UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTÉ DES SCIENCES ET TECHNIQUES FÈS
DÉPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ETUDES

MASTER SCIENCES ET TECHNIQUES
SYSTÈMES INTELLIGENTS & RÉSEAUX

LA RECONNAISSANCE AUTOMATIQUE DU LOCUTEUR PAR LES RÉSEAUX DE NEURONES PROFONDS

LIEU DE STAGE : LABORATOIRE DES SYSTÈMES INTELLIGENTS ET APPLICATIONS À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE FÈS

RÉALISÉ PAR : SOUFIANE HOURRI

SOUTENU LE 13/06/2016

ENCADRÉ PAR :

MR JAMAL KHARROUBI

DEVANT LE JURY COMPOSÉ DE :

PR. JAMAL KHARROUBI
PR. KHALID ABBAD
PR. FATIHA MRABTI
PR. LOUBNA LAMRINI

ANNÉE UNIVERSITAIRE 2015-2016

Remerciements

Avant toute chose, je tiens à remercier Monsieur **Jamal Kharroubi**, encadrant de mon stage, pour l'attention particulière qu'il m'a accordée au long des jours de mon stage, ainsi que monsieur **Ayoub Bouziane**, doctorant au sein du laboratoire des systèmes intelligents et applications, pour son précieux aide.

Réaliser ce travail n'aurait pas été possible sans **ma famille** qui a toujours encouragé et soutenu toutes mes idées et mes projets, aussi loin qu'ils peuvent être parfois.

Egalement, je souhaite que toute personne, qui a contribué de près ou de loin au bon déroulement de mon stage de fin d'études, puisse trouver ici l'expression de ma reconnaissance et ma gratitude.

Merci

Résumé

La reconnaissance automatique du locuteur est le processus de la reconnaissance de ceux qui parlent par les caractéristiques de leurs voix. Dans ce rapport nous étudions la reconnaissance automatique du locuteur en mode indépendant du texte, et plus précisément, la vérification du locuteur en utilisant un réseau de neurones profond comme extracteur des vecteurs Bottleneck, et un réseau de neurones artificiel, séparateurs à vaste marge, K plus proches voisins et les distances comme classifieurs. Nous avons trouvé que les réseaux de neurones donnent le taux d'erreur le plus bas pour les locuteurs masculins et féminins.

Mots clés : vérification du locuteur, DNN, PMC, KPPV, SVM, Bottleneck.

Abstract

Automatic speaker recognition is the process of automatically recognizing who is speaking from characteristics of voices by speech input. In this report we study the text-independent speaker recognition and specially speaker verification using deep neural networks as feature extractor for Bottleneck features and neural network, Support Vector Machine, K nearest neighbors and distances as speaker classifiers, we found that neural networks shows the lowest equal error rate for both male and female speakers.

Keywords: speaker verification, DNN, MLP, KNN, SVM, Bottleneck features.

Table des matières

Liste des figures

Liste des tables

Liste des abréviations

RAL : Reconnaissance automatique du locuteur.

IAL : Identification automatique du locuteur.

VAL : Vérification automatique du locuteur.

UBM : Universal background model.

MLP : Multilayer perceptron (Anglais).

PMC : Perceptron multicouche.

SVM : Séparateurs à vaste marge.

SVM : Support vector machine (Anglais).

DNN : Deep Neural Network (Anglais).

BN : Bottleneck.

KNN : K nearest neighbor.

KPPV : K plus proche voisins.

CNN : Convolutional neural network (Anglais).

MBR : Machines de Boltzmann restreintes.

DBN : Deep belief network (Anglais).

EER : Equal Error Rate (Anglais).

Introduction générale

La Faculté des Sciences et Techniques de Fès intègre dans le cursus de la formation Master Sciences et Techniques ses étudiants un stage de fin d'études au choix, soit un stage de fin d'études techniques au sein d'une entreprise ou bien un stage de fin d'études de recherche au sein d'un laboratoire, ce qui permet aux étudiants de mettre en pratique leurs connaissances théoriques et pratiques sur terrain, et facilite leur intégration dans le monde professionnel et de recherche après l'obtention de leurs diplômes. Dans ce cadre, on a choisi d'effectué notre stage de fin d'études au sein du laboratoire des Systèmes Intelligents et Applications et spécialement dans les systèmes de communication et traitement des connaissances sous l'encadrement du professeur monsieur Jamal Kharroubi.

Le sujet de notre stage de fin d'études est la reconnaissance automatique du locuteur par les réseaux de neurones profonds. La RAL est le processus de détecter automatiquement l'identité de celui qui parle en se basant sur les informations incluses dans son signal vocal, elle fait l'objet de travaux de recherches par nombreuses équipes de recherche dans le monde, elle s'est limitée longtemps à l'identification et la vérification, cette dernière se fait en calculant un modèle stochastique sur la base de l'expression vocale du locuteur à reconnaitre. Une fois calculé, ce modèle est comparé à des modèles pré entraînés sur la base de différents enregistrements prononcés par les locuteurs. Or l'identification consiste à reconnaitre un locuteur particulier parmi un ensemble fini de locuteurs possibles.

Les applications potentielles des systèmes de reconnaissance de locuteur incluent le contrôle d'accès à distance de bases de données, les services d'information et de réservation à distance, les services bancaires à distance, etc. la tendance actuelle montre une évolution vers l'exécution de diverses transactions en utilisant les téléphones mobiles.

Les réseaux de neurones sont une technique d'apprentissage inspirée des réseaux de neurones biologique du cerveau humain, ils sont utilisés pour la classification, la régression, et maintenant pour l'extraction des caractéristiques en utilisant plusieurs couches cachées et l'extraction des caractéristiques sous forme de vecteurs Bottleneck. L'utilisation des réseaux de neurones a commencée par la discrimination entre les locuteurs par des réseaux de neurones artificiels avec une seule couche cachée mais avec un nombre très limité de locuteurs, et maintenant elle a réussi d'être parmi les extracteurs de caractéristiques en reconnaissance automatique de la parole, et actuellement en reconnaissance automatique du locuteur.

Le présent rapport se décline en trois parties principales :

- La première partie permet de donner une vision générale sur la reconnaissance automatique du locuteur et les différentes tâches traitées par les systèmes RAL.
- La deuxième partie est consacré à une description détaillée des réseaux de neurones artificiels et des réseaux de neurones profonds.
- La troisième partie est dédiée au protocole expérimental ainsi que les différentes méthodes utilisées, et présentation des résultats.

Chapitre 1

La reconnaissance automatique du locuteur

I. Introduction

Ce chapitre introduit l'état de l'art de la reconnaissance automatique du locuteur, ainsi que les différentes tâches traitées par les chercheurs de la reconnaissance automatique du locuteur. Nous présentons aussi la structure générale d'un système de reconnaissance automatique du locuteur.

II. La reconnaissance automatique du locuteur

1. Généralités

La reconnaissance automatique du locuteur – RAL – est un sous-problème de la caractérisation automatique du locuteur. Cette dernière est un domaine vaste ou la machine a pour tâche d'extraire du signal de la parole les informations suffisantes d'un individu qui renseignent sur ces spécificités : identité, caractéristiques physique, émotivité, état pathologique, particularités régionales, etc. Elle s'applique à différents thèmes de recherche traitant des informations extralinguistiques véhiculées par la voix tels que la classification d'individus, ou l'étude psychique ou physiologique d'une personne. Or, la reconnaissance automatique du locuteur repose sur la reconnaissance de l'identité d'une personne à l'aide de sa voix en profitant de la variabilité de la parole entre locuteurs.

La reconnaissance automatique du locuteur peut être répartie en deux catégories, la première est la reconnaissance automatique du locuteur en mode dépend du texte ou le système a déjà l'information à priori sur le message que le locuteur est censé prononcer [1]. Cette première catégorie, peut se divisée en trois sous-catégories. Dans la première, le locuteur doit prononcer le même message à chaque fois dans l'étape de la reconnaissance. Pour la deuxième, la machine demande au locuteur à chaque fois de prononcer un message différent. Et la dernière, la machine fait prononcer au locuteur une séquence de phonèmes. La deuxième catégorie est la reconnaissance automatique du locuteur indépendante du texte. Dans cette catégorie, la machine effectue l'opération de la reconnaissance du locuteur seulement en se basant sur le signal de la parole indépendamment de la message prononcé [2].

2. Tâches de la reconnaissance automatique du locuteur

La reconnaissance automatique du locuteur est un domaine où plusieurs tâchent s'en dérivent. Les tâches principales de la reconnaissance automatique du locuteur sont : l'identification automatique du locuteur, la vérification automatique du locuteur, l'indexation par locuteur et le suivi de locuteur.

a. Identification automatique du locuteur

L'identification automatique du locuteur – IAL – est le processus qui permet de chercher la voix d'un locuteur parmi un ensemble de voix de locuteurs enregistrés dans la base de données.

La Figure 1 illustre, d'un point de vue schématique, l'identification automatique du locuteur. Une séquence de parole est donnée en entrée du système d'IAL. Pour chaque locuteur connu du système, la séquence de parole est comparée à une référence caractéristique du locuteur. L'identité du locuteur dont la référence est la plus proche de la séquence de parole est donnée en sortie du système d'IAL.

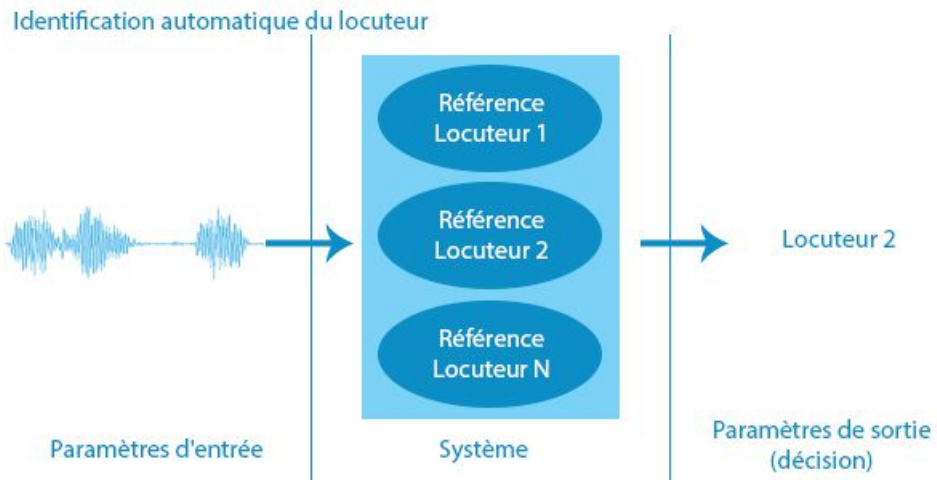

Figure 1: La tâche d'IAL. Principe de base de la tâche d'Identification Automatique du Locuteur

En IAL, deux modes sont proposés : l'identification en ensemble fermé pour lequel on suppose que la séquence de parole est effectivement prononcée par un locuteur connu par le système, et l'identification en ensemble ouvert pour lequel le locuteur peut ne pas être connu.

b. Vérification automatique du locuteur

La vérification du locuteur – VAL – ou bien l'authentification est le processus décisionnel permettant de déterminer, au moyen d'un signal vocal, la véracité de l'identité revendiquée par un individu (Figure 2). L'identité ainsi que le signal vocal constituent les deux entrées du système de VAL. L'identité, nécessairement connue du système, désigne automatiquement la référence caractéristique d'un locuteur. Une mesure de similarité est calculée entre cette référence et le signal vocal, puis comparée à un seuil de décision. Dans le cas où la mesure de similarité est supérieure au seuil, l'individu est accepté. Dans le cas contraire, l'individu est considéré comme un imposteur et rejeté.

c. Indexation par locuteur

La tâche d'indexation automatique par locuteur (Figure 3) consiste à cibler les interventions de différents locuteurs dans un flux audio. C'est-à-dire, indiquer à quel moment un individu prend la parole et qui est cet individu. Le signal de la parole est pris dans les paramètres d'entrée, et aucune information n'est donnée au système sur le nombre de locuteurs présents dans le document ou leur identité. Le principe de fonctionnement de la tâche d'indexation par locuteur repose

généralement sur une phase de segmentation aveugle en locuteurs suivi d'une phase de regroupement. En sortie, chaque locuteur est identifié par un système IAL avec ses instants d'interventions.

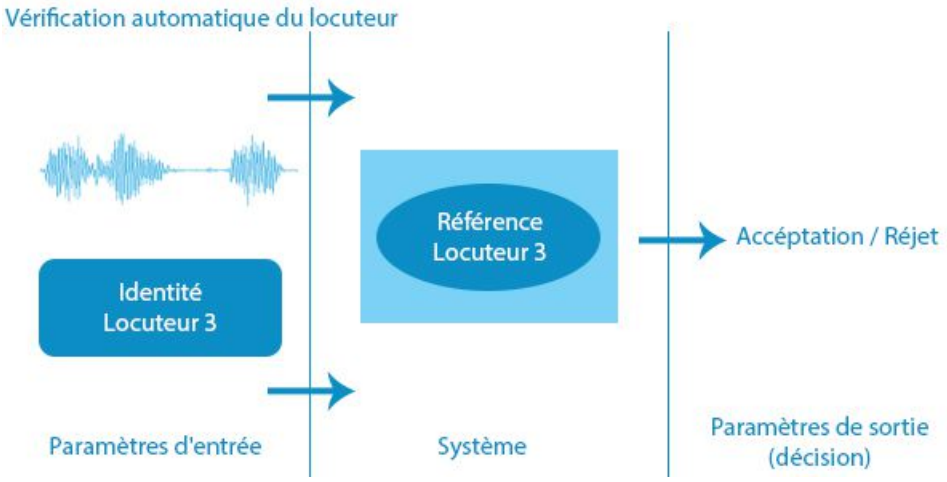

Figure 2: La tâche de VAL. Principe de base de la tâche de Vérification Automatique du Locuteur

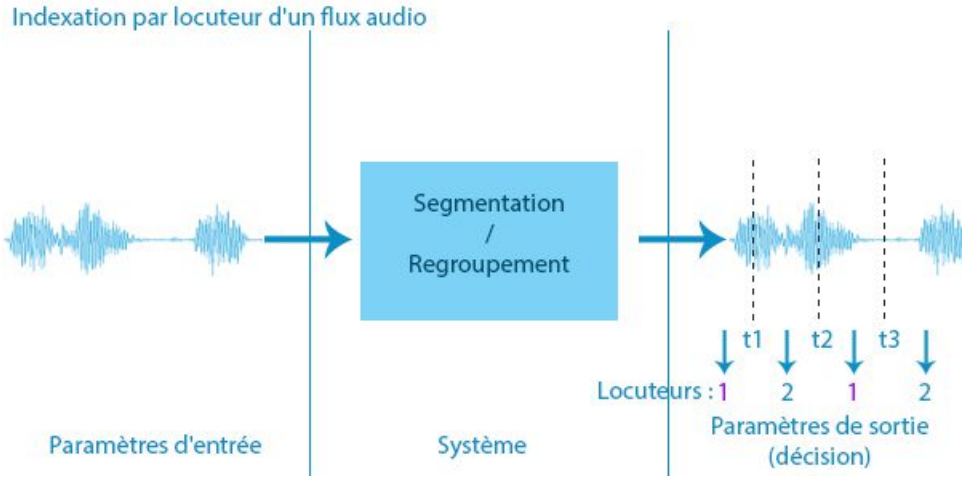

Figure 3 : La tâche d'indexation par locuteur d'un flux audio. Principe de base.

d. Suivi de locuteurs

La tâche de suivi de suivi de locuteurs (Figure 6) est une version simplifiée de l'indexation par locuteur d'un flux audio. Le principe reste le même : déterminer les interventions d'un ou plusieurs locuteurs, appelés locuteurs cibles, dans un flux audio. La simplification réside dans le fait que le système de suivi de locuteurs connait à priori les locuteurs présents dans le document à indexer ou, du moins, ceux dont il doit détecter les interventions. Il possède une référence caractéristique pour chacun des locuteurs. Malgré cette simplification, le suivi de locuteurs reste une tâche très complexe. Trois grandes approches sont recensées dans la littérature :

1- Une segmentation aveugle en locuteurs (Figure 4), identique à celle employée pour l'indexation par locuteur d'un flux audio, est appliquée sur le signal de test. Les segments sont soumis à un système de VAL classique afin de déterminer les segments appartenant effectivement au locuteur cible [3].

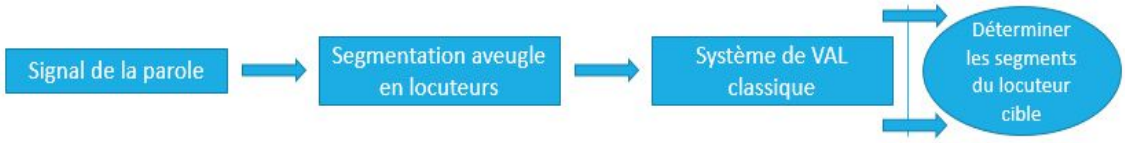

Figure 4 : La première approche de suivi de locuteurs qui repose sur la segmentation aveugle en locuteurs

2- Le signal de test est découpé en une suite de blocs de trames (Figure 5) de taille fixe. Sur ces chaque bloc on lance un système de VAL. Un processus de décision, à base de seuils, permet en phase finale d'accepter ou de rejeter les blocs appartenant au locuteur cible [4], [5].

Figure 5 : La deuxième approche de suivi de locuteurs qui consiste à découper le signal en une suite de blocs de trames de taille fixe

3- La troisième approche est similaire à la précédente excepté pour le processus de décision. Dans ce cas, la décision repose sur un HMM ergodique composé d'états correspondant au locuteur cible, à un modèle générique de parole et à un modèle générique de non parole [6].

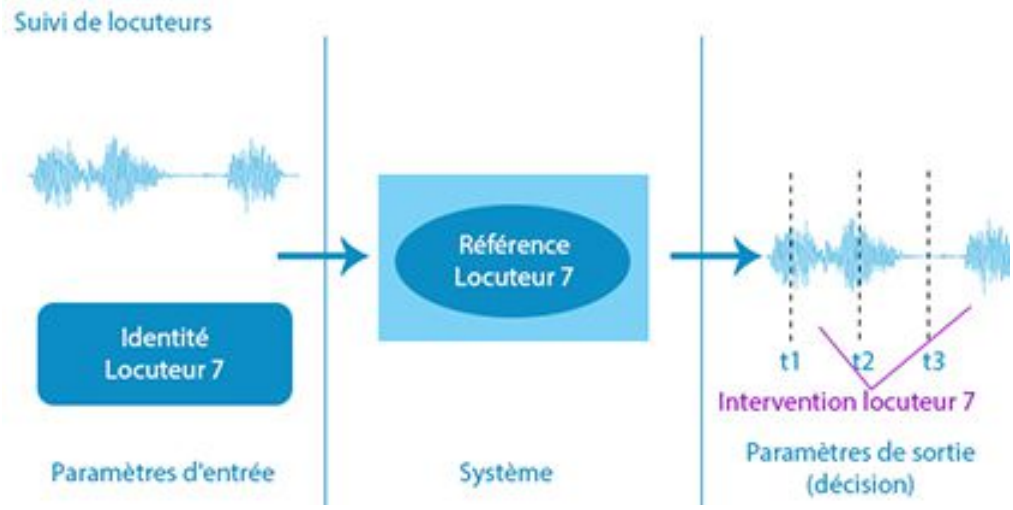

Figure 6 : La tâche de suivi de locuteurs. Principe de base

3. Mise en place d'un système de RAL

La mise en place d'un système de RAL pour une application donnée se décompose en deux phase distinctes. La première phase est la phase essentielle qui consiste à construire des références ou modèles de chaque client (locuteur connu) du système. Par conséquence, il faut collecter auprès des clients de l'application des signaux de parole dits d'apprentissage, lors de sessions d'entrainement. La deuxième phase est la phase de test ou de reconnaissance qui consiste, pour chaque client, à se présenter devant le système RAL.

4. Problèmes rencontrés en RAL

Le signal de parole est un signal très complexe puisqu'il est constitué d'information linguistiques, d'informations caractéristiques du locuteur, d'informations relatives au matériel utilisé pour la transmission ou l'enregistrement du signal, etc. En outre, le signal de parole est très redondant. Cette caractéristique est d'ailleurs reconnue pour faciliter la communication entre deux personnes dans un environnement très bruyant. Par ces différents aspects, le signal de parole présente une très grande variabilité.

La capacité des systèmes de RAL à différencier plusieurs individus repose essentiellement sur la variabilité inter-locuteur i.e. la disposition du signal de parole à varier entre différents individus. Cette variabilité peut être aussi sur le même individu mais elle est considérée en tant que variabilité intra-locuteur, elle est induite par l'évolution naturelle ou volontaire de la voix d'une personne. Néanmoins, le signal de parole renferme d'autre type de variabilité qui rendent problématique la tâche de reconnaissance, telle la variabilité due au matériel. Par ailleurs, les systèmes de RAL doivent faire face à d'autres difficultés liées d'avantage au domaine applicatif, comme l'utilisation des systèmes dans des conditions difficiles, les tentatives d'imposture, etc.

III. Structure des systèmes de RAL et techniques associées

En générale, un système de RAL se résume à l'enchainement de trois processus principaux qui sont : la paramétrisation, la reconnaissance et la décision (Figure 7). Le processus de paramétrisation est basé sur des techniques communes à d'autres domaines comme la reconnaissance automatique de la parole. Par contre, dans les deux autres processus, reconnaissance et décision, les principes mis en œuvre sont étroitement liés à la tâche visée (IAL, VAL ou Indexation).

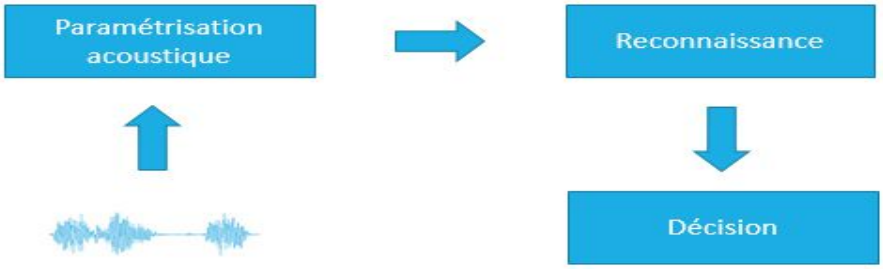

Figure 7 : Structure générale d'un système de reconnaissance automatique de locuteur

1. Paramétrisation acoustique

Le processus de paramétrisation acoustique consiste à extraire du signal de paroles les informations pertinentes en vue de la reconnaissance. Car le signal de parole ne peut être exploité directement due à sa complexité. Donc, une représentation simplifiée est utilisée en utilisant généralement des vecteurs de paramètres acoustiques, calculés périodiquement sur le signal de parole.

La première étape de la paramétrisation acoustique consiste à décomposer le signal de parole, à cadence régulière, en trames de signal. Un traitement particulier est ensuite appliqué à ces trames afin de produire les vecteurs de paramètres acoustiques.

La littérature propose un grand nombre de traitements selon la nature des informations à extraire du signal de parole. Mais on considère juste trois grandes classes de paramètres (Figure 8) : les paramètres de l'analyse spectrale, les paramètres prosodiques et les paramètres dynamiques [7].

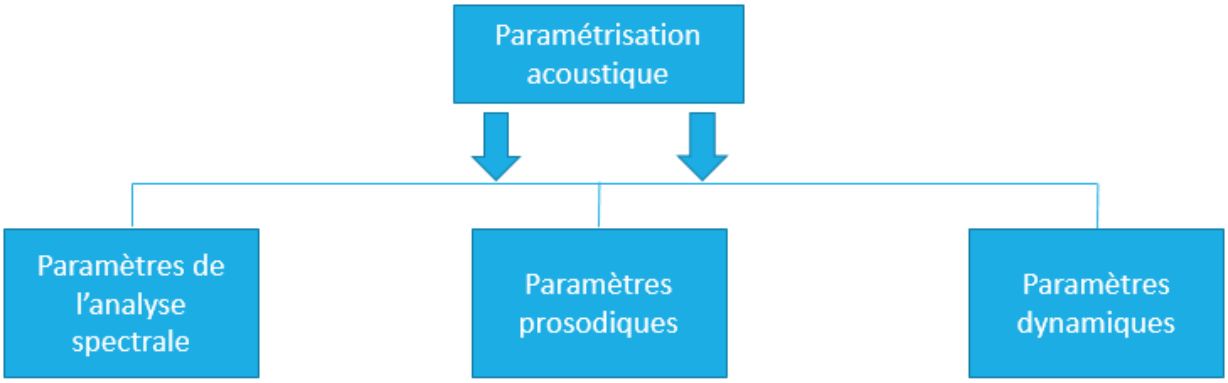

Figure 8 : les trois types les plus utilisés en RAL

a. Paramètres de l'analyse spectrale

L'analyse spectrale est l'analyse la plus utilisée en RAL. Les paramètres qui en découlent sont généralement représentatifs des caractéristiques physiques de l'appareil phonatoire de chaque individu.

Les paramètres les plus pertinents en RAL sont :

- Coefficients issus d'une analyse par prédiction linéaire : LPCC *(Linear Predictive Cepstral Coefficients)* ou LPC *(Linear Predictive Coefficients)* ;
- Coefficients spectraux issus d'une analyse en banc de filtres : LFSC *(Linear Frequency Cepstral Coefficients)* ou MFSC *(Mel Frequenc Spectral Coefficients)* ;
- Coefficients spectraux issus d'une analyse en banc de filtres : LFCC *(Linear Frequency Cepstral Coefficients)* ou MFCC *(Mel Frequency Cepstral Coefficients)*.

b. Paramètres prosodiques

Les paramètres prosodiques représentent le style d'élocution d'un locuteur : vitesse d'élocution, durée et fréquence des pauses, … ainsi que les caractéristiques de la source glottale (fréquence fondamentale, énergie, taux de voisement, …).

Néanmoins, ces paramètres, notamment la fréquence fondamentale et ses variations [8] sont généralement associés aux paramètres de l'analyse spectrale pour améliorer les performances des systèmes de RAL, car ils ne sont pas suffisamment discriminants pour être utilisés seuls dans un système de RAL.

c. Paramètres dynamiques

L'information dynamique véhiculée par le signal de parole est une source potentielle d'informations pour la caractérisation du locuteur [7]. Les paramètres dynamiques les plus répandus demeurent les coefficients dérivés des vecteurs de paramètres instantanés, appelés coefficients Delta (première dérivée) et Delta-Delta (second dérivée). D'autres paramétrisations sont proposées dans la littératures pour exploiter es informations dynamiques du signal telles que l'utilisation des Composants Principales Temps-Fréquence, la concaténation de trames successives de signal, etc.

2. Reconnaissance

Le processus de reconnaissance s'appuie généralement, pour les tâches d'IAL, de VAL et de suivi de locuteurs, sur une modélisation des caractéristiques de chaque locuteur connu du système. Cette modélisation est réalisée à partir des données d'apprentissage collectées au cours des sessions d'enrôlement. Une mesure de similarité est ensuite calculée entre un modèle client et un signal de parole, puis transmise au processus de décision.

La littérature propose quatre grandes approches pour la construction des modèles clients (Figure 9) : les approches vectorielles, statistique, prédictive et connexionniste. Nous présentons brièvement les fondements de chacune de ces approches.

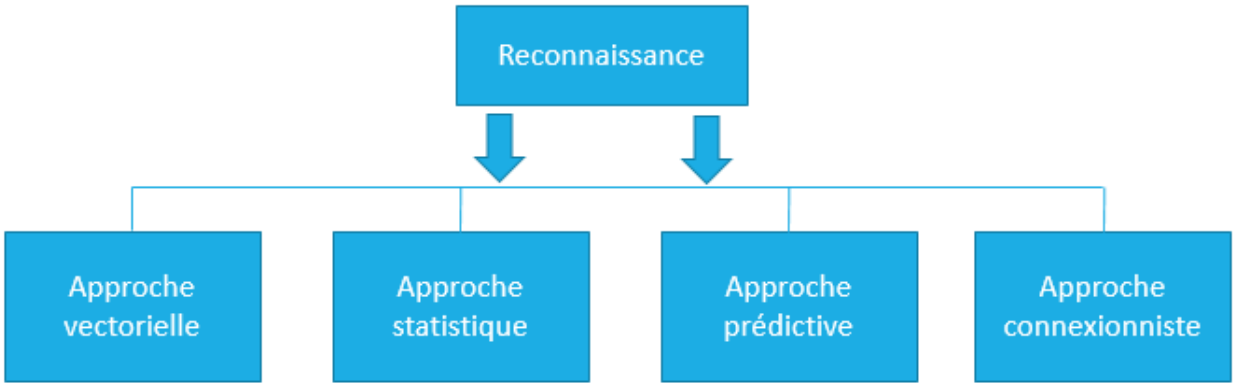

Figure 9 : Les différentes approches existent en reconnaissance automatique du locuteur

a. L'approche vectorielle

Dans l'approche vectorielle, un modèle de locuteur est un ensemble de vecteurs de paramètres représentatifs de l'espace acoustique construit lors de la phase de paramétrisation des signaux d'apprentissage. Lors de la reconnaissance, une distance entre cet ensemble de vecteurs et les vecteurs de paramètres issus des signaux de test est calculée. Cette approche compte deux grandes techniques : la programmation dynamique et la quantification vectorielle. Pour la programmation dynamique (*Dynamic Time Warping : DTW*), il consiste à aligner temporellement une séquence de vecteurs de paramètres de test avec une séquence de vecteurs d'apprentissage. Dans ce cas, le modèle de locuteur est tout simplement l'ensemble des vecteurs de paramètres obtenus après paramétrisation des signaux d'apprentissage. Une distance est calculée entre vecteurs d'apprentissage et de test et moyennée sur l'ensemble de la séquence. Pour la quantification vectorielle (*Vector Quantisation : VQ*), il repose sur un partitionnement de l'espace acoustique en sous-espaces. Chaque sous-espace est associé à leur vecteur centroïde. Dans ces conditions, un modèle de locuteur est composé d'un ensemble de vecteurs centroïdes, appelé dictionnaire de quantification (*codebook*). Lors de la phase de reconnaissance, une distance est calculée entre un vecteur de test et chaque vecteur centroïde du dictionnaire. La distance minimale est assignée au vecteur de test. La distance d'une séquence de vecteurs de test est obtenue par moyenne des distances minimales attribuées à chacun des vecteurs de test.

b. L'approche statistique

L'approche statistique consiste à représenter une séquence de vecteurs acoustiques issus de la paramétrisation par des statistiques à long terme. Lors de la reconnaissance, le spectre moyen estimé sur les vecteurs de test est comparé, à l'aide d'une distance spectrale, au spectre moyen issu de l'apprentissage.

c. L'approche connexionniste

L'approche connexionniste repose sur la discrimination entre locuteurs, elle consiste à fournir à un réseau de neurones un ensemble de signaux de parole issus d'une population de locuteurs clients afin que ce dernier apprenne comment discriminer un locuteur des autres. Un modèle client

se présente sous la forme d'un ou plusieurs réseaux de neurones pour lequel la séquence de vecteurs d'apprentissage du client concerné ainsi que celles des autres clients du système sont fournies en entrée. Lors de la reconnaissance, la vraisemblance pour qu'une séquence de vecteurs de test soit produite par un réseau de neurones est calculée.

d. L'approche prédictive

L'approche prédictive repose sur le principe qu'une trame de signal peut être prédite par la seule observation des trames précédentes. Par conséquent, cette approche est considérée dans la littérature comme une approche dynamique i.e. une approche tenant compte des informations dynamiques véhiculées par le signal de parole. Elle s'appuie principalement sur l'estimation d'une fonction de prédiction, propre à chaque locuteur et apprise sur les signaux d'apprentissage. Lors de la reconnaissance, une erreur de prédiction peut être calculée entre une trame prédite et la trame réellement observée dans la séquence de test. L'erreur de prédiction moyenne constitue alors la mesure de similarité entre le signal de test et le modèle de locuteur.

IV. Conclusion

Dans ce chapitre nous avons présenté les différents types de systèmes de RAL ainsi que les différentes approches utilisées dans l'étape de la reconnaissance dans un système RAL.

Chapitre 2

Les réseaux de neurones artificiels, les réseaux de neurones profonds

I. Introduction

Les réseaux de neurones artificiels sont une inspiration des réseaux de neurones biologiques, dans ce chapitre, nous introduisons les réseaux de neurones biologiques, nous passons par le premier réseau de neurones appelé perceptron, et comment le perceptron multicouche a pallié aux problèmes du perceptron, nous expliquons aussi la notion de "profond" pour les réseaux de neurones.

II. Les réseaux de neurones artificiels

1. Origine biologique

Comment l'homme fait-il raisonner, parler, calculer, apprendre… ? Comment s'y prendre pour créer une intelligence artificielle ? Deux approches ont été essentiellement explorées :

- Approche 1 : Procéder d'abord à l'analyse logique des tâches relevant de la cognition humaine et tenter de les reconstituer par programme. C'est cette approche qui a été privilégiée par l'intelligence artificielle symbolique et la psychologie cognitive classiques. Cette démarche est étiquetée sous le nom de cognitivisme.
- Approche 2 : Puisque la pensée est produite par le cerveau ou en est une propriété, commencer par étudier comment celui-ci fonctionne. C'est cette approche qui a conduit l'étude des réseaux de neurones. On désigne par connexionnisme la démarche consistant vouloir rendre compte de la cognition humaine par des réseaux de neurones.

La seconde approche a donc mené à la définition et à l'étude de réseaux de neurones qui sont des réseaux complexes d'unités de calcul élémentaires interconnectées. Il existe deux courants de recherche sur les réseaux de neurones : un premier motivé par l'étude et la modélisation des phénomènes naturels d'apprentissage pour lequel la pertinence biologique est importante ; un second motivé par l'obtention d'algorithmes efficaces ne se préoccupant pas de la pertinence biologique. Nous nous plaçons du point de vue du second groupe. En effet, bien que les réseaux de neurones aient été définis à partir de considérations biologiques, pour la plupart d'entre eux, et en particulier ceux étudiés dans ce chapitre, de nombreuses caractéristiques biologiques ne sont pas prises en compte.

Les neurones reçoivent les signaux (impulsions électriques) par des extensions très ramifiées de leur corps cellulaire (les dendrites) et envoient l'information par de longs prolongements (les axones). Les impulsions électriques sont régénérées pendant le parcours le long de l'axone. La durée de chaque impulsion est de l'ordre d'une milliseconde et son amplitude d'environ 100 mV.

Les contacts entre deux neurones, de l'axone à une dendrite, se font par l'intermédiaire des synapses (Figure 10). Lorsqu'une impulsion électrique atteint la terminaison d'un axone, des neuromédiateurs sont libérés et se lient à des récepteurs post-synaptiques présents sur les dendrites.

Chaque neurone intègre en permanence jusqu'à un millier de signaux synaptiques. Ces signaux n'opèrent pas de manière linéaire : il y a un effet de seuil.

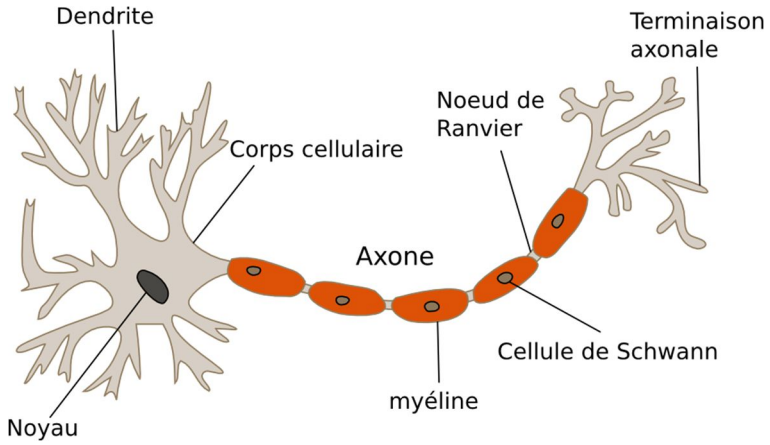

Figure 10 : Modèle du neurone biologique

2. Le perceptron

a. Fonctionnement

Le perceptron a été inventé en 1957 par *Frank Rosenblatt* [9] au laboratoire d'aéronautique de l'université Cornell. C'est un modèle inspiré des théories cognitives de *Friedrich Hayeh* et de *Donald Hebb*. Le perceptron peut être vu comme le type de réseau de neurones le plus simple. C'est un classifieur linéaire monocouche et n'a qu'une sortie à laquelle toutes les entrées sont connectées. Les entrées et la sortie sont booléennes.

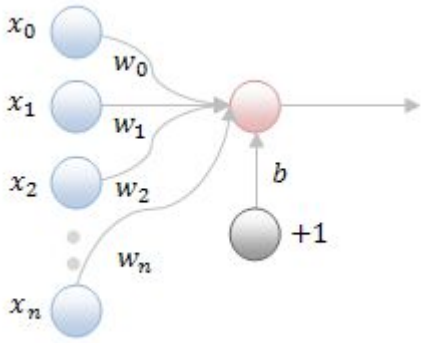

Figure 11 : Modélisation du perceptron

Le résultat de la sortie est calculé selon le relation suivante :

$$Sortie = \begin{cases} 0 \; si \; \omega.x \; + b \leq 0 \\ 1 \; si \; \omega.x \; + b > \; 0 \end{cases}$$

Avec $\omega = \{\omega_0, \omega_1, \dots, \omega_n\}$, $x = \{x_0, x_1, \dots, x_n\}$ et $b = +1$ $(Biais\ utilisé)$

b. Apprentissage

Donald Hebb a proposé une règle d'apprentissage des réseaux de neurones artificiels. Cette règle suggère que lorsque deux neurones sont excités conjointement, il se crée ou renforce un lien les unissant. Dans le cas d'un neurone artificiel seul utilisant la fonction signe comme fonction d'activation cela signifie que :

$$\omega'_i = \omega_i + \alpha(y.x_i)$$

Où ω'_i représente le poids i corrigé et α représente le pas d'apprentissage.

Mais le problème avec cette règle vient de s'inapplicabilité dans certains cas bien que la solution existe.

Frank Rosenblatt n'était pas loin de cette règle, la différence entre eux s'illustre dans le calcul de l'erreur observée en sortie :

$$\omega'_i = \omega_i + \alpha(y_t - y)x_i$$

Où y_t représente la sortie attendue, ω'_i le poids i corrigé et α le pas d'apprentissage.

L'apprentissage dans les réseaux de neurones est un apprentissage supervisé en se basant sur une base de données (Entrée, sortie désirée). Il consiste à modifier les poids des connexions entre les neurones selon une règles de modification. Généralement, il existe deux algorithmes pour faire apprendre à un réseau de neurones monocouche.

- Apprentissage par descente de gradient.
- Apprentissage par l'algorithme de Windrow-Hoff

Table 1 : Représentation des deux algorithmes d'apprentissage des réseaux de neurones monocouches : Descente de gradient et Windrow-Hoff

Apprentissage par descente de gradient	Apprentissage par Windrow-Hoff
Entrée : n poids reliant les n informations au neurone ayant des valeurs quelconques N exemples (X_k, yk) où X_k est un vecteur à n composantes x_i, chacune représentant une information de cet exemple **Sortie** : les n poids modifiés POUR 1 <= i <= n dw_i = 0 FIN POUR POUR TOUT exemple e = (Xk, yk) Calculer la sortie sk du neurone POUR 1 <= i <= n di = dw_i + alpha*(yk - sk)*x_i	**Entrée** : n poids reliant les n informations au neurone ayant des valeurs quelconques N exemples (X_k, yk) où X_k est un vecteur à n composantes x_i, chacune représentant une information de cet exemple Le taux d'apprentissage alpha **Sortie** : les n poids modifiés POUR TOUT exemple = (Xk,yk) Calculer la sortie sk du neurone POUR 1 <= i <= n w_i = w_i + alpha*(yk - sk)*x_i FIN POUR FIN POUR

```
        FIN POUR
FIN POUR

POUR 1 <= i <= n
        w_i = w_i + dw_i
FIN POUR
```

Ces deux algorithmes (Table 1) consistent à comparer le résultat qui était attendu par les exemples puis à minimiser l'erreur commise sur les exemples. Mais l'algorithme le plus utilisé c'est l'algorithme de Windrow-Hoff, ou encore ''la règle Delta'', qui n'est en fait qu'une variante de l'algorithme de la descente de gradient. En effet la méthode élaborée par *Windrow* et *Hoff* consiste à modifier les poids après chaque exemple, et non pas après que tous les exemples aient défilé. Ceci va donc minimiser l'erreur de manière précise, et ce sur chaque exemple. Instinctivement, on constate bien que le réseau de neurones va s'améliorer nettement mieux et va tendre bien plus rapidement à classifier parfaitement (ou presque) chacun des exemples, bien que des méthodes plus efficaces encore existent.

c. Limitation du perceptron

Un perceptron linéaire à seuil à n entrées divise l'espace des entrées R^n en deux sous-espaces délimités par un hyperplan (Figure 12). Réciproquement, tout ensemble linéairement séparable peut être discriminé par un perceptron. Mais la plupart des problèmes de classification ne sont pas linéairement séparable, ce qui rend la mission de classification impossible au perceptron pour les problèmes non linéairement séparables.

Le meilleur exemple à donner pour les problèmes non linéairement séparables c'est le OU Exclusif (XOR) qui ne peut pas être calculé par un perceptron linéaire à seuil (Figure 12).

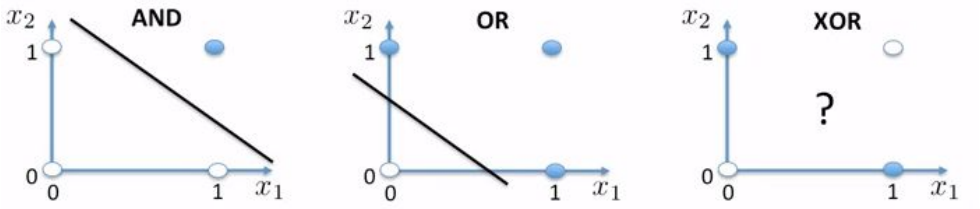

Figure 12 : Représentation des problèmes linéairement séparable

La seule solution revient à décortiquer un problème non linéairement séparable aux problèmes linéairement séparables. Prenons l'exemple du XOR, on pourrait remplacer x_1 par NAND(x_1, x_2) i.e. NON ET, et x_2 par OR(x_1, x_2) (Figure 13).

Donc, nous pourrons créer un réseau de neurones avec deux perceptrons dont chaque perceptron représente un problème linéairement séparable (Figure 14) avec : $y_1 = OR(x_1, x_2)$, $y_2 = NAND(x_1, x_2)$ et $y = AND(y_1, y_2)$

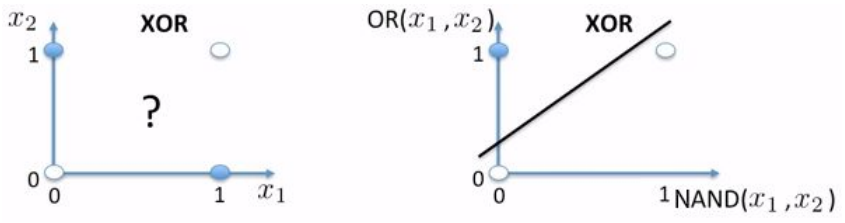

Figure 13 : Représentation de la fonction du OU Exclusif

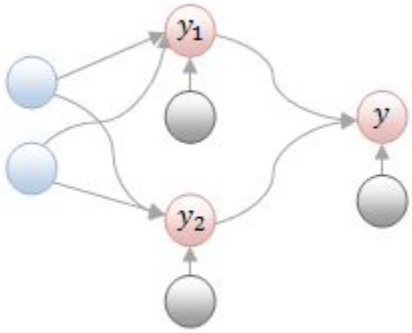

Figure 14 : Réseau de neurones avec deux perceptrons

3. Perceptron multicouche

Nous avons précédemment étudié les perceptrons et nous avons vu que les neurones de sortie étaient chacun connectés aux mêmes informations. Nous les avons perçus comme une couche (alignés verticalement). Ainsi, une couche est constituée de neurones étant connectés aux mêmes informations mais n'étant pas connectés entre eux. Il s'agit maintenant de généraliser le perceptron. On peut ainsi disposer les neurones en plusieurs couches. Ainsi les informations en entrée sont connectées à tous les neurones de la seconde couche, et ainsi de suite jusqu'à la dernière couche, appelée couche de sortie. Toutes les couches exceptée la couche de sortie sont considérées comme couches cachées. Toutefois, il a été prouvé que dans la plupart des cas, un réseau à deux couches (informations » couche cachée » couche de sortie) où chaque neurone de la couche cachée a comme fonction d'activation la fonction sigmoïde et chaque neurone de la couche de sortie a comme fonction d'activation une fonction linéaire permet d'approximer une fonction continue. Il s'agit du théorème de Cybenko. Mais pour des fonctions discontinues, nous n'avons aucune garantie.

La couche cachée d'un perceptron multicouche permet plus d'interaction et instinctivement, on est conscient que notre réseau de neurone pourra apprendre des fonctions plus complexes. On a toujours le choix de la fonction d'activation.

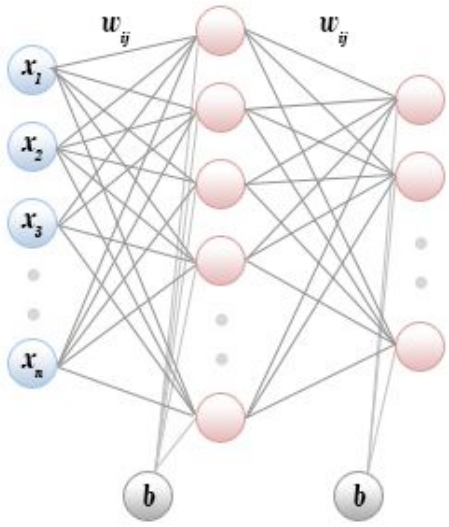

Figure 15 : Modélisation d'un perceptron multicouche avec une seule couche cachée

a. Fonction d'activation

La fonction d'activation, ou fonction de transfert, est une fonction qui doit renvoyer un réel proche de 1 quand les ''bonnes'' informations d'entrée sont données et un réel proche de 0 quand elles sont ''mauvaises''. On utilise généralement des fonctions à valeurs dans l'intervalle réel [0,1]. Quand le réel est proche de 1, on dit que l'unité est inactive. En effet, si les fonctions d'activations sont linéaires, alors le réseau est l'équivalent d'une régression multilinéaire. L'utilisation du réseau de neurone est toutefois bien plus intéressante lorsque l'on utilise des fonctions d'activations non linéaires.

Il y a bien sûr beaucoup de fonctions d'activations possibles, c'est-à-dire répondant aux critères que nous avons données, toutefois dans la pratique il y en a principalement deux qui sont utilisées :

- La fonction de Heaviside
- La fonction sigmoïde

La fonction de Heaviside (Figure 16) est définie par :

$$\forall x \in R \begin{cases} f(x) = 1 \; si \; x \geq 0 \\ f(x) = 0 \; sinon \end{cases}$$

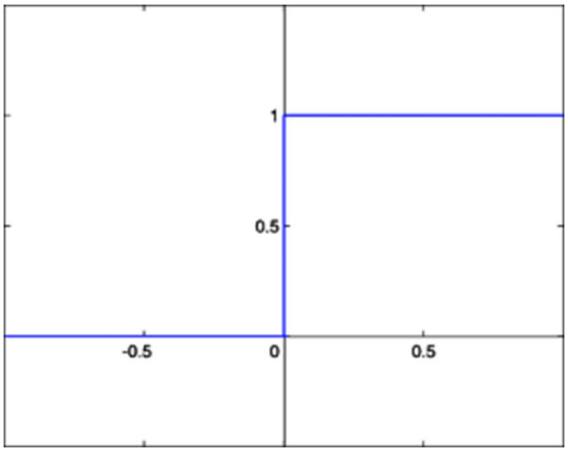

Figure 16 : Graphe de la fonction de Heaviside

La fonction sigmoïde (Figure 17) est définie par : $\forall x \in R, f(x) = \frac{1}{1+e^{-x}}$

Cette fonction présente l'avantage d'être dérivable ainsi que de donner des valeurs intermédiaires des réels compris entre 0 et 1, par opposition à la fonction de Heaviside qui elle renvoie soit 0 soit 1.

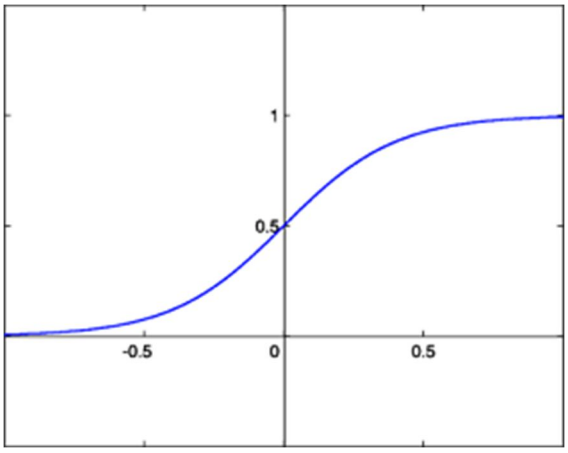

Figure 17 : Graphe de la fonction Sigmoïde

b. Apprentissage

Da la même manière que le perceptron monocouche, le perceptron multicouche est lui aussi capable d'apprentissage. En effet, il existe également un algorithme permettant de corriger les poids vis-à-vis d'un ensemble d'exemples données. Cet algorithme est appelé : Algorithme de rétro-propagation du gradient.

Cet algorithme utilise la même règle de modification des poids (la règle delta) que l'algorithme de Windrow-Hoff. L'algorithme va être donnée dans sa version la plus générale, i.e. avec plusieurs couches cachées. On notera g la fonction d'activation qu'elle soit indéfiniment dérivable. On

notera s_i la sortie du neurones i de la couche de sortie et y_i la sortie attendue pour ce même neurone. Enfin, pour des neurones d'une couche cachée, on notera o_i la sortie calculée du neurone.

```
Entrée : un exemple, sous la forme (vecteur_x,vecteur_y);
epsilon le taux d'apprentissage
un Perceptron MultiCouches avec q-1 couches cachées C1, ..., Cq-1,
une couche de sortie Cq.

Répéter
        Prendre un exemple (vecteur_x,vecteur_y) et calculer g(vecteur_x)

        Pour toute cellule de sortie i   di <- si(1-si)(yi-si) finPour
        Pour chaque couche de q-1 à 1
          Pour chaque cellule i de la couche courante
                di = oi(1-oi) * Somme
                [pour k appartenant aux indices des neurones prenant en entrée la
sortie du neurone i] de dk*w_ki
          finPour
        finPour
        Pour tout poids w_ij <- w_ij + epsilon*di*x_ij finPour
finRépéter
```

La variable ''di'' apparait deux fois dans le code. Il s'agit de deux variables différentes, car en fait on suppose que les neurones sont numérotés de sorte que l'on puisse associer à un identifiant un neurone et réciproquement. Par conséquent, le 'i' de ''di'' identifie un neurone et ainsi on peut effectuer la dernière boucle de manière uniforme sans différencier pour la couche de sortie et les couches cachées.

L'algorithme de rétro-propagation du gradient est une extension de l'algorithme de Windrow-Hoff. En effet, dans les deux cas, les poids sont mis à jour à chaque présentation d'exemple et donc on tend à minimiser l'erreur calculée pour chaque exemple et pas l'erreur globale. Cette méthode donne de bon résultats pratiques. Dans la plupart des cas on rencontre peu de problèmes dus aux minima locaux, mais il y en a. Toutefois, il est moins performant que d'autres algorithmes de propagation d'erreur : il tend moins rapidement vers des poids plus ou moins optimaux. Pour la condition d'arrêt pour le REPETER. C'est de nous de fixer le critère. On peut par exemple répéter cela jusqu'à ce que l'erreur sur chaque exemple descende en dessous d'un certain nombre. Pour le choix de l'architecture initiale du réseau, il reste un problème difficile. Ce choix peut être fait par l'expérience.

III. L'apprentissage profond

Le deep learning est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires. Ces techniques ont permis des progrès importants et rapides dans les domaines de l'analyse du signal sonore ou visuel et notamment de la reconnaissance faciale, de la reconnaissance vocale, de la vision par ordinateur, du traitement automatisé du langage. Dans les années 2000, ces progrès ont suscité des investissements privés,

universitaires et publics importants, notamment de la part du GAFA (Google, Apple, Facebook et Amazon).

En octobre 2015, le programme alphaGo ayant appris à jouer au jeu de go (jeu de plateau originaire de la chine) [10] par la méthode du deep learning a battu par 5 parties à 0 le champion européen Fan Hui. En mars 2016, le même programme a battu le champion du monde Lee Sedol 4 parties à 1 [11].

1. Les réseaux de neurones profonds

Les réseaux de neurones profonds sont des réseaux de neurones qui font leur apprentissage d'une manière profonde, la partie cachée du réseau est constituée de plusieurs couches cachées (Figure 18), chaque couche cachée joue deux rôles, elle est la fois la couche de sortie de la couche précédente et la couche d'entrée de la couche suivante.

L'idée des réseaux de neurones profonds c'est de donner le pouvoir à chaque couche cachée de contribuer dans la phase de la reconnaissance. Prenons le cas de la reconnaissance de forme, les neurones d'entrée vont être les pixels de l'image, ensuite les neurones de la première couche cachée peuvent reconnaitre les bords de la forme, les neurones de la deuxième couche cachée peuvent reconnaitre des formes plus complexes (Triangle, Rectangles …), etc.

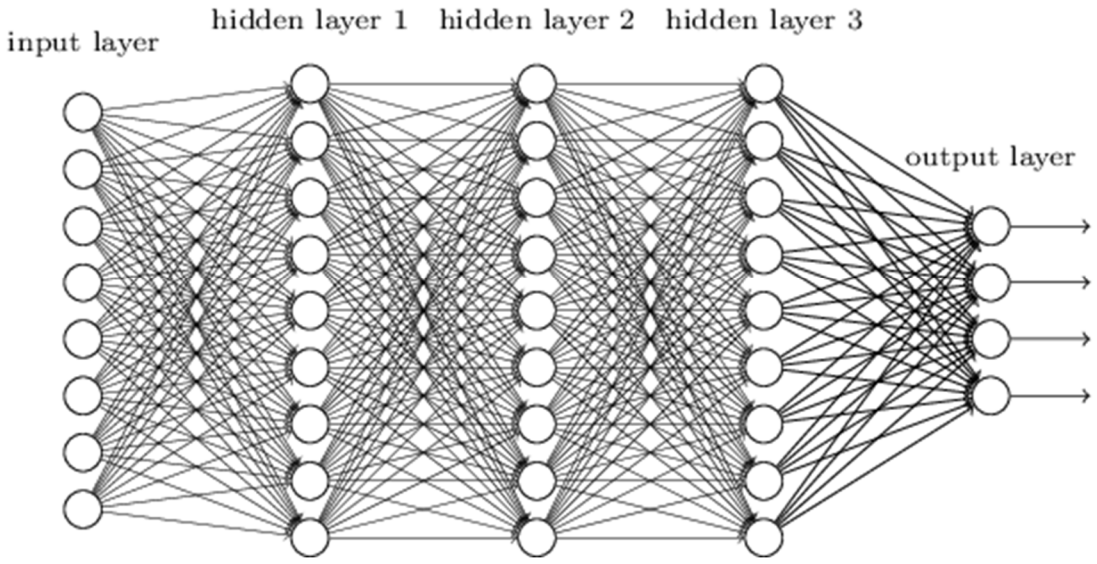

Figure 18 : L'architecture générale du réseau de neurones profond

Le premier réseau de neurones profond selon [12] a été publié par Alexey Girgoverich Ivankhnenko en 1965, il est été constitué avec 8 couches cachées en utilisant l'algorithme d'apprentissage appelé GMDH (GroupMethod of Data Handling).

Mais le problème avec ce type de réseaux c'est que l'apprentissage peut prendre un temps énorme et le sur-apprentissage.

2. Les réseaux de neurones à convolution

Les réseaux de neurones à convolution ont devenu la méthode du choix pour le traitement d'images [13]. Un réseau de neurones à convolution est composé d'un ou plusieurs couches à convolutions (Figure 19). Chaque couche à convolution contient plusieurs couches, et chaque couche est connectée avec la couche d'entrée avec un poids et un biais commun avec tous ces neurones pour que chaque couche détermine un comportement différent des autres couches (Figure 21). Par la suite, les couches à convolution subiront une réduction de dimension (Figure 22) en prenant le maximum d'une fenêtre de neurones et constituant une autre couche. Et à la fin les neurones de cette couche seront les neurones d'entrée de la couche sigmoïde pour faire l'apprentissage (Figure 19).

Les images ont l'avantage d'avoir une structure bidimensionnelle, c'est pour cela les réseaux de neurones à convolution profitent de cet avantage pour donner de meilleurs résultats en traitement d'images.

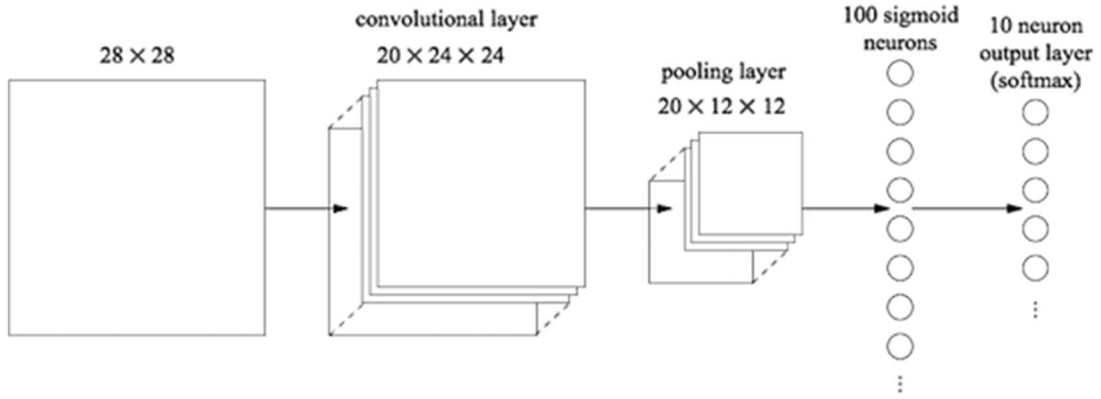

Figure 19 : Architecture du réseau de neurones à convolution

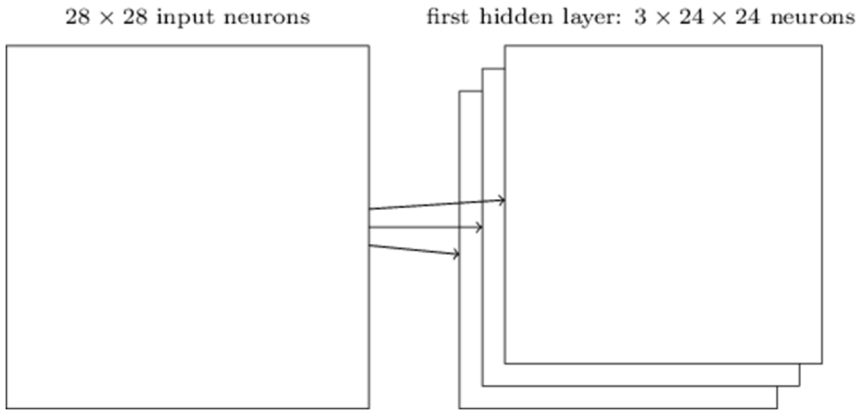

Figure 20 : La première couche à convolution contient 3 couches connectées avec la couche d'entrée

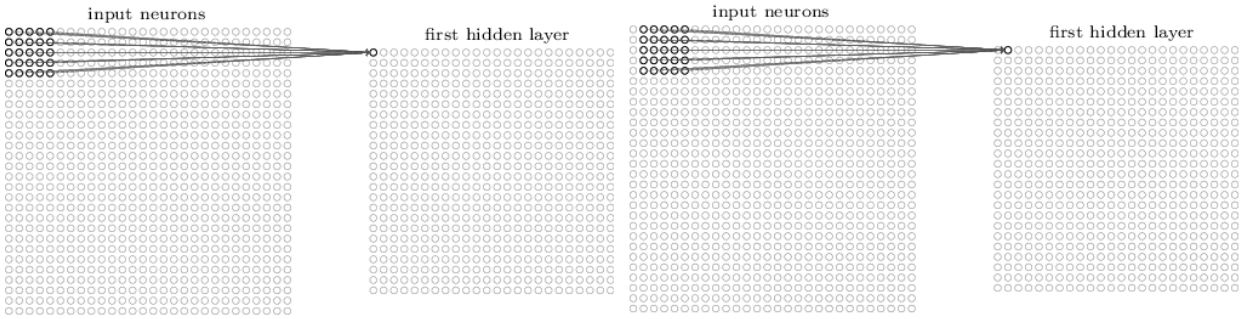

Figure 21 : La connexion des neurones de la première couche cachée de la couche à convolution avec les neurones de la couche d'entrée, chaque couche est connectée avec un poids et un biais commun

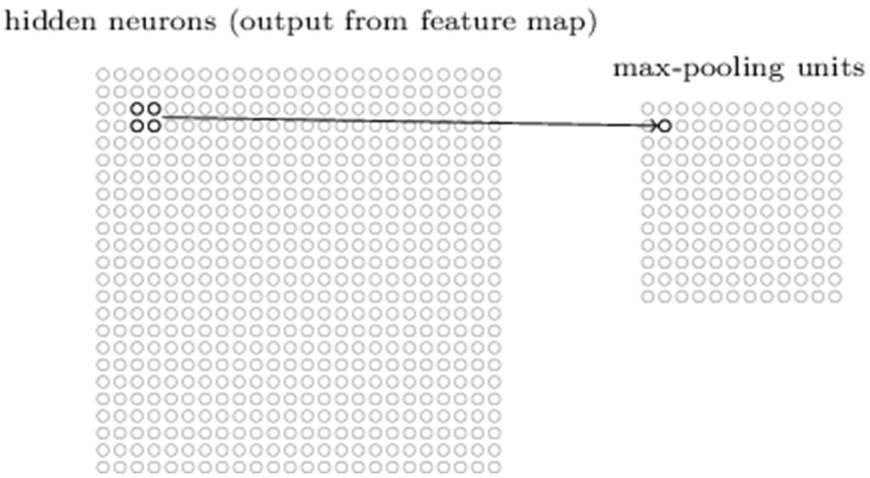

Figure 22 : Réduction de la dimension des couches à convolution

3. Réseaux de croyance profonde

Les réseaux de croyance profonde (Figure 23) sont des modèles génératifs probabilistes composés de plusieurs couches stochastiques de variables latentes (neurones cachés), avec des connexions entres les couches du réseau mais pas entre les neurones des couches [14].

Quand l'apprentissage d'un ensemble de données est en mode non-supervisé, le réseau de croyance profonde peut apprendre la reconstruction des données de l'entrée. Donc, les couches actent comme des extracteurs de caractéristiques dans l'entrée. Après cette étape d'apprentissage, le réseau de croyance profonde peut s'entrainer d'une manière supervisée pour faire la classification.

Les réseaux de croyance profonde peuvent être vu comme composition de réseaux non-supervisés simples (MBR : machines de Boltzmann restreintes) (Figure 23).

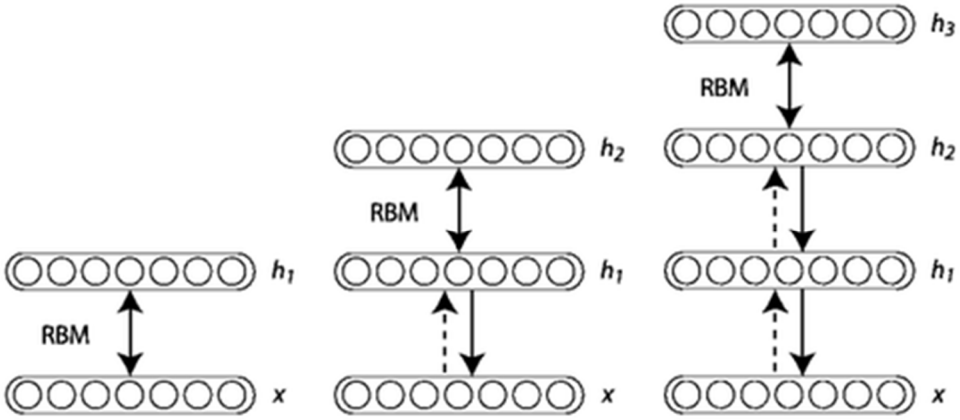

Figure 23 : Architecture des réseaux de croyance profonde

IV. Conclusion

Les réseaux de neurones artificiels sont des réseaux de neurones à une seule cachée, car la plupart des fonctions numériques peuvent être approximée par seulement une couche cachée. Les réseaux de neurones profonds sont des réseaux à plusieurs couches cachées qui peuvent classifier des objets plus complexes théoriquement mais ils sont difficiles à apprendre.

Chapitre 3

La reconnaissance automatique du locuteur par les réseaux de neurones profonds, protocole expérimental et résultats

I. Introduction

Dans ce chapitre, nous allons présenter la base de données des locuteurs sur laquelle nous avons travaillé ainsi que le protocole expérimental de notre approche, et à la fin nous allons analyser les résultats obtenus.

II. Généralités

1. MFCC

La première étape à faire dans un système de reconnaissance automatique du locuteur est l'extraction des caractéristiques, autrement dit, l'identification des composants du signal audio qui seront bénéficiaire pour la reconnaissance automatique du locuteur et rejeter les autres composants.

Le point, concernant la parole, qu'il faut comprendre est que la voix générée par l'homme est filtrée par la forme de l'appareil phonatoire y compris la langue, les dents etc. si on peut approximer cette forme on peut en déduire une représentation précise du phonème produit. La forme de l'appareil phonatoire se manifeste dans l'enveloppe du spectre de puissance d'une durée courte, et le rôle de MFCC est la représentation de cet enveloppe avec une précision.

Les MFCC [15] ou bien les coefficients spectraux issus d'une analyse en banc de filtres (*Mel Frequency Cepstral Coefficients)* sont largement utilisés en reconnaissance automatique de la parole et en reconnaissance automatique du locuteur. Ils sont introduits par *Davis* et *Mermelstrein* en 1980 [16], et depuis cette date ils restent l'état de l'art de l'extraction des caractéristiques.

En bref, les étapes de l'algorithme :

- Découper le signal en petite trames,
- Pour chaque trame, calculer le périodogramme du spectre de puissance,
- Appliquer le *Mel Filterbank* aux spectres de puissance, sommer l'énergie dans chaque filtre,
- Prendre le logarithme de toutes les énergies de *Filterbank*
- Prendre le DCT (*Discrete Cosine Transforme*) ou bien la transformée en cosinus discrète des énergies de *log filterbank*,
- Garder les coefficients 2-13 de DCT, et jeter le reste.

2. UBM

UBM ou bien Universal Background Model est un modèle utilisé en systèmes de la VAL qui permet de prendre les caractéristiques d'un grand nombre de locuteurs est de construire un modèle qui représente les caractéristiques générales des locuteurs en mode indépendant du texte. En vérification le locuteur inconnu est comparé avec le modèle UBM et avec la référence du locuteur sur lequel la vérification est effectuée.

III. Travaux existants

Avant de pouvoir réaliser l'approche de notre projet de fin d'études sur la reconnaissance automatique du locuteur en utilisant les réseaux de neurones profonds nous avons étudié plusieurs articles dans ce domaine et nous venons dans cette section de présenter les articles les plus pertinents.

Nos premières recherches étaient sur la reconnaissance automatique du locuteur par les réseaux de neurones artificiels i.e. avec une seule couche cachée. Nous avons trouvé que les résultats sont très intéressant en utilisant le réseau de neurones artificiel en tant que classifieur, mais pour un nombre très limité de locuteurs comme ils ont montré dans ce travail [17], ils ont trouvé sur une base de données de 10 locuteurs un taux de 100% avec un taux des faux acceptations de 10%, et pour 14 locuteurs un taux de 94% et 12% comme taux des faux acceptations. Donc les réseaux de neurones artificiels sont efficaces pour la classification des locuteurs sur des bases de données d'une dizaine de locuteurs, et spécialement en mode dépendant du texte [18], [19].

Le premier article avec les réseaux de neurones profonds était sur la VAL en mode dépendant du texte [1]. L'idée de ce travail, sponsorisé par Google, revient à prendre les vecteurs MFCC des expressions « Ok Google » tirés de plusieurs locuteurs et les faire passer dans le réseau. L'apprentissage du réseau est un apprentissage supervisé repose sur l'utilisation de ce dernier en tant que systèmes d'extraction des caractéristiques de chaque locuteur. Chaque d-vecteur (vecteur discriminant) représente les caractéristiques d'un locuteur, il est calculé de la dernière couche cachée du réseau de neurones profond (Figure 24).

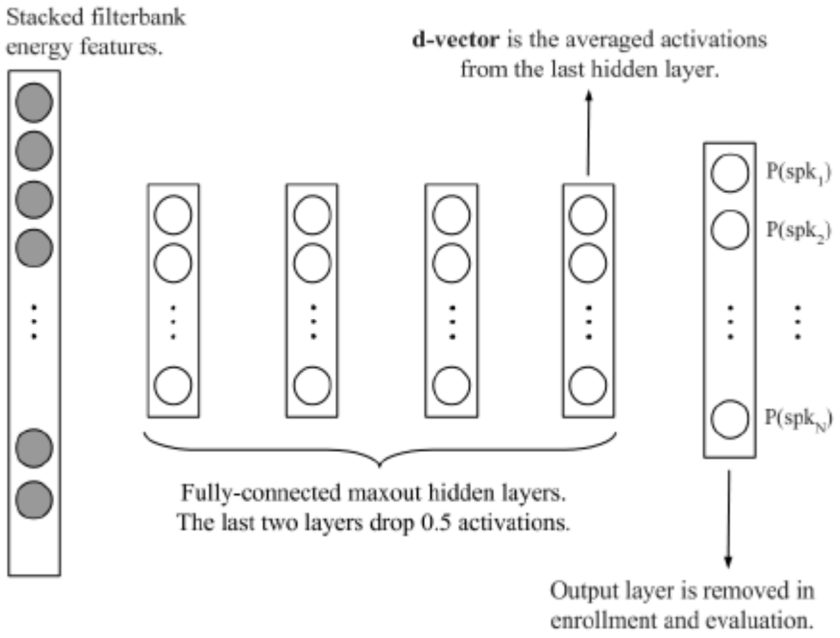

Figure 24 : architecture du réseau de neurones utilisé dans la vérification automatique du locuteur mode dépendant du texte

L'étape d'apprentissage représente l'étape où le modèle de chaque locuteur est entrainé. Ensuite, dans l'étape de classification, une décision, selon un seuil, est réalisée en comparant le d-vecteur obtenu avec le d-vecteur du modèle. Pour les résultats ils ont obtenu 4.54% comme taux d'erreur EER (Equal Error Rate) pour 4 expressions appris par le système pour chaque locuteur, et 2% pour 20 expressions appris par le système pour chaque locuteur.

Un autre travail traite un nouveau Framework pour la reconnaissance automatique du locuteur, où l'extraction des caractéristiques nécessaires pour l'état de l'art du modèle i-vecteur est monitoré par les réseaux de neurones profonds [20], leur idée revient à remplacer le standard GMM pour la production des alignements de trames par les réseaux de neurones profonds qui guident la modélisation du locuteur. Par conséquent, ils ont diminué le taux d'erreur de 1.99% à 1.39%. Plusieurs articles ont aussi suivi le même chemin de remplacement de UBM/GMM par DNN/i-vecteur [21], [22].

Dans la même idée d'extraction des caractéristiques, il y a une autre manière d'extraire les caractéristiques à partir d'un réseau de neurones profonds, plusieurs articles resservent une couche cachée du réseau appelée la couche Bottleneck, la plupart d'articles réduisent la taille de cette couche positionnée huitième à six neurones [23], [24], [25].

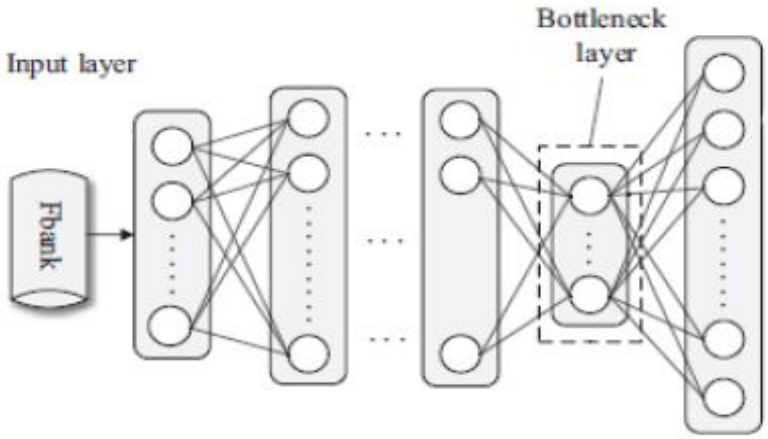

Figure 25 : Architecture d'un réseau de neurones profond avec la couche Bottleneck

Les réseaux de neurones à convolution et les réseaux de croyance profonde sont aussi utilisés et ils sont très performants dans les milieux bruyants [26] [27].

IV. Base de données

Dans le but d'acquérir un ensemble adéquat de parole d'entrainement et de test pour la reconnaissance automatique du locuteur, une base de données a été demandée. Par conséquent, plusieurs bases de données peuvent aider les chercheurs pour comparer l'efficacité de différentes approches sur des données communes et de sélectionner la plus prometteuse. Pour cette raison, on peut trouver plusieurs bases de données comme TIMIT[28] qui est utilisées largement en

reconnaissance du locuteur et qui comporte un grand nombre de locuteurs, elle contient 630 locuteurs (438 Hommes/ 192 Femmes). D'autres bases de données sont utilisées en reconnaissance automatique du locuteur comme Gandalf[29], CSLU[30], SIVA[31], COST250[32], ELSDSR[33], etc.

Mais ces bases de données sont très coûteuses, pour cette raison nous avons décidé de réaliser notre propre base de données qui sera open source pour les chercheurs. Ce travail a été réalisé avec deux doctorants du laboratoire LSIA, sous l'encadrement de monsieur Jamal Kharroubi, enseignant chercheur à la Faculté des Sciences et Techniques de Fès.

Les locuteurs de notre base de données sont collectés du site web www.voxforge.org, pour la langue française, anglaise, espagnole, italienne et deutsche, et pour la langue arabe nous l'avons collectée manuellement. La base de données est constituée de 159 Hommes et 59 Femmes et 50 UBM pour la base de développement, et de même pour la base d'évaluation, au total nous avons 536 locuteurs. La Figure 26 présente la répartition de différentes langues constituantes notre base de données.

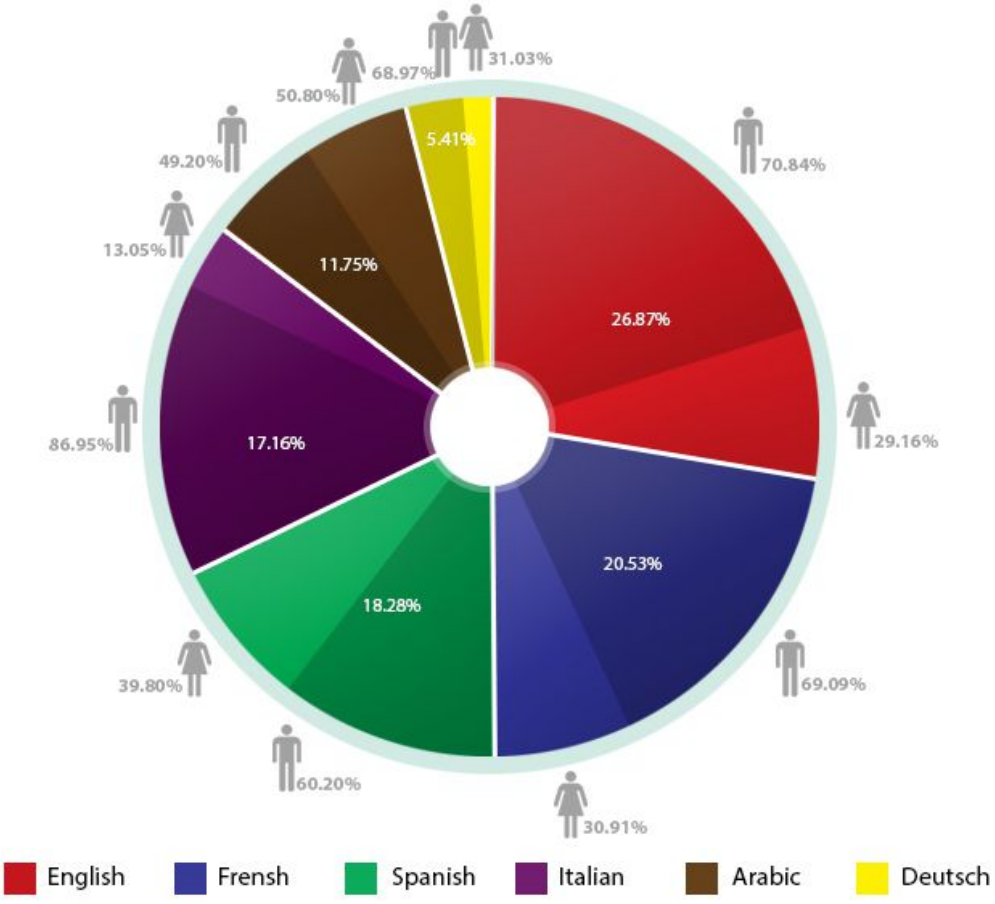

Figure 26 : Architecture de la base de données FSCSR Speech Corpus

V. Protocole expérimental

1. Extraction des caractéristiques

a. Extraction des MFCC

Après que nous avons expliqué comment extraire les vecteurs MFCC d'un signal audio, nous prenons maintenant la base de données contenant 159 hommes et 59 femmes en plus de 50 locuteurs constituants l'UBM. Chaque locuteur a 30 seconds d'apprentissage et un ensemble d'enregistrements de test de 6 seconds. Après l'extraction des vecteurs MFCC chaque second audio donne 100 vecteurs MFCC, car chaque 10 ms est représentée par un vecteur MFCC, et chaque vecteur MFCC contient 13 valeurs réelles.

b. Extraction des vecteurs Bottleneck

Notre approche consiste à diviser l'ensemble de vecteurs de chaque locuteur i.e. l'ensemble de vecteurs d'apprentissage de 30 seconds, en 23 blocs, le premier bloc contient tous les vecteurs d'apprentissage, ensuite on divise les 30 seconds en deux blocs, en trois blocs et en six blocs, à la fin on fait glisser une fenêtre d'une taille de 600 vecteurs avec un pas de 200 vecteur pour aboutir à 11 vecteurs de plus (Figure 27, Table 2).

Concernant les enregistrements de test et les enregistrements de UBM, on prend le bloc de tous les vecteurs.

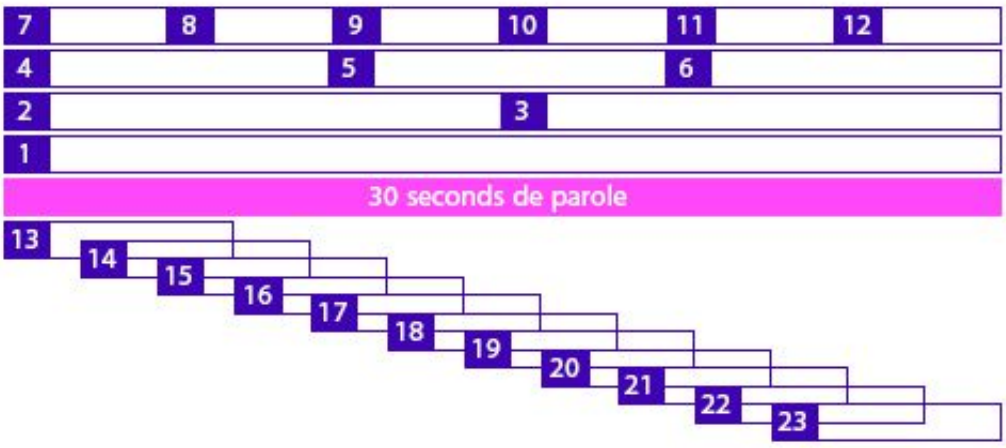

Figure 27 : Représentation de 30 seconds d'apprentissage d'un locuteur par 23 blocs de vecteurs

Table 2 : Nombre de vecteurs de chaque bloc (23 blocs en 30 seconds)

Blocs	Nombre de vecteurs
1	3000
2, 3	1500
4, 5, 6	1000
7, 8, 9, 10, 11, 12	500
13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23	600

Après la présentation de chaque locuteur par 23 blocs, et pour chaque bloc, on pend chaque vecteur MFCC et on le place dans l'entrée de notre réseau de neurones (Figure 28):

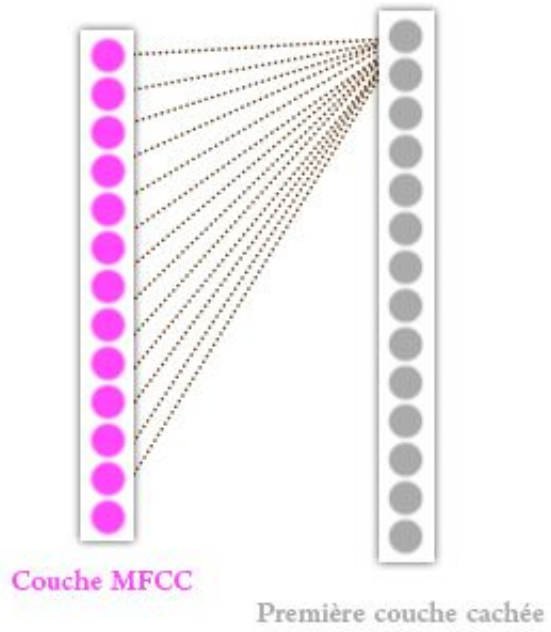

Figure 28 : Connexion entre les neurones de la première couche cachée et la couche d'entrée MFCC

Chaque neurone de la première couche cachée est connecté avec les neurones de couche d'entrée MFCC selon la relation suivante :

$$x_j^{(1)} = \sum_{j=0}^{j=13} \sum_{i=0}^{i=12} \omega_{i,j} . x_i^{(0)}$$

Avec $\omega_{i,j}$ est le poids entre le neurone i de la couche d'entrée et le neurone j de la première couche cachée. Ensuite, nous rajoutons 6 autres couches cachées, et chaque neurone est connecté avec les autres neurones dans les 7 couches cachées avec la relation suivante :

$$x_j^{(k)} = \sum_{j=0}^{j=13} \sum_{i=0}^{i=13} \omega_{i,j} . x_i^{(k-1)}$$

Avec k = {2, …, 7}.

Et à la fin nous calculons les neurones de la huitième couche cachée avec la relation suivante :

$$x_j^{(8)} = \sum_{j=0}^{j=5} \sum_{i=0}^{i=13} \omega_{i,j} . x_i^{(7)}$$

L'idée maintenant revient à construire un système d'extraction des caractéristiques des locuteurs, nous avons construit ce système à l'aide d'un réseau de neurones constitué d'une couche d'entrée (vecteurs MFCC) de 13 neurones, et 7 couches cachées de 14 neurones, et la dernière couche appelée la couche Bottleneck de seulement 6 neurones (Figure 29).

Pour chaque bloc on calcule le vecteur Bottleneck de chaque vecteur MFCC, puis on calcule la moyenne des vecteurs pour avoir un seul vecteur Bottleneck par chaque bloc. Par la suite on normalise les vecteurs Bottleneck en divisant chaque valeur du vecteur sur la somme des valeurs du vecteurs.

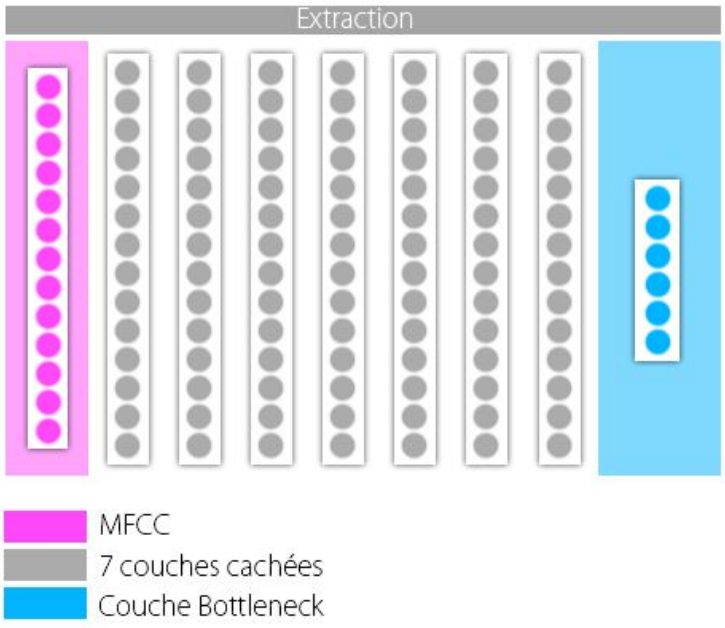

Figure 29 : Architecture du système d'extraction des caractéristiques Bottleneck

L'étape d'extraction des caractéristiques permet de représenter chaque locuteur par 23 vecteurs de 6 valeurs pour l'entrainement, et pour chaque 6 seconds de test par un vecteur Bottleneck, de même pour l'UBM, chaque locuteur est représenté par un vecteur Bottleneck.

2. Méthodes utilisées

Afin d'évaluer le système d'extraction des vecteurs Bottleneck pour la VAL, nous considérons qu'un locuteur est représenté par 23 vecteurs Bottleneck, et le modèle UBM est représenté par 50 vecteurs Bottleneck, et chaque test est représenté par un seul vecteur Bottleneck. Nous optons pour les méthodes de classifications suivantes : la classification selon la distance, les K plus proches voisins, le perceptron multicouche et les SVM ou bien les Séparateurs à Vaste Marge.

a. Distances

Les distances sont le moyen le plus simple pour tester notre système, pour se faire, plusieurs distances sont disponibles :

- La distance la plus célèbre est la distance Euclidienne, elle est calculée selon la relation suivante :

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

  Avec p et q sont deux vecteurs d'une dimension n.
- La deuxième distance que nous avons utilisée est la distance de Manhattan, nommée aussi taxi-distance, elle est calculée selon la relation suivante :

$$d(p,q) = \sum_{i=1}^{n}|p_i - q_i|$$

  Avec p et q sont deux vecteurs d'une dimension n.

La décision prise sur chaque vecteur de test est calculée selon la relation suivante :

$$Décision = \begin{cases} \mathbf{1}\ si\ \dfrac{\frac{1}{23}\sum_{i=1}^{23} d_i(V_{BN}(S_{test}), V_{BN}(S^i_{train}))}{\frac{1}{50}\sum_{i=1}^{50} d_i(V_{BN}(S_{test}), V_{BN}(S^i_{UBM}))} \leq \varepsilon \\ \mathbf{0}\ sinon \end{cases}$$

Avec $V_{BN}(S_{test})$ représente le vecteur Bottleneck, $V_{BN}(S^i_{train})$ le vecteur Bottleneck i d'entrainement parmi les 23 vecteurs représentants la référence du locuteur cible et $V_{BN}(S^i_{UBM})$ représente le vecteur i parmi les 50 vecteurs représentants la référence du modèle UBM.

d : c'est la distance utilisé (Euclidienne ou Manhattan).

Normalement dans cette équation le fait que le numérateur est supérieur au dénominateur donne une idée que le vecteur de test plus proche aux vecteurs d'entrainement qu'aux vecteurs du modèle UBM, i.e. le seuil qu'on peut utiliser est $\varepsilon = 1$, mais les expériences ont montré que le meilleur seuil à utiliser est $\varepsilon = 0.55$.

Nous avons utilisé plusieurs distances comme la distance de Minkowski et la distance de Tchebychev, mais les plus répondus sont Manhattan et Euclidienne.

b. KPPV

La deuxième méthode que nous avons utilisée c'est les K Plus Proches Voisins, en anglais KNN (K-Nearest Neighbors), l'algorithme KPPV figure parmi les plus simples algorithmes d'apprentissage artificiel. Dans le contexte de classification d'une nouvelle observation, l'idée fondatrice simple est de faire voter les plus proches voisins de cette observation. La classe de l'observation est déterminée en fonction de la classe majoritaire parmi les k plus proches voisins de l'observation.

La méthode KPPV est donc une méthode à base de voisinage, non-paramétriques. C'est une extension de l'idée du plus proche voisin, qui est largement et communément utilisée en pratique. La plus proche observation n'est plus la seule observation utilisée pour la classification. Nous utilisons désormais les k plus proches observations. Ainsi la décision est en faveur de la classe majoritairement représentée par les k voisins.

Dans notre cas, les entrées les entrées sont les vecteurs de la référence du locuteur cible et les vecteurs du modèle UBM, si le vecteur de test est plus proche d'un vecteur de la référence du locuteur la sortie sera 1, et elle sera 0 si le vecteur de test est plus proche d'un vecteur du modèle UBM. Le choix de k doit être impaire pour avoir une majorité qui aide à la décision et pour ne pas avoir une égalité. Pour $k = 1$ nous aurons juste un vecteur plus proche, pour $k = 3$ nous aurons trois vecteurs plus proches etc. (Figure 30).

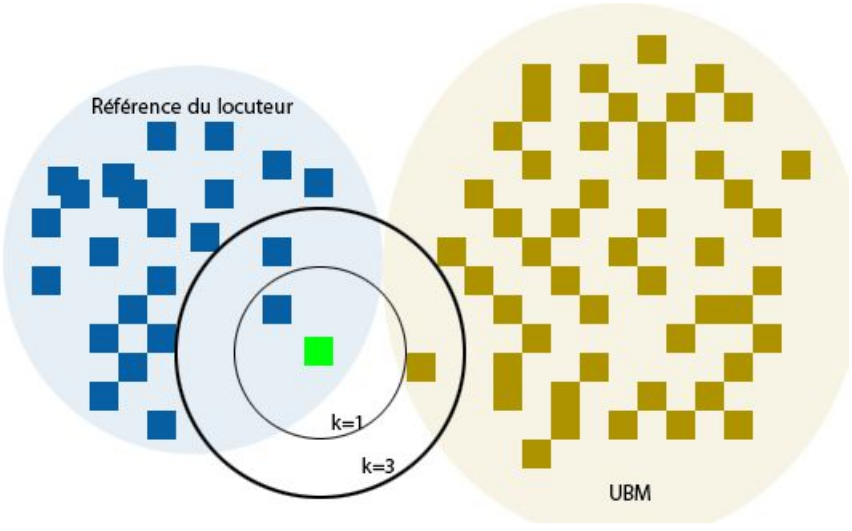

Figure 30 : Classification par les KPPV d'un vecteur Bottleneck de test (vert), pour k=1 le plus proche vecteur c'est le vecteur appartenant aux vecteurs représentant la référence du locuteur (bleu), et pour k = 3, deux vecteurs de la référence sont proches et un vecteur du modèle UBM est proche.

c. Perceptron multicouche

La troisième méthode que nous avons utilisée c'est le réseau de neurones artificiel avec une seule couche cachée. Nous avons bien expliqué l'algorithme du perceptron multicouche dans le deuxième chapitre donc nous ne revenons pas sur ce point.

Notre réseau de neurones artificiel vient juste après le système d'extraction des vecteurs Bottleneck, son entrée est exactement la couche Bottleneck qui est constitué de 6 neurones, la couche cachée du réseau de neurones artificiel est constituée de 9 neurones et la couche de sortie est formé d'un seul neurone (Figure 32).

La fonction d'activation que nous avons utilisée c'est la fonction tangente hyperbolique, car elle a l'avantage de présenter des valeurs entre -1 et 1 au lieu de l'intervalle [0,1] de la fonction sigmoïde.

La fonction de la tangente hyperbolique est définie comme suit :

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

La Figure 31 représente la courbe de la fonction $tanh$:

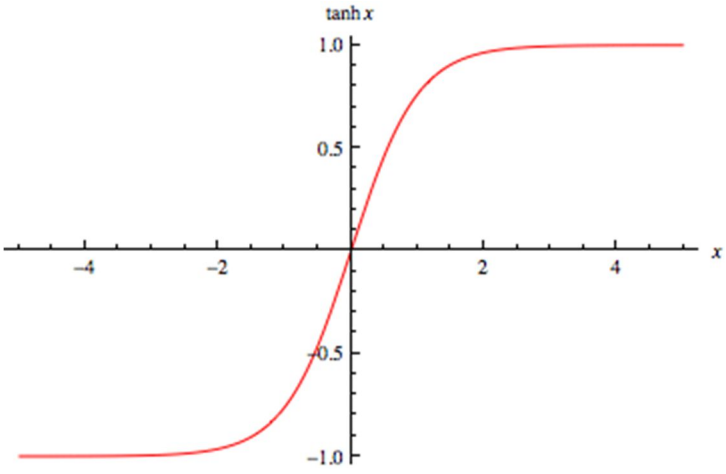

Figure 31 : Graphe représentatif de la fonction Tanh

La démarche pour faire apprendre les vecteurs consiste à passer les 23 vecteurs de la référence du locuteur, et en utilisant l'algorithme de propagation et de rétro propagation pour les converger vers 1 et en même temps apprendre les 50 vecteurs du modèle UBM, et les converger vers -1. Nous avons utilisé un nombre fixe d'itération 80,000.

Pour tester il suffit de prendre un vecteur Bottleneck de test et lui appliqué l'algorithme de propagation, si la sortie est proche de -1 le système n'accepte pas et si la sortie est proche de 1 le système accepte.

La décision est prise selon la relation suivante :

$$Décision = \begin{cases} \mathbf{1}\ sortie \leq \varepsilon \\ \mathbf{0}\ sinon \end{cases}$$

Nous avons fixé le seuil d'acceptation à $\varepsilon = 0.64$ par expérience.

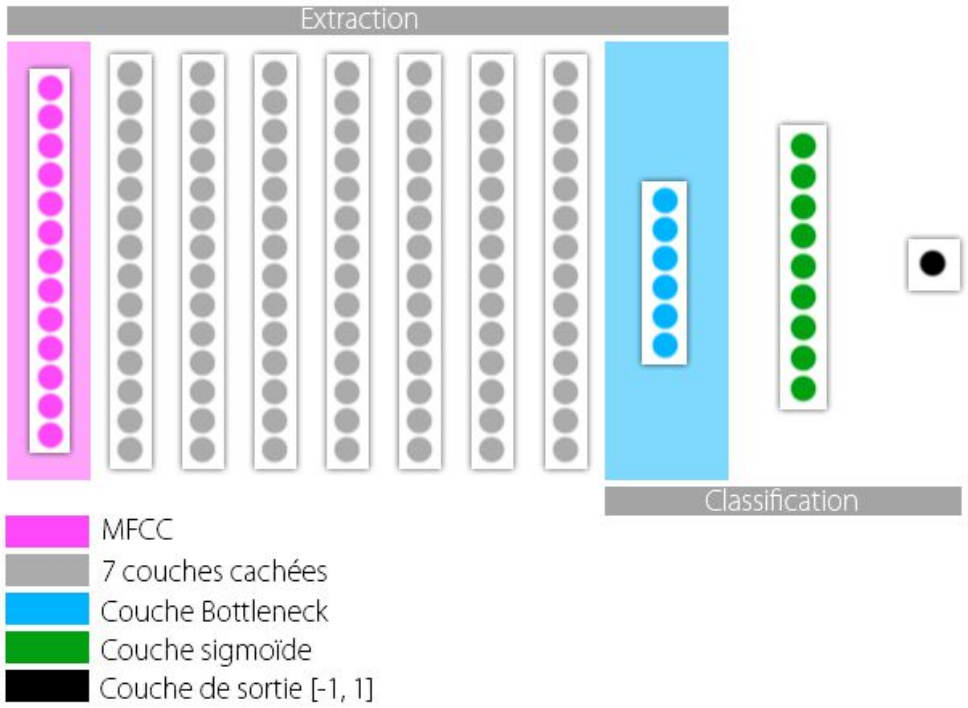

Figure 32 : Architecture du réseau de neurones profond qui est utilisé pour l'extraction des vecteurs Bottleneck qui sont utilisé par la suite comme entrées du réseau de neurones artificiel pour classifier les vecteurs Bottleneck, le voisinage de -1 pour les vecteurs appartenant à la classe UBM, et le voisinage de 1 pour les vecteurs appartenant à la classe des vecteurs de la référence du locuteur cible.

d. SVM

Les Machines à Vecteurs de Support ou Séparateurs à Vaste Marge sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les SVM sont une généralisation des classifieurs linéaires. Ils reposent sur deux idées clés : la notion de marge maximale et la notion de fonction noyau. La première idée clé est la notion de marge maximale où la marge est une distance entre la frontière de séparation et les échantillons les plus proches (Figure 33). Ces derniers sont appelés vecteurs supports. Dans les SVM, la frontière de séparation est choisie comme celle qui maximise la marge. Ce choix est justifié par la théorie de statistique de l'apprentissage, qui montre que la frontière de séparation de marge maximale possède la plus petite capacité [34], le problème est de trouver cette frontière séparatrice optimale, à partir d'un ensemble d'apprentissage.

La deuxième idée clé des SVM est transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, dans lequel il est probable qu'il existe une séparatrice linéaire. Ceci est réalisé grâce à une fonction noyau, qui doit respecter les conditions

du théorème de Mercer [35], et qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace. Les fonctions noyau permettant de transformer un produit scalaire dans un espace de grande dimension, ce qui est coûteux, en une simple évaluation ponctuelle d'une fonction.

Les SVM peuvent, comme nous avons dit, être utilisés pour résoudre des problèmes de classification. La résolution de ce problème passe par la construction d'une fonction h qui a un vecteur d'entrée x fait correspondre une sortie y :

$$y = h(x)$$

Nous avons un problème de discrimination à deux classes : la classe de la référence du locuteur avec les 23 vecteurs, et la classe du modèle UBM avec les 50 vecteurs, donc $y \in \{0, 1\}$, 0 pour les vecteurs UBM et 1 pour les vecteurs de la référence du locuteur, et $X = V_{BN}(S_{train}) + V_{BN}(S_{UBM})$.

Nous avons utilisé une bibliothèque appelée *sklearn* [36] en Python pour appliquer les SVM plus facilement à notre système d'extraction des vecteurs Bottleneck, cette dernière permet d'en classifier en choisissant une fonction noyau parmi quatre existantes :

- Linéaire : $< x, x' >$
- Polynomiale : $(\gamma < x, x' > + r)^d$
- RBF : $exp(-\gamma\ |x, x'|^2\ )$
- Sigmoïde : $tanh(\gamma < x, x' > + r\ )$

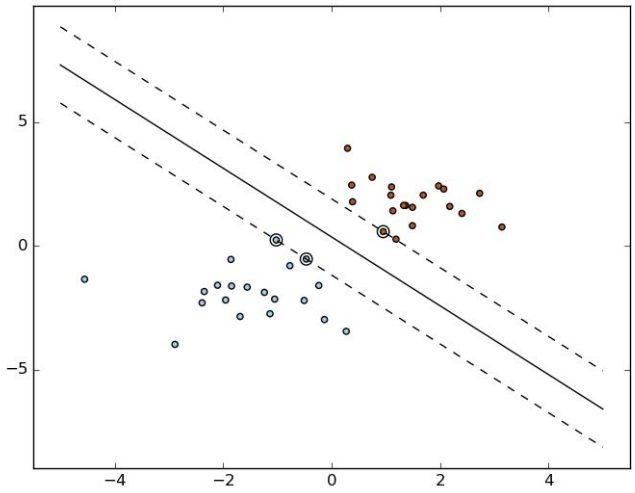

Figure 33 : Séparation entre deux classes (points bleus et points rouges) par un hyperplan

3. Protocole expérimental détaillé

Après le calcul des vecteurs Bottleneck de tous les enregistrements de la base de données, on prend chaque vecteur de test et on le compare avec toutes les références des locuteurs existants, si le vecteur de test du locuteur x correspond à la référence du locuteur x on le considère comme un client ou bien un vrai accès, sinon on le considère comme un faux rejet. Mais si le vecteur de test du locuteur x correspond à une référence d'un locuteur autre que x on le considère comme un faux accès, sinon il est considéré comme un vrai rejet.

A la fin on obtient une table qui contient, pour chaque locuteur, le nombre des vrais accès et les faux rejets ainsi que les faux accès et vrais rejets.

On note TFA le Taux des Faux Accès, VR les Vrais Rejets, FA les Vrais Accès et FR les Faux rejets, donc : TFA = FA / Nombre total des accès non clients (imposteur)

$$TFA = \frac{FA}{FA + VR}$$

On note aussi TFR le Taux des Faux Rejets, VA les Vrais Accès, FR les Faux Rejets et FR les Faux Rejets, donc : TFR = Nombre des FR / Nombre total des accès clients

$$TFR = \frac{FR}{FR + VA}$$

Alors le Taux d'égal à erreur ou bien le EER (Equal Error Rate) c'est le nombre où TFA = TFR. Le meilleur EER c'est le plus petit.

VI. Analyse des résultats

En utilisant la distance Euclidienne, nous avons trouvé 16.42% comme taux d'erreur pour les locuteurs masculins, et 16.32% comme taux d'erreur pour les locuteurs féminins (Table 3).

Table 3 : Le taux de la vérification ainsi que le taux d'erreur pour le système des hommes et le système des femmes sur la base de données FSCSR Speech Corpus en utilisant la distance Euclidienne.

Distance Euclidienne	
	EER
Homme	16.42%
Femme	16.32%

Mais pour la distance Manhattan, nous avons pu diminuer le taux d'erreur par 18.75%, donc le taux d'erreur était 13.53% pour les locuteurs masculins, et 13.40% pour les locuteurs féminins (Table 4). Nous avons tracé la courbe DET pour les locuteurs masculins en utilisant les deux distances Euclidienne et Manhattan (Figure 34)

Table 4 : Le taux de la vérification ainsi que le taux d'erreur pour le système des hommes et le système des femmes sur la base de données FSCSR Speech Corpus en utilisant la distance Manhattan.

Distance Manhattan	
	EER
Homme	13.53%
Femme	13.40%

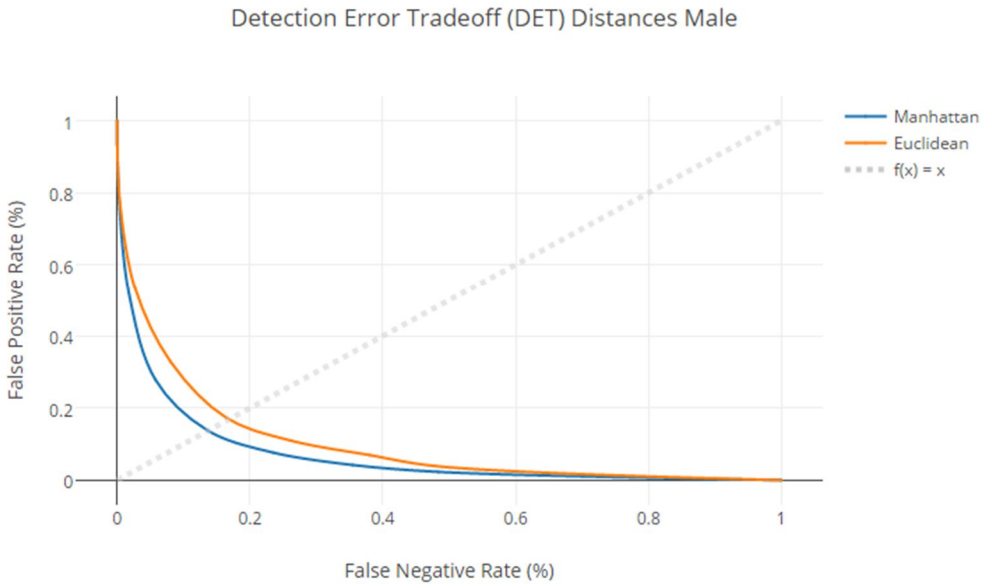

Figure 34 : Les courbes DET de vérification automatique du locuteur en utilisant la distance Euclidienne et la distance Manhattan

Ensuite nous avons utilisé les K plus proches voisins par la distance Manhattan, pour le plus proche voisin (K=1), ça donne le même résultat que nous avons obtenu par la vérification à l'aide de la distance Manhattan. Pour K=3 le taux d'erreur a diminué vers 11.22% pour les hommes et 10.72% pour les femmes, mais le meilleur taux d'erreur obtenu était pour 5 voisins, nous avons obtenu 7.93% pour les hommes et 7.51% pour les femmes. Les résultats sont représentés dans le tableau (Table 5).

Table 5 : Le taux de la vérification ainsi que le taux d'erreur pour le système des hommes et le système des femmes sur la base de données FSCSR Speech Corpus en utilisant les K les plus proches voisins pour K = 1, K = 3, K = 5 et K = 7 en utilisant la distance Manhattan.

KPPV par la distance : Manhattan	
K = 1	
	EER
Homme	13.53%
Femme	13.40%

K = 3	
	EER
Homme	11.22%
Femme	10.72%

K = 5	
	EER
Homme	7.93%
Femme	7.51%

K = 7	
	EER
Homme	8.59%
Femme	8.10%

La représentation des résultats sous forme des courbes DET montre que la courbe la plus significative est la courbe que convient à K=5 (Figure 35).

Figure 35 : Les courbes DET de la VAL en utilisant les K plus proche voisins par la distance Manhattan, pour 1 voisin, 3 voisins, 5 voisins et 7 voisins.

Le troisième système que nous avons utilisé, c'est le système de vérification en utilisant un réseau de neurones artificiel, ce système présente le plus petit taux d'erreur obtenu parmi tous les systèmes utilisés, nous avons trouvé 6.78% comme taux d'erreur sur les locuteurs masculins, et 5.24% et sur les locuteurs féminins (Table 6).

Table 6 : Le taux de la vérification ainsi que le taux d'erreur pour le système des hommes et le système des femmes sur la base de données FSCSR Speech Corpus en utilisant un réseau de neurones artificiel avec 9 neurones dans la couche sigmoïde, et comme fonction d'activation la tangente hyperbolique.

MLP	
	EER
Homme	6.78%
Femme	5.24%

Nous avons également testé les SVM, et nous avons trouvé 9.73% comme taux d'erreur sur les locuteurs masculins et 8.64% sur les locuteurs féminins.

Table 7 : Le taux de la vérification ainsi que le taux d'erreur pour le système des hommes et le système des femmes sur la base de données FSCSR Speech Corpus en utilisant les SVM et RBF comme fonction noyau.

SVM	
	EER
Homme	9.73%
Femme	8.64%

VII. Conclusion

Dans ce chapitre nous avons présenté notre approche basée sur l'extraction des vecteurs Bottleneck, ensuite nous avons parlé des différentes méthodes utilisées pour évaluer notre système d'extraction des caractéristiques, nous avons remarqué que la vérification à l'aide d'un réseau de neurones artificiel donne les meilleurs taux d'erreur.

Conclusion et perspectives

Le principal thème d'étude de ce projet de fin d'études a été la reconnaissance automatique du locuteur par les réseaux de neurones profonds. Nous avons commencé par le lecture et l'analyse de plusieurs articles portent sur la reconnaissance automatique du locuteur, et d'autres articles sur l'utilisation des réseaux de neurones artificiels et profonds dans la RAL et la reconnaissance automatique de la parole. Ensuite nous avons réalisé notre approche qui utilise un réseau de neurones profond comme extracteur de vecteurs Bottleneck, et les distances, KPPV, SVM et PMC comme classifieurs.

Dans ce présent rapport, nous avons débuté par le premier chapitre qui était un chapitre introductif sur la RAL et les différentes tâches effectuées par un système RAL, ainsi que les approches utilisées par les chercheurs. Le deuxième chapitre traitait les réseaux de neurones artificiels, en commençant du plus simple réseau de neurones nommé le perceptron, et en passant sur sa version évaluée, le perceptron multicouche, et à la fin nous avons étudié les réseaux de neurones profonds et ses différentes architectures. Le troisième chapitre présentait notre approche utilisée, qui se repose sur l'utilisation d'un réseau de neurones profond à 8 couches cachées, pour l'extraction des vecteurs Bottleneck, qui sont utilisés par la suite comme vecteurs représentatifs des locuteurs, et par conséquent des vecteurs d'entrée de plusieurs classifieurs. Les résultats de notre approche étaient très intéressants en utilisant les réseaux de neurones artificiels comme classifieur, nous avons obtenu 5.24% EER pour les locuteurs féminins, et 6.78% EER pour les locuteurs masculins. Dans une autre méthode de classification, nous avons trouvé que la distance Manhattan donne des résultats plus intéressants que la distance Euclidienne, c'est pour cela, nous avons utilisé dans les K plus proche voisins la distance Manhattan, et nous avons trouvé que l'utilisation de cinq voisins est le point critique de cette méthode, pour lequel on trouve les résultats les plus intéressants.

Ce stage de fin d'études m'a permis de travailler en équipe, en rédigeant mon premier article dans le domaine de la recherche avec lequel nous avons participé à une conférence IEEE CIST'16, qui aura lieu 24-26 Octobre 2016 à Tanger. Cet article porte sur la base de données « FSCSR Speech Corpus » qui est une base de données open sources sur laquelle nous avons testé le système décrit dans ce rapport.

Les travaux de recherche présentés dans ce travail peuvent être poursuite de plusieurs façons, on peut trouver des résultats plus intéressant en testant différentes architectures des réseaux de neurones profonds, le premier travail qu'il faut faire, c'est d'essayer d'avoir des vecteur Bottleneck très représentatifs pour chaque locuteur et en même temps ces vecteurs doivent être plus dispersés dans l'espace. Ceci revient à faire un préapprentissage non supervisé ou bien semi-supervisé pour le réseau d'extraction des vecteurs Bottleneck. La deuxième chose à faire, c'est le passage à la voix sur IP « VoIP » car c'est le domaine le plus intéressant dont on peut très bien bénéficier des résultats de la reconnaissance automatique du locuteur en temps réel.

Références

[1] E. Variani, X. Lei, E. McDermott, I. L. Moreno, et J. Gonzalez-Dominguez, « Deep neural networks for small footprint text-dependent speaker verification », in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, p. 4052-4056.

[2] H. Hattori, « Text-independent speaker recognition using neural networks », in *, 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1992. ICASSP-92*, 1992, vol. 2, p. 153-156 vol.2.

[3] J. F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, et C. Wellekens, « A speaker tracking system based on speaker turn detection for NIST evaluation », in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings*, 2000, vol. 2, p. II1177-II1180 vol.2.

[4] A. Rosenberg, I. Magrin-Chagnolleau, et S. Parthasarathy, « Speaker Detection in Broadcast Speech Databases », présenté à Proceedings of International Conference on Spoken Language Processsing, 1998.

[5] J.-F. Bonastre, P. Delacourt, C. Fredouille, S. Meignier, T. Merlin, et C. J. Wellekens, « Différentes stratégies pour le suivi de locuteur », 2000.

[6] S. Meignier, J. F. Bonastre, C. Fredouille, et T. Merlin, « Evolutive HMM for multi-speaker tracking system », in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings*, 2000, vol. 2, p. II1201-II1204 vol.2.

[7] C. Fredouille, *Approche statistique pour la reconnaissance automatique du locuteur : informations dynamiques et normalisation bayesienne des vraisemblances*. Avignon, 2000.

[8] B. S. Atal, « Automatic recognition of speakers from their voices », *Proc. IEEE*, vol. 64, n° 4, p. 460-475, avr. 1976.

[9] F. Rosenblatt, « The perceptron: a probabilistic model for information storage and organization in the brain », *Psychol. Rev.*, vol. 65, n° 6, p. 386-408, nov. 1958.

[10] « Le jeu de Go de Pierre Aroutcheff ». [En ligne]. Disponible sur: http://bibliographie.jeudego.org/le_jeu_de_go_de_pierre_aroutcheff_1.php. [Consulté le: 21-avr-2016].

[11] « AlphaGo | Google DeepMind ». [En ligne]. Disponible sur: http://deepmind.com/alpha-go. [Consulté le: 21-avr-2016].

[12] J. Schmidhuber, « Deep learning in neural networks: An overview », *Neural Netw.*, vol. 61, p. 85-117, janv. 2015.

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et L. D. Jackel, « Backpropagation Applied to Handwritten Zip Code Recognition », *Neural Comput.*, vol. 1, n° 4, p. 541-551, déc. 1989.

[14] « Deep belief networks - Scholarpedia ». [En ligne]. Disponible sur: http://www.scholarpedia.org/article/Deep_belief_networks. [Consulté le: 05-mai-2016].

[15] X. Huang, A. Acero, et H.-W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, 1st éd. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[16] S. Davis et P. Mermelstein, « Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences », *IEEE Trans. Acoust. Speech Signal Process.*, vol. 28, n° 4, p. 357-366, août 1980.

[17] F. Mueen, A. Ahmed, Sanaullah, et A. Gaba, « Speaker recognition using artificial neural networks », in *IEEE Students Conference, 2002. ISCON '02. Proceedings*, 2002, vol. 1, p. 99-102 vol.1.

[18] S. A. Firoz, S. A. Raji, et A. P. Babu, « Speaker and text dependent automatic emotion recognition from female speech by using artificial neural networks », in *World Congress on Nature Biologically Inspired Computing, 2009. NaBIC 2009*, 2009, p. 1411-1413.
[19] N. S. Dey, R. Mohanty, et K. L. Chugh, « Speech and Speaker Recognition System Using Artificial Neural Networks and Hidden Markov Model », in *2012 International Conference on Communication Systems and Network Technologies (CSNT)*, 2012, p. 311-315.
[20] Y. Lei, N. Scheffer, L. Ferrer, et M. McLaren, « A novel scheme for speaker recognition using a phonetically-aware deep neural network », in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, p. 1695-1699.
[21] Y. Liu, P. Karanasou, et T. Hain, « An investigation into speaker informed DNN front-end for LVCSR », in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, p. 4300-4304.
[22] G. Saon, H. Soltau, D. Nahamoo, et M. Picheny, « Speaker adaptation of neural network acoustic models using i-vectors », in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, p. 55-59.
[23] S. H. Ghalehjegh et R. C. Rose, « Deep bottleneck features for i-vector based text-independent speaker verification », in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, p. 555-560.
[24] M. McLaren, Y. Lei, et L. Ferrer, « Advances in deep neural network approaches to speaker recognition », in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, p. 4814-4818.
[25] « INTERSPEECH 2015 Abstract: Tian et al. » [En ligne]. Disponible sur: http://www.isca-speech.org/archive/interspeech_2015/i15_1151.html. [Consulté le: 10-mai-2016].
[26] « Application of convolutional neural networks to speaker recognition in noisy conditions ». [En ligne]. Disponible sur: https://www.researchgate.net/publication/290245535_Application_of_convolutional_neural_networks_to_speaker_recognition_in_noisy_conditions. [Consulté le: 10-mai-2016].
[27] P. Safari, O. Ghahabi, et J. Hernando, « Feature classification by means of deep belief networks for speaker recognition », in *Signal Processing Conference (EUSIPCO), 2015 23rd European*, 2015, p. 2117-2121.
[28] « TIMIT Acoustic-Phonetic Continuous Speech Corpus - Linguistic Data Consortium ». [En ligne]. Disponible sur: https://catalog.ldc.upenn.edu/LDC93S1. [Consulté le: 16-mai-2016].
[29] H. Melin, « Gandalf-a Swedish telephone speaker verification database », in *, Fourth International Conference on Spoken Language, 1996. ICSLP 96. Proceedings*, 1996, vol. 3, p. 1954-1957 vol.3.
[30] M. N. Ronald Cole, « The Cslu Speaker Recognition Corpus », 1999.
[31] M. Falcone et A. Gallo, « The ldquo;SIVA rdquo; speech database for speaker verification: description and evaluation », in *, Fourth International Conference on Spoken Language, 1996. ICSLP 96. Proceedings*, 1996, vol. 3, p. 1902-1905 vol.3.
[32] H. Melin, « Databases For Speaker Recognition: Activities In Cost250 Working Group 2 », 2000.
[33] L. Feng et L. K. Hansen, *A New Database for Speaker Recognition*. .
[34] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, et B. Scholkopf, « Support vector machines », *IEEE Intell. Syst. Their Appl.*, vol. 13, nº 4, p. 18-28, juill. 1998.

[35] J. Mercer, « Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations », *Philos. Trans. R. Soc. Lond. Math. Phys. Eng. Sci.*, vol. 209, nº 441-458, p. 415-446, janv. 1909.
[36] « 1.4. Support Vector Machines — scikit-learn 0.17.1 documentation ». [En ligne]. Disponible sur: http://scikit-learn.org/stable/modules/svm.html. [Consulté le: 19-mai-2016].