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

Elzayyat Marwa

&

Bouguera Feriel

Rehaussement de la parole basée sur l'apprentissage profond (Deep Learning)

Proposé par : Ykhlef Farid

Année Universitaire 2022-2023

Remerciements

En premier lieu, nous remercions Dieu tout puissant pour nous avoir accordé la volonté, la santé et le courage nécessaires pour mener à bien nos études. Sa guidance et sa protection ont été des sources d'inspiration et de force tout au long de notre parcours académique.

Nous tenons également à exprimer notre reconnaissance envers notre encadreur, M. F.YKHLEF, dont l'aide précieuse, la patience et les orientations ont été déterminantes pour la réussite de notre travail. Sa disponibilité, son expertise et son intérêt constant pour notre projet nous ont grandement aidés et nous lui en sommes profondément reconnaissants.

Nous adressons également nos remerciements aux membres du jury qui ont accepté d'évaluer notre travail. Leur expertise et leurs commentaires constructifs nous ont permis d'améliorer la qualité de notre recherche.

Nous remerciements vont également à toutes les personnes qui ont apporté leur contribution, qu'elle soit directe ou indirecte, à l'élaboration de ce travail. Votre soutien, vos conseils et votre collaboration ont été d'une valeur inestimable, et nous vous sommes reconnaissants pour votre précieuse contribution.

Enfin, nous souhaitons exprimer notre profonde gratitude envers nos enseignants qui ont enrichi nos connaissances et nous ont guidés tout au long de notre formation professionnelle. Leur dévouement, leur expertise et leur passion pour l'enseignement ont joué un rôle essentiel dans notre parcours académique, et nous leur sommes infiniment reconnaissants.

Nous tenons à remercier du fond du cœur chacune de ces personnes pour leur soutien inestimable et leur contribution à la réalisation de notre travail de fin d'étude. Votre engagement et votre accompagnement ont été des éléments clés de notre réussite, et nous vous en sommes profondément reconnaissants.

Dédicaces

Je dédie mon succès à ceux qui m'ont appris la générosité, le succès et la patience. Tout au long de sa vie, il ne m'a jamais empêché de le soutenir et je suis fier de porter son nom. Je demande à Dieu de prolonger sa vie afin qu'il puisse voir les fruits tant attendus.

Mon père

Pour ma présence angélique dans ma vie, qui représente l'amour, la tendresse et le dévouement, pour mon sourire dans la vie et le sens de mon existence, pour

Ma mère

Que Dieu la protège, dont la supplication a été le secret de ma réussite. Je voudrais également exprimer ma gratitude à mes frères : Mohamed et Abdelhak et à ma chère sœur : Malak qui ont été à mes côtés et m'ont soutenu de toutes leurs forces.

Je n'oublie pas ma binôme, Elzayyat Marwa. Nous avons parcouru ensemble un chemin long, difficile. Je vous félicite sincèrement

Feriel

Dédicaces

À celle qui incarne l'amour et la tendresse infinis,

À celle qui a été à mes côtés dans les moments les plus difficiles,

Ma chère mère, que Dieu la protège.

À celui qui n'a jamais lésiné sur son soutien tout au long de sa vie,

Mon cher père

À mes chères sœurs : Haizia, Rania, Kholoud

À mes neveux : Ahmed, Adam,

Et à ma binome, Bouguera Feriel,

Qui nous avons partagé ensemble les difficultés de ce projet, nous avons surmonté les obstacles et savouré la joie de la réussite.

MARWA

ملخص:

في هذا المشروع ، اقترحنا استخدام نموذج مدرب مسبقًا يعتمد على التعلم العميق لتقليل الضوضاء في الإشارة الصوتية. لهذا ، استخدمنا بيئة معينة ، تسمى "Colaboratory" ، وغالبًا ما يتم اختصارها إلى "Colab" المناسبة للتعلم الآلي وتحليل البيانات. سمح لنا النموذج المستخدم بالحصول على إشارات كلام خالية من الضوضاء. في هذه الدراسة ، استكشفنا أيضًا المفاهيم النظرية بدءًا من الأهمية الكبرى للتعلم العميق ، متبوعًا بشرح أساسيات الذكاء الاصطناعي والتعلم الآلي والتعلم العميق ، بالإضافة إلى تطورها على مر عقود من الزمن.

كلمات المفاتيح: تحسين الخطاب ؛ تعلم عميق ؛ شبكة RNN

Résumé :

Dans ce projet, nous avons proposé l'utilisation d'un modèle pré-entraîné basé sur le Deep Learning pour réduire le bruit dans un signal audio. Pour cela, nous avons utilisé un environnement particulièrement, nommé "Colaboratory", souvent raccourci en "Colab" adapté au Machine Learning et à l'analyse de données. Le modèle utilisé nous a permet d'obtenir des signaux de parole dépourvus de bruit. Dans cette étude, nous avons aussi exploré les concepts théoriques en commençant par l'importance majeure du Deep Learning, suivi de l'explication des bases de l'intelligence artificielle, du Machine Learning et du Deep Learning, ainsi que de leur évolution au fil des décennies.

Mots clés : Rehaussement de la parole ; Deep Learning ; RNN network.

Abstract :

In this project, we proposed the use of a pre-trained model based on Deep Learning to reduce noise in an audio signal. For this, we used a particular environment, named "Colaboratory", often shortened to "Colab". This latter is suitable for Machine Learning and data analysis. The model used allowed us to obtain noise-free speech signals. In this study, we also explored the theoretical concepts starting with the major importance of Deep Learning, followed by the explanation of the basics of Artificial Intelligence, Machine Learning and Deep Learning, as well as their evolution over time decades.

Keywords : Speech enhancement ; Deep Learning ; RNN network.

Listes des acronymes et abréviations

AI : Artificial Intelligence

ANN : Artificial Neural Network

CNN : Convolution Neural Network.

CUDA : Compute Unified Device Architecture.

DCT : Discrete Cosine Transform.

DFT : Discrete Fourier Transform.

DNN : Deep Neural Network.

FC : Fully-Connected neural networks

FT : Fourier Transform.

GPU : Graphics Porcessing Unit.

LSTM : Long Short Term Memory.

LTU : Linear Threshold Units

MMSE : Minimum Mean Squared Error.

MMSE-LSA : mmse logarithmic spectral amplitude estimator.

RBM : Restricted Boltzmann Machines.

RNN : Recurrent Neural Network.

SGD : Similar to stochastic Gradient Decent learning methodology

SNR : Signal to noise ratio.

STFT : Short-Time Fourier Transform.

TPU :Tensor Processing Units.

VAD : Voice activity dectetor.

Table des matières

Liste des figures

Introduction générale

La parole joue le rôle le plus important dans la communication humaine en tant que support d'information le plus courant et le plus pratique. Dans les situations d'écoute quotidiennes, les signaux de parole sont souvent corrompus par les bruits ambiants lors de leur acquisition, entraînant une dégradation de la qualité et de l'intelligibilité de la parole pour un auditeur.

Le rehaussement de la parole vise à améliorer la qualité de la parole de la parole dégradée. Avec la croissance exponentielle de l'intelligence artificielle dans divers domaines et son rôle essentiel dans le développement de nombreux secteurs, l'utilisation du Machine Learning et du Deep Learning est devenue cruciale.

Récemment, les chercheurs ont eu recours à l'apprentissage en profondeur comme principal outil d'amélioration de la parole, qui comporte souvent des modèles déterministes adoptant une formation supervisée.

Des systèmes intelligents récents de traitement audio intègrent souvent des mécanismes pour réduire de manière plus précise le bruit dans les scènes audio. Ce mécanisme est considéré comme un sous-domaine de la vision par ordinateur, qui repose sur des techniques telles que le Machine Learning et le Deep Learning.

Au cours des dernières décennies, le domaine de l'apprentissage a été dominé par un sous-type de réseau neuronal appelé réseau neuronal récurrent (RNN), adapté aux tâches liées aux signaux sonores.

Le réseau neuronal est entraîné sur diverses fonctionnalités, telles que la réduction du bruit dans les signaux audio, l'amélioration de la qualité, ainsi que l'augmentation de la puissance et de la clarté du son.

Dans cette étude, nous explorerons les concepts théoriques commençant par l'importance majeure du Deep Learning, suivi de l'explication des bases de l'intelligence artificielle, du Machine Learning et du Deep Learning, ainsi que de leur évolution au fil des décennies. Nous aborderons également les réseaux de neurones, pour enfin nous concentrer sur les réseaux de neurones récurrents.

Une fois que nous aurons présenté les concepts généraux et les problèmes résolus, nous examinerons le défi de la réduction du bruit dans les sons. Nous décrirons comment créer un modèle en utilisant différentes structures en appliquant deux modèles distincts, à savoir "torch" et "torchaudio". Nous évaluerons les résultats de nos tests en utilisant plusieurs bibliothèques afin de faciliter la mise en œuvre et d'accélérer le processus d'apprentissage.

Ce mémoire de master est divisé en trois chapitres. Le premier chapitre présente une revue des méthodes de rehaussent de la parole en passant par les techniques classiques traditionnelles pour aller vers celles qui sont basée sur l’apprentissage profond (Deep Learning) , en détaillant notre approche méthodologique et les choix techniques effectués. Le second chapitre introduit des définitions sur le Deep Learning et des réseaux de neurones récurrents (RNN). Enfin, le troisième chapitre présente les résultats expérimentaux obtenus. La conclusion générale à la fin de ce mémoire achèvera le travail.

Chapitre 1 Techniques de rehaussement de la parole

1.1 Introduction

Dans la plupart des applications du monde réel, le signal de parole propre est déformé par une combinaison de différentes catégories de distorsions avant d'être capturé par un appareil. Dans les environnements clos, par exemple, les distorsions sont généralement modélisées comme un mélange de bruit de fond additif et de réverbération. Pour faire face aux distorsions dans de tels environnements, plusieurs types de systèmes d'amélioration de la parole ont été proposés, allant des systèmes à canal unique basés sur une simple soustraction spectrale [***1***] aux systèmes à plusieurs étages qui exploitent les signaux de plusieurs microphones [***2***].

Récemment, les réseaux de neurones profonds (DNN) [***3***] ont été utilisés avec succès dans une large gamme d'applications, ayant obtenu des résultats de pointe dans des tâches telles que la reconnaissance automatique de la parole [***4***] [***5***] et la classification d'images [***6***]. La raison de ces progrès est double. Premièrement, l'évolution des systèmes informatiques et des accélérateurs matériels tels que les unités de traitement graphique (GPU) a conduit à une augmentation significative de la puissance de calcul et des capacités de stockage, qui sont deux conditions importantes pour former avec succès un réseau de neurones à plusieurs couches. Deuxièmement, les progrès des algorithmes d'optimisation et des architectures de réseau tels que les réseaux de

neurones récurrents et convolutifs ont amélioré la capacité de ces systèmes à apprendre les dépendances spatiales et temporelles des données.

1.2 Revue sur les méthodes traditionnelles du rehaussement de la parole

Le rehaussement de la parole est une étape du traitement numérique du signal audio ayant pour objectif d'augmenter la qualité de la parole, c'est-à-dire d'améliorer la clarté, l'intelligibilité, la capacité de compréhension et l'intelligibilité de la parole à l'aide d'un algorithme/filtre.

Dans cette section, nous résumons les différentes techniques classiques de débruitage audio utilisées dans la littérature.

1.2.1 Soustraction spectrale

Le développement des méthodes de rehaussement de la parole remonte à 1979 lorsque S. Boll a proposé une méthode de suppression du bruit basée sur la soustraction spectrale [***1***].

Une estimation du spectre de bruit, dérivée du signal mesuré alors que l'activité non vocale ou le début/la fin d'un signal vocal a été soustraite du spectre vocal bruyant pour obtenir un estimateur de soustraction spectrale. En d’autres termes, la soustraction spectrale est une méthode de restitution du spectre de puissance ou du spectre d'amplitude d'un signal observé en bruit additif, par soustraction d'une estimation du spectre de bruit moyen au spectre du signal bruité. Le spectre de bruit est généralement estimé et mis à jour à partir des périodes où le signal est absent et où seul le bruit est présent. Par hypothèse, le bruit est considéré comme un processus stationnaire ou à variation lente, et que le spectre de bruit ne change pas de manière significative entre les périodes de mise à jour.

Une version de celle-ci est la soustraction spectrale en amplitude. Fondamentalement, cela signifie qu'une estimation $\hat{B}(m,f)$ du spectre d'amplitude du bruit est soustraite du spectre d'amplitude de l'entrée bruitée $Y(m,f)$ tel que :

$$\hat{X}(m,f) = Y(m,f) - \hat{B}(m,f) \quad (1.1)$$

où $\hat{X}(m,f)$ est le résultat du traitement (estimée du spectre du signal non bruité). Le signal de sortie débruité $\hat{x}(t)$ du système est ainsi obtenu en retransformant $\hat{X}(m,f)$ en domaine temporel.

En raison de la quasi-stationnarité de la parole, le traitement doit être effectué trame par trame d'où m est la trame courante et f est l'indice de fréquence.

Différentes lois sont apparues donnant naissance à la soustraction spectrale de puissance définies [*7*] :

$$\left|\hat{X}(m,f)\right|^2 = |Y(m,f)|^2 - \left|\hat{B}(m,f)\right|^2 \quad (1.2)$$

Il peut être intéressant d'exprimer la soustraction spectrale puissance comme une opération de filtrage :

$$\hat{X}(m,f) = G_{SS}(m,f)Y(m,f) \quad (1.3)$$

où :

$$G_{SS}(m,f) = \frac{\sqrt{|Y(m,f)|^2 - \left|\hat{B}(m,f)\right|^2}}{Y(m,f)} = \sqrt{\frac{SNR_{pos}(m,f)}{1 + SNR_{pos}(m,f)}}$$

$$SNR_{pos}(m,f) = \frac{|Y(m,f)|^2}{\left|\hat{B}(m,f)\right|^2} - 1$$

$SNR_{pos}(m,f)$ est le rapport signal à bruit a posteriori.

1.2.2 Filtre de Wiener

La règle du filtre de Wiener est dérivée de la théorie du filtre optimal [***8***]. Elle est basée sur la minimisation de l'erreur quadratique moyenne entre la parole $X(m,f)$ et l'estimation $\hat{X}(m,f)$:

$$E\left\{\left|X(m,f)-\hat{X}(m,f)\right|^2\right\} \tag{1.4}$$

où :

$\hat{X}(m,f) = G_W(m,f)Y(m,f)$

Par hypothèse, la parole et le bruit obéissent à une distribution normale et ne sont pas corrélés.

Il est supposé :

$$E\{|Y(m,f)|^2\} = E\{|B(m,f)|^2\} + E\{|X(m,f)|^2\} \tag{1.5}$$

Après un certain traitement des équations (par lequel les valeurs attendues disparaissent également), on retrouve avec le filtre de Wiener [***9***] :

$$G_W(m,f) = \frac{SNR_{prio}(m,f)}{1+SNR_{prio}(m,f)} \tag{1.6}$$

$SNR_{prio}(m,f)$ est le rapport signal à bruit a priori.

Les méthodes de la soustraction spectrale ainsi que le filtrage de Wiener parviennent à réduire de manière très efficace le niveau de bruit de fond. Néanmoins, le bruit résiduel qui subsiste dans le signal rehaussé après modification spectrale est très gênant du point de vue perceptif [***10***].

Par conséquent, il convient, même au prix d'une complexité légèrement plus élevée, de recourir à une règle de suppression plus sophistiquée. Parmi ces méthodes, la règle de suppression d'Ephraïm et Malah [***11***] que nous allons développer dans la section suivante semble être un choix particulièrement judicieux de par sa capacité à prévenir l'apparition de bruit musical tout en restant très efficace en réduction de bruit.

1.2.3 Estimateur d'amplitude spectrale logarithmique MMSE (MMSE-LSA)

Pour être plus cohérent avec la perception auditive humaine, les auteurs dans [***11***] ont proposé un estimateur d'amplitude de l'erreur quadratique moyenne minimale (MMSE) dans le domaine log-spectral au lieu du domaine spectral de puissance tel qu'utilisé par la règle de soustraction spectrale.

Il minimise l'erreur quadratique moyenne des spectres logarithmiques du signal de parole original non perturbé et du signal de sortie traité.

$$E\left\{\left(\log\left(X(m,f)\right)-\log\left(\hat{X}(m,f)\right)\right)^2\right\} \tag{1.7}$$

Ainsi on obtient l'estimateur au sens MMSE-LSA :

$$G_{LSA}(m,f)=\frac{SNR_{prio}(m,f)}{1+SNR_{prio}(m,f)}\exp\left(\frac{1}{2}\int_{v(m,f)}^{\infty}\frac{\exp(-t)}{t}dt\right) \tag{1.8}$$

avec :

$$v(m,f)=\frac{SNR_{prio}(m,f)}{1+SNR_{prio}(m,f)}\left(SNR_{pos}(m,f)+1\right)$$

1.2.4 Estimation du bruit

L'estimation du bruit $\hat{B}(m,f)$ est tirée des pauses de parole qui sont identifié à l'aide d'un détecteur d'activité vocale (VAD). Un détecteur d'activité vocale donne des valeurs de zéro « 0 » et un « 1 » comme indicateurs de l'activité vocale dans chaque trame, ce qui permet de mettre à jour l'estimation du spectre de bruit de fond pendant les trames qui ont zéro VAD, en utilisant la formule :

$$\left|\hat{B}(m,f)\right|^2=\lambda\left|\hat{B}(m-1,f)\right|^2+(1-\lambda)|Y(m,f)|^2 \tag{1.9}$$

où $|Y(m,f)|^2$ est le spectre de la parole bruitée et λ st le facteur d'oubli.

1.3 Impact du Deep Learning sur le rehaussement de la parole (qualité et intelligibilité)

La plupart des systèmes de rehaussement de la parole dans la littérature n'utilisent qu'un minimum d'informations sur l'environnement. Les méthodes classiques de réduction de bruit, par exemple, nécessitent souvent une estimation du rapport signal sur bruit (SNR) a priori et une estimation du spectre de bruit.

Les approches traditionnelles font fréquemment des hypothèses sur la corruption qui pourraient ne pas tenir dans tous les scénarios. Par exemple, une hypothèse très utilisée de statistiques de bruit stationnaires ou à variation lente dans les procédés classiques de rehaussement de la parole a pour résultat que ces algorithmes ont des performances médiocres pour les bruits non stationnaires ou transitoires.

Cependant, les performances de ces méthodes sont limitées et échouaient donc dans des scénarios acoustiques sophistiqués. Au cours de la dernière décennie, l'apprentissage en profondeur (Deep Learning) en tant qu'outil principal pour développer des systèmes d'information basés sur les données a conduit à des avancées révolutionnaires dans le rehaussement de la parole. Dans ce contexte, le rehaussement de la parole est traité comme un problème d'apprentissage supervisé, qui ne souffre pas des problèmes rencontrés par les méthodes traditionnelles.

Ce qui suit présente les éléments constitutifs de cette méthode.

1.4 Rehaussement de la parole basée sur l'apprentissage en profondeur

Dans le domaine du rehaussement de la parole, les modèles de réseaux neuronaux servent souvent de fonction de mappage de certaines caractéristiques, soigneusement conçues ou brutes, à des cibles telles que la parole propre elle-même ou quelque chose qui peut être utilisé pour reconstruire la parole propre (par exemple, le spectrogramme d’amplitude de la parole propre).

Nouvellement, les méthodes basées sur les réseaux de neurones profonds (DNN : Deep Neural Network) ont attiré beaucoup d'attention en raison de progrès remarquables des

ressources informatiques matériel et logiciels. En termes de logiciels, l'apparition de plates-formes informatiques telles que l'architecture de dispositif unifiée de calcul (CUDA : Compute Unified Device Architecture) et les bibliothèques d'apprentissage en profondeur telles que Tensorflow [***12***] et PyTorch [***13***] ont simplifié la mise en œuvre d'algorithmes basés sur les DNN.

Côté matériel, les unités de traitement graphique (GPU) et les unités de traitement de tenseur (TPU) ont contribué par leur vitesse de calcul élevée pour les algorithmes basés sur les DNN. Cependant, les méthodes basées sur les DNN surpassent les méthodes traditionnelles à grande échelle. De plus, les DNN peut offrir un traitement à faible délai, ce qui est crucial pour de nombreuses applications en temps réel.

Les différents types des DNN, tels que les réseaux de neurones entièrement connectés (FC : Fully-Connected neural networks), les réseaux de neurones récurrents (RNN : Recurrent Neural Networks) et les réseaux de neurones convolutionnels (CNN : Convolutional Neural Networks), ont d'abord été étudiés dans le domaine de la vision par ordinateur. Dans ce domaine, les images traitées sont généralement en 2 dimensions (2D) (ou 3D pour les canaux RVB). Contrairement à l'image, le signal de parole est 1D dans le domaine temporel et présente de fortes corrélations entre les échantillons consécutifs. Pour bénéficier des approches basées sur les DNN dans le domaine du traitement de la parole, la transformée de Fourier à court terme (TFCT) de la parole dans le domaine temporel est souvent utilisée.

La TFCT d'un signal de parole peut devenir une représentation 2D (calcul du spectrogramme) dans le domaine temps-fréquence (TF) où les axes horizontal et vertical représentent les trames temporelles et les intervalles de fréquence. En conséquence, la majorité des approches basées sur les DNN dans le domaine de la vision par ordinateur peuvent être utilisées dans le domaine du traitement de la parole avec seulement quelques ajustements [***14***].

1.4.1 Principe

Dans le domaine du rehaussement de la parole, les modèles de réseaux de neurones servent souvent à cartographier certaines caractéristiques, soigneusement conçues ou

brutes, à des cibles telles que la parole propre elle-même afin d'être utilisées pour reconstruire la parole propre (spectrogramme propre parole).

Dans un premier temps, deux problèmes devraient être résolus pour le réseau, retenir un réseau neuronal approprié et choisir un ensemble de cibles de fonctionnalité approprié.

Cette approche peut présenter un inconvénient majeur est que l'ensemble des fonctions choisies peut ne pas être optimale pour des modèles de réseaux de neurones spécifiques, et qu'il peut donc ne pas produire de bonnes performances en termes d'intelligibilité et de complexité du modèle. Il est pratique de faire face à ces deux problèmes en tant que tâches distinctes dans le cas de réseaux simples avec des structures fixes, telles que DNN et LSTM entièrement connectées (LSTM : Long Short-Term Memory seront détaillé dans le prochain chapitre).

1.4.2 Système de rehaussement de la parole basée sur DNN

En règle générale, un réseau de neurones est entrainé en tant que fonction de mappage pour convertir certaines caractéristiques de la parole bruyante en certaines cibles pouvant être utilisées pour reconstruire une parole propre. Ces méthodes de rehaussement de la parole utilisant des réseaux de neurones se sont concentrées sur l'estimation de l'amplitude spectrale de la parole propre, étant donné que l'estimation de la phase spectrale avec des réseaux de neurones est difficile en raison de l'effet d'enveloppement.

Plusieurs techniques ont été étudiées et adoptées pour être adapter à différentes architectures du réseau de neurones. Ces techniques sont généralement fondées sur trois principes :

- cibles basées sur un masque [***15***].
- cibles basées sur un spectrogramme [***16***].
- cibles de forme d'onde dans le domaine temporel [***17***].

Les masques du rehaussement de la parole sont constitués d'éléments temps-fréquence qui appliquent une atténuation sur le spectrogramme de la parole bruyante pour filtrer la composante de bruit.

Dans le domaine temporel, un signal de parole bruitée $y(t)$ est généralement modélisé comme :

$$y(t) = x(t) + b(t) \tag{1.10}$$

où t, $x(t)$ et $b(t)$ représentent respectivement : l'indice temps, le signal de la parole propre non bruitée et le signal bruit.

En appliquant la TFCT, on obtient :

$$Y(m,f) = X(m,f) + B(m,f) \tag{1.11}$$

où m est l'indice trame et f est l'indice de fréquence.

Le rehaussement de la parole vise à extraire la composante de parole propre du mélange bruyant observé. Pour les approches basées sur la cartographie spectrale, la sortie est l'amplitude spectrale estimée de la parole propre, qui est notée $\left|\hat{X}(m,f)\right|$.

Pour les approches basées sur le mappage du masque, le masque $\widehat{M}(m,f)$ est estimé et peut être utilisée pour estimer l'amplitude spectrale de la parole propre par une opération de multiplication.

$$\hat{X}(m,f) = \widehat{M}(m,f)Y(m,f) \tag{1.12}$$

Le schéma fonctionnel de l'apprentissage est illustré à la Figure (1.1). Dans la phase d'entrainement, les signaux de parole propre et les signaux de bruit sont mélangés dans diverses conditions du rapport signal sur bruit (SNR), où l'extraction des caractéristiques et le calcul de la cible sont suivis.

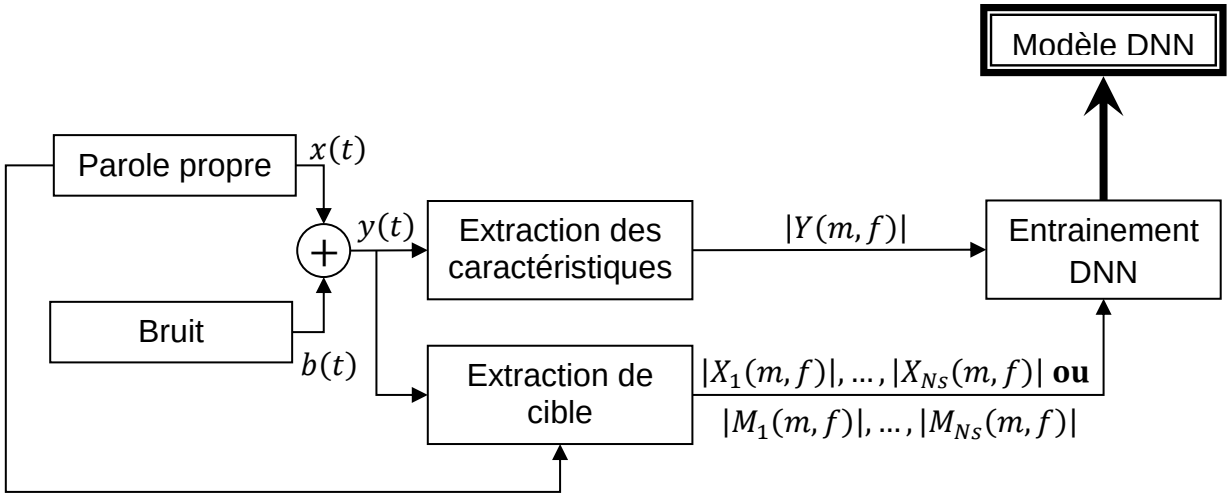

Figure 1.1 : Phase d'entrainement.

$|Y(m,f)|$ est utilisé pour désigner la caractéristique et $|X_{ns}(m,f)|$ avec $ns = 1, \dots, Ns$ pour désigner la cible. Ns est le nombre d'étages.

Dans cette technique, deux types de cibles sont adoptés, l'amplitude du spectre et le masque.

Les cibles pour différents niveaux du bruit sont envoyées au modèle d'apprentissage en profondeur pour l'entrainement.

Dans la phase de test (figure (1.2)), les caractéristiques bruyantes de l'ensemble de données de test sont introduites dans le modèle d'apprentissage en profondeur entrainé pour estimer les cibles souhaitées dans chaque unité Temps-Fréquence à différentes étapes.

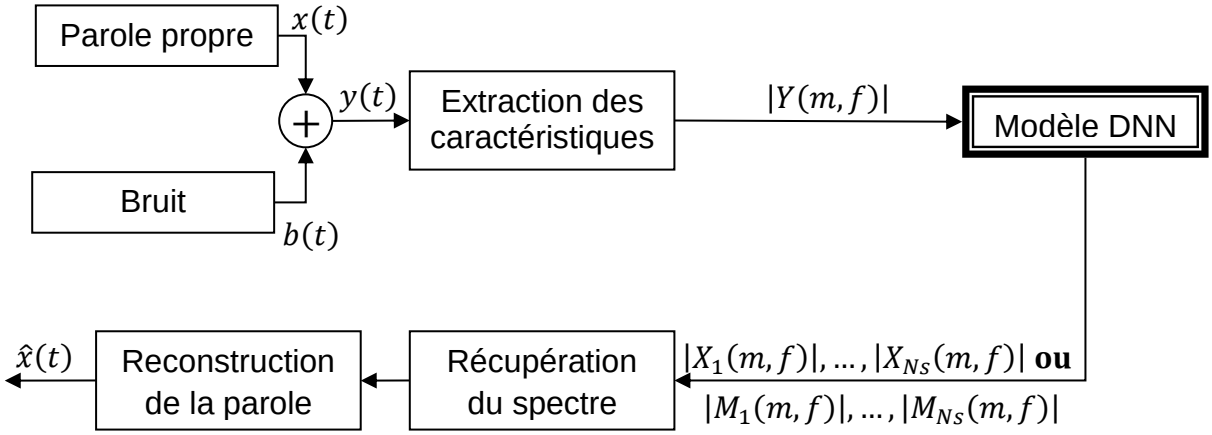

Figure 1.2 : Phase de test.

Une fois l'amplitude spectrale de la parole est estimée, le signal de parole sera reconstruit dans le domaine temporel par la transformée de Fourier inverse.

1.5 Analyse audio à court-terme

1.5.1 Principe de l'analyse à court-terme

La majorité des algorithmes de traitement de la parole qui utilisent la transformée de Fourier à court-terme traitent le spectre d'amplitude à court-terme, tout en rejetant le spectre de phase à court-terme ou en le laissant inchangé. Cela est en partie dû au spectre de phase à court-terme, qui est calculé sur de petites durées de fenêtre d'analyse de 20 à 40 ms, et qui contient peu d'informations utiles et est donc (principalement) sans importance pour le traitement de la parole (bien qu'il soit admis que le spectre de phase contribue dans une certaine mesure aux aspects de naturel et de qualité de la parole). La thèse ci-dessus a été soutenue par de nombreuses études présentées dans la littérature. Les résultats d'expériences récentes de perception de la parole suggèrent cependant que le spectre de phase (à de petites durées de fenêtre d'analyse de 20 à 40 ms) contient une quantité importante d'informations utiles, à condition que la fonction de fenêtre d'analyse soit soigneusement sélectionnée. Il a été signalé que l'utilisation de fonctions de fenêtres d'analyse non effilées (telles que la fenêtre rectangulaire) améliore considérablement l'intelligibilité du spectre de phase. Cette amélioration a été attribuée

aux caractéristiques spectrales des fenêtres d'analyse non effilées et, en particulier, à leur faible dynamique spectrale.

1.5.2 Traitement de courte durée des signaux audio

Dans les sections sur l'analyse à court terme (TFCT), nous avons discuté sur les méthodes d'analyse des signaux de parole, et nous avons constaté que nous devions diviser le signal en segments plus courts et appliquer des fonctions de fenêtrage. Ici, nous discutons des étapes supplémentaires que nous devons prendre en compte lorsque nous voulons traiter des signaux, c'est-à-dire lorsque nous voulons modifier le signal.

Pour traiter un signal fenêtré, nous avons donc besoin des étapes :

- fenêtrage
- transformée temps-fréquence telle que la DFT (facultatif)
- appliquer le traitement souhaité
- transformée temps-fréquence inverse (lorsque la DFT a été appliquée)

Les transformées temps-fréquence telles que la transformée de Fourier discrète (DFT) et la transformée en cosinus discrète (DCT) sont orthonormées et ont des algorithmes rapides bien connus pour leurs inverses.

A savoir, près des extrémités de la fenêtre, la fonction de fenêtrage va progressivement à zéro et donc l'inverse de la fonction de fenêtrage se rapproche de l'infini. Cela conduit clairement à des problèmes numériques; de très petits changements dans le signal fenêtré peuvent conduire à des échantillons arbitrairement grands après fenêtrage inverse. Nous avons donc besoin d'une méthode qui ne repose pas sur l'inversion de la fonction de fenêtrage.

1.5.3 Overlap-add

La majorité du traitement de la parole moderne est basée sur la méthode d'Overlap-add (Chevauchement-ajout), où le signal d'entrée est fenêtré dans des segments qui se chevauchent, de sorte que lorsque les parties qui se chevauchent sont additionnées, on peut parfaitement reconstruire le signal d'origine.

Le scénario le plus courant où l'Overlap-add ne peut pas être utilisé concerne les applications qui nécessitent un délai algorithmique très faible. Par exemple, les microphones de scène lors de concerts et de théâtres nécessitent un faible retard, de sorte que l'interaction entre les personnes sur scène ne soit pas perturbée.

Algorithme

- Application de la fonction de fenêtrage.
- Modifier/traiter la fenêtre avec votre algorithme de choix.
- Application de la fonction de fenêtrage encore une fois.
- Ajouter des segments qui se chevauchent pour obtenir un signal de sortie.

1.5.4 Fenêtre et reconstruction

Les segments qui se chevauchent sont additionnés. Le fenêtrage doit être choisi de manière à ce que la reconstruction soit exactement égale à l'original.

Tout l'intérêt du fenêtrage pour le traitement est de parvenir à modifier le signal fenêtré et pouvoir ensuite synthétiser le signal modifié.

Habituellement, on effectue une transformation temps-fréquence sur le signal fenêtré et effectuer des modifications dans le domaine fréquentiel.

1.5.5 La transformée de Fourier à court terme (TFCT)

L'Overlap-add est généralement combiné avec la prise de transformées de Fourier discrètes du signal fenêtré, ainsi qu'une transformée inverse après traitement. Cet algorithme est connu sous le nom de transformée de Fourier à court terme (TFCT) et c'est le domaine le plus couramment utilisé pour le traitement de la parole et de l'audio. C'est si courant que souvent, lorsque on parle d'une transformée temps-fréquence en conjonction avec des algorithmes de traitement, on entend implicitement la TFCT.

Algorithme

- Application de la fonction de fenêtrage.
- Appliquer la transformée de Fourier discrète (DFT) sur le signal fenêtré.

- Modifier/traiter la fenêtre avec votre algorithme de choix.
- Appliquer la DFT inverse sur les fenêtres modifiées.
- Appliquer à nouveau la fonction de fenêtrage.
- Ajouter des segments qui se chevauchent pour obtenir un signal de sortie.

1.6 Conclusion

Ce chapitre introduit dans un premier temps les techniques classiques de référence qui contribuent à la réduction de bruit à savoir la soustraction spectrale, le filtre de Wiener et l'estimateur d'amplitude spectrale logarithmique. Ces techniques traditionnelles nécessitent une estimation en permanence du bruit. Pour cela, un technique récursive basée sur la détection d'activité vocale à été présentée comme outil précieux pour améliorer la qualité des signaux débruités. Ces techniques peuvent être appliquées à diverses applications, telles que la suppression du bruit de fond dans les enregistrements audio, la réduction du bruit de mesure dans les capteurs, ou encore la réduction du bruit de canal dans les communications sans fil. Aussi, ces techniques présentent des limitations pour des niveaux de bruit élevés. Pour cette raison, il convient de choisir la technique appropriée en fonction de la nature du signal, du type de bruit et de l'application. De plus, avec l'avènement de nouvelles technologies telles que l'apprentissage automatique et l'intelligence artificielle, de nouvelles méthodes de réduction de bruit plus sophistiquées et plus efficaces sont développées et pourraient être utilisées pour de futures applications. En conséquence, dans un deuxième temps, ce chapitre fait un trait sur les techniques basées sur le Deep Learning DNN, d'où un système de rehaussement de la parole basée sur DNN a été présenté.

Chapitre 2 Deep Learning

2.1 Introduction

Ce chapitre explore le concept de base de l'intelligence artificielle et met en évidence ses principaux sous-domaines, tels que l'apprentissage automatique (Machine Learning) et l'apprentissage en profondeur (Deep Learning).

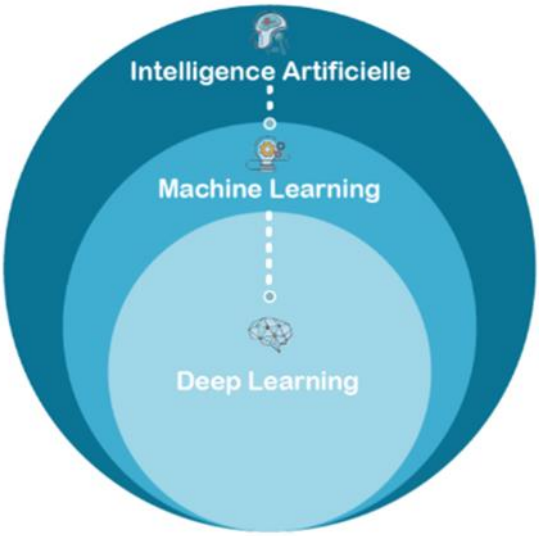

Figure 2.1 : Intelligence Artificielle, Machine Learning et Deep Learning.

2.2 Intelligence artificielle

2.2.1 Historique

Tout au long de l'histoire, les êtres humains ont constamment cherché à concevoir des outils pour améliorer leurs capacités et augmenter leur pouvoir. Dans les temps anciens, des inventions ingénieuses comme les machines d'Archimède dépassaient les limites de la force humaine brute. De plus, les humains ont également développé des

outils pour faciliter les tâches intellectuelles. Au premier siècle de notre ère, Héron d'Alexandrie a conçu la première "machine à sous" qui distribuait une quantité précise d'eau lorsqu'une pièce était insérée, utilisant des boucles de rétroaction dans un mécanisme hydraulique. Cet exemple montre que les principes de l'automatisation et de la pensée systémique, qui sont à la base de l'informatique et de l'intelligence artificielle, étaient déjà établis et appliqués il y a plus de deux mille ans.

Les premiers ordinateurs n'étaient pas si différents des machines de l'Antiquité. La différence significative réside dans la possibilité d'effectuer des opérations logiques impliquant des concepts tels que "et" et "ou", grâce à l'invention des transistors. Par la suite, la miniaturisation a permis d'augmenter le nombre de composants élémentaires sur une surface donnée, rendant ainsi possible l'exécution de calculs de plus en plus complexes. Malgré les évolutions technologiques, le désir de reproduire, voire de surpasser, la capacité de raisonnement humain reste une constante tout au long de l'histoire humaine.

2.2.2 Définition de l'Intelligence Artificielle

L'Intelligence Artificielle (IA) est un domaine de l'informatique qui vise à concevoir des systèmes intelligents en intégrant une grande quantité de connaissances et de capacités de traitement. Son objectif est de résoudre des problèmes complexes en un temps raisonnable en utilisant des connaissances spécifiques du domaine concerné. L'IA implique la programmation d'ordinateurs pour effectuer des tâches qui sont mieux réalisées par des humains et pour automatiser des activités liées à la pensée humaine, telles que la prise de décision, la résolution de problèmes et l'apprentissage.

L'IA s'intéresse également à l'étude des facultés mentales en utilisant des modèles informatiques et en simulant le raisonnement humain pour résoudre des problèmes complexes. Son objectif est d'automatiser des comportements intelligents et de produire des biens et services de manière plus économique en utilisant des ressources artificielles telles que des robots, qui sont efficaces, infatigables et obéissants. En résumé, l'IA est une approche artificielle qui permet de créer des artefacts selon des méthodes rigoureuses, en simulant certains comportements humains afin d'obtenir une meilleure efficacité et rentabilité [***18***].

2.3 Machine Learning

2.3.1 Définition

Le Machine Learning est une discipline contemporaine qui analyse les données pour détecter des motifs et des structures récurrents. En utilisant des méthodes statistiques, il est possible d'effectuer des prévisions basées sur ces analyses. En d'autres termes, le Machine Learning exploite les données pour identifier des schémas et fournir des analyses prédictives. Bien que certains des premiers algorithmes de Machine Learning, tels que le célèbre Perceptron, remontent aux années 1950, cette discipline est en constante évolution. Grâce à l'augmentation de la puissance de calcul et à la disponibilité de grandes quantités de données, le Machine Learning a connu une accélération significative [***19***].

2.3.2 Utilisation du Machine Learning

Le Machine Learning est une méthode précieuse pour résoudre des problèmes lorsque les données sont abondantes mais que les connaissances sont peu accessibles ou peu développées. Par conséquent, cette technique peut également faciliter l'apprentissage humain. Les algorithmes d'apprentissage peuvent générer des modèles qui révèlent l'importance relative de certaines informations ou la manière dont elles interagissent pour résoudre un problème spécifique [***19***].

2.3.3 Types de Machine Learning

a. *Apprentissage supervisé*

- **Définition**

Le Supervised Learning, également appelé apprentissage supervisé, est le paradigme d'apprentissage le plus couramment utilisé en Machine Learning et Deep Learning. Comme son nom l'indique, ce type d'apprentissage implique la supervision de l'apprentissage de la machine en lui fournissant des exemples de données pour lui enseigner comment accomplir une tâche spécifique [***20***].

- **Fonctionnement**

En utilisant l'apprentissage supervisé, la machine peut apprendre à effectuer une tâche en examinant des exemples de cette tâche. Par exemple, elle peut être capable de reconnaître une photo de chien après avoir été entraînée sur des millions de photos de chiens. De même, elle peut apprendre à traduire le français en chinois après avoir analysé des millions d'exemples de traduction français- chinois. En général, en examinant des millions d'exemples d'associations, la machine peut apprendre la relation qui relie une entrée à une sortie spécifique [***20***].

b. ***Apprentissage non supervisé***

- **Définition**

L'apprentissage non supervisé est une branche du Machine Learning qui se focalise sur l'analyse et le regroupement de données non étiquetées. Les algorithmes d'apprentissage non supervisé permettent de découvrir des modèles ou des groupes au sein des données avec peu ou pas d'intervention humaine. D'un point de vue mathématique, l'apprentissage non supervisé implique l'observation de multiples occurrences d'un vecteur X et l'apprentissage de la distribution de probabilité P(X) de ces occurrences [***20***].

- **Fonctionnement**

Les algorithmes non supervisés sont opérationnels dès la réception des données, sans nécessiter de formation spécifique préalable. Ils sont capables de prendre leurs propres décisions en triant les variables et en déterminant leur interdépendance. Un avantage notable de cette méthode est qu'elle ne requiert pas de données étiquetées. Le système explore les données et établit les règles en conséquence. Les algorithmes d'apprentissage non supervisé suivent un processus bien défini pour produire une sortie [***20***].

c. *Apprentissage semi-supervisé*

L'apprentissage semi-supervisé consiste à apprendre à partir d'un ensemble de données partiellement étiqueté. Le principal avantage de cette approche est d'éviter l'étiquetage complet de tous les exemples d'apprentissage, ce qui peut être fastidieux et chronophage, surtout lorsque les données sont volumineuses. Par exemple, dans le cas de la classification d'images, la collecte d'un grand nombre d'images peut être nécessaire, mais l'étiquetage manuel de chaque image peut demander beaucoup d'efforts. De plus, les étiquettes fournies par les êtres humains peuvent refléter leurs biais, qui seraient alors reproduits par un algorithme entièrement supervisé. L'apprentissage semi-supervisé peut être une solution pour éviter ces problèmes [***19***].

d. *Apprentissage par renforcement*

Dans le cadre de l'apprentissage par renforcement, le système d'apprentissage interagit avec son environnement en prenant des actions. En retour, il reçoit une récompense qui peut être positive si l'action a été bénéfique ou négative si elle a été préjudiciable. Il est possible que cette récompense soit retardée et ne soit reçue qu'après une longue séquence d'actions, comme dans le cas d'un système apprenant à jouer au jeu de go ou aux échecs. L'apprentissage par renforcement implique donc la recherche d'une politique optimale pour maximiser les récompenses obtenues sur la durée [***19***].

2.4 Deep Learning

2.4.1 Historique et évolution du Deep Learning

a. *Les premiers réseaux de neurones*

En 1943, Warren McCulloch et Walter Pitts, deux mathématiciens et neuroscientifiques, ont développé les premiers modèles de réseaux de neurones. Dans leur article intitulé "A LogicalCalculus of the Ideas Immanent in Nervous Activity", ils ont proposé une représentation mathématique du fonctionnement d'un neurone biologique. Bien que le modèle de neurone de McCulloch Pitts soit limité dans ses capacités et ne possède pas de mécanisme d'apprentissage, il a posé les bases des réseaux de neurones artificiels et de l'apprentissage profond [***21***].

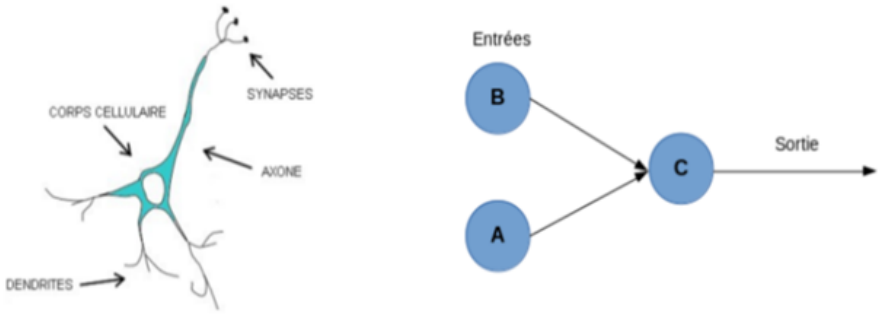

Figure 2.2 : à gauche le schéma d'un neurone biologique et à droite le schéma du neurone formel de 1943.

Chaque matériau dans l'univers a une particule sous-atomique ou le bloc de construction. Les réseaux de neurones ne font pas exception, le réseau de neurones artificiels (ANN : Artificial Neural Network) ou un perceptron à plusieurs niveaux est constitué de plusieurs perceptrons interconnectés qui sont à leur tour conçus par plusieurs unités de seuil linéaire (LTU : Linear Threshold Units). Par conséquent, un perceptron peut être considéré comme le réseau de neurones artificiels le plus simple.

b. *Unités de seuil linéaire (LTU)*

Chaque LTU a une entrée et une sortie sous forme de nombre, les entrées sont associées par un poids. La sortie est calculée en ajustant la somme pondérée des entrées et en appliquant une fonction échelon à la somme pondérée. Une fonction échelon est définie par la caractéristique d'augmenter ou de diminuer brusquement après un intervalle fixe. Les fonctions d'étape sont un type de fonctions d'activation utilisées pour le calcul de réseau neuronal et la raison pour laquelle elles sont utiles est due au fait qu'elles apportent une sorte de modèle à l'ensemble de données autrement complexe. Par conséquent, ils aident à mapper les entrées et les variables aux sorties en introduisant une non-linéarité dans l'ensemble de données.

c. *Fonctions d'activation*

Les fonctions d'activation tentent d'établir si un neurone doit être activé ou non en calculant une somme pondérée et en ajoutant un biais supplémentaire. Le but de la fonction d'activation est d'introduire une non-linéarité dans la sortie d'un neurone.

d. *Le perceptron*

Initialement, le perceptron était une simulation exécutée sur un IBM 704. Il comprenait une couche de "neurones" située entre des unités sensorielles et des unités d'activation dans le but de reconnaître des lettres de l'alphabet.

Un perceptron est composé d'une seule couche de LTU, chaque neurone étant connecté l'un à l'autre. La façon dont les perceptrons sont formés est le calcul de la variance de la sortie et du résultat attendu. Chaque instance d'entrées fait une prédiction et pour chaque neurone de sortie qui a produit une mauvaise prédiction, les poids sont renforcés pour la connexion d'entrée. Nous observons que cela est similaire à la méthodologie d'apprentissage décent du gradient stochastique (SGD).

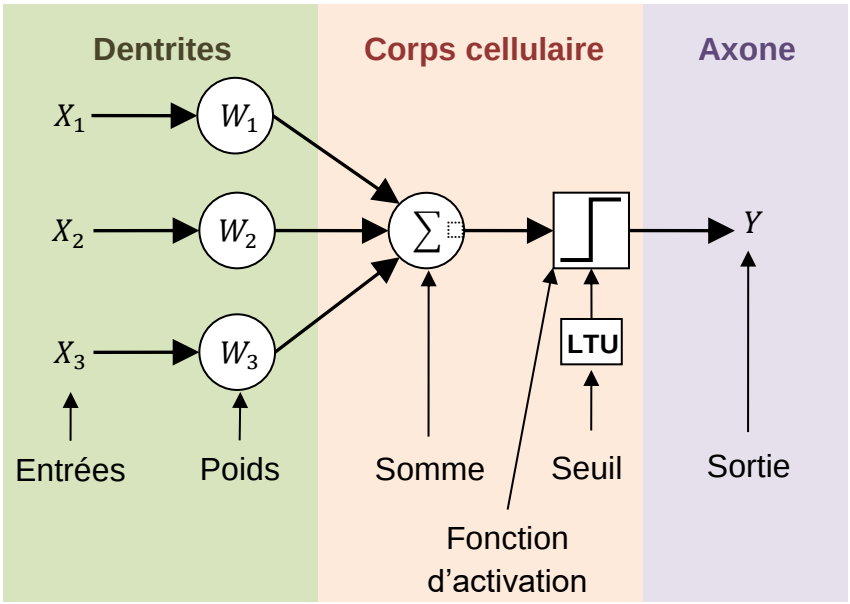

Figure 2.3 : neurone artificiel.

2.4.2 Long Short Term Memory (LSTM)

Les réseaux de mémoire longue à court terme (LSTM : Long Short Term Memory) sont un type de réseau neuronal récurrent capable d'apprendre la dépendance d'ordre dans les problèmes de prédiction de séquence. Il s'agit d'un comportement requis dans des domaines problématiques complexes tels que la traduction automatique, la reconnaissance vocale, etc.

Les LSTM sont un domaine complexe d'apprentissage en profondeur. Il peut être difficile de comprendre ce que sont les LSTM et comment des termes tels que bidirectionnel et séquence à séquence se rapportent au domaine.

Depuis 1997, l'architecture de neurones Long Short Term Memory a connu une croissance significative ces dernières années. Cette tendance s'explique en partie par l'amélioration constante des performances des machines, qui bénéficient des avancées technologiques en matière de modèles neuronaux. Par ailleurs, le machine learning dispose aujourd'hui d'outils mathématiques puissants qui ont permis de résoudre les problèmes de rétropropagation. En effet, c'est en 1997 que Sepp Hochreiter et Jürgen Schmidhuber ont conceptualisé le réseau de neurones LSTM [***22***].

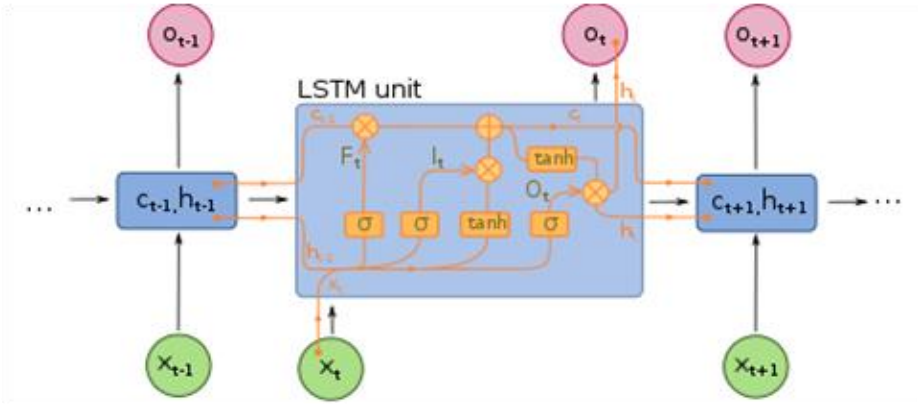

Figure 2.4 : Schéma d'un réseau LSTM à une unité.

2.4.3 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 [***3***]. Il existe de nombreux types de fonctions d'activation, voici quelques exemples :

a. *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} [***3***].

$$\text{sigmaoide}(x) = \sigma(x) = \frac{1}{1 + \exp(-x)} \qquad (2.1)$$

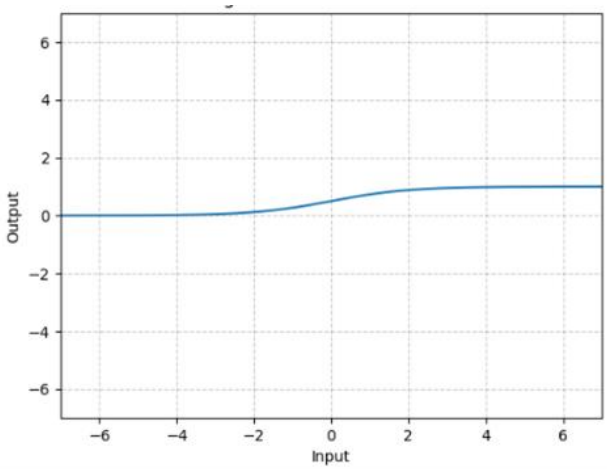

Figure 2.5 : Représentation graphique de la fonction sigmoïde.

b. *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, +∞) [***3***].

$$\mathrm{ReLU6}(x) = \min(\max(0, x), 6) \qquad (2.2)$$

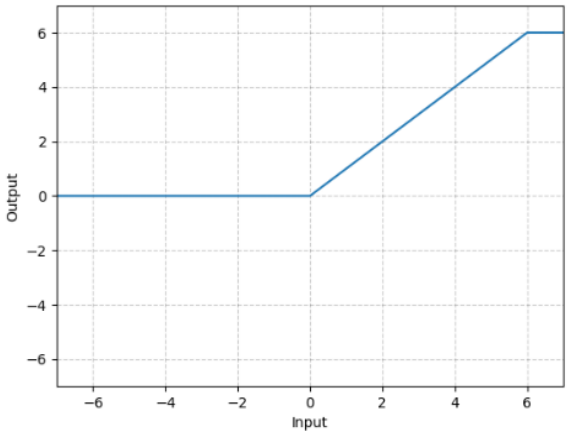

Figure 2.6 : Représentation graphique de la fonction ReLU.

c. *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 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 [***3***].

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_i)} \qquad (2.3)$$

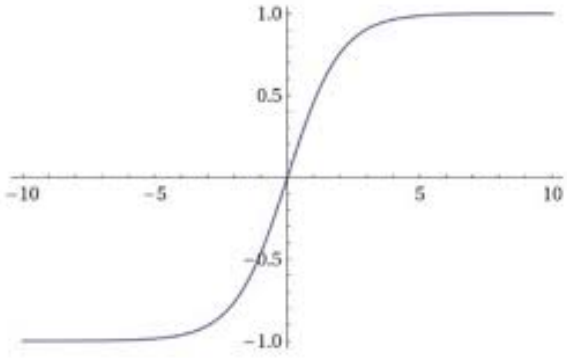

Figure 2.7 : Représentation graphique de la fonction softmax.

2.5 Conclusion

Dans ce chapitre, nous avons couvert une introduction à l'intelligence artificielle et à l'apprentissage automatique (Machine Learning), en mettant l'accent sur l'apprentissage en profondeur (Deep Learning). Nous avons présenté un aperçu des différentes applications du Deep Learning et des réseaux de neurones, tout en expliquant leur fonctionnement.

Chapitre 3 Réalisation pratique

3.1 Introduction

Afin de mettre en pratique le modèle Deep Learning pour réduire le bruit d'un clip audio, nous utilisons le langage de programmation Python et ses librairies associées. Dans ce cadre, on commence par installer Python sur l'ordinateur, ainsi que l'environnement Anaconda, ce qui facilite le processus de programmation. Ensuite, nous installons les bibliothèques nécessaires pour effectuer cette opération. L'étape de formation du modèle nécessite l'utilisation d'un grand nombre de clips audio. Enfin, nous mettons en œuvre le Deep Learning. Cette section examine les programmes et outils utilisés, la structure du réseau RNN (Recurrent Neural Network), ainsi que les résultats obtenus.

3.2 Langage de programmation utilisé (python)

Python est un langage de programmation puissant et convivial, offrant des structures de données de haut niveau et une approche simple mais efficace de la programmation orientée objet. Sa syntaxe élégante, son typage dynamique et son interprétation le rendent idéal pour la création de scripts et le développement rapide d'applications dans divers domaines, sur différentes plateformes.

L'interpréteur Python et sa vaste bibliothèque standard sont disponibles gratuitement sous forme de sources ou de binaires pour toutes les principales plates-formes. Ils sont téléchargeables sur le site https://www.python.org/. Python peut être redistribué librement, et le site propose également une large gamme de modules, programmes et outils tiers, ainsi que des ressources de documentation pour aider au développement.

3.3 L’environnement utilisé (Anaconda)

Anaconda est une distribution Python complète qui offre une vaste sélection d'outils logiciels. En plus de Python, il offre également un support pour la programmation en R. Une distribution Python est une collection de logiciels qui facilite le travail avec Python, et Anaconda est l'une de ces distributions, comprenant plusieurs programmes. Parmi les programmes inclus dans Anaconda se trouvent deux outils populaires : Jupyter Notebook et Spyder. Ces programmes sont spécialisés dans les applications liées aux données scientifiques, mais ne sont pas adaptés au développement de sites web ou d'autres types d'applications.

Spyder 3 IDE , un IDE pour python sous anaconda, a été choisi (précédemment nommé pydee dans les versions précédentes). Bien que le codage python puisse être effectué ici, il existe quelques différences par rapport à jupyter Notebook.

Bien que Spyder ne soit pas aussi interactif que Jupyter. Il est plus efficace pour créer une application complète de science des données.

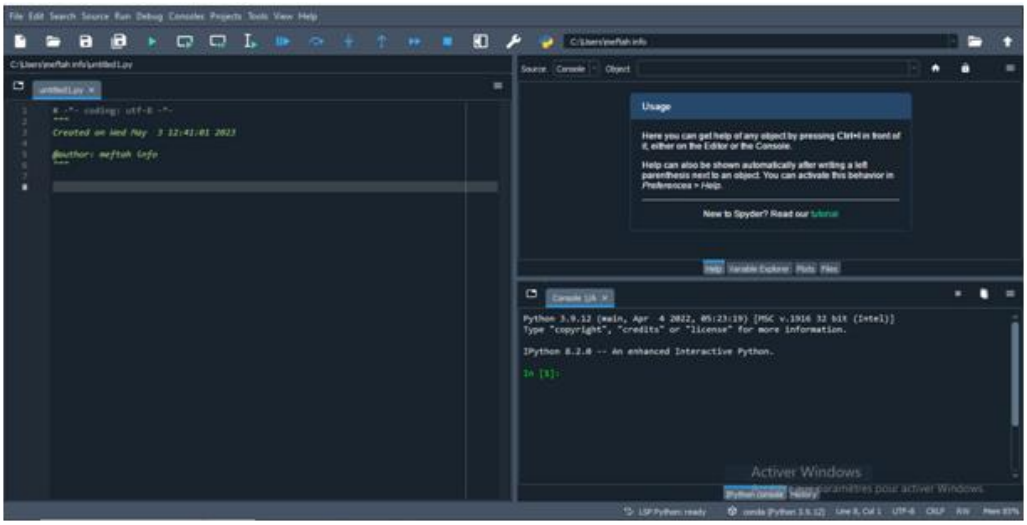

Figure 3.1 : IDE spyder interface sur Windows.

3.4 Mise en œuvre de l'environnement

Pour créer un environnement, nous avons utilisé l'interface en ligne de commande d'Anaconda et entré la commande suivante :

```
(base) C:\Users\meftah info> creat -n env_name python=3.9
```

Figure 3.2: Génération d'un environnement dans anaconda.

Une fois l'environnement créé, nous l'activons en exécutant la commande suivante :

```
(base) C:\Users\meftah info> active env_name
```

Figure 3.3: Activation d'un environnement.

Il existe deux commandes pour installer des bibliothèques dans Anaconda :

Conda install et pip install.

3.4.1 Installation des bibliothèques

a. ***Torch*** 

Torch est une bibliothèque de tenseurs légère et flexible qui utilise le langage de programmation Lua. Elle est largement utilisée par les chercheurs en apprentissage automatique, ce qui signifie que de nombreuses nouvelles idées de réseaux de neurones profonds sont d'abord mises en œuvre dans Torch et rendues disponibles sous forme d'extensions open source. Ainsi, les avancées scientifiques les plus récentes dans le domaine de l'apprentissage en profondeur sont disponibles dès leur création dans Torch.

Cependant, un inconvénient de Torch est que sa documentation peut parfois être en retard. Par conséquent, à moins de trouver un exemple précis sur GitHub, il peut être

difficile de déterminer quels modules de Torch utiliser et comment les utiliser. Par exemple, si vous souhaitez entraîner vos réseaux de neurones en utilisant plusieurs GPU, il peut être difficile de trouver des informations précises dans la documentation de Torch.

b. *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.

c. *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 "fromnumpy 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. Voici quelques caractéristiques importantes des tableaux numpy :

- Les tableaux numpy ne peuvent contenir que des objets du même type.

- La bibliothèque numpy propose de nombreuses fonctions permettant un accès rapide aux données (par exemple, la recherche et l'extraction), des manipulations variées (comme le tri) ainsi que des calculs statistiques et scientifiques, y compris des opérations élément par élément et des opérations matricielles.

- Les tableaux numpy sont plus performants en termes de vitesse et de gestion de la mémoire que les structures de données courantes en Python telles que les listes et les tuples.

- Les tableaux numpy servent de base à de nombreux packages dédiés au calcul scientifique en Python.

- Une matrice est un tableau à deux dimensions, tandis qu'un vecteur est un tableau avec une seule ligne.

- En ce qui concerne l'indexation, numpy propose différentes méthodes d'accès aux éléments, similaires à celles des listes (par exemple, l'utilisation de tranches d'éléments avec ":"), ainsi qu'une indexation par une liste (ou un tableau) d'entiers. Cela permet de faire référence aux éléments spécifiés par les indices correspondants. Il est également possible d'utiliser un tableau de booléens de même taille pour effectuer une indexation. Dans ce cas, seuls les emplacements correspondant aux valeurs True seront référencés.

- Les opérations d'indexation mentionnées ci-dessus sont également utilisables en écriture. On peut assigner une seule valeur qui sera copiée dans toutes les cases correspondantes, ou un tableau de valeurs de même forme.

- Contrairement à d'autres conteneurs, la plupart des opérateurs agissent élément par élément sur les tableaux numpy.

- Il est possible de créer des tableaux de tableaux. Lorsqu'un tableau contient des tableaux de même type et de même longueur, il devient multidimensionnel et prend en charge une indexation multiple, avec un indice par axe séparé par des virgules.

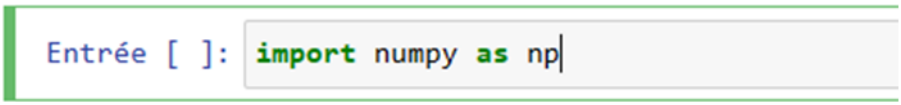

Figure 3.4: numpy python.

d. *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.

Matplotlib est conçu avec la philosophie selon laquelle vous devriez être en mesure de créer des graphiques simples avec seulement quelques commandes, voire une seule ! 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 [52].

e. *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.

3.5 L’environnement Colab

Colaboratory, ou « Colab » en abrégé, est un produit de Google Research. Colab permet à quiconque d'écrire et d'exécuter du code Python arbitraire via le navigateur, et est particulièrement bien adapté à l'apprentissage automatique, à l'analyse de données et à l'éducation.

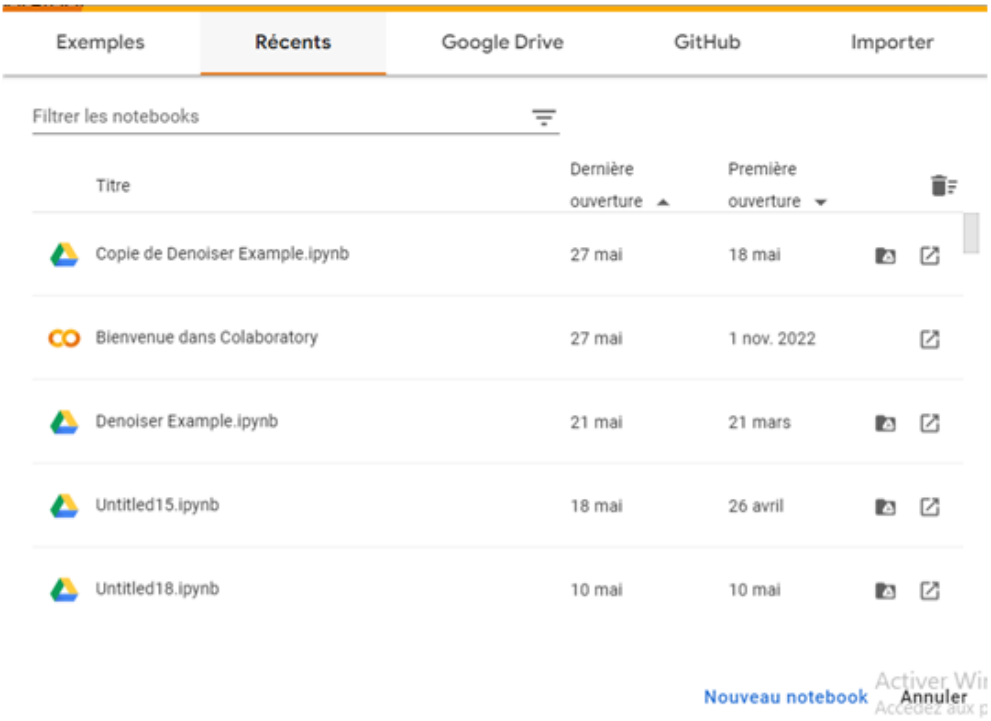

Figure 3.5: Google Colab.

Lorsqu'on utilise Google Colab dans le navigateur Web Chrome, on peut générer de nouveaux blocs-notes même si on a déjà des blocs-notes ouverts. Google Colab est une plateforme qui repose sur Jupyter et offre gratuitement des fonctionnalités pour former des modèles d'apprentissage automatique et d'apprentissage en profondeur (Deep Learning). Vous avez la possibilité de choisir entre l'utilisation du CPU, du GPU ou même du TPU pour l'entraînement. Toutefois, l'utilisation d'un GPU est généralement préférée en raison de sa rapidité d'exécution et de sa puissance de calcul supérieure par rapport aux autres options, telles que le CPU ou le TPU.

3.6 Modèles pré-entrainés

Il est important de souligner que lors de nos expérimentations, nous avons rencontré des problèmes liés à la faiblesse du réseau Internet et au manque de machines puissantes compatibles avec le développement du Deep Learning. Ces contraintes ont limité notre capacité à réaliser des expériences dans des conditions idéales.

Ces dernières années, de nombreux chercheurs et praticiens en apprentissage automatique ont commencé à exploiter de grands modèles pré-entrainés, en particulier pour des applications en vision par ordinateur et en TAL (Traitement Automatique Du Langage). Ces modèles pré-entrainés sont entrainés sur une très grande quantité de données à l'aide d'une infrastructure de cloud computing à grande échelle et sont ensuite affinés ou utilisés comme dorsale ou pour la tâche en aval.

Ce travail vise à étudier si et comment des modèles pré-entraînés comme Demucs [***23***] et Conv-Tasnet [***24***] peuvent être utilisés pour améliorer la qualité des modèles de rehaussement de la parole neuronale.

3.7 L'entraînement du modèle Demucs

3.7.1 Définition d'époque et augmentations

Les intervalles d'entraînement ont été générés à partir des extraits audio en parcourant tous les extraits de 11 secondes avec un pas d'une seconde. Un décalage de temps aléatoire entre 0 et 1 seconde était appliqué, et les 10 secondes d'audio à partir de cette position ont été utilisés pour l'entraînement [***23***].

Pour augmenter la variabilité des données d'entraînement, les sources audio ont été mélangées en permutant aléatoirement les données, générant ainsi un nouveau mixage pour êtres mises à l'échelle de manière aléatoire avec un facteur uniforme entre 0,25 et 1,25. De plus, chaque source subit une inversion de phase en étant multipliée par ±1 [***23***].

Lorsque des données supplémentaires ont été utilisées pour l'entraînement, l'augmentation de la hauteur et du tempo n'a pas était appliquée à l'algorithme Conv-Tasnet, car cela entraîne une détérioration de ses performances.

3.7.2 L’architecture Demucs

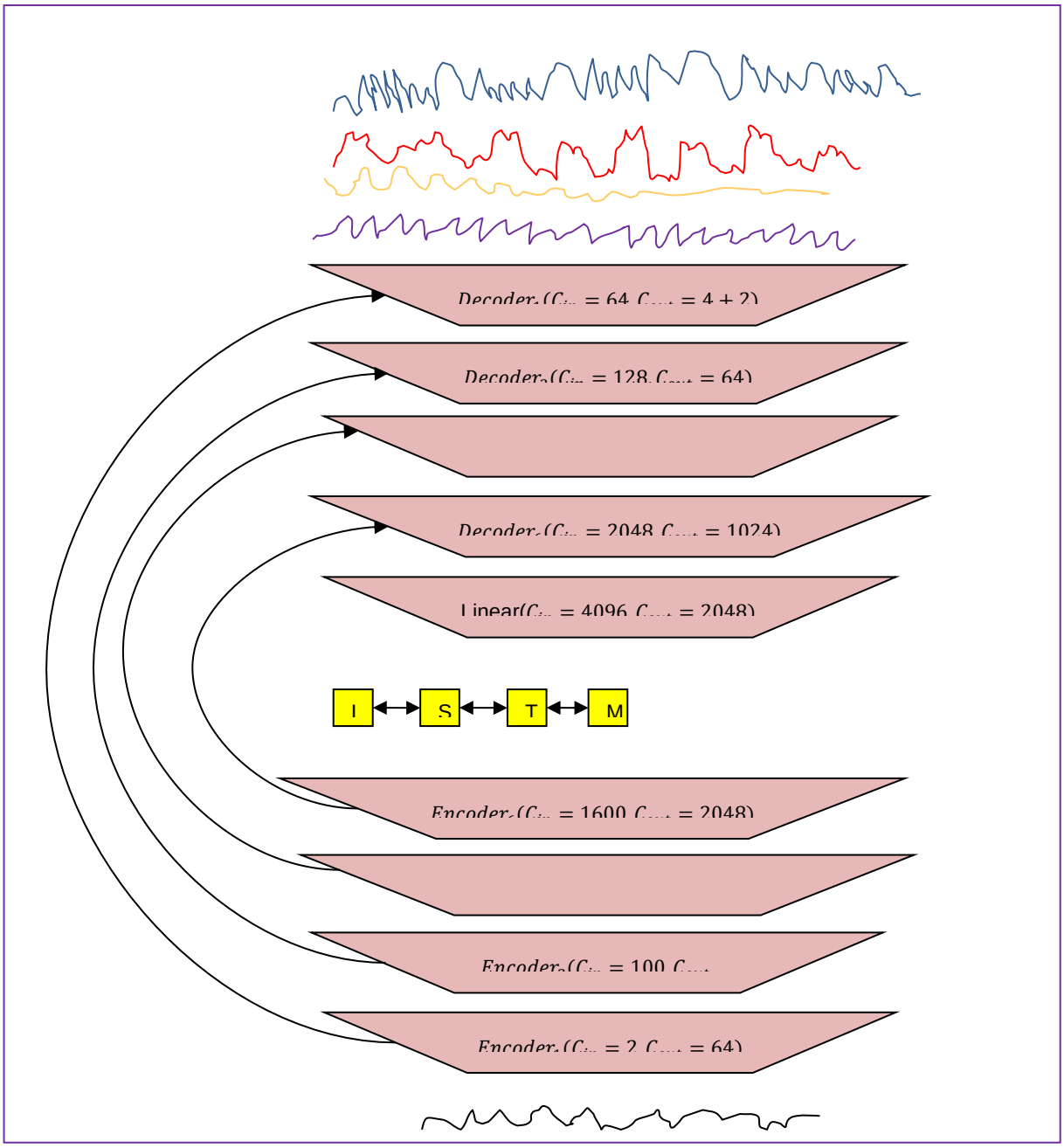

Figure 3.6: L’architecture Demucs.

Demucs consiste en un encodeur et un décodeur convolutionnels multicouches avec des connexions de saut U-net [***25***], et un réseau de modélisation de séquence appliqué sur la sortie des encodeurs. Il est caractérisé par son nombre de couches L, le nombre initial de canaux cachés H, la taille du noyau de couche K et la foulée S et le facteur de rééchantillonnage U.

Les couches de l'encodeur et de décodeur sont numérotées de 1 à L (dans l'ordre inverse pour le décodeur, donc les couches à la même échelle ont le même indice). Comme nous nous concentrons sur l'amélioration de la parole monophonique, l'entrée et la sortie du modèle n'ont qu'un seul canal.

3.8 Le test du modèle Demucs

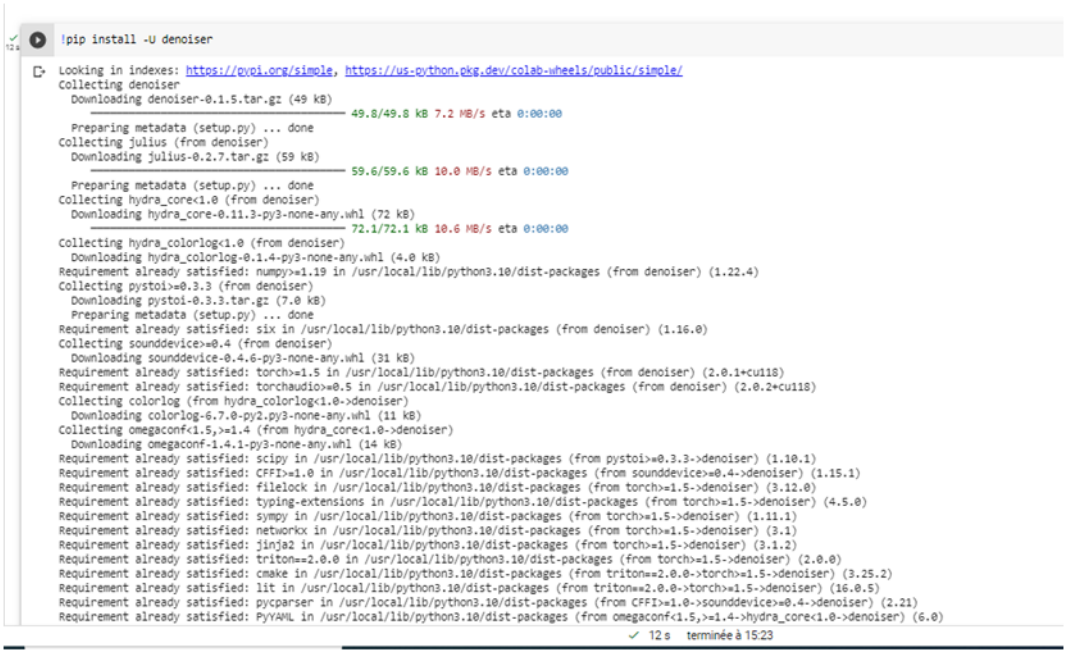

Figure 3.7: installation du débruiteur sous Colab.

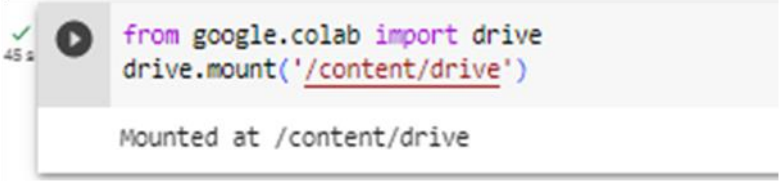

Figure 3.8: Connexion de Google Colab avec Google drive.

```
from IPython import display as disp
import torch
import torchaudio
from denoiser import pretrained
from denoiser.dsp import convert_audio
```

Figure 3.9: Installation des bibliothèques.

```
[6] model = pretrained.dns64().cuda()
    wav, sr = torchaudio.load('/content/drive/MyDrive/noisy_testset_wav/noisy_testset_wav/p232_006.wav')
    wav = convert_audio(wav.cuda(), sr, model.sample_rate, model.chin)
    with torch.no_grad():
        denoised = model(wav[None])[0]
    disp.display(disp.Audio(wav.data.cpu().numpy(), rate=model.sample_rate))
    disp.display(disp.Audio(denoised.data.cpu().numpy(), rate=model.sample_rate))
```

0:05 / 0:05

0:05 / 0:05

Figure 3.10: Importation des fichiers audio pour le test.

```
import numpy as np
import matplotlib.pyplot as plt
import librosa

# Load the audio file
audio_path = ('/content/drive/MyDrive/noisy_testset_wav/noisy_testset_wav/p232_006.wav')
audio, sr = librosa.load(audio_path, sr=None)

# Compute the spectrogram
spectrogram = librosa.stft(audio)

# Convert to dB scale
spectrogram_db = librosa.amplitude_to_db(np.abs(spectrogram))

# Display the spectrogram
plt.figure(figsize=(10, 6))
librosa.display.specshow(spectrogram_db, sr=sr, x_axis='time', y_axis='hz')
plt.colorbar(format='%+2.0f dB')
plt.title('Spectrogram')
plt.xlabel('Time')
plt.ylabel('Frequency')
plt.show()
```

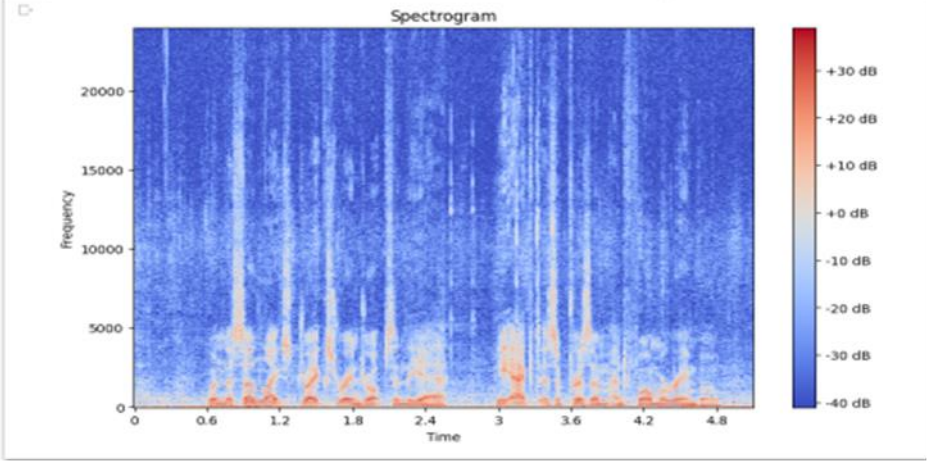

Figure 3.11: Tracé du spectrogramme.

3.9 Conclusion

Dans ce chapitre, nous avons exploré l'efficacité des algorithmes RNN dans la réduction du bruit et l'optimisation du signal de parole. Grâce à leur capacité à exploiter les informations temporelles et à modéliser les relations complexes entre les signaux de parole bruyants et propres, les RNN ont démontré des résultats prometteurs. L'utilisation des RNN dans ce contexte permet de reconstruire le signal de parole propre en utilisant les informations contextuelles des séquences de données antérieures, même lorsque le signal est partiellement masqué par le bruit. Cela ouvre des perspectives intéressantes pour améliorer la qualité de la parole et rendre les applications de traitement de la parole plus performantes.

Conclusion générale

Dans ce mémoire, nous avons rédigé une introduction à l'intelligence artificielle et à l'apprentissage automatique, en mettant l'accent sur l'apprentissage profond, plus précisément le deep learning. Nous avons présenté un aperçu des différentes applications du deep learning et des réseaux neuronaux, en mettant l'accent sur son utilisation pour réduire le bruit dans les clips audio.

Dans cette étude, notre objectif était de présenter le fonctionnement du deep learning pour réduire le bruit dans les clips audio en utilisant un modèle qui tire parti des fonctions de deep learning disponibles dans les bibliothèques sélectionnées.

Nous avons choisi d'utiliser un algorithme d'apprentissage en profondeur basé sur les réseaux de neurones récurrents (RNN) en raison de sa capacité à traiter efficacement les signaux audio contaminés par des bruits indésirables. Ce type d'algorithme apprend à partir de grandes quantités de données et est capable de reconnaître et de supprimer le bruit dans les signaux audio, ce qui améliore considérablement leur qualité d'enregistrement.

L'un des principaux avantages de l'utilisation de l'apprentissage en profondeur pour la réduction du bruit est sa capacité à s'adapter à différents types de bruit. Les modèles RNN peuvent apprendre à détecter et à supprimer diverses caractéristiques du bruit, qu'il s'agisse d'un bruit constant, d'un bruit impulsionnel ou d'un bruit de fond variable.

De plus, les algorithmes d'apprentissage en profondeur peuvent être entraînés sur de grandes quantités de données, ce qui leur permet d'acquérir une connaissance approfondie des caractéristiques du bruit et des signaux audio correspondants. Cela leur permet d'améliorer efficacement les performances de débruitage lorsqu'ils sont appliqués à de nouvelles données.

Il convient cependant de noter que les performances de débruitage dépendent également de la qualité des données d'entraînement utilisées pour former le modèle. Il est essentiel d'utiliser des ensembles de données représentatifs et de haute qualité pour obtenir de bons résultats. De plus, l'optimisation et le réglage des hyperparamètres de l'algorithme peuvent jouer un rôle important dans l'obtention de performances optimales.

Il est également important de souligner que lors de nos expérimentations, nous avons rencontré des problèmes liés à la faiblesse du réseau Internet et au manque de machines puissantes compatibles avec le développement du deep learning. Ces contraintes ont limité notre capacité à réaliser des expériences dans des conditions idéales.

A la fin, la réduction du bruit par le biais du deep learning, en particulier avec l'algorithme RNN, présente un grand potentiel pour améliorer la qualité des signaux audio en supprimant les bruits indésirables. Avec des ensembles de données appropriés et un réglage adéquat des hyper paramètres, ces algorithmes peuvent fournir des résultats prometteurs dans diverses applications audio.

Bibliographie

[1] S. Boll, "A spectral subtraction algorithm for suppression of acoustic noise in speech," *in IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '79*, vol. 4, pp. 200–203, Apr. 1979.

[2] E. A. P. Habets, "Single- and multi-microphone speech dereverberation using spectral enhancement," *Citeseer*, vol. 68, no. 4, 2007.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[4] Geoffrey Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82-97, 2012.

[5] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6645–6649, May 2013.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *in Advances in neural information processing systems*, pp. 1097–1105, 2012.

[7] P. Vary, "Noise suppression by spectral magnitude estimation—mechanism and theoretical limits," *EURASIP Signal Processing*, vol. 8, pp. 387-400, 1985.

[8] S. V. Vaseghi, *Advanced Signal Processing and Digital Noise Reduction*.: Wiley Teubner, 1996.

[9] A. A. Azirani, R. L. B. Jeannes, and G. Faucon, "Speech Enhancement Using a Wiener Filtering Under Signal Prescence Uncertainty," *in Proceedings European Signal Processing Conference*, September 1996.

[10] Olivier Cappé, "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor," *IEEE transactions on Speech and Audio Processing*, pp. 345-349, Apr. 1994.

[11] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Trans. on ASSP*, pp. 443-445, 1985.

[12] M. Abadi, A. Agarwal, and al. (2015) Software available from tensorflow.org. [Online]. http://download.tensorflow.org/paper/whitepaper2015.pdf

[13] A. Paszke, S. Gross, and al. (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. [Online]. https://arxiv.org/abs/1912.01703

[14] Z. Ouyang, "Single-channel speech enhancement based on deep neural network," Concordia University, Mémoire de Master 2020.

[15] D. S. Williamson, Y. Wang, and D. Wang, "Complex ratio masking for monaural speech separation," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 24, no. 3, pp. 483-492, March 2016.

[16] S.W. Fu, T.Y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," *IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1-6, 2017.

[17] D. Rethage, J. Pons, and X. Serra, "A wavenet for speech denoising," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5069-5073, April 2018.

[18] Seyyed Mohammad Reza Farshchi, *Une introduction avancée à l'intelligence artificielle: La science de l'intelligence artificielle (French Edition)*.: Éditions universitaires européennes, June 22, 2020.

[19] V. Cherkassky and F. Mulier, *Learning from Data : Concepts, Theory, and Methods.* New York: Wiley, 1998.

[20] Anaimalaza Serge Randrianasolo, *Conception et mise en place d'un système d'apprentissage non-supervisé*., September 15, 2016.

[21] Warren S. McCulloc and Walter Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133 , 1943.

[22] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, December 1997.

[23] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. https://arxiv.org/abs/1911.13254.

[24] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing* , 2019.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *International Conference on Medical image computing and computer-assisted intervention*, 2015.