : Réseau profond multicouche avec une couche d'entrée et trois couches cachées Fully Connected et une couche de sortie.

Les couches Fully Connected peuvent être activées par une multitude de fonctions d'activation.

2.9.2 Les bases de données

L'apprentissage des CNN nécessite une importante quantité de données, ce sont les "Datasets" (bases de données). Pour des applications en vision artificielle (vision par ordinateur), elles sont généralement organisées sous forme d'images enregistrées sous un format de fichiers spécifiques, et de "Labels" (dans le cas d'un apprentissage supervisé) qui désignent la classe correspondante à l'image. Chaque image est associée à son Label (Classe).

Les images sont généralement divisées en deux catégories, images d'apprentissage et images de test.

Chaque base de données a sa capacité (Nombre d'images conservées) et son nombre de classes (Nombre de Labels).

2.9.3 Quelques architectures des réseaux de neurones convolutifs

a. L'architecture LeNet-5

Appelé aussi "LeNet-5", cette architecture est utilisée pour la reconnaissance des caractères décimaux manuscrits de 0 jusqu'à 9. Elle est composée d'une couche d'entrée prenant des données de dimensions 32×32 avec une seule chaine.

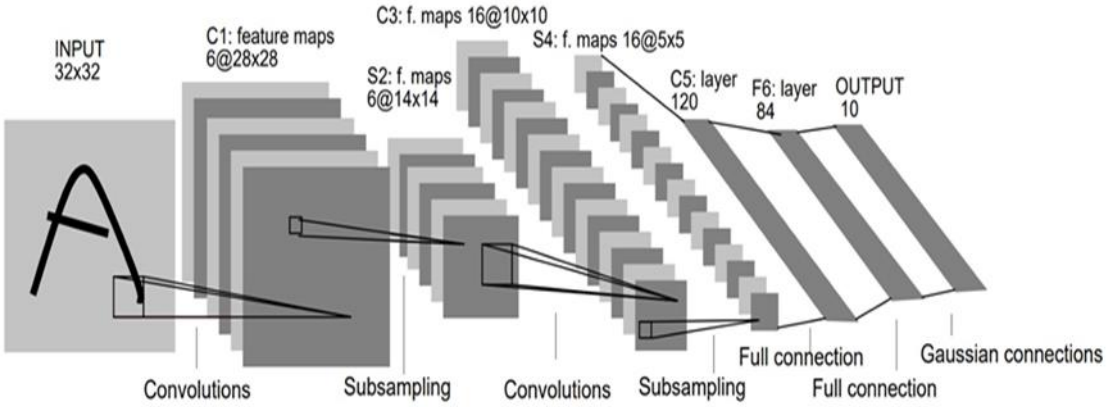


Figure 2-18 : Architecture de LeNet-5.

b. **L'architecture AlexNet**

AlexNet est la première architecture de CNN à grande échelle. Elle a abouti à un redémarrage des réseaux de neurones profonds, elle a gagné la compétition ILSVRC en 2012. Elle prend des images de 22×224×3 (en RGB) et contient huit couche dont les cinq premières sont des couches de convolution et les trois dernières sont des couches Fully Connected [24]. Cette architecture se représente comme suite :

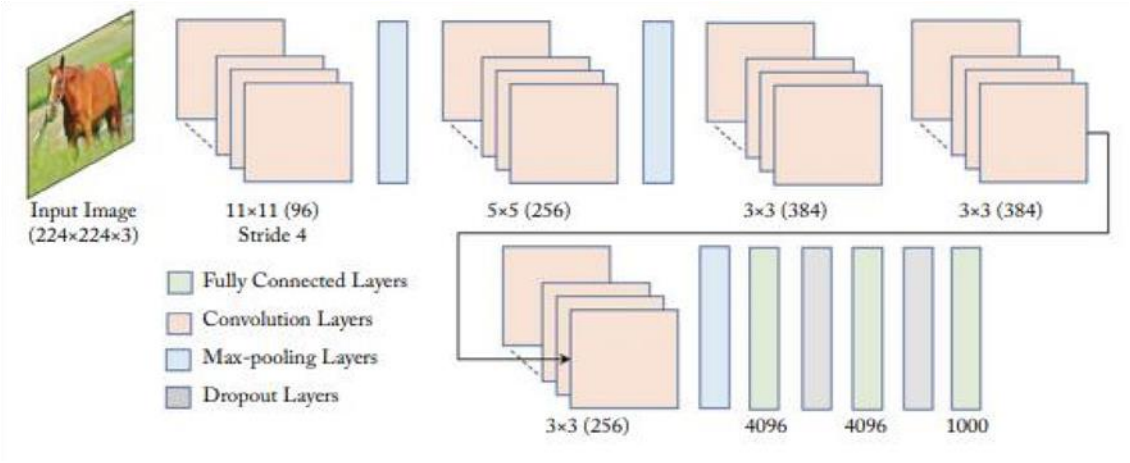


Figure 2-19 : Architecture d'AlexNet

c. L'architecture VGGNet

VGGNet est l'une des architectures les plus populaires, elle est proposée en 2014 par K.Simonyan et A.Zisserman avec une utilisation stricte de filtres de convolution de taille 3×3 avec une profondeur de 16 et 19 couches communément appelées : VGGnet-16 et VGGnet-19 respectivement. Ci-après, la Figure illustre l'architecture VGGnet-16 avec treize couches de convolution et trois couches Fully Connected [25].

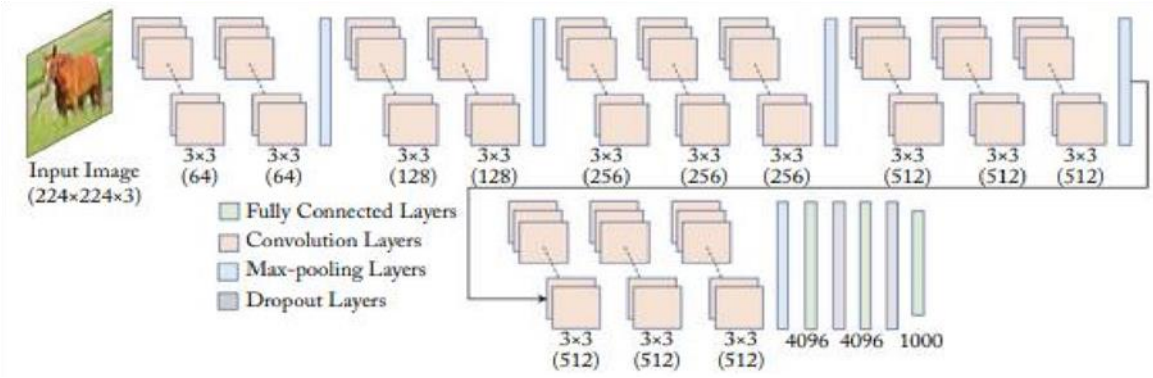


Figure 2-20 : L'architecture VGGNet.

2.9.4 Choix des hyperparamétres

Les CNNs utilisent plus d'hyperparamètres qu'un MLP standard. Même si les règles habituelles pour les taux d'apprentissage et des constantes de régularisation s'appliquent toujours, il faut prendre en considération les notions de nombre de filtres, leur forme et la forme du max pooling.

a. Nombre de filter

Comme la taille des images intermédiaires diminue avec la profondeur du traitement, les couches proches de l'entrée ont tendance à avoir moins de filtres tandis que les couches plus proches de la sortie peuvent en avoir davantage. Pour égaliser le calcul à chaque couche, le produit du nombre de caractéristiques et le nombre de pixels traités est généralement choisi pour être à peu près constant à travers les couches. Pour préserver l'information en entrée, il faudrait maintenir le nombre de sorties intermédiaires (nombre d'images intermédiaire multiplié par le nombre de positions de pixel) pour être croissante (au sens large) d'une couche à l'autre.

Le nombre d'images intermédiaires contrôle directement la puissance du système, dépend du nombre d'exemples disponibles et la complexité du traitement.

b. Forme du filtre

Les formes de filtre varient grandement dans la littérature. Ils sont généralement choisis en fonction de l'ensemble de données. Les meilleurs résultats sur les images de MNIST (28 x 28) sont habituellement dans la gamme de 5 × 5 sur la première couche, tandis que les ensembles de données d'images naturelles (souvent avec des centaines de pixels dans chaque dimension) ont tendance à utiliser de plus grands filtres de première couche de 12 × 12, voire 15 × 15.

Le défi est donc de trouver le bon niveau de granularité de manière à créer des abstractions à l'échelle appropriée et adaptée à chaque cas.

c. Forme du max pooling

Les valeurs typiques sont 2 × 2 (figure 2.22). De très grands volumes d'entrée peuvent justifier un pooling 4 × 4 dans les premières couches. Cependant, le choix de formes plus grandes va considérablement réduire la dimension du signal, et peut entraîner la perte de trop d'information.

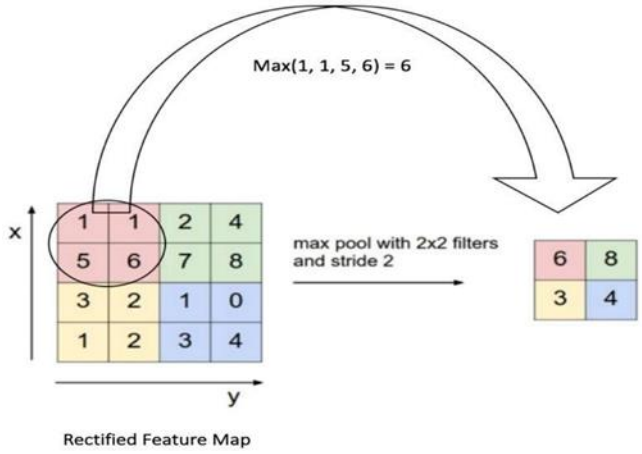


Figure 2-21 : Exemple de max pooling (2×2).

2.9.5 L'entraînement d'un nouveau CNN

La création d'un réseau de neurones convolutifs est une tâche difficile et coûteuse car elle nécessite une bonne expérience, du matériel et la quantité de données nécessaires. La première étape consiste à fondre l'architecture du réseau, c'est-à-dire le nombre de couches, la taille et les opérations matricielles qui les relient, puis la formation consiste à optimiser les paramètres du réseau pour réduire l'erreur de classification en sortie. Le temps d'exécution peut prendre plusieurs jours pour les meilleurs réseaux CNN car les unités de traitement graphique (GPU) fonctionnent sur des centaines de milliers d'images [26].

2.9.6 Avantage du CNNs

Un avantage majeur des réseaux convolutifs est l'utilisation d'un poids unique associé aux signaux entrant dans tous les neurones d'un même noyau de convolution. Cette méthode réduit l'empreinte mémoire, améliore les performances [27] et permet une invariance du traitement par translation. C'est le principal avantage du CNN par rapport au MLP, qui lui considère chaque neurone indépendant et donc affecte un poids différent à chaque signal entrant. Lorsque le volume d'entrée varie dans le temps (vidéo ou son), il devient intéressant de rajouter un paramètre de temporisation (delay) dans le paramétrage des neurones.

Comparés à d'autres algorithmes de classification de l'image, les réseaux de neurones convolutifs utilisent relativement peu de pré-traitement. Cela signifie que le réseau est responsable de faire évoluer tout seul ses propres filtres (apprentissage sans supervision), ce qui n'est pas le cas d'autres algorithmes plus traditionnels. L'absence de paramétrage initial et d'intervention humaine est un atout majeur des CNN.

2.10 Les métriques de mesure de la performance des modèles

L'évaluation des performances des modèles est une tâche critique et complexe à la fois. Par conséquent, cela doit être fait avec soin afin que les résultats rapportés soient fiables. Cette section explique comment nous pouvons évaluer les résultats de notre modèle, ce qui rend un modèle meilleur qu'un autre. Plusieurs métriques ont été proposées pour évaluer la performance prédictive des problèmes de régression et de classification.

2.10.1 LOSS

La fonction LOSS est un élément essentiel de l'entraînement du modèle. Elle quantifie la qualité d'exécution d'une tâche par un modèle en calculant un seul nombre. LOSS Si les prédictions du modèle sont totalement erronées, la perte sera un nombre élevé. S'ils sont plutôt résulta bons, ce sera proche de zéro.

$$Loss == -\log(Y_{Pred})$$

2.10.2 RMSE (Root Mean Square Error)

L'erreur quadratique moyenne (RMSE) est une formule populaire pour mesurer le taux d'erreur d'un modèle de régression. Cependant, il ne peut être comparé qu'entre des modèles dont les erreurs sont mesurées dans les mêmes unités.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}}$$

a=Cible réelle.

p =Cible prévue.

2.10.3 MAE (Mean Absolut error)

L'erreur absolue moyenne a la même unité que les données d'origine et ne peut être comparé qu'entre des modèles dont les erreurs sont mesurées dans les mêmes unités. Son ampleur est généralement similaire à celle du RMSE, mais légèrement plus petite. a et p sont défini dans l'erreur quadratique moyenne.

$$MAE = \frac{\sum_{i=1}^{n} |p_i - a_i|}{n}$$

2.10.4 La matrice de confusion

Il s'agit d'un tableau de taille n × n pour visualiser les résultats des modèles prédictifs dans les problèmes de classification, où n est le nombre de classes dans l'ensemble de données (voir tableau ci-dessous). Dans cette matrice on croise les classes cibles réelles avec les classes prédites obtenues. Ceci nous donne le nombre d'instances correctement classées et mal classées.

Table 2-1 : Matrice de confusion pour une classification binaire

		Classes actuels	
		Positive	**Négative**
Classes prédites	**Positive**	VP	FP
	Négative	FN	VN

- VP : vrais positifs est le nombre d'instances positives correctement classifiées.
- FP : faux positifs est le nombre d'instances négatives et qui sont prédites comme positives.
- FN : faux négatifs est le nombre d'instances positives classifiées comme négatives.
- VN : vrais négatifs est le nombre d'instances négatives correctement classifiées.

À partir de la matrice de confusion on peut calculer plusieurs métriques, parmi eux :

2.10.5 Accuracy

Est une métrique de classification. L'exactitude est proportion de prédictions correctes, c'est-à-dire les vrais positifs et les vrais négatifs, parmi le nombre total de prédication. Ce nombre est donc compris entre 0 (parfaite inexactitude) et 1 inclus (parfaite exactitude).

$$Accuracy = \frac{(VP + VN)}{(VP + VN + FP + FN)}$$

2.10.6 Recall

La sensitivité ou le rappel (Recall) est le pourcentage des instances positives correctement identifiées.

$$Recall = \frac{VP}{(VP + FN)}$$

2.10.7 Precision

La précision (Precision) est le pourcentage de prédictions positives qui sont correctes.

$$Precision = \frac{VP}{(VP + FP)}$$

2.10.8 F1-Score

Le F1- Score est une métrique pour évaluer la performance des modèles de classification à 2 classes ou plus .il est particulièrement utilisé pour les problèmes utilisant des données déséquilibrées. Le F1-Score permet de résumer les valeurs de la Precision et du Recall en une seule métrique. Mathématiquement, le F1-Score est défini comme étant la moyenne harmonique de la Precision et du Recall, ce qui se traduit par l'équation suivante :

$$F1 - Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}$$

2.11 Conclusion

Dans ce chapitre, nous avons pu aborder la définition de l'intelligence artificielle ainsi que quelques domaines et parmi eux le Machine Learning (ML) et le Deep Learning (DL). Ensuite, nous avons présentes les principaux concepts de bases sur les réseaux de neurones artificiels. Puis nous avons décrit d'une manière générale les réseaux de neurones convolutifs (CNNs), les différentes opérations utilisées. Nous avons colorés ce chapitre avec les différentes architectures les plus populaires et les plus largement utilisées dans ce réseau.

Ce qui nous permet de commencer la mise en œuvre de notre étude, il est nécessaire de choisir une architecture et appliquer les opérations nécessaires qui présenteront dans le prochain et le dernier chapitre de notre travail.

Chapitre 3 Fonctionnement et résultats

3.1 Introduction

L'objectif de la détection d'activité vocale est d'identifier et de détecter la présence ou l'absence de signaux vocaux dans un flux audio. Il s'agit d'une tâche fondamentale dans le domaine du traitement du signal audio et de la reconnaissance de la parole.

La détection d'activité vocale trouve de nombreuses applications pratiques notamment : Systèmes de reconnaissance de la parole, Compression audio, Application de traitement audio, Applications de sécurité etc.

Nous avons utilisé l'apprentissage en profondeur dans cette étude car on vise à améliorer la précision, la robustesse et l'efficacité des modèles de détection, tout en ouvrant de nouvelles perspectives pour l'exploitation de la voix dans divers domaines d'application.

Initialement, nous abordons l'environnement de travail, la langue utilisée et les ressources mobilisées. Par la suite, nous détaillons les étapes de déploiement de l'algorithme suggéré, jusqu'aux résultats expérimentaux obtenus à l'issue des tests effectués.

3.2 Environnement de développement

3.2.1 Google Colab

Google Colab est un service cloud gratuit basé sur Jupyter Notebook et le langage de programmation Python, qui permet de développer des applications d'apprentissage profond sans être limité par les contraintes matérielles. Il offre la possibilité d'utiliser des bibliothèques telles que TensorFlow et Keras. Une

caractéristique distinctive de Colab par rapport aux autres services cloud gratuits est la mise à disposition gratuite d'une unité de traitement graphique (GPU - Graphic Processing Unit) [28] [29].

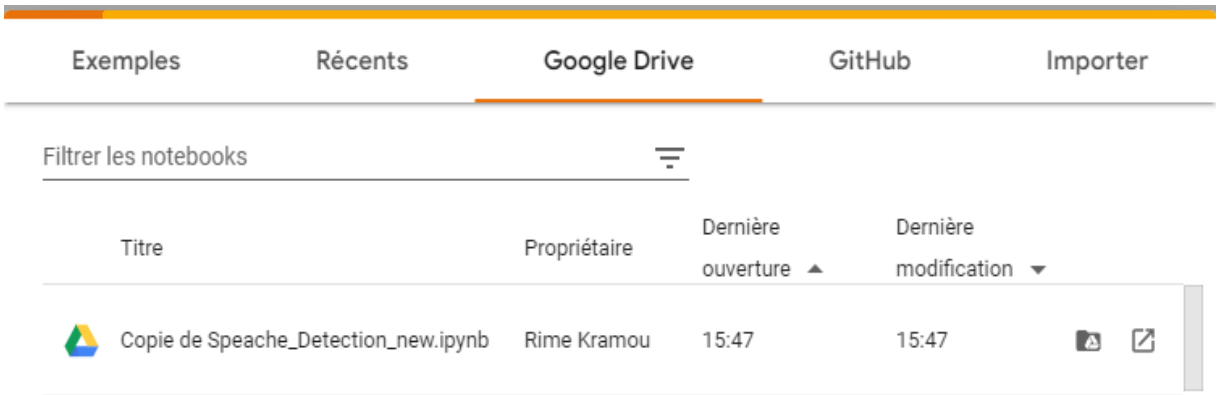


Figure 3-1 : Environnement de Google Colab

Nous comptons sur le GPU car il améliore les performances et effectue des calculs intenses.

3.2.2 Python

Python est un langage de programmation open source de haut niveau, interpréteur et Multi-paradigme pour la programmation générique, créé par Guido van Rossum, à paraître en 1991, préconisait une programmation impérative structurée, fonctionnelle et orientée objet. Il a un système de type dynamique et un Gestion automatique de la mémoire. L'interpréteur Python peut être utilisé pour de nombreux systèmes d'exploitation.

Le langage Python prend en charge les principaux aspects du cycle de vie de l'application. L'apprentissage automatique et l'apprentissage en profondeur car il a de nombreux Contribuer au développement des bibliothèques dans ce domaine, en plus de favoriser la gestion fichier audio [30].

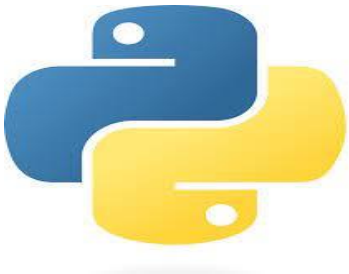


Figure 3-2 : logo python.

3.3 Bibliothèques utilisées

Librosa : est un progiciel python pour l'analyse de la musique et du son. Il fournit les éléments de base nécessaires à la création de systèmes de recherche d'informations musicales [31].

NumPy **:** est une bibliothèque fondamentale en python, pour traiter les tableaux homogènes multidimensionnels et aussi travailler avec l'algèbre linéaire [32].

Matplotlib : est une bibliothèque de traçage, qui peut représenter les données sous plusieurs visualisations [33].

Pandas : est une bibliothèque Python utilisée pour travailler avec des ensembles de données. Elle a des fonctions pour analyser, nettoyer, explorer et manipuler les données [34].

sklearn.preprocessing : fournit plusieurs fonctions d'utilité et classes de transformateurs communs pour modifier les vecteurs caractéristiques bruts en une représentation plus convenable pour les estimateurs en aval [35].

TensorFlow : est une bibliothèque de logiciels open source pour le calcul numérique puissant. Son architecture flexible permet de déployer facilement la puissance de calcul sur les plates-formes (CPU, GPU, TPU) et les ordinateurs de bureau, des clusters de serveurs aux appareils mobiles et périphériques [36].

Keras : est une API d'apprentissage en profondeur écrite en Python et exécutée sur la plate-forme d'apprentissage automatique TensorFlow. Il a été conçu pour permettre une expérimentation rapide. La clé d'une bonne recherche est de pouvoir passer de l'idée au résultat le plus rapidement possible [37].

3.4 Ensembles de données

L'ensemble de données des commandes vocales est une tentative de générer un ensemble de données standard d'entrainement et d'évaluation pour une détection vocale.

L'objectif est de fournir un moyen de construire et de tester de petits modèles qui détectent quand un seul mot est dit, d' un ensemble de dix mots cibles ou moins, avec aussi peu de faux positifs que possible du bruit de fond ou de la parole sans rapport. Cette tâche est souvent connue sous le nom de tâches de mots clés.

3.4.1 LibriSpeech

Représente un groupe de signaux de parole lisibles en anglais, dérivés de livres audio basés sur le projet LibriVox. Le but de leur utilisation est d'entrainer et d'évaluer les systèmes de reconnaissance automatique de la parole (RAP). Cet ensemble de données open source contient 1000 heures de parole [38].

3.4.2 TIDIGITS

Contient 25.000 séquences de chiffres par 300 haut-parleurs différents, enregistrés dans une salle tranquille par les contributeurs rémunérés. L'ensemble de données n'est disponible que sous une licence commerciale du consortium de données de mesure de réseau et est stocké au format de fichier NIST SPHERE, qui s'est avéré difficile à décoder en utilisant un logiciel moderne. Nos premières expériences sur le repérage de mots clés ont été réalisées à l'aide de cet ensemble de données [39].

3.4.3 CHiME-5

Enregistré 50 heures de discours dans les maisons des gens, stocké sous forme de fichiers WAV 16 kHz, et disponible sous une licence restreinte. C'est aligné au niveau de la phrase [40].

3.5 Evaluation et résultats

L'ensemble de données LibriSpeech est en format FLAC et l'ensemble de données CHiME-5 est en format WAV. Donc, le format diffère pour chaque ensemble de données, ce qui a conduit à standardiser les données, en convertissant l'ensemble de données LibriSpeech au format WAV, en utilisant pydub qui nécessite la bibliothèque ffmpeg.

L'ensemble de données final comprenait 105 829 déclarations de 35 mots, répartis en catégories et fréquences [41].

3.5.1 Propriété de données

Chaque énoncé est stocké en une seconde (ou moins) Fichier de format WAVE, avec les données de l'échantillon codées comme valeurs PCM linéaires 16 bits à canal unique, à 16 kHz taux. Il y a 2.618 haut-parleurs enregistrés, chacun avec un identificateur hexadécimal unique à huit chiffres attribué comme décrit ci-dessus. Les fichiers non compressés coûtent environ 3,8 Go sur le disque, et peut être stocké comme une archive tar compressée gzip de 2,7 Go [42].

3.5.2 Extraction de caractéristiques

Un DataFrame appelé "x_test" qui contient les caractéristiques audio extraites du jeu de données de test. Les caractéristiques audio sont stockées dans la variable "test_audio". Ensuite, un autre DataFrame appelé "df" est créé à partir des échantillons de test pour avoir une représentation tabulaire des données de test. La variable "y_test" est extraite du DataFrame "df" et contient les étiquettes de classe correspondantes aux échantillons de test. Enfin, la fonction "value_counts()" est utilisée pour compter le nombre d'occurrences de chaque classe dans la variable "y_test". Cela permet de connaître la répartition des classes dans le jeu de données de test.

3.5.3 Encodage

L'encodage one-hot des étiquettes de classe pour les ensembles d'entraînement, de validation et de test, ce qui permet de représenter les classes sous forme de

vecteurs binaires pour une utilisation ultérieure dans des modèles d'apprentissage automatique. A partir de notre code nous avant les résultats suivants.

- Pour l'ensemble d'entraînement (y_train), le résultat affiché est (1000, 2), ce qui signifie qu'il y a 1000 échantillons dans cet ensemble, et chaque échantillon est représenté par un vecteur binaire de longueur 2, où la position 0 indique la première classe et la position 1 indique la deuxième classe.
- Pour l'ensemble de validation (y_val), le résultat affiché est (132, 2), indiquant qu'il y a 132 échantillons dans cet ensemble et que chaque échantillon est représenté par un vecteur binaire de longueur 2.
- pour l'ensemble de test (y_test), le résultat affiché est (400, 2), ce qui signifie qu'il y a 400 échantillons dans cet ensemble, et chaque échantillon est représenté par un vecteur binaire de longueur 2.

3.5.4 Modèle CNN

La construction et l'entraînement d'un modèle de réseau neuronal convolutif (CNN) pour la classification d'activité vocale. La Création d'une instance de rappel d'arrêt anticipé (Early Stopping Callback), Ce rappel permet d'arrêter l'entraînement du modèle si la valeur de la fonction de perte (loss) sur l'ensemble de validation cesse de s'améliorer pendant un certain nombre d'époques défini par le paramètre patience. Restore_best_weights = True permet de restaurer les poids du modèle correspondant à la meilleure performance sur l'ensemble de validation.

1. Construction du modèle : Le modèle est construit en utilisant la classe Sequential de Keras, qui permet de définir un empilement linéaire de couches. Les couches ajoutées sont les suivantes :

- Une couche de convolution (Conv2D) avec 32 filtres, une taille de noyau de (3, 3), une fonction d'activation ReLU et une taille d'entrée de (16, 8, 1).
- Une couche de max pooling (MaxPooling2D) avec une taille de fenêtre de (2, 2).
- Une couche de dropout (Dropout) avec un taux de 0.25 pour régulariser le modèle et réduire le surapprentissage.

- Une deuxième couche de convolution avec 64 filtres, une taille de noyau de (3, 3) et une fonction d'activation ReLU.
- Une deuxième couche de max pooling.
- Une deuxième couche de dropout.
- Une couche de mise en forme (Flatten) pour aplatir les sorties des couches précédentes en un vecteur.
- Trois couches fully connected (Dense) avec des fonctions d'activation ReLU, des couches de dropout intermédiaires pour la régularisation, et une couche de sortie avec une fonction d'activation softmax pour la classification en deux classes.

2. Compilation du modèle : Le modèle est compilé en spécifiant la fonction de perte categorical_crossentropy pour la classification multiclasse, l'optimiseur Adam et la métrique d'évaluation de l'exactitude (accuracy).

3. Entraînement du modèle : Le modèle est entraîné en utilisant la méthode fit de Keras. Les données d'entraînement (x_train et y_train) sont fournies, ainsi que le nombre d'époques (epochs), le rappel d'arrêt anticipé (early_stopping_callback), la taille du batch (batch_size) et les données de validation (x_val et y_val).

```
[ ] Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 16, 8, 32)         320

 max_pooling2d (MaxPooling2D  (None, 8, 4, 32)         0
 )

 dropout (Dropout)           (None, 8, 4, 32)          0

 conv2d_1 (Conv2D)           (None, 8, 4, 64)          18496

 max_pooling2d_1 (MaxPooling  (None, 4, 2, 64)         0
 2D)

 dropout_1 (Dropout)         (None, 4, 2, 64)          0

 flatten (Flatten)           (None, 512)               0

=================================================================
Total params: 18,816
Trainable params: 18,816
Non-trainable params: 0
```

Figure 3-3 : Les paramètres du modèle CNN.

Ces informations donnent une vue d'ensemble de l'architecture du modèle, y compris les types de couches utilisées, les formes de sortie de chaque couche et le nombre total de paramètres. Cependant, pour obtenir une compréhension plus complète du modèle, il serait nécessaire d'examiner les autres couches et la sortie finale du modèle.

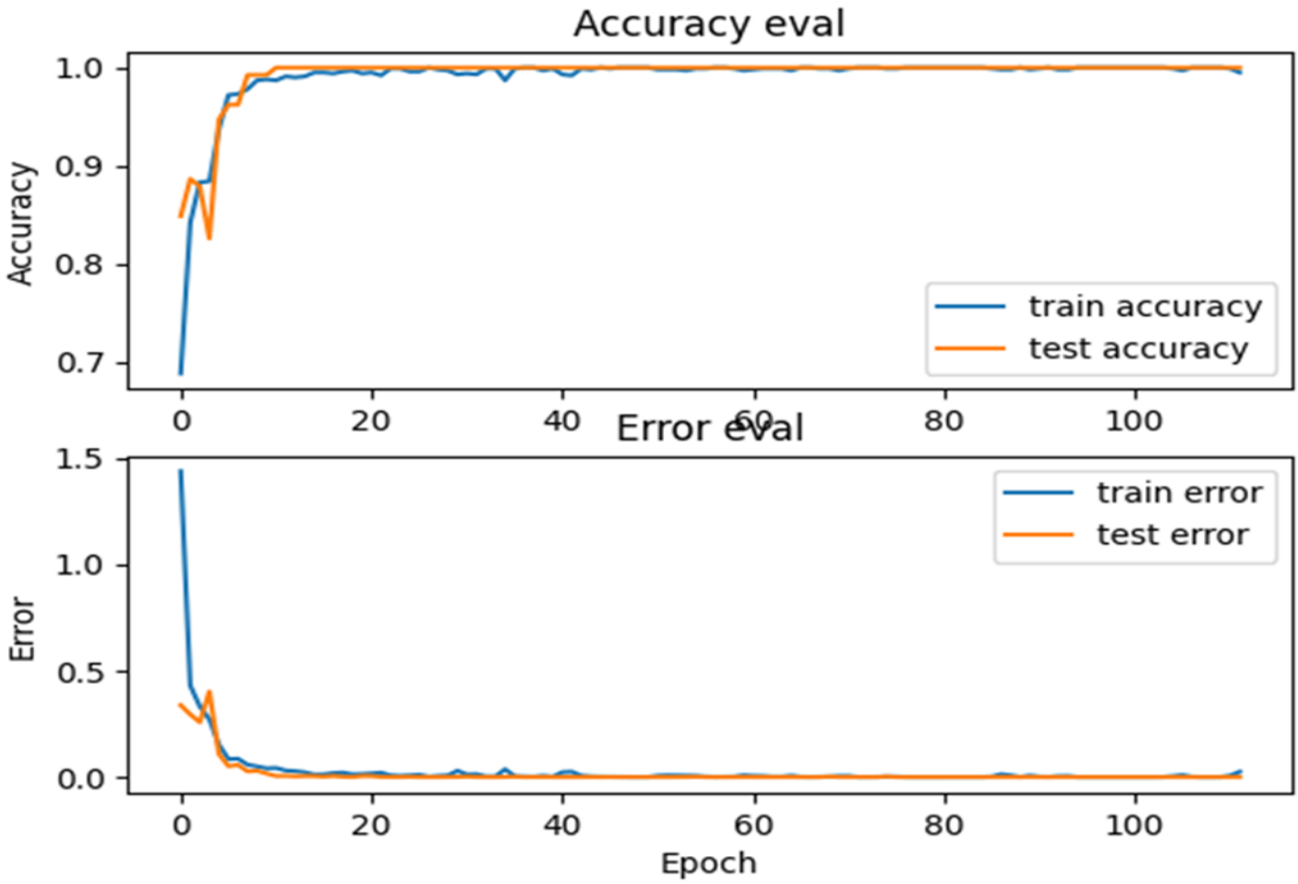
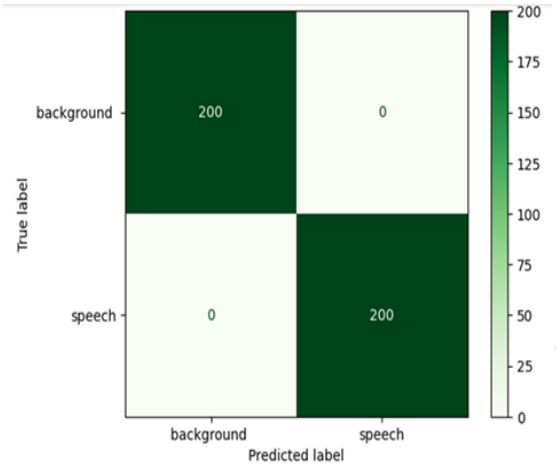


Figure 3-4 : Accuracy et l'erreur du model CNN

Cela permet de visualiser graphiquement les performances du modèle pendant l'entraînement, en montrant comment l'exactitude évolue et comment l'erreur diminue au fil des époques. Cela peut aider à évaluer la progression de l'entraînement et à détecter d'éventuels problèmes tels que le su apprentissage.

3.6 Evaluation

Le modèle a obtenu de bons résultats avec une grande précision pour les deux classes, en particulier pour la classe "speech".



	precision	recall	f1-score	support
background	1.00	1.00	1.00	200
speech	1.00	1.00	1.00	200
accuracy			1.00	400
macro avg	1.00	1.00	1.00	400
weighted avg	1.00	1.00	1.00	400

```
[[200   0]
 [  0 200]]
```

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

F1 Score: 1.0

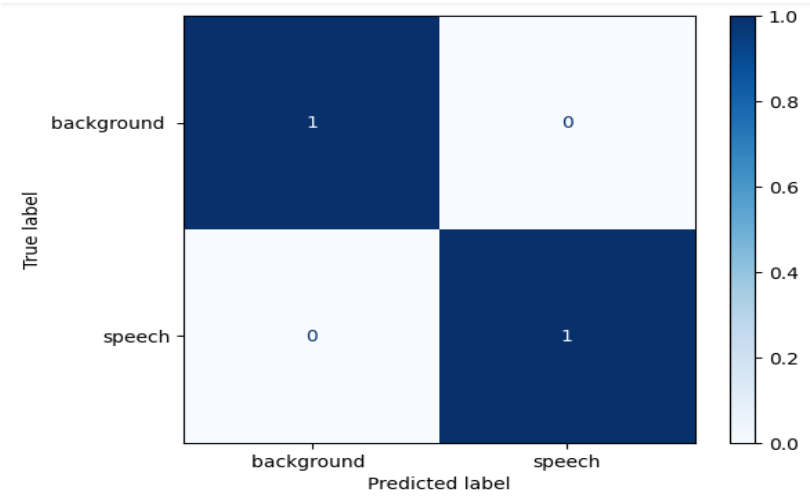


Figure 3-5 : Evaluation Measures

Les résultats d'évaluation indiquent que le modèle a une performance parfaite avec une précision, un rappel et un F1-score de 1.00 pour les classes "background" et "speech". L'accuracy globale est également de 1.00, ce qui signifie que le modèle a prédit correctement toutes les classes pour toutes les observations dans l'ensemble de données évalué.

Lorsque nous traitons un signal audio contenant de la parole, il est essentiel d'utiliser des techniques appropriées pour extraire les caractéristiques pertinentes.

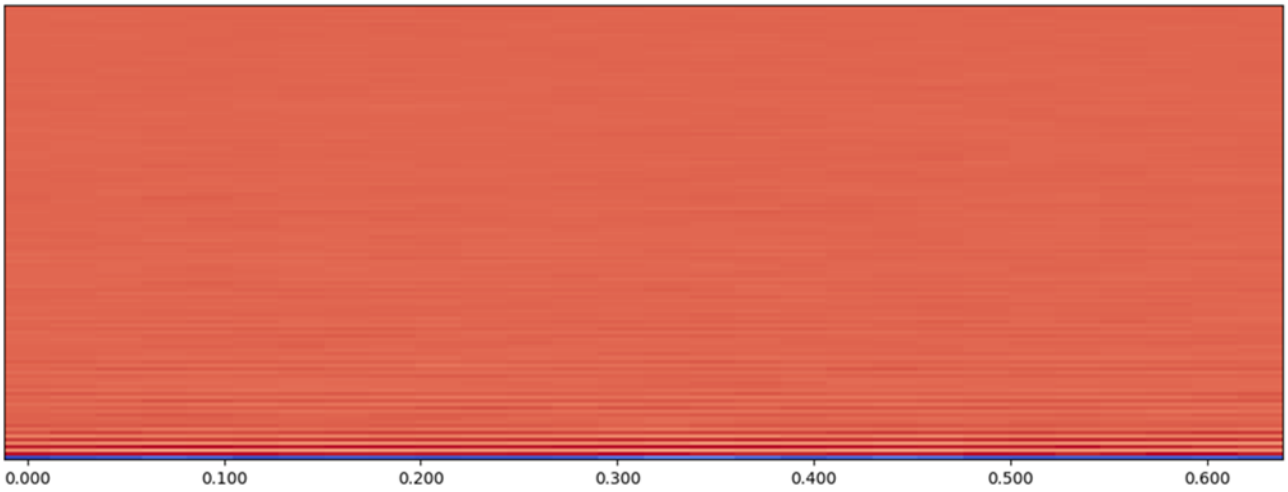


Figure 3-6 : représente les coefficients MFCC du signal audio.

Une approche couramment utilisée est l'application de la transformée de Fourier à court terme (STFT) sur le signal. La STFT permet d'analyser le signal audio à la fois dans le domaine temporel et fréquentiel pour garder tous les deux.

La STFT divise le signal audio en petits segments de temps, puis calcule-la transformée de Fourier de chaque segment. Cela permet de visualiser les variations de fréquence dans le signal tout au long du temps. L'axe horizontal de cette représentation est le temps, tandis que l'axe vertical représente la fréquence.

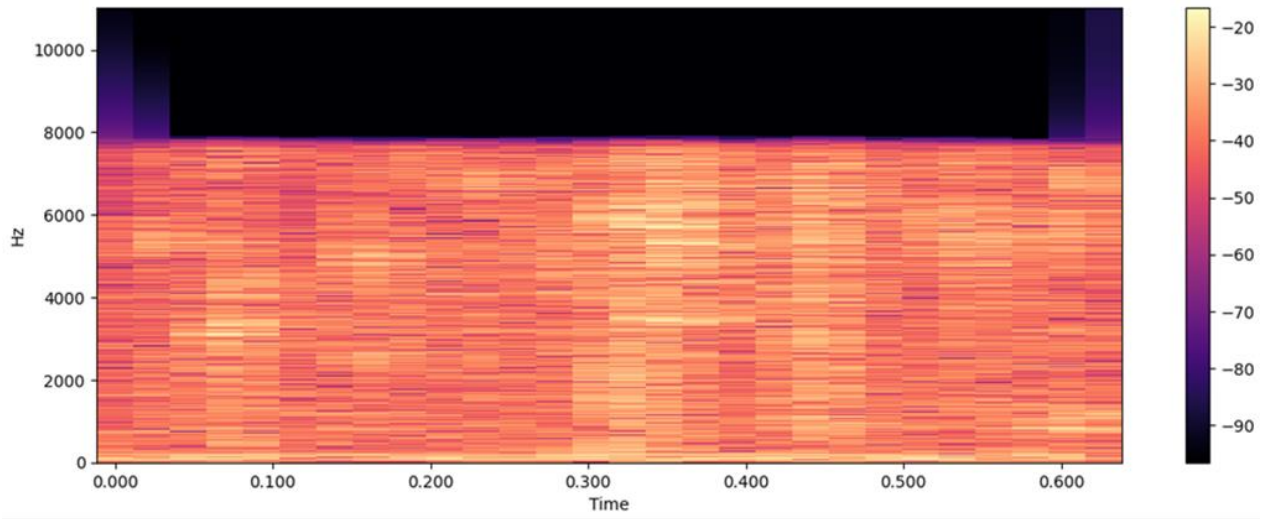


Figure 3-7 : spectrogramme représente la variation de l'amplitude du signal audio

Cependant, le spectrogramme ne fournit pas directement les caractéristiques les plus informatives pour la détection d'activité vocale. C'est là que les coefficients cepstraux de fréquence mel (MFCCs) entrent en jeu. Les MFCCs sont calculés en appliquant une transformation en cosinus discrète (DCT) sur le spectrogramme. Cette transformation réduit la dimensionnalité des données et se concentre sur les composantes fréquentielles les plus discriminantes

3.7 Conclusion

Les résultats montrent que le modèle a une très bonne performance avec une précision élevée, une capacité à bien identifier les échantillons des deux classes et un bon équilibre entre la précision et le rappel.

Conclusion générale

Dans ce projet, nous nous sommes concentrés sur la détection de l'activité vocale basée sur les aspects les plus importants de la classification binaire car elle fait la distinction entre les zones actives et inactives dans un environnement bruyant. Notre objectif principal était donc de résoudre ce problème en utilisant l'apprentissage en profondeur.

Nous avons donc commencé nos recherches sur les réseaux de neurones artificiels et les approches d'apprentissage en profondeur et expliqué certaines architectures universelles. Nous avons également présenté la plupart des concepts de base du traitement du signal, et bien que la VAD soit étudiée depuis plusieurs décennies, nous avons mis en évidence les principales approches qui ont contribué au développement de la VAD au fil du temps.

Dans la mise en œuvre, nous nous sommes appuyées sur l'architecture CNN.

Nous avons mené des expériences en utilisant LibriSpeech et TiDiGiTs , CHiME-5 , y compris la détermination de différents niveaux de bruit. Egalement, nous nous sommes appuyés sur MFCC et leurs dérivés pour extraire les caractéristiques.

Etant donné que la mise en œuvre des expériences est basée sur GPU, et en raison du manque d'espace suffisant dans l'environnement de développement, nous avons trouvé que CNN produisait les meilleurs résultats de performance dans des ensembles sans bruit ou de faibles niveaux de bruit. Comme critères d'évaluation, nous avons utilisé la précision, F1-score, Accracy, recall, RMS et le Loss.

Bibliographie

[1] R. Johny Elton P. Vasuki, J. Mohanalin, “Voice Activity Detection Using Fuzzy Entropy and Support Vector Machine Support Vector Machine”, ed. Knuth Kevin H. 2016.

[2] Ourdighi Asmaa,” Contribution à l’étude de la robustesse des réseaux de neurones impulsionnels dans la reconnaissance de la parole”, Thèse de Doctorat, Informatique : Reconnaissance des formes et intelligence artificielle, Université des sciences et de la technologie d’Oran Mohamed Boudiaf, 2017, pages : 86-87.

[3] Roberto Chiodi, ” Détection d’activité vocale basée sur la transformée en ondelettes ”, Thèse présenté comme exigence partielle de la maitrise en génie électrique, Université du Québec, Canada, 2010, pages : 11-12.

[4]Agnel Waghela Rohan Reddy, Shivangi Rai, Aditya Pawar, Namrata Gharat SUV. “Detection Algorithm for Speech Signals”, International Journal of Advanced Research in Computer Science and Software Engineering. 2014.

[5] Subramanian Hariharan, "Audio signal classification". 2004.

[6] Rao Preeti. “Audio Signal Processing”, ed. Mahadeva Bhanu Prasad and S. R...- India: Springer-Verlag, 2007.

[7] Hadri Cherif, « La recherche des paramètres de la trace acoustique et son application dans la reconnaissance de la parole », Thèse de Magistère, Option Systèmes intelligents, Université Badji Mokhtar-Annaba, 2008, page :21.

[8] Charles C. Introduction aux ondelettes [Revue].

[9] Sherry Vijh Parminder Singh and Manjot Kaur Gill Feature Extraction Using MFCC for Speech Recognition [Revue]. - Ludhiana (India): Guru Nanak Dev Engineering College.

[10] Manjutha M Gracy J, Dr P Subashini, Dr M Krishnaveni utomated Speech Recognition System – A Literature Review [Revue] // International Journal of Engineering Trends and Applications (IJETA) – Volume 4 Issue 2, . - India: [s.n.], 2017.

[11] Urmila Shrawankar, Dr. Vilas Thakare. Techniques for feature extraction in speech recognition system: A comparative study [Revue].

[12] Laurent BUNIET, « Traitement automatique de la parole en milieu bruité : étude de modèles connexionnistes statiques et dynamiques », thèse de Doctorat, spécialité informatique, Université Henri Poincaré – Nancy I, Français, 1997, pages : 23.

[13] Cyril Plapous, « Traitements pour la réduction de bruit. Application à la communication parlée. », Thèse de Doctorat, Traitement du signal et Télécommunications, université de Rennes 1, France, 2005, pages : 11-12.

[14] J. Perr, «Rudiments d'acoustique et de traitement du signal», LinuxFocus article number 271, 14 01 2005.

[15] Laurence Vidrascu. « Analyse et détection des émotions verbales dans les interactions orales », Thèse de Doctorat, Informatique, Université Paris 11, France, 2007, pages : 93-94.

[16] Abdelouahed Sara (ep) Slimani, « Etude et réalisation d'une plateforme tele medicale dediée à l'évaluation objective et au suivi des dysphonies chroniques d'origine laryngée par analyse spectro-temporelle du signal vocal », Thèse de Doctorat, Spécialité : " Génie Biomédical ", Université Abou Bekr Belkaid Tlemcen, 2015, pages : 19-20.

[17] Aziza Yassamine, « Mod´elisation AR et ARMA de la Parole pour une Vérification Robuste du Locuteur dans un Milieu Bruité en Mode Dépendant du Texte », Th`ese de Magistère, Spécialité : 'Communication', Université Ferhat Abbas-Setif1-UFAS (Algérie), 2013, pages : 7-8-9.

[18] Amrane Abdessalem et Ould Ammar Kheirreddine, « Nouvelle technique automatique de réduction de bruit acoustique basée sur le principe de séparation

aveugle de source », Thèse de Master, Spécialité : Réseau et Télécommunications, Université Saad Dahlab de Blida, 2019, pages : 8-9-10.

[19] M. V. Droogenbroeck, Principes des télécommunications analogiques, Institut Montefiore/Service de Télécommunications et d'Imagerie éd., université de liège, 2013, p. 19.

[20] P. Kim, MATLAB deep learning: with machine learning, neural networks and artificial intelligence, eng, sér. For professionals by professionals. New York: Springer, 2017, ISBN: 978-1-4842-2845-6 978-1-4842-2844-9.

[21] Alex Graves Abdel-rahman Mohamed and Geoffrey Hinton. Speech Recognition with deep recurrent neural networks [Revue] // arXiv: 1303.5778v1 [cs.NE]. - 2013. - p. 2.

[22]Mohammed Msaaf Fouad Belmajdoub. L'application des réseaux de neurone de type "feedforward" dans le diagnostic statique. [Revue] // ffhal-01260830. - Tanger, Maroc : [s.n.], Dec 2015. - pp. 2-4

[23] Berthelier, A., Yan, Y., Chateau, T., Blanc, C., Duffner, S. et Garcia, C., 2020, juin. Compression de réseaux convolutifs par utilisation d'un terme de clarté l1/l2 sur les noyaux. Dans RFIAP

[24] Visetti, YM, 1991. Des systèmes experts aux systèmes à base de connaissances : à la recherche d'un nouveau schéma régulateur. Intellectica , 12 (2), pp.221-279

[25] Kennedy, J., 2006. Intelligence d'essaim. Dans Manuel d'informatique inspirée de la n

[26] Phil Kim; Matleb Deep learning with machine learning, Neural networks and Artificial intelligence.

[27] A. Krizhevsky, I. Sutskever et G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks », Advances in neural Processing Systems de traitement. 2012.

[28] https : //medium.com/deep-learning-turkey/google-colab-free-gpu-tutorial-e113627b9f5d consulter le : 28-09-2020.

[29] https: //moov.ai/fr/blog/deep-learning-avec-google-colab/ consulter le: 28-09-2020.

[30] https ://www.python.org./

[31] https://librosa.org/doc/latest/index.html

[32] https://datascientest.com/numpy

[33] https://datascientest.com/matplotlib-tout-savoir

[34] https://www.w3schools.com/python/pandas/pandas_intro.asp

[35]https://scikitlearn.org/0.15/modules/preprocessing.html#:~:text=The%20sklearn.,suitable%20for%20the%20downstream%20estimators.

[36] https://pypi.org/project/tensorflow/

[37] https://keras.io/about/

[38] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.

[39] R. G. Leonard and G. R. Doddington. (1992) A speaker-independent connected-digit,database.[Online],Available:
https://catalog.ldc.upenn.edu/docs/LDC93S10/tidigits.readme.html

[40] (2018) The 5th chime speech separation and recognition challenge. [Online]. Available: http://spandh.dcs.shef.ac.uk/chime_challenge/data.html

[41] (2018) Implementation of set assignment algorithm. [Online]. Available: https://github.com/tensorflow/tensorflow/blob/master/tensorflow/examples/speech_commands/input_data.py#L61.