République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Batna 2 – Mostefa Ben Boulaïd
Faculté de Technologie
Département de Génie Industriel

Thèse

Préparée au sein du Laboratoire d'Automatique et Productique

Présentée pour l'obtention du diplôme de :

Doctorat en Sciences en Génie Industriel

Option : Génie Industriel

INTÉGRATION D'UN MODULE DE RECONNAISSANCE DE LA PAROLE AU NIVEAU D'UN SYSTÈME AUDIOVISUEL - APPLICATION TÉLÉVISEUR

Par

Naima ZERARI

Ingénieur d'état en Informatique et Magister en Génie Industriel

Soutenue publiquement le : 21/04/2021 devant le jury composé de :

M. MOUSS Mohamed Djamel	Professeur	U. Batna 2	Président
M. ABDELHAMID Samir	MCA	U. Batna 2	Rapporteur
M. HARRAG Abdelghani	Professeur	U. Setif 1	Examinateur
Mme BOUFAIDA Zizette	Professeur	U. Constantine 2	Examinateur
M. BOUFAIDA Mahmoud	Professeur	U. Constantine 2	Examinateur
M. MELKEMI Kamal Eddine	Professeur	U. Batna 2	Examinateur
M. BOUZGOU Hassen	Professeur	U. Batna 2	Invité

AVRIL 2021


بسم الله الرحمن الرحيم

REMERCIEMENTS

En premier lieu, je remercie ALLAH le tout puissant qui m'a accordé la volonté, la santé et le courage d'accomplir ce travail.

Je remercie chaleureusement mon directeur de thèse, le Docteur *Samir Abdelhamid*. Ce moment est pour moi, l'occasion de le remercier et lui témoigner ma plus grande gratitude et ma reconnaissance. Je le remercie pour être à l'origine de ce thème de recherche.

Je souhaite aussi exprimer toute ma reconnaissance au Professeur *Hassen Bouzgou* pour ses conseils avisés. Je le remercie de m'avoir encouragé et aidé dans les périodes difficiles et pénibles durant la réalisation de ce travail.

J'accorde une place à part dans ces remerciements au Docteur *Christian Raymond* qui m'a accueilli au sein de l'Institut National des Sciences Appliquées de Rennes (INSA), et qui m'a fourni toute l'aide dont j'avais besoin. Je le remercie de m'avoir donné la chance d'être accueillie au sein de son laboratoire de recherche.

Je voudrais également remercier le Professeur *Mouss Mohamed Djamel* pour l'honneur qu'il m'a fait en acceptant de présider ce jury de thèse. Je remercie très sincèrement les Professeurs : *Harrag Abdelghani*, *Boufaida Zizette*, *Boufaida Mahmoud*, *Melkemi Kamal Eddine* et *Bouzgou Hassen* pour avoir accepté d'examiner ce travail et pour leur présence dans ce jury.

Ce travail a été réalisé au sein du Laboratoire d'Automatique et Productique. Je tiens à exprimer mon respect et ma gratitude vers sa directrice Professeur *Leila Hayet Mouss*.

Je remercie également tous mes collègues du laboratoire et tout particulièrement les Docteurs : *Hanane Zermane*, *Samia Aitouche*, *Karima Aksa* et *Rafik Bensaadi* qui m'ont patiemment écouté parler de mes travaux. Je tiens à souligner leurs conseils et leur bonne volonté qui ont largement contribué à l'aboutissement de ce travail.

A toute personne qui a contribué de près ou de loin à l'élaboration de ce travail, je dis, MERCI.

DÉDICACES

Je dédie ce travail à :

- Mes très chers parents ;

- Mon mari et mes enfants ;

- Ma famille et ma belle-famille ;

- Mes amies.

Naima

ملخص

تقترح هذه الأطروحة تصميم وإنجاز نظام التعرف التلقائي على الكلام الذي يهدف إلى التحكم عن بعد في نظام سمعي بصري -جهاز تلفزيون. ينقسم النظام المقترح "طرف إلى طرف" إلى جزئين: الأول يهدف إلى استخراج أفضل الخصائص من الإشارة الصوتية المدخلة. تحقيقا لهذه الغاية، تم فحص وإختبار العديد من تقنيات إستخراج الخصائص. فيما يتعلق بالجزء الثاني، تم اقتراح العديد من تقنيات التعلم العميق، بهدف التكيف والتعرف على الخصائص المستخرجة لإعطاء نوع الأمر المراد التعرف عليه.

تم التحقق من صحة المنهجيات المختلفة المقدمة في هذه الأطروحة بناءً على مجموعتين من البيانات الحقيقية، الأولى استعملت كتقييم أولي للمنهجيات المقترحة، بينما تم إنشاء الثانية خصيصًا لنظام التعرف التلقائي المقترح في هذه الأطروحة. النتائج التي تم الحصول عليها أكدت فعالية المنهجيات المقترحة.

يبقى التحدي الذي يواجه الأعمال المستقبلية هو تقييم هذا النوع من الأنظمة في ظل ظروف أكثر واقعية مع إشارات صوتية صادرة من بيئات عالية الضجيج.

الكلمات الرئيسية: التعرف التلقائي على الكلام؛ استخراج الخصائص؛ التعلم العميق؛ الشبكات العصبية؛ تصنيف؛ التلفاز.

ABSTRACT

This thesis proposes to design and realize an automatic speech recognition system intended to remotely control an audiovisual system, namely a television. The global "end-to-end" system is divided into two blocks : the first seeks to extract the best characteristics from the input voice signal. To this end, several feature extraction techniques will be examined and tested. With respect to the second block, we propose several deep learning techniques, with the aim to adapt and recognize the characteristics extracted to finally give the class of the utterance.

The validation of the different methodologies presented in this thesis was carried out based on two real data sets, the first being considered for an initial evaluation, whereas, the second is especially created for the ASR system proposed in this thesis. The results obtained confirmed the effectiveness of the proposed approaches.

The challenge for future works is to evaluate this type of system under more realistic conditions with voice signals issued from noisy environments.

Keywords : *Automatic speech recognition ; Features extraction ; Deep learning ; Artificial Neural networks ; Classification ; TV system.*

RÉSUMÉ

Cette thèse propose de concevoir et réaliser un système de reconnaissance automatique de la parole destiné à commander à distance un système audiovisuel à savoir : un Téléviseur.

Le système global "bout en bout" se scinde en deux blocs : le premier cherche à extraire les meilleures caractéristiques à partir du signal vocal d'entrée. A cet effet, plusieurs techniques d'extraction de caractéristiques vont être examinées et testées. Concernant le deuxième bloc, nous mettons en évidence une multitude de techniques relevant du domaine de l'apprentissage profond, dont l'impact est d'adapter et de d'affirmer les caractéristiques extraites pour donner en final la classe de l'énoncé.

La validation des différentes méthodologies présentées dans cette thèse a été effectuée sur la base de deux jeux de données réelles, le premier est tenu compte pour une évaluation initiale, tandis que le second est conçu exclusivement pour le système ASR proposé dans cette thèse. Les résultats obtenus ont certifié l'efficience des approches proposées.

Le défi pour les travaux futurs est d'évaluer ce type de système dans des conditions plus réalistes avec des signaux vocaux issus des milieux bruités.

Mots-clés : *Reconnaissance automatique de la parole ; Extraction des caractéristiques ; Apprentissage profond ; Réseaux de neurones ; Classification ; Système TV.*

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ACRONYMES

Dans un soucis de clarté vis-à-vis de la littérature du domaine, nous allons utiliser les acronymes de la langue anglaise.

ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
BiRNN	Bidirectional Recurrent Neural Network
CAE	Convolutional Auto-Encoder
CHMM	Continues Hidden Markov Models
CNN	Convolutional Neural Networks
DBN	Dynamic Bayes Network
DBeN	Deep Belief Networks
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DL	Deep Learning
DNN	Deep Neural Networks
DTW	Dynamic Time Warping
EBP	Error Back-Propagation
FB	Filter bank
FFNN	Feed-Forward Neural Network
FFT	Fast Fourier Transform
GLSTM	Grid LSTM
GMM	Gaussian Mixture Model
GRNN	General Regression Neural Network
GRU	Gated Recurrent Unit

LISTE DES ACRONYMES

H-LSTM	Highway LSTM
HCI	Human Computer interaction
HMM	Hidden Morkov Model
IDCT	Inverse Discrete Cosine Transform
KNN	k-Nearest Neighbors
LPC	Linear Predictive Coding
LPCC	Linear Prediction Cepstral Coefficients
LSTM	Long Short-Term Memory
MFCC	Mel-Frequency Cepstral Coefficients
MFCC-MT	Multitaper Frequency Cepstral Coefficients
MGB	Multi-Genre Broadcast
ML	Machine Learning
MLP	Multi-Layer Perceptron
MR-WER	Multi-reference Word Error Rate
MSA	Modern Standard Arabic
MSE	Mean Square Error
PLP	Perceptual Linear Prediction
RASTA	RelAtive SpecTrAl
RBM	Restricted Boltzmann Machine
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Networks
SVM	Support Vector Machine
TDNN	Time-Delay Neural Networks
WER	Word Error Rate
ZCR	Zero Cross Rate

CHAPITRE

1

INTRODUCTION GÉNÉRALE

Sommaire

1.1 Contexte et problématique de la thèse

La parole est incontestablement le mode de communication le plus naturel que les humains utilisent pour interagir entre eux. Dans ce contexte, concevoir une machine qui imite le comportement humain, en particulier la capacité d'utiliser la parole d'une manière naturelle et répondre correctement au langage parlé, a attiré l'attention des ingénieurs et des scientifiques durant le dernier siècle [1].

Depuis les années 30, lorsque *Homer Dudley*, des laboratoires Bell, a proposé un modèle de système qui analyse et synthétise la parole [2], le besoin d'utiliser la parole pour concevoir des machines intelligentes, a été progressivement étudié, d'une simple machine qui répond à un petit ensemble de mots, à une machine sophistiquée qui répond au langage naturel couramment parlé et qui prend en compte les conditions variables dans laquelle la parole est produite.

La reconnaissance automatique de la parole (Automatic Speech Recognition : ASR) en tant que outil de modélisation, peut répondre à ce besoin, en raison de ses applications massives qui peuvent être développées pour aider les humains dans leurs tâches quotidiennes. Elle peut être considérée comme une technologie émergente pour permettre et améliorer les interactions homme-homme et homme-machine.

Sur la base des progrès majeurs de la modélisation statistique de la parole dans les années 80, les systèmes ASR se trouvent aujourd'hui dans des tâches qui nécessitent une interface homme-machine, telles que le traitement automatique des appels dans les réseaux téléphoniques et les systèmes d'information basés sur des requêtes telles que les informations de voyage, bulletins météorologiques, etc. [3].

Il existe différents catégorisations de modèles pour les systèmes ASR qui peuvent être catégorisés en fonction : 1) des énoncés, 2) de la taille du vocabulaire et 3) de la dépendance du locuteur [3].

En ce qui concerne les énoncés, les différentes classifications sont données ci-dessous :

- *Mots isolés :* les mots sont prononcés isolément.
- *Mots connectés :* les énoncés à reconnaître sont des séquences de mots isolés d'un vocabulaire spécifique. Dans ce cas, la reconnaissance est basée sur la reconnaissance individuelle des mots isolés.
- *Parole continue :* les énoncés sont prononcés naturellement, c'est-à-dire sans pauses entre phonèmes, syllabes, mots ou phrases.
- *Parole spontanée :* les énoncés ne répondent pas à une question ou à une directive spécifique. Elle représente évidemment la classe la plus difficile à reconnaître.

Quant à la taille du vocabulaire, la précision du système dépend de la complexité et les exigences du traitement. Certaines applications sont conçues pour traiter quelques mots d'autres nécessitent un vocabulaire étendu. Les différentes catégories sont brièvement discutées ci-dessous :

- *Vocabulaire de petite taille :* contenant des dizaines de mots, ce qui signifie que le système a la capacité de reconnaître un nombre limité de mots.
- *Vocabulaire de taille moyenne :* se compose d'une centaines de mots.
- *Vocabulaire de grande taille :* comprend des milliers de mots .
- *Vocabulaire de très grande taille :* englobe des millions de mots ou plus.

La dernière catégorisation concerne la dépendance vis-à-vis du locuteur, où on peut trouver les catégories suivantes :

- *Dépendant du locuteur :* là, le système ASR reconnait de manière unique les caractéristiques d'un seul locuteur.
- *Indépendant du locuteur :* contrairement au mode dépendant du locuteur, ce type de système est destiné à reconnaître différents locuteurs.
- *Adaptation au locuteur :* défini pour améliorer le système indépendant du locuteur pendant que le locuteur utilise le système, il est développé pour adapter son fonctionnement aux caractéristiques des nouveaux locuteurs.

Par rapport à la technique de reconnaissance utilisée, de nombreux systèmes ASR modernes peuvent être conçus et créés à l'aide de plusieurs techniques de classification [4], à savoir : Modèles de Markov caché (Hidden Morkov Model : HMM) [5,6], Déformation temporelle dynamique (Dynamic Time Warping : DTW) [7], Réseaux bayésiens dynamiques (Dynamic Bayes Network : DBN) [8], Machine à vecteurs de support (Support Vector Machine : SVM) [9, 10], K-plus proches voisins (K-Nearest Neighbors : KNN) [11]. La technique de reconnaissance la plus utilisée est celle basée sur les réseaux de neurones artificiels (Artificial Neural Network : ANN) [12, 13].

Récemment, une nouvelle famille de techniques appelée réseaux de neurones profonds a été appliquée avec succès aux différents problèmes d'ASR [14–17].

À partir de la présentation des différentes catégorisations mentionnées ci-dessus, les travaux présentés dans cette thèse se situent comme suit :

- la classe des mots isolés est considérée pour les deux différents problèmes traités (chiffres et commandes TV),
- un vocabulaire de petite taille est utilisé (10 mots dans chaque problème),
- vis-à-vis de la dépendance du locuteur, le système indépendant est celui adopté dans les deux problèmes traités,
- en ce qui concerne la technique de classification utilisée, le système développé s'appuie sur l'emploi des techniques basées sur l'apprentissage profond.

Étant donné ce contexte, deux défis vont être abordés et discutés lors de cette étude :

1. Le premier consiste à choisir la meilleure technique d'extraction des caractéristiques adaptée pour chaque problème de reconnaissance.
2. Le second, revient à choisir la meilleure technique de classification donnant la meilleure performance de reconnaissance.

1.2 Objectifs de la thèse

L'objectif principal de cette thèse est de concevoir et réaliser un système ASR qui a pour but de commander vocalement un téléviseur. Le système proposé se compose de deux blocs de traitement où plusieurs approches vont être étudiées.

Dans le premier, nous allons examiner plusieurs techniques d'extraction qui ont pour but d'extraire les caractéristiques les plus pertinentes afin de bien représenter les signaux de parole.

Dans le second, le problème du choix de la technique de classification va être étudié en examinant différentes architectures de classification qui vont être implémentées et testées avec différentes configurations.

Afin de valider les deux méthodologies proposées ci-dessus, deux jeux de données réelles vont être utilisés. Le premier est considéré en tant que référentiel (Benchmark) pour évaluer et comparer l'efficacité des approches proposées avec quelques travaux utilisant le même jeu de données dans la littérature [18, 19].
Le second jeu de données contenant les différentes commandes vocales TV a été créé avec la participation de plusieurs locuteurs ayant des catégories d'âge et de genre distinctes.

1.3 Structure de la thèse

Le chapitre 1 introduit la thèse avec ses principaux constituants, en mettant l'accent sur l'importance des systèmes de reconnaissance automatique de la parole dans la vie quotidienne de l'être humain et en exposant leurs différentes catégorisations. Aussi, il fournit les grands axes servant à comprendre le contexte des approches proposées et se termine par un aperçu de cette thèse et ses principales contributions.

Le chapitre 2 propose les éléments essentiels à la compréhension du domaine de recherche ASR et le contexte qui lui est associé.

Le chapitre 3 explique en détails les notions, les concepts de base et les paradigmes liés à l'apprentissage machine, en introduisant d'une manière simple le principe de classification. Aussi, ce chapitre présente quelques techniques de classification basées sur l'apprentissage profond.

Le chapitre 4 présente un résumé succinct de l'état-de-l'art des principaux travaux trouvés dans la littérature en se basant soit sur la catégorie du problème à traiter soit sur la technique utilisée. En outre, les infrastructures logicielles existantes ont été énumérées.

Les différentes méthodologies proposées dans cette thèse sont expliquées en détails dans le chapitre 5 avec les résultats obtenus et les discussions.

Enfin, une conclusion générale passe en revue les principales contributions de cette thèse et propose des lignes directrices pour les travaux futurs.

1.4 Contributions de la thèse

Les principales contributions de cette thèse sont :

1. En ce qui concerne le choix de la technique d'extraction, plusieurs méthodologies seront proposées pour extraire d'une manière efficace les caractéristiques pertinentes les plus appropriées au problème considéré dans cette thèse.

2. Quant au choix de la technique d'apprentissage, plusieurs approches issues de l'apprentissage profond pour traiter et adapter la non-uniformité des séquences vocales à savoir : les réseaux de neurones à base de cellules (Long Short-Term Memory : LSTM) et les réseaux de neurones récurrents à portes (Gated Recurrent Unit : GRU) avec différentes configurations (forward, backward et bidirectionnel) vont être proposées où le résultat de traitement est introduit à un classifieur neuronal.

CHAPITRE

2

APERÇU GÉNÉRAL SUR L'ASR

Sommaire

2.1 Introduction

La communication humaine occupe une place privilégiée dans toute société dû à l'utilisation de la parole comme moyen de communication le plus naturel et le plus simple utilisé par l'être humain. A cet effet, la parole peut jouer également un rôle clé dans le développement des interfaces d'interactions homme-machine modernes. Afin de réaliser de telles interfaces, il est essentiel que le processus de communication humain soit imité au niveau de la conception. En conséquence, le domaine de la reconnaissance automatique de la parole est né. Celui-ci fait souvent référence aux sciences et technologies permettant le développement et l'implémentation des algorithmes sur des machines dont le but de les manipuler vocalement.

La recherche dans ce domaine a réalisé de louables avancées au cours des dernières décennies, motivées par les progrès en matière de traitement du signal, d'algorithmes, d'architectures des ordinateurs et du matériel utilisé.

Pour cet objectif, ce chapitre présente essentiellement une introduction inhérente au domaine de la reconnaissance automatique de la parole et ses principales composantes. Une perspective historique sur les inventions clés qui ont permis des progrès dans la reconnaissance vocale est présentée. Ce chapitre met aussi l'accent sur la structure de base d'un système ASR.

2.2 La parole humaine

La parole constitue le mode de communication le plus naturel dans toute société humaine du fait que son apprentissage s'effectue dès l'enfance. La parole se définit comme étant un signal réel, continu, d'énergie finie et non stationnaire, généré par l'appareil vocal humain [20]. Elle offre un moyen facile aux humains pour établir une communication bien claire.

La section 2.2.1 présente globalement quelques principes liés à la production de la parole du point de vue articulatoire et acoustique et dans la section 2.2.2 ceux liés à sa perception.

2.2.1 La production de la parole

La production de la parole est l'une des activités humaines les plus complexes. Ceci, n'est peut-être pas tout à fait surprenant dans la mesure où bon nombre de processus neurologiques et physiologiques complexes sont impliqués dans la génération de la parole.

Elle désigne un phénomène acoustique qui se produit par l'appareil phonatoire faisant intervenir différents organes, en l'occurrence : le diaphragme, les poumons, la trachée, le pharynx, le larynx et les cavités buccale et nasale [20,21].

La production de la parole commence dans le cerveau, où s'effectue la création du message et la structure lexico-grammaticale. Une fois le message créé, une représentation de la séquence sonore et un certain nombre de commandes, à exécuter par les organes de l'appareil phonatoire, pour produire l'*élocution* sont nécessaires. La production physique des sons se fait comme suit :

1. L'air est expulsé des poumons, par la force musculaire qui fournit la source d'énergie, traverse la trachée avant d'arriver dans le larynx où il va rencontrer les cordes vocales. Ainsi, la vibration des cordes vocales produit des vibrations acoustiques représentant les différents sons. La vibration des cordes vocales s'effectue selon deux mécanismes : mécanisme lourd et léger. Le mécanisme lourd est plus particulièrement utilisé par les femmes et les enfants. Alors que le mécanisme léger est essentiellement utilisé par les hommes.

2. Les sons obtenus par la vibration des cordes vocales ne constituent pas encore des mots. A cet effet, une intervention du reste de l'appareil vocal s'effectue pour en devenir un son. Par ailleurs, la transformation du son se réalise dans la cavité du pharynx. Cette dernière, avec les différentes cavités (larynx, bouche et fosses nasales) jouent le rôle de *résonateur* [1].

3. Ensuite, le son *laryngé* [2] est transformé en parole par modulation de différentes manières. Celle-ci est effectuée dans le conduit vocal, grâce aux mouvements de nombreux articulateurs qui peuvent être actifs ou passifs : lèvres supérieures et inférieures, dents supérieures et inférieures, la position de la langue et le voile du palais. Les sons de la parole se distinguent les uns des autres en fonction de l'endroit et de la manière dont ils sont articulés [22].

1. Appareil ou milieu produisant un phénomène de résonance, où certaines fréquences sont amplifiées, d'autres sont atténuées.
2. Voix à l'état brut, telle qu'elle se présente à la sortie des cordes vocales avant de passer dans les différentes cavités de résonance.

La Figure 2.1 illustre les différents constituants de l'appareil phonatoire.

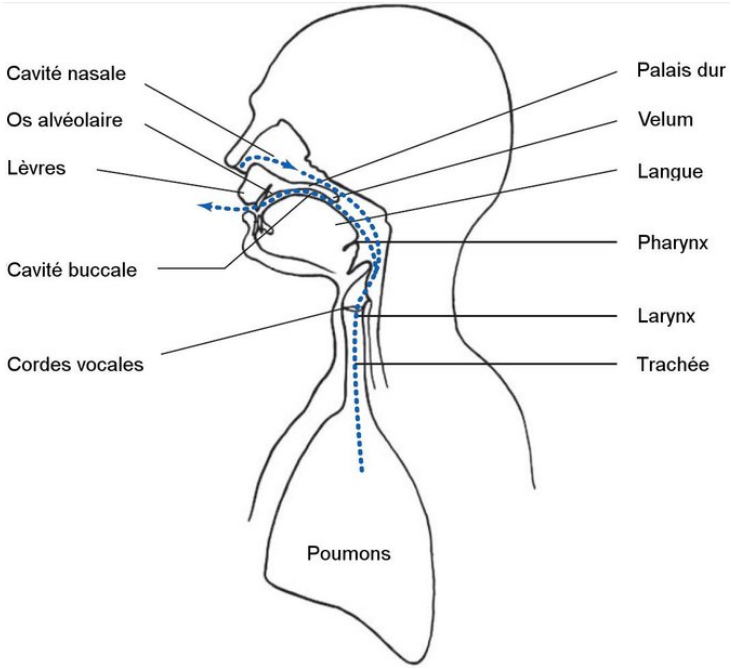


FIGURE 2.1 – Les différents constituants de l'appareil phonatoire.

2.2.2 La perception de la parole

Le système auditif est divisé de manière anatomique et fonctionnelle en trois zones : oreille externe, oreille moyenne et oreille interne, comme indiqué sur la Figure 2.2.

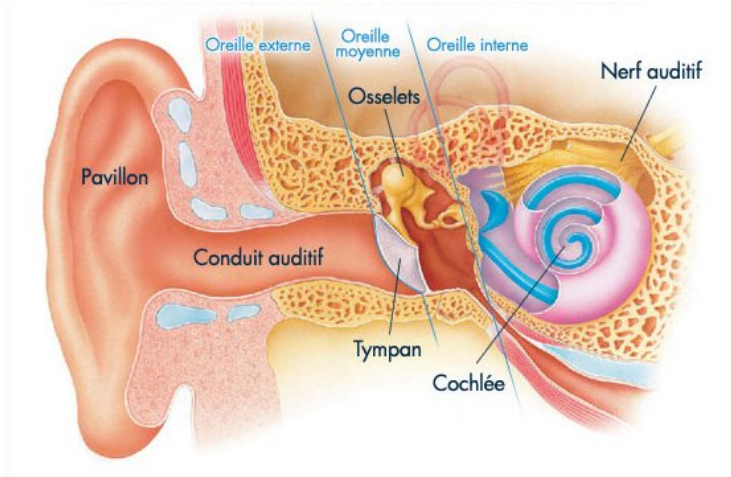


FIGURE 2.2 – Structure de l'oreille humaine.

L'oreille externe est composée du pavillon auriculaire et du canal auditif externe. Le pavillon qui représente la partie la plus visible de l'oreille externe capte

le son, participe à son amplification et le dirige vers le canal auditif externe. Son effet de filtrage permet de sélectionner les sons dans la bande de fréquences de la parole humaine. Il ajoute également des informations directionnelles, indiquant d'où vient le son. Quant à l'oreille moyenne, elle est composée du tympan et d'une cavité remplie d'air (cavité tympanique) permettant également l'amplification du son. La troisième zone désigne l'oreille interne constituée d'un labyrinthe osseux rempli de liquide comportant deux parties fonctionnelles principales : le système vestibulaire et la cochlée. La première composante intervient dans l'équilibre tandis que la seconde possède des capacités d'analyse sonore exceptionnelles, aussi bien en fréquence qu'en intensité. Elle renferme la membrane basilaire dont le rôle est la décomposition des sons selon leurs fréquences. L'oreille interne agit comme un capteur, qui transforme les ondes sonores mécaniques en un signal électrique envoyé au cerveau [22].

La Figure 2.3 décrit les principales étapes de la perception humaine (partie droite) et illustre également la transcription de ces étapes dans le domaine du traitement du signal (partie gauche) [23].

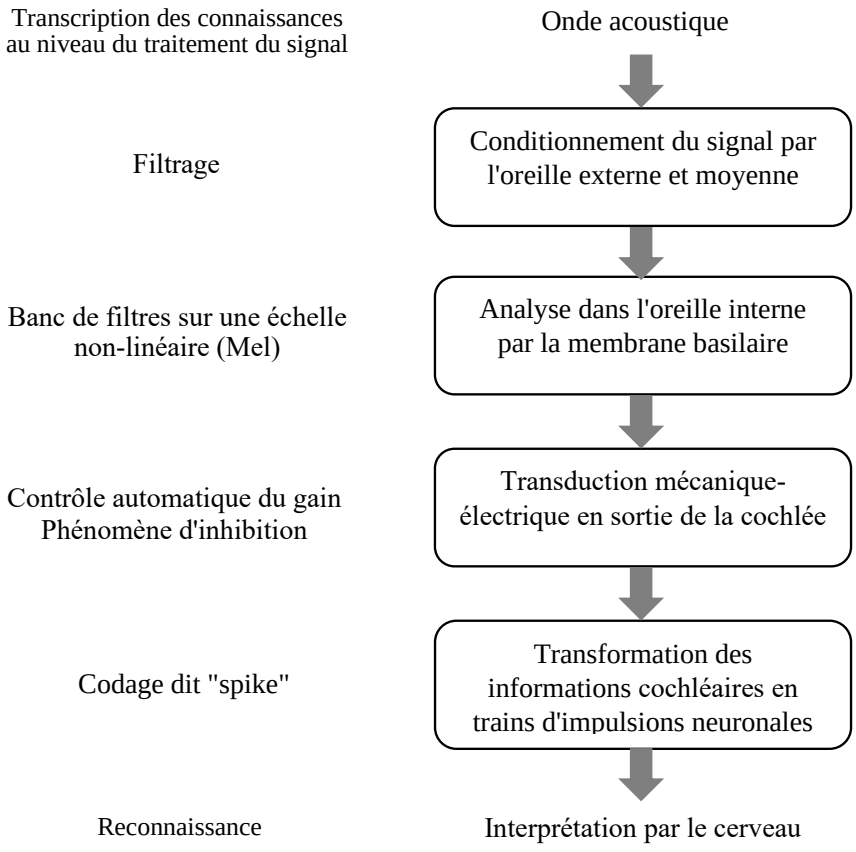


FIGURE 2.3 – Analogie entre perception humaine et machine.

2.3 Paramètres acoustiques du signal de parole

La parole est un processus naturel, variable dans le temps qui peut être directement représenté sous la forme de signal analogique. Ce dernier est un vecteur acoustique porteur d'informations d'une grande complexité, variabilité et redondance.

Analyser un tel signal est une tâche difficile vu le grand nombre de paramètres associés. Néanmoins, trois principaux paramètres s'imposent : la fréquence fondamentale, le spectre fréquentiel et l'énergie. Ces paramètres sont appelés *traits acoustiques* et sont énumérés ci-après [24,25] :

1. **La fréquence fondamentale** ou F0 d'un son est une caractéristique en acoustique propre à chaque personne. Elle est fonction de plusieurs paramètres physiologiques tel que le volume de la glotte et la longueur de la trachée. Elle se définit par la cadence du cycle d'ouverture et de fermeture des cordes vocales pendant la phonation des *sons voisés*[3]. La fréquence fondamentale varie d'un locuteur à un autre selon le genre et l'âge comme suit [26] :
   - de 80 Hz à 200 Hz pour une voix d'homme ;
   - de 150 Hz à 450 Hz pour une voix de femme ;
   - de 200 Hz à 600 Hz pour une voix d'enfant.
2. **Le spectre fréquentiel** est la représentation d'un signal dans le domaine fréquentiel (ensemble de fréquences en progression arithmétique). Une importante caractéristique permettant l'identification de tout locuteur par sa voix nommée *timbre*[4].
3. **L'énergie** correspond à l'intensité sonore. Elle est généralement plus puissante pour les segments voisés de la parole que pour les segments non-voisés.

La Figure 2.4 illustre un exemple réel du signal de parole pour le mot "tash-ghil" dont la signification est "allumer".

3. issus d'une articulation avec une vibration des cordes vocales.
4. Wikipédia : différencie deux sons de même hauteur et de même amplitude.

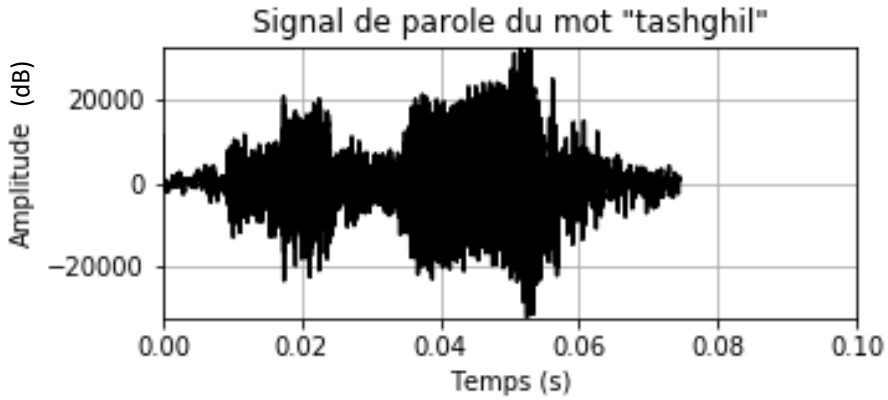


FIGURE 2.4 – Signal enregistré du mot "tashghil" (allumer).

Dans une perspective de reconnaissance, le signal de parole est considéré comme étant un signal très complexe, variable et souvent bruité. Cette complexité du signal est due à différents facteurs en particulier : la redondance, la continuité, les effets de coarticulation, les conditions d'enregistrement et la variabilité intra-locuteurs et inter-locuteurs. Une brève description de ces facteurs est donnée ci-après [27] :

- **Redondance :** le signal de parole présente plusieurs types d'information : les sons, l'identité du locuteur, le genre, l'état émotionnel, la syntaxe et la sémantique des mots prononcés. L'intelligibilité de la parole est remarquablement robuste aux distorsions du signal acoustique. Il a été montré que même si on supprime ou on masque par du bruit des morceaux du signal de parole à intervalle régulier, le signal reste intelligible, ce qui montre une redondance phonétique dans le signal [28]. Cette redondance offre une certaine résistance au bruit, toutefois, elle rend l'extraction des informations pertinentes par un ordinateur plus délicate.

- **Continuité et coarticulation :** la production d'un son dépend fortement du son qui le précède et celui qui le suit en raison de l'anticipation du geste articulatoire. Cette forte articulation des mots rend la tâche de reconnaissance difficile.

- **Conditions d'enregistrement :** L'enregistrement du signal de parole dans de mauvaises conditions rend la tâche d'extraction des caractéristiques pertinentes, nécessaires pour la reconnaissance difficile. En effet, les perturbations transportées par le microphone (le type, la distance, l'orientation) et l'environnement (bruit, réverbération [5]) compliquent amplement la reconnaissance de la parole.

5. Wikipédia : est la persistance du son dans un lieu après l'interruption de la source sonore. La réverbération est le mélange d'une quantité de réflexions directes et indirectes donnant un son confus qui décroît progressivement.

- **Variabilité :** le signal vocal de deux prononciations à contenu phonétique égal est distinct pour un même locuteur (variabilité intra-locuteur) ou pour des locuteurs différents (variabilité inter-locuteur). Ces deux types de variabilités sont expliquées ci-après [29,30] :

  – ***Variabilité intra-locuteur :*** identifie les différences dans le signal produit par un même locuteur. Elle est liée à l'origine biologique de sa production. Le signal de parole ne transmet pas uniquement le message linguistique mais également un grand nombre d'informations sur le locuteur lui-même : genre, âge, origines régionales et sociales, état de santé, état émotionnel et le rythme d'élocution et l'intensité de prononciation (voix normale, voix criée, voix chuchotée). Toutes ces informations dépendent du locuteur et des conditions de prononciation du message.

  La Figure 2.5 montre deux signaux à contenu phonétique égal, prononcé par le même locuteur.

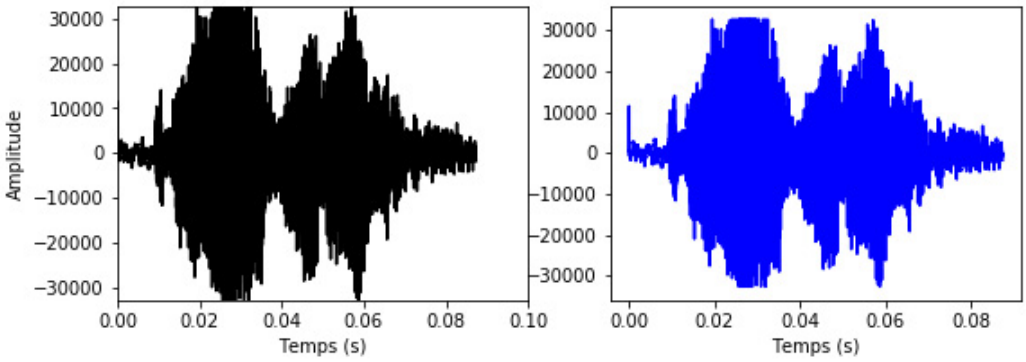
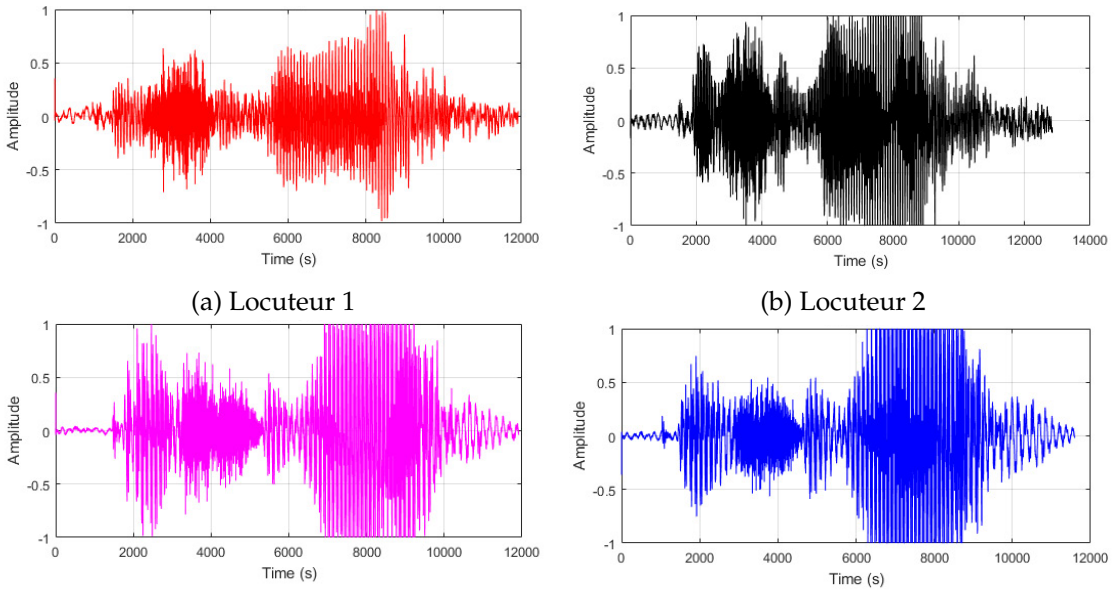


FIGURE 2.5 – Signaux de parole à contenu phonétique égal produit par le même locuteur.

  – ***Variabilité inter-locuteur :*** compte tenu de sa nature physiologique, elle représente un phénomène important dans le domaine de la reconnaissance de la parole. La Figure 2.6 permet de constater les différences d'amplitudes et de durées lors de la prononciation d'un même mot par différents locuteurs.



(a) Locuteur 1

(b) Locuteur 2

(c) Locuteur 3

(d) Locuteur 4

FIGURE 2.6 – Différents signaux du même mot "tashghil" prononcé par différents locuteurs.

A ces difficultés se joint le fait que le signal de parole, suite à sa production transite par un milieu (l'air en premier lieu puis le microphone et le câblage) contenant des perturbations provoquant une détérioration. En effet, on compte plusieurs interactions telles que :

- d'autres sons qui peuvent s'additionner au signal de parole ;
- la forme du signal sonore peut être affectée par la géométrie de la pièce (effet d'écho) ;
- le signal acoustique peut être modifié lors de sa conversion par le microphone.

Ces interactions amplifient d'autant la variabilité du signal de parole et augmentent les difficultés pour le reconnaître.

2.4 La reconnaissance automatique de la parole

La reconnaissance automatique de la parole étant une branche de l'intelligence artificielle, vise principalement à convertir automatiquement le signal de parole en une séquence de mots via un algorithme implémenté sous forme de module logiciel ou matériel. Ainsi, l'objectif de la reconnaissance automatique

de la parole est le développement des techniques et systèmes permettant de recevoir le signal naturel de la parole en entrée et rendre à la sortie sa signification (résultat de la reconnaissance) [31].

Un système ASR est tout système permettant à la machine la compréhension et le traitement des informations fournies oralement par un utilisateur humain.

Les systèmes ASR peuvent être classés en plusieurs catégories différentes selon les types d'énoncés qu'ils sont capables de reconnaître. Ces catégories sont basées sur le fait que l'une des difficultés de l'ASR est la capacité de déterminer quand un locuteur commence et termine un énoncé [32].

- **Reconnaissance des mots isolés :** les systèmes de reconnaissance des mots isolés acceptent un seul mot à la fois. Souvent, ces systèmes ont des états "Ecouter /Ne pas écouter", où ils exigent que le locuteur marque une pause entre les mots. La reconnaissance de mots isolés convient aux situations où le locuteur est tenu à ne donner au système ASR qu'une seule réponse ou des mots isolés désignant des commandes. Ce type de système est celui étudié dans cette thèse.

- **Reconnaissance des mots connectés :** les systèmes de reconnaissance des mots enchaînés permettent de traiter des mots séparés par des pauses. Il sont similaires à ceux des mots isolés, mais il permettent à des énoncés séparés d'être *exécutés ensemble* avec une pause minimale entre eux.

- **Reconnaissance de la parole continue :** Les systèmes de reconnaissance de la parole continue permettent aux utilisateurs de parler presque naturellement et traitent de la parole où les mots sont connectés ensemble au lieu d'être séparés par des pauses. En conséquence, les limites des mots inconnues, la coarticulation et la vitesse d'élocution affectent leurs performances. Ces systèmes sont parmi les plus difficiles à créer, car ils utilisent des méthodes spéciales pour déterminer la limite des mots.

- **Reconnaissance de la parole spontanée :** la parole spontanée peut être considérée comme une parole naturelle dont le contenu n'est pas connu préalablement. Un système ASR traitant de la parole spontanée devrait être capable de gérer une diversité de fonctionnalités de parole naturelles telles que des mots exécutés en même temps (légers bégaiements et des non-mots tels que : "um" , "ah", etc.).

2.4.1 Histoire du développement de l'ASR

Les premières tentatives de mise en œuvre de la reconnaissance automatique de la parole sur la machine ont commencé dans les années 1950 [33].

- Le premier système ASR significatif a été réalisé en 1952 par *Davis*, *Biddulph* et *Balashek* aux laboratoires Bell pour la reconnaissance des chiffres isolés. Avec ce système, un seul locuteur pouvait être reconnu [34].

- Dans les années 1960 et 1970, de nombreuses techniques fondamentales pour l'ASR ont émergé. Ces techniques incluent la transformée de Fourier rapide (Fast Fourier Transform : FFT) [35], l'analyse cepstrale [36] et le codage prédictif linéaire (Linear Predictive Coding : LPC) [37–39] pour l'extraction des coefficients. Alors que, la technique de déformation temporelle dynamique (Dynamic Time Warping : DTW) [40] pour mesurer la similarité entre les séquences qui peuvent varier au cours du temps et la technique des modèles de Markov cachés (Hidden Markov Models : HMM) [41] pour la reconnaissance.

- Dans les années 1980, le problème des mots connectés était au centre des intérêts de cette époque. De plus, l'approche de la reconnaissance des formes a basculé des méthodes à base de modèles aux méthodes de modélisation statistique. En particulier, l'approche HMM a été pleinement étudiée et implémentée par différents laboratoires : Bell [20], CMU [42] et IBM [43]. L'approche HMM était la technique clé introduite au cours de cette période. En sus, une autre technique qui a été ré-introduite à la fin des années 1980, c'était l'idée de réseaux de neurones artificiels. Les réseaux de neurones ont été introduits pour la première fois dans les années 1950 [44], mais n'ont pas produit de résultats notables au départ.

- Depuis les années 1990, les chercheurs ont donné un grand intérêt à la tâche de reconnaissance de la parole continue à grand vocabulaire (Large Vocabulary Continuous Speech Recognition). Pendant ce temps, de nombreuses techniques ont aussi été mises au point tels que les réseaux de neurones récurrents à base de cellules (Long short-term memory : LSTM), proposés en 1997 par *Sepp Hochreiter* et *Jürgen Schmidhuber* [45].

- A partir des années 2000, le domaine de l'apprentissage profond (Deep Learning : DL) fût introduit. Il a relancé l'utilisation des réseaux de neurones en traitement automatique de la parole [46]. En 2007, LSTM a commencé à révolutionner la reconnaissance automatique de la parole, surpassant les modèles traditionnels dans certaines applications du domaine [47]. En 2014, *Kyunghyun Cho* [48] a proposé une variante simplifiée appelée réseaux de neurones récurrents à portes (Gated Recurrent Unit : GRU). Ces

deux variantes des réseaux de neurones récurrents seront appliquées dans les différentes méthodologies proposées dans cette thèse et qui seront présentées dans le chapitre 3.

2.4.2 Approches de la reconnaissance de la parole

En reconnaissance automatique de la parole, généralement trois approches sont met en évidence, en l'occurrence : l'approche globale, l'approche analytique et l'approche statistique [49]. La première considère une phrase ou un mot comme une structure globale à reconnaitre, ceci est réalisé par l'entremise d'une comparaison avec des références (phrase/mot) enregistrées. Tandis que la deuxième, utilisée pour la parole continue, tente d'analyser une phrase en tant que chaîne d'unités élémentaires via un décodage acoustico-phonétique exécuté par des modules linguistiques. Alors que l'approche statistique exploite les niveaux linguistiques en transformant le signal de parole en une suite de vecteurs acoustiques qui vont être considérés comme échantillons d'apprentissage pour construire le modèle de reconnaissance.

2.4.2.1 Approche globale

L'approche globale considère l'ensemble des mots prononcés comme étant une seule unité indépendamment de la langue. Elle considère seulement l'aspect acoustique de la parole. Cette approche est destinée généralement pour la reconnaissance des mots isolés ou enchaînés appartenant à des vocabulaires réduits [27].

Dans les systèmes de reconnaissance globale, une phase d'apprentissage est indispensable, où l'utilisateur prononce l'ensemble des mots du vocabulaire pour son application. Pour chaque mot prononcé, une analyse acoustique est accomplie afin d'extraire les informations pertinentes sous forme de vecteurs de caractéristiques acoustiques qui vont être par la suite à sauvegarder. Ainsi, les méthodes globales associent un ou plusieurs exemples de références acoustiques à chaque mot enregistré.

Pour reconnaitre un nouveau mot prononcé, les caractéristiques acoustiques du mot vont êtres comparées à toute les caractéristiques de référence déjà stockées par l'entremise d'un critère de ressemblance, ainsi, le mot ressemblant le plus au mot prononcé est alors considéré.

2.4.2.2 Approche analytique

L'approche analytique est destinée beaucoup plus aux problèmes de la reconnaissance de la parole continue (grands vocabulaires). Cette approche permet de segmenter le signal de parole en un ensemble de constituants élémen-

taires (mot, phonème, biphone, triphone, syllabe) [6] [1], puis les décoder, et enfin régénérer la phrase prononcée successivement en utilisant des modules linguistique (niveaux lexical, syntaxique ou sémantique). Le processus de l'ASR dans cette approche peut être décomposé en deux opérations :

1. Segmentation du signal de parole sous forme d'une suite de petits segments ;

2. Identification des segments sous forme d'unités phonétiques (Décodage).

2.4.2.3 Approche statistique

Issue de la théorie de l'information, cette approche se base sur le même principe des méthodes globales (avec phase d'apprentissage et de reconnaissance) mais avec l'exploitation des niveaux linguistiques. De ce fait, une analyse acoustique est nécessaire pour transformer le signal de la parole en une suite de vecteurs acoustiques (caractéristiques). Ces derniers sont considérés comme des exemples d'apprentissage pour construire des modèles de reconnaissance qui vont classifier les nouveaux signaux de parole. Cette approche est adoptée dans les différentes méthodologies proposées dans cette thèse.

2.4.3 Structure d'un système ASR

La plupart des systèmes de la reconnaissance automatique de la parole fonctionnent sur des principes probabilistes et formulent le problème comme suit :

Le but d'un système ASR est de trouver la séquence de mots $\hat{W}$ ($\hat{W} = w_1, w_2$,-..., w_k) qui maximise la probabilité à posteriori $P(W/O)$. Cette dernière représente la probabilité que la séquence d'observations acoustiques extraite du signal de parole $O(O = o_1, o_2, ..., o_n)$ génère la séquence de mots W. Cependant, cette probabilité n'est pas quantifiable du moment où un locuteur qui prononce deux fois le même mot, génère deux signaux différents. Ainsi, il est difficile d'estimer cette probabilité à partir d'un jeu de données. A cet effet, le théorème de Bayes est utilisé afin de reformuler cette probabilité comme suit [50, 51] :

$$P(W|O) = \frac{P(W)P(O|W)}{P(O)}, \tag{2.1}$$

La séquence de mot $\hat{W}$ est celle qui maximise l'équation (2.1), comme formulé par l'équation (2.2).

6. phonème : élément sonore du langage parlé, considéré comme une unité distinctive. Biphone : séquence de deux phonèmes consécutifs. Triphone : séquence de trois phonèmes consécutifs. Syllabe : voyelle, consonne ou groupe de consonnes et de voyelles se prononçant d'une seule émission de voix.

$$\hat{W} = \underset{w}{\operatorname{argmax}}\, P(W|O) = \underset{w}{\operatorname{argmax}}\, \frac{P(W)\, P(O|W)}{P(O)}, \tag{2.2}$$

A partir de la formule (2.2), la probabilité de la séquence d'observations acoustiques $P(O)$ n'est pas calculable pour des raisons identiques à celles qui n'ont pas permis de calculer $P(W|O)$. Aussi, la séquence d'observations acoustiques O est la même pour toutes les hypothèses W. Et $P(O)$ représente une valeur constante dont il est possible de l'ignorer pour calculer $\hat{W}$. On a donc :

$$\hat{W} = \underset{w}{\operatorname{argmax}}\, P(W)\, P(O|W). \tag{2.3}$$

La structure générale et les composants de base de tous les systèmes de reconnaissance automatique de la parole aujourd'hui se définit par :

- un analyseur acoustique permettant l'extraction des caractéristiques O,
- un modèle acoustique $P(W|O)$ effectuant le décodage (la reconnaissance).

Ainsi, la structure générale d'un système ASR se base principalement sur les phases fonctionnelles illustrées par la Figure 2.7 et détaillées ci-dessous :

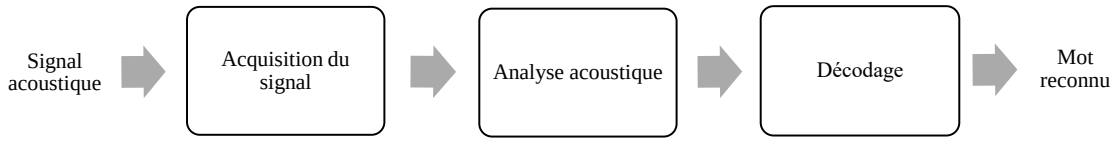


FIGURE 2.7 – Structure générale d'un système ASR.

Le signal acoustique présente dans le domaine temporel, une redondance qui nécessite un traitement préalable à toute tentative de reconnaissance. Le rôle de l'analyse acoustique est d'extraire des paramètres (coefficients ou caractéristiques) les plus pertinents, adaptés à la tâche voulue et ainsi diminuer la dimensionnalité du signal en faisant appel à des traitements appropriés dans le but de réduire le temps de traitement et l'espace mémoire associés. Ces paramètres sont représentés sous la forme d'une suite discrète de vecteurs, appelés communément *vecteurs caractéristiques du signal de parole* [52].

Les sections suivantes expliquent les différentes phases schématisées dans la Figure 2.7.

2.4.3.1 Acquisition du signal de parole

Le signal acoustique de la parole est capturé par un microphone dont la position et la qualité sont importantes pour un bon enregistrement. Pour analyser ce

signal en utilisant un ordinateur, il est fondamental de convertir le signal analogique (continu) en un signal numérique (discret). Cette conversion est effectuée par un convertisseur analogique/numérique à travers les trois étapes suivantes : 1) l'échantillonnage, 2) la quantification et 3) le codage. Le résultat de ce module d'acquisition dépend principalement du matériel utilisé. La Figure 2.8 illustre ces différentes opérations [53].

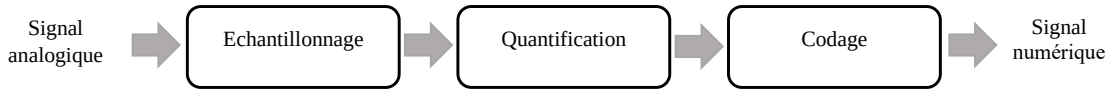


FIGURE 2.8 – Processus de numérisation d'un signal analogique

1. **Échantillonnage** : permet de transformer le signal continu $x(t)$ en un signal discret $x(n)$ définit aux instants d'échantillonnage. Il consiste à prendre des échantillons instantanés du signal à des intervalles de temps régulier, comme le montre la Figure 2.9 (b). La fréquence d'échantillonnage est typiquement de 16 à 20 kHz pour la parole de bonne qualité [53].

   A partir de la Figure 2.9 (b), il est facile de noter que si peu d'échantillons sont considérés, la forme originale du signal ne peut être récupérée. Alors que si l'échantillonnage est effectué à un taux plus élevé, un grand nombre d'échantillons est à gérer, et par conséquent, la contrainte d'un espace mémoire plus élevé est imposée. Un compromis est alors indispensable. Ce dernier est obtenu par l'application du théorème de *Shannon*, qui nécessite l'utilisation d'une fréquence d'échantillonnage F_e définie par :

   $$F_e = 1/T_e. \tag{2.4}$$

   où T_e représente la période d'échantillonnage.

   Afin de pouvoir récupérer le signal d'origine sans distorsion, la fréquence d'échantillonnage doit être supérieure ou égale à deux fois la fréquence la plus élevée présente dans le signal définie par la formule :

   $$F_e \geq 2f_{max}. \tag{2.5}$$

2. **Quantification** : consiste à découper l'amplitude du signal échantillonné en valeurs discrètes comme illustré par la Figure 2.9 (c). La qualité des données prélevées dépend de la fréquence de quantification : plus la fréquence est élevée plus le signal numérique se rapproche de l'analogique.

3. **Codage** : consiste à attribuer le nombre binaire correspondant à toute valeur prélevée (valeur numérique de l'amplitude) au signal lors de la quantification. Ainsi, le choix du nombre de bits sur lequel les valeurs sont codées lors de la quantification reflète la qualité de la numérisation : lorsque

le nombre de bits augmente, la qualité de numérisation augmente également. En outre, un codage de bonne qualité requiert en général 16 bits [53].

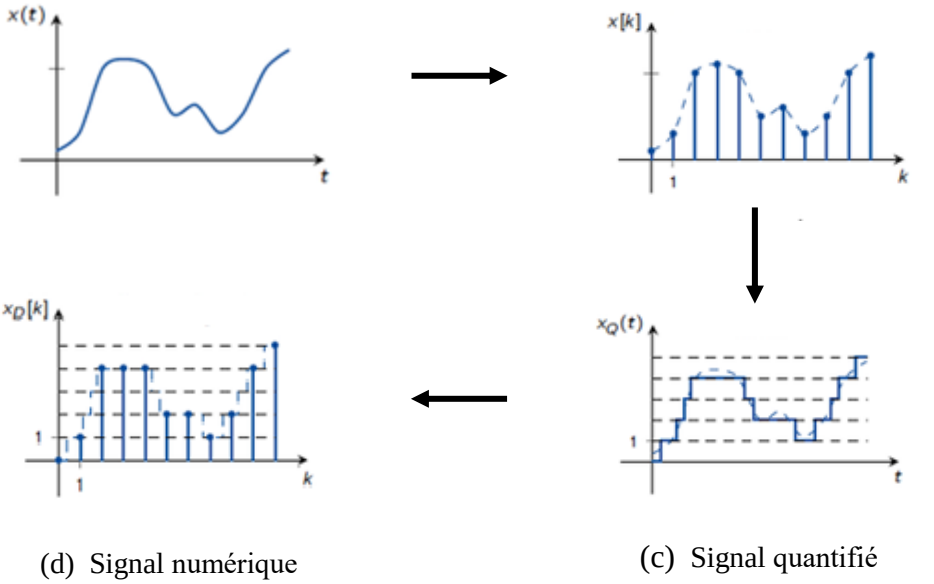


FIGURE 2.9 – Étapes de la conversion analogique-numérique.

A la fin de l'étape de codage, le signal numérique résultant de la conversion du signal analogique est obtenu (voir Figure 2.9 (d)).

Généralement, un ordinateur n'a pas l'habilité de gérer des phénomènes auxiliaires tels que les bruits additifs et les atténuations. Ainsi, l'ASR n'est pas aussi robuste que l'appareil auditif humain. En effet, un pré-traitement du signal est fondamental pour la phase de reconnaissance. IL permet d'améliorer le signal par la suppression du bruit et les distorsions du canal [54].

2.4.3.2 Analyse acoustique

Dans le domaine de l'ASR, la caractérisation de la parole est l'un des domaines d'intérêt les plus importants. C'est également une tâche cruciale car tout le processus de reconnaissance dépend de la qualité de ces caractéristiques. Dans ce contexte, l'analyse acoustique permet d'extraire les vecteurs caractéristiques les mieux adaptés à partir du signal de parole qui doivent être [54] :

- *pertinents* : les vecteurs acoustiques doivent être déterminants pour la solution avec une taille raisonnable pour limiter le coût de leurs calculs dans le module de décodage.

- *discriminants* : les vecteurs acoustiques doivent fournir une représentation caractéristique des sons de base et les rendre aisément séparables.
- *robustes* : les vecteurs acoustiques doivent être insensibles suffisamment aux variations de niveau sonore ou à un bruit de fond.

Quelques détails des techniques d'extraction les plus utilisées seront présentés dans la section 2.5.

2.4.3.3 Décodage

Les traitements de cette phase sont responsables du décodage, aussi appelé reconnaissance de la parole. Ils visent à déterminer le modèle correspondant au mieux au nouveau signal de parole (non utilisé lors de la phase de conception du modèle de reconnaissance).

Différentes approches de décodage sont définies dans la littérature et ont été présentées dans la section 2.4.2. Notons que la plupart des efforts de recherches se penchent actuellement sur l'approche statistique où certaines de ses techniques seront discutées dans le chapitre 3.

2.4.4 Applications de l'ASR

Les applications de l'ASR sont multiples et peuvent variées selon leurs types. Les évolutions des techniques de l'ASR ont permis aux systèmes d'évoluer et d'être de plus en plus efficaces. Aussi, La plupart des systèmes ASR sont des **systèmes dépendants** ou **indépendants** du locuteur. Les systèmes dépendants du locuteur nécessitent une phase d'apprentissage où de nombreuses heures de parole sont généralement indispensables. Cependant, les systèmes indépendants du locuteur ne nécessitent aucune phase d'apprentissage des données et sont souhaitables pour de nombreuses applications où l'apprentissage est difficile à mener. Cette section décrit brièvement les quatre grands types de systèmes qui existent en reconnaissance de la parole [55] :

1. **Commandes vocales :** les systèmes à commandes vocales offrent une interaction entre l'utilisateur et la machine grâce à des commandes vocales, qui s'utilisent généralement dans les systèmes embarqués. Ces commandes représentent des mots isolés que l'utilisateur prononce dans le but d'interagir avec le système. Ce type de système ASR est celui adopté par la présente étude.
2. **Systèmes de compréhension :** principalement, ils permettent de dialoguer avec une machine. Ainsi, l'utilisateur prononce une suite de mots-clés que

le système est capable de reconnaître. A la différence des systèmes à commandes vocales, ce type de système utilise en plus un dispositif de compréhension des mots pour les interpréter et répondre en conséquence. Les systèmes de compréhension utilisent un vocabulaire restreint et un mode indépendant du locuteur. L'utilisation de ces systèmes se limitent habituellement à l'interrogation d'une base de données et de standards téléphoniques automatisés [56].

3. **Systèmes de dictée automatique :** le rôle de ces systèmes est la transcription d'un texte dicté par un utilisateur de la meilleure manière possible. Toutefois, le texte transcrit doit respecter les règles orthographiques et grammaticales propres à la langue considérée. Ce type de système ne prend pas en charge la compréhension du texte à transcrire et qui engendre des erreurs de transcription. Ces systèmes sont fréquemment utilisés pour transcrire des compte-rendus ainsi que des rapports. Dans ces cas, l'utilisateur doit adapter sa locution car il est conscient qu'il s'adresse à un ordinateur.
   Avec l'objectif d'obtention de meilleures précisions en temps réel, ces systèmes à grand vocabulaire sont devenus très dépendants de l'utilisateur. En outre, une phase d'apprentissage nécessitant du temps, est vitale pour permettre au système d'apprendre des modèles spécifiques de la voix de son utilisateur [57].

4. **Systèmes de transcription grand vocabulaire :** l'objectif de ces systèmes est de transcrire des documents audio non préparés par extraction du maximum d'informations de l'enregistrement. Le signal audio étant un signal riche en informations de différentes natures (informations sur les utilisateurs, les frontières des phrases, les zones de musique ou encore les hésitations des utilisateurs) pouvant enrichir la transcription en mots. Un système de transcription grand vocabulaire se compose de plusieurs modules : un module permettant la transcription en mots du signal, un autre module permettant l'extraction des informations additionnelles disponibles dans le signal audio (tel que le module de la reconnaissance automatique du locuteur). Ces systèmes de transcription traitent de multiples documents de diverse nature pouvant être des enregistrements de réunions, d'émissions de télévision, de journaux radiophoniques et de compte-rendus [51, 58].

2.5 Techniques d'extraction des caractéristiques du signal de parole

La redondance et la variabilité du signal de parole ne permettent pas son utilisation directe dans un système ASR, d'où la nécessité d'une analyse acoustique. Cette analyse également appelée extraction de caractéristiques est l'ensemble de méthodes utilisées pour extraire de l'information à partir de ce signal tout en maintenant le pouvoir discriminant du signal et en réduisant sa dimensionnalité [54].

La phase d'extraction des caractéristiques est connue par différentes appellations [23] :

- *Codage* : Cette désignation est certainement héritée des premiers travaux en transmission de la parole. Le codage LPC-10 (Linear Predictive Coding) utilisé en télécommunications en est un exemple édifiant. Cette appellation a continué d'exister avec l'utilisation de LPC en reconnaissance.

- *Extraction des paramètres* : représente l'appellation la plus échangée dans le domaine de l'ASR. Elle désigne l'ensemble des techniques de codage populaires (LPC, Coefficients Cepstraux à Fréquence Mel (Mel-Frequency Cepstral Coefficients : MFCC)), etc. ainsi que l'extraction d'autres paramètres, le pitch [7] en est un exemple.

- *Extraction des caractéristiques* : Cette appellation est proposée ces derniers temps pour le domaine du traitement de la parole. Elle désigne l'ensemble des éléments utilisés pour la classification. Il est nécessaire de préciser la différence avec les paramètres, vu que les caractéristiques sont orientés vers un processus d'apprentissage et non pas vers la reconstruction du signal. Toutefois, l'appellation "extraction de caractéristiques" peut être analogue à celle de "extraction de paramètres". Dans cette thèse, l'appellation *extraction des caractéristiques* est adoptée.
  L'extraction des caractéristiques recherche une représentation du signal de parole appropriée à l'application considérée, communément, le vecteur caractéristiques est constitué de plusieurs caractéristiques, à titre d'exemple nous citons :

  - Vecteur code : représentant le signal de parole issu de la technique employée : MFCC, PLP, LPC, etc. ;
  - Paramètres Δ et $\Delta\Delta$: dérivées première et seconde du vecteur code. Ces paramètres permettent la modélisation de la dynamique du signal de parole ;

7. La fréquence fondamentale du signal vocal perçue par l'oreille.

- La fréquence fondamentale (pitch) ;
- Le taux de passage par zéro (Zero Cross Rate : ZCR) et sa dérivée ;
- Énergie et sa dérivée.

Dans ce qui suit, les analyses, largement utilisées pour la création du vecteur contenant les caractéristiques discriminatifs du signal de parole sont détaillées, tout en mettant l'évidence en particulier sur l'analyse des Coefficients Cepstraux à Fréquence Mel (MFCC), vu sa large utilisation dans le domaine de l'ASR. Parmi ces différentes analyses, nous présentons :

- Analyse par codage prédictif linéaire (Linear Predictive Coding :LPC) [37] ;
- Analyse par coefficients cepstraux de prédiction linéaire (Linear Prediction Cepstral Coefficients : LPCC) [37] ;
- Analyse par coefficients cepstraux à échelle de Mel (Mel Frequency Cepstral Coefficients : MFCC) [59] ;
- Analyse par la prédiction linéaire perceptuelle (Perceptual Linear Prediction : PLP) [60] ;
- Analyse spectrale relative (RelAtive SpecTrAl : RASTA) [61].

2.5.1 Analyse par codage prédictif linéaire

Le codage prédictif linéaire (LPC) est un modèle paramétrique du signal de parole pris du modèle humain de la production de la parole [54]. LPC a été largement usité en particulier dans le traitement du signal vocal depuis son introduction à la fin des années 1960 [62].

Cette technique s'appuie particulièrement sur l'hypothèse que la parole peut être modélisée par un processus linéaire, qui cherche à prédire le signal $s(n)$ à un instant n à partir des p échantillons précédents. Néanmoins, la parole étant un processus non parfaitement linéaire, la somme pondérée du signal sur p pas de temps engendre une erreur qui doit être corrigée par l'introduction du terme $e(n)$ (erreur de prédiction d'ordre p) illustrée par la formule (2.6) [25].

Le codage par prédiction linéaire s'admet alors à déterminer les coefficients a_k (représentant les coefficients de prédiction linéaire d'ordre p) qui minimisent l'erreur $e(n)$, en utilisant un ensemble de signaux constituant les données d'apprentissage. Le choix de l'ordre p est un accord entre précision spectrale, temps de calcul et mémoire de calcul [63].

$$s(n) = \sum_{k=1}^{p} a_k \,.\, s(n-k) + e(n). \tag{2.6}$$

où

$s(n-k)$ représente les k échantillons précédents.

Les différentes étapes qui définissent la technique LPC sont expliquées ci-dessous et illustrées par la Figure 2.10.

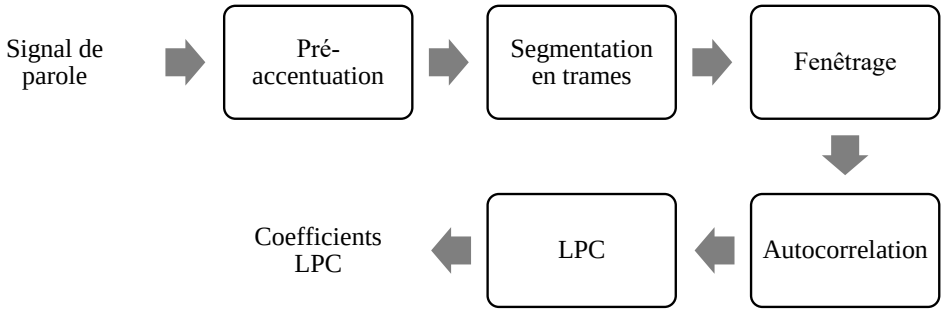


FIGURE 2.10 – Schéma fonctionnel de la technique LPC.

- ***Pré-accentuation*** : augmentation systématique des amplitudes relatives de certaines composantes spectrales du signal de parole pour mieux couvrir le bruit de fond.
- ***Segmentation*** : dans cette étape, le signal est scindé en trames composées de M échantillons, chacune de 20 à 40 ms avec un chevauchement standard de 10 ms entre chaque deux trames adjacentes.
- ***Fenêtrage*** : les trames résultantes sont multipliées par la fenêtre de Hamming afin d'adoucir la transition du signal sur les bords de la trame.
- ***Calcul des LPCs*** : dans cette étape, la méthode d'auto-corrélation est appliquée sur les trames fenêtrées.

2.5.2 Analyse par coefficients cepstraux de prédiction linéaire

La technique des coefficients cepstraux de prédiction linéaire (LPCC) est principalement dérivée de l'analyse prédictive linéaire, où les paramètres LPCCs (les p premiers coefficients cepstraux C_n) sont calculés en utilisant la formule (2.7) suivante :

$$\begin{aligned} c_1 &= a_1, \\ c_n &= \sum_{k=1}^{n-1}(1-\frac{k}{n})a_k c_{n-k} + a_n \qquad 1 < n \leq p. \end{aligned} \tag{2.7}$$

où

c_i : le coefficient cepstre d'ordre i ; a_i : le coefficient prédicteur linéaire.

LPCC a été créée pour répondre aux limites de LPC en délivrant des coefficients moins corrélés à la place de ceux fortement corrélés fournis par LPC. La Figure 2.11 illustre le schéma fonctionnel de l'extraction des LPCCs.

Il à noter que toutes les étapes de calcul des LPCs sont maintenues dans le calcul des LPCCs. Les caractéristiques LPCC sont calculées en introduisant les coefficients cepstraux dans les paramètres LPC [64].

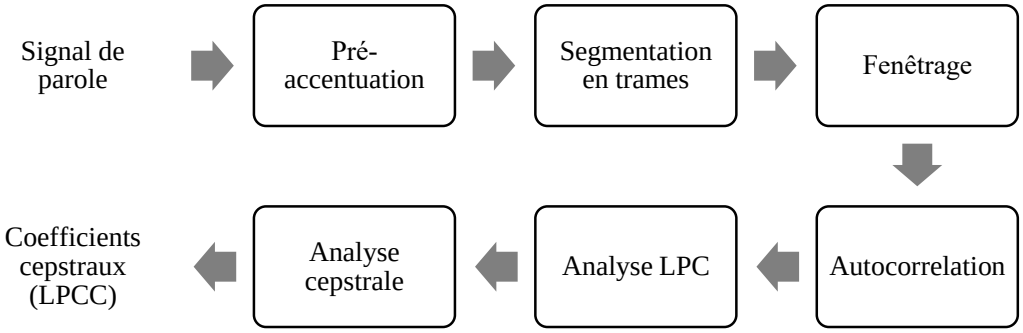


FIGURE 2.11 – Schéma fonctionnel de l'extraction des LPCCs.

2.5.3 Analyse par prédiction linéaire perceptuelle

Une autre analyse, appelée la prédiction linéaire perceptuelle (PLP) développée dans [65], utilise le même principe de base que la technique LPC vu qu'elle utilise le spectre à court terme du signal. En outre, PLP utilise les connaissances issues de la psycho-acoustique [8] du système auditif humain pour optimiser l'utilisation du spectre. Cet aspect a rendu cette analyse plus proche de l'audition humaine ce qui lui a permis de fournir des paramètres plus robustes. Cette technique représente une alternative pour la technique des coefficient cepstraux (MFCC) (cf. section (2.5.5)).

Le processus de calcul des coefficients PLP peut être décrit par la Figure 2.12.

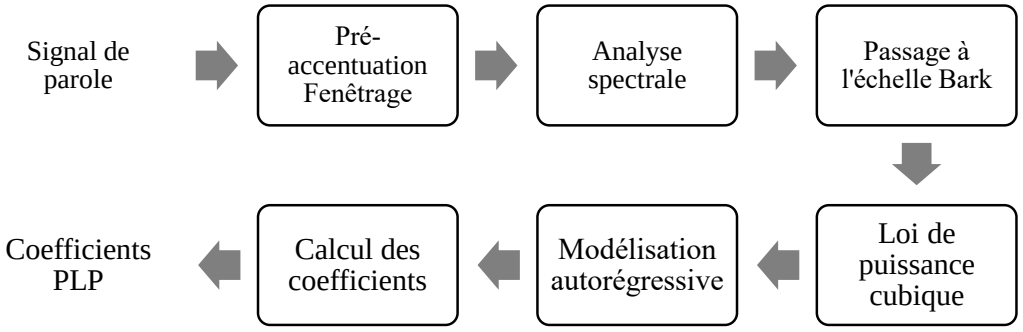


FIGURE 2.12 – Calcul des coefficients PLP.

8. Wikipédia : étudie les rapports entre les perceptions auditives de l'être humain et les sons qui parviennent à ses oreilles.

2.5.4 Analyse spectrale relative

L'analyse spectrale relative (RASTA) est dérivée de l'analyse PLP et proposée dans [61]. La conception de base est de supprimer les variations trop lentes ou trop rapides par filtrage sur le spectre d'amplitude, dans le but de ne retenir que les variations liées au signal produit par l'être humain. Les articulateurs ne peuvent pas bouger trop rapidement, alors si les caractéristiques changent trop rapidement, elles ne sont peut-être pas issues de la parole. De plus, si les caractéristiques changent trop lentement, ce changement ne sera pas perçu.

L'analyse RASTA est souvent combinée avec l'analyse PLP, donnant RASTA-PLP [66], ceci dans le but d'augmenter la robustesse des paramètres utilisés par les systèmes ASR.

2.5.5 Analyse des Coefficients cepstraux à échelle Mel

L'analyse des Coefficients cepstraux à échelle Mel (MFCC) a été présentée par Davis et Mermelstein en 1980 dans [59]. Elle a été exploitée depuis cette date avec succès dans les différentes tâches d'ASR.
Cette analyse s'appuie sur un calcul de coefficients cepstraux à *échelle Mel* qui se rapproche de la perception fréquentielle de l'oreille humaine. L'idée principale est de moyenner le spectre dans des bandes de fréquence correspondant au filtrage effectué par la membrane basilaire.

Afin d'extraire les caractéristiques des signaux de parole, l'analyse MFCC utilise un certain nombre de bancs de filtres Mel, de 15 à 24 filtres triangulaires espacés linéairement jusqu'à 1 kHz et logarithmiquement au-dessus de 1 kHz, pour lisser et capturer les différentes caractéristiques linguistiques du spectre du signal de parole [24].
Les différentes phases de la technique MFCC sont expliquées ci-dessous et illustrées dans la Figure 2.13.

Il est à noter qu'une attention particulière est donnée à cette technique compte tenu de son utilisation dans les différentes expériences de reconnaissance menées dans cette thèse.

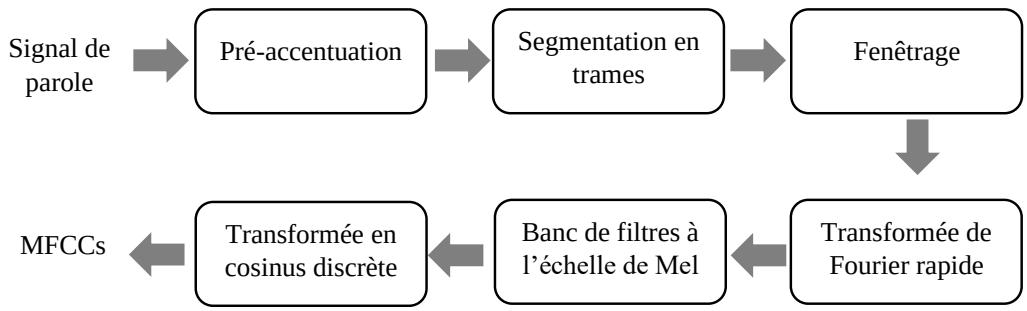


FIGURE 2.13 – Schéma fonctionnel des étapes de calcul des MFCCs.

2.5.5.1 Pré-accentuation

Les recherches menées dans le domaine de l'ASR, ont montré que les segments vocaux tels que les voyelles ont plus d'énergie aux basses fréquences que dans les hautes fréquences ; ceci est causé par la nature de l'impulsion glottale. La pré-accentuation permet d'amplifier l'énergie des hautes fréquences pour les rendre plus appropriées pour le modèle de reconnaissance. Elle est effectuée en faisant passer le signal échantillonné d'origine $x[n]$ dans un filtre passe-haut de premier ordre dont l'équation est la suivante [1] :

$$y[n] = x[n] - \alpha x[n-1]. \tag{2.8}$$

où

$y[n]$ désigne le signal de sortie.

$x[n]$ est la séquence d'échantillons obtenue à partir du signal temporel continu $x(t)$.

α : facteur de pré-accentuation prenant une valeur comprise dans $[0.9, 1.0]$.

2.5.5.2 Segmentation en trames

Le signal de parole représente un processus aléatoire non-stationnaire à long terme, toutefois il est considéré stationnaire dans des fenêtres temporelles d'analyse de l'ordre de 20 à 30 ms. A cet effet, après la phase de pré-accentuation, pour avoir des caractéristiques acoustiques stables, le signal de parole doit alors être divisé en un certain nombre de trames et examiné sur chacune d'elles où la propriété de stationnarité à court terme est vérifiée avec, généralement, un chevauchement de fenêtres de 10 ms. L'intérêt de ce chevauchement est l'obtention d'une continuité temporelle des caractéristiques. De ce fait, à partir de chaque trame, un ensemble de paramètres est dérivé pour former le vecteur caractéristiques [24, 67].

2.5.5.3 Fenêtrage

Le fenêtrage de la trame est utilisé pour minimiser les discontinuités du signal au début et à la fin de chaque trame. Dans le domaine du traitement de la parole, plusieurs types de fenêtres de pondération, également appelées fenêtres d'observation, sont définies dans la littérature et employées, où chaque fenêtre peut être décrite par trois paramètres : sa largeur appelée taille de trame, le décalage entre les fenêtres successives appelé décalage de trame ou chevauchement et la forme de la fenêtre.

Dans le domaine temporel, le fenêtrage consiste à multiplier la valeur du signal $s[n]$ par la valeur de la fenêtre $w[n]$ à l'instant n et fournit le signal de sortie $y[n]$ (cf. équation (2.9)).

$$y[n] = s[n] * w[n]. \tag{2.9}$$

où

$y[n]$ est le signal de sortie à l'instant n.
$s[n]$ est le signal d'entrée à l'instant n.
$w[n]$ représente la fenêtre de pondération employée.

Dans la littérature, de nombreuses fenêtres de pondération sont définies, parmi lesquelles : la fenêtre rectangulaire, la fenêtre de Hann, la fenêtre de Blackman et la fenêtre de Hamming.

- ***Fenêtre rectangulaire :*** est la forme de fenêtre la plus simple. L'équation (2.10) et la Figure 2.14 décrivent cette fenêtre [68, 69].

$$w[n] = \begin{cases} 1 & 0 \leq n \leq N-1, \\ 0 & sinon \end{cases} \tag{2.10}$$

où
N est le nombre d'échantillons dans la fenêtre.

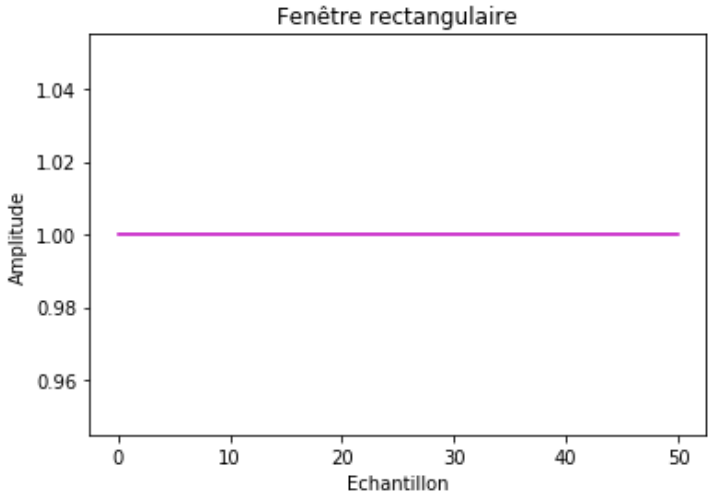


FIGURE 2.14 – Fenêtre rectangulaire.

La fenêtre rectangulaire présente l'inconvénient de la coupure brutale à ses limites dû aux discontinuités qui posent des problèmes lors de l'analyse de Fourier.

- ***Fenêtre de Hann :*** cette fenêtre est décrite par l'équation (2.11) et illustrée dans la Figure 2.15. La fenêtre de Hann réduit complètement les données à zéro au début et à la fin de la trame.

$$w[n] = \begin{cases} 0.50 - 0.50\cos\dfrac{2\pi n}{N} & 0 \leq n \leq N-1, \\ 0 & sinon \end{cases} \tag{2.11}$$

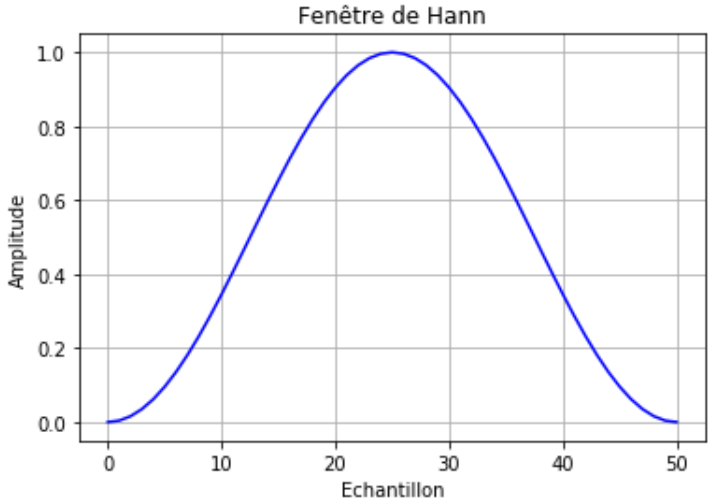


FIGURE 2.15 – Fenêtre de Hann.

- ***Fenêtre de Blackman :*** un autre type de fenêtre est la fenêtre de Blackman illustrée par l'équation (2.12) et la Figure 2.16 respectivement.

$$w[n] = \begin{cases} 0.42 - 0.50\cos\dfrac{2\pi n}{N} + 0.08\cos\dfrac{2\pi n}{N} & 0 \leq n \leq N-1, \\ 0 & sinon \end{cases} \tag{2.12}$$

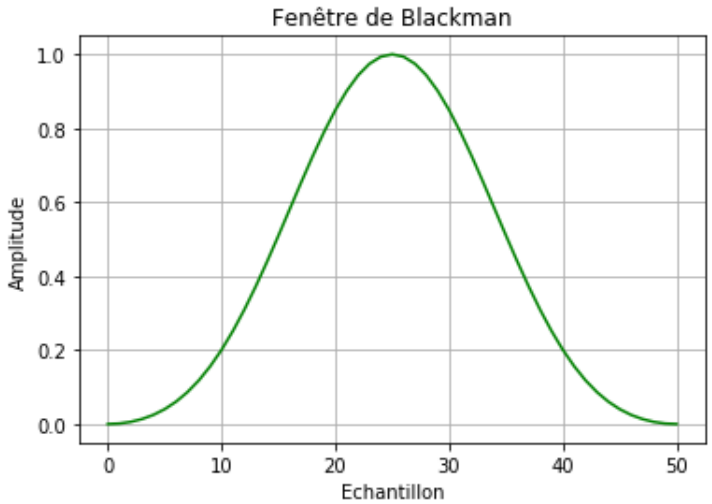


FIGURE 2.16 – Fenêtre de Blackman.

- ***Fenêtre de Hamming :*** une autre forme des fenêtres de pondération est la fenêtre de Hamming. Elle a été proposée par *Richard Wesley Hamming* et peut être considérée comme une forme optimisée de la fenêtre Hann [70]. Elle a pour rôle l'atténuation de la valeur du signal à zéro lorsqu'elle s'approche des bords de la fenêtre pour éviter les discontinuités.
  L'équation (2.13) et la Figure 2.17 illustrent respectivement la fenêtre de Hamming.

$$w[n] = \begin{cases} 0.54 - 0.46\cos\dfrac{2\pi n}{N} & 0 \leq n \leq N-1, \\ 0 & sinon \end{cases} \tag{2.13}$$

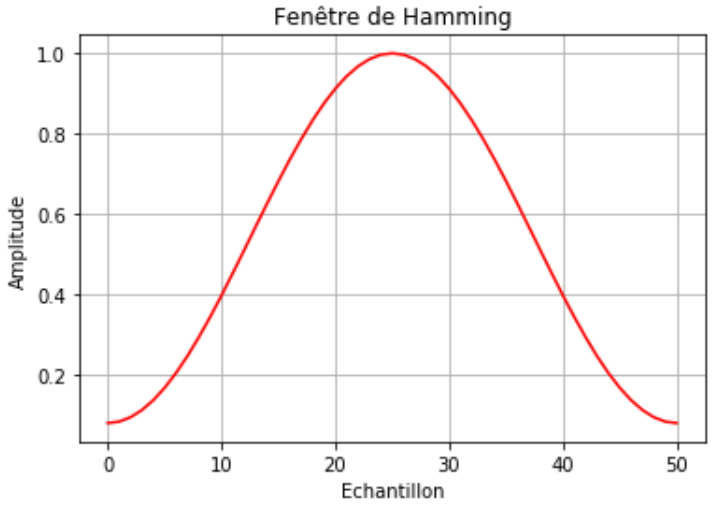


FIGURE 2.17 – Fenêtre de Hamming.

Pour le domaine de l'ASR, différentes études [70–73] ont montré que la forme de fenêtre la plus appropriée est la fenêtre de Hamming. Celle-ci est adoptée par la technique MFCC.

2.5.5.4 Transformée de Fourier discrète

Après avoir effectué un fenêtrage pour atténuer la discontinuité du signal en début et en fin de trame, la phase suivante consiste à appliquer la transformée de Fourier discrète (Discrete Fourier Transform : DFT) qui se calcule via l'algorithme de la transformée de Fourier rapide (Fast Fourier Transform : FFT). Cet algorithme largement utilisé pour évaluer la fréquence du spectre du signal permet de convertir chaque trame fenêtrée de N échantillons du domaine temporel au domaine fréquentiel. La DFT est utilisée pour extraire les informations

spectrales de chaque fenêtre du signal d'entrée. A la sortie de la DFT, une valeur complexe représentant l'amplitude et la phase de chaque composante fréquentielle de la trame est obtenue. Le calcul de la DFT est donné par l'équation (2.14) [1,74].

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N} \qquad k = 0, \ldots, N-1 \tag{2.14}$$

où

X_k est la sortie DFT.

N est le nombre d'échantillons dans la trame.

2.5.5.5 Banc de filtres à l'échelle Mel et Log

L'échelle Mel a été inspirée pour la première fois par *Stevens et Volkman* en 1937 [75]. Elle modélise la membrane basilaire et redistribue les fréquences en fonction de la fréquence perçue (voir Figure 2.3) [23,68].

Le banc de filtres à l'échelle Mel est constitué d'une série de filtres passe-bande de forme triangulaire qui se chevauchent dont l'objectif est la réduction de la taille des caractéristiques impliquées. Ce type de structure de filtre est largement utilisé pour la modélisation spectrale auditive dans le domaine du traitement de la parole en particulier dans le cadre du calcul des coefficients cepstraux de fréquence Mel [76]. Chaque réponse de filtre commence à une valeur d'amplitude nulle à l'extrémité inférieure et augmente linéairement jusqu'à la fréquence centrale, puis décroît linéairement à zéro à son extrémité supérieure. Les filtres sont disposés de telle sorte que le premier filtre commence à une fréquence nulle et se termine à la fréquence centrale du filtre suivant. Ensuite, le deuxième filtre commence à la fréquence centrale du filtre précédent et se termine à la fréquence centrale du filtre suivant, etc. La Figure 2.18 illustre la forme générale du banc de filtres à l'échelle Mel.

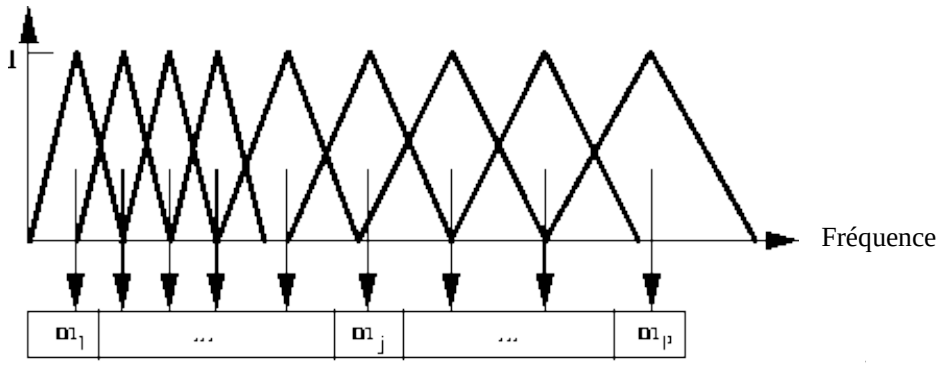


FIGURE 2.18 – Banc de filtres à l'échelle Mel.

Les bancs de filtres à l'échelle Mel convertissent la puissance du spectre obtenu à partir de la transformation FFT sur l'échelle Mel en utilisant l'équation (2.15).

$$Mel(f) = 2595 * \log_{10}(1 + f/700) \tag{2.15}$$

où

f représente la fréquence dans l'échelle linéaire et $Mel(f)$ est celle perçue.

La relation entre la fréquence en Hertz et celle de l'échelle Mel est linéaire en dessous de 1000 Hz et logarithmique au dessus de 1000 Hz comme illustré dans la Figure 2.19 [76].

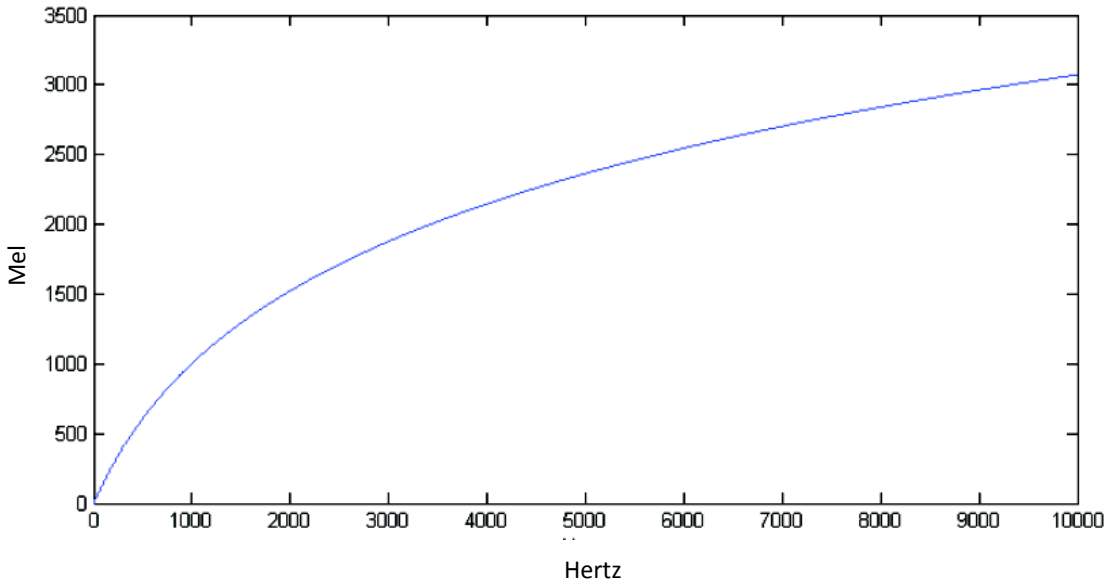


FIGURE 2.19 – Relation entre la fréquence en Hertz et en échelle Mel.

Enfin, l'utilisation de l'opérateur *Log* rend les estimations des coefficients moins sensibles aux variations d'entrée, telles que les variations dues au rapprochement ou à l'éloignement de la bouche du haut-parleur du microphone [76]. Les coefficients log du banc de filtres à l'échelle Mel (Filter bank : FB) peuvent être calculés à partir des sorties des filtres par l'équation (2.16) :

$$S(m) = 20\, log_{10}\left(\sum_{k=0}^{N-1} |X(k)|H(k)\right), \qquad 0 < m < M \tag{2.16}$$

où,

M est le nombre de filtres à l'échelle Mel de 20 à 40.

$X(k)$ est la FFT de la trame.

$H(k)$ est la fonction de transfert du filtre Mel.

Les coefficients obtenus à la sortie des filtres peuvent être utilisés directement pour la reconnaissance de la parole, néanmoins, d'autres coefficients plus discriminatifs, plus robustes au bruit et bien particulièrement décorrélés entre eux sont privilégiés obtenus en utilisant la transformation en cosinus discrète (cf. 2.5.5.6).

2.5.5.6 Transformée en cosinus discrète

La transformation en cosinus discrète (Discrete Cosine Transform : DCT) est une transformation mathématique linéaire ayant la capacité de générer des coefficients décorrélés et de concentrer la majeure partie de l'énergie du signal dans un nombre réduit de coefficients. La DCT convertit le signal du domaine fréquentiel au domaine temporel, avec la possibilité de le reconvertir dans le domaine fréquentiel en utilisant la DCT inverse (Inverse Discrete Cosine Transform : IDCT) [77]. Les coefficients c_n sont calculés par l'équation (2.17) [78] :

$$C(n) = \sum_{m=0}^{N-1} S(m) \cos\left(\pi n \left(m - \frac{1}{2}\right) / M\right), \quad 0 \leq n \leq M \tag{2.17}$$

où
$C(n)$: les coefficients MFCC.
S_m : spectre logarithmique.
N : le nombre d'échantillons dans chaque trame.
M : le nombre des bancs de filtres.

Cette transformation dé-corrèle les sorties du banc de filtres à échelle Mel, et les premiers coefficients sont concaténés pour former le vecteur de caractéristiques MFCCs comme le montre la Figure 2.20.

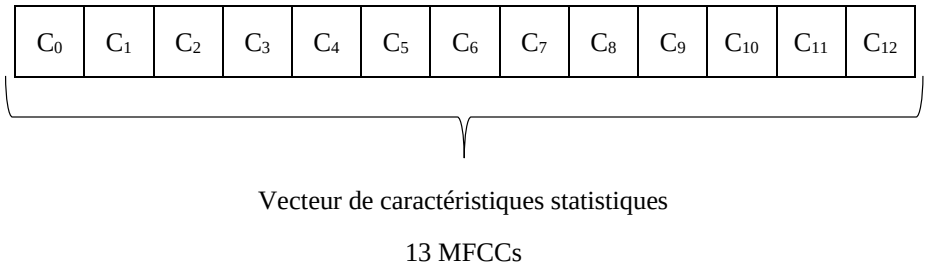


FIGURE 2.20 – Les caractéristiques statiques MFCCs.

2.5.5.7 Coefficients Delta et delta-delta

Bien que les vecteurs de caractéristiques statiques comme les MFCCs fournissent une bonne estimation des spectres locaux, ils ne parviennent pas à capturer les aspects dynamiques de la parole humaine. Ces derniers sont très im-

portants pour distinguer les différentes prononciations. Les performances d'un système ASR peuvent être considérablement améliorées en ajoutant des dérivées temporelles aux paramètres statiques de base. Pour cet objectif, Furui [64] a proposé l'utilisation des paramètres dynamiques qui permettent d'introduire une information sur la dynamique temporelle du signal. En particulier, il a proposé les dérivées du premier ordre et les dérivées du second ordre. Les dérivées du premier ordre aussi appelées *coefficients delta*, sont issus des coefficients cepstraux, alors que les dérivées du second ordre, appelées *coefficients delta-delta* sont issues des coefficients delta. Les coefficients delta sont calculés utilisant l'équation (2.18) [68,79] :

$$\Delta C_i = \frac{\sum_{k=-N}^{N} k\, C_i(n+k)}{\sum_{k=-N}^{N} k^2}. \tag{2.18}$$

où

ΔC_i représente le coefficient delta calculé à la $n^{\text{ième}}$ trame pour le $i^{\text{ième}}$ coefficient cepstral C_i.

N est le nombre de trames à travers lesquelles le cepstre de delta est calculé avec des valeurs typiques égales à 2 ou 4 [68,79].

De même, les coefficients delta-delta peuvent être calculés en utilisant la même équation mais en utilisant les coefficients delta au lieu des coefficients cepstraux originaux (MFCCs).

L'équation (2.18) peut être simplifiée en fixant $N = 1$ et en ignorant le dénominateur qui donne l'équation (2.19) utilisée dans de nombreuses applications [68].

$$\Delta C_i = C_i(n+1) - C_i(n-1). \tag{2.19}$$

Les coefficients dynamiques delta et delta-delta sont traditionnellement concaténés avec les coefficients statiques pour former un vecteur caractéristiques unique contenant à la fois les informations statiques et dynamiques dans le signal de parole.

$$x_k = \begin{pmatrix} c_k \\ \Delta c_k \\ \Delta\Delta c_k \end{pmatrix}$$

Cette paramétrisation des caractéristiques est illustrée sur la Figure 2.21.

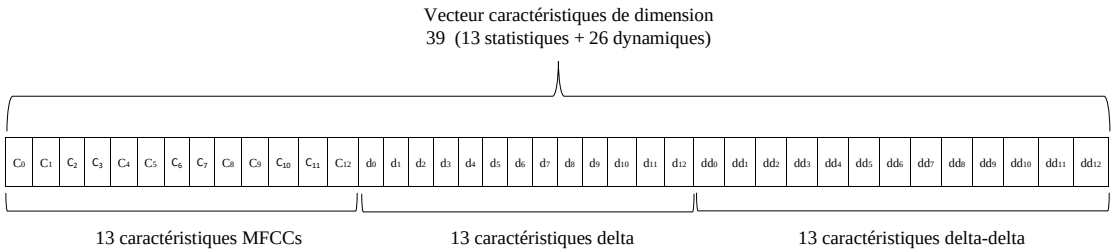


FIGURE 2.21 – Concaténation des caractéristiques statiques et dynamiques.

Une étude comparative des différentes représentations du signal, LPC, LPCC et MFCC a montré que le codage MFCC est considéré comme la technique de codage la plus utilisée, ce qui la rend la technique de référence dans un grand nombre d'applications de traitement de la parole [59].

2.6 Mesures de performance

L'évaluation des performances de reconnaissance des systèmes ASR doit être mesurée sur des données différentes de celles utilisées pour l'apprentissage. Les performances des systèmes ASR peuvent être évaluées principalement avec trois mesures : la précision de la reconnaissance, la complexité et la robustesse [80]. Ces mesures sont expliquées ci-dessous :

2.6.1 Précision de reconnaissance

La précision de la reconnaissance est la mesure la plus importante et la plus simple des performances des systèmes ASR. Pratiquement, les données vocales collectées sont partitionnées en ensemble d'apprentissage et ensemble de test. L'ensemble d'apprentissage, qui contient généralement la plupart des données disponibles, est utilisé pour l'estimation des paramètres des modèles acoustiques. Les données restantes forment l'ensemble de test, qui est utilisé pour mesurer les performances du système développé sur de nouveaux signaux non vus pendant l'apprentissage.

2.6.2 Complexité

La complexité est un autre problème qui doit être pris en compte dans la plupart des systèmes ASR. En général, la complexité d'un système ASR fait référence aux complexités de calcul et du modèle. La complexité de calcul concerne le temps d'exécution dans chaque module du système. Pour la plupart des implémentations pratiques où la tâche de reconnaissance doit être terminée en

temps réel, la complexité de calcul doit certainement être bien prise en compte. Tandis que la complexité du modèle est généralement mesurée par le nombre de paramètres distincts du modèle.

2.6.3 Robustesse

Bien que la précision est cruciale pour les performances de la reconnaissance automatique, la robustesse est également d'une grande importance pour les systèmes ASR. À l'heure actuelle, la plupart des systèmes ASR sont entraînés sur un ensemble d'exemples (échantillons) de parole collectés dans certaines conditions prévues. Ils fonctionneraient bien si les conditions de fonctionnement correspondent aux conditions prévues. Les aspects importants des conditions de fonctionnement comprennent le niveau de bruit de fond, le bruit et la distorsion du canal, la différence de locuteur, le style de parole et l'écart syntaxique, la spontanéité de la parole, etc. En pratique, l'écart de ces conditions par rapport à celles supposées lors de la phase de conception peut entraîner une dégradation substantielle des performances.

2.7 Conclusion

Ce chapitre a tenté d'introduire le domaine de la reconnaissance automatique de la parole en présentant initialement la parole humaine comme acteur principal, ensuite les caractéristiques qui sont liées à la difficulté de sa reconnaissance sont présentées, puis l'architecture principale des systèmes ASR est détaillée avec une variété d'applications des systèmes ASR qui sont mises en exergue. Par la suite, un bref panorama des techniques d'extraction des caractéristiques du signal de parole les plus utilisées : LPC, LPCC, PLP et MFCC, et qui décrivent fidèlement les propriétés les plus pertinentes, ont été discutées. En fin du chapitre, les standards de mesures de performances les plus utilisés par la communauté sont discutés et analysés.

CHAPITRE

3

CONCEPTS FONDAMENTAUX DE L'APPRENTISSAGE MACHINE

Sommaire

3.1 Introduction

Ce chapitre donne un aperçu général sur les concepts fondamentaux de l'apprentissage machine, définit étant une discipline scientifique qui s'intéresse à découvrir et à apprendre des relations intrinsèques dans et à partir des données, c'est-à-dire extraire des informations, découvrir des modèles, prédire des informations manquantes sur la base des données observées.

Ce chapitre se focalise principalement sur la présentation du processus d'apprentissage machine et ses différents paradigmes définis dans la littérature en mettant l'accent sur la technique de classification dans le contexte de l'apprentissage profond, plus précisément dans le cadre des applications en relation avec le traitement automatique de la parole devenu de plus en plus important ses dernières années dans plusieurs thématiques de recherche et applications industrielles. Ceci afin de proposer une solution à la problématique exposée dans cette thèse, à savoir la reconnaissance automatique des commandes TV vocales.

3.2 Apprentissage humain

L'apprentissage permet à l'être humain d'avoir de la flexibilité dans sa vie quotidienne ; il lui permet de s'adapter aux nouvelles circonstances et d'apprendre de nouvelles astuces. Les parties importantes de l'apprentissage de l'être humain dans ce contexte sont :

- La mémorisation : permettant de reconnaitre que la dernière fois qu'il était dans cette situation (données déjà vues).
- L'adaptation : il a essayé une action particulière qui a donné un résultat.
- La généralisation : si le résultat est correct, il va donc l'essayer à nouveau, dans le cas échéant, il va essayer quelque chose de différent. Le concept de généralisation concerne la reconnaissance de la similitude entre différentes situations, afin que les choses qui s'appliquent à une situation puissent être utilisées dans une autre. C'est ce qui rend l'apprentissage utile, car il est possible d'utiliser les connaissances dans de nombreuses situations différentes [81].

Afin d'imiter le fonctionnement du cerveau humain, les scientifiques ont essayé d'implémenter le même mécanisme dans le but d'intégrer une certaine intelligence à la machine, d'où l'apprentissage machine.

3.3 Apprentissage machine

Pour résoudre un problème sur un ordinateur, il est fondamental d'avoir un algorithme. Celui-ci se définit étant une suite d'instructions qui doit être exécutée pour transformer l'entrée en sortie. A titre d'exemple, soit un algorithme de tri. L'entrée de cet algorithme est un ensemble de nombres et la sortie est leur liste ordonnée. Pour cette tâche de tri , il peut y avoir différents algorithmes où il est intéressant de trouver le plus efficace, nécessitant le moins d'instructions ou de mémoire ou les deux à la fois [82].
Pour certaines tâches, cependant, il n'existe pas d'algorithme, mais il existe des exemples de données.

Actuellement, grâce aux progrès des technologies informatique et électronique, de grandes quantités de données peuvent être stockées et traitées, ainsi que d'y accéder à partir d'emplacements physiquement éloignés sur un réseau informatique. La plupart des appareils d'acquisition de données sont désormais numériques et enregistrent des données fiables.
Soit par exemple, une chaîne de magasins qui compte des centaines de magasins dans tout le pays vendant des milliers de produits à des millions de clients. Les terminaux de point de vente enregistrent les détails de chaque transaction : date, code d'identification client, marchandises achetées et leur montant, montant total dépensé, etc. Cela équivaut généralement à des gigaoctets de données chaque jour. Cette chaîne de magasin souhaite prédire qui sont les clients potentiels d'un produit. Encore une fois, l'algorithme pour cela n'est pas évident ; il change dans le temps et selon la situation géographique. Les données stockées ne deviennent utiles que lorsqu'elles sont analysées et transformées en informations pourront être utilisées pour faire des prédictions. C'est dans ces cas qu'intervient l'apprentissage machine [82].

L'apprentissage machine (Machine Learning : ML) aussi appelé apprentissage artificiel ou apprentissage automatique est un domaine de recherche de l'intelligence artificielle et plus généralement de l'informatique, comme illustrée par la Figure 3.1, qui implique la recherche et le développement des programmes informatiques qui s'améliorent automatiquement sur la base de leurs expériences [83].

Il est définit dans [84] comme étant la science qui permet aux ordinateurs d'apprendre sans être explicitement programmé. Une autre définition de l'apprentissage machine plus moderne est donnée dans [83] en exprimant qu'une machine peut apprendre lorsque sa performance à réaliser une certaine tâche s'améliore avec de nouvelles expériences. Par ailleurs, l'objectif de l'apprentissage machine est de créer des modèles qui apprennent par le biais des exemples : il s'appuie sur l'utilisation des données numériques (résultats de simulations ou de mesures). A l'issue de l'apprentissage à partir d'exemples, le modèle construit doit être capable de généraliser, c'est-à-dire, capable de fournir un ré-

sultat correct, avec des données qu'il n'a pas vu durant l'apprentissage. En particulier, l'apprentissage machine peut se définir comme un ensemble de techniques qui peuvent détecter automatiquement les modèles dans les données, puis utiliser ces modèles pour prédire de nouvelles données [85].

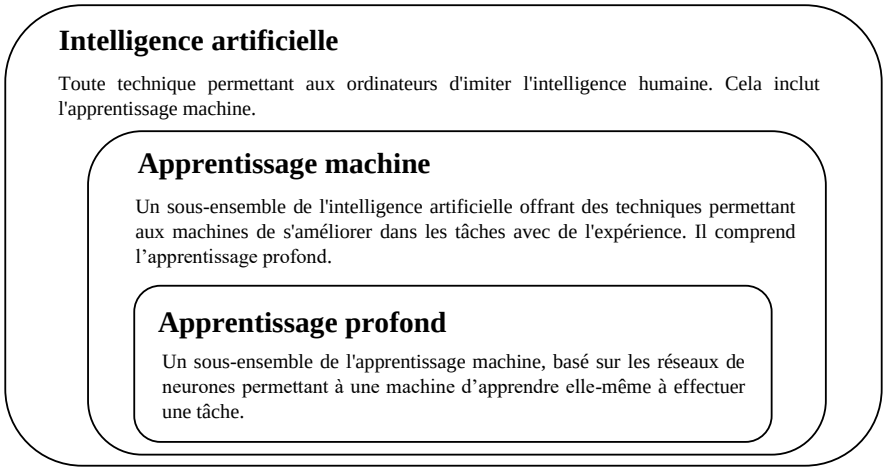


FIGURE 3.1 – Intelligence artificielle, apprentissage machine et apprentissage profond.

3.3.1 Processus de l'apprentissage machine

Cette section examine brièvement le processus d'apprentissage machine, illustré par six étapes, par lequel les algorithmes d'apprentissage machine peuvent être sélectionnés, appliqués et évalués [81]. La Figure 3.2 explique ce processus.

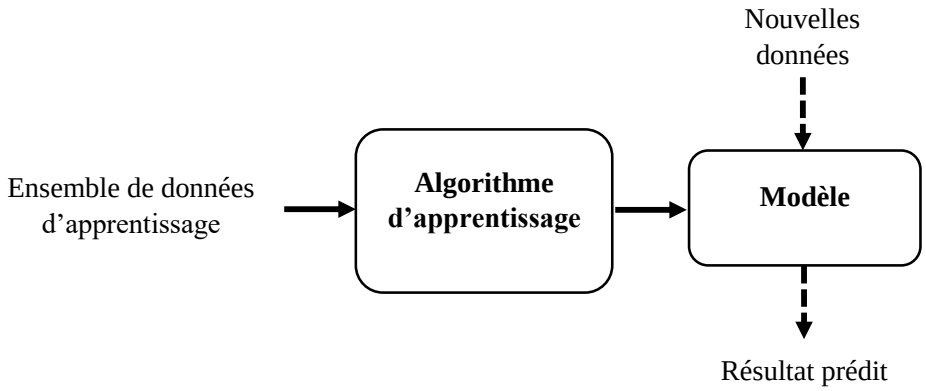


FIGURE 3.2 – Processus de l'apprentissage machine.

1. *Collecte et préparation des données*
   Il s'agit de déterminer quel type de données est nécessaire pour résoudre le problème considéré pour les collecter, et par conséquent, créer le jeu de

données. Cette étape doit être fusionnée avec l'étape suivante de sélection des caractéristiques, de sorte que seules les données requises soient collectées.
Pour l'apprentissage supervisé, la classe (donnée cible) de chaque exemple est également nécessaire, ce qui peut requérir la participation d'experts dans le domaine concerné. Aussi, la quantité de données doit être prise en compte. Les algorithmes d'apprentissage machine ont besoin de quantités importantes de données, de préférence sans trop de bruit. Toutefois, avec une taille de jeu de données accrue, les coûts de calcul augmentent, et le point idéal auquel il y a suffisamment de données sans surcharge de calcul excessive est généralement difficile à prévoir.

2. *Sélection des caractéristiques*
   Ce palier consiste à identifier les caractéristiques les plus utiles pour le problème examiné. Cela nécessite une connaissance préalable du problème et des données. Il est également nécessaire que les caractéristiques soient résistantes au bruit et à toute autre corruption de données pouvant survenir au cours du processus de collecte.

3. *Choix d'algorithme*
   Le choix d'un ou de plusieurs algorithmes appropriés dépend du type de problème à résoudre, de l'ensemble de données et du niveau de complexité du problème.

4. *Sélection des paramètres et des modèles*
   Pour de nombreux algorithmes, il existe des paramètres qui doivent être définis manuellement ou qui ont besoin des expérimentations pour identifier les valeurs appropriées. Lors de cette étape, le jeu de données est scindé en deux parties ; les données d'apprentissage et les données de test. Les données d'apprentissage seront utilisées pour construire et analyser le modèle, alors que les données de test seront utilisées pour sa validation.

5. *Apprentissage*
   Compte tenu de l'ensemble de données, de l'algorithme utilisé et des paramètres, l'apprentissage devrait être simplement l'utilisation de ressources de calcul afin de construire un modèle de données dans le but de prédire les sorties sur de nouvelles données.

6. *Évaluation*
   Après avoir élaboré un modèle à l'aide de l'ensemble de données d'apprentissage, et avant que le système puisse être déployé, il doit d'abord subir une évaluation pour une éventuelle modification. Un ensemble de données d'évaluation est utilisé pour vérifier l'efficience du modèle et la précision avec laquelle il peut prédire. Une fois la précision calculée, toute

autre amélioration du modèle peut être mise en œuvre à ce stade. Des méthodes telles que le réglage des paramètres et la validation croisée peuvent être utilisées pour améliorer les performances du modèle.

7. *Test*
   Dans cette étape, aucune opération de réglage des paramètres n'est permise. Le modèle construit doit être testé sur des données jamais vues lors de la phase d'apprentissage.

3.3.2 Paradigmes d'apprentissage machine

L'apprentissage machine est un domaine très large, par conséquent, il s'est divisé en plusieurs sous-domaines traitant de différents types de tâches d'apprentissage. Une taxonomie approximative des paradigmes d'apprentissage, visant à fournir un aperçu des différents types dans ce vaste domaine est décrite dans cette section.
Les systèmes d'apprentissage machine sont classés en fonction du type et de la façon de supervision humaine pendant la phase d'apprentissage. Quatre grandes catégories sont à distinguer [86] comme illustré dans la Figure 3.3.

- Apprentissage supervisé.
- Apprentissage non-supervisé.
- Apprentissage semi-supervisé.
- Apprentissage par renforcement.

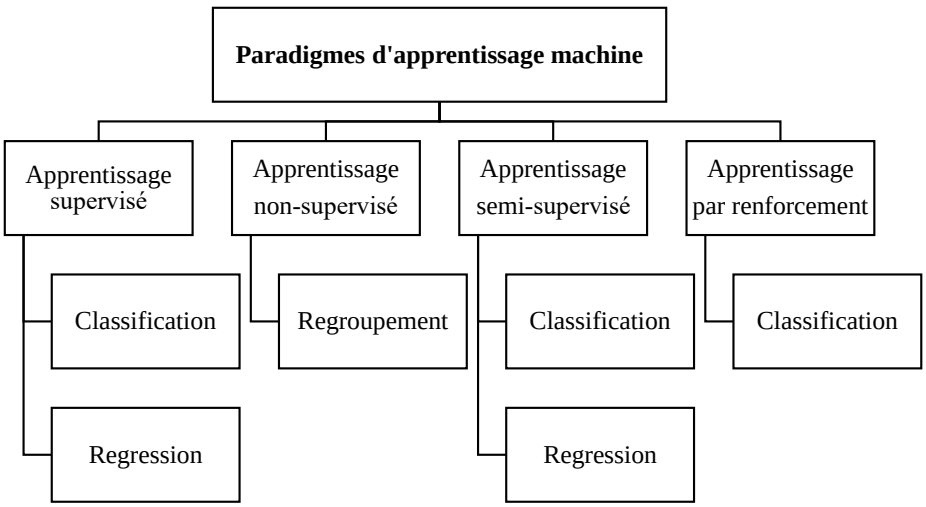


FIGURE 3.3 – Taxonomie des paradigmes de l'apprentissage machine.

3.3.2.1 Apprentissage supervisé

L'apprentissage supervisé est l'apprentissage à partir d'un ensemble d'exemples d'apprentissage étiquetés fournis par un superviseur externe compétent, autrement dit, une expertise humaine est nécessaire pour étiqueter les données. Chaque exemple est une description d'une situation accompagnée d'une étiquette (une classe, qui peut consister en des valeurs numériques ou nominales) de l'action correcte que le système doit prendre pour cette situation. Il s'agit d'identifier une classe à laquelle l'instance appartient.

L'objectif de ce type d'apprentissage est que le système extrapole ou généralise ses réponses pour qu'il agisse correctement dans des situations non présentes dans l'ensemble de l'apprentissage [87]. Ainsi, dans ce type d'apprentissage, l'utilisateur fournit à l'algorithme des paires d'entrées/sorties souhaitées (X,y) illustrées dans la Figure 3.4, et l'algorithme trouve un moyen de produire la sortie souhaitée à partir des entrées. En particulier, l'algorithme est capable de créer une sortie pour une entrée qu'il n'a jamais vue auparavant [82].

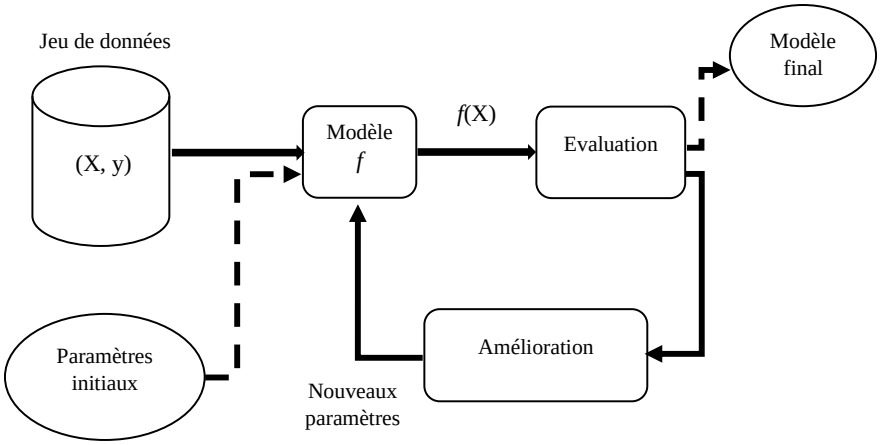


FIGURE 3.4 – Processus de l'apprentissage supervisé.

3.3.2.2 Apprentissage non-supervisé

En pratique, la majorité des données produites ne sont pas étiquetées. Mais pour autant, elles recèlent d'énormes quantités d'informations qui ne demandent qu'à être valorisées. C'est dans ces cas de figure que peut servir l'apprentissage non-supervisé.

L'apprentissage non-supervisé est un autre type d'apprentissage où seules les données d'entrée sont connues et aucune sortie n'est fournie à l'algorithme. Le rôle de cet algorithme d'apprentissage non-supervisé est d'extraire lui même les connaissances de ces données, autrement dit, découvrir des groupes d'exemples similaires (homogènes) dans les données comme le montre la Figure 3.5 : c'est l'opération de regroupement (clustering) [88, 89].

Ainsi, les techniques d'apprentissage non-supervisées ne reposent pas sur des données étiquetées et tentent de trouver des modèles dans un ensemble de données sans interaction humaine.

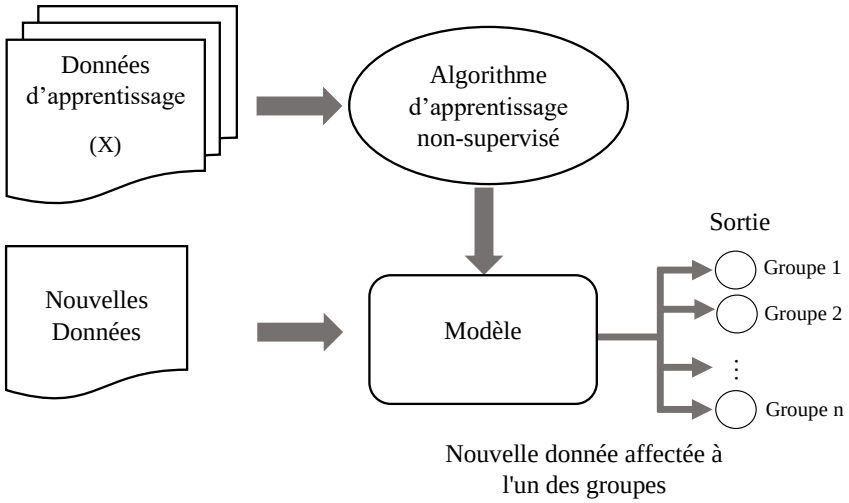


FIGURE 3.5 – Processus de l'apprentissage non-supervisé.

Pour les tâches d'apprentissage supervisées et non-supervisées, il est nécessaire d'avoir une représentation des données d'entrée qu'un ordinateur peut comprendre. Ces données sont souvent considérées comme étant un tableau, où chaque ligne représente un exemple de données (instance) et chaque colonne représente la propriété ou la caractéristique qui décrit cet exemple de données [89]. Il est nécessaire de distinguer les sorties discrètes des sorties continues des algorithmes d'apprentissage machine. Les sorties discrètes ont tendance à provenir d'un ensemble distinct et fini de valeurs. Par exemple, une sortie discrète peut être une valeur entière. Les sorties continues quant à elles, ont tendance à être des valeurs appartenant à un ensemble continu, avec un nombre potentiellement infini de valeurs. Par exemple, la taille d'un adulte peut être comprise entre 1.30 et 1.90 mètres, et pourra prendre les valeurs 1.65 ou 1.6598.
La Table 3.1 considère les cas d'utilisation de ce type de données avec l'apprentissage supervisé et non-supervisé. Ces derniers sont utilisés pour une variété de tâches, les principales étant la classification, la régression, le regroupement et la réduction de dimensionnalité.

TABLE 3.1 – Type de données vs type d'apprentissage.

	Apprentissage supervisé	**Apprentissage non-supervisé**
Discrète	Classification	Regroupement
Continue	Régression	Réduction de dimensionnalité

3.3.2.3 Apprentissage semi-supervisé

La communauté de l'apprentissage machine s'est penchée vers le paradigme d'apprentissage semi-supervisé dans le but d'améliorer significativement la qualité de l'apprentissage. Il se situe alors entre l'apprentissage supervisé qui utilise des données étiquetées et l'apprentissage non-supervisé qui utilise des données non étiquetées (voir Figure 3.6). Cette combinaison fait référence à une quantité de données non étiquetées supérieure à celle des données étiquetées. Par ailleurs, l'apprentissage semi-supervisé est très important dans les scénarios du monde réel où toutes les données disponibles sont une combinaison de données étiquetées et non étiquetées [90].

L'apprentissage semi-supervisé peut être utilisé par exemple en co-apprentissage, dans lequel deux classifieurs apprennent un ensemble de données en utilisant chacun un ensemble de caractéristiques distinctes et indépendantes. A titre d'exemple si les données sont des images à classifier en adultes et enfants l'un pourra utiliser la taille et l'autre la voix.

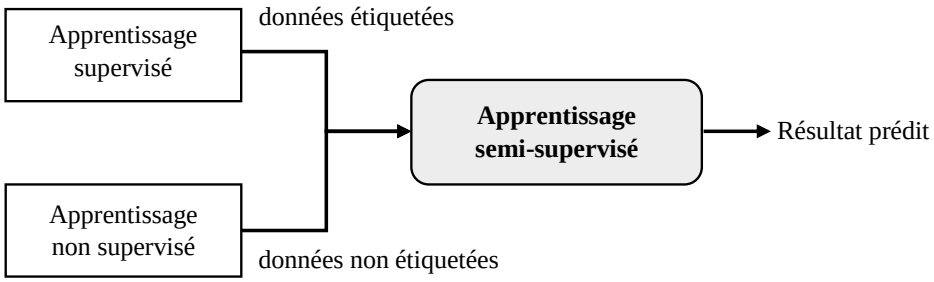


FIGURE 3.6 – Processus de l'apprentissage semi-supervisé.

3.3.2.4 Apprentissage par renforcement

L'apprentissage par renforcement est un autre type d'apprentissage machine. Il se base sur le concept de récompense. Il s'agit d'apprendre par une machine comment faire correspondre des situations à des actions afin de maximiser un signal de récompense numérique.

Les actions à entreprendre ne sont pas connues au préalable, comme dans d'autres types d'apprentissage machine, mais doivent être découvertes (celles qui rapportent le plus de récompense) [87]. Il peut être formulé comme un processus de décision de Markov d'un agent interagissant avec l'environnement afin de maximiser la récompense future.

À chaque instant t, étant donné l'état actuel e_t (et la récompense actuelle r_t), l'agent doit apprendre une stratégie qui sélectionne la décision ou l'action optimale a_t. L'action aura un impact sur l'environnement qui induit le prochain signal de récompense r_{t+1} (qui peut être positif, négatif ou nul) et produit également l'état suivant e_{t+1} (voir Figure 3.7).

L'apprentissage par renforcement continu avec un processus d'essais et d'erreurs jusqu'à ce qu'il apprenne une stratégie optimale ou sous-optimale [87]. La recherche par essais/erreurs et récompense différée représentent deux caractéristiques distinctes les plus importantes de l'apprentissage par renforcement. A cet effet, un agent (tel qu'un robot) observe l'environnement où il apprend, par des méthodes d'essai et d'erreur, à prendre une décision bien spécifique, effectue des actions ciblées, puis reçoit en retour des récompenses. Avec ce type d'apprentissage, l'agent doit apprendre par lui-même [86].

Ainsi, dans l'apprentissage par renforcement, les algorithmes choisissent une action dans un environnement et sont ensuite récompensés (positivement ou négativement) pour avoir choisi cette action. L'algorithme s'ajuste ensuite et modifie sa stratégie afin d'atteindre un objectif d'obtenir plus de récompenses [87].

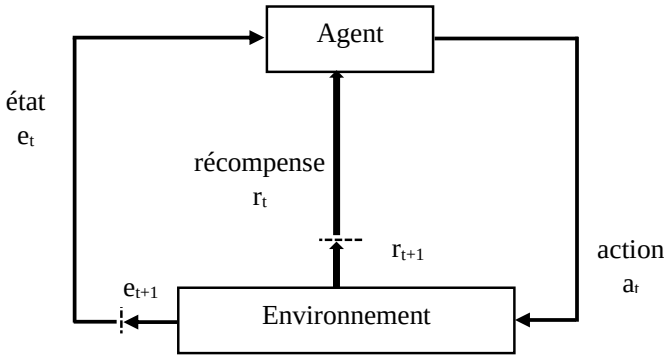


FIGURE 3.7 – Interaction agent-environnement dans l'apprentissage par renforcement.

Ce type d'apprentissage est bien favorable pour de nombreuses applications robotiques. Il diffère de l'apprentissage supervisé et non-supervisé par le signal de récompense qui indique simplement si l'action prise par l'agent est bonne ou mauvaise (aucun détail sur la meilleure action). En outre, il n'utilise ni les données d'apprentissage ni les étiquettes [87].

3.3.3 Classification

En apprentissage machine, la classification fait référence à la tâche d'identification de la classe à laquelle appartient un exemple de données bien spécifique, compte tenu des informations relatives à ses caractéristiques. Par exemple, étant donné l'image d'une fleur, la longueur et la largeur des pétales, la longueur et la largeur des sépales représentent des caractéristiques qui peuvent être utilisées pour identifier le type de la fleur. Les données caractéristiques peuvent être discrètes ou continues, tandis que les étiquettes de classes doivent être discrètes

(par nature ou par codage d'étiquette) pour tout problème de classification. Souvent, les valeurs des étiquettes de classe sont des chaînes de caractères (par exemple « noir » « blanc ») et doivent être associées à des valeurs numériques avant d'être fournies à un algorithme d'apprentissage. Ceci est souvent appelé codage d'étiquette, où un entier unique est attribué à chaque étiquette de classe, par exemple « noir »= 0, « blanc »= 1. Notons que cette thèse, s'inscrit dans le cadre de résolution de ce type de problème.

Types de classification

Essentiellement, deux types de classification sont à distinguer : 1) la classification binaire et 2) la classification multi-classes.

Classification binaire

La classification binaire fait référence aux tâches de classification qui ont deux étiquettes de classe. Typiquement, les tâches de classification binaire impliquent une classe qui est l'état normal et une autre classe qui est l'état anormal. La classe de l'état normal se voit attribuer l'étiquette de classe 0 et la classe de l'état anormal se voit attribuer l'étiquette de classe 1. Il est courant de modéliser une tâche de classification binaire avec un modèle qui prédit une distribution de probabilité de Bernoulli pour chaque exemple. La distribution de Bernoulli est une distribution de probabilité discrète qui couvre le cas où un événement aura un résultat binaire 0 ou 1. Pour la classification, cela signifie que le modèle prédit la probabilité d'un exemple appartenant à la classe 1.

Classification multi-classes

La classification multi-classes fait référence aux tâches de classification qui ont plus de deux étiquettes de classe. Contrairement à la classification binaire, la classification multi-classes n'utilise pas la notion d'état normal et anormal. Dans ce type de classification, les exemples prennent une étiquette de classe parmi une gamme d'étiquettes de classes déterminées préalablement. Le nombre d'étiquettes de classe dépend du problème traité.

3.4 Applications de l'apprentissage machine

L'apprentissage machine admet une multitude d'applications pratiques, notamment [91] :

- Classification des textes ou des documents : inclut des problèmes tels que l'attribution d'un sujet à un texte ou à un document, ou la détermination

automatique si le contenu d'une page Web est inapproprié ; ce type d'application comprend également la détection des spam.

- Traitement du langage naturel : la plupart des tâches dans ce domaine ont pour objectif l'extraction des informations et la signification d'un contenu textuel. Ceci inclue la recherche de documents dans des bases documentaires, la traduction automatique, le résumé automatique et l'analyse syntaxique, etc.

- Traitement de la parole : inclue la reconnaissance automatique de la parole, la synthèse vocale, la vérification du locuteur, l'identification du locuteur, ainsi que des sous-problèmes tels que la modélisation du langage et la modélisation acoustique.

- La vision par ordinateur : comprend la reconnaissance d'objets, l'identification d'objets, la détection de visages, la reconnaissance optique de caractères, la récupération d'images basée sur le contenu.

- La biologie computationnelle : comprend la prédiction de la fonction des protéines, l'identification des sites clés ou l'analyse des réseaux de gènes et de protéines.

- De nombreux autres problèmes tels que la détection des fraudes pour les cartes de crédit, les compagnies de téléphone ou les compagnies d'assurance, l'intrusion dans le réseau, l'apprentissage de jeux tels que les échecs et le backgammon, le contrôle non assisté de véhicules tels que les robots ou les voitures, le diagnostic médical, la conception des systèmes de recommandation, les moteurs de recherche ou les systèmes d'extraction d'informations sont traités à l'aide de techniques d'apprentissage machine.

Cette liste n'est en aucun cas exhaustive, la plupart des problèmes de prédiction rencontrés dans la pratique peuvent être considérés comme des problèmes d'apprentissage et le domaine d'application pratique de l'apprentissage machine ne cesse de s'étendre.

3.5 Apprentissage profond

Depuis 2006, l'apprentissage structuré profond, ou plus communément appelé apprentissage profond ou apprentissage hiérarchique (Deep Learning : DL), est devenu un nouveau domaine de recherche en apprentissage machine [92]. Les premiers travaux ont montré qu'un perceptron linéaire ne peut pas être un classifieur universel, et qu'un réseau avec une fonction d'activation non-linéaire avec plusieurs couches cachées pouvant être également hétérogènes, peut en revanche l'être : c'est l'apprentissage profond.

Le terme *profond* dans l'apprentissage profond vient de l'utilisation de plusieurs couches dans le réseau. Les diverses définitions ou descriptions de haut niveau de l'apprentissage profond sont étroitement liées, nous citons la suivante : un sous-domaine de l'apprentissage machine basé sur des algorithmes d'apprentissage de plusieurs niveaux de représentation afin de modéliser des relations complexes entre les données. Les caractéristiques et les concepts de niveau supérieur sont ainsi définis en termes de niveaux inférieurs, et une telle hiérarchie de fonctionnalités est appelée une architecture profonde [93].

Les réseaux de neurones profonds (Deep Neural Networks : DNN) [94], les réseaux profond de croyance (Deep Belief Networks : DBN) [95], les réseaux de neurones récurrents (Recurrent Neural Networks : RNN) [96] et les réseaux de neurones convolutionnels (Convolutional Neural Networks : CNN) [97] représentent des architectures d'apprentissage profond qui ont été appliquées à une diversité de domaines tels que la reconnaissance automatique de la parole, la vision par ordinateur, le traitement du langage naturel, la traduction automatique, l'analyse d'images médicales, etc. où ils ont donné des résultats comparables et dans certains cas dépassant les performances des experts humains.

3.5.1 Réseaux de neurones artificiels

Alors que les ordinateurs modernes deviennent de plus en plus puissants, les scientifiques cherchent à utiliser efficacement les machines pour des tâches relativement simples pour les humains. Dans ce contexte, le développement de réseaux de neurones artificiels (Artificial Neural Network : ANN) a commencé il y a environ une soixantaine d'années, motivé par le désir d'essayer à la fois de comprendre le cerveau humain et d'égaler certaines de ses capacités.

Les Réseaux de Neurones représentent des modèles qui symbolisent des fonctions mathématiques avec un nombre important de paramètres. La section 3.5.1.1 va s'attacher à présenter le neurone biologique alors que la section 3.5.1.2 va expliquer le neurone artificiel tout en mettant l'évidence sur la correspondance entre le neurone biologique et l'artificiel.

3.5.1.1 Neurone biologique

Le cerveau humain se compose d'environ 10^{11} neurones (mille milliards), avec 1000 à 10000 synapses (connexions) par neurone. Le neurone est une cellule comportant un corps cellulaire, centre de contrôle de celui-ci, qui effectue la somme des informations qui lui parviennent (voir Figure 3.8). Le corps cellulaire se ramifie pour former les dendrites qui permettent l'acheminement de l'information de l'extérieur vers le corps du neurone. Il traite l'information et l'achemine tout au long de l'axone pour la transmettre à d'autres neurones. La

jonction entre deux neurones est appelée la synapse [98].

Les réseaux de neurones biologiques effectuent facilement certaines fonctions telles que la mémorisation, l'apprentissage par l'exemple, la généralisation, la reconnaissance des formes et le traitement du signal. C'est à partir du principe que le comportement intelligent provient de la structure et du comportement des neurones biologiques que les recherches ont abouti aux neurones formels ou encore appelé artificiels.

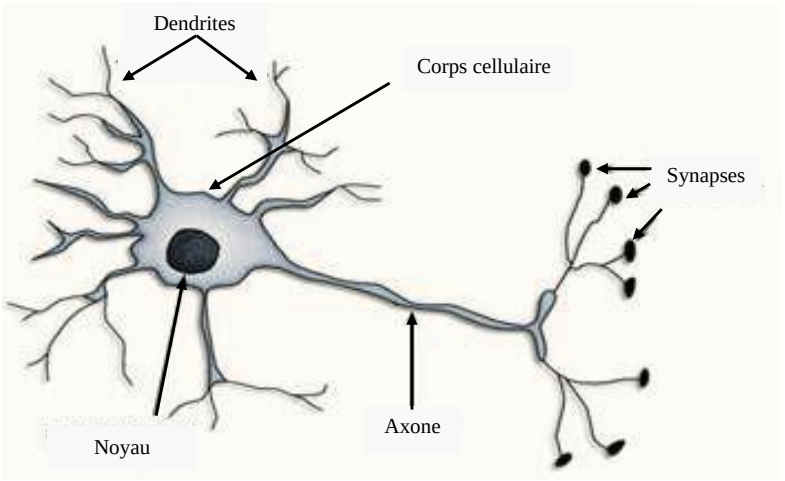


FIGURE 3.8 – Le neurone biologique.

3.5.1.2 Neurone artificiel

Le neurone artificiel est la forme mathématique du neurone biologique. Il représente un processeur élémentaire qui reçoit des valeurs en entrée x_n associées à des poids w_{in} représentatifs de la force de la connexion. Aussi, le neurone artificiel renvoie en sortie une seule valeur. Celle-ci peut être diffusée à plusieurs neurones en aval [81]. La structure d'un neurone artificiel est illustrée dans la Figure 3.9.

La Table 3.2 montre une mise en correspondance entre le neurone biologique et le neurone artificiel.

TABLE 3.2 – Mise en correspondance neurone biologique et neurone artificiel.

Neurone biologique	**Neurone artificiel**
Dendrites	Signal d'entrée
Synapes	Poids de connexion
Corps cellulaire	Fonction d'activation
Axone	Signal de sortie

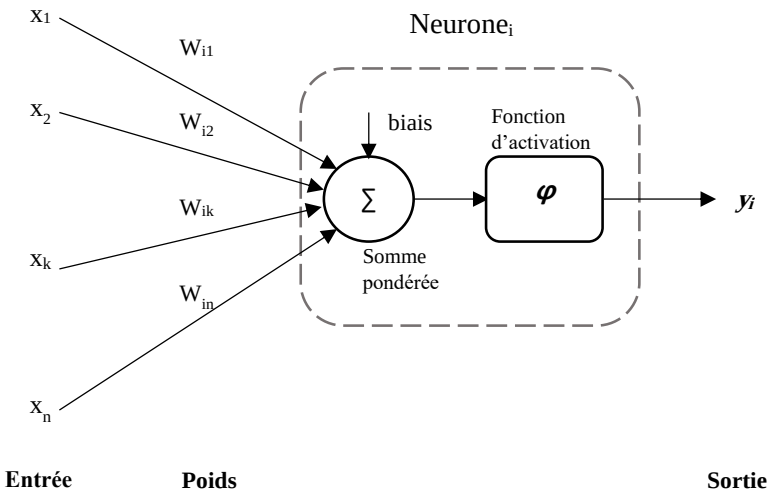


FIGURE 3.9 – Structure d'un neurone artificiel.

La sortie du neurone est calculée par la fonction combinaison représentée par le produit scalaire, à laquelle une fonction d'activation est appliquée dans le but d'obtenir la valeur de sortie y_i en utilisant l'équation (3.1).

$$y_i = f\left(\sum_{k=1}^{n}(w_{ik}x_k) + b\right). \tag{3.1}$$

où

n est le nombre des entrées, b est le biais et f est la fonction d'activation.

Plusieurs types de fonction d'activation sont définies dans la littérature, les plus utilisées sont la fonction sigmoïde exprimée par l'équation (3.2), la tangente hyperbolique décrite par l'équation (3.3) et la fonction Unité Linéaire Rectifiée (Rectified Linear Unit : ReLU) exprimée par l'équation (3.4) [99].

$$f(x) = \frac{1}{1+e^{-x}}. \tag{3.2}$$

$$f(x) = \frac{e^{-x}-1}{e^{-x}+1}. \tag{3.3}$$

$$f(x) = \max(0, x). \tag{3.4}$$

Un réseau de neurones artificiels est formé d'un ensemble de neurones, fortement connectés entre eux et dont le fonctionnement est parallèle.
Il représente ainsi un système de traitement de l'information qui a certaines caractéristiques de performance en commun avec le réseau de neurones biologiques. Il a été développé comme généralisations de modèles mathématiques

de la cognition humaine ou de la biologie neuronale, sur la base des hypothèses suivantes [99] :

1. Le traitement de l'information se produit au niveau des neurones ;
2. Les signaux sont transmis entre les neurones via des liens de connexion.
3. Chaque lien de connexion a un poids associé qui, dans un réseau neuronal typique, multiplie le signal transmis ;
4. Chaque neurone applique une fonction d'activation (généralement non linéaire) au produit scalaire de ses entrées pour déterminer son signal de sortie.

Ainsi, un réseau de neurones se caractérise par [99] :

- son architecture : schéma de connexions entre les neurones,
- son algorithme d'apprentissage : méthode de détermination des poids de connexions,
- ses fonctions d'activation.

3.5.2 Réseaux de neurones profonds

Un réseau neuronal profond (DNN) est un perceptron multi-couches conventionnel avec de nombreuses couches cachées, souvent plus de deux. La Figure 3.10 illustre un DNN avec un total de cinq couches qui inclue une couche d'entrée, trois couches cachées et une couche de sortie. Le terme réseau neuronal profond a été initialement introduit pour désigner des perceptrons multicouches avec de nombreuses couches cachées, mais a ensuite été étendu pour désigner tout réseau de neurones avec une structure profonde [14].

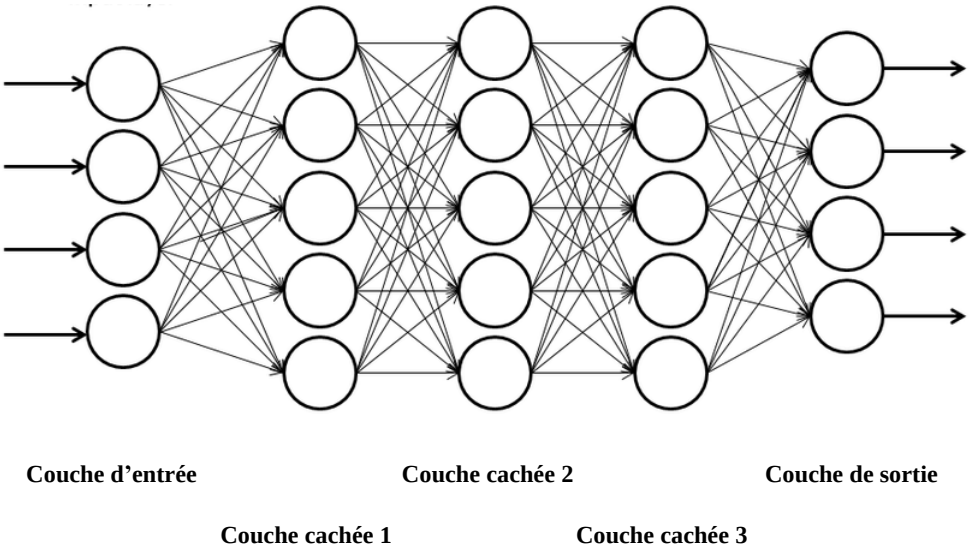


FIGURE 3.10 – Un exemple d'un réseau de neurones profond avec une couche d'entrée, trois couches cachées et une couche de sortie.

Plusieurs types de réseaux de neurones avec différentes propriétés ont été développés depuis l'apparition des neurones formels dans les années quarante [44]. Plus particulièrement, deux types de réseaux sont à distinguer [100].

- Les réseaux de neurones statiques, dits aussi acycliques, ou non bouclés, étant donné l'absence des cycles.
- Les réseaux de neurones dynamiques, appelés aussi récurrents, ou bouclés, vu la présence d'au moins un cycle dans le réseau.

3.5.3 Les réseaux de neurones acycliques

Un réseau de neurones acyclique (Feed-Forward Neural Network : FFNN) effectue une ou plusieurs fonctions algébriques à ses entrées par composition des fonctions exécutées par chacun de ses neurones. Le flux d'information circule des entrées vers les sorties sans retour en arrière dans ce type de réseau comme le montre la Figure 3.11.

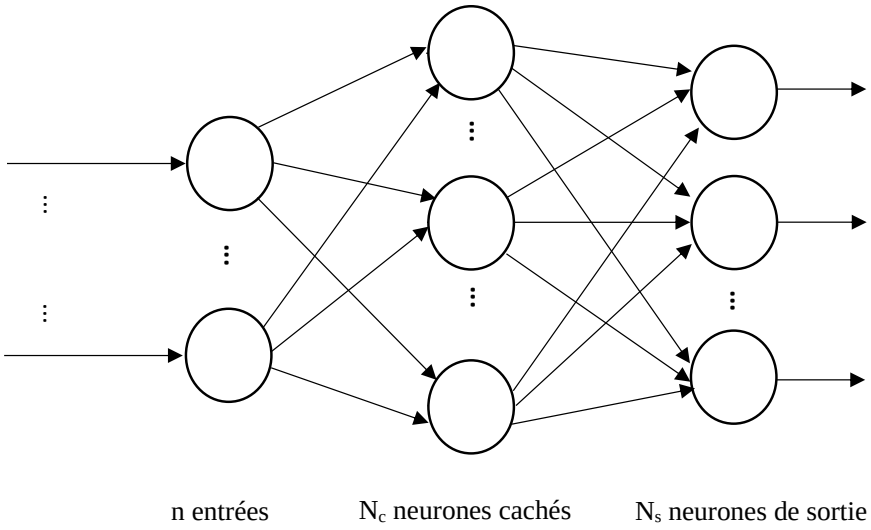


FIGURE 3.11 – Réseau de neurones à n entrées, une couche cachée de Nc neurones et une couche de sortie à Ns neurones.

Dans ce type de réseau, le temps ne représente aucun rôle fonctionnel. Le temps nécessaire pour effectuer le calcul de la fonction réalisée par chaque neurone est insignifiant et ce calcul peut être considéré instantané [100]. A cet effet, les réseaux de neurones acycliques sont souvent désignés par « réseaux statiques », par opposition aux réseaux récurrents ou « dynamiques » où la notion de temps est importante et non négligeable.

Généralement, pour les réseaux acycliques, différentes variantes existent, parmi lesquels : les réseaux de type perceptron multi-couches [101], convolutifs [102] et Radial Basis Function [103]. Néanmoins, la variante la plus utilisée est le perceptron multi-couches.

3.5.3.1 Perceptron multi-couches

Le type de réseau de neurones le plus courant et le plus utilisé est le perceptron multi-couches (Multi-Layer Perceptron : MLP) créé par Frank Rosenblatt à la fin des années cinquante [101]. Un MLP est composé d'un certain nombre de neurones artificiels hautement interconnectés fonctionnant en parallèle et organisés en couches, avec un flux d'informations à action directe (pas de boucles). L'architecture du perceptron multi-couches est illustrée dans la Figure 3.12 qui montre qu'elle comprend une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. La couche d'entrée se définit par un nombre de neurones égal à la dimension des données, alors que la couche de sortie se définit par un nombre de neurones égal au nombre de classes à discriminer.

Dans un MLP, les signaux circulent consécutivement à travers les différentes couches, de la couche d'entrée à la couche de sortie. Pour chaque couche, chaque

neurone élémentaire calcule un produit scalaire entre un vecteur de poids et le vecteur de sortie donné par la couche précédente. Une fonction d'activation est ensuite appliquée au résultat pour produire une entrée pour la couche suivante [104]. La structure d'un MLP se caractérise par trois paramètres : le nombre de couches cachées, le nombre de neurones dans chaque couche cachée et les fonctions d'activation utilisées.

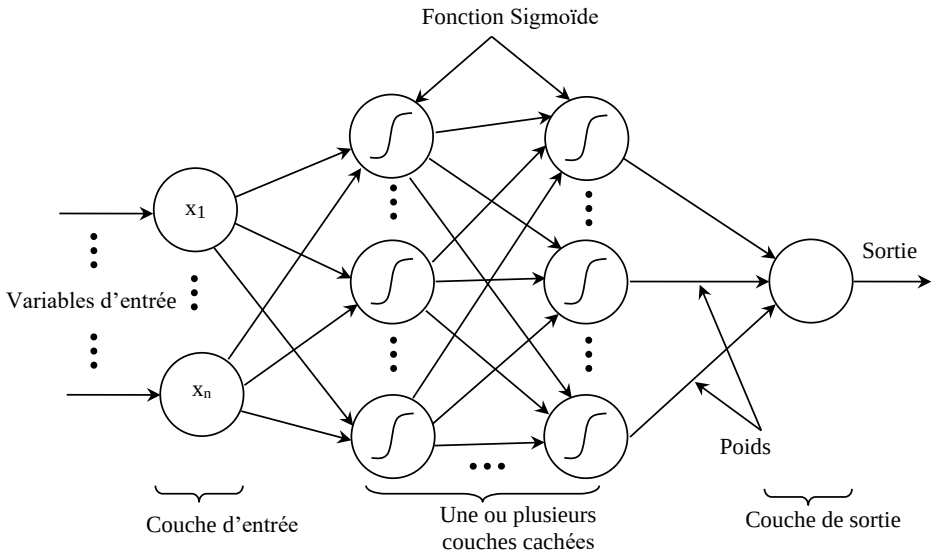


FIGURE 3.12 – Architecture d'un réseau perceptron multi-couches.

Typiquement, les MLPs sont utilisés pour les problèmes de classification supervisée qui nécessite la disponibilité d'un ensemble de paires d'entrées/ sorties (données d'apprentissage) liées par une relation que le réseau va « apprendre » en ajustant ses paramètres durant la phase d'apprentissage. Cet ajustement des poids est réalisé par l'algorithme de rétro-propagation de l'erreur (Error Back-Propagation : EBP) [104].

L'exécution de cet algorithme se répète autant de fois que nécessaire jusqu'à ce qu'un certain critère de convergence soit atteint. Durant la phase de classification, le réseau reçoit en entrée, une nouvelle donnée pour laquelle, il doit prendre une décision [105, 106].

3.5.3.2 Algorithme de rétro-propagation de l'erreur

L'algorithme de rétro-propagation du gradient de l'erreur se déroule principalement en deux étapes : *la propagation avant (Forward pass)*, et *la propagation arrière (Backward pass)*. Au cours de la *propagation avant* les sorties du réseau sont calculées à partir des entrées (comme décrit préalablement), et pendant la *propagation arrière* les dérivées partielles d'une fonction de coût E (fréquemment l'erreur quadratique moyenne (Mean Square Error : MSE) entre la sortie prédite

et la sortie souhaitée par rapport aux paramètres du réseau sont rétro-propagés. Ensuite, une mise à jour des poids du réseau est réalisée en fonction de cette dérivée partielle avec l'équation (3.5). Ainsi, pour obtenir la relation entrée / sortie souhaitée du réseau, les poids des connexions entre les neurones sont ajustés. Ce processus se poursuit jusqu'à ce que la différence entre la sortie du réseau et la sortie souhaitée soit égale à une erreur de seuil prédéfinie. D'autre part, le processus d'apprentissage doit être répété pour le reste des paires entrée-sortie existant dans les données d'apprentissage [104].

$$\Delta W_{ij} = -\epsilon \frac{\partial E}{\partial w_{ij}}. \tag{3.5}$$

où ϵ représente le taux d'apprentissage.

3.5.4 Réseaux de neurones récurrents

Les réseaux de neurones récurrents (Recurrent Neural Network : RNN) définissent une famille de réseaux de neurones qui contiennent une boucle (Figure 3.13), contrairement à un réseau de neurones acycliques (FFNN), où l'information se propage de couche en couche sans retour en arrière possible. Cette caractéristique de boucle les rend ainsi récurrents et permettant aux informations de persister en eux [107]. Ils sont de nature récurrente, car ils exécutent la même fonction pour chaque entrée de données puisque la sortie de l'entrée actuelle dépend du dernier calcul [48]. Pour prendre une décision, le réseau considère l'entrée actuelle et la sortie obtenue de la phase précédente. Ainsi, dans un RNN, toutes les entrées sont liées les unes aux autres.

Les réseaux neuronaux récurrents simples contiennent uniquement une boucle tandis que d'autres réseaux neuronaux récurrents plus complexes sont composés d'une ou plusieurs portes qui leur permettent de modéliser les informations à retenir et à oublier [107].

Les RNNs sont adaptés pour des données d'entrée de taille variable. Ils conviennent en particulier pour l'analyse de séries temporelles et sont utilisés bien particulièrement en traitement automatique de la parole [108].

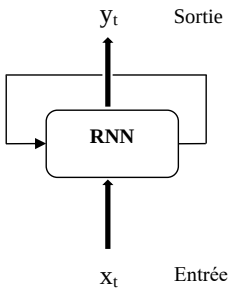


FIGURE 3.13 – Structure d'un RNN simple.

La Figure 3.14 illustre un RNN déroulé. Elle montre qu'un RNN (à gauche ; présence d'un cycle) peut être vu comme une séquence de réseaux neuronaux (absence de cycle) où les différents pas de temps sont visualisés et l'information est passée d'une étape à l'autre. Là, le RNN prend le x_{t-1} de la séquence d'entrée, puis il résulte h_{t-1} (équation (3.6)) qui, avec x_t, forme l'entrée pour l'étape suivante. Ainsi, le h_{t-1} et x_t représentent l'entrée pour l'étape suivante. Similairement, h_t est l'entrée avec x_{t+1} pour l'étape suivante et ainsi de suite. De cette façon, le RNN garde en mémoire le contexte pendant l'apprentissage.

$$h_t = f(h_{t-1}, X_t). \tag{3.6}$$

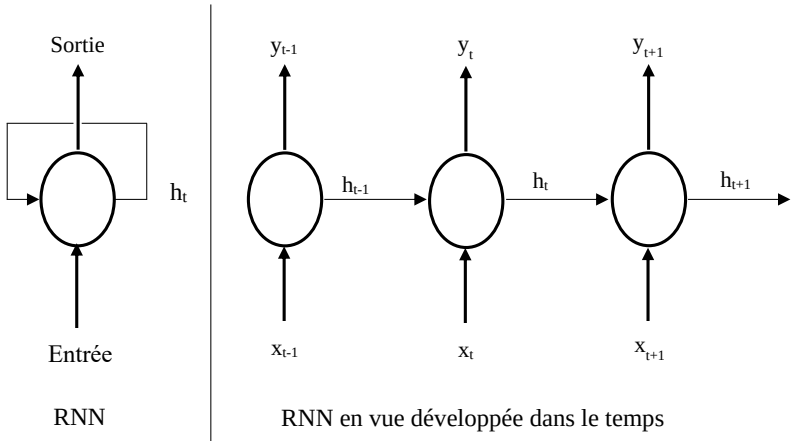


FIGURE 3.14 – Un RNN déroulé.

3.5.4.1 Réseaux récurrents bidirectionnels

Une évolution des réseaux de neurones récurrents a été dénotée par l'introduction des réseaux de neurones récurrents bidirectionnels (Bidirectional Recurrent Neural Network : BiRNN) dans les travaux de [109]. Ces réseaux exploitent l'information « passée » et « future » afin de réaliser de meilleures prédictions. La Figure 3.15 schématise l'architecture d'un BiRNN.

Un réseau récurrent bidirectionnel comporte une couche appelée *couche forward* qui réalise une récurrence dans le sens temporel et une couche appelée *couche backward* appliquant la récurrence en sens inverse. Ces deux couches cachées sont utilisées pour produire la sortie du réseau pour le même temps t. En sortie, chaque RNN résulte une séquence, ensuite les deux séquences obtenues seront concaténées.
L'apprentissage des BiRNNs est effectué d'une manière identique à celle des réseaux unidirectionnels, avec prise en compte des sens de propagation différents pour les deux couches cachées [109].

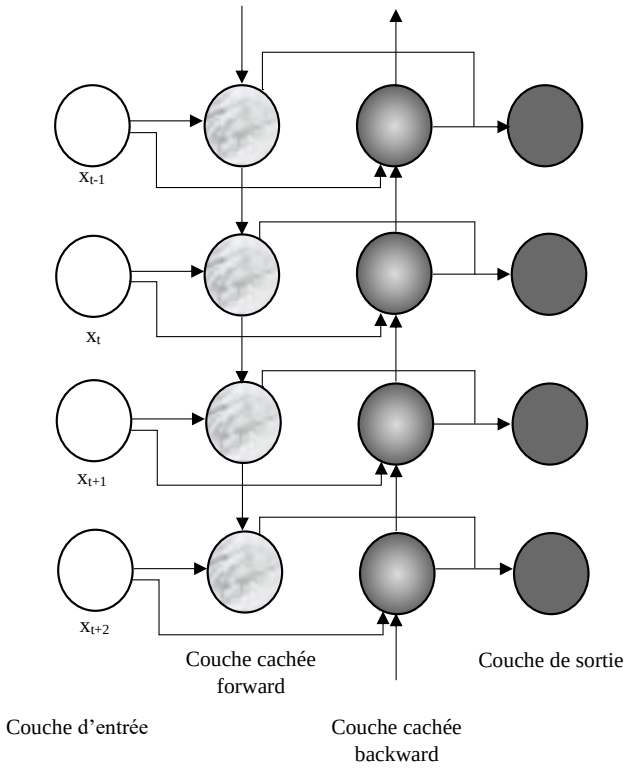


FIGURE 3.15 – Schéma fonctionnel d'un réseau récurrent bidirectionnel.

Des faiblesses de ce type de réseau apparaissent dès que les séquences deviennent longues, tel est le cas des phrases sous forme textuelle ou sous forme de signal. En effet, le gradient diminue au fil du temps et n'impacte que faiblement les premières itérations [110].

Face à ce problème, les travaux de *Hochreiter et Schmidhuber* [45] proposent une nouvelle approche comme solution ; les réseaux de neurones à base de cellules (Long Short-Term Memory : LSTM). Cette variante des RNNs est devenue la technique de plus en plus utilisée pour le traitement des séquences temporelles vu les performances obtenues dans des tâches aussi nombreuses que variées.

3.5.4.2 Réseaux de neurones à base de cellules

Les réseaux de neurones à base de cellules (LSTM) représentent une architecture de réseau de neurones récurrents permettant de pallier le problème de la disparition du gradient [111]. Le modèle LSTM introduit des portes logiques multiplicatives permettant de conserver et d'accéder à l'information pertinente sur de longs intervalles.

Une cellule LSTM possède une mémoire interne contrôlée par trois portes : 1) une porte d'entrée, 2) une porte d'oubli et 3) une porte de sortie, utilisées pour

le contrôle du flux d'information, car elles permettent de transférer plus d'informations pertinentes entre les temps t_i et permettent aussi d'apprendre des dépendances de plus longs termes [110, 112].

L'architecture d'une cellule LSTM est schématisée dans la Figure 3.16, où elle montre que la cellule LSTM est pilotée par trois entrées :

x_t : entrée courante,

h_{t-1} : état caché précédent,

C_{t-1} : état précédent de la cellule.

Et fournie deux sorties :

h_t : la sortie courante représente l'information prédite par la cellule LSTM,

C_t : état courant de la cellule représente la mémoire du réseau qui évolue à chaque temps t pour être utilisée par la cellule au temps $t+1$.

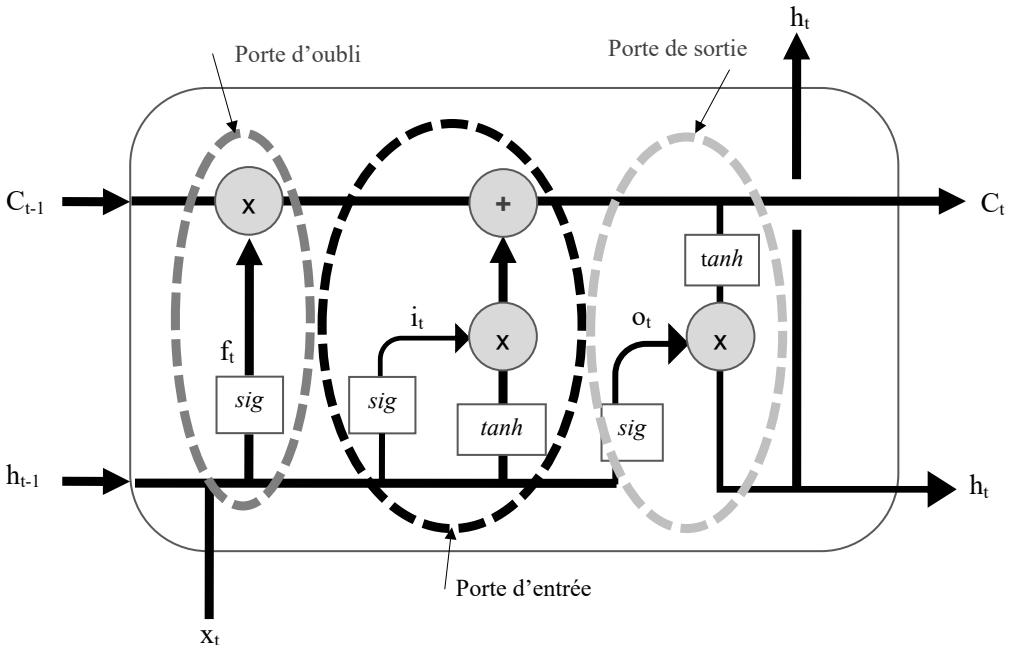


FIGURE 3.16 – Schéma fonctionnel d'une cellule LSTM.

La porte d'oubli : contrôle la quantité d'information mémorisée, elle décide quelles informations doivent être oubliées ou conservées. Les informations de l'état caché précédent h_{t-1} et les informations de l'entrée actuelle x_t sont transmises via la fonction sigmoïde. Ainsi, à partir de h_{t-1} et x_t, cette porte produit un vecteur f_t en utilisant l'équation (3.7) dont les valeurs sont comprises entre 0 et 1 (plus près de 0 signifie oublier l'information et plus proche de 1 signifie la mémoriser).

La porte d'oubli procède comme un filtre pour *oublier* certaines informations

de l'état de la cellule. A cet effet, une multiplication terme à terme s'effectue entre f_t et c_{t-1}, ce qui a tendance à annuler les composantes de c_{t-1} proches de 0. Un état de cellule filtré, est alors obtenu. [113].

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \tag{3.7}$$

où W_f : les poids de la porte d'oublie et b_f est le biais.

La porte d'entrée : permet la mise à jour de l'état de la cellule en filtrant l'information utile contenue dans les entrées x_t et la prédiction précédente h_{t-1}, qui doit être introduite dans l'état de la cellule c_t.
A partir de l'état caché précédent h_{t-1} et l'entrée actuelle x_t, cette porte produit un filtre i_t (équation (3.8)), de valeurs comprises entre 0 et 1 pour décider quelles valeurs seront mises à jour, de façon similaire à la porte d'oubli.
En parallèle, un vecteur $\tilde{C}_t$ (équation (3.9)) est généré, à partir de l'état caché précédent et l'entrée actuelle x_t par la fonction *tanh*. $\tilde{C}_t$ est le vecteur candidat pour mettre à jour l'état de la cellule. Ensuite, une multiplication de la sortie issue de la fonction *tanh* par celle de la fonction *sigmoide* est effectuée ($i_t \times \tilde{C}$). La sortie *sigmoide* a pour rôle de décider quelles sont les informations à retenir de la sortie *tanh*.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i). \tag{3.8}$$

$$\tilde{C} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c). \tag{3.9}$$

Etat de la cellule : Le calcul de l'état de la cellule se base sur la porte d'oubli et de la porte d'entrée, par la multiplication de la sortie de la porte d'oubli avec l'ancien état de la cellule. Cette opération permet d'oublier certaines informations de l'état précédent (non nécessaire pour la nouvelle prédiction). Le résultat filtré obtenu sera additionné avec la sortie de la porte d'entrée via l'équation (3.10) afin de mémoriser dans l'état de la cellule ce que le LSTM a considéré pertinent (parmi l'état caché précédent et les entrées).

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{C}. \tag{3.10}$$

Porte de sortie : la porte de sortie décide quel devrait être le prochain état caché (contenant des informations sur les entrées précédentes). De façon analogue à f_t et i_t, la porte de sortie produit un filtre o_t, de valeurs entre 0 et 1, en utilisant l'équation (3.11). En outre, les valeurs du nouveau état courant c_t sont normalisées entre -1 et 1 par le biais d'une fonction d'activation *tanh*. Ensuite, un filtrage par la porte de sortie o_t est réalisé dans le but d'obtenir la sortie h_t définie par l'équation (3.12).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{3.11}$$

$$h_t = o_t * tanh(c_t). \tag{3.12}$$

Le nouvel état de cellule et celui de l'état caché sont ensuite dirigés vers le pas de temps suivant.

Il a été démontré que les réseaux LSTM apprennent les dépendances à long terme plus facilement que les architectures récurrentes simples sur des tâches difficiles de traitement de séquences temporelles [114]. Récemment, une autre variante et alternative aux LSTMs a été étudiée et utilisée appelée réseaux de neurones récurrents à portes (Gated Recurrent Units : GRU).

3.5.4.3 Réseaux de neurones récurrents à portes

Les réseaux de neurones récurrents à portes (GRU) représentent la nouvelle génération de réseaux de neurones récurrents qui ont été proposés par *Cho et al.* en 2014 [48]. Ils ont été créés dans le but de gérer efficacement la mémoire à court et long terme grâce à leurs systèmes de portes. Ils permettent de mémoriser et d'oublier leurs états en fonction du signal d'entrée.

Les GRUs sont similaires aux LSTMs ; cependant, ils ont été affinés à l'aide d'une porte de *mise à jour* dans leur structure. La porte de mise à jour est une combinaison d'une porte d'entrée et d'une porte d'oubli [115–117].

La structure d'un GRU est illustrée dans la Figure 3.17 et les équations qui régissent ses fonctions sont définies comme suit [116] :

$$\begin{cases} z_t = \sigma(W_z \cdot [h_{t-1}, x_t]), \\ r_t = \sigma(W_r \cdot [h_{t-1}, x_t]), \\ \tilde{h}_t = tanh(W_h \cdot [r_t * h_{t-1}, x_t]), \\ h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t. \end{cases} \tag{3.13}$$

où

l'entrée (les caractéristiques) est représentée par x_t,
la prédiction par h_t,
z_t représente la porte de mise à jour,
r_t représente la porte de réinitialisation,
W_z vecteur des poids qui pondère l'entrée de la porte de mise à jour, W_r pondère l'entrée de la porte de réinitialisation et W_h pondère les données qui vont se combiner pour définir l'état caché courant,
σ et $tanh$ sont les fonctions d'activation utilisées dans cette structure afin de maintenir les informations circulant à travers le GRU dans une plage spécifique (valeurs normalisées) [117].

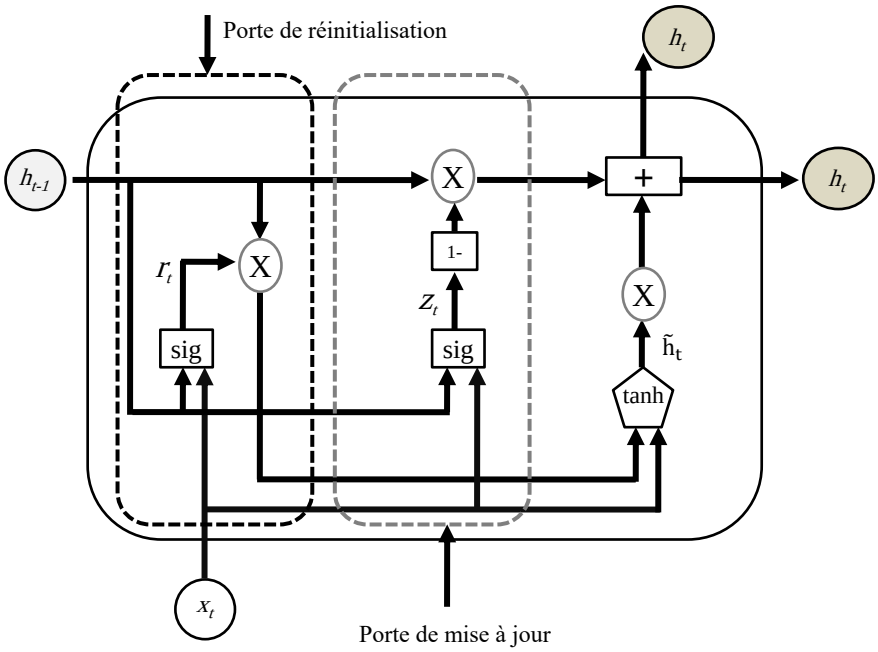


FIGURE 3.17 – Schéma d'une cellule GRU.

Comme illustré dans la Figure 3.17, la cellule GRU dispose de deux portes et un état en sortie.

Porte de réinitialisation : cette porte est utilisée pour aider le réseau à décider de la quantité d'*information passée* à oublier. L'état caché précédent h_{t-1}, concaténé avec les données d'entrée x_t, passent par une sigmoïde, dans le but de garder uniquement les données pertinentes, ensuite, une multiplication par l'état caché précédent h_{t-1} est effectuée : ainsi seulement les données importantes de l'état caché précédent seront conservées. A cet effet, cette porte permet de perdre une partie de l'état précédent.

Porte de mise à jour : cette porte agit de manière similaire aux portes d'oubli et d'entrée de LSTM : elle a pour rôle la prise de décision relative aux informations à conserver et de celles à oublier. Les données d'entrées x_t et l'état caché précédent h_{t-1} sont concaténés afin d'être introduites à une fonction sigmoïde dont le rôle est de déterminer les informations pertinentes.

Sortie du réseau GRU : une combinaison de l'entrée x_t du réseau et l'état caché précédent h_{t-1} (partiellement effacé par la porte de réinitialisation) est effectuée suivie d'une normalisation par la fonction d'activation *tangente*. Ensuite, une annulation de toutes les données décidées inutiles pour la prédiction (par la sortie de la porte de mise à jour) est réalisée puis l'état caché précédent est ajouté.

3.6 Conclusion

Ce chapitre a effectué un tour d'horizon des différents paradigmes de l'apprentissage machine et en particulier l'apprentissages supervisé. Il a aussi évoqué les réseaux de neurones récurrents, qui découlent de l'apprentissage profond, en particulier les architectures de type LSTM et GRU qui peuvent être utilisées pour produire des représentations vectorielles adaptées aux algorithmes de classification. Ceci offre ainsi la possibilité d'utiliser un modèle adapté aux données temporelles combiné avec un algorithme de classification donné.

Par ailleurs, dans le cadre d'une tâche de reconnaissance des commandes TV vocales enregistrées (séries temporelles), notre proposition s'est inspirée des réseaux de neurones récurrents, en particulier les LSTMs, les GRUs et les modèles bidirectionnels combinés avec un réseau MLP. L'implémentation de notre proposition, sera décrite dans le suivant chapitre.

CHAPITRE

4

REVUE DE LITTÉRATURE SUR LES SYSTÈMES ASR ARABE PAR APPRENTISSAGE PROFOND

Sommaire

4.1 Introduction

Le domaine de recherche sur les systèmes ASR a connu un grand développement lié à la découverte de nouveaux algorithmes et avancées en mathématiques, électronique et informatique. Parmi les techniques les plus émergentes utilisées dans le domaine de l'ASR, nous trouvons l'apprentissage profond [114]. Les Différents algorithmes et systèmes développés ont été appliqués pour les différentes langues parlées autour du monde, notamment la langue arabe, qui est l'une des langues les plus parlées et les moins considérées en termes de systèmes ASR développés.

La langue arabe est la langue officielle dans 22 pays, connus sous le nom du monde arabe (voir Figure 4.1).

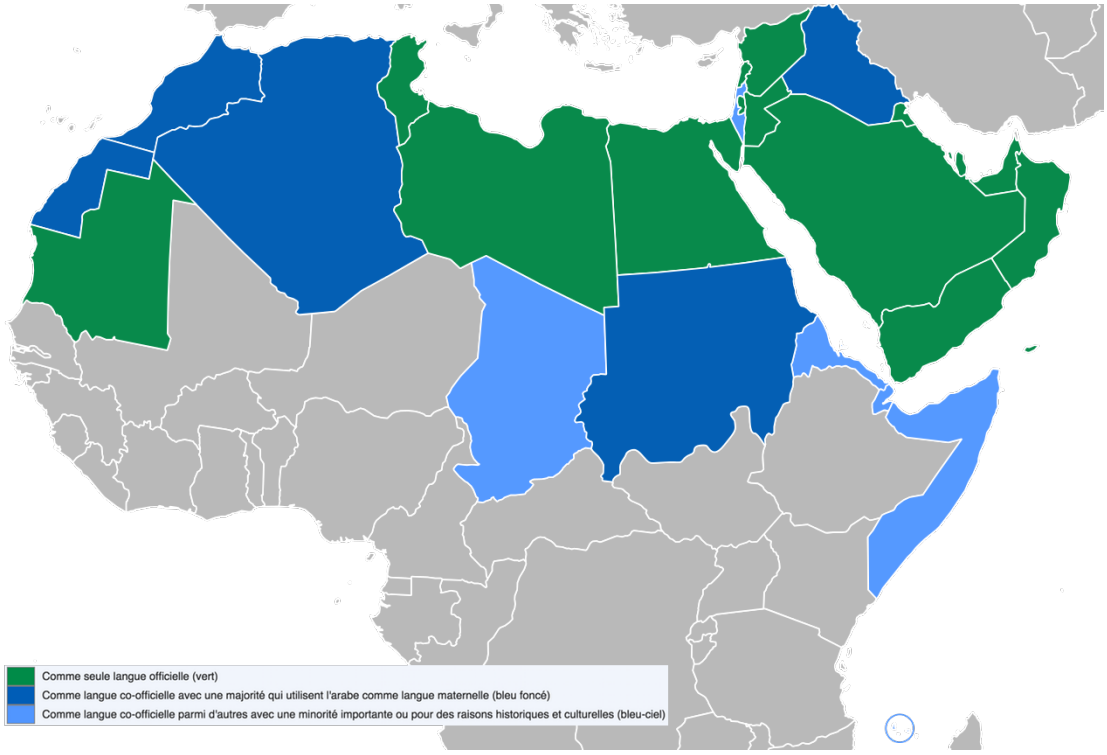


FIGURE 4.1 – Pays utilisant la langue arabe (Source : Université de Stockholm, Wikipédia).

Sur la base du nombre de locuteurs natifs, la langue arabe est l'une des six langues officielles des Nations Unies (l'Anglais, le Chinois, l'Espagnol, le Français, le Russe et l'Arabe) [118].

Ce chapitre donne un large aperçu des différents travaux effectués dans le domaine de l'ASR en langue arabe. En outre, il met en exergue les services et les boîtes à outils disponibles pour le développement des systèmes ASR en langue arabe. Principalement, il tente à exposer les éléments suivants :

- Les systèmes ASR arabe utilisant les techniques d'apprentissage machine classique ;
- Les différentes études sur l'ASR en langue arabe basées sur l'apprentissage profond ;
- Les services et les boîtes à outils disponibles pour développer un système ASR en langue arabe.

Il est à noter que cette revue de la littérature ne concerne qu'une partie succincte des travaux trouvés dans la littérature de la communauté, pour plus de détails nous renvoyons le lecteur à [14, 119–123].

4.2 Contexte général

La reconnaissance automatique de la parole, également connue sous plusieurs appellations, reconnaissance vocale ou reconnaissance vocale par ordinateur, a pour objectif la reconnaissance de la parole et de faire évoluer les techniques et les systèmes d'acquisition de la parole dans la machine.

La parole est le principal moyen de communication entre les humains ce qui motive les efforts de recherche pour lui permettre de devenir le moyen d'interaction homme-machine (Human Computer interaction : HCI) viable, considérée comme partie intégrante des HCIs. Principalement différentes catégories de systèmes ASR sont définies : 1) système à mots isolés, 2) système à mots connectés, 3) système à parole continue et 4) système à parole spontané, qui peuvent être dépendants ou indépendants du locuteur.

Dans ce chapitre, différents travaux basés essentiellement sur les réseaux de neurones, désignés comme une classe importante des techniques de reconnaissance les plus émergentes seront discutés.

Yu et Deng [14] ont donné un aperçu complet des progrès récents dans le domaine de la reconnaissance automatique de la parole en mettant l'accent sur les modèles d'apprentissage profond, y compris les réseaux de neurones profonds (DNN) et un bon nombre de leurs variantes. Il s'agit du premier livre d'ASR dédié à l'approche de l'apprentissage profond. En plus du traitement mathématique rigoureux du sujet, le livre présente également des idées et les fondements théoriques d'une série de modèles d'apprentissage profond très réussis.

Dans [124], les auteurs ont discuté la reconnaissance de phonèmes implémentée à l'aide de la technique des réseaux neuronaux en utilisant un algorithme puissant au stade du pré-traitement, via un filtrage passe-bas gaussien

pour améliorer la qualité du signal et réduire le bruit. Ils ont expliqué les étapes de la reconnaissance des phonèmes, les procédures de pré-traitement telles que l'obtention du signal, l'échantillonnage, la quantification, la détermination de l'énergie puis l'utilisation du réseau neuronal pour améliorer les performances de reconnaissance du système.

Ahmed et Ghabayen dans [125] ont proposé trois approches d'ASR en langue arabe. Pour la modélisation de la prononciation, ils ont proposé une génération variantes de prononciation avec arbre de décision. Pour la modélisation acoustique, ils ont proposé une approche hybride pour adapter le modèle acoustique natif en utilisant un autre modèle acoustique natif. Concernant le modèle de langage, il a été amélioré à l'aide du texte traité. Les résultats expérimentaux ont montré que l'approche du modèle de prononciation proposée a une réduction du taux d'erreur de mot (Word Error Rate : WER) d'environ 1%. La modélisation acoustique a réduit le WER de 1,2% et la modélisation du langage adapté montre une réduction du WER de 1,9%.

Une autre recherche effectuée par *Emami et Mangu* [126] a proposé l'usage de modèles de langage par réseau neuronal pour l'ASR en langue arabe, en utilisant une représentation distribuée des mots. Le modèle proposé permet une généralisation plus robuste et mieux adaptée au problème de la rareté des données. Les auteurs ont étudié différentes configurations du modèle probabiliste neuronal, en expérimentant des paramètres tels que l'ordre des N-gram, le vocabulaire de sortie, la méthode de normalisation, la taille et les paramètres du modèle. Des expériences ont été menées sur des informations (news) et des conversations diffusées en arabe. Les modèles de langage par réseau neuronal optimisé ont montré des améliorations significatives par rapport au modèle N-gram de base.

Une revue de littérature sur l'ASR arabe a été présentée par *Al-Anzi et Abu-Zeina* dans [120]. Elle expose le problème de la discrétisation facultative de l'écriture arabe, en mettant en évidence les progrès réalisés dans le domaine de l'ASR arabe qui incluent des jeux de données, des phonèmes, des modèles de langage et des modèles acoustiques. Les auteurs ont mis l'accent sur le problème de la pénurie des jeux de données vocaux (de parole continue) disponibles gratuitement. Ils ont montré également la nécessité de traiter de grands jeux de données ou ceux de référence (Benchmarks).

Kirchhof et al. dans [127] ont étudié les améliorations de la modélisation de la langue arabe en développant divers modèles de langue basés sur la morphologie. Ils ont présenté quatre approches différentes de la modélisation du langage basée sur la morphologie, y compris une nouvelle technique appelée modèles de langage factorisés.

Les auteurs de [128] ont présenté un système de reconnaissance de la parole arabe naturel à grand vocabulaire indépendant du locuteur. Ils ont précisé que leur travail était destiné à être un banc d'essai pour des recherches ultérieures sur le problème ouvert de la réalisation d'une conversation homme-machine en

langage naturel. Le système proposé a résolu un certain nombre de problèmes difficiles liés à la langue arabe, par exemple la génération d'une transcription entièrement vocalisée et d'un dictionnaire d'orthographe basé sur des règles. Le système ASR arabe développé est basé sur les outils Sphinx et les outils HTK. L'apprentissage du système a été réalisé sur 7.0 heures d'un jeu de données des informations diffusées en arabe de 7.5 heures et testé pendant la demi-heure restante.

Le premier outil de reconnaissance arabe basé sur SPHINX-IV a été proposé par *Hyassat et Abu Zitar* dans [129]. Les auteurs ont proposé une boîte à outils automatique capable de produire un dictionnaire de prononciation pour le Saint Coran et la langue arabe standard à la fois. Ils ont développé trois jeux de données, à savoir le Saint Coran HQC-1 d'environ 18.5 heures, le jeu de données de commande et de contrôle CAC-1 d'environ 1.5 heure et le jeu de données des chiffres arabes de moins d'une heure de mots. Les taux d'erreur obtenus par chaque système sont respectivement 46.182%, 1.818% et 0.787%.

Dans [130], les auteurs ont proposé une nouvelle approche multilingue pour l'ASR arabe dialectale. Ils ont construit plusieurs modèles acoustiques avec un jeu de données des informations diffusées de parole arabe standard moderne (Modern Standard Arabic : MSA). L'arabe familier égyptien a été choisi par les auteurs comme un exemple typique de dialecte arabe. Ils ont rassemblé un jeu de données de chiffres parlés (connectés) en arabe familier égyptien pour évaluer leur approche. Ils ont pu utiliser des modèles acoustiques de MSA en tant que modèles multilingues pour décoder l'arabe égyptien. Un taux de reconnaissance égal à 99,34% est atteint dans ce travail.

Les auteurs dans [131] ont conçu et présenté un jeu de données de parole en dialecte arabe algérien et un jeu de données de parole arabe standard moderne composé d'énoncés prononcés par 300 locuteurs natifs algériens sélectionnés dans onze régions d'Algérie. Ils ont montré que le modèle global de reconnaissance vocale monophone créé, avec un taux de précision de 91.65%, pourrait constituer un modèle de base utile pour d'autres études utilisant des systèmes ASR complexes dédiés à MSA.

4.3 Systèmes ASR arabe par apprentissage machine

Cette section se focalise essentiellement sur les techniques de reconnaissance basées sur les ANNs. De plus, les systèmes ASR peuvent être classifiés en différentes classes en fonction du type d'énoncés qu'ils peuvent reconnaître. Ces derniers se présentent en quatre types : 1) mots isolés, 2) mots connectés, 3) parole continue et 4) parole spontanée où quelques travaux de recherches sont examinés plus en détail dans les sous-sections suivantes :

4.3.1 Mots isolés

Amrouche et al. [132] ont présenté les résultats du réseau neuronal de régression générale (General Regression Neural Network : GRNN) appliqué à la reconnaissance de mots isolés arabe. Le modèle se compose de deux phases : une phase de pré-traitement qui consiste en une normalisation segmentaire et une extraction de caractéristiques et une phase de classification qui utilise des réseaux de neurones basés sur une estimation de densité non paramétrique. Afin d'accomplir une telle comparaison, le GRNN et le MLP ont été testés. Les résultats obtenus en utilisant un grand ensemble de chiffres arabe ont montré que les réseaux de neurones basés sur la régression générale améliorent davantage le taux de reconnaissance que ceux basés sur l'erreur de rétro-propagation.

Une nouvelle approche pour implémenter un système ASR pour la parole isolée arabe est décrite dans [133]. Elle est basée sur les réseaux de neurones Elman récurrents modulaires (Modular Recurrent Elman Neural Networks). Les résultats obtenus par cette approche peuvent concurrencer les approches traditionnelles de l'ASR basées sur les HMMs.

Un système ASR basé sur les RNNs a été proposé dans [134]. Il a été conçu et testé pour la reconnaissance automatique des chiffres arabe. Le système qui reconnaît les mots isolés a été implémenté à la fois en mode multi-locuteurs et en mode indépendant du locuteur. Les auteurs ont utilisé la technique MFCC pour l'extraction des caractéristiques et un RNN pour la classification des chiffres inconnus. Ce système de reconnaissance a atteint 99,5% de reconnaissance correcte des chiffres dans le cas du mode multi-locuteurs et 94,5% dans le cas du mode indépendant du locuteur.

Dans l'étude [135], un système ASR basé sur un ANN a été conçu et testé pour la reconnaissance automatique des chiffres arabe. Le système reconnaît vocalement les chiffres isolés dans un mode multi-locuteurs où un algorithme d'alignement temporel a été utilisé pour compenser les différences de longueur d'énonciation et les désalignements entre les phonèmes. Les caractéristiques des trames ont été extraites à l'aide de la technique MFCC pour réduire la quantité d'informations dans le signal d'entrée. Enfin, le réseau neuronal a classifié les chiffres inconnus avec un taux de reconnaissance de 99,48%.

Une approche générale de bout en bout pour l'apprentissage des séquences a été proposée dans [136]. Cette approche se base sur les LSTMs et les GRUs pour traiter la longueur de séquence non uniforme des énoncés de parole. Une phase d'extraction des caractéristiques pertinentes a été effectuée par la technique MFCC afin d'être utilisées par la suite par un LSTM/GRU. Ensuite un MLP a été utilisé pour la classification. Les différentes architectures au problème de la reconnaissance des chiffres arabe parlés [137]. Le système proposé a surpassé d'un grand écart les résultats publiés précédemment par d'autres auteurs sur le même jeu de données.

4.3.2 Mots connectés

Il est à noter que les recherches menées dans cette classe de mots, spécialement en langue arabe, sont relativement rares, en conséquence, on s'est limité de discuter uniquement les deux travaux ci-dessous.

Ghulam et al. [138] ont mené des expériences sur la tâche de reconnaissance des phonèmes connectés constituant des chiffres arabe. Chaque phonème a été modélisé par un HMM à trois états. La transition d'état était de gauche à droite. Les fonctions de densité de probabilité d'observation ont été modélisées à l'aide d'un modèle de mélange gaussien (Gaussian Mixture Model : GMM). Toutes les expériences d'apprentissage et de reconnaissance ont été mises en œuvre avec le package HTK [139]. L'apprentissage a été effectué en utilisant la parole normale, tandis que les tests ont été effectués en utilisant la parole normale et la voix désordonnée.

Les auteurs dans [140] ont développé un système ASR basés sur la technique HMM pour la langue arabe standard. Ils ont analysé l'effet du fenêtrage variable des trames (taille et période), le nombre de paramètres acoustiques résultant des méthodes d'extraction de caractéristiques traditionnellement utilisées en ASR, l'unité de reconnaissance de parole, le nombre gaussien par état HMM et le nombre de ré-estimations intégrées de l'algorithme Baum-Welch. Pour évaluer le système ASR proposé, un jeu de données de chiffres connectés à plusieurs locuteurs est collecté, transcrit et utilisé dans toutes les expériences, où le taux de reconnaissance était 94,02%.

4.3.3 Parole continue

Kabache et Guerti en [141] ont travaillé sur la reconnaissance des consonnes spécifiques à la langue arabe. Ils ont utilisé plusieurs techniques : PLP, RASTA-PLP et LPC dans la phase d'extraction et un réseau MLP pour la phase de reconnaissance. Les résultats obtenus ont montré que les techniques PLP et RASTA-PLP donnent de meilleurs résultats par rapport à la technique LPC. Ensuite, les auteurs ont précisé que la combinaison de RASTA-PLP avec l'énergie et le taux de passage par zéros augmente considérablement le taux de reconnaissance de leur système.

Les auteurs dans [142] ont présenté trois différents systèmes ASR basés sur des architectures MLP. Ils ont construit manuellement un jeu de données de phonèmes arabe. L'analyse MFCC a été utilisée pour extraire les caractéristiques du signal d'entrée. Les données ont été normalisées et utilisées pour l'apprentissage et le test des trois différents systèmes. Les taux de reconnaissance de ces systèmes étaient respectivement 47.52%, 44.58% et 46.63%.

Une comparaison de plusieurs techniques d'ASR appliquées à un ensemble limité des informations d'actualités diffusées en arabe a été présentée dans [143].

Les différentes approches ont toutes été appris sur 50 heures de transcription audio de la chaîne d'information Al-Jazeera. Les meilleurs résultats ont été obtenus à base des caractéristiques i-vector par un apprentissage utilisant le critère d'erreur téléphonique minimale combiné avec un apprentissage séquentiel d'un DNN.

Un système ASR des informations et des conversations diffusées en arabe utilisant la boîte à outils Kaldi a été proposé dans [144]. Le système a utilisé 200 heures de données issues du jeu de données GALE (GALE Phase 2 Arabic Broadcast Conversation Speech). La boîte à outils MADA pour la normalisation et la voyellisation de texte avec 36 phonèmes a été utilisée.

Dans l'étude [145], les auteurs ont développé leur système dans le cadre du défi de diffusion multi-genres (Multi-Genre Broadcast : MGB-2) 2016 en langue arabe qui a pour but la transcription parole-texte des enregistrements de la chaîne TV Aljazeera. Ils ont utilisé les caractéristiques dérivées de la technique GMM dans l'apprentissage d'un DNN, combiné avec un réseau neuronal à retardement (Time-Delay Neural Networks : TDNN). L'utilisation de deux approches différentes a pour but de phonétiser automatiquement les mots arabes, et la stratégie de sélection des données d'apprentissage pour les modèles acoustiques et linguistiques.

Un système ASR en arabe a été développé par les auteurs de [146]. Ils ont utilisé un jeu de données vocal de 1200 heures qui a été mis à disposition pour le défi de diffusion multi-genres arabe 2016 (MGB). Différentes topologies de DNNs ont été modélisées, notamment; propagation avant/arrière, Convolutionnelle, TDNN, LSTM, Highway LSTM (H-LSTM) et Grid LSTM (GLSTM). Les meilleures performances ont été fourni par le réseau neuronal G-LSTM avec un WER de 18,3%.

Les auteurs de [147] ont réalisé une extraction des caractéristiques par la technique MFCC multi-fenêtres (Multitaper Frequency Cepstral Coefficients : MFCC-MT) et des caractéristiques Gabor MFCC (GF-MFCC). Trois systèmes de classification ont été utilisés à savoir : CHMM (Continues Hidden Markov Models), DNN et un classifieur hybride HMM-DNN. Les auteurs ont utilisé un jeu de données de 3 heures contenant 440 phrases de 20 locuteurs avec des étiquettes générées par l'alignement de Viterbi en utilisant la boîte à outils HTK.

4.3.4 Parole spontanée

Graciarena et al. [148] ont présenté deux modèles de locuteurs acoustiques différents : les modèles de mélange gaussien cepstraux et les machines à vecteur de support à base de régression linéaire à maximum de vraisemblance et un ANN comme technique de combinaison. Ils ont présenté leurs résultats sur la partie arabe du jeu de données NIST, dans des conditions avec/sans bruit. La technique de combinaison a fournit une réduction significative de l'erreur sur

les systèmes individuels dans des conditions bruyantes.

Les auteurs dans [149] ont présenté une étude comparative entre deux moteurs d'identification automatique des locuteurs à partir de la parole arabe spontanée. Le premier moteur est basé sur les modèles de Markov cachés continus (Continious Hidden Markov Model : CHMM) tandis que le second est basé sur les ANNs. Les MFCCs ont été sélectionnés pour décrire le signal de parole. La distribution générale de densité gaussienne a été développée pour le moteur basé sur CHMM, alors que le réseau Elman a été développé pour le moteur basé sur un ANN. Le taux d'identification s'est avéré être de 100% pour les deux moteurs lors des expériences dépendant du texte. Cependant, pour les expériences indépendantes du texte, les performances du moteur basé sur CHMM ont surpassé celles du moteur basé sur un ANN.

Dans [150], les auteurs ont décrit leur système de compréhension de la langue arabe. Le système adapte l'approche stochastique à la grammaire probabiliste sans contexte (approche basée sur des règles). Les auteurs ont affirmé que leur système a surpassé certains systèmes internationaux existants dans la communauté.

4.4 Techniques d'apprentissage profond pour ASR arabe

L'apprentissage profond regroupe différentes techniques qui peuvent être appliquées aux problèmes d'ASR. Dans cette revue de littérature, nous nous concentrons davantage sur la technique des réseaux de neurones artificiels de base et cela va être consacré uniquement aux travaux de recherche en relation avec la langue arabe.

4.4.1 Réseaux de neurones

L'utilisation des modèles de langage à base d'un ANN pour les ASRs des informations et des conversations diffusées en arabe ont été étudiées par les auteurs dans [126]. Différentes architectures avec des configurations diverses d'un modèle neuronal probabiliste ont été discutées. La validation des expériences a été faite sur la base d'un jeu de données réelles. L'architecture ANN proposée basée sur les modèles de langage a surpassé le modèle de base N-gram.

Dans l'étude [151], les auteurs ont proposé un système hybride de reconnaissance des chiffres arabe intégrant un ANN et un HMM. La principale innovation dans ce travail réside dans l'utilisation d'un ANN optimal pour déterminer la bonne classe du chiffre en question. Comparativement à l'approche classique de Kohonen, les résultats obtenus par le système hybride proposé sont encourageants et satisfaisants.

La recherche [152] a abordé un problème d'ASR qui consiste à reconnaître les lettres arabe parlées, qui sont trois lettres de hijaiyah ('sa', 'sya' et 'tsa') qui ont une prononciation indentique lorsqu'elles sont prononcées par des locuteurs indonésiens mais qui ont en fait, différentes prononciations. La recherche a utilisé la technique MFCC pour l'extraction des caractéristiques et la technique ANN pour la classification. Les résultats obtenus par l'approche proposée étaient 92,42% comme précision moyenne et une précision de reconnaissance de chaque lettre 92,38%, 93,26% et 91,63% respectivement.

4.4.2 Réseaux de neurones récurrents

Les réseaux de neurones récurrents (RNNs) comptent parmi les meilleurs modèles appliqués aux données séquentielles. Ils permettent à la fois la propagation avant et la propagation arrière, ce qui est bien adapté aux données de la parole, qui peuvent être considérées comme des séquences temporelles.

Dans l'étude [134], un système ASR basé sur les RNNs a été conçu et testé pour reconnaitre les dix chiffres arabes (de zéro à neuf) (aussi discutée dans la section 4.3.1 des mots isolés). L'architecture RNN proposée a atteint 99,5% de reconnaissance correcte des chiffres dans le cas du mode multi-locuteurs et 94,5% dans le cas du mode indépendant du locuteur.

Une autre application des RNNs est proposée dans [133] (aussi discutée dans la section 4.3.1 des mots isolés), où les auteurs ont présenté une nouvelle approche pour implémenter un système ASR pour la parole isolée. Ils ont utilisé un RNN Elman modulaire, c'est-à-dire que pour chaque mot de l'ensemble du vocabulaire, un RNN séparé est appliqué. La modularité adopte une approche "diviser pour régner" en divisant le problème complexe en plusieurs problèmes bien plus simples. Le vocabulaire utilisé est composé de 6 mots arabe : "manzel" (maison), "hirra" (chat), "chajara" (arbre), "tariq" (route), "ghinaa" (chant), "zeina" (zeina étant un nom propre). L'apprentissage est divisé en deux étapes : un apprentissage cohérent composé de 48 énoncés et un apprentissage discriminant possédant 20 énoncés. Les résultats obtenus, entre 85% et 100%, ont été comparés avec ceux des HMMs.

4.4.3 Réseaux de neurones profonds

Les réseaux de neurones profonds (DNNs) sont des modèles d'apprentissage machine récents et extrêmement puissants. Récemment, ils sont employés dans le domaine de l'ASR [123, 153–155].

Dans [156], les auteurs ont proposé une architecture DNN entièrement connectée avec des unités Maxout. La technique MFCC a été utilisée pour l'extraction des caractéristiques du signal de parole. L'apprentissage et le test du DNN ont été effectués sur un jeu de données composé de phonèmes arabe consonantiques

enregistrés à partir de 20 locuteurs Malais. Les résultats de l'architecture proposée a été comparés avec la machine Boltzmann restreinte (Restricted Boltzmann Machine : RBM), le réseau de croyance profond (DBeN), le réseau neuronal convolutif (CNN) , le ANN et à l'auto-encodeur convolutif (Convolutional Auto-Encoder : CAE).

Les auteurs dans [157] ont testé 6 techniques DNN différentes sur un système ASR, en comparant les performances à plusieurs modèles d'apprentissage machine classiques selon le type de problème à classifier (classification binaire et multi-classes). Les résultats expérimentaux ont montré que les variantes des RNNs bidirectionnels ont atteint la meilleure précision sur le jeu de données commentaire arabe en ligne [158]. Celui-ci représente un référentiel à grande échelle de dialectes arabes avec des étiquettes manuelles pour 4 variétés de dialecte. Les résultats obtenus ont surpassé de manière significative toutes les méthodes de base concurrentielles.

Dans l'étude [15], les auteurs ont proposé une système ASR basé sur un DNN bout en bout qui s'appuie sur des architectures LSTM/GRU pour reconnaitre des commandes TV vocales en langue arabe. Les caractéristiques extraites par MFCC ont été introduites aux différentes structures profondes à savoir : avant, arrière et bidirectionnelle. Ensuite un MLP a été adopté pour classifier les commandes. Le système proposé a donné des précisions individuelles de reconnaissance correcte allant de 95.98% jusqu'à 99.64 %.

4.5 ASR arabe avec les services par apprentissage profond

Le développent d'un ASR basé sur l'apprentissage profond est une tâche difficile dont le succès dépend de la disponibilité d'un vaste référentiel de données d'apprentissage. La disponibilité de frameworks open source en apprentissage profond et d'interfaces de programmation d'application (Application Programming Interface : API) stimule le développement et la recherche des systèmes ASR arabe. Il existe plusieurs services et frameworks qui offrent aux développeurs de puissantes capacités d'apprentissage profond pour implémenter les différents systèmes ASR.

4.5.1 Services API

L'une des applications distinguées est le service "Cloud Speech-to-Text" de Google [159], qui utilise un DNN pour convertir la parole (entre autres la langue arabe) ou un fichier audio en texte. Ce service permet d'introduire le mot prononcé (à convertir en texte) au système de traduction qui va être par la suite traduit. Le service propose une API pour les développeurs avec plusieurs fonc-

tionnalités de reconnaissance. Un autre service est l'API "Microsoft Speech To Text " [160], offrant une aide aux développeurs pour créer des systèmes ASR en utilisant les DNNs. IBM cloud fournit aussi un service nommé "Watson" [161] pour l'ASR qui prend en charge la MSA.

4.5.2 Boîtes à outils

L'une des boîtes à outils les plus utilisées est nommée Kaldi [162], elle est gratuite et open source, destinée à être utilisée par les chercheurs et les professionnels de l'ASR utilisant un DNN prenant en charge la langue arabe. L'utilisation de Kaldi pour la construction d'un système ASR des informations diffusées en arabe est présentée dans [163].

Un autre travail présenté par*Manohar et al.* [164], où la boîte à outils Kaldi pour le défi de diffusion multi-genre arabe (MGB-3) qui traite le dialecte arabe égyptien a été utilisée. Les auteurs ont effectué une étude comparative sur l'efficacité de l'utilisation de Kaldi en adoptant le taux d'erreur de mot multi-référence (Multi-reference Word Error Rate : MR-WER) pour mesurer l'efficacité du système proposé.

4.5.3 Frameworks

Appelés aussi infrastructures logicielles, désignent un ensemble cohérent de composants logiciels structurels servant à créer les grandes lignes d'une architecture logicielle. L'un des principaux frameworks offrant des capacités d'apprentissage profond pour les développeurs est la Tensorflow [165]. Elle représente la principale bibliothèque Open Source pour le développement des modèles d'apprentissage machine.

4.6 Conclusion

Ce chapitre a pour but de donner un aperçu général sur les différents travaux en relation avec le développement des systèmes ASR, en particulier en lange arabe. En outre, il a présenté une brève revue de littérature des travaux utilisant différentes techniques et bien particulièrement les réseaux de neurones profonds. La littérature a couvert différentes recherches présentées selon : 1) les énoncés à reconnaitre : mots isolés, mots connectés, parole continue et parole spontanée d'une part, et d'autre part selon : 2) les techniques d'apprentissage utilisées. De plus les principales 3) infrastructures logicielles et services disponibles en ligne, dédiés au développement des systèmes ASR de la reconnaissance automatique de la parole ont été exposés.

CHAPITRE

5

RECONNAISSANCE DES CHIFFRES ET COMMANDES TV PARLÉS

Sommaire

5.1 Introduction

Ce chapitre propose une nouvelle approche pour la reconnaissance automatique de la parole, plus particulièrement celle des mots isolés. Dans ce contexte, les méthodologies proposées traitent deux problèmes distincts à savoir : 1) reconnaissance des chiffres parlés en langue arabe et 2) reconnaissance des commandes vocales TV prononcées en langue arabe. La reconnaissance sera effectuée dans le cadre des séries temporelles représentant les signaux vocaux enregistrés (chiffres/commandes TV).

L'implémentation de l'approche proposée pour la tâche de reconnaissance comprend trois phases principales :

La première phase représente la création du jeu de données avec la participation d'un nombre élevé de locuteurs pour pouvoir modéliser la variabilité entre les locuteurs (sexe et âge), elle se base sur le modèle *locuteur indépendant* (discuté dans le chapitre 2).

La deuxième consiste à appliquer plusieurs techniques d'extraction afin d'obtenir les caractéristiques les plus pertinentes pour les utilisées dans la tâche de reconnaissance. Dans ce but, nous utilisons trois types de techniques et par conséquent trois types de caractéristiques (MFCCs, FBs et delta/double delta).

Dans la troisième phase, les caractéristiques obtenues par les différentes techniques précédemment citées sont fournies à un réseau de neurones récurrent pour convertir le format variable des séries temporelles à un format des vecteurs de taille fixe. Ces derniers sont envoyés au classifieur comme entrée afin d'effectuer la classification des différents chiffres/commandes vocaux. A cet effet, deux types de réseaux de neurones récurrents et un réseau de neurones conventionnel sont utilisés. Les réseaux de neurones récurrents proposés sont de deux types LSTM et GRU alors que le réseau de neurones conventionnel proposé est de type MLP.

L'évaluation expérimentale et les résultats obtenus sont donnés sur la base de deux jeux de données réelles.

5.2 Implémentation de l'approche proposée

A la base de la structure générale des systèmes de reconnaissance de la parole, nous proposons un système composé de quatre modules distincts [31] :

La Figure 5.1 illustre le schéma du principe du système proposé.

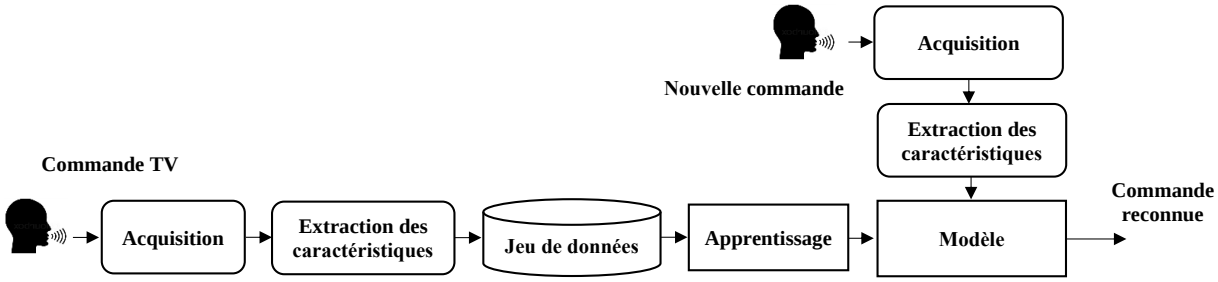


FIGURE 5.1 – Schéma de principe du système ASR proposé pour la reconnaissance des commandes TV.

Pour atteindre l'objectif visé par cette étude, la création de notre propre jeu de données (commandes TV prononcées en langue arabe) est jugée nécessaire pour la validation des méthodologies proposées. Ainsi, l'élaboration du jeu de données s'effectue en plusieurs phases en l'occurrence :

- Acquisition des différentes commandes ;
- Pré-traitement ;
- Extraction des caractéristiques ;
- Étiquetage.

La Figure 5.2 illustre le processus de création du jeu de données.

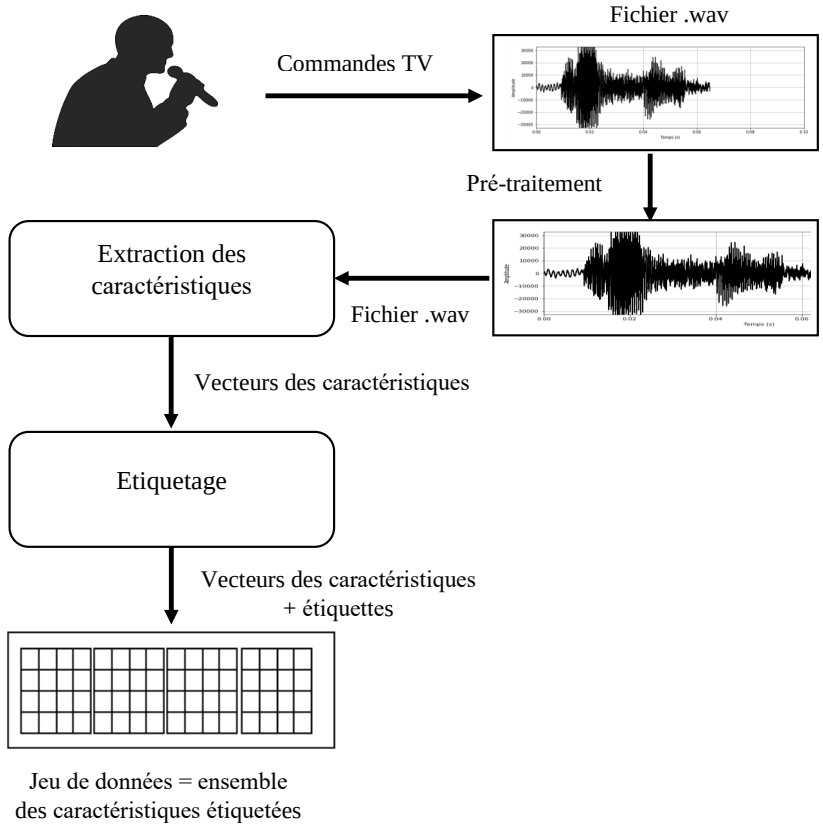


FIGURE 5.2 – Phases de création du jeu de données des commandes TV.

5.2.1 Acquisition

La première phase de la création du jeu de données consiste à réaliser l'enregistrement des différentes commandes par les différents locuteurs. Les signaux audio sont enregistrés par le bais d'un microphone dont la position et la qualité sont importantes pour la performance du système de reconnaissance. Le signal de parole analogique est initialement converti en un signal électrique analogique dans le microphone, ensuite le convertisseur analogique-numérique le convertit en échantillons numériques discrets. Dans les expériences de cette thèse, le signal de la parole a été enregistré par la définition des principaux paramètres présentés dans la Table 5.1.

TABLE 5.1 – Paramètres utilisés pour l'enregistrement du signal de parole.

Paramètres	**Valeurs**
Type de microphone	Microphone à main
Technologie	statique à petite membrane
Directivité	cardioïde

5.2.2 Pré-traitement

Le résultat de la phase précédente est fourni en entrée à cette deuxième phase de pré-traitement. Celle-ci consiste à réaliser un filtrage, effectué par un programme, afin de déterminer les zones de parole et les conserver tout en éliminant les zones de non parole (silence). Cette opération permet de réduire la longueur du signal de parole conduisant à une accélération de la phase d'extraction des caractéristiques les plus pertinentes. Le rôle de cette phase justifie son importance et sa nécessité.

5.2.3 Extraction des caractéristiques

Les fichiers filtrés (signaux vocaux) représentent l'entrée de la troisième phase qui a pour rôle l'extraction des caractéristiques les plus pertinents de chaque signal vocal contenant de grandes quantités d'informations. Cette phase est cruciale car elle met l'accent sur les informations contribuant à la détection du mot prononcé et ignore les informations non pertinentes du signal. Ceci permet de réduire la quantité d'informations à traiter.

Dans la littérature, différentes techniques d'extraction des caractéristiques pour représenter les aspects pertinentes du spectre de la parole à court terme sont définies. Un certain nombre de ces propositions sont motivées par les résultats de la recherche dans le domaine de la reconnaissance de la parole. Parmi ces propositions, les caractéristiques MFCCs qui sont les plus utilisées dans les systèmes ASR [166]. Une fois, cette phase d'extraction des caractéristiques est achevée, les vecteurs MFCCs de chaque mot enregistré sont obtenus. L'ensemble des MFCCs constituent le jeu de données des commandes non étiquetées.

Dans cette thèse, les expériences sont menées avec trois types de caractéristiques :

1. Caractéristiques Banc de Filtres (FBs),
2. Caractéristiques MFCCs ;
3. Caractéristiques Delta-Delta combinées avec les MFCCs.

5.2.4 Etiquetage

Cette phase se base sur l'étiquetage des caractéristiques pertinentes obtenues en attribuant une classe spécifique (type de commande) à chaque vecteur de caractéristiques à travers un programme développé dans ce but. La nécessité de cette phase se justifie par le type de problème à traiter. Ce dernier est défini étant un problème de classification supervisée.
A la fin de cette phase, le jeu de données des commandes vocales étiquetées est obtenu.

5.2.5 Construction du système de reconnaissance

La nature des données (multidimensionnelles) traitées ne permet pas d'utiliser un classifieur d'apprentissage machine ordinaire.

Par conséquent, l'approche proposée s'appuie sur une architecture récurrente profonde, plus précisément les réseaux à base de LSTM/GRU qui sont utilisés pour encoder les données et unifier leurs tailles. Ensuite, un réseau MLP est utilisé pour classifier les exemples (entrées vocales).

L'idée principale de l'approche proposée est de reconnaitre les commandes vocales enregistrées en traitant le signal vocal comme étant des séries temporelles. Ces dernières formant le jeu de données, sont utilisées pour entrainer le classifieur proposé, en l'occurrence le réseau MLP.

Le schéma bloc du modèle de reconnaissance proposé s'interprète par la Figure 5.3.

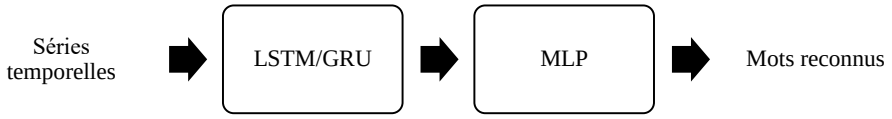


FIGURE 5.3 – Schéma bloc du modèle de reconnaissance proposé.

Premièrement, le réseau LSTM (ou le réseau GRU) code la séquence des MFCCs en tant que vecteur de taille fixe. Ensuite, le réseau MLP reçoit ce vecteur de taille fixe et réalise une classification des MFCCs. Cela s'effectue à travers différentes étapes comme suit :

1. **Encodage des données sous forme de matrice**
   Dans le cas du jeu de données Commande TV, l'encodage donne une matrice de dimension $(7000, 198, 13)$ (cf. section 5.3.2).
   où :
   - 7000 correspond au nombre des exemples de l'ensemble d'apprentissage,
   - 198 représente la taille de la plus longue séquence de caractéristiques MFCCs (correspondant à la commande enregistrée),
   - 13 désigne le nombre de caractéristiques MFCCs utilisés dans cette étude.
   Dans le cas où la taille de la séquence n'atteint pas 198, l'opération de *padding* [1] est effectuée à la séquence pour atteindre une taille maximale de 198.

2. **Encodage de séquence de taille fixe**
   La couche LSTM reçoit les données résultantes de l'étape précédente et tente d'encoder la séquence en tant que vecteur de taille fixe. Dans ce qui suit, le choix de l'approche bidirectionnelle est justifié par les résultats trouvés dans plusieurs recherches [107, 136].

1. Technique de remplissage par des zéros (zéro padding) consiste à ajouter aux N points du signal, une séquence de M valeurs nulles afin d'obtenir un plus grand nombre de points [69].

3. **Classification avec MLP**
   Le vecteur résultant de l'encodage de LSTM est envoyé au réseau MLP avec une seule couche cachée. Les différents paramètres ont été fixés intuitivement comme suit :

   (a) La sortie de la couche récurrente est définie par 100 neurones. Ces derniers sont fixés pour les couches récurrentes *forward* ou *backward* et $2 * 50$ pour les modèles bidirectionnels dont les sorties sont concaténées pour obtenir le vecteur final de sortie.

   (b) La taille de la couche cachée est fixée à 50 avec la fonction d'activation non linéaire "unité linéaire rectifiée" (ReLU). Le principal avantage de l'utilisation de la fonction ReLU par rapport aux autres fonctions d'activation est qu'elle n'active pas tous les neurones en même temps.

   (c) La taille de la couche de sortie est définie par le nombre de classes (10 classes : Digits/Commandes) à l'aide d'une fonction d'activation standard *softmax* [2] avec perte d'entropie croisée (cross entropy loss [3]).

Par ailleurs, tous les modèles sont entrainés en 50 époques et le meilleur modèle issu de la phase d'apprentissage est conservé pour l'évaluation finale (test). Afin d'avoir une meilleure estimation des performances du modèle, toutes les expériences sont répétées 10 fois. A la fin de ces 10 itérations, la moyenne des résultats obtenus durant chaque itération est calculée.

Dans le but de régulariser le réseau de neurones, la technique de *dropout* est implémentée. Cette technique permet de supprimer temporairement les neurones (cachées et visibles) du réseau, ainsi que toutes ses connexions entrantes et sortantes, comme illustré dans la Figure 5.4.

La technique dropout est utilisée pour éviter le sur-apprentissage qui s'interprète comme un apprentissage « par coeur » des données et peut empêcher la généralisation des données. Pour le choix des neurones à supprimer, il s'effectue aléatoirement. Ainsi, dans notre cas, deux couches de dropout sont insérées dans l'architecture proposée, la première couche dropout est insérée à la sortie du LSTM / GRU et la deuxième à la sortie de la couche cachée du MLP comme le montre la Figure 5.5.

Les deux couches dropout sont utilisées avec une probabilité de dropout de 0.2 et de 0.5 respectivement [167].

2. Fonction qui prend en entrée un vecteur z de K nombres réels, et le normalise en une distribution de probabilité constituée de K probabilités proportionnelles aux exponentielles des nombres d'entrée.

3. Mesure les performances d'un modèle de classification dont la sortie est une valeur de probabilité entre 0 et 1. La perte d'entropie croisée augmente à mesure que la probabilité prédite diverge de l'étiquette réelle.

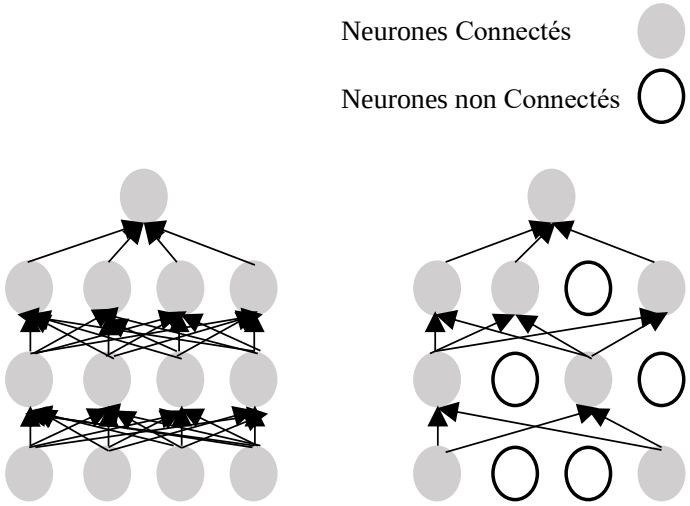


FIGURE 5.4 – Architecture du réseau de neurones : (a) avant dropout et (b) après dropout.

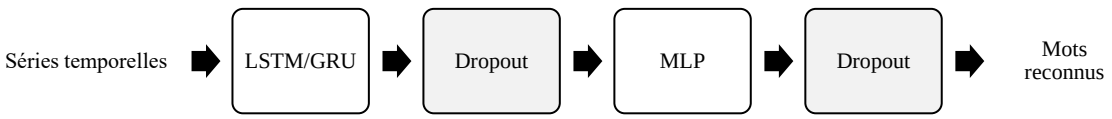


FIGURE 5.5 – Architecture proposée avec dropout.

La Figure 5.6 illustre les détails de l'architecture de l'approche proposée basée sur le réseau neuronal avec une topologie LSTM bidirectionnelle.

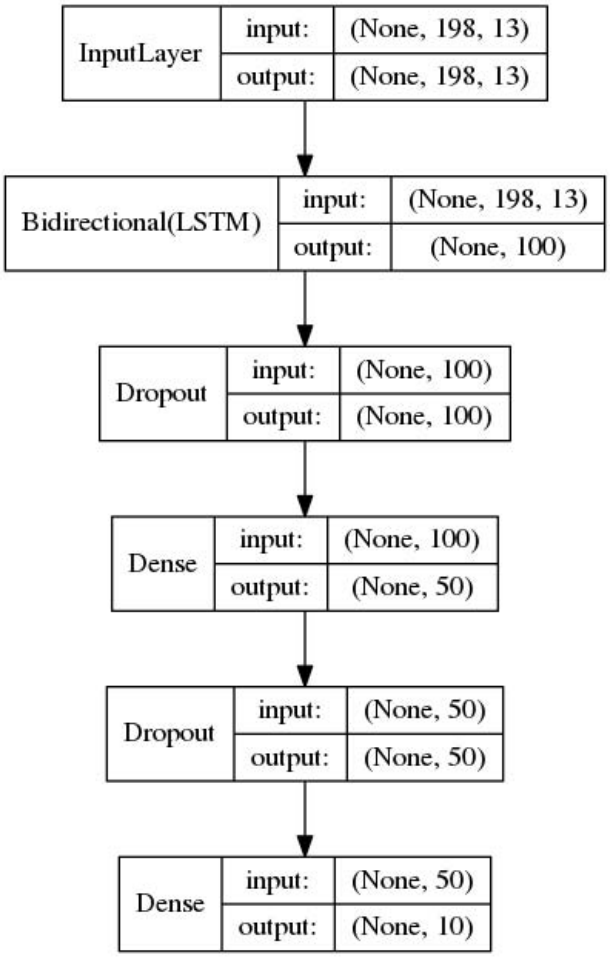


FIGURE 5.6 – Détails du modèle proposé pour le jeu de données commandes TV.

La Figure 5.6 montre que modèle dans le cas du jeu de données commandes TV, prend comme entrée les caractéristiques MFCCs issues de la phase d'extraction dont la taille est (None,198,13). Puis, un LSTM bidirectionnel va convertir (encoder) les entrées précédentes en un vecteur de taille fixe, une valeur de 100 dans ce cas est choisie. Ensuite, deux couches dropout sont insérées avant et après le classifieur MLP adopté dans les différentes expériences. Finalement, une dernière couche (sortie) est insérée à la fin du réseau proposé, prenant comme taille 10 qui représente les 10 classes reflétant les différentes commandes TV.

5.3 Données expérimentales

Pour évaluer, valider et justifier les différentes méthodologies proposées dans cette thèse, deux ressources de données ont été utilisées :

1. La première ressource représente le jeu de données des chiffres arabes parlés de 0 à 9 qui a été considéré comme Benchmark (jeu de données de référence) pour une validation initiale des différentes méthodologies proposées dans la présente thèse. Pour plus de détails, nous référons le lecteur aux références [19, 137].

2. La seconde ressource, quant à elle comporte les commandes vocales prononcées en langue arabe. La création de cette ressource fait partie des travaux de la présente thèse.

Les sous-sections 5.3.1 et 5.3.2 sont dédiées à la description des deux jeux de données utilisés dans nos expériences.

5.3.1 Jeu de données des chiffres parlés

Le premier jeu de données utilisé est celui des chiffres arabe parlés. Il regroupe les 10 premiers chiffres de 0 à 9 illustrés dans la Figure 5.7. Ce jeu de données a été conçu par le laboratoire d'Automatique et Signaux de l'Université de Badji-Mokhtar, Annaba, Algérie. Un nombre de 88 locuteurs définis par 44 hommes et 44 femmes de langue maternelle arabe ont été invités à prononcer tous les chiffres dix fois. Par ailleurs, le jeu de données comprend 8800 exemples (10 chiffres x 10 répétitions x 88 locuteurs) [19]. Chaque exemple contient les 13 caractéristiques MFCCs extraites de chaque chiffre enregistré.

Les caractéristiques extraites (MFCCs) de l'ensemble des chiffres parlés du premier jeu de données ont été calculées en utilisant les différents paramètres illustrés dans la Table 5.2 [19].

TABLE 5.2 – Paramètres de calcul des MFCCs pour le jeu de données des chiffres parlés.

Paramètres	**Valeurs**
Taux d'échantillonnage	11025 Hz, 16 bits
Filtre pré-accentuation	1-0.97*Z^{-1}
Fenêtrage	Hamming

Chiffres	Prononciation en arabe
0	صفر
1	واحـد
2	اثنان
3	ثلاثـة
4	أربعـة
5	خمسـة
6	ستـة
7	سبعـة
8	ثمانيـة
9	تسعـة

FIGURE 5.7 – Chiffres arabes et leurs prononciations.

5.3.2 Jeu de données des commandes TV

Pour créer le jeu de données des commandes TV, l'enregistrement des différentes commandes par les différents locuteurs a été effectué à différents lieux, à savoir :

- Établissements scolaires (primaire, moyen et lycée) : des élèves, des collégiens et des lycéens ont collaboré pour la réalisation des différents enregistrements.
- Université de Batna 2, département Génie Industriel : des enseignants du département ont contribué à la création du jeu de données. Aussi, des étudiants du département ont participé à l'enregistrement.
- Autres lieux, plus particulièrement les demeures.

Le jeu de données regroupe dix commandes prononcées en langue arabe pour commander vocalement à distance un téléviseur. Les différentes commandes sont illustrées dans la Figure 5.8.

N°	Commande Arabe	Signification
1	تشغيل	**Allumer**
2	اغلاق	**Eteindre**
3	أمام	**Suivant**
4	خلف	**Précédent**
5	رفع	**Augmenter**
6	خفض	**Diminuer**
7	صامت	**Muet**
8	قائمة	**Liste**
9	خروج	**Sortir**
10	ايقاف	**Quitter**

FIGURE 5.8 – Les commandes TV en langue arabe et leurs significations.

Par ailleurs, La création du jeu de données a fait intervenir plusieurs locuteurs natifs arabes (50 locuteurs / 50 locutrices), appartenant à des catégories d'âges distinctes comme le montre la Table 5.3.

TABLE 5.3 – Distribution des locuteurs du jeu de données commandes TV selon leurs genres et catégories.

Genres	**Masculin**		**Féminin**		
Catégories	Adulte	Enfant	Adulte	Enfant	*Total*
Locuteurs	37	13	31	19	100
Commandes prononcées	3700	1300	3100	1900	10000

Les différents locuteurs participant à l'enregistrement ont été invités à enregistrer 10 fois, chacune des 10 commandes illustrées sur la Figure 5.8. Ainsi, à la fin de l'enregistrement de chaque locuteur, un nombre de 100 fichiers de format *.wav* est obtenu. Par conséquent, le jeu de données contient 10000 exemples (10 commandes x 10 répétition x 100 locuteurs).

L'extraction des caractéristiques MFCCs consiste à découper le signal de la commande TV prononcée en blocs de taille fixe, typiquement 25ms, décalés les uns par rapport aux autres d'une durée constante de 10ms. Ensuite extraire un vecteur des caractéristiques cepstrales (les MFCCs) de chaque bloc en faisant intervenir les différents paramètres énumérés dans la Table 5.4.

TABLE 5.4 – Paramètres de calcul des MFCCs du jeu de données des commandes TV.

Paramètres	**Valeurs**
Taux d'échantillonnage	16000 Hz, 16 bits
Filtre pré-accentuation	1-0.97*Z^{-1}
Fenêtrage	Hamming
Taille de la fenêtre	256
Taille FFT	512
Filtres Linéaire	13
Filtres Log	27
Coefficients Cepstraux	13

5.3.3 Répartition des données

Dans une perspective d'apprentissage machine, le jeu de données des commandes TV est réparti en deux sous-ensembles nommés : *ensemble d'apprentissage* (X_{TR}) et *ensemble de test* (X_{TST}), ayants pour rôle de fixer les paramètres du modèle construit et de tester le modèle optimal (qui donne le meilleur score lors de la phase d'apprentissage) sur des données non-vues.

A cet effet, plusieurs techniques de validation peuvent être utilisées pour obtenir le modèle optimal. Pour plus de détails, nous référons le lecteur à [168].

Le premier sous-ensemble regroupe approximativement les deux tiers (2/3) des données tandis que le second contient le un tiers (1/3) restant, comme illustré par la Figure 5.9.

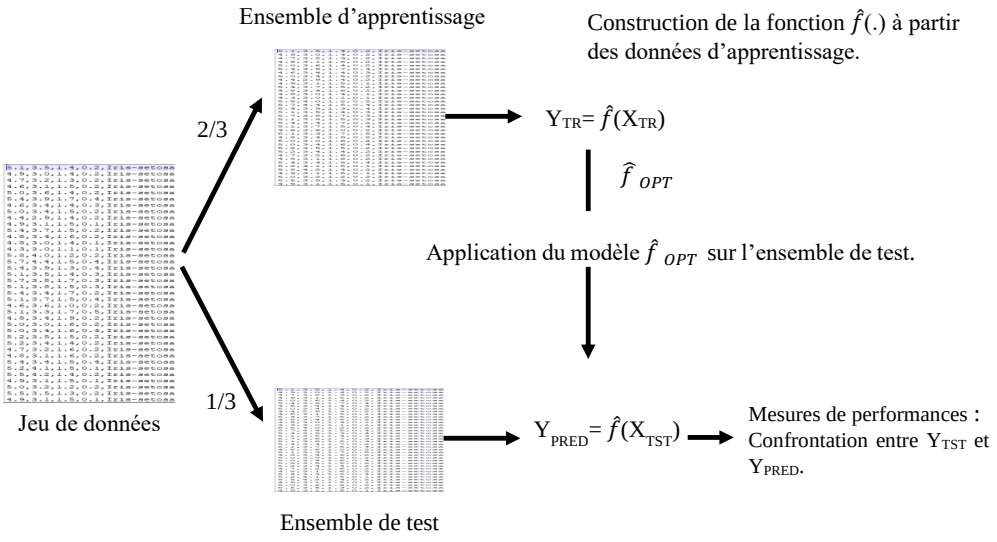


FIGURE 5.9 – Création et utilisation des ensembles d'apprentissage et de test.

où :
X_{TR} : caractéristiques d'entrée de l'ensemble d'apprentissage,
Y_{TR} : vecteur des classes de l'ensemble d'apprentissage,
X_{TST} : caractéristiques d'entrée de l'ensemble de test,
Y_{TST} : vecteur des classes de l'ensemble de test,
$\widehat{f}_{OPT}$: fonction optimale issue de la phase d'apprentissage,
Y_{PRED} : valeurs prédites par la $\widehat{f}_{OPT}$ sur l'ensemble de test.

5.3.3.1 Ensemble d'apprentissage

Le sous-ensemble d'apprentissage obtenu de la répartition du jeu de données, est utilisé pour construire un modèle approprié, entre autres, découvrir une relation prédictive entre les données liant l'exemple à son étiquette (sa classe). La plupart des approches cherchent dans les données d'apprentissage des relations empiriques qui ont tendance à s'adapter aux données ; cela signifie qu'ils peuvent identifier clairement des relations dans les données qui ne sont pas valables en général. A titre d'exemple, dans un modèle de réseau neuronal, l'ensemble d'apprentissage est utilisé pour fixer les poids du réseau.

5.3.3.2 Ensemble de test

Le sous-ensemble de test est un ensemble de données indépendant des données d'apprentissage, mais qui suit généralement la même distribution de probabilité que les données d'apprentissage. Afin d'évaluer l'erreur de prédiction du modèle appris, les performances doivent être mesurées sur de nouvelles données différentes de celle d'apprentissage. La précision du modèle sur l'ensemble de test peut donner une évaluation raisonnable des performances du modèle construit sur des nouvelles données.

5.3.4 Sélection du modèle

Apprendre les paramètres d'une fonction de prédiction et la tester sur les mêmes données est une erreur méthodologique : un modèle qui ne fait que répéter les étiquettes des échantillons qu'il vient de voir aurait un score parfait mais ne parviendrait pas à prédire quoi que ce soit d'utile sur des données encore invisibles. Cette situation s'appelle le sur-ajustement (*overfitting*). Pour l'éviter, il est important lors de l'exécution d'une expérience d'apprentissage automatique supervisée de conserver une partie des données disponibles sous la forme d'un ensemble de test (X_{TST}, Y_{TST}). L'organigramme 5.10 fait référence au flux de travail de la technique de la validation croisée typique durant la phase d'apprentissage (création du modèle). Les meilleurs paramètres peuvent être déterminés par des techniques de recherche de grille.

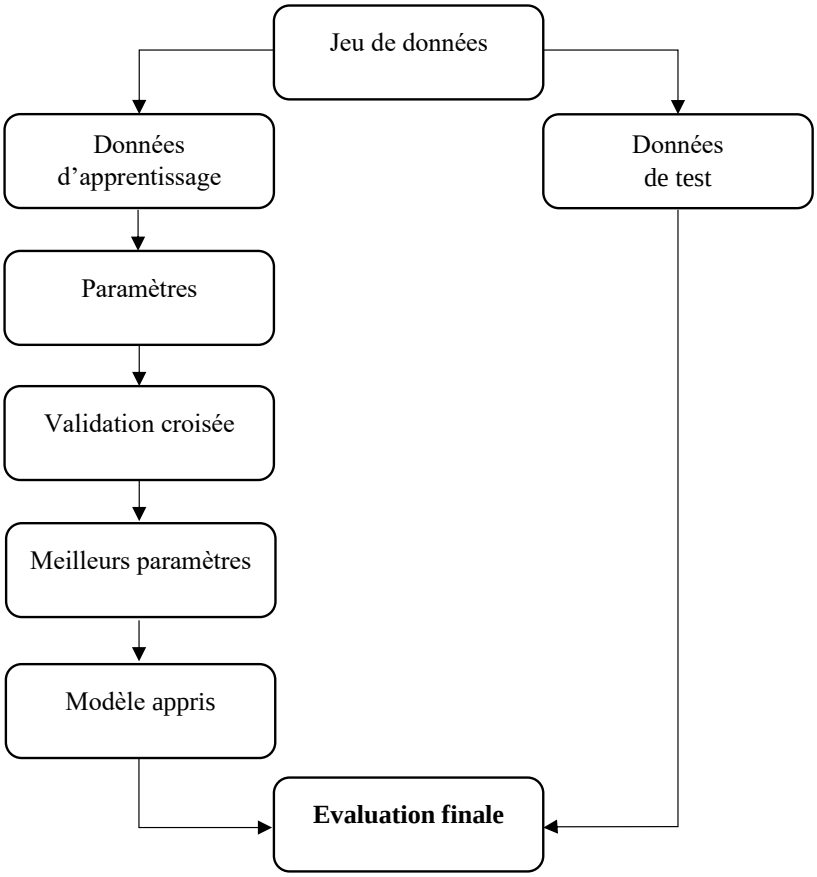


FIGURE 5.10 – Organigramme de validation croisée pour la création d'un modèle d'apprentissage.

Lors de l'évaluation des différents paramètres (*hyperparamètres*) pour les modèles de prédiction, qui est dans notre cas un classifieur, il existe toujours un risque de sur-ajustement de l'ensemble d'apprentissage car les paramètres peuvent être modifiés jusqu'à ce que le prédicteur fonctionne de manière optimale. De cette façon, les connaissances sur l'ensemble de test peuvent "s'infiltrer" dans le modèle, ainsi ses compétences de généralisation diminuent.

Pour résoudre ce problème, une autre partie de l'ensemble de données peut être considérée en tant que ensemble de validation, l'apprentissage se déroule sur l'ensemble d'apprentissage, ensuite l'évaluation s'effectue sur l'ensemble de validation, et lorsque l'expérience semble réussir, l'évaluation finale peut être effectuée sur l'ensemble de test. Cependant, en partitionnant les données disponibles en trois ensembles (Hold-out) [104], nous réduisons considérablement le nombre d'exemples pouvant être utilisés pour l'apprentissage du modèle, et les résultats peuvent dépendre d'un choix aléatoire particulier pour la paire d'ensembles (apprentissage, validation).

La solution à ce problème est une procédure appelée validation croisée [168]. Un ensemble de test doit toujours être présenté pour l'évaluation finale, mais

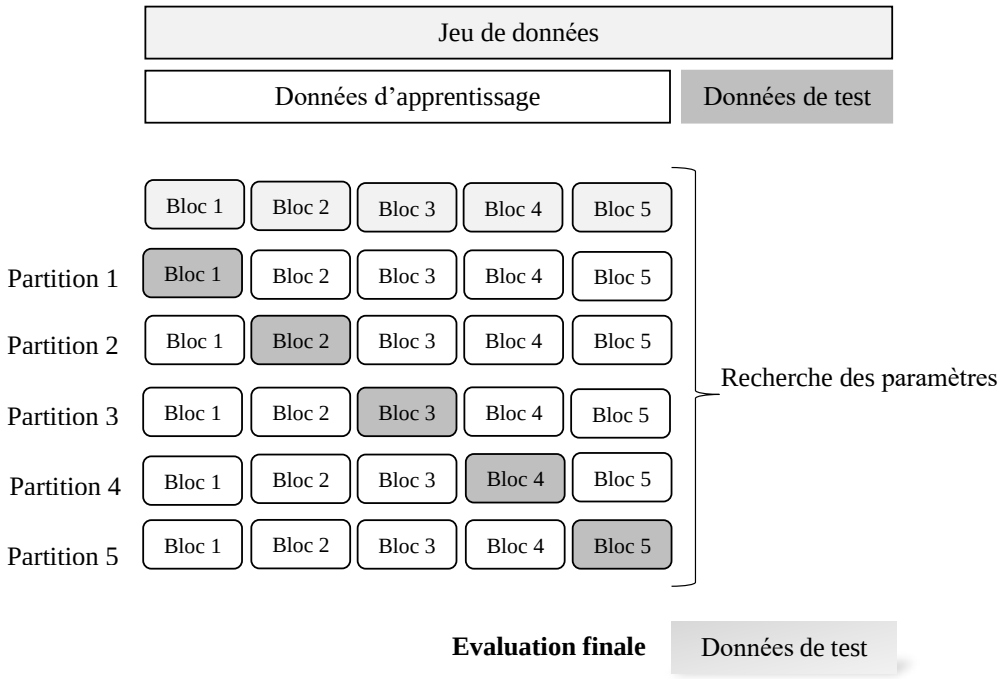


FIGURE 5.11 – Principe de la validation croisée à k-blocs.

l'ensemble de validation n'est plus nécessaire lors de la création de la validation croisée. Dans l'approche de base, appelée validation croisée k-blocs, l'ensemble d'apprentissage est divisé en k ensembles plus petits. La procédure suivante est suivie pour chacun des k-blocs (voir Figure 5.11) :

- le modèle est entrainé en utilisant $k-1$ blocs comme données d'apprentissage ;
- le modèle résultant est validé sur la partie restante des données, c'est-à-dire qu'il est utilisé comme ensemble de test pour mesurer la performance du modèle en cours.

La mesure de performance rapportée par la validation croisée de k-blocs est alors la moyenne des valeurs calculées dans la boucle. Cette approche peut être coûteuse en calcul, mais ne gaspille pas trop de données (comme c'est le cas lors du Hold-out), ce qui est un avantage majeur dans des problèmes où le nombre d'exemples est très petit.

5.4 Environnement de travail

Les différentes expériences sont exécutées sous *Python* [4]. Ce dernier s'impose actuellement dans plusieurs domaines comme un langage de programmation

4. Python est un langage de programmation interprété, portable, extensible, gratuit et dynamique permettant d'avoir une approche modulaire et orientée objet. Il est développé en 1989

de référence. Il représente l'outil parfait pour implémenter les techniques de l'apprentissage machine et de l'apprentissage profond [169].

Python permet de tirer profit des équipements informatiques modernes (CPU multicore, GPU), et aussi de gérer d'une manière efficace les grands ensembles de données (Big Data). Par ailleurs, différentes librairies logicielles offrant des solutions standardisées à de nombreux problèmes du ML, en particulier Scikit-learn, numpy, Tensorflow et Keras sont disponibles [170].

5.5 Critères de performance utilisés

L'évaluation de la performance d'un système ASR se mesure par plusieurs critères d'évaluation [171]. Celles utilisées pour l'évaluation de la qualité des prédictions du modèle proposé pour la reconnaissance des commandes vocales sont des métriques standard en l'occurrence : la *F-mesure* et le *taux d'erreur*. Ces mesures standard ont une interprétation intuitive, ce qui peut faciliter la compréhension de la façon dont le système ASR pourrait être amélioré [172]. Ces métriques sont disponibles dans la librairie appelée *scikit-learn*, une bibliothèque libre Python destinée à l'apprentissage machine, proposant de nombreuses fonctions intégrées pour analyser les performances des modèles.

En classification, chaque exemple dispose d'une classe réelle, qui lui est associée, et le système fournit une classe prédite. Ainsi, il est primordial de définir les paramètres, Vrais positifs, Faux positifs, Faux négatifs, Vrais négatifs, utilisés pour le calcul des métriques d'évaluation [173] :

- **Vrais positifs :** classe réelle = 1, classe prédite = 1 ;
- **Faux positifs :** classe réelle = 0, classe prédite = 1 ;
- **Faux négatifs :** classe réelle = 1, classe prédite = 0 ;
- **Vrais négatifs :** classe réelle = 0, classe prédite = 0.

Les paramètres "Vrais positifs" et "Vrais négatifs" désignent les exemples correctement prédits. Alors que les "Faux positifs" et "Faux négatifs" se produisent lorsque la classe réelle est en contradiction avec la classe prédite.
Ces différents paramètres peuvent également être expliqués par la Table 5.5.

5.5.1 Métriques utilisées

Pour évaluer les performances de reconnaissance de nos architectures, quatre indicateurs statistiques habituels sont utilisés dans cette thèse, à savoir : la précision, le rappel, la F-mesure (F1) et le taux d'erreur. Ils sont définis comme suit :

par *Guido van Rossum.*

TABLE 5.5 – Paramètres utilisés par les métriques d'évaluation.

		Classe prédite	
		Positif	Négatif
Classe réelle	Positif	*Vrais positifs*	*Faux négatifs*
	Négatif	*Faux positifs*	*Vrais négatifs*

5.5.1.1 Précision

La métrique "précision" obtenue sur une classe c donne le pourcentage des exemples prédits pertinents parmi tous les exemples affectés par le système d'ASR à la classe c. La métrique "précision" reflète ainsi la capacité du système à éviter les faux positifs pour cette classe. Elle est exprimée par l'équation (5.1) qui peut s'écrire sous la forme de l'équation (5.2) :

$$Précision_c = \frac{Nombre\ d'exemples\ correctement\ affectés\ à\ la\ classe\ c}{Nombre\ d'exemples\ affectés\ à\ la\ classe\ c}, \tag{5.1}$$

$$Précision_c = \frac{Vrais\ positifs}{Vrais\ positifs + Faux\ positifs}. \tag{5.2}$$

Ainsi, à partir de la précision calculée sur chaque classe, la précision moyenne exprimée par l'équation (5.3) peut être obtenue :

$$Précision\ moyenne = \frac{\sum_{c=1}^{C} Précision_c}{C}. \tag{5.3}$$

où
C représente le nombre de classes.

5.5.1.2 Rappel

Cette métrique permet de calculer pour une classe c, le pourcentage des exemples prédits pertinents par rapport au nombre total des exemples appartenant réellement à cette classe. Par ailleurs, cette métrique explique la capacité du système à éviter les faux négatifs pour une classe c. Elle s'exprime par les équations (5.4) ou bien (5.5) :

$$Rappel_c = \frac{Nombre\ d'exemples\ correctement\ affectés\ à\ la\ classe\ c}{Nombre\ d'exemples\ appartenant\ à\ la\ classe\ c}, \tag{5.4}$$

$$Rappel_c = \frac{Vrais\ positifs}{Vrais\ positifs + Faux\ négatifs}. \tag{5.5}$$

Similairement, le *rappel moyen* est obtenu par la moyenne du rappel calculé sur chaque classe. Il est exprimé par l'équation (5.6) :

$$Rappel\ moyen = \frac{\sum_{c=1}^{C} Rappel_c}{C}. \tag{5.6}$$

5.5.1.3 F-mesure

Aussi appelée F1, représente une métrique largement utilisée. Elle est obtenue par une moyenne harmonique des scores de précision moyenne et de rappel moyen à travers l'équation (5.7) [173] :

$$F\text{-}mesure = 2 * \frac{(Précision\ moyenne\ *\ Rappel\ moyen)}{(Précision\ moyenne\ +\ Rappel\ moyen)} \tag{5.7}$$

Ainsi, la F-mesure réalise une évaluation moyenne entre les différentes classes. Ce qui signifie que l'évaluation n'est pas globale. Par conséquent, la F-mesure donne une importance identique à toutes les classes.

5.5.1.4 Taux d'erreur

Il permet de calculer le pourcentage des commandes non correctement prédites par le système. Le calcul de cette métrique s'effectue sur la totalité du jeu de données en utilisant l'équation 5.8 :

$$Taux\ d'erreur = \frac{Nombre\ de\ prédictions\ fausses}{Nombre\ total\ de\ mots\ à\ reconnaître}. \tag{5.8}$$

5.6 Application 1 : Résultats obtenus avec le jeu de données des chiffres parlés

Les différentes expériences menées dans le cadre de cette thèse proposent l'utilisation des différents types d'encodeurs basés sur les réseaux LSTMs et GRUs avec les stratégies de recherche suivantes :

- **Recherche avant *(Forward)* :** commence avec un ensemble de caractéristiques vide. Dans chaque étape ultérieure, elle procède en ajoutant soit une caractéristique aléatoire (recherche plus rapide), soit une caractéristique qui optimise un critère quelconque (recherche plus lente), soit en ajoutant la caractéristique voisine par référence à l'ordre (ordre de la séquence temporelle).
  Dans ce cas de figure, l'inclusion se fait d'une manière ascendante en commençant de la caractéristique la plus ancienne allant vers la plus récente.

- **Recherche arrière *(Backward)* :** contrairement à la recherche avant, la recherche arrière s'effectue de manière descendante, en commençant par la caractéristique la plus récente tout en allant vers la plus ancienne.

- **Recherche bidirectionnelle *(Bidirectional)* :** cette stratégie remplace les techniques de recherche unique (avant et arrière) par deux recherches plus petites, l'une à partir du point initial et l'autre à partir du point objectif. La recherche se termine lorsque les deux recherches se croisent, c'est-à-dire, on combine les deux stratégies précédentes en même temps, une recherche dans le sens ascendant est effectuée parallèlement à une recherche au sens descendant.

Dans la présente thèse, les différentes architectures proposées avec les différentes stratégies de recherche (voir Figure 5.5) reçoivent en entrée les séquences de caractéristiques MFCCs et donnent en sortie la classe du chiffre parlé (mot à reconnaître). Initialement, le réseau de neurones encode la séquence des caractéristiques MFCCs sous forme d'un vecteur de taille fixe qui alimentera un classifieur MLP pour enfin classifier les chiffres parlés.

Pour cet objectif, le jeu de données des chiffres arabe parlés est partitionné en deux sous-ensembles : un sous-ensemble d'apprentissage contenant 6600 exemples ayant pour dimension $(6600, 93, 13)$, et un sous-ensemble de test ayant comme dimension $(2200, 93, 13)$.
où

- 6600 est le nombre des exemples (nombre de lignes) dans le sous-ensemble des données d'apprentissage,

- 2200 est le nombre des exemples dans le sous-ensemble des données de test,

- 93 est la taille de la plus longue séquence de caractéristique MFCCs (correspondant à la plus longue durée d'enregistrement). Lorsque la taille de la séquence est inférieure à 93, la séquence est complétée par un vecteur de zéros (*padding*) d'une taille de 13 jusqu'à ce que la taille maximale de 93 est atteinte,

- 13 est le nombre de caractéristiques MFCCs utilisés dans cette étude.

Les résultats obtenus sont comparés à ceux présentés dans [18] et [19], et comme indiqué dans la Table 5.6, en termes des critères de performance : précision, rappel, F1 et taux d'erreur.

TABLE 5.6 – Comparaison des résultats de l'approche proposée avec ceux des approches publiées dans [18,19] utilisant le même jeu de données des chiffres parlés. Les meilleurs résultats sont présentés en gras.

(a) Résultats de l'approche bidirectionnelle.

Chif.	Précision	Rappel	F1	%Erreur
0	95.98	98.73	97.33	4.14
1	98.92	99.86	99.39	1.09
2	99.63	98.91	99.27	1.09
3	98.67	97.45	98.06	2.55
4	99.64	99.32	99.48	0.68
5	99.32	99.91	99.61	0.68
6	99.81	96.36	98.06	3.64
7	98.55	98.82	98.68	1.45
8	98.32	98.41	98.36	1.68
9	98.96	99.91	99.43	1.05
Tout	98.77	98.77	98.77	1.23

(b) Comparaison avec les approches en termes de % succès.

Chif.	[18]	[19]	BiLSTM proposé
0	91.00	85.55	**95.86**
1	**99.00**	98.36	98.91
2	91.50	92.91	**98.91**
3	88.00	94.09	**97.45**
4	81.50	89.91	**99.32**
5	94.50	94.00	**99.32**
6	84.50	93.82	**96.36**
7	89.50	90.18	**98.55**
8	92.50	99.00	**98.32**
9	91.00	93.36	**98.95**
Tout	90.35	93.12	**98.77**

Pour une meilleure exploration des méthodologies présentées dans cette thèse, différentes architectures alternatives vont être examinées et testées sur le jeu de données des chiffres parlés. Il convient de noter que toutes les expériences sont menées uniquement avec les caractéristiques MFCCs. A partir de la Table 5.7, il est évident que l'architecture la plus performante est celle du "GRU-arrière", avec une légère amélioration par rapport aux architectures bidirectionnelles mais avec un très grand nombre de paramètres conduisant à un temps de calcul assez important.

TABLE 5.7 – Résultats moyennés sur 10 expériences sur le jeu de données des chiffres avec différents encodeurs. Les meilleurs résultats sont en gras.

Type d'encodeur	**Nombre de paramètres**	**F1**	**%Erreur**
*LSTM-bidirectionnel 2*50*	31.560	98.77	1.23
*GRU-bidirectionnel 2*50*	25.060	98.63	1.37
GRU-avant 100	40.060	97.26	2.74
GRU-arrière 100	40.060	**98.85**	**1.15**
LSTM-avant 100	51.560	97.41	2.59
LSTM-arrière 100	51.560	98.33	1.67

5.7 Application 2 : Résultats obtenus avec le jeu de données commandes TV

Dans cette partie d'expériences, les différentes stratégies de recherche expliquées en section 5.6 sont reconduites. Par conséquent, nous utilisons différents réseaux récurrents en l'occurrence : les LSTMs, les GRUs et les stratégies de recherche correspondantes *avant*, *arrière* et *bidirectionnelle*, sur le jeu de données créé (les commandes TV).

Pour cet objectif, le jeu de données des commandes TV est subdivisé en deux sous-ensembles : un sous-ensemble d'apprentissage contenant 7000 exemples ayant pour dimension $(7000, 198, 13)$, tandis que $(3000, 198, 13)$ est la dimension du sous-ensemble de test.
où

- 7000 est le nombre des exemples dans le sous-ensemble des données d'apprentissage,
- 3000 est le nombre des exemples dans le sous-ensemble des données de test,
- 198 est la taille de la plus longue séquence de coefficients MFCCs (correspondant à la plus longue durée d'enregistrement), lorsque la taille de la séquence est inférieure à 198, la séquence est complétée par un vecteur de zéros (*padding*) d'une taille de 13 jusqu'à ce que la taille maximale de 198 est atteinte,
- 13 est le nombre des caractéristiques MFCCs utilisées pour l'élaboration de ce jeu de données.

Les approches utilisées et leurs spécificités sont énumérées ci-dessous où les tailles des réseaux sont fixées expérimentalement :

- LSTM-bidirectionnel de taille 50 ;
- GRU-bidirectionnel de taille 50 ;
- GRU-bidirectionnel de taille 67 ;
- GRU-bidirectionnel de taille 80 ;
- LSTM-avant de taille 100 ;
- GRU-avant de taille 100.

Ces caractéristiques distinctes issues des différentes techniques, sont utilisées pour évaluer l'approche de reconnaissance adoptée dans cette thèse avec les différentes architectures utilisées, pour reconnaître les différentes commandes TV.

- 13 caractéristiques statiques : MFCCs ;
- 39 caractéristiques dynamiques (13 MFFCs+ 13 Delta + 13 delta delta) ;
- 40 caractéristiques Banc de filtres (FBs).

A partir des résultats obtenus par l'approche proposée, nous constatons que les résultats fournis par l'utilisation des FBs comme entrée, avec les mêmes paramètres que ceux utilisés par les MFCCs, sont moins efficaces. En utilisant un nombre élevé de caractéristiques (40 caractéristiques FBs), le réseau exige la définition de plus de paramètres qui lui permettent un meilleur apprentissage. En conséquence, nous avons effectué l'expérience en augmentant la taille du GRU à 100 neurones et la couche cachée suivante de 50 à 75. Les résultats obtenus sont reportés dans la Table 5.8.

TABLE 5.8 – Résultats obtenus avec plus de paramètres sur le jeu de données des commandes TV en utilisant GRU-bidirectionnel avec les FBs.

Type d'encodeur	**Nombre de paramètres**	**F1**	**%Erreur**
*GRU-bidirectionnel 2*100*	100,435	95.7	4.3

En effet, avec l'utilisation de plus de paramètres dans le réseau, il deviendra plus efficace, mais reste toujours moins performant avec les FBs qu'avec les MFCCs.

En outre, dans les expériences réalisées avec les 39 caractéristiques delta-delta, le réseau a besoin de plus de paramètres pour un apprentissage adéquat. Ainsi, il s'avère que les coefficients FBs calculés dans les premières étapes des MFCCs sont fortement corrélés [174], ce qui peut créer un effet négatif sur la précision des algorithmes d'apprentissage machine proposés (voir la Table 5.9).

TABLE 5.9 – Résultats moyennés sur 10 expériences sur le jeu de données des commandes TV avec les FBs. Les meilleurs résultats sont en gras.

Type d'encodeur	**Nombre de paramètres**	**F1**	**%Erreur**
*LSTM-bidirectionnel 2*50*	41.960	81.3	18.70
*GRU-bidirectionnel 2*50*	32.860	84.8	15.20
*GRU-bidirectionnel 2*67*	50.676	87.1	12.90
*GRU-bidirectionnel 2*80*	66.640	**88.6**	**11.40**
LSTM-avant 100	61.960	79.13	20.87
GRU-avant 100	47.860	85.26	14.73

En raison du problème de la forte corrélations des FBs, la transformation discrète en cosinus (DCT) est utilisée pour décorréler les coefficients des bancs de filtres et produire une représentation compressée. Typiquement, pour la reconnaissance automatique de la parole, les coefficients cepstraux résultants (dans notre cas 13) sont conservés et le reste est rejeté. Les résultats sont rapportés dans la Table 5.10.

TABLE 5.10 – Résultats moyennés sur 10 expériences sur le jeu de données des commandes TV avec différents encodeurs utilisant les MFCCs. Les meilleurs résultats sont en gras.

Type d'encodeur	Nombre de paramètres	F1	%Erreur
*LSTM-bidirectionnel 2*50*	31.560	96.23	3.77
*GRU-bidirectionnel 2*50*	25.060	96.14	3.86
*GRU-bidirectionnel 2*67*	40.224	96.93	3.07
*GRU-bidirectionnel 2*80*	54.160	97.06	2.96
LSTM-avant 100	51.560	97.03	2.97
GRU-avant 100	40.060	**97.11**	**2.89**

La Table 5.11 illustre les différents résultats obtenus en utilisant les coefficients dynamiques (double delta). Ces caractéristiques, ajoutées aux caractéristiques statiques (MFCCs) fournissent davantage d'informations sur l'évolution du signal.

TABLE 5.11 – Résultats moyennés sur 10 expériences sur le jeu de données des commandes TV avec différents encodeurs utilisant les MFCCs + delta-delta. Les meilleurs résultats sont en gras.

Type d'encodeur	Nombre de Paramètres	F1	%Erreur
*LSTM-bidirectionnel 2*100*	133.110	97.36	2.64
*GRU-bidirectionnel 2*100*	105.110	**97.66**	**2.34**
LSTM-avant 200	217.330	97.23	2.77
GRU-avant 200	169.330	97.27	2.73

Toutes les variantes des encodeurs présentées jusqu'à maintenant ont donné de bons résultats. Par ailleurs, les expériences réalisées dans cette partie, confirment que les architectures bidirectionnelles sont plus efficaces que les monodirectionnelles.

La comparaison des résultats illustrés par les Tables 5.9, 5.10 et 5.11 montre que ceux obtenus utilisant les caractéristiques dynamiques sont meilleurs que ceux obtenus par les MFCCs et les FBs. Ceci prouve l'utilité d'utiliser les caractéristiques dynamiques qui peuvent avoir un impact positif sur la précision de la classification. Il convient de noter que les expériences réalisées avec les caractéristiques dynamiques nécessitent davantage de paramètres pour permettre un meilleur apprentissage du réseau, ce qui augmente d'autant la complexité du calcul.

En outre, il est nécessaire de noter que les modèles *arrière* révèlent parfois des difficultés à converger et lorsqu'ils convergent, ils présentent des performances plus faibles. Le fait que les modèles *arrière* soient moins efficaces dans cette tâche peut expliquer les performances équivalentes relatives entre les modèles *avant* et les modèles *bidirectionnels* pour un nombre équivalent de paramètres.

En résumé, dans le cas du jeu de données des chiffres parlés, toutes les variantes des encodeurs donnent de bons résultats et surpassent ceux cités par [19] et [18] avec au moins 5% de précision. Alors que pour le jeu de données conçu, celui des commandes TV en langue arabe, tous les encodeurs montrent des résultats satisfaisants et comparables où les performances globales (en terme de F-mesure) sont supérieures à 97% pour tous les modèles.

5.8 Conclusion

Ce chapitre a donné les détails des résultats obtenus lors de l'implémentation des différentes méthodologies proposées pour traiter la problématique principale de cette thèse. En outre, il a mis en évidence les différentes variantes de l'approche proposée. Celle-ci se base essentiellement sur un système à deux blocs à savoir : les réseaux de neurones récurrents (LSTM / GRU) qui ont comme tâche de traiter les séries temporelles du signal de la parole, qui vont être introduites via plusieurs formes de caractéristiques, en l'occurrence : 1) MFCCs, 2) FBs et 3) delta-delta combinées avce les MFCCs à un classifieur universel (MLP) utilisé pour reconnaître les mots prononcés.

Pour tester et valider les méthodologies proposées, deux jeux de données ont été utilisés : le premier concerne un benchmark représenté par le jeu de données des chiffres arabe parlés, a été utilisé dans une première phase pour tester initialement les différentes approches développées, tandis que le second a été créé principalement pour répondre à la thématique principale de la thèse.

Les résultats expérimentaux obtenus sur les deux jeux de données, sont avérés prometteurs et présentent de bonnes performances pour toutes les architectures proposées.

CHAPITRE

6

CONCLUSION ET PERSPECTIVES

Sommaire

6.1 Conclusion

Au cours des dernières décennies, la recherche en reconnaissance automatique de la parole a été activement menée dans le monde entier, encouragée par les progrès du traitement du signal, des algorithmes, des architectures et du matériel. Les systèmes ASR ont été développés pour une large gamme d'applications, allant de la reconnaissance de mots-clés de petit vocabulaire, aux systèmes interactifs vocaux de commande et de contrôle de vocabulaire de taille moyenne, à la dictée vocale à grand vocabulaire, à la compréhension spontanée de la parole et à la traduction vocale avec un domaine limité.

Cette thèse a exploré le problème de la conception et la réalisation d'un système ASR pour commander automatiquement un téléviseur. Cet objectif a été réalisé sur la base d'un système à deux blocs qui a nécessité au préalable une étape de collecte de données afin de valider expérimentalement les différentes méthodologies proposées dans cette thèse. Ainsi, différents locuteurs avec différentes catégories d'âge et de genre ont participé à la création de ce jeu de données dans des conditions d'enregistrement normales (environnement réel). Par

la suite, et afin d'éliminer les silences associés à l'enregistrement des différentes commandes, une étape de pré-traitement par le biais d'un filtrage a été effectuée.

Concernant le premier bloc du système proposé, nous avons examiné plusieurs techniques d'extraction des caractéristiques, considérées comme les plus utilisées dans la littérature, à savoir : MFCC, banc de filtres et les caractéristiques dynamiques (double delta), qui ont servi à mieux caractériser les signaux de parole d'entrée. Cependant, la variabilité des séquences issues de la phase d'extraction, a exigé l'ajout d'une autre opération nommée padding, dont le rôle est l'uniformisation des tailles des séquences.

Ce qui est du second bloc, différents types de réseaux profonds, qui ont montré leurs efficacités dans les problèmes liés aux séquences temporelles, ont été implémentés et testés à savoir : LSTM et GRU avec trois différentes configurations (avant, arrière et bidirectionnelle). A noter que les résultats obtenus sont très satisfaisants, comparativement à quelques résultats de travaux reportés dans la littérature.

6.2 Perspectives

Le problème de la conception et de la réalisation d'un système ASR pour commander automatiquement un téléviseur par le biais de la parole naturelle a été abordé dans cette thèse. Nous mentionnons ici quelques directions de recherches futures possibles par rapport aux différentes méthodologies présentes dans cette étude.

1. *Techniques d'extraction :* le problème du choix de la technique d'extraction revient toujours dans l'esprit des concepteurs des systèmes ASR. A cet effet, l'exploration de nouveaux algorithmes de sélection de caractéristiques sera d'une grande importance pour bien définir les caractéristiques les plus pertinentes d'une part, et mieux s'adapter aux différentes situations d'enregistrements, notamment, dans le cas des environnements hautement bruités d'autres part, ceci afin d'améliorer la robustesse et l'exactitude des systèmes ASR. Dans ce contexte, nous proposons les x-vectors.

2. *Techniques de classification :* utilisation d'autres techniques relevant du domaine DL qui ne font pas appel à des techniques d'extraction.

3. *Langue d'entrée :* concevoir des systèmes qui se basent sur la langue courante parlée en prenant en compte les différents dialectes régionaux.

4. *Catégorie des locuteurs :* concevoir des systèmes destinés à d'autres catégories de personnes ayant des troubles ou des difficultés de prononciations (bégaiement, dyslalie, etc.)

Cette thèse constitue une synthèse et une extension utiles de la littérature actuelle sur la conception des systèmes ASR en langue arabe. Nous espérons que les futurs chercheurs trouveront l'utilité dans les méthodes proposées et découvriront de nouvelles directions stimulantes basées sur les lignes directrices définies dans les perspectives.

BIBLIOGRAPHIE

[1] Daniel Jurafsky and James H Martin. *Speech & language processing*. Pearson Education India, 2000. 1, 18, 29, 33

[2] Homer Dudley. Synthesizing speech. *Bell Laboratories Record*, 15 :98–102, 1936. 1

[3] Chin-Hui Lee, Frank K Soong, and Kuldip K Paliwal. *Automatic speech and speaker recognition : advanced topics*, volume 355. Springer Science & Business Media, 2012.

[4] Michelle Cutajar, Edward Gatt, Ivan Grech, Owen Casha, and Joseph Micallef. Comparative study of automatic speech recognition techniques. *IET Signal Processing*, 7(1) :25–46, 2013. 3

[5] Biing Hwang Juang and Laurence R Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3) :251–272, 1991. 3

[6] Mark Gales and Steve Young. *The application of hidden Markov models in speech recognition*. Now Publishers Inc, 2008. 3

[7] Talal Bin Amin and Iftekhar Mahmood. Speech recognition using dynamic time warping. In *2008 2nd international conference on advances in space technologies*, pages 74–79. IEEE, 2008. 3

[8] Geoffrey Zweig and Stuart Russell. Speech recognition with dynamic bayesian networks. 1998. 3

[9] Aravind Ganapathiraju, Jonathan E Hamaker, and Joseph Picone. Applications of support vector machines to speech recognition. *IEEE transactions on signal processing*, 52(8) :2348–2355, 2004. 3

[10] Aravind Ganapathiraju. *Support vector machines for speech recognition*. PhD thesis, Mississippi State University, 2019. 3

[11] Samira Hazmoune, Fateh Bougamouza, Smaine Mazouzi, and Mohamed Benmohammed. A new hybrid framework based on hidden markov models and k-nearest neighbors for speech recognition. *International Journal of Speech Technology*, 21(3) :689–704, 2018. 3

[12] Richard P Lippmann. Review of neural networks for speech recognition. *Neural computation*, 1(1) :1–38, 1989. 3

[13] Joe Tebelskis. *Speech recognition using neural networks*. PhD thesis, Carnegie Mellon University, 1995. 3

[14] Dong Yu and Li Deng. *Automatic speech recognition : A Deep Learning Approach*. Springer, 2016. 3, 54, 68

[15] Naima Zerari, Samir Abdelhamid, Hassen Bouzgou, and Christian Raymond. Bidirectional deep architecture for arabic speech recognition. *Open Computer Science*, 9(1) :92–102, 2019. 3, 76

[16] Albert Zeyer, Patrick Doetsch, Paul Voigtlaender, Ralf Schlüter, and Hermann Ney. A comprehensive study of deep bidirectional lstm rnns for acoustic modeling in speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2462–2466. IEEE, 2017. 3

[17] Yasufumi Moriya and Gareth JF Jones. Lstm language model adaptation with images and titles for multimedia automatic speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 219–226. IEEE, 2018.

[18] Nacereddine Hammami and Mokhtar Sellam. Tree distribution classifier for automatic spoken arabic digit recognition. In *2009 International Conference for Internet Technology and Secured Transactions,(ICITST)*, pages 1–4. IEEE, 2009. 4, 97, 102

[19] Nacereddine Hammami and Mouldi Bedda. Improved tree model for arabic speech recognition. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 5, pages 521–526. IEEE, 2010. 4, 87, 97, 102

[20] Lawrence Rabiner. *Fundamentals of speech recognition*. PTR Prentice Hall, 1993. 7, 8, 16

[21] William J Hardcastle and Alain Marchal. *Speech production and speech modelling*, volume 55. Springer Science & Business Media, 2012. 8

[22] Nacereddine Hammami. *Contribution to the automatic speech recognition of arabic language ans its applications*. PhD thesis, University of Annaba, 2014. 8, 10

[23] Chetouani Mohamed. *Codage neuro-prédictif pour l'extraction de caractéristiques de signaux de parole*. PhD thesis, Université Pierre & Marie Curie, 2004. 10, 24, 33

[24] Asmaa Amehraye. *Débruitage perceptuel de la parole*. PhD thesis, Télécom Bretagne, 2009. 11, 28, 29

[25] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon. *Spoken language processing : guide to algorithms and system development*. Prentice Hall, 2001. 11, 25

[26] René Boite. *Traitement de la parole*. PPUR presses polytechniques, 2000. 11

[27] Abdenour Hacine-Gharbi. *Sélection de paramètres acoustiques pertinents pour la reconnaissance de la parole*. PhD thesis, Université d'Orléans, France et Université Ferhat Abbas-Sétif, Algérie, 2012. 12, 17

[28] George A Miller and Joseph CR Licklider. The intelligibility of interrupted speech. *The Journal of the Acoustical Society of America*, 22(2) :167–173, 1950. 12

[29] Laurent Buniet. *Traitement automatique de la parole en milieu bruité : étude de modèles connexionnistes statiques et dynamiques*. PhD thesis, Université Henri Poincaré-Nancy 1, France, 1997. 13

[30] Othman Lachhab. *Reconnaissance Statistique de la Parole Continue pour Voix Laryngée et Alaryngée*. PhD thesis, Université Mohamed V-Agdal, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes, 2017. 13

[31] Joseph Mariani. *Reconnaissance de la parole*. Hermès science, 2002. 15, 80

[32] Atma Prakash Singh, Ravindra Nath, and Santosh Kumar. A survey : Speech recognition approaches and techniques. In *2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pages 1–4. IEEE, 2018. 15

[33] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2) :257–286, 1989. 16

[34] Ken H Davis, R Biddulph, and Stephen Balashek. Automatic recognition of spoken digits. *The Journal of the Acoustical Society of America*, 24(6) :637–642, 1952. 16

[35] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90) :297–301, 1965. 16

[36] B Boguert, MJ Healey, and JW Tukey. The frequency analysis of time series for echoes : Cepstrum pseudo-autocovariance cross-cepstrum and shape cracking. *Ed. New York : Wiley*, 1963. 16

[37] John D Markel and AH Jr Gray. *Linear prediction of speech*, volume 12. Springer Science & Business Media, 1976. 16, 25

[38] Fumitada Itakura. Analysis synthesis telephony based on the maximum likelihood method. In *The 6th international congress on acoustics, 1968*, pages 280–292, 1968. 16

[39] Bishnu S Atal and Suzanne L Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *The journal of the acoustical society of America*, 50(2B) :637–655, 1971.

[40] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1) :43–49, 1978. 16

[41] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state markov chains. *The annals of mathematical statistics*, 37(6) :1554–1563, 1966. 16

[42] Kai-Fu Lee. *Automatic speech recognition : the development of the SPHINX system*, volume 62. Springer Science & Business Media, 1988. 16

[43] Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1997. 16

[44] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4) :115–133, 1943. 16, 55

[45] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997. 16, 60

[46] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition : The shared views of four research groups. *IEEE Signal processing magazine*, 29(6) :82–97, 2012. 16

[47] Santiago Fernández, Alex Graves, and Jürgen Schmidhuber. An application of recurrent neural networks to discriminative keyword spotting. In *International Conference on Artificial Neural Networks*, pages 220–229. Springer, 2007. 16

[48] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv :1406.1078*, 2014. 16, 58, 63

[49] Claude Barras. *Reconnaissance de la parole continue : adaptation au locuteur et contrôle temporel dans les modèles de Markov cachés*. PhD thesis, Université Paris VI, 1996. 17

[50] Thomas Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53) :370–418, 1763. 18

[51] Christian Raymond. *Décodage conceptuel : co-articulation des processus de transcription et compréhension dans les systèmes de dialogue*. PhD thesis, Université d'Avignon et des Pays de Vaucluse, 2005. 18, 23

[52] Jean-Paul Haton, Christophe Cerisara, Dominique Fohr, Yves Laprie, and Kamel Smaïli. *Reconnaissance automatique de la parole : Du Signal à son Interprétation*. Dunod, 2006. 19

[53] Stanley Mneney. *An Introduction to Digital Signal Processing*. River Publishers, 2009. 20, 21

[54] Joseph Mariani. *Analyse, synthèse et codage de la parole*. Hermès science, 2002. 21, 24, 25

[55] Vincent Jousse. *Identification nommée du locuteur : exploitation conjointe du signal sonore et de sa transcription*. PhD thesis, Université du Maine, 2011. 22

[56] Michael F McTear. *Spoken dialogue technology : toward the conversational user interface*. Springer Science & Business Media, 2004. 23

[57] Randy Allen Harris. *Voice interaction design : crafting the new conversational speech systems*. Elsevier, 2004. 23

[58] George Saon and Jen-Tzung Chien. Large-vocabulary continuous speech recognition systems : A look at some recent advances. *IEEE Signal Processing Magazine*, 29(6) :18–33, 2012. 23

[59] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4) :357–366, 1980. 25, 28, 37

[60] Hynek Hermansky, B Hanson, and Hisashi Wakita. Perceptually based linear predictive analysis of speech. In *ICASSP'85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 509–512. IEEE, 1985. 25

[61] Hynek Hermansky and Nelson Morgan. Rasta processing of speech. *IEEE transactions on speech and audio processing*, 2(4) :578–589, 1994. 25, 28

[62] Bishnu S Atal and Manfred R Schroeder. Adaptive predictive coding of speech signals. *Bell System Technical Journal*, 49(8) :1973–1986, 1970. 25

[63] O Douglas and O Shaughnessy. Speech communications : Human and machine. *IEEE press, Newyork*, pages 367–433, 2000. 25

[64] Sadaoki Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(2) :254–272, 1981. 27, 36

[65] Hynek Hermansky. Perceptual linear predictive (plp) analysis of speech. *the Journal of the Acoustical Society of America*, 87(4) :1738–1752, 1990. 27

[66] Hynek Hermansky, Nelson Morgan, Aruna Bayya, and Phil Kohn. Rasta-plp speech analysis. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, volume 1, pages 121–124, 1991. 28

[67] Jean-Claude Junqua and Jean-Paul Haton. *Robustness in automatic speech recognition : fundamentals and applications*, volume 341. Springer Science & Business Media, 2012. 29

[68] Hassan Soliman Moftah Mohsen. *Arabic dialect identification using unsupervised motif discovery in speech Signal*. PhD thesis, Université Ain Shams, Egypt, 2018. 30, 33, 36

[69] E. Tisserand, J.F. Pautex, and P. Schweitzer. *Analyse et traitement des signaux - 2e éd. : Méthodes et applications au son et à l'image*. Sciences de l'ingénieur. Dunod, 2009. 30, 83

[70] KM Muraleedhara Prabhu. *Window functions and their applications in signal processing*. CRC press, 2013. 32

[71] Ooi Chia Ai, M Hariharan, Sazali Yaacob, and Lim Sin Chee. Classification of speech dysfluencies with mfcc and lpcc features. *Expert Systems with Applications*, 39(2) :2157–2165, 2012. 32

[72] K Sreenivasa Rao and Shashidhar G Koolagudi. *Robust emotion recognition using spectral and prosodic features*. Springer Science & Business Media, 2013. 32

[73] Zakir Ali, Arbab Waseem Abbas, TM Thasleema, Burhan Uddin, Tanzeela Raaz, and Sahibzada Abdur Rehman Abid. Database development and automatic speech recognition of isolated pashto spoken digits using mfcc and k-nn. *International Journal of Speech Technology*, 18(2) :271–275, 2015.

[74] Frank J Owens. *Signal processing of speech*. Macmillan International Higher Education, 1993. 33

[75] Stanley Smith Stevens, John Volkmann, and Edwin B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3) :185–190, 1937. 33

[76] V.S.Dharun M.E. *Intelligent system speech recognition*. PhD thesis, Manonmaniam Sundaranar University, 2012. 33, 34

[77] Derzu Omaia, JanKees vd Poel, and Leonardo V Batista. 2d-dct distance based face recognition using a reduced number of coefficients. In *2009 XXII Brazilian Symposium on Computer Graphics and Image Processing*, pages 291–298. IEEE, 2009. 35

[78] Julien Epps and Eliathamby Ambikairajah. Use of the discrete cosine transform for gene expression data analysis. In *Proc. Workshop on Genomic Signal Processing and Statistics*. Citeseer, 2004. 35

[79] Eliathamby Ambikairajah, Haizhou Li, Liang Wang, Bo Yin, and Vidhyasaharan Sethu. Language identification : A tutorial. *IEEE Circuits and Systems Magazine*, 11(2) :82–108, 2011. 36

[80] Jean-Claude Junqua and Jean-Paul Haton. On the use of a robust speech representation. In *Robustness in Automatic Speech Recognition*, pages 233–272. Springer, 1996.

[81] Stephen Marsland. *Machine learning : an algorithmic perspective*. CRC press, 2015. 40, 42, 52

[82] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2010. 41, 45

[83] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL : McGraw Hill*, 45(37) :870–877, 1997. 41

[84] Arthur L Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3) :210–229, 1959. 41

[85] Kevin P Murphy. *Machine learning : a probabilistic perspective*. MIT press, 2012. 42

[86] Rudolph Russell. *Machine Learning : Step-by-Step Guide To Implement Machine Learning Algorithms with Python*. 2018. 44, 48

[87] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning : An introduction.*, volume 135. MIT press Cambridge, 1998. 45, 47, 48

[88] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006. 45

[89] Andreas C Müller, Sarah Guido, et al. *Introduction to machine learning with Python : a guide for data scientists*. " O'Reilly Media, Inc.", 2016. 45, 46

[90] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1) :1–130, 2009. 47

[91] Mohri Mehryar, Rostamizadeh Afshin, and Talwalkar Ameet. *Foundations of machine learning*. MIT press, 2018. 49

[92] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553) :436–444, 2015. 50

[93] Li Deng and Dong Yu. Deep learning : methods and applications. *Foundations and trends in signal processing*, 7(3–4) :197–387, 2014. 51

[94] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009. 51

[95] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5) :5947, 2009. 51

[96] Frank Seide, Gang Li, and Dong Yu. Conversational speech transcription using context-dependent deep neural networks. In *Twelfth annual conference of the international speech communication association*, 2011. 51

[97] Hamed Habibi Aghdam and Elnaz Jahani Heravi. Guide to convolutional neural networks. *New York, NY : Springer*, 10 :978–973, 2017. 51

[98] Howard B Demuth, Mark H Beale, Orlando De Jess, and Martin T Hagan. *Neural network design*. Martin Hagan, 2014. 52

[99] Laurene V Fausett. *Fundamentals of Neural Networks : Architectures, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, 1993. 53, 54

[100] Mohamed Yessin Ammar. *Mise en œuvre de réseaux de neurones pour la modélisation de cinétiques réactionnelles en vue de la transposition batch/continu*. PhD thesis, Institut National Polytechnique, Toulouse, France, 2007. 55, 56

[101] F Rosenbaltt. The perceptron–a perciving and recognizing automation. *Report 85-460-1 Cornell Aeronautical Laboratory, Ithaca, Tech. Rep.*, 1957. 56

[102] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998. 56

[103] DS Broomhead and D Lowe. Multivariable functional interpolation and adaptive networks, complex systems, vol. 2. 1988. 56

[104] Hassen Bouzgou. *Automatic Analysis of Highdimensional Signals : Advanced Wind Speed Forecasting Techniques*. LAP LAMBERT Academic Publishing, 2012. 57, 58, 92

[105] Charles Pelletier. *Classification des sons respiratoires en vue d'une détection automatique des sibilants*. PhD thesis, Université du Québec à Rimouski, 2006. 57

[106] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088) :533–536, 1986. 57

[107] Vedran Vukotić, Christian Raymond, and Guillaume Gravier. A step beyond local observations with a dialog aware bidirectional gru network for spoken language understanding. 2016. 58, 83

[108] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013. 58

[109] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11) :2673–2681, 1997. 59

[110] Killian Janod. *La représentation des documents par réseaux de neurones pour la compréhension de documents parlés*. PhD thesis, Université dŠAvignon et des Pays de Vaucluse, France, 2017. 60, 61

[111] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02) :107–116, 1998. 60

[112] Tarik A Rashid, Polla Fattah, and Delan K Awla. Using accuracy measure for improving the training of lstm with metaheuristic algorithms. *Procedia Computer Science*, 140 :324–333, 2018. 61

[113] Noshin Tasnim and Farhana Yasmeen. *An in depth analysis of neural network with application in finance*. PhD thesis, Brac University, 2019. 62

[114] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016. 63, 67

[115] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv :1412.3555*, 2014. 63

[116] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *International conference on machine learning*, pages 2067–2075, 2015. 63

[117] Sydney Mambwe Kasongo and Yanxia Sun. A deep gated recurrent unit based model for wireless intrusion detection system. *ICT Express*, 2020. 63

[118] Mohamed Elmahdy, Rainer Gruhn, and Wolfgang Minker. *Novel techniques for dialectal arabic speech recognition*. Springer Science & Business Media, 2012. 67

[119] Biing-Hwang Juang and Lawrence R Rabiner. Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, 1 :67, 2005. 68

[120] Fawaz Al-Anzi and Dia AbuZeina. Literature survey of arabic speech recognition. In *2018 International Conference on Computing Sciences and Engineering (ICCSE)*, pages 1–6. IEEE, 2018. 68, 69

[121] Wajdan Algihab, Noura Alawwad, Anfal Aldawish, and Sarah AlHumoud. Arabic speech recognition with deep learning : A review. In *International Conference on Human-Computer Interaction*, pages 15–31. Springer, 2019. 68

[122] Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Teodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, et al. Automatic speech recognition and speech variability : A review. *Speech communication*, 49(10-11) :763–786, 2007. 68

[123] Jayashree Padmanabhan and Melvin Jose Johnson Premkumar. Machine learning in automatic speech recognition : A survey. *IETE Technical Review*, 32(4) :240–251, 2015. 68, 75

[124] Khalaf Khatatneh et al. A novel arabic speech recognition method using neural networks and gaussian filtering. *International Journal of Electrical, Electronics & Computer Systems*, 19(1), 2014. 68

[125] Basem HA Ahmed and Ayman S Ghabayen. Arabic automatic speech recognition enhancement. In *2017 Palestinian International Conference on Information and Communication Technology (PICICT)*, pages 98–102. IEEE, 2017. 69

[126] Ahmad Emami and Lidia Mangu. Empirical study of neural network language models for arabic speech recognition. In *2007 IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, pages 147–152. IEEE, 2007. 69, 74

[127] Katrin Kirchhoff, Dimitra Vergyri, Jeff Bilmes, Kevin Duh, and Andreas Stolcke. Morphology-based language modeling for conversational arabic speech recognition. *Computer Speech & Language*, 20(4) :589–608, 2006. 69

[128] Mansour Alghamdi, Moustafa Elshafei, and Husni Al-Muhtaseb. Arabic broadcast news transcription system. *International Journal of Speech Technology*, 10(4) :183–195, 2007. 69

[129] Hussein Hyassat and Raed Abu Zitar. Arabic speech recognition using sphinx engine. *International Journal of Speech Technology*, 9(3-4) :133–150, 2006. 70

[130] Mohamed Elmahdy, Rainer Gruhn, Wolfgang Minker, and Slim Abdennadher. Modern standard arabic based multilingual approach for dialectal arabic speech recognition. In *2009 Eighth International Symposium on Natural Language Processing*, pages 169–174. IEEE, 2009. 70

[131] Sid Ahmed Selouani and Malika Boudraa. Algerian arabic speech database (algasd) : corpus design and automatic speech recognition application. *Arabian Journal for Science and Engineering*, 35(2) :157–166, 2010. 70

[132] Abderrahmane Amrouche and J Michel Rouvaen. Arabic isolated word recognition using general regression neural network. In *2003 46th Midwest*

Symposium on Circuits and Systems, volume 2, pages 689–692. IEEE, 2003. 71

[133] MM El Choubassi, HE El Khoury, CE Jabra Alagha, JA Skaf, and MA Al-Alaoui. Arabic speech recognition using recurrent neural networks. In *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology (IEEE Cat. No. 03EX795)*, pages 543–547. IEEE, 2003. 71, 75

[134] Yousef Ajam Alotaibi. Spoken arabic digits recognizer using recurrent neural networks. In *Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, 2004.*, pages 195–199. IEEE, 2004. 71, 75

[135] Yousef Ajami Alotaibi. A simple time alignment algorithm for spoken arabic digit recognition. *Engineering Sciences*, 20(1), 2009. 71

[136] Naima Zerari, Samir Abdelhamid, Hassen Bouzgou, and Christian Raymond. Bi-directional recurrent end-to-end neural network classifier for spoken arab digit recognition. In *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–6. IEEE, 2018. 71, 83

[137] Uci machine learning repository, university of california, school of information and computer science. https://archive.ics.uci.edu/ml/datasets/Spoken+Arabic+Digit/ (Consulté le : 28/12/2020). 71, 87

[138] Ghulam Muhammad, Tamer A Mesallam, Khalid H Malki, Mohamed Farahat, Mansour Alsulaiman, and Manal Bukhari. Formant analysis in dysphonic patients and automatic arabic digit speech recognition. *Biomedical engineering online*, 10(1) :41, 2011. 72

[139] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. The htk book. *Cambridge university engineering department*, 3(175) :12, 2002. 72

[140] R Walha, F Drira, H El-Abed, and AM Alimi. On developing an automatic speech recognition system for standard arabic language. *International Journal of Electrical and Computer Engineering*, 6(10) :1138–1143, 2012. 72

[141] M Kabache and M Guerti. Application des réseaux de neurones à la reconnaissance des phonèmes spécifiques à lŠarabe standard. *SETIT 2005*, 2005. 72

[142] N Hmad and Tony Allen. Biologically inspired continuous arabic speech recognition. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 245–258. Springer, 2012. 72

[143] Patrick Cardinal, Ahmed Ali, Najim Dehak, Yu Zhang, Tuka Al Hanai, Yifan Zhang, James R Glass, and Stephan Vogel. Recent advances in asr applied to an arabic transcription system for al-jazeera. In *Fifteenth annual conference of the international speech communication association*, 2014. 72

[144] Ahmed Ali, Yifan Zhang, Patrick Cardinal, Najim Dahak, Stephan Vogel, and James Glass. A complete kaldi recipe for building arabic speech recognition systems. In *2014 IEEE spoken language technology workshop (SLT)*, pages 525–529. IEEE, 2014. 73

[145] Natalia Tomashenko, Kévin Vythelingum, Anthony Rousseau, and Yannick Estève. Lium asr systems for the 2016 multi-genre broadcast arabic challenge. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 285–291. IEEE, 2016. 73

[146] Tuka AlHanai, Wei-Ning Hsu, and James Glass. Development of the mit asr system for the 2016 arabic multi-genre broadcast challenge. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 299–304. IEEE, 2016. 73

[147] L Bouchakour and M Debyeche. Improving continuous arabic speech recognition over mobile networks dsr and nsr using mfccs features transformed. 2018. 73

[148] Martin Graciarena, Sachin Kajarekar, Andreas Stolcke, and Elizabeth Shriberg. Noise robust speaker identification for spontaneous arabic speech. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–245. IEEE, 2007. 73

[149] Hesham Tolba. Comparative experiments to evaluate the use of a chmm-based speaker identification engine for arabic spontaneous speech. In *2009 2nd IEEE International Conference on Computer Science and Information Technology*, pages 241–245. IEEE, 2009. 74

[150] Chahira Lhioui, Anis Zouaghi, and Mounir Zrigui. A combined method based on stochastic and linguistic paradigm for the understanding of arabic spontaneous utterances. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 549–558. Springer, 2013. 74

[151] Mohamed Ettaouil, Mohamed Lazaar, and Zakariae En-Naimani. A hybrid ann/hmm models for arabic speech recognition using optimal code-

book. In *2013 8th International Conference on Intelligent Systems : Theories and Applications (SITA)*, pages 1–5. IEEE, 2013. 74

[152] Elvira Sukma Wahyuni. Arabic speech recognition using mfcc feature extraction and ann classification. In *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, pages 22–25. IEEE, 2017. 75

[153] Zhen-Hua Ling, Shi-Yin Kang, Heiga Zen, Andrew Senior, Mike Schuster, Xiao-Jun Qian, Helen M Meng, and Li Deng. Deep learning for acoustic modeling in parametric speech generation : A systematic review of existing techniques and future trends. *IEEE Signal Processing Magazine*, 32(3) :35–52, 2015. 75

[154] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks : A systematic review. *IEEE Access*, 7 :19143–19165, 2019. 75

[155] Abdelaziz Abdelhamid, Hamzah Alsayadi, Islam Hegazy, and Zaki Fayed. End-to-end arabic speech recognition : A review. In *ESOLECŠ19 : The Nineteenth Conference on Language Engineering*, 09 2020. 75

[156] Ali AbdAlmisreb, Ahmad Farid Abidin, and Nooritawati Md Tahir. Maxout based deep neural networks for arabic phonemes recognition. In *2015 IEEE 11th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 192–197. IEEE, 2015. 75

[157] Mohamed Elaraby and Muhammad Abdul-Mageed. Deep models for arabic dialect identification on benchmarked data. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 263–274, 2018. 76

[158] Omar Zaidan and Chris Callison-Burch. The arabic online commentary dataset : an annotated dataset of informal arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, pages 37–41, 2011. 76

[159] Google cloud speech-to-text. https://cloud.google.com/speech-to-text/ (Consulté le : 30/10/2020). 76

[160] Microsoft speech-to-text. https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/ (Consulté le : 30/10/2020). 77

[161] Ibm watson speech-to-text. https://www.ibm.com/watson/services/speech-to-text/. (Consulté le : 30/10/2020). 77

[162] Kaldi. http://kaldi-asr.org/models.html/ (Consulté le : 30/10/2020). 77

[163] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No. : CFP11SRW-USB. 77

[164] Vimal Manohar, Daniel Povey, and Sanjeev Khudanpur. Jhu kaldi system for arabic mgb-3 asr challenge using diarization, audio-transcript alignment and transfer learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 346–352. IEEE, 2017. 77

[165] Tensorflow. https://www.tensorflow.org/?hl=fr (Consulté le : 30/10/2020). 77

[166] Alexander Waibel and Kai-Fu Lee. *Readings in speech recognition*. Elsevier, 1990. 82

[167] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1) :1929–1958, 2014. 84

[168] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4 :40–79, 2010. 90, 92

[169] Emmanuel Jakobowicz. *Python pour le data scientist : Des bases du langage au machine learning*. Dunod, 2018. 94

[170] François Chollet et al. Keras : The python deep learning library. *ascl*, pages ascl–1806, 2018. 94

[171] Hui Jiang. Confidence measures for speech recognition : A survey. *Speech communication*, 45(4) :455–470, 2005. 94

[172] I Dan Melamed, Ryan Green, and Joseph Turian. Precision and recall of machine translation. In *Companion Volume of the Proceedings of HLT-NAACL 2003-Short Papers*, pages 61–63, 2003. 94

[173] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn : Machine learning in python. *the Journal of machine Learning research*, 12 :2825–2830, 2011. 94, 96

[174] Md Sahidullah and Goutam Saha. Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition. *Speech communication*, 54(4) :543–565, 2012. 100