République Algérienne Démocratique et Populaire

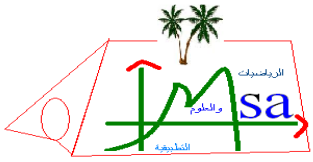
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université de Ghardaïa

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Laboratoire de Mathématiques et Sciences Appliquées



Mémoire présenté pour l'obtention du diplôme de **Master en Informatique**

Spécialité : Systèmes Intelligents pour l'Extraction de Connaissances

Thème

Détection d'activité vocale utilisant l'apprentissage profond

Présenté par :

Meroua Khene et Sabrina Souid

Devant le jury :

M. Youcef Mahdjoub	MAA	Université de Ghardaïa	Président
M. Abdelkader Bouhani	MAA	Université de Ghardaïa	Examinateur
M. Slimane Bellaouar	MCB	Université de Ghardaïa	Coencadreur
M. Abderrahmane Adjila	MAA	Université de Ghardaïa	Encadreur

Année universitaire : 2019 - 2020

Dédicaces

Je dédie ce mémoire

À cette source de tendresse, de patience et de générosité qu'ils m'ont apportée et cette confiance qu'ils ont placée en moi. À mes chers parents, qui m'a soutenu et encouragé durant ces années d'études. Auxquelles je dois ce que je suis aujourd'hui, Je prie le bon Dieu de vous donnez une longue vie et une santé de fer.

À mes chers frères et sœurs pour tout l'amour dont vous m'avez apporté. À mes grands-mères et toute ma famille et mes proches et à ceux qui me donnent de la vivacité.

À mes chères amies en souvenir des plus beaux instants qu'on a passés ensemble. À tous ceux qui par un mot m'ont donné la force de continuer je vous remercie de tout cœur.

Khene Meroua

Dédicaces

Je dédie ce travail

Aux chers gens et rapprochez-les de mon cœur, ma chère mère et mon cher père, qui m 'ont aidé et soutenu, et leurs supplications bénies ont eu le plus grand effet dans la conduite du navire de recherche jusqu 'à ce qu 'il s 'ancre dans cette image.

À mon frère et mes chères sœurs.

À mes professeurs et aux personnes de crédit qui m 'ont submergé d 'amour, d 'appréciation, de conseils, d 'orientation et de conseil.

À ma chère collègue et sœur, Meroua khene. Pour

chacun de la famille Souid et messaitfa.

J 'offre un dédicace spécial aux bourgeons de la famille, mes neveux

Mohammad Taha et Ossama.

À mes chers amis, surtout Massi ben mira .

À tous les étudiants Master 2 informatique promo 2019/2020.

A tous ceux qui m 'ont donné la force de continuer, je vous remercie du fond du cœur.

Souid Sabrina

Remerciement

Nous remercions tout d'abord Allah SWT, tout puissant, de nous avoir donné la volonté, l'aide, la patience et le courage pour accomplir ce projet, ainsi que la force et l'audace pour dépasser toutes les difficultés.

A notre encadrant, M. Abderrahmane Adjila. Votre compétence, votre encadrement ont toujours suscité notre profond respect. On vous remercie pour votre accueil et vos conseils. Veuillez trouvez ici, l'expression de nos gratitudes et de notre grande estime.

Nos profondes gratitudes s'orientent vers M.Slimane Bellaouar pour nous aider, ses judicieux conseils et son support permanent.

Messieurs les jurys, vous nous faites un grand honneur en acceptant de juger ce travail.

On voudrait également remercier à toute personne ayant aidé de près ou de loin dans la réalisation de ce travail.

Résumé

La détection d'activité vocale (DAV) est considérée comme l'une des principales techniques pour de nombreuses applications vocales. C'est une méthode importante dans le traitement de la parole, car elle détecte la présence ou l'absence d'une voix humaine. Auparavant, les performances de la DAV étaient basées sur des méthodes qui dépendent du traitement du signal, mais ne donnaient pas des performances satisfaisantes dans des environnements à bruit élevé, donc l'apprentissage profond est devenu une alternative. À partir de là, nous avons adoptés dans l'étude expérimentale sur trois structures pour l'apprentissage profond qui sont la mémoire à long et court terme (LSTM), l'unité récurrente fermée (GRU) et un réseau DenseNet, et nous avons également utilisés les deux bases de données pour la parole et le bruit, qui sont LibriSpeech et QUT-NOISE successivement. Nous avons mesurés la précision du WebRTC dans des environnements à faible bruit avec diverses sensibilités et nous avons obtenus une précision de 98%.

Mots clés :Réseaux de neurones, apprentissage profond, réseaux de neurones récurrents, Réseaux de neurones convolutifs, détection d'activité vocale, rapport signal sur bruit.

Abstract

Voice activity detection (VAD) is considered to be one of the main techniques for many voice applications. It is an important method in speech processing, because it detects the presence or absence of a human voice. Previously, the performance of the VAD was based on methods that depend on signal processing, but did not give satisfactory performance in high noise environments, so deep learning became an alternative. From there, We adopted in the experimental study of three structures for deep learning, long short-term memory (LSTM), gated recurrent unit (GRU) and a DenseNet network, and we also used the two databases for speech and noise, which are LibriSpeech and QUT-NOISE successively. We measured the accuracy of WebRTC in low noise environments with various sensitivities and we got accuracy of 98%.

Key words : Neural networks, deep learning, recurrent neural networks, Convolutional Neural Networks, voice activity detection, signal-to-noise ratio.

ملخص

يعد اكتشاف النشاط الصوتي كتقنية من التقنيات الرئيسية للعديد من تطبيقات الكلام، فهو طريقة مهمة في معالجة الكلام، حيث يقوم بالكشف عن وجود أو غياب الصوت البشري. سابقا، تم الاعتماد في أداء اكتشاف النشاط الصوتي على طرق تعتمد على معالجة الاشارات، لكن لم تعطي أداءً مرضيًا في البيئات ذات الضوضاء العالية، لذلك أصبح التعلم العميق بديلاً. ومنه اعتمدنا في الدراسة التجريبية على ثلاث بنيات للتعلم العميق وهي الذاكرة طويلة قصيرة المدى LSTM، الوحدة المتكررة ذات البوابات GRU وشبكة DenseNet، كما استخدمنا مجموعتي البيانات للكلام والضوضاء، وهي LibriSpeech وQUT-NOISE على التوالي. قمنا بقياس دقة WebRTC في البيئات ذات ضوضاء منخفضة وبحساسيات مختلفة وحصلنا على دقة 98%.

الكلمات المفتاحية: الشبكات العصبية، التعلم العميق، الشبكات العصبية المتكررة، الشبكات العصبية التلافيفية، كاشف النشاط الصوتي، نسبة الإشارة إلى الضوضاء.

Table des matières

Table des figures

Liste des tableaux

liste des abréviations

ADAM	**Ada**ptive **M**oment Estimation
AMR	**A**daptatif **M**ulti-**R**ate
ATRAC	**A**daptive **T**ransform **A**coustic **C**oding
AUC	**A**rea **U**nder the **C**urve
BLSTM	**B**idirectionnel **LSTM**
BPTT	**B**ack-**P**ropagation **T**hrough **T**ime
BRNN	**B**idirectionnel **RNN**
CDA	**C**ompact **D**isc **A**udio
CE	**C**ross **E**ntropy
CLDNN	**C**onvolutional **L**ong short terme memory **D**eep**N**eural **N**etworks
CNN	**C**onvolutional **N**eural **N**etworks
DAV	**D**étection **A**ctivité **V**ocale
dB	**d**éci**B**els
DNN	**D**eep **N**eural **N**etworks
EER	**E**qual **E**rror **R**ate
FAR	**F**alse **A**cceptance **R**ate
FFT	**F**ast **F**ourier **T**ransform
FIR	**F**inite **I**mpulse **R**esponse
FL	**F**ocal **L**oss
FLAC	**F**ree**L**ossless **A**udio **C**odec
FRR	**F**alse **R**ejection **R**ate
GMM	**G**aussian **M**ixture **M**odel
GOF	**G**oodness **O**f **F**it

GPU	**G**raphic **P**rocessing **U**nit
GRU	**G**ated **R**ecurrent **U**nit
HMM	**H**idden **M**arkov **M**odel
KS	**K**olmogorov **S**mirnov
LLR	**L**og**L**ikelihood **R**atio
LPC	**L**inear **P**redictive **C**oding
LRT	**L**ikelihood **R**atio **T**est
LSTM	**L**ong **S**hort **T**erme **M**emory
MFCC	**M**el **F**requency **C**epstral **C**oefficients
MS	**M**odulation **S**pectrum
PLP	**P**erceptual **L**inear **P**redictive
PMC	**P**erceptron **M**ulti-**C**ouche
RAP	**R**econnaissance **A**utomatique de la **P**arole
RASTA-PLP	**R**el**A**tive **S**pac**T**r**A**l-PLP
ReLU	**Re**ctified **L**inear **U**nit
RMSE	**R**oot **M**ean **S**quare **E**rror
RNA	**R**éseaux de **N**eurones **A**rtificiels
RNN	**R**ecurrent **N**eural **N**etworks
ROC	**R**eceiver **O**perating **C**haracteristic
RSB	**R**apport **S**ignal sur **B**ruit
SGD	**S**tochastic **G**radient **D**escent
SM	**S**tate **M**achine
Tanh	**T**angente **h**yperbolique
TFCT	**T**ransformée de **F**ourier à **C**ourt **T**erme

TFD **T**ransformée de **F**ourier **D**iscrète

WAV **Wav**e form

WebRTC **Web R**eal-**T**ime **C**ommunication

WER **W**ord **E**rror **R**ate

WMA **W**indows **M**edia de **A**udio

ZCR **Z**ero **C**rossing **R**ate

Chapitre 1

Introduction générale

La plupart du temps, les systèmes de reconnaissance automatique de la parole (RAP) nécessitent l'unité de détection d'activité vocale (DAV) comme étape de prétraitement afin d'identifier les parties de la voix qui contiennent la parole, tel que la technique de la DAV est pour distinguer la parole et la non-parole dans le signal audio.

La DAV représente un défi pour la parole bruité, en particulier dans le scénario de faible rapport signal sur bruit (RSB), en raison de la grande variation des signaux de paroles et non paroles.

Ces dernières années, la plupart des recherches se sont concentrées sur les méthodes d'apprentissage profond qui prennent la DAV comme problème de classification binaire, car les réseaux de neurones récurrents ont montré de bons résultats en tant qu'architecture qui gère efficacement les données séquentielles.

Afin de détecter une activité vocale dans des environnements bruyants, nous avons mené une étude sur une expérimentation impliquant la performance de trois architectures différentes : les réseaux de neurones récurrents de mémoire à long et court terme (LSTM-RNN), les réseaux de neurones récurrents d'unité récurrente fermée (GRU), et la mise en œuvre d'un réseau DenseNet qui est l'un des architectures d'un réseau de neurone convolutif, qui fonctionne mieux avec moins de complexité. Nous avons utilisés deux ensembles de données dans cette expérimentation, l'ensemble de données de la parole "LibriSpeech" et l'ensemble de données du bruit "QUT-NOISE".

Pour compléter cette étude, nous appuyons sur 4 chapitres :

Le premier chapitre : Nous présentons dans ce chapitre les concepts de base des réseaux de neurones, ensuite nous introduisons l'approche d'apprentissage profond où nous concentrons sur les réseaux de neurones convolutifs et les réseaux de neurones récurrents.

Le deuxième chapitre : Comprend tout ce qui concerne le traitement du signal, à commencer par la génération de la parole, le bruit, comment convertir un signal analogique en numérique et comment stocker des fichiers audios. Passant à la définition de la DAV et leur principe, nous introduisons diverses techniques pour extraire les caractéristiques du signal audio.

Le troisième chapitre : Nous appuyons sur la présentation des études précédentes qui incluent la DAV de différentes manières, ce chapitre est basé sur 3 approches, une approche basée sur le traitement du signal, une approche basée sur les modèles statistiques et la troisième basée sur l'apprentissage profond.

Le quatrième chapitre : Présente les expérimentations menées sur la DAV avec différentes architectures de l'apprentissage profond, et nous discutons également des différents résultats obtenus.

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

Chapitre 2

Apprentissage profond

2.1 Introduction

L'intelligence artificielle a été développée pour simuler le comportement du cerveau humain. Les premières tentatives de modélisation du cerveau sont anciennes, même avant l'ère de l'informatique. Dans ce sens, les scientifiques ont pensé à essayer d'imiter le fonctionnement de l'esprit humain et ont découvert que le neurone est l'élément le plus important pour la formation et la collecte du cerveau. Dans cette perspective, des études ont commencé sur le mécanisme des réseaux neuronaux biologiques pour simuler leur travail sur l'ordinateur pour résoudre des problèmes complexes. Par conséquent, des réseaux neuronaux artificielles ont été créés.

L'apprentissage profond est un sous-ensemble de l'apprentissage automatique où ce dernier est l'un des domaines de l'intelligence artificielle. L'apprentissage profond est basé sur l'idée des réseaux neuronaux artificielles, donc ces concepts sont interconnectés même s'ils ne sont pas équivalents.

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

2.2 Neurones biologiques

Le système nerveux central contient des cellules appelées neurones, dont le nombre dépasse 10^{12} neurones, ce qui signifie l'équivalent d'un plus de 1000 milliards de neurones.

Il se compose de trois composants de base, qui sont illustrés dans la figure 2.1.

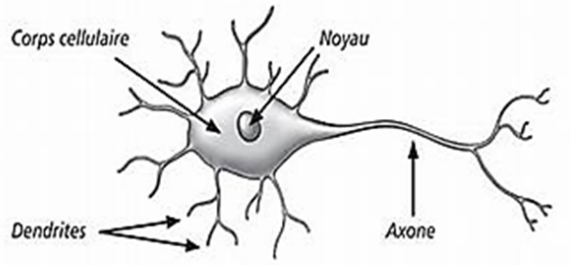


FIGURE 2.1 – Neurone biologique

1) Corps cellulaire : Le corps cellulaire contient un noyau, qui est le principal responsable de la réalisation de transformations biochimiques qui aident à la synthèse de

certains éléments qui garantissent la vie du neurone.

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

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

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

2.3 Du neurone biologique au neurone formel

Le neurone formel est considéré comme un modèle mathématique ou une fonction algebrique modélisant les principes applicables dans le neurone biologique, dont la valeur dépend de parametre appellés poids, voir la figure 2.2. Cette transition et cette modélisation sont bien illustrées dans la table 2.1.

TABLE 2.1 – Analogie entre le neurone biologique et le neurone formel

Neurone biologique	**Neurone formel**
Synapses	Poids des connexions
Axones	Signal de sortie
Dendrites	Signal d'entrée
Noyau ou Somma	Fonction d'activation

Le neurone formel se compose d'un noyau, d'une réticulation et d'une entrée réticulée, qui reçoit un certain nombre d'entrées accompagnées de poids, appelés poids synaptique, qui est considéré comme un signe de la force de contact.

En général, il est connu du neurone formel qu'il effectue trois opérations appliquées à ses entrées :

-Pondération : Multipliez les entrées par des valeurs appelées poids.

-sommation : Agrégation des valeurs obtenues après pondération.

-Activation : Passez la valeur obtenue après l'ajout dans la fonction d'activation.

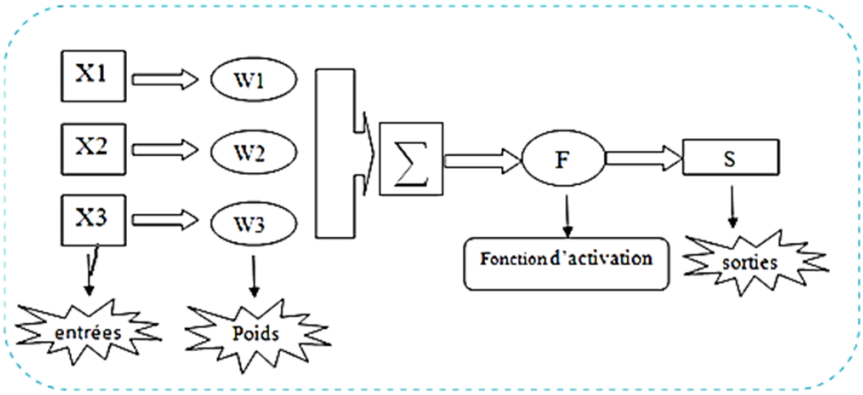


FIGURE 2.2 – Exemple d'un neurone formel

2.4 Réseaux de neurones artificiels (RNA)

Les réseaux de neurones ont reçu beaucoup d'attention de la part des scientifiques et des chercheurs. Ils sont connus comme un modèle informatique composé de petits éléments simples qui s'inspirent du système nerveux biologique et fonctionnent en parallèle appelés neurones.

La structure des réseaux de neurones est représentée par la présence de la couche d'entrée, une couche plus cachée en plus de la couche de sortie étroitement connectée par des neurones (nœuds), comme montre la figure 2.3.

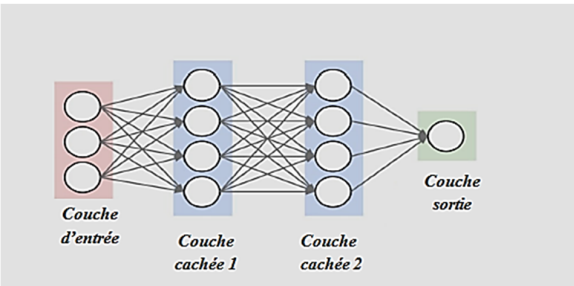


FIGURE 2.3 – Réseau de neurone artificiel

• **Fonction d'activation**

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

1) Fonction sigmoïde : La fonction sigmoïde est l'une des fonctions d'activation non linéaires les plus importantes et les plus couramment utilisées, limitant les valeurs à varier entre 0 et 1 de sorte que si les valeurs sont des nombres positifs, le résultat est 1, et si les valeurs sont des nombres négatives ou nul, le résultat remise à 0, comme illustré par la figure 2.4 [1].

La fonction sigmoïde est donnée par la relation suivante :

$$f(x) = \frac{1}{1 + \mathrm{e}^{-x}} \qquad (2.1)$$

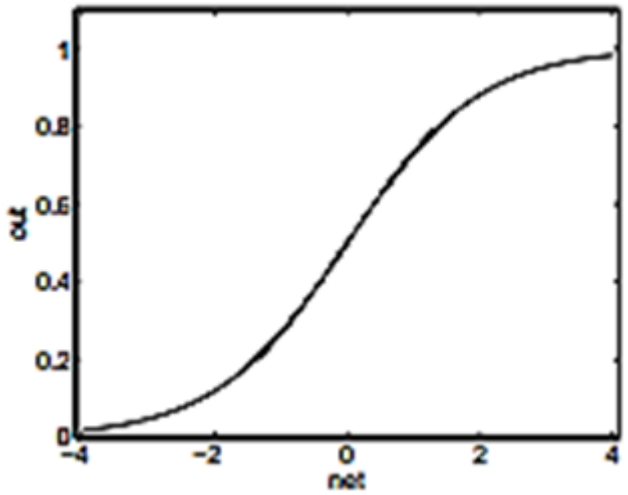


FIGURE 2.4 – Graphique de la fonction sigmoïde

2) Fonction Tangente hyperbolique (tanh) : La fonction tangente hyperbolique est une version similaire à la fonction sigmoïde sauf qu'au point de sortie est spécifié entre -1 et 1, comme montre la figure 2.5 [1].

La fonction tangente hyperbolique Connu par la relation :

$$f(x) = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{\mathrm{e}^{x} + \mathrm{e}^{-x}} \qquad (2.2)$$

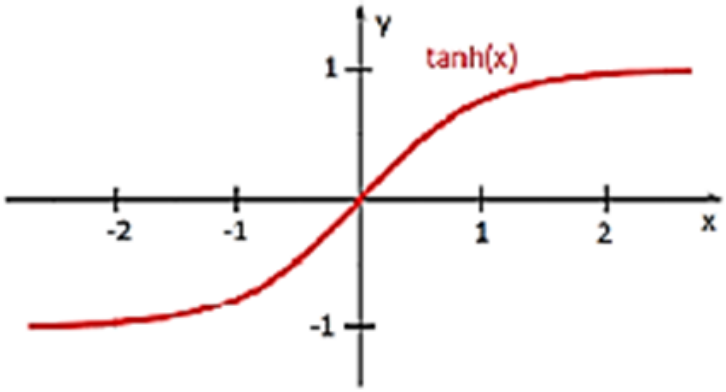


FIGURE 2.5 – Graphique de la fonction tangente hyperbolique

3) Fonction Unité Linéaire Rectifiée (Rectified Linear Unit/ReLU) : Le résultat de cette fonction est 0 si les valeurs d'entrée sont négatives, au contraire les valeurs de sortie restent les mêmes que les valeurs d'entrée, comme illustrée par la figure 2.6 [1].

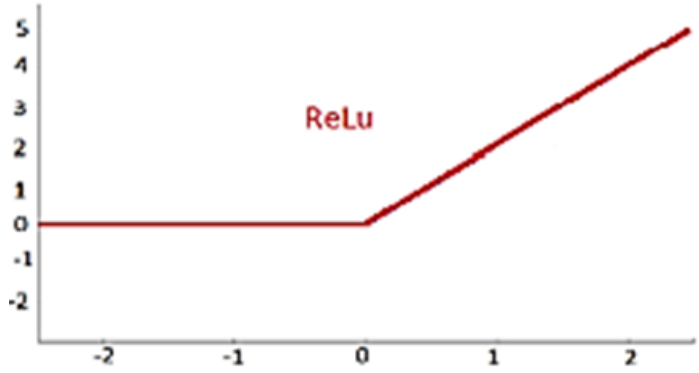


FIGURE 2.6 – Graphique de la fonction ReLU

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

$$f(x) = max(0, x) \qquad (2.3)$$

Où :

$$f(x) = \begin{cases} 0 \quad si\ x < 0 \\ \\ x \quad si\ x \geq 0 \end{cases} \qquad (2.4)$$

4)Fonction softmax : La fonction softmax est une régression logistique dans un cas si nous voulons aborder plusieurs classes, elle est utilisée dans les problèmes de multiclassification [1].

Softmax est calculé avec la formule ci-dessous :

$$f(x_i) = \frac{\mathrm{e}^{x_i}}{\sum_{i=1}^{k}(\mathrm{e}^{x_c})} \qquad (2.5)$$

Où K est le nombre du classes.

• **Perceptron**

1)Perceptron simple : Le perceptron simple ou monocouche, a été développé par Rosenblatt, composé d'un neurone, comme illusté dans la figure 2.7, il est défini par une fonction d'activation comme suit [2] :

$$f(x) = \begin{cases} 1 \qquad\quad \text{si } y > 0 \\ \\ 0(ou-1) \quad si\ y \leq 0 \end{cases} \qquad (2.6)$$

où y est le produit scalaire des entrées avec les poids, c'est-à-dire :

$$y = \vec{w}.\vec{x} = \sum_{i=1}^{n}(w_i x_i) \qquad (2.7)$$

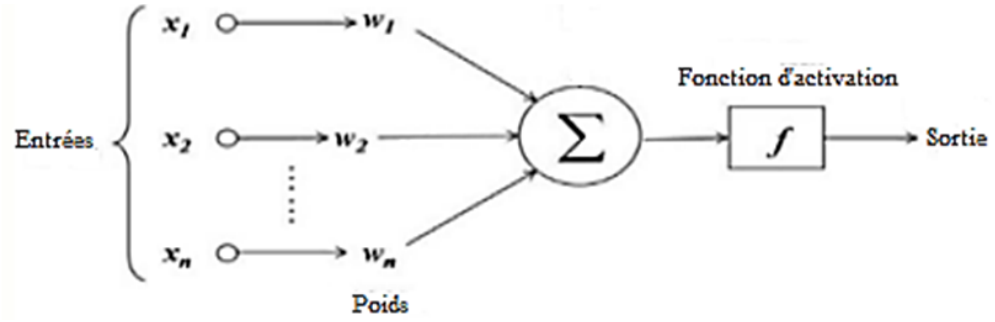


FIGURE 2.7 – Architecture d'un neurone simple

2)Perceptron Multicouche(PMC) : Ce modèle se compose d'une couche d'entrée, au moins une couche cachée en plus d'une couche de sortie, comme illustrée dans la figure 2.8 [2].

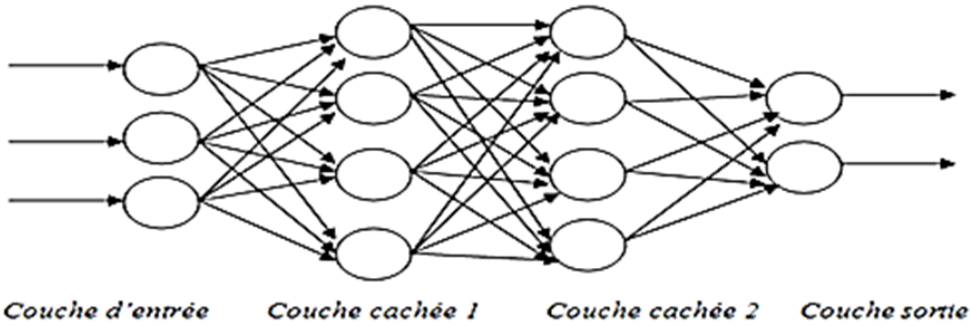


FIGURE 2.8 – Architecture de PMC

2.5 Apprentissage profond (Deep Learning)

L'apprentissage profond est un ensemble d'algorithmes d'apprentissage automatique, car il s'appuie sur des réseaux de neurones artificiels pour modéliser les processus d'abstraction de haut niveau dans les données. Les modèles d'apprentissage profond ont la capacité d'extraire des caractéristiques à partir des données brutes grâce à plusieurs couches de traitement constituées de multiples transformations linéaires et non linéaires. Ces caractéristiques sont reconnues étape par étape à travers chaque couche et avec un minimum d'effort humain. Alors, les modèles d'apprentissage profond peuvent donc atteindre une plus grande précision et parfois dépasser le niveau de performance humaine, car ces modèles permettent de faire des classifications directes sur les images, les textes ou les sons.

- **Apprentissage automatique Apprentissage profond**

L'apprentissage profond lié aux algorithmes qui peuvent apprendre plusieurs niveaux de représentation afin de modéliser les relations complexes entre les données. Bien que l'apprentissage automatique fonctionne avec des caractéristiques connexes qui sont généralement extraites manuellement à partir des entrées, puis ces caractéristiques sont utilisées pour créer un modèle qui effectue par exemple une tâche de classification. Donc, cette extraction manuelle des caractéristiques est à la fois difficile et coûteuse. Pour l'apprentissage profond les caractéristiques associées sont automatiquement extraites à partir des entrées, comme le montre la figure 2.9.

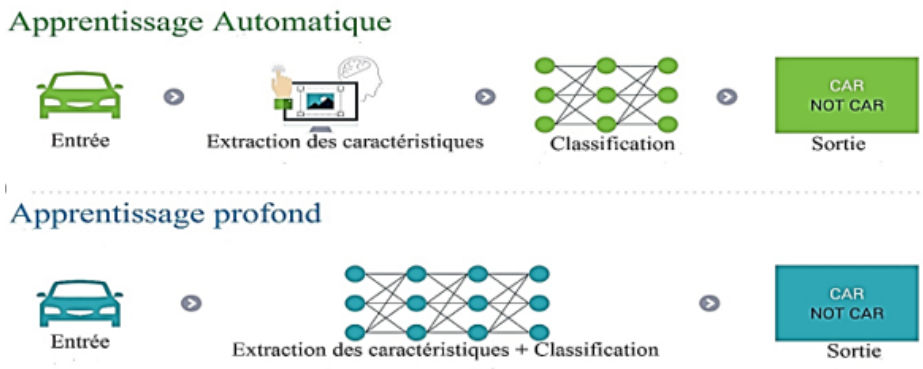


FIGURE 2.9 – Différence entre apprentissage automatique et apprentissage profond

- **Pourquoi l'apprentissage profond**

Les algorithmes d'apprentissage automatique permettent de résoudre divers problèmes, mais nous constatons parfois des lacunes, des erreurs et des inexactitudes pour résoudre certains problèmes, tels que la reconnaissance d'objets et la reconnaissance

vocale. Alors, l'apprentissage profond est venu résoudre ces problèmes. Le principal avantage des réseaux d'apprentissage profond est qu'ils continuent de s'améliorer à mesure que le volume de données augmente, contrairement aux algorithmes d'apprentissage automatique traditionnels, ils ne sont pas utiles lorsque vous travaillez avec des données de grande dimension, comme le montre la figure 2.10. Donc, l'apprentissage profond est très efficace dans ces situations.

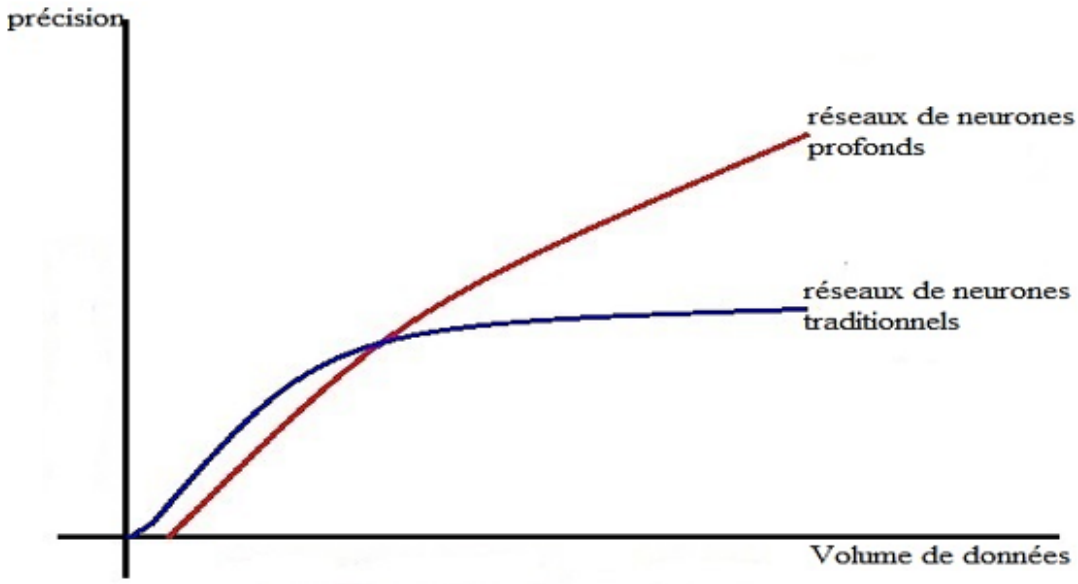


FIGURE 2.10 – Précision de l'apprentissage automatique et l'apprentissage profond en terme de volume de données

L'apprentissage profond dispose de plusieurs architectures. Dans la suite, nous nous focalisons sur les deux architectures les plus couramment utilisées.

2.6 Réseaux de neurones convolutifs

Le réseau de neurone convolutif (Convolutional Neural Networks ou CNN) est développé par LeCun en 1989. Le réseau CNN est un type de réseau feedforward qui permet d'extraire des caractéristiques, qui tire son travail du système visuel humain [3].

Les CNN sont largement utilisés dans les classifications des images, la détection des objets, etc.

2.6.1 Couches de CNN

Nous présentons les couches du CNN en dessous, lorsque ces couches sont empilées, nous obtenons la structure CNN comme le montre la figure 2.11 [4].

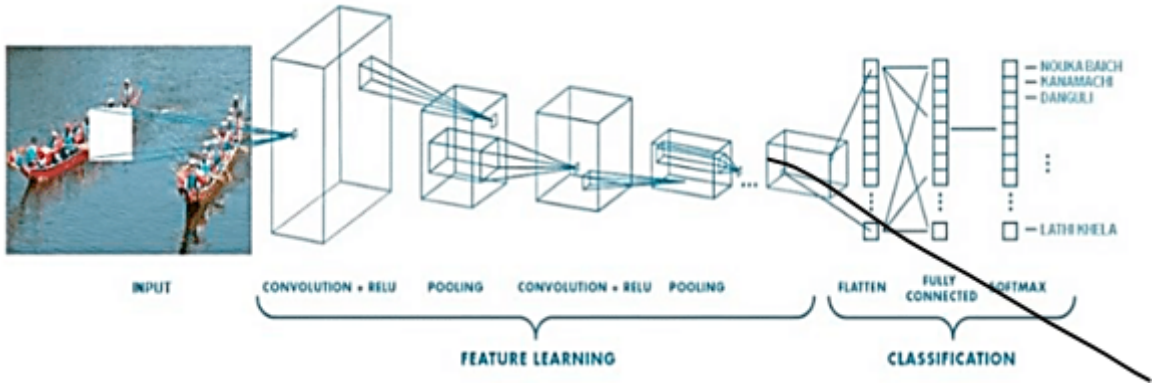


FIGURE 2.11 – Structure du réseau de neurone convolutif

1) Couche de convolution (convolution layer) : La couche la plus importante dans le réseau CNN, elle joue un rôle essentiel dans l'extraction des caractéristiques.

La figure 2.12 illustre l'opération de la convolution qui prend une image qui représente une matrice compose de pixels de 0 et 1, elle est une dimension de 7×7 pour appliquer un calcul à l'aide d'un noyau ou d'un filtre (3x3), afin de produire une matrice Moins de dimensions (5x5) ou ce qu'on appelle une carte des caractéristiques (Features Map).

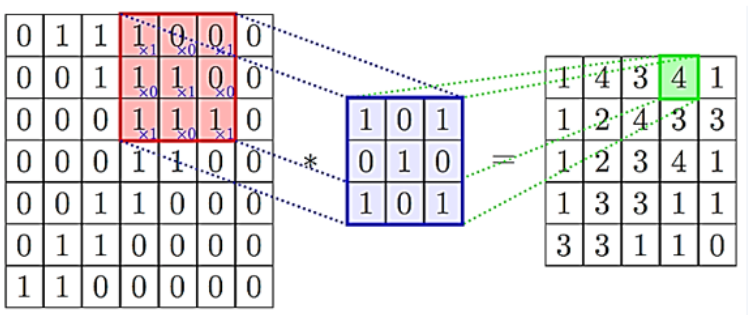


FIGURE 2.12 – Couche de convolution

2) Couche de correction (ReLU layer) : Pour améliorer l'efficacité du traitement, une fonction d'activation est représentée par la couche ReLU doit être incluse, dont le but est d'augmenter la non-linéarité de réseau CNN. Toutes les valeurs égales ou inférieures à zéro mettre en zéro, tandis que les valeurs positives restent les mêmes, comme illustré dans la figure 2.13.

3) Couche de mise en commun (Pooling layer) : C'est une couche qui réduit les dimensions de chaque carte tout en préservant les informations importantes.

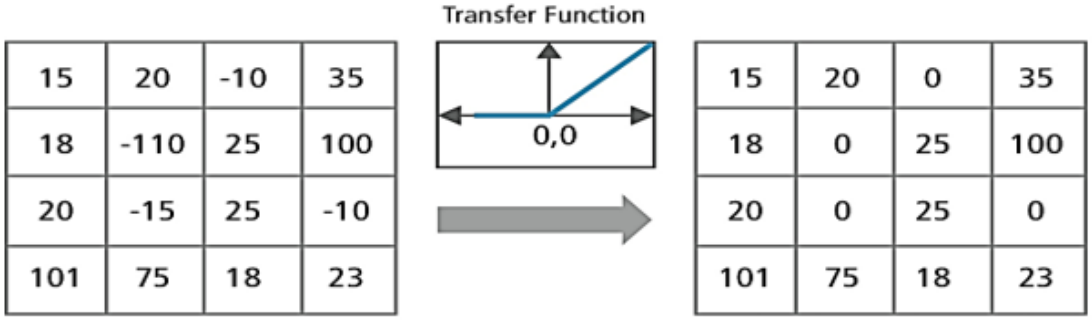


FIGURE 2.13 – Couche Relu

Il existe plusieurs types de pooling, le type le plus utilisé est le max pooling qui prend la grande valeur lors on applique généralement un filtre carré (2x2) à la couche ReLU, comme illustré dans la figure 2.14.

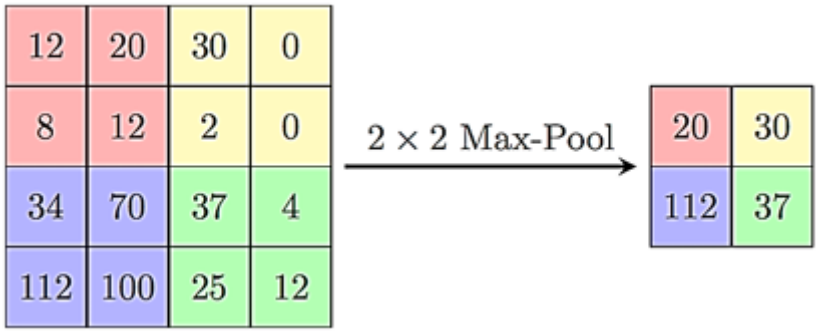


FIGURE 2.14 – Couche du pooling

4) Couche d'aplanissement (flattening layer) : Cette couche transforme tout ce qui est produit à partir de la couche du pooling en un tableau unidimensionnel, c'est-à-dire en les rendant plates, comme indique la figure 2.15.

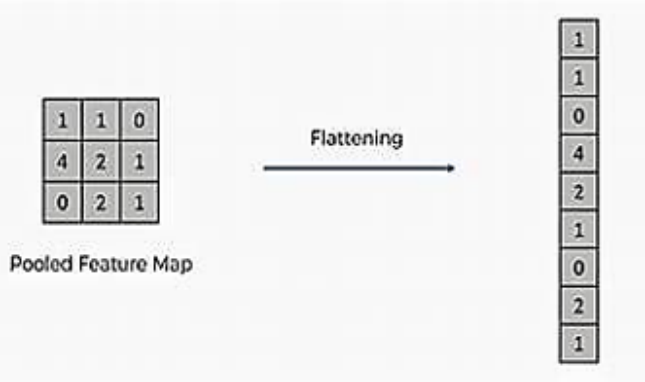


FIGURE 2.15 – Couche d'aplanissement

5) Couche entièrement connecté (fully connected layer) : Cette couche a pour but de réaliser un réseau de neurone à partir du vecteur obtenu par la couche d'aplanissement, comme montre la figure 2.16. Chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante. La fonction Softmax (ou sigmoid) est utilisé dans cette couche pour faire des classifications et obtenir des sorties.

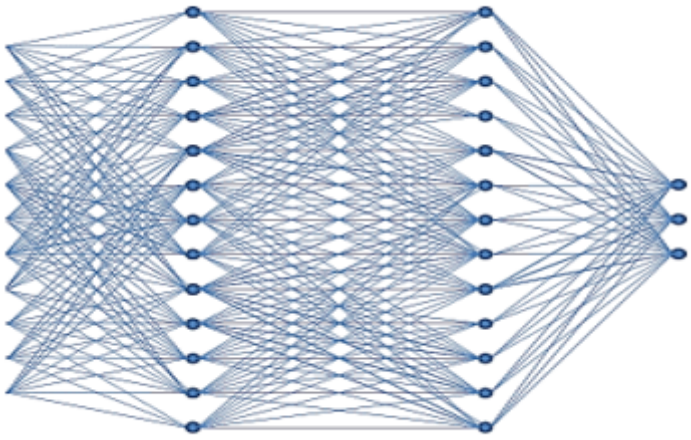


FIGURE 2.16 – Couche entièrement connecté

2.6.2 Différentes architectures de CNN

Les réseaux de neurones convolutifs ont plus que les architectures utilisées dans la littérature. À travers le projet ImageNet qui est une grande base de données visuelles, nous pouvons en apprendre davantage sur les différentes architectures des réseaux de neurones convolutifs.

ImageNet organise un concours annuel de logiciels, connu sous le nom de défi de la reconnaissance visuelle à grande échelle ImageNet (ILSVRC) pour les défis de classifi-

cation et de divulgation. Commençant par LeNet-5 du LeCun et d'autres en 1998, cette architecture était limitée. Puis en 2012, AlexNet a remporté le concours ImageNet en réduisant les erreurs de 26% à 15,3% car cette structure était plus profonde dans les couches convolutives empilées que LeNet.

En 2013, ZFNet a atteint le meilleur taux d'erreur de 14,8% en l'ajustant dans les paramètres de AlexNet. En 2014, une structure est apparue GoogleNet ou ce qu'on appelle aussi Inception V1 par les chercheurs de Google, où il a atteint un taux d'erreur de 5,67% où il s'approche de la performance humaine. Il est également apparu à partir de la même année VGGNet, mais c'est une architecture naïve à cause du problème de la disparition des gradients. Cependant les réseaux VGG sont utiles pour transférer l'apprentissage et les petites tâches de classification. Jusqu'à ILSVRC 2015, Kaiming he et autres fourni l'architecture ResNet ou par ce qui est connu comme le réseau neurone résiduel, cette structure a atteint un taux d'erreur de 3,57% qui surpasse les performances humaines dans cet ensemble de données [5].

Après ces architectures, les chercheurs ont souligné de meilleures structures, parmi lesquelles réseaux résiduels étendu, puis ResNext qui est une extension de la précédente. Il est également apparu une structure DenseNet et d'autres structures grâce à la compétition [6].

2.7 Réseaux de neurones récurrentes

Le réseau de neurone récurrente (Recurrent Neural Network ou RNN) est développé par Rumelhart et d'autres en 1986. C'est l'un des réseaux d'apprentissage profond associés aux modèles de séquence similaire à son travail dans les réseaux de neurones traditionnels, mais il se distingue d'eux en contenant des boucles et des itérations au sein du réseau grâce à l'utilisation des informations précédentes, qu'il stocke sous forme de mémoire pour les aider à prévoir et à estimer les valeurs suivantes [3].

Le RNN est largement utilisé dans de nombreuses applications telles que la reconnaissance vocale, la composition musicale, la traduction automatique, la reconnaissance de l'écriture manuscrite, l'apprentissage de la grammaire, les cotations boursières, voitures d'auto-conduisez, etc.

2.7.1 Fonctionnement de RNN

Le RNN s'appuie dans son travail sur 3 couches, la première est la couche d'entrée qui contient N unités d'entrées représentent une série de vecteurs à travers le

temps tel que $t : \{\ldots, x_{t-1}, x_t, x_{t+1}, \ldots\}$ où $x_t = (x_{t_1}, x_{t_2}, \ldots, x_{t_N})$. Les unités d'entrées sont connectées aux unités cachées dans la deuxième couche (la couche cachée répétée) par une matrice de poids w_{IH}. La couche cachée se compose de M unités cachées $h_t = (h_{t_1}, h_{t_2}, \ldots, h_{t_M})$, connectés les uns aux autres en forme répétitive (voir la figure 5.2), cette couche détermine la mémoire système comme suit [7] :

$$h_t = f_H(o_t) \qquad (2.8)$$

où :

$$o_t = w_{IH}x_t + w_{HH}h_{t-1} + b_h \qquad (2.9)$$

$f_H(.)$: Fonction d'activation de la couche cachée ;

b_h : Vecteur de biais des unités cachées.

Les unités cachées sont connectées à la troisième couche (la couche de sortie) en fonction du poids w_{HO}. La couche de sortie comprend P unités $y_t = (y_{t_1}, y_{t_2}, \ldots, y_{t_p})$, il est calculé comme suit :

$$y_t = f_O(w_{HO}h_t + b_O) \qquad (2.10)$$

$f_O(.)$: Fonction d'activation de la couche de sortie.

b_O : Vecteur de biais dans la couche de sortie.

2.7.2 Types de RNN

Étant donné que RNN ne se limite pas au traitement des entrées de taille fixe, mais a plutôt la capacité de calculer différentes longueurs de séquences, donc RNN est classé à un certain nombre de types en termes de nombre d'entrées et de nombre de sorties, comme le montre la figure 5.3 [8].

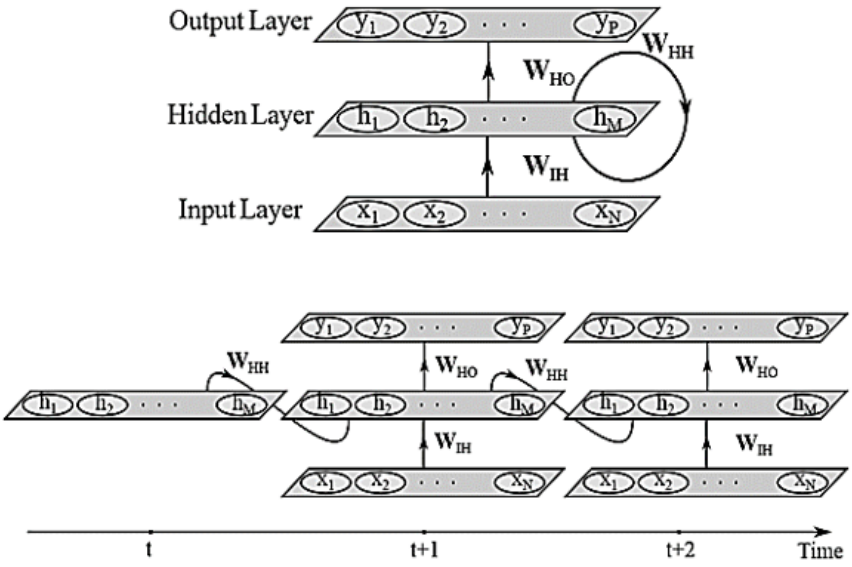


FIGURE 2.17 – Structure du réseau de neurone récurrent[7]

Plusieurs-à-un : Il a de nombreuses entrées (une séquence) qui se traduisent par une sortie, telle qu'une évaluation des sentiments du texte.

Un-à-plusieurs : Dépend d'une seule entrée pour créer une séquence (plusieurs sorties) comme la création d'une appellation explicatif à partir d'une image.

Plusieurs-à-Plusieurs (direct) : Il a de nombreuses entrées pour produire de nombreuses sorties. Alors que le nombre d'entrées et de sorties est souvent égal dans ce type, comme exemple la reconnaissance vocale.

Plusieurs-à-Plusieurs (indirect) : Dans ce type, le nombre d'entrées n'est pas égal au nombre de sorties, de sorte que le traitement est en deux parties, la première partie dans laquelle toutes les entrées sont traitées en une seule fois (encodeur) et dans la deuxième partie les sorties sont produites (décodeur), comme la traduction.

Un-à-un : Ce type représente un traitement simple sans séquencement, comme les réseaux RNA.

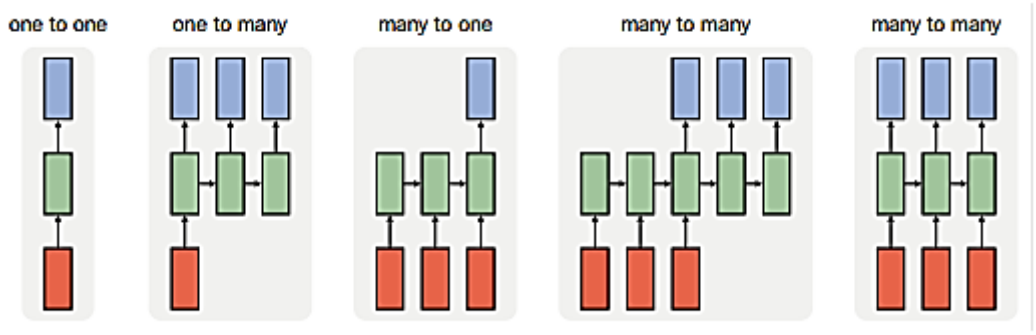


FIGURE 2.18 – Différents types du réseau RNN[8]

2.7.3 Rétropropagation dans le temps

L'algorithme BPTT (Back-Propagation Through Time en anglais) est une extension de la propagation inverse standard qui mis à jour les pondérations en s'appuyant sur l'équation d'erreur pour formuler le réseau à partir de l'arrière vers l'avant, sur la base de sa réduction, comme illustré dans la figure 5.4.

Le BPTT est utilisé pour entraîner le RNN, il détecte le réseau à un certain moment pour propager les erreurs dans le temps en raison des paramètres du réseau RNN : $\theta = \{w_{HH}, w_{IH}, w_{HO}, b_h, b_o\}$ et h_t est l'état caché du réseau au temps t, les gradients sont les suivants [7] :

$$\frac{dL}{d\theta} = \sum_{t=1}^{T} \frac{dL_t}{d\theta} \qquad (2.11)$$

où :

$$\frac{dL_t}{d\theta} = \sum_{k=1}^{t} (\frac{dL_t}{dh_t} . \frac{dh_t}{dh_k} . \frac{dh_k^+}{d\theta}) \qquad (2.12)$$

L'équation (2.14) représente l'expansion du gradient de la fonction de perte dans le temps t.

Où $\frac{dh_k^+}{d\theta}$ est la dérivée partielle qui représente l'étendue de l'influence des paramètres de l'ensemble θ sur la fonction de perte à partir du pas de temps t au pas de temps k. Pour transférer l'erreur dans le temps comme suit :

$$\frac{dh_t}{dh_k} = \prod_{i=k+1}^{t} \frac{dh_i}{dh_{i-1}} \qquad (2.13)$$

Selon les entrées et l'activation de l'unité cachée dans les pas de temps précédents on a :

$$\prod_{i=k+1}^{t} \frac{dh_i}{dh_{i-1}} = \prod_{i=k+1}^{t} w_{HH}^{T} diag\,|f'_H(h_{i-1})| \qquad (2.14)$$

où :

$f'(.)$: Dérivé du $f(.)$.

$Diag(.)$: Matrice diagonale.

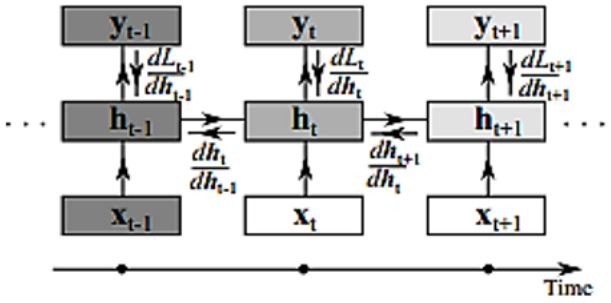


FIGURE 2.19 – Rétropropagation dans le temps[7]

2.7.4 RNN bidirectionnel

Le RNN traditionnel ne regarde que le contexte précédent lors de l'entraînement des données, alors que cela n'est pas suffisant (par exemple dans la reconnaissance vocale) mais plutôt utile pour regarder le contexte futur également [1]. C'est pourquoi en 1997, Schuster et Paliwal ont découvert une nouvelle structure appelée BRNN (Bidirectional RNN en anglais)[3].

Le BRNN examine les deux directions du traitement des données avec deux couches cachées distinctes (voir la figure 5.5) pour estimer la couche de sortie [1].

La séquence cachée vers l'avant du début à la fin dans une direction temporelle positive :

$$\overrightarrow{h_t} = f_H(w_{\overrightarrow{IH}} x_t + w_{\overrightarrow{HH}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \qquad (2.15)$$

avec $t = 1, \ldots, T$

La séquence cachée vers l'arrière de la fin au début dans une direction temporelle négative :

$$\overleftarrow{h_t} = f_H(w_{\overleftarrow{IH}} x_t + w_{\overleftarrow{HH}} \overleftarrow{h}_{t-1} + b_{\overleftarrow{h}}) \qquad (2.16)$$

avec $t = 1, \ldots, T$

Mettre à jour la couche de sortie y_t :

$$y_t = w_{\overrightarrow{HO}} \overrightarrow{h_t} + w_{\overleftarrow{HO}} \overleftarrow{h_t} + b_o \qquad (2.17)$$

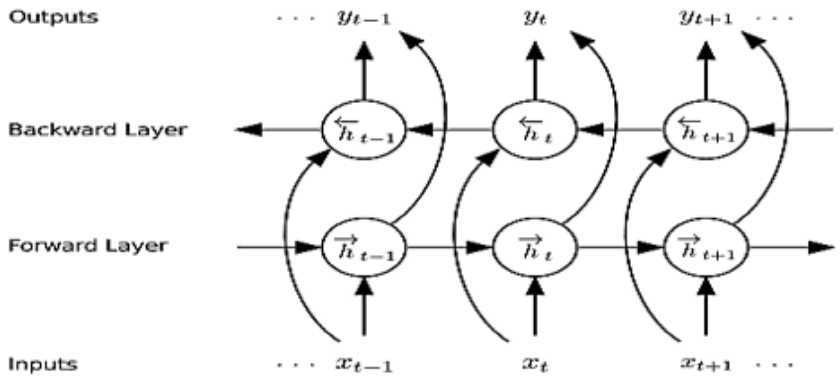


FIGURE 2.20 – RNN bidirectionnel[1]

Le réseau BRNN est entrainé par BPTT, et bien que BRNN prédit une direction à la fois négative et positive en même temps, il présente des défauts en fournissant des connaissances de séquence du début à la fin en avance, et en augmentant la complexité du calcul [7].

2.7.5 Limites de RNN

Le RNN peuvent être constitué d'un grand nombre de couches profondes, ce qui cause des problèmes et devient inexpérimenté. Nous abordons les deux principaux problèmes confrontés par RNN [8].

1)Problème de gradient explosif : C'est un gonflage des gradients lors de l'entrainement du RNN en raison de la détonation de composants à long terme qui conduit à l'instabilité du réseau.

2) Problème de disparition des gradients : C'est un problème plus difficile à détecter et à résoudre que son prédécesseur, où les gradients se dégradent en très petites valeurs de leur propagation dans le temps (entrainement RNN par BPTT). La conversion rapide à zéro des composants à long terme et l'incapacité du modèle à stocker ces composants rendent le modèle incapable d'apprendre.

Pour résoudre ces problèmes, nous abordons dans la suite les deux solutions les plus courantes.

2.7.6 Mémoire à long et court terme

La solution de mémoire à long et court terme (Long Short Terme Memory ou LSTM) a été introduit par Schmidhuber et Hochreiter en 1997. C'est une architecture utilisée dans les réseaux de neurones récurrent, elle forme la modélisation de dépendances séquentielles à long terme. L'unité LSTM se compose de 3 portes : la porte d'oubli, l'entrée (la porte de mise à jour des entrées) et la porte de sortie, comme le montre la figure 5.6 [9].

Ces portails utilisent une fonction d'activation segmoid, le résultat d'activation (0) signifie que la masse cellulaire ne produit aucune information, et le résultat d'activation (1) indique la masse cellulaire exploite l'information, elle la stocke [8].

La porte de l'oubli dépend de la mémorisation de l'état de la cellule ou de son oubli, sa formule comme suit :

$$f_t = \sigma(w_f h_{t-1} + U_f x_t + b_f) \qquad (2.18)$$

Le portail d'entrée est principalement destiné à préserver les informations impor-

tantes et arrête ainsi tout composant du vecteur d'entrée inutile :

$$i_t = \sigma(w_i h_{t-1} + U_i x_t + b_i) \qquad (2.19)$$

Les entrées à partir des sorties précédentes h_{t-1} et les entrées actuelles x_t sont ajoutées et écrasées via une fonction tanh :

$$a_t = \tanh(w_c h_{t-1} + U_c x_t + b_c) \qquad (2.20)$$

À travers les deux portes précédentes, une mémoire de répétition est créée pour LSTM :

$$C_t = f_t * C_{t-1} + i_t * a_t \qquad (2.21)$$

Le portail de sortie à travers laquelle les informations pouvant être éjectées du vecteur d'état de cellule où :

$$O_t = \sigma(w_o h_{t-1} + U_o x_t + b_o) \qquad (2.22)$$

Enfin, l'écrasement avec la fonction tanh pour obtenir la valeur finale :

$$h_t = O_t * \tanh(C_t) \qquad (2.23)$$

LSTM a plusieurs architectures qui sont présentées dans la table 2.2, la structure qui est devenue accessible dans de nombreuses applications est LSTM bidirectionnelle (comme les applications de détection d'activité vocale).

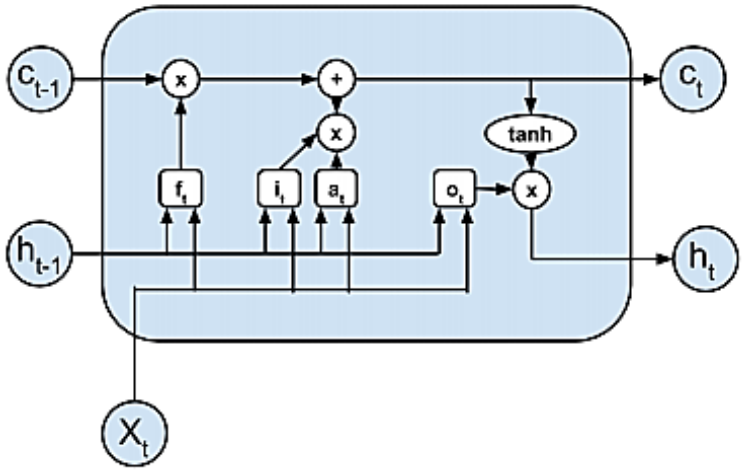


FIGURE 2.21 – Structure de la cellule LSTM[8]

TABLE 2.2 – Différents architectures de LSTM[7]

LSTM	LSTM Bidirectionnel est plus puissant et efficace que le LSTM ordinaire. En augmentant l'amplitude du BRNN en empilant des couches cachées dans l'espace des cellules LSTM, BLSTM est formé, il présente les mêmes avantages que BRNN.
S-LSTM	
LSTM empilé	
LSTM Bidirectionnel	
LSTM multidimensionnel	
Grille LSTM	
Différentiel RNN	
LSTM local-global	
LSTM correspondant	
Fréquence-Temps LSTM	

Puisque LSTM a résolu des problèmes RNN, il améliore également les prévisions, leur efficacité est grande et leur précision. Cependant, il a un ensemble de points négatifs dans la multiplication des cellules de mémoire, ce qui entraîne plus de besoin de mémoire que RNN, ce qui prend du temps et la complexité dans le calcul[7].

2.7.7 Unité Récurrente Fermée

Depuis LSTM est une solution aux problèmes du RNN, mais il augmente les besoins en mémoire en plus d'autres inconvénients (comme mentionné précédemment) qui a conduit à la suggestion d'une nouvelle structure nommée GRU (Gated Recurrent Unit)[7].

GRU est proposée en 2014, c'est par LSTM que GRU dispose également de portails, le premier est le portail de mise à jour qui combine les portails d'oubli et d'entrée de LSTM. L'activation de ce portail est liée aux unités avec des dépendances à long terme. La deuxième porte, cela de réinitialisation qui permet à l'unité d'oublier le passé si elle est fermée ($r_t = 0$) signifie que l'unité est capable de lire uniquement le premier code de la chaîne d'entrée, l'activation de ce portail est liée aux unités avec des dépendances à court terme(voir la figure 5.7)[10]

Portail de mise à jour :

$$z_t = \sigma(w_z x_t + U_z h_{t-1}) \qquad (2.24)$$

Porte de réinitialisation :

$$r_t = \sigma(w_r x_t + U_r h_{t-1}) \qquad (2.25)$$

w et U : Matrices de poids.

L'activation du filtre (similaire à Eq (1.20)) est calculée comme suit :

$$\tilde{h_t} = \tanh(w_h x_t + U_h(r_t * h_{t-1})) \qquad (2.26)$$

La valeur finale de GRU (achèvement linéaire) :

$$h_t = (1 - z_t)h_{t-1} + z_t\tilde{h_t} \qquad (2.27)$$

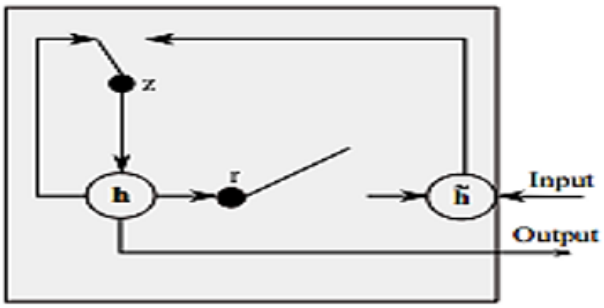


FIGURE 2.22 – Unité récurrente fermée[7]

Étant donné que GRU réduit les besoins en mémoire par rapport à LSTM, il résout également les problèmes RNN, mais aussi avoir des inconvénients, dont la multiplicité des états cachés qui compliquent le calcul et les besoins en mémoire plus que RNN [7].

2.7.8 Deep RNN

La multiplicité des couches cachées RNN et leur empilement les unes sur les autres forme le Deep RNN. La séquence d'entrée de pièce actuelle est produite par la séquence de sortie de couche unique précédente et ainsi de suite. Alors, les séquences vectorielles cachées h^n sont calculées de $n = 1$ à N couches où $t = 1, \ldots, T$ en forme répétitif comme suit [1] :

$$h_t^n = H(w_{h^{n-1}h^n} h_t^{n-1} + w_{h^n h^n} h_{t-1}^n + b_h^n) \qquad (2.28)$$

où :

$h^0 = x$, les sorties y_t sont :

$$y_t = w_{h^N y} h_t^N + b_y \qquad (2.29)$$

Parmi les avantages du Deep RNN, dans la séquence d'entrée le Deep RNN Distingué entre toutes les différences, et aussi le réseau peut s'adapter rapidement aux nœuds d'entrée changeants, donc le réseau est témoin d'une évolution à l'état caché. Cependant, à cause de la profondeur du réseau, elle devient plus vulnérable à la disparition des gradients, en plus de la complexité de calcul[7].

2.8 conclusion

Dans ce chapitre, nous avons présentés les principaux concepts de bases sur les réseaux de neurones artificiels, puis nous avons étudiés l'apprentissage profond et nous nous sommes focalisés principalement sur les structures CNN et RNN.

Étant donné que ce chapitre comprend divers concepts d'apprentissage profond, cela ne suffit pas pour terminer nos recherches, car nous devons également clarifier des notions de base liés à notre recherche, et c'est ce que nous aborderons dans le chapitre suivant.

Chapitre 3

Notions de base sur la détection d'activité vocale

3.1 introduction

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

Pour atteindre notre objectif, dans ce chapitre nous abordons brièvement quelques concepts de base, qui sont : La parole, bruit, rapport signal sur bruit et traitement numérique des signaux. A partir de là, nous allons passer à une étude sur la DAV, en plus d'énoncer le principe applicable.

3.2 Signal de la parole

•**Parole :** C'est la capacité d'exprimer et de percevoir des sons pour permettre aux êtres humains de communiquer entre eux [11].

•**Son :** Physiquement, c'est la propagation des changements de pression dans un milieu (gazeux, liquide, solide), formant des ondes, il est donc considéré comme une vibration mécanique. Ainsi, le son est un signal perçu par le sens de l'ouïe. Donc, le son peut être caractérisé par [12] :

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

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

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

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

3.2.1 Production du signal de la parole

La personne utilise ses systèmes respiratoire et digestif pour produire le signal de la parole, où les sons sont produits par l'air expiré des poumons. Après que l'air dans les poumons se soit déplacé de la trachée à travers le larynx, où il touche les cordes vocales, qui en elles-mêmes effectuent le processus d'ouverture et de fermeture de la glotte aux

voies vocales s'étendant du pharynx aux lèvres, un signal acoustique est émis,comme illustré par la figure 3.1 [13].

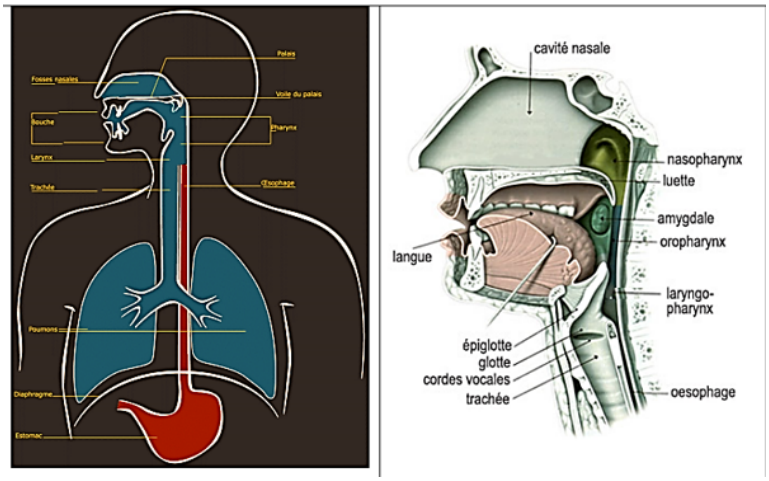


FIGURE 3.1 – Appareil phonatoire[13]

Ce signal acoustique distinctif avec une énergie non stationnaire limitée et sa structure complexe atteignent l'oreille (Figure 3.2) et exactement à l'oreille interne, où se trouve le nerf auditif, qui à son tour transmet le message parlé au cerveau pour que ce dernier l'interprète [11].

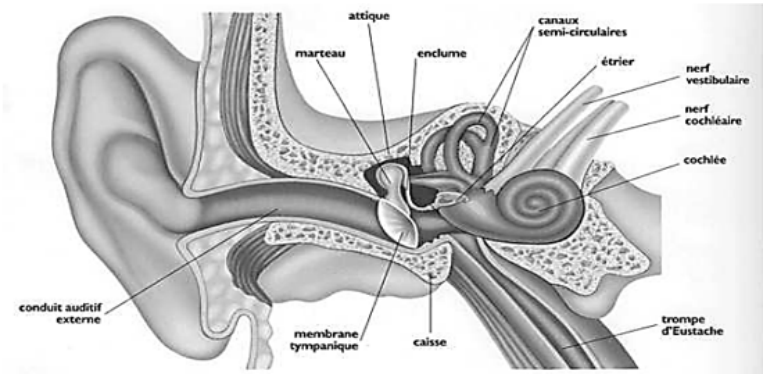


FIGURE 3.2 – Appareil auditif humain[11]

3.2.2 Caractéristiques du signal de la parole

Comme nous l'avons déjà dit à propos du signal vocal, il a une structure complexe, où il est parfois périodique et autre aléatoire, de sorte que les sons de la parole sont divisés en deux catégories, qui sont les suivantes [14] :

•**Sons voisés :** Ce sont des signaux semi-périodiques, résultant de vibrations périodiques des cordes vocales et également dus au conduit vocal et à sa configuration

semi-stable , comme le montre la figure 3.3.

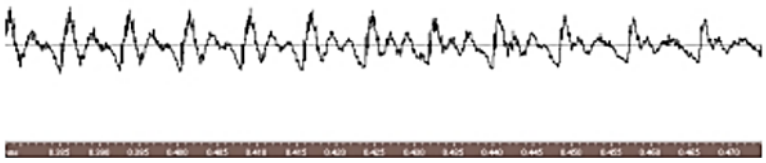


FIGURE 3.3 – Exemple de son voisé[14]

•**Sons non voisés :** Les cordes vocales ne vibrent pas car elles sont dans une position écartée. Donc, ces signaux ne sont pas de structure périodique, comme le montre la figure 3.4.

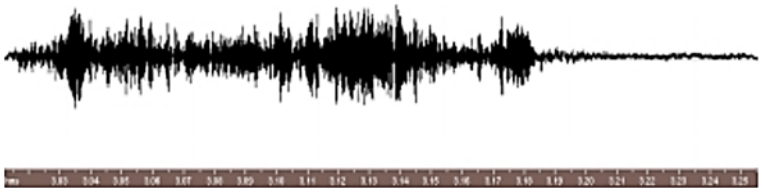


FIGURE 3.4 – Exemple de sons non voisé[14]

3.2.3 Paramètres du signal de la parole

De manière générale, le signal vocal est caractérisé par trois paramètres, qui sont les suivants [15] :

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

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

$$E = \sum_{n=0}^{N-1} s^2(n) \qquad (3.1)$$

$s(n)$: Segment analysé.

N : Taille de la trame.

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

Pour mettre en œuvre les méthodes d'analyse et de modélisation efficaces utilisées pour le traitement à court terme du signal de la parole, des fenêtres qui sont un ensemble des trames (une trame d'une durée comprise entre 20 et 30 ms) sont utilisées, car le recouvrement entre ces fenetres constitue une garantie de la continuité temporelle des caractéristiques d'analyse. La densité spectrale de puissance de la transformée de Fourier donnée par :

$$\left|\hat{S}(k)\right|^2, 0 \leq k \leq \frac{N}{2} \qquad (3.2)$$

Où :

$$\hat{S}(K) = \hat{S}(f = \frac{N}{K}) = \sum_{n=0}^{N-1} s(n).w(n)\mathrm{e}^{(}\frac{-j2\pi nk}{N}), 0 \leq k \leq N-1 \qquad (3.3)$$

$\hat{S}(k)$: Spectre complexe.

$w(n)$: Fenêtre de temps.

L'équation(3.3) représente une transformation du signal pondérée (transformée de Fourier à court terme (TFCT) pour un signal échantillonné).

3.2.4 Quelques propriétés du signal de la parole

Tout signal de la parole peut être décrit par les propriétés suivantes [16] :

-Quasi-stationnaire.

-Périodique.

-La bande passante de tout signal est au moins 4 KHz, car cette valeur en elle-même est suffisante pour comprendre la voix humaine.

-La valeur de la fréquence fondamentale est comprise entre les deux valeurs 80 et 350 Hz.

3.3 Bruit

3.3.1 Définition

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

3.3.2 Types de bruit [17]

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

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

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

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

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

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

3.3.3 Rapport signal sur bruit (RSB)

Le RSB mesure la quantité de bruit dans le signal utile. La qualité de transmission est qualifiée en raison du quotient de division de la puissance du signal utile P_S par la puissance du signal de bruit P_N [18].

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

$$RSB = 10\log_{10}(\frac{P_S}{P_N}) \qquad (3.4)$$

3.4 De l'analogique au numérique

3.4.1 Classification du son

Le son est classé selon deux groupes principaux [19] :

• **Son analogique (Un signal continu) :** Semblable à l'onde sonore produite par le son analogique, cette onde sonore est enregistrée. En fonction du changement de pression de l'air réside la valeur de la tension sonore analogique, où au moyen d'un microphone cette pression est convertie en tension, qui est capturée.

• **Son numérique (Un signal discontinu) :** Il représente des séquences binaires (0 et 1), car la pression acoustique captée est convertie en une tension mesurable. Donc, cette grandeur analogique continue est représentée par une courbe variant en fonction du temps.

3.4.2 Numérisation du son

Pour convertir le signal analogique en signal numérique, les trois opérations de base suivantes sont appliquées :

a) Echantillonnage : Ce processus dépend de la division du temps, car lorsque le signal analogique entre dans l'ordinateur, il est mesuré plusieurs fois par seconde. À partir de là, le son est coupé en bandes ou ce que l'on appelle des échantillons, comme illustré par la figure 3.5. La fréquence d'échantillonnage représente le nombre d'échantillons disponibles par seconde d'audio. Plus la fréquence d'échantillonnage est élevée,

plus la traduction numérique du signal est proche de l'original analogique [20].

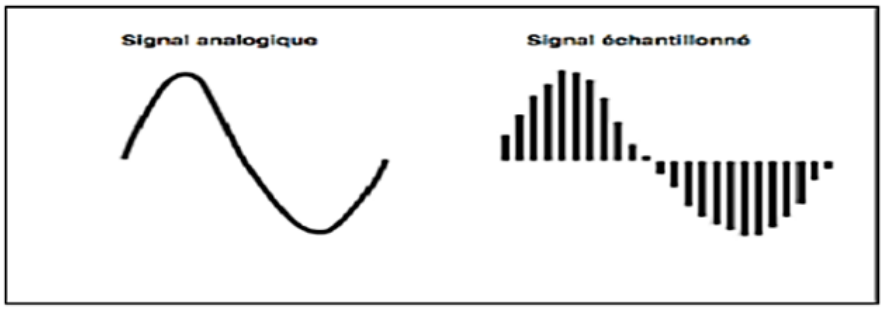


FIGURE 3.5 – Échantillonnage d'un signal audio[21]

b) Quantification : Ce processus est exprimé en bits, où 16 bits et 24 bits font partie des valeurs couramment utilisées en audio. La quantification permet d'attribuer une valeur d'amplitude à chaque échantillon en créant une échelle pour les valeurs discrètes, comme le montre dans la figure 3.6.

Inévitablement, l'une des valeurs spécifiées par l'échelle de quantification doit être attribuée à l'amplitude de chaque échantillon. La valeur d'amplitude de l'échantillon est approximée au palier le plus proche, si elle se situe entre deux paliers de l'échelle de quantification, cette approximation conduit à l'apparition d'une erreur de quantification.

Plus la résolution de quantification est élevée, les petites différences d'amplitude du signal échantillonné sont approximées. À partir de là, plus le nombre de bits est élevé, plus le nombre de paliers est important et l'erreur de quantification est réduite.

Ainsi, en fonction de la résolution (en bits) et de la fréquence d'échantillonnage se trouve la précision de la forme d'onde numérique de la forme d'onde du signal analogique [21].

c) Codage : Les valeurs quantitatives sont représentées par des séquences binaires de 0 et 1, ce qui rend le signal numérique bien exploité par l'ordinateur [21].

3.5 Fichier audio numérique

3.5.1 Format de fichier audio numérique

Représente le format des données, au moyen duquel les sons sont stockés sous forme numérique dans des ordinateurs. Il existe de nombreux types de formats [22].

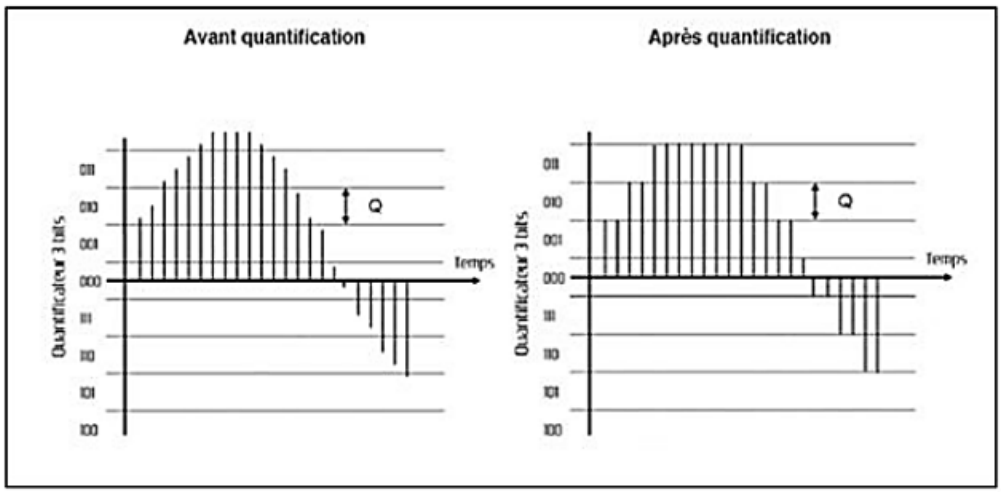


FIGURE 3.6 – Signal échantillonné avant et après quantification[21]

3.5.2 Types de formats de fichier audio numérique

1) Les formats audios compressés avec perte

a) MP3 : C'est une abréviation pour MPEG 1/2 Audio Layer 3, créé par l'Institut Fraunhofer en 1993, ce format faire une compression d'une séquence audio dans un très petit fichier, car il filtre le fichier de toutes les données inaudibles, donc il est difficile pour l'oreille humaine d'entendre perte de qualité. Son extension **.mp3** [20][23].

b) WMA (Windows Media Audio) : Il a été créé en 1999 par Microsoft grâce aux recommandations MPEG4, puisque ce fichier est de petite taille par rapport au fichier MP3, il est lu par le programme windows player media, son extension **.wma** [22][23].

2) Les formats audios compressés sans perte

a) ATRAC (Adaptive Transform Acoustic Coding) : Il a été développé par Sony en 1992, il a ensuite subi diverses modifications car ce format était classé dans les techniques de compression sonore avec perte et sans perte de données [22].

b) FLAC (Free Lossless Audio Codec) : Un format de codec audio open source,

développé en 2000 par Josh Coalson, tel qu'il a été incorporé sous les logan Xiph.org dans l'année 2003, ce format préserve la qualité et produit des fichiers de plus petite taille [24].

3) Les formats audios sans compression

a) WAV (Wave form : forme d'onde) : Il a été créé par Microsoft, d'où ce format a été dérivé de la spécification du format de fichier d'échange de ressources. La taille de ce fichier est basée sur l'utilisation d'un codeur modifié, un code d'impulsion qui représente pour le traitement numérique du signal, où un audio d'une minute peut prendre environ 10 MO, son extension **.wav** [20][23].

b) CDA (Compact Disc Audio) : Un format pour Windows créé par Microsoft, qui affiche les pistes du CD audio lorsqu'elles sont insérées dans le lecteur de CD, son extension **.cda** [22].

3.6 Détection d'activité vocale(DAV)

3.6.1 Définition

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

Cette technique distingue des zones parole et des zones non parole. Le codage de la parole et la reconnaissance vocale font partie des principales applications liées à DAV [25].

3.6.2 Principe et Fonctionnement

Comme nous avons vus avant, DAV fait la distinction entre deux zones du signe de la parole. Idéalement, DAV produit "1" s'il y a de voix (zone active), ou "0" s'il n'y a pas de voix (zone inactive), voir la figure 3.7 [26].

Le signal audio d'entrée est découpé en trame, par la plupart méthodes DAV. La

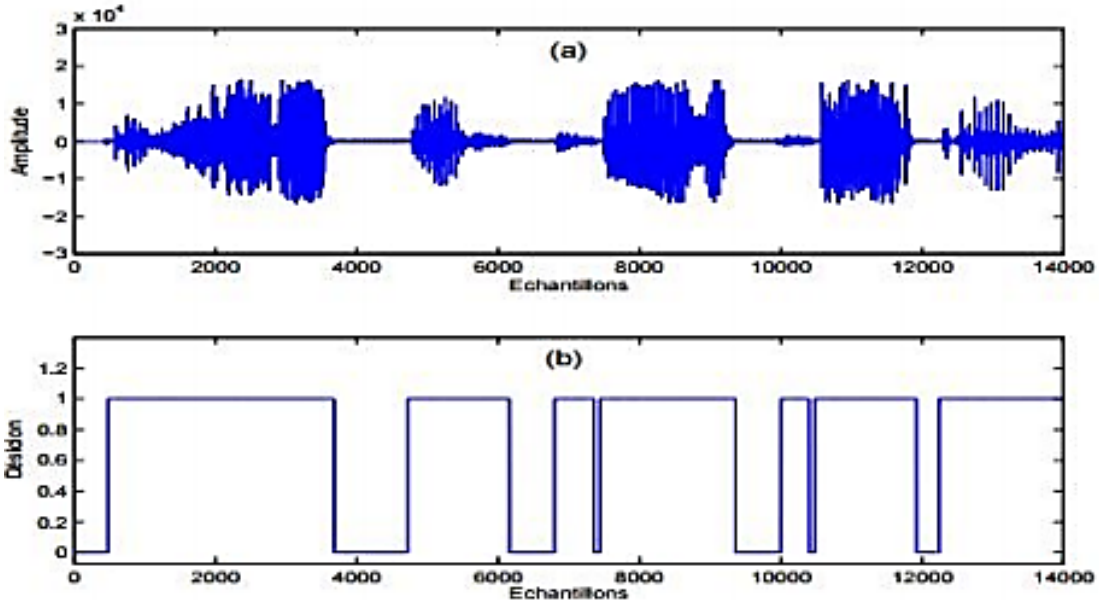


FIGURE 3.7 – Exemple illustrant le principe de la DAV [26]

trame fait partie d'un signal de longueur fixe qui est ordonnée en quelques millisecondes, c'est-à-dire qu'elle peut être de 10 à 50 ms. Par conséquent, la DAV décide la trame qui contient de la parole ou qui ne contient pas de parole, alors il est possible de découper les trames avec ou sans chevauchement. La figure 3.8 clarifie les types de découpages sur un signal audio, où chaque découpage représente un pourcentage du chevauchement [27].

Généralement, chaque algorithme DAV suit le processus général qui décrite dans la figure 3.9 pour fournir une décision 0 ou 1 à partir d'un signal audio.

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

1) Domaine temporelle : Ce domaine comprend de nombreuses techniques d'analyse, notamment :
a) Taux de passage à zéro(ZCR : Zero Crossing Rate) : Dans un intervalle de temps ou un terme donné, le ZCR mesure le nombre de fois des changements de l'amplitude du signal. Alors, si les échantillons successifs ont des signes algébriques différents, un passage à zéro se produit selon le contexte des signaux à temps discret.[28]

Le ZCR s'agit une simple mesure du bruit pour les signaux complexes. Aussi, il est utilisé pour faire une estimation approximative de la fréquence fondamentale pour les signaux à une seule voix [29].

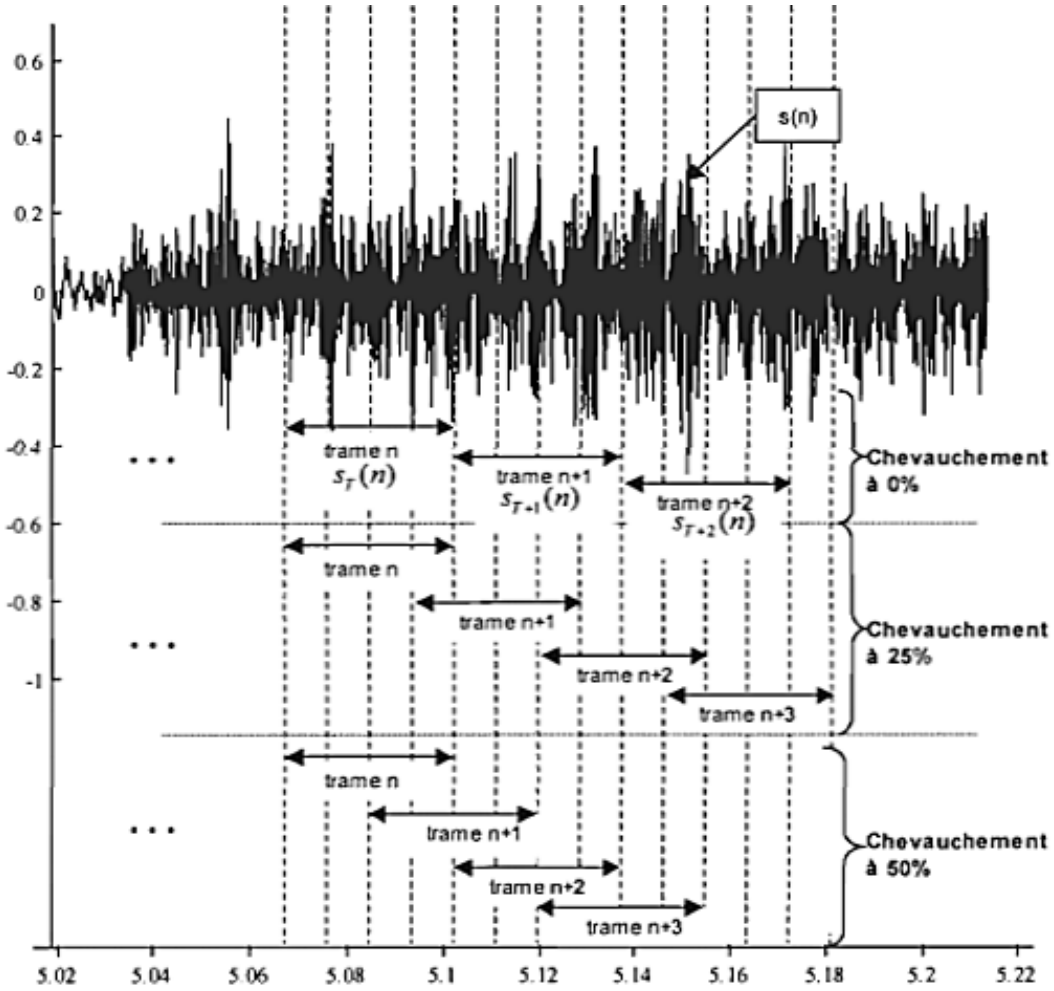


FIGURE 3.8 – Découpage du signal audio à trois taux de chevauchement différents [27]

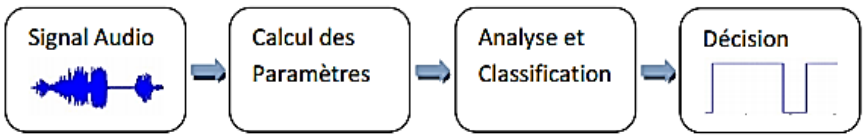


FIGURE 3.9 – Processus général d'un algorithme de la DAV [26]

b) Énergie à court terme : Représente l'enveloppe temporelle du signal, car elle exprime la valeur quadratique moyenne des valeurs de forme d'onde dans la trame de données.Grâce à la variation que cette énergie crée au fil du temps, cette variation peut devenir un indicateur fort du contenu du signal sous-jacent [30].

2) Domaine spectrale : Il existe de nombreuses techniques d'analyse dans ce domaine, mais nous fournirons une description de certaines des techniques suivantes :

a) Transformée de Fourier discrète(TFD) : La TFD est une technique d'analyse, il est définit comme suite [31] :

$$\hat{S}(K) = f\{s[n]\} = \sum_{n=0}^{N-1} s[n]\mathrm{e}^{(\frac{-j2\pi nk}{N})} \qquad (3.5)$$

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

K : Variable discrète de fréquence.

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

b) Ondelettes : C'est l'un des outils d'analyse du signal, l'analyse par ondelette est apparue au début des années 80. Cette analyse offre une large gamme de fonctions de base parmi lesquelles on peut choisir la plus appropriée pour une application donnée. La transformée en ondelettes offre la possibilité d'analyser un signal simultanément dans le domaine du temps et celui des fréquences, où il a été nommé «la technique d'analyse temps-fréquence» [32].

c) MFCC(Mel Frequency Cepstral Coefficients) : C'est une technique la plus couramment utilisée pour extraire des caractéristiques, car elle prend le domaine fréquentiel comme une base principale et fonctionne avec une approximation plus proche de l'audition humaine [33].

Le processus de cette technique illustre dans la figure 3.10.

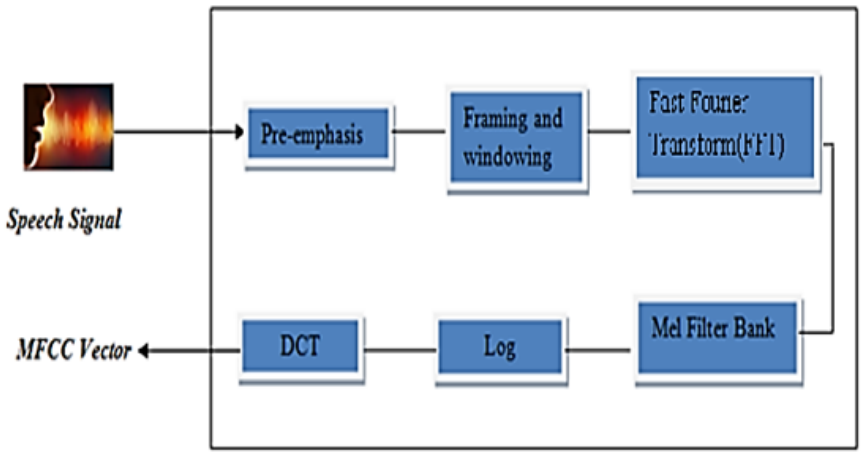


FIGURE 3.10 – Schéma fonctionnel pour l'extraction de caractéristiques MFCC [34]

les étapes pour produire les caractéristiques MFCC sont décrites dans l'ordre suivant [34] :

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

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

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

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

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

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

d) LPC(Linear Predictive Coding) : Le codage prédictif linéaire est une technique d'analyse de la parole utilisée pour réduire la somme des différences quadratiques entre le signal de parole d'origine et le signal de parole estimé sur une période de temps limitée. Il est considéré comme une technique statique pouvant produire des paramètres d'ordre inférieur. LPC est utile pour encoder la qualité de parole avec un taux de bit faible [33] [35].

e) PLP(Perceptual Linear Predictive) : Le Prédictif linéaire perceptif est une technique basée sur le spectre à court terme de la parole. Il est similaire à l'analyse LPC et MFCC, car il est au but de décrire plus précisément la psychophysique de l'audition humaine dans le processus d'extraction des caractéristiques [35].

Le processus de cette méthode se résume à trois étapes de traitement consécutives [36] :

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

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

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

f) RASTA-PLP : RASTA est une abréviation de RelAtive SpacTrAl, cette méthode a été développée en raison des limitations rencontrées par l'algorithme PLP, afin d'atténuer les effets des distorsions spectrales linéaires. Donc, le principe de la méthode RASTA-PLP est de remplacer le spectre à courte terme par un spectre estimé, où par passage à travers un filtre chaque canal fréquentiel est modifié. Lors de l'exécution de ce filtrage dans le domaine spectral logarithmique, il supprime les composantes spectrales fixes et de celui-ci les effets convolutifs du canal de communication sont également supprimés [36].

3.7 Conclusion

Dans ce chapitre, nous avons donnés quelques notions de base sur le traitement de signal, ainsi nous avons introduits le principe de la DAV et nous avons donnés aussi une brève description sur des différents techniques d'extraction des caractéristiques d'un signal. Dans le chapitre suivant, nous présenterons le chemin évolutif vers les méthodes DAV.

Chapitre 4

État de l'art

4.1 Introduction

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

Ce chapitre donne un bref aperçu des théories et des recherches antérieures basées sur la DAV. Nous avons menés une étude bibliographique afin de déterminer les sujets à aborder. Donc, nous avons Atteins à étudier 3 approches principales que nous traiterons dans ce chapitre.

4.2 Approche fondé sur le traitement du signal

La plupart des méthodes DAV ont été développées dans la littérature pour les télécommunications. Donc, dans cette approche, nous avons abordés l'énoncé des deux algorithmes les plus courants qui sont utilisés comme méthodes de référence traditionnelles dans la littérature.

4.2.1 Algorithme DAV G729 Annexe B

G729 a été proposé en 1996 par l'union internationale des télécommunications. G729 est un codeur de la parole pour la communication fixe et multimédia, sa propre extension B qui s'appelle G729.B présente la DAV [37].

G729.B est utilisé pour prendre en charge la transmission discontinue avec DAV. Donc, dans cet algorithme le signal audio est divisé en trames de 10 ms, où extraire un ensemble de paramètres pour chaque trame et utiliser le modèle de bruit de fond pour la prise de décision [38].

Les paramètres extraits comprennent le calcul de la différence de puissance en bande basse et bande pleine entre le signal d'entrée et le modèle de bruit, en plus des fréquences spectrales (distorsion spectrale). Enfin, la différence de croisement zéro entre le signal d'entrée et le modèle de bruit est calculée.

Les avantages de cet algorithme, il contient un système de correction de décision pour réduire les erreurs de classification. Aussi, dans des conditions de bruit modéré, il prouve son efficacité et sa force.

Mais l'un de ses inconvénients est que lorsque le niveau de bruit est élevé, il produit un faible taux de détection de la parole.

4.2.2 DAV de AMR (Adaptatif Multi-Rate)

Des standards pour deux options de DAV du codeur de la parole à multi-taux adaptatif ont été proposés par l'Institut européen des normes de communication au cours de l'année 1998 [37].

Par Système Global pour les Communications Mobiles, AMR a été développé pour les systèmes de communication mobile de troisième génération.

Les deux DAV que nous appelons AMR1 et AMR2 fonctionnent comme suit [38] :

Grâce à l'utilisation des banques de filtres avec de concentration sur des bandes de fréquences plus élevées, AMR1 décompose le signal audio en 9 sous-bandes. Aussi, il calcule les estimations de puissance ainsi que le RSB pour chaque sous-bande.

Pour la prise de décision qui est suivie d'un schéma qui corriger la décision préliminaire, la somme des RSB est d'abord comparée au seuil adaptatif.

Quant à l'option AMR2 utilise des banques de filtres FFT, alors elle est similaire à AMR1. AMR2 a 16 sous-bande, en trames de non parole il adapte l'énergie du bruit de fond de chaque bande.

En général, comme point fort de AMR, il fonctionne bien dans des conditions de bruit variables. Mais, à cause de son comportement conservateur, la précision de non parole est détériorée.

4.3 Approche fondé sur les modèles statistiques

•Test du rapport de vraisemblance

Ou bien LRT (Likelihood Ratio Test), est un test statistique sur la base duquel l'ajustement de deux modèles statistiques est comparée pour les données de l'échantillon, parmi de ces deux modèles que faire une compétition entre eux, un modèle non contraint(tous les paramètres), et un modèle contraint (l'hypothèse nulle) [39].

•Le modèle de Markov caché [40]

HMM (Hidden Markov Model en anglais), c'est un modèle statistique apparu dans les années 1970 en raison du problème de la reconnaissance automatique de la parole (RAP).

La parole peut être distinguée par un processus aléatoire dont les paramètres sont correctement estimés, d'où l'idée des modèles HMM et leur utilisation, car ils se sont avérés efficaces dans de nombreux domaines de la RAP.
Le modèle de Markov caché est défini par : $\emptyset=\{A,B,\pi\}$ tel que :

$A =\{ a_{ij}\}$: Une matrice des probabilités de transition entre états ;

$B = \{ b_i(K)\}$: Une matrice des probabilités d'émission des observations dans chaque état ;

$\pi=\{ \pi_i\}$: Une matrice de distribution de l'état initial.

Le modèle le plus couramment utilisée dans la modélisation acoustique est le modèle HMM gauche-droit où il n'est pas possible de revenir à l'état précédent, Comme le montre la figure 4.1.

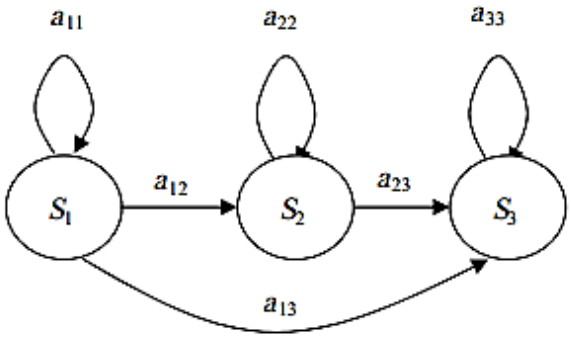


FIGURE 4.1 – Exemple de HMM à 3 états gauche-droit

•Modèle de mélange gaussien

GMM (Gaussian Mixture Model en anglais), C'est un modèle statistique qui représente la densité du mélange. Les variables aléatoires sont modélisées comme la somme de nombreux gaussiens appelés noyaux, où GMM est utilisé pour estimer la distribution de ces variables pratiquement, à partir de lui, des paramètres sont déterminés pour chaque gaussien (variance, moyenne, amplitude), où Ces derniers sont améliorés au moyen de maximum de vraisemblance afin d'atteindre la distribution souhaitée, grâce à l'algorithme d'espérance-maximisation cette procédure est effectuée répétitivement [41].

4.3.1 Etude (Joon-Hyuk Chang et autres, 2006)

Cette étude montre une DAV basé sur multiples modèles statistiques. Où dans cette étude le travail des différents modèles statistiques proposés est vérifié afin de décrire la distribution des composantes spectrales dans les différents types de bruit.

La plupart des modèles statistiques basés sur l'utilisation de l'hypothèse gaussienne dans le domaine de la TFD ne constituent pas une observation réelle, c'est pourquoi certains algorithmes DAV ont été proposés qui dépendent de plusieurs modèles statistiques.

Semblable au modèle gaussien dans l'analyse des propriétés statistiques, en particulier pour représenter la distribution de chaque paramètre TFD, également les fonctions de densité de probabilité Laplacien et Gamma complexes ont été combinées.

Pour chaque modèle paramétrique, LRT a été dérivée, nouvelles méthodes basées sur l'optimisation dans des conditions de bruit variables ont été attient, car ces méthodes dépendent de l'estimation prédictive au lieu de l'estimation de la décision dirigée traditionnelle.

Le test de qualité d'aptitude (the goodness-of-fit : GOF) a été utilisé pour évaluer chaque modèle paramétrique. Ce test mesure l'étendue de l'écart du modèle supposé par rapport à la distribution expérimentale, parmi les tests GOF un test a été choisi Kolmogorov–Smirnov (KS) par internet, selon laquelle une méthode adaptative a été proposée pour choisir le meilleur modèle statistique en fonction des valeurs de RSB et de la diversité du bruit.

Avec l'aide de ROC (Receiver Operating Characteristic), la performance de l'approche DAV proposée a été évaluée dans des environnements à bruit stationnaire et non stationnaire.

Où les résultats suivants ont été atteints :

-Les deux modèles Laplacien et Gamma sont plus précis que le modèle gaussien pour la modélisation des spectres vocaux bruités.

-Une amélioration de la précision de la DAV.

Malgré les résultats obtenus, la complexité du calcule parmi l'un des inconvénients [42].

4.3.2 Etude (Xulei Bao et Jie Zhu, 2012)

Cette étude comprenait la DAV basé sur une nouvelle approche de la reconnaissance vocale qui utilise un modèle de mélange gaussien caché HMM/GMM.

Généralement, certaines méthodes dépendent d'un grand nombre de mélanges GMM et dépendent de la détermination d'un seuil expérimental, mais elles ne sont pas plafonnées avec une grande précision et ne sont pas appropriées dans certains cas. Par conséquent, pour réduire le nombre de mélanges et ne pas recourir à seuil expérimental, pour améliorer la précision de DAV. Donc, il faut s'appuyant sur HMM basé sur GMM.

L'algorithme proposé dans cette étude est basé sur l'hypothèse que la non-parole peut être considérée comme un son supplémentaire qui correspond aux phonèmes traditionnels de la langue mandarin.

Également certaines caractéristiques vocales avancées qui sont : statistiques d'ordre élevé, informations sur la structure harmonique et MFCC pour représenter chaque trame de la parole / non-parole ont été incorporées, qui sont utilisées dans le principe du maximum de vraisemblance avec un algorithme Baum-Welch dans le modèle HMM/GMM hybride. De plus, la probabilité maximale des signaux détectés est recherchée en utilisant un algorithme de décodage Viterbi.

Une méthode différente basée sur des méthodes de reconnaissance vocale et de suppression du bruit a été présentée également pour démontrer que la méthode traditionnelle d'amélioration de la parole en utilisant uniquement DAV bien qu'elle donne des résultats précis, mais elle n'est pas efficace et insuffisamment pour RAP qui dégrade dans les systèmes RSB bas.

Des expériences ont été menées et une comparaison a été faite entre le travail de la

méthode proposée et certaines méthodes DAV courantes, en particulier ces méthodes DAV dans le même groupe de parole mandarin.

Les résultats ont montré des bonnes performances et l'efficacité de l'algorithme proposé. À l'avenir, certaines limitations devraient être envisagées pour réduire la distorsion de la parole et l'algorithme DAV plus puissant [43].

4.4 Approche fondé sur l'apprentissage profond

La plupart des architectures d'apprentissage profond ont été utilisées dans la DAV, car elles ont montrées une excellente amélioration de la détection par rapport aux méthodes basées sur le traitement du signal et les modèles statistiques.

Dans cette partie, nous nous étudions 5 études pour démontrer les procédures utilisées pour la DAV et les résultats obtenus.

4.4.1 Etude (Thad Hughes and Keir Mierle, 2013)

Cette étude comprenait un modèle RNN pour la détection d'activité vocale. Pour prédire la probabilité de parole de chaque trame, le système DAV est parfois basé sur GMM suivi de HMM, où HMM-DAV système se compose généralement d'un petit nombre d'états cachés, ce qui conduit à l'échec du traitement de chaque trame indépendamment, le modèle ne se souvient pas beaucoup du passé, donc une structure RNN a été proposée dans cette étude afin de remédier à ces limitations.

Un modèle RNN-DAV pour cette étude a été conçu et configuré sous la forme de plusieurs couches des réseaux de neurones avec alimentation direct et avec des connexions répétées. Les nœuds RNN reçoivent des entrées des nœuds dans les étapes de temps précédentes, l'état est stocké et traité à plusieurs reprises pour créer la séquence. Les nœuds RNN calculent les fonctions quadratiques de leurs entrées, suivies d'une option non linéaire facultative. Les entrées sont des caractéristiques PLP trois dimensions, Ceres Solver est une bibliothèque C++ open source qui est utilisée pour résoudre les problèmes de moindres carrés non linéaires dans cette étude pour effectuer l'optimisation pendant l'entraînement, en plus d'utiliser la différenciation automatique. L'entraînement est réalisé en deux étapes, la première dans laquelle les coefficients d'alimentation directe sont entraînés et la réparation de tous les paramètres répétitifs, où la mémoire est fournie pour RNN, la deuxième étape optimise tous les paramètres ensemble.

Des expériences ont été menées sur quatre systèmes de RNN parmi lesquels le système RNN complètement identique au modèle de cette étude, deux systèmes légèrement

différents du modèle précédent, tandis que ce dernier système a une structure traditionnelle similaire à PMC. Une base de référence forte qui consiste en GMM et une machine à états SM (State Machine en anglais) réglée manuellement pour l'homogénéité temporelle. Cette base de référence forte a été comparé avec les performances des quatre systèmes RNN précédents.

L'évaluation de la performance du modèle correspondant à l'étude se faire avec la vie réelle en l'incluant dans le système RAP, ainsi l'évaluation a été menée sur la base de faux accepter, faux rejet et le taux d'erreur de mot (WER : Word Error Rate).

Les résultats ont montré que le modèle RNN de cette étude est nettement supérieure aux systèmes DAV basés sur GMM + SM au Pourcentage de 26% en réduisant les fausses alarmes, ce qui entraîne une réduction de 17% du temps total de calcul de la reconnaissance vocale tout en réduisant le taux d'erreur de mot de 1% relativement. Le modèle RNN de cette étude a également montré sa supériorité sur le RNN traditionnel similaire à PMC. À l'avenir, s'appuyer sur des algorithmes génétiques peut conduire à de meilleures solutions [44].

4.4.2 Etude (Phuttapong Sertsi et autres, 2017)

Cette étude visait à démontrer l'amélioration du pouvoir de DAV en proposant une structure composée de LSTM et de spectre de modulation MS (Modulation Spectrum). LSTM dépendant sur ses trois portes, le MS représente également une fluctuation temporelle du signal de parole, où le tracé spectral représente une série de spectres à court terme résultant de l'application d'une transformation Fourier d'une courte portion du signal de la parole, ce tracé décrit le contenu fréquentiel de cette parole qui change avec le temps. MS se caractérise par des caractéristiques spécifiques telles que l'emplacement du pic, la bande passante, la valeur Q ou la pente après le pic.

Généralement, la DAV basée sur LSTM est entrainée par des signaux vocaux bruités pour améliorer sa force, à partir des caractéristiques vocales fortes de l'entrée se complète cette force. Généralement MFCC, énergie et Pitch sont utilisées pour entraîner LSTM-DAV traditionnel, Mais dans cette étude, il a été suggéré d'ajouter des informations MS à ces caractéristiques, la valeur Q a été choisie pour représenter MS de la parole et du bruit. Cette valeur provient du modèle lisse de MS résultant de plusieurs trames pour les signaux vocaux, car le nombre insuffisant de trames entraîne une différence MS qui se traduit par une erreur dans le calcul de la valeur Q et à partir de là, la zone sous la courbe de MS a été utilisée.

Le modèle d'étude a été examiner sous plusieurs types de bruit (Aurora5) et avec différentes valeurs de RSB, également la base de données vocale Lotus a été utilisée,

alors que les signaux vocaux propres ont 22 heures de données vocales pour 40 locuteurs (20 femmes et 20 hommes). Boîte à outils Kaldi a été utilisé pour extraire MFCC et Pitch à partir des données d'entraînement et à partir de la première caractéristique de MFCC la caractéristique MS a été calculée, l'entraînement a été effectuée en utilisant un ensemble de boîte à outils keras. Les données d'évaluation se composent de trois groupes différents, qui comprenaient une comparaison entre le modèle traditionnel et le modèle proposé.

Deux expériences ont été menées, la première pour tester le modèle d'étude sur différentes valeurs de RSB, la seconde pour le tester dans des conditions bruyantes (invisibles) de la vie réelle, après quoi l'évaluation a été menée sur la base du taux de fausse acceptation et le taux de faux rejet.

Les résultats de cette étude sont les suivants :

La capacité de la méthode proposée à déterminer les sections de parole et de non-parole dans des environnements bruyants avec RSB faible, ainsi qu'une valeur de précision améliorée de 3.5% a été obtenue dans des conditions bruyante dans des environnements réels. À l'avenir, les performances de la méthode proposée seront améliorées en considérant une autre caractéristique de MS telle que la pente [45].

4.4.3 Etude (Yeonguk Yu et autre, 2018)

Cette étude a traité un modèle de DAV basé sur BLSTM-RNN et le mécanisme de l'attention. Pour juger des trames, il est nécessaire de regarder la trame précédente et la trame suivante, par conséquent la structure BLSTM-RNN a été proposé. Le mécanisme d'attention a été proposé également pour se concentrer sur les trames importantes.

La structure du système DAV-LSTM consistait à créer deux systèmes N1 et N2 chacun d'eux contient un numéro spécifique de LSTM cellules, la valeur de sortie de ces deux modèles était de (-1) pour la non-parole et de (1) pour la parole, le coût de cette valeur est estimé par la valeur de l'erreur moyenne de la racine carrée (RMSE), où elle est réduite pendant l'entraînement.

La confusion est formée à partir de trames contradictoires adjacents. Pour estimer DAV, la plage des trames environnants doit être ajustée différemment. Pour juger les trames, il faut considérer également la distance entre les trames, le nombre de trames et le type des caractéristiques du trame à partir de la technique du mécanisme d'attention qui est formé d'un encodeur qui convertit l'état caché de LSTM à un matrice, il se compose également d'un décodeur qui montre l'état important parmi les états cachés.

Donc, la structure générale se compose de 5 parties, codage basé sur la séquence LSTM, codage de trame, attention de trame, classification de trame, spécification d'architecture pour le modèle proposé qui comprenait la dernière partie de la structure générale, où deux modèles ont été créés AN1 et AN2, chacun d'eux contient un certain nombre de (couches RNN et cellules LSTM) et une couche d'attention.

Le jeu de données utilisé est TiMit qui a été généré à travers 4 opérations, l'édition de données, l'ajout de bruit, l'extraction de caractéristiques (RASTA-PLP) et enfin la création d'étiquettes pour décrire les valeurs d'étiquette, où l'étiquette de la trame de parole a été attribuée une valeur de (1) et la trame de non-parole est (-1), pour les modèles N1, N2 et AN1. Comme pour le modèle AN2 une valeur de (0) est spécifiée pour la trame de non-parole et (1) pour la trame de parole.

Les expérimentations ont été menées selon les quatre modèles précédents dans des environnements Python et Tensorflow, des processus d'entraînement et d'évaluation sont exécutés et l'expérimentation est répétée cinq fois. Pour l'évaluation des performances, un taux d'erreur égal est utilisé.

Où le modèle AN2 montre une amélioration par rapport aux autres modèles d'environ 3%, le problème de cadre déroutant dans le modèle AN2 est résolu par le mécanisme d'attention. Où les résultats a conduit au démontrer la précision du modèle AN2 de cette étude dans une meilleur détection d'activité vocale [46].

4.4.4 Etude (Tianjiao Xu, 2019)

Cette étude visait à démontrer l'amélioration de l'utilisation des données par l'apprentissage en deux étapes dans la DAV basée sur CNN-LSTM. La combinaison de DNN, CNN et LSTM a conduit à la suggestion d'un modèle hybride appelé CLDNN "réseaux neuronaux profonds de mémoire à court long terme convolutive" proposé par Sainath et autres, afin de profiter de l'avantage de chacun, car DNN est bon pour assigner des caractéristiques dans un espace plus séparable, CNN est bon pour extraire des caractéristiques et LSTM est bon pour traiter les données de séquence. Où ce modèle a été utilisé par Zago et autres pour la DAV qui a prouvé son efficacité et il a apporté une excellente amélioration dans le problème de DAV.

Au moyen du système ShuffleNode proposé par Sainath et autres, les éléments de la carte des caractéristiques ont été arrangés aléatoirement de sorte que pendant la formation du modèle, afin de remplir les fonctions de régularisation.

Dans cette étude, pour réduire la variance de fréquence de l'entrées, CLDNN applique CNN en bas, puis passe à LSTM pour l'intention de modélisation temporelle et de là à DNN. Où CNN dans la première étape est entrainer sur des données au niveau de la trame et à partir de cela une expression caractéristique de haut niveau est obtenue, qui dans la deuxième étape est entraînée à l'aide de données de séquence par LSTM.

Toutes les expérimentations ont été menées sur la base de données TiMit, et les caractéristiques banque de filtres log-mel 40 dimensions ont été utilisées.

La méthode proposée a été comparée à quatre méthodes DAV pour : CNN pur, LSTM pur, une méthode statistique et la méthode originale CLDNN qui sont sous quatre types de bruit visible, les quatre nouveaux types de bruit de fond avec des valeurs RSB différentes.

Pour l'amélioration, tous les modèles ont été entrainer par un optimiseur Adam. Pour l'évaluation, AUC et ROC sont utilisés. L'étude a trouvé les résultats suivants :

La méthode proposée réalise une amélioration relative par rapport à CLDNN original de plus de 2.89% dans le cas de l'adaptation de bruit, et de plus de 1.07% dans des conditions inégalées.

Avantages de cette étude, la capacité discriminatoire et La capacité de généraliser.

Cependant, pour obtenir un système DAV précis, la méthode proposée utilise des

données d'apprentissage très limitées.[47]

Pour passer à la prochaine étude, il faut d'abord savoir ce que c'est le WebRTC ?

WebRTC : Communication en temps réel pour le web,c'est une technologie open source gratuite pour l'échange de communications audio et vidéo humaines en temps réel, disponible dans les navigateurs Web et les applications mobiles. Ces connexions sont effectuées sans installer de composants supplémentaires ou d'applications externes [48].

4.4.5 Etude(Lu Ma, 2020)

Cette étude visait à combiner les deux modèles DNN et GMM pour la DAV, étant donné que le modèle GMM a une capacité de modélisation limitée, en particulier dans l'environnement sonore complexe. Par conséquent, pour améliorer la précision de la DAV, on peut recourir à un modèle DNN, cependant DNN rencontre des inconvénients lors de l'application de données de tailles et de types différents. Donc, la combinaison de ces deux modèles pourrait mieux détecter la parole.

Le modèle DNN dépend de deux phases, la première étape dans laquelle 29 dimensions des caractéristiques de banque des filtres sont extraites, après avoir effectué certaines opérations à travers lesquelles deux éléments sont produits, dont l'un montre la probabilité de la parole et l'autre probabilité du silence. En raison de la limitation des appareils et du temps de réponse, seules deux couches de DNN simple ont été utilisées. Quant à la deuxième étape, elle dépend de la préparation des données pour entraîner DNN, donc deux types de corpus ont été fabriqués, le premier représente des données de simulation, à travers de l'ajout de bruit aux fichiers purs et à mesure que les vecteurs de caractéristiques sont calculés. La trame de la parole propre (après avoir calculé l'énergie) est comparée au seuil, à partir de celui-ci une désignation supplémentaire a été utilisée contrairement à 0 et 1 qui est 0.5 où pendant les transitions il montre une incertitude dans la parole. Le deuxième type de corpus est les données pratiques, par des appareils réels ce type de données a été collecté.

Le modèle GMM, il a été dérivé graphiquement et utilisé à partir d'une source ouverte de WebRTC, le rapport de vraisemblance logarithmique (LLR) a été exprimée, où LLR total est obtenir en pondérant les six sous-bandes dans le WebRTC. Pour distinguer d'abord la parole et le bruit, LLR pour chaque sous-bande est comparé à un seuil spécifique et le LLR total est comparé à un autre seuil pour détecter si est une parole ou non.

Dans le schéma de combinaison, la mise à jour des coefficients d'un modèle GMM

est amélioré au moyen de la probabilité de parole et de silence de DNN. Pour vérifier les performances, un schéma de contrôle a été conçu qui détecte les points finaux de la parole grâce à l'utilisation de tampon de données interne et tampon de données externe.

Les opérations de validation faire grâce à deux étapes, y compris la phase de test en fournissant 4 plates-formes différentes avec des algorithmes différents. En raison de limitations matérielles, le modèle DAV qui est utilisé pour les deux appareils, il est attribué à l'un des deux modèles respectivement. Quant à la deuxième étape, elle a été représentée dans l'étape de simulation, où un bruit d'un type différent a été ajouté dans la parole propre avec une différence dans les valeurs de RSB. la DAV avec un modèle GMM qui a plus d'efficacité que la DAV avec le modèle DNN en raison d'un entrainement insuffisante.

Enfin, on est parvenu au résultat que le schéma proposé en intégrant DNN et GMM est supérieur et efficace en comparant à l'exécution du travail d'un schéma indépendant pour chacun d'eux [49].

Les points forts de cette étude étaient la combinaison de l'apprentissage supervisé et non supervisé.

•Performance de DAV

Pour mesurer la précision des performances des modèles basés sur la DAV, les métriques suivantes sont utilisées :

RMSE (Root Mean Square Error) : L'erreur quadratique moyenne mesure l'erreur du modèle dans la prédiction des données quantitatives, donc l'utilisation de RMSE au but d'évaluer la précision des modèles entraînés, ou comme heuristique pour entraîner le modèle sur la base de sa réduction [50].

RMSE est donné par la relation suivante :

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \qquad (4.1)$$

avec :

$\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$: Valeurs prédites.

$y_1, y_2, \ldots, y_n$: Valeurs observées.

n : Nombre d'observations.

WER (Word Error Rate) : Taux d'erreur de mot, l'une des mesures courantes pour mesurer la précision des applications de reconnaissance vocale, il est basé sur l'erreur dans les mots. Il est calculé comme suit [51].

$$WER = \frac{S + I + D}{N} \qquad (4.2)$$

S : Le nombre de substitutions intervenues entre les mots.

I : Le nombre de mots insérés.

D : Le nombre de mots manquants.

N : Le nombre de mots déjà prononcés.

FAR (False Acceptance Rate) : Taux de fausse acceptation est la probabilité de déterminer la trame de parole pendant un intervalle de temps silencieux, il est donné par la relation suivante [45] :

$$FAR = \frac{N_{FA}}{N_s} \times 100\% \qquad (4.3)$$

N_{FA} : Nombre de trames non parole détectées comme trames de parole.

N_s : Nombre total de trames.

FRR (False Rejection Rate) : Taux de faux rejet est la probabilité de déterminer la trame de silence pendant la période de parole est donnée par la relation suivante [45] :

$$FRR = \frac{N_{FR}}{N_s} \times 100\% \qquad (4.4)$$

N_{FR} : Le nombre de trames de parole détectées comme trames non parole.

EER (Equal Error Rate) : Taux d'erreur égal représente la valeur commune entre FAR et FRR, c'est-à-dire le point d'intersection entre les deux courbes FAR et FRR, où plus la valeur de EER est faible, plus la précision est élevée [52].

AUC (Area Under the Curve) : Zone sous la courbe est considéré comme une mesure de séparabilité.

ROC (Receiver Operating Characteristic) : Caractéristique de fonctionnement du récepteur est un courbe de probabilité représentée par le taux de vrai positif et le taux de faux positif.

Ainsi, une courbe AUC-ROC mesure la performance tout en fixant différents seuils dans le problème de classification, également elle montre la capacité du modèle à différencier les classes [53].

4.5 Aspects de bénéficier des études précédentes

1) Montrer les étapes nécessaires pour les procédures de la DAV.

2) reconnaissance des différentes métriques utilisées, pour mesurer les performances de DAV.

3) Reconnaissance des différents résultats obtenus, en particulier ceux trouvés dans l'apprentissage profond

4.6 conclusion

Dans ce chapitre, nous avons donnés quelques études antérieures menées dans la littérature sur les performances de la DAV de diverses manières.

Nous avons classés ces études sous trois approches, approche basée sur le traitement du signal et l'autre basée sur les modèles statistiques, le dernier approche basée sur l'apprentissage profond. Ce qui a conduit à la description des avantages et des limites de certains méthodes et Nous avons classés quelques aspects de bénéficier pour ces études. Ce qui nous permet de commencer la mise en œuvre de notre étude, qui présentera dans le prochain et le dernier chapitre de notre travail.

Chapitre 5

Expérimentations

5.1 Introduction

Dans cette partie pratique de ce projet, nous avons utilisés le code source principal obtenu à partir de : https ://github.com/nicklashansen/voice-activity-detection.

Les étapes de mise en œuvre de l'approche proposée sont présentées dans le cadre de la DAV afin d'évaluer ses performances selon des critères précis.

Tout d'abord, nous abordons l'environnement de travail, la langue et les ressources utilisés, puis nous présentons les étapes de mise en œuvre de l'algorithme proposé, jusqu'aux résultats expérimentaux obtenus après les tests qui ont été réalisés.

5.2 Environnement de développement

5.2.1 google colab

Google Colab est un service cloud gratuit, basé sur Jupyter Notebook et le langage de programmation Python, pour développer les applications d'apprentissage profond sans avoir de contrainte matérielle utilisant des bibliothèques telles que TensorFlow et Keras. Colab se distingue des autres services cloud gratuits, en fournissant une unité de traitement graphique GPU (Graphic Processing Unit) gratuite [54] [55].

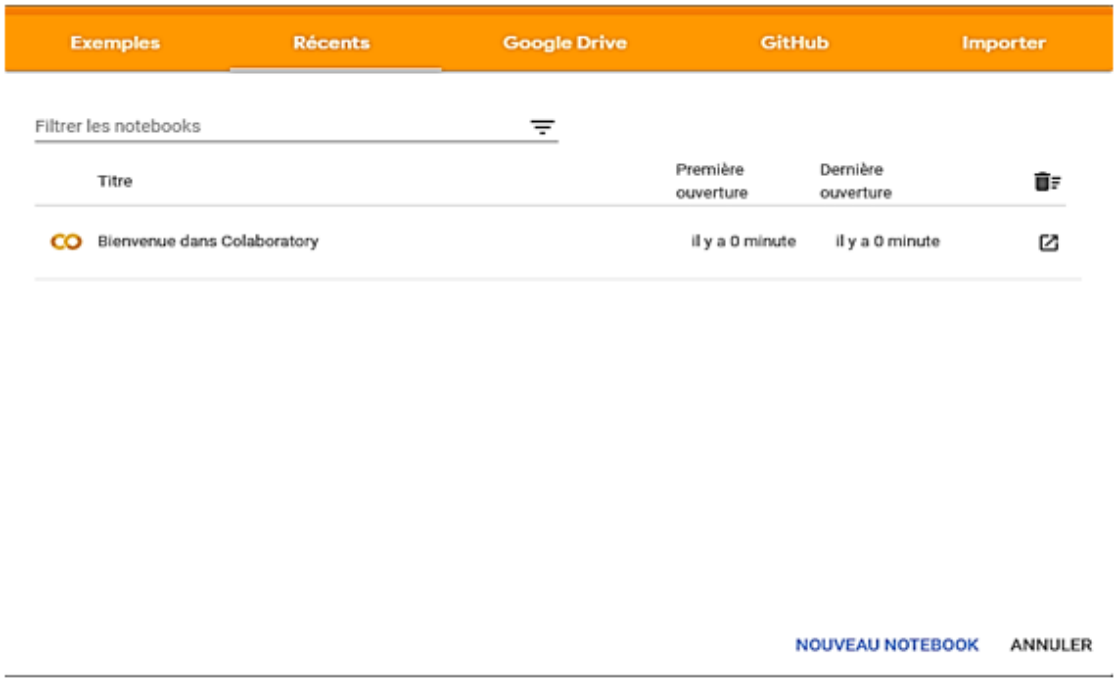


FIGURE 5.1 – Environnement de Google Colab

La figure 5.1 représente l'interface initiale de Google Colab. Pour développant notre

approche, nous nous sommes appuyés sur le GPU, car il accélère les performances et effectue des calcules élevées.

5.2.2 Python

Python est un langage de programmation open source de haut niveau, interprété et multi paradigme utilisé pour la programmation générale, il a été créé par Guido van Rossum pour être publié en 1991, il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il dispose d'un système de type dynamique et d'une gestion automatique de la mémoire. Les interpréteurs de Python sont disponibles pour de nombreux systèmes d'exploitation.

Le langage Python prend en charge les principaux aspects du cycle de vie de l'apprentissage automatique et de l'apprentissage profond, car il dispose de nombreuses bibliothèques qui aident à développer ce domaine, en plus de faciliter la gestion des fichiers audio [56].

FIGURE 5.2 – Logo de python

5.3 Bibliothèques utilisées

NumPy : Est une bibliothèque fondamentale en python, pour traiter les tableaux homogènes multidimensionnels et aussi travailler avec l'algèbre linéaire [57].

Matplotlib : Est une bibliothèque de traçage, qui peut représenter les données sous plusieurs visualisations [57].

Scikit-learn ou Sklearn : Est une robuste bibliothèque pour créer des modèles d'apprentissage automatique, cette bibliothèque s'appuie sur la fourniture de divers algorithmes de classification [58].

Python-Speech-features : Est une bibliothèque qui fournit les caractéristiques MFCC et les énergies de banque de filtres pour la RAP [59].

Soundfile : Est une bibliothèque audio basée sur NumPy. Il peut lire et écrire des fichiers audios pour les représenter sous forme de tableaux NumPy [60].

Torche : Est une bibliothèque open source pour l'apprentissage automatique, qui sert un langage de script et un cadre pour les calculs scientifiques, car elle fournit divers algorithmes d'apprentissage profond [61].

5.4 Ensembles de données

Dans ces expérimentations, nous utilisons deux ensembles de données qui sont définis comme suit :

5.4.1 LibriSpeech

Représente un groupe de discours lisibles en anglais, dérivés de livres audio basés sur le projet LibriVox. Le but de leur utilisation est d'entrainer et d'évaluer les systèmes de RAP. Cet ensemble de données open source contient 1000 heures de parole échantillonnées à 16 kHz.

LibriSpeech a été divisé en plusieurs parties. Dans ces expérimentations, le sous-ensemble **Train-clean360** a été utilisé, qui consistait en 360 heures d'enregistrements vocaux propres, dont (36 heures, 10%) ont été utilisés et échantillonnés au hasard de manière uniforme.

Cet ensemble de données est disponible sur : http ://www.openslr.org/12/

5.4.2 QUT-NOISE

Est un ensemble de données qui fournit un bruit naturel pour une parole propre, car cet ensemble comprend différents types de bruits courants auxquels l'utilisateur est exposé dans la vie quotidienne (café, maison, rue, voitures, acoustique) afin de simuler des scénarios réels avec un bruit de fond. Les sous-ensembles du bruit de café et de maison de cet ensemble de données contiennent un discours de fond subtil, ce qui conduit à un étiquetage incorrect de l'audio, car ce discours est techniquement une voix, donc

ces deux sous-ensembles ont été exclus.

L'ensemble de données QUT-NOISE est disponible sur :
https ://research.qut.edu.au/saivt/databases/qut-noise-databases-and-protocols/

5.5 Pré-traitement

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

Dans une distribution uniforme, l'ensemble de données LibriSpeech est découpé en tranches ou fragments de longueur variable entre 1000 ms et 5000 ms, aussi avec cette longueur variable similaire, une quantité égale de tranches contenant du silence est créée, afin de créer un seul ensemble de données avec un modèle aléatoire de parole et de silence, grâce au fusionnement et la mélange des deux ensembles de tranches précédentes. Alors, on obtient une distribution 50/50 de parole et de silence pour un total de 72 heures.

Afin de simuler un environnement varié, trois niveaux de bruit différents (aucun, faible(-15dB) et élevé(-3dB)) sont ajoutés à l'audio de la parole, après tout un bruit est normalisé.

Le WebRTC DAV est utilisé pour étiqueter l'audio en utilisant une taille de trame de 30 ms et une sensibilité de zéro, cette sensibilité indique une mesure statistique de la performance d'un test de classification binaire pour la DAV, elle mesure la proportion de trames positifs correctement identifiés.

5.5.1 Extraction des caractéristiques

Puisqu'un vecteur de caractéristiques MFCC ne décrit que l'enveloppe spectrale d'énergie d'une seule trame, il semble que la parole aurait également des informations dans la dynamique, en représentant les trajectoires des coefficients MFCC dans le temps. A partir de là, le calcul des trajectoires MFCC et leur ajout au vecteur de caractéristiques d'origine conduit à une augmentation relative et à une amélioration des performances de RAP [62].

Alors, à l'aide d'une taille de trame de 30 ms, les coefficients delta qui aussi appelés coefficients différentiels, sont calculés avec les MFCC, comme montre la figure 5.3. Dans cette expérimentation, nous obtenons 12 coefficients MFCC et à partir de là nous obtiendrons également 12 coefficients delta, qui se combinent pour donner un vecteur de caractéristiques de longueur 24, pour chaque trame audio résultant en un ensemble de caractéristiques bidimensionnel, comme indiqué sur la figure 5.4.

```
[[ 2.71596432e+01  5.47562838e+00  4.53414631e+00 -1.12650881e+01
  -5.35936785e+00  1.04649341e+00 -1.51445675e+01 -1.31209219e+00
  -1.75492311e+00 -1.08300323e+01  9.90102053e-01  5.16945696e+00
   4.30280596e-01 -4.43186969e-01  4.88138556e-01  9.25511003e-01
  -1.51928797e-01 -6.37707651e-01 -1.10511258e-01 -4.70654696e-01
  -6.50285363e-01 -3.80999386e-01  1.73394513e+00 -1.19245738e-01]
 [ 2.85016499e+01  4.17739773e+00  5.83749533e+00 -9.36771965e+00
  -6.96123171e+00 -2.31404066e-01 -1.51406298e+01 -2.04787326e+00
  -2.52827978e+00 -1.23413706e+01  7.65062380e+00  6.28943157e+00
   6.38171434e-01 -7.61947155e-01  7.65742600e-01  1.33868325e+00
   2.61046905e-02 -9.29551721e-01 -1.89037159e-01 -8.16654503e-01
  -1.09646964e+00 -6.22445643e-01  1.29897451e+00 -5.05945444e-01]
 [ 2.86400433e+01  3.90880871e+00  6.32316446e+00 -7.58621788e+00
  -5.31807995e+00 -1.50309610e+00 -1.56990929e+01 -3.29747510e+00
  -4.61967134e+00 -1.19793596e+01  6.32956696e+00  4.01324129e+00
   5.70975125e-01 -8.79170179e-01  7.70557344e-01  1.76451123e+00
   6.32661760e-01 -1.16239870e+00  1.35291323e-01 -1.16887832e+00
  -1.14402664e+00 -3.81975591e-01  3.05643171e-01 -6.73624039e-01]]
```

FIGURE 5.3 – Matrice MFCC avec Deltas obtenue

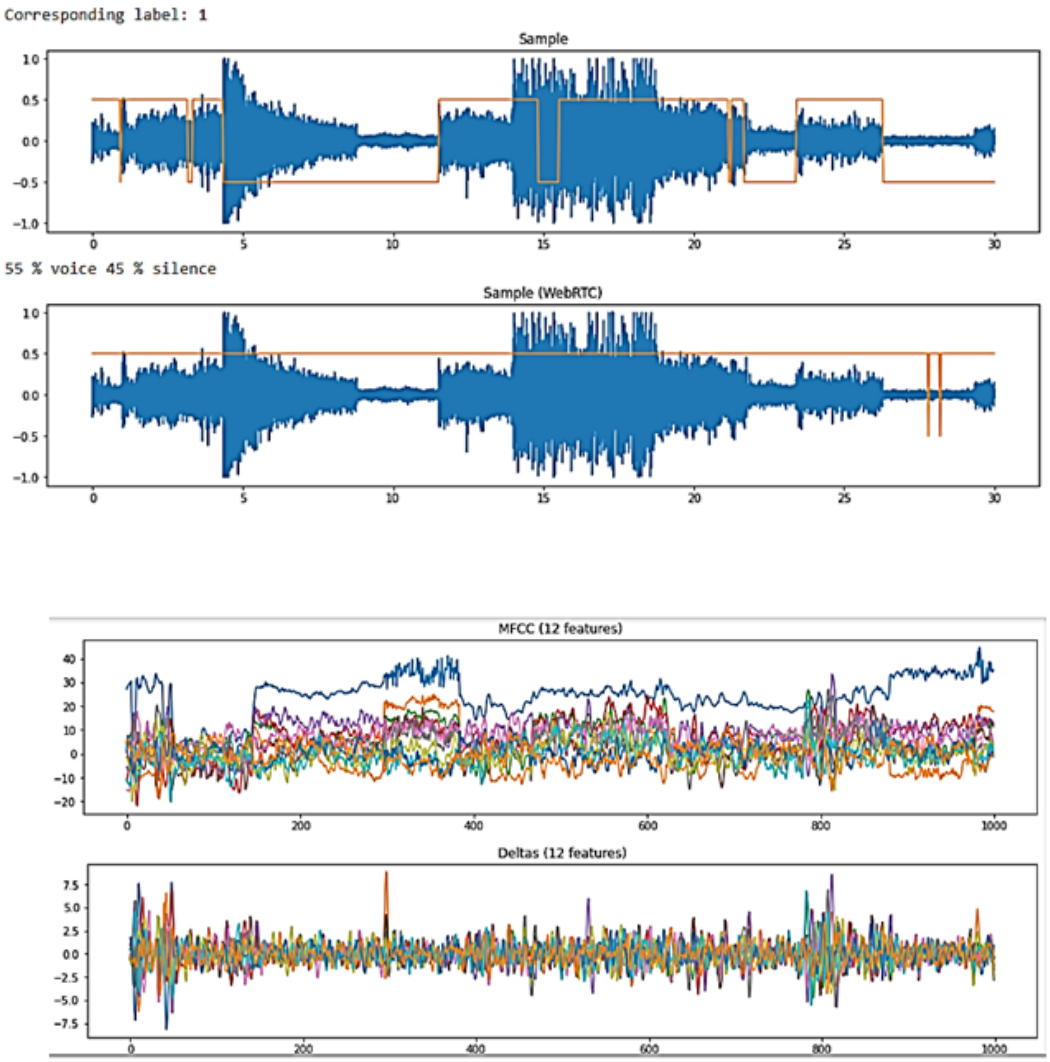


FIGURE 5.4 – Extraction des caractéristique MFCC et Deltas après l'ajout de bruit

5.6 Architectures utilisés

Dans cette expérimentation, trois architectures distinctes ont été utilisées pour DAV, qui sont : LSTM-RNN, GRU-RNN avec des caractéristiques dérivées des couches convolutives et l'implémentation de DenseNet.

Où deux tailles de réseau ont été utilisées pour chacune des architectures mentionnées ci-dessus. Un petit réseau avec 10 000 paramètres, et un grand réseau avec 30 000 paramètres. Donc pour évaluer DAV dans des environnements bruyants, six modèles distincts sont utilisés.

• **LSTM :** Cette architecture se compose d'une couche LSTM unidirectionnelle avec 30 cellules, selon la taille du réseau elle se compose d'une ou deux couches complète-

ment connectées. Il est considéré comme une ligne de base de performance.

• **GRU :** Cette architecture se compose de 3 ou 4 couches convolutives fermée selon la taille du réseau, une couche unidirectionnelle GRU avec 30 cellules et enfin une ou deux couches complètement connectées suivies d'une couche de sortie Softmax.

Dans l'implémentation des mécanismes de porte par le GRU est de façon similaire que LSTM, mais le GRU manque une unité de mémoire, ce qui expose l'état complétement caché à la cellule.

Pour capturer des motifs à court terme dans le temps, une convolution uni-dimensionnelle avec un rembourrage nul (zero-padding) et une taille de noyau fixe de 3 (90 ms) est appliquée le long de l'axe temporel. De sort, que chaque canal dans la couche convolutive est représenté par une caractéristique.

Les mécanismes de porte trouvés dans les LSTMs et GRUs sont similaire que la convolution fermée, tel qu'une entrée est convoluée indépendamment avec deux ensembles différents de filtres, des filtres réguliers et des filtres de porte, où la fonction sigmoïde est appliquée à la sortie des filtres de porte, d'autre part, la sortie des filtres réguliers passe par une fonction tanh (voir la figure 5.5).

$$h_k = \tanh(w_{f,k-1} * x_{k-1}) o \sigma(w_{g,k-1} * x_{k-1}) \qquad (5.1)$$

où :

h_k : Etat caché ;

x_{k-1} : Entrée ;

$w_{f,k-1}$,$w_{g,k-1}$: Poids.

o : Multiplication par élément.

L'équation (5.1) représente le calcule complet d'un état caché, tel que le produit des deux matrices résultantes transmis en tant que sortie finale de cette couche.

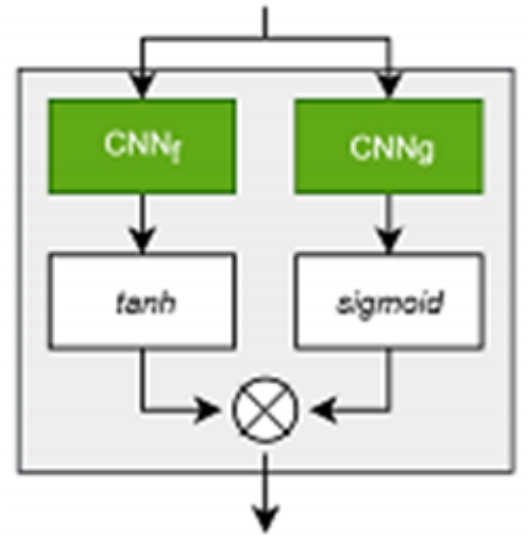


FIGURE 5.5 – Convolution gated telle qu'utilisée dans le GRU-RNN

Les connexions résiduelles n'ont pas améliorés les performances du modèle, donc ont été abandonnées. En raison du nombre limité de couches convolutives ainsi que du fait que les CNN peu profonds sont moins susceptibles de disparaître des gradients. Donc, deux techniques sont appliquées chaque fois que cela est approprié, qui sont la normalisation par lots (Batch normalization) et le décrochage (dropout).

•**DenseNet :** Cette architecture a été adaptée à la nature temporelle de la VAD. En connectant les couches CNN via la concaténation, DenseNet tente de maximiser le flux d'informations à travers les couches CNN, donc d'améliorer les propriétés d'anticipation des couches CNN. Où la couche CNN dilatée avec max-pooling représente ce qui est composé de la couche initiale, permettant une capture large de la fenêtre en mouvement tout en gardant le nombre de paramètres bas.

Dans deux blocs denses reliés par une couche de transition (lequel essaie de réduire le sur apprentissage), la sortie de la couche CNN dilatée est transmise. Ensuite, la sortie du dernier bloc dense est passée à travers une seule couche CNN avec max-pooling et enfin une seule couche Softmax.

A partir de la quantité de couches dans chaque bloc dense et le nombre de canaux utilisé dans les couches CNN, on peut déterminer la principale différence entre le petit et le grand DenseNet (voir la figure 5.6).

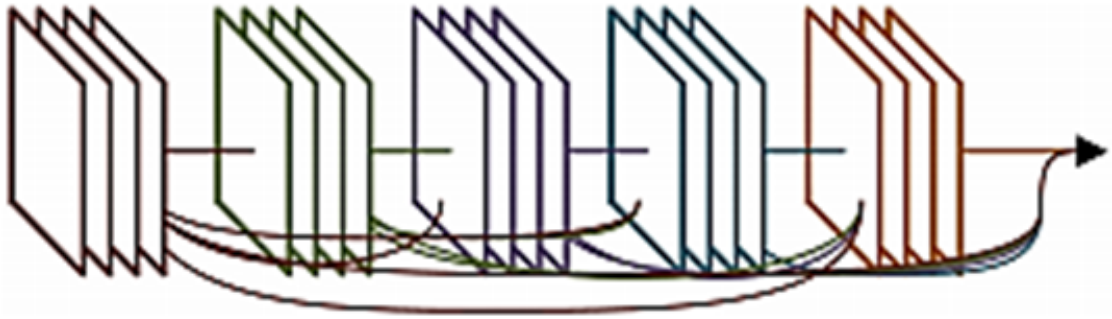


FIGURE 5.6 – Illustration d'un bloc dense tel qu'utilisé dans le DenseNet

5.7 Entraînement

• **Fonction de perte**

Dans cette expérimentation, la fonction de perte inclus la perte focale (Focal Loss : FL) avec l'utilisation de l'entropie croisée (Cross Entropy : CE).

Comme critère pendant l'entrainement, la perte focale est appliquée pour maximiser les performances dans des environnements bruyants. FL introduit à partir de la perte d'entropie croisée pour la classification binaire :

$$CE(p,y) = \begin{cases} -\log(p) & si\ y = 1 \\ -\log(1-p) & sinon \end{cases} \tag{5.2}$$

$y \in [\pm 1]$: Qui spécifie la classe de vérité terrain.

$p \in [0,1]$: Pour la classe avec l'étiquette $y = 1$, p indique la probabilité estimée du modèle.

Alors, p_t est définie comme suite :

$$p_t = \begin{cases} p & si\ y = 1 \\ 1-p & sinon \end{cases} \tag{5.3}$$

Donc :

$$CE(p,y) = CE(p_t) = -\log(p_t) \tag{5.4}$$

La perte FL ajoute à la perte CE un facteur de modulation $(1 - p_t)^\gamma$ où γ est un paramètre de focalisation accordable ($\gamma \geq 0$), alors :

$$FL(p_t) = (1 - p_t)^\gamma CE(p_t) = -(1 - p_t)^\gamma \log(p_t) \qquad (5.5)$$

Pour $\gamma = 0$, le facteur de modulation égale à 1, donc la perte est égale à celle de CE.

Exemple : pour une prédiction $p_t = 0.95$ et avec $\gamma = 2$, une perte FL résultant est 400 fois plus petite que CE.

Donc, γ sert à sous-pondérer l'influence d'échantillons bien classés (où $p_t \gg 0.5$). Pour minimiser les pertes, obligeant l'optimiseur à se concentrer beaucoup plus sur des échantillons mal classés, tout en exploitant toujours la disponibilité d'échantillons faciles [63].

•Hyperparamètres

Les hyperparamètres qui illustrés dans la figure 5.7 sont utilisés pour entraîner les modèles sur les trois niveaux de bruit pendant 15 époques et l'utilisation de FL comme critère, avec une valeur de γ attribuée à chaque modèle.

```
BATCH_SIZE = 2048
FRAMES = 30
FEATURES = 24
STEP_SIZE = 6
```

FIGURE 5.7 – Hyperparamètres utilisés

30 trames, indique une fenêtre de 900 ms.

•Métriques d'optimisation

les métriques d'optimisation suivantes sont utilisées pondant l'entrainement des modèles [64] :

1)Stochastic gradient descent (SGD) : La descente de gradient stochastique parmi l'une des méthodes d'optimisation, elle est couramment utilisée pour l'apprentissage profond, leur principe est de limiter les coûts de travail à travers de trouver la meilleure

configuration des paramètres des réseaux de neurones, en ajustant itérativement ces configurations.

2)Momentum : C'est un algorithme d'optimisation, il est venu accélérer SGD et résoudre ses différents problèmes.

3)ADAM (Adaptive Moment Estimation) : C'est une technique de calcul des taux d'apprentissage adaptatif pour tous les paramètres qui incluent l'entraînement des gradients. Adam nécessite moins de mémoire pour s'exécuter et réduit également les coûts de calcul.

Dans cette expérience, SGD est appliqué à DenseNet avec Momentum de 0.7 et taux d'apprentissage (learning rate : 1). Aussi, Adam est appliqué aux LSTM-RNN et GRU-RNN avec dégradation des pondérations (weight decay : 1e-5) et le taux d'apprentissage initial de 1e-3.

Sans chevauchement de la parole, les modèles sont évalués sur un ensemble de test généré à partir de la même source.

5.8 Résultats expérimentaux et discussion

Les performances des six modèles sont évaluées en fonction des trois niveaux de bruit au moyen de courbes ROC et l'utilisation de AUC comme mesure de performance initial. Aussi, FAR est calculé par rapport à FRR de 1%.

Une valeur γ a été spécifiée pour chaque modèle, où $\gamma = 2$ a été spécifiée pour tous les modèles, sauf le petit LSTM-RNN pour lequel $\gamma = 0$ a été spécifiée, afin d'obtenir de meilleurs résultats.

Étant donné que le prétraitement des données a pris 16 heures et que l'entrainement des modèles a pris environ 3 heures au début, mais en raison de la grande taille des données utilisées, et bien que nous ayons essayés d'utiliser un volume plus petit, l'espace de stockage dans l'environnement de développement n'était pas suffisant pour atteindre le reste des résultats, et à partir de là, nous ferons une comparaison des résultats obtenus précédemment.

La figure 5.8 représente les courbes ROC pour chacun des trois niveaux de bruit. Alors, nous analysons les résultats obtenus ci-dessous.

Dans le cas d'un bruit élevé, FL réalise un effet positif sur le grand LSTM de AUC=0.969, par rapport à le petit LSTM (AUC=0.965) qui n'a pas été affecté par FL. Le grand GRU réalise AUC de 0.968, qui approche que le grand LSTM, tandis que le petit GRU avec AUC= 0.961, qui surpassait le grand et le petit DenseNet.

Dans le cas du faible bruit, le grand GRU réalise AUC=0.990 par rapport au petit GRU (AUC=0.989) et grand LSTM (AUC=0.988), tandis que le grand DenseNet de AUC=0.984, il surpassait les petits LSTM et DenseNet.

S'il n'y a pas de bruit, le grand LSTM et le petit GRU réalisent un AUC similaire de 0.992, tandis que le grand GRU réalise AUC=0.991. Le petit LSTM et le grand DenseNet rapprochent avec des valeurs AUC=0.988 et AUC=0.989 respectivement, par conséquent le petit DenseNet atteindre à AUC=0.983.

On peut conclure que tous les modèles atteignent un niveau élevé de AUC. Donc, chaque architecture fonctionne sur l'optimisation de AUC à tous les niveaux de bruit avec du nombre de paramètres augment.

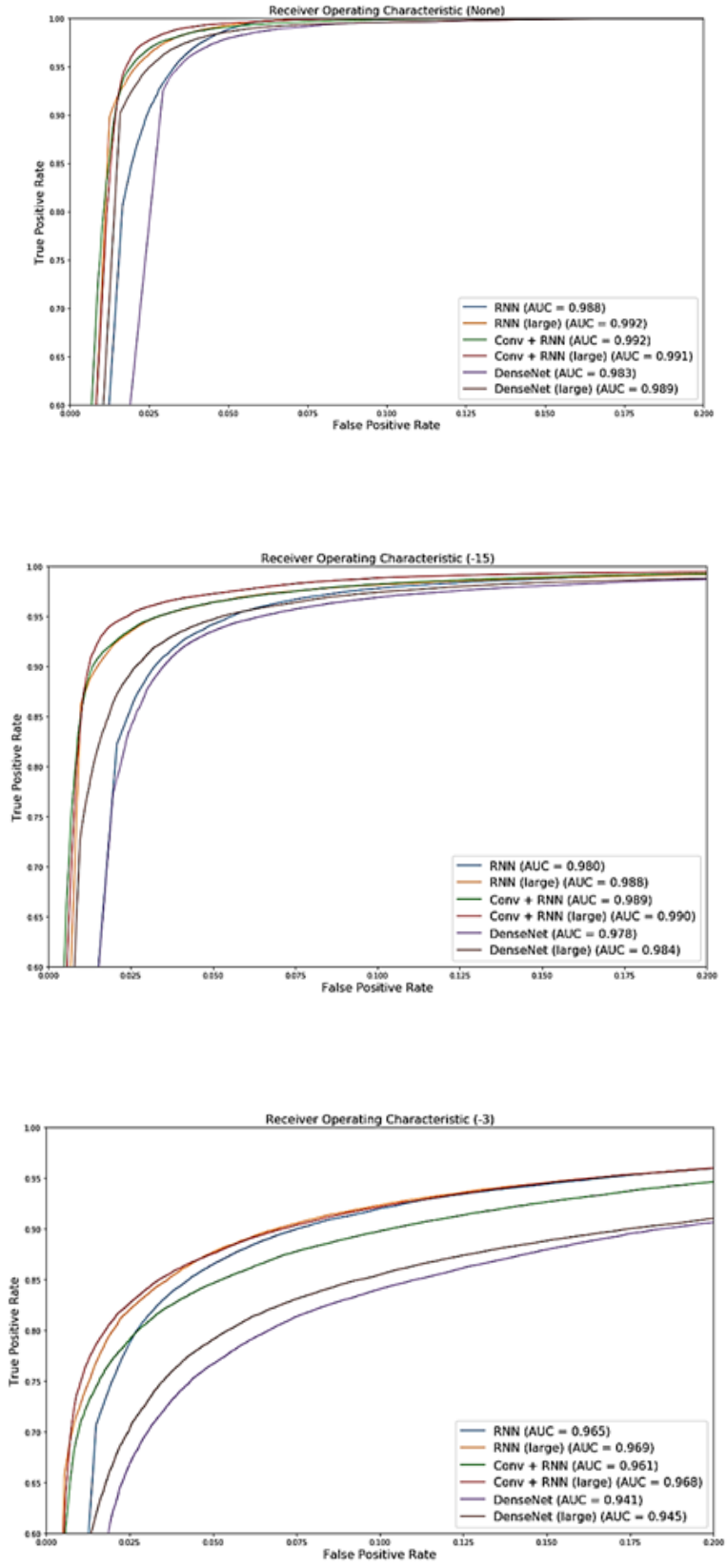


FIGURE 5.8 – Courbes ROC pour chacun des trois niveaux de bruit

TABLE 5.1 – FAR pour FRR fixe à 1% pour chacun des trois niveaux de bruit

	Petit réseau			Grand réseau		
	LSTM-RNN	GRU-RNN	DenseNet	LSTM- RNN	GRU-RNN	DenseNet
FAR (bruit : aucune)	5.11%	4.71%	6.86%	4.34%	**3.61%**	5.75%
FAR (bruit : -15 dB)	17.18%	15.58%	24.45%	16.84%	**11.14%**	23.15%
FAR (bruit : -3 dB)	**48.13%**	63.10%	58.14%	51.75%	58.99%	61.02%

Le tableau 5.1 représente les résultats FAR obtenus à FRR de 1% et à différents niveaux de bruit. Donc, le FAR augmente de manière significative pour tous les modèles avec l'augmentation du bruit. Les résultats sont évalués sur la base de FAR faible.

Dans le cas d'un bruit fort, les LSTM petit et grand surpassent grandement avec FAR de 48.13% et 51.75% respectivement. Le petit GRU réalise une valeur élevée de FAR de 63.10%, tandis que le petit DenseNet réalise une valeur FAR meilleur par apport aux grands GRU et DenseNet.

Dans le cas d'un faible bruit, le grand GRU réalise un FAR de 11.14 % par rapport aux grand et petit LSTM, tandis que le petit DenseNet obtient la pire valeur de toutes les valeurs dans ce cas.

Dans l'absence de bruit, le grand GRU surpasse avec une valeur de FAR égal à 3.16%. Les modèles restants présentent des valeurs semi-rapproche, le petit DenseNet atteint la valeur la plus élevée 6.86%.

Dans les environnements du bruit élevé, on peut dire que les modèles LSTM sont meilleurs que les quatre modèles qui utilisent CNN, car ils fonctionnent mal en raison de la convolution qui peut être plus susceptible de classer le bruit comme son. Contrairement, dans des environnements sans bruit ou avec un bruit faible, l'utilisation du CNN avec RNN donne les meilleurs résultats. En général, l'augmentation du nombre de paramètres à un effet positif sur les performances.

Car le WebRTC s'agit d'un API, nous avons trouvés la précision du WebRTC de près de 98%, sur les niveaux de bruit faible et de sensibilités différents.

5.9 Conclusion

Dans ce chapitre, nous avons présentés les modèles proposés dans cette expérimentation pour évaluer les performances de la DAV dans des environnements bruyants. Nous avons utilisés les deux ensembles de données LibriSpeech et QUT-NOISE. Pour extraire les caractéristiques, nous avons calculés les coefficients MFCC avec Delta.
En raison de l'espace insuffisant dans Google Colab et le fait que cette expérimentation est basée sur GPU, nous n'avons pas atteints les résultats de la performance des modèles. Alors, nous avons essayés de faire des comparaisons sur les résultats obtenus précédemment dans cette expérimentation.

Chapitre 6

Conclusion générale

Dans ce projet, nous avons focalisés sur la détection d'activité vocale qui est basée sur les problèmes les plus importants de la classification binaire, car elle différencie les zones actives et inactives dans des environnements bruyants. Donc, notre objectif principal était de résoudre ce problème en utilisant l'apprentissage profond.

Alors, nous avons commencés notre étude à partir des réseaux de neurones artificielles, ainsi que d'une approche d'apprentissage profond, en clarifiant certaines architectures courantes. Aussi, nous avons présentés la plupart des concepts de base sur les traitements de signale, et lors que la DAV a été étudié pendant plusieurs décennies, nous avons montrés les approches les plus importantes qui ont fonctionnés pour développer la DAV au fil du temps, Jusqu'à l'apprentissage profond.

Dans la mise en œuvre, nous nous sommes appuyés sur trois architectures LSTM-RNN, GRU-RNN et DenseNet où chaque architecture est basée sur deux tailles du réseau, un petit réseau avec 10 000 paramètres et un grand réseau avec 30 000 paramètres.

Nous avons menés des expériences en utilisant LibriSpeech et QUT-NOISE, y compris la détermination de différents niveaux de bruit . Également, nous nous sommes appuyés sur MFCC et leurs dérivés pour extraire les caractéristiques.

Étant donné que la mise en œuvre des expériences est basée sur GPU, et en raison du manque d'espace suffisant dans l'environnement de développement, nous n'avons pas atteint les résultats de la performance des modèles. Par conséquent, nous avons fait des comparaisons sur les résultats qui sont étés précédemment trouvés dans cette expérimentation.

À partir de ces comparaisons, nous avons trouvés qu'une combinaison de CNN et RNN produisait les meilleurs résultats de performance dans des ensembles sans bruit

ou de faibles niveaux de bruit.

Comme une métrique d'évaluation simple, nous avons trouvés la précision du WebRTC de près de 98%, sur les trois niveaux de bruit et de sensibilités différents.

À l'avenir, le R-CNN (Recurrent convolutional neural network) peut être considéré comme une alternative aux réseaux CNN classiques, dans des environnements très bruyants.

Bibliographie

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

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

[3] Ian Goodfellow Y oshua Bengio and Aaron Courville. Deep Learning [Livre]. -[s.l.] : MIT Press, 2016.

[4] Prabhu. Understanding of Convolutional Neural Network (CNN) | Deep Learning [En ligne]. - 2018. - https ://medium.com/@RaghavPrabhu/understandingof-convolutional neural-network- cnn-deep-learning-99760835f148.

[5] Das Siddharth. Architectures CNN : LeNet, AlexNet, VGG, GoogLeNet, ResNet et plus... [En ligne]. - 16 11 2017. - https ://medium.com/analytics-vidhya/cnns-architectures lenetalexnet-vgg-googlenet- resnet-and-more-666091488df5.

[6] Jordan Jeremy. Architectures courantes dans les réseaux de neurones convolutifs. [En ligne]. - 19 04 2018. - https ://www.jeremyjordan.me/convnetarchitectures/.

[7] Hojjat Salehinejad Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. Recent Advances in Recurrent Neural Networks [Revue] // arXiv :1801.01078v3 [cs.NE] . - 2018. - pp. 2-14.

[8] Valentino Zocca Gianmario Spacagna, Daniel Slater and Peter Roelants. Python Deep Learning [Livre]. - Birmingham B3 2PB, UK : Packt Publishing Ltd, 2017. - pp. 165-175.

[9] Gibson Josh Patterson and Adam. Deep Learning A Practitioner's Approach [Livre]. - the United States of America : O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2017. - pp. 150-154.

[10] Rajib Rana Julien Epps, Raja Jurdak, Xue Li, Roland Goecke, Margot Brereton and Jeffrey Soar. Gated Recurrent Unit (GRU) for Emotion [Revue] // arXiv :1612.07778v1 [cs.HC] . - 2016. - pp. 3-4.

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

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

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

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

[15] Aziza Yassamine, « Modélisation AR et ARMA de la Parole pour une Vérification Robuste du Locuteur dans un Milieu Bruité en Mode Dépendant du Texte », Thèse de Magistère, Spécialité : 'Communication', Université Ferhat Abbas-Setif1-UFAS (Algerie), 2013, pages :7-8-9.

[16] Nazaret García Trascasa, « Start-and-End Point Detection at the Input of Speech Recognition Application », Thèse de Master, Spécialité : 'sciences en ingénierie et gestion des télécommunications', Université polytechnique de Catalogne, pages :5.

[17] Amrane Abdessalem et Ould Ammar Kheirreddine, « Nouvelle technique automatique de réduction de bruit acoustique basée sur le principe de séparation aveugle de source », Thèse de Master, Spécialité : Réseau et Télécommunications, Université Saad Dahlab de Blida, 2019, pages :8-9-10.

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

[19] Rezzoug Wahiba et Mankour Amina, « Utilisation de la reconnaissance vocale dans un jeu éducatif pour enfants », Thèse de Master, Spécialité :SIC, Université Abou Bekr Belkaid Tlemcen, 2017, pages : 14-15.

[20] Boukada Yassine et Bemmoussat Mohammed Hamza, « Mise au point d'une application de reconnaissance vocale », Thèse de Master, Spécialité : Réseaux et systèmes distribués, Université Abou Bekr Belkaid-Tlemcen, 2018, pages : 21-23.

[21] Belghitri Karima, « Système sécurisé à base vocale », Thèse de Master, Spécialité : Réseaux et systèmes distribués, Université Abou Bekr Belkaid-Tlemcen, 2015.

[22] Wikipédia Format de fichier audio [Revue]. - 10 11 2011. - pp. 1, 4, 5.

[23] Carmel Lucie Quelques formats de fichiers courants [Revue] // SCI6052 Information documentaire numérique . - 2010. - pp. 4, 5.

[24] Russo David FLAC Format Preservation Assessment [Revue] // British Library / éd. Team British Library Digital Preservation. - 18 01 2018.

[25] R. Johny Elton P. Vasuki, J. Mohanalin Voice Activity Detection Using Fuzzy Entropy and Support Vector Machine Support Vector Machine [Revue] // Entropy / éd. Knuth Kevin H.. - 08 2016, p :1.

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

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

[28] Agnel Waghela Rohan Reddy, Shivangi Rai, Aditya Pawar, Namrata Gharat SUV. Detection Algorithm for Speech Signals [Revue] // International Journal

of Advanced Research in Computer Science and Software Engineering. - 04 2014.

[29] Subramanian Hariharan. Audio signal classification [Revue]. - 2004.

[30] Rao Preeti. Audio Signal Processing [Revue] / éd. Mahadeva Bhanu Prasad and S. R.. - India : Springer-Verlag, 2007.

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

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

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

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

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

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

[37] Julie Séris, « Étude de quelques méthodes de détection d'activité vocale dans des environnements industriels bruités », mémoire présenté comme exigence partielle à l'obtention de la maitrise en génie électrique, Université du Québec, Canada, 10 Avril 2006, pages :32 et 38.

[38] Evgeny Karpov Zaur Nasibov, Tomi Kinnunen, Pasi Fränti. Combining Voice Activity Detection Algorithms by Decision Fusion [Revue]. - Joensuu, Finland : [s.n.], 2011.

[39] https ://en.wikipedia.org/wiki/Likelihood-ratio-test consulter le 05/09/2020.

[40] Le Viet Bac, « Reconnaissance automatique de la parole pour des langues peu dotées », thèse de Doctorat, spécialité Informatique, Université Joseph-Fourier - Grenoble 1, France, 2006, pages : 14-17.

[41] Asbai Nassim, « Identification et Authentification de Locuteurs, par les Techniques de Fusion des Paramètres et des Modèles dans un Environnement Réel », thèse de Doctorat, Electronique : spécialité du Télécommunication, Université des Sciences et de Technologie Houari Boumediene, 2015, page : 48.

[42] Joon-Hyuk Chang, Nam Soo Kim, Sanjit K. Mitra, Life Fellow, Voice Activity Detection Based on Multiple Statistical Models [Revue] // IEEE Transactions On Signal Processing, VOL. 54, NO.. - 6 June 2006.

[43] Xulei Bao and Jie Zhu A novel voice activity detection based on phoneme recognition using statistical model [Revue] // Bao and ZhuEURASIP Journal on Audio, Speech, and Music Processing. - 2012.

[44] Mierle Thad Hughes and Keir. Recurrent Neural Networks For Voice Activity Detection [Revue] // 978-1-4799-0356-6/13/S31.00 ©2013 IEEE ICASSP 2013. -2013.

[45] Phuttapong Sertsi Surasak Boonkla, Vataya Chunwijitra, Nattapong Kurpukdee, and Chai Wutiwiwatcha. Robust Voice Activity Detection based on LSTM recurrent neural networks and Modulation Spectrum [Revue] // Proceedings of APSIPA Annual Summit and Conference 2017. - Malaysia : [s.n.], 12 - 15 December 2017.

[46] Yeonguk Yu Yoon-Joong Kim A Voice Activity Detection Model Composed of Bidirectional LSTM and Attention Mechanism [Revue] // 978-1-5386-7767-4/18/S31.00 ©2018 IEEE . - 2018.

[47] Tianjiao Xu Hao Li, Hui Zhang and Xueliang Zhang Improve Data Utilization with Twostage Learning in CNN-LSTM-based Voice Activity Detection [Revue] // Proceedings of APSIPA Annual Summit and Conference 2019. - Lanzhou, China : [s.n.], 18-21 November 2019.

[48] https ://en.wikipedia.org/wiki/WebRTC consulter le 05/09/2020.

[49] Lu Ma Xiaomeng Zhang, Pei Zhao, Tengrong Su Voice Activity Detection Scheme by Combining DNN Model with GMM Model [Revue] // arXiv :2005.08184v1 [cs.SD]. - 17 May 2020.

[50] Moody James What does RMSE really mean ? [En ligne] // towardsdatascience. - 5 09 2019. - https ://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e.

[51] https ://blog.deepgram.com/what-is-word-error-rate/ consulter le 06/09/2020.

[52] https ://www.webopedia.com/TERM/E/equal-error-rate.html consulter le 06/09/2020.

[53] Narkhede Sarang Understanding AUC - ROC Curve [En ligne] // towards data science. -26 Jun 2018. - https ://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

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

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

[56] https ://www.python.org./ consulter le : 28-09-2020.

[57] https ://le-datascientist.fr/top-10-des-bibliotheques-python-pour-la-data-science consulter le : 29-09-2020.

[58] https ://fr.bitdegree.org/tutos/bibliotheque-python/ consulter le : 29-09-2020.

[59] https ://python-speech-features.readthedocs.io/en/latest/ consulter le : 29-09-2020.

[60] https ://pysoundfile.readthedocs.io/en/latest/ consulter le : 29-09-2020.

[61] https ://fr.qwe.wiki/wiki/Torch-(machine-learning) consulter le : 29-09-2020.

[62] http ://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/ consulter le : 18-10-2020.

[63] Dollar Tsung-Yi Lin Priya Goyal Ross Girshick Kaiming He Piotr. Focal Loss for Dense Object Detection [Revue]. - [s.l.] : Computer Science, IEEE International Conference on Computer Vision (ICCV), 2017.

[64] Ruder Sebastian An overview of gradient descent optimization algorithms [Revue] // arXiv :1609.04747v2 [cs.LG] . - Irlande : [s.n.], 15 Jun 2017.