UNIVERSITÉ DU QUÉBEC

MÉMOIRE
PRÉSENTÉ À
L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INGÉNIERIE

PAR
MIGUEL GARCÍA

PROTOTYPE DE SYSTÈME DE RECONNAISSANCE DE PAROLE PAR RÉSEAU DE NEURONES UTILISANT UNE ANALYSE PAR DÉMODULATION.

JANVIER 1997

Bibliothèque
Paul-Émile-Boulet

UQAC
Université du Québec
à Chicoutimi

Mise en garde/Advice

Afin de rendre accessible au plus grand nombre le résultat des travaux de recherche menés par ses étudiants gradués et dans l'esprit des règles qui régissent le dépôt et la diffusion des mémoires et thèses produits dans cette Institution, **l'Université du Québec à Chicoutimi (UQAC)** est fière de rendre accessible une version complète et gratuite de cette œuvre.

Motivated by a desire to make the results of its graduate students' research accessible to all, and in accordance with the rules governing the acceptation and diffusion of dissertations and theses in this Institution, the **Université du Québec à Chicoutimi (UQAC)** is proud to make a complete version of this work available at no cost to the reader.

L'auteur conserve néanmoins la propriété du droit d'auteur qui protège ce mémoire ou cette thèse. Ni le mémoire ou la thèse ni des extraits substantiels de ceux-ci ne peuvent être imprimés ou autrement reproduits sans son autorisation.

The author retains ownership of the copyright of this dissertation or thesis. Neither the dissertation or thesis, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

Table des matières

Liste des figures

Liste des tableaux

Sommaire

L'analyse par démodulation est un outil effectuant des transformations sur des signaux de parole, de façon à faire ressortir les caractéristiques d'enveloppe à la sortie d'un banc de filtres cochléaires.

Ce travail se propose d'étudier la pertinence de l'analyse par démodulation et d'élaborer une architecture à réseaux de neurones capable d'en extraire les paramètres les plus importants afin de concevoir un système pouvant reconnaître des voyelles avec plusieurs locuteurs.

Remerciements

Je désirerais remercier ici toutes les personnes qui m'ont accordées leur aide et leur temps pour m'aider à terminer ce travail.

Tout d'abord, merci à monsieur Jean Rouat, qui a patiemment lu et relu mes chapitres au fur et à mesure que je les écrivais, et qui a toujours répondu à mes questions malgré ses nombreux déplacements et les miens. Fax, téléphone, courrier, Internet, tous les moyens de communications ont été utilisés pour continuer à travailler ensemble et à effectuer un suivi rigoureux de mes travaux.

Je désirerais aussi remercier madame Chantale Dumas pour la rapidité avec laquelle elle a toujours su m'aider dans le côté administratif de mon séjour à l'UQAC. Étant souvent absent lorsque je devais m'inscrire ou demander des prolongation de délai ou autre, c'est à elle que je dois d'avoir mis la main sur les bon formulaires à remplir au bon moment.

Merci à monsieur Enrique Gochicoa, qui a immédiatement accepté de mettre à ma disposition 24 heures sur 24 les ordinateurs et les locaux de l'Inter

American Bank lors de mon séjour à Washington D.C., pour me permettre de travailler le sujet de ma maîtrise.

Merci à tous les membres de ma famille, en particulier à mon épouse Imelda pour leur soutien et leur encouragement, qui m'ont permis de mener à terme ce travail.

Enfin, merci aussi à tous les gens qui, comme moi, pensent que cette planète est si petite qu'on ne devrait pas l'étouffer avec des notions de frontières, de pays, de races et de nations. Merci à ces gens qui me permettent de continuer mon cheminement et de croire que l'humanité a quand même quelque chose de bon en elle et mérite qu'on travaille à en améliorer le sort.

1

Introduction

Aucun système, jusqu'à ce jour, n'a pu effectuer très efficacement la reconnaissance de la parole continue, en temps réel et dans un millieu bruité. Différents outils ont été développés pour permettre la reconnaissance de parole continue. Les objectifs de ce travail sont d'une part d'étudier la pertinence de l'analyse par démodulation (utile en reconnaissance de parole continue), et d'autre part d'élaborer une architecture à réseaux de neurones pourvant s'en servir pour la reconnaissance de parole continue.

Nous avons donc développé un système à réseaux de neurones pouvant reconnaître une catégorie de sons voisés dans un contexte de parole continue. Le réseau de neurones utilisé est de type DYSTAL (DYnamically STable Associative Learning). Il analyse un signal de parole pré-traité sous forme d'image 3-D, et en compare des portions avec une série de patrons qui ont été obtenus suite à un apprentissage Pavlovien (série de stimuli conditionnés et non-conditionnés). Lors de cet apprentissage, des patrons sont crées au besoin en présentant un stimulus conditionné de pair avec un stimulus non-conditionné. Après l'apprentissage, le stimulus non-conditionné sera reproduit à la sortie du réseau si on lui présente une entrée similaire au stimulus conditionné avec lequel il a appris. Le système de reconnaissance de parole ainsi que ses performances sont présentés.

Le prototype a été testé sur les lettres de l'alphabet en français à partir de 6 locuteurs. L'évaluation s'est limitée aux voyelles.

Nous montrons dans ce travail que l'approche que nous avons étudiée peut constituer la base d'un système de reconnaissance de parole plus

complexe capable de reconnaître une grande variété de sons, mais nous montrons aussi quelques faiblesses de l'analyse par démodulation (manque de caractéristiques discriminantes entre certains sons). Enfin, nous indiquons quelques améliorations qui pourraient être apportées à notre système pour une meilleure reconnaissance (génération de patrons artificiels et apprentissage supervisé).

1.1. Problématique.

Nous décrirons ici la problématique à laquelle nous avons été confrontée au départ et les objectifs que nous nous sommes fixé face à cette problématique.

La plupart des système de reconnaissance de parole développés jusqu'à ce jour ont des performances très limitées lorsqu'il s'agit de travailler dans un millieu bruité et de reconnaître des sons ou des mots intégrés à l'intérieur d'un signal de parole continue. Un des problèmes majeurs auquels font face les chercheurs est donc d'élaborer une technologie pouvant reconnaître de la parole continue, et en même temps travailler dans un millieu bruité.

L'analyse par démodulation est un outil permettant d'identifier et de faire ressortir certaine caractéristiques uniformes des signaux de parole. L'analyse par démodulation s'applique bien sur des signaux de parole continue.

1.2. Objectifs.

Comme nous l'avons mentionné dans la section précédente, le système réalisé utilise en entrée les signaux de sortie donnés par l'analyse par démodulation (des images 3D, comme le verra le lecteur un peu plus loin), afin d'évaluer le potentiel de l'analyse par démodulation.

Le système devait, bien entendu, utiliser et valider (ou invalider) l'analyse par démodulation et démontrer que cette analyse pouvait être de quelque utilité en reconnaissance de la parole continue. L'architecture du système de reconnaissance restait à déterminer.

Notre travail devait aussi permettre de déterminer les faiblesses de l'analyse par démodulation et en identifier les causes, tout en mettant en valeur ses points forts par rapport à la reconnaissance de parole continue.

Afin de vérifier et de valider notre approche, nous avons étudié la reconnaissance de quelques voyelles, car ce sont les sons les plus faciles à reconnaître. Un échec à ce point aurait conduit à un abandon de l'idée.

Donc, pour résumer, nous avions deux objectifs principaux. Premièrement, la vérification de la pertinence de l'analyse par démodulation en reconnaissance de parole continue, et deuxièmement, l'élaboration d'un système de reconnaissance de parole capable d'utiliser les signaux de sortie de l'analyse par démodulation.

2

Les signaux de parole

Nous parlerons dans ce premier chapitre des signaux de parole, de ce qu'ils sont et de comment ils sont émis et détectés par les différents organes des appareils phonatoire et auditif.

Dans un premier temps, nous introduisons les signaux de parole et leurs caractéristiques physiques, puis nous parlerons de l'appareil phonatoire servant à la production de la parole. Viendra ensuite l'appareil auditif, servant au décodage des signaux de parole.

Il est à noter que ce chapitre expose des notions très générales sur les signaux de parole. Ces notions pourront être retrouvées dans pratiquement tout volume didactique traitant de la reconnaissance de la parole. Néammoins, nous retiendrons deux références principales [29] et [6]. Le lecteur pourra y trouver des renseignements sur les signaux de parole et sur leurs principales caractéristiques utilisées en reconnaissance de la parole.

2.1. Les signaux de parole.

Le signal de parole est l'entité de base sur laquelle travaillent les chercheurs concevant des systèmes de reconnaissance de parole. Un signal de parole se présente sous la forme donnée à la figure 2.1, et est produit par des

fluctuations de la pression de l'air engendrées par l'appareil phonatoire humain ([29], chapitre 4). La figure 2.1 montre ces fluctuations en fonction du temps. Un signal de parole est donc non-stationnaire, c'est-à-dire que ses propriétés statistiques (moyenne, écart-type...) varient en fonction du temps. Pour donner un ordre de grandeur, la figure 2.1 représente les variations de pression, ou les intensités sonores, pour la prononciation par une femme de la lettre "f" de l'alphabet. Le signal de cette figure a une durée d'environ 1.0 seconde.

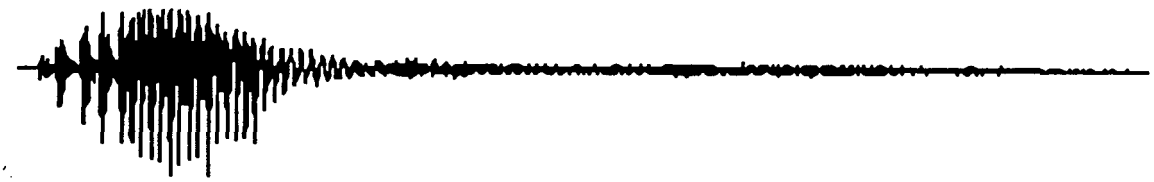


Figure 2.1. Aspect physique d'un signal de parole.

Un signal de parole peut être vu comme une suite de segments ou sons présentant plus ou moins de caractéristiques communes. Les segments de

parole sont appelés phones ou phonèmes. Ils peuvent se diviser en deux grandes catégories: les signaux périodiques (fig. 2.2) et les signaux qui prennent la forme d'un bruit (fig. 2.3). Les signaux périodiques sont produits lorsqu'il y a vibration des cordes vocales, tandis que les signaux non périodiques, quant à eux, sont produits lorsque l'air passe librement dans le conduit vocal. La figure 2.4 montre la classification des phonèmes pour la langue française.

Figure 2.2. Son périodique, ou voisé.

Figure 2.3. Son fricatif.

La figure 2.2 montre une portion d'environ 80ms du son /ɛ/ de la prononciation de la lettre "f" montrée à la figure 2.1. La très forte périodicité est bien mise en valeur ici.

La figure 2.3 est une portion d'environ 130 ms du son /ɛ/ de la lettre "f". En comparant la figure 2.2 et 2.3, on voit bien la différence entre un son non-voisé (figure 2.3) et un son voisé (figure 2.2).

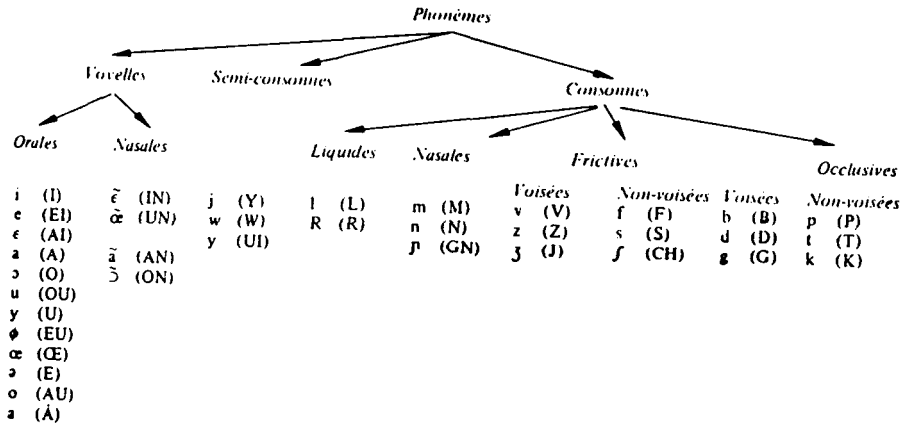


Figure 2.4. Phonèmes de la langue française [6]

Donc, ce qui frappe, à première vue, lorsqu'on étudie les sons voisés, c'est leur périodicité. La périodicité d'un son voisé est déterminée par la

fréquence de vibration des cordes vocales. Cette fréquence, appelée fréquence fondamentale, peut varier:

- de 50 à 200 Hz pour une voix masculine,

- de 150 à 450 Hz pour une voix féminine,

- de 200 à 600 Hz pour une voix d'enfant.

Nous voyons donc que la périodicité des signaux de parole est une caractéristique non-uniforme, qui varie en fonction des individus [6] p. 4.

2.2. Les formants.

Nous introduirons ici une des plus importantes caractéristiques des signaux de parole, qui est beaucoup utilisée pour l'analyse de la parole, car elle est uniforme si on étudie des groupes distincts de locuteurs. Cette caractéristique est le formant.

L' appareil phonatoire humain peut être grossièrement comparé à un tube acoustique, certaines fréquences sont atténuées et d'autres amplifiées. En fait, un tube acoustique agit exactement comme un filtre, amplifiant les fréquences proches de ses fréquences de résonance et atténuant les autres.

L'appareil phonatoire humain a la particularité de pouvoir modifier à volonté la position et la forme de ses différents organes, comme nous le verrons plus loin. Il change de ce fait la position de ses "filtres" et modifie les fréquences auxquelles il est sensible. Pour chaque son voisé, certaines fréquences sont amplifiées et d'autres sont atténuées. Les fréquences amplifiées ont pour nom *formants.* Ces formants sont donc en fait les fréquences de résonance de l'appareil phonatoire humain. Comme ce dernier peut changer ses caractéristiques physiques pour prononcer plusieurs sons, plusieurs formants peuvent donc être produits.

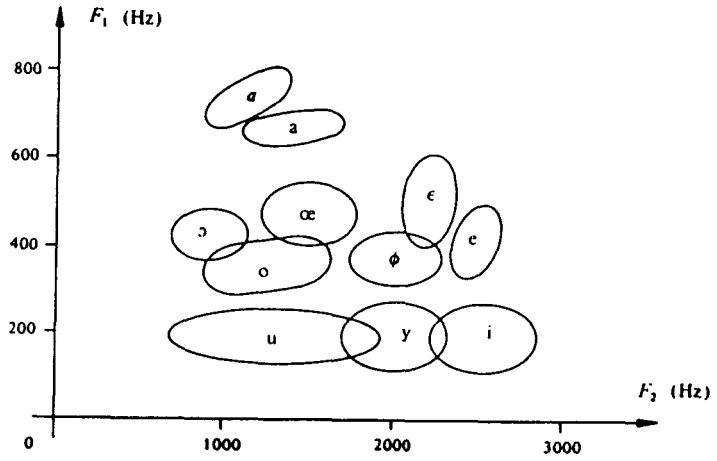


Figure 2.5. Premiers et deuxièmes formants pour différents sons de la langue française [6]

La figure 2.5 montre les formants pour différents sons de la langue française. Sur cette figure, seulement les deux premiers formants, F1 et F2, sont représentés. Il y a en effet souvent un troisième et même un quatrième formant. Il est intéressant de constater que les frontières entre les sons se recoupent, ce qui complique encore plus la reconnaissance.

2.3. Production des signaux de parole.

Les signaux de parole sont produits par l'appareil phonatoire humain, qui est montré sur la figure 2.6.

La parole est le résultat de l'action volontaire et coordonnée des appareils respiratoire et masticatoire. Cette action se déroule sous le contrôle du système nerveux central qui reçoit en permanence des informations par rétroaction auditive et par les sensations cénesthétiques.

L'appareil respiratoire fournit l'énergie nécessaire lorsque l'air est expiré par la trachée. Au sommet de celle-ci se trouve le larynx où la pression de l'air est modulée avant d'être appliquée au conduit vocal, qui s'étend du pharynx jusqu'aux lèvres.

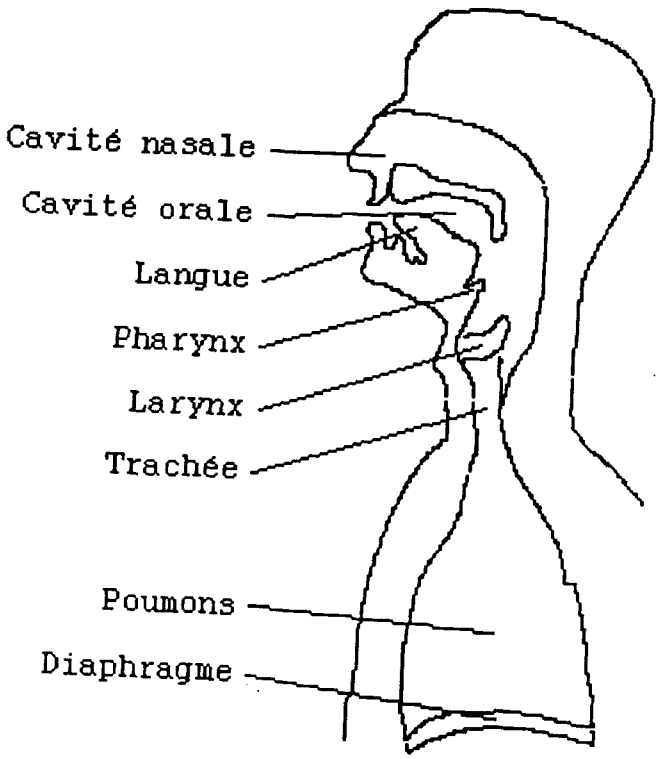


Figure 2.6. L'appareil phonatoire humain.

Le larynx est un ensemble de muscles et de cartilages mobiles qui entourent une cavité située à la partie supérieure de la trachée (figure 2.7). Les

cordes vocales sont en fait deux lèvres symétriques (replis musculaires) placées en travers du larynx; ces lèvres peuvent fermer complètement le larynx et, en s'écartant, déterminer une ouverture triangulaire appelée glotte. L'air y passe librement pendant la respiration et la voix chuchotée, et aussi pendant la phonation des sons sourds ou non voisés. Les sons voisés résultent au contraire d'une vibration périodique des cordes vocales; des impulsions périodiques de pression sont ainsi appliquées au conduit vocal.

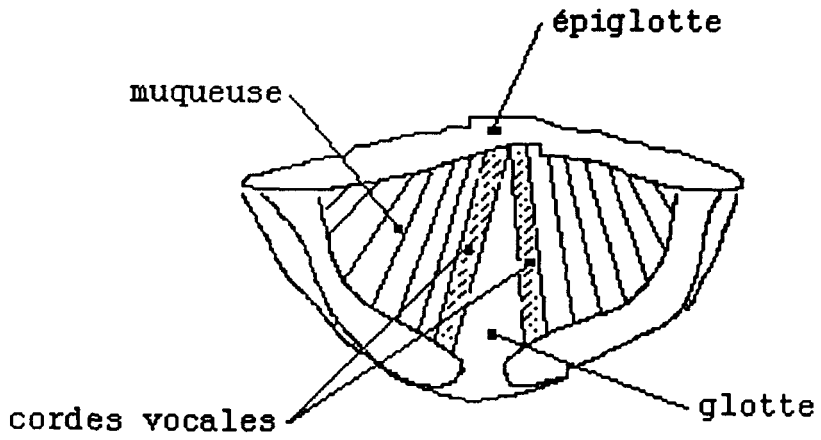


Figure 2.7. Le larynx.

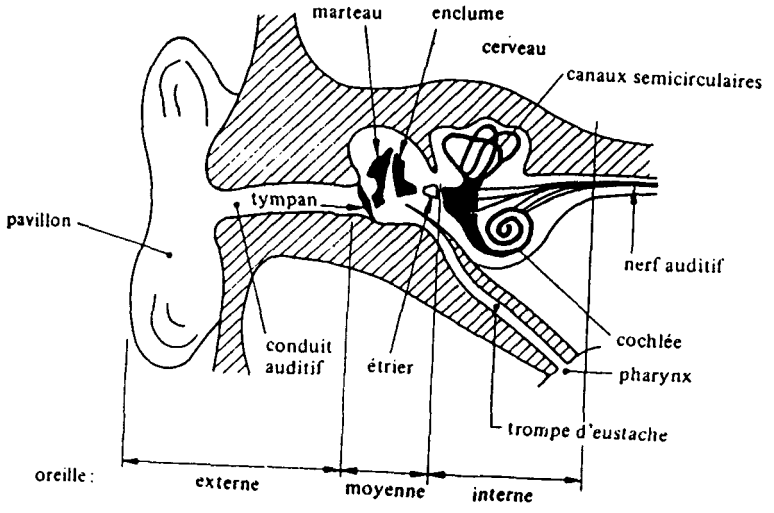


Figure 2.8. L'appareil auditif. [6]

2.4. Détection des signaux de parole.

La reconnaissance de la parole consiste, évidemment, en la détection et la classification des signaux de parole. Nous parlerons ici de l'oreille humaine, qui peut être d'une grande utilité en servant de modèle pour un système auditif . Il est important de faire la différence entre l'audition et la perception de la parole.

L'audition est le travail fait par l'oreille pour capter les variations de la pression de l'air causées par les ondes sonores. La perception de la parole est le travail effectué par le cerveau pour convertir les signaux provenant de l'oreille en des concepts intelligibles. La perception de la parole est une faculté qui doit être apprise alors que l'audition est innée.

Si les mécanismes de l'audition sont plus ou moins compris aujourd'hui, ceux de la perception de la parole le sont moins encore.

La figure 2.8 représente l'appareil auditif périphérique humain. L'appareil auditif périphérique comprend l'oreille externe, l'oreille moyenne et l'oreille interne.

Le conduit auditif relie le pavillon au tympan: c'est un tube acoustique de section uniforme fermé à une extrémité. Son premier mode de résonance est situé vers 3000 Hz, ce qui accroît la sensibilité du système auditif dans cette gamme de fréquences.

Le mécanisme de l'oreille interne (chaîne: marteau - enclume - étrier) permet une adaptation d'impédance entre l'air et le millieu liquide de l'oreille interne.

Les vibrations de l'étrier sont transmises au liquide de la cochlée. Celle-ci contient la membrane basilaire, un organe qui transforme les vibrations mécaniques en impulsions nerveuses. La cochlée contient environ 35000 cellules ciliées qui sont raccordées au nerf auditif. Il y a deux types de cellules

ciliées dans la cochlée, les internes et les externes. Les quelques 3000 cellules ciliées internes sont responsables de la détection des fréquences [2]. La figure 2.9 donne un exemple de réponse en fréquence d'une cellule ciliée. La fréquence de résonance $f_{O,\,i}$ dépend de la position occupée par la cellule sur la cochlée. Comme on peut le remarquer sur la figure 2.9, au-delà de cette fréquence, la réponse s'atténue très rapidement.

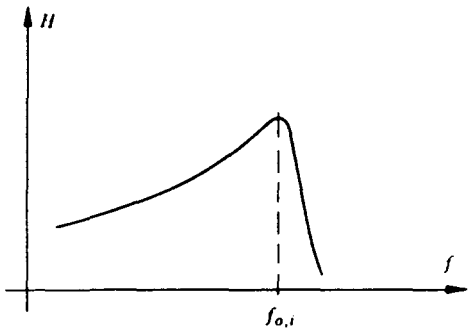


Figure 2.9. Réponse en fréquence d'une cellule de la cochlée.

Le système auditif humain est surtout sensible dans une gamme de fréquences situées entre 800 Hz et 8000 Hz; les limites extrêmes sont respectivement 20 et 20000 Hz.

La figure 2.10 représente la réponse en fréquence pour plusieurs cellules ciliées, en montrant l'échelle de fréquence et le facteur d'atténuation (variant de 0 à 1.0), 1.0 étant évidemment le maximum. On indique aussi, en millimètres, l'emplacement des cellules sur la cochlée. Nous voyons donc que les cellules les plus éloignées sur la cochlée sont les cellules qui sont sensibles aux basses fréquences.

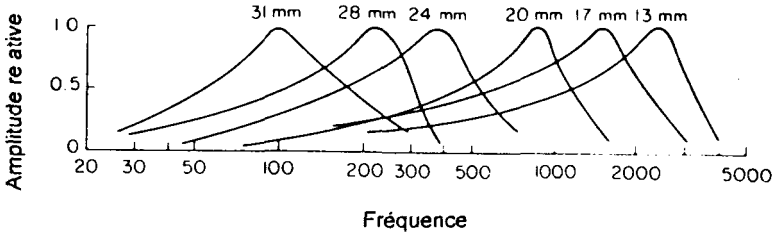


Figure 2.10. Caractéristiques de plusieurs cellules ciliées.

3

La reconnaissance de la parole

La reconnaissance de la parole est, comme nous l'avons déjà mentionné, la capacité qu'a une machine de pouvoir analyser et interpréter des signaux de parole. Construire des machines capables de dialoguer avec des humains est un des buts des recherches en reconnaissance de la parole.

Ce chapitre propose un bref historique de la reconnaissance de la parole, pour ensuite survoler la technologie présentement utilisée pour effectuer la reconnaissance de la parole. Nous verrons deux types d'analyse basées sur les techniques spectrales de la transformée de Fourier, soit FFT [29] pp. 220-224, et LPC [29] pp. 336-379, qui sont des analyses supposant une stationnarité des signaux de parole sur une échelle de temps pré-déterminée. Nous verrons aussi un type d'analyse n'assumant pas cette stationnarité [36] [22] [35].

Nous verrons ensuite les algorithmes les plus couramment utilisés, de la programmation dynamique [4] [33] aux réseaux de neurones en général [10] [21] en passant par les systèmes de Markov [16][32][34].

Ceci nous amènera, dans un dernier temps, à introduire la problématique à laquelle sont confrontés les chercheurs en reconnaissance de la parole. Nous verrons aussi les différentes solutions envisagées.

3.1. Historique.

La reconnaissance de la parole intéresse sérieusement les chercheurs depuis quelques décennies seulement. Quoique ce soit un domaine de recherche très moderne, l'analyse de Fourier, mise au point au XIX^{ième} siècle, constitue un outil mathématique important pour la reconnaissance de la parole.

La compréhension élémentaire de la production de la parole ne date pas d'hier. En effet, des systèmes mécaniques parlant (quelques mots) sont mis au point dès la fin du XVIII^{ième} siècle.

Le système auditif, quant à lui, est passablement plus complexe, et les premières tentatives de modélisation apparaissent dans les années 40. On peut dire que l'ère moderne de la reconnaissance de la parole a débuté vers les années 30, où apparaissent les premiers signaux modulés.

La perception des phonèmes (sons) comme étant des signaux distincts et pouvant être classés et traités voit le jour vers 1950. Le premier ouvrage connu, *Acoustic Theory of Speech Production*, de G. Fant [11], voit le jour en 1960, donnant le départ à des recherches qui allaient bientôt intéresser tous les pays industrialisés. Les premières méthodes d'analyse (cepstre et prédiction linéaire) sont développées.

Le début des années 70 voit une recrudescence des recherches en reconnaissance de la parole, et l'apparition de la méthode de programmation dynamique (Dynamic Time Warping). L'analyse et traitement des signaux

numériques fait des bonds de géant et des systèmes de reconnaissance de parole opérationnels sont développés.

Les développements majeurs ces dernières années concernent des nouvelles techniques d'analyse et de reconnaissance, par exemple les modèles statistiques de Markov [16] [32] [34] et les modèles connectionnistes (réseaux de neurones) [10] [8], permettent d'entrevoir les systèmes de l'avenir. Bien qu'étant encore loin de la machine capable de dialoguer avec les humains, nous pouvons cependant deviner que les systèmes de reconnaissance de parole seront de plus en plus complexes et performants avec les années.

La reconnaissance de la parole par réseaux de neurones, quoique immature pour l'instant, présente une approche intéressante et plausible pour concevoir des systèmes dont les performances approcheront et peut-être éclipseront celles des humains.

3.2. Technologie de la reconnaissance de la parole.

Les méthodes pour concevoir un système effectuant la reconnaissance de la parole peuvent se ranger, en gros, dans deux catégories.

La première, la plus complexe et celle ayant les perspectives à long terme les plus intéressantes, est l'approche analytique. Dans cette approche, on

considère que les signaux de parole sont divisés en unités élémentaires, les phonèmes ou sons. L'analyse mathématique pour concevoir de tels systèmes est très complexe, car les phonèmes présentent des aspects très différents selon le locuteur et l'environnement (bruit).

La seconde catégorie est l'approche globale. On considère ici que l'unité élémentaire de la parole est le mot, par exemple. Les systèmes commercialisés jusqu'à ce jour utilisent principalement cette approche.

Il est important de bien comprendre que la conception d'un système capable de dialoguer de façon naturelle avec une personne doit intégrer l'approche analytique sans nécessairement s'y limiter. C'est pourquoi nous nous y intéressons.

Plusieurs systèmes de reconnaissance de parole sont aujourd'hui opérationnels, bien qu'aucun d'eux ne soit un système global fonctionnant dans tous les cas. Les technologies utilisées en reconnaissance de parole ont permis de réaliser des machines capables de reconnaître [40]:

- soit une centaine de mots en mode indépendant du locuteur,

- soit des phrases à syntaxe contrainte et vocabulaire d'un millier de mots en mode plurilocuteur,

- soit des phrases (naturelles mais respectant une syntaxe de l'écrit) prononcées par mots isolés (20000 mots) ou syllabes isolées (200 000 mots) en mode monolocuteur.

3.2.1. Les analyses.

Nous verrons ici les analyses les plus courantes effectuées sur les signaux de parole. Ces différentes analyses sont plus ou moins efficaces, selon qu'elles supposent la stationnarité d'un signal de parole. Les analyses conventionnelles (FFT, LPC) supposent la stationnarité des signaux de parole à l'intérieur d'une fenêtre d'analyse, alors qu'il est évident qu'un signal de parole est non-stationnaire [22] lorsque vu dans son ensemble, mais quasi-stationnaire si on le considère sur un intervalle de temps fini. Cependant, il faut bien l'avouer, les analyses conventionnelles sont encore celles qui sont les plus performantes aujourd'hui.

Nous introduirons aussi brièvement les opérateurs non-linéaires, qui se conforment avec la non-stationnarité des signaux de parole sur un intervalle de temps donné. Nous reparlerons en détail dans le chapitre 4 de l'opérateur Dyn, que nous utilisons dans notre système.

Une analyse effectuée sur un signal de parole doit faire ressortir le plus possible les paramètres indépendants du locuteur, et neutraliser les autres. Une

bonne analyse doit aussi exploiter la redondance des signaux de parole, pour présenter seulement l'information essentielle et indépendante du locuteur.

3.2.1.1. Fenêtres et échantillonnage.

Dans la plupart des analyses, seulement une partie d'un signal de parole est considérée à la fois. Cette partie de signal est contenue dans ce que l'on appelle une fenêtre [29], pp. 206-211. Le choix d'une fenêtre adéquate, peut s'avérer crucial pour concevoir un bon système de reconnaissance de parole. Une fenêtre doit être suffisamment courte pour que les propriétés du signal de parole ne changent pas significativement à l'intérieur de cette fenêtre. Une fenêtre doit aussi être suffisamment longue pour englober assez d'échantillons pour permettre un calcul adéquat des paramètres de sortie.

Se servir d'une fenêtre signifie multiplier l'amplitude d'un signal de parole par l'amplitude de la fenêtre, pour tout instant t. La fenêtre la plus simple est la fenêtre rectangulaire *r(n)*, dont l'équation est la suivante:

$w(n) = r(n) = 1$ pour $0 <= n <= N-1$

n représente ici l'échantillon considéré dans le signal et N représente le nombre d'échantillons englobés par la fenêtre.

Ceci nous amène à parler de l'échantillonnage, qui permet de convertir un signal de parole analogique en une suite de nombres. La figure 3.1 montre, à gauche, une onde sinusoïdale pure. Lorsqu'on échantillonne un signal, on remplace carrément le signal par une série de valeurs numériques. La figure 3.1 de droite montre notre onde sinusoïdale échantillonnée. Une fenêtre sélectionne simplement un groupe d'échantillons consécutifs. La figure 3.1 montre, à droite, trois fenêtres se chevauchant. On choisit en effet très souvent des fenêtres se chevauchant à l'intérieur d'un même signal.

L'échantillonnage se fait, dans la plupart des cas, à une fréquence uniforme pour un même signal. 32 kHz et 16 kHz sont des fréquences d'échantillonnage que nous utiliserons.

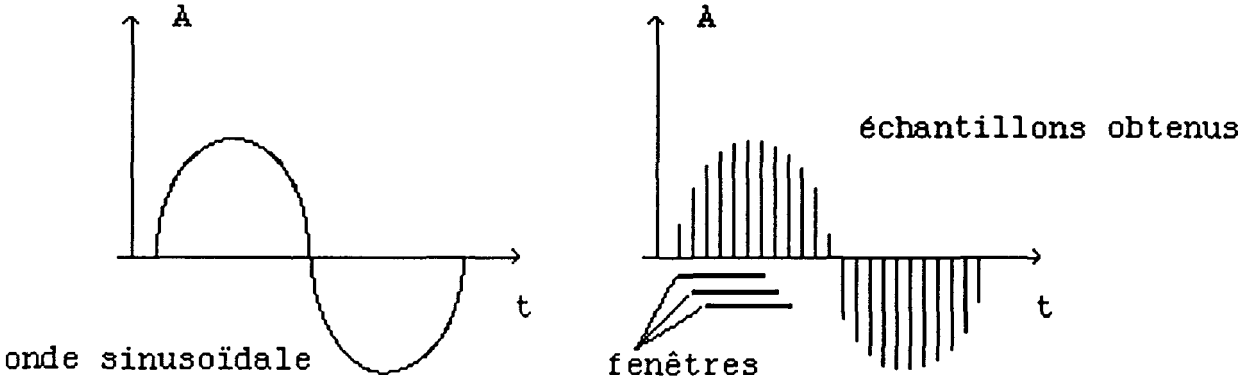


Figure 3.1. Échantillonnage d'une onde sinusoïdale.

Plusieurs types de fenêtres sont utilisés. Les fenêtres les plus courantes sont les fenêtres de Hamming, de Bartlett, rectangulaires, de Blackman et de Kaiser. La figure 3.2 montre la forme de ces différentes fenêtres. Nous parlerons plus loin des fenêtres que nous avons utilisées pour notre système.

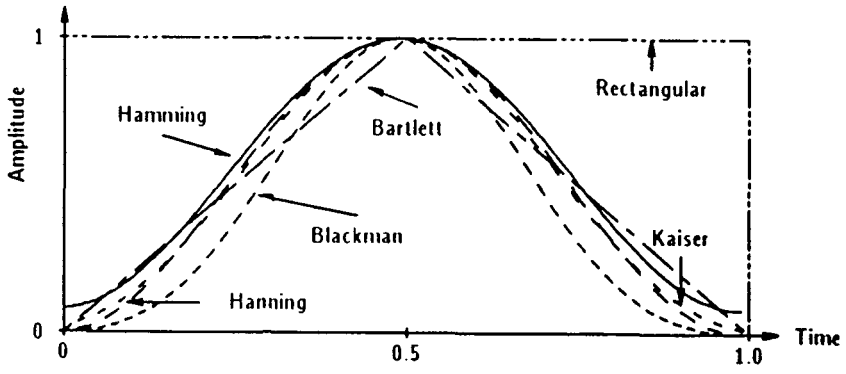


Figure 3.2. Fenêtres les plus utilisées [29].

3.2.1.2. La FFT (Fast Fourier Transform).

La FFT n'est pas autre chose qu'une transformée de Fourier effectuée sur une fenêtre d'un signal de parole. La transformée de Fourier est calculée ici à l'aide d'un algorithme optimisé pour le traitement numérique. La transformée de Fourier nous donne ce que nous appellons le spectre d'un signal. Comme nous l'avons mentionné plus tôt, l'analyse de Fourier suppose la stationnarité du signal, alors qu'un signal de parole n'est pas stationnaire. Il faut donc choisir des fenêtres étroites pour de bonnes performances de reconnaissance [29]. La FFT permet de générer un outil extrêmement utilisé en analyse de la parole, le spectrogramme. Un spectrogramme donne la transformée de Fourier en fonction du temps et de la fréquence. Un exemple de spectrogramme à bandes larges est donné à la figure 3.3. Les zones ombragées que l'on voit sur l'image représentent l'amplitude de la transformée de Fourier. Le spectrogramme de la figure 3.3 correspond à la phrase anglaise "The birch canoe slid on the smooth planks".

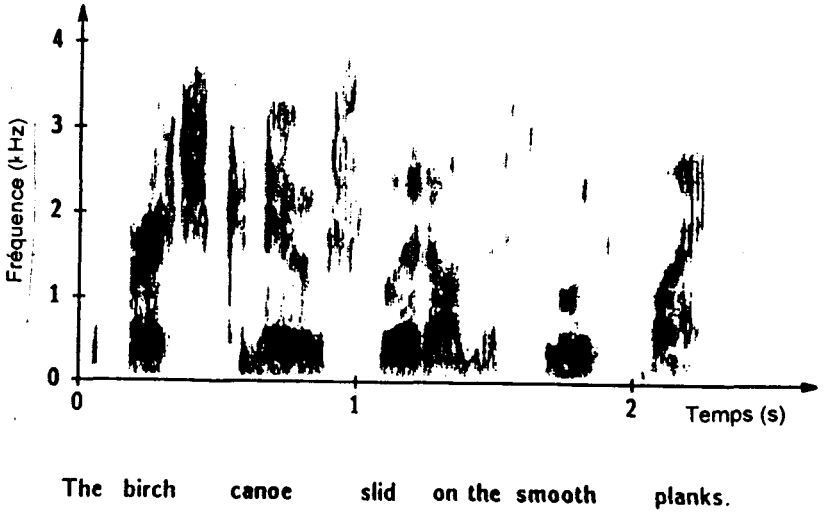


Figure 3.3. Un spectrogramme [29].

Un outil très semblable au spectrogramme est l'analyse cepstrale [29] pp. 226-231, qui, au moyen du cepstre du signal, donne d'autres informations utiles en reconnaissance de la parole.

3.2.1.3. L'analyse LPC (Linear Predictive Coding).

Une autre analyse supposant la quasi-stationnarité d'un signal est l'analyse LPC [29] pp. 336-379. Au fil des ans, le LPC est devenu l'analyse la plus utilisée dans les systèmes de reconnaissance de la parole. Le LPC permet d'obtenir des spectrogrammes et des cepstrogrammes, comme la transformée de Fourier, mais en utilisant une méthodologie différente.

3.2.1.4. Les analyses non-stationnaires.

Les analyses non-stationnaires, encore au stade expérimental, peuvent faire appel à ce qu'on nomme des opérateurs non-linéaires. Les analyses non-stationnaires sont, par définition, des analyses qui tiennent compte de la non-stationnarité des signaux de parole. L'analyse non-stationnaire que nous étudierons est l'analyse par démodulation. Comme elle fera l'objet de tout un

chapitre, nous nous bornerons à dire ici qu'elle permet d'extraire des paramètres impossibles à obtenir avec les analyses conventionnelles (FFT, LPC).

3.2.1.5. La programmation dynamique.

La méthode de programmation dynamique repose sur un algorithme d'anamorphose temporelle (AAT) (Dynamic Time Warping) [4]. Cet AAT, popularisé dans les années 70, a d'abord permis une reconnaissance de mots isolés. Environ dix ans plus tard, une méthode beaucoup plus performante et fonctionnant en temps réel est proposée [33]. Plusieurs techniques basées sur la programmation dynamique voient alors le jour.

L'algorithme d'alignement temporel sert à pallier aux fluctuations non linéaires d'un signal de parole dans le temps. En effet, pour un même mot prononcé deux fois par un même locuteur, on obtiendra deux signaux de même allure mais pas identiques. Un système basé sur la programmation dynamique tient compte des compressions et dilatations temporelles pour pouvoir décider que deux signaux désignent le même mot [25].

3.2.1.6. Les réseaux de neurones

Les réseaux de neurones constituent une approche indépendante des systèmes vus jusqu'à présent pour la reconnaissance de parole et la reconnaissance des formes en général. Comme nous consacrons le chapitre 5 en entier aux réseaux de neurones, nous ne nous attarderons pas ici à les décrire. Disons seulement qu'ils offrent une approche intéressante pour réaliser des systèmes de reconnaissance de parole performants et robustes au bruit.

3.2.1.7. Les modèles de Markov.

C'est l'équipe de Jelinek à IBM [3] qui a fait connaître en 1975 les modèles de Markov en reconnaissance des formes. Depuis, ces modèles sont très utilisés et se retrouvent dans pratiquement tous les systèmes de reconnaissance de parole commercialisés jusqu'à ce jour. Bien qu'une approche statistique pour la reconnaissance de la parole semblait un peu excentrique à l'époque, les résultats furent nettement supérieurs à tout ce qui se faisait alors, d'où leur grande popularité.

Un système de Markov est en réalité constitué d'une série d'états, représentants un signal sonore. Ces états sont reliés en séquences par des liens probabilistiques. On appelle "chaîne de Markov" une séquence d'états caractérisant un mot, un son ou une phrase. Lorsqu'on fait de la reconnaissance

de parole en utilisant un système Markovien, on choisit parmi plusieurs chaînes de Markov celle qui a la probabilité la plus grande de correspondre au mot ou à la phrase que l'on veut reconnaître. Le lecteur désireux de se familiariser avec les systèmes de Markov peut consulter [29] pp. 459-468 [34] [38].

Les systèmes de Markov, quoiqu'ayant de très bonnes performances avec plusieurs locuteurs, ont beaucoup de difficulté lorsqu'ils sont confrontés à un millieu bruité. De plus, il faut beaucoup de temps pour "faire apprendre" à reconnaître plusieurs mots à un système de Markov.

3.3. Conclusion.

La principale difficulté à laquelle font face les chercheurs aujourd'hui est la reconnaissance de parole continue pour un grand vocabulaire. En effet, s'il est relativement facile de reconnaître des mots isolés, il en est tout autrement lorsque l'on doit reconnaître des mots prononcés de façon normale, c'est-à-dire intégrés dans un message de parole. Les modèles statistiques de Markov ont été conçus, initialement, pour la reconnaissance de mots isolés. Bien que plausible, un système de reconnaissance de parole continue par modèles statistiques de Markov devient extrêmement compliqué. Les modèles de Markov sont ce qu'il y a de plus performant sur le marché aujourd'hui. Ils fonctionnent donc mieux que les autres systèmes de reconnaissance que nous avont décrits précédemment.

Le système que nous proposons ici pourrait constituer une autre alternative aux systèmes de Markov.

4

L'analyse par démodulation

Nous verrons dans ce chapitre ce qu'est l'analyse par démodulation et en quoi elle est utile pour traiter des signaux de parole. Nous verrons aussi comment nous appliquons l'analyse par démodulation de façon concrête pour le traitement de la parole.

Lorsque nous travaillons sur des signaux de parole, il est important d'isoler et d'accentuer l'information qui peut être utile pour distinguer un phone d'un autre, et de neutraliser l'information redondante ou inutile. Le principal problème auquel les chercheurs sont confrontés est de savoir quels sont les paramètres importants à extraire d'un signal de parole. On doit, lorsqu'on fait de la reconnaissance de parole, rendre les signaux les plus uniformes et indépendants du locuteur possible. Lorsqu'on utilise les analyses conventionnelles vues au chapitre 3, on peut obtenir de l'information spectrale sur les signaux. Cette information est relativement dépendante du locuteur. Cependant, en se servant des analyses conventionnelles, une partie de l'information temporelle est définitivement neutralisée. L'analyse par démodulation [36] permet de tirer parti de cette information temporelle.

L'analyse par démodulation utilise ce qu'on appelle des opérateurs non-linéaires [36] [22] [18]. Ces opérateurs, contrairement aux analyses conventionnelles que nous avons survolé au chapitre 3, respectent la non-stationnarité des signaux de parole. Pour se familiariser avec les opérateurs non linéaires, les travaux de Teager [19] [20] sont indéniablement à consulter. L'intérêt de cette approche, quoique certain, n'a pas encore été vérifié sur des systèmes concrets. Ce travail se propose, entre autre, de remplir cette lacune.

4.1. Observations de base.

La recherche ayant comme sujet la démodulation de la parole est motivée par des observations montrant que le cerveau humain possède des cellules spécialisées dans la détection de modulation d'amplitude (AM) et de modulation de fréquence (FM) [43]. Il est intéressant de constater que cette détection de modulation s'observe aussi chez les animaux [12] [41]. Ces travaux montrent que la détection de modulation AM et FM constitue une partie importante du système auditif.

L'approche que nous étudierons ici considère le phénomène naturel de démodulation automatique du signal par des cellules hautement spécialisées. Cette démodulation se fait avant le post-traitement par d'autres réseaux de neurones biologiquement moins bien connus. Nous modéliserons le phénomène de démodulation par ce que nous appellerons des opérateurs non-linéaires. Le post-traitement sera laissé au réseau DYSTAL [1], que nous présenterons plus loin.

Nous rappellerons ici que notre hypothèse de travail est que le phénomène de démodulation des signaux sonores étudié ici est une faculté innée de l'appareil auditif humain et par conséquent n'est le résultat d'aucun apprentissage. Par contre, le traitement des signaux obtenus par démodulation est, lui, le résultat d'un apprentissage.

Cette opération de traitement par opérateurs non-linéaires est combinée avec un banc de filtres pour donner des images 3D, que nous étudierons plus loin.

4.2. Opérateur Dyn, banc de filtres et images 3D.

L'opérateur non-linéaire que nous nous proposons de présenter dans cette section est l'opérateur Dyn [35]. L'opérateur Dyn calcule le produit de l'enveloppe d'un signal par sa dérivée en tenant compte de facteurs physiques et mécaniques.

Étant donné que, dans l'analyse que nous utilisons, l'opérateur Dyn est étroitement lié à un banc de 24 filtres et à une visualisation 3D, nous optons pour présenter tous ces éléments de notre système en même temps. Voyons l'exemple qui suit. D'abord, deux considérations:

1/ En théorie, un signal de parole voisée pourrait être considéré comme une somme finie d'ondes sinusoïdales pures, de fréquence et de phase différentes. Considérons le signal suivant:

$$s(t) = A_1\cos(\omega_1 t) + A_2\cos(\omega_2 t) + \ldots + A_n\cos(\omega_n t)$$

Ce signal pourrait représenter très grossièrement un son voisé, dans lequel tous les ω_i sont multiples entre eux.

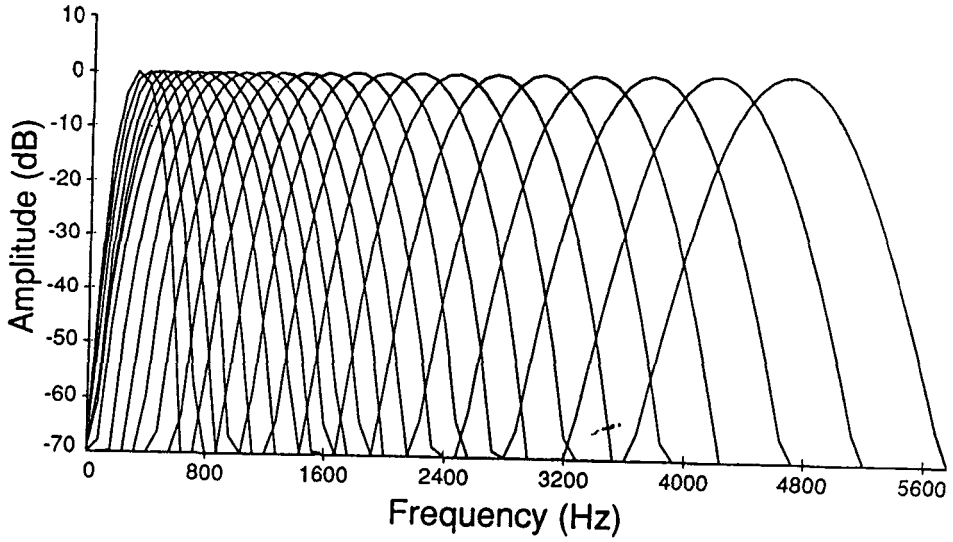
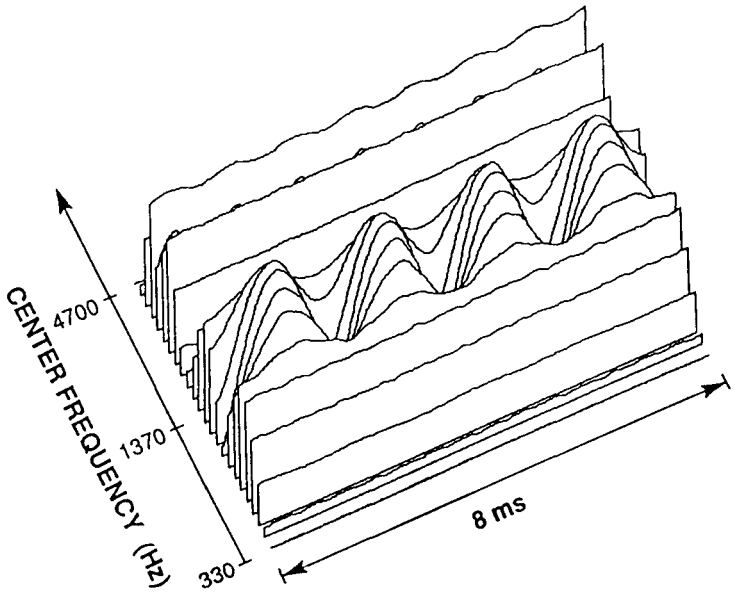


Figure 4.1. Réponses en fréquence du banc de filtres [36].

2/ En plus de l'opérateur Dyn, notre système est constitué d'un banc de 24 filtres dont la réponse en fréquence est donnée à la figure 4.1. Ce banc de

filtre peut être vu comme comme une approximation linéaire et très grossière de la cochlée (voir chapitre 2).



Envelope output (A(t)) for f(t).
$f(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin(2\pi f_3 t) + \sin(2\pi f_4 t)$;

$f_1 = 700$ Hz; $f_2 = 1200$ Hz; $f_3 = 2500$ Hz and $f_4 = 3400$ Hz.

Figure 4.2. Enveloppe du signal [36].

Prenons, comme exemple, un signal constitué de la somme de 4 fréquences:

f1 = 700 Hz

f2 = 1200 Hz

f3 = 2500 Hz

f4 = 3400 Hz

Nous avons là un signal de parole hypothétique, dont la sonorité serait près du schwah /^/ du mot anglais "but".

La figure 4.2 montre les enveloppes obtenues par transformée de Hilbert à partir des sorties de notre banc de filtres pour le signal constitué des 4 fréquences. Nous avons là une image 3D. L'opérateur Dyn donne, en fait, le produit de l'enveloppe d'un signal par la dérivée de l'enveloppe, dans un format semblable à l'image de la figure 4.2. C'est ce produit qui sera traité par notre réseau de neurones.

Sur la figure 4.2, nous avons les réponses en fréquence sur un axe représentant les enveloppes des sorties données par les 24 canaux de notre banc de filtres.

Les deux autres axes de nos images 3D représentent le temps et l'amplitude des enveloppes. Sur la figure 4.2, nous observons une série d'ondes sinusoïdales entrecoupées de bandes d'amplitudes continues. Les filtres pour lesquels les bandes d'amplitude sont continues permettraient de mesurer directement les composantes du signal d'entrée, si nous avions plus de filtres dans notre banc de filtres. Ceci s'explique par le fait que le calcul de l'enveloppe d'un signal met en évidence les amplitudes (A_n) du signal que nous avons examiné précédemment (pour les sorties à enveloppe constante).

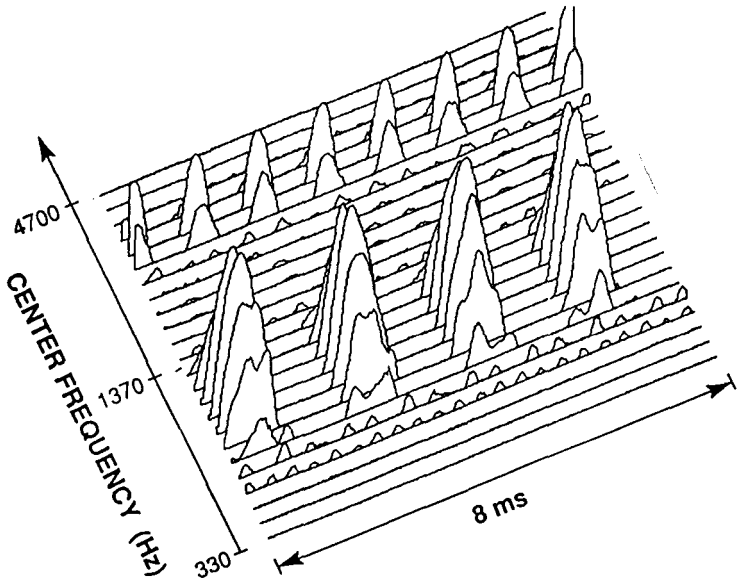
Nous nous intéresserons plutôt aux oscillations entre les bandes d'amplitudes continues. Ces oscillations sont une mesure directe des valeurs (f2 - f1) et (f4 - f3). Ceci s'explique encore par le fait que, lorsque plusieurs fréquences sont détectées par un même canal du banc de filtres, et qu'on calcule l'enveloppe de ce signal, on fait ressortir les battements provoqués par ces composantes. Il faut, pour qu'un signal soit détecté par un filtre, que la fréquence de ce signal soit comprise à l'intérieur de la bande passante du filtre.

Lorsque deux signaux purs (ondes sinusoïdales, par exemple) ont une fréquence comprise à l'intérieur de la bande passante d'un filtre, et qu'on calcule l'enveloppe de la sortie du filtre, on obtient une mesure de la différence de fréquence entre ces signaux (comme sur les canaux près de 1300Hz de la figure 4.2).

Sur la figure 4.2, f3 - f2 n'est pas vue parce que f3 et f2 ne se retrouvent pas sur la bande passante d'un seul filtre. Aucun filtre du banc de filtres n'a une largeur de bande passante suffisamment large pour "voir" f3 et f2 simultanément. Cet exemple pourrait caractériser grossièrement le système auditif.

L'information qui nous intéresse ici est non pas la mesure des fréquences de départ f1 ... f4 mais la mesure de la différence de ces fréquences (f2 - f1), (f4 - f3), etc... La raison est que la mesure directe des fréquences ne peut se faire avec précision, dû au petit nombre de filtres disponibles. La figure 4.3 représente la sortie rectifiée de l'opérateur Dyn. C'est la dérivée de l'enveloppe des sorties du banc de filtres (figure 4.2) multipliée par les enveloppes elles-

mêmes. La figure 4.3 constitue un exemple d'entrée pour notre réseau de neurones, quoique dans les expériences présentées dans ce mémoire, on ne considère pas le signal rectifié.



Low-pass-filtered and rectified Dyn output for f(t)
$f(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t) + \sin(2\pi f_3 t) + \sin(2\pi f_4 t)$;
$f_1 = 700$ Hz; $f_2 = 1200$ Hz; $f_3 = 2500$ Hz and $f_4 = 3400$ Hz.

Figure 4.3. Dérivée des enveloppes multipliées par les enveloppes du signal de sortie des filtres et rectification [36].

4.3. Analyse des signaux de parole.

Pour faire suite à l'exemple de la section 4.2, nous montrerons ici comment l'analyse par démodulation est utilisée sur des signaux de parole réels. L'analyse d'un signal de parole se fait exactement comme dans l'exemple de la section 4.2. La différence est que les valeurs f1, f2, f3 et f4 de notre exemple sont ici les formants. L'information que nous allons chercher ici est la différence de fréquence entre certains formants, qui est en fait la mesure la plus précise qui peut être faite avec notre système.

4.3.1. Filtrage.

La version dont nous nous servons pour notre pré-traitement comprend donc un banc de 24 filtres centrés de 330 Hz à 4700 Hz. Ces filtres simulent partiellement l'analyse de fréquence accomplie par la cochlée. Ce sont des filtres exponentiels avec les largeurs de bande rectangulaires équivalentes proposées par Patterson [30] et Moore et Glasberg [24].

La sortie donnée par chacun de ces filtres est un signal passe-bande centré autour de f_i. La fréquence f_i est la fréquence centrale du canal i. Le signal obtenu ainsi peut être considéré comme modulé en amplitude et en phase [9] avec une fréquence porteuse de f_i.

4.3.2. Démodulation.

Dans notre système, nous utilisons l'opérateur Dyn (sans rectification) décrit plus haut pour effectuer la démodulation. On peut considérer [36] que la sortie donnée par l'opérateur Dyn, filtrée passe-bas, est égale aux fluctuations de l'amplitude du signal au carré, d/dt $[A_i^2(t)]$. Les données de parole avec lesquelles les opérateurs non-linéaires travaillent sont des données échantillonnées à 32 kHz. Cette fréquence d'échantillonnage nous permet d'avoir le nombre d'échantillons nécessaire à l'intérieur d'un filtre pour effectuer la démodulation. On peut voir sur les figures 4.4 à 4.6 les images 3D données par l'opérateur Dyn. La figure 4.4 représente une tranche de 14.5 ms du son /a/ prononcé par un homme. La figure 4.5 représente 13.25 ms du son /i/ prononcé par un homme et la figure 4.6, quant à elle, représente 13.5 ms du son /i/, cette fois prononcé par une femme.

4.3.3. Les images 3D.

Regardons plus attentivement les images 3D obtenues avec l'opérateur Dyn. Nous voyons, à la figure 4.4, la modulation typique pour le son /a/. Si nous regardons attentivement, nous voyons deux séries de trois pics. Nous avons deux fois la répétition du "patron" prononcé, donc nous avons deux impulsions glottales dans cette image (voir chapitre 2). L'information qui nous intéresse ici est non l'impulsion glottale, dépendante du locuteur, mais l'intervalle de temps

entre les pics d'une même série. Par exemple, sur la figure 4.4, l'intervalle entre les deux premiers pics (encerclés sur la figure) est une mesure directe de 1 / (f2 - f1), ou l'inverse de la différence entre les deux premiers formants (voir analogie avec l'exemple de la section 4.2).

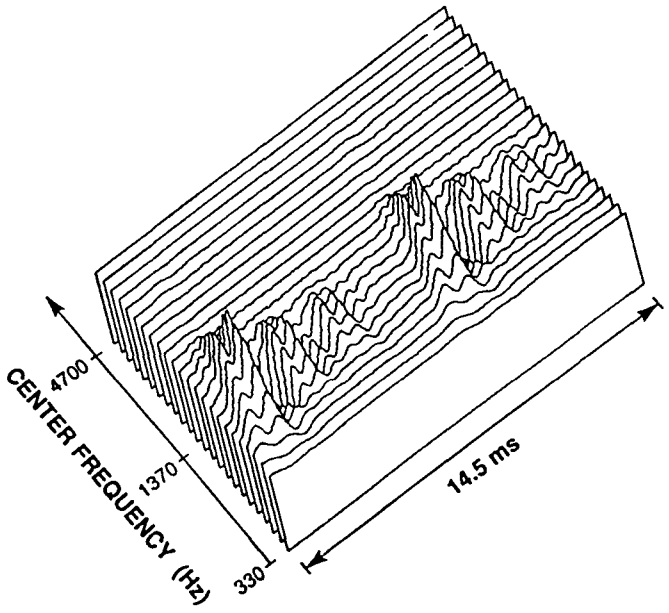


Figure 4.4. Le son /a/, homme [37].

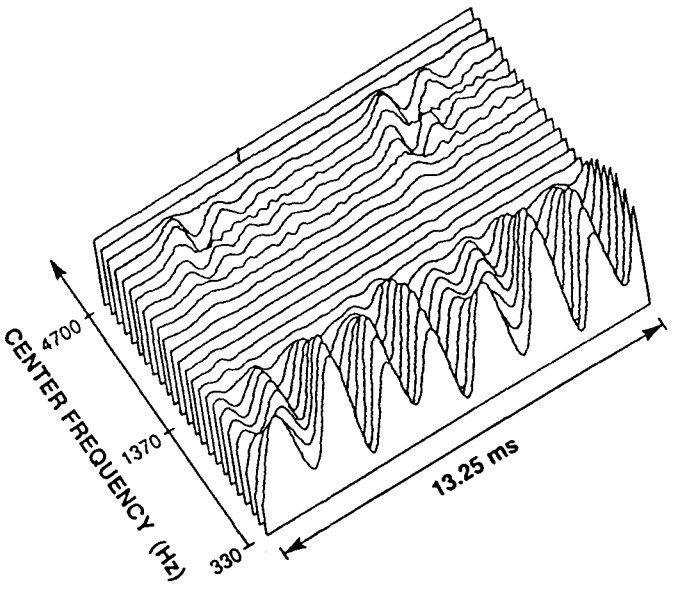


Figure 4.5. Le son /i/, homme[37].

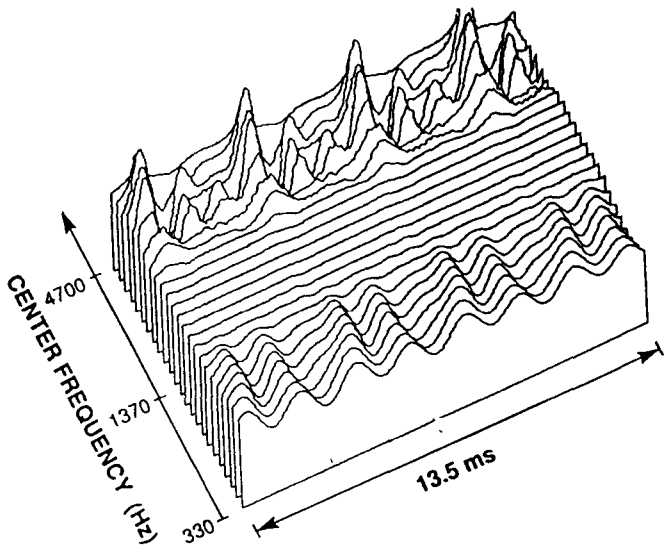


Figure 4.6. Le son /i/, femme [37].

Nous pouvons tirer le même genre d'information des figures 4.4 et 4.5. Pour le son /i/, les paramètres à considérer sont F4-F3 et F3-F2 dans les hautes fréquences. L'information est cependant moins évidente à reconnaître ici pour un observateur inexpérimenté. Il est important de retenir que, sur l'échelle temporelle constituant une dimension de nos images 3D, nous retrouvons des mesures directes des différences de fréquences entre les formants, lorsque ceux-ci sont suffisamment proches l'un de l'autre pour être englobés à l'intérieur

d'un même filtre (largeur de bande des filtres finie, voir figure 4.1), encore une fois en analogie avec le système auditif de l'humain.

En plus des valeurs des différences de fréquences entre les formants, les images 3D conservent l'information spectrale classique [29] pp. 64 - 78. En effet, sur l'échelle de fréquence (banc de filtres), variant de 330 à 4700 Hz, nous retrouvons une distribution spectrale qui donne aussi des informations utiles pour la discrimination entre les sons, similairement à ce qu'on obtient par FFT, par exemple [29], pp. 204 - 242.

4.4. En guise de conclusion.

Nous ajouterons ici que l'opérateur Dyn, en plus de l'information spectrale conventionnelle, nous donne des mesures des modulations AM présentes dans les signaux de parole voisés. Ces mesures peuvent être considérées sous forme spectrale et sous forme temporelle grâce aux images 3D.

Cette information temporelle pourrait aider à reconnaître des sons qui présentent une difficulté majeure pour les systèmes de reconnaissance de parole utilisés aujourd'hui. De plus, l'information au niveau temporel donnée par Dyn n'est pas conservée par les systèmes de reconnaissance de parole utilisés aujourd'hui. En résumé, l'opérateur Dyn pourrait s'insérer dans un modèle de l'appareil auditif humain inspiré d'observations biologiques.

Sur ce, nous référerons le lecteur désireux d'approfondir ses connaissances sur les analyses des signaux de parole les plus utilisées et sur la technologie de la reconnaissance de la parole à [29], un ouvrage général et très complet faisant, à notre avis, le tour de la question.

5

Reconnaissance de parole par réseaux de neurones

Nous aborderons ici dans un premier temps les réseaux de neurones en général, puis nous étudierons différentes architectures à réseaux de neurones appliquées à la reconnaissance de la parole.

Les architectures à réseaux de neurones sont de plus en plus utilisées dans plusieurs domaines de pointe de la science et de l'ingénierie. Citons entre autres la planification financière, l'analyse et traitement de signaux numériques, la robotique et plusieurs autres domaines. Un des domaines les plus importants de l'analyse et traitement des signaux numériques est la reconnaissance de la parole.

Les réseaux de neurones sont un domaine de la science en pleine évolution. Le chercheur s'apercevra très vite s'il consulte plusieurs ouvrages sur le sujet que bien que certains termes de base sont bien définis, d'autres plus particuliers le sont moins bien. Un réseau de perceptrons utilisant un algorithme à rétropropagation peut être appelé un réseau de perceptrons dans certaines références, ou un réseau à rétropropagation dans d'autres. Nous nous excusons d'avance pour les erreurs qui pourraient occasionner des confusions et qui ne manqueront pas de se produire dans ce chapitre.

5.1. Introduction.

Un réseau de neurones est en quelque sorte un circuit électronique capable de s'auto-programmer, si on lui présente des informations pertinentes, pour effectuer une tâche précise. Comme nous ne ferons ici qu'un bref tour d'horizon des architectures à réseaux de neurones adaptées à la reconnaissance de la parole, le lecteur désireux d'approfondir ses connaissances sur ce sujet pourra se référer à un excellent article [44] de Bernard Widrow.

Nous croyons utile ici d'aborder une question importante, à savoir comment transmettre l'information sonore à un réseau de neurones, ou plus exactement quel signal lui présenter.

Un réseau de neurones peut traiter de l'information échantillonnée, c'est-à-dire une série de nombres qui déterminent un signal. Par exemple, à un intervalle de temps déterminé, on va mesurer la valeur d'un signal et cette valeur constituera un échantillon qui pourra être traité par un réseau de neurones.

La plupart du temps, les échantillons sont présentés par groupe à l'entrée d'un réseau de neurones. Ces groupes sont déterminés par des fenêtres qui peuvent soit être contiguës ou soit se chevaucher (voir chapitre 2). Il est évident que le choix de la taille et du chevauchement des fenêtres aura des répercussions sur la sortie que nous obtiendrons. [42] donne un bon aperçu général de la façon de traiter l'information que l'on présente à un réseau de neurones.

En plus de choisir une façon de présenter le signal au réseau de neurones, le chercheur devra aussi choisir le signal à présenter. Le signal le plus simple à présenter est sans doute le signal-parole échantillonné pur, qui donne de très piètres résultats lorsque traité par des réseaux de neurones. La raison en est simple. La parole contient beaucoup d'information inutile et de redondances dont les humains font facilement abstraction, mais qui posent des problèmes autrement plus ardus aux ordinateurs (voir chapitres précédents).

Une bonne façon de procéder est donc d'effectuer un pré-traitement sur le signal avant de le présenter à un réseau de neurones. La façon d'effectuer ce pré-traitement et le choix des paramètres à prioriser fait encore aujourd'hui l'objet de controverses et le chercheur pourra se familiariser avec les différentes tendances en cours en consultant la documentation appropriée. Nous pouvons toutefois ajouter que, en règle générale, les structures formantiques (caractéristiques reliées aux sons lorsqu'il y a vibration des cordes vocales) sont à conserver.

Il est à noter que les conditions énumérées ci-haut font qu'il est plus difficile pour un réseau de neurones de différencier les sons non voisés (certaines consonnes) que les sons voisés (les voyelles, par exemple). Les références [29] et [6] font un tour d'horizon assez complet de l'état des connaissances sur ce sujet.

5.2. Les composantes de base des architectures à réseaux de neurones.

Avant de discuter des différentes architectures employées aujourd'hui dans la conception de systèmes de reconnaissance de parole par réseaux de neurones, il est impératif de se familiariser avec les composantes de base qui constituent tout système à réseau de neurones.

5.2.1. Le neurone électronique.

Le neurone électronique est l'élément de base qui constitue tout réseau de neurones. Il est constitué de diverses lignes d'entrées (X_0 à X_n) ayant chacune un poids (valeur par laquelle on multiplie l'entrée). Le concept de poids est très important car il est directement relié à celui d'apprentissage. En effet, lorsqu'un réseau "apprend", il ajuste des poids associés aux lignes d'entrée de ses neurones de façon à ce que le réseau se comporte de manière adéquate selon diverses excitations.

Le neurone électronique est aussi muni d'un sommateur qui fait la somme de ses entrées. Le sommateur peut avoir à sa sortie une fonction de transfert ou propager l'information telle quelle. La figure 5.1 représente un neurone électronique. Les poids (W_0 à W_n) ainsi que le traitement possible de la sortie par une fonction de transfert y sont représentés.

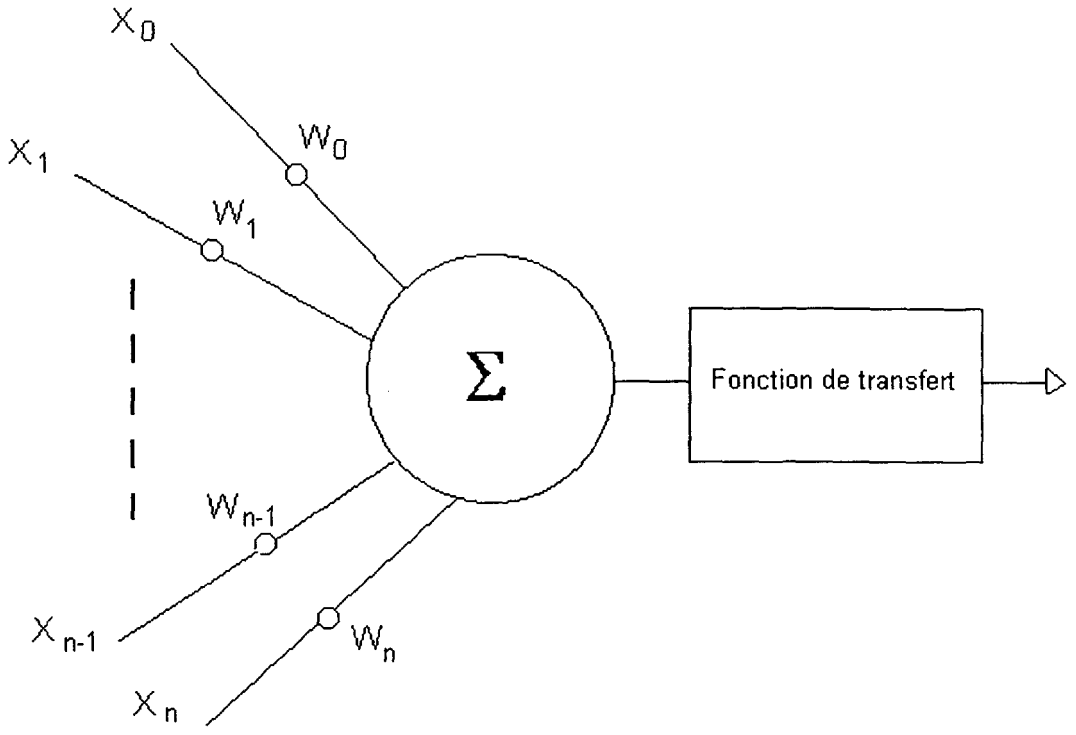


Figure 5.1. Un neurone électronique.

5.2.2. Le perceptron binaire.

Il fut inventé par Frank Rosenblatt vers le millieu des années 50, et fut à l'origine de l'intérêt qu'on porte aujourd'hui aux machines capables d'apprendre. Il s'agit en fait d'un neurone électronique dont la fonction de transfert est la suivante (voir figure 5.2): La sortie Y_j prend la valeur 1 si la sommation de toutes

les entrées multipliées par leurs poids respectifs est supérieure à 0. Autrement, la valeur de sortie est nulle.

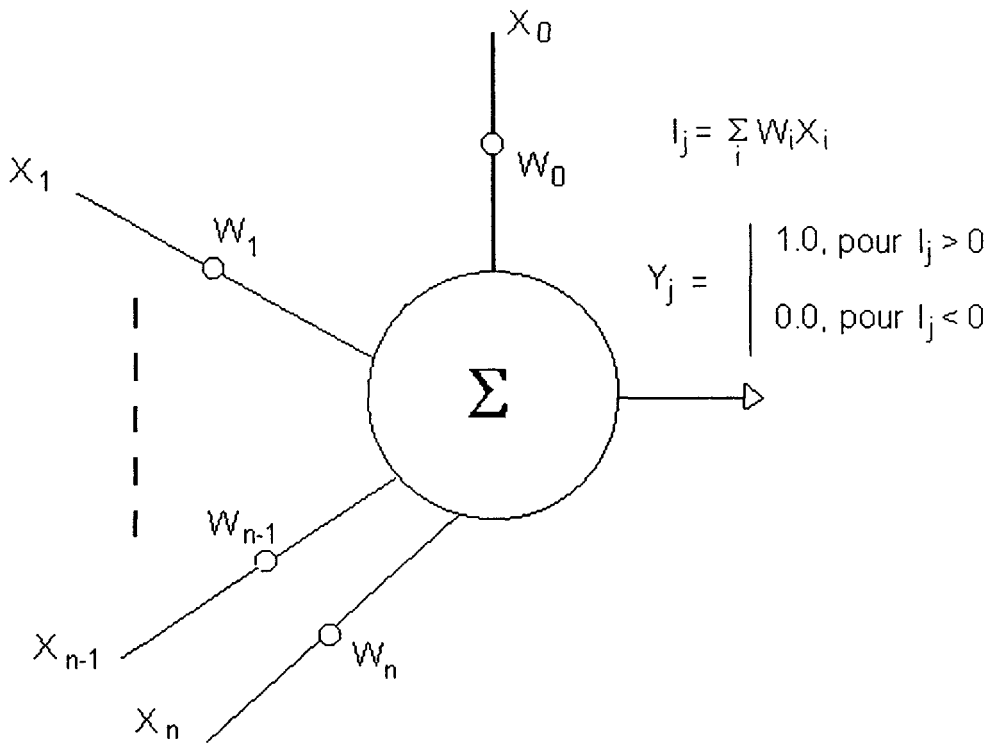


Figure 5.2. Perceptron binaire.

À titre d'information, on peut souligner les travaux de Minsky et Papert [23], qui firent remarquer la limite de discrimination des perceptrons, ce qui fit stagner la recherche dans ce domaine pendant quelques années.

5.2.3. Adaline et madaline.

Similaire au perceptron, l'adaline possède cependant une structure un peu plus complexe. En effet, on y a ajouté une boucle de retour qui sert à l'implantation d'un algorithme d'apprentissage, et qui sert à ajuster les poids.

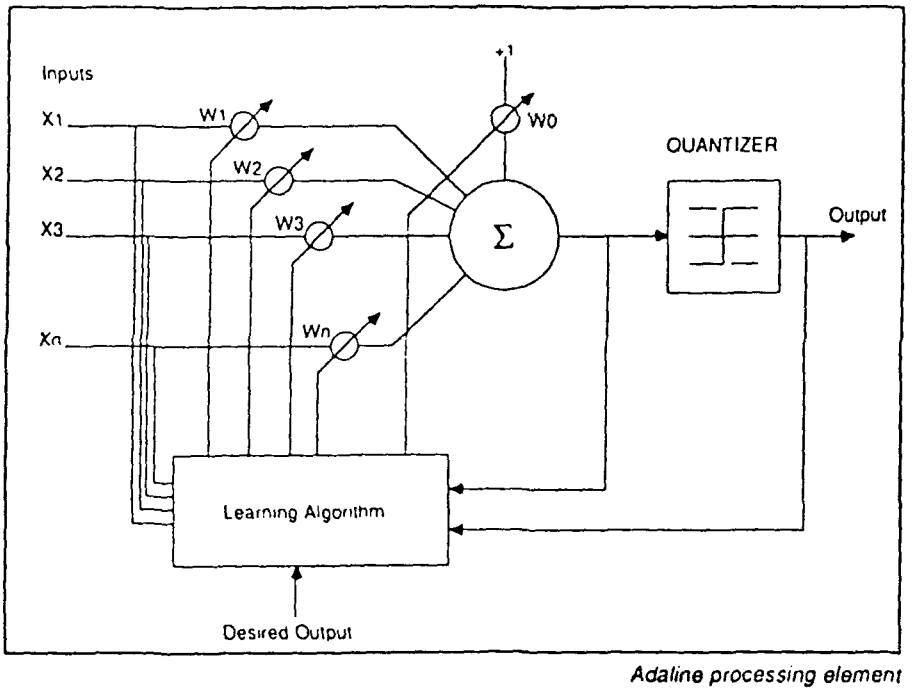


Figure 5.3. Un adaline [26].

Il est à noter que la valeur des poids doit être fixée au hasard initialement, ceci devient évident lorsqu'on se familiarise avec les différentes règles d'apprentissage utilisées pour les réseaux d'adalines. Comme il n'est pas notre but ici d'en faire la description, nous référerons le lecteur à un bref aperçu de la façon de procéder pour l'apprentissage d'un adaline [26].

La fonction de transfert à la sortie d'un adaline donne la sommation des poids multipliés par leurs entrées respectives. Un quantificateur donnant soit la valeur -1 soit la valeur 0 peut aussi être ajouté. La figure 5.3 représente un adaline. On peut remarquer que ce modèle présente beaucoup de similitudes avec le perceptron.

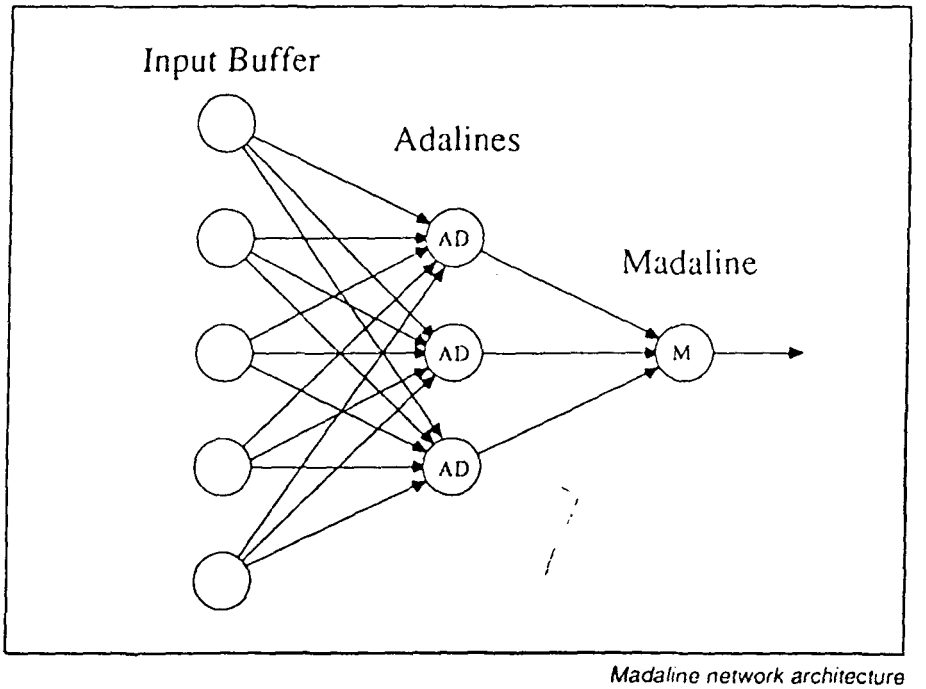


Figure 5.4. Un madaline [26].

Le madaline est une architecture comprenant plusieurs adalines en parallèle et une seule sortie, le madaline. Des réseaux à base de madalines comprennent plusieurs madalines et leurs adalines. La figure 5.4 représente un madaline. Dans un madaline, si la moitié ou plus des adalines d'entrée a pour valeur de sortie +1, la sortie donnée par le madaline sera +1. Autrement, elle sera à -1.

5.2.4. Autres éléments.

D'autres éléments légèrement différents de ceux décrits plus haut sont aussi utilisés dans des configurations particulières, comme l'élément de Hopfield ou l'élément d'un réseau à rétropropagation. Nous retrouvons aussi toute la série d'éléments apparentés au perceptron et à l'adaline mais fournissant une fonction de sortie différente. Les fonctions les plus couramment utilisées sont la sigmoïde, la tangente hyperbolique et la sinusoïde.

Les éléments fournissant ce type de fonction à leur sortie sont utilisés dans la plupart des architectures à réseaux de neurones récentes.

5.3. Les architectures à réseaux de neurones pour la reconnaissance de la parole.

Plusieurs réseaux de neurones effectuant de la reconnaissance de parole ont vu le jour ces dernières années, avec des résultats qui peuvent être qualifiés de concluants. Ces réseaux ont différentes architectures, en fait chacun d'eux est unique, mais nous pouvons cependant les classer par catégories. Nous nous proposons ici de voir d'abord l'architecture de base des différentes catégories, pour étudier ensuite quelques exemples concrets.

Les architectures étudiées ici n'ont pas toutes fait l'objet de la même attention de la part des chercheurs, aussi une comparaison est très difficile à faire. Nous n'émettrons donc pas de jugement sur la meilleure architecture à utiliser, car nous croyons que l'état des recherches en reconnaissance de parole par réseaux de neurones n'est pas encore assez avancé pour se prononcer. Des modèles hybrides ont même été proposés, qui allient les réseaux de neurones à la programmation dynamique (méthode d'optimisation), ou à des modèles utilisant une approche stochastique ou probabilistique [15], [39] et [28].

Le chercheur désireux d'approfondir et d'améliorer son travail devrait donc effectuer plusieurs essais d'apprentissage avec plusieurs réseaux pour voir celui qui est le plus approprié pour le système qu'il veut construire.

Comme on le verra dans les exemples concrets étudiés, en plus d'une disparité d'architecture des réseaux, on a une disparité des tâches accomplies. En effet, certains systèmes fonctionnent avec un seul locuteur, d'autres ne

distinguent que quelques sons, d'autres encore nécessitent un fractionnement préalable des phrases parlées en phonèmes. C'est pourquoi, comme nous l'avons mentionné précédemment, il est encore trop tôt pour une "recette infaillible et universelle", et nous nous bornerons à un bref tour d'horizon.

5.3.1. Le réseau de perceptrons à simple couche.

Peu utilisé, car avec ce type d'architecture la discrimination non linéaire est impossible [44]. Des systèmes de reconnaissance pouvant fonctionner avec quelques sons ont cependant été réalisés avec cette architecture.

La figure 5.5 représente un réseau de perceptrons à simple couche. On y voit la couche d'entrée qui reçoit les échantillons du signal et la couche de sortie constituée de perceptrons. Nous remarquerons que sur tout réseau de neurones la couche d'entrée est en quelque sorte un registre et n'effectue aucun traitement du signal. Elle se borne à le propager vers les neurones de la couche suivante.

Le réseau de perceptrons à simple couche, d'intérêt purement académique, ne mérite pas que nous nous y attardions d'avantage, car un système complet de reconnaissance de parole utilisant uniquement cette architecture est impossible à réaliser dans la pratique.

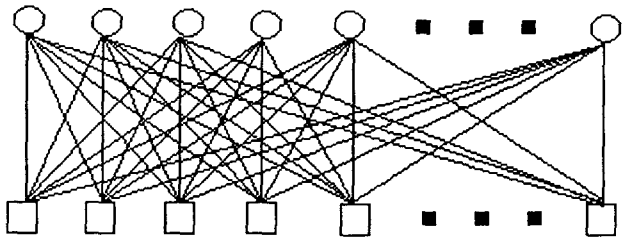


Figure 5.5. Réseau de perceptrons simple couche.

5.3.2. Le réseau de perceptrons à plusieurs couches.

Ce réseau, très intéressant, comporte deux couches ou plus de perceptrons ainsi qu'une couche d'entrée, donc trois couches au minimum. Un réseau de perceptrons à trois couches est illustré à la figure 5.6. On y remarque une couche d'entrée, une couche de sortie et une couche cachée.

Il est à noter que le nombre d'éléments sur chaque couche est arbitraire et dépend du problème qu'on a à résoudre. Ceci soulève le problème de la ou des couches cachées. En effet, s'il est assez facile pour le chercheur de déterminer combien d'éléments d'entrée ou de sortie il utilisera pour son application, c'est un tout autre problème de savoir combien d'éléments placer dans la ou les couches cachées ou simplement de savoir combien de couches cachées placer dans son architecture.

Là encore, il n'y a pas de recette magique, et l'expérience se révèle supérieure à la théorie. Le chercheur procédera donc à plusieurs essais pour optimiser son réseau, il évaluera la rapidité avec laquelle celui-ci apprend selon le nombre d'éléments cachés qu'il possède.

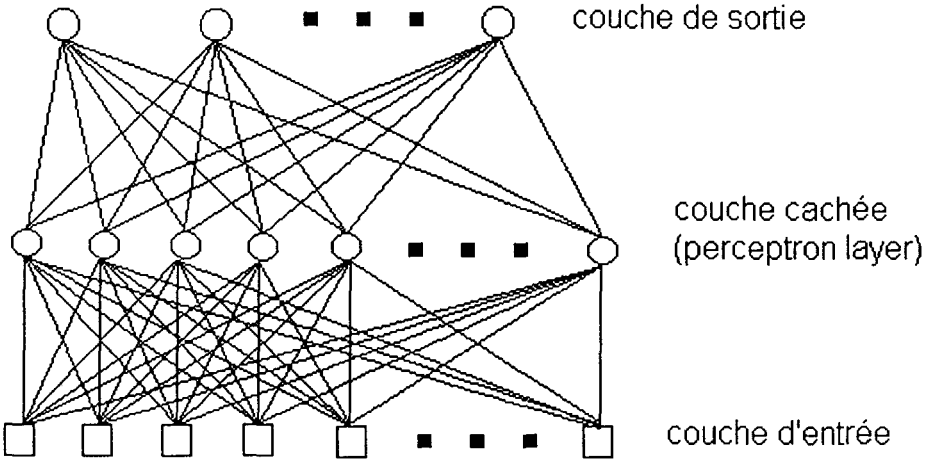


Figure 5.6. Réseau de perceptrons multicouche.

Selon certains chercheurs français [14] le réseau de perceptrons serait moins précis pour la reconnaissance de parole par rapport aux machines de Boltzmann, que nous étudierons plus loin, mais aurait un certain avantage au niveau de la rapidité. Cette étude se borne cependant à des systèmes reconnaissant des voyelles seulement et il est très hasardeux de se prononcer dans le cas d'un système plus élaboré.

Nous terminerons en disant que la majorité des systèmes élaborés jusqu'à aujourd'hui utilisent des réseaux de perceptrons multicouche avec un algorithme d'apprentissage à rétropropagation, dont nous reparlerons plus loin.

5.3.3. Le réseau d'adalines.

Il est constitué d'une couche d'entrée, d'une couche centrale (couche adaline) et d'une couche de sortie. Comme la couche d'entrée, la couche de sortie est simplement un registre où se propage l'information. Il y a toujours le même nombre d'éléments à la couche de sortie qu'à la couche centrale.

La figure 5.7 représente un réseau d'adalines. Ce type d'architecture n'est pas utilisé pour la reconnaissance de la parole.

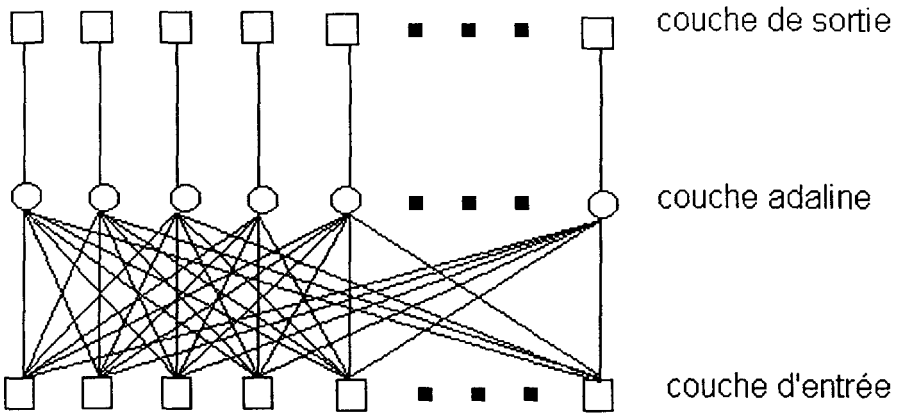


Figure 5.7. Réseau d'adalines.

5.3.4. Le réseau de madalines.

Le réseau de madalines est constitué de trois couches, comme le réseau d'adalines, mais la troisième couche est constituée de madalines, neurones reliés de façon symétrique à la couche centrale adaline. La différence entre un réseau de madalines et un réseau de perceptrons multicouche avec une seule couche cachée est en fait cette couche cachée. Elle est constituée d'adalines dans le réseau de madalines et de perceptrons dans le réseau de perceptrons multicouche.

Il est plus facile de visualiser la configuration d'un tel réseau en regardant la figure 5.8. On peut voir que les madalines de la troisième couche sont reliés aux adalines de la deuxième couche alternativement.

Encore là, nous n'avons pas trouvé d'exemples concrets implémentant un réseau de madalines dans un système de reconnaissance de la parole, mais nous croyons qu'il serait intéressant d'étudier cette possibilité. Une des limitations de ce réseau pour la reconnaissance de la parole est son aspect réponse binaire.

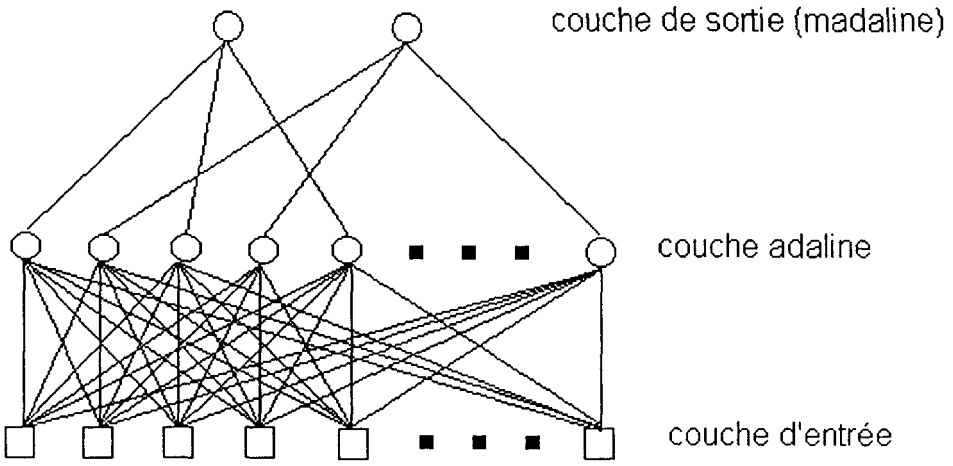


figure 5.8. Réseau de madalines.

5.3.5. Les réseaux de Hopfield.

Les architectures vues jusqu'à présent sont ce qu'on appelle des architectures à propagation avant. Une fois les poids ajustés, la sortie d'un élément quelconque ne retourne jamais à l'entrée d'un élément appartenant à la même couche ou à une couche inférieure. Ceci n'est plus le cas dans les architectures à réseaux de Hopfield.

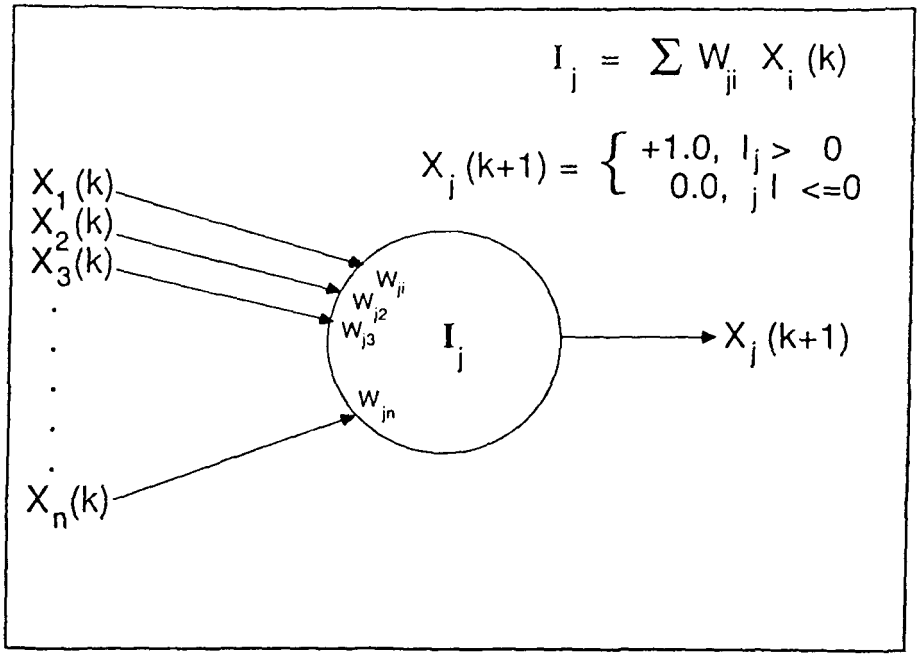


Figure 5.9. Un neurone de Hopfield [26].

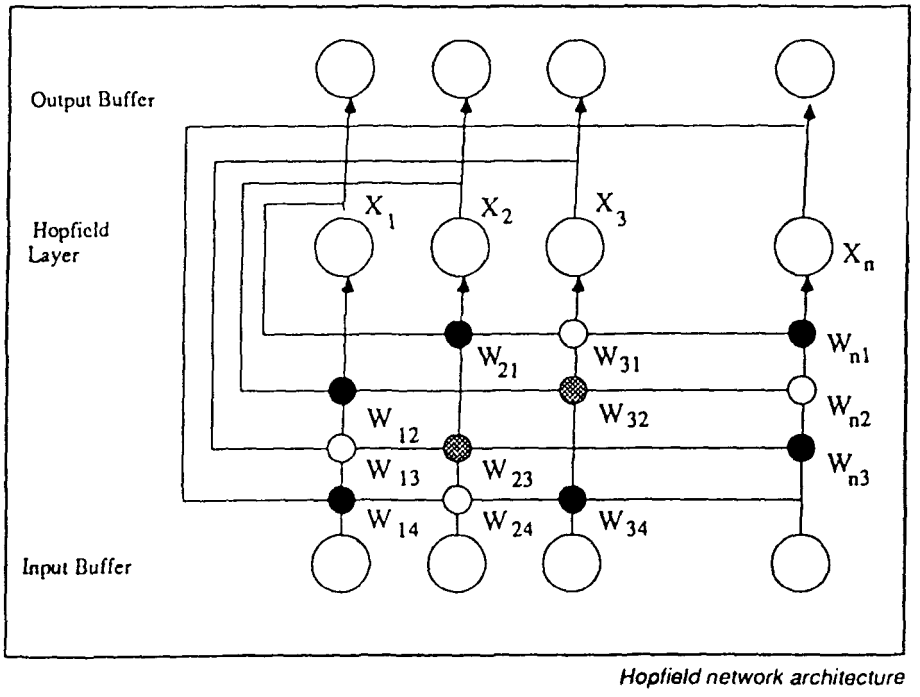


Figure 5.10. Architecture d'un réseau Hopfield [26].

Nous nous servons ici de l'élément de Hopfield. C'est une unité similaire au perceptron, même si la non-linéarité n'est pas la même, et on pourra faire la comparaison en regardant la figure 5.9. La différence principale est que les éléments de Hopfield sont faits pour être chacun connectés à tous les autres.

La figure 5.10 montre un réseau de Hopfield. On y remarque la configuration spéciale des éléments. Comme nous l'avons mentionné, la sortie de chaque élément de la couche Hopfield est connectée aux entrées de tous les autres sauf lui-même. On peut apercevoir sur cette figure les poids associés à chaque entrée d'élément. Par exemple, la ligne de sortie de l'élément X_3 sera rétropropagée dans tous les autres éléments, et avant d'entrer dans chaque élément, sa valeur sera multipliée par un poids. Le poids W_{21} signifie que nous considérons la sortie de l'élément 1 rétropropagée dans l'élément 2.

Nous avons ici une architecture à trois couches, dont la couche entrée et la couche sortie sont de simples registres. La couche centrale, sur laquelle est implantée l'architecture à rétropropagation décrite plus haut, se nomme la couche de Hopfield. Il est à noter que dans les réseaux de Hopfield, chaque couche possède le même nombre d'éléments.

Les réseaux de Hopfield utilisent pour leur apprentissage un algorithme particulier, l'algorithme de Hopfield [26].

Quoiqu'ils constituent une approche intéressante pour la conception de systèmes de reconnaissance de parole, les réseaux de Hopfield semblent plutôt mis à l'écart dans ce domaine. Nous citons cependant un réseau de Hopfield de

120 neurones utilisé pour la reconnaissance de voyelles et de certaines consonnes [13].

5.3.6. Les machines de Boltzmann.

Ces architectures sont semblables aux réseaux de Hopfield, sauf qu'elles combinent une approche déterministique (modèles probabilistiques apparentés aux modèles de Markov) à la technologie des réseaux de neurones.

Les machines de Boltzmann, comme les réseaux de Hopfield, font appel à des concepts d'énergie minimale. Ils possèdent en effet à chaque instant une énergie. Lorsqu'on change une entrée, on donne de l'énergie au système, qui change d'état.

En gros, une machine de Boltzmann essaiera de rétablir son énergie au niveau minimum. Le chercheur intéressé peut consulter la référence [13] pour plus d'information et de bibliographie sur ce type de réseau.

L'approche de Boltzmann est intéressante à considérer pour la reconnaissance de la parole. Citons une machine de Boltzmann reconnaissant 11 sons voisés de la langue anglaise en mode monolocuteur [31].

5.3.7. Les architectures à rétropropagation.

La rétropropagation est en fait une règle d'apprentissage mise au point à partir du concept du perceptron. La figure 5.11 montre un neurone faisant partie d'un réseau à rétropropagation.

Ceci prouve que le concept de perceptron est mal défini et ne comprend pas seulement le modèle proposé par Flank Rosenblatt vers 1950 [44].

Le chercheur rigoureux devra appeler réseaux de perceptrons seulement les réseaux utilisant la règle du perceptron comme algorithme d'apprentissage. Les réseaux ayant une architecture à rétropropagation devraient êtres appelés réseaux à rétropropagation.

Les architectures à rétropropagation sont en tout point semblables aux réseaux de perceptrons. On y trouve une couche d'entrée, une couche de sortie, et au moins une couche cachée.

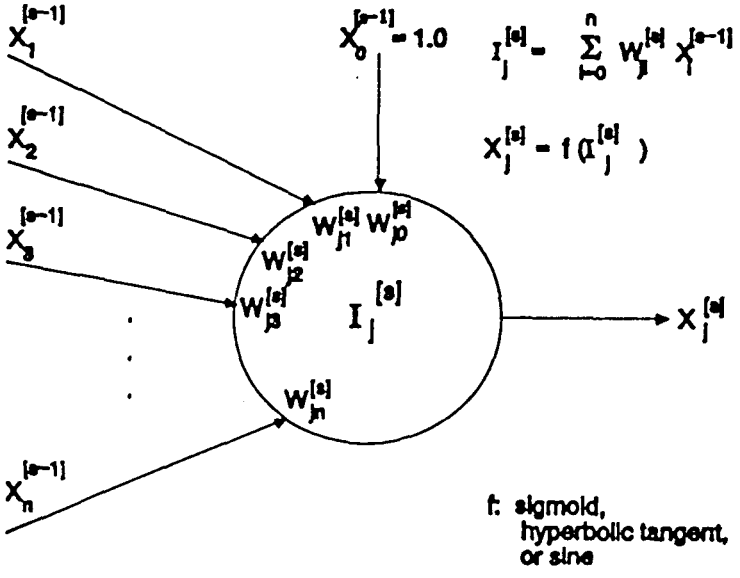


figure 5.11. Une unité d'un réseau à rétropropagation [27].

5.3.8. Autres architectures.

D'autres architectures sont également disponibles au chercheur voulant résoudre un problème en ayant recours aux réseaux de neurones. Ces modèles se différencient surtout par leur règles d'apprentissage.

Les plus notables sont les réseaux à contre-propagation [27], mais nous trouvons aussi plusieurs autres réseaux utilisant d'autres règles d'apprentissage. Ces réseaux n'ont pas, jusqu'à aujourd'hui, fait l'objet de beaucoup d'attention de la part des chercheurs pour le développement d'applications en reconnaissance de la parole. Un de ces réseaux est le réseau de type Dystal, que nous utilisons dans notre système de reconnaissance de parole et que nous verrons plus loin. Dans notre référence [27], on pourra voir une brève description de ces réseaux et avoir accès à une large bibliographie sur chacun d'eux.

Encore là, il peut être utile pour le chercheur expérimenté désireux de vérifier et d'améliorer son travail de considérer ces réseaux. Certains peuvent se révéler à la longue d'une certaine utilité pour la reconnaissance de la parole.

Peu ou pas de systèmes effectuant la reconnaissance de la parole ont été développés en utilisant ces architectures, d'où l'originalité de notre système de reconnaissance.

5.4. Quelques exemples.

Nous décrirons ici quelques architectures fonctionnelles qui effectuent la reconnaissance de la parole. Il est important de préciser que quelque soit l'architecture utilisée pour implanter son système de reconnaissance de parole, le chercheur devra élaborer une méthode pour transformer les signaux de parole avant d'effectuer la reconnaissance. En effet, comme nous l'avons déjà dit, le signal de parole présenté tel quel à un système de reconnaissance contiendra trop d'informations inutiles et ne sera pas reconnu.

Ceci est dû au fait que les signaux de parole sont extrêmement différents suivant les locuteurs et même pour un seul locuteur (voir les chapitres précédents).

Il faut donc accentuer les caractéristiques qui uniformisent les signaux de parole et neutraliser celles particulières à chaque locuteur. Ceci constitue le principal problème auquel est confronté le chercheur en reconnaissance de la parole, mais nous en avons déjà parlé dans les chapitres précédents.

Nous répéterons seulement qu'il est capital de comprendre que l'originalité d'un système et son efficacité reposent aujourd'hui sur les transformations préalables effectuées sur le signal plutôt que sur les architectures à réseaux de neurones utilisées pour la reconnaissance.

Évidemment, ces transformations peuvent être faites par un réseau de neurones, mais en général il est plus efficace d'utiliser d'autres outils.

5.4.1. Réseau utilisant des connexions à délai temporel pour la reconnaissance de parole continue.

Le réseau que nous présentons ici est un réseau de perceptrons, c'est-à-dire que la règle d'apprentissage utilisée est celle du perceptron. Le réseau est décrit en détail dans la référence [42].

Ce système détecte en quelque sorte des séquences qui correspondent à des mots. On peut généraliser ce concept en imaginant que nous avons une série de détecteurs de sons. Un signal de parole correspond à un mélange de sons. Le travail consiste à détecter un agencement de ces sons et à reconnaître de ce fait un mot. La figure 5.12 représente le fonctionnement du système.

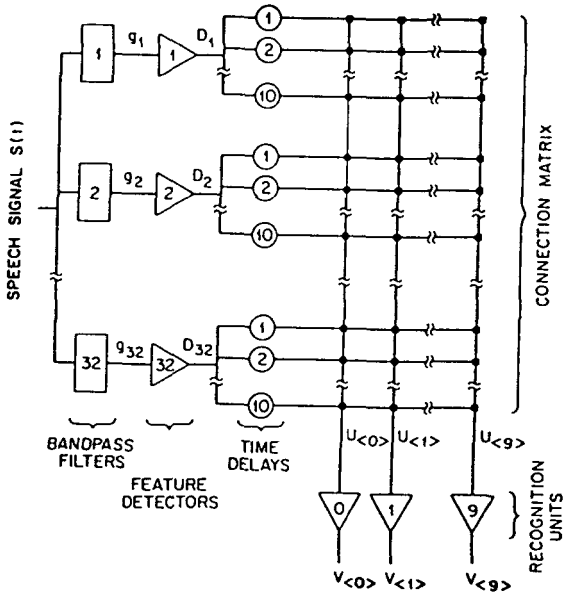


Figure 5.12. Réseau utilisant des connexions à délai temporel [42].

On y voit que le signal de parole est d'abord filtré passe-bande par une série de 32 filtres (bandpass filters). Chacune des 32 sorties passe-bande est ensuite traitée par un détecteur de paramètres (feature detector), de façon à uniformiser les signaux donnés par les filtres passe-bande. Ensuite, pour vérifier l'agencement de ces sons et reconnaître un mot, on envoie tous les signaux dans des détecteurs de délais (time delays). Si les délais concordent, une détection est réussie et un signal est produit à la sortie du système, dans l'une des unités de reconnaissance (recognition units).

Ce système permet la reconnaissance de quelques dizaines de mots prononcés normalement.

5.4.2. Réseau de perceptrons multicouche pour la reconnaissance de voyelles.

Ce travail [10] est encore réalisé dans le but de pouvoir construire un système de parole continue. Un shéma du système est donné à la figure 5.13.

Nous retrouvons d'abord une unité détectrice de sons, capable de détecter 15 sons différents de la langue anglaise, y compris le silence ou l'absence de son. Suit ensuite une unité détectrice de phonèmes, capable d'identifier le phonème correspondant au son. L'unité détectrice de phonèmes

pourra éventuellement être couplée à une unité détectrice de mots, capable de reconnaître un mot à partir de plusieurs phonèmes.

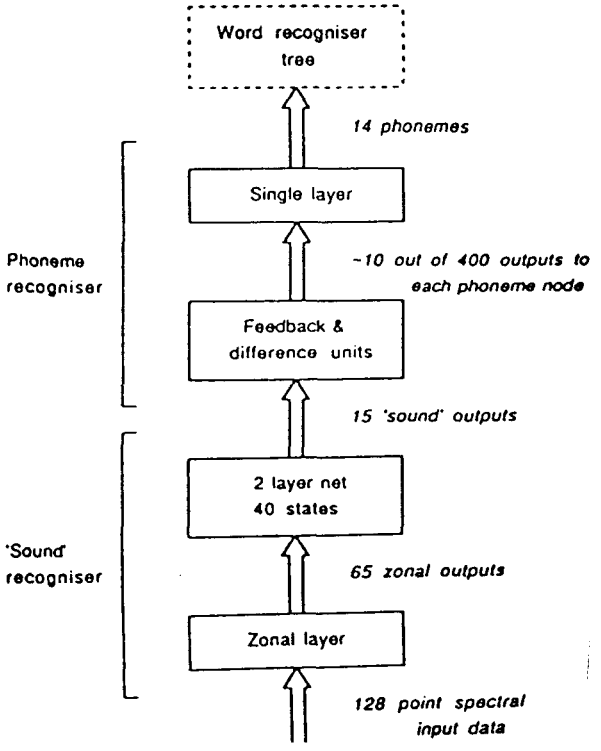


Figure 5.13. Réseau de perceptrons multicouche pour la reconnaissance de voyelles [10].

La figure 5.14 représente le réseau de perceptrons constituant l'unité détectrice de sons. On remarque que 128 éléments spectraux d'un signal de parole échantillonné à 10kHz constituent les entrées de ce réseau. Les 15 sorties correspondent chacune à l'un des 15 sons qu'il est possible de détecter avec ce système et sont les entrées de l'unité de reconnaissance des phonèmes.

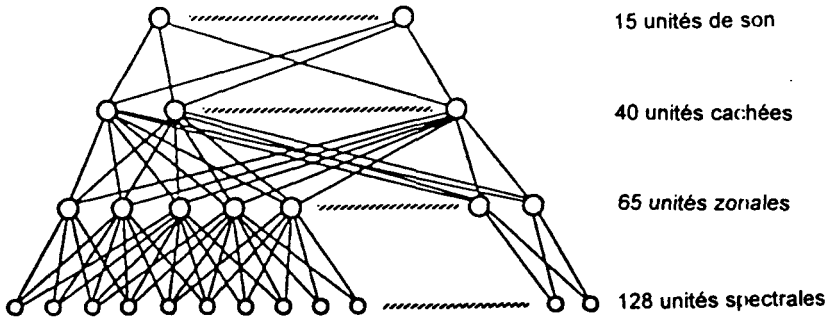


Figure 5.14. Unité détectrice de sons [10].

5.5. Conclusion.

Les réseaux de neurones se révèlent de plus en plus l'outil par excellence pour la réalisation d'un système de reconnaissance de parole qui sera éventuellement capable de comprendre la parole prononcée naturellement et dans un milieu bruité.

La recherche sur la reconnaissance de la parole en utilisant des architectures à réseaux de neurones en est encore aujourd'hui à un stade embrionnaire, aussi la comparaison entre différentes architectures est prématurée. C'est pourquoi nous nous sommes contentés de brosser un tableau général des types de systèmes actuellement utilisés. Les prochaines années verrons sans doute une architecture prédominer sur certaines autres, car on se sera rendu compte de son efficacité par l'expérience. D'ici là, il convient d'explorer les différentes possibilités et de continuer à rechercher l'architecture qui, implantée dans un système plus complexe, donnera les meilleurs résultats.

Nous n'avons pas présenté ici Dystal, le réseau que nous avons choisi pour concevoir notre système de reconnaissance. La raison en est simple: le réseau Dystal fait l'objet de tout le chapitre 6. Comme le lecteur pourra le remarquer, Dystal est plus évolué que ses prédécesseurs que nous venons de présenter.

6

Réseaux de type DYSTAL

6.1. Introduction.

Nous présentons ici les réseaux de neurones de type DYSTAL, pour DYnamically STable Associative Learning. Comme nous l'avons mentionné auparavant, le réseau de neurones utilisé pour notre système de reconnaissance est un réseau de type DYSTAL.

Les réseaux DYSTAL sont différents des réseaux de neurones conventionnels que nous avons présenté dans le chapitre précédent, parce qu'ils sont dérivés directement d'observations biologiques faites en laboratoire. Ils sont par le fait même des modèles plus près de la réalité que les réseaux de neurones conventionnels, dit formels.

Afin de ne pas nous embrouiller dans les lignes qui suivent, définissons dès à présent quelques termes que nous utiliserons dans ce chapitre et dans les autres à venir.

- Réseaux de neurones biologiques (RNB): Biens que nous trouvions ce terme mal choisi, nous l'utiliserons néammoins, pour des raisons pratiques. Le terme correct serait: Réseaux de neurones artificiels dérivés d'observations neurobiologiques. Dans ce texte, RNB est toujours employé pour désigner des réseaux de neurones artificiels. On ne se réfère donc pas aux réseaux de neurones du cerveau humain ou des animaux.

- Systèmes biologiques naturels ou réseaux de neurones biologiques naturels: Ce terme désignera les systèmes neuronaux des organismes vivants.

- Réseaux de neurones conventionnels (RNC): Les réseaux de neurones artificiels tels que nous les connaissons, ou, pour être plus précis, tous les réseaux de neurones qui ne sont pas biologiques.

Les RNB sont différents des RNC du fait qu'ils sont le résultat d'une collaboration étroite entre ingénieurs et informaticiens d'une part et neurobiologistes d'autre part. Les RNB sont donc plus aptes, fondamentalement, à reproduire le comportement de systèmes biologiques naturels. Les RNC, quant à eux, ont été développés par des informaticiens et des mathématiciens avec peu ou pas de connaissances en neurobiologie.

Les RNB sont en quelque sorte des réseaux de neurones conventionnels hybrides. En plus du circuit neuronal (réseau de neurones), ils sont munis d'une série de patrons définis au cours de l'apprentissage. Cette série de patrons peut être comparée, de façon simpliste, à la mémoire des systèmes neuronaux biologiques naturels.

Beaucoup de systèmes intelligents développés jusqu'à ce jour font appel à des techniques de reconnaissance des formes pour la prise de décision. Dans ce contexte, les réseaux de neurones biologiques sont plus complets que leur parents conventionnels, et potentiellement plus adaptés au domaine de l'intelligence artificielle tel que nous le connaissons aujourd'hui.

Il est cependant trop tôt pour faire pencher la balance du côté des RNB ou des RNC. Nous pouvons cependant dire, sous toute réserve, que les RNB

sont un outil très puissant qui pourrait s'avérer capital dans quelques années pour l'avancement de la science dans le domaine de l'intelligence artificielle.

6.2. Généralités.

Le réseau de type DYSTAL est un réseau tiré d'observations effectuées sur une espèce d'escargot marin, *Hermissenda crassicornis*, et sur des lapins. Le principe de base est l'association d'un stimulus conditionné (cs) avec un stimulus non-conditionné (ucs). Cette association n'est possible qu'à la suite d'un apprentissage, au cours duquel le cerveau (ou le réseau DYSTAL) réagit aux stimuli conditionnés en produisant les stimuli non-conditionnés qui ont été appris. On se rappellera la célèbre expérience des chiens de Pavlov, qui salivaient au seul son d'une clochette, qui précédait inévitablement la distribution de nourriture.

6.3. L'apprentissage des réseaux DYSTAL.

Le réseau DYSTAL, comme tout autre réseau de neurones, doit être entraîné préalablement, pour faire son travail efficacement. Nous rappellerons qu'un réseau de neurones artificiel a presque toujours au moins deux modes de

fonctionnement: le mode normal, dans lequel on peut considérer que les propriétés du réseau sont statiques dans le temps (système stationnaire), et le mode d'apprentissage, dans lequel le réseau modifie constamment ses propriétés internes, en fonction des données qu'on lui présente en entrée. Ce dernier mode est le mode d'apprentissage. Dans les lignes qui suivent, nous verrons comment DYSTAL se comporte dans ce mode d'apprentissage.

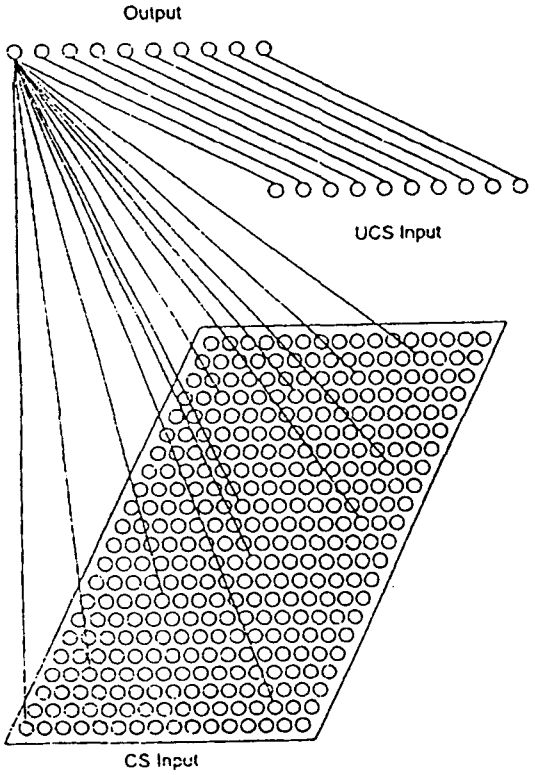


Figure 6.1. Grille d'entrée cs [5].

Pour entraîner un réseau DYSTAL, on lui présente une série de stimuli conditionnés (cs), associés chacun avec son stimulus non-conditionné (ucs). Si le réseau n'a jamais vu le stimulus conditionné qui lui est présenté, il créera un patron (patch) qui sera stocké en mémoire. Le patron créé représentera le stimulus cs et le stimulus ucs qui lui est associé. Il n'existe pas de limite au

nombre de patrons qu'un réseau DYSTAL peut garder en mémoire, à part la limite physique de la machine qui supporte son implantation. Une routine d'élimination des patrons désuets doit être créée, afin de ne pas surcharger la mémoire.

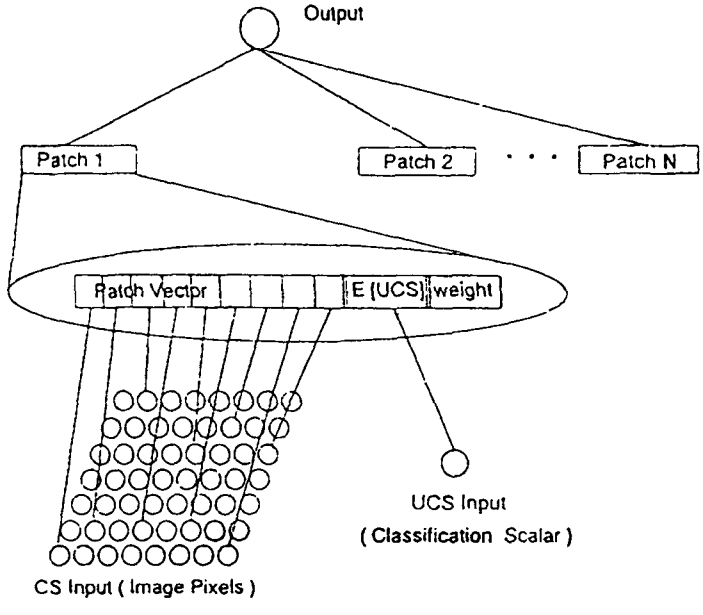


Figure 6.2. Détail d'un neurone DYSTAL [5].

Dans un réseau DYSTAL, un stimulus peut être une grille d'entrée contenant des valeurs binaires. La figure 6.1 montre une telle grille. Une grille similaire à celle de la figure 6.1 constitue l'entrée cs du réseau DYSTAL agissant au sein de notre système de reconnaissance de parole.

Chaque patron d'un réseau DYSTAL contient une entrée complète cs, l'entrée ucs qui lui a été associée ainsi qu'un poids qui mesure la fréquence d'utilisation (l'importance relative du patron dans le réseau). La figure 6.2 représente un neurone d'un réseau DYSTAL avec les patrons qui lui sont associés, et montre en détail le contenu d'un de ces patrons.

Comme on le voit sur la figure 6.2, un patron DYSTAL est en fait constitué d'une série de valeurs numériques représentatives de l'entrée cs. Nous appellerons donc l'entrée cs un vecteur. Ce vecteur n'est pas nécessairement un vecteur à 2 dimensions. Comme nous le verrons dans les chapitres suivants, les vecteurs cs de notre système caractérisent des images en 3 dimensions. Une implémentation plus complexe de DYSTAL peut contenir des vecteurs à n dimensions. On voit aussi sur la figure 6.2 qu'à chaque vecteur est associée une entrée ucs. Cette entrée n'est pas présentée lors du fonctionnement normal de DYSTAL. Le travail du réseau consiste donc, lorsqu'il n'est pas en apprentissage, à reproduire un stimulus non conditionné (ucs) lorsqu'on lui présente un stimulus contidionné (cs). En plus d'une représentation numérique des entrées cs et ucs, on retrouve aussi dans chaque patron un poids qui lui est associé. Le poids est une mesure de l'importance qui est attachée à un patron donné lors de la reconnaissance et est lié à la fréquence d'apparition de ce patron lors de l'apprentissage et de la reconnaissance.

Comme nous l'avons déjà mentionné, une des différences majeures de DYSTAL avec les RNC est que les poids ne sont pas associés aux connexions entre les neurones (ou synapses) mais bel et bien aux patrons qui sont stockés en mémoire.

Au début de l'apprentissage, aucun patron n'existe. Les patrons seront créés un par un au cours de l'apprentissage. Le processus de création des patrons peut être facilement compris si nous examinons ce qui ce passe lors d'un cycle d'apprentissage.

La première étape est la présentation d'une entrée cs associée à son entrée ucs. DYSTAL, dans un premier temps, compare l'entrée cs avec toutes celles qu'il a déjà emmagasinées dans ses patrons. La comparaison se fait à l'aide d'une formule mesurant la similarité entre l'entrée et un patron. La formule est la suivante:

$$S^j = \mathrm{corr}(P^j, I) = \frac{\sum_i (P_i^j - \bar{P}^j)(I_i - \bar{I})}{\sqrt{\left(\sum_i (P_i^j - \bar{P}^j)^2 \sum_i (I_i - \bar{I})^2\right)}}$$

Formule 6.1.

Il est important de bien comprendre le sens de cette formule, car elle est capitale pour l'apprentissage des réseaux DYSTAL.

S^j: Valeur de la similarité entre une entrée cs (I) et le patron j (contenu dans la mémoire de DYSTAL).

corr (P^j, I): Valeur de la corrélation entre le patron j et une entrée cs (I).

P^j_i: Valeur numérique du $i^{ième}$ élément du vecteur cs du $j^{ième}$ patron de la mémoire de DYSTAL.

I_i: Valeur numérique du $i^{ième}$ élément du vecteur cs de l'entrée qui est présentée à DYSTAL.

$\bar{P}^j$ et $\bar{I}$: Valeur moyenne de toutes les valeurs numériques comprises dans les vecteurs cs du $j^{ième}$ patron et de l'entrée I qui est présentée à DYSTAL.

L'application de cette formule donne une valeur S^j comprise entre -1 et 1. Cette valeur est une mesure directe de la similarité entre un patron et une entrée présentée au réseau, 1 étant une similarité parfaite (identité) et -1 étant une dissimilarité parfaite. La valeur 0, quant à elle, représente une orthogonalité entre les vecteurs, soit une absence totale de similarité ou de dissimilarité.

Nous disions donc que la première étape de l'apprentissage de DYSTAL est la présentation d'une entrée cs associée à son entrée ucs. Cette entrée est comparée avec tous les patrons que DYSTAL a en mémoire, et une valeur de similarité est calculée pour chaque patron. Si la similarité est en dessous d'une valeur prédéfinie, un nouveau patron est crée. Sinon, le patron le plus semblable à l'entrée cs est mis à jour. Un patron très similaire à une entrée cs est laissé tel quel, n'ayant pas besoin de mise à jour. Le patron cs est mis à jour de façon graduelle, en appliquant la formule suivante:

$$P_i^m(t) = \frac{(t-1)\left[P_i^m(t-1)\right] + I_i}{t}$$

Formule 6.2.

$P^m{}_i$ est la valeur du $i^{ième}$ élément du patron P. t est le temps, ou, pour être plus précis, le nombre de fois que le patron a été modifié. $P^m{}_i(0) = 0$. I_i est l'élément i de l'entrée I. Nous voyons donc qu'à t = 1, i.e. lors de la création du patron, la valeur de ce dernier est égale à la valeur de l'entrée présentée au réseau de neurone, puisqu'il s'agit d'un nouveau patron qui n'a jamais été modifié. Nous voyons aussi que l'importance du changement effectué par une entrée sur un patron diminue avec le temps, et tends vers 0 lorsque t devient grand. La valeur finale est la moyenne de toutes les entrées correspondant au même patron. Il en est de même pour l'entrée ucs qui est présentée lors de l'apprentissage. La formule de la variation de l'entrée ucs pour un patron donné sera donc:

$$\cup^{m}(t) = \frac{(t-1)\left[\cup^{m}(t-1)\right] + UCS(t)}{t}$$

Formule 6.3.

U^m est le stimulus ucs moyen, t est le nombre de fois que le patron est mis à jour et UCS(t) est le stimulus ucs qui est présenté à l'instant t lors de l'apprentissage.

Bien entendu, aucune évidence biologique ne prouve que ces équations sont utilisées lors de l'apprentissage d'un animal ou d'un être humain. Cependant, le cerveau d'un organisme vivant apprend à ajuster sa mémoire en fonction du temps pour réfléter les valeurs les plus plausibles d'un stimulus cs et d'un stimulus ucs [5].

En plus d'ajuster les stimuli, lors d'un cycle d'apprentissage, les poids des patrons sont mis à jour, eux aussi. La mise à jour se fait de la façon suivante:

-Un sous-ensemble de tous les patrons répondant à un signal ucs donné est constitué.

-Le poids du patron le plus semblable à l'entrée cs est augmenté, tous les autres sont décrémentés. Le poids réflète donc l'utilisation d'un patron donné. La formule déterminant l'incrémentation d'un poids est la suivante:

$$W^m(t) = \left| 1 - W^m(t-1) \right| * a + W^m(t-1)$$

Formule 6.4.

Où a est l'incrément d'apprentissage et $W^m(0) = 0$. En plus d'incrémenter le poids du patron trouvé, le poids de tous les autres patrons est décrémenté de la façon suivante:

$$W^k(t) = W^k(t-1) * (1-b)$$

Formule 6.5.

où b est le décrément d'apprentissage. Les poids convergent ici de la même façon que lorsqu'on effectue l'apprentissage avec un RNC.

La sortie donnée par un neurone DYSTAL est égale au produit du calcul de la similarité et du stimulus ucs trouvé avec le patron le plus semblable. Donc:

$$N = S^m * U^m$$

Formule 6.6.

6.4. Fonctionnement normal.

En mode de fonctionnement normal, seulement l'entrée cs est présentée au réseau. Le réseau choisit le patron le plus semblable à l'entrée qui lui est présentée, et donne la sortie ucs correspondante. Il est important ici de souligner l'approche neuronale de DYSTAL. Il ne faudrait pas voir un réseau DYSTAL comme un simple "pattern matching system", pardonnez-moi l'anglicisme. Dystal, en plus de mémoriser des patrons, peut aussi les ajuster dynamiquement pour mieux s'adapter aux conditions ambiantes. Dystal peut réagir à un tout nouveau stimulus en créant un patron adapté à ce stimulus. En fait, Dystal crée lui-même les patrons qu'il reconnaîtra plus tard. L'architecture des réseaux DYSTAL, bien décrite dans [1] et dans la section suivante, se prête en tout point au parallélisme, car nous pouvons en effet avoir plusieurs neurones de sortie ayant chacun leur patrons et travaillant de façon indépendante tout en ayant accès aux mêmes informations d'entrée.

6.5. DYSTAL et le problème du XOR, une implémentation "hardware".

Afin de bien comprendre le fonctionnement d'un réseau de type DYSTAL, nous allons expliquer en détail comment il peut résoudre avec succès le problème du XOR.

Nous tenons à préciser que les explications qui suivent montrent une implémentation "hardware" de Dystal, c'est à dire ce qui se passerait à l'intérieur

d'un neuroprocesseur Dystal, à la différence de ce que nous avons expliqué plus tôt (figures 6.1 et 6.2), qui est une implémentation "software". Il va sans dire que le système que nous avons développé et dont nous parlerons plus loin est une implémentation "software" de Dystal.

La fonction XOR, très bien connue des informaticiens, a la table de vérité présentée à la figure 6.3.

Entrée 1	Entrée 2	Sortie
0	0	0
0	1	1
1	0	1
1	1	0

Figure 6.3. Table de vérité du XOR.

Il s'agit ici de conditionner le réseau à produire les sorties en fonction des entrées.

Nous avons donc un réseau DYSTAL de 2 entrées et de une sortie. Les deux entrées, comme nous l'avons vu précédemment, sont les stimuli conditionnés cs. En plus des entrées cs, nous aurons évidemment une entrée ucs correspondant à la sortie désirée (pour l'apprentissage seulement).

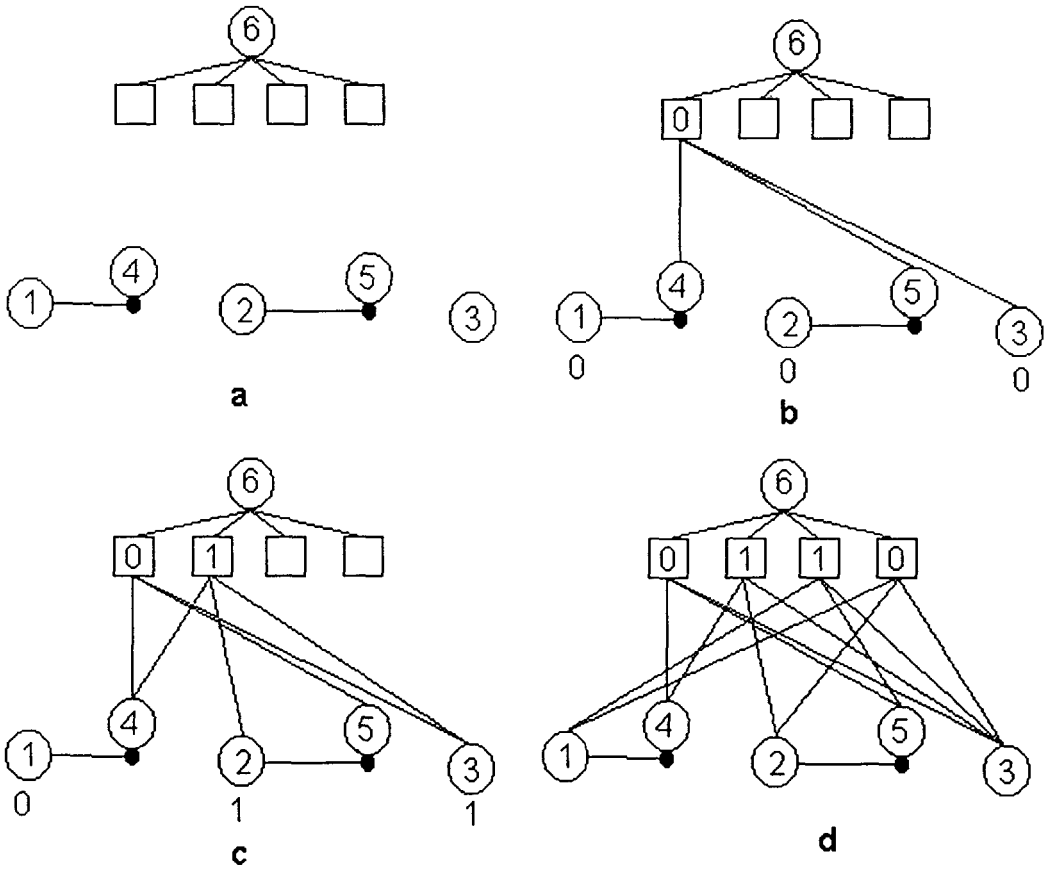


Figure 6.4. Apprentissage de DYSTAL pour la fonction XOR.

Au départ, le réseau présente l'aspect de la figure 6.4.a. Les neurones 1 et 2 sont les neurones d'entrée recevant les entrées cs. Le neurone 3 sert pour l'apprentissage, c'est lui qui reçoit l'entrée ucs. Les neurones 4 et 5 sont ce qu'on appelle des interneurones, permettant des connexions indirectes au neurone de sortie. Les interneurones sont en état constant d'excitation, à moins qu'une connexion inhibitrice ne vienne les arrêter. Les connexions inhibitrices provenant des neurones d'entrée et se rendant aux interneurones sont montrées sur la figure 6.4, elles sont terminées par un cercle noir. Dans tous les réseaux DYSTAL, chaque neurone d'entrée doit avoir son interneurone, pour lui permettre d'inverser son signal à la sortie. Les grilles représentent les patrons qui seront définis au cours de l'apprentissage. Le nombre de patrons maximum que peut contenir un réseau DYSTAL est 2^n, n étant le nombre de neurones recevant l'entrée cs. Comme, dans notre cas, nous n'avons que 2 entrées cs, il n'y a possibilité que de 4 patrons.

Pour les réseaux DYSTAL ayant beaucoup d'entrées cs, le nombre maximum de patrons n'est jamais atteint, d'où l'importance d'utiliser l'allocation dynamique de mémoire pour la création de patrons.

Présentons maintenant notre table de vérité au réseau DYSTAL. La première entrée est 0 0 avec 0 comme ucs. Comme le réseau n'a aucun patron de créé, il n'a pas besoin de faire de calculs de similarité. Un nouveau patron est créé. Comme les signaux d'entrée sont à zéro, les connexions serons établies entre les interneurones, qui sont toujours actifs à moins d'être inhibés par un neurone cs envoyant un signal. La figure 6.4.b montre les connexions qui sont établies.

La seconde entrée est 0 1 avec 1 comme sortie désirée (ucs). Comme le réseau possède déjà un patron, la similarité (corrélation) est calculée avec ce dernier. La similarité entre 0 0 et 0 1 n'étant pas très élevée, un nouveau patron est créé. Nous obtenons la figure 6.4.c. La similarité est calculée selon la formule 6.1.

Il est important de noter que, dans notre exemple, cette formule ne fonctionne pas dans tous les cas. Nous laissons le soin au lecteur de vérifier notre affirmation. La raison est mathématique, le calcul de la similarité avec cette formule est impossible si toutes les entrées ont la même valeur. Cependant, dans les systèmes complexes qui ont plusieurs dizaines d'entrées, avoir toutes les entrées à la même valeur est pratiquement impossible. Une façon de contourner le problème, dans notre cas, serait d'insérer une valeur "bidon" dans nos patrons, mais, comme le lecteur est averti, nous lui épargnerons cette complication.

L'apprentissage terminé, nous obtenons un réseau qui présente l'architecture de la figure 6.4.d. Ce réseau est prêt pour le mode de fonctionnement normal. Supposons que nous lui présentons l'entrée 1 - 0. Nous n'avons maintenant plus d'entrée ucs, DYSTAL doit la trouver lui-même. Le neurone 3 ne reçoit donc ici aucune excitation. Le neurone 1, étant excité, enverra des impulsions aux patrons 3 et 4, en même temps qu'il inhibera l'interneurone 4. Le neurone 2, quand à lui, ne sera pas excité, et l'interneurone 5 sera donc en mesure d'envoyer ses impulsions aux patrons 2 et 4. Le patron 4 recevant 2 impulsions, sera activé et choisi comme le bon patron. La sortie que donnera le neurone de sortie sera donc 0, soit la valeur stockée dans le patron.

6.6. Conclusion.

Nous avons présenté dans ce chapitre le réseau DYSTAL ainsi que son fonctionnement. Nous avons donc tous les outils en main pour comprendre le système de reconnaissance dans son ensemble.

Il serait intéressant de comparer les performances d'un réseau DYSTAL, que nous pouvons qualifier de système à réseaux de neurones hybride, avec celles d'un RNC. Les performances d'apprentissage de DYSTAL sont en effet très intéressantes. À titre d'exemple, nous avons développé un système de reconnaissance de caractères manuscrits (les chiffres de 0 à 9). Avec un jeu de 11 patrons, DYSTAL s'est avéré d'une efficacité surprenante. Il a réussi à reconnaître une vingtaine d'entrées qu'il n'avait jamais "vues". Il est intéressant de noter que l'apprentissage de ces 30 patrons demande extrêmement peu de temps (quelques secondes).

Dans le prochain chapitre, nous présenterons le système de reconnaissance dans son entier, de la partie analyse par démodulation à la partie DYSTAL.

7

Le système de reconnaissance

Nous décrirons ici le système de reconnaissance en son entier, à partir des opérations de pré-traitement par analyse par démodulation jusqu'à la partie reconnaissance par réseau Dystal.

Une attention toute particulière sera apportée à l'interaction entre les différentes composantes de notre système et à la façon dont nous les avons implantées.

7.1. Introduction.

Le système présenté ici est en quelque sorte un système à réseaux de neurones hybride. La partie reconnaissance est faite par un réseau de neurones de type DYSTAL, tandis que la partie pré-traitement est en quelque sorte une implantation de l'analyse par démodulation, exposée dans le chapitre 4. Dans un premier temps, nous parlerons du pré-traitement fait sur les signaux présentés à DYSTAL, pour ensuite expliquer comment fonctionne la partie reconnaissance en tant que tel.

7.2. L'analyse.

Comme nous l'avons expliqué au chapitre 4, l'analyse par démodulation produit une sortie filtrée en fréquence sur 24 canaux. Nous appelons ceci un banc de filtres de 24 canaux.

Les filtres constituant le banc de filtres sont centrés sur des fréquences variant de 330Hz (en conformité avec le téléphone) à 4700Hz. Le filtre centré à 330Hz est quand même assez large pour aller chercher des formants dont la fréquence est plus basse, comme ceux du /i/, du /y/ et du /u/. Cette gamme de fréquences peut contenir la partie la plus importante de l'information fréquencielle de la parole. On peut voir la réponse en fréquence du banc de filtres au complet sur la figure 4.1 (voir chapitre 4).

Comme ce qui nous intéresse ici est la différence entre les formants et non la valeur des formants en tant que telle, une différence F2 - F1 (plus de 1000Hz) aussi grande que dans les sons /i/, /y/ et /u/ ne pourra être "vue" par aucun filtre, même si sa fréquence centrale est très basse.

La sortie de chaque filtre est donc un signal filtré passe-bande centré autour d'une fréquence centrale f_i où f_i est la fréquence centrale du canal i. Le signal de sortie $s_i(t)$ du canal i peut être vu comme un signal modulé en amplitude et en phase dont la fréquence porteuse est f_i. Le signal aura donc la forme suivante.

$$s_i(t) = A_i(t)\cos[\omega_i(t) + \phi_i(t)]$$

Formule 7.1.

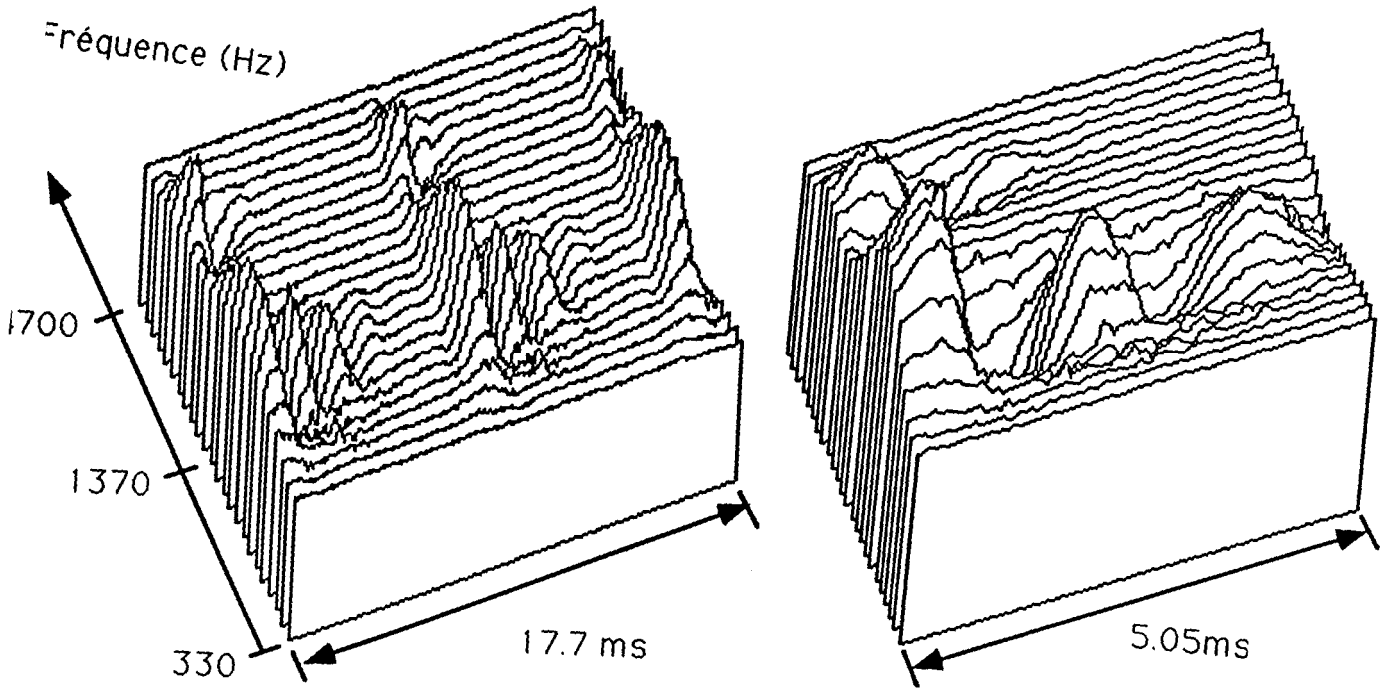
$A_i(t)$ est l'amplitude de $s_i(t)$ et $\phi_i(t)$ est la phase. Nous extrayons ensuite l'enveloppe des signaux $s_i(t)$ par la formule suivante:

$$A_i(t) = \sqrt{s_i(t)^2 + s_i(t)_q^2}$$

Formule 7.2.

où $s_i(t)_q$ est la transformée de Hilbert de $s_i(t)$. Ensuite, nous obtenons des images 3D en calculant $A_i(t) \cdot A_i(t)'$. $A_i(t)'$ est évidemment la dérivée de $A_i(t)$ par rapport au temps. Des exemples d'images 3D sont donnés au chapitre 4 (figures 4.4, 4.5 et 4.6). Ces images 3D représentent donc les signaux $A_i(t) \cdot A_i(t)'$ en fonction du temps en x et de la fréquence en y. Nous invitons le lecteur à se référer au chapitre 4 pour plus de précisions sur la technique utilisée dans l'analyse de nos signaux de parole.

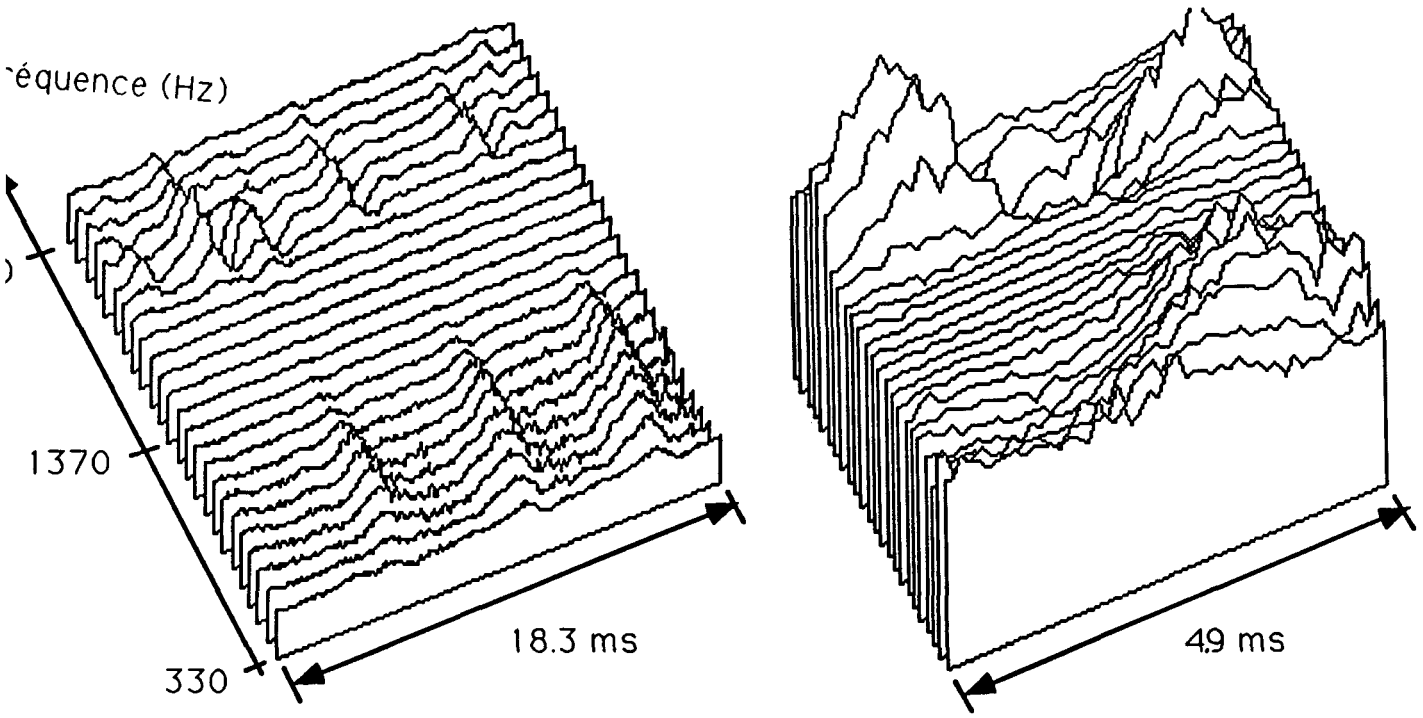
Les signaux de parole que nous analysons sont échantillonnés à 32 kHz. Les images 3D que nous obtenons à la sortie de l'analyse par démodulation consistent en 24 canaux échantillonnés à 16 kHz.



a. Image 3D.

b. Patron correspondant.

Figure 7.1. Voyelle /a/, homme.



a. Image 3D.

b. Patron correspondant.

Figure 7.2. Voyelle /ε/, femme.

7.3. L'apprentissage avec DYSTAL.

Précisons ici que l'apprentissage dont nous parlons se situe dans un contexte de simulation, donc nous n'utilisons pas exactement la méthode "hardware" décrite au chapitre 6 (XOR).

La prochaine étape consiste à définir des patrons qui seront appris par DYSTAL. Les patrons de notre système consistent en une portion d'image 3D, que nous essayons de rendre la plus indépendante du locuteur possible.

Nous voyons à la figure 7.1.a une image 3D pour le son /a/ prononcé par un homme. La figure 7.1.b montre le patron tiré de cette image 3D. Un patron débutera inévitablement au début d'une impulsion glottale (le plus haut et le premier pic d'une série). Par exemple, sur la figure 7.1.a, nous voyons deux séries de pics consécutifs. Le patron isole en fait une de ces séries. Nous conservons seulement les deux ou trois premiers pics car nous pouvons généraliser en disant que tous les pics suivant les deux ou trois premiers sont plus dépendants du locuteur.

Similairement, on peut observer sur les figures 7.2.a et 7.2.b la voyelle /ε/ prononcé par une femme. La figure 7.2.a représente la figure 3D et la figure 7.2.b le patron correspondant.

Dans le mode d'apprentissage, DYSTAL considère les patrons qui correspondent aux images 3D sélectionnées et les stocke en mémoire, exactement comme expliqué dans le chapitre 6. Pour chaque patron qui est appris, un stimulus non-conditionné (ucs) est bien entendu présenté à DYSTAL. Le stimulus non-conditionné est en fait un caractère ASCII caractérisant le son correspondant au patron. Par exemple, le patron de la figure 7.1.b. est présenté avec la lettre a comme stimulus non-conditionné. Nous rappelons que le patron représente le stimulus conditionné (cs).

Une association est donc faite par DYSTAL entre le patron de la figure 7.1.b et la lettre a, représentant le son /a/.

À chaque patron qui est présenté à DYSTAL, une similarité est calculée entre le nouveau patron et chaque patron ayant le même stimulus ucs. Si la similarité entre le nouveau patron et un des patrons déjà mémorisés est au-delà d'une valeur prédéterminée, 0.9 dans notre cas, le patron n'est pas mémorisé. Autrement, le patron est mémorisé et stocké avec les autres.

Comme nous l'avons déjà mentionné dans le chapitre précédent, un patron DYSTAL est constitué, en plus d'une image 3D et d'un caractère associé au son représenté par l'image, d'un poids représentant la fréquence d'utilisation du patron en question. Nous référons le lecteur au chapitre précédent pour les règles de gestion des poids associés aux patrons.

7.4. La phase de reconnaissance.

Durant la phase de reconnaissance, nous présentons à DYSTAL une image 3D représentant un segment de parole continue. Ce segment pourrait être calculé en temps réel de façon continue, en ayant préalablement optimisé la phase de prétraitement, bien entendu. Il n'y a pas de contraintes au niveau de la longueur du segment. Le système génère aussi facilement une image 3D pour un court mot que pour une longue phrase.

Lors de la phase de reconnaissance, une fenêtre est déplacée le long du signal prétraité (sortie de l'analyse par démodulation). La similarité est calculée entre cette fenêtre et les patrons stockés dans la mémoire de DYSTAL. Nous créons ensuite des images statiques en utilisant l'algorithme qui est donné ci-dessous. Se référer aux figures 7.1 et 7.2 pour une meilleure compréhension.

Les figures 7.1.a et 7.2.a représentent des exemples de sortie de l'analyse par démodulation, on peut y observer 24 canaux, montrant chacun un signal en fonction du temps. Dans un premier temps, une fenêtre est déplacée le long de l'échelle de temps de l'image 3D. Les similarités maximum sont calculées entre cette fenêtre et chacun des sons stockés en mémoire. Par exemple, la figure 7.3 représente un réseau DYSTAL ayant été entraîné avec 3 sons, /a/, /i/ et /ou/. On y voit 3 neurones, chacun spécialisé dans la détection d'un seul son. À chaque neurone est associé une série de patrons.

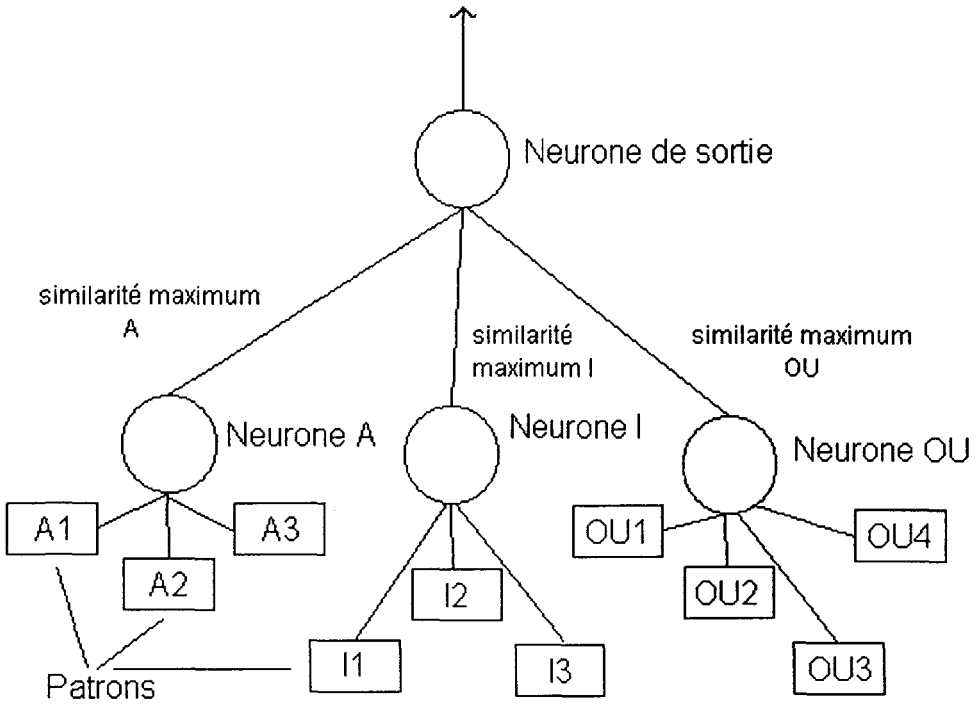


Figure 7.3. Réseau Dystal reconnaissant 3 sons.

1. Les similarités maximum sont calculées. Dans notre exemple, trois valeurs seront donc calculées à cette étape. Une similarité maximum avec le "a", une similarité maximum avec le "i", et une similarité maximum avec le "ou". Il est intéressant de noter que la fenêtre qui est déplacée le long de l'image 3D est de longueur variable. En effet, comme chaque patron mémorisé par DYSTAL est de longueur variable, la fenêtre s'ajuste automatiquement à la longueur de chaque patron, de façon à effectuer le calcul de la similarité.

2. La fenêtre est décalée d'une unité temporelle, et l'étape 1 est réalisée de nouveau. On stocke, à chaque unité temporelle, les valeurs des similarités maximum, de façon à avoir par la suite une image 3D simplifiée. Nous aurions dans notre exemple une image 3D à trois canaux, après le calcul des similarités maximum. La figure 7.4 montre le décalage de la fenêtre de longueur variable. On y voit, à chaque instant t, que les similarités maximum avec tous les patrons sont calculées.

3. Prendre une fenêtre de 10ms de signal provenant de la sortie de l'étape 2. Localiser le maximum de cette fenêtre. La localisation du maximum est indépendante des canaux.

4. Extraire de la fenêtre de 10 ms un segment de 2 ms. Ce segment débute par le maximum trouvé en (2). Le segment considéré ne comprend donc en principe que l'information provenant d'une impulsion glottale puisque sa durée ne dépasse pas 2 ms. Comme nous l'avons déjà mentionné dans les chapitres précédents, il est essentiel dans la reconnaissance de la parole d'éliminer les données dépendantes du locuteur, pour accentuer celles uniformes à tous les locuteurs. La fréquence de vibration des cordes vocales est déterminée par les impusions glottales, facteur hautement dépendant du locuteur. Si l'on ne considère qu'une impulsion glottale, on élimine ainsi ce facteur indésirable.

5. On place le segment de 2 ms dans un tableau temporaire.

6. On décale ensuite la fenêtre de 10 ms de 5 ms sur le signal des similarités donné par l'étape 2. On a donc des fenêtres qui se chevauchent. On

recommence le processus à partir de (3), jusqu'à ce que tout notre signal soit traité.

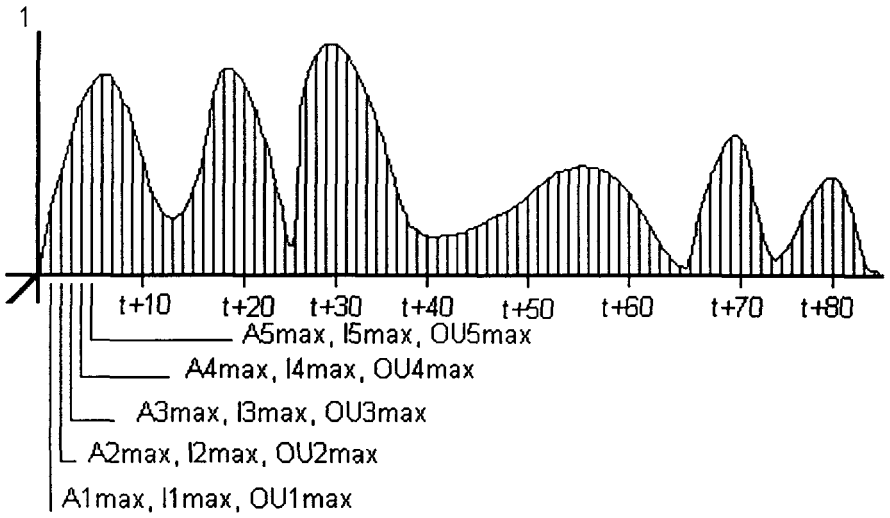


Figure 7.4. Décalage de la fenêtre de calcul des similarités.

Nous avons donc vu que la sortie de la première couche de notre réseau de neurones DYSTAL donne un tableau "statique" des similarités maximums pour chaque sons qu'il a stocké en mémoire. Nous avons maintenant deux solutions pour calculer le son correspondant. On peut soit calculer la moyenne de tous les canaux sur un intervalle déterminé et choisir la plus grande comme étant le son correspondant, soit encore élaborer un étage supérieur capable de traiter l'information provenant de la première couche de DYSTAL. La figure 7.5 donne un exemple de sortie que nous obtenons avec Dystal.

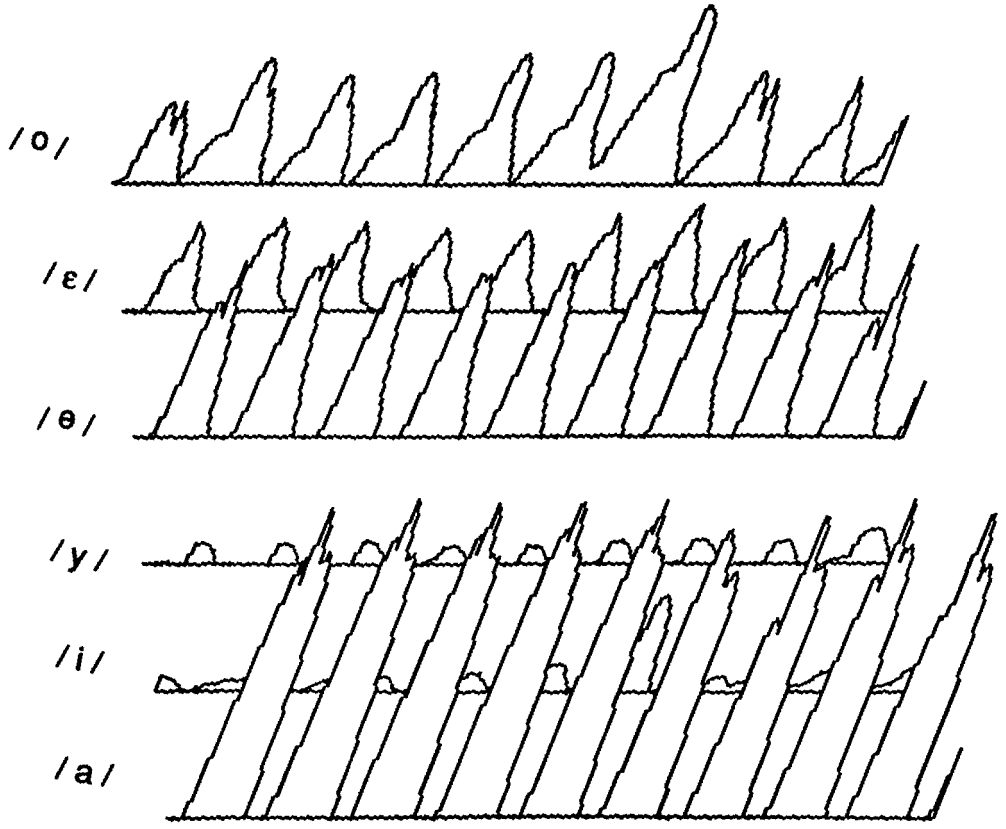


figure 7.5. Sortie de la première couche, voyelle /a/ prononcée par un homme. Les similarités sont calculées pour 6 voyelles, / a /, / i /, / y /, / ə /, / ɛ /, / o /.

7.5. L'étage supérieur du réseau DYSTAL.

Nous décrirons ici sommairement la couche supérieure de notre système de reconnaissance de parole. Il est important de noter que la pertinence d'utiliser ou non cette couche supérieure reste à déterminer.

La couche supérieure n'est rien d'autre qu'un réseau DYSTAL semblable en tout point à celui décrit précédemment, sauf que les entrées d'apprentissage qui lui sont présentées sont différentes de celles que nous avons décrites.

En effet, le stimulus conditionné de la deuxième couche consiste en une série de trois ou quatre segments de 2ms des images 3D données par la premième couche. Ces images "statiques" sont appariées à un stimulus ucs semblable à celui déjà décrit (une simple représentation ASCII du son considéré), par exemple le caractère ASCII a pour le son /a/.

En mode de fonctionnement normal, la sortie de la première couche est présentée à l'entrée de la deuxième couche, et une similarité maximum est donnée à la sortie de la deuxième couche. Cette fois, la similarité maximum est calculée indépendamment des sons contenus en mémoire, c'est-à-dire que la deuxième couche ne produira qu'une sortie ucs, soit la sortie ucs du patron ayant la similarité la plus grande avec l'entrée qui lui a été présentée. On a donc en sortie de la deuxième couche le stimulus ucs correspondant au patron ayant obtenu la similarité maximum avec le signal d'entrée. Ici, les patrons ne sont pas classés par sons, comme dans la première couche.

On décale ensuite le signal d'entrée d'une unité et on recommence le processus, jusqu'à ce que tout le signal soit traité.

Nous avons dû modifier légèrement la deuxième couche pour des raisons pratiques que nous verrons dans le chapitre 8. Ce qui doit être retenu est que la deuxième couche est un réseau Dystal traitant les images statiques obtenues par la première couche.

7.6. Conclusion

Nous avons décrit ici notre système de reconnaissance en son entier. Nous verrons au prochain chapitre les résultats obtenus suite aux expériences que nous avons faites avec notre système.

Nous désirons attirer l'attention du lecteur sur la deuxième couche, qui demandera certainement des expériences supplémentaires de validation. En effet, comme nous l'avons mentionné plus haut, nous ne sommes pas convaincu de la pertinence de cette deuxième couche, pour la reconnaissance des quelques sons avec lesquels nous avons effectué nos expériences.

La deuxième couche sera possiblement beaucoup plus utile dans le cas on l'on augmentera le nombre de sons contenus dans la mémoire de DYSTAL.

On aura ici sûrement besoin d'un système plus complexe, capable de faire face à une plus grande variété de combinaisons.

Nous reparlerons de la deuxième couche de Dystal dans le prochain chapitre.

8

Expériences et résultats

Nous décrirons ici les expériences que nous avons effectuées ainsi que les résultats obtenus. Nous conclurons ce chapitre par des recommandations qui devraient être ultérieurement appliquées au système que nous avons développé. Les expériences sont décrites ici dans l'ordre où elles ont été effectuées.

8.1. Première expérience, premier locuteur homme, première couche.

Le but de cette expérience est de voir les sons qui sont bien reconnus et ceux qui sont problématiques.

Dans un premier temps, nous avons fait apprendre à Dystal plusieurs patrons pour les sons de la langue française suivants: /a/, /i/, /y/, /ø/, /ɛ/ et /o/. Lors de l'apprentissage, nous avons présenté à Dystal trois patrons de / a /, deux hommes et une femme, cinq patrons de /i/, trois hommes et deux femmes, trois patrons de /y/, un homme et deux femmes, trois patrons de / ø/, un homme et deux femmes, trois patrons de /ɛ/, un homme et deux femmes et 3 patrons de /o/, un homme et deux femmes. Nous avons ensuite testé avec des signaux de parole provenant d'un locuteur homme qui n'avaient jamais été présentés à DYSTAL. En fait, DYSTAL, n'avait jamais appris aucun patron provenant de ce locuteur auparavant. Voyons un peu ce que nous avons obtenu.

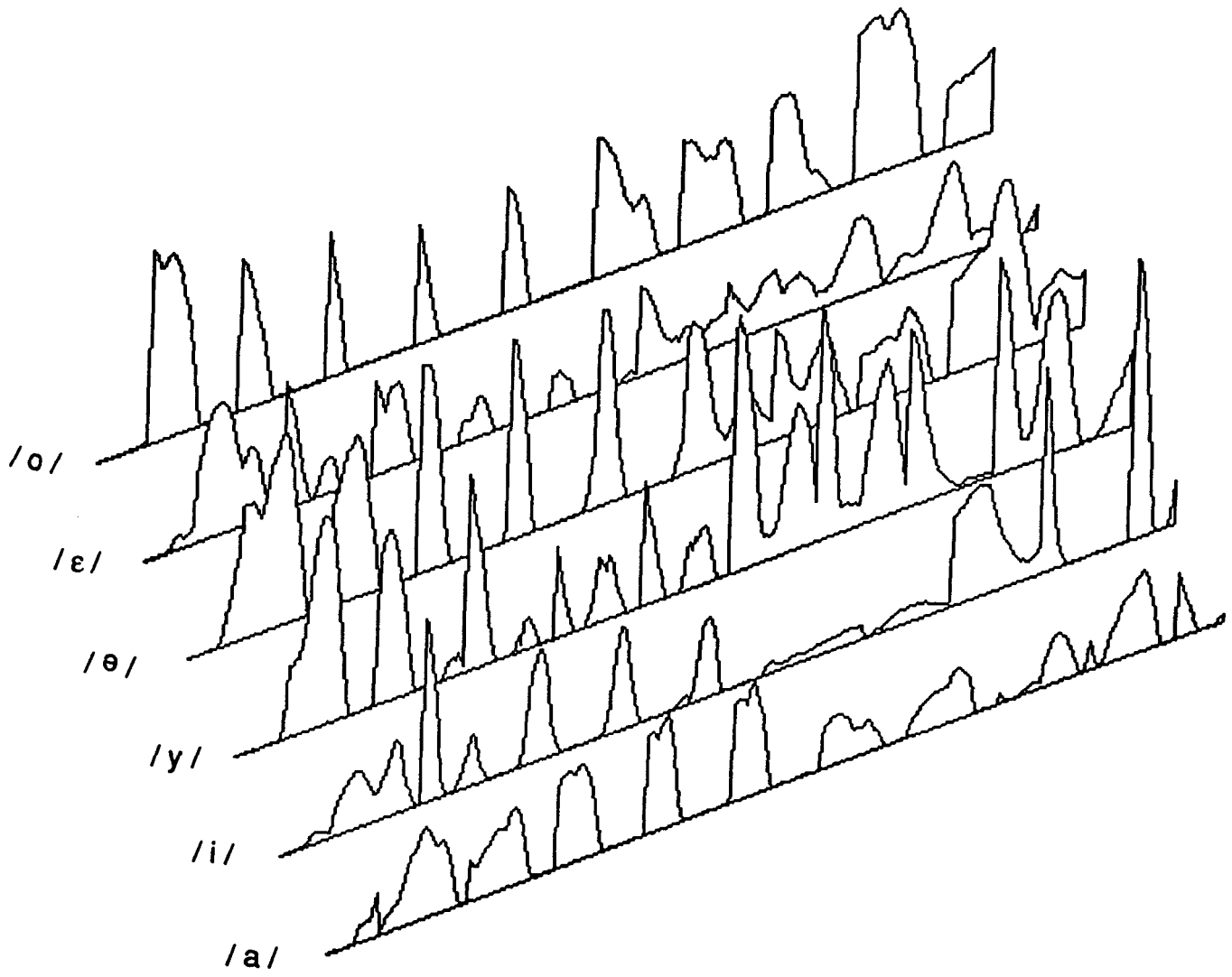


figure 8.1. Similarités pour b.sim.

8.1.1. Signaux présentés et résultats obtenus.

Les signaux que nous avons présenté à DYSTAL pour cette première expérience sont en fait les sons de la prononciation des 26 lettres de l'alphabet. Nous avons donc obtenu, pour chaque son, un fichier de similarité représentant la sortie des 6 neurones de DYSTAL en fonction du temps. Nous avons donc là une mesure de la similarité de chaque son avec chacun des patrons en fonction du temps.

La figure 8.1. représente cette similarité pour la prononciation de la lettre b, soit le son / e /. On y voit chacun des 6 canaux avec la valeur de similarité respective donnée en fonction du temps.

Nous nous sommes intéressé lors de cette première expérience à plusieurs informations pouvant être tirées du fichier de similarité. La première information qui nous intéresse est la moyenne de similarité dans chacuns des canaux (neurones) de notre système en fonction des sons qui sont présentés en entrée. Le tableau 8.1 montre cette moyenne. Le chiffre en gras montre la moyenne maximum trouvée pour chacun des 6 canaux. La moyenne dont nous parlons est calculée en prenant le signal en entier ou un segment de signal (chiffres en italique), et en faisant la moyenne de toutes les valeurs de similarités comprises dans le signal ou dans le segment de signal.

Pour les lettres prononcées qui contiennent plus d'un son, par exemple la lettre f ou la lettre c, contenant chacunes une voyelle et une consonne, il est intéressant de suivre l'évolution des moyennes en fonction du temps. Nous

avons donc développé un autre utilitaire "inter", qui calcule cette moyenne sur des segments de signal (200 échantillons). Les valeurs en italiques sur le tableau 8.1 représentent ces "inter-moyennes", pour les lettres contenant plus d'un son (toutes les lettres sauf les voyelles a, e, i, o, et u).

La figure 8.2 montre les sons ayant eu la moyenne la plus forte et le patron auquel ils sont associés.

Tableau 8.1. Moyenne des similarités en fonction des sons présentés.

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. /ə /	sim. / ɛ /	sim./o/
a	**0.387**	0.042	0.019	0.178	0.099	0.137
b	0.052	0.102	**0.143**	0.123	0.084	0.101
	0.021	***0.059***	*0.050*	*0.032*	*0.035*	*0.023*
	0.076	*0.052*	*0.111*	***0.126***	*0.062*	*0.081*
	0.076	*0.108*	***0.187***	*0.178*	*0.106*	*0.160*
	0.049	***0.238***	*0.127*	*0.125*	*0.124*	*0.108*
	0.050	*0.092*	***0.235***	*0.180*	*0.118*	*0.153*
	0.023	*0.014*	***0.158***	*0.065*	*0.032*	*0.051*
c	0.035	**0.131**	0.064	0.054	0.059	0.052
	0.018	***0.108***	*0.019*	*0.009*	*0.032*	*0.002*
	0.016	***0.068***	*0.009*	*0.006*	*0.019*	*0.001*
	0.037	***0.138***	*0.070*	*0.059*	*0.081*	*0.041*
	0.050	***0.263***	*0.080*	*0.095*	*0.076*	*0.123*
	0.062	*0.045*	***0.185***	*0.130*	*0.105*	*0.117*
d	0.047	0.100	**0.135**	0.115	0.116	0.071
	0.049	***0.083***	*0.082*	*0.061*	*0.072*	*0.058*
	0.036	*0.072*	***0.146***	*0.089*	*0.084*	*0.050*
	0.068	***0.183***	*0.142*	*0.156*	*0.180*	*0.090*
	0.043	*0.087*	***0.180***	*0.168*	*0.158*	*0.100*
	0.018	*0.019*	***0.104***	*0.078*	*0.027*	*0.021*
e	0.167	0.089	0.098	**0.173**	0.124	0.155
f	0.086	0.144	0.126	**0.190**	0.170	0.170
	0.088	***0.247***	*0.103*	*0.182*	*0.193*	*0.142*

	0.148	*0.129*	*0.230*	*0.358*	*0.266*	***0.362***
	0.040	*0.072*	*0.065*	*0.070*	***0.077***	*0.064*
	0.035	*0.096*	*0.074*	*0.077*	***0.097***	*0.008*
g	0.031	**0.116**	0.102	0.074	0.077	0.072
	0.012	***0.050***	*0.029*	*0.025*	*0.036*	*0.012*
	0.054	*0.121*	***0.150***	*0.129*	*0.092*	*0.116*
	0.027	***0.155***	*0.087*	*0.037*	*0.116*	*0.018*
	0.022	***0.159***	*0.084*	*0.035*	*0.048*	*0.101*
	0.045	*0.118*	***0.146***	*0.131*	*0.100*	*0.112*
	0.032	*0.024*	***0.154***	*0.124*	*0.062*	*0.083*
h	0.095	**0.136**	0.061	0.095	0.097	0.051
	0.140	*0.128*	*0.065*	*0.131*	*0.081*	*0.042*
	0.213	*0.171*	*0.109*	*0.198*	*0.135*	*0.097*
	0.090	*0.071*	*0.059*	*0.116*	*0.102*	***0.119***
	0.020	***0.167***	*0.036*	*0.019*	*0.084*	*0.002*
	0.022	***0.153***	*0.039*	*0.023*	*0.096*	*0.001*
	0.006	***0.041***	*0.015*	*0.003*	*0.012*	*0.003*
i	0.023	**0.116**	0.093	0.063	0.056	0.034
j	0.030	**0.101**	0.077	0.057	0.070	0.036
	0.034	*0.034*	***0.069***	*0.047*	*0.045*	*0.024*
	0.049	*0.108*	***0.123***	*0.093*	*0.114*	*0.065*
	0.023	***0.178***	*0.033*	*0.029*	*0.083*	*0.012*
	0.019	***0.158***	*0.036*	*0.034*	*0.081*	*0.022*
	0.021	*0.092*	***0.147***	*0.094*	*0.055*	*0.057*
	0.029	***0.075***	*0.071*	*0.059*	*0.062*	*0.040*
	0.040	*0.037*	***0.045***	*0.035*	*0.032*	*0.036*
k	**0.266**	0.067	0.085	0.195	0.115	0.126
	0.168	*0.069*	*0.133*	***0.188***	*0.127*	*0.106*
	0.384	*0.055*	*0.078*	*0.215*	*0.113*	*0.150*
	0.243	*0.077*	*0.039*	*0.179*	*0.102*	*0.120*
l	0.076	0.111	0.124	**0.157**	0.118	0.109
	0.039	***0.143***	*0.064*	*0.084*	*0.083*	*0.072*
	0.055	***0.195***	*0.090*	*0.106*	*0.122*	*0.070*
	0.162	*0.139*	*0.226*	***0.250***	*0.205*	*0.190*
	0.096	*0.079*	*0.152*	***0.274***	*0.148*	*0.187*
	0.043	*0.029*	***0.110***	*0.104*	*0.057*	*0.040*
	0.023	*0.007*	***0.043***	*0.039*	*0.027*	*0.027*
m	0.074	0.097	**0.142**	**0.142**	0.120	0.099
	0.100	*0.204*	*0.124*	*0.182*	***0.211***	*0.090*
	0.188	*0.127*	*0.189*	***0.213***	*0.197*	*0.162*
	0.028	*0.094*	***0.170***	*0.131*	*0.065*	*0.096*
	0.021	*0.037*	***0.150***	*0.121*	*0.072*	*0.097*

	0.030	*0.017*	***0.075***	*0.062*	*0.051*	*0.048*
n	0.100	0.120	0.127	**0.152**	0.106	0.112
	0.122	***0.216***	*0.103*	*0.186*	*0.162*	*0.182*
	0.201	*0.235*	*0.143*	***0.251***	*0.205*	*0.227*
	0.155	*0.096*	*0.159*	***0.195***	*0.104*	*0.109*
	0.035	*0.068*	***0.175***	*0.126*	*0.050*	*0.052*
	0.024	*0.034*	***0.097***	*0.070*	*0.050*	*0.039*
	0.038	*0.034*	***0.059***	*0.042*	*0.034*	*0.026*
o	0.163	0.050	0.141	0.228	0.252	**0.287**
p	0.060	0.167	**0.168**	0.136	0.150	0.073
	0.076	*0.189*	*0.171*	*0.162*	***0.199***	*0.070*
	0.065	***0.222***	*0.160*	*0.115*	*0.157*	*0.070*
	0.024	*0.028*	***0.178***	*0.130*	*0.053*	*0.082*
q	0.050	0.081	**0.166**	0.118	0.090	0.045
	0.041	*0.090*	***0.106***	*0.059*	*0.073*	*0.005*
	0.079	*0.116*	***0.212***	*0.166*	*0.129*	*0.066*
	0.036	*0.051*	***0.187***	*0.139*	*0.076*	*0.068*
	0.018	*0.020*	***0.124***	*0.078*	*0.042*	*0.023*
r	0.163	0.066	0.080	**0.174**	0.125	0.128
	0.208	*0.114*	*0.025*	*0.082*	*0.076*	*0.052*
	0.251	*0.066*	*0.056*	*0.154*	*0.107*	*0.108*
	0.232	*0.067*	*0.131*	***0.371***	*0.270*	*0.306*
	0.089	*0.047*	*0.113*	***0.190***	*0.129*	*0.148*
	0.032	*0.032*	***0.076***	*0.068*	*0.037*	*0.020*
s	0.073	0.105	0.109	**0.142**	0.128	0.108
	0.090	*0.138*	*0.110*	***0.195***	*0.162*	*0.174*
	0.149	*0.112*	*0.224*	***0.313***	*0.248*	*0.206*
	0.080	*0.070*	*0.152*	***0.172***	*0.123*	*0.153*
	0.018	***0.091***	*0.017*	*0.007*	*0.031*	*0.001*
	0.026	***0.116***	*0.041*	*0.019*	*0.074*	*0.000*
t	0.043	0.091	**0.142**	0.117	0.111	0.072
	0.042	*0.069*	*0.088*	*0.073*	***0.096***	*0.027*
	0.068	*0.185*	*0.130*	*0.148*	***0.200***	*0.100*
	0.035	*0.049*	***0.238***	*0.168*	*0.081*	*0.109*
	0.015	*0.031*	***0.086***	*0.047*	*0.028*	*0.035*
u	0.078	0.051	**0.260**	0.173	0.168	0.027
v	0.084	0.083	**0.177**	0.165	0.112	0.141
	0.086	*0.070*	***0.185***	*0.152*	*0.086*	*0.157*
	0.089	*0.047*	***0.141***	***0.141***	*0.077*	*0.109*
	0.125	*0.148*	*0.189*	***0.205***	*0.155*	*0.157*
	0.067	*0.087*	*0.204*	***0.205***	*0.161*	*0.183*

	0.021	*0.038*	***0.151***	*0.079*	*0.050*	*0.054*
w	0.075	0.085	**0.176**	0.165	0.110	0.134
	0.051	*0.064*	***0.096***	*0.081*	*0.083*	*0.056*
	0.069	*0.096*	*0.113*	***0.118***	*0.101*	*0.089*
	0.054	*0.110*	***0.113***	*0.105*	*0.093*	*0.060*
	0.119	*0.102*	*0.178*	***0.181***	*0.117*	*0.152*
	0.126	*0.072*	***0.326***	*0.293*	*0.171*	*0.273*
	0.060	*0.058*	*0.179*	***0.202***	*0.097*	*0.171*
	0.041	*0.100*	***0.222***	*0.165*	*0.101*	*0.125*
	0.046	*0.052*	***0.155***	*0.109*	*0.057*	*0.029*
x	0.023	**0.130**	0.055	0.033	0.043	0.018
	0.023	***0.236***	*0.006*	*0.006*	*0.046*	*0.007*
	0.024	***0.187***	*0.093*	*0.051*	*0.051*	*0.040*
	0.037	*0.046*	***0.140***	*0.086*	*0.054*	*0.037*
	0.019	***0.090***	*0.025*	*0.012*	*0.033*	*0.002*
	0.014	***0.087***	*0.011*	*0.008*	*0.032*	*0.001*
y	0.060	0.116	**0.159**	0.120	0.138	0.074
	0.051	***0.198***	*0.092*	*0.062*	*0.087*	*0.036*
	0.071	*0.120*	***0.181***	*0.106*	*0.151*	*0.069*
	0.058	*0.133*	***0.193***	*0.155*	*0.172*	*0.109*
	0.112	*0.109*	*0.227*	***0.241***	*0.228*	*0.166*
	0.063	*0.117*	***0.179***	*0.150*	*0.162*	*0.084*
	0.029	*0.040*	***0.117***	*0.050*	*0.059*	*0.020*
	0.023	*0.084*	***0.092***	*0.035*	*0.086*	*0.005*
z	0.053	0.075	**0.128**	0.114	0.088	0.091
	0.035	*0.050*	*0.059*	*0.059*	***0.067***	*0.059*
	0.014	***0.058***	*0.017*	*0.008*	*0.015*	*0.008*
	0.075	*0.141*	***0.198***	*0.168*	*0.116*	*0.119*
	0.110	*0.154*	*0.206*	***0.222***	*0.171*	*0.159*
	0.105	*0.054*	*0.221*	***0.235***	*0.170*	*0.216*
	0.024	*0.034*	***0.125***	*0.080*	*0.060*	*0.066*
	0.024	*0.046*	***0.100***	*0.060*	*0.033*	*0.028*
	0.016	*0.038*	***0.046***	*0.027*	*0.036*	*0.031*

/a/	/i/	/y/	/ə/	/ɛ/	/o/
a k	c g h i j x	b d m p q t u v w y x	e f l m n r s		o

figure 8.2. Moyennes maximales.

Une autre information qui peut être intéressante est l'aire sous la courbe de similarité de chacuns des sons. Nous considérons cette information plus pertinente que la moyenne du signal, puisqu'il s'agit d'une mesure directe de la similarité trouvée en fonction des canaux. Le tableau 8.2 présente cette aire en pourcentage relatif. L'aire est calculée à partir de la formule suivante.

$Aire_i = [\Sigma_j sim_{i,j}] / Aire\ totale$

où $sim_{i,j}$ est la valeur de similarité associée au canal i et de coordonnée j sur l'échelle de l'image statique (c.f. section 7.4). j varie de 1 à n, n étant la dernière coordonnée du vecteur Sim_i.

L'aire totale est la sommation de toutes les similarités de tous les canaux. Les segments de parole que nous étudions actuellement étant constitués de

quelques sons seulement, nous avons calculé l'aire sous la courbe pour l'ensemble du signal. Dans le cas d'une analyse sur de la parole continue, il est évident que ce calcul devra être fait en considérant une fenêtre de largeur fixe.

tableau 8.2. Aire sous la courbe.

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. /ə /	sim. / ɛ /	sim./o/
a	**44.85**	4.97	2.17	20.62	11.50	15.90
b	8.55	16.86	**23.63**	20.35	13.96	16.65
c	8.76	**33.10**	16.14	13.73	15.04	13.22
d	8.01	17.10	**23.11**	19.78	19.89	12.11
e	20.69	11.08	12.14	**21.49**	15.38	19.21
f	9.70	16.20	14.26	**21.44**	19.20	19.19
g	6.74	**24.39**	21.55	15.68	16.35	15.29
h	17.77	**25.37**	11.32	17.79	18.18	9.57
i	6.07	**30.12**	24.06	16.32	14.49	8.93
j	8.13	**27.25**	20.67	15.40	18.76	9.80
k	**31.18**	7.82	10.01	22.81	13.46	14.73
l	10.95	16.01	17.85	**22.59**	16.98	15.63
m	10.94	14.35	21.11	**21.13**	17.80	14.66
n	13.97	16.69	17.75	**21.27**	14.74	15.58
o	14.52	4.45	12.55	20.38	22.51	**25.59**
p	7.97	22.07	**22.29**	18.08	19.94	9.65
q	9.06	14.80	**30.11**	21.53	16.32	8.18
r	22.21	8.93	10.93	**23.65**	16.94	17.35
s	10.96	15.81	16.42	**21.34**	19.29	16.19
t	7.51	15.76	**24.65**	20.31	19.29	12.48
u	10.31	6.71	**34.34**	22.86	22.24	3.54
v	11.07	10.85	**23.17**	21.69	14.72	18.51

w	10.11	11.40	**23.56**	22.21	14.79	17.92
x	7.74	**42.87**	18.37	10.89	14.29	5.84
y	9.04	17.43	**23.77**	17.91	20.72	11.12
z	9.70	13.61	**23.36**	20.86	15.97	16.50

8.1.2. Interprétation des résultats.

Regardons ces données d'un peu plus près, pour tâcher de comprendre leur signification. L'interprétation n'en est certes pas très aisée, mais nous tâcherons d'établir des règles générales.

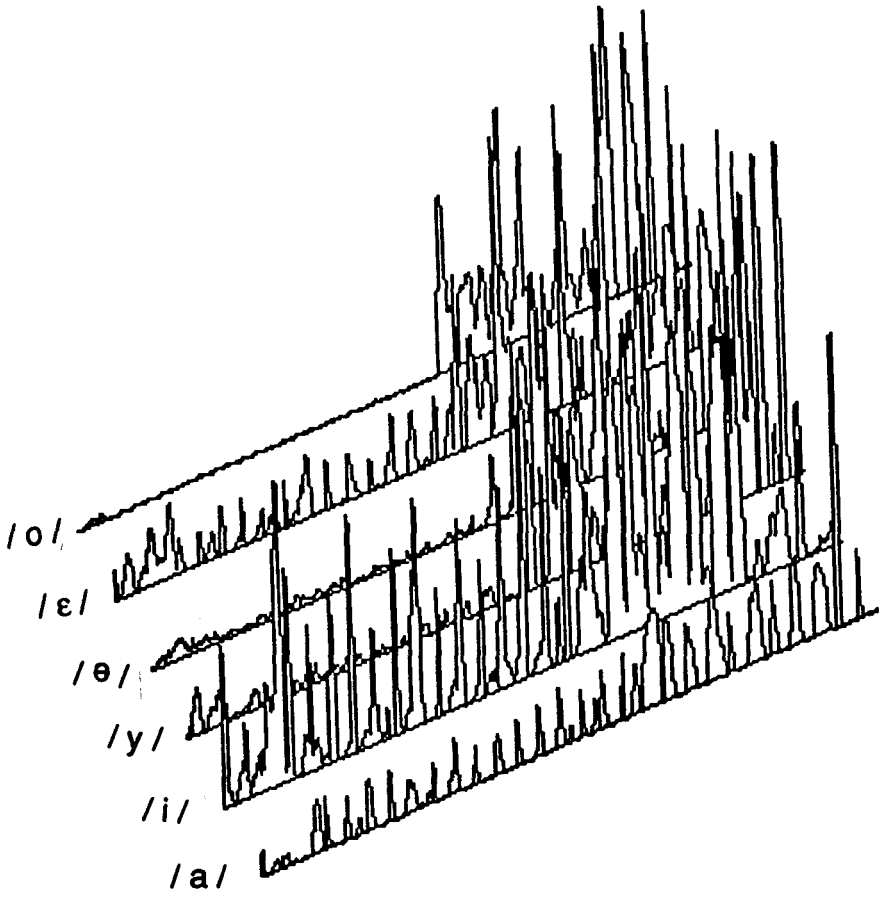


figure 8.3. Sortie, fichier c.sim.

Nous remarquons que le / a / n'est jamais confondu. Nous n'avons aucun problème à le différencier des autres sons. Le / a / des lettres a et k est toujours reconnu sans faute.

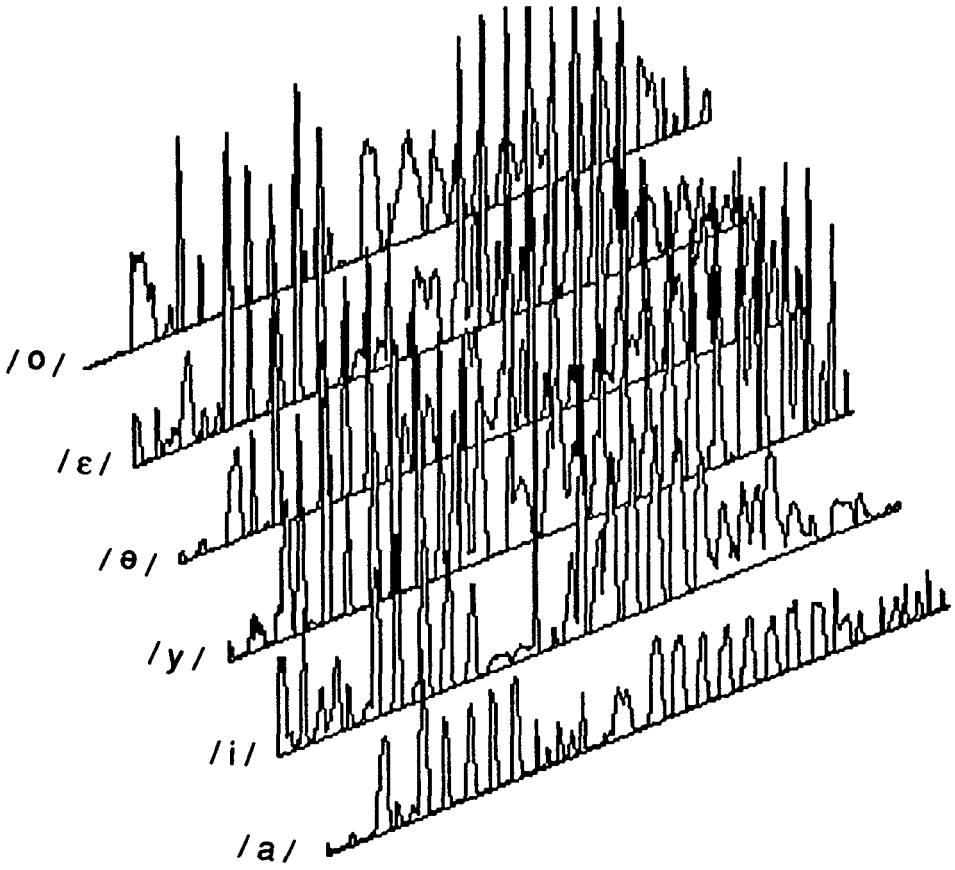


figure 8.4. Sortie, fichier d.sim.

Le / i / ne semble pas non plus causer de problèmes, à en juger par les similarités intermédiaires des lettres i, j, x et y. Nous voyons à un moment ou à un autre une bonne similarité dans le canal du / i /.

Le son / e /, quand à lui, se trouve réparti entre le / i / et le / y /. En effet, le 2ième formant de /e/ est à mi-chemin entre le 2ième formant de /i/ et le 2ième formant de /y/. Il semble y avoir un lien entre le son fricatif qui précède le / e / et le patron reconnu. Nous remarquons qu'une consonne frictive suivie du son / e / est reconnue comme un / i / et qu'une consonne occlusive suivie du son / e / est reconnue comme un / y /. Cela reste évidemment à vérifier avec d'autres expériences. On peut voir sur les figures 8.3 et 8.4 la similarité de sortie des fichiers c.sim et d.sim. Nous remarquons dans la partie fricative du c une similarité assez importante dans le canal 2, soit celui du / i /. Les oscillations entre le / e / et le / y / sont bien mises en évidence dans le détail des valeurs moyennes (tableau 8.1).

Le / e / et le / ε / sont peu ou pas différenciés. La confusion entre le / e / et le / ε / peut cependant s'expliquer. Nous rappelons que notre système accentue la différence de fréquence entre deux formants. Voyons ces différences:

/ e /: F1-F2: 900Hz
F2-F3: 900Hz
F3-F4: 1100Hz

/ ε /: F1-F2: 1400Hz
F2-F3: 800Hz
F3-F4: 1000Hz

Nous observons ici une différence notable seulement au niveau de F1-F2. Comme nos filtres de basses fréquences n'englobent pas une aussi grande différence de fréquence (voir chapitre 4, figure 4.1), il est normal de confondre le

/ e / et le / ε /, étant donné que F2-F3 et F3-F4 se ressemblent. Nous vérifierons ces affirmations dans les autres expériences.

Les autres sons ne semblent pas poser un problème majeur à notre système de reconnaissance.

Il est intéressant de constater que la prononciation des lettres contenant plusieurs sons est en fait de la parole continue. On peut voir les similarités osciller entre plusieurs valeurs en fonction du temps. Ceci pourrait constituer un critère d'analyse pour un éventuel étage supérieur de notre système.

8.2. Deuxième expérience, premier homme, deuxième couche du réseau de neurones.

Le but de cette expérience est de tester la pertinence de la deuxième couche de notre réseau.

Nous avons ici présenté les fichiers sortie de la première couche à l'entrée de la deuxième couche décrite précédemment. Nous nous sommes servis ici encore des utilitaires csup et moyenne pour interpréter les données contenues dans les fichiers de sortie de la couche supérieure. Des patrons ont été crées pour les mêmes voyelles que dans la première expérience.

8.2.1. Résultats obtenus.

Tableau 8.3. Moyenne des similarités en fonction des sons présentés, deuxième couche

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. /ə/	sim. / ε /	sim./o/
a	0.438	0.000	0.073	0.350	0.058	0.276
b	0.049	0.131	0.237	0.180	0.125	0.169
c	0.023	0.405	0.053	0.068	0.136	0.035
d	0.059	0.136	0.275	0.204	0.160	0.149
e	0.207	0.081	0.065	0.251	0.065	0.172
f	0.070	0.172	0.184	0.239	0.130	0.277
g	0.008	0.267	0.142	0.091	0.145	0.068
h	0.125	0.285	0.106	0.185	0.142	0.109
i	0.018	0.244	0.177	0.100	0.135	0.066
j	0.017	0.244	0.146	0.091	0.145	0.044
k	0.315	0.006	0.087	0.310	0.072	0.226
l	0.082	0.129	0.214	0.227	0.133	0.211
m	0.080	0.083	0.278	0.212	0.132	0.206
n	0.088	0.075	0.229	0.198	0.123	0.169
o	0.159	0.000	0.196	0.335	0.103	0.513
p	0.053	0.275	0.305	0.199	0.173	0.106
q	0.058	0.109	0.340	0.197	0.155	0.119
r	0.217	0.025	0.162	0.313	0.088	0.256
s	0.055	0.236	0.130	0.166	0.140	0.183
t	0.057	0.123	0.318	0.209	0.154	0.158
u	0.063	0.087	0.356	0.224	0.156	0.169
v	0.063	0.052	0.256	0.226	0.114	0.230
w	0.062	0.063	0.264	0.208	0.125	0.226
x	0.012	0.354	0.113	0.050	0.139	0.017
y	0.060	0.130	0.301	0.206	0.163	0.150
z	0.040	0.114	0.204	0.153	0.124	0.149

Comme dans la première expérience, nous donnons ici l'information que nous considérons la plus pertinente, à savoir l'aire sous la courbe en pourcentage relatif (tableau 8.4). À titre de comparaison, nous donnons aussi l'aire sous la courbe en pourcentage relatif obtenue dans la première expérience (chiffre entre parenthèses).

tableau 8.4. Aire sous la courbe en pourcentage relatif, deuxième couche.

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. /ə/	sim. / ɛ /	sim./o/
a	**36.69**	0.03	6.10	29.20	4.86	23.13
	(45)	(5)	(2)	(21)	(12)	(16)
b	5.55	14.67	**26.58**	20.26	13.97	18.98
	(9)	(17)	**(24)**	(20)	(14)	(17)
c	3.20	**56.31**	7.39	9.45	18.83	4.85
	(9)	**(33)**	(16)	(14)	(15)	(13)
d	5.99	13.85	**27.98**	20.71	16.32	15.09
	(8)	(17)	**(23)**	(20)	(20)	(12)
e	24.60	9.63	7.68	**29.66**	7.74	20.40
	(21)	(11)	(12)	**(21)**	(15)	(19)
f	2.67	**65.49**	7.01	9.10	4.93	10.56
	(10)	(16)	(14)	**(21)**	(19)	(19)
g	1.14	**37.28**	19.76	12.67	20.21	8.89
	(7)	**(24)**	(22)	(16)	(16)	(15)
h	13.20	**29.91**	11.14	19.39	14.95	11.44
	(18)	**(25)**	(11)	(18)	(19)	(10)
i	2.46	**33.00**	23.8	13.56	18.26	8.94
	(6)	**(30)**	(24)	(16)	(14)	(9)
j	2.54	**35.45**	21.27	13.20	21.15	6.43
	(8)	**(27)**	(21)	(15)	(19)	(10)

k	**31.21**	0.55	8.59	30.77	7.16	21.8
	(31)	(8)	(10)	(23)	(13)	(15)
l	8.24	12.93	21.44	**22.84**	13.36	21.12
	(11)	(16)	(18)	**(23)**	(17)	(16)
m	8.05	8.41	**27.99**	21.37	13.36	20.79
	(11)	(14)	(21)	**(21)**	(18)	(15)
n	10.0	8.54	**25.97**	22.44	13.90	19.13
	(14)	(17)	(18)	**(21)**	(15)	(16)
o	12.22	0.00	14.99	25.59	7.87	**39.35**
	(15)	(4)	(13)	(20)	(23)	**(26)**
p	4.78	24.83	**27.38**	17.91	15.57	9.56
	(8)	(22)	**(22)**	(18)	(20)	(10)
q	5.97	11.13	**34.72**	20.24	15.83	12.16
	(9)	(15)	**(30)**	(22)	(16)	(8)
r	20.43	2.34	15.22	**29.46**	8.32	24.24
	(22)	(9)	(11)	**(24)**	(17)	(17)
s	6.06	**25.98**	14.32	18.25	15.34	20.03
	(11)	(16)	(16)	**(21)**	(19)	(16)
t	5.55	12.10	**31.30**	20.47	15.16	15.47
	(8)	(16)	**(25)**	(20)	(19)	(12)
u	5.99	8.22	**33.75**	21.21	14.81	15.94
	(10)	(7)	**(34)**	(23)	(22)	(4)
v	6.79	5.52	**27.17**	24.00	12.12	24.41
	(11)	(11)	**(23)**	(22)	(15)	(19)
w	6.52	6.63	**27.81**	21.95	13.13	23.83
	(10)	(11)	**(24)**	(22)	(15)	(18)
x	1.71	**51.76**	16.55	7.26	20.34	2.42
	(8)	**(43)**	(18)	(11)	(14)	(6)
y	5.89	12.91	**29.81**	20.40	16.15	14.82
	(9)	(17)	**(24)**	(18)	(21)	(11)
z	5.08	14.54	**25.97**	19.55	15.86	18.98
	(10)	(14)	**(23)**	(21)	(16)	(17)

y	5.89	12.91	**29.81**	20.40	16.15	14.82
	(9)	(17)	**(24)**	(18)	(21)	(11)
z	5.08	14.54	**25.97**	19.55	15.86	18.98
	(10)	(14)	**(23)**	(21)	(16)	(17)

8.2.2. Interprétation des résultats.

À première vue, rien de nouveau ne peut être obtenu avec la deuxième couche. Il semble cependant que les écarts entre les similarités sont accentués, mais les catégories de sons reconnus (voir figure 8.2) restent sensiblement les mêmes. Dans les prochaines expériences, nous laisserons de côté les sons des lettres w et y, car les mesures de moyennes et d'aire pour ces sons ne sont pas significatives, étant donné qu'il s'agit ici de sons composés. Nous les analyserons lorsque nous aborderons la section sur la parole continue.

8.3. Troisième expérience, deuxième homme, première couche du réseau Dystal.

Nous répétons ici l'expérience décrite dans la section 8.1, mais avec un autre locuteur homme. Voyons les résultats que nous obtenons. Nous ne montrons pas ici les valeurs de similarités moyennes, seulement les valeurs d'aire sous la courbe en pourcentage relatif, étant donné que ces deux informations sont redondantes. Il va sans dire que Dystal n'avait jamais effectué d'apprentissage sur les signaux présentés.

8.3.1. Résultats obtenus.

Le tableau 8.5 donne l'aire sous la courbe en pourcentage relatif pour les sons analysés. Les chiffres en gras représentent les plus gros pourcentages pour un son donné. Nous croyons bon aussi de donner les moyennes intermédiaires pour quelques sons (tableau 8.6). Ces valeurs peuvent être comparées à celles du tableau 8.1.

tableau 8.5. Aire sous la courbe, troisième expérience

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. /ə/	sim. / ɛ /	sim./o/
a	**41.40**	5.50	2.82	26.45	13.47	10.36
b	11.63	**22.49**	19.72	15.83	17.38	12.95
c	7.53	**39.82**	16.81	13.05	16.66	6.13
d	10.21	**28.31**	16.52	14.78	16.92	13.25
e	17.55	9.38	17.94	**22.98**	17.61	14.53
f	10.27	28.47	17.64	12.84	**24.61**	6.17
g	7.42	**41.54**	15.29	10.06	19.62	6.07
h	17.90	22.79	10.76	17.04	**22.94**	8.58
i	5.63	**52.14**	8.22	5.42	24.37	4.22
j	6.81	**46.65**	11.23	5.20	26.57	3.54
k	**30.31**	8.17	7.14	25.48	14.89	14.00
l	11.71	**21.87**	14.02	20.07	19.00	13.34
m	14.17	12.81	**23.12**	20.02	18.64	11.25
n	11.81	14.44	**21.88**	21.46	17.81	12.60
o	12.73	1.55	11.91	**27.20**	20.61	25.99
p	9.11	**32.07**	21.76	11.52	19.28	6.26
q	8.06	15.00	**33.64**	10.66	30.24	2.40
r	16.60	9.43	13.80	**24.96**	19.49	15.71

s	11.95	**28.35**	14.19	14.74	21.48	9.28
t	6.47	**44.83**	13.36	9.51	18.53	7.31
u	11.48	9.49	**32.36**	18.12	22.96	5.60
v	7.42	**34.46**	17.94	14.48	17.12	8.58
x	8.34	**52.38**	10.22	4.62	22.60	1.84
z	11.24	18.70	**20.80**	19.47	16.54	13.25

Tableau 8.6. Moyennesintermédiaires des similarités en fonction des sons présentés.

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. /ə/	sim. / ε /	sim./o/
b						
	0.110	*0.033*	*0.107*	***0.131***	*0.119*	*0.117*
	0.115	*0.067*	*0.135*	*0.135*	***0.150***	*0.116*
	0.033	***0.197***	*0.095*	*0.029*	*0.102*	*0.021*
	0.028	***0.226***	*0.135*	*0.064*	*0.047*	*0.051*
	0.040	***0.118***	*0.081*	*0.079*	*0.070*	*0.054*
	0.043	*0.016*	*0.084*	***0.120***	*0.079*	*0.083*
c						
	0.022	***0.147***	*0.018*	*0.011*	*0.060*	*0.002*
	0.040	***0.162***	*0.061*	*0.049*	*0.091*	*0.031*
	0.019	***0.189***	*0.067*	*0.048*	*0.053*	*0.031*
	0.043	***0.145***	*0.137*	*0.114*	*0.066*	*0.037*
d						
	0.120	*0.023*	*0.140*	***0.208***	*0.166*	*0.194*
	0.082	*0.084*	*0.127*	*0.128*	***0.131***	*0.122*
	0.028	***0.235***	*0.054*	*0.011*	*0.057*	*0.008*
	0.027	***0.305***	*0.055*	*0.018*	*0.060*	*0.012*
	0.027	***0.156***	*0.094*	*0.047*	*0.060*	*0.031*
f						
	0.082	*0.159*	*0.122*	*0.162*	***0.168***	*0.106*
	0.078	***0.152***	*0.081*	*0.100*	*0.124*	*0.052*
	0.043	*0.072*	*0.074*	*0.031*	***0.085***	*0.007*
	0.028	***0.127***	*0.077*	*0.021*	*0.090*	*0.006*
	0.028	***0.134***	*0.048*	*0.026*	*0.078*	*0.007*

	0.026	***0.141***	*0.069*	*0.025*	*0.117*	*0.002*
	0.028	*0.082*	*0.072*	*0.023*	***0.094***	*0.004*
g						
	0.030	***0.162***	*0.083*	*0.058*	*0.075*	*0.036*
	0.031	***0.205***	*0.059*	*0.018*	*0.128*	*0.007*
	0.025	***0.190***	*0.028*	*0.011*	*0.085*	*0.007*
	0.022	***0.241***	*0.049*	*0.018*	*0.058*	*0.013*
	0.053	***0.168***	*0.123*	*0.097*	*0.104*	*0.059*
	0.053	*0.050*	*0.070*	***0.126***	*0.052*	*0.077*
o						
	0.219	*0.036*	*0.114*	***0.311***	*0.265*	*0.282*
	0.190	*0.011*	*0.172*	*0.427*	*0.337*	***0.428***
	0.179	*0.023*	*0.224*	***0.468***	*0.333*	*0.452*
	0.156	*0.021*	*0.193*	***0.391***	*0.271*	*0.363*

8.3.2. Interprétation des résultats.

Il sera intéressant de comparer ici les résultats obtenus avec ceux de la première expérience. Voyons un peu les sons reconnus. Nous reproduisons sur la figure 8.3 le contenu de la figure 8.2 en y ajoutant les résultats de la troisième expérience.

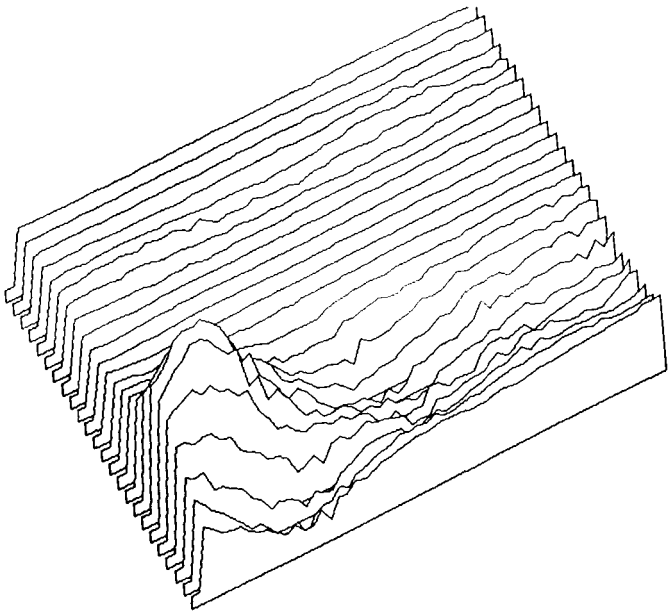
Expérience	/a/	/i/	/y/	/ə/	/ɛ/	/o/
1	a k	c g h i j x	b d m p q t u v w y x z	e f l m n r s		o
3	a k	b c d f g i j l p s t v x	n m q u z	e o r	h	

figure 8.3. Similarités maximales pour les sons des expériences 1 et 3.

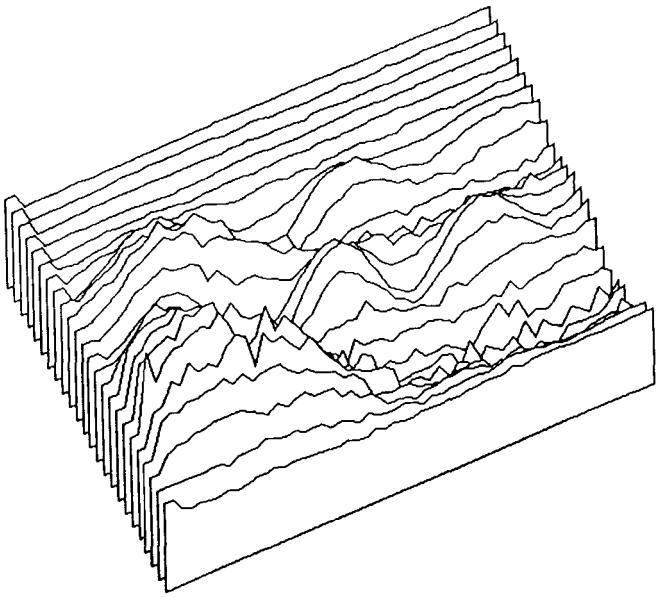
Nous constatons que quelques sons ont changé de place. Les sons qui ont été bien reconnus dans les deux expériences sont / a /, / i /, / y / et / ə /. Nous voyons très clairement sur la figure 8.3 que le son /é/ peut être reconnu comme un / i / ou un / y /, tandis que le son / ɛ /, quant à lui, oscille entre le / u / et le / ə / . Le son / o / a été reconnu globalement comme un / ə / dans la deuxième expérience. Cependant, si nous examinons les valeurs de moyennes intermédiaires, on constate que le / o / est bien reconnu dans un des segments. Nous rappelons que ces résultats doivent être interprétés avec un grain de sel, il faut en effet, pour dire si un son a été reconnu ou pas, examiner en détail les valeurs de moyenne intermédiaires.

La reconnaissance du son / i / dans la prononciation de lettres comme b, c, d, g est bien mise en évidence au tableau 8.6. Le / ɛ / de f est ici mieux reconnu que dans la première expérience.

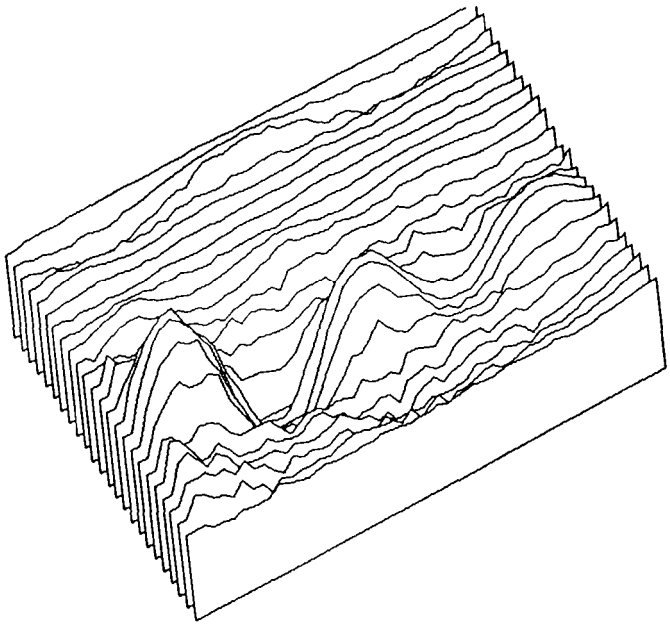
On peut essayer de comprendre pourquoi le / o / est confondu en examinant les patrons de /ə / et de / o / que Dystal avait en mémoire lors de la



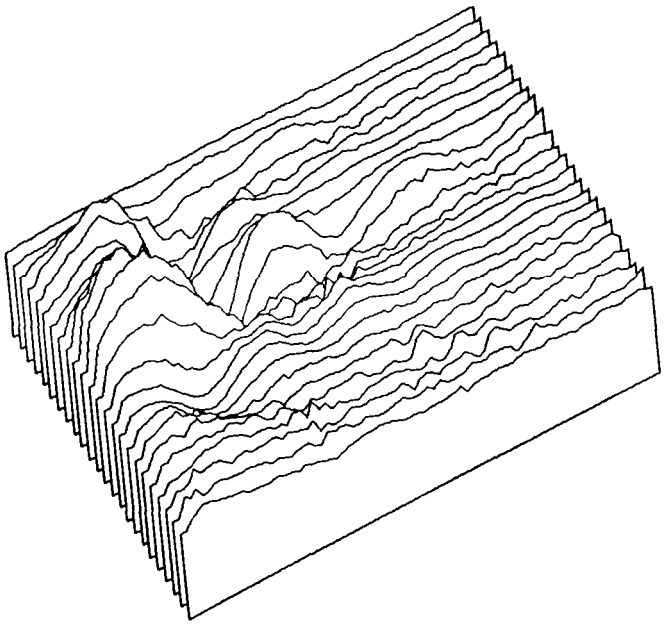
/ə/, première femme



/ə/, deuxième femme

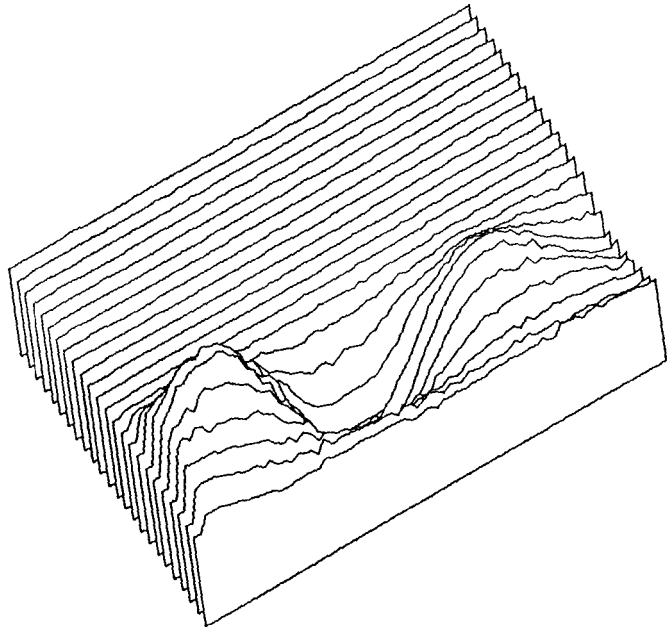


/ə/, premier homme

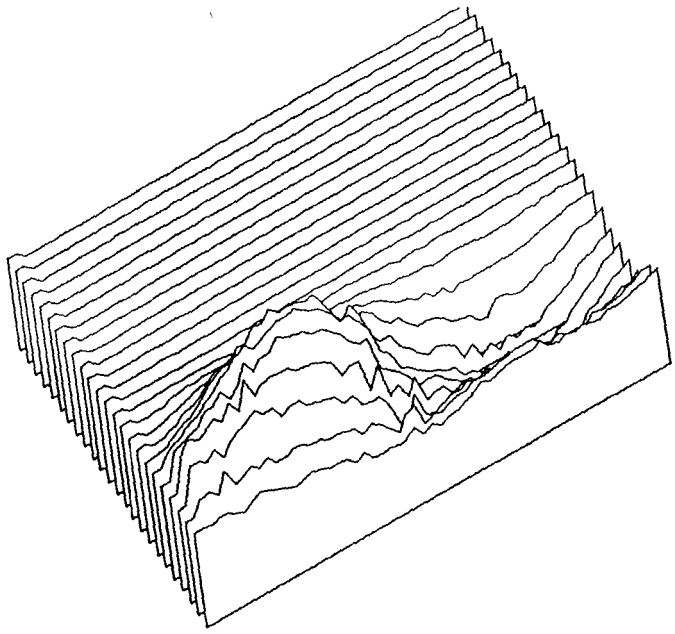


/ə/, deuxième homme

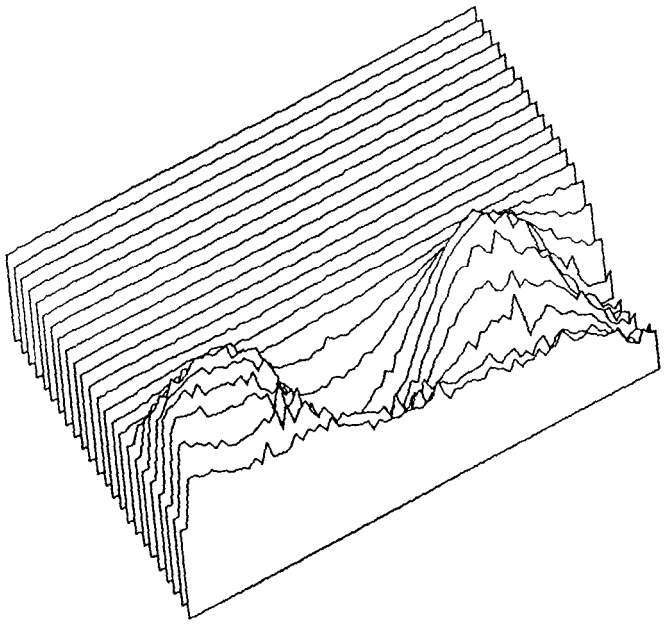
figure 8.4. Patrons de /ə/ utilisés.



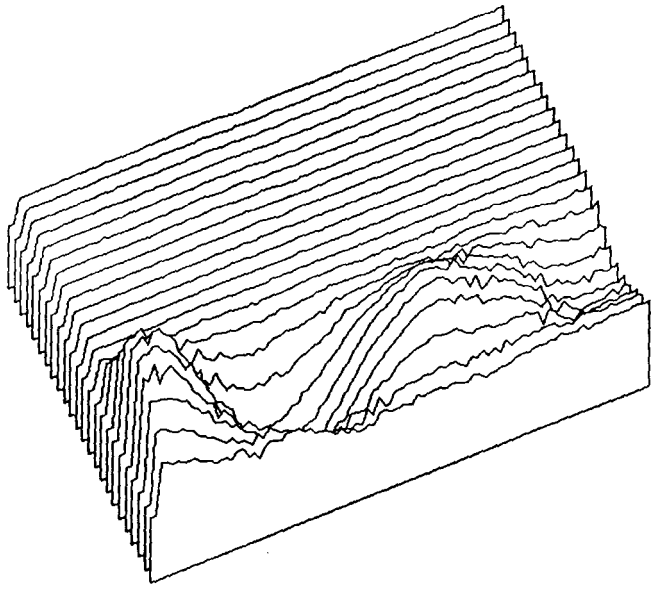
/ o /, première femme



/ o /, deuxième femme



/ o /, premier homme



/ o /, deuxième homme

figure 8.5. Patrons de / o / utilisés.

première et de la troisième expérience. Nous avons effectué ces deux expériences avec 4 patrons de /ə / et 4 patrons de / o /, dont seulement 3 utilisés dans chaque expérience, pour une raison bien simple: Nous ne voulions pas présenter à Dystal les mêmes sons lors de l'apprentissage et de la simulation.

Les patrons sont montrés sur les figures 8.4. et 8.5.

Dystal avait donc appris, dans la première expérience, avec les patrons de la première et de la deuxième femme, ainsi qu'avec ceux du deuxième homme. Par contre, dans la deuxième expérience, il avait appris avec ceux de la première et de la deuxième femme, ainsi que ceux du premier homme.

Pourquoi peut-il y avoir confusion entre le / ə / et le / o /? Nous serions porté à dire, à première vue, que le patron de /e/ de la première femme ressemble drôlement au patron de / o / de la deuxième femme. Si nous éliminons tout simplement ces deux patrons et que nous relançons la simulation pour le /o/, nous obtenons les résultats suivants:

Tableau 8.7. Aire sous la courbe, troisième expérience, après suppression de deux patrons.

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. /ə /	sim. / ɛ /	sim./o/
oh1a	14.52	4.45	12.55	20.38	22.51	**25.59**
oh2a	12.73	1.55	11.91	**27.20**	20.61	25.99
oh1	16.96	5.98	15.64	11.31	**25.94**	24.17
oh2	15.37	2.20	13.59	19.20	**26.03**	23.61

oh1a: Voyelle /o/, premier homme, avant suppression des deux patrons

oh2a: Voyelle /o/, deuxième homme, avant suppression des deux patrons

oh1: Voyelle /o/, premier homme, après suppression des deux patrons

oh2: Voyelle /o/, deuxième homme, après suppression des deux patrons

La suppression des deux patrons a diminué de près de 50% la reconnaissance du / ə /, d'environ 2% la reconnaissance du / o / et a augmenté d'environ 20% la reconnaissance des autres sons. Nous voulons ici souligner la variation de similarité qu'on peut obtenir en ajoutant ou supprimant quelques patrons et nous voulons aussi mettre en évidence l'importance d'un choix judicieux des patrons à utiliser.

8.4. Quatrième expérience. Est-il avantageux de supprimer les 6 premiers canaux de basses fréquences?

On nous a souvent demandé, au cours de ce travail, s'il pouvait être avantageux de supprimer des canaux du banc de filtres qui sont moins important, de façon à éviter d'embrouiller le réseau de neurones avec de l'information inutile. C'est ce que nous vérifierons dans cette expérience.

La question: Pourquoi supprimer les 6 canaux de plus basses fréquences se posera sûrement. La réponse est que l'information de ces canaux n'est peut-être pas pertinente, et que les sons ayant de l'information dans ces canaux (le

/ a /, par exemple) peuvent être reconnus quand-même. Voyons ce que nous obtenons.

8.4.1. Résultats obtenus.

Nous voyons sur le tableau 8.8 les aires sous la courbe pour quelques sons de l'expérience 3.

Tableau 8.8. Aire sous la courbe, quatrième expérience.

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. / ɵ /	sim. / ɛ /	sim./o/
a	**39.66**	4.91	3.49	28.87	9.39	13.67
b	15.70	**28.17**	19.00	14.94	16.11	6.09
i	5.97	**52.99**	7.22	2.98	28.28	2.56
o	15.79	1.10	7.57	**30.67**	14.23	30.63

8.4.2. Interprétation des résultats.

Le / a / est reconnu avec moins de précision que dans la troisième expérience, il fallait s'y attendre, puisqu'on coupe de l'information en basse fréquence. Légère amélioration pour le / i /, 52.99 contre 52.14, pas de

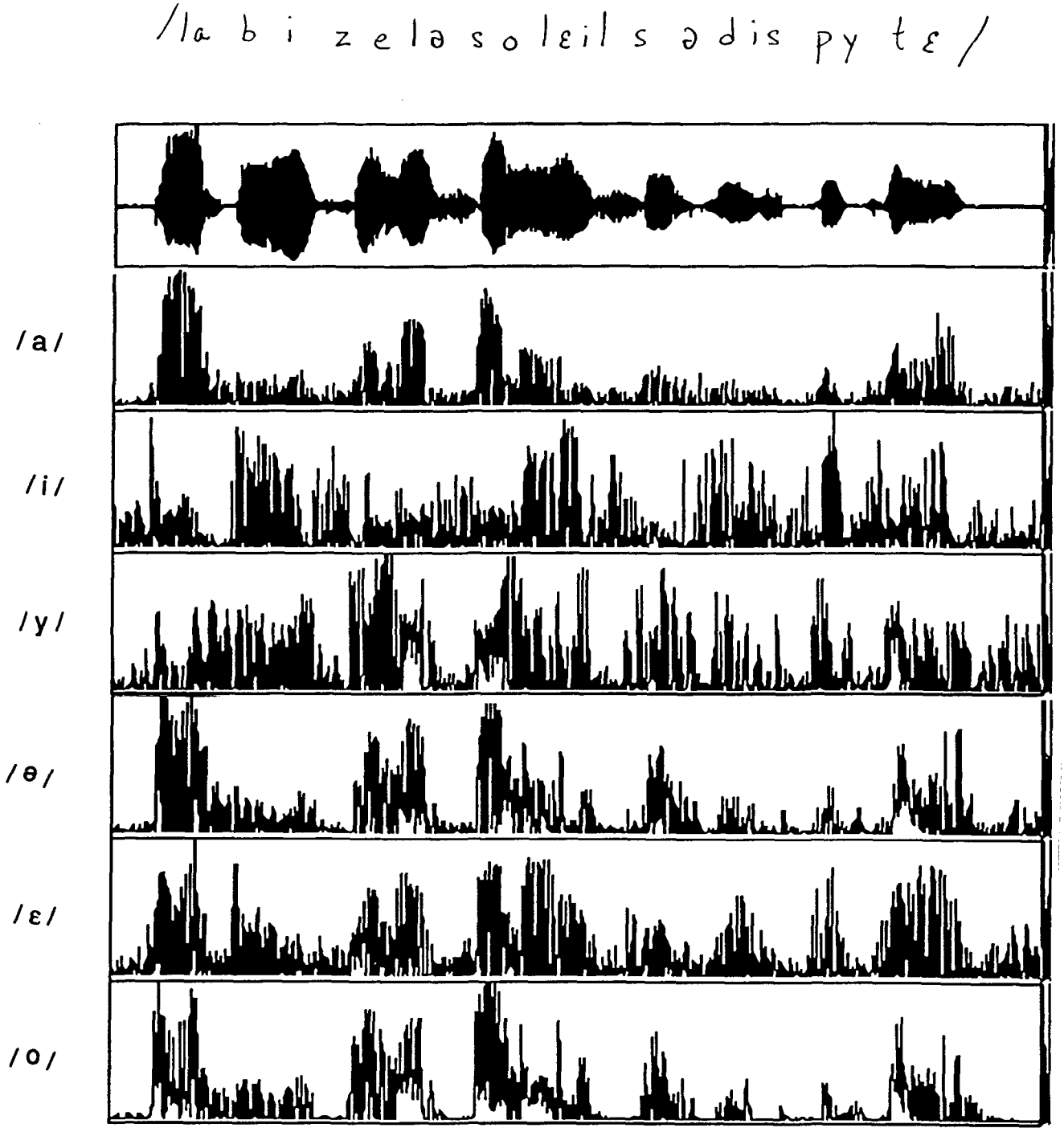


figure 8.5. La parole continue.

différence majeure, bref, rien qui ne laisse penser que supprimer des canaux serait une bonne solution pour optimiser le système. Nous n'émettrons donc aucune recommendation à ce sujet, et nous n'irons pas plus loin sur cette piste.

8.5. Cinquième expérience: la parole continue.

Il nous reste ici à vérifier comment notre système se débrouille lorsqu'il est confronté à de la parole continue. Nous analyserons ici un morceau de parole continue. La phrase analysée est la suivante "La bise et le soleil se disputaient", prononcée par une femme.

8.5.1. Résultats obtenus.

La figure 8.5 montre l'aspect physique du signal de parole ainsi que les similarités dans chacun des six canaux de sortie en fonction du temps. Nous y avons aussi ajouté la phrase écrite ainsi que les voyelles reconnues ayant les moyennes maximales.

Nous donnons aussi au tableau 8.9 les valeurs de moyenne intermédiaires, pour une meilleure compréhension. Sur ce tableau, le symbole ~ est utilisé pour une période de silence.

tableau 8.9. Moyennes intermédiaires pour la phrase: "La bise et le soleil se disputaient".

Son	sim. / a /	sim. / i/	sim. / y /	sim. /ə/	sim. / ɛ /	sim./o/
~	0.004	0.063	0.021	0.012	0.023	0.010
~	0.027	0.108	0.052	0.060	0.066	0.060
/ l a /	0.228	0.096	0.045	**0.282**	0.177	0.252
/ a /	0.204	0.082	0.089	**0.255**	0.197	0.238
~	0.041	0.019	0.103	0.081	0.049	0.076
/ b /	0.029	0.140	0.084	0.055	0.111	0.063
/ i /	0.029	**0.187**	0.095	0.058	0.094	0.065
/ i /	0.034	**0.143**	0.091	0.057	0.079	0.070
/ i /	0.018	0.051	**0.111**	0.053	0.053	0.046
/ z /	0.017	0.104	0.028	0.014	0.043	0.006
/ z /	0.081	0.077	0.135	0.186	0.176	0.249
/ e /	0.072	0.048	0.163	**0.218**	0.144	0.210
/ le /	0.128	0.100	0.210	0.283	0.214	**0.284**
/ e /	0.062	0.085	0.115	**0.128**	0.088	0.126
/ s /	0.016	**0.073**	0.028	0.024	0.046	0.010
/ c /	0.133	0.101	0.113	**0.219**	0.171	0.217
/ c /	0.199	0.115	0.256	**0.430**	0.284	0.398
/ l /	0.069	0.129	0.126	0.184	0.166	0.176
/ ɛ /	0.055	**0.161**	0.057	0.091	0.151	0.151
/ ɛ /	0.031	**0.156**	0.051	0.074	0.150	0.098
/ i /	0.029	0.074	0.074	0.075	0.075	0.071
/ s /	0.019	**0.103**	0.030	0.018	0.049	0.003
/ se /	0.029	0.051	0.057	**0.072**	0.067	0.046
/ e /	0.054	0.055	0.124	**0.195**	0.103	0.150
/ e /	0.021	0.038	**0.100**	0.056	0.038	0.033
/ d /	0.017	0.087	0.035	0.025	0.033	0.010

/ di /	0.021	**0.135**	0.074	0.047	0.089	0.038
/ i /	0.023	**0.102**	0.065	0.035	0.076	0.027
/ s /	0.015	**0.067**	0.027	0.017	0.029	0.008
/ s /	0.007	**0.055**	0.029	0.016	0.041	0.007
/ p /	0.029	0.140	0.102	0.069	0.112	0.052
/ py /	0.011	0.072	0.056	0.034	0.039	0.034
/ yt /	0.022	0.091	0.029	0.014	0.064	0.007
/ t /	0.080	0.111	0.190	0.249	0.176	0.257
/ t ɛ /	0.050	0.134	0.076	0.114	**0.135**	0.108
/ ɛ /	0.099	0.140	0.044	0.114	0.097	0.055
~	0.016	0.040	0.047	0.035	0.037	0.030
~	0.014	0.046	0.047	0.025	0.053	0.007
~	0.023	0.037	0.073	0.044	0.066	0.003
~	0.019	0.051	0.065	0.032	0.056	0.004

8.5.2. Interprétation des résultats.

Nous pouvons voir sur le tableau 8.9 que le son / i / est toujours reconnu sans équivoque. Le son / s / est toujours, quant à lui, reconnu comme un / i /, comme nous l'avions mentionné précédemment. Le son / a / n'a pas obtenu de forte similarité probablement parce que nos patrons étaient élaborés à partir du / a / de "pâte" et non du / a / de "la" ou de "patte". Le / ə / a été bien reconnu. Nous voyons donc, par cette expérience, que notre système peut très bien fonctionner avec de la parole continu, mais qu'il pourrait à la rigueur discriminer un son dont il n'a appris aucun patron en se basant sur une séquence de reconnaissance des sons qu'il connaît. Je m'explique. Lors de la prononciation du / s / de la figure 8.8, on remarque que le / i / est reconnu, suivi du /ə /. On peut donc en déduire que le son qui est présenté à Dystal est "se" par la reconnaissance successive de / i / et / ə /. Ceci laisse entrevoir l'utilité d'une

seconde couche, qui pourrait reconnaître une séquence de sons et non des sons purs.

La figure 8.5, quant à elle, nous donne des informations intéressantes sur le choix des patrons. Nous voyons, en observant les valeurs de similarités en fonction du temps pour les sons / ə / et / ɛ /, que l'allure générale est pratiquement identique pour ces deux sons. Comme nous en avons déjà parlé précédemment, il est difficile de différencier ces deux sons avec un outil tel que l'analyse par démodulation. Il faudrait donc créer des patrons plus représentatifs pour ces deux sons pour exploiter au maximum les facteurs qui peuvent les différencier.

8.6. Généralisation avec plusieurs locuteurs.

Nous avons répété en partie les expériences avec des femmes, pour examiner les valeurs de similarités obtenues. Ces expériences se sont faites dans les mêmes conditions que la première expérience.

8.6.1. Résultats obtenus.

Les femmes n'ayant fondamentalement pas la même fréquence glottale que les hommes, nous croyons qu'une différenciation préalable, (une partie reconnaissance pour femmes et une partie reconnaissance pour hommes) serait à considérer dans un système plus évolué. Nous donnons ici quelques exemples des moyennes intermédiaires pour une femme (tableau 8.10).

tableau 8.10. Moyennes intermédiaires pour quelques lettres de l'alphabet prononcées par une femme.

Lettre	sim. / a /	sim. / i/	sim. / y /	sim. /ə/	sim. / ɛ /	sim./o/
a						
	0.265	*0.066*	*0.006*	*0.096*	*0.019*	*0.038*
	0.170	*0.076*	*0.004*	*0.071*	*0.012*	*0.060*
	0.145	*0.071*	*0.020*	*0.056*	*0.039*	*0.075*
	0.124	*0.046*	*0.022*	*0.081*	*0.036*	*0.055*
c						
	0.019	***0.121***	*0.027*	*0.017*	*0.030*	*0.004*
	0.028	***0.148***	*0.026*	*0.029*	*0.035*	*0.004*
	0.018	***0.137***	*0.036*	*0.035*	*0.088*	*0.004*
	0.100	*0.106*	*0.094*	***0.243***	*0.219*	*0.239*
	0.087	*0.090*	*0.100*	***0.221***	*0.179*	*0.214*
	0.076	*0.063*	*0.038*	***0.100***	*0.093*	*0.090*
e						
	0.246	*0.081*	*0.084*	***0.401***	*0.266*	*0.318*
	0.223	*0.080*	*0.075*	***0.395***	*0.255*	*0.315*
	0.045	*0.093*	*0.061*	*0.101*	*0.096*	***0.111***
f						
	0.031	***0.070***	*0.028*	*0.044*	*0.058*	*0.015*
	0.184	*0.121*	*0.101*	***0.367***	*0.271*	*0.349*
	0.183	*0.105*	*0.121*	***0.296***	*0.243*	*0.259*
	0.022	***0.159***	*0.081*	*0.051*	*0.093*	*0.005*
	0.025	***0.147***	*0.078*	*0.043*	*0.081*	*0.004*
	0.031	***0.163***	*0.044*	*0.033*	*0.057*	*0.010*

	0.034	*0.056*	*0.108*	*0.070*	***0.117***	*0.042*
	0.094	*0.018*	*0.025*	*0.077*	*0.045*	*0.050*
	0.059	*0.014*	*0.021*	*0.045*	*0.028*	*0.021*
	0.036	***0.046***	*0.029*	*0.040*	*0.034*	*0.014*
h						
	0.179	*0.058*	*0.028*	*0.147*	*0.088*	*0.118*
	0.232	*0.115*	*0.092*	***0.274***	*0.219*	*0.197*
	0.020	***0.170***	*0.019*	*0.023*	*0.031*	*0.012*
	0.024	***0.227***	*0.051*	*0.038*	*0.116*	*0.003*

8.7. Recommandations.

Nous émettrons ici quelques recommandations et indiquerons les pistes à suivre pour de futures recherches avec Dystal et l'analyse par démodulation.

8.7.1. Importance du choix des patrons.

L'analyse par démodulation, comme toute autre méthode d'analyse pour la reconnaissance de la parole, peut facilement confondre deux sons. Nous ne saurons trop insister sur l'importance du choix préalable des sons à utiliser. Les échantillons de sons qui constituent les patrons utilisés pour la phase d'apprentissage devraient être triés avec soin pour ne présenter aucune confusion. Si, après visualisation, un patron se révèle flou (beaucoup de bruit) ou peu représentatif, il devrait aussitôt être éliminé du processus de sélection.

On peut voir sur la figure 2.5 du chapitre 2 que les frontières de délimitations entre les sons sont mal définies et se recoupent en plusieurs endroits. Un patron, puisque servant de référence, ne devrait être sujet à aucune confusion. Il devrait caractériser sans équivoque le son qu'il représente.

8.7.2. Génération de patrons artificiels.

Il serait intéressant de générer des patrons artificiels "parfaits" caractérisant des groupes de locuteurs typiques. En effet, en ne supervisant pas l'apprentissage et en lui présentant n'importe quel locuteur, on ne fait qu'accroître la confusion générée par les délimitations entre les sons mal définis dont nous avons parlé à la section précédente.

Notre système de reconnaissance pourrait s'insérer comme un étage supérieur dans un système qui ferait préalablement du tri sur les caractéristiques d'un locuteur donné. Un locuteur pourrait être classé dans un groupe donné selon sa fréquence glottale, puis pris en charge par un réseau Dystal n'ayant appris que des patrons ayant des caractéristiques proches de ce locuteur. La sixième expérience laisse croire qu'un tri sur le sexe (directement relié à la fréquence glottale) serait à considérer.

Nous avons choisi, lors de nos expériences, 6 sons que nous avons décidé d'analyser. La première chose à faire serait de générer des patrons ayant

des différences entre les formants parfaites, c'est-à-dire ne portant aucunement à confusion. Il ne faudra donc définitivement pas présenter n'importe quel locuteur à Dystal lors de la phase d'apprentissage.

8.7.3. Apprentissage supervisé.

Nous croyons qu'une bonne reconnaissance ne pourra être faite sans apprentissage supervisé, à moins d'établir un algorithme de tri des patrons, c'est-à-dire de rejet systématique des patrons non caractéristiques. De cette façon, on évitera à Dystal de travailler avec des patrons portant à confusion (on peut se référer aux patrons de /e/ montrés précédemment.

8.7.4. Révision de l'analyse par démodulation.

L'analyse par démodulation permet très bien de faire la distinction entre certains sons, mais reste peu ou pas appropriée pour des sons comme le "ou", par exemple. Il serait intéressant d'examiner en détail les sorties générées par l'analyse par démodulation, afin d'établir quels sont les sons qui peuvent être le plus facilement discriminés et quels sont ceux avec lesquels l'analyse par

démodulation ne donne aucune caractéristique discriminante. On pourrait à la rigueur revoir cette analyse pour trouver les points à améliorer.

8.7.5. Délimitation de "frontières" de reconnaissance.

En plus de revoir les patrons qui sont présentés lors de l'apprentissage, l'étape de la simulation devra elle aussi être révisée. Il serait intéressant d'ajouter à la phase de simulation un algorithme déterminant statistiquement un facteur de précision sur la reconnaissance. On pourrait ainsi déterminer le degré de certitude de reconnaissance sur chaque son en même temps que les similarités sont calculées.

9

Conclusion

À ce point, permettons nous de rappeler les objectifs que nous nous étions assignés.

Les objectifs principaux étaient d'une part de vérifier la pertinence de l'analyse par démodulation et d'autre part d'élaborer une architecture de reconnaissance de parole capable de traiter les paramètres d'analyse.

Quoique très intéressante pour la reconnaissance de parole continue, l'analyse par démodulation devra cependant être révisée, car il y a définitivement des sons qui sont difficiles à différencier avec cette technologie. Nous avons identifié quelques-uns de ces sons dans les chapitres précédents. Les sons qui sont confondus possèdent en général des différences de fréquences entre les formants similaires, par exemple, si nous examinons le /e/ et le /ɛ/, nous remarquons que les différences de fréquences entre les formants sont pratiquement les mêmes.

/e/: F1-F2: 900Hz
F2-F3: 900Hz
F3-F4: 1100Hz

/ɛ/: F1-F2: 1400Hz
F2-F3: 800Hz
F3-F4: 1000Hz

Comme nous l'avons déjà mentionné, la seule différence notable entre ces deux sons se trouve au niveau de F1-F2. Comme nos filtres de basses

fréquences n'englobent pas une aussi grande différence de fréquence (chapitre 3, figure 3.1), il est donc normal de confondre ces deux sons, étant donné que F2-F3 et F3-F4 sont très similaires. Nous mettons ici le doigt sur un problème de l'analyse par démodulation.

Comme nous l'avons vu au chapitre précédent, les échantillons de sons présentés à Dystal pour l'apprentissage doivent être choisis avec soin, pour éviter toute confusion entre les sons ayant des caractéristiques similaires. Nous devons, lors de l'apprentissage, chercher à accentuer les caractéristiques particulières de chaque son, afin de bien faire ressortir les différences. Le mieux serait la génération de patrons artificiels "parfaits".

L'analyse par démodulation devra être améliorée pour la reconnaissance de sons comme /u/, /e/ ou /ɛ/. Les sorties fournies par l'analyse par démodulation pour ces sons porte définitivement à confusion.

Bien que nous ne pensons pas que l'architecture à réseaux de neurones que nous avons développé aura besoin de modifications majeures pour reconnaître plus de sons et s'intégrer à un système de reconnaissance de parole plus complet, nous croyons cependant que l'apprentissage supervisé s'imposera de lui-même lorsque l'on ajoutera des sons à ceux déjà appris par notre système (voir les recommandations ci-dessus).

Nous avons établi dans ce travail les bases d'un système de reconnaissance de parole (pré-traitement + réseau de neurones). Comme les sorties de l'analyse par démodulation sont données pour des segments de

signaux de parole, un segment pourrait être présenté à l'entrée du réseau de neurones aussitôt après sa génération, pour être traité sur le champs. Cette manipulation pourrait constituer la base d'un système en temps réel. Le lecteur comprendra que ce que nous avons développé actuellement ne peut fonctionner en temps réel, car nous n'avons pas tenu compte dans ce travail de l'aspect performance et optimisation. Nous mentionnons donc le temps réel seulement parce que nous considérons que cela est un avantage potentiel de notre système qui ne doit pas être passé sous silence.

L'approche que nous avons choisie d'étudier ici est intéressante parce qu'elle pourrait constituer la base d'un système de reconnaissance de parole fonctionnant en temps réel et évidemment parce qu'elle devrait être relativement robuste au bruit.

Nous croyons avoir ouvert un domaine de recherche encore très vaste et peu ou pas étudié jusqu'à présent. Nous espérons que ce travail saura inspirer d'autres chercheurs et leur sera de quelque utilité.

Annexe

Aspect technique et chaîne de traitement

Le moment est maintenant venu de se plonger plus en détail dans l'aspect technique de notre chaîne de traitement. Nous verrons ici un exemple typique de traitement normal et de traitement de masse que nous avons eu à faire lors de nos expériences. Nous montrerons aussi la structure organisationnelle du compte dans lequel nous avons fait nos expériences et organisé nos résultats.

Le lecteur devrait être en mesure, après la lecture de ce chapitre, de reconstituer nos expériences et d'élaborer les siennes.

A.1 Précisions techniques sur la chaîne de traitement.

La première étape et le point de départ de notre système est, bien entendu, un fichier de parole. Ce fichier contient une série de nombres ASCII, car, bien sûr, tous les fichiers de parole sont échantillonnés en fonction du temps. Les fichiers de parole portent l'extension .txt, et sont échantillonnés à 32 KHz. Les fichiers de parole sont obtenus à l'aide d'enregistrements traités par un convertisseur analogue - numérique.

Avant d'être traité par DYSTAL, un fichier de parole doit être pré-traité. Nous avons automatisé le pré-traitement décrit au chapitre 3 en une seule fonction qui produit un fichier .dys (pouvant être présenté à DYSTAL), qui est en fait le résultat de l'analyse par démodulation (images 3D). Par exemple, la

commande *ptrait a.txt* produira le fichier a.dys en utilisant la méthode décrite au chapitre 3.

Le fichier .dys est présenté à la première couche de DYSTAL, qui produit un fichier .sim, contenant les valeurs de similarités trouvées par chacuns des neurones DYSTAL. Les fichiers .sim sont aussi des images 3D, à la différence que les canaux ne sont plus des filtres numériques mais des valeurs de similarité en fonction des différents neurones. Le traitement de a.dys par Dystal, par exemple, produira le fichier a.dys.sim. Nous avons décrit au paragraphe 6.4, sections 3, 4 et 5 la façon dont les fichiers .sim sont obtenus.

Le fichier .sim peut être traité par la deuxième couche, pour produire un fichier portant l'extension .2nd.

Pour faciliter l'interprétation des résultats, nous avons développé deux utilitaires de traitement, "moyenne" et "aire".

Le premier de ces utilitaires, "moyenne", calcule la moyenne de similarité dans chaque neurone de sortie de la première et de la deuxième couche de Dystal (6 dans notre cas).

Le deuxième, "aire", calcule l'aire sous la courbe pour chaque canal (neurone) de sortie de dystal. La figure A.1 présente un shéma logique de l'opération de traitement des signaux de parole effectuée par notre système. L'extension des fichiers obtenus à chaque étape est aussi indiquée.

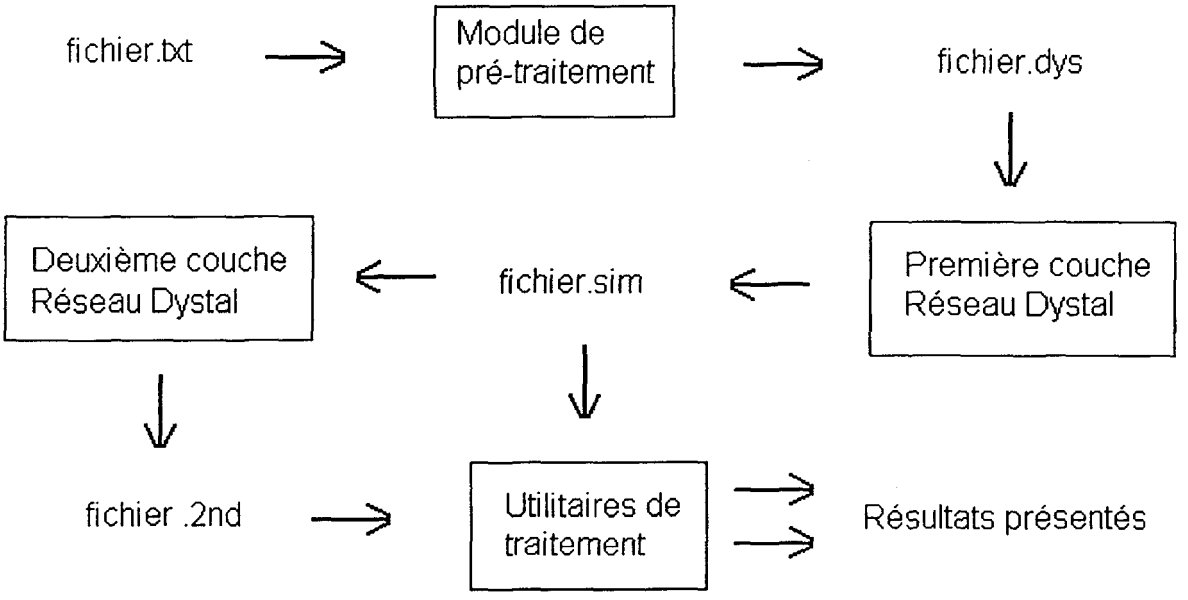


figure A.1. Séquences de traitement, pour la première couche.

A.2. La deuxième couche.

La chaîne de traitement peut se poursuivre, si on le désire, par l'envoi du fichier de sortie de la première couche à l'entrée de la deuxième couche. Le fichier d'entrée accepté par la deuxième couche est le .sim de sortie de la première couche.

La deuxième couche donnera comme sortie un fichier .2nd semblable au fichier .sim, mais représentant les similarités obtenues par le calcul avec les patrons de la deuxième couche.

A.2.1. Contrainte technique: la pile de données de Windows 3.1.

Pour réaliser les expériences qui vous sont présentées, nous avons travaillé à la fois sur un ordinateur IBM PC muni de Microsoft Windows 3.1 et sur une station de travail SUN Sparc 2+. Ces machines réservent, pour chaque application, une certaine quantité de mémoire vive qu'elles peuvent utiliser à leur guise. Cette quantité de mémoire est égale à 5k par application pour Windows 3.1. Nous appelons cette quantité la pile de données.

Comme la deuxième couche de notre réseau Dystal travaille avec des patrons dont les données (stimuli cs) sont stockées sous forme de nombres avec virgule flottante qui prennent chacuns 8 octets de mémoire vive, la pile de données est vite remplie.

La solution que nous avons adoptée est tout simplement d'exécuter la simulation de la deuxième couche avec Dystal n'ayant mémorisé qu'un seul son, c'est-à-dire 3 ou 4 patrons seulement. Il est évident que cette solution n'est que temporaire, il faudra tôt ou tard augmenter la taille de la pile si on prévoit une utilisation plus intensive du logiciel. Il est à noter que ce problème se présentera aussi pour la première couche. Les patrons de la première couche étant

constitués d'entiers (4 octets de mémoire), une augmentation du nombre de patron entraînera la saturation de la pile.

A.3. Détails techniques sur la chaîne de calcul.

A.3.1. Le signal de parole initial.

La première chose à faire est d'obtenir un fichier de parole numérisé à 32kHz. Ce fichier peut être obtenu à l'aide d'un convertisseur analogue - digital. Voici les premières lignes du fichier a.txt, un fichier de parole.

```
a
32000
-169
-254
-178
.....
```

La première ligne d'un fichier de parole est le nom de fichier, donnée peu importante et n'étant pas utilisée par nos programmes de traitements. La deuxième ligne est la fréquence d'échantillonnage (32kHz). À partir de la troisième ligne, nous retrouvons le signal de parole, dont chaque échantillon est un entier compris entre -32000 et +32000.

A.3.2. Le pré-traitement.

La prochaine étape est le pré-traitement du signal de parole. Le pré-traitement est automatisé et se fait en exécutant le programme ptrait.

ptrait a.txt

Cette commande générera en fait deux fichiers, a.dys, a.acc. Voyons un peu le contenu et l'utilité de ces deux fichiers.

Le fichier a.acc est un fichier pré-accentué. Le fichier a.dys est l'analyse par démodulation du fichier a.acc.

La pré-accentuation tend à amplifier les composantes haute fréquence des signaux de parole qui sont la plupart du temps très faibles par rapport aux impulsions basse fréquence. Tous nos signaux sont systématiquement pré-accentués avant d'être traités. Nous utilisons la formule suivante pour pré-accentuer nos fichiers numériques.

$y(n) = x(n) - A\,x(n-1)$, avec $y(0) = 0.$ et $n >= 1.$

A = 0.94, le facteur standard de pré-accentuation pour un signal de parole.

Nous devons aussi glisser un mot sur le fichier a.nom, un fichier temporaire normalisé entre -32000 et +32000 qui sera créé, puis détruit lors du

pré-traitement. Le fichier a.nom est produit avant le fichier a.acc, à partir du fichier .txt. On veut seulement s'assurer ici que toutes nos données sont bien normalisées entre -32000 et +32000. Si le fichier .txt ne l'était pas, il le sera avant d'être pré-accentué.

Le fichier a.acc est pré-traité selon la méthode expliquée au chapitre 3, poúr donner le fichier .dys, pouvant être présenté à Dystal.

A.3.3. Dystal et les fichiers .dys.

Examinons brièvement un fichier portant l'extension .dys. C'est un fichier échantillonné à 16kHz, constitué de lignes comprenant chacune 24 entiers entre -32000 et +32000. Chacun de ces 24 entiers correspond à la sortie d'un filtre du banc de filtres utilisé lors de l'analyse par démodulation.

La première ligne du fichier .dys est la fréquence d'échantillonnage (16kHz).

Nous devons maintenant nous servir du programme Dystal pour générer les fichiers de similarité (.sim).

A.3.3.1. Pré-requis.

Pour faire fonctionner Dystal, nous avons besoin des éléments suivants:

1. L'exécutable "dystal", version compilée du fichier source "dystal.c".

2. Le fichier d'entête "dystal.h", qui contient de l'information essentielle à la configuration du réseau de neurones.

3. Un fichier de parole pré-traité (actuellement une grille de 24 x ??? entiers).

4. Une série de patrons (fichiers .pat).

5. Le fichier de configuration du réseau de neurones "dystal.cnf". N.B. Ce fichier n'est qu'optionnel. S'il n'est pas présent dans le répertoire de travail de Dystal, il sera automatiquement recréé lors de la première utilisation de Dystal.

A.3.3.2. Exemple d'utilisation

Voici un exemple d'utilisation de notre programme, qui, nous l'espérons, aidera le lecteur à se servir de dystal et aussi de la couche supérieure, qui fonctionne exacement de la même façon.

1. Détruire le fichier de configuration "dystal.cnf". Ceci effacera de la mémoire tous les patrons stockés auparavant par Dystal.

2. Exécuter le programme principal "dystal".

3. Un menu apparaîtra à l'écran. Trois options sont disponibles: Apprentissage, Simulation et Quitter. Comme le fichier "dystal.cnf" vient d'être détruit, on ne peut effectuer la simulation maintenant, car Dystal ne possède aucun patron en mémoire. Choisissons donc 1. Apprentissage.

4. "Nom de la grille cs: " apparaît à l'écran. Entrer a1.pat ou tout autre nom portant l'extension .pat se trouvant dans le répertoire de travail de dystal.

5. "Valeur de l'entree ucs: " est maintenant affiché. Entrer un caractère, correspondant au son relié au patron. La réponse logique est "a" si l'on a choisi "a1.pat" comme patron. En effet, on désire associer le stimulus conditionné contenu dans la grille a1.pat au son /a/, qui représente, on s'en souviendra, le stimulus non-conditionné.

6. Dystal affiche ensuite les valeurs de similarité trouvées avec les patrons disponibles dans sa mémoire. Comme ceci est notre premier patron, aucune valeur de similarité est affichée. Il est à noter qu'un nouveau patron est créé seulement si dystal ne trouve pas en mémoire de patron ayant une similarité supérieure à 0.8 avec le patron couramment traité.

7. De la même façon, on peut faire "apprendre" à Dystal tous les patrons se trouvant dans son répertoire de travail.

8. Revenir au menu principal, et cette fois choisir 2. Simulation.

9. "Nom de la grille d'entrée: " apparaît alors à l'écran. Entrer a.txt.dys ou tout autre nom de fichier portant l'extension .dys. Attention, l'exécution du programme prend un certain temps, soyez patient!!!

10. Un fichier contenant les similarités trouvées en comparant les groupes de patrons correspondants à chaque son appris sera constitué. Le fichier portera l'extension .sim.

11. Lorsque Dystal aura analysé tout le fichier .dys, le menu principal réapparaîtra à l'écran. Choisir 3. Quitter.

A.3.4. Les fichiers de similarités (.sim).

Les fichiers de similarités sont les images statiques qui ont été décrites au chapitre précédent. Un fichier .sim est une suite de groupes de 6 nombres en virgule flottante, dont la valeur représente une similarité et est comprise entre -1.0 et 1.0. Les 6 nombres d'une ligne d'un fichier .sim représentent la similarité maximale, à un instant donné, avec les sons suivants, dans l'ordre:

/a/, /i/, /y/, /ø/, /ɛ/, /o/.

A.3.5. La deuxième couche et les fichiers .2nd.

Les fichiers .sim peuvent être reconnus par la deuxième couche "csup", de la manière expliquée en A.3. Il n'y a pas d'images statiques dans les fichiers de sortie .2nd, seulement des similarités calculées. Les fichiers .2nd ont exactement le même format que les fichiers .sim.

A.3.6. Autres précisions techniques.

Nous croyons bon ici de parler de la taille de la grille que nous utilisons, car nous supposons qu'un autre utilisateur devra la changer pour l'adapter à ses besoins. La grille utilisée pour l'analyse avec Dystal est une grille de 24 colonnes par 126 lignes maximum. Les deux seuls paramètres à changer pour utiliser une grille de taille différente sont les variables RANGEES et COL, définies au début du fichier "dystal.h". COL définit le nombre de colonnes que contient la grille (24) et RANGEES le nombre de rangees maximal pour le calcul de similarité (126). RANGEES x COL correspond donc à la taille maximale qu'un patron peut avoir en mémoire. Les autres variables définies dans le fichier "dystal.h" sont des

paramètres de configuration pour les images statiques générées en sortie (voir chapitre 6). Nous conseillons de ne pas modifier ces valeurs, à moins de modifier le programme dystal.c.

A.4. Traitement de masse.

Comme nous avons souvent eu à faire du traitement de masse, nous avons développé des utilitaires de traitement qui exécutent les procédures requises pour traiter une série de fichiers. Nous croyons bon d'en glisser un mot ici, car ces utilitaires sont en fait des scripts Unix pouvant facilement être modifiés par un utilisateur pour de nouvelles expériences.

Le script global à lancer pour le traitement de masse est simule.scr (simule2.scr pour la deuxième couche). On peut le lancer (sur Unix) avec la commande suivante:

```
simule.scr > simule.out &
```

simule2.scr est le script maître pour la simulation de masse. Il peut appeler plusieurs autres scripts. À titre d'exemple, voyons un cas typique d'une simulation sur la deuxième couche. On désire traiter ici 26 sons correspondant à la prononciation des lettres de l'alphabet. Voici le contenu du script simule2.scr.

simule2.scr

```
rm csup.cnf
        /* On efface ici le fichier de configuration pour en créer un nouveau */
csup < learn_a.ent > learn_a.sor
        /* Appel du script d'apprentissage pour le son /a/ */
compute.scr
        /* Création des fichiers .2nd pour le son /a/ */
results2.scr > results2_a.out
        /* Interprétation des résultats avec les utilitaires de traitement */
rm csup.cnf
        /* On recommence avec le son /i/ */
rm *.sor
csup < learn_i.ent > learn_i.sor
compute.scr
results2.scr > results2_i.out
rm csup.cnf
rm *.sor
csup < learn_f.ent > learn_f.sor
compute.scr
results2.scr > results2_f.out
rm csup.cnf
rm *.sor
csup < learn_u.ent > learn_u.sor
compute.scr
results2.scr > results2_u.out
rm csup.cnf
rm *.sor
csup < learn_e.ent > learn_e.sor
compute.scr
results2.scr > results2_e.out
rm csup.cnf
rm *.sor
csup < learn_o.ent > learn_o.sor
compute.scr
results2.scr > results2_o.sor
rm csup.cnf
rm *.sor
```

Les fichiers learn_*.ent ne contiennent que les commandes nécessaires au logiciel pour l'apprentissage des sons appropriés. Par exemple, learn_a.ent contiendra les commandes nécessaires à l'apprentissage des patrons du son a.

compute.scr est un fichier appelant csup avec les paramètres nécessaires à la simulation d'un ou plusieurs sons.

Les fichiers les plus intéressants à consulter sont en fait les fichiers learn_*.out, qui contiennent les données interprétées par les utilitaires moyenne et aire, dont nous avons parlé plus haut.

Les fichiers portant l'extension .sor sont en fait des fichiers tampons, peu importants et pouvant être détruits sitôt une simulation terminée.

A.5. Structure organisationnelle du compte e271.

Les expériences qui ont été décrites au chapitre 7 ont été réalisées avec les scripts, fichiers et programmes contenus dans le compte e271. Nous présentons ici l'organisation des répertoires principaux de ce compte.

e271 possède des répertoires contenus dans deux partitions de la machine dsa_vlsi (/nfs/dsa/sug3 et /sug2).

On y trouvera les répertoires suivants:

/Traitement: Répertoire contenant les fichiers .txt, .acc et .dys obtenus par l'utilitaire ptrait. Ces fichiers sont contenus dans une série de sous répertoires identifiés par le son de correspondance.

/dystal: Répertoire contenant dystal.c, dystal et dystal.h. Répertoire de travail de la première couche de dystal dans lequel on doit copier les fichiers .dys à traiter. Ce répertoire contient tous les patrons utilisés jusqu'à présent par la première couche (fichiers .pat).

/dystal/2ndlayer: Répertoire contenant csup.c, csup et csup.h, les fichiers constituant la deuxième couche du réseau Dystal. On doit y copier tous les fichiers .sim qu'on veut traiter. Ce répertoire contient aussi tous les patrons utilisés jusqu'à présent par la deuxième couche (*2.pat).

Références

[1] D.L. Alkon, K.T. Blackwell, G.S. Barbour, A.K. Rigler and T.P. Vogl. "Pattern-Recognition by an Artificial Network Derived from Biologic Neuronal Systems". Biol. Cybern, 62, pp. 363-376, 1990.

[2] J.M. Aran, A.Dancer, J.M. Dolmazon, R. Pujol et P. Tran Ba Huy, "Physiologie de la cochlée", Inserm/SFA, 1988.

[3] L. R. Bahl & F. Jelinek, "Decoding for Channels with Insertions, Deletions and Substitutions with Applications to Speech Recognition". IEEE Trans. on Information & Theory, vol. IT-21, pp. 404-411, 1975.

[4] R. Bellman, "Dynamic Programming". Princetown University Press, 1957.

[5] K.T. Blackwell, T.P. Vogl, S.D. Hyman, G.S. Barbour and D.L. Alkon, "A New Approach to Hand-Written Character Recognition". Pattern Recognition, vol. 25, no. 6, pp. 655-666, 1992.

[6] René Boite & Mural Kunt, "Traitement de la parole". Presses polytechniques romandes, 1987.

[7] Bregman, "Auditory scene analysis". IEEE International Conference On Pattern Recognition", pp. 168-175, 1984.

[8] John S. Bridle, "Neural Networks or Hidden Markov Models for Automatic Speech Recognition: Is there a Choice?". NATO ASI Series, Volume F75, pp. 225-233, Springer-Verlag Berlin Heidelberg, 1992.

[9] A.B. Carlson, "Communications systems: An introduction to Signals and Noise in Electrical Communication". Mc Graw Hill, 1986.

[10] F. Fallside, "Neural Networks for continuous speech recognition". Speech recognition and understanding, recent advances, trends and applications, NATO ASI series F, computer and systems sciences, vol. 75, 1992, pp. 237-256.

[11] G. Fant, "Acoustic Theory of Speech Production". Mouton: The Hague, 1960.

[12] R.B. Gardner & J.P. Wilson, "Evidence for direction-specific channels in the processing of frequency modulation". JASA 66, 704-709, 1979.

[13] B. Gold, "Hopfield model applied to vowel and consonant discrimination". M.I.T. Lincoln Lab., Lexington, MA, Tech. Rep. 747, June 1986.

[14] F. Guyot, F. Alexandre, C. Dingeon and J.P. Haton, "The cortical column as a model for speech recognition: principles and first experiments". Speech recognition and understanding, recent advances, trends and applications, NATO ASI series F, computer and systems sciences, vol. 75, 1992, pp. 275-291.

[15] W. Huang and R. Lippman, "A neural net approach for speech recognition", Proceedings of the IEEE-ICASSP, New-York, 1988, pp. 412-453.

[16] F. Jelinek, "Continuous Speech Recognition by Statistical Methods". Proceedings of the IEEE, vol. 64, pp. 532-536, April 1976.

[17] J.F. Kaiser, "On a simple algorithm and its generalization to continuous signals". Proceedings of IEEE-ICCASSP'90, Albuquerque, 381-384.

[18] James F. Kaiser, "Some Useful Properties of Teager's Energy Operators". Department of Electrical and Computer Engineering, Rutgers University, Piscataway, New Jersey.

[19] James F. Kaiser, "On a Simple Algorithm to Calculate the Energy of a Signal". Proceedings of IEEE-ICASSP'90, Albuquerque, 381-384.

[20] James F. Kaiser. "On Teager's energy algorithm and its generalization to continuous signals." IEEE DSP workshop, Now Paltz, NY, september 1991, pp. III-149 - III-152.

[21] Ben J.A. Kröse & P. Patrick van der Smagt, "An Introduction to Neural Networks". University of Amsterdam, Faculty of Mathematics and Computer Science, 1993.

[22] Petros Maragos, Thomas F. Quatieri & James F. Kaiser, "Speech Nonlinearities, Modulation and Energy". IEEE-ICASSP 1991, pp. 421-424.

[23] M. Minsky and S. Papert, "Perceptrons", MIT Press, Cambridge, Massachusetts, 1969.

[24] B.C.J. Moore & B.R. Glasberg, "Suggested formulae for calculating auditory-filter bandwiths and excitation patterns". Jour. Acoust. Soc. Amer., 74, 750-753, 1983.

[25] C. S. Myers & L. R. Rabiner, "A level Building Dynamic Time Warping Algorithm for Connected Word Recognition". IEEE-ASSP, vol. ASSP-29, #2, pp. 284-297.

[26] NeuralWorks Professionnal II, volume 1, "Neural Computing", NeuralWare Inc, 1989.

[27] NeuralWorks Professional II/Plus and NeuralWorks Explorer, "Neural Computing". NeuralWare Inc, 1991.

[28] H. Ney. "Stochastic grammars and pattern recognition". Speech recognition and understanding, recent advances, trends and applications, NATO ASI series F, computer and systems sciences, vol. 75, 1992, pp. 345-381.

[29] Douglas O'Shaughnessy, "Speech Communication, Human and Machine". Addison-Wesley Publishing Company, 1990.

[30] R.D. Patterson, "Auditory filter shapes derived with noise stimuli". Jour. Acoust. Soc. Amer., 59, 3, 640-654, 1976.

[31] R.W. Prager, T.D. Harrison and F. Fallside, "Boltzmann machines for speech recognition". Computer speech language, vol. 1, pp. 3-27, 1986.

[32] Lawrence R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". Proceedings of the IEEE, vol. 77, no. 2, February 1989, pp. 257-283.

[33] L. R. Rabiner & C. E. Schmidt, "Application of Dynamic Time Warping to Connected Digit Recognition", IEEE-ASSP, vol. 28, #4, pp. 377-388.

[34] L. R. Rabiner, "The Theory of Hidden Markov Models and its Application to Speech Recognition". NATO ASI Recent Advances in Speech Understanding and Dialog Systems, Bad Windsheim, Germany, 1987, pp. 237-274.

[35] J. Rouat, "A Nonlinear Speech Analysis based on Modulation Information", Speech Recognition and Coding, New Advances and Trends, edited by Antonio J. Rubio Ayuso, Juan M. López Soler, Springer, NATO-ASI series, vol. 147, 1995, pp. 341-344.

[36] J. Rouat, "Nonlinear operators for speech analysis", dans "Visual Representations of Speech Analysis" de M. Cook & S. Beet.. J Wiley, 1992.

[37] J. Rouat, "A Spectro-Temporal Analysis of Speech Based on Nonlinear Operators". Proceedings of the International Conference on Spoken Language Processing, Banff, october 12 to 16, Vol. 2, pp. 1629-1632.

[38] J. Rouat, "Thèse de Ph. D.". Université de Sherbrooke, Québec, Canada, 1988.

[39] H. Sakoe et al. "Speaker-independant word recognition using dynamic programming neural networks". Proceedings of the IEEE-ICASSP, Glasgow, 1989, pp. 24-32.

[40] Christel Sorin, "Reconnaissance et synthèse automatique de la parole". Centre national d'étude des télécommunications, 1991.

[41] B.W. Tansley & J.B. Suffield, "Time course of adaptation and recovery of channels selectively sensitive to frequency and amplitude modulation". JASA 74, 765-775, 1983.

[42] K.P. Unnikrishman, John J. Hopfield and David W. Tank. "Connected-digit speaker-dependant speech recognition using a neural network with time-delayed connections". IEEE transactions on signal processing, vol. 39, no. 3, march 1991, pp. 108-119.

[43] G. H. Wakefield & N. F. Viemeister. "Selective Adaptation to Linear Frequency-Modulated Sweeps: Evidence for Direction-Specific FM Channels?", JASA 75, pp. 1588-1592, 1984.

[44] B. Widrow. "30 years of adaptive neural networks: perceptron, madaline, and backpropagation". Proceedings of the IEEE, vol 78, no. 9, September 1990, pp. 1415-1439.