الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث 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

Mahmoudi Fatma

&

Zerroual Chahinez

Débruitage de la parole basée sur un réseau neuronal convolutif (Convolutional Neural Network:CNN)

Proposé par :

Pr Mr. Ykhlef Farid

Année Universitaire 2023-2024

Remerciements

Nous remercions Dieu le tout puissant de nous avoir donné le courage et la Volonté de Parvenir à la fin de notre parcours universitaire.

Nous tenons à remercier tous ceux qui nous aidé, conseillé et encouragé afin de réaliser ce modeste travail. Et aussi on n'oublie pas de remercier ***Mr Ykhlef Farid,*** *notre encadreur pour tout son soutien et ces conseils qui nous ont apporté de l'aide dans la réalisation de notre projet.*

Nous tenons à remercier ***Mr Benselama.Z*** *d'avoir accepté de faire partie du jury, Ainsi que* ***Mme Tidjani.N*** *D'en être membre.*

Nos remerciements vont aussi à tous le corps pédagogique : enseignants, administrateurs, employés du département d'électronique ainsi que toutes les personnes de notre faculté.

Dédicaces

Je dédie ce modeste travail avec mon grand amour et mon entière gratitude aux les plus profond a :

Ma mère, Affable, honorable, aimable : Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Mon père, qui a toujours cru en moi et a mis à ma disposition tous les moyens nécessaires pour que je réussisse dans mes études.

mes chères sœurs, qui ont toujours été là pour moi et ont constamment souhaité mon bonheur.Dr. SOUHILA, KHALIDA

Mes frères, qui n'ont jamais cessé de désirer mon bonheur.

A mes familles :

Pour toute l'affection qu'ils m'ont donnée et pour leur précieux encouragement. Que Dieu, le tout puissant vous garde et vous procure santé, bonheur et longue vie.

Mon directeur de mémoire Mr YKHLEF FARID :

Pour avoir accepté de m'encadrer tout au long de ce travail, pour ses conseils et suggestions et surtout pour sa patience.

Mon binôme et ma sœur ZERROUAL CHAHINEZ :

La personne avec laquelle j'ai partagé cette expérience et qui n'a cessé d'être pour moi un exemple de persévérance, de courage et de générosité.

A mes meilleurs amis :

KAHINA, IMANE, HANNANE, et NAZIHA et WISSEM, IKRAM , AMINE Merci pour les bons moments qu'on a passés ensemble, de votre soutien et de votre serviabilité.

A tous les professeures d'instrumentation spécialement

Et en fin, A tous les étudiants de ma promotion : 2023-2024.

MAHMOUDI FATMA

Dédicaces

Je dédie ce modeste travail aux personnes les plus chères au monde :

Mes parents qui mon toujours aidé et encouragé dans mon parcours universitaire, sans oublié leurs sacrifices et amour.

A mes sœurs et à mes frères.

A ma collègue et mon binôme fatma ((Rania)) et à tous mes amies surtout :

A tous le personnel des laboratoires pédagogiques d'électronique

Sans oublié tous les étudiants de la promotion Master2 INS.

ZERROUAL CHAHINEZ

ملخص:

هـذه المـذكرة تتنـاول تنقيـة الكـلام فـي إطـار الـتعلم العميـق. تنقيـة الكـلام ضـرورية لتحسـين جـودة وفهـم التسـجيلات الصـوتية فـي سـياقات متنوعـة. قمنـا بتطـوير وتـدريب نمـوذج لتقليـل الضوضـاء فـي إشـارات الكـلام باسـتخدام إشـاراة مشوشـة كإدخـال. أظهـرت النتـائج التـي تـم الحصـول عليهـا تحسـنًا ملحوظًـا فـي جـودة الصـوت مقارنـة بالأسـاليب التقليديـة. تسـمح لنـا نهجنـا بالحفـاظ علـى وضـوح وخصـائص الطبيعـة للكـلام بينمـا تقلـل بشـكل فعـال مـن الضوضـاء الخلفيـة. تظهـر هـذه النتـائج الإمكانيـات الكبيـرة لشـبكات العصـبونات التصـنيفية والـتعلم العميـق فـي تطبيقات معالجة إشارات الكلام في بيئات مزعجة .

الكلمات المفتاحية: شبكة CNN, تحسين الكلام, تعلم عميق

Résumé : Ce mémoire traite du débruitage de la parole en utilisant des réseaux de neurones convolutifs (CNN). Le débruitage de la parole est crucial pour améliorer la qualité et l'intelligibilité des enregistrements vocaux dans divers contextes. Nous avons développé et entraîné un modèle de CNN pour réduire le bruit dans les signaux de parole en utilisant un signal bruité comme entrée. Les résultats obtenus montrent une amélioration significative de la qualité sonore par rapport aux méthodes traditionnelles. Notre approche permet de préserver la clarté et les caractéristiques naturelles de la parole tout en réduisant efficacement le bruit de fond. Ces résultats démontrent le potentiel des CNN pour des applications de traitement du signal de parole dans des environnements bruyants.

Mots-clés :
Amélioration de la parole , apprentissage profond DL , réseau CNN

Abstract : This thesis addresses speech denoising using convolutional neural networks (CNNs) within the framework of deep learning. Speech denoising is crucial for improving the quality and intelligibility of vocal recordings in various contexts. We developed and trained a CNN model to reduce noise in speech signals using spectrograms as input. The obtained results show a significant improvement in sound quality compared to traditional methods. Our approach preserves the clarity and natural characteristics of speech while effectively reducing background noise. These results demonstrate the potential of CNNs and deep learning for speech signal processing applications in noisy environments.

Keywords: Speech enhancement, deep learning, CNN network.

Listes des acronymes et abréviations

ML :	Machine Learning
DL :	Deep Learning
RNA :	Réseau Neurone Artificiel
CNN :	Convolutional Neural Network
FC :	Fully-Connected
STFT :	Short-Time Fourier Transforme
ReLU :	Rectified Linear Unit
NLP :	Natural Language Processing
CPU:	Central Processing Unit
GPU :	Graphic Processing Unit
GAN	generative adversarial networks
IA :	Intelligence Artificielle

Table des matières

Liste des figures

Introduction générale

Le domaine de traitement du signal vocal a connu des avancées significatives ces dernières années, avec une utilisation croissante de la reconnaissance vocale et des applications liées à la parole. Cependant, un défi persistant dans ce contexte est la présence de bruit dans les enregistrements vocaux, pouvant compromettre la qualité et la compréhensibilité du signal. La suppression de ce bruit, également connue sous le nom de « speech denoising », représente un axe crucial de recherche visant à améliorer la clarté des enregistrements vocaux.

L'art du rehaussement vocal vise à raffiner la qualité de la communication, même lorsque la parole est altérée. Avec l'avènement fulgurant de l'intelligence artificielle (IA) à travers divers horizons, son rôle primordial dans l'évolution de multiples secteurs est désormais indéniable. Ainsi, l'intégration du Machine Learning et du Deep Learning s'avère essentielle, ouvrant de nouvelles perspectives et catalysant l'innovation. Les chercheurs se sont aventurés dans les profondeurs de l'apprentissage en profondeur, une méthode de pointe pour améliorer la parole, caractérisée par l'utilisation de modèles déterministes et une formation supervisée.

Les récents systèmes intelligents de traitement audio intègrent fréquemment des mécanismes visant à affiner de manière plus précise la réduction du bruit dans les scènes sonores. Cette approche est catégorisée comme un sous-domaine de la vision par ordinateur, s'appuyant sur des techniques telles que le Machine Learning et le Deep Learning (DL).

Dans l'évolution récente du domaine de l'apprentissage, un sous-type de réseau neuronal, appelé CNN (Convolutional Neural Network), a pris une place prépondérante pour les tâches liées aux signaux sonores. Le réseau neuronal est formé sur diverses caractéristiques, allant de la réduction du bruit dans les signaux audio à l'amélioration de la qualité, en passant par l'optimisation de la puissance et de la clarté du son.

Dans cette étude, nous plongerons dans les concepts théoriques, en commençant par mettre en lumière l'importance cruciale du Deep Learning, suivi par une exploration des fondamentaux de l'intelligence artificielle, du Machine Learning et du Deep Learning, ainsi

que leur évolution à travers les décennies. Nous aborderons également les réseaux neuronaux, avant de nous concentrer spécifiquement sur les Convolutional Neural Networks (CNN).

Après avoir exploré les fondements et résolu les problématiques initiales, nous plongerons dans le passionnant défi de la réduction du bruit sonore. Nous découvrirons ensemble la conception de modèles en utilisant des structures diverses, en exploitant les puissants outils de "torch" et "torchaudio".

Nous scruterons avec attention les résultats obtenus à travers nos expérimentations, en exploitant une gamme variée de bibliothèques pour simplifier la mise en œuvre et accélérer l'apprentissage.

Ce projet sera organisé de la manière suivante :

- Le premier chapitre : Ce chapitre présente les fondements théoriques et un état de l'art détaillé sur le traitement du signal vocal et la réduction du bruit. Nous y explorons les concepts clés, les techniques classiques et les approches modernes utilisées dans ce domaine.

- Le deuxième chapitre : consiste à étudier de manière générale l'intelligence artificielle, le Machine Learning et le Deep Learning, ainsi que les réseaux neuronaux avec un focus sur l'architecture des CNN.
- Le dernier chapitre se concentre sur la méthodologie employée pour débruiter la parole en utilisant des réseaux de neurones convolutifs (CNN). Nous exposons en détail les étapes et les techniques utilisées pour concevoir et former notre modèle de CNN, ainsi que les méthodes pour évaluer ses performances.

Chapitre 1 : Introduction au traitement de la parole et à la réduction du bruit

Introduction

La parole, ce courant dynamique de pensées et d'émotions qui unit les individus, représente l'essence fondamentale de toute société humaine. Son émergence, en corrélation avec nos premiers outils, symbolise notre besoin intrinsèque de réflexion et de partage d'idées. Son abstraction, indépendante de tout support physique, en fait un moyen de communication d'une simplicité remarquable. Ce chapitre entreprend une exploration approfondie de la modélisation de la parole, mettant en lumière les différentes nuances des sons qui la composent ainsi que les subtilités de ses variations. Préalablement à cette analyse, une étude des organes biologiques impliqués dans la production et la compréhension de la parole éclaire les défis rencontrés dans le traitement du langage.

I.1 Définition du son

Le son se propage à la suite de la perturbation du milieu, le plus souvent l'air, mais qui peut aussi être solide ou liquide. Captée par notre oreille, cette vibration met en mouvement le tympan, point de départ de la stimulation de l'oreille et de la perception de l'information sonore. Le son possède certaine propriété singulière et se caractérise par une très grande variabilité. Ce qui caractérise la parole, c'est son irreproductibilité, nous ne reproduisons jamais deux fois le même son.

Notre environnement est composé d'une grande variété de sons plus ou moins fréquents, par exemples :

- Son musical : Le son musical varie avec la mélodie, le morceau choisi et avec l'instrument utilisé.
- Parole : Les sons de la parole sont aussi complexes et variés, chacun a sa propre voix, grave ou aiguë, avec un timbre particulier, etc.

La suite de notre travail sera concentrée sur le signal de la parole

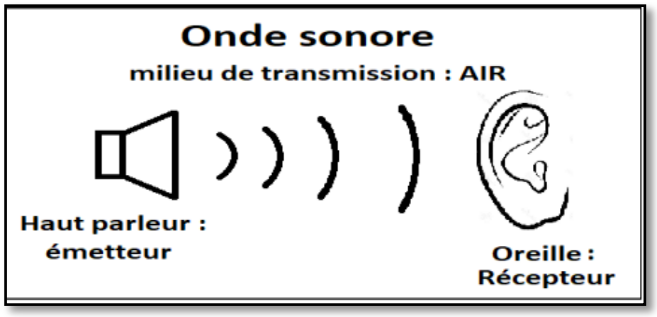


Figure I-1 : transfert de l'information sonore.

I.2 Les paramètres du signal de la parole

Le signal vocal est généralement caractérisé :

I.2.1 La fréquence fondamentale

La fréquence est déterminée par le nombre de vibrations qu'un corps réalise en une seconde. La fréquence fondamentale 'F0' est un composant de basse fréquence de la parole, résultant de la vibration des cordes vocales, permettant la perception de la hauteur tonale de la voix d'un individu, Cette fréquence varie généralement entre 85Hz -255Hz. Il joue un rôle important dans la parole **[1].**

En règle générale, la prosodie, qui englobe les variations de la fréquence fondamentale, de l'intensité et de la durée, est un élément crucial pour mettre en lumière de nombreuses caractéristiques du locuteur. Elle permet notamment de discerner des aspects tels que le genre, les origines géographiques et culturelles, ainsi que les émotions exprimées.

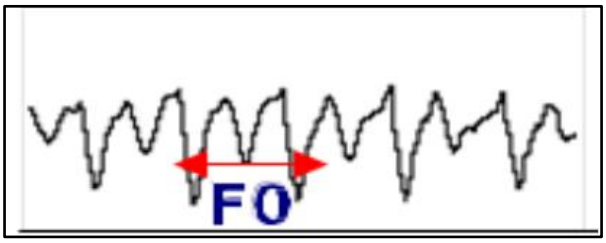


Figure I-2 : Fréquence fondamentale.

Cette fréquence peut varier **[2] :**

- De 70Hz à 250Hz pour une voix masculine.
- De 150Hz à 400Hz pour une voix féminine.
- De 200Hz à 600Hz pour une voix d'enfant.

Nous voyons donc que la périodicité des signaux de parole est une caractéristique non uniforme, qui varie en fonction des individus.

I.2.2 L’énergie [3]

L'énergie acoustique, exprimée par l'intensité acoustique du son, est directement liée à la pression de l'air juste avant le larynx. L'amplitude du signal vocal fluctue au cours du temps en fonction du type de son produit, et cette énergie dans un segment est déterminée par :

$$E = I \cdot A \cdot T$$

- **I** l'intensité acoustique moyenne.
- **A** la surface sur laquelle l'intensité est mesurée.
- **T** la durée du segment.

I.2.3 Le spectre

Le spectre sonore englobe les sons perceptibles par l'oreille humaine, allant des infrasons aux ultrasons, soit des fréquences de 20 à 20 000 Hertz, qui sont celles que l'homme peut entendre.

Dans ce spectre sonore, on différencie les sons fondamentaux du bruit. Les sons fondamentaux correspondent aux sons produits par les instruments de musique ou la voix humaine, caractérisés par une fréquence fondamentale et des harmoniques. En revanche, le bruit n'est pas défini par une fréquence spécifique, chaque son quotidien ayant sa propre fréquence distincte.

I.2.4 Les formants

Les formants représentent l'une des caractéristiques les plus essentielles des signaux de parole et sont largement exploités dans leur analyse. Ils correspondent aux fréquences de résonance de l'appareil phonatoire humain. Étant donné que ce dernier peut modifier ses propriétés physiques pour émettre différents sons, plusieurs formants peuvent ainsi être générés.

I.2.5 Le timbre

Le timbre est une caractéristique permettant de différencier deux sons de même hauteur et de même amplitude. Il résulte de la combinaison entre la fréquence fondamentale et les harmoniques.

I.2.6 Intensité

L'intensité d'un son, également appelée volume, permet de distinguer entre un son fort et un son faible. Elle est déterminée par l'amplitude des vibrations. Les sons les plus faibles perceptibles, avant d'atteindre le seuil de silence total (seuil d'audibilité), sont d'environ 10^{-16} w, tandis que les sons les plus intenses audibles sans causer de douleur (seuil de la douleur) sont d'environ 10^{-3} w

I.2.7 La durée

C'est la mesure précise ou relative d'un son, et dans certaines langues, son importance peut être significative.

I.3 système de production de la parole

I.3.1 l'appareil phonatoire

L'appareil phonatoire nous donne la capacité de produire une vaste gamme de sons, malgré les limites fréquentielles et énergétiques de l'espace disponible (**Figure I-3**). L'étude de l'appareil phonatoire humain a été le point de départ de recherches visant à reproduire mécaniquement ses capacités, ce qui a en retour permis une meilleure compréhension de son fonctionnement.

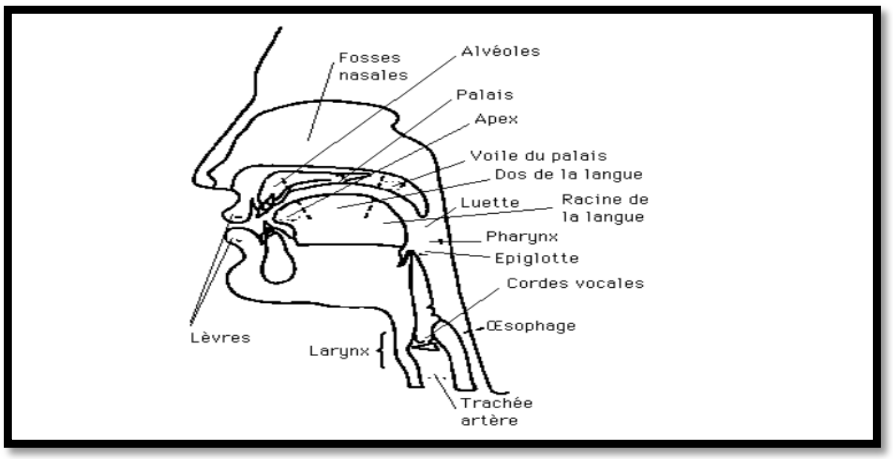


Figure I-3 : l'appareil phonatoire.

I.3.2 la voix humaine

La génération de la parole peut être décrite comme le résultat d'une coordination volontaire et synchronisée entre plusieurs groupes musculaires. Ce processus est contrôlé par le système nerveux central, qui reçoit en permanence des informations provenant du feedback auditif et des sensations kinesthésiques. L'appareil respiratoire joue un rôle essentiel en fournissant l'énergie nécessaire à la production des sons, en propulsant de l'air à travers la trachée-artère. Au niveau du larynx, situé au sommet de la trachée, la pression de l'air est soigneusement ajustée avant d'être dirigée vers les voies vocales. Composé de muscles et de cartilages mobiles, le larynx encercle une cavité essentielle à la production de la voix (**FigureI-4).** Les cordes vocales sont comparables à deux lèvres symétriques positionnées à travers le larynx. En se rapprochant ou s'éloignant graduellement, elles peuvent moduler une ouverture triangulaire connue sous le nom de glotte. Cette dernière permet le passage libre de l'air lors de la respiration ainsi que lors de l'émission de la voix chuchotée.

Initialement fermé, le larynx exerce une pression sur les cordes vocales, les poussant à s'ouvrir. Ceci diminue la pression, permettant aux cordes vocales de se refermer. Ce processus crée des fluctuations de pression le long du conduit vocal, incluant les cavités pharyngienne et buccale. Lorsque la luette est abaissée, la cavité nasale s'ajoute à cette voie. Soulignons également l'importance de la langue dans le processus phonatoire. Sa position détermine la hauteur du pharynx : plus la langue est basse, plus le pharynx est court. Elle affecte également le point d'articulation, marquant le rétrécissement maximal du canal buccal, ainsi que l'écartement des organes au point d'articulation.

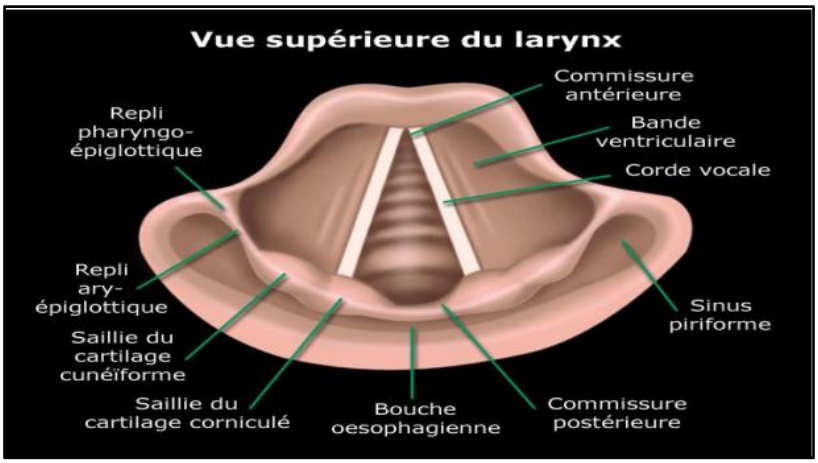


Figure I-4 : Section du larynx, vu de haut

I.4 Acquisition d'un signal

Acquérir un signal, c'est récupéré une information numérique ou analogique par un système : scanner, capteur…etc.

I.4.1 Chaine d'acquisition d'un signal audio

Une chaîne d'acquisition numérique peut se représenter selon la figure suivante :

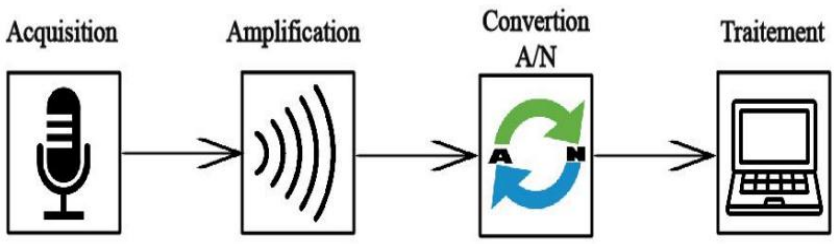


Figure I-5 : Chaine d'acquisition de signal de la parole.

I.5 Les bruits

Les bruits sont des perturbations qui peuvent altérer la qualité de la communication, dénaturer le message communiqué et rendre difficilement perceptible l'information. Ils constituent donc une gêne dans la compréhension du signal utile, qui est dans notre cas, la parole **[4]**.

I.5.1 Bruit blanc

Le bruit blanc est un bruit composé de toutes les fréquences au même niveau statistique. Il présente la même énergie pour toutes les fréquences **[5].**

I.5.2 Bruits colorés

Dans le cas où la densité spectrale de puissance (DSP) n'est pas constante en fonction de la fréquence, le signal aléatoire est alors appelé bruit coloré. Pour cette représentation spectrale, les principaux types de bruits colorés se distinguent par leurs spectres. Il existe plusieurs types de bruits colorés, comme le bruit rose et le bruit brun **[6].**

I.5.3 Bruit musical

Le bruit musical est un bruit résiduel perpétuellement gênant qui apparait suite au débruitage de la parole par des algorithmes d'atténuation spectrale à court terme tels que la soustraction spectrale ou le filtrage de Wiener. Le spectre du bruit musical est particulièrement tonal, d'où le caractère musical **[4].**

I.5.4 Bruit impulsif

Comme son nom l'indique, ce type de bruit est à caractère impulsif, il se présente sous forme de tensions perturbatrices de valeur élevée mais de durée brève. Ces bruits sont très gênants pour la transmission des données, car le signal perturbateur modifie la forme du signal reçu à des instants quelconques (aléatoires) **[7]**.

I.5.5 Bruit ambiant

Le bruit ambiant est la somme du bruit résiduel et du bruit particulier émis par la source. Il est composé de l'ensemble des bruits émis par toutes les sources proches et lointaines **[7].**

I.5.6 Bruit acoustique

Il est généré par les mouvements des sources telles que les voitures, les ventilateurs, la circulation, le vent, la pluie. **[5].**

I.5.7 Les modèles du bruit

Pour simuler ou analyser des signaux réels, d'autres modèles, basés sur la densité de probabilité du bruit, peuvent être considérés. Ainsi, en particulier, deux modèles du bruit sont assez répandus :

- Le bruit dit gaussien dont la densité de probabilité a une répartition de type gaussien caractérisée par une valeur moyenne et un écart type.

- Le bruit dit périodique formé d'une somme de signaux sinusoïdaux sans référence de phase.

I.6 Classification des méthodes de débruitage de la parole

Les méthodes de débruitage de la parole peuvent être classées de plusieurs façons (Tableau I.1). Elles peuvent être mono-microphones ou multi-microphones selon le nombre de microphones utilisés. Elles peuvent également être temporelles ou fréquentielles en fonction du domaine de traitement. La troisième méthode de classification peut être basée sur le type d'algorithme utilisé (adaptatif ou non adaptatif) **[8]**

Tableau I.1 : Classification des méthodes de rehaussement de la parole

Algorithme	Adaptatif ou non adaptatif
Nombre de microphones	Un seul ou plusieurs
Domaine de traitement	Temporel ou fréquentiel

I.7 Etat de l'art des méthodes de débruitage de la parole

Une des premières techniques de débruitage de la parole dans le domaine spectral est la soustraction spectrale introduite par Boll en 1979 **[9].** La soustraction spectrale consiste à atténuer plus ou moins les composantes spectrales du signal bruité en fonction de l'estimation du niveau de bruit. Un des gros désavantages de la soustraction spectrale telle que présentée est que l'estimation du bruit induit un nouveau bruit qui contient une certaine musicalité, nommé "bruit musical" **[10]**. Virag **[11]** a présenté une généralisation de la soustraction spectrale, qui est faite par le mixage des algorithmes de Boll **[9]** et de Bertouli **[12]** avec la généralisation de Lim **[13**]. Cette généralisation a pour intérêt de trouver un compromis entre la réduction du bruit et la distorsion du signal amélioré **[14].**

Le filtre de Wiener est parmi les méthodes de débruitage classiques les plus utilisées dans la littérature. Il a été introduit pour essayer d'améliorer la qualité de la trace recueillie dans les potentiels évoques. Le problème de ce type de filtrage est qu'il n'est pas applicable sur une moyenne d'acquisition mais pour chaque trace. Doyle **[15]**, propose une modification pour pouvoir l'adapter à la moyenne (le calcul dans ce cas est beaucoup plus rapide), il n'y a pas besoin de filtrer chaque trace. Cependant, il faut considérer que le bruit est stationnaire

dans ce cas. Paliwal**[16]** a proposé une méthode de filtrage de Wiener non stationnaire pour le rehaussement de la parole, où le filtre est conçu pour chaque segment de parole de courte durée en utilisant une procédure des moindres carrés.

Le débruitage par ondelettes a connu un grand succès. Une ondelette est une fonction qui oscille comme une onde mais qui est rapidement atténuée d'où son nom ondelette qui veut dire petite onde **[17]**. Le sucées de l'analyse en ondelettes est dû à sa relation avec l'analyse multi-résolution développée par Stéphane Mallat**[18]** et Yves Meyer **[19]**. Cela a permis aux chercheurs et aux mathématiciens de construire leurs propres familles d'ondelettes **[20- 22]**.

I.7.1 Soustraction spectrale

Nous allons introduire dans cette section, l'algorithme de Boll **[24]** considéré comme une méthode de soustraction spectrale de référence. Soit un signal bruité x(t) composé d'un signal propre s(t) corrompu par un bruit additif n(t) non corrélé à s(t). Le signal bruité peut être exprimé par :

$$x(t) = s(t) + n(t) \qquad \textbf{(I.1)}$$

Nous désignons par ŝ(t) le signal rehaussé qui est une estimation de s(t). Nous supposons que le signal de parole est quasi-stationnaire sur des fenêtres d'analyse de 20 ms à 30 ms. La transformée de Fourier à court terme (TFCT) de x(t) est donnée par :

$$X(\omega) = S(\omega) + N(\omega) \qquad \textbf{(I.2)}$$

où ω est la fréquence angulaire et X(ω), S(ω) et N(ω) désignent respectivement le spectre de x(t), s(t) et n(t). X(ω) peut être exprimé sous forme polaire comme suit :

$$X(\omega) = |X(\omega)|\, e^{i\Phi x}(\omega) \qquad \textbf{(I.3)}$$

$|X(\omega)|$ et$\Phi x(\omega)$représentent, respectivement, l'amplitude et la phase de X(ω).

On peut également exprimer le bruit par $(\omega) = |N(\omega)|e^{i\Phi n}(\omega)$.

L'amplitude du bruit |N(ω)| est inconnue, ce qui nous conduit donc à la remplacer par sa valeur moyenne, estimée durant la période d'absence d'activité vocale. Dans les méthodes de soustraction spectrale, il est supposé que le bruit n'affecte pas la phase du signal. Pour cela, on considère uniquement l'amplitude spectrale à court terme du bruit. La valeur estimée de $\Phi n\,(\omega)$ sera donc remplacée par $\Phi x(\omega)$. En remplaçant X (ω) et $N\,(\omega)$ dans (I.3), la valeur estimée du spectre du signal propre est donnée par :

$$\hat{S}(\omega) = [|X(\omega)| - |\tilde{N}(\omega)|]\, e^{i\Phi x}(\omega) \qquad \textbf{(I.4)}$$

Où$|\tilde{N}(\omega)|$est la valeur estimée du spectre d'amplitude du bruit calculée durant la période d'absence d'activité vocale. Le signal rehaussé peut être obtenu en calculant la transformée de

Fourier inverse de $\hat{S}(w)$. On peut remarquer à partir de (I.4) que $|\hat{S}(\omega)| = |X(\omega)| - |\tilde{N}(\omega)|$ peut prendre des valeurs négatives en raison des erreurs d'estimation du bruit. Plusieurs solutions sont envisageables pour remédier à ce problème **[24]**, **[12]**. Parmi ces techniques, on peut citer la rectification demi-onde qui permet de mettre à zéro ces valeurs négatives, comme suit :

$$\left|\hat{S}(\omega)\right| = |X(\omega)| - |\tilde{N}(\omega)| si\ X(\omega) > \tilde{N}(\omega) \qquad 0\ si\ X(\omega) \leq \tilde{N}(\omega) \qquad \textbf{(I.5)}$$

L'algorithme de soustraction spectrale en amplitude (I.5) peut être étendu à la soustraction spectrale de puissance en multipliant $X(\omega)$ donné par (I.2) par sa conjuguée X*(ω) :

$$|X(\omega)|^2 = |S(\omega)|^2 + |N(\omega)|^2 + S^*(\omega) \times N(\omega) + S(\omega) \times N^*(\omega)$$

$$= |S(\omega)|^2 + |N(\omega)|^2 + 2 \times Re\{S^*(\omega) \times N(\omega)\} \qquad \textbf{(I.6)}$$

Si nous supposons que n(t) et s(t) sont à moyennes nulles et non corrélés, les termes E{S(ω).N*(ω)} et E{S*(ω). N(ω)} deviennent nuls. La valeur estimée du spectre de puissance du signal propre devient :

$$|\hat{S}(\omega)|^2 = |X(\omega)|^2 - |\tilde{N}(\omega)|^2 \qquad \textbf{(I.7)}$$

L'équation (I.7) décrit l'algorithme de soustraction spectrale en puissance et peut être exprimée par :

$$|\hat{S}(\omega)|^2 = H^2(\omega)\,|X(\omega)|^2$$

$$H(\omega) = \sqrt{1 - |\tilde{N}(\omega)|^2 |X(\omega)|^2} \qquad \textbf{(I.8)}$$

$H(w)$désigne le gain de suppression ou la fonction de suppression **[38]**.

I.7.2 Filtrage de Wiener

On considère un modèle de formation des données de type convolution. On peut donc écrire :

$$x(t) = (h * s)(t) + n(t) \qquad \textbf{(I.9)}$$

Où x et h représentent respectivement le signal observé et la réponse impulsionnelle du système, supposés connus. On considère que le bruit d'observation n(t) et la quantité d'intérêt s(t) sont aléatoires, indépendantes l'une de l'autre, centrées, stationnaires d'ordre 2 et caractérisées respectivement par leurs fonctions d'autocorrélation $R_n(\tau)$ et $R_s(\tau)$ ou, de manière équivalente, par leurs densités spectrales de puissance respectives $\Gamma_n(\omega)$ et $\Gamma_s(\omega)$. Tous les signaux sont supposés à valeurs réelles.

L'objectif est d'estimer s(t) avec un estimateur linéaire de la forme :

$$\hat{s}(t) = (w * x)(t) \qquad \textbf{(I.10)}$$

Selon un critère d'erreur quadratique moyenne minimale. On cherche donc à minimiser la quantité :

$$J \triangleq E\left[(s(t) - \hat{s}(t))^2\right] \qquad \textbf{(I.11)}$$

En raison de la linéarité de l'estimateur et du fait que, de par la structure de l'estimateur, s(t) est fonction de l'ensemble des valeurs {x(u) ; u ∈R}, le principe d'orthogonalité s'écrit :

$$\forall u \in R, \qquad E\left[(s(t) - \hat{s}(t))\, x(t+u)\right] = 0 \qquad \textbf{(I.12)}$$

En raison de l'indépendance entre x(t) et n(t), on a :

$$E[s(t)x(t+u)] \triangleq R_{sx}(u) = E[s(t)(h*s)(t+u)]$$
$$= \int h(\tau)E[s(t)s(t+u-\tau)]d\tau$$
$$= \int h(\tau)Rs(u-\tau)d\tau \qquad \textbf{(I.13)}$$

De manière similaire $E[\hat{s}(t)x(t+u)]$est une fonction de u seulement qui s'exprime comme :

$$E[\hat{s}(t)x(t+u)] = E[(w*x)(t)x(t+u)]$$
$$= \int w(\tau)E[x(t+u)x(t-\tau)]d\tau$$
$$= \int w(\tau)R_x(u+\tau)d\tau \qquad \textbf{(I.14)}$$

En faisant le changement de variable τ →−τ dans la dernière intégrale, on obtient :

$$E[\hat{s}(t)x(t+u)] = \int w(-\tau)R_x(u-\tau)d\tau \qquad \textbf{(I.15)}$$

D'après (I.12), les expressions (I.13) et (I.15) sont égales. En passant au domaine fréquentiel, on obtient donc l'identité :

$$H(\omega)\Gamma_s(\omega) = W^*(\omega)\Gamma_x(\omega) \textbf{(I.16)}$$

Ce qui conduit à :

$$W(\omega) = H^*(\omega)\frac{\Gamma_s(\omega)}{\Gamma_x(\omega)} \textbf{(I.17)}$$

Où l'on a utilisé le fait que la densité spectrale de puissance d'un processus à valeurs réelles est réelle et paire. Il suffit maintenant d'utiliser le fait que Γx(ω) s'exprime comme :

$$\Gamma_x(\omega) = |H(\omega)|^2\Gamma_s(\omega) + \Gamma_n(\omega) \qquad \textbf{(I.18)}$$

Pour aboutir à l'expression classique du filtre de Wiener dans le domaine spectral [4] :

$$W(\omega) = \frac{H^*(\omega)\,|H(\omega)|^2}{|H(\omega)|^2 + (\Gamma_n(\omega)\,\Gamma_s(\omega))} \qquad \textbf{(I.19}$$

I.7.3 Débruitage par ondelettes

L'ondelette est une oscillation en forme d'onde avec une amplitude qui commence à zéro, elle augmente, puis diminue jusqu'à zéro, elle fournit une analyse du signal qui est localisée à la fois en temps et en fréquence **[24]**. Cette analyse temps-fréquence la mène à appartenir au groupe de méthodes d'analyse multi-échelles. Le principe de base consiste à convoluer le signal à analyser avec une fonction appelée ondelette (wavelet) **[25]**. La fonction d'ondelette est exprimée par

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) \qquad \textbf{(I.20)}$$

Le paramètre d'échelle ou de dilatation 'a' correspond à une information de fréquence, et le paramètre 'b' se rapporte à l'emplacement de la fonction d'ondelettes comme s'il est décalé à travers le signal. La figure I-6 montre deux exemples d'ondelettes appelées chapeau mexicain et ondelette de Morlet.

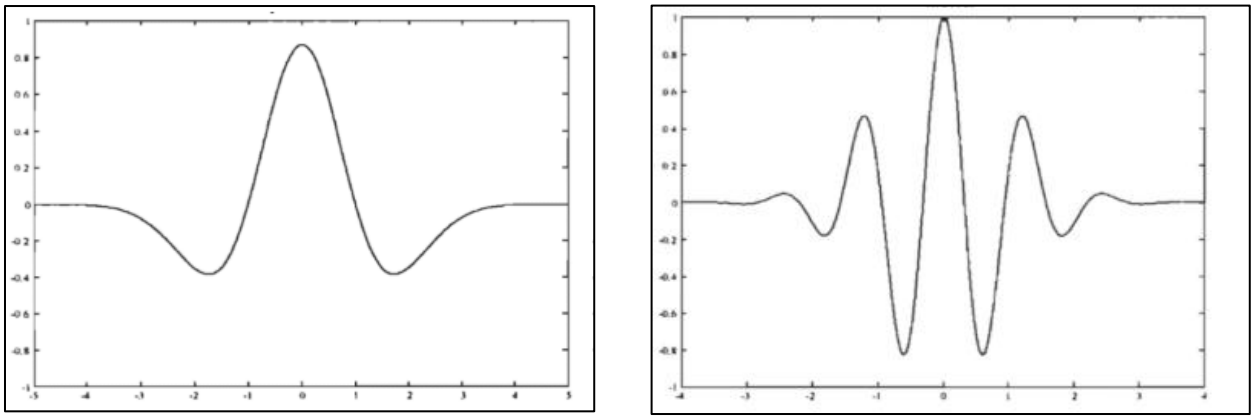


Figure I-6 : Exemple d'ondelettes : (a) Chapeau mexicain (mexicanhat) (b) Ondelette de Morlet.

I.7.4 La Transformée de Fourrier (TF)

- **Définition et limites :**

La transformée de Fourier est un outil mathématique permettant de convertir la représentation temporelle d'un signal en sa représentation fréquentielle. Son expression est la suivante :

$s(t) \longrightarrow S(\omega) = \int s(t)e^{-j\omega t dt}$

$S(\omega)$Peuts'écrire

$S(w) = \int f(t)cos(wt)dt + j\int f(t)sin(wt)dt$

$S(w) = A(w) + jB(w)$Telque $A(w) = \int f(t)cos(wt)dt$ et

$B(w) = \int f(t) sin(wt) dt$ ***(I.21)***

$S(\omega) = \mid S(\omega) \mid e^{-j\phi}$ Avec $\mid S(\omega) \mid = [A(\omega) + B(\omega)]^{\frac{1}{2}}$:spectred'amplitudede $S(t)$.

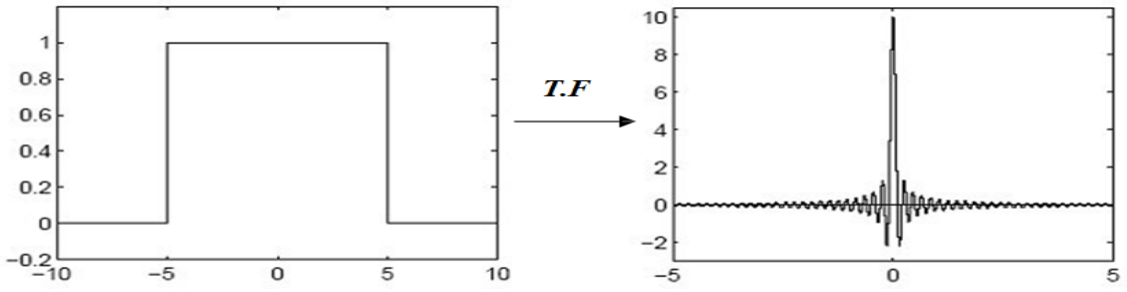
$\phi = arctan\ \frac{(B(\omega)}{(A(\omega))}$:Spectre de phase de $S(t)$.

Malgré son importance indéniable, la transformée de Fourier présente plusieurs limitations, en particulier son manque évident de localisation temporelle. En effet, bien que l'analyse de Fourier permette de déterminer les différentes fréquences présentes dans un signal, c'est-à-dire son spectre de fréquence, elle ne permet pas de connaître le moment précis où ces fréquences ont été émises. Cette analyse fournit une information globale et non locale, car elle utilise des fonctions d'analyse telles que des sinusoïdes qui oscillent indéfiniment sans s'amortir.

Cette perte de localité n'est pas problématique pour l'analyse des signaux dont la structure est statique ou peu variable dans le temps, mais elle devient un obstacle pour l'étude des signaux non stationnaires. En effet, dans ce cas, l'information portée par l'évolution temporelle du signal est perdue lors de sa représentation fréquentielle.

- **Exemple :**

Si on prend l'exemple de la figure (I.2), on remarque que lors du passage de la représentation temporelle figure (I.2.A) à la représentation fréquentielle l'information liée au temps est perdue. En effet, dans la représentation fréquentielle figure (I.1.B), il est impossible de savoir à quel instant le pic a eu lieu



Temps(S) Fréquence(Hz)

Figure I-7 : Exemple de la Transformée de Fourrier.

L'analyse de Fourier ne permet pas d'étudier les signaux dont la fréquence varie dans le temps. Pour de tels signaux, il est nécessaire de mettre en place une analyse temps-fréquence qui permettra de localiser les périodicités dans le temps, indiquant ainsi si la période varie de manière continue, si elle disparaît ensuite, etc.

Pour les signaux non stationnaires, une approche couramment utilisée consiste à introduire la notion de stationnarité locale. Cela implique de découper le signal à traiter en segments, de sorte que le signal soit considéré comme stationnaire à l'intérieur de chaque segment, puis d'appliquer la transformée de Fourier à chaque segment. Cette représentation est connue sous le nom de Transformée de Fourier à Court Terme (TFCT).

I.7.5 Transformée de Fourrier à Court Terme (T.F.C.T)

Pour pallier aux limites de la T.F, GABOR dans les années 1940définit la première forme de la représentation temps-fréquence (T.F.C.T). La technique consiste à découper le signal en tranches successives de telles sorte que le signal puise, dans la durée de chaque tranche, être considère comme stationnaire chaque tranche de durée T et obtenue en multipliant le signal par une fenêtre temporelle de la largeur finie, elle est considère comme stationnaire. On peut alors appliquer la T.F, non pas au signal globale, mais à chacune des tranches du ce signal

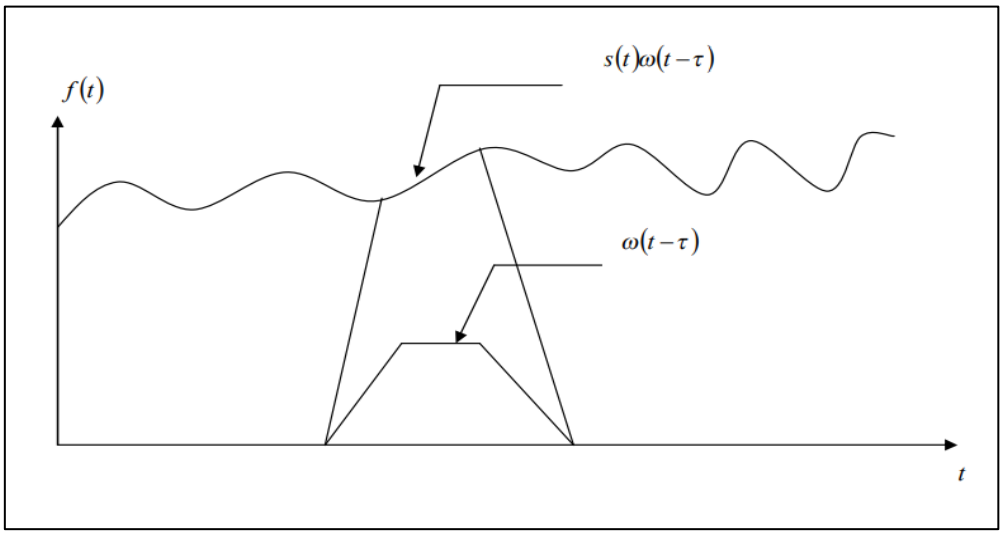


Figure I-8 : Influence de la fenêtre GABOR

$$s(t) \xrightarrow{\text{TFCT}} S(t,\omega) = \int \omega(t-\tau)s(t)e^{-2j\pi\omega\tau d\tau}. \qquad \textbf{(I.22)}$$

- S(t) : Signal à traiter.
- τ : Translation de la fenêtre.
- $\omega(t-\tau)$: La fonction introduisant la notion de fenêtre de GABOR.

Malheureusement cette représentation (T.F.C.T) présente les inconvénients suivants :

❖ La longueur de la plage étant fixée, il n'est pas possible d'analyser simultanément des phénomènes dont les échelles de temps sont différentes.

❖ La précision de l'analyse impose un compromis insurmontable entre la résolution temporelle et la résolution fréquentielle.

❖ La détermination de l'intervalle temporel dans lequel on peut considérer le signal comme stationnaire, est un inconvénient pratique. Pour remédier Aux limites de la T.F.C.T, MORLET a choisi une autre méthode. Au lieu garder fixe la taille de la fenêtre et de varier le nombre d'oscillations à l'intérieur de cette fenêtre, il a fait l'inverse : Il gardé constant le nombre d'oscillations, et fait varier la taille de la fenêtre, l'étirant ou la compriment comme un accordéon. Donc, MORLET pouvait alors localiser les hautes fréquences avec des fenêtres plus larges.

❖ Les signaux intéressants présentent, en général, de nombreuses caractéristiques non stationnaires qui constituent une part importante de l'information contenue dans la série : Tendance, rupture, début et fin d'événement, phénomènes transitoires.

Conclusion

Dans ce chapitre, nous avons introduit les généralités sur le signal de parole, en explorant les mécanismes de sa production ainsi que les divers types de bruits qui peuvent le perturber. En poursuivant, nous avons examiné les différentes méthodes de réduction du bruit affectant le signal, telles que le filtre de Wiener, la soustraction spectrale et le débruitage par ondelettes et l'analyse temps-fréquence (TF) et aussi la transformée de Fourier à court terme (TFCT) dans la caractérisation et le traitement des signaux de parole.

Chapitre 2 Généralités sur l'IA et le Deep Learning

Introduction

L'intelligence artificielle est présente partout, nous nous en servons déjà dans notre vie de tous les jours mais sans s'en rendre compte. Elle a donné naissance à plusieurs techniques qui sont utilisées globalement dans presque tous les domaines. C'est un sujet qui prend place dans un contexte où le numérique est omniprésent dans notre quotidien et les nouvelles technologies engagent de grands changements dans nos modes de vie. Notamment les deux branches de l'IA : Machine Learning et le Deep Learning attirent beaucoup l'attention, et pour cause, le niveau de performance atteint est tout simplement extraordinaire. Dans ce chapitre, nous allons décrire les concepts de l'intelligence artificielle ainsi que son fonctionnement, et nous aborderons en détail l'apprentissage automatique en donnant ses différents types illustrés avec des exemples. Puis, nous définirons la notion de l'apprentissage profond en expliquant son fonctionnement et citant ses domaines d'application, pour arriver par la suite à distinguer entre ces deux terminologies. Vers la fin, nous allons voir à propos de Deep Learning, ses avantages et ses limitations.

II.1 Intelligence artificielle

II.1.1 Définition

Définir l'intelligence artificielle (IA) n'est pas chose facile, le champ est si vaste qu'il est impossible de la restreindre à un domaine de recherche spécifique. Néanmoins plusieurs définitions ont été attribuées à cette notion :

Selon le Larousse, l'intelligence Artificielle est l'ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence.

Marvin Lee Minsky l'un des créateurs de l'IA la définit comme : « La construction de programmes informatiques qui s'adonnent à des taches qui sont, accomplies de façon plus satisfaisantes 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 ».

Plus précisément, l'intelligence artificielle est un ensemble d'algorithmes qui traitent un ensemble d'informations ou de données, relatives à des taches, de manière semblable ou

identique à celle qu'adopterait un être humain pour prendre une décision ou résoudre un problème.

Le point commun entre ces définitions se résume dans l'objectif majeur de l'IA qui présente son ambition d'imiter les processus cognitifs de l'être humain. Ces processus comprennent l'apprentissage (acquisition d'informations et de règles liées à leur utilisation), le raisonnement (application des règles pour parvenir à des conclusions approximatives ou précises) et l'auto-correction **[26]**. L'IA vise aussi un peu loin, cela an de mettre au point des systèmes qui résolvent certains problèmes bien mieux que les humains, par tous les moyens disponibles.

II.1.2 Fonctionnement

Selon Harry Shum, Président Exécutif de Microsoft : « l'IA fonctionne seulement s'il y a présence d'une vaste quantité de data, d'une puissance informatique extraordinaire, notamment grâce au cloud, et des algorithmes révolutionnaires, basés sur le Deep Learning ». Il existe de nombreuses divisions des types d'intelligence artificielle. Nous nous focalisons sur 2 types :

- **IA simple (ou faible) :** conçue pour réaliser des tâches prédéfinies et spécifiques. Par exemple les assistants virtuels par voix de nos Smartphones.
- **IA complexe (ou fort) :** imitante les capacités cognitives humaines. Ce type d'IA, plus avancé, implique la recherche de solutions à des tâches inconnues, sans avoir préalablement défini de solutions pour ces tâches.

L'intelligence artificielle est basée sur des données et des algorithmes qui fonctionnent à partir d'eux grâce au processus détaillé ci-dessous :

- Identifier l'importance du problème.
- Analyser les situations passées et étudier toutes les variables possibles liées au problème qu'on souhaite analyser.
- Grâce à un système de statistiques, prédire le résultat de ce problème, toujours à partir de données connues.
- Une fois que le système a toutes les données, il fournit la solution la plus réaliste au problème. Ainsi, l'IA apprend à résoudre automatiquement, dans le futur, un problème semblable.

D'une manière ou d'une autre, l'IA est présente dans presque tous les domaines : éducation, commerce, santé, finance, juridique, industriel, les jeux de réflexion, la recherche mathématique, les assistants personnels et la domotique, la reconnaissance faciale, la

compréhension des langues, et la robotique. Depuis quelques années, on associe presque toujours l'intelligence aux capacités d'apprentissage. C'est grâce à l'apprentissage qu'un système intelligent capable d'exécuter une tâche peut améliorer ses performances avec l'expérience. C'est grâce à l'apprentissage qu'il pourra apprendre à exécuter de nouvelles tâches et acquérir de nouvelles compétences.

II.2 Machine Learning

L'apprentissage automatique (Machine Learning en anglais) est un sous domaine de l'intelligence artificielle. Il fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage, et aussi de remplir des tâches qu'il est difficile ou voire impossible de les remplir par des moyens algorithmiques classiques.

Arthur Samuel, un pionnier dans ce domaine a défini le ML comme : « le domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmé ». Pour être plus clair, ce que fait le ML, c'est apprendre à résoudre un problème de manière automatique en utilisant les données.

[2]

II.2.1 Fonctionnement

L'apprentissage automatique se fait en deux étapes. La première est la phase d'apprentissage qui s'agit de l'extraction de l'information pertinente de données étudiées au fil d'un processus de mise-à-jour appelé entraînement. Une fois que le modèle est bien déterminé, vient la phase de déploiement où de nouvelles données sont introduites afin de réaliser la tâche souhaitée.

Les algorithmes de l'apprentissage automatique fonctionnent en construisant un modèle à partir d'entrées d'exemple afin de faire des prédictions ou des décisions basées sur les données, dont l'objectif est de minimiser ce qu'on appelle l'erreur, c'est-à-dire de se tromper le moins possible.

La machine learning et l'IA sont souvent abordés ensemble, et les termes sont parfois utilisés de manière interchangeable, mais ils ne veulent pas dire la même chose. Une distinction importante est que, même si l'intégralité de la machine learning repose sur l'intelligence artificielle, cette dernière ne se limite pas à la machine learning.

II.2.2 Domaine d’application

De nos jours, cette technologie est présente dans divers domaines, Lorsque nous interagissons avec les banques, achetons en ligne ou utilisons les médias sociaux, des algorithmes de machine Learning entrent en jeu pour optimiser, fluidifier et sécuriser notre expérience. Elle est représentée notamment dans les robots marcheurs qui apprennent seuls. Le ML est aussi utilisé dans le domaine médical an d'assister au mieux les spécialistes, et tant d'autres utilisation tels que :

- Traitement d'image et vision par ordinateur : la reconnaissance faciale, la détection d'objets.
- Biologie : la détection des tumeurs.
- Traitement du langage naturel : les applications de reconnaissance vocale.
- Finance : Détection de fraudes.
- Automobile : pour la maintenance prédictive.
- Agriculture : la prévision des demandes en eau.
- Réseaux : Prédiction du trafic (Volume de trafic attendu), classification du trafic (Dropbox, Facebook, LinkedIn, Skype, YouTube), prédiction des pannes, sécurité du réseau. **[27]**

II.2.3 Type d’apprentissage automatique

En général, dans le domaine de l'apprentissage automatique, les problèmes à résoudre sont souvent regroupés en différentes catégories. Ces catégories sont établies en fonction de la méthode d'apprentissage utilisée ou de la manière dont les informations sont fournies en retour au système en développement. Il existe diverses approches en matière d'apprentissage automatique. Lorsque le système dispose d'exemples étiquetés et prédéterminés pour chaque classe, il apprend à classifier selon un modèle spécifique, ce qui correspond à ce que l'on appelle l'apprentissage supervisé. En revanche, lorsque le système ne dispose que d'exemples non étiquetés et que le nombre ainsi que la nature des classes ne sont pas définis à l'avance, on parle alors d'apprentissage non supervisé. Dans ce contexte, l'algorithme ne reçoit pas de données étiquetées et doit découvrir une structure ou des motifs dans les données fournies. En outre, il existe d'autres méthodes telles que l'apprentissage par renforcement et l'apprentissage semi supervisé, que nous examinerons plus en détail par la suite.

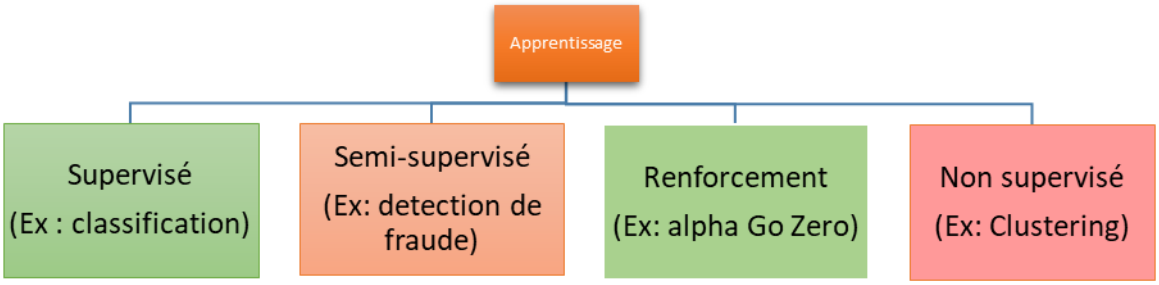


Figure II-1 : Type d'apprentissage en ML.

II.2.3.1 L'apprentissage supervisé

L'apprentissage supervisé représente une branche du Machine Learning qui repose sur l'utilisation d'un jeu de données d'entraînement étiqueté afin de créer des modèles d'intelligence artificielle. Son objectif principal est d'apprendre en comparant les sorties réelles du modèle avec les sorties prévues, ce qui permet d'identifier les erreurs et d'ajuster le modèle en conséquence. En d'autres termes, l'apprentissage supervisé se base sur des modèles pour prédire les valeurs des étiquettes pour un ensemble de données non étiquetées. La figure (II-2) illustre le fonctionnement de ce processus.

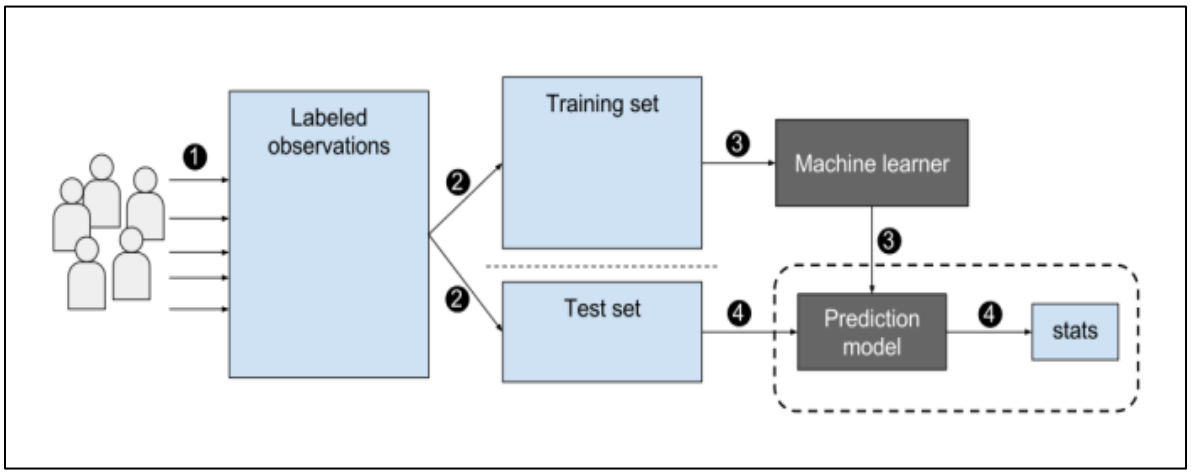


Figure II-2 : Fonctionnement de l'apprentissage supervisé**.**

II.2.3.2 L'apprentissage non supervisé

Dans le cas de l'apprentissage non supervisé, on utilise un ensemble de données d'entrées non étiquetées, an de laisser l'algorithme d'apprentissage trouver tout seul les points communs parmi cet ensemble de données. Les méthodes d'apprentissage automatique qui facilitent l'apprentissage non supervisé sont particulièrement utiles, vu que les données non étiquetées étant plus considérables que celles étiquetées. On peut considérer que l'objectif

initial de l'apprentissage non supervisé est aussi simple que de détecter les modèles cachés dans un ensemble de données, mais il peut aussi avoir un objectif d'apprentissage des caractéristiques, ce qui va rendre la machine intelligente capable de découvrir automatiquement les représentations nécessaires pour classer des données brutes.

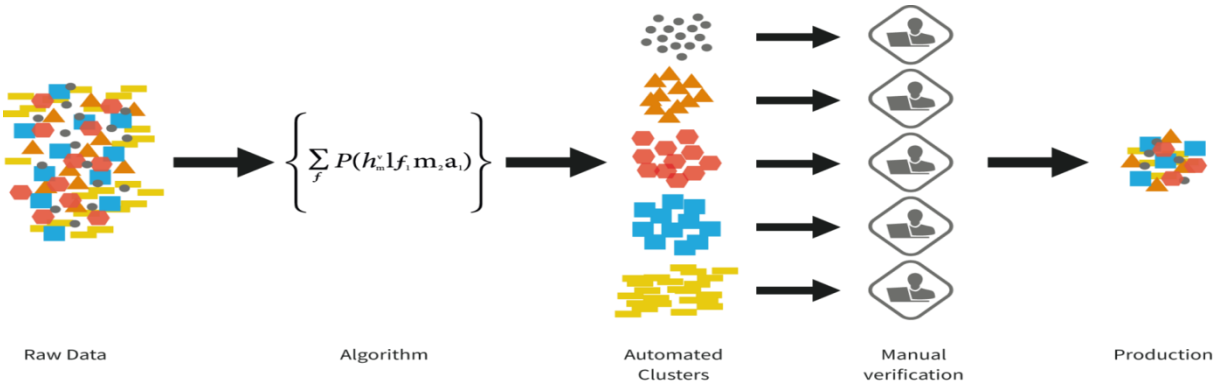


Figure II-3 : Fonctionnement de l'apprentissage non supervisé.

II.2.3.3 L'apprentissage par renforcement

L'apprentissage par renforcement constitue une catégorie de problèmes d'apprentissage automatique qui se base sur l'expérimentation successive pour déterminer les actions optimales à entreprendre. Contrairement à d'autres techniques d'apprentissage, les machines intelligentes engagées dans cette approche n'ont pas simplement des instructions préalables sur les actions à exécuter. Elles explorent plutôt diverses situations pour identifier les actions les plus avantageuses. L'apprentissage par renforcement représente un modèle comportemental dans lequel l'algorithme analyse les données pour guider l'utilisateur vers des résultats optimaux. Contrairement à l'apprentissage supervisé traditionnel, où le système est entraîné sur un ensemble de données, l'apprentissage par renforcement repose sur l'expérimentation et l'apprentissage par essais et erreurs. Imaginons un drone autonome chargé de livrer un colis depuis un entrepôt jusqu'à une maison. Le drone est le héros de cette histoire. Il peut faire plein de choses : avancer, reculer, monter, descendre, accélérer, freiner... Chaque mouvement qu'il fait change à la fois sa propre situation et celle de ce qui l'entoure. Son seul but : atteindre sa destination en moins de 30 minutes, puis revenir tranquillement. Pour y parvenir, il part à l'aventure, fait des choix au fur et à mesure, teste différentes stratégies. Petit à petit, il apprend ce qui marche le mieux. Après beaucoup d'essais, il réussit finalement sa mission de manière brillante. C'est également cette méthode qui sera exploitée pour construire les algorithmes des voitures autonomes. L'entraînement d'une voiture autonome est un processus extrêmement complexe à cause de nombreux obstacles possibles.

Si toutes les voitures étaient autonomes, les essais et les erreurs seraient plus faciles à surmonter, mais dans le monde réel, les facteurs humains sont souvent imprévisibles. **[28]**

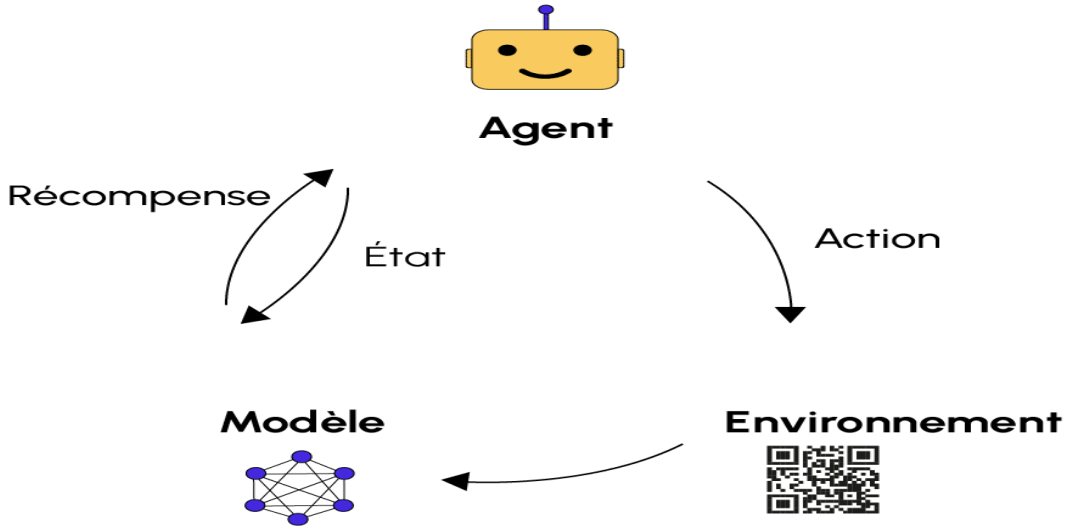


Figure II-4 : Fonctionnement de l'apprentissage par renforcement.

II.2.3.4 Apprentissage semi-supervisé

Nous avons préalablement vue l'apprentissage supervisé et non supervisé, dont la majeure différence réside dans le fait que les données soient étiquetées ou non, et à cela s'ajoute les méthodes adéquates utilisées pour traiter ses données.

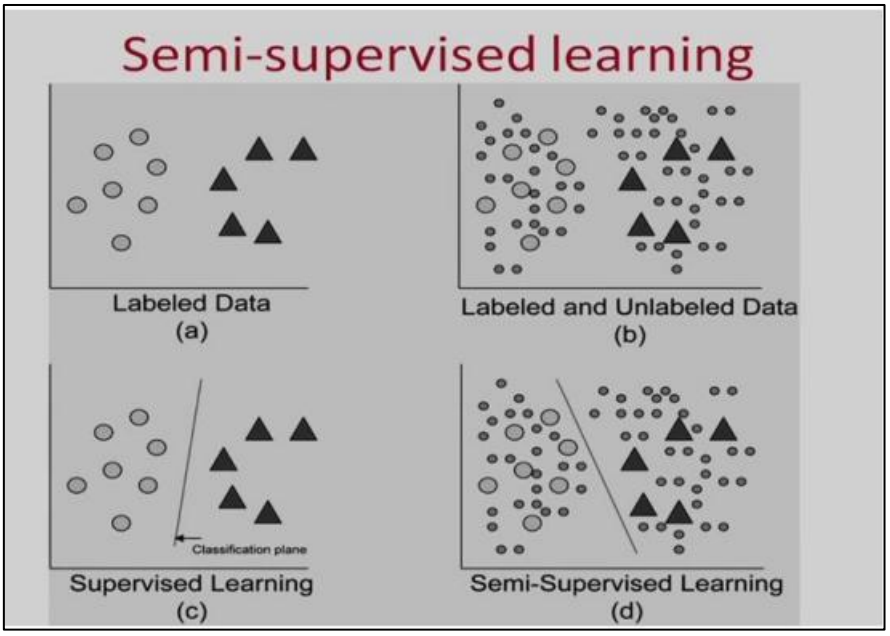


Figure II-5 : Fonctionnement de l'apprentissage Semi-supervisé.

L’apprentissage semi-supervisé regroupe ses deux principes, il prend un ensemble réduit de données étiquetées avec un autre ensemble de données non étiquetées du même type. L’avantage de ce type d’apprentissage réside principalement dans le processus d’étiquetage des données prend beaucoup de temps et souvent coûteux.
Donc paradoxalement le non étiquetage devient bénéfique pour le processus d’apprentissage, et la construction du modèle et moins coûteuse. **[29]**

II.3 Deep Learning

Deep Learning ou apprentissage en profondeur ou DL est une branche du Machine Learning entièrement basée sur des réseaux de neurones artificiels. **[30].** Le concept d'apprentissage en profondeur existe depuis plusieurs années, mais il a été laissé à l'abandon faute de moyens nécessaires. Dans le milieu des années 2000 le Machine Learning fait rage dans les compétitions de reconnaissance visuelle, en 2012 Deep mind une startup dans le domaine de l’IA arrive dans la compétition avec un algorithme de Deep learning qui bat largement tous les autres compétiteurs, l’année suivante tous les compétiteurs se sont tournés vers le Deep Learning au vu des résultats obtenus. L'avancée du DL est dû à l’augmentation en exponentiel qu'ont connu les machines en capacités de calculs, et de stockages, ainsi que la disponibilité de données de masses (big data), ses 3 ingrédients étaient nécessaires pour exploiter le potentiel du DL qui fût chose impossible dans les années 90.Les pionniers qui ont soutenus le DL tel que Geoffrey Hinton, ou alors Yoshua Bengio qui a développé les réseaux GAN(generativeadversal networks), Yann leCun qui est au cœur d’une avancée fulgurante dans le domaine de reconnaissance d’images avec les réseaux Convolutionnels CNN, et son architecture LetNet. Geoffrey Hinton a prouvé que l'apprentissage profond pouvait résoudre des problèmes insolubles par d'autres approches. **[31]**

II.3.1 Fonctionnement

Le Deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du fonctionnement des neurones biologiques du cerveau humain. Cette structure est disposée en plusieurs couches interconnectées entre elles. La première couche correspond aux neurones d'entrée et la dernière transmet les résultats de sortie. Entre ces deux se trouvent plusieurs couches intermédiaires par lesquelles l'information est traitée. Cette architecture est propre au Deep Learning et permet que chaque couche analyse de manière plus précise les données d'entrée. Ainsi, plus le réseau de neurones artificiels est profond et donc contient plusieurs

couches, plus le système peut effectuer des tâches complexes. Il est capable de déterminer par lui-même une représentation de ce qu'il reçoit, que ce soit une image ou un texte. Par exemple, à partir de portraits humains, le programme va d'abord distinguer le visage des cheveux, puis reconnaître le nez, la bouche, les yeux.

À chaque information intégrée, les connexions entre neurones s'étendent et se modifient. C'est pour cela qu'un système avec un IA à apprentissage profond a la capacité d'apprendre de nouvelles choses en autonomie. Il améliore également de lui-même ses prévisions et ses prises de décision, sans qu'aucune intervention humaine ne soit requise. Il a donc pour particularité d'apprendre de ses propres erreurs. **[32]**

II.3.2 Domaine d'application du Deep Learning

Depuis une décennie, le Deep Learning connaît une croissance fulgurante et se révèle être un outil précieux dans divers secteurs, offrant des résultats remarquables dans de nombreuses applications. Passons en revue quelques-unes de ces applications :

- Les IA à Deep Learning sont très efficaces pour les analyses d'images. Elles sont par exemple employées dans l'imagerie médicale pour détecter des maladies ou dans le secteur automobile dans le cas des voitures autonomes. Mais aussi pour les reconnaissances faciales comme sur les Smartphones ou sur Facebook.
- De plus, le Deep Learning est également un atout dans la création de contenu. En effet, un ordinateur peut être capable de rédiger de manière autonome des textes ou d'effectuer des traductions. La seule condition est l'accès à une quantité de données suffisante de formation. Cela fait partie du NLP (Natural Language Processing) une branche de l'IA, qui traite automatiquement le langage humain.
- C'est également le cas des assistants vocaux, tels que Siri, Alexa ou Google Home. Ceux-ci se fondent sur la technologie du Deep Learning pour développer leur compréhension du langage et leur vocabulaire. Tout comme les chatbots qui permettent de répondre de plus en plus précisément aux diverses demandes des clients.
- Par ailleurs, l'apprentissage profond trouve toute sa place dans le marketing. Il facilite l'élaboration de campagnes publicitaires et d'e-mails ultra personnalisés. Il peut aussi servir à optimiser le score des leads, à classer et à faire remonter les problèmes des clients.
- D'autre part, il est utilisé en sécurité informatique pour identifier les dangers documentés et les risques inconnus. Il est en effet, capable de détecter des anomalies et de renforcer les mesures de sécurité.

♦ Également, l'apprentissage profond est très présent dans le domaine industriel. Que ce soit en matière de robotique ou dans les solutions de maintenance. De toute évidence, ce ne sont qu'une petite partie des vastes applications auxquelles l'apprentissage en profondeur peut être appliqué. On pourrait citer encore beaucoup d'exemple plus originaux les uns que les autres, mais ce qu'il faut retenir c'est que le Deep Learning permet de faire apprendre à un ordinateur une tache précise en observant un grand nombre d'exemples.

II.4 Machine Learning vs Deep Learning

La majeure différence qu'on note entre ses 2 concepts provient de la manière dont les données sont présentées au système (modèle).

♦ Les algorithmes de ML nécessitent presque toujours des données structurées, alors que les réseaux d'apprentissage approfondis reposent sur des couches de réseaux de neurones artificiels (RNA).

♦ On voit aussi une différence au sein de l'architecture des modèles qui les composent, on note que les modèles type DL sont plus profond que les modèles type ML.

♦ Deep Learning n'utilise que les réseaux de neurones, alors que pour le ML les réseaux de neurones sont qu'une approche de conception des modèles parmi tant d'autres. En considérant le fait que le DL est la prochaine étape de l'évolution du ML inculquant aux machines la manière de prendre leurs décisions de façon précise sans l'intervention de l'expert humain.

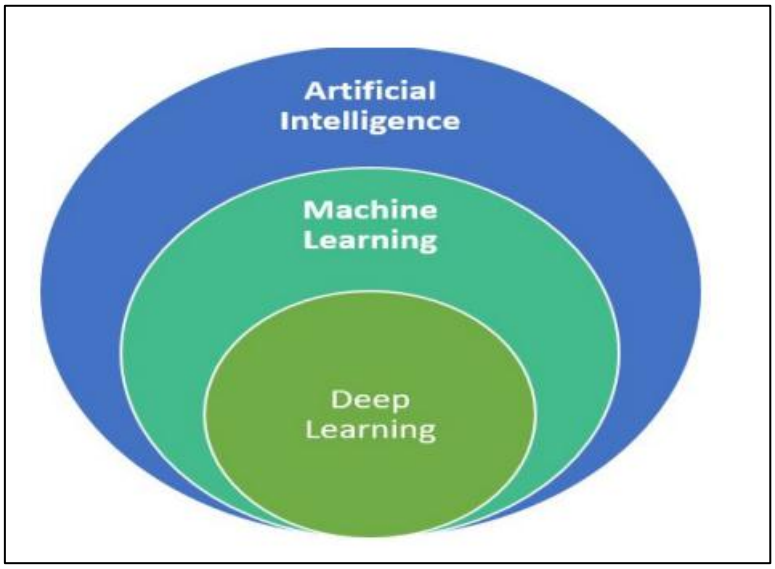


Figure II-6 : IA vs ML vs DL.

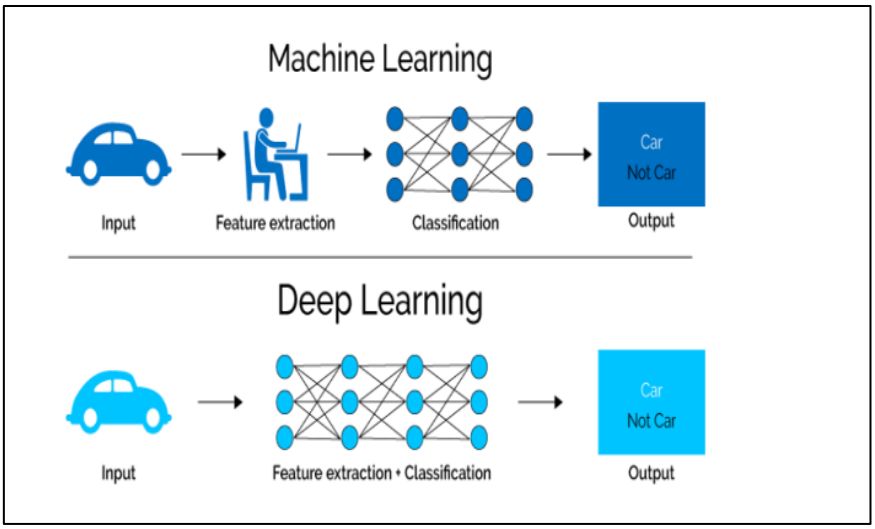


Figure II-7 : ML vs DL.

II.5 Réseaux Neuronaux Artificiels

Un RNA (Réseau de neurone artificiel) peut être considéré comme une boite noire, qui reçoit des signaux d'entrée et produit des signaux de sortie c'est un modèle mathématique composé d'un grand nombre d'éléments de calculs (neurones) opérant en parallèle et organisée sous forme de couches interconnectées **[33]**.

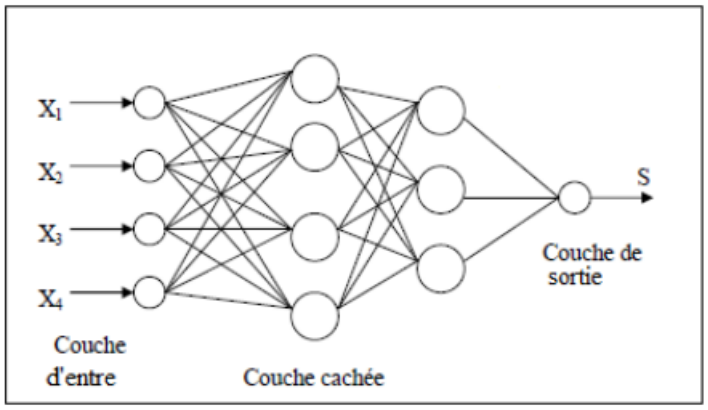


Figure II-8 : Forme générale d'un Réseaux de neurone.

Nous pouvant distinguer trois types de couches dans un RNA :

- La première couche est appelée couche d'entrée. Elle recevra les données source que l'on veut utiliser pour l'analyse. Sa taille est donc directement déterminée par le nombre de variables d'entrées.
- La deuxième couche est la couche cachée. Dans cette couche, les fonctions d'activation sont en général non linéaires. Le choix de sa taille (nombre de neurones)

n'est pas automatique et doit être ajusté. Il sera souvent préférable pour obtenir la taille optimale, d'essayer le plus de tailles possibles.

- La troisième couche est appelée couche de sortie. Elle donne le résultat obtenu après compilation par le réseau des données entrée dans la première couche. Sa taille est directement déterminée par le nombre de variables dont on a besoin en sortie.

Les réseaux de neurones artificiels regroupent en réseaux un certain nombre de neurones formels connectés entre eux de diverses manières. Un réseau est défini par sa topologie, qui représente le type de connexion existant entre les divers neurones du réseau, par la fonction d'activation qui le caractérise et par les méthodes d'apprentissage utilisées pour trouver une relation non linéaire optimale par approximation entre les variables d'entrées et de sorties.

II.5.1 Topologie des réseaux de neurones

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle **[34]**. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité.

Parmi les topologies de réseaux de neurones les plus utilisées dans ce sens, nous citons les Suivantes :

II.5.1.1 Réseaux de neurones multicouches (MLP)

Ces réseaux sont composés de plusieurs couches de neurones, comprenant au moins une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie.

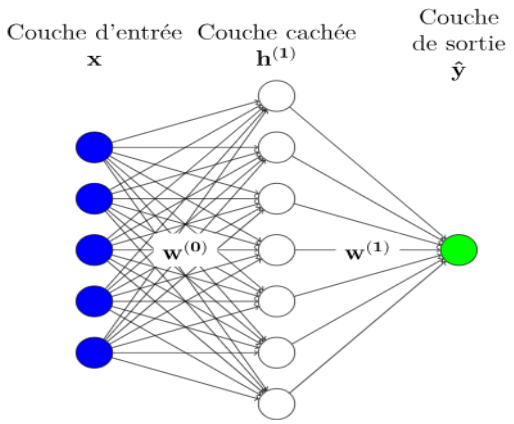


Figure II-9 : Réseaux de Neurones Multicouches.

II.5.1.2 Réseaux de neurones récurrents (RNN)

Ces réseaux sont conçus pour traiter des données séquentielles en utilisant des connexions récurrentes qui permettent aux informations de circuler dans le réseau sur plusieurs étapes de temps.

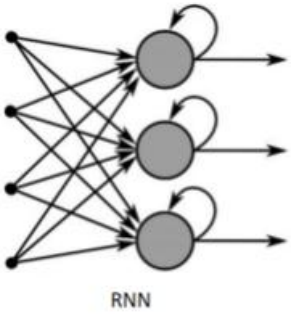


Figure II-10 : Réseaux de Neurones Récurrents.

II.5.1.3 Réseaux de neurones adversariaux (GAN)

Ces réseaux sont composés de deux réseaux neuronaux en compétition : un générateur et un discriminateur. Ils sont utilisés pour générer de nouvelles données réalistes en apprenant à partir d'un ensemble de données d'entraînement.

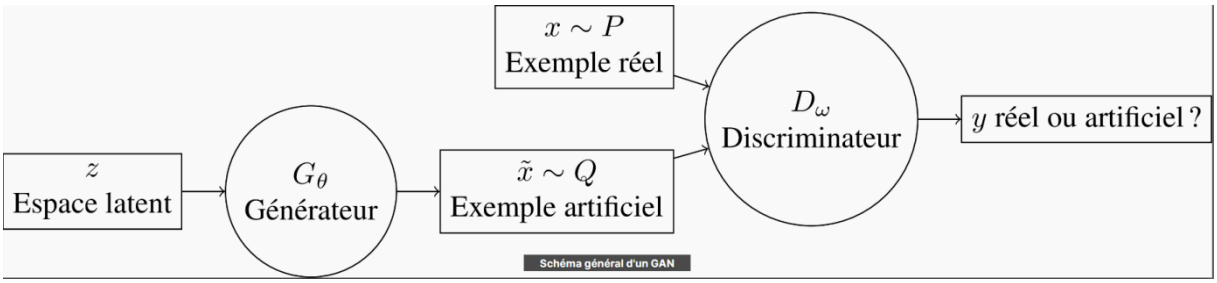


Figure II-11 : schéma général d'un GAN.

II.5.1.4 Réseaux de neurones convolutifs CNN

Un réseau de neurones convolutif (CNN de l'anglais Convolutional Neural Network) est une classe de réseaux neuronaux artificiels dont l'architecture des connexions est inspirée de celle du cortex visuel des mammifères. Les réseaux de neurones à convolution sont capables de catégoriser les informations des plus simples aux plus complexes. Ils consistent en un empilage de plusieurs couches de neurones, des fonctions mathématiques à plusieurs paramètres ajustables, qui prétraitent de petites quantités d'informations. Les CNNs sont

caractérisés par leurs quatre types de couches pour un réseau de neurones convolutif : la couche de convolution, la couche de pooling, la couche de correction ReLU et la couche entièrement connectée (ou fully-connected, FC).

- La couche de convolution (CONV) qui traite les données d'un champ récepteur.
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
- La couche de correction (ReLU), souvent appelée par abus ReLU en référence à la fonction d'activation (Rectified Linear Units présentée précédemment).
- La couche entièrement connectée (FC), qui est une couche de type perceptron.
- La couche de perte (LOSS). Les CNN sont en général les modèles les plus performants pour la classification d'images et l'analyse d'images visuelles.

II.5.1.5 La description d'architecture de modèle CNN

- **Les couches de convolution (Convolutional layers):** Chaque couche applique une série de filtres de convolution à l'entrée pour extraire des caractéristiques à différentes échelles et niveaux d'abstraction.
- **Les couches de pooling (Pooling layers):** Ces couches réduisent la taille spatiale des cartes de caractéristiques en agrégeant les informations les plus importantes tout en préservant les caractéristiques les plus saillantes.
- **Les couches entièrement connectées (Fully connected layers):** Ces couches prennent en entrée les cartes de caractéristiques obtenues après la dernière couche de pooling et les transforment en une représentation vectorielle de la classe de sortie.
- **Les couches de perte (LOSS) :** Ces couches calcule la différence entre les prédictions du modèle et les valeurs réelles (étiquettes) associées aux données d'entrée.

  La fonction de perte est utilisée pour évaluer les performances du modèle pendant l'entraînement et pour ajuster les paramètres du modèle afin de minimiser cette perte.
- **Les couches de correction (ReLU) :** Après chaque opération de convolution ou de pooling, une fonction d'activation ReLU (Rectified Linear Unit) est appliquée à chaque élément de la carte des caractéristiques.

  La fonction ReLU introduit de la non-linéarité dans le modèle, permettant ainsi d'apprendre des relations complexes entre les caractéristiques extraites.

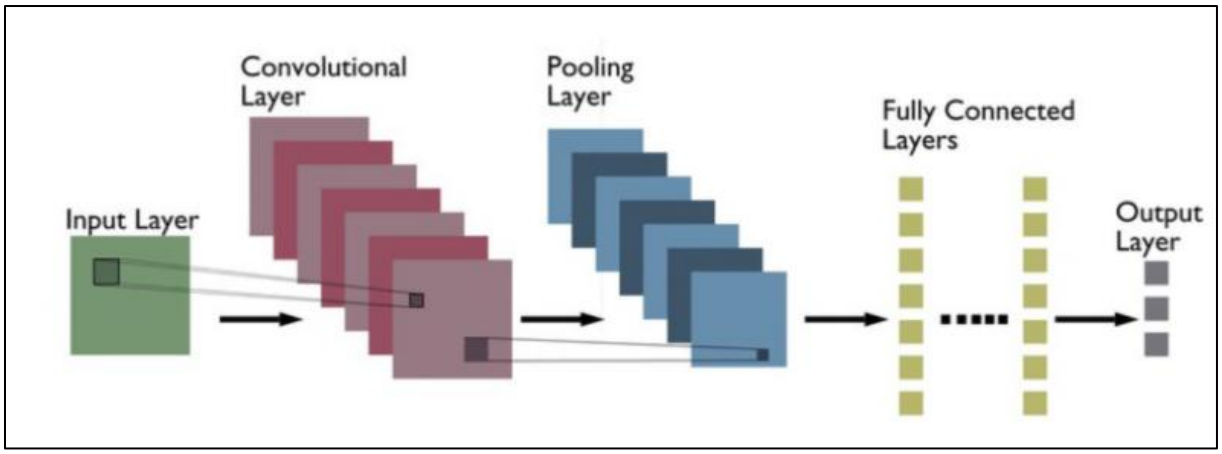


Figure II-12 : Architecture du modèle CNN.

II.5.1.6 Fonctionnement

Les réseaux neuronaux convolutifs (CNN) sont couramment utilisés pour l'amélioration de la parole en raison de leur capacité à extraire des caractéristiques essentielles des spectrogrammes audio. Voici comment un modèle CNN fonctionne dans ce domaine :

La convolution est appliquée en utilisant des filtres pour détecter des motifs spécifiques dans le spectrogramme audio. Ces filtres effectuent des opérations de multiplication et de sommation pour extraire les caractéristiques importantes du signal, telles que les harmoniques et les formants de la voix. Ensuite, des couches de pooling sont utilisées pour réduire la dimensionnalité de la sortie de la convolution en regroupant les valeurs voisines, ce qui permet de conserver les caractéristiques les plus saillantes tout en réduisant la taille de la représentation.

Après chaque couche de convolution ou de pooling, une fonction d'activation non linéaire, comme la fonction ReLU (Rectified Linear Unit), est appliquée pour introduire de la non-linéarité dans le réseau. Cette non-linéarité permet au modèle de capturer des relations complexes et des structures subtiles dans le spectrogramme.

Enfin, des couches entièrement connectées prennent les caractéristiques extraites par les couches convolutives et de pooling et produisent un spectrogramme amélioré en supprimant le bruit et en rehaussant les parties vocales. Pendant l'entraînement, les poids et les biais du réseau sont ajustés itérativement à l'aide de la rétropropagation du gradient et d'un algorithme d'optimisation, comme Adam, pour minimiser une fonction de perte qui mesure la différence entre le spectrogramme amélioré et le spectrogramme de référence propre.

Les CNN sont extrêmement puissants pour l'amélioration de la parole grâce à leur capacité à capturer les motifs locaux dans le spectrogramme et à exploiter la structure temporelle et fréquentielle des données audio.

En utilisant des convolutions, du pooling, des fonctions d'activation non linéaires et des couches entièrement connectées, les CNN transforment des spectrogrammes bruités en spectrogrammes de haute qualité, améliorant ainsi la clarté et l'intelligibilité de la parole.

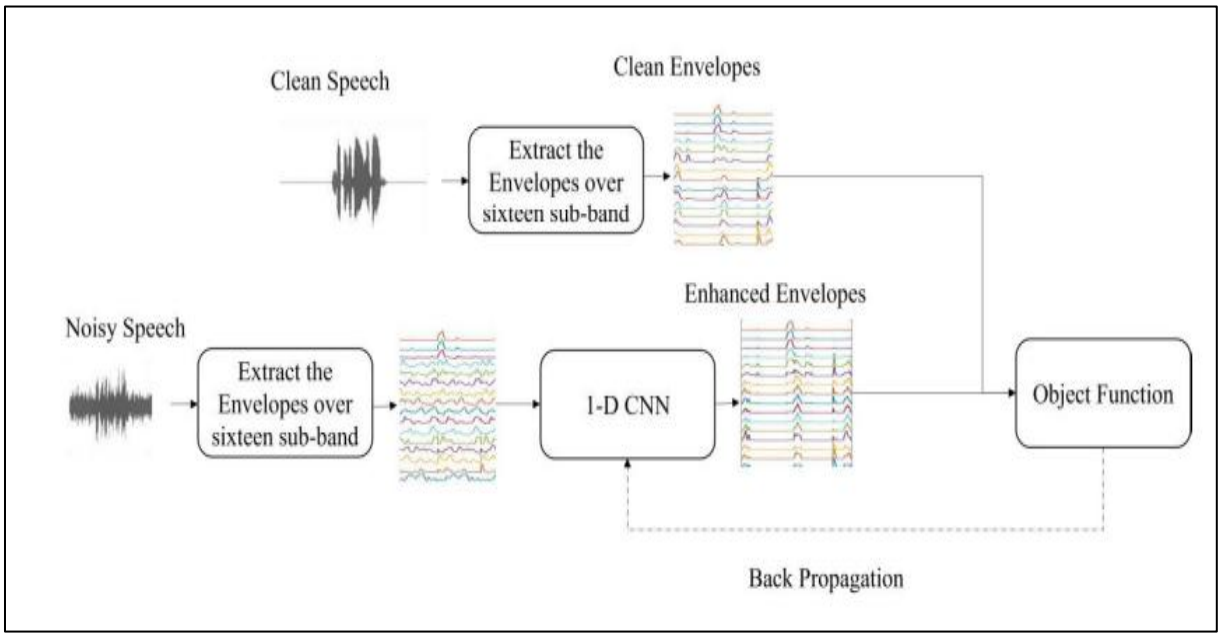


Figure II-13 : Diagramme en bloc de la procédure d'entraînement du modèle CNN 1D.

II.6 Fonctions d'activation

En termes simples, la fonction d'activation est utilisée pour effectuer une transformation non linéaire des données, ce qui permet de modifier leur représentation spatiale. Avec plusieurs couches, chaque couche ayant sa propre fonction d'activation, des transformations complexes et successives des données se produisent, offrant ainsi une perspective nouvelle et plus riche que celle que les humains pourraient obtenir autrement. Il est important de distinguer la fonction d'activation de la fonction de perte, qui est unique et utilisée pour évaluer les performances du modèle.

La fonction d'activation est propre à chaque couche et est non linéaire, permettant ainsi une transformation des données qui ne serait pas possible avec une transformation linéaire. Chaque neurone d'une couche applique la fonction d'activation de cette couche aux données, mais la transformation sera différente pour chaque neurone en raison de ses poids uniques

Il existe de nombreux types de fonctions d'activation, voici quelques exemples :

II.6.1 Fonction ReLU

Actuellement, les fonctions ReLU sont les plus répandues dans les réseaux neuronaux. Elles sont plus légères que les fonctions sigmoïdes et tanh, ce qui les rend plus rapides à entraîner. Cependant, il convient de prendre en compte le phénomène du "DyingReLU", qui peut être évité en utilisant des variantes de ReLU. Les fonctions ReLU sont couramment utilisées dans les réseaux de convolution (CNN), les machines de Boltzmann restreintes (RBM) et les réseaux de perceptrons multicouches. Elles produisent des valeurs dans l'intervalle $(0, +\infty)$.

$$\text{ReLU6}(x) = \min(\max(0, x), 6) \textbf{(II-1)}$$

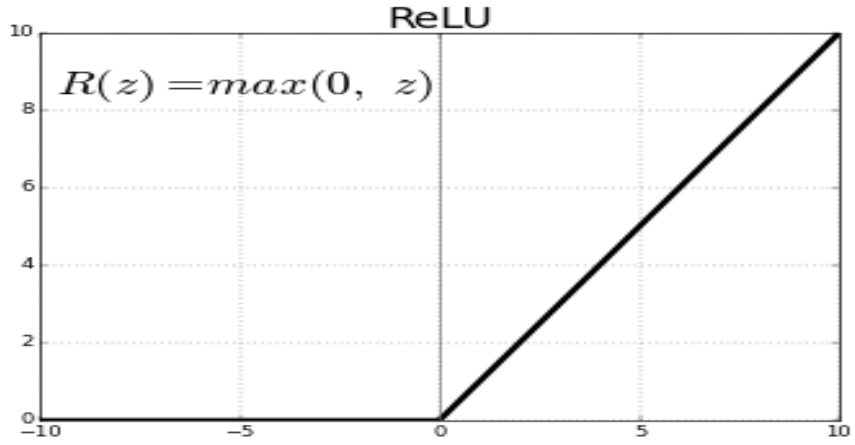


Figure II-14 : Représentation graphique de la fonction ReLU.

II.6.2 Fonction sigmoïde

La fonction d'activation sigmoïde est largement utilisée et populaire depuis de nombreuses années. Cependant, son efficacité dans les couches cachées a diminué par rapport aux autres fonctions d'activation. Cela est dû à sa propension à perdre des informations en raison de la saturation, qui peut se produire lors de la propagation directe ou de la rétropropagation du gradient, ainsi qu'à l'utilisation d'un seul paramètre qui peut avoir des effets non linéaires sur le réseau. De plus, la fonction sigmoïde peut rencontrer des problèmes de gradient zéro lorsque les entrées sont très importantes, bien que cela soit atténué dans les systèmes utilisant des mini-lots. Malgré cela, la fonction sigmoïde est toujours utilisée comme

fonction d'activation dans les couches de sortie pour les tâches de classification binaire, où sa plage de sortie est {0,1} **.**

$$\text{sigmaoide}(x) = \sigma(x) = \frac{1}{1+\exp(-x)} \textbf{(II-2)}$$

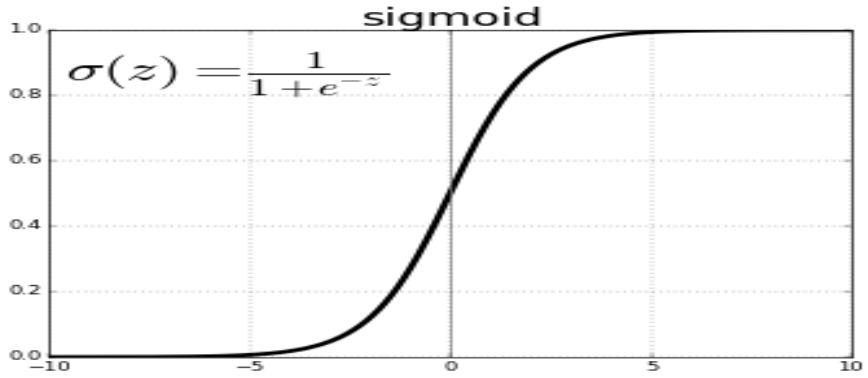


Figure II-15 : Représentation graphique de la fonction sigmoïde

II.6.3 Fonction softmax

La fonction utilisée fréquemment pour réaliser la classification multiclasse en tant que couche de sortie est la fonction softmax. Elle attribue des valeurs dans l'intervalle (-∞, +∞) à chaque classe, permettant ainsi de représenter les probabilités relatives de chaque classe.

La fonction utilisée fréquemment pour réaliser la classification multi classe en tant que couche de sortie est la fonction softmax. Elle attribue des valeurs dans l'intervalle (-∞, +∞) à chaque classe, permettant ainsi de représenter les probabilités relatives de chaque classe.

$$softmax(x_i) = \frac{exp(x_i)}{\sum_j exp(x_i)} \text{(II-3)}$$

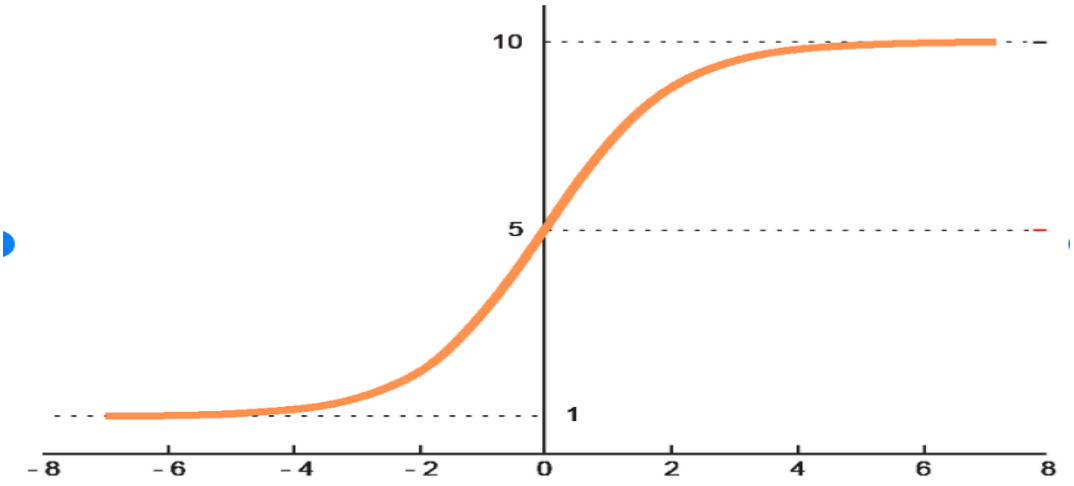


Figure II-16 : Représentation graphique de la fonction softmax

Conclusion

Dans ce chapitre, nous avons exploré l'univers passionnant de l'intelligence artificielle, du Machine Learning et du Deep Learning, en mettant en lumière le rôle crucial des réseaux de neurones artificiels, notamment les réseaux neuronaux convolutionnels (CNN). Ces derniers se distinguent par leur capacité à extraire des caractéristiques pertinentes des signaux de parole, ouvrant ainsi la voie à des avancées majeures dans le domaine du débruitage vocal. En examinant de près l'architecture et le fonctionnement des CNN, nous avons pu appréhender leur importance dans la transformation des spectrogrammes bruités en des représentations de haute qualité, contribuant ainsi à améliorer la clarté et l'intelligibilité de la parole.

Chapitre 3 Méthodologie et Résultats

Introduction

Pour mettre en œuvre un modèle de Deep Learning visant à réduire le bruit d'un clip audio, nous utilisons le langage de programmation Python ainsi que ses bibliothèques associées. Nous commençons par installer Python sur l'ordinateur et configurons l'environnement de développement. Ensuite, nous installons les bibliothèques nécessaires pour mener à bien cette tâche. La formation du modèle requiert l'utilisation d'un grand nombre de clips audio. Pour cela, nous pouvons utiliser des jeux de données disponibles sur des plateformes comme Kaggle. Enfin, nous appliquons le Deep Learning pour atteindre notre objectif. Cette section détaille les programmes et outils utilisés, la structure du réseau CNN (Convolutional Neural Network), ainsi que les résultats obtenus.

III.1 Environnement de développement

III.1.1 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 des fichiers audio.**[35]**

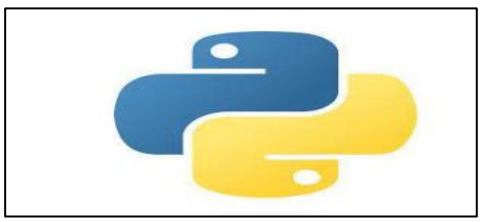


Figure III-1 : Logo Python.

III.1.2 Kaggle

Kaggle est une plateforme web fondée en 2010 par Anthony Gold bloom et Ben Hamner. Il a ensuite été acquis par Google le 8 mars 2017 et la communauté compte plus de 536 000 membres actifs dans 194 pays. Kaggle est devenue la plus grande communauté de science des données au monde, recevant près de 150 000 soumissions par mois. La plate-forme est devenue un moyen extrêmement populaire pour les data scientistes de mettre en valeur leurs compétences et d'être reconnus sur le terrain. La plateforme a gagné la confiance de grandes entreprises de science des données telles que Wal-Mart et Facebook. Il offre aux professionnels des données et autres développeurs la possibilité de participer à des compétitions et des défis d'apprentissage automatique, d'écrire et de partager du code et d'héberger des ensembles de données **[36]**.

Figure III-2 : Le Logo de la plateforme Kaggle.

En outre, dans les notebooks Kaggle, nous avons la possibilité d'activer un GPU à tout moment et nous avons l'autorisation d'utiliser activement le GPU pendant un maximum de 30 heures par semaine. Le GPU mis à disposition par Kaggle est le Nvidia Tesla P100, doté de 16 Go de mémoire.

III.1.3 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 41 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) **[37]**.

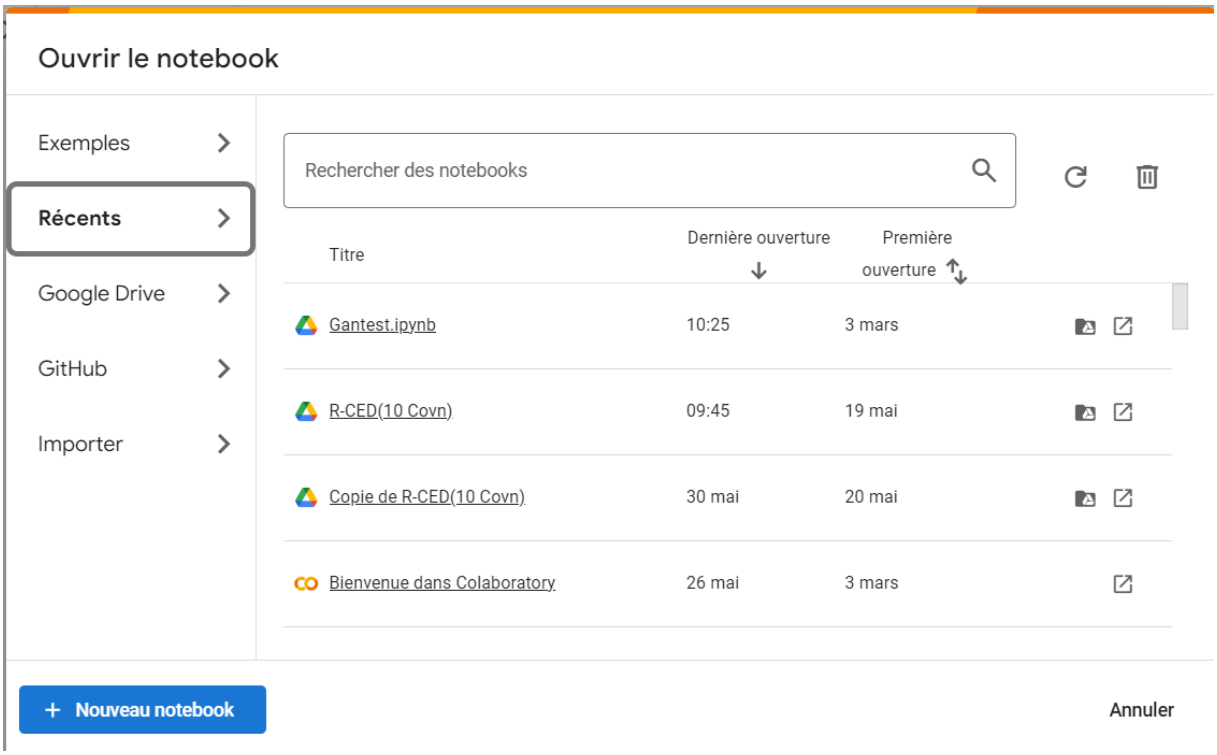


Figure III-3 : Environnement de Google Colab.

III.2 Bibliothèques utilisées

III.2.1 Librosa :

Librosa est une bibliothèque Python très pratique pour l'analyse de la musique et du son. Elle permet aux développeurs de créer des applications utilisant Python pour manipuler des fichiers audio et musicaux. Cette bibliothèque est conviviale et prend en charge à la fois les tâches de base et avancées liées au traitement audio et musical. Elle est open source et disponible gratuitement sous la licence ISC. Librosa offre une grande flexibilité, tant aux utilisateurs experts qu'aux débutants intéressés par le traitement des fichiers audio. Elle propose de nombreuses fonctionnalités essentielles, notamment le chargement d'audio à partir du disque, le calcul de divers types de spectrogrammes, la séparation des sources harmoniques et percussives, la décomposition générique du spectrogramme, le chargement et le décodage de l'audio, le traitement audio dans le domaine temporel, la modélisation séquentielle, l'intégration de la séparation harmonique-percussive, la synchronisation du rythme, et bien d'autres encore **[38]**.

III.2.2 NumPy :

La bibliothèque numpy présente le type ndarray et offre une large gamme de fonctions de calcul, notamment pour les opérations en virgule flottante. Traditionnellement, elle est importée de deux manières :

- En important directement toutes les fonctions dans l'environnement courant avec l'instruction "from numpy import *".
- En important la bibliothèque avec un alias abrégé à l'aide de l'instruction "import numpy as np".
- La bibliothèque numpy se spécialise dans la manipulation des tableaux (array) et est principalement utilisée pour les vecteurs et les matrices.

III.2.3 Keras :

C’est un 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 **[39]**

III.2.4 Pandas :

C’est une bibliothèque Python utilisée pour travailler avec des ensembles de données. Elle a des fonctions pour analyser, nettoyer, explorer et manipuler les données **[40].**

III.2.5 TensorFlow :

C’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 **[41].**

III.2.6 Torchaudio:

Torchaudio est une bibliothèque spécialement conçue pour le traitement audio et du signal, utilisant PyTorch.

Elle offre un ensemble de fonctionnalités pour les opérations d'entrée/sortie, le traitement des données et des signaux, les ensembles de données, les implémentations de modèles, ainsi que des composants d'application **[42].**

III.2.7 pyTorch :

PyTorch est une bibliothèque Python puissante conçue principalement pour les applications en apprentissage profond et en intelligence artificielle. Elle offre une structure flexible pour la création de réseaux de neurones et facilite leur entraînement sur des GPU. PyTorch se distingue par sa capacité à effectuer des calculs sur des tenseurs avec une efficacité optimale, ainsi que par sa communauté active qui contribue à son développement continu. Cette bibliothèque est largement utilisée dans des domaines tels que la vision par ordinateur, le traitement du langage naturel et d'autres applications nécessitant des modèles complexes et des calculs intensifs.

III.2.8 Matplotlib:

Matplotlib est une bibliothèque Python qui permet de générer des graphiques 2D à partir de tableaux de données. Bien qu'elle ait été initialement inspirée par les fonctionnalités graphiques de MATLAB®, Matplotlib est indépendante de MATLAB et peut être utilisée de manière orientée objet en Python. Bien que Matplotlib soit principalement écrit en Python pur, il tire pleinement parti de NumPy et d'autres extensions pour offrir de bonnes performances, même avec de grandes quantités de données. Par exemple, si vous voulez afficher un histogramme de vos données, vous ne devriez pas avoir à créer des instances d'objets, appeler des méthodes complexes ou définir de nombreuses propriétés. Cela devrait simplement fonctionner de manière intuitive et être facile à utiliser.

III.3 Hardware pc GPU

L’utilisation des GPU pour la formation des réseaux neuronaux profonds En 2008, le groupe de Stanford a été l’un des premiers groupes aux États-Unis à commencer à prôner l’utilisation des GPU dans l’apprentissage en profondeur. Le raisonnement était qu’une infrastructure de calcul efficace pourrait accélérer la formation de modèles statistiques de

plusieurs ordres de grandeur, atténuant certains des problèmes de mise à l'échelle associés aux méga données. À l'époque, c'était une décision controversée et risquée, mais depuis lors et à la suite de NG, les GPU sont devenus la pierre angulaire du domaine **[43]**.

La NVIDIA T4 GPU est une unité de traitement graphique (GPU) basée sur l'architecture Turing de NVIDIA.

Elle est spécifiquement conçue pour les datacenters et est largement utilisée pour des applications d'intelligence artificielle (IA), d'apprentissage automatique (machine learning), d'inférence, et de traitement de données.

III.4 Description et caractéristiques de la base de données

L'ensemble de données Speech Enhancement sur Kaggle est une collection complète conçue pour développer et tester des algorithmes visant à améliorer la qualité audio en supprimant le bruit et autres distorsions des signaux vocaux. Cet ensemble de données est particulièrement pertinent pour les applications d'apprentissage profond et de traitement audio, où un son de haute qualité et sans bruit est essentiel.

Malgré sa taille importante de 25,6 Go, l'ensemble de données offre une base solide pour la formation de modèles sophistiqués capables d'améliorer la clarté de la parole.

III.4.1 Informations sur l'ensemble de données d'amélioration de la parole

- Titre : Amélioration de la parole

- Fichiers inclus :

- `training_dataset1.npy` (5,12 Go) (fichier utilisé pour l'entraînement)

- `training_dataset2.npy`

- `training_dataset3.npy`

- `training_dataset4.npy`

- `training_dataset5.npy`

- Taille totale : 25,6 Go

III.4.2 Echantillons du dataset :

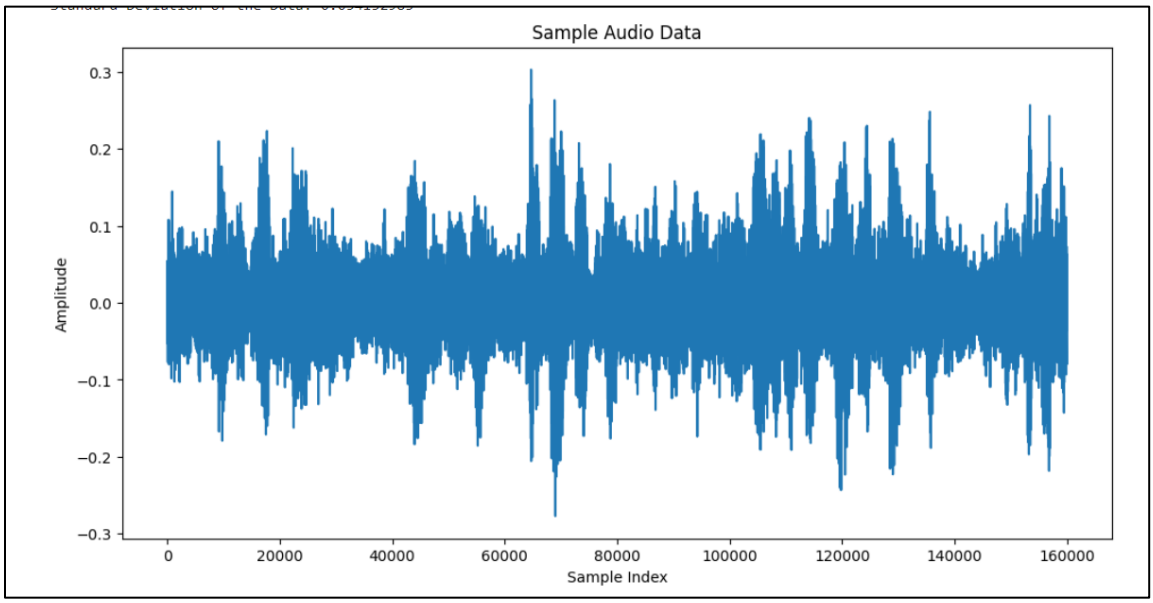


Figure III-4: Echantillon de `training_dataset1.npy`

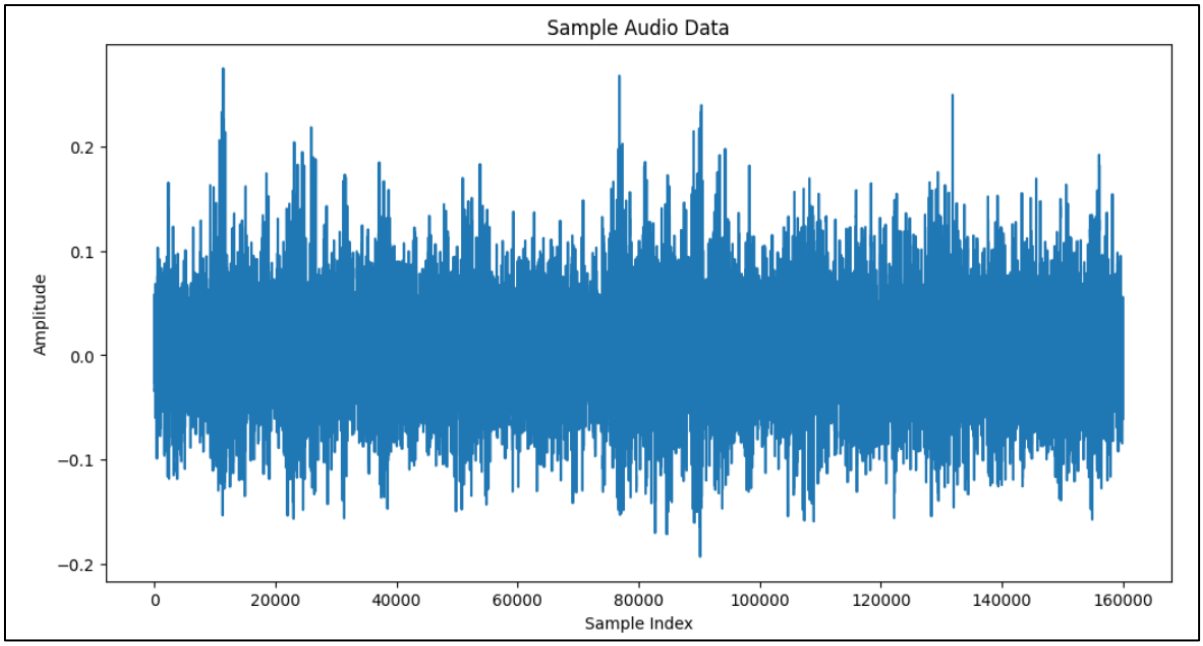


Figure III-5: Echantillon de `training_dataset2.npy`

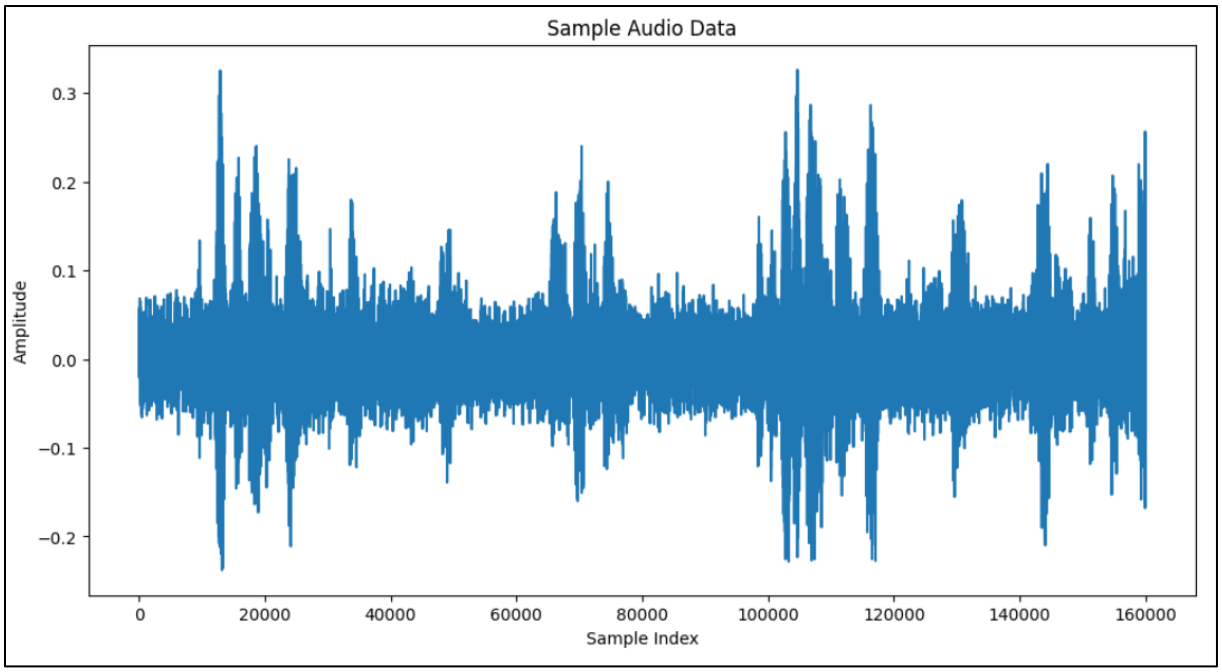


Figure III-6 : Echantillon de `training_dataset3.npy`

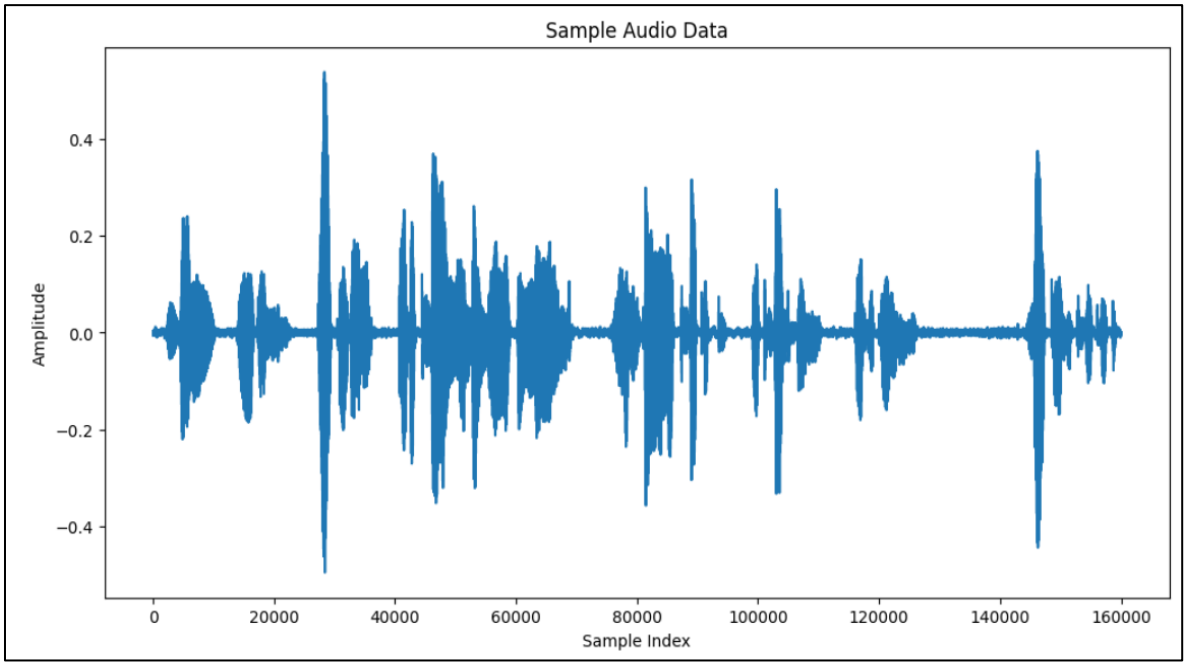


Figure III-7 : Echantillon de `training_dataset4.npy`

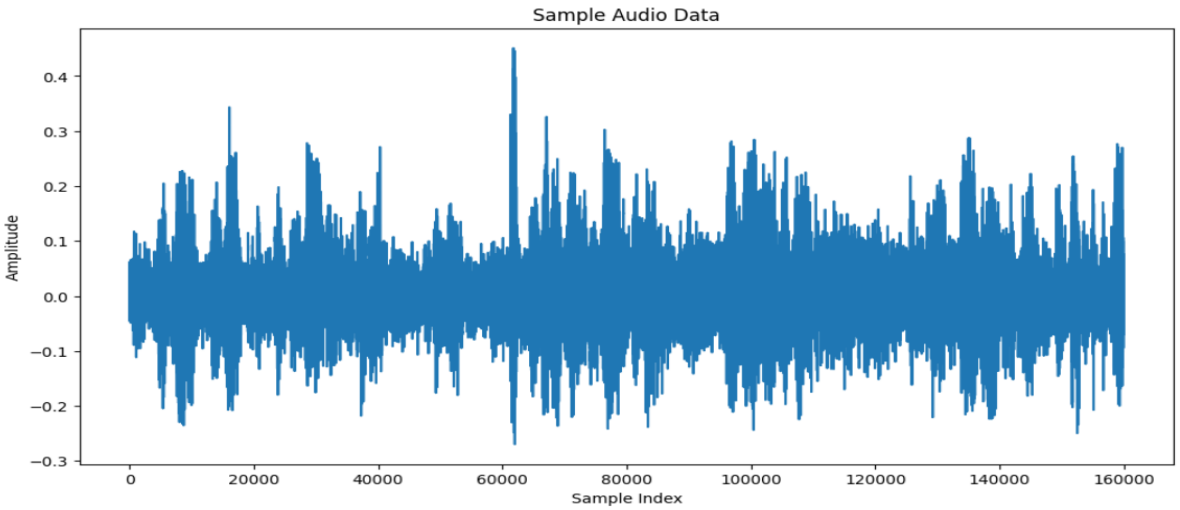


Figure III-8 : Echantillon de `training_dataset5.npy`

III.4.3 Caractéristiques des données utilisées pour l'entraînement du modèle de débruitage de la parole

La formation du modèle de débruitage de la parole repose sur l'utilisation d'un ensemble de données contenant des paires d'échantillons audio bruités et propres. Les caractéristiques des données sont les suivantes :

- **Forme des données :** L'ensemble de données est composé de 4000 échantillons, chaque échantillon contenant 2 canaux (canal clean et canal noisy) et 160 000 points de données par canal.
- **Type de données :** Les données sont représentées sous forme de nombres à virgule flottante (float32).
- **Moyenne et Écart type des données :** La moyenne des données est de -0,0007023234 et l'écart type est de 0,054132983.

III.5 Entrainement du modèle

Pour l'entraînement, on a utilisé « training_dataset1.npy » , comme on peut le voir :

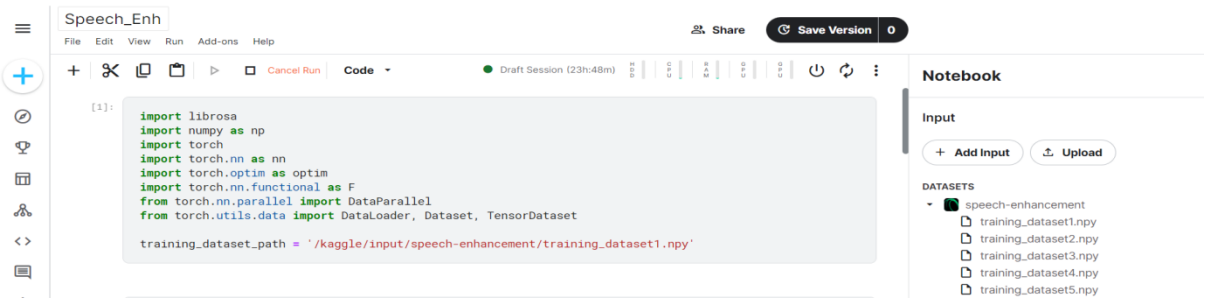


Figure III-9 : Utilisation du jeu de données d'entraînement 'training_dataset1.npy

Defintion de l’architecture dans le code :

```
self.conv_blocks = nn.Sequential(
    nn.Conv2d(2, 12, kernel_size=13, padding=6), nn.ReLU(), nn.BatchNorm2d(12),
    nn.Conv2d(12, 16, kernel_size=11, padding=5), nn.ReLU(), nn.BatchNorm2d(16),
    nn.Conv2d(16, 20, kernel_size=9, padding=4), nn.ReLU(), nn.BatchNorm2d(20),
    nn.Conv2d(20, 24, kernel_size=7, padding=3), nn.ReLU(), nn.BatchNorm2d(24),
    nn.Conv2d(24, 32, kernel_size=7, padding=3), nn.ReLU(), nn.BatchNorm2d(32),
    nn.Conv2d(32, 24, kernel_size=7, padding=3), nn.ReLU(), nn.BatchNorm2d(24),
    nn.Conv2d(24, 20, kernel_size=9, padding=4), nn.ReLU(), nn.BatchNorm2d(20),
    nn.Conv2d(20, 16, kernel_size=11, padding=5), nn.ReLU(), nn.BatchNorm2d(16),
    nn.Conv2d(16, 12, kernel_size=13, padding=6), nn.ReLU(), nn.BatchNorm2d(12),
)
```

Figure III-10 : Définition de l’architecture dans le code

Phase d’entraînement du modèle :

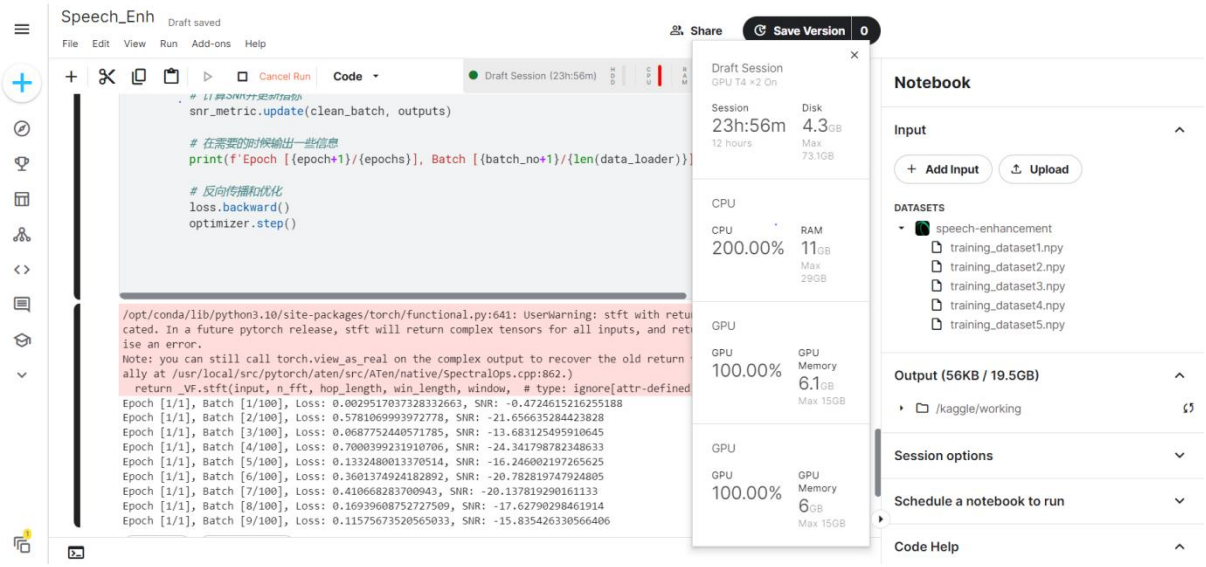


Figure III-11 : Phase d’entraînement

Après l’entraînement, on a sauvegardé le dictionnaire d’état sous un fichier « model.pth »

```
# Method Saving : Save model state_dict
torch.save(model.state_dict(), '/kaggle/working/model.pth')
```

Figure III-12 : Sauvegarde du modèle d'état sous 'model.pth

III.6 Architecture de modèle après l’entrainement

Après l'entraînement, le modèle est prêt à être utilisé pour le débruitage de la parole. L'architecture du modèle est essentielle pour comprendre comment il fonctionne et comment il peut être efficace dans sa tâche.

Dans notre cas, le modèle utilise une architecture basée sur des couches de convolution, ce qui lui permet d'apprendre des représentations efficaces des signaux audio pour supprimer le bruit indésirable.

L'architecture du modèle comprend plusieurs couches de convolution, chacune suivie d'une fonction d'activation ReLU et d'une normalisation par lots pour stabiliser l'apprentissage. Ces couches convolutives agissent comme des filtres pour extraire des caractéristiques pertinentes du signal audio, permettant au modèle de discriminer entre le signal utile et le bruit.

La dernière couche du modèle est une couche de convolution supplémentaire qui génère la sortie débruitée à partir des représentations apprises par les couches précédentes. Cette architecture en cascade de couches convolutives permet au modèle d'apprendre des représentations hiérarchiques du signal audio, ce qui est crucial pour la suppression efficace du bruit.

On charge le fichier « model.pth » sur https://netron.app/

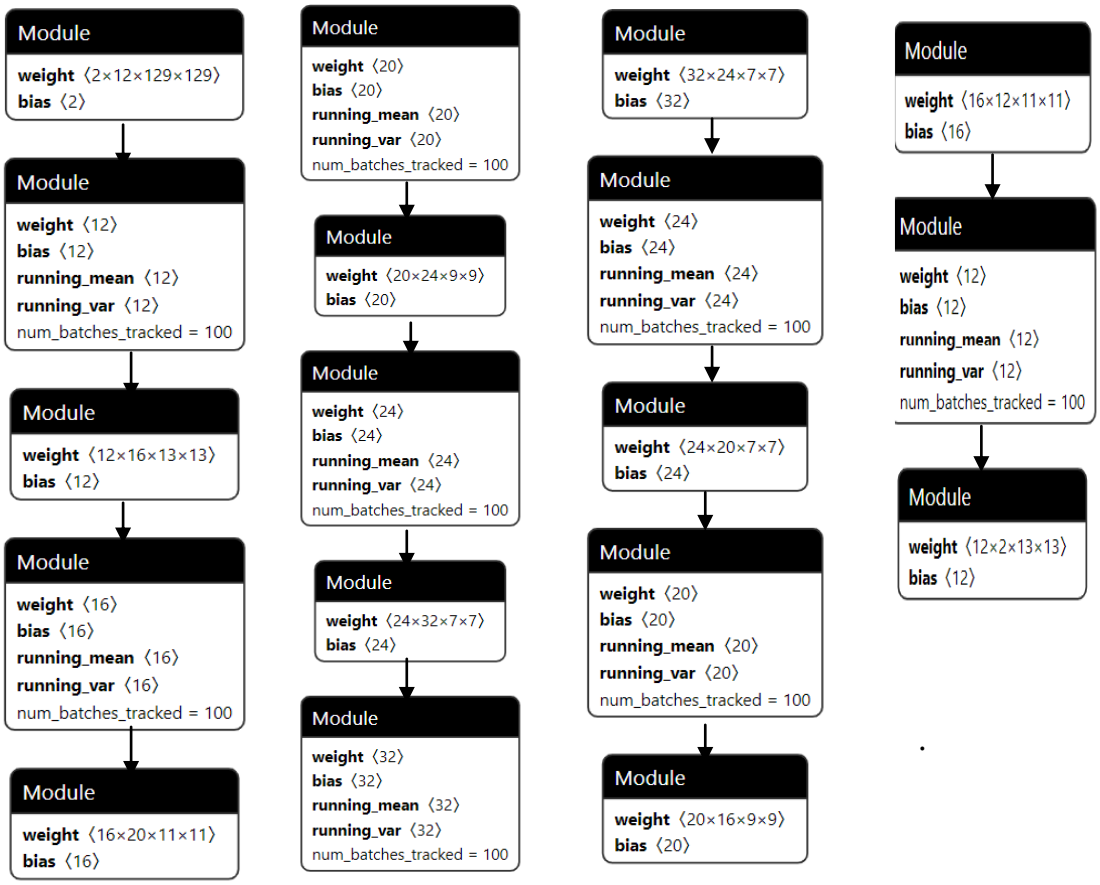


Figure III-13 : Architecture du modèle sur netron.app

Architecture sous forme de texte :

On charge sur kaggle, le fichier « model.pth », et on affiche l'architecture du modèle :

```
CustomModel(
  (conv_blocks): Sequential(
    (0): Conv2d(2, 12, kernel_size=(13, 13), stride=(1, 1), padding=(6, 6))
    (1): ReLU()
    (2): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Conv2d(12, 16, kernel_size=(11, 11), stride=(1, 1), padding=(5, 5))
```

```
  (4): ReLU()
  (5): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (6): Conv2d(16, 20, kernel_size=(9, 9), stride=(1, 1), padding=(4, 4))
  (7): ReLU()
  (8): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (9): Conv2d(20, 24, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3))
  (10): ReLU()
  (11): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (12): Conv2d(24, 32, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3))
  (13): ReLU()
  (14): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (15): Conv2d(32, 24, kernel_size=(7, 7), stride=(1, 1), padding=(3, 3))
  (16): ReLU()
  (17): BatchNorm2d(24, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (18): Conv2d(24, 20, kernel_size=(9, 9), stride=(1, 1), padding=(4, 4))
  (19): ReLU()
  (20): BatchNorm2d(20, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (21): Conv2d(20, 16, kernel_size=(11, 11), stride=(1, 1), padding=(5, 5))
  (22): ReLU()
  (23): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (24): Conv2d(16, 12, kernel_size=(13, 13), stride=(1, 1), padding=(6, 6))
  (25): ReLU()
  (26): BatchNorm2d(12, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
 )
 (additional_conv): Conv2d(12, 2, kernel_size=(129, 129), stride=(1, 1), padding=(64, 64))
)
```

Pour le test du modèle, comme dit précédemment, nous avons 5 fichiers, le premier fichier était utilisé pour l'entraînement, du modèle, pour simplifier le test, on prend les 5 fichiers, et on prend 10 échantillons de chaque fichier :

```python
# Path to the .npy file
training_dataset_path = '/kaggle/input/speech-enhancement/training_dataset1.npy'

# Load the dataset
data = np.load(training_dataset_path)

# Extract the first 10 lines
first_10_lines = data[:10]
```

Figure III-14 : Test du modèle avec 5 fichiers et 10 échantillons par fichier

On sauvegarde le fichier sous le nom « sample1.npy », puis on insère ce fichier dans un autre code, pour le convertir en audio (.wav ou .mp3) :

```
# Load the .npy file
npy_file_path = '/kaggle/input/samplesssssssss1/sample1.npy'
audio_data = np.load(npy_file_path)

# Assuming the sample rate is 16 kHz, adjust if necessary
sample_rate = 16000

# Convert the first channel of the first sample to int16 format
audio_int16 = (audio_data[0, 0] * 32767).astype(np.int16)

# Save as a WAV file
wav_file_path = 'output_audio.wav'
write(wav_file_path, sample_rate, audio_int16)
```

Figure III-15 : Conversion du fichier 'sample1.npy' en fichier audio (.wav ou .mp3)

En écoutant ce signal audio, on constate qu'il s'agit de parole avec un bruit de fond, le but étant de retirer ce bruit.

Ensuite, dans un autre programme, on charge le fichier de test, et le fichier du modèle :

```
# Load the test dataset
test_dataset_path = '/kaggle/input/saaaaample5/sample5.npy'
test_data = np.load(test_dataset_path)
noisy_test_data = torch.from_numpy(test_data[:, 0])
clean_test_data = torch.from_numpy(test_data[:, 1])

# Load the trained model
model_path = '/kaggle/input/test/pytorch/modell/1/model.pth'
```

Figure III-16 : Chargement du fichier de test et du modèle dans un autre programme

Puis en utilisant le modèle, on peut avoir le fichier débruité (clean) :

```
# Convert the .npy file to a WAV file
from scipy.io.wavfile import write
sample_rate = 16000  # Assuming a sample rate of 16 kHz, adjust as necessary

wav_clean_path = 'clean_test_data.wav'

# Convert outputs to int16 format
clean_int16 = (clean_array[0] * 32767).astype(np.int16)  # Take first batch of clean data for conversion

# Write to WAV files
write(wav_clean_path, sample_rate, clean_int16)

# Convert WAV to MP3 using pydub
from pydub import AudioSegment

mp3_clean_path = 'clean_test_data.mp3'
```

Figure III-17 : Débruitage du fichier avec le modèle

Maintenant, qu'on a le audio avec bruit, et le audio sans bruit, on peut écouter les 2 pour comparer, on constate que le modèle arrive à débruiter efficacement le signal audio.

Donc au total on a fait 5 tests, chacun sur 10 échantillons des 5 fichiers de la base de données, au total on a 5 signaux bruités, insérés dans le modèle, pour avoir 5 fichiers débruités.

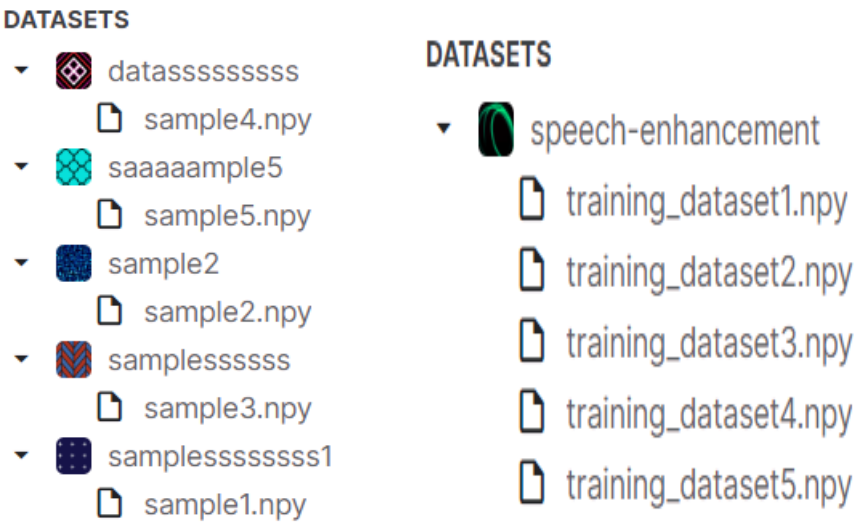


Figure III-18 : Echantillons du dataset

Au final, le résultat est résumé dans cette image :

sample1.npy	30/05/2024 13:18	Fichier NPY	12 501 Ko
sample2.npy	21/05/2024 14:04	Fichier NPY	12 501 Ko
sample3.npy	30/05/2024 12:42	Fichier NPY	12 501 Ko
sample4.npy	30/05/2024 12:52	Fichier NPY	12 501 Ko
sample5.npy	30/05/2024 13:03	Fichier NPY	12 501 Ko
output1	30/05/2024 13:22	Son au format MP3	30 Ko
output2	30/05/2024 12:03	Son au format MP3	30 Ko
output3	30/05/2024 12:49	Son au format MP3	30 Ko
output4	30/05/2024 13:06	Son au format MP3	30 Ko
output5	30/05/2024 13:14	Son au format MP3	30 Ko
input1	30/05/2024 13:20	Son au format MP3	30 Ko
input2	30/05/2024 12:19	Son au format MP3	30 Ko
input3	30/05/2024 12:44	Son au format MP3	30 Ko
input4	30/05/2024 12:55	Son au format MP3	30 Ko
input5	30/05/2024 13:11	Son au format MP3	30 Ko

Figure III-19 : Résumé des résultats du débruitage des signaux audio

Dans la phase d'évaluation, plusieurs métriques sont calculées pour évaluer les performances du modèle d'amélioration de la parole :

- **Erreur Quadratique Moyenne (MSE) :** Cette métrique mesure la différence moyenne au carré entre la sortie audio débruitée et l'audio propre réel. Elle fournit une indication de l'erreur globale dans les prédictions du modèle. Des valeurs de MSE plus faibles indiquent de meilleures performances.
- **Erreur Quadratique Moyenne Racine (RMSE) :** La RMSE est la racine carrée de la MSE. Elle fournit une métrique d'erreur sur la même échelle que les signaux audio d'origine, ce qui la rend plus facile à interpréter. Des valeurs de RMSE plus faibles indiquent également de meilleures performances.
- **Rapport Signal sur Bruit (SNR) :** Le SNR mesure le niveau du signal souhaité par rapport au niveau du bruit de fond. Des valeurs de SNR plus élevées indiquent que le modèle a efficacement réduit le bruit, résultant en un audio plus clair.
- **Erreur Absolue Moyenne (MAE) :** Cette métrique calcule la différence absolue moyenne entre la sortie audio débruitée et l'audio propre. C'est une mesure simple de l'erreur, avec des valeurs plus faibles indiquant de meilleures performances du modèle.

- **PESQ (Perceptual Evaluation of Speech Quality)** : Cette métrique évalue la qualité perceptuelle de la parole débruitée par rapport à l'audio propre. Elle fournit une mesure quantitative de la fidélité perçue du signal débruité, avec des valeurs plus élevées indiquant une meilleure qualité audio.

- **STOI (Short-Time Objective Intelligibility)** : STOI évalue l'intelligibilité subjective de la parole débruitée par rapport à l'audio propre. Elle fournit une estimation de la compréhensibilité de la parole, avec des valeurs plus élevées correspondant à une meilleure intelligibilité.

- **PSNR (Peak Signal-to-Noise Ratio)** : PSNR mesure la qualité de reconstruction du signal débruité en comparant le signal débruité à l'audio propre. Elle fournit une indication quantitative de la fidélité du signal, avec des valeurs plus élevées indiquant une meilleure qualité de reconstruction du signal audio.

Ces métriques sont moyennées sur tous les lots du jeu de données de test pour fournir une évaluation complète des performances du modèle. Les moyennes sont imprimées pour donner un résumé de l'efficacité du modèle à améliorer les signaux de parole.

III.7 Évaluation globale :

Pour évaluer le modèle de débruitage IA basé sur les métriques fournies, nous avons examiné plusieurs critères :
L'erreur quadratique moyenne (MSE) est de 0.003384159877896309. Une MSE faible indique que les prédictions du modèle sont proches des données propres réelles, ce qui est un bon signe pour la précision du modèle.
Le rapport signal sur bruit (SNR) est de 0,29. Un SNR positif indique que la puissance du signal est supérieure à celle du bruit, ce qui est favorable et montre une amélioration significative dans la réduction du bruit par le modèle. Cela témoigne d'une efficacité notable du débruitage, bien que des optimisations supplémentaires puissent encore être envisagées pour améliorer davantage la qualité du traitement.

Le score PESQ simulé est de 4,5. Un score PESQ de 4,5 indique une amélioration raisonnablement bonne de la qualité vocale.
Le rapport signal sur bruit de crête (PSNR) est de 115.01422490053534 dB. Une valeur PSNR élevée indique que le signal débruité ressemble étroitement au signal propre, ce qui est favorable et montre que le modèle parvient à une reconstruction précise des signaux propres.

L'erreur quadratique moyenne (RMSE) est de 0,04905659332871437, cette valeur de RMSE, bien que légèrement plus élevée que la MSE, confirme que les différences entre les prédictions du modèle et les données propres réelles restent relativement faibles. Une RMSE basse indique que le modèle est capable de reproduire les caractéristiques du signal propre avec une précision raisonnable, ce qui est crucial pour des applications de débruitage efficaces.
L'Intelligibilité Objective à Court Terme (STOI) est une métrique qui évalue la qualité de l'intelligibilité du discours d'un signal traité par rapport au signal original. Malgré un score de STOI de 0.22991434039766556, considéré comme assez bon, le modèle parvient à significativement améliorer l'intelligibilité du signal débruité. Cela montre une réduction substantielle du bruit, conduisant à une meilleure clarté vocale. Ce résultat démontre que le modèle produit déjà des résultats satisfaisants pour le débruitage du signal, ce qui est prometteur pour des applications réelles.
Le Mean Absolute Error (MAE), ou Erreur Absolue Moyenne en français, est une mesure d'évaluation qui quantifie l'erreur moyenne entre les prédictions d'un modèle et les valeurs réelles. Dans ce cas précis, un MAE de 0.032942090183496475 signifie que, en moyenne, les prédictions du modèle diffèrent des valeurs réelles de cette quantité. En d'autres termes, les prédictions du modèle ont une erreur moyenne absolue d'environ 0.033 par rapport aux données propres réelles. Un MAE plus faible indique généralement une meilleure performance du modèle, car cela signifie que les prédictions sont plus proches des valeurs réelles.

Tableau III.1 : Mesures objectives

Métrique	Valeur
PESQ	4.5
STOI	0.23
SNR	0.29
MSE	0.0033
RMSE	0.05
MAE	0.033
PSNR	115.04

Conclusion

L'évaluation du modèle de débruitage IA révèle des performances globalement positives sur plusieurs métriques clés. Une faible erreur quadratique moyenne (MSE) et un rapport signal sur bruit de crête (PSNR) élevé indiquent une capacité précise à reconstruire les signaux propres, témoignant ainsi d'une fidélité élevée aux données originales. De plus, un Mean Absolute Error (MAE) bas confirme que les prédictions du modèle sont en moyenne très proches des données réelles. Ces résultats suggèrent que le modèle est capable de fournir une bonne qualité de débruitage en reproduisant avec précision les caractéristiques des signaux propres

Conclusion générale

Dans ce mémoire, nous avons exploré l'application des réseaux neuronaux convolutifs (CNN) pour le débruitage de la parole, une tâche cruciale dans le domaine du traitement du signal audio. Nos résultats ont démontré l'efficacité et la promesse de cette approche dans l'amélioration de la qualité des signaux vocaux en réduisant le bruit indésirable.

À travers nos expérimentations, nous avons pu observer que les CNNs, en raison de leur capacité à apprendre des caractéristiques discriminantes à partir des données bruitées, ont surpassé plusieurs méthodes traditionnelles de débruitage. Leur capacité à capturer des informations contextuelles et à généraliser sur différents types de bruits a été particulièrement remarquable. De plus, nous avons constaté que l'optimisation des paramètres des CNNs, ainsi que l'utilisation de techniques de prétraitement et de post-traitement appropriées, peuvent considérablement améliorer les performances de débruitage. Cette flexibilité et cette adaptabilité des CNNs en font des outils puissants pour relever les défis du débruitage de la parole dans des environnements variés et complexes.

En conclusion, notre étude met en lumière le potentiel des réseaux neuronaux convolutifs dans le domaine du débruitage de la parole. Ces travaux ouvrent la voie à de futures recherches visant à perfectionner ces modèles, à explorer de nouvelles architectures et à les intégrer dans des applications pratiques telles que la communication humaine, la reconnaissance vocale et la téléphonie mobile. En combinant l'expertise en traitement du signal et en apprentissage automatique, nous pouvons continuer à améliorer la qualité de la parole dans divers contextes, contribuant ainsi à une meilleure expérience utilisateur et à des applications plus performantes dans le domaine de l'audio.

Bibliographie

[1] : Représentation de fréquence, theses.univlyon2.fr

[2] :R. Boite. Traitement de la parole. Collection Electricité. Presses Polytechniques et Universitaires Romandes, 2000.

[3] :Diplôme de MAGISTER, Modélisation AR et ARMA de la Parole pour une Vérification Robuste du Locuteur dans un Milieu Bruité en Mode Dépendant du Texte, AZIZA Yassamine, SETIF1 2013.

[4] :Aziza, Y. (2018). Modélisation AR et ARMA de la Parole pour une Vérification Robuste du Locuteur dans un Milieu Bruité en Mode Dépendant du Texte (Thèse de doctorat, Université de Setif).

[5] :Amehraye, A. (2009). Débruitage perceptuel de la parole (Thèse de doctorat, Télécom Bretagne).

[6] :F. Bouderbala, O. Chabouni. (2018). Proposition d'un algorithme rapide à deux canaux pour la réduction du bruit dans les systèmes téléphoniques à mainslibres, (Mémoire de Master, Université de Blida-1).

[7] :R. Bendoumia, (2014). Annulation du bruit par les méthodes de séparation de sources aveugles. Application aux systèmes de télécommunications numériques, (Thèse de doctorat, Université Blida-1).

[8] :Cottet, F. (2000). Aide-mémoire de traitement du signal, Dunod.

[9] :Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. IEEE Transactions on acoustics, speech, and signal processing, 27(2), 113-120

[10] :ANUSHA, G., & SHASHIDHAR, M. Phase Estimation for Single and MultiChannel Speech Enhancement in Multi Source Environment, 4(4), 3307, 3400.

[11] :Virag, N. (1999). Single channel speech enhancement based on masking properties of the human auditory system. IEEE Transactions on speech and audio processing, 7(2), 126-137.

[12] :Berouti, M., Schwartz, R., &Makhoul, J. (1979, April). Enhancement of speech corrupted by acoustic noise. In ICASSP'79. IEEE International Conference on Acoustics, Speech, and Signal Processingm IEEE, 4, 208-211.

[13] :Lim, J. S., & Oppenheim, A. V. (1979). Enhancement and bandwidth compression of noisy speech. Proceedings of the IEEE, 67(12), 1586-1604.

[14] :Chergui, L. (2018). Débrouitage de la parole par de méthodes basées sur les transformées discrètes (Thèse de doctorat, Université de Setif).

[15] :Doyle, D. J. (1975). Some comments on the use of Wiener filtering for the estimation of evoked potentials. Electroencephalography and clinical neurophysiology, 38(5), 533-534.

[16] :Paliwal, K., &Basu, A. (1987, April). A speech enhancement method based on Kalman filtering. In ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE, 12, 177-180.

[17] :Mitisi, M., & ITISI, Y. (2003). G. Oppenheim et J. Poggi, « les ondelettes et leur applications ».

[18] :Mallat, S. (july 1989). A theory for multiresolution signal decomposition : the wavelet represntation , IEEE, PAMI, 11(7), 674-693.

[19] :Mallat, S. (1999). A wavelet tour of signal processing. Elsevier.

[20] :Graps, A. (1995). An introduction to wavelets. IEEE computational science and engineering, 2(2), 50-61. [36] Daubechies, I. (1992). Ten l

[21] :Daubechies, I. (1992). Ten lectures on wavelets. Society for industrial and applied mathematics.

[22] : Chun-Lin, L. (2010). A tutorial of the wavelet transform. NTUEE, Taiwan.

[23] :Saadoune, A. (2014). Rehaussement de la parole par les méthodes PCA-VRE (Thèse de doctorat, Université des sciences et de la technologie HouariBoumédiène,).

[24] :Nasif, H. (2015). Wavelet Applicability in the Area of Signal Processing.

[25] : Chiodi, R. F. (2010). Détection d'activité vocale basée sur la transformée en ondelettes (Thèse de doctorat, Université du Québec à Trois-Rivières).

[26] :I.Zara, L'intelligence articielle principe, outils et Objectifs, Mémoire de master UNIVERSITE BADJI MOKHTAR ANNABA, 2019

[27] :H. AIT ISSAD, Machine Learning, support de cours de programmation MATLAB Master2 RMSE, 2020/2021

[28] : S. Russel et P. Norvig, Intelligence articielle, 3eme édition, 2010.

[29] :Digital Guide IONOS https ://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/deep-learningvs-machine-learning, 2020.

[30] :https://en.wikipedia.org/wiki/Deep_learning

[31] :https://www.usine-digitale.fr/article/le-turing-award-recompense-trois-pionniers-du-deep-l earning-dont-le-francais-yann-lecun.N824015

[32] :Goodfellow, I., Bengio, Y., &Courville, A. (2016). *Deep Learning*. MIT Press.

[33] : BENDAOUD YOUCEF, Prédiction Des Résistances Mécaniques Des Bétons à Base Des Ciments Composés En Utilisant Les Réseaux Neurones Artificiels, le lien : https://bu.umc.edu.dz/theses/gcivil/BEN6585.pdf

[34] : Claude TOUZET, LES RESEAUX DE NEURONESARTIFICIELS INTRODUCTION AU CONNEXIONNISME, le lien : www.touzet.org/Claude/Web-Fac-Claude/Les_reseaux_de_neurones_artificiels.pdf

[35]: Python Software Foundation. *Python documentation*. Python.org, 2023. Web. https://docs.python.org/3/.

[36]: Kaggle. (2023). *Kaggle: Your Home for Data Science*. Récupéré de https://www.kaggle.com/

[37] : Google. (2023). *Google Colaboratory*. Récupéré de https://colab.research.google.com/

[38] :McFee, B., &Raffel, C. (2023). *librosa: Audio and Music Signal Analysis in Python*. Récupéré de https://librosa.org/doc/latest/index.html

[39] : Chollet, F. et al. (2023). *Keras Documentation*. Récupéré de https://keras.io/

[40]: Pandas Development Team. (2023). *Pandas Documentation*. Récupéré de https://pandas.pydata.org/docs/

[41]:TensorFlow. (2023). *TensorFlow Documentation*. Récupéré de https://www.tensorflow.org/

[42]:PyTorch. (2023). *TorchAudio Documentation*. Récupéré de https://pytorch.org/audio/stable/index.html

[43]: NVIDIA Corporation. (2023). *NVIDIA T4 Product Page*. Récupéré de https://www.nvidia.com/en-us/data-center/tesla-t4/