

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE SAAD DAHLAB DE BLIDA

FACULTÉ DES SCIENCES DE L'INGENIEUR

Département d'Aéronautique de Blida



MEMOIRE DE FIN D'ETUDE

**Pour l'obtention d'un diplôme
d'ingénieur d'Etat en Aéronautique**

Spécialité : Navigation Aérienne

Option : Installation

THEME

*Utilisation Des Réseaux De Neurone
Pour La Reconnaissance
Automatique De La Parole*

Encadré par :

Mr Mouhamed RAHMOUNI

Mr Abdel halim BENOURED

Présenté par ::

Mme Nesrine BOUSSOUAT

Promotion : 2010/2011

Remerciements



Après aimer et aider, remercier est assurément le troisième le plus beau verbe dans toutes les langues.

Ce travail est l'aboutissement d'un projet personnel, depuis longtemps attendu, né d'une volonté de voir mon rêve vif en réalité.

*Tout d'abord, je remercie **ELLEH** de m'avoir attribué assez le courage, la volonté et la patience pour arriver à terme de ce travail.*

***MES PARENTS.** Merci. Que puis-je dire d'autres Merci, parce que vous êtes toujours là; pour tout. Parce que je sais que vous m'aimez et que vous vous foutez éperdument de ce mémoire. Et dans ce monde, c'est tout ce que nous pouvons demander à nos parents. Nous avons grandi avec des histoires de pirates et d'îles aux Trésors*

*Tout en commençant par mon promoteur Mr **Rahmouni Mouhamed** pour m'a suivi et pour l'aide et ainsi pour sa gentillesse.*

*Je tiens également à remercier chaleureusement mon promoteur Mr **Abd el halim BENOUARED** pour avoir accepté de diriger ce mémoire, ainsi pour ses conseils précieux, ses orientations qui m'ont été de grande importance.*

*Je tiens à exprimer mes plus vifs remerciements à **Mr. Smaïne MUGFULATI** pour son aide particulière ainsi pour sa gentillesse, ses conseils précieux ses directives m'ont permis d'acquiescer le maximum de savoir. Qu'il trouve ici l'expression de ma profonde gratitude.*

*Je tiens aussi à remercier chaleureusement mon **président** de jurée et mes **examineurs** et également à mes **enseignants** de l'IAB.*

*Il m'est agréable de remercier chaleureusement **Mon Mari** et tous les membres de ma famille pour leurs sacrifices, leur affection et leur compréhension.*

Enfin un profond respect à tous ceux qui ont contribué de près ou de loin dans la réalisation de travail.

Dédicaces



Ce modeste travail n'est que le fruit d'un travail d'une volonté qui s'agrandissait de plus en plus .Ce but tant espéré n'a pu se réaliser sans la présence de tous ceux qui m'ont aidé, soutenu, encouragé et surtout supporté épreuve dans les moments les plus difficiles de cette épreuve .pour cela ,je dédie ce travail à toutes les personnes qui me sont chères, en particulier :

Mes chers parents

*A la fleur de ma vie; ma sœur **LOLA**
(allah yarhamha)*

A mes très chers beaux parents

A mon marie qui m'a beaucoup encouragé

*A mes adorable sœurs **MAMI** et **MINA**
et belles sœurs **Kamilia**, **Sihem** ,**Chahra** ,**Ahlem** ,**Achwak** et **Amina***

Sans oublié mes frères et mon beaux frère

*Ma tante **Samia**, et Oncle **Miloud***

*Et bien sur à Mes nièces **Lola** et **Doudou**,
mes neveux **Haitem**, **Aymen** et **Iade***

*Et une dédicace spéciale pour mon **BEBE** qui m'a donné beaucoup de courage
A toutes les personnes que j'ai rencontrées : **Khadija**, **Imene** ,**Dhrifa**, **Meriem**,
Hadjer et tous le groupe de l'**INSTALLATIONS** sans exception avec les quelle
j'ai partagé les meilleurs moments et toute les personnes que j'ai oubliées de leur
dédie ce modeste travail.*

Résumés

ملخص

على متن الطائرة، الكلمة تسمح للطيار أن تتوفر لديه وسيلة إضافية للتواصل مع الآلة، والتعرف التي تكون نموذج معطيات الدخول MFCC الاتوماتيكي على الكلمة يحتاج الي معالجة الاشارة و استخراج المعاملات في الشبكة العصبية المصطنعة للتعلم، و لكن للحصول على كفاءة عالية تضاهي الخبراء نستخدم الشبكة العصبية المصطنعة

Résumé

A bord des avions, la parole permet au pilote d'avoir à sa disposition un moyen supplémentaire d'interaction avec la machine, la reconnaissance automatique de la parole nécessite un traitement du signal et la paramétrisation pour extraire les coefficients MFCC (coefficients cepstraux) qui seront le patron d'entrée pour l'apprentissage d'un réseau neurones, afin de fournir une performance comparable à l'expert humain on a utilisé un réseau de neurone.

Abstract

On the aircraft board, the speech allows the pilot to have an additional and available means of interaction with the machine, the automatic speech recognition requires a signal processing and parameterization to extract MFCC coefficients that will be the pattern input for training a neural network, but to provide a performance comparable to human expert, a neurone network is used.

SOMMAIRE

RESUMES.....	4
LISTES DES FIGURES ET TABLEAUX.....	5
INTRODUCTION.....	7
Reconnaissance automatique de la parole (RAP)	7
Classification	7
Difficultés de la RAP	7
Les méthodes de la RAP	8
La reconnaissance globale	8
La reconnaissance analytique.....	9
Les approches de la RAP	9
La comparaison dynamique	9
Les modèles markoviens	9
Les modèles connexionnistes.....	10
Les systèmes hybrides	10
CHAPITRE I TRAITEMENT DE LA PAROLE	11
I.1-Introduction	11
I.2 Etude Phonétique	12
I.2.1-production de la parole	12
I.2.2 Audition	12
I.3 Le pré-traitement de la parole.....	14
I.3.1 Acquisition	14
I.3.2 Préaccentuation	15
I.3.3-Fenêtrage	16
I.3.3.1 Fenêtre rectangulaire	16
I.3.3.2 Fenêtre de Hamming	16
I.3.3.3 Fenêtre de Hanning	17
I.4 Méthodes d'analyse d'un signal vocal	17
I.4.1-Analyse spectrale	17
I.4.1.1-La transformation de Fourier	17
I.4.1.2 Le banc de filtres (vocodeur à canaux).....	18
I.4.2Analyse temporelle	19
I.4.2.1.Energie totale	19
I.4.2.2.La densité de passage par zéro (DPZ).....	19
I.4.3 Analyse basée sur la modélisation de la parole	19
I.4.4 Analyse homomorphique (Cepstrale).....	20
I.4.4.1Calcul du cepstre complexe	20
I.4.4.2 Ambiguïté de la phase	22

I.4.4.3 Définition du cepstre réel	23
I.4.4.4-Coefficients MFCC	25
I.4.4.5 Coefficients LFCC	25
CHAPITRE II LES RESEAUX DE NEURONES	26
II.1 Introduction	26
.....	
II.1.1 Fondement biologique	26
II.2 Modélisation du problème.....	28
II.2.1 Neurone formel	28
II.2.1.1 Constitution d'un réseau de neurone formel	29
II.2.1.1.1 La fonction d'entrée $f()$	29
II.2.1.1.2-fonction d'activation	30
II.3 Les réseaux de neurones formels	31
II.3.1 Les réseaux non bouclés	32
II.3.2 Les réseaux bouclés	32
II.3.3- Structure d'interconnexion.....	32
II.4 Analogie avec le modele biologique.....	34
II.5 Les réseaux de neurones artificiels	34
II.5.1. Les réseaux de neurones statiques	34
II.5.1.1 Les réseaux monocouches	35
II.5.1.2 Les réseaux multicouches.....	35
II.5.2 Les réseaux de neurones dynamiques	36
II.6 Apprentissage	36
II.6.1 Définition.....	36
II.6.2 Protocoles d'apprentissages.....	36
II.6.3 Les type d'apprentissage	36
II.6.3.1 Apprentissage non supervisé	36
II.6.3.2 Apprentissage supervisé.....	37
II.6.3.3 Apprentissage renforcé	39
II.6.4 Règles d'apprentissage	39
II.6.4.1 La loi de Hebb, un exemple d'apprentissage non supervisé.....	39
II.6.4.2 La règle d'apprentissage du Perceptron un exemple d'apprentissage supervisé	40
II.6.5 Algorithme du retro propagation.....	42
II.6.6 Algorithme de rétro propagation du gradient	46
II.7 Type de réseaux de neurones.....	48
II.7.1 L'Adaline.....	48
II.7.2 Le Perceptron multicouche.....	49
II.8 Conclusion.....	50

CHAPITRE V RESULTATS DES SIMULATION.....	51
III.1 Introduction	51
III.2 Présentation des applications	51
III.3 Présentation de l'interface des applications	52
III.3.1 La première phase	52
III.3.2 La deuxième phase	55
III.4 Simulation des résultats	57
III.4.1 Simulation de l'application 1: Energie, DPZ, La valeur moyenne :.....	57
III.4.3 Simulation de l'application 3 : DSP.....	58
III.4.2 Simulation de l'application 2 : FFT	58
III.4.4 Simulation de l'application 4 : préaccentuation.....	59
III.4.5 Simulation de l'application de reconnaissance	60
III.6 Conclusion.....	62
CONCLUSION GENERALE.....	63
BIBLIOGRAPHIE.....	64

Liste des figures et tableaux

INTRODUCTION :

Figure 1 Eléments intervenants dans la reconnaissance	09
---	----

CHAPITRE I :

Fig I.1 L'appareil phonatoire	12
Fig I.2 Le système auditif	13
Fig I.3 Le champ auditif humain.	13
Fig I.4 Acquisition	15
Fig I.5 La Préaccentuation	16
Fig I.6 - Traitement par transformé de Fourier	18
Fig I.7 - Analyse par vocodeur à canaux	18
Fig I.8- Modèle de production de la parole	20
Fig I.9-Modèle source-filtre	20
Fig I.10 Calcul du cepstre	22
Fig. I.11 calcul des coefficients cepstraux	23
Fig I.12 calcul des coefficients MFCC	25

CHAPITRE II :

Fig II.1 Neurone biologique	28
Fig II.2 Neurone formel	29
Table II.1 Les fonctions d'activations	31
Fig II.3- Réseau multicouche	32
Fig II.4 Réseau à connexions locales	33
Fig II.5- Réseau à connexions récurrentes	33
Fig II.6- Réseau à connexion complète	33
Table II.2 Compatibilité entre le neurone et un model artificiel	34
Fig II.7 Exemple le réseau multicouche	35
Fig II.8 Apprentissage non supervisé	37
Fig II.9. Surface d'erreur avec un minima local	38
Table II.3 La loi de Hebb	39
Fig II.10. Partition de l'espace d'entrée de dimension 2 réalisée par un Perceptron	42
Fig II.11 Réseau de neurones multicouches	43
Fig II.12 Organigramme d'algorithme d'entraînement par rétro propagation des erreurs.	48
Fig II.13 Architecture d'un PMC à trois couches	49

Chapitre III

Fig III.1 : L'interface principale de l'analyse de la parole	52
Fig III.2 l'interface de l'analyse temporelle d'un signal	52
Fig III.3 l'interface de FFT	53
Fig III.4 L'interface de DSP	54
Fig III.5 Interface de préaccentuation	54
Fig III.6 Interface de reconnaissance	55
FigIII.7 : L'interface de l'apprentissage	56
Fig III.8 Schéma explicatif de fonctionnement de réseau de neurone	56
Fig.III.9: signal $S(f)$ du mot « TEST »	57
Tab III.1 Tableau des paramètres de mot test	58
Fig III.10 : La FFT pour le mot « TEST »	58
Fig III.11: DSP du mot « TEST »	59
Fig III.12 : signal pré accentué de mot « Test »	59
Fig III.13 : le spectrogramme de mot « Test »	60
Fig III.14 : Test de reconnaissance du mot « test »	61
Fig III.15 : Exemple de mot n'est pas reconnu	62

Reconnaissance automatique de la parole (RAP) :

Le but ultime poursuivi en reconnaissance de la parole est la communication en langage naturel avec une machine (ordinateur, machine, robot,...).

Depuis près de cinquante ans, la reconnaissance de la parole par machine a fait l'objet d'un très grand effort de recherche, mais les performances des systèmes réalisées n'ont pu égaler, encore, celles de l'être humain, notamment dans les conditions réalistes de fonctionnement tel que la parole spontanée ; présence de bruit ambiant, etc...

Reconnaître et comprendre la parole demeurent parmi les grands défis de l'informatique. Les problèmes à résoudre sont considérables et de natures pluridisciplinaires notamment le traitement du signal, l'intelligence artificielle, la reconnaissance des formes, la phonétique, la linguistique et les neurosciences intervenant à des degrés divers dans les solutions.

Classification :

Les systèmes de reconnaissance peuvent être classés par ordre de difficulté croissante :

- Reconnaissance des mots isolés appartenant à un vocabulaire limité (moins de 100 mots) ;le système est adapté à un locuteur donné :système unilocuteur.
- Mêmes spécifications mais le vocabulaire est beaucoup plus étendu (quelques milliers de mots)
- Mêmes spécifications mais le système est indépendant du locuteur : système multilocuteur
- Reconnaissance des mots enchainés : les mots appartiennent à un vocabulaire très limité (les dix chiffres par exemple) mais ils sont prononcés sans pause dans un ordre quelconque.
- Reconnaissance de phrases courtes basées sur un vocabulaire limité : le système est unilocuteur
- Mêmes spécifications mais le système est multilocuteur.
- Reconnaissance de la parole continue prononcée par un seul locuteur, ou par locuteur quelconque

Difficultés de la RAP :

Le signal de la parole possède des caractéristiques qui compliquent son interprétation et augmentent le nombre de données à traiter.

Il présente un caractère redondant, c'est à dire qu'il renferme plusieurs types d'informations (les sons, la syntaxe et la sémantique de la phrase, l'identité du locuteur et son état émotionnel). Si cette redondance lui confère une bonne résistance au bruit, elle oblige à extraire du signal les informations pertinentes, en essayant de ne pas trop les dégrader.

Le signal est très variable selon le locuteur, c'est la variabilité interlocuteur (timbres différents, différences morphologiques homme ou femme). Mais également pour un même locuteur, on parle alors de variabilité intralocuteur, due à l'état émotionnel, la voix chantée, parlée, chuchotée. Il s'ajoute aussi les variabilités dues au milieu (le bruit perturbe la prise de son et augmente la variabilité intra locuteur) et à l'acquisition du signal.

Le signal est continu, c'est à dire que lorsqu'on écoute parler une personne, on perçoit une suite de mots, alors que l'analyse du signal vocal ne permet de déceler aucun séparateur. Le même problème de segmentation se retrouve à l'intérieur du mot. Celui-ci est perçu comme une suite de sons élémentaires (les phonéticiens trouveront le même nombre de phonèmes dans une phrase) que l'analyse ne permet pas d'isoler en segments distincts du signal acoustique. Le signal de la parole est évolutif.

Il y a également le phénomène de coarticulation. C'est l'effet contextuel que produit un phonème sur ses voisins. Il est provoqué par le fait que, lors de la prononciation d'un phonème, l'appareil articulaire se prépare pour la production du suivant.

Ces caractéristiques compliquent la tâche d'un système de RAP qui doit être capable de « décider qu'un [a] prononcé par un adulte masculin est plus proche d'un [a] prononcé par un enfant, dans un mot différent, dans un environnement différent et avec un autre microphone, que d'un [o] prononcé dans la même phrase par le même adulte masculin ».

Les méthodes de la RAP :

Un système de reconnaissance de la parole reçoit en entrée un signal acoustique et délivre en sortie une suite de symboles discrets.

Il existe deux méthodes différentes :

- **La reconnaissance globale :**

L'unité de base considérée est le mot donc elle évite la segmentation. Tous les mots prononcés sont supposés être séparés par des silences de quelques dixièmes de secondes. Les images acoustiques des mots sont isolées les unes des autres à partir de la courbe d'énergie du signal vocal. Après stockage des mots dans un dictionnaire, la comparaison choisit le mot le plus proche.

- **La reconnaissance analytique :**

Cette approche considère, par contre, comme unité : les phonèmes, les syllabes, les diphtongues, les allophones, etc. Le problème majeur est le phénomène de coarticulation (les phonèmes ne sont pas isolés mais liés les uns aux autres), la réalisation acoustique d'un phonème change avec celui qui le

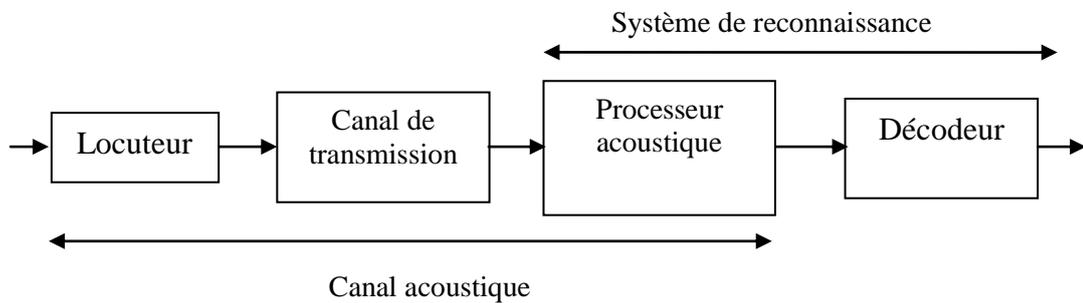


Fig 1 Elements intervenants dans la reconnaissance.

Précède et qui le suit. Cette tâche concerne des informations acoustiques, phonétiques et linguistiques. Cette approche est utilisée dans la reconnaissance de la parole continue.

Les approches de la RAP :

- **La comparaison dynamique :**

Elle permet d'intégrer des séquences de mots ou de phonèmes par l'alignement temporel. Les deux systèmes de comparaison dynamique les plus utilisés sont :

La programmation dynamique (ou en anglais DTW) : Il s'agit de mettre en correspondance deux formes afin d'obtenir une coïncidence optimale au sens d'un critère défini en fonction du domaine étudié (spectre dans la parole).

La DTW, Dynamic time warping, consiste à effectuer une normalisation temporelle au cours de la phase de comparaison entre un mot test et les mots références ; déterminer le début et la fin des formes acoustiques. La DTW comprime les mots dans le temps pour les faire coïncider à un modèle fixe.

- **Les modèles markoviens :**

Chaque mot est représenté par un modèle de MARKOV caché. Le critère d'apprentissage le plus utilisé est celui du maximum de vraisemblance. On cherche à déterminer les paramètres des modèles de façon à minimiser la probabilité de générer l'ensemble des mots ou phonèmes du

corpus d'apprentissage. Ce modèle autorise l'élision et l'ajout de plusieurs trames ; il peut donc coder l'aspect temporel d'une forme et la variabilité de la parole.

- **Les modèles connexionnistes :**

La modélisation par les réseaux de neurones permet de prendre en compte la variabilité de la parole. La classification des formes est effectuée en utilisant différentes sources d'information de manière auto-organisée. Ce mécanisme qui permet de combiner les connaissances de diverses origines est représenté par les coefficients de pondération du réseau. Cette approche permet donc de passer de l'extraction des caractéristiques du signal à la reconnaissance phonétique en faisant interagir différents indices acoustiques sans avoir à expliciter la stratégie de contrôle à adopter pour combiner au mieux plusieurs attributs phonétiques. Ces modèles seront détaillés dans ce qui suivra.

- **Les systèmes hybrides :**

Les systèmes hybrides connexionnistes et markoviens sont certainement une voie de recherche prometteuse. D'un point de vue théorique, de nombreux travaux de recherches sont effectivement encore nécessaires pour améliorer les algorithmes d'apprentissage des réseaux et les structures actuelles. Enfin, d'un point de vue expérimental, une validation de cette approche sur une autre base de données devrait permettre d'avoir plus de certitudes quant à l'apport des réseaux et des méthodes hybrides dans le domaine de la reconnaissance de la parole.

CHAPITRE 1

TRAITEMENT DE LA PAROLE

I.1-Introduction

La parole est la manière naturelle et, en conséquence, la forme la plus commune de communication humaine, elle se distingue des autres sons par des caractéristiques acoustiques ayant leurs origines dans les mécanismes de production. Les sons de parole sont produits soit par les vibrations des cordes vocales (voisement), soit par l'écoulement turbulent de l'air dans le conduit vocal, soit lors du relâchement d'une occlusion de ce conduit (bruit).

La parole est un signal réel, continu, d'énergie finie et non stationnaire. Sa structure est complexe et variable avec le temps

Deux limitations fondamentales méritent d'être prises en compte : les limitations du système auditif et celles du système vocal chez l'être humain.

Le système auditif humain est surtout sensible dans une gamme de fréquence située entre 800 Hz à 8.000 Hz; les limites extrêmes sont respectivement 20 et 20.000 Hz.

Par contre, le système vocal est encore plus limité, en résumé, pour des sons Vocaliques à des fréquences au-dessus de 4 kHz, les hautes fréquences sont plus de 40 dB en dessous du sommet du spectre [1].

I.2 Etude Phonétique :

I.2.1-production de la parole

L'appareil respiratoire fournit l'énergie nécessaire à la production de sons, en poussant de l'air à travers la trachée-artère. Au sommet de celle-ci se trouve le *larynx* ou la pression de l'air est modulée avant d'être appliquée au conduit vocal composé des cavités pharyngienne et buccale pour la plupart des sons. Lorsque la *luette* est en position basse, la cavité nasale vient s'y ajouter en dérivation.

L'air y passe librement pendant la respiration et la voix chuchotée, ainsi que pendant la phonation des sons non-voisés (ou *sourds*). Les sons voisés (ou *sonores*) résultent au contraire d'une vibration périodique des cordes vocales. Notons pour terminer le rôle prépondérant de la langue dans le processus phonatoire, elle détermine la hauteur du pharynx, le *lieu d'articulation*, ainsi que l'*aperture*, écartement des organes au point d'articulation ; elle permet au conduit vocal d'avoir une géométrie et un volume extrêmement variable [9].

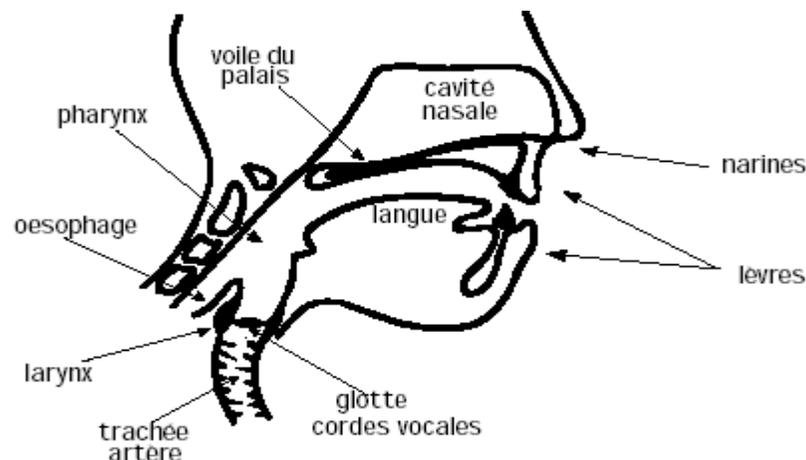


Fig I.1 : L'appareil phonatoire

I.2.2 Audition

Les ondes sonores sont recueillies par l'appareil auditif, ce qui provoque les sensations auditives. Ces ondes de pression sont analysées dans l'**oreille interne** qui envoie au cerveau l'influx nerveux qui en résulte. [7]

L'appareil auditif comprend l'**oreille externe**, l'**oreille moyenne**, et l'**oreille Interne**.

Le mécanisme de l'oreille interne (**marteau, étrier, enclume**) permet une adaptation d'impédance entre l'air et le milieu liquide de l'oreille interne. Les vibrations de l'étrier sont transmises au liquide

de la **cochlée**. Celle-ci contient la **membrane basilaire** qui transforme les vibrations mécaniques en impulsions nerveuses.

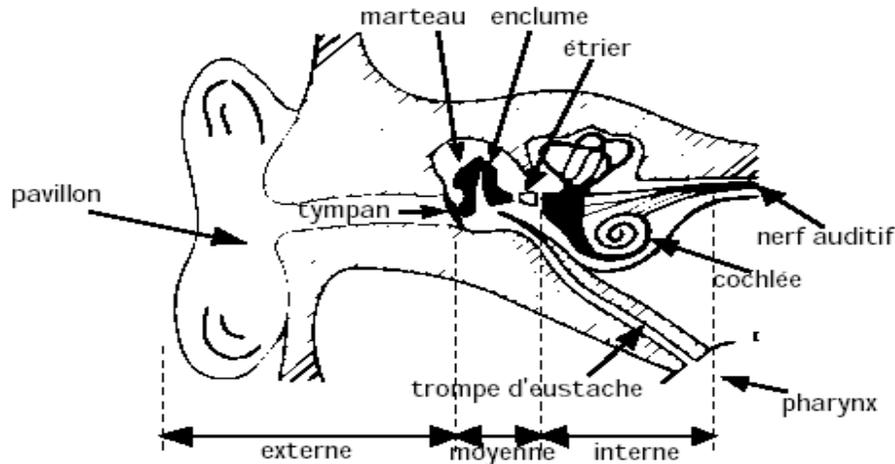


Fig I.2 : Le système auditif

Ainsi, l'oreille ne répond pas également à toutes les fréquences. La figure 1.3 présente le champ auditif humain, délimité par la courbe de **seuil de l'audition** et celle du **seuil de la douleur**. Sa limite supérieure en fréquence (16000 Hz, variable selon les individus) fixe la fréquence d'échantillonnage maximale utile pour un signal auditif (32000 Hz).

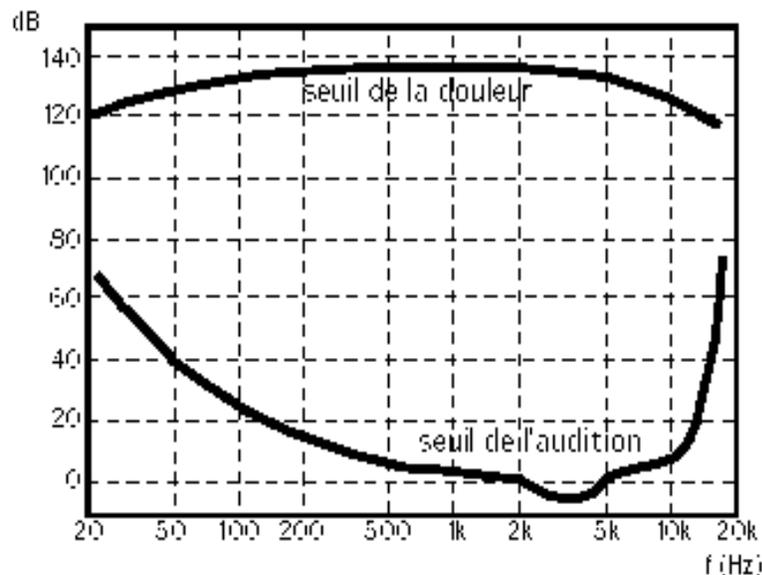


Fig I.3 : Le champ auditif humain.

I.3 Le pré-traitement de la parole :

Le traitement de la parole est aujourd'hui une composante fondamentale des sciences de l'ingénieur. Située au croisement du traitement du signal numérique et du traitement du langage (c'est-à-dire du traitement de données symboliques), cette discipline scientifique a connu depuis les années 60 une expansion fulgurante, liée au développement des moyens et des techniques de télécommunications.

De nature redondante et très complexe, la parole nécessite une représentation et un traitement particuliers, dans le but de minimiser sa redondance d'une part, et de l'adapter au milieu des machines d'autre part [2].

I.3.1 Acquisition :

Le traitement de la parole suppose toujours en premier lieu une analyse du signal vocal converti au préalable en signal électrique par un microphone ; Puisque les ordinateurs ne peuvent pas manipuler des sources analogiques, on doit convertir les signaux au format numérique avec un convertisseur A/N.

De la sorte, on va pouvoir travailler sur une représentation spectrale du signal, décomposant ses différentes fréquences avec leurs amplitudes et leurs harmoniques, aboutissant à des « traits » qu'on appelle formants du signal.

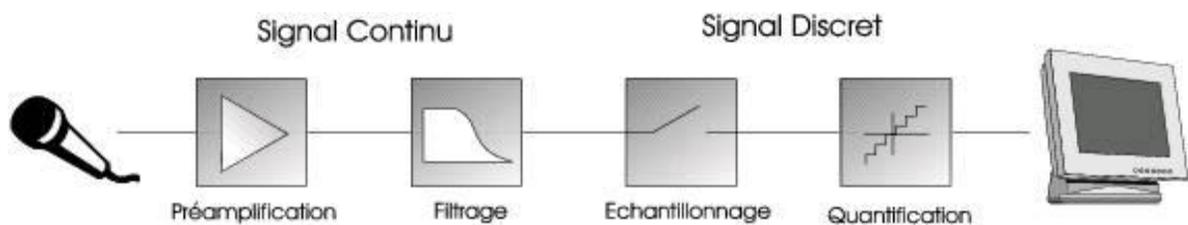
La raison d'une analyse numérique est qu'elle est plus aisée pour un traitement sophistiqué et qu'elle est beaucoup plus fiable. Le développement rapide des ordinateurs et des circuits intégrés en conjonction avec la croissance des communications numériques a encouragé l'application des techniques numériques au traitement du signal.

La conversion analogique/numérique consiste en l'échantillonnage, la quantification et le codage.

L'échantillonnage est le processus de représentation d'un signal continûment variable comme une séquence de valeurs. La quantification conduit à représenter approximativement chaque échantillon dans un ensemble finit de valeurs.

Le codage consiste à assigner un numéro réel à chaque valeur .Avant l'échantillonnage, un filtre passe-bas de fréquence de coupure égale à la moitié de la fréquence d'échantillonnage est inséré pour éviter l'effet dénommé «repliement» ou « aliasing » postulé par le théorème de Nyquist-Shannon ; Ce filtre est donc appelé filtre « anti-repliement » ou « antialiasing».

Il y a deux paramètres qui affectent la qualité du son. Le premier est la fréquence d'échantillonnage (sampling rate) : On la mesure en Hertz (Hz) et des valeurs typiques pour le son sont 4 kHz, 8 kHz, 11.025 kHz, 22.05 kHz, 44.1 kHz et 48 kHz. Cependant d'après le théorème de Shannon, il faut choisir cette fréquence un peu plus grande que la moitié de la bande intéressante parce que les composants électroniques ne sont pas idéaux et qu'il est donc impossible de réaliser un filtre parfait. Le deuxième paramètre qui affecte la qualité est la quantification en un nombre de bits fixé. Typiquement, ce nombre varie entre 8, 12, 14 ou 16 bits et détermine la dynamique et le rapport signal à bruit. Généralement, il s'agit d'une représentation uniforme mais une amélioration peut être obtenue avec des quantifications non linéaires [2].



FigI.4 : Acquisition

I.3.2 Préaccentuation :

En général, le signal vocal se caractérise par une perte de 6 dB/Octave, due à l'influence de la source d'excitation et au rayonnement des lèvres. Une perte de 6 dB/Octave veut dire que les hautes fréquences ont une énergie plus faible que celle des basses fréquences. Pour palier à cet inconvénient la préaccentuation permet d'égaliser les sons aigus avec les sons graves (figure I.5)

Le procédé le plus simple est d'appliquer un filtre de pré-accentuation donné par la fonction de transfert :

$$H(Z) = 1 - \mu * Z^{-1} \quad \text{où } 0 \leq \mu \leq 1.$$

Dans le domaine des signaux discrets (*échantillonnés*) $S(n)$ le problème consiste habituellement à calculer : $Y(n) = X(n) - \mu * X(n-1)$ pour $n \geq 0$.

Le facteur de préaccentuation μ , est pris entre 0.9 et 1 (souvent **0.95**). Comme conséquence, la préaccentuation introduit une légère distorsion spectrale.

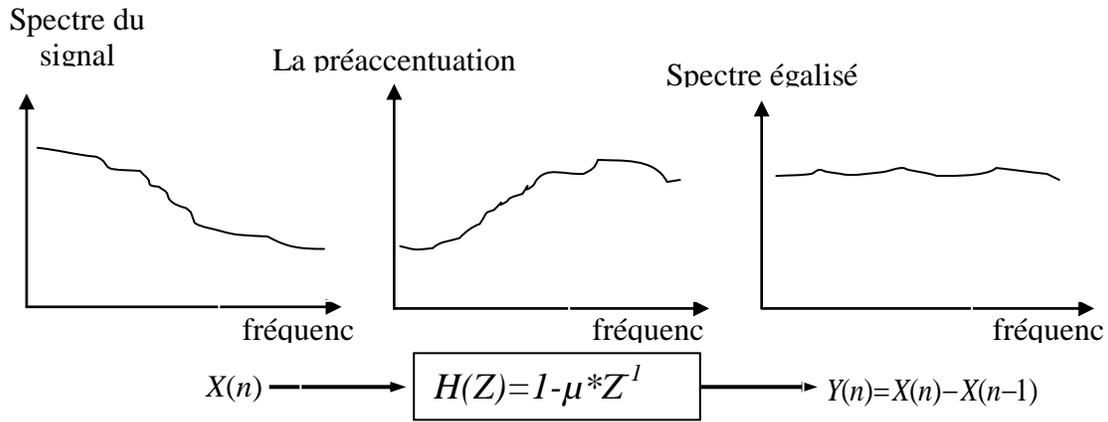


Fig I.5 : La Préaccentuation

I.3.3-Fenêtrage

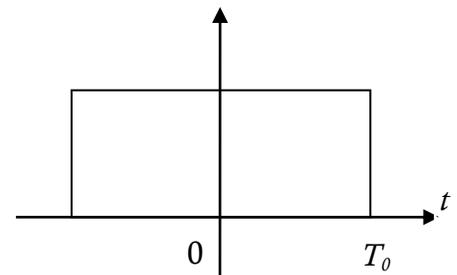
En réalité, le signal vocal n'est pas stationnaire et le conduit vocal se déforme de façon continue. Cependant l'objectif du fenêtrage est de subdiviser le signal à analyser en intervalles de temps assez courts (chacune de durée 30ms environ), sur lesquels il est pseudo stationnaire, c'est à dire *ces propriétés sont presque invariantes*(quasi- stationnaire) .

Il existe, dans la littérature, différents types de fenêtres ; on citera :

I.3.3.1 Fenêtre rectangulaire :

Il s'agit simplement de la partition pure et dure : tout si on est dans la tranche, rien dehors.

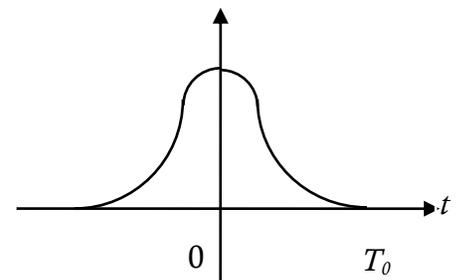
$$WR(k) = \begin{cases} 1 & \text{pour } |nT| \leq T_0 \\ 0 & \text{ailleurs} \end{cases}$$



I.3.3.2 Fenêtre de Hamming :

Couramment utilisée, elle est codifiée en tables pour éviter le calcul du cosinus.

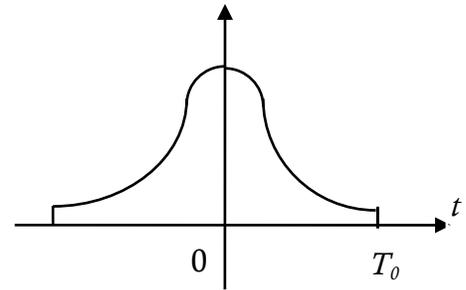
$$WH(k) = \begin{cases} a + (1 - a) \text{Cos}(2pk/N) & \text{pour } |nT| \leq T_0 \\ 0 & \text{partout ailleurs} \end{cases} \quad \text{avec } a = 0,54$$



I.3.3.3 Fenêtre de Hanning :

C'est un cas de la fenêtre de Hamming avec $a=1/2$.

$$Wh(k) = \begin{cases} 1/2(1 + \cos(2\pi k/N)) & \text{pour } |nT| \leq T_0 \\ 0 & \text{partout ailleurs} \end{cases}$$



Parmi ces fenêtres, la fenêtre de Hamming est la plus convenable à la parole, car elle entraîne un minimum de distorsion spectrale du signal de parole, par rapport aux autres fenêtres. (Atténuation du rapport du lobe principal au lobe secondaire est égale à -41 dB, c'est à dire que la concentration de l'énergie dans le lobe principal est égale à 99.96%).

$T_0=1/f_e$ est la période d'échantillonnage.

T est la moitié de longueur de la fenêtre.

I.4 Méthodes d'analyse d'un signal vocal :

Plusieurs approches ont été proposées pour l'analyse de la parole, ayant toutes pour but d'extraire le minimum d'information pouvant définir complètement le signal de parole. Parmi ces approches il y a celles qui agissent dans le domaine fréquentiel, celles qui agissent dans le temps, d'autres permettant d'avoir une analyse conjointe temps-fréquence, et finalement les méthodes basées sur la modélisation du système des phonations.

I.4.1-Analyse spectrale :

Elle est fondée sur une décomposition du signal sans connaissance à priori de sa structure fine ou de sa source. Dans ce domaine, on trouve principalement la TFD, et le vocodeur à canaux.

I.4.1.1-La transformation de Fourier :

Lorsqu'on veut analyser une entité complexe, une des procédures largement utilisées consiste à la décomposer en une somme d'entités plus simples. L'idée donc est d'exprimer le signal vocal par une combinaison linéaire discrète de fonctions élémentaires de forme simple ; c'est le cas de la transformée de Fourier discrète pour l'estimation du spectre.

La TFD est définie par :

$$S(n) = \sum_{k=0}^{N-1} s(k) \times e^{-j\pi \frac{nk}{N}}$$

$s(k)$ est un signal numérique.

N est la longueur du support du signal.

Pratiquement la TFD est évaluée par un algorithme rapide appelé FFT (Fast Fourier Transform). Elle s'opère sur des durées limitées du signal vocal, en prélevant les échantillons de parole à l'aide d'une fenêtre temporelle glissante, En général les fenêtres successives se recouvrent.

Ces fenêtres doivent avoir une largeur si l'on veut que la FFT ait un sens : En général, on prend 256 à 512 points, le recouvrement est par exemple la moitié soit 128 ou 256 respectivement.

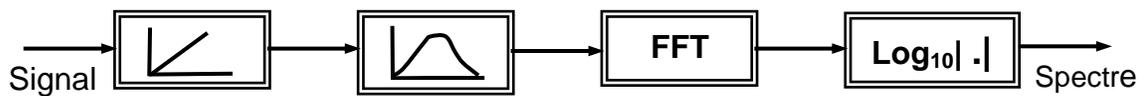


Fig I.6 : Traitement par transformé de Fourier

I.4.1.2 Le banc de filtres (vocodeur à canaux) :

L'estimation de l'enveloppe spectrale du signal peut se faire à l'aide des filtres en découpant la bande utile en sous-bandes(canaux), dans lesquelles on évalue l'intensité du signal. A la limite, lorsque les sous-bandes sont de largeur nulle, on tend vers une transformée de Fourier si les filtres sont idéaux. Le schéma classique d'un vocodeur à canaux est représenté sur la figure II.7.

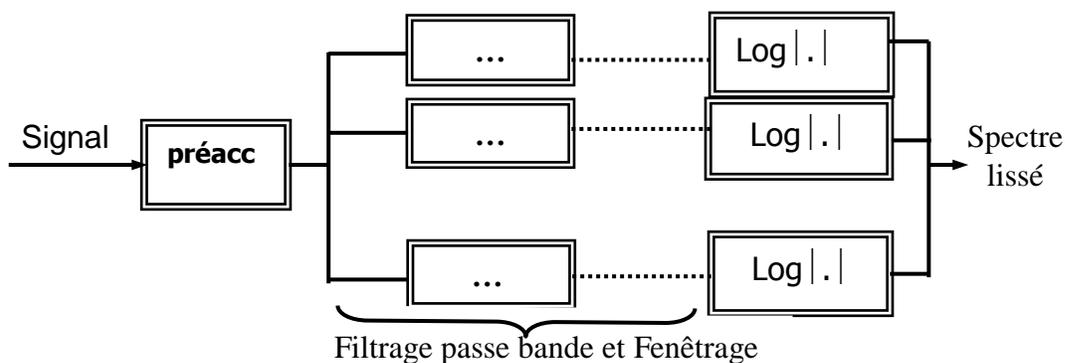


Fig II.7 :Analyse par vocodeur à canaux

I.4.2 Analyse temporelle :

I.4.2.1. Energie totale

Elle est évaluée par :
$$E = \frac{1}{N} \sum_{k=0}^{N-1} s^2(k)$$

N est la longueur de la fenêtre.

S(k) est la signal à traité.

Elle joue un rôle important pour délimiter les mots, et la localisation des voyelles qui sont caractérisées par une forte énergie par rapport aux autres phonèmes.

I.4.2.2. La densité de passage par zéro (DPZ) :

Ce critère consiste à compter le nombre de passage par zéro pour des trames identiques à celles définies précédemment en commençant du début de l'enregistrement, respectivement de la fin. Si ce nombre dépasse un certain seuil, calculé expérimentalement à partir d'échantillons de silence, alors on est en présence de parole ; début du mot, respectivement fin du mot.

$$DPZ = \frac{1}{2} \sum_{k=0}^{k-1} |sign(s(k+1)) - sign(s(k))|$$

Elle est utilisée pour distinguer le signal de parole du silence et un son voisé d'un son non voisé.

I.4.3 Analyse basée sur la modélisation de la parole :

Dans ce domaine on trouve essentiellement les systèmes de prédiction linéaire noté LPC et les systèmes de traitement homomorphe (cepstral) ; Elles sont applicables sur des modèles de parole basés sur une connaissance à priori de la production d'un signal de parole. Avant de présenter ces deux méthodes, il est utile d'examiner l'élaboration d'un modèle de parole [1].

I.4.3.1. Modèle de la parole :

La parole est formée par l'excitation du conduit vocal. On peut considérer le conduit vocal comme un système variant dans le temps, qui impose ses propriétés de transfert selon la forme d'excitation qui lui est appliquée. Si on admet que les excitations du conduit vocal sont relativement indépendantes, la production de la parole peut se résumer dans le modèle de la figure II.8. Dans ce modèle la source d'excitation est soit un générateur d'impulsions périodiques, avec une période dont

l'inverse est appelé le fondamental (pour les sons voisés), soit un générateur de bruit blanc (pour les sons non voisés).

Le conduit vocal peut être modélisé par un filtre numérique variant dans le temps, dont les coefficients varient d'une tranche de temps à l'autre.

Un contrôle de gain entre la source et le conduit ajoute une flexibilité supplémentaire pour le niveau sonore [10].

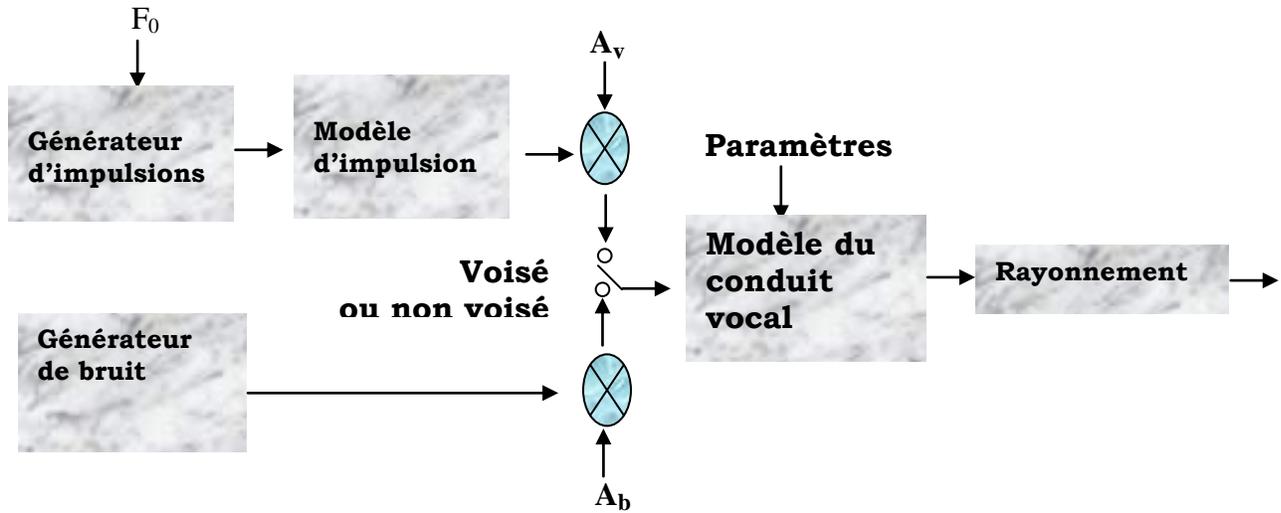


Fig I.8 : Modèle de production de la parole

I.4.4 Analyse homomorphique (Cepstrale):

I.4.4.1 Calcul du cepstre complexe :

Le défaut majeur de la FFT pour le calcul du spectre vocal, réside dans l'intermodulation source/conduit qui rend difficile la mesure des formants et du fondamental. L'analyse cepstrale est une méthode qui vise à séparer leurs contributions respectives par déconvolution. Pour cela on fait l'hypothèse que le signal vocal $x(n)$ est produit par un signal excitateur $g(n)$ (source glottique) traversant un système linéaire passif de réponse impulsionnelle $h(n)$ (conduit vocal).

D'après ces hypothèses, on aura le système suivant::

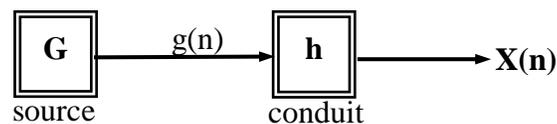


Fig I.9-Modèle source-filtre

Donc on peut écrire pour tout $n > 0$:

$$x(n) = g(n) * h(n)$$

Pour déconvoluer $x(n)$, c'est à dire pour retrouver les deux composantes $g(n)$ et $h(n)$, il faut se donner une classe de fonctions admissibles pour $g(n)$ (ou pour $x(n)$), ci on suppose que $g(n)$ est une séquence d'impulsions (périodique pour les sons voisés). Il est évident que l'ensemble de ces hypothèses est très limitatif : en toute rigueur cette analyse ne s'applique théoriquement qu'aux parties stables des sons périodiques (voyelles longues, par exemple), dans la pratique, cependant, cette méthode fournit des résultats acceptables sur l'ensemble du signal.

Pour déconvoluer plus aisément $x(n)$, il suffit de transposer le problème par homomorphisme dans un espace où l'opérateur de convolution « $*$ » correspond à un opérateur d'addition « $+$ ». Soit D_{*}^{+} cet homomorphisme.

D_{*}^{+} est un homomorphisme (application) qui applique l'espace vectoriel des signaux d'entrées muni de la loi « $*$ » (convolution), sur l'espace vectoriel

des signaux de sortie muni de la loi « $+$ » (addition), donc on est en face de la situation suivante :

$$x(n) = g(n) * h(n) \xrightarrow{D_{*}^{+}} \hat{x}(n) = \hat{g}(n) + \hat{h}(n)$$

Après séparation de $\hat{g}(n)$ et de $\hat{h}(n)$ si la transformation inverse D_{+}^{*} existe, on aura :

$$\hat{g}(n) \xrightarrow{D_{+}^{*}} g(n)$$

$$\hat{h}(n) \xrightarrow{D_{+}^{*}} h(n)$$

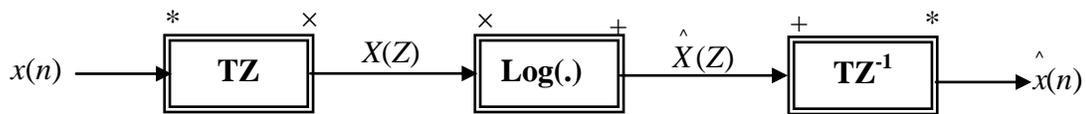
L'intérêt de la méthode réside dans le fait que $\hat{g}(n)$ et $\hat{h}(n)$ sont facilement séparables par un filtrage temporel est ceci grâce à l'hypothèse simplificatrice sur $g(n)$.

Les homomorphismes D_{*}^{+} et D_{+}^{*} sont inverses l'un de l'autre, et se définissent par :

$$D_{*}^{+} = TZ(\) \circ \text{Log}(\) \circ TZ^{-1}(\)$$

$$D_{+}^{*} = TZ(\) \circ \text{Exp}(\) \circ TZ^{-1}(\)$$

Ce qui donne le système schématisé dans la figure II.10

Fig I.10 : Calcul du cepstre **complexe**

Où :

- TZ est la transformée en Z (TZ⁻¹ sa transformée inverse).
- La fonction log est utilisée pour le passage du domaine de la loi « . » (La multiplication) au domaine de la loi « + » (l'addition), cette fonction n'est valable que pour les signaux positifs, toutefois, étant donné que la majorité des signaux courants sont bipolaires (positifs et négatifs), donc il faut faire appel à fonction log complexe. Soit :

$$X(Z) = |X(Z)| \times \exp[j \text{Arg}(X(Z))]$$

donc :

$$\hat{X}(Z) = \log [X(Z)] = \log |X(Z)| + j \text{Arg} [X(Z)]$$

- Exp. doit être aussi la fonction exponentielle complexe.

D'après le schéma de la figure II.10 on a :

$$X(Z) = TZ [x(n)]$$

$$\hat{X}(Z) = \log [X(Z)] \text{ (la fonction log est complexe)}$$

$$\hat{x}(n) = TZ^{-1} [X(Z)]$$

Tout ceci peut être résumé par la notation suivante :

$$\hat{x}(n) = D_{\#} [x(n)]$$

Le signal $\hat{x}(n)$ est appelé cepstre complexe associé au signal $x(n)$.

I.4.4.2 Ambiguïté de la phase :

Le problème qui se pose ici est que $\text{Arg} [X(Z)]$ n'est définie qu'à 2π près (la valeur principale), c'est à dire que l'on peut ajouter un multiple entier de 2π à la partie imaginaire du log complexe sans changer le résultat.

Ceci montre que l'homomorphisme tel qu'il est défini n'est pas une transformation biunivoque. Pour contourner ce problème, on a introduit la notion du cepstre réel.

I.4.4.3 Définition du cepstre réel :

La difficulté du logarithme complexe (à cause de la phase) peut être levée dans le cas de la parole (où l'on ne s'intéresse que rarement à l'information de phase) en prenant un log module ($\log | \cdot |$), ce qui garanti inversibilité sans calcul particulier de la phase.

Soit DM_{*}^{+} cet homomorphisme et DM_{*}^{-} son inverse :

$$DM_{*}^{+} = TZ(\cdot) \circ \log | \cdot | \circ TZ^{-1}(\cdot)$$

$$DM_{*}^{-} = TZ(\cdot) \circ \exp | \cdot | \circ TZ^{-1}(\cdot)$$

Les coefficients du cepstre réel sont définis par :

$$C_n = DM_{*}^{+} [x(n)]$$

Ces coefficients sont réels, ils conservent le spectre d'amplitude, par contre l'information de phase est perdue.

En pratique, on peut remplacer avantageusement la transformée en Z par une transformation de Fourier rapide (FFT), celle ci possède les même propriétés de linéarité que la transformée en Z. (figure I.11)



Fig I.11: calcul des coefficients cepstraux réels

Dans ce cas les coefficients \tilde{C}_n sont donnés par :

$$\tilde{C}_n = \frac{1}{N} \sum_{k=0}^{N-1} \log | X(k) | e^{jk1 2\pi / N}$$

Sous certaines conditions, on peut admettre que $\tilde{C}_n \cong C_n$ donc on peut dire que les \tilde{C}_n sont les coefficients cepstraux approchés prenant leurs valeurs dans un domaine pseudo temporel appelé domaine QUEFRENTIEL.

I.4.4.3.1 Application au signal vocal :

Si $x(n)$ correspond à un son voisé, on peut écrire :

$$x(n) = [p(n) * h(n)] \bullet w(n).$$

Où :

- $p(n)$ est un train d'impulsions de période p_0 ; p_0 étant l'inverse de la fréquence du fondamental

$$p(n) = \sum_k \delta(n - kp_0)$$

- $h(n)$ est la réponse impulsionnelle (elle met en évidence la contribution de l'onde glottique, le conduit vocal, et les lèvres).
- $w(n)$ est la fonction fenêtre.

Or la procédure de déconvolution décrite antérieurement exige que le signal observé soit bien sûr un produit de convolution. C'est la raison pour laquelle on fait souvent l'hypothèse que la fenêtre $w(n)$ recouvre un nombre suffisant M de périodes du fondamental. Dans ce cas, sa variation est faible sur la durée effective de la réponse impulsionnelle ce qui permet d'écrire :

$$x(n) \cong [p(n) * w(n)] \bullet h(n)$$

Soit :

$$P_w(n) = p(n) \bullet w(n) = \sum_{k=0}^{M-1} w(kp_0) \delta(n - kp_0)$$

Il s'ensuit que

$$P(e^{i\theta}) = W(e^{jp_0\theta}).$$

et donc

$$\hat{p}_w(n) = \hat{w}(n/p_0)$$

A cause de l'ambiguïté de la phase, on a intérêt à considérer le cepstre réel, qui vaut :

$$\tilde{C}(n) = \tilde{p}_w(n) + \tilde{h}(n)$$

où :

- $\tilde{p}_n(n)$ est une séquence d'impulsions séparée de p_0 échantillons.
- $\tilde{h}(n)$ décroît rapidement (en $1/n$) avec n , et devient rapidement négligeable du moins pour $n \geq p_0$

Dans ces conditions, on peut admettre que les premiers coefficients contiennent essentiellement la contribution du conduit vocal et que les pics périodiques ($\tilde{p}_w(n)$) reflètent les impulsions de la source.

On peut séparer la contribution de $\tilde{p}(n)$ par un filtrage temporel en utilisant tout simplement une fenêtre.

Soit $F(n)$ cette fenêtre telle que :

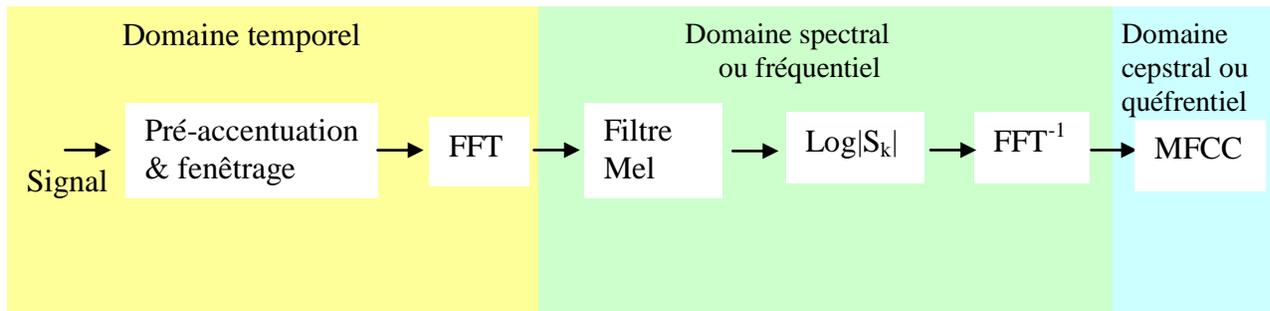
$$F(n) = \begin{cases} 1 & n < p_0 \\ 0 & \text{autrement} \end{cases}$$

Et finalement il ne reste que $h(n)$, définissant le filtre numérique qui modélise le conduit vocal.

I.4.4.4-Coefficients MFCC :(Mel-scaled Frequency Cepstral Coefficients)

Les paramètres MFCC sont des coefficients cepstraux obtenus à partir des énergies d'un banc de filtre en échelle de fréquence Mel. Il s'agit en fait d'un calcul classique des coefficients cepstraux auquel on a rajouté, avant le logarithme un filtre de Mel. Ces résultats sont intéressants, car le calcul d'une dizaine de coefficients cepstraux est alors suffisante pour des expériences de RAP.

$MFCC_i = \sum_{k=1}^{20} Xk \cos \pi_i \frac{(k-0.5)}{20}$ avec $i=1,2,\dots,p$; 20 est le nombre de filtre et p est le nombre des coefficients.



FigI.12 calcul des coefficient MFCC

I.4.4.5 Coefficients LFCC :

Ces coefficients sont calculés avec la relation suivante :

$$LFCC_i = \sum_{k=1}^{20} Xk \cos \left(\pi_i \frac{k}{K} \right) \text{ avec } i=1,2,\dots,p, \text{ et } K \text{ est le nombre des coefficient de la TFD}$$

CHAPITRE II

LES RESEAUX DE NEURONES

II.1 Introduction :

Depuis la création de la terre, l'homme n'a pas cessé de poser des questions, sur le monde qu'il entoure, par exemple comment on peut faire la différence entre deux choses, comment on peut parler, calculer.... et quelle est la partie qui fait tout ce travail ? La réponse pour toutes ces questions : c'est le cerveau, ce dernier est un dispositif très complexe, il peut réaliser des millions d'opération en une seconde, mais la question que se pose : pourrait –il existe un jour des machines ou des circuits électriques qui peuvent remplacer notre cerveau ?

Depuis quelques années, des chercheurs de part le monde entier relèvent ce défi, or en 1943 les deux américains de Chicago, le neurophysiologue WARREN MCCULLOCH et le logicien WALTER PITTS en inventent le premier neurone formel qui portera leurs noms (neurone de MCCULLOCH-PITTS) et qui sert à calculer certaines fonctions logiques. Quelques années plus tard en 1949, HEBB souligna l'importance du couplage synoptique dans les processus d'apprentissages, cette dernière repose sur le fait que si deux neurones sont connectes entre eux et activés en même temps, les connexions qui les relieent doivent être renforcées, dans le cas contraire, elles ne sont pas modifiées.

En 1958 FRANK ROSENBLAT décrivit le premier réseau de neurone artificiel qui conçoit le fameux < perceptron>. Ce modèle a réussi de calculer certaines fonctions logiques, et l'utilisé aussi dans le domaine de la reconnaissance des formes simples, et cela jusqu'à la fin des années 60(1959) car les deux mathématiciens américains MARVIN MINSKY et SEYMOUR PAPERTE ont prouvé les limitations du système, ils ont publié un livre dans le quel ils décrivent une série de problème de reconnaissance des formes que le perceptron ne peut pas résoudre, et

cette dernière est la cause pour stopper toutes les recherches sur la technologie des réseaux neuronaux. Mais au début des années 80 (1982) le chercheur JOHN HOPFIELD montre l'utilité de l'usage du réseau neurones formels. Grâce à tous ces chercheurs, on trouve aujourd'hui que l'utilisation des réseaux neuronaux est très vaste et qui peuvent être utilisés dans l'industrie, télécommunication, pour capter une image[4].

II.1.1 Fondement biologique :

Le fondement des réseaux des neurones provient d'une hypothèse proposée par de nombreux biologistes pour recréer le comportement intelligent du cerveau, et qu'il faut s'appuyer sur son architecture, pour tenter de l'imiter. De ce fait, la base du réseau des neurones est le simple neurone biologique lui-même. Il est alors perçu comme un mini processeur qui traite les informations qu'il reçoit pour produire une information unique.

Pour former le réseau des neurones, il y aura une architecture de connexion des minis processeurs fonctionnant en parallèle.

Le neurone (figure 1.1) est une cellule différenciée appartenant au système nerveux, qui assure le contrôle de toutes les fonctions de l'organisme. La structure du neurone comprend quatre parties importantes :

- Le corps cellulaire.
- Les dendrites.
- L'axone.
- La synapse.

Le corps cellulaire : C'est la partie qui traite les informations et de commande du neurone.

Les dendrites : Ce sont les récepteurs principaux des signaux qui parviennent des neurones adjacents.

L'axone : C'est un cylindre fixe qui se termine par de très fine arborisation. Il peut mesurer jusqu'à un mètre, son rôle est de transmettre l'activité neuronale générée par une cellule nerveuse aux fibres musculaires.

Les synapses : C'est la jonction de l'axone d'un neurone à dendrite d'un autre neurone.

Les neurones du cerveau sont interconnectés par le biais de prolongement ou dendrites.

Ils reçoivent des informations (les signaux) venant des neurones voisins à travers les synapses.

Les neurones transmettent ses propres influx neurone par une fibre ramifiée à son extrémité qu'on appelle « l'axone »

Les synapses servent à limiter plus au moins l'amplitude des signaux qui passent d'un neurone vers les autres.

Quand un neurone a reçu suffisamment d'énergie, il est excité, donc il envoie à son tour une décharge d'énergie vers les autres neurones qui lui sont reliés.

Le schéma d'un neurone biologique est représenté sur la figure (1.1).

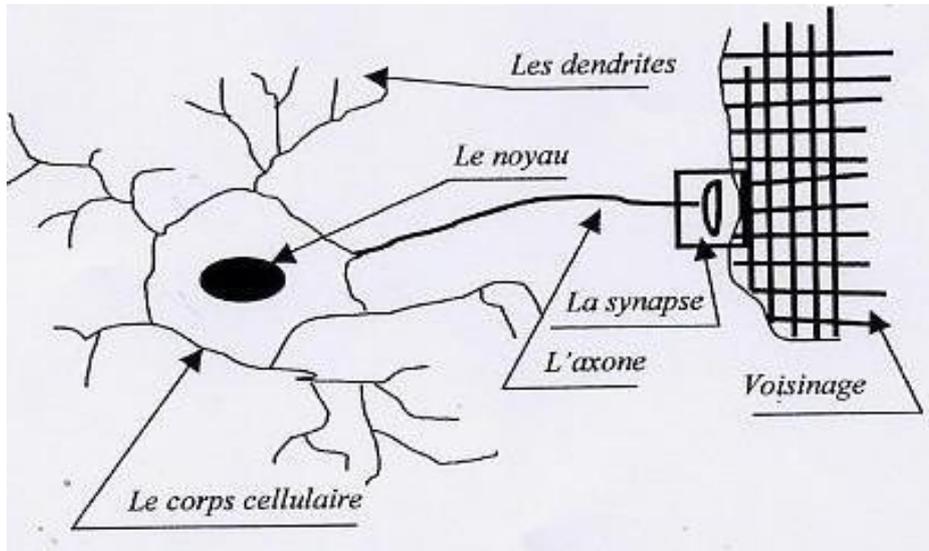


Fig II.1 Neurone biologique.

Mécanisme :

L'impulsion nerveuse est une manifestation de la communication intercellulaire. C'est une décharge électrique prenant naissance à l'extrémités de l'axone (segment initiale).

Cette décharge se propage le long de l'axone pour arriver au synapses ou se déroule des interactions fortement complexes qui transforment la décharge en un signal biochimique et cela par la libération de neurotransmetteurs (acétylcholine, adrénaline,...).

Ces neurotransmetteurs auront pour effets d'exciter les neurones qui les reçoivent ,ou au contraire d'inhiber l'influx nerveux [5].

II.2 Modélisation du problème :

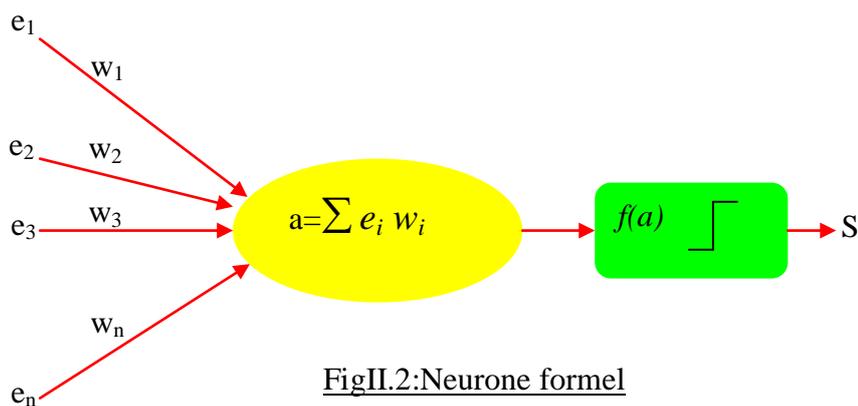
C'est à partir de l'hypothèse que le comportement intelligent émerge de la structure et du comportement des éléments de base du cerveau que les réseaux de neurones artificiels se sont développés. Les réseaux de neurones artificiels sont des modèles, à ce titre ils peuvent être décrits par leurs composants, leurs variables descriptives et les interactions des composants.

II.2.1 Neurone formel :

Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amont.

A chacune de ces entrées est associé un poids w , abréviation de weight (poids en anglais) représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique, qui se ramifie ensuite pour alimenter un nombre variable de neurones avals. A chaque connexion est associé un poids [5].

“Un neurone formel fait une somme pondérée des potentiels d’actions qui lui proviennent des autres neurones, puis s’active suivant la valeur de cette sommation pondérée. Si cette sommation dépasse un certain seuil, le neurone est activé et transmet une réponse dont la valeur est celle de son activation. Si le neurone n’est pas activé, il ne transmet rien. ”



FigII.2:Neurone formel

II.2.1.1 Constitution d’un réseau de neurone formel :

Le neurone formel est un automate caractérisé par un petit nombre de fonctions mathématique. Il traite un signal d’entrée recueilli à travers ses connexions entrantes pour fournir le signal de sortie calculé par la fonction de transfert. Son état est caractérisé par trois variables E, A, S calculées au moyen de trois fonctions successives : la fonction d’entrée $f()$, la fonction d’activation $g()$ et la fonction de sortie $h()$.

II.2.1.1.1 La fonction d’entrée $f()$:

Dans un réseau de n neurones chacun d’entre eux reçoit en général de 1 à n stimulations en provenance des autres cellules. Ces valeurs, binaires ($\{1, 0\}$ ou $\{1, 1-\}$) ou réelles, sont en général pondérées puis additionnées par la fonction $f()$:

$$E = f(x) = \sum_{j=1}^n W_j X_j \quad \dots\dots\dots (1)$$

W_j est la pondération appliquée par le neurone à l'entrée j . Si le neurone dont on calcule l'entrée est le neurone i , on note W_{ij} la pondération correspondant à l'arc reliant le neurone j au neurone i . On parle alors du poids de la connexion et l'équation (1) s'écrit comme suivant :

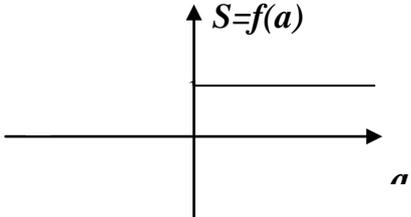
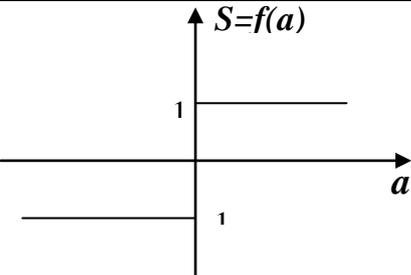
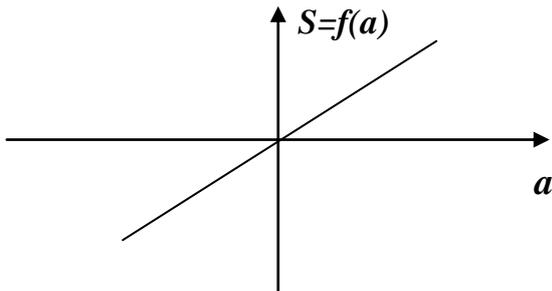
$$E = f(x) = \sum_{j=1}^n W_{ij} X_j$$

Certaines cellules utilisent des fonctions d'entrée différentes. Ainsi, par exemple, les unités Sigma Pi ($\Sigma\Pi$) réalisent des sous-produits pondérés entre groupes d'entrées avant d'additionner les résultats obtenus.

II.2.1.1.2-fonction d'activation :

Il est clair que la fonction d'activation joue un rôle très important dans le comportement du neurone. Elle retourne une valeur représentative de l'activation du neurone, cette fonction a comme paramètre la somme pondérée des entrées ainsi que le seuil d'activation.

La nature de cette fonction diffère selon le réseau. On en compte divers types, parmi elles :

<p><u>Fonction de Heaviside (seuil) :</u></p> $S = \begin{cases} 1 & \text{si } a \geq 0 \\ 0 & \text{si } a < 0 \end{cases}$	
<p><u>Fonction signe :</u></p> $S = \begin{cases} 1 & \text{si } a \geq 0 \\ -1 & \text{si } a < 0 \end{cases}$	
<p><u>Fonction linéaire :</u></p> $S = \lambda a$	

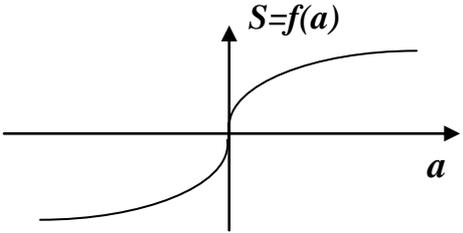
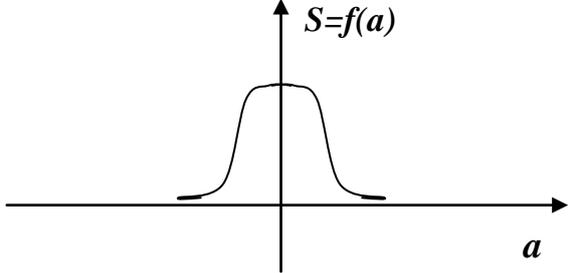
<p>Fonction sigmoïde :</p> $S = th(a)$	
<p>Fonction gaussienne :</p> $S = exp(-a^2/\sigma)$	

Table II.1 : Les fonctions d'activations

II.3 Les réseaux de neurones formels :

Nous pouvons dire qu'un réseau de neurones peut être défini comme un graphe orienté et pondéré.

Les neurones étant les nœuds, qui seront connectés par des liens appelés liens synaptiques ou synapses.

Ces liens synaptiques sont pondérés par des poids judicieusement choisis.

La propagation de l'activation à travers les liens synaptiques des neurones en aval vont influencer les autres neurones en amont, cette activation sera pondérée par le lien qu'elle prendra ainsi on appellera poids synaptique le poids de chaque liaison synaptique.

Ils sont décrits par la matrice des poids synaptiques :

$$\begin{pmatrix}
 W_{11} & W_{12} & \dots & \dots & \dots & W_{1m} \\
 W_{21} & W_{22} & \dots & \dots & \dots & W_{2m} \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & \dots & W_{ij} & \dots & \dots \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 \dots & \dots & \dots & \dots & \dots & \dots \\
 W_{n1} & W_{n2} & \dots & \dots & \dots & W_{nm}
 \end{pmatrix}$$

W_{ij} : poids de la liaison du neurone "j" vers le neurone "i".

L'activation totale du réseau est décrite par un vecteur d'activation :

$A = (a_1, a_2, \dots, a_n)$ a_i : activation du neurone "i" $i = 1, 2, 3, \dots, n$

L'architecture d'un réseau de neurone influence considérablement sur son comportement globale. On distingue deux grandes architectures de réseaux de neurones :

II.3.1 Les réseaux non bouclés :

Un réseau de neurone non bouclé, si son graphe ne possède pas de cycle. Dans tel réseau, l'information circule de l'entrée vers la sortie sans aucun retour. Un réseau de neurone non bouclé est dit acyclique. Citons l'exemple des réseaux multicouches

II.3.2 Les réseaux bouclés :

Un réseau est bouclé, si son graphe possède au moins un cycle. Le réseau bouclé fait ramener une ou plusieurs valeurs à l'entrée. Un réseau de neurone bouclé est donc un système dynamique, régi par des équations différentielles ; comme l'immense majorité des applications sont réalisées par des programmes d'ordinateurs, on se place dans le cadre des systèmes à temps discret, où les équations différentielles sont remplacées par des équations aux différences [4].

II.3.3- Structure d'interconnexion

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

- **Réseau multicouche** (au singulier) :

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche et les connexions ne se font qu'avec les neurones des couches avales.

Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie.

Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées [3].

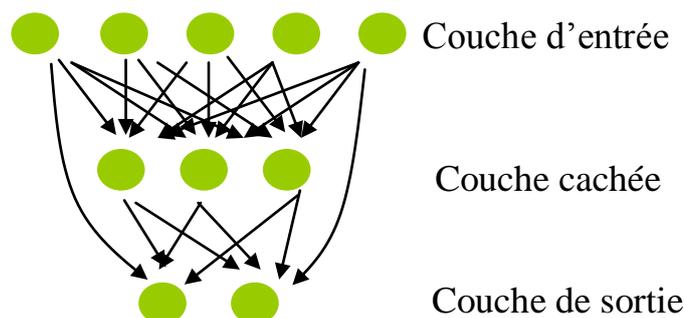


Fig II.3: Réseau multicouche

- **Réseau à connexions locales :**

Il s'agit d'une structure multicouche, mais qui à l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale (fig2.3). Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique [3].

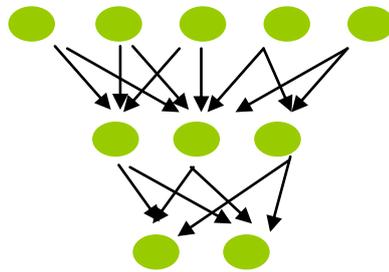


Fig II.4 Réseau à connexions locales

- **Réseau à connexions récurrentes :**

Les connexions récurrentes ramènent l'information en arrière par rapport au sens de propagation défini dans un réseau multicouche. Ces connexions sont le plus souvent locale

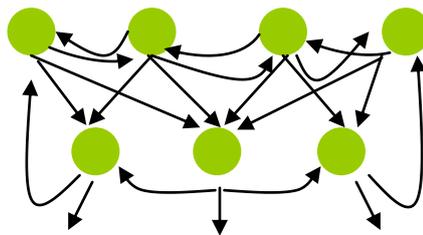


Fig II.5 Réseau à connexions récurrentes

- **Réseau à connexion complète :**

C'est la structure d'interconnexion la plus générale (fig2.5). Chaque neurone est connecté à tous les neurones du réseau.

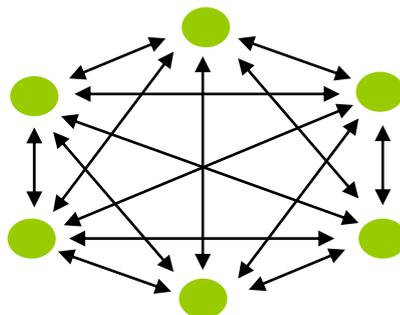


Fig II.6 Réseau à connexion complète

Il existe de nombreuses autres topologies possibles, mais elles n'ont pas eu à ce jour la notoriété des quelques unes que nous avons décrites ici.

II.4 Analogie avec le modèle biologique

Le tableau suivant résume la comptabilité entre les modèles neuronaux biologiques et artificiels.

Cerveau	→	réseau de neurone
Neurone	→	élément processeur
Dendrites	→	somme pondérée
Corps cellulaire	→	fonction d'activation
Axone	→	fonction de sortie
Synapse	→	poids

Table II.2 Compatibilité entre le neurone et un modèle artificiel.

II.5 Les réseaux de neurones artificiels :

Le réseau de neurone artificiel est constitué par un ensemble de neurones associés en couches, qui fonctionnent en parallèle. Chaque couche fait un traitement indépendant des autres et transmet le résultat de son analyse à la couche suivante jusqu'à la couche de sortie, en passant soit par aucune, une ou plusieurs couches intermédiaires dites cachées. Par exemple pour reconnaître une fonction mathématique, il faut une phase de comportement qui est appelée « la phase d'apprentissage ».

On distingue deux grandes catégories de réseaux neuronaux :

- Les réseaux neuronaux statiques.
- Les réseaux neuronaux dynamiques.

II.5.1. Les réseaux de neurones statiques :

Dans ce type de réseau, l'information circule dans un seul sens, de l'entrée vers la sortie, or chaque neurone d'une couche reçoit ses entrées à partir des neurones de la couche précédente. Ces réseaux peuvent être utilisés pour les problèmes de classification pour réaliser une transformation non linéaire.

$$U = G(x) \quad \text{ou} \quad x \in \mathbb{R}^n \quad U \in \mathbb{R}^m$$

On cite quelques réseaux de neurone statiques :

- Les réseaux monocouches.
- Les réseaux multicouches (MLP Multi Layer Perceptron).

II.5.1.1 Les réseaux monocouches :

Les réseaux monocouches sont constitués d'une couche d'entrée formée de plusieurs neurones connectés à une couche de connexion modifiable. Le mode de calcul est une sommation pondérée des activations des neurones d'entrées associées à chaque neurone de sortie.

II.5.1.2 Les réseaux multicouches

Apparus en 1985, les réseaux multicouches sont aujourd'hui les modèles les plus employés. Plusieurs couches de traitement leur permettent de réaliser des associations non linéaires entre l'entrée et la sortie.

On sait depuis les années soixante que les possibilités de traitement des réseaux multicouches sont supérieures à celle du Perceptron, cependant l'algorithme d'apprentissage manqué. Pour la couche de sortie, on peut appliquer l'apprentissage du Perceptron, mais comment modifier les poids pour les connexions qui ne sont pas en relation avec un neurone de sortie ?

Le problème est ramené à l'obtention d'une estimation de la valeur désirée pour chaque neurone de la couche cachée. Le rétro propagation de gradient est une solution à ce problème.

Cet algorithme a été proposé indépendamment par trois équipes en 1985, dont Y. le Cun. Des recherches bibliographiques ont montré qu'il s'agit en fait d'une redécouverte. Concluons que le Faire savoir est aussi important que le Savoir dans le domaine scientifique.

Le principe utilisé par la rétro propagation ("back propagation" en anglais) de gradient est la minimisation d'une fonction dépendante de l'erreur. Il s'agit d'une méthode générale, largement employée dans d'autres domaines tels que la physique. (Voir figure 1.3).

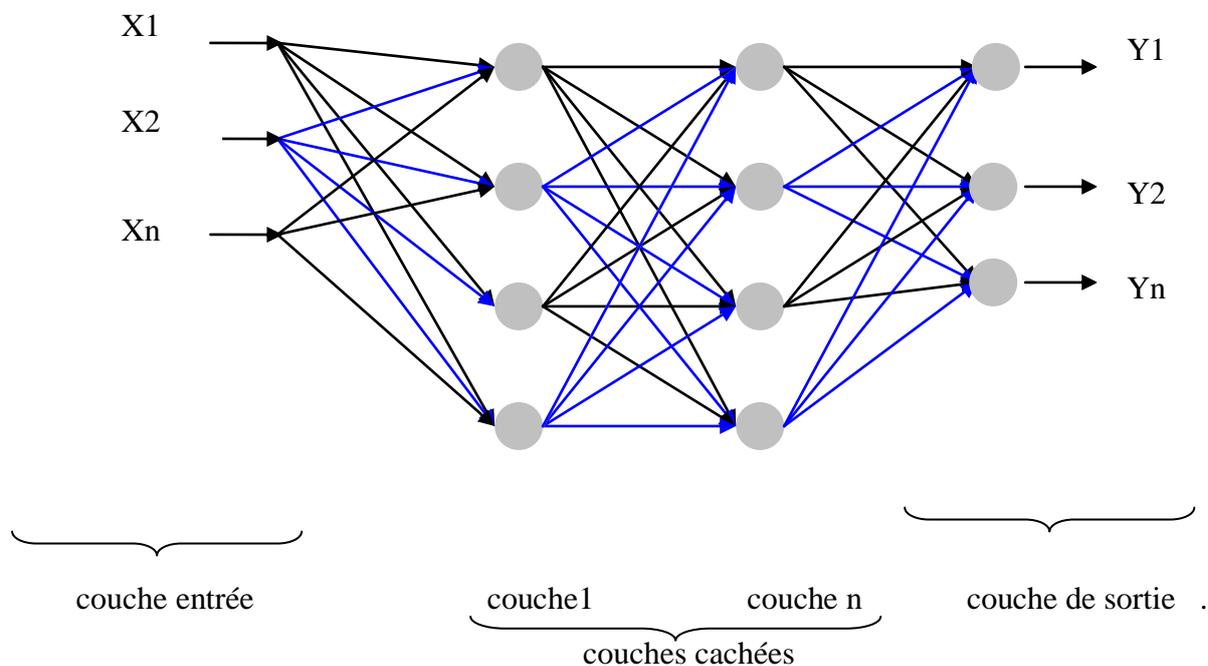


Fig II.7 Exemple le réseau multicouche.

II.5.2 Les réseaux de neurones dynamiques :

Les réseaux dynamiques, ou réseaux récurrents sont organisés tel que chaque neurone reçoit sur ses entrées une partie ou la totalité de l'état du réseau (sortie des autres neurones) en plus des informations interne. Pour les réseaux dynamiques, l'influence entre les neurones s'exerce dans les deux sens. L'état global du réseau dépend aussi de ses états précédents.

L'équation du neurone dans ce cas est décrite par les équations différentielle ou des équations à la différence. Ce type d'architecture de réseau est très important, car on l'utilise pour l'identification et la commande des systèmes dynamiques (avion, moteur, robot...).

II.6 Apprentissage :

II.6.1 Définition :

L'apprentissage est une phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement.

Dans les algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions [4].

II.6.2 Protocoles d'apprentissages:

Presque la totalité des réseaux de neurones ont en commun un même protocole d'apprentissage celui-ci comporte quatre étapes :

- **Etape 1 :** Initialisation des poids synaptiques avec des petites valeurs aléatoires.
- **Etape 2 :** Présentation du patron d'entrée et propagation de l'activation des neurones.
- **Etape 3 :** Calcul de l'erreur, dans le cas d'un apprentissage supervisé cette erreur dépend de la différence entre l'activation des neurones et le patron de référence
- **Etape 4 :** Calcul du vecteur de correction à partir des valeurs des erreurs ,avec lequel on effectue la correction des poids synaptiques.

Les étapes 2-3-4 sont répétées jusqu'à la fin de l'apprentissage.

II.6.3 Les type d'apprentissage :

Les techniques d'apprentissage se subdivisent en trois grandes familles :

II.6.3.1 Apprentissage non supervisé :

Dans un apprentissage non supervisé le réseau ne dispose pas d'un comportement de référence mais la modification des poids est fonction d'un comportement local.

On présente à l'entrée du réseau des exemples connus et le réseau s'organise lui-même. Comme exemple d'apprentissage non supervisé, la règle de HEBB qui s'applique aux connexions entre neurones. HEBB a proposé l'idée que le cerveau humain s'adapte à son

environnement en modifiant l'efficacité des connexions entre neurones. Elle est basée sur le concept que si 2 neurones connectés entre eux sont activés simultanément, alors il faut renforcer la connexion qui les relie. Dans le cas contraire, elle n'est pas modifiée.

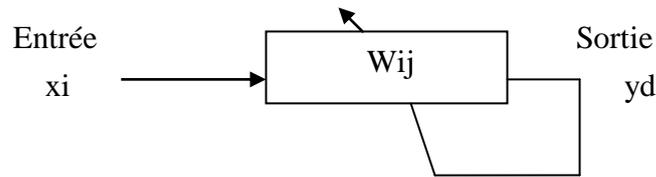


Fig II.8 Apprentissage non supervisé

II.6.3.2 Apprentissage supervisé :

Dans cette apprentissage nous supposons que chaque patron d'entrée lui est associée une sortie désirée.

Ainsi pour chaque propagation des activations lors de la présentation de l'entrée une sortie sera générée. Celle-ci sera comparée à la sortie attendue par la suite.

Cette comparaison déterminera l'erreur du réseau pour chaque entrée (patron).

$D(t) = (d_{i1}, d_{i2}, \dots, d_{is})$ $i_k \in S$: valeurs des sorties désirées.

$A(t) = (a_{i1}, a_{i2}, \dots, a_{is})$ $i_k \in S$: valeurs des sorties obtenues.

$E(t) = (e_{i1}, e_{i2}, \dots, e_{is})$ $i_k \in S$: erreur du réseau.

$e_{ik} = f_E(a_{ik}, d_{ik})$ $i_k \in S$

f_E : c'est la fonction d'erreur qui fait la comparaison (supervision).

Il est évident que si l'on veut obtenir un comportement adéquat du réseau il faut minimisée l'erreur du réseau.

Cela sera fait par approximation successive, tout en modifiant les valeurs des poids synaptiques du réseau.

Ainsi nous avons :

$$\Delta W = f_w(W, E, \eta). \text{ : Correction des poids.}$$

ΔW est fonction des poids W_{ij} , de l'erreur E ainsi que d'un paramètre η appelé pas d'apprentissage.

Les règles de l'apprentissage supervisé sont des descentes de gradient. Si l'erreur « E » a des solutions qui peuvent être évaluée par une valeur scalaire, nous pouvons définir une surface d'erreur ou chaque point représente une solution possible qui lui est associée une hauteur déterminée par la grandeur de son erreur.

Donc, la tâche à accomplir est de progresser de solution en solution tout en minimisant l'erreur a chaque pas.

Ceci revient à choisir une solution dans chaque étape dans la direction inverse du gradient de « E ».

Ainsi, la solution final aura la plus petite erreur (localement), donc un gradient tendant vers zéro.

Deux hypothèses doivent être faite sur la nature de la surface :

- **La continuité :**

La continuité de la surface doit être assuré, car dans le cas contraire la recherche ne peut aboutir à une solution.

Mais cette condition ne pose pas trop de problèmes en pratique.

- **Les minimums doivent être acceptables :**

Car la solution obtenue par la suite risque d'être loin de la solution optimale si nous nous trouvons dans un minimum local.

Cette dernière hypothèse est l'une des faiblesses les plus importantes de cette technique de recherche.

Vue la complexité de la surface d'erreur, elle peut comprendre des minimas locaux de l'erreur et cela bien au dessus de l'erreur optimale.

La vérification d'une solution ne peut être effectuée, car la forme de la surface de l'erreur est méconnaissable.

Divers solutions ad hoc ont été utilisées pour palier ses inconvénients parmi elles :

- Recherche stochastique dite de: «Monté-Carlo ».
- Modification des pas d'apprentissages, et cela lorsqu'on se trouve dans un minima locale, nous modifions la valeur du pas de façons à «ce débloquent »et à continuer l'apprentissage.

[3].

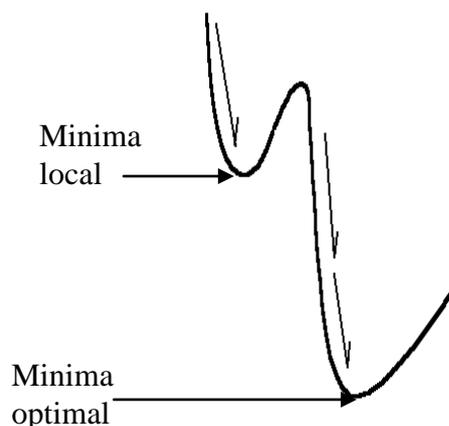


Fig II.9 Surface d'erreur avec un minima local

La descente stochastique :

Elle a été proposée par WINDROW et HOFF dans les années 60. Au lieu de minimiser l'erreur globale due à l'ensemble des vecteurs d'apprentissage, ils ont proposé de minimiser itérativement l'erreur due à chaque exemple d'apprentissage. Mathématiquement on cherche à minimiser itérativement chaque terme de la somme au lieu de minimiser la somme.

II.6.3.3 Apprentissage renforcé :

Ce type d'apprentissage est moins classique que les deux précédents, il a d'une part en commun avec l'apprentissage supervisé la présence d'un critère qui dirige l'évolution de l'apprentissage, et d'autre part, il ne nécessite que des entrées pour définir les sorties désirées comme dans le cas de non supervisé.

Le réseau de neurone génère une ou plusieurs valeurs de sorties basées sur la performance de calcul des valeurs d'entrées. Les résultats de ces calculs dépendent des valeurs des poids assignées à chaque entrée obtenue après la phase d'apprentissage.

II.6.4 Règles d'apprentissage :

II.6.4.1 La loi de Hebb, un exemple d'apprentissage non supervisé:

La loi de Hebb (1949) s'applique aux connexions entre neurones, Elle s'exprime de la façon suivante :

"Si 2 cellules sont activées en même temps alors la force de la connexion augmente".

La modification de poids dépend de la coactivation des neurones pré synaptique et post synaptique, ainsi que le montre la table III.2. x_i et x_j sont respectivement les valeurs d'activation des neurones i et j , Δw_{ij} (dérivée partielle du poids) correspond à la modification de poids réalisée.

x_i	x_j	Δw_{ij}
0	0	0
0	1	0
1	0	0
1	1	+

Table II.3 La loi de Hebb

La loi de Hebb peut être modélisée par les équations suivantes ($w(t+1)$ est le nouveau poids, $w_{ij}(t)$ l'ancien) :

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$$

$\Delta w_{ij}(t) = x_i \cdot x_j$ (la coactivité est modélisée comme le produit des deux valeurs d'activation)

L'algorithme d'apprentissage modifie de façon itérative (petit à petit) les poids pour adapter la réponse obtenue à la réponse désirée. Il s'agit en fait de modifier les poids lorsqu'il y a erreur seulement.

1/ Initialisation des poids et du seuil S à des valeurs (petites) choisies au hasard.

2/ Présentation d'une entrée $E_i = (e_1, \dots, e_n)$ de la base d'apprentissage.

3/ Calcul de la sortie obtenue x pour cette entrée :

$a = \Delta (w_i \cdot e_i) - S$ (la valeur de seuil est introduite ici dans le calcul de la somme pondérée)

$x = \text{signe}(a)$ (si $a > 0$ alors $x = +1$ sinon $a = 0$ alors $x = -1$)

4/ Si la sortie x est différente de la sortie désirée d_i pour cet exemple d'entrée E_i alors modification des poids (μ est une constante positive, qui spécifie le pas de modification des poids) :

$w_{ij}(t+1) = w_{ij}(t) + \mu \cdot (x_i \cdot x_j)$

5/ Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement (i.e. modification des poids), retour à l'étape 2.

II.6.4.2 La règle d'apprentissage du Perceptron un exemple d'apprentissage supervisé :

La règle de Hebb ne s'applique pas dans certain cas, bien qu'une solution existe. Un autre algorithme d'apprentissage a donc été proposé, qui tient compte de l'erreur observée en sortie.

L'algorithme d'apprentissage du Perceptron est semblable à celui utilisé pour la loi de Hebb. Les différences se situent au niveau de la modification des poids.

1/ Initialisation des poids et du seuil S à des valeurs (petites) choisies au hasard.

2/ Présentation d'une entrée $E_i = (e_1, \dots, e_n)$ de la base d'apprentissage.

3/ Calcul de la sortie obtenue x pour cette entrée :

$a = \Delta (w_i \cdot e_i) - S$

$x = \text{signe}(a)$ (si $a > 0$ alors $x = +1$ sinon $a = 0$ alors $x = -1$)

4/ Si la sortie x du Perceptron est différente de la sortie désirée dl pour cet exemple d'entrée Ei alors modification des poids (μ le pas de modification) :

$$w_i(t+1) = w_i(t) + \mu \cdot (dl - x) \cdot e_i$$

Rappel : $dl = +1$ si E est de la classe 1, $dl = -1$ si E est de la classe 2 et $(dl - x)$ est une estimation de l'erreur.

5/ Tant que tous les exemples de la base d'apprentissage ne sont pas traités correctement (i.e. modification des poids), retour à l'étape 2.

Exemple de fonctionnement de l'algorithme d'apprentissage du Perceptron:

Base d'exemples d'apprentissage :

e1	e2	d	(eX)
1	1	1	(1)
-1	1	-1	(2)
-1	-1	-1	(3)
1	-1	-1	(4)

1/ Conditions initiales : $w_1 = -0.2$, $w_2 = +0.1$, $S = 0$, ($\mu = +0.1$)

2/ $a(1) = -0.2 + 0.1 \cdot -0.2 = -0.3$

3/ $x(1) = -1$ (la sortie désirée $d(1) = +1$, d'où modification des poids)

4/ $w_1 = -0.2 + 0.1 \cdot (1 + 1) \cdot (+1) = 0$

$w_2 = +0.1 + 0.1 \cdot (1 + 1) \cdot (+1) = +0.3$

2/ $a(2) = +0.3 - 0.2 = +0.1$

3/ $x(2) = +1$ Faux

4/ $w_1 = 0 + 0.1 \cdot (-1 - 1) \cdot (-1) = +0.2$

$$w_2 = +0.3 + 0.1 \cdot (-1 - 1) \cdot (+1) = +0.1$$

$$2-3/ a(3) = -0.2 - 0.1 - 0.2 = -0.5 \text{ Ok}$$

$$2-3/ a(4) = +0.2 - 0.1 - 0.2 = -0.1 \text{ Ok}$$

$$2-3/ a(1) = +0.2 + 0.1 - 0.2 = +0.1 \text{ Ok}$$

$$2-3/ a(2) = -0.2 + 0.1 - 0.2 = -0.1 \text{ Ok}$$

5/ Tous les exemples de la base ont été correctement traités, l'apprentissage est terminé.

Le Perceptron réalise une partition de son espace d'entrée en 2 classes (1 et 2) selon la valeur de sa sortie (+1 ou -1). La séparation de ces deux zones est effectuée par un hyperplan (fig. 3). L'équation de la droite séparatrice est :

$$w_1 \cdot e_1 + w_2 \cdot e_2 - S = 0$$

se comportant comme un ET booléen. Les 4 exemples de la base d'apprentissage sont les 4 arêtes du carré. Les paramètres du Perceptron sont : $w_1 = 0.2$, $w_2 = 0.1$ et $S = -0.2$.

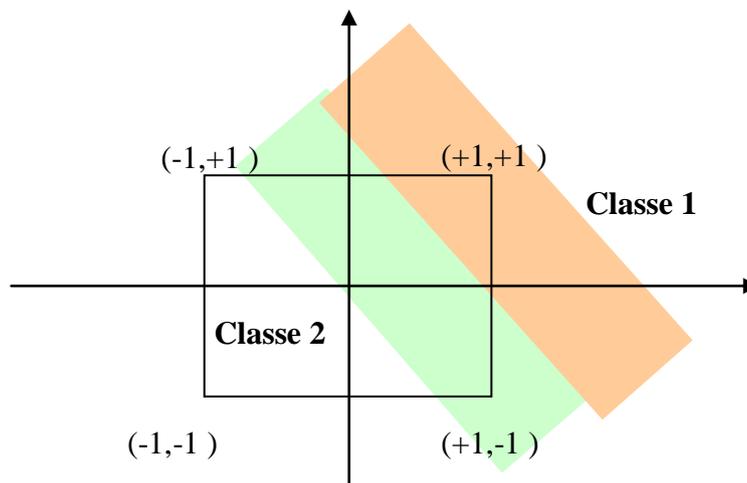


Fig II.10 :Partition de l'espace d'entrée de dimension 2 réalisée par un Perceptron

II.6.5 Algorithme du retro propagation :

L'algorithme de rétro propagation du gradient est utilisé dans le perceptron multicouche, la phase la plus importante où ce dernier est utilisé est bien évidemment l'apprentissage.

On présente au réseau des entrées et on lui demande de modifier sa pondération de telle sorte que l'on trouve la sortie correspondante. L'algorithme consiste dans un premier temps à propager vers l'avant les entrées jusqu'à obtenir une sortie calculée par le réseau.

La seconde étape compare la sortie calculée à la sortie réelle connue.

On modifie alors les poids de telle sorte qu'à la prochaine itération, l'erreur commise entre la sortie calculée et connue soit minimisée.

Malgré tout, il ne faut pas oublier que l'on a des couches cachées. On rétro propage alors l'erreur commise vers l'arrière jusqu'à la couche d'entrée tout en modifiant la pondération, on répète ce processus sur tous les exemples jusqu'à ce que l'on obtienne une erreur de sortie négligeable.

Exemple

Afin de voir un petit peu mieux ce qui se passe, on considère un réseau assez simple composé d'une couche d'entrée, une seule couche cachée et une couche de sortie.

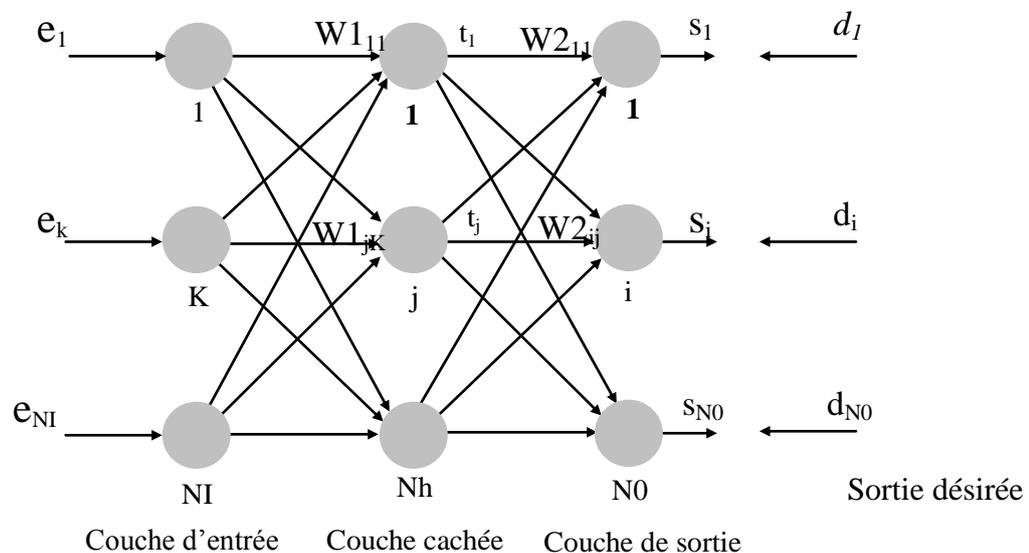


Fig II.11 Réseau de neurones multicouches

Notations :

i : Neurone de la couche de sortie .

j : Neurone de la couche cachée

k : Neurone de la couche d'entrée

N_I, N_h, N_O : dimension de la couche d'entrée, de la couche cachée et de la couche de sortie respectivement.

w_{ij} : représente le poids de la connexion entre le neurone i et le neurone j de la couche qui le précède.

s_j : la sortie du neurone j

t_j : l'entrée totale du neurone j

f : fonction d'activation de type sigmoïde.

e_k : valeur d'entrée du neurone k

s_i : valeur de la sortie du neurone i

d : valeur des sorties désirées.

Les étapes :

1 – On commence tout d'abord les calculs par une initialisation aléatoire des poids.

2 – On calcule les sorties du réseau, s_i : $i=1, N_0$, en fonction des entrées e_i : $i=1, N_1$ en utilisant les formules de calcul d'activation d'un neurone.

La fonction quadratique de l'erreur à pour expression (qui est un critère de performance)

$$E(w) = \frac{1}{2} \sum_{l=1}^{N_0} (s_l - d_l)^2 \quad \dots\dots\dots (3)$$

La règle de modification des poids est donnée par :

$$W_{ij}(k+1) = w_{ij}(k) - \eta \frac{\partial E}{\partial w_{ij}} \quad \dots\dots\dots (4)$$

Où η c'est le pas de descente (pas d'apprentissage), $0 < \eta < 1$

Pour Couche de sortie

Pour Neurone i

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial t_i} * \frac{\partial t_i}{\partial w_{ij}} \quad \dots\dots\dots (5)$$

$$\text{Avec : } t_i = \sum_{m=1}^{N_h} w_{im} s_m \quad \dots\dots\dots (6)$$

On calcule erreur du neurone i

$$\delta_i = \frac{\partial E}{\partial t_i} \quad \dots\dots\dots (7)$$

$$\delta_i = \frac{\partial E}{\partial t_i} = \frac{\partial E}{\partial s_i} * \frac{\partial s_i}{\partial t_i} = (s_i - d_i) * f' \quad \dots\dots\dots (8)$$

$$\frac{\partial t_i}{\partial w_{ij}} = \delta_i * s_j = f'(t_i) * (s_i - d_i) * s_j \quad \dots\dots\dots (9)$$

La valeur δ_i est proportionnelle à la différence de la valeur désirée et calculée et enfin à la dérivée de la fonction d'activation.

En effet, plus la pente est forte, plus on doit corriger.

On calcul l'erreur δ_i , $i=1, N_0$ selon l'équation (8) ainsi que les gradients selon (5)

On ajuste les poids vers les neurones de sortie selon la règle donnée par l'équation (4)

Couche cachée.

Pour Neurone j

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial t_j} * \frac{\partial t_j}{\partial w_{ij}} \quad \dots\dots\dots (10)$$

Avec

$$t_j = \sum_{m=1}^{N_I} w_{jm} e_m \quad \dots\dots\dots (11)$$

L'erreur du neurone j de la couche cachée :

$$\delta_j = \frac{\partial E}{\partial t_j} \quad \dots\dots\dots (12)$$

$$\delta_j = \frac{\partial E}{\partial t_j} = \frac{\partial E}{\partial s_j} * \frac{\partial s_j}{\partial t_j} = f'(t_j) * \frac{\partial E}{\partial s_j} \quad \dots\dots\dots (13)$$

$$\frac{\partial E}{\partial s_j} = \sum_{n=1}^{N_0} \frac{\partial E}{\partial t_n} * \frac{\partial t_n}{\partial s_j} = \sum_{n=1}^{N_0} \delta_n \frac{\partial t_n}{\partial s_j} \quad \dots\dots\dots (14)$$

$$\frac{\partial t_n}{\partial s_j} = \frac{\partial}{\partial s_j} \left(\sum_{m=1}^{N_h} w_{nm} s_m \right) = w_{nj} \quad \dots\dots\dots (15)$$

$$\frac{\partial E}{\partial s_j} = \sum_{n=1}^{N_0} \delta_n w_{nj} \quad \dots\dots\dots (16)$$

Donc :

$$\delta_j = \sum_{n=1}^{N_0} \delta_n w_{nj} f'(t_j) = f'(t_j) \sum_{n=1}^{N_0} \delta_n w_{nj} \dots\dots\dots (17)$$

δ_n : L'erreur propagée à partir du neurone n de la couche de sortie vers les neurones de la couche cachée

$$\frac{\partial t_i}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}} \left(\sum_{m=1}^{N_i} w_{jm} e_m \right) = e_k \dots\dots\dots (18)$$

Donc :

$$\frac{\partial E}{\partial w_{jk}} = \delta_j e_k = \sum_{n=1}^{N_0} \delta_n w_{nj} f'(t_j) e_k \dots\dots\dots (19)$$

L'erreur δ_j est proportionnelle à la dérivée de la fonction d'activation, car plus la pente est importante, plus il faut corriger. En plus, elle est proportionnelle à la somme pondérée des erreurs de sortie. Ainsi, le neurone de sortie ayant le plus gros poids et la plus grosse erreur interviendra le plus dans la correction.

On ajuste les poids selon la règle de correction (4) (on effectue les mêmes calculs pour les neurones de la couche cachée, $j=1, N_h$)

Après la phase d'apprentissage, toute la mémorisation des connaissances est gravée dans le réseau. C'est la phase la plus longue en calcul connexionniste.

II.6.6 Algorithme de rétro propagation du gradient

A partir de l'exemple présenté, la procédure générale d'apprentissage par rétro propagation du gradient s'effectuera suivant les étapes suivantes :

1. Initialisation des poids synaptiques et des seuils d'une manière aléatoire.
2. Initialisation d'un accumulateur d'erreur à zéro. $E=0$
3. Présentation du patron d'entrée et propagation d'activation.
4. Calcul de l'erreur de sortie pour le patron de référence en cours et ajustement des poids depuis la couche de sortie jusqu'à la couche d'entrée suivant :

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = -\eta \delta_i^n \delta_j^{n-1} \dots\dots\dots (20)$$

n : indice de la couche.

δ_i^n : L'erreur du neurone i de la couche (n) .

δ_j^{n-1} : La sortie du neurone j de la couche d'indice $(n - 1)$.

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij} \quad \dots\dots\dots (21)$$

5 .calcul de : $E = E + \sum_{j=1}^{N_0} (y_j^i - y_{d_j}^i)^2 \quad \dots\dots\dots (22)$

y_j^i : La sortie du neurone de la sortie j lors du passage de l'exemple i.

$y_{d_j}^i$: La sortie désirée pour l'exemple i au neurone j de la couche de sortie.

N_0 : Nombre de neurones dans la couche de sortie.

6. Répéter depuis 3 jusqu'au dernier patron de référence dans la base d'exemples.

D'apprentissage (corpus d'apprentissage).

7. Calcul de l'erreur moyenne (où l'erreur quadratique moyenne)

$$E_{moy} = \frac{E}{N} \quad \dots\dots\dots (23)$$

Où : N c'est le nombre de patron de référence dans le corpus d'apprentissage.

8. Si $E_{moy} = E_{moy_{accepté}}$ aller en 9 sinon en 2.

9. Sauvegarde des poids du réseau.

Les étapes de 2 jusqu'à 7 constituent une époque d'apprentissage.

Donc l'algorithme de rétro propagation du gradient est un outil performant et simple mais les études pratiques ont montré que son application appelle les remarques suivantes :

1. Il n'existe pas de théorème garantissant la convergence du réseau, et l'on se contentera le plus souvent d'une erreur faible. Donc la valeur initiale des poids influe sur la solution finale.
2. Le temps d'apprentissage est relativement long.
3. La structure du réseau est fixée.

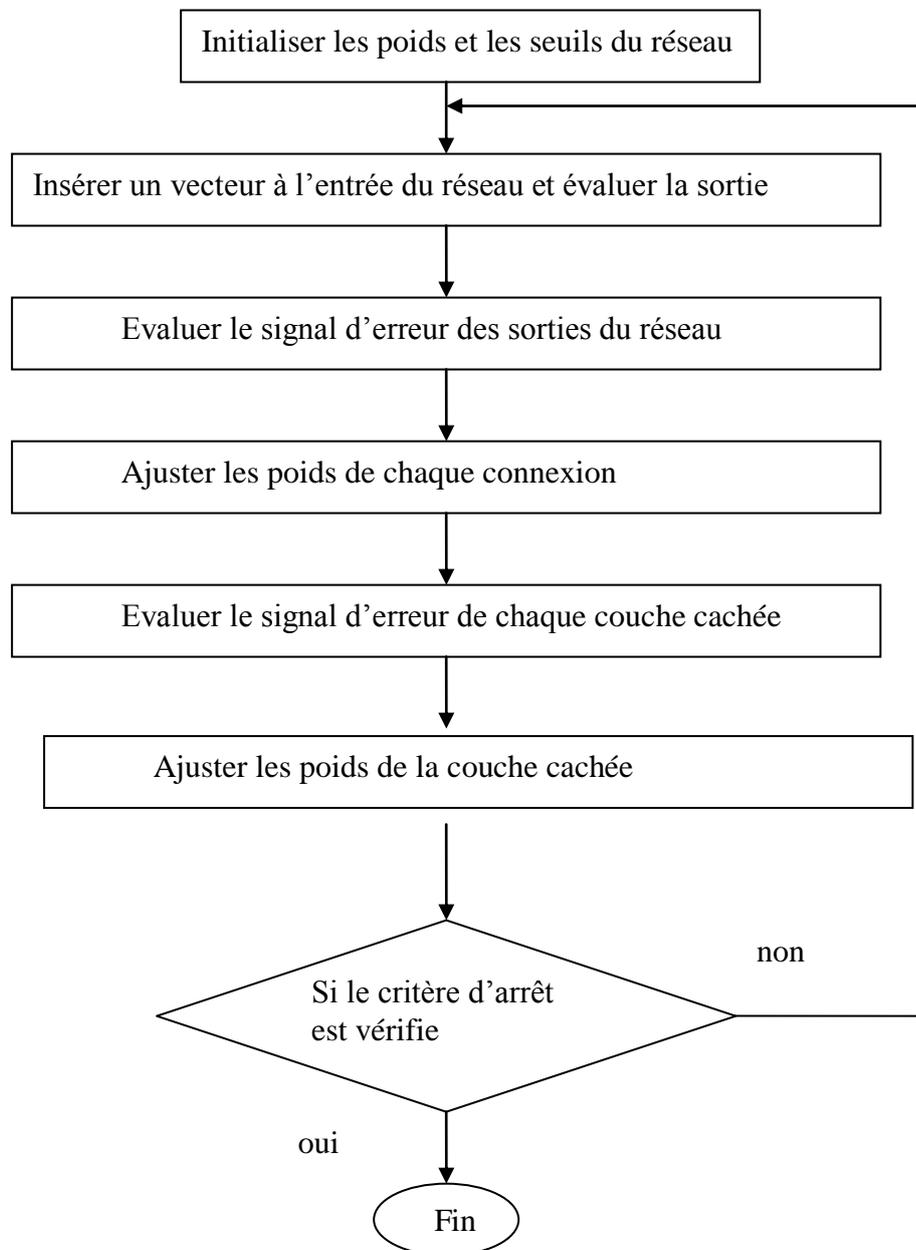


Fig II.12 Organigramme d'algorithme d'entraînement par rétro propagation des erreurs.

II.6.7 Type de réseaux de neurones :

Il existe une multitude de réseaux de neurones différents les uns des autres par leurs architectures et leurs méthodes d'apprentissages [3].

II.6.7.1 L'Adaline :

Peu de temps après que Rosenblatt proposa son perceptron, Windrow et Hoff proposèrent un autre modèle de neurone qu'ils baptisèrent l'Adaline (pour Adaptive Linear Neuron) en 1960.

Ce modèle trouva diverses applications comme composante dans les antennes adaptatives et les modems à haute vitesse. L'Adaline est un seul neurone à valeur d'activation continue et une fonction d'activation linéaire.

$$a_i = net_i = \sum W_{ij} a_j - \theta_i$$

a_i : Activation du neurone i .

W_{ij} : Poids du neurone j vers le neurone i .

θ_i : Seuil du neurone i .

a_j : Activation du neurone précédant j .

La règle d'apprentissage utilisée est la règle de "Widrow-Hoff" ou règle Delta :

$$\Delta W_{ij} = \eta a_j (d_i - a_i).$$

Ou :

η : Est le pas d'apprentissage.

a_j : Activation du neurone j .

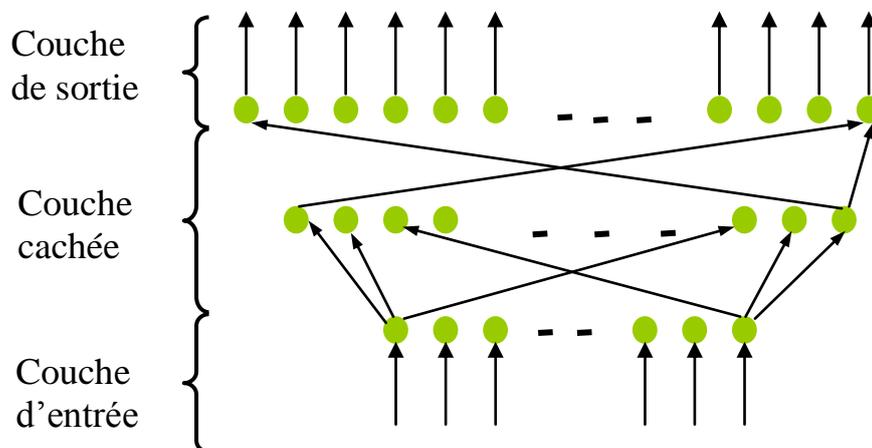
d_i : Activation attendue du neurone i .

a_i : Activation du neurone i .

II.6.7.2 Le Perceptron multicouche

Ce modèle fut le premier proposé en 1958 par "Frank Rosenblatt". Le perceptron était constitué de trois couches appelées :

- Rétine (neurones d'entrées)
- Aire d'association (neurones cachés)
- Neurones réponse (neurones de sortie)



FigII.13 : Architecture d'un PMC à trois couches

Le perceptron multicouche est sans doute le plus simple et le plus connu des réseaux de neurones. La structure est relativement simple : une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées. Chaque neurone n'est relié qu'aux neurones des couches précédentes, mais à tous les neurones de la couche précédente. La fonction d'activation utilisée est en générale une somme pondérée [5].

II.8 Conclusion :

Dans ce chapitre, nous avons présenté les concepts de base pour la compréhension et l'utilisation des réseaux de neurones. Nous avons procédé par une présentation biologique des réseaux de neurones et de la notion du neurone formel qui est une présentation mathématique, puis nous avons donné un aperçu sur les différentes classes d'apprentissage. A la fin du chapitre nous avons étudié en détails la méthode de rétro propagation du gradient comme algorithme d'apprentissage.

CHAPITRE III

RESULTATS DES SIMULATIONS

III.1 Introduction :

Dans ce chapitre nous donnerons les résultats et simulation des algorithmes et organigrammes citées dans les chapitres précédents à savoir les paramètres importants des signaux de la parole sous le format « WAV » et la reconnaissance de mots. Pour cela on a utilisé le logiciel Matlab pour réaliser une interface à l'aide de son guide pour faciliter les opérations de dialogue des différents sous programmes.

Mais avant d'entamer l'étape des simulations, on donne un aperçu sur les applications qu'on a introduit pour cette étude.

III.2 Présentation des applications :

Dans ce qui suit, on va diviser notre travail en deux phases principales :

La première phase :

Concernant le traitement de la parole dont on va montrer les quatre paramètres du signal de la parole (l'analyse temporelle, l'analyse spectrale, la densité spectrale de puissance et la préaccentuation) voir la figure III.1

La deuxième phase :

Elle concerne les phases de la reconnaissance automatique de la parole voir figure III.6

III.3 Présentation de l'interface des applications :

III.3.1 La première phase :

Pour étudier le traitement de la parole nous avons réalisé une interface avec Matlab, pour présenter les différents paramètres de signal de la parole. Cette interface permet de faire varier un certain nombre de paramètres et visualiser un certain nombre de résultats. La figure III.1 montre cette interface.

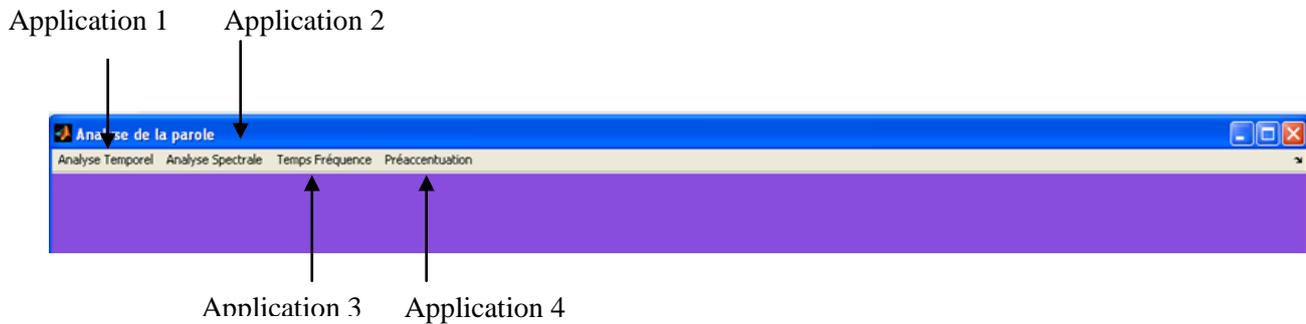


Fig III.1 : L'interface principale de l'analyse de la parole

Application 1 :

La figure III.2 visualise l'interface de l'analyse temporelle contenant 02 zones :

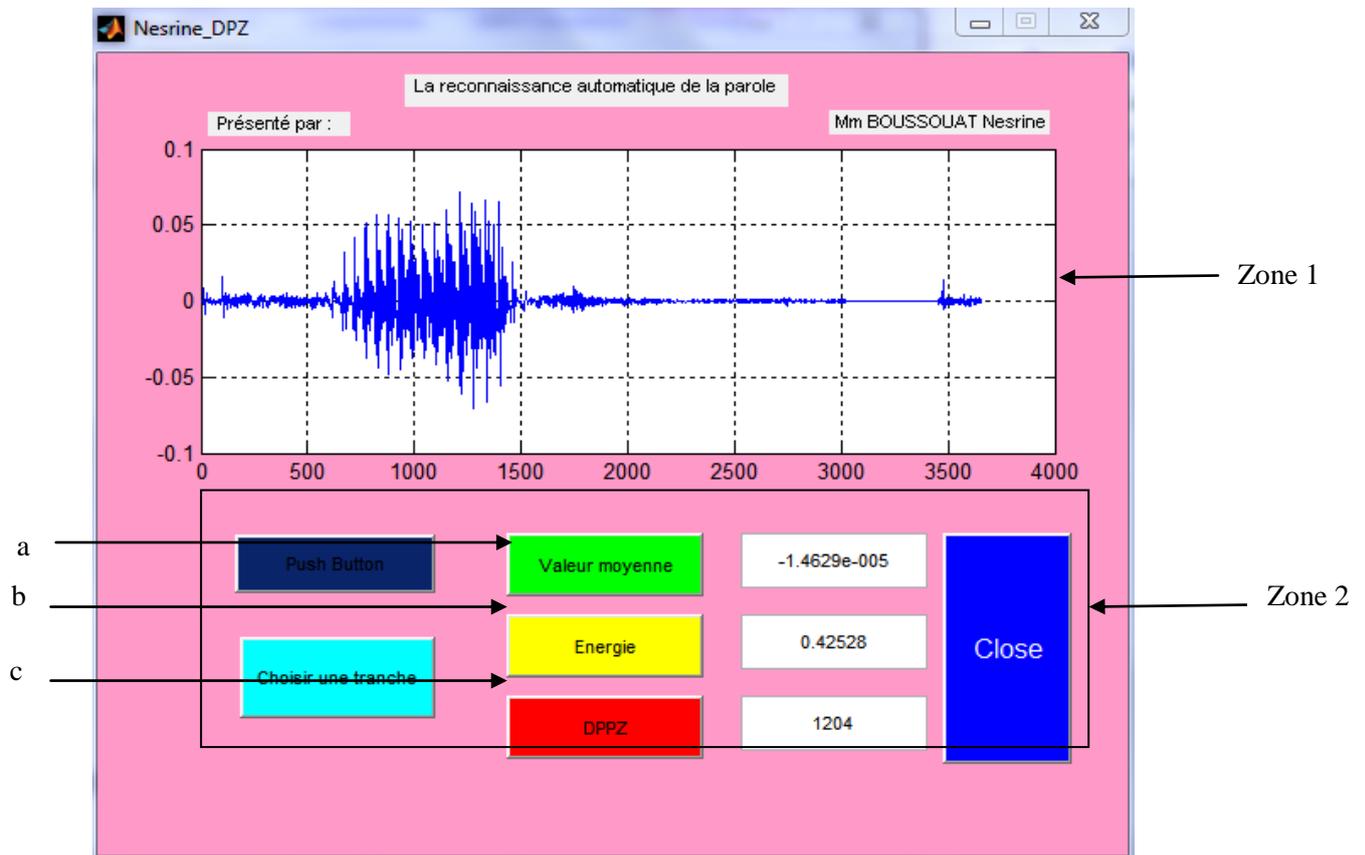


Fig III.2 l'interface de l'analyse temporelle d'un signal

Zone 1 : Sert à visualiser le signal de la parole

Zone 2 : Contient des différents boutons qui servent à calculer les paramètres du signal

- a- Le calcul de la valeur moyenne
- b- Le calcul de l'énergie
- c- Le calcul de DPZ

Application 2 :

Représente l'analyse spectrale dont on va étudier la FFT. La figure III.3 montre le menu

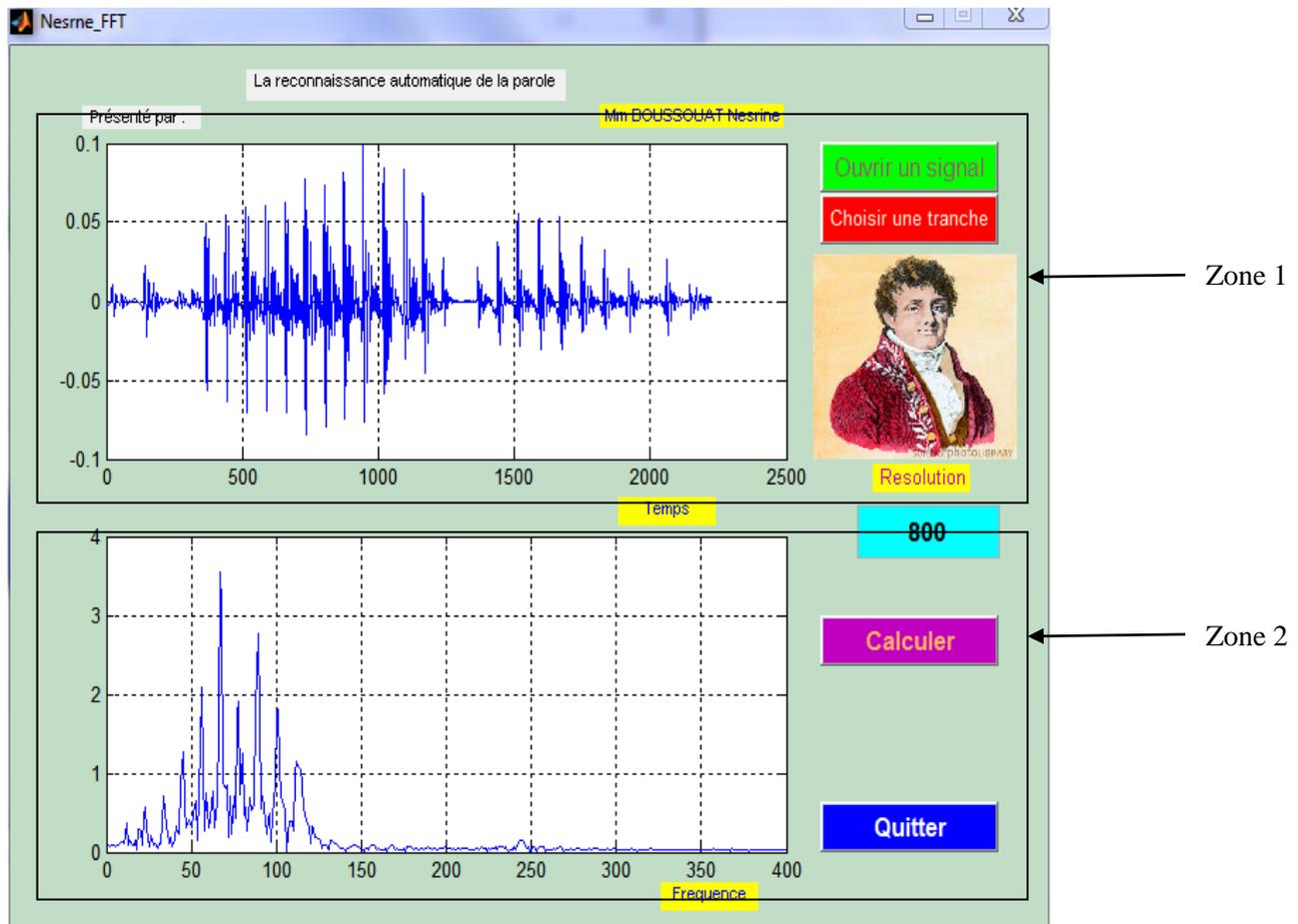


Fig III.3 l'interface de FFT

La zone 1 sert à visualiser le signal de la parole et la zone 2 sert à visualiser sa transformée de fourrier

Application 3 :

Représente les signaux DSP (signal et sa puissance) .La figure III.4 montre le menu

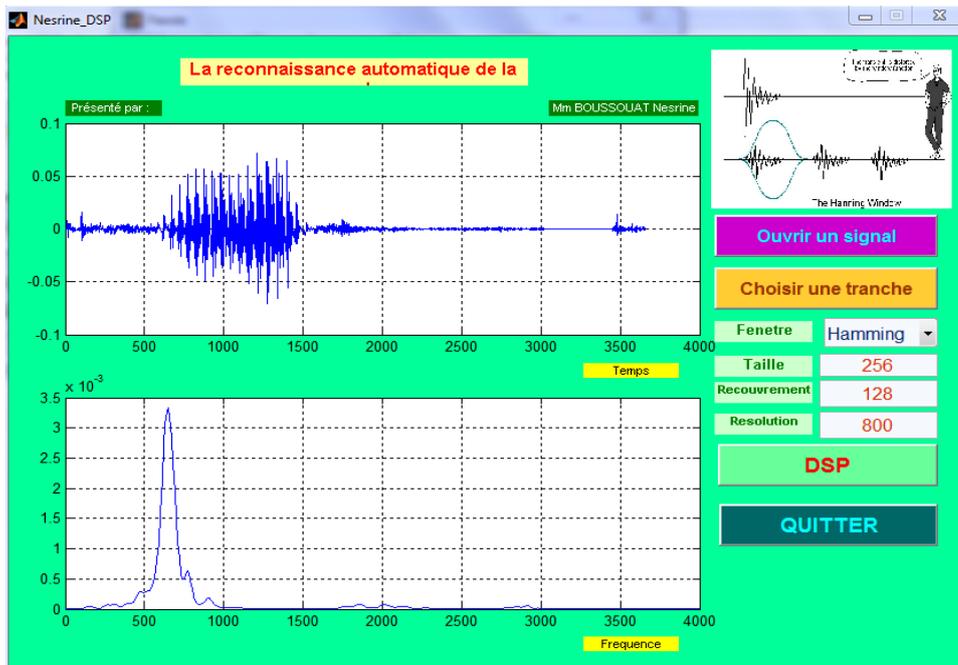


Fig III.4 L'interface de DSP

Application 4 :

La figure III.5 visualise interface de **préaccentuation** contenant deux boutons.

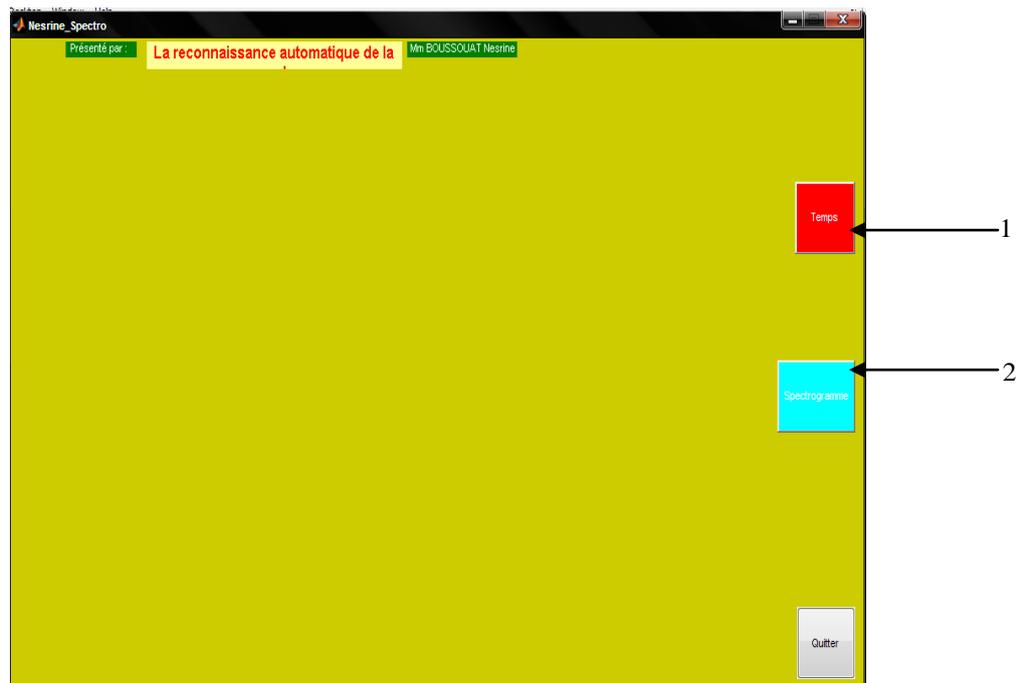


Fig III.5 Interface de préaccentuation

Bouton **temps** donne le signal original et le signal pré accentué et bouton **spectrogramme** donne leurs spectrogrammes

III.3.2 La deuxième phase :

Pour étudier la reconnaissance de la parole nous avons réalisé un autre menu avec Matlab guide voir figure III.6 .Ce menu permet de visualiser les résultats de reconnaissance des mots que nous avons utilisée pour les confirmer.

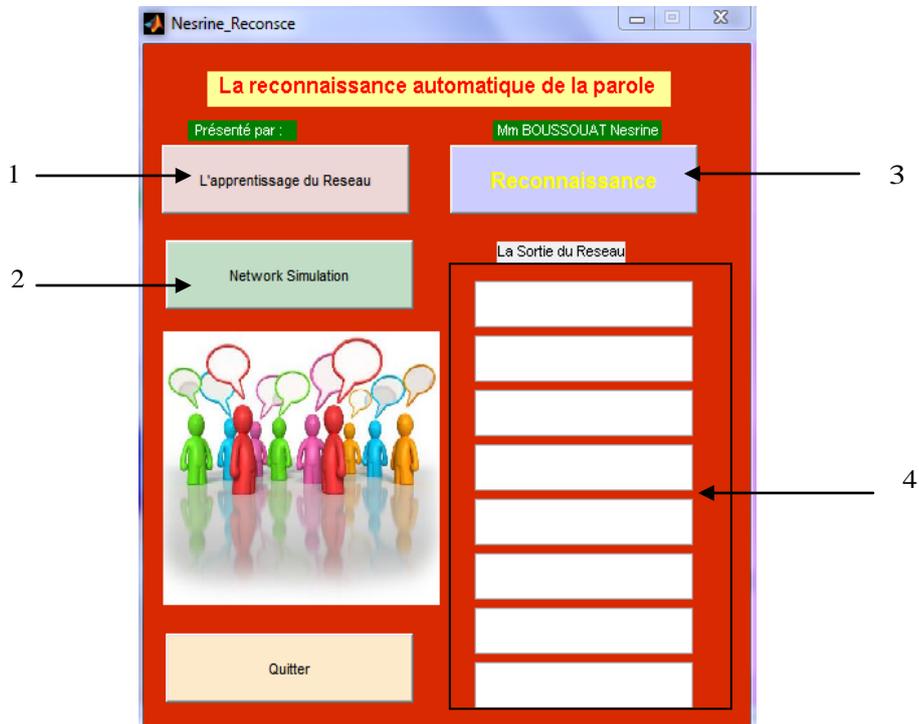


Fig III.6 Interface de reconnaissance

1-L'apprentissage du Réseau : avec ce bouton on fait l'apprentissage des mots, pour notre cas on a introduit 07 mots anglais enregistrés avec des bonnes conditions prise d'une base de donnée américaine ces mots sont : « **Aptitude** », « **Bidding** », « **Man** », « **Test** », « **Wash** », « **Water** », « **Year** » dans l'ordre suivant

Pour cela nous avons choisis un réseau de neurone à 3couche (PMC)

Quand on applique un RN pour la classification, le nombre de neurone de sortie égale au nombre de classes, le $i^{\text{ème}}$ neurone correspond à la $i^{\text{ème}}$ classe, ce qui nous a mené dans le calcul de la sortie désirée de forcé à 1 le neurone qui représente la classe du mot propagé dans le réseau et de 0 ailleurs, dans le but à apprendre au réseau que le mot propagé appartient à cette classe. Par exemple :

« **Aptitude** » correspond à une sortie du réseau de neurone du type « 1 0 0 0 0 0 »

« **Test** » correspond à une sortie du réseau de neurone du type « 0 1 0 0 0 0 »

Dans cette phase lorsque on clic dans ce bouton il fait la procédure de l'apprentissage .voir la figure III.7

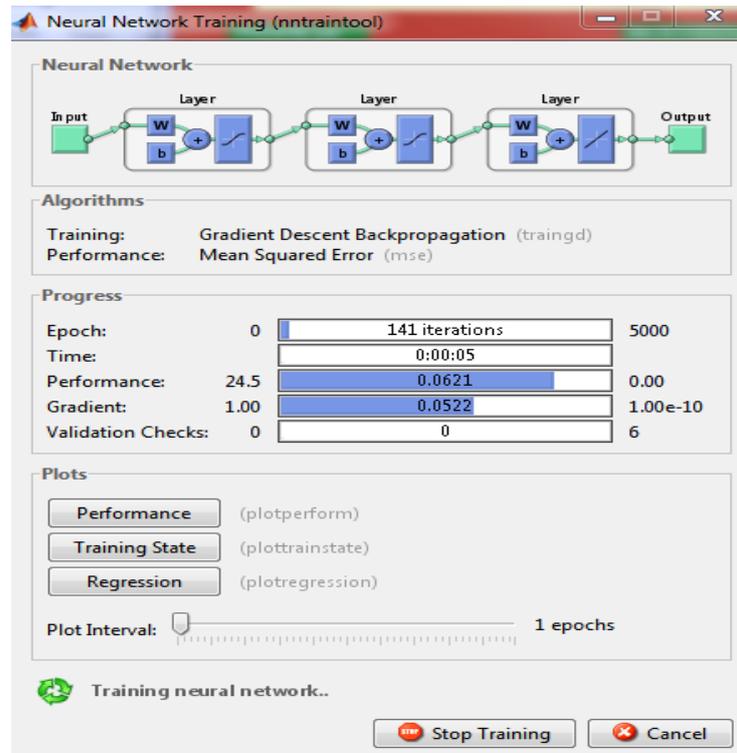


Fig III.7 : L'interface de l'apprentissage

2-Network simulation : En cliquant sur ce bouton il nous donne la figure III.8:

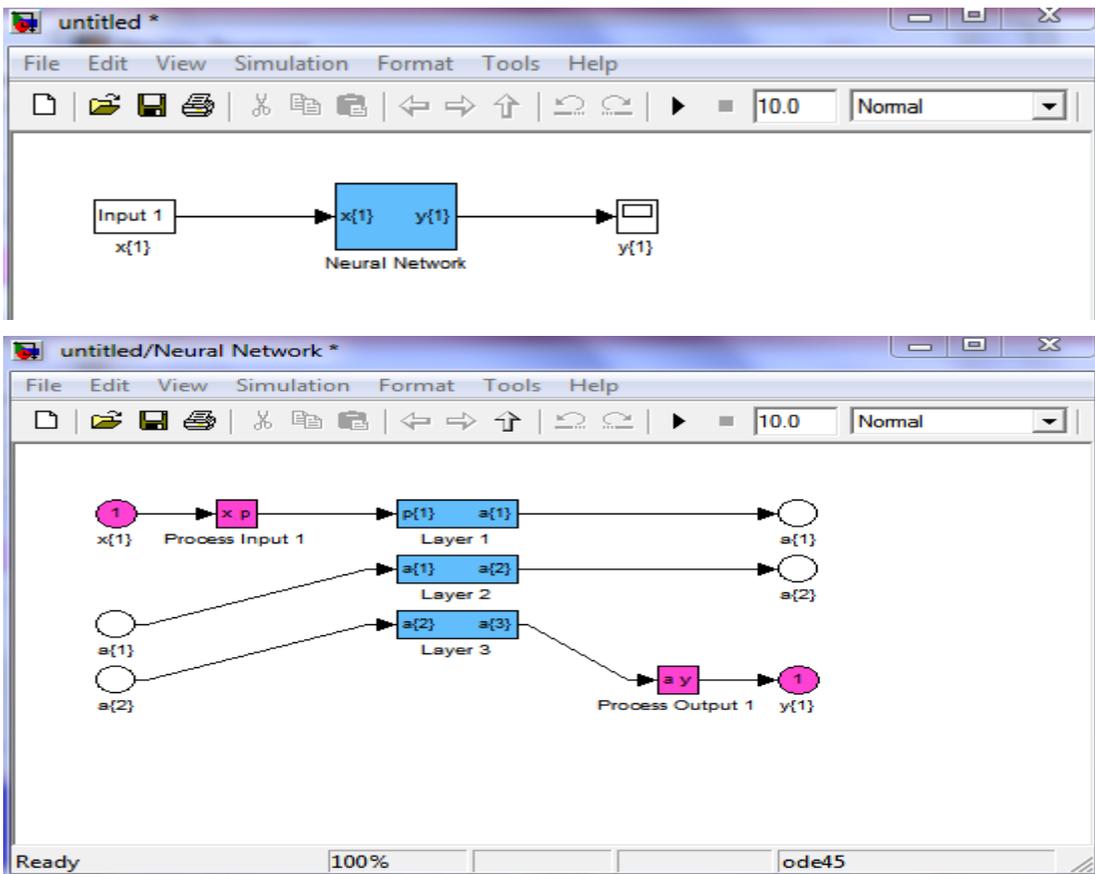


Fig III.8 Schéma explicatif de fonctionnement de réseau de neurone

3-Reconnaissance : En cliquant sur ce bouton il nous donne un menu pour choisir un mot. Puis on vérifie sa reconnaissance à travers la zone « **La Sortie du Réseau** »

Pour cela on a utilisé plusieurs sous programmes :

- On convertit le signal sous la forme numérique en utilisant la commande « **wavread** » de Matlab.
- On extrait les coefficients cepstraux de signal à partir de sous programme « Nesrine_MFCC ».
- On lui applique les paramètres du réseau de neurone trouvés au cours de l'apprentissage.
- On applique la commande « **sim** » de Matlab pour déterminer la sortie de réseaux de neurone et on fait une comparaison.

4-La Sortie du Réseau : c'est une zone d'affichage qui contient 8 cases : les 7 premières représentent les cases des mots à tester. si je trouve un nombre très proche de 1 il suffit de repaire sa position pour dire qu'il a reconnu le mot à tester dans le cas contraire c'est un mot inconnu.

Et la dernière case affiche le résultat de test de reconnaissance.

III.4 Simulation des résultats :

Dans cette étape on va présenter les différents résultats et discussions obtenus pour les différentes applications. par exemple si on prend le mot « Test » on aura :

III.4.1 Simulation de l'application 1: Energie, DPZ, La valeur moyenne :

La figure III.9 représente le signal $S(f)$ du mot « Test ». le tableau III.1 représente les paramètres énergie DPZ et la valeur moyenne pour différents bandes de fréquence (voir figure III.2).

- On observe que l'énergie sa augmente avec la tranche de l'intervalle et donc lorsque l'intervalle dépasse 2000 ms l'énergie varie faiblement
- La valeur moyenne est autour de 0 parce que le signal est à peu près symétrique par rapport a l'axe des temps.
- Le DPZ augmente avec la durée de signal.

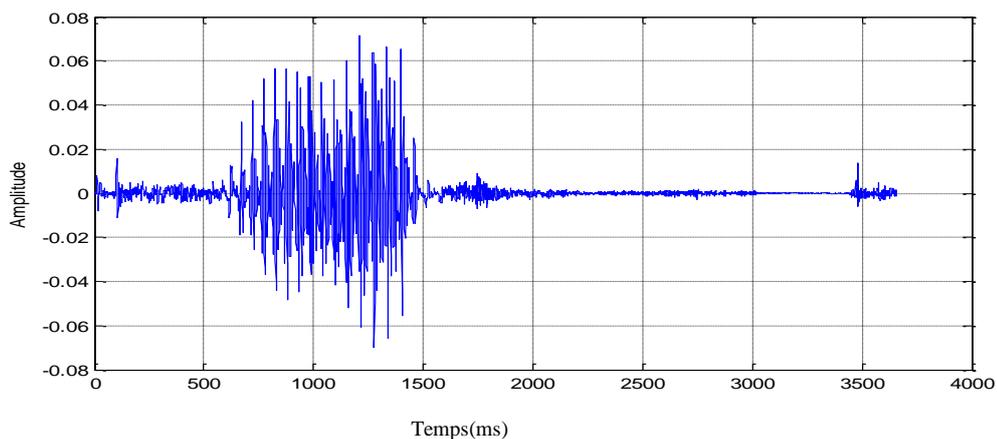


Fig.III.9: signal $S(f)$ du mot « TEST »

Les résultats :

Intervalle	0-500	0-1000	0-1500	0-2500	0-3500
Les paramètres					
Energie	0.0026888	0.14277	0.42183	0.4298	0.42499
DPZ	145	247	329	1136	1138
Valeur moyenne	1.5536 e-005	1.9732 e-005	6.0051 e-006	-9.6192 e-006	-1.0986 e-005

Tab III.1 Tableau des paramètres du mots test

III.4.2 Simulation de l'application 2 : FFT

Cette simulation représente la « FFT » du mot « Test »

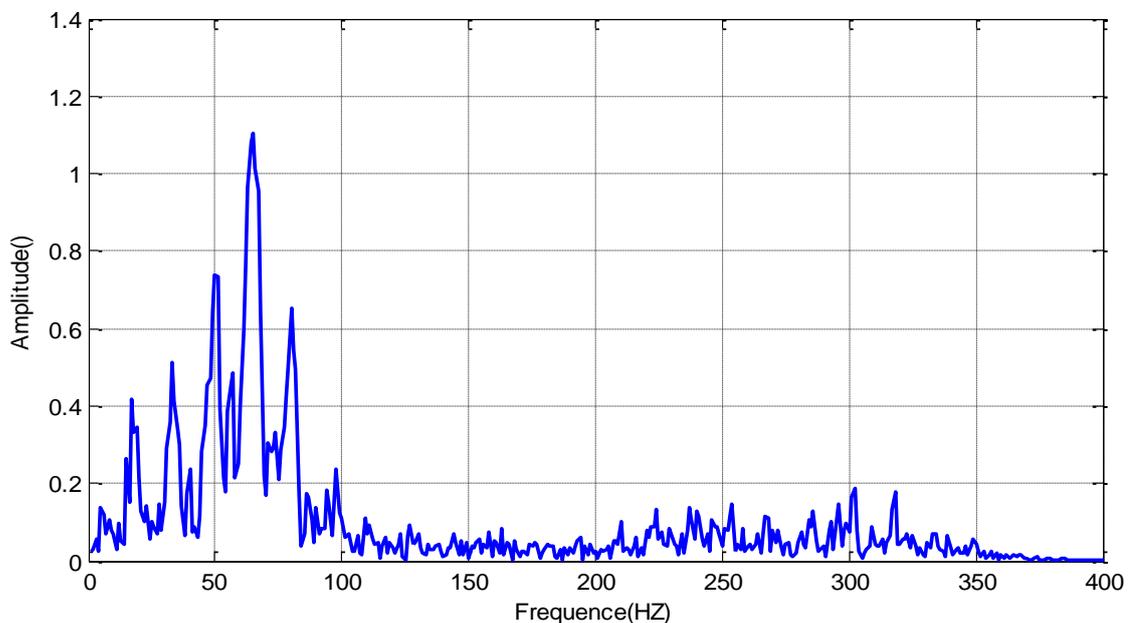


Fig III.10 : La FFT pour le mot « TEST »

Commentaires :

On remarque dans la figure III.10 que le spectre du mot s'étale jusqu'à à peu près 350HZ et que presque la totalité de la puissance du mot est contenu dans la bande 0-100 HZ

III.4. 3 Simulation de l'application 3 : DSP

on a utilisé la fonction « psd » de Matlab pour calculer la densité spectrale de puissance

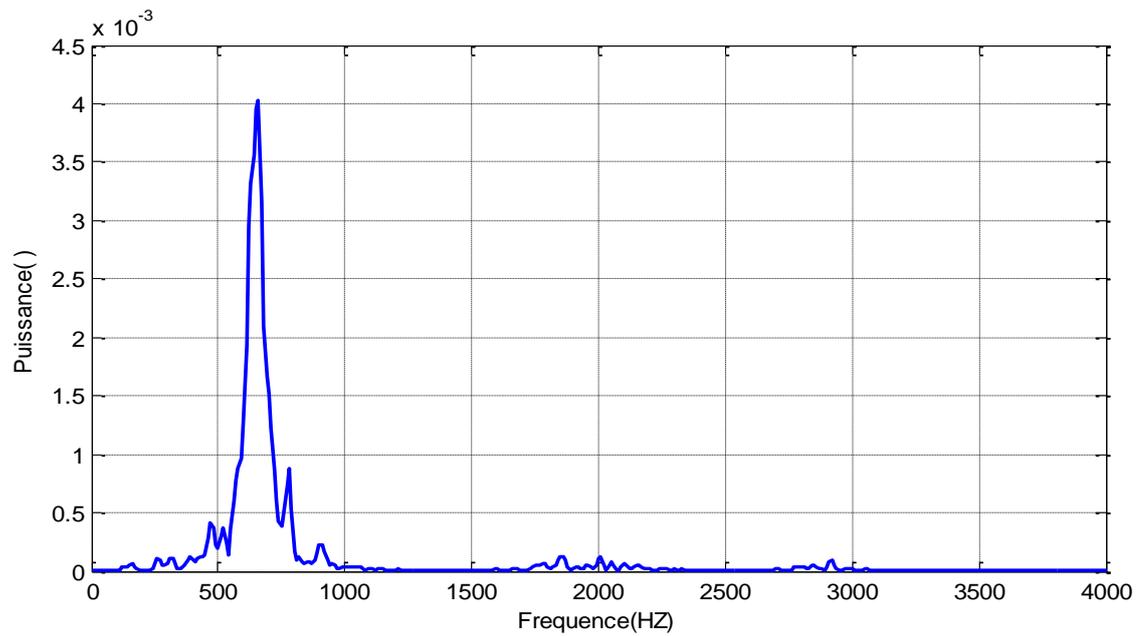


Fig III.11: DSP du mot « TEST »

Commentaires :

On remarque dans la figure III.11 que la densité spectrale de puissance de mot s'étale jusqu'à 4000 Hz alors que presque la totalité de puissance comprise dans la bande entre 450 Hz et 1000 Hz.

III.4.4 Simulation de l'application 4 : préaccentuation

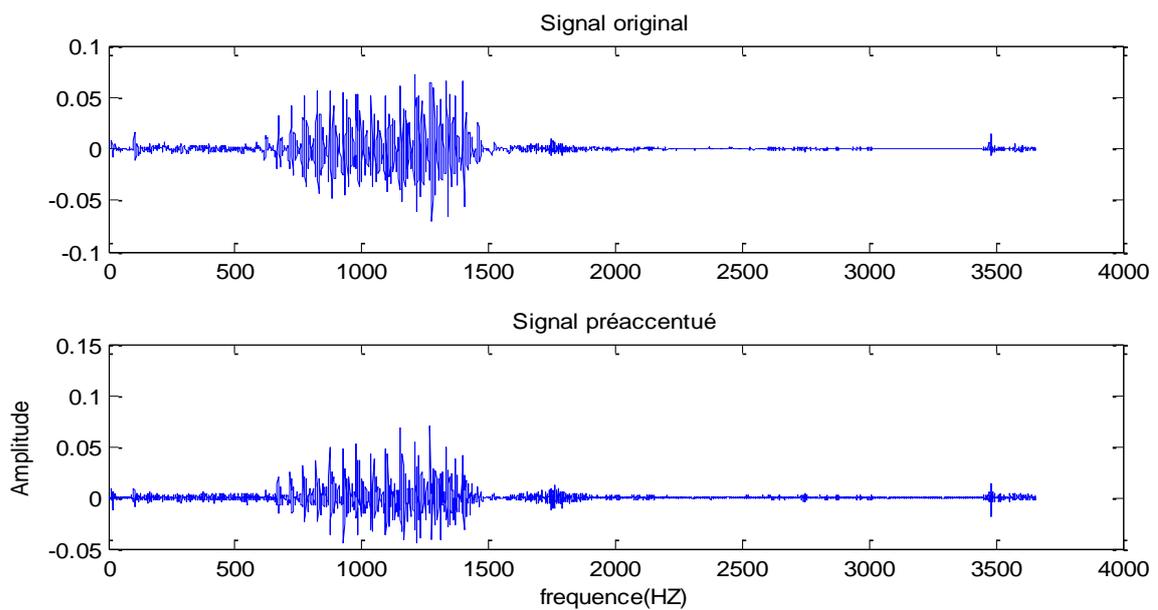


Fig III.12 : signal pré accentué de mot « Test »

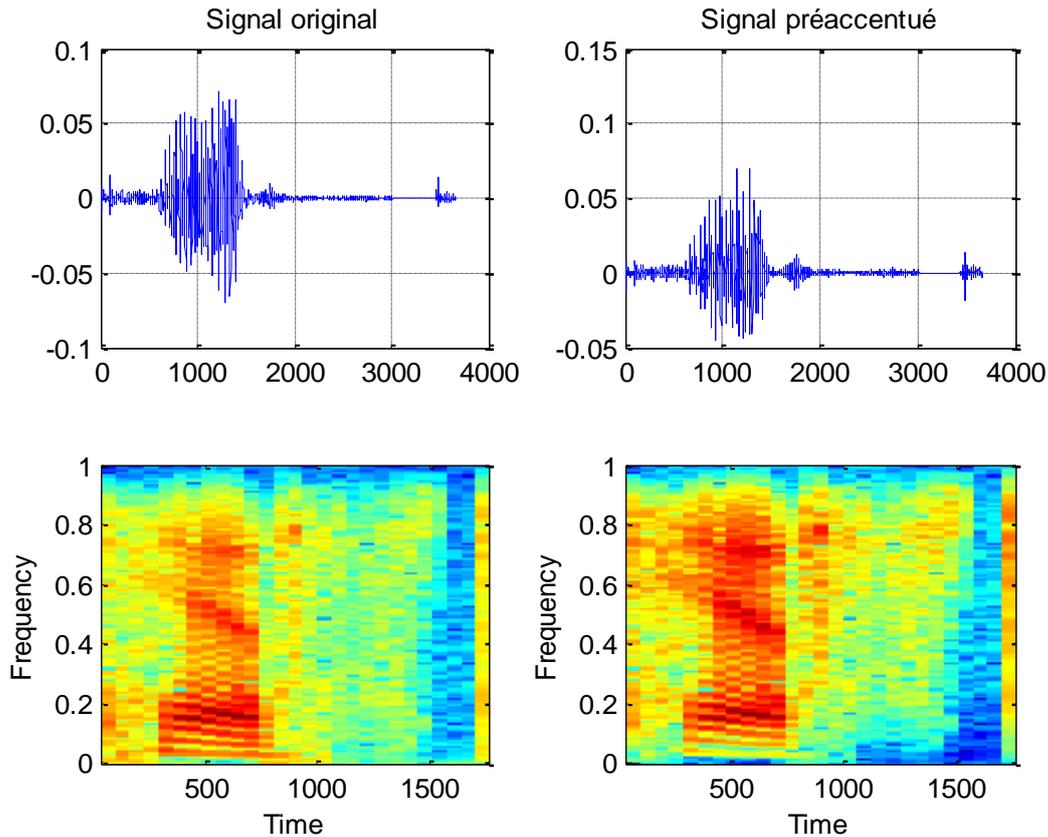


Fig III.13 : le spectrogramme de mot « Test »

Commentaires :

Dans la figure III.13 le spectrogramme donne l'énergie de signal en fonction de la couleur : Plus la couleur est foncée plus l'énergie est grande donc on remarque dans la figure de gauche avant préaccentuation la couleur de fréquence haute n'est pas foncée donc le signal a une énergie faible. Dans la figure droite après préaccentuation on observe que la couleur haute est plus foncée donc on a bien augmenté la puissance de fréquence haute.

III.4.5 Simulation de l'application de reconnaissance :

Dans cette application on prend un exemple de deux mot le premier est reconnu et l'autre n'est pas reconnu

Exemple 1 : on prend l'exemple du mot « **Wash** » les résultats sont montrés dans la figure III.14

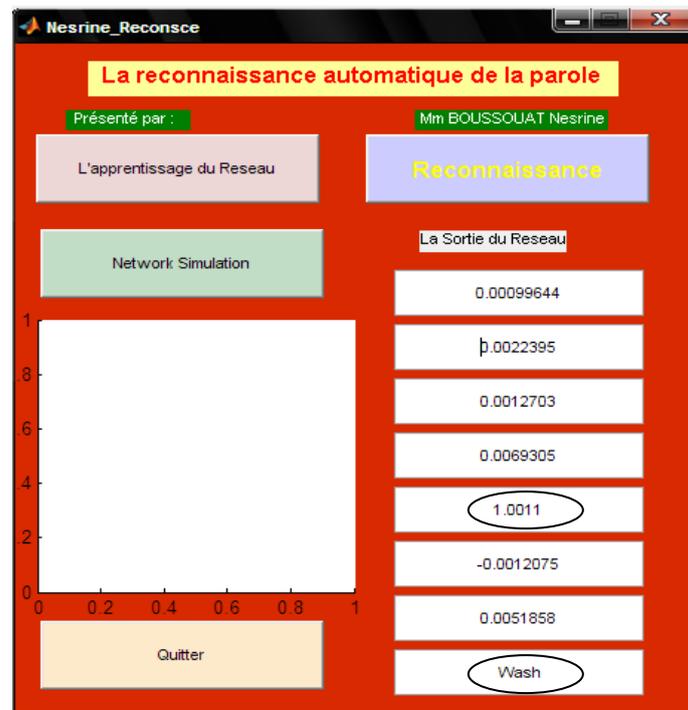


Fig III.14 : Test de reconnaissance du mot « wash »

On remarque que le mot que nous avons testé est affiché dans la dernière case de « la sortie du réseau » car la correspondance est de même type de « 0 0 0 0 1 0 0 » qui correspond au mot « Wash », lors de la reconnaissance.

Exemple2 : dans cet exemple on prend le mot « Test » prononcé de deux façon différentes ; une fois avec une voix d'une femme et l'autre d'un homme,

- « Test » de la voix de la femme : les résultats sont montrés dans la figure III.15



Fig III.15 le mot « Test » de la voix de la femme

On remarque que le mot est reconnu

- « Test » de la voix de l'homme: les résultats sont montrés dans la figure III.16

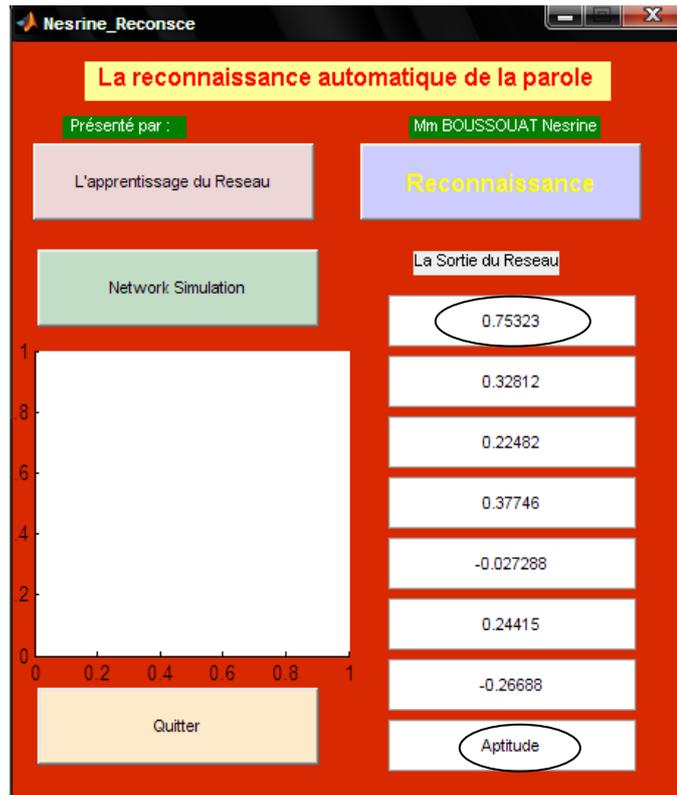


Fig III.16 : Exemple de mot n'est pas reconnu

On remarque que le mot « Aptitude » est affiché dans la dernière case de « la sortie du réseau » au lieu le mot « Test » que nous avons fait le test, car le réseau choisi le maximum des valeurs qui situé dans la 1^{ère} case qui correspond le mot « Aptitude », donc le mot « Test » n'est pas reconnu.

A partir de deux résultats on remarque que le réseau a des difficultés pour reconnaître le mot si on change la prononciation car dans cet exemple il n'est pas appris bien sur la voix de l'homme

III.6 Conclusion :

D'après l'analyse des résultats des simulations on a pu constater que notre réseau neurone donne de bon résultat pour les mots appris néanmoins si on change de prononciation notre réseau a des difficultés pour bien reconnaître le mot.

Conclusion générale

Nous avons présenté dans ce mémoire, une étude de reconnaissance de la parole en utilisant les réseaux neurone capables d'apprendre des mots donnée pour identifié une voix en se basant sur deux types de critères, l'analyse de la parole et la reconnaissance automatique de la parole.

D'après l'analyse des résultats des simulations on a pu constater que la reconnaissance de la parole par les réseaux neurone a donné des résultats satisfaisants pour les mots appris durant le test de reconnaissance de même il nous a donné aussi des résultats négatifs (pourcentage faible) pour les mots inconnus. On a remarqué que l'opération d'apprentissage s'effectue dans un temps raisonnable avec une capacité de reconnaissance acceptable. Ceci nous permet d'élargir la base des données des mots à apprendre et que l'on peut les généraliser sur toutes les applications pareilles de reconnaissance.

A partir de ces constatations, nous suggérons de poursuivre ce travail en cherchant à appliquer les réseaux neurone en utilisant beaucoup de mots avec différents prononciations.

Bibliographie

[1] **BOITE R.** et **KUNT M.**, *Traitement de la parole*, Lausanne, **PRESSES**

POLYTECHNIQUES ROMANDES, 1987.

[2] **CALIOPE**, *La parole et son traitement automatique*, Paris, **MASSON, 1989**

[3] **DAVALOP E.** et **NAIM P.**, *des réseaux de neurones*, Ed. **MASSON 19**

[4] **TOUZET**, *les réseaux de neurones artificiels, introduction au connexionnisme,*
Cours, exercices et travaux pratiques.

[5] **JODOIN J.F.**, *les réseaux neuromimétiques.*

- *Principes et définitions.*
- *Modèles et applications.*

Ed.HERMES, 1994.

[6] **Bernadette B. M.**, *la logique floue et ses applications*, Ed. **Addison-Wesley 1994**

[7] **BENYAHIA M** et **BENZAIR H.**, *navigation neuro-flou par apprentissage*

Supervisé et semi-supervisé d'un robot mobile.

[8] **ABLA K.**, *réseaux neuro-flous théorie et applications.*

« *Je parle, donc je suis ?* » *Un bilan des développements récents en traitement automatique de la parole* **T. Dutoit** **Faculté Polytechnique de Mons 2000.**