

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR -ANNABA
UNIVERSITY
UNIVERSITE BADJI MOKHTAR
ANNABA



جامعة باجي مختار
- عنابة -

Faculté des Sciences de l'Ingénieur

Année : 2005

Département d'Informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme de **MAGISTER**

UN SYSTEME HYBRIDE AG/PMC POUR LA RECONNAISSANCE DE LA PAROLE ARABE

Option :

Intelligence Artificielle Distribuée (IAD)

Par

Lotfi AMIAR

DIRECTEUR DE MEMOIRE : Mokhtar SELLAMI Professeur U. ANNABA

Devant le jury

PRESIDENT : M.T. KHADIR Maître de Conférence U. ANNABA

EXAMINATEURS : M. K. KHOLLADI Maître de Conférence U. CONSTANTINE

B. TIGHIOUART Maître de Conférence U. ANNABA

Z. ZEMIRLI Maître de Conférence I.N.I. ALGER

ملخص

من بين الابتكارات الحديثة في الذكاء الاصطناعي (IA)، تكامل عديد من الطرق في أنظمة تسمى أنظمة هجينة ذكية. عدد من البحوث في هذا السياق تتوجه نحو تكامل شبكات الأعصاب (RN) و الخوارزميات الوراثية (AG).

مشكلة (أو صعوبة) إعادة التعرف الأوتوماتيكي للكلام (RAP) هو ميدان بحث و دراسة نشيط منذ سنوات الخمسينات (50). من بين الأنماط الأكثر استعمالاً في هذا الميدان، شبكات الأعصاب، خصوصاً الشبكات المتعددة الطبقات (PMC). لقد صدر مؤخراً تمديد لهذه الأنماط، ما نتج عنها ولادة أنماط هجينة.

مساهمتنا تدخل في إطار إعادة التعرف على الأرقام العربية المنطوقة بصفة مستقلة عن المتكلم.

إثر استخلاص الوسائط السمعية من الإشارة الصوتية، مستعملين بذلك تحليل RASTA-PLP، نقترح بعد ذلك نظام هجين للتعليم و إعادة التعرف الذي يسمح باتصال القدرات المتميزة، المتمثلة في المقاومة ضد الضجيج للشبكات المتعددة الطبقات و قدرة الخوارزميات الوراثية من أجل تقريب مجموعة وسائط باستعمال بحث كلي، بهدف إيجاد نظام إعادة التعرف الصوتي التقريبي بأحسن الأداءات.

ABSTRACT

One innovation in artificial research in the integration of multiple artificial intelligent paradigms into hybrid intelligent systems. Some researches in this area deal with the integration of neural network and genetic algorithms.

The problem of the Automatic Speech Recognition (ASR) is an active field of studies since the fifties. Among the most used models in this field we can find the Neural Networks (NNs), particularly, Multi-Layer Perceptron (MLP). Recently, an extension of these models was developed giving rise to the hybrid models.

The contribution presented in this thesis, enters within the framework of the Arabic isolated digits recognition independently from the speaker.

After extraction of the speech signal acoustic parameters by using RASTA-PLP analysis (*RelAtive SpecTrAl processing-Perceptual Linear Predictive*), an hybrid system is proposed for the training and recognition which makes it possible to join the discriminating capacities, resistance to the noise of Multi-Layer Perceptron (MLP) and the ability of the Genetic Algorithm (GA) to optimize a set of parameters by a global search in order to obtain optimal speech recognition system of better performances.

Keywords

Speech recognition, GA, MLP, RASTA-PLP, training, classifier selection.

RESUME

Une des innovations récentes en intelligence artificielle (IA) est l'intégration de ses multiples paradigmes dans des systèmes dits systèmes hybrides intelligents. Nombre de recherches dans ce contexte s'orientent vers l'intégration des réseaux de neurones et des algorithmes génétiques.

Le problème de la reconnaissance automatique de la parole (RAP) est un domaine d'étude active depuis les années 50. Parmi les modèles les plus utilisés dans ce domaine, les réseaux de neurones (RNs), plus particulièrement, le perceptron multi-couches (PMC). Dernièrement, une extension de ces modèles a été mise au point donnant naissance aux modèles hybrides.

Notre contribution rentre dans le cadre de la reconnaissance des chiffres Arabe isolés indépendamment du locuteur.

Après extraction des paramètres acoustiques du signal vocal, en utilisant l'analyse RASTA-PLP (*RelAtive SpecTrAl processing-Perceptual Linear Predictive*), nous proposons après, un système hybride pour l'entraînement et la reconnaissance qui permet de rejoindre les capacités discriminantes, la résistance au bruit des Perceptrons Multi-Couches (PMCs) et la capacité des Algorithmes Génétiques (AGs) pour l'optimisation d'un ensemble de paramètres par une recherche globale, en vue de trouver un système de reconnaissance vocale optimal de meilleurs performances.

Mots Clef

Reconnaissance de la parole, AG, PMC, RASTA-PLP, Apprentissage, sélection de classifieurs.

DEDICACE

Je dédie ce travail aux êtres qui me sont les plus chers au monde : mes parents.

A mes frères et ma petite sœur Fadoua.

A ma grand-mère.

A la mémoire de mon grand-père.

A mes oncles et tantes, en particulier, ma tante Rachida.

A toutes ma famille.

A tous mes amis.

A tous les gens qui me sont chère.

REMERCIEMENTS

Au terme de ce travail, je voudrai exprimer ma profonde gratitude envers Dieu tout puissant qui, grâce à son aide, j'ai pu finir ce travail.

Mes remerciements les plus vifs sont adressés à Monsieur Mokhtar Sellami, Professeur à l'université Badji Mokhtar, Annaba, pour avoir accepté de diriger mon travail, pour le sujet qu'il m'a proposé, pour le confiance qu'il m'a témoigné, pour ses précieux critiques, conseils, encouragements et son soutien constant qu'il veuille trouver dans ce mémoire ma profonde gratitude et mon grand respect.

Je tiens très sincèrement à remercier monsieur M.T. Khadir, Maître de conférence à l'université de Annaba, pour avoir accepté de présider ce jury.

Mes remerciements les plus intenses sont adressés à monsieur M. Kolladi, Maître de conférence à l'université de Constantine, à madame B. Tighiouart, Maître de conférence à l'université de Annaba, à monsieur Z. Zemirli, Maître de conférence à I. N. I d'Alger, pour l'honneur qu'ils mon fait en acceptant, sans aucune hésitation de faire partie de ce jury et d'examiner ce travail.

Je tiens aussi à remercier tous les membres du laboratoire LRI, tout particulièrement madame L. Meslati, pour l'aide qu'elle m'a apporté, pour ses précieuses conseils, qu'elle trouve dans ce mémoire ma profonde gratitude et mon grand respect.

Ma gratitude s'adresse aussi à tous les enseignants qui ont assuré ma formation.

Enfin, je remercie toute ma famille, pour son soutien indéfectible, pour ses encouragements, et pour sa précieuse collaboration.

Liste des Tableaux

Tab	Titre	Page
1.1	Exemples d'applications de la reconnaissance de la parole	14
3.1	Analogie entre algorithme génétique et nature	58
3.2	Codage des entiers	61

Liste des Figures

Fig	Titre	N°
1.1	Coupe de l'appareil phonatoire humain (d'après [97])	6
1.2	Modèle de production de la parole	7
1.3	Schéma d'une interface homme-machine vocale	11
1.4	Système classique de reconnaissance automatique de parole	12
1.5	Représentation temporelle du signal acoustique de la parole	15
1.6	Exemple de 2 signaux temporels (à gauche) et de 2 spectrogrammes (à droite) d'une même phrase prononcée par deux locuteurs différents (signal extrait du corpus TIMIT)	20
1.7	Modèle simplifié de modèle de production de la parole	22
1.8	Calcul des cepstres par analyse spectrale	22
1.9	Répartition des filtres triangulaires sur les échelles fréquentielle et Mel	24
1.10	Algorithme de calcul des <i>MFCCs</i>	24
1.11	Analyse PLP	25
1.12	Comparaison des spectres dérivés d'une analyse PLP et LPC	26
1.13	La méthode RASTA-PLP	27
1.14	Les variations en fonction du temps (a) du coefficient cepstral $MFCC_l$ (b) de sa première dérivée delta- $MFCC_l$ (avec $n_{\Delta}=2$) et (c) de sa deuxième dérivée delta-delta- $MFCC_l$ (avec $n_{\Delta\Delta}=2$)	29
1.15	Systèmes de reconnaissance de mots isolés (approche globale)	31
2.1	Hypothèse biologique de génération d'un comportement intelligent	38
2.2	Exemple de neurone biologique	39
2.3	Schématisation d'un neurone. À gauche, les dendrites et le corps de la cellule ; au centre, l'axone ; à droite, les axones terminaux.	40
2.4	Mise en correspondance neurone biologique / neurone formel	40
2.5	Exemple de neurone formel	41
2.6	Exemple de fonctions binaires à seuil (d'après [21])	42
2.7	Exemple de fonctions à saturation (d'après [21])	42
2.8	Exemple de fonctions non linéaires dérivables (a) fonction tangentielle (b) fonction sigmoïdale (d'après [21])	42
2.9	Définition des couches d'un réseau multicouche	44
2.10	Réseau avec raccourcis	45

2.11	Réseau à connexions locales	45
2.12	Réseau à connexions récurrentes	46
2.13	Réseau à connexions complète	46
2.14	Un Perceptron Multi-Couches	48
2.15	Descente de gradient sur une surface d'erreur et minima locaux	49
3.1	Différentes branches des algorithmes évolutionnaires	55
3.2	Amélioration de l'adaptation des individus au cours de l'évolution	56
3.3	Niveaux d'organisation de l'algorithme génétique	58
3.4	Principe général d'un algorithme génétique simple	60
3.5	Exemple d'application de la <i>Roulette Wheel Selection</i>	65
3.6	Croisement à 1 point (one point crossover)	66
3.7	Croisement à 2 points (two points crossover)	67
3.8	Croisement uniforme (uniform crossover)	67
3.9	Opération de Mutation	69
4.1	Convergence des stratégies de recherche de [78]	75
4.2	Schéma général du système de reconnaissance vocal	80
4.3	Chaîne de traitement acoustique d'un système de reconnaissance de la parole	80
4.4	Mise en forme du signal vocal	82
4.5	Modèles de population de SRAPs	84
4.6	Structure générale d'un individu (SRAP)	85
4.7	Représentation génotypique du réseau	86
4.8	Schéma représentant la structure du chromosome	90

Table des Matières

ملخص.....	i
ABSTRACT.....	ii
RESUME.....	iii
DEDICACE.....	iv
REMERCIEMENTS.....	v
LISTES DES TABLEAUX.....	vi
LISTES DES FIGURES.....	vii
TABLES DES MATIERES.....	ix
INTRODUCTION GENERALE.....	1
CHAPITRE 1 : RECONNAISSANCE AUTOMATIQUE DE LA PAROLE.....	4
1. Introduction.....	4
2. Quelques définitions.....	5
2.1. Modes de fonctionnement.....	5
2.2. Mode d'élocution.....	5
2.3. Mesure de performance.....	5
3. L'appareil phonatoire.....	6
4. Traitement de la parole.....	7
4.1. Classification des tâches.....	8
4.1.1. Codage de la parole.....	8
4.1.2. Synthèse vocale.....	8
4.1.3. Reconnaissance vocale.....	8
4.1.4. Vérification ou l'identification du locuteur.....	9
5. Reconnaissance de la parole.....	9
5.1. Les objectifs d'un système de reconnaissance de parole.....	9
5.2. Schéma général d'un système de reconnaissance de parole.....	10
5.2.1. Interface audio.....	12
5.2.2. Détection de parole.....	12
5.2.3. Analyse acoustique.....	13
5.2.4. Décodage acoustique.....	13
5.3. Applications de la reconnaissance de la parole.....	14
6. Le signal acoustique.....	14
7. Complexité du signal de parole.....	15
7.1. Redondance.....	16
7.2. Continuité et coarticulation.....	16
7.3. Variabilité.....	17
7.3.1. Variabilité inter-locuteurs.....	17
7.3.2. Variabilité intra-locuteurs.....	17
8. Analyse et représentation du signal de parole.....	18
8.1. Représentation non paramétrique.....	18
8.1.1. Energie du signal.....	18
8.1.2. Transformée de Fourier.....	19
8.2. Représentation paramétrique.....	20
8.2.1. Codage prédictif linéaire (LPC).....	21

8.2.2. Représentation cepstrale.....	22
8.2.3. Coefficients cepstraux (MFCC)	23
8.2.4. Coefficients PLP.....	25
8.2.5. Coefficients RASTA-PLP.....	27
8.2.6. Dérivées première et seconde.....	28
8.2.7. Réduction de l'espace de représentation.....	29
8.3. Choix d'une représentation du signal.....	30
9. Approches de reconnaissance vocale.....	30
9.1. Approche analytique.....	30
9.2. Approche globale.....	31
10. Méthodes de classification et de reconnaissance vocale.....	32
10.1. La classification automatique.....	33
10.2. La classification statistique.....	33
10.2.1. Décision Bayésienne.....	34
10.2.2. La méthode des <i>k</i> -plus proches voisins (<i>k</i> -ppv)	34
10.3. La classification stochastique.....	34
10.4. La classification neuronale.....	35
11. Conclusion.....	35
CHAPITRE 2 : LES RESEAUX DE NEURONES.....	37
1. Introduction.....	37
2. Neurone biologique.....	38
3. Neurone formel.....	40
3.1. Structure.....	41
3.2. Comportement.....	41
4. Classification et propriétés.....	43
4.1. Type d'Architecture du Réseau.....	43
4.1.1. Les réseaux de neurones non bouclés.....	43
4.1.2. Les réseaux de neurones bouclés.....	45
4.2. Apprentissage.....	47
4.2.1. Apprentissage supervisé.....	47
4.2.2. Apprentissage non supervisé.....	48
5. Le modèle du Perceptron Multi-Couches.....	48
5.1. Architecture.....	48
5.2. Apprentissage.....	49
6. Revue des différentes applications des réseaux de neurones.....	50
7. Avantages et limites d'utilisation des réseaux de neurones.....	51
8. Conclusion.....	52
CHAPITRE 3 : LES ALGORITHMES GENETIQUES.....	54
1. Introduction.....	54
2. Principes généraux.....	55
3. Présentation des algorithmes génétiques.....	57
4. Mécanisme d'un algorithme génétique.....	58
5. Codage (représentation) d'un individu.....	61
6. Génération de la population initiale.....	62
7. Fonction d'adaptation (<i>fitness</i>)	63
8. Opérateurs génétiques.....	63
8.1. Opérateur de sélection (ou reproduction)	64
8.1.1. Sélection ordonnée (<i>Rank selection</i>).....	64

8.1.2. Roulette biaisée (roulette wheel selection)	64
8.1.3. Sélection par tournoi (Tournament selection)	65
8.1.4. Sélection uniforme (Uniform selection)	65
8.2. Opérateur de croisement (recombinaison ou crossover)	66
8.2.1. Croisement à "1 point" (one-point)	66
8.2.2. Croisement à "2 points" (two-points)	67
8.2.3. Croisement uniforme (uniform crossover)	67
8.3. Opérateur de mutation.....	68
9. Principaux paramètres d'un algorithme génétique.....	69
10. Discussion sur les algorithmes génétiques.....	70
11. Conclusion.....	71

CHAPITRE 4 : PROPOSITION D'UN SYSTEME HYBRIDE POUR LA RECONNAISSANCE DE LA PAROLE..... 73

1. Introduction.....	73
2. Orientation vers les approches hybrides.....	73
3. Pourquoi les algorithmes génétiques.....	74
4. Intégration Algorithme Génétique/Réseaux de Neurones.....	76
5. Notre choix.....	78
6. Architecture générale du système de reconnaissance proposé.....	79
7. Paramétrisation du signal vocal.....	80
7.1. Base de données.....	80
7.2. Acquisition du signal vocal.....	81
7.3. Extraction des paramètres acoustiques.....	82
7.3.1. Mise en forme du signal vocal.....	82
7.3.2. Paramètres acoustiques.....	83
8. Présentation du modèle de population.....	84
8.1. Architecture du réseau neuronal.....	85
8.2. Codage des chromosomes (Représentation des solutions)	85
8.3. Génération et initialisation de population.....	86
8.4. Evaluation.....	87
9. Evolution.....	87
9.1. Sélection ou reproduction.....	87
9.2. Croisement.....	87
9.3. Mutation.....	88
10. Stratégie de sélection du meilleur individu PMC.....	89
10.1. Critère de sélection.....	89
10.2. Représentation.....	90
10.3. Algorithmes de recherche.....	91
10.3.1. Les algorithmes génétiques.....	91
10.3.2. La recherche Tabou (Tabu search)	91
10.3.3. La recherche par accroissement basé sur la population (PBIL).....	93
11. Conclusion.....	94

CONCLUSION ET PERSPECTIVES..... 95

REFERENCES BIBLIOGRAPHIQUES..... 97

Introduction générale

La conception et la réalisation des systèmes informatiques, tels que par exemple des serveurs ou des gestionnaires d'informations, destinés à des usagers professionnels ou occasionnels, doivent intégrer de façon explicite et intentionnelle le traitement de tous les aspects de la communication homme-machine. En effet, nombreux sont les exemples d'échecs de systèmes, aux conséquences économiques souvent lourdes, dus à une conception superficielle de l'interface gérant les interactions entre les usagers et le système. La prise en compte de nombreuses facettes de la communication homme-machine nécessite différents modèles (de l'utilisateur, du dialogue, de connaissances..., etc) à définir ou à adapter à partir des résultats existants.

Parmi les moyens de communication d'un système, la parole se révèle l'un des plus intéressants. Du point de vue de l'utilisateur et du système, l'usage de la parole avec celui naturellement associé de la langue naturelle facilite la manipulation des informations fournies ou reçues du système. D'autre part, la parole s'est avérée être le plus performant des médias lors de nombreuses expériences effectuées en laboratoire ou en situation réelle. Ces caractéristiques justifient les recherches effectuées sur le traitement de la parole et la progression, lente mais régulière, du nombre de systèmes commercialisés et mis à la disposition du public. Cependant, les résultats encore peu fiables (sans restriction d'environnement) des techniques de reconnaissance de la parole et les difficultés de compréhension du langage naturel en parole spontanée constituent des obstacles qui limitent la mise en place d'un plus grand nombre de systèmes oraux. Il apparaît alors intéressant de rechercher des méthodes et des moyens pour pallier les problèmes posés par la reconnaissance et la compréhension de la parole. Les capacités de dialogue du système, l'ajout de moyens de communication supplémentaires (multimodalité) et l'architecture même du système forment une bonne partie de ces méthodes et moyens.

Les réseaux de neurones peuvent être regardés comme des modèles non linéaires paramétriques possédant des propriétés d'approximation universelle. Les réseaux de neurones ont une lointaine inspiration biologique : un réseau de neurones est composé d'unités simples (neurones) connectées à travers des liens caractérisés par des valeurs numériques (poids synaptiques). Pour résoudre (modéliser) un problème à l'aide de réseau neuronal, il est

nécessaire de trouver une architecture adéquate du réseau (identification du modèle) et des valeurs optimales pour les poids des connexions (estimation du modèle).

Les algorithmes génétiques sont des algorithmes de recherche (et sous certaines conditions des algorithmes d'optimisation) qui peuvent être utilisés, entre autres, pour déterminer le réseau neuronal capable de modéliser un problème. D'inspiration biologique, les algorithmes génétiques ont un caractère stochastique et travaillent sur une population d'individus, chaque individu représentant une solution alternative. Un individu est décrit par une chaîne de valeurs numériques (génome). Les individus de la population sont évalués par rapport à un critère qui dépend du problème à résoudre. Une descendance est ensuite générée en privilégiant les meilleurs individus trouvés (sélection) et en utilisant des opérateurs comme la recombinaison et la mutation. La descendance remplace la population courante et l'algorithme continue.

Dernièrement, une extension de ces deux paradigmes d'IA (les réseaux de neurones et les algorithmes génétiques) a été mise au point donnant naissance aux modèles hybrides. Cette hybridation se base essentiellement sur la complémentarité qui peut exister entre ces deux paradigmes et tente ainsi de les faire coopérer pour résoudre un problème de reconnaissance vocale, où chacun d'entre eux prendra à sa charge le traitement d'une tâche qui s'accomode le mieux avec son style de raisonnement et où chacun viendrait pallier les inconvénients de l'autre sans pour autant en renforcer les faiblesses.

Nous nous intéressons à la conception et au développement d'un système de reconnaissance vocal des chiffres isolés indépendant de locuteur. La reconnaissance par les méthodes les plus performantes de l'état de l'art reste insuffisante ; cette faiblesse est un facteur limitant des systèmes de RAP. Nous cherchons à améliorer les performances des SRAP en mettant comme objectif l'augmentation du taux de reconnaissance, appliquant ainsi une nouvelle technique qui combine efficacement les AGs et les RNs, particulièrement, les perceptrons multi-couches (PMCs) ayant comme principal but d'éliminer les limites et les faiblesses soulevées par chacun d'entre eux.

Le système proposé permet de rejoindre les capacités discriminantes, la capacité des AGs d'effectuer une recherche dans l'espace global des solutions potentielles, et des RNs qui se concentrent plutôt vers une recherche locale dans l'espace du voisinage (dont la solution trouvée, supposée optimale, ne l'est peut-être pas si l'on considère l'espace global). On peut supposer donc que ces deux méthodes sont complémentaires.

L'organisation générale du présent mémoire est articulée en 4 chapitres comme suit :

Le chapitre 1 présente l'état de l'art dans le domaine de la reconnaissance de la parole où nous décrivons les étapes nécessaires au développement d'un système de reconnaissance vocale. Une revue de littérature sur les différentes méthodes de reconnaissance est élaborée, ainsi que les techniques d'analyse du signal vocal afin d'extraire des paramètres pertinents qui nous serviront de données pour la reconnaissance.

Le chapitre 2 et le chapitre 3 sont consacrés à la présentation des deux paradigmes d'IA (les réseaux de neurones et les algorithmes génétiques), leurs avantages et leurs limites.

Le chapitre 2 est consacré à l'état de l'art des réseaux de neurones, entre autre le PMC adopté dans notre projet. Une description de la méthode d'apprentissage utilisée est aussi présentée.

Le chapitre 3 est une revue de littérature détaillée sur les algorithmes génétiques. Dans cette partie, nous expliquons les différentes étapes de l'algorithme génétique, les opérateurs de reproduction (croisement et mutation) et le choix des paramètres.

Au chapitre 4, nous présentons notre système hybride développé pour la reconnaissance de la parole. En premier, nous mettons le point sur l'orientation vers les approches hybrides. Nous montrerons par la suite les différentes méthodes de combinaison qui peuvent être effectuées entre les AGs et les RNs.

Nous achèverons ce mémoire par une conclusion et quelques perspectives.

Chapitre 1

Reconnaissance Automatique de la Parole

“For millions of years, mankind live just like the animals, Then something happened which unleashed the power of our imagination: We learned to talk.”

Pink Floyd
Keep Talking

*“Lorsque **moi** j’emploie un mot”, répliqua Heumpty Deumpty d’un ton de voix quelque peu dédaigneux, “il signifie exactement ce qu’il me plaît qu’il signifie... ni plus, ni moins”.*

“La question”, dit Alice, “est de savoir si vous avez le pouvoir de faire que les mots signifient autre chose que ce qu’ils veulent dire”.

“La question”, riposta Heumpty Deumpty, “est de savoir qui sera le maître... un point, c’est tout”.

Lewis Carroll
De l’autre côté du miroir

1. Introduction

La parole est le principal moyen de communication dans toute société humaine. Son apparition peut être considérée comme concomitante à l'apparition des outils, l'homme ayant alors besoin de raisonner et de communiquer pour les façonner [88]. Son abstraction par rapport à un support physique en fait un moyen de communication très simple à utiliser. L'ère industrielle a par ailleurs permis de mettre en place des moyens d'enregistrement, et donc de sauvegarde, qui permettent à la parole de se hisser au rang de l'écrit pour la conservation de la connaissance.

L'importance de la parole fait que toute interaction homme-machine devrait plus ou moins passer par elle. D'un point de vue humain, la parole permet de se dégager de toute obligation de contact physique avec la machine, libérant ainsi l'utilisateur qui peut alors effectuer d'autres tâches. Sans pour autant imposer la parole là où elle pourrait être un frein à l'interaction (il est par exemple difficile d'imaginer une application graphique où seule la parole serait utilisée), son utilisation permettrait de commencer à limiter l'emploi des claviers, tablettes graphiques et autres.

La Reconnaissance Automatique de la Parole (RAP) par les machines est que des personnes puissent dialoguer en langage naturel avec des ordinateurs. Elle constitue un thème de recherche qui fascine le public et demeure un défi pour les chercheurs et les performances des Systèmes de Reconnaissance Automatique de la Parole (SRAP) actuels pour les langues latines (anglais, français, allemand, espagnol,...) ne sont plus très loin de l'attente du grand public. Nous cherchons à rendre ces machines accessibles par la voix sans avoir besoin d'un clavier ou d'une souris.

Les recherches en RAP ont véritablement commencé dans les années soixante-dix, notamment avec le projet ARPA (Advanced Research Projects Agency) [106] de "compréhension de la parole", lancé en 1971. Les conclusions de ce premier projet ARPA marquaient la fin d'une période optimiste où une partie de la communauté espérait des progrès rapides des SRAPs. Depuis, les SRAPs expérimentaux mono-locuteur, en mots isolés et à petit vocabulaire des années quatre-vingt, les laboratoires de recherche travaillent maintenant sur des systèmes multi-locuteurs, en parole continue et à très grand vocabulaire. Aujourd'hui, la recherche en RAP tente de mieux comprendre le processus humain de génération et de compréhension de la parole, tant d'un point de vue mécanique par le biais de l'étude et de la modélisation des organes biologiques en charge de ces tâches, que d'un point de vue mathématique par le développement de méthodes de classification toujours plus fines et exactes.

2. Quelques définitions

2.1. Modes de fonctionnement

Les différents modes de fonctionnement qui caractérisent un système de reconnaissance sont les suivants :

- a. *Mono-locuteur (dépendant du locuteur)*: un seul locuteur peut utiliser le système de reconnaissance de la parole à un instant donné après l'avoir adapté à sa voix.
- b. *Multi-locuteur* : le système de reconnaissance peut être utilisé par un groupe restreint de personnes.
- c. *Indépendant du locuteur* : n'importe quel locuteur peut utiliser le système de reconnaissance.

2.2. Mode d'élocution

Le mode d'élocution caractérise la façon dont on peut parler au système de reconnaissance. Il existe trois modes d'élocution distincts [74, 84] :

- a. *Mots isolés*: chaque mot est prononcé isolément (précédé et suivi d'une pause). Ce mode, purement acoustique, ne nécessite aucune notion de contexte, de sémantique ou de syntaxe, c'est-à-dire aucun modèle de langage.
- b. *Mots connectés* : le système reconnaît des séquences de quelques mots (phrases) sans pause entre les mots. Comme dans le mode "mots isolés", le mode de "mots connectés" est purement acoustique et ne fait pas intervenir de modèles de langage.
- c. *Parole continue* : le locuteur peut prononcer des phrases au sens habituel du terme, sans avoir à respecter des contraintes. C'est le mode le plus sophistiqué. Son but est d'associer, à la partie acoustique, une partie modèle de langage. Cela permet une dictée en continu.

2.3. Mesure de performance

La mesure de performance ou taux de reconnaissance permet de mesurer l'efficacité du système de reconnaissance testé. Ce taux varie fortement selon le type de canal de transmission utilisé (microphone, téléphone), la taille du vocabulaire et le mode d'élocution. Il existe différentes valeurs mesurant les performances d'un système de reconnaissance :

- a. *Taux de reconnaissance*: le pourcentage de mots ou de phrases correctement reconnues.
- b. *Taux de substitution* : le pourcentage de mots ou de phrases pour lesquels le système fait une erreur de reconnaissance.

- c. *Taux de rejet* : le pourcentage de mots ou de phrases que le système n'a pas compris.
- d. *Taux d'omission* : le pourcentage de mots ou de phrases non détectés.
- e. *Taux d'insertion* : le pourcentage de réponses inopinées.

3. L'appareil phonatoire

La reconnaissance approfondie du mécanisme de production de la parole chez l'homme contribue à faciliter la compréhension des caractéristiques du signal vocal et d'identifier la forme de la source d'excitation pour bien analyser le signal produit [24]. Une coupe de l'appareil phonatoire humain est fournie en figure 1.1.

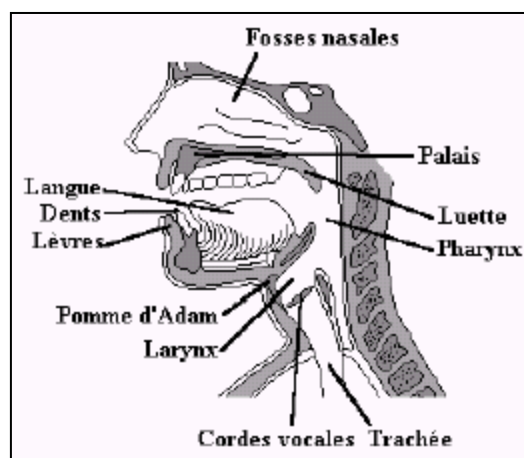


Figure 1.1 : Coupe de l'appareil phonatoire humain (d'après [97])

La parole se distingue des autres sons par des caractéristiques acoustiques ayant leurs origines dans les mécanismes de production. Le signal de parole appartient à la classe des signaux acoustiques produits par des vibrations de couches d'air. Les fluctuations de la pression de l'air produisent des variations de ce signal en fonction du temps qui peuvent être enregistrées de façon analogique ou digitale. Ceci constitue une représentation élémentaire du signal de parole [18]. Le principe de production de la parole est montré dans la figure 1.2. Le diaphragme expulse l'air des poumons, produisant ainsi l'énergie nécessaire à la parole. L'air arrive via la trachée-artère dans le larynx où se trouvent les cordes vocales. Si le son à produire est non voisé ou chuchoté, les lèvres symétriques constituant les cordes vocales forment une grande ouverture triangulaire nommée glotte et l'air continue son chemin. Si le son est voisé, les cordes vibrent périodiquement (ouverture brusque, fermeture plus progressive) ce qui module le son en impulsions périodiques de pression dont la fréquence est nommée fondamentale ou *pitch* (de 80 à 600 Hz). L'air arrive alors dans le conduit vocal proprement dit, avec d'abord la cavité

pharyngienne, puis en parallèle la cavité buccale et la cavité nasale. Cette dernière peut être utilisée ou non grâce au voile du palais qui permet une isolation totale de la cavité nasale.

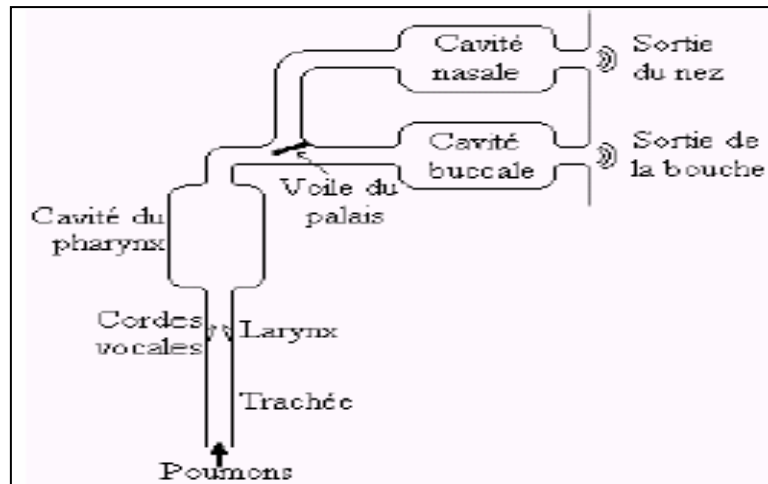


Figure 1.2 : Modèle de production de la parole

4. Traitement de la parole

Par traitement de la parole, on entend le traitement de l'information contenue dans le signal vocal. L'objectif poursuivi est la transmission ou l'enregistrement de ce signal, ou encore sa synthèse ou sa reconnaissance [18].

Le traitement automatique de la parole est aujourd'hui une composante fondamentale des sciences de l'ingénieur. C'est un champ de recherche riche mais difficile qui fait collaborer de nombreuses disciplines : traitement du signal, théorie de l'information, statistiques, algorithmique, linguistique, phonétique, acoustique, voire biologie et neurosciences...etc jusqu'à l'intelligence artificielle (IA) pour le développement de systèmes experts [18]. Située au croisement du traitement du signal numérique et du traitement du langage (c'est-à-dire du traitement de données symboliques), cette discipline scientifique a connu depuis les années soixante une expansion fulgurante, liée au développement des moyens et des techniques de télécommunications.

L'importance particulière du traitement de la parole dans ce cadre plus général s'explique par la position privilégiée de la parole comme vecteur d'information dans notre société humaine.

L'extraordinaire singularité de cette science, qui la différencie fondamentalement des autres composantes du traitement de l'information, tient sans aucun doute au rôle fascinant que joue le cerveau humain à la fois dans la production et dans la compréhension de la parole et à l'étendue

des fonctions qu'il met, inconsciemment, en œuvre pour y parvenir de façon pratiquement instantanée [37].

4.1. Classification des tâches

Les objectifs principaux poursuivis en traitement de la parole sont les suivants:

- Un codage efficace du signal vocal ;
- La synthèse du signal vocal ;
- La reconnaissance de la parole, base indispensable comme la synthèse pour le développement d'un système de dialogue homme-machine ;
- L'identification ou la vérification d'un locuteur.

Toutes ces applications exigent le développement de méthodes d'analyse efficaces du signal vocal.

4.1.1. Codage de la parole

Le codage de la parole est une opération qui consiste à ne prendre en compte que la valeur du signal analogique, ou celle de ses paramètres représentatifs, à des instants discrets du temps (*échantillonnage*) et à la représenter par un nombre réduit parmi l'ensemble fini des valeurs discrètes attribués au signal (*quantification*) [24].

4.1.2. Synthèse vocale

Qu'est-ce que la synthèse de la parole?.

Une simple réponse à cette question pourrait être : "*la production de la parole par une machine*". Autrement dit: "*la synthèse de la parole est un terme générique qui désigne la sortie vocale d'un système informatique*".

Son principal objectif est de produire des sons de parole à partir d'une représentation phonétique du message et permet la transmission d'informations sous forme orale soit en l'absence d'écran, soit en complément avec celui-ci [24].

4.1.3. Reconnaissance vocale

La reconnaissance de la parole est très souvent basée sur une représentation paramétrique du signal. Son but est la communication en langage naturel avec une machine. Il s'agit là de deux objectifs différents que l'on peut assigner à un système : la reconnaissance conduisant à une application de type dictée automatique (*la machine à écrire phonétique*), tandis que la compréhension automatique cherche à accéder à la signification de l'énoncé parlé.

4.1.4. Vérification ou l'identification du locuteur

Il convient de chercher à reconnaître non pas ce qui a été dit, mais l'identité de la personne qui parle, à partir de son "*empreinte*" vocale. La reconnaissance du locuteur regroupe en fait deux tâches distinctes:

- a. *La vérification du locuteur*: Il s'agit ici, après que le locuteur ait décliné son identité vocalement, de vérifier l'adéquation du message vocal qu'on lui demande d'émettre avec la référence acoustique du locuteur qu'il prétend être [24].
- b. *Identification du locuteur*: Là, il convient de comparer un message vocal avec un ensemble de références acoustiques correspondant à plusieurs personnes (un nombre fini et préétabli de locuteurs), et de déterminer par cet examen quelle est la personne qui a parlé [24].

5. Reconnaissance de la parole

La notion de reconnaissance est souvent confondue avec la compréhension qui consiste à chercher la signification de la parole prononcée. Par contre, la reconnaissance réalise seulement sa dictée sans chercher à savoir le sens de la parole.

Il y a quelques années, la recherche en reconnaissance de la parole était considérée pour le grand public comme un aimable passe-temps où l'on ne se préoccupait que de problèmes sans fondement réel.

Aujourd'hui, les temps ont changés, et la reconnaissance de la parole fait l'objet de plusieurs recherches récentes afin d'élargir les domaines d'interactions homme-machine (cette machine qui peut être un ordinateur, une machine à écrire, ou même un robot). En effet, la communication avec une machine capable de distinguer quelques mots de commandes prononcés isolément ou encore avec un ordinateur à qui on pourrait dicter un texte, ou lui demander une information, nécessite, évidemment, la création d'un *système de reconnaissance de parole*.

5.1. Les objectifs d'un système de reconnaissance de parole

Le développement de l'utilisation de la reconnaissance de la parole requiert :

- d'améliorer la robustesse de la reconnaissance vis à vis de l'environnement d'émission (bruit) ou des perturbations liées à la transmission (adaptation au canal de transmission utilisé, microphone ou ligne téléphonique) [28] ;

- d'améliorer le "confort d'usage" de ces systèmes en permettant à l'utilisateur de s'exprimer le plus librement possible (détection de mots clés au sein d'énoncés complets, augmentation de la taille du vocabulaire pouvant être reconnu) [28] ;
- de rendre cette technologie indépendante de l'application (aujourd'hui, une adaptation au vocabulaire, longue et coûteuse reste nécessaire : elle requiert la constitution de "bases de données de parole" correspondant aux enregistrements des mots devant être reconnus par plusieurs centaines de locuteurs) et l'adapter à différentes langues [28] ;
- d'améliorer la fiabilité de ces systèmes en les adaptant au mieux aux divers contextes d'applications (passage d'un mode de fonctionnement "indépendant du locuteur" à un mode "dépendant du locuteur", par exemple) [28] ;
- de préparer l'intégration optimale de modules de reconnaissance au sein des systèmes de "compréhension" qui seront nécessaires dans les systèmes de dialogue homme-machine du futur [28].

5.2. Schéma général d'un système de reconnaissance de parole

La parole s'est imposée comme le mode de communication entre individus par excellence. De nos jours, la parole devient également une interface pour la communication entre l'homme et la machine. Le problème de la RAP consiste à extraire, à l'aide d'un ordinateur, l'information contenue dans un signal de parole. Depuis plus de deux décennies, des recherches intensives dans ce domaine ont été accomplies par de nombreux laboratoires internationaux. Même si depuis les balbutiements de la RAP, avec des "machines" pouvant comprendre quelques mots prononcés isolément par une seule personne, il y a eu de nombreux progrès importants qui ont été accomplis grâce au développement d'algorithmes puissants ainsi qu'aux avancées en traitement du signal, nous n'avons pas encore vu naître la machine qui pourra comprendre n'importe quel interlocuteur humain s'exprimant de façon tout à fait courante. En effet, bien que quelques systèmes existent (Dragon system, en reconnaissance de grand vocabulaire [9], SPHINX, en reconnaissance de parole continue [85]), ils ne permettent pas d'assurer un réel dialogue homme-machine. Pour atteindre cette capacité de dialogue, il faut d'une part pouvoir reconnaître les formes acoustiques prononcées mais aussi les interpréter. Les aspects cognitifs et perceptifs sont donc liés.

Nous présentons dans ce qui suit une description générale du fonctionnement d'un système de reconnaissance automatique de la parole. La figure 1.3 ci-dessous illustre les différentes étapes du système de reconnaissance.

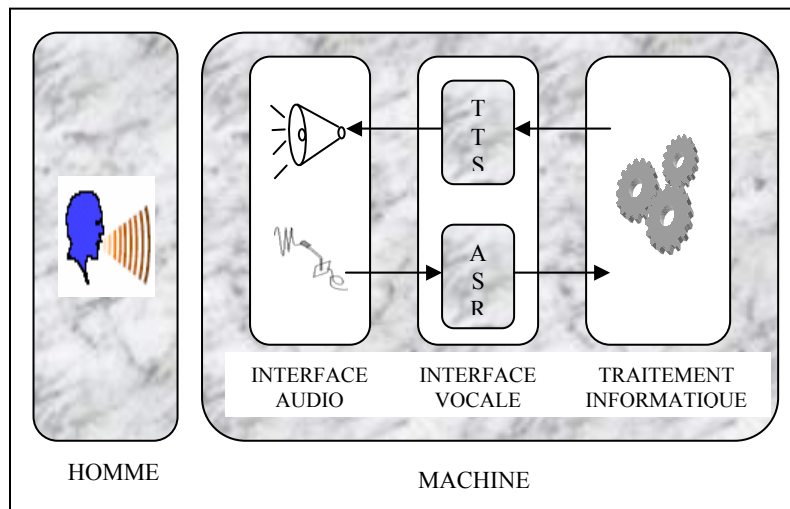


Figure 1.3 : Schéma d'une interface homme-machine vocale

Le locuteur convertit l'information qu'il souhaite communiquer à la machine en parole, c'est-à-dire en une séquence de sons structurés et associés pour former des mots et des phrases. La parole se propage alors sous la forme d'une onde acoustique jusqu'à un microphone qui constitue la porte d'entrée du système de *Reconnaissance Automatique de Parole* (Automatic Speech Recognition, ASR). Le signal délivré par le microphone y est décodé, c'est-à-dire l'information linguistique encodée par le locuteur dans le signal de parole est extraite sous la forme d'un mot ou d'une séquence de mots. Cette information peut ensuite être interprétée par la machine et l'action appropriée peut être prise. Par exemple, la machine peut générer un message vocal grâce à un système de *Synthèse Automatique de Parole* (Text-To-Speech, TTS) et l'émettre à destination du locuteur au moyen d'un haut-parleur.

Les systèmes de reconnaissance de parole modernes reposent sur une architecture séquentielle et modulaire (figure 1.4), dans laquelle une onde acoustique de parole est mesurée et analysée afin d'en extraire son contenu linguistique sous la forme d'un ou d'une séquence de mots [19, 51].

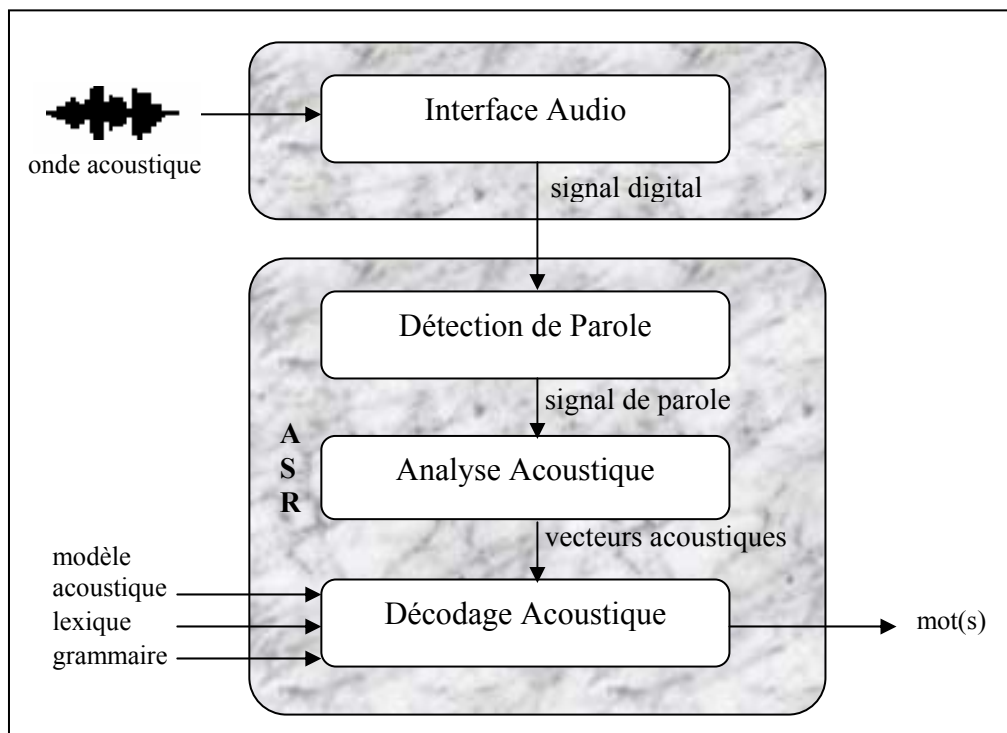


Figure 1.4 : Système classique de reconnaissance automatique de parole

5.2.1. Interface audio

Ce module réalise l'acquisition de l'onde acoustique de parole captée par un microphone et la convertit en une forme exploitable par la machine. Il existe de nombreux types de microphones mais tous assurent la même fonction : transformer les fluctuations de pression causées par l'onde acoustique de parole en un signal électrique. Ce signal subit ensuite une conversion analogique-numérique, c'est-à-dire qu'il est discrétisé à la fois en temps (échantillonnage) et en valeur (quantification). On obtient ainsi un signal digital sous la forme d'une séquence d'échantillons qui mesurent l'amplitude du signal du microphone à des instants régulièrement espacés, et l'amplitude de chaque échantillon est représentée sous sa forme digitale, utilisable par la machine.

5.2.2. Détection de parole

Ce module a pour fonction de détecter les segments où la parole est présente dans le signal digital. Seuls ces segments qui constituent le signal de parole sont transmis au module suivant. Le rôle du détecteur de parole est de limiter le coût de calcul et éviter d'exécuter abusivement le processus ASR lorsque des événements sonores imprévus se produisent. Cette fonctionnalité est parfois réalisée de manière manuelle : le locuteur est invité à presser un bouton pour activer le processus ASR lorsqu'il prononce le mot ou la phrase à reconnaître (mode *push-to-talk*).

5.2.3. Analyse acoustique

Le signal de parole démontre une très grande variabilité. Il est en effet peu probable de mesurer deux signaux de parole totalement semblables même si des mots identiques sont prononcés par le même locuteur. Cette variabilité rend le processus de reconnaissance automatique de parole excessivement complexe. Le rôle du module d'analyse acoustique est de traiter le signal de parole de manière à réduire cette variabilité.

5.2.4. Décodage acoustique

Pour pouvoir reconnaître un mot, le système ASR doit avoir appris à quoi ressemblent les réalisations acoustiques de ce mot en termes de vecteurs acoustiques. Durant une phase d'apprentissage, on présente au système ASR des exemples pour chacun des mots possibles. Un modèle statistique représentatif de la distribution des vecteurs acoustiques pour le mot considéré peut alors être construit. On obtient ainsi un *modèle acoustique* de la parole, c'est-à-dire un ensemble de modèles statistiques pour chacun des mots considérés, qui est stocké dans le système ASR. Cependant, une modélisation acoustique basée sur le mot devient problématique lorsque le nombre de mots possibles devient trop important. En particulier, la collecte des données pour réaliser la phase d'apprentissage devient fastidieuse voire irréaliste. On préfère généralement utiliser des unités linguistiques plus petites que le mot mais en nombre plus restreint pour décrire la parole. Un choix classique est le *phonème*. La plupart des langues peuvent être définies complètement par quelques dizaines de phonèmes. Ils constituent l'ensemble des sons élémentaires dont est composé l'énoncé de tout mot ou de toute phrase. Durant la phase d'apprentissage, un modèle statistique est calculé pour chaque phonème et les modèles de mots sont obtenus par concaténation des modèles de phonèmes. Une telle approche nécessite de disposer d'une transcription phonétique de chaque mot, c'est-à-dire de sa prononciation en termes de phonèmes. Pour cela, on recourt à un *lexique* contenant une ou plusieurs prononciations pour chaque mot.

En phase d'utilisation, étant donné un signal de parole, le module de *décodage acoustique* recherche la suite de mots dont la séquence de modèles associés est la plus « proche » de la série de vecteurs acoustiques observés. Ce problème est complexe car ni le nombre de mots, ni leur durée ne sont connus au préalable.

5.3. Applications de la reconnaissance de la parole

Les applications de la reconnaissance automatique de parole sont nombreuses. Elles existent là où la parole peut remplacer ou compléter une interface existante pour communiquer avec une machine, par exemple, pour accéder à un service ou contrôler une fonctionnalité d'un équipement. La parole s'impose parfois comme le seul mode de communication, par exemple dans les applications *mains-libres* où l'utilisateur ne touche pas l'équipement [35]. Le tableau 1.1 présente quelques exemples d'applications démontrant l'intérêt de la reconnaissance automatique de parole.

Domaine	Applications
Téléphonie	Automatisation de transactions téléphoniques (ex : opérations bancaires), self-service téléphonique pour l'accès à des services d'information (ex : consultation des bulletins météorologiques), etc.
Automobile	Contrôle mains-libres des équipements tels que la radio, le conditionnement dans le système de navigation, le téléphone sans fil (ex : voice dialing), les systèmes télématiques, etc.
Multimédia	Logiciels de dictée vocale, interaction vocale dans les logiciels pédagogiques (ex : apprentissage des langues) et ludiques (ex : jeux vidéo), etc.
Médical	Aide aux personnes handicapées.
Industriel	Contrôle vocal de machines, application pour la gestion de stocks, etc.

Tableau 1.1 : Exemples d'applications de la reconnaissance de la parole

6. Le signal acoustique

Le signal de parole est d'abord un signal sonore. C'est donc une onde matérielle de faible puissance se déplaçant dans l'air. C'est un phénomène de nature acoustique porteur d'un message [37]. L'information d'un message parlé réside dans les fluctuations de la pression de l'air, engendrées, puis émises par l'appareil phonatoire. Ces fluctuations constituent le *signal vocal*. Avant leur détection par l'oreille qui procède à une certaine analyse, elles sont sujet à des perturbations importantes : bruit ambiant, résonances, phénomène d'écho, ...etc. Les résultats sont transmis au cerveau qui les interprète [18, 89, 128, 129].

Le signal vocal représente la combinaison d'éléments simples et brefs du signal sonore appelé *les phonèmes*, qui permettent de distinguer les différents mots. La parole est un signal réel, continu, d'énergie finie et non stationnaire. Sa structure est complexe et variable avec le temps [57].

Sa composition est la suivante (figure 1.5) :

- Pseudo-périodique (D) → sons voisés;
- Aléatoire (A) → sons fricatifs;
- Impulsionnel (C) → phase explosive des sons occlusifs. (B est un bruit).

En acoustique, le son se définit classiquement au moyen de son *intensité*, sa *hauteur* qui est fixé par la fréquence de vibration des cordes vocales, appelée *fréquence du fondamental* ou *pitch* en anglais (F_0). Elle est de:

- 80 à 200 Hz pour une voie masculine;
- 50 à 450 Hz pour une voie féminine;
- 200 à 600 Hz pour une voie d'enfant.

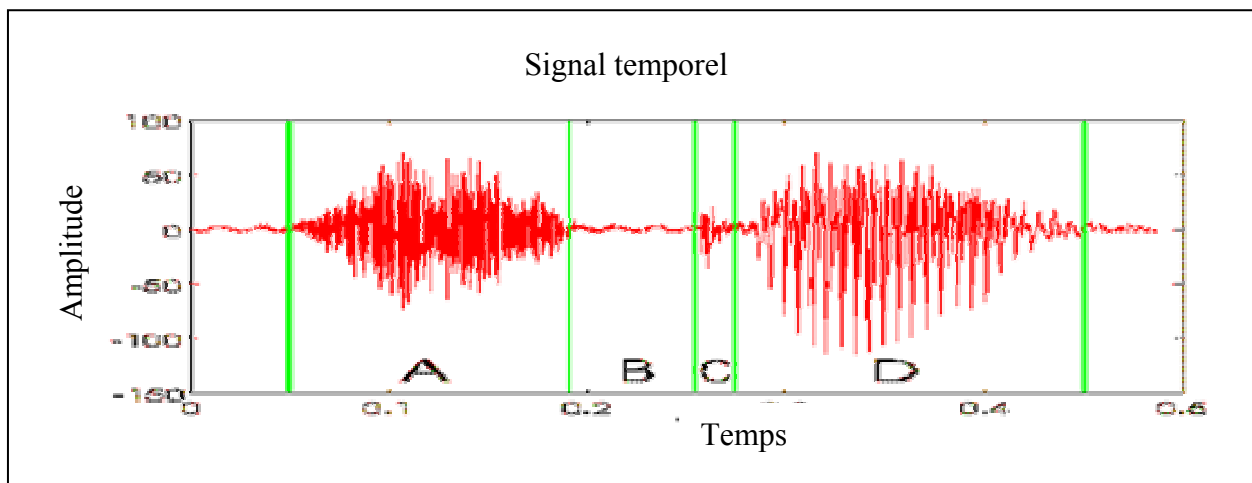


Figure 1.5 : Représentation temporelle du signal acoustique de la parole

Deux sons de même *intensité* et de même *hauteur* se distinguent par le *timbre* qui est déterminé par les amplitudes relatives des harmoniques du fondamental. Ces amplitudes nommées *formants* se caractérisent par les maximums de la fonction de transfert (transmittance) du conduit vocal. En général, les trois premiers formants sont essentiels pour caractériser le spectre du signal vocal [57].

7. Complexité du signal de parole

Le signal de parole est complexe, car sa structure résulte de l'interaction entre la production des sons et leur perception par l'oreille, avec comme objectif la transmission d'un message. En observant un enregistrement de parole, on constate que les phonèmes – les sons « élémentaires » identifiés par les linguistes – n'apparaissent pas comme des éléments isolés, mais s'enchaînent et se modifient par des phénomènes dits de coarticulation. Par ailleurs, il y a une grande

variabilité du signal, d'un locuteur à l'autre, ou en fonction des conditions d'enregistrement et de bruit de fond. Tout ceci contribue à la difficulté de la reconnaissance automatique de la parole [12].

7.1. Redondance

Le signal acoustique présente, dans le domaine temporel, une redondance qui rend indispensable un traitement préalable à toute tentative de reconnaissance. Il existe en effet une grande disproportion entre le débit du signal enregistré et la quantité d'information cherchée pour une tâche de reconnaissance. Un signal échantillonné à 16 kHz sur 16 bits représente un débit de 256 kbit/s, alors qu'une tâche de reconnaissance phonétique recherche typiquement une dizaine de phonèmes à la seconde, soit une compression de près de 10^4 du débit initial.

Ce constat motive la recherche d'une représentation plus compacte du signal. Il existe un grand nombre de paramètres possibles, souvent dérivés d'une analyse spectrale ou auto-régressive (AR). Le choix de paramètres adaptés à un type de problème dépend des conditions d'enregistrement du signal et surtout de l'usage ultérieur qui doit en être fait, puisque ce qui n'est pas pertinent vis-à-vis du contenu lexical peut le devenir pour l'identification du locuteur. Le choix de cette représentation ne résout pas les difficultés provenant de la continuité de la production et plus généralement de la variabilité de la parole.

7.2. Continuité et coarticulation

Tout discours peut être retranscrit par des mots, qui peuvent à leur tour être décrits comme une suite de symboles élémentaires appelés phonèmes par les linguistes. Cela laisse supposer que la parole est un processus séquentiel, au cours duquel des unités indépendantes se succèdent. Malheureusement, les spécialistes de phonétique eux-mêmes ont parfois des difficultés à identifier individuellement ces unités discrètes dans le signal, même si quelques événements acoustiques particuliers peuvent être détectés. La parole est en réalité un flux continu, et il n'existe pas de pause entre les mots qui pourrait faciliter leur localisation automatique par les systèmes de reconnaissance.

De plus, les contraintes introduites par les mécanismes de production créent des phénomènes de coarticulation. La production d'un son est fortement influencée par les sons qui le précèdent mais aussi qui le suivent en raison de l'anticipation du geste articulatoire. Ces effets s'étendent sur la durée d'une syllabe, voire même au-delà, et sont amplifiés par une élocution rapide. L'identification correcte d'un segment de parole isolé de son contexte est parfois impossible. La

prise en compte des phénomènes de coarticulation ne suffit pourtant pas à prédire la réalisation acoustique d'une phrase en raison de la grande variabilité de la parole.

7.3. Variabilité

La parole est un phénomène *a priori* très simple à comprendre, mais le problème posé en reconnaissance automatique de la parole est que la machine n'ayant aucune connaissance propre en compréhension de la parole. Tout système de RAP doit donc être défini par l'homme lui-même.

Cet apprentissage inconscient a été la cause d'une certaine naïveté lors des premières années de recherche en RAP. Depuis lors, la liste des différentes tâches qu'il faudra résoudre s'est précisée mais n'est peut-être pas encore exhaustive. Au rang des difficultés rencontrées se trouvent les problèmes de variabilité. On distingue généralement deux sources de variabilité qui peuvent rendre deux prononciations d'un même énoncé très différentes: la variabilité inter-locuteurs et la variabilité intra-locuteur.

7.3.1. Variabilité inter-locuteurs

La variabilité inter-locuteurs est *a priori* la plus importante. Les différences physiologiques entre locuteurs, qu'il s'agisse de la longueur du conduit vocal ou du volume des cavités résonantes, modifient la production acoustique; à cet égard, les voix d'hommes, de femmes et d'enfants sont les plus caractéristiques. A cela s'ajoutent les habitudes acquises en fonction du milieu social et géographique, comme les accents régionaux.

7.3.2. Variabilité intra-locuteurs

La variabilité intra-locuteur est plus réduite, mais n'est pas négligeable. Elle identifie les différences dans le signal produit par une même personne. Cette variation peut résulter de l'état physique ou moral du locuteur, par exemple la fatigue ou le rhume, les conditions physiologiques, comme le stress, et même le bruit de fond pendant l'élocution qui peut, parfois, être considérée comme une variabilité intra-locuteur, influent sur la production et sur la prosodie, entraînant des variations complexes du rythme d'élocution, de la mélodie et de l'intensité du discours. Une maladie des voies respiratoires peut ainsi dégrader la qualité du signal de parole de manière à ce que celui-ci devienne totalement incompréhensible, même pour un être humain. Ces phénomènes sont encore mal modélisés et compliquent la réalisation des systèmes de reconnaissance.

8. Analyse et représentation du signal de parole

Le traitement numérique des signaux connaît depuis trois décennies un développement fulgurant. Une multitude de méthodes puissantes de traitement des signaux peuvent désormais être mise en œuvre grâce aux techniques numériques. L'étude de la parole a été un des domaines importants qui a bénéficié et qui continue à bénéficier du traitement numérique des signaux [18].

Nous avons constaté dans le paragraphe précédent (§ 7.) que le signal de parole présente de la redondance et contient des informations jugées superflues pour la reconnaissance, ce qui justifie la recherche d'une représentation plus compacte. Cette représentation consiste à extraire un nombre réduit de coefficients représentatifs du signal traité, elle est généralement déduite d'une analyse à court-terme. Les paramètres extraits doivent être :

- a. *pertinents* : extraits de mesures suffisamment fines, ils doivent être précis mais leur nombre doit rester raisonnable afin de ne pas avoir de coût de calcul trop important dans le module de décodage.
- b. *discriminants* : ils doivent donner une représentation caractéristique des sons de base et les rendre facilement séparables.
- c. *robustes* : ils ne doivent pas être trop sensibles à des variations de niveau sonores ou à un bruit de fond.

Pour cela, il existe différentes techniques d'analyse vocale et de représentation du signal, chacune d'elles est basée sur une forme particulière du signal acoustique.

8.1. Représentation non paramétrique

Le signal de parole peut être analysé dans le domaine temporel ou dans le domaine spectral par des méthodes non paramétriques, sans faire l'hypothèse d'un modèle pour rendre compte du signal observé. Les représentations les plus souvent retenues sont l'énergie du signal et la transformée de Fourier.

8.1.1. Energie du signal

Après la phase de numérisation et surtout de quantification (§5.2.1.), le paramètre intuitif pour caractériser le signal ainsi obtenu est l'énergie. Cette énergie correspond à la puissance du signal. Elle est souvent évaluée sur plusieurs trames de signal successives pour pouvoir mettre en évidence des variations. La formule de calcul de ce paramètre est :

$$E(\text{fen\^etre}) = \sum_{n \in \text{fen\^etre}} |n|^2$$

Equation 1 : Calcul de l'\'energie d'un signal \'echantillonn\'e

Il existe des variations de ce calcul. L'une des plus utilis\'ees r\'ealise une simple somme des valeurs absolues des amplitudes des \'echantillons pour all\'eger la charge de calcul, les variations restant les m\^emes. D'autres, comme celle de Taboada et al. [134] proposent la modification suivante du calcul int\'egrant une normalisation par rapport au bruit ambiant.

$$E(\text{fen\^etre}) = \left(\sum_{n \in \text{fen\^etre}} \frac{|n|^2}{R} \right)$$

Equation 2 : Calcul de l'\'energie normalis\'e par rapport au bruit ambiant

Dans cette \'equation, R est la valeur moyenne de l'\'energie du bruit. Le r\'esultat de ce calcul tend vers 0 lorsque la portion consid\'eree est une zone o\`u il n'y a que le bruit de fond. Tout le probl\^eme de cette variante r\'eside dans l'estimation du facteur de normalisation R .

8.1.2. Transform\'ee de Fourier

Joseph Fourier a montr\'e que toute onde physique peut \^etre repr\'esent\'ee par une somme de fonctions trigonom\'etriques appel\'ee s\'erie de Fourier. Elle comporte un terme constant et des fonctions sinuso\'idales d'amplitudes diverses. Ainsi un son sinuso\'idal ne comporte qu'une seule raie spectrale correspondant \`a la fr\'equence de sa fonction sinus. Un son complexe est compos\'e d'une multitude de ces raies spectrales qui repr\'esentent sa composition fr\'equentielle [15].

Dans le cas d'une s\'equence d'\'echantillons, il est alors possible de calculer une Transform\'ee de Fourier Discr\^ete (TFD, *Discret Fourier Transform* – DFT – en anglais). L'\'equation 3 donne le calcul de la TFD pour une s\'equence $X(n)$ comportant N \'echantillons.

$$X(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(k) e^{-jk2\pi(n/N)}$$

Equation 3 : Formule de la Transform\'ee de Fourier Discr\^ete

L'intensit\'e en d\'ecibels (dB) du spectre est directement visualisable sous la forme d'un spectrogramme pour une \'evaluation qualitative du signal (figure 1.6).

$|X(n)|^2$ repr\'esente le spectre d'\'energie, il exprime la r\'epartition fr\'equentielle de l'\'energie du signal; il repr\'esente aussi un estimateur de la densit\'e spectrale court-terme, si on proc\^ede \`a une

pondération du signal par une fenêtre d'analyse. Le spectre du signal vocal est très utilisé dans le traitement et la reconnaissance de la parole et en particulier, la densité spectrale [14].

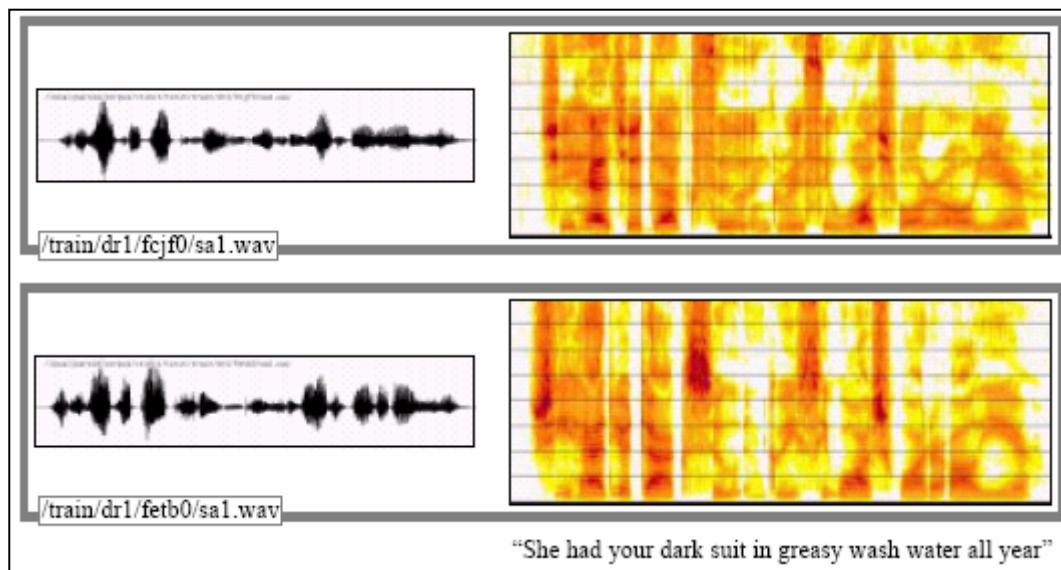


Figure 1.6 : Exemple de 2 signaux temporels (à gauche) et de 2 spectrogrammes (à droite) d'une même phrase prononcée par deux locuteurs différents (signal extrait du corpus TIMIT)

En 1965, Cooley et Tukey [29] ont proposé un algorithme de calcul rapide de transformée de Fourier discrète, la *Fast Fourier Transform* (FFT, Transformée de Fourier Rapide – TFR – en français). La seule limitation de cet algorithme est que la taille de la séquence dont on veut obtenir la FFT doit être une puissance de 2. Le temps de calcul d'une FFT est environ 10 fois inférieur à celui d'une TFD classique. Le nombre de paramètres spectraux calculés sur une trame par FFT reste trop élevé pour un traitement automatique ultérieur. L'énergie du spectre est calculée à travers un banc de filtres numériques couvrant la bande passante, ce qui permet de ne conserver qu'une vingtaine de valeurs d'énergie par exemple sur une bande passante de 8kHz.

8.2. Représentation paramétrique

Les méthodes précédentes ne font pas d'hypothèse sur la nature du signal analysé. Si l'on dispose d'un modèle adapté au signal de parole, l'estimation des paramètres du modèle doit permettre une meilleure caractérisation du signal. L'analyse par prédiction linéaire et l'analyse cepstrale utilisent un modèle de production. Des modèles issus d'études sur la perception sont aussi développés dans les laboratoires de recherche.

8.2.1. Codage prédictif linéaire (LPC)

L'analyse par prédiction linéaire est une alternative intéressante au calcul du spectre à court-terme pour plusieurs raisons : elle est basée sur un modèle de production simple mais cohérent, et ce modèle est facilement inversible, c'est-à-dire qu'il est possible de calculer les paramètres du modèle à partir des algorithmes rapides. Le cadre théorique est celui d'une modélisation source-filtre. Le conduit vocal est modélisé comme une succession de tubes de sections cylindriques qui sont des résonateurs. En tenant compte des autres facteurs participant à la production comme les cavités nasales résonantes et le rayonnement des lèvres, le filtre est modélisé par un modèle Auto-Régressif à Moyenne Ajustée (*ARMA*), dont la transmittance dans l'espace z est une fraction rationnelle. L'estimation des paramètres d'un modèle *ARMA* est délicate. Il est préférable de simplifier encore ce modèle et de modéliser le filtre comme un filtre linéaire tout-pôles qui produit un signal Auto-Régressif (*AR*); la théorie des développements limités montre que tout filtre *ARMA* peut être approximé par un filtre *AR* d'ordre suffisamment élevé.

Le codage prédictif linéaire (*LPC*, *Linear Predictive Coding*) est une méthode de codage et de représentation de la parole [95]. Elle repose principalement sur l'hypothèse que la parole peut être modélisée par un processus linéaire. Il s'agit donc de prédire le signal à un instant n à partir des p échantillons précédents, à laquelle s'ajoute un bruit blanc gaussien $e(n)$ de variance σ^2 :

$$\text{Le codage par prédiction linéaire } s(n) = \sum_{k=1}^p a_k s(n-k) + e(n) \quad \text{es coefficients } a_k \text{ qui minimisent} \quad (1.1)$$

l'erreur $e(n)$ par la méthode des moindres carrés. Cette minimisation conduit aux équations de Yule-Walker qui expriment le vecteur des coefficients $A = (1, a_1, \dots, a_p)^t$ comme :

$$R.A = (\sigma^2, 0, \dots, 0)^t \quad (1.2)$$

où la matrice d'autocorrélation R du signal est une matrice de Toëplitz constituée des $p+1$ premières coefficients d'autocorrélation :

$$R = [R_{ij}]_{1 \leq i, j \leq p+1} \quad \text{avec} \quad R_{ij} = r_{|i-j|} \quad (1.3)$$

Un algorithme rapide pour résoudre l'équation (1.2) et calculer les coefficients auto-régressifs $(a_i)_{i=1..p}$ a été développé par Levinson en 1947 et modifié par Durbin en 1960 [92]. Les applications de la prédiction linéaire en analyse de la parole sont nombreuses et anciennes [6]. Une transformation directe des coefficients auto-régressifs vers les coefficients cepstraux est possible.

8.2.2. Représentation cepstrale

L'analyse cepstrale est basée sur une connaissance du mécanisme de production de la parole. Elle est largement utilisée, car elle permet d'étudier séparément la source (signal excitateur) et la fonction de transfert du conduit vocal.

On part de l'hypothèse que le spectre instantané du signal de parole est le résultat de la convolution du spectre instantané du signal de la source par le filtre correspondant au conduit (figure 1.7) :

$$S(n)=E(n)*H(n) \tag{1.4}$$

avec :

$S(n)$: le signal temporel, $E(n)$: le signal excitateur et $H(n)$: la contribution du conduit.

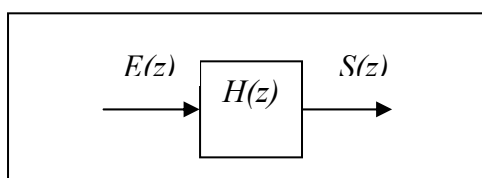


Figure 1.7 : Modèle simplifié de modèle de production de la parole

Le but de l'analyse cepstrale est de séparer ces deux contributions par déconvolution. Une transformation en Z permet de transformer la convolution en produit :

$$S(z)=E(z).H(z) \tag{1.5}$$

Ce produit devient une somme si les amplitudes spectrales sont estimées selon une échelle logarithmique, on obtient alors :

$$\text{Log}|S(z)|=\text{Log}|E(z)|+\text{Log}|H(z)| \tag{1.6}$$

Une transformation de Fourier inverse permet de retourner dans le domaine temporel et fournit le Cepstre. Cette technique est relativement lourde : elle nécessite deux transformées de Fourier et un calcul de logarithme (figure 1.8).

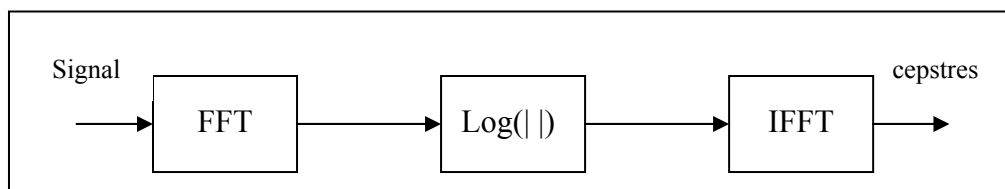


Figure 1.8 : Calcul des cepstres par analyse spectrale

Une première transformée de Fourier (FFT) est alors calculée pour obtenir un spectre du signal. Ces coefficients sont ensuite transformés par logarithme module ($\text{Log}(| |)$). La convolution étant un opérateur multiplicatif, ce passage par les logarithmes permet de passer les coefficients dans

un espace additif. Une transformée de Fourier inverse (IFFT) permet alors d'obtenir un cepstre dont un coefficient représente le fondamental, les autres coefficients permettant d'obtenir le spectre de la convolution effectuée sur le fondamental. Cette méthode de calcul des cepstres est élémentaire [24], il existe également des méthodes itératives effectuant un lissage, ce qui permet d'obtenir des cepstres de meilleure qualité.

Une extension possible des cepstres est leur passage dans un espace fréquentiel non linéaire proche de l'audition humaine. Il est ainsi possible de modifier la procédure de calcul précédente pour que les coefficients obtenus soient répartis selon une échelle Mel. Une telle procédure, proposée dans [31], permet d'obtenir des coefficients cepstraux à échelle Mel non linéaire, MFCC pour *Mel-scaled Frequency Cepstral Coefficients*, aussi nommés *Mel Frequency Cepstral Coefficients*.

8.2.3. Coefficients cepstraux (MFCC)

Les coefficients cepstraux (MFCC) ont été très utilisés en RAP du fait des bons résultats qu'ils ont permis d'obtenir. [31] avait d'ailleurs comparé cette méthode à d'autres du même ordre avec des conclusions qui, déjà, laissaient entrevoir la qualité des informations extraites par la méthode MFCC. Parmi les méthodes auxquelles les MFCCs avaient été comparés se trouvaient des méthodes fondées sur la prédiction linéaire (§ 8.2.1.).

La motivation d'extraction de coefficients MFCCs est leur correspondance avec la réponse en fréquence d'une oreille humaine. Pour transformer une fréquence linéaire en une fréquence Mel, on peut utiliser la formule de transformation suivante :

$$M_{mels} = x \cdot \log\left(1 + \frac{f}{y}\right)$$

Equation 4 : Correspondance entre l'échelle Mel et la fréquence en Hertz

où f est la fréquence en Hz, M_{mels} est la fréquence mel-échelle de f .

Plusieurs valeurs sont utilisées pour x et y . En 1989, on trouvait dans [24] $x=1000/\log(2)$ et $y=1000$. De nos jours, les valeurs les plus couramment utilisées sont $x=2595$ et $y=700$.

Pourtant l'utilisation de cette unité n'est pas suffisante. Pour avoir une largeur de bande relative qui reste constante, le banc de filtres Mel est construit à partir de filtres triangulaires positionnés uniformément sur l'échelle Mel donc non uniformément sur l'échelle fréquentielle. Cette répartition est illustrée ci-dessous :

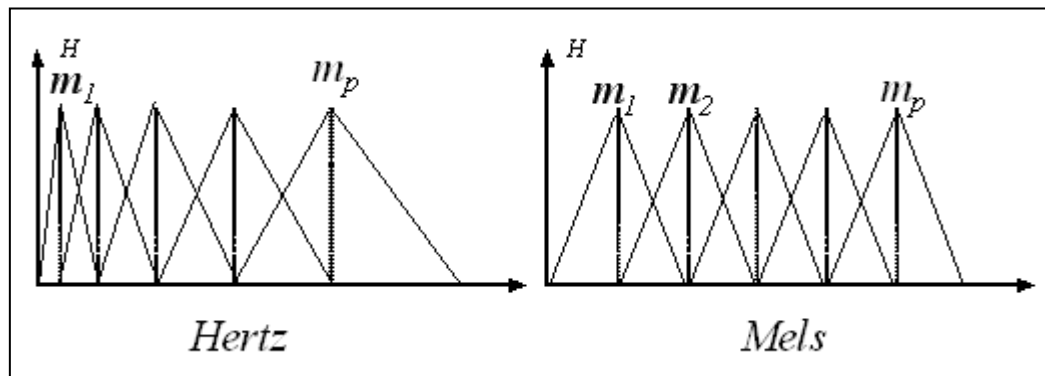


Figure 1.9 : Répartition des filtres triangulaires sur les échelles fréquentielle et Mel

Sur cette illustration, m_p correspond au nombre de filtres que l'on souhaite. Lorsque ce banc de filtres est en place, il est alors possible de calculer les coefficients MFCCs. L'algorithme peut être décrit comme suit :

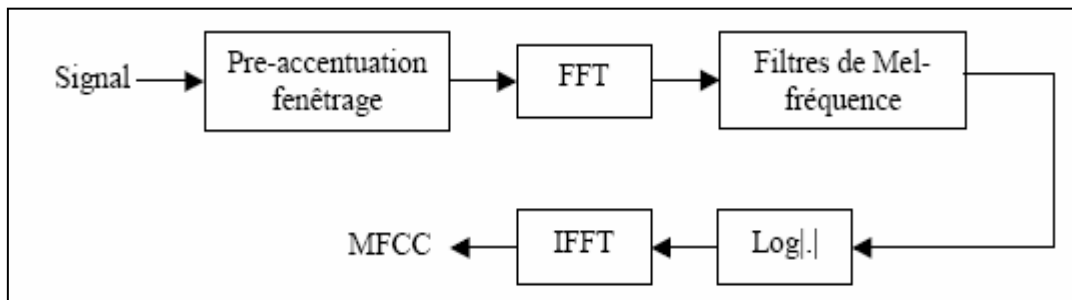


Figure 1.10 : Algorithme de calcul des *MFCCs*

Il est possible de choisir le nombre de paramètres générés en sorties de cet algorithme. Dans la littérature, le nombre de coefficients utilisés varie de 5 à plus d'une quarantaine en fonction de l'utilisation qui en est faite : reconnaissance de la parole, de la langue ou identification du locuteur par exemple. En ce qui concerne le nombre de filtres, nombreux sont ceux qui choisissent 30 pour un signal avec une bande passante de 0 à 8kHz [137]. En pratique, à partir d'une vingtaine de filtres en échelle Mel, une dizaine de coefficients cepstraux sont généralement considérés comme suffisants pour les expériences de reconnaissance de la parole et du locuteur [27].

S. Seneff décrit une modélisation basée sur les propriétés du système auditif humain. Un banc de filtres en bandes critiques répartis sur l'échelle Bark constitue l'entrée d'un modèle non-linéaire. Des paramètres BACC (*Bark Auditory Cepstral Coefficients*) en échelle Bark peuvent donc être calculés [125]. Les différences entre les deux échelles Mel et Bark sont peu importantes.

8.2.4. Coefficients PLP

La méthode *PLP*, [58, 59, 103], *Perceptual Linear Prediction* (ou *Perceptually based Linear Prediction*), est une méthode inspirée du principe de prédiction linéaire. Elle combine ce principe à une représentation du signal qui suit l'échelle humaine de l'audition. Elle est à l'origine de toute une famille de techniques de traitement du signal de parole. Le schéma fonctionnel de l'analyse PLP est montré dans la figure 1.11.

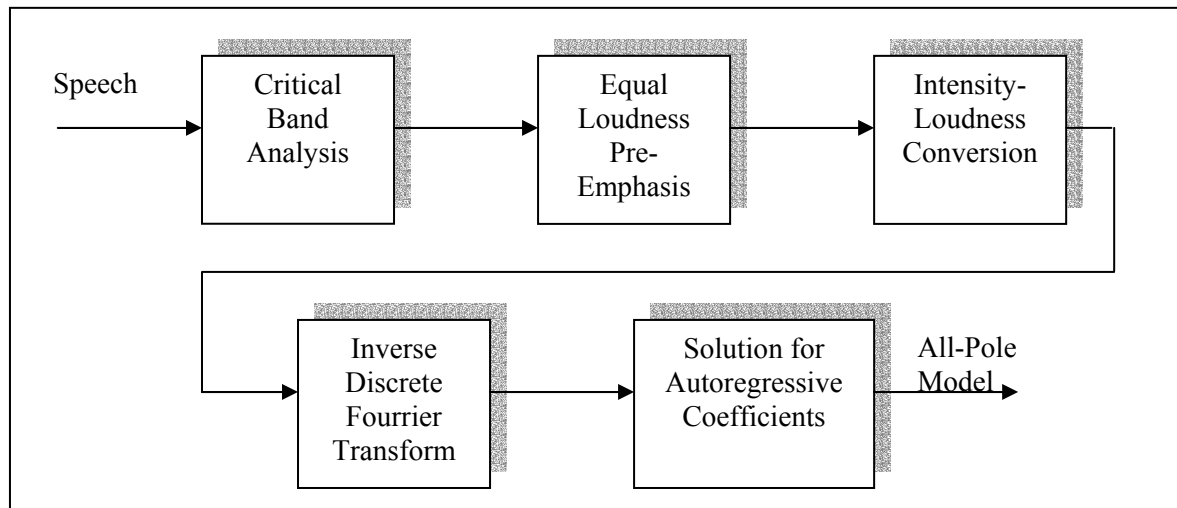


Figure 1.11 : Analyse PLP

La détermination des premiers paramètres perceptuels se base uniquement sur une des propriétés remarquables de l'oreille humaine. L'oreille intègre des zones de fréquence en bandes, appelées *bandes critiques* [112]. Cette méthode peut être résumée en trois phases de traitements successifs. Le signal de parole est tout d'abord analysé pour obtenir un spectre suivant une échelle d'audition. Ce spectre est ensuite modifié par une interpolation et une transformée de Fourier inverse, le signal obtenu étant passé dans un filtre pour réduire la dimension du spectre et augmenter la résolution fréquentielle. Une troisième étape, qui peut être omise, permet de reconstruire un signal de parole par filtrage inverse, passage dans le domaine fréquentiel hertzien et désaccentuation.

La première étape est précisément constituée par :

- une analyse en bandes critiques selon une échelle Bark par un banc de filtres ;
- une préaccentuation des valeurs obtenues selon une courbe suivant approximativement les mêmes principes que les traitements effectués par l'oreille, avec accentuation des basses fréquences et atténuation des hautes fréquences ;
- une application de la loi de préaccentuation de Steven.

La deuxième étape est, elle, constituée des phases suivantes :

- une interpolation des sorties des filtres du banc pour obtenir un spectre sur une échelle fréquentielle auditive ;
- une transformée de Fourier inverse qui permet de ramener le spectre obtenu dans le domaine temporel ;
- une résolution d'un ensemble d'équations linéaires pour obtenir les coefficients issus d'un filtre tout pôle d'ordre 5 (ce qui permet d'obtenir au moins deux sommets caractéristiques selon [58]).

Cette méthode a pour avantage de permettre une analyse et/ou un codage de la parole qui respectent le principe de la prédiction linéaire, qui suivent l'échelle fréquentielle observable dans l'oreille et, enfin, qui réduisent l'espace de représentation. Hermensky et al. [59] ont montré que les coefficients PLP introduisent une meilleure robustesse au bruit que les coefficients cepstraux classiques, comme illustré sur la figure 1.12.

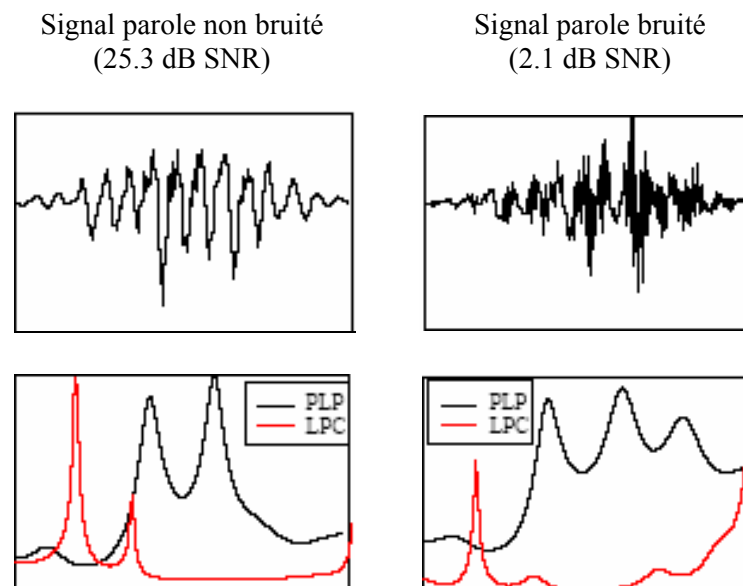


Figure 1.12 : Comparaison des spectres dérivés d'une analyse PLP et LPC

L'oreille humaine est également plus sensible aux variations relatives de la valeur d'un signal acoustique qu'à ses variations absolues. L'intégration de cette nouvelle propriété dans la technique PLP a conduit à l'élaboration des paramètres RASTA-PLP, RASTA étant l'acronyme de *RelAtive SpecTrAl* [61]. Ces paramètres bénéficient de très bonnes propriétés de robustesse. Aujourd'hui, les paramètres RASTA-PLP sont pratiquement toujours employés dans le domaine de la reconnaissance vocale et du locuteur. Mais, il n'existe pas de relations biunivoques entre ces paramètres et les paramètres LPC. Ils ne sont donc pas utilisés dans les techniques de codage de parole.

8.2.5. Coefficients RASTA-PLP

La méthode PLP [59], dont l'algorithme repose sur des spectres à court terme de la parole, résiste difficilement aux contraintes qui peuvent lui être imposées par la réponse fréquentielle d'un canal de communication. Pour atténuer les effets de distorsions spectrales linéaires, Hermansky [60, 61] propose de modifier l'algorithme PLP en remplaçant le spectre à court terme par un spectre estimé où chaque canal fréquentiel est modifié par passage à travers un filtre. Cette modification est à la base de la méthode RASTA-PLP. La mise en place de ce filtrage permet, lorsqu'il est effectué dans le domaine spectral logarithmique, de supprimer les composantes spectrales constantes, supprimant ainsi les effets de convolution du canal de communication. L'application de l'analyse RASTA-PLP sur une trame d'analyse est schématisée dans la figure 1.13.

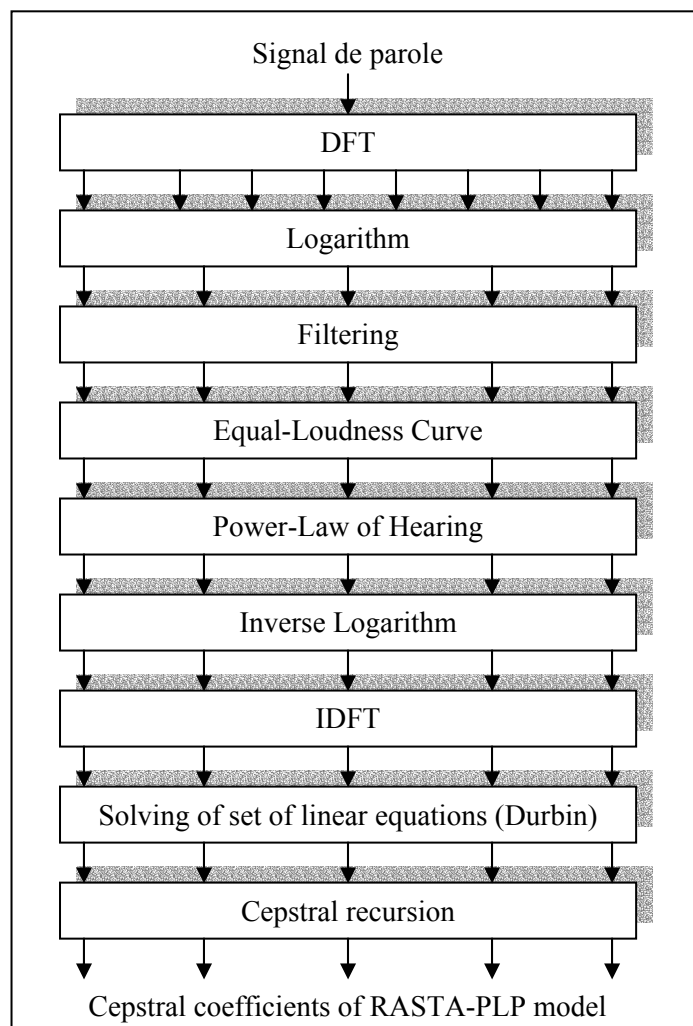


Figure 1.13 : La méthode RASTA-PLP

En effet, supposons que la parole soit corrompue par un bruit convolutif (exemple d'un changement des caractéristiques fréquentielles d'un canal de transmission causé par l'utilisation

d'un nouveau microphone). Une telle distorsion apparaît sous la forme d'une constante additive dans le logarithme du spectre de la parole. La plupart des paramètres acoustiques et donc des systèmes de reconnaissance sont affectés par de tels phénomènes. Le filtrage RASTA supprime les composantes spectrales (ou log spectrales) qui varient plus lentement ou plus rapidement que le signal de parole et donc leurs influences sur les systèmes de reconnaissance de la parole.

Différentes études réalisées avec cette méthode [61, 62, 63] ont permis de confirmer les bonnes qualités de cette méthode relativement aux distorsions et ses moindres qualités face aux bruits qualifiés d'additifs, signe de la présence de plusieurs sources sonores dans un même environnement. Pour améliorer encore la méthode PLP, [104] définit la méthode J-RASTA, plus résistante aux bruits additifs que ne l'est la méthode RASTA, par adjonction d'un filtrage passe-bas dans le domaine spectral.

8.2.6. Dérivées première et seconde

Le but final de l'extraction des paramètres est de modéliser la parole, un phénomène très variable. Par exemple, même si elle a de l'importance, la simple valeur de l'énergie n'est pas suffisante pour donner toute l'information portée par ce paramètre. Il est donc souvent nécessaire de recourir, en plus des paramètres initiaux, à des informations sur l'évolution dans le temps de ces paramètres. Pour cela, les dérivées première et seconde sont calculées pour représenter la variation ainsi que l'accélération de chacun des paramètres. Soit $f_k(t)$ le coefficient d'indice k de la trame t , alors le coefficient différentiel $\Delta f_k(t)$ correspondant est calculé sur $2n_{\Delta}+1$ trames d'analyse (typiquement $n_{\Delta}=2$ ou 3) par :

$$\Delta f_k(t) = \frac{\sum_{i=-n_{\Delta}}^{n_{\Delta}} i \cdot f_k(t+i)}{\sum_{i=-n_{\Delta}}^{n_{\Delta}} i^2} \quad (1.7)$$

Ces coefficients, appelés aussi coefficients delta, estiment la pente de la régression linéaire d'un coefficient à un instant donné. Leur utilisation améliore sensiblement les performances des systèmes de reconnaissance. Par exemple, l'usage de ces coefficients delta améliore de 6% le taux de reconnaissance phonétique du système SPHINX sur la base TIMIT [86].

Des coefficients différentiels du second ordre peuvent aussi contribuer à l'amélioration des systèmes de reconnaissance. Ces coefficients, notés par $\Delta\Delta f_k(t)$ sont calculés par régression linéaire des coefficients delta sur $n_{\Delta\Delta}$ trames (typiquement $n_{\Delta\Delta}=1$ ou 2). [86] constate une amélioration faible mais significative du taux de reconnaissance. La figure 1.14 ci-dessous

illustre les variations en fonction du temps d'un coefficient cepstral $MFCC_1$ et de ses deux dérivées.

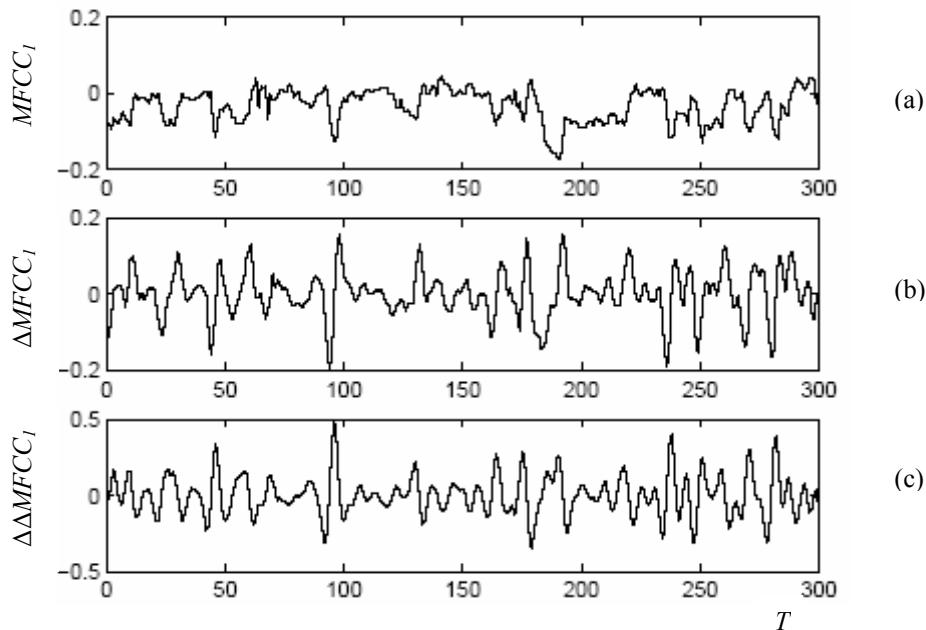


Figure 1.14 : Les variations en fonction du temps (a) du coefficient cepstral $MFCC_1$ (b) de sa première dérivée delta- $MFCC_1$ (avec $n_{\Delta}=2$) et (c) de sa deuxième dérivée delta-delta- $MFCC_1$ (avec $n_{\Delta\Delta}=2$)

8.2.7. Réduction de l'espace de représentation

Même si la robustesse de la représentation obtenue est accrue, l'utilisation des coefficients différentiels dans les systèmes de reconnaissance entraîne un doublement ou même un triplement de la taille des vecteurs de paramètres gérés par les systèmes de reconnaissance. Cette augmentation est coûteuse en ressources de calcul et de mémoire et introduit une redondance de l'information représentée. Il est donc important de ne garder que des paramètres discriminants. La méthode majoritairement utilisée, de nos jours, est l'analyse discriminante linéaire [130], LDA pour *Linear Discriminant Analysis* en anglais. Cette technique s'apparente à l'analyse en composantes principales (ACP). Elle permet l'obtention de paramètres considérés comme discriminants en appliquant une transformation linéaire de l'espace d'entrée de taille n vers un espace de taille réduite q ($q < n$). L'application de cet algorithme maximise la séparation des classes qui sont affectées à chaque vecteur acoustique et ainsi améliore la robustesse de la représentation. [56] ont d'ailleurs montré que l'utilisation d'une telle analyse permet de pallier certaines catégories de bruits.

8.3. Choix d'une représentation du signal

Nous avons brièvement parcouru dans ce paragraphe les méthodes utilisées pour extraire les paramètres acoustiques qui seront fournis au système de reconnaissance. Nous n'avons fait que survoler ces techniques de calcul des coefficients représentatifs du signal vocal. Le choix d'une représentation du signal parmi celles présentées est en définitive guidé par les résultats expérimentaux dans la tâche considérée.

Cette étude soulève deux problèmes principaux. En premier lieu, la grande variété des méthodes et de leurs conditions d'utilisation font qu'il est difficile d'en dégager la meilleure. Elles ont toutes des avantages et des inconvénients, et aucune méthode ne convient parfaitement à la résolution du problème. Le choix doit se faire donc selon le but à atteindre. En second lieu, la difficulté qui provient principalement du bruit qui entache les formes et qui peut rendre les parties importantes à extraire moins nettes. Pour remédier à ce problème, les méthodes d'extractions de caractéristiques sont accompagnées de méthodes de prétraitement pour filtrer le bruit et préparer les données pour le traitement.

9. Approches de reconnaissance vocale

On distingue usuellement en reconnaissance de la parole deux approches, l'approche analytique et l'approche globale. La première cherche à traiter la parole en décomposant le problème, le plus souvent en procédant à un décodage acoustico-phonétique exploité par des modules de niveau linguistique. La seconde consiste à identifier globalement un mot ou une phrase en les comparant avec des références enregistrées.

9.1. Approche analytique

L'approche analytique cherche à résoudre le problème de la parole en isolant des unités acoustiques courtes en procédant à une segmentation en entités élémentaires de base étiquetées ou identifiées, comme les phonèmes, les syllabes...etc (l'étiquetage étant une identification partielle) [24]. Les systèmes analytiques ont l'avantage de ne pas avoir lieu à se préoccuper de la taille du vocabulaire et de s'adapter facilement à tout nouveau locuteur, par contre, leur mise au point est beaucoup plus délicate.

Pour bien mener une tâche de reconnaissance, il est nécessaire d'intégrer dans un même système un ensemble de niveaux supérieurs de reconnaissance: niveau lexical, syntaxique (contraintes liées aux règles de la grammaire), sémantique (le sens des mots), et pragmatique (la cohérence de la phrase, compte tenu de l'application).

Les principaux obstacles de la reconnaissance analytique résultent de la reconnaissance très limitée que l'on a de différents niveaux notamment pour les niveaux phonétiques et sémantiques, mais aussi de les intégrer dans un même système. Les méthodes globales, développées pour la reconnaissance de mots isolés, ne font pas d'hypothèses sur la structure phonétique des mots, ce qui évite une erreur pénalisante au début du traitement [57].

9.2. Approche globale

Les méthodes globales s'appliquent aux systèmes pour lesquels l'unité de décision est l'entité lexicale "le mot". L'idée de base de cette approche est de donner au système au moins une image acoustique de chacun des mots qu'il devra identifier par la suite, ces images sont stockées dans une base ou bien dans un ensemble appelé *ensemble des références*. Cette opération est faite lors de la phase d'apprentissage, où chacun des mots est prononcé une ou plusieurs fois. La reconnaissance globale d'un mot ou d'un ensemble de mots (phrase) consiste à comparer globalement des mots isolés avec des références enregistrées. La comparaison est réalisée par l'exploitation de critères de comparaison, comme par exemple *l'alignement temporelle dynamique* (Dynamic Time Warping ou DTW en anglais). S'il y a identité ou forte ressemblance entre les deux modèles, le mot est déclaré reconnu.

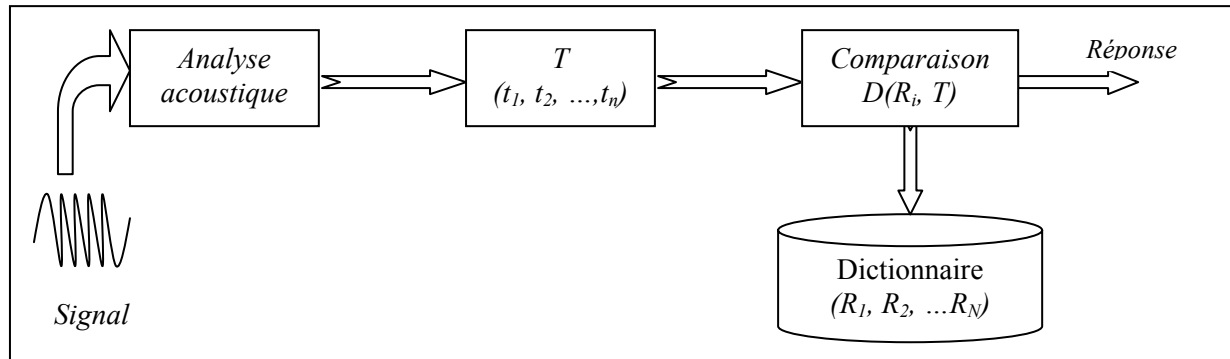


Figure 1.15 : Systèmes de reconnaissance de mots isolés (approche globale)

Cette méthode a pour avantage d'éviter les effets de coarticulation, c'est-à-dire l'influence réciproque des sons à l'intérieur des mots. Cependant, il faut noter que cette approche est limitée aux petits vocabulaires prononcés par un nombre restreint de locuteurs. C'est donc la méthode analytique qui semble utilisable pour les applications à grands vocabulaires où les mots ne sont pas mémorisés dans leur intégralité, mais traités en tant que suite de phonèmes [57].

10. Méthodes de classification et de reconnaissance vocale

La classification ou la reconnaissance dans un SRAP regroupe les deux tâches d'apprentissage et de décision. Elles tentent toutes les deux, à partir de la description en paramètres extraits dans l'étape d'analyse et de paramétrisation (§8.), d'attribuer une forme acoustique à un modèle (ou à une classe) de référence.

Apprentissage

L'apprentissage est une opération d'importance capitale pour un système de reconnaissance, car le soin apporté à cette opération conditionne les performances du système. Cela exige de l'apprentissage de bien définir les classes des formes acoustiques et leurs modèles de manière à bien distinguer les familles homogènes des formes et donc à identifier les nouvelles par rapport à elles, par exemple, en fournissant un bon choix de formes de références ou en donnant au système les bons critères de modélisation [14].

Pendant cette phase, un dictionnaire de référence (représenté par les groupes de mots ou modèles de référence formant des classes) est créé. Le résultat de l'apprentissage est soit la réorganisation ou le renforcement de modèles existants en tenant compte de l'apport de la nouvelle forme acoustique, soit la création d'une nouvelle classe représentant la forme entrée.

En outre, l'apprentissage est dit *supervisé*, si la tâche d'apprentissage est guidée par un superviseur (concepteur) qui indique à la nouvelle forme, la classe qui la contienne, ou apprentissage *non supervisé*, si les classes sont créées automatiquement, sans l'intervention d'un opérateur, à partir d'échantillons de référence et de règles de regroupement.

Décision

La décision est l'ultime étape de la reconnaissance. A partir de la description en paramètres, elle recherche, parmi les modèles d'apprentissage en présence, ceux qui sont les plus "*proches*", et cela en un temps aussi court que possible.

La décision peut conduire à un succès si la réponse est unique (un seul modèle répond à la description de l'image acoustique). Elle peut conduire à une confusion (substitution) si la réponse est multiple (plusieurs modèles correspondent à la description). Enfin, la décision peut conduire à un rejet de la forme si aucun des modèles ne correspond à sa description. Dans les deux premiers cas, la décision peut être accompagnée d'une mesure de vraisemblance appelée aussi *score* ou *taux de reconnaissance* [14].

Dans ce contexte, il existe plusieurs méthodes de classification qui ont contribué à la création des SRAPs regroupées parmi les catégories suivantes :

- a- classification automatique ;
- b- classification statistique ;
- c- classification stochastique ;
- d- classification neuronale.

10.1. La classification automatique

Nous considérons qu'une forme acoustique est représentée par un vecteur de n composantes correspondant à la mesure de n caractéristiques observées sur elle. Classifier un ensemble de vecteurs consiste à regrouper ces vecteurs en classes. Bien entendu, une classe est un ensemble de vecteurs ayant des caractéristiques semblables.

On exige donc de la classification de vérifier les deux propriétés suivantes:

- a- *compacité* : les points représentant une classe donnée sont plus proches entre eux que des points de toutes les autres classes.
- b- *Séparabilité* : les classes sont bornées et il n'y a pas de recouvrement entre elles.

En pratique, ces propriétés sont rarement respectées à cause du bruit et de distorsion des signaux. La décision d'attribuer un vecteur de mesures candidat à une classe est fondée sur la notion de *proximité*. Il en est de même pour la constitution de classes lors de l'apprentissage [57, 18]. Parmi les méthodes de classification automatique, on distingue la programmation dynamique [57]

10.2. La classification statistique

Ces méthodes sont basées sur une formalisation statistique simple issue de la théorie de l'information et qui est aujourd'hui classique pour décomposer le problème de la reconnaissance de la parole. Cette formalisation a été proposée par *F.Jelinek* en 1976 [71].

Les méthodes de classification statistiques consistent à faire correspondre des vecteurs de caractéristiques de longueur fixe à un espace partitionné. Dans ces méthodes, la classification peut être aussi simple qu'un classifieur à distance qui compare les caractéristiques de la forme à reconnaître avec la valeur moyenne des caractéristiques de chaque classe, puis, attribue la forme à la classe ayant les valeurs de caractéristiques les plus proches. Parmi les nombreuses théories et méthodes, on peut citer à titre d'exemples :

- a- décision Bayésienne ;

b- la méthode des k -plus proches voisins (k -ppv).

10.2.1. Décision Bayésienne

Soit un problème caractérisé par un ensemble de N observations, $x = \{x_1, x_2, \dots, x_N\}$ réparties en M classes (C_1, C_2, \dots, C_M) avec leur probabilité *a priori* $P(C_i)$, probabilité d'avoir la classe C_i , calculée à partir des fréquences d'occurrence des exemples de cette classe. Le rôle de la *théorie Bayésienne* est de fournir une fonction de décision qui minimise le coût moyen d'erreur par décision prise. Pour une probabilité d'erreur minimum, il suffit de construire un système qui à chaque x associe la classe dont la densité en x est la plus forte ; elle nécessite donc la connaissance de la probabilité conditionnelle d'appartenance à une classe C_i donnée, soit $P(x/C_i)$ [34].

L'avantage majeur des classifieurs statistiques réside dans leur apprentissage qui peut être automatique.

10.2.2. La méthode des k -plus proches voisins (k -ppv)

Cette méthode de décision est liée à la notion de "*proximité*" (ou *ressemblance*). L'idée de cette technique est simple. La forme acoustique à classer est comparée à d'autres déjà classées, et on lui affecte la classe la plus représentée parmi les k plus proches [14].

Dans le cas particulier $k=1$, c'est la classe de la forme acoustique la plus proche de celle à classer qui lui est affectée. Cette notion de voisinage est quantifiée par une mesure de similarité. La mesure de similarité la plus utilisée est *la distance euclidienne* [34].

10.3. La classification stochastique

L'approche stochastique utilise un modèle pour la comparaison, prenant en compte une plus grande variabilité de la forme. Cette dernière est considérée comme un signal continu observable dans le temps à différents endroits constituant des "*état d'observations*".

Le modèle stochastique décrit ces états à l'aide de probabilités de transitions d'état à état et de probabilités d'observation par état. La comparaison consiste à chercher dans ce graphe le chemin le plus probable correspondant à une suite d'éléments observés dans la chaîne d'entrée.

Ces méthodes sont robustes et fiables du fait de l'existence de bon algorithmes d'apprentissage. La reconnaissance est très rapide car les modèles comprennent généralement peu d'états et le calcul est relativement faible.

10.4. La classification neuronale

Les réseaux de neurones possèdent des propriétés propres à leur style de raisonnement ce qui les rendent très attrayant dans plusieurs domaines. Un des domaines d'application les plus répandus des réseaux de neurones est la reconnaissance des formes [14, 17].

L'utilisation d'un réseau de neurones en classification dépend du codage des sorties. Dans la méthode généralement employée, chaque unité de la couche de sortie représente une classe possible pour les formes en entrée. L'introduction d'une forme inconnue en couche d'entrée du réseau neuronal et après propagation des résultats du calcul élémentaire de chaque neurone vers la couche de sortie, l'élément ayant la plus grande valeur permet alors de choisir à quelle classe affecter la forme en entrée [138]. Un problème de classification peut donc être résolu en trouvant une fonction de transfert qui associe un ensemble de formes de départ à un ensemble de classes d'arrivée. Les réseaux de neurones sont donc des approximateurs universels de fonctions de transfert [105].

Les méthodes neuronales présentent les avantages suivants [105] :

- a- Puissance d'approximation ;
- b- Robustesse pour des tâches difficiles ;
- c- Parallélisme dans le traitement des données.

Cependant, les réseaux de neurones ont des problèmes et des limites propres à eux. En plus de temps d'apprentissage qui est lent, l'inconvénient majeur des méthodes neuronales réside dans la détermination de la topologie du réseau de neurones à utiliser. En effet, la structuration du réseau (nombre de couches cachées, nombre de neurones dans chaque couche cachée, la stratégie de connectivité : locale ou globale, ...etc) se fait avant le processus d'apprentissage, d'une manière aléatoire ou en utilisant des heuristiques.

11. Conclusion

La parole est la faculté naturelle de s'exprimer et de communiquer la pensée, les idées et les émotions par un système de sons articulés ; c'est le moyen de communication privilégié entre les humains qui sont les seuls à utiliser un tel système structuré. Une reconnaissance approfondie du mécanisme de production de la parole chez l'homme contribue à faciliter la compréhension des caractéristiques du signal vocal et d'identifier la forme de la source d'excitation pour bien analyser le signal produit. Ce qui nous a paru nécessaire de présenter

dans ce chapitre un rappel théorique concis, à savoir, une brève description de l'appareil phonatoire qui permettra de mieux comprendre les phénomènes de production de la parole.

Par suite, nous avons présenté le principe des différentes phases nécessaires dans le processus de reconnaissance de la parole. Une analyse détaillée des deux phases qui sont l'extraction des caractéristiques et de classification a été jugée utile pour l'introduction du chapitre 4. Le but recherché par la présentation des méthodes de reconnaissance n'est pas un exposé exhaustif, mais plutôt une situation de la méthode neuronale que nous utilisons par rapport aux autres existantes qui permettent de concevoir des systèmes de reconnaissance qui répondent comme le système naturel quand ils sont simulés par les mêmes données. Cela est effectué sans que le mode de fonctionnement du processus de traitement s'inspire d'une manière quelconque de celui qu'ils sont censées simuler. On parle alors de simulation comportementale.

Mais, il est évident que les systèmes les plus performants sont ceux qui cherchent en plus une simulation fonctionnelle, c'est-à-dire ceux qui résultent d'une utilisation des réseaux de neurones artificiels. Ces derniers seront l'objet du chapitre 2.

Chapitre 2

Les Réseaux de Neurones

Les dogons distinguent deux types de parole, qu'ils nomment parole sèche et parole humide. La parole humide est celle qui fut donnée aux hommes : c'est le son audible. La parole sèche, ou parole première, est l'attribut de l'Esprit Premier. C'est une parole indifférenciée, sans conscience de soi, qui existe en l'homme comme en toute chose. Mais l'homme ne la connaît pas : c'est la pensée divine, dans sa valeur potentielle, et, sur notre plan microscopique, c'est l'inconscient.

M. Griaule et G. Dieterlen
Revue de la Société des Africanistes

1. Introduction

L'informatique est la science du traitement automatique de l'information. Son développement est souvent confondu avec celui des machines de traitement : les ordinateurs. Depuis les débuts (ENIAC 1946) jusqu'à aujourd'hui, les ordinateurs sont devenus de plus en plus puissants.

Cependant, cette augmentation de puissance ne permet pas toujours de résoudre les problèmes d'une application informatique dans un domaine particulier. L'idée s'est donc installée que ce n'était peut être pas tant le matériel que le logiciel qui pêchait par manque de puissance. La construction de logiciels s'appuie sur plusieurs approches. Deux parmi les plus utilisées sont l'approche algorithmique et l'approche basée sur la connaissance.

Une approche algorithmique nécessite l'écriture (avant la transcription dans un quelconque langage de programmation) du processus à suivre pour résoudre le problème. Lorsque le problème est complexe, ce peut être une étape coûteuse ou impossible. D'autre part, les ordinateurs sont des machines complètement logiques (et même binaires) qui suivent à la lettre chacune des instructions du programme. C'est un avantage lorsque tous les cas ont été prévus à l'avance par l'algorithmicien. Ce n'est hélas pas toujours possible.

La seconde approche possible est celle de l'intelligence artificielle (appelée IA par commodité), avec pour applications les plus connues les systèmes experts. Ici, la résolution du problème est confiée à un ensemble de règles données par l'expert humain du domaine. Il n'en demeure pas moins que toutes les règles doivent avoir été exprimées préalablement au traitement, et que le programme demeure binaire dans son exécution. Les cas qui n'ont pas été prévus par l'expert ne seront pas correctement traités. L'introduction de la logique floue ne change pas la nature des limitations d'emploi du programme : l'exécution reste totalement déterministe. En fait, l'approche basée sur la connaissance se limite à des domaines d'application où la modélisation de la connaissance, par exemple sous forme de règles, est possible. Ces domaines sont souvent ceux des sciences dites "exactes" comme l'électronique, la mécanique, la physique, etc, par opposition aux sciences dites "humaines" comme la médecine, la psychologie, la philosophie, etc, où la connaissance est plus empirique. L'IA se révèle donc être principalement un moyen commode de stocker de la connaissance sous forme explicite.

Ces deux approches ne suffisent pas à répondre à tous les problèmes existants. Citons les domaines de la reconnaissance de formes (images ou signaux), du diagnostic, de la traduction automatique, de la compréhension du langage, depuis longtemps explorés à l'aide des approches algorithmiques et à base de connaissances, qui n'ont pas rencontré le succès escompté. Pourtant,

des êtres vivants relativement simples sont capables de réaliser certaines de ces opérations sans difficulté.

Une troisième approche au traitement automatique de l'information semble donc s'offrir à nous, où l'on cherche à s'inspirer du traitement de l'information effectué par le cerveau. L'hypothèse principale, à la base de l'essor des Réseaux de Neurones Artificiels (RNA), est que le comportement intelligent est sous-tendu par un ensemble de mécanismes mentaux. Ces mécanismes étant basés sur des processus neurophysiologiques, nous supposons donc que la structure du système nerveux central est à la base du développement d'un comportement intelligent.

La figure 2.1 reprend l'hypothèse proposée par de nombreux biologistes : pour recréer le comportement intelligent du cerveau, il faut s'appuyer sur son architecture, en fait, tenter de l'imiter.

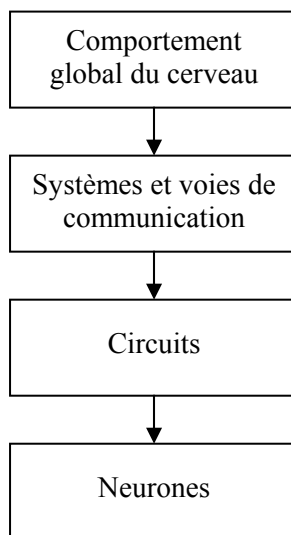


Figure 2.1 : Hypothèse biologique de génération d'un comportement intelligent

2. Neurone biologique

Les réseaux connexionnistes sont des assemblages fortement connectés d'unités de calcul, les *neurones formels*. Ces derniers ont pour origine un modèle du *neurone biologique*, dont ils ne retiennent d'ailleurs qu'une vision fort simplifiée (voir figure 2.2). Un neurone biologique est constitué de trois parties distinctes qui assurent : une collecte d'information, l'intégration de cette information et la restitution de l'information vers d'autres cellules.

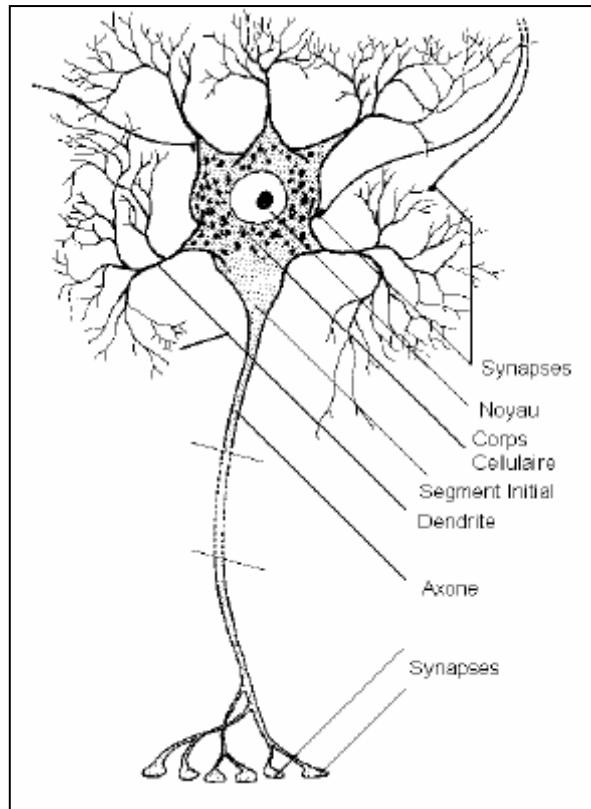


Figure 2.2 : Exemple de neurone biologique

La collecte de l'information est effectuée par les dendrites du neurone qui réceptionnent l'information des unités afférentes par l'intermédiaire des connexions synaptiques. Cette information est alors acheminée, grâce à un processus électro-chimique, vers la cellule elle-même qui intègre cette information au sein de son noyau, également appelé soma où se déroulent les activités propres à sa vie cellulaire. Cette information, une fois traitée, est répercutée en sortie de la cellule vers l'axone qui propage cette information vers d'autres cellules via les axones terminaux et les connexions synaptiques, structures spécialisées dans la communication avec les autres neurones. Cette communication entre cellules nerveuses s'effectue via des impulsions nerveuses. Les impulsions sont générées à l'extrémité somatique de l'axone et vont vers les terminaisons axonales. Là, elles affecteront tous les neurones reliés au neurone générateur, par l'intermédiaire de jonctions appelées synapses entre les terminaisons axonales et les autres cellules (voir figure 2.3) [77].

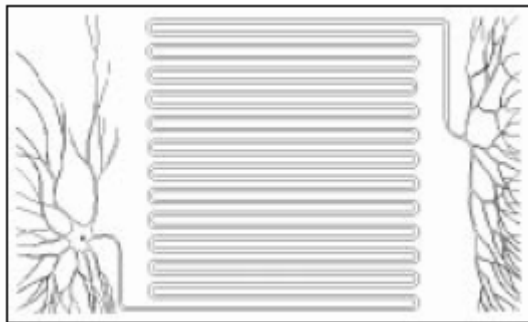


Figure 2.3 : Schématisation d'un neurone. À gauche, les dendrites et le corps de la cellule ; au centre, l'axone ; à droite, les axones terminaux.

Après ces premières études descriptives du neurone vinrent les premiers modèles de neurone formel. En 1943, McCulloch et Pitts proposèrent un modèle mathématique de neurone qui s'avéra un point de départ des recherches en connexionnisme [96, 54, 30, 98].

3. Neurone formel

Cet héritage de la neurobiologie forme une composante importante de l'étude des réseaux connexionnistes, et le souci de maintenir une certaine correspondance avec le système nerveux humain (figure 2.4) a animé et continue à animer une part importante des recherches dans ce domaine. Malgré cet héritage, l'essentiel des travaux d'aujourd'hui ont pour objet les réseaux de neurones formels et non son corrélat neurobiologique. Vu comme des systèmes de calcul, les réseaux de neurones possèdent plusieurs propriétés qui les rendent intéressants d'un point de vue théorique, et fort utile en pratique.

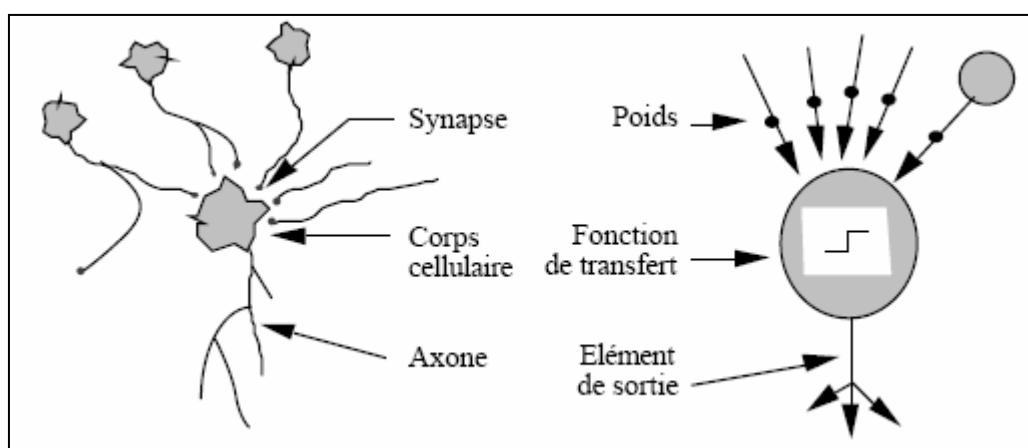


Figure 2.4 : Mise en correspondance neurone biologique / neurone formel

3.1. Structure

Un réseau connexionniste est constitué par un graphe orienté et pondéré. Les noeuds de ce graphe sont des automates simples nommés *neurones formels* ou tout simplement *unités du réseau* (figure 2.5). Les unités sont dotées d'un *état interne*, que l'on appelle *état d'activation*. Les unités peuvent propager leur état d'activation aux autres unités du graphe en passant par des arcs pondérés appelés *connexions*, *liens synaptiques* ou bien *poids synaptiques*. La règle qui détermine l'activation d'un neurone en fonction de l'influence venue de ses entrées et de leurs poids respectifs s'appelle *règle d'activation* ou *fonction d'activation*. Les *changements* apportés aux valeurs *des poids synaptiques* ou dans la structure d'interconnexion des unités du réseau sont responsables des changements de comportement d'activation de ce réseau, ce qui lui permet *d'apprendre* un nouveau comportement. Ainsi, le réseau est capable d'établir des associations entrée-sortie (stimulus et réponse) afin de bien résoudre un problème. La méthode utilisée pour modifier le comportement d'un réseau s'appelle *règle d'apprentissage*.

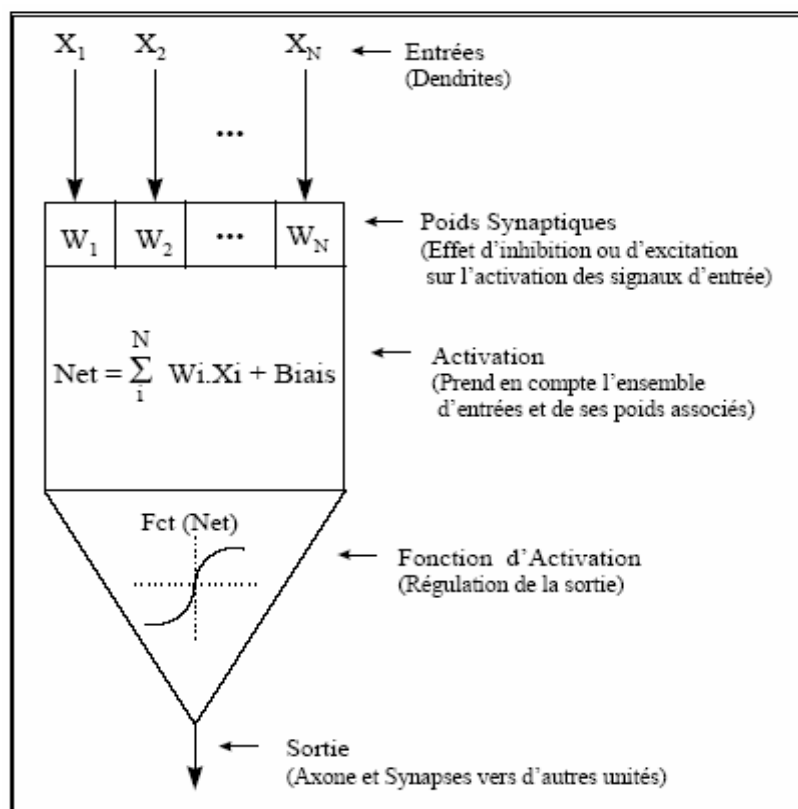


Figure 2.5 : Exemple de neurone formel

3.2. Comportement

On distingue deux phases. La première est habituellement le calcul de la somme pondérée des entrées (Net) selon l'expression suivante :

$$\text{Net} = \sum_i^N W_i \cdot X_i \quad (2.1)$$

A partir de cette valeur, une fonction de transfert calcule la valeur de l'état du neurone. C'est cette valeur qui sera transmise aux neurones avals. La fonction de transfert se doit être non linéaire, mais cette non linéarité peut être exprimée de différentes façons. Nous présentons ci-dessous un ensemble de telle fonctions non linéaires utilisées dans le domaine du connexionnisme.

La figure 2.6 présente des fonctions binaires à seuil qui correspondent à un mécanisme tout ou rien. Parmi celles-ci se trouve la fonction utilisée par McCulloch et Pitts.

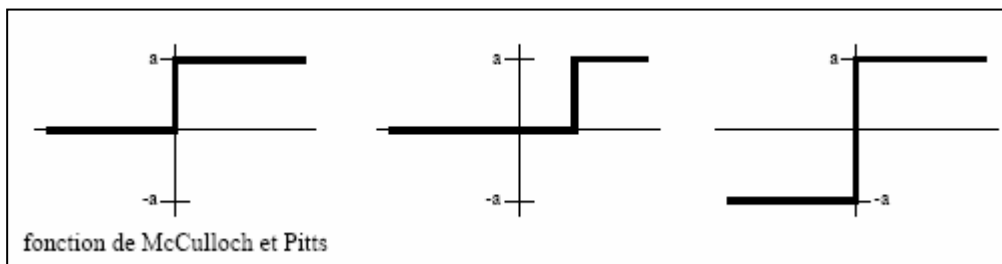


Figure 2.6 : Exemple de fonctions binaires à seuil (d'après [21])

La figure 2.7 présente elle une généralisation de ces fonctions : les fonctions à saturation ou linéaires par morceaux à seuil.

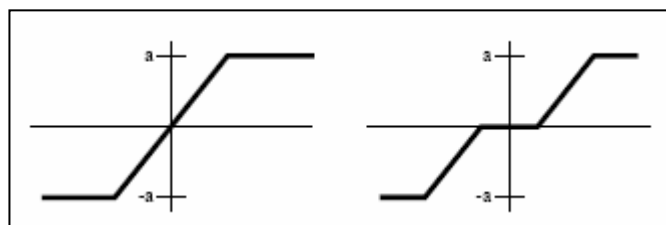


Figure 2.7 : Exemple de fonctions à saturation (d'après [21])

La figure 2.8 présente, enfin, les fonctions non linéaires dérivables qui sont actuellement utilisées puisque la dérivabilité de ce type de fonctions est aujourd'hui une condition nécessaire au processus d'apprentissage.

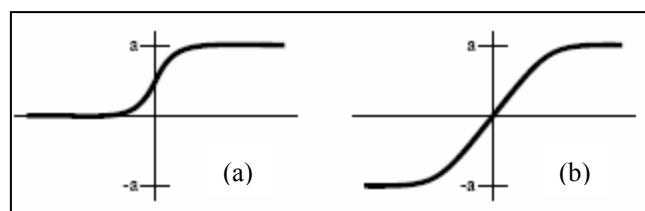


Figure 2.8 : Exemple de fonctions non linéaires dérivables
(a) fonction tangentielle (b) fonction sigmoïdale (d'après [21])

(a) fonction tangentielle : $Fct(Net) = \tanh(Net)$; (2.2)

(b) fonction sigmoïdale : $Fct(Net) = \frac{1}{1+e^{-Net}}$. (2.3)

Nous constatons que les équations décrivant le comportement des neurones artificiels n'introduisent pas la notion de temps. En effet, et c'est le cas pour la plupart des modèles actuels de réseaux de neurones, nous avons affaire à des modèles à temps discret, synchrone, dont le comportement des composants ne varie pas dans le temps [5].

4. Classification et propriétés

Comme nous venons de le voir, un neurone réalise simplement une fonction non linéaire, paramétrée, de ses variables d'entrée. L'intérêt des neurones réside dans les propriétés qui résultent de leur association en réseaux, c'est-à-dire de la *composition* des fonctions non linéaires réalisées par chacun des neurones.

La grande quantité de modèles connexionnistes existants [121, 73] nous oblige à tout d'abord les classer selon leurs différentes propriétés, afin de mieux comprendre les avantages et les inconvénients du choix d'un modèle plutôt que l'autre. Il n'existe pas une seule façon de classer les modèles existants, mais plusieurs, selon les attributs choisis, tels que : l'architecture d'interconnexion des unités du réseau, le type d'apprentissage, la forme pour traiter et représenter les données, etc [42].

4.1. Type d'Architecture du Réseau

Les connexions entre les neurones qui composent le réseau décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité. On distingue deux types de réseaux de neurones : les réseaux non bouclés et les réseaux bouclés.

4.1.1. Les réseaux de neurones non bouclés

Un réseau de neurones non bouclé réalise une (ou plusieurs) fonctions de ses entrées, par composition des fonctions réalisées par chacun des neurones. Il est donc représenté graphiquement par un ensemble de neurones « connectés » entre eux, l'information circulant des entrées vers les sorties sans « retour en arrière » : si l'on représente le réseau comme un graphe dont les noeuds sont les neurones et les arêtes les « connexions » entre ceux-ci, le graphe d'un réseau non bouclé est *acyclique* : si l'on se déplace dans le réseau, à partir d'un neurone quelconque, en suivant les connexions, on ne peut pas revenir au même neurone de départ. La

représentation de la topologie d'un réseau par un graphe est très utile, notamment pour les réseaux bouclés, comme nous le verrons dans le paragraphe suivant. Les neurones qui effectuent le dernier calcul de la composition de fonctions sont les *neurones de sortie* ; ceux qui effectuent des calculs intermédiaires sont les *neurones cachés* [36]. Les architectures les plus importantes dans ce type de réseau sont :

- *Réseaux à une seule couche* : les unités sont toutes sur le même niveau. Dans ce type d'architecture, les unités sont connectées directement aux entrées et sont aussi les sorties du réseau. Les réseaux à une seule couche ont normalement des connexions latérales (entre les neurones d'une même couche). Comme un exemple de ce type d'architecture, on cite les réseaux du type "Kohonen Feature Map".

- *Réseaux à couches unidirectionnels* : les unités sont organisées en plusieurs niveaux bien définis, que l'on appelle des couches. Chaque unité d'une couche reçoit ses entrées à partir de la couche précédente et envoie ses sorties vers la couche suivante (*feed-forward nets*). La figure 2.9 montre un exemple de réseaux à trois couches. Cette architecture à trois couches (entrée, couche cachée et sortie) est très utilisée dans la pratique. Le modèle du PMC [73, 139] se compose en général d'une architecture de ce type, c'est-à-dire avec une seule couche cachée (*hidden layer*), mais rien n'empêche d'avoir plus d'une seule couche cachée dans ce modèle.

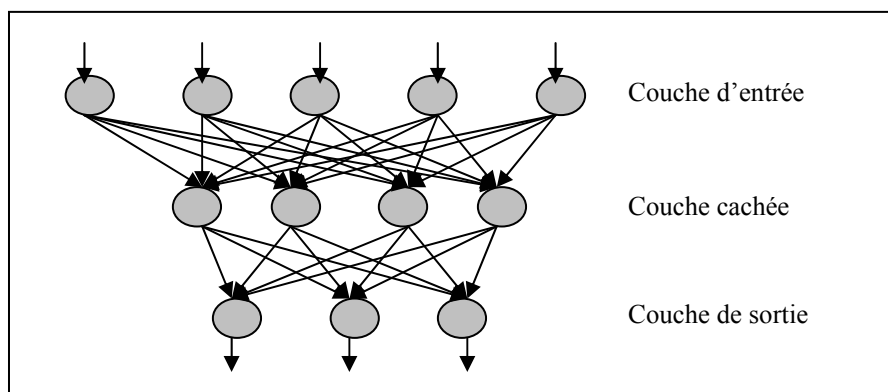


Figure 2.9 : Définition des couches d'un réseau multicouche.

Un autre type d'interconnexions dans les réseaux à couches sont les raccourcis (*short-cuts*) qui permettent de lier des unités en passant à travers des niveaux. Ainsi, on peut 'sauter' d'une couche à l'autre avec les raccourcis, à condition de ne pas créer une boucle (voir figure 2.10).

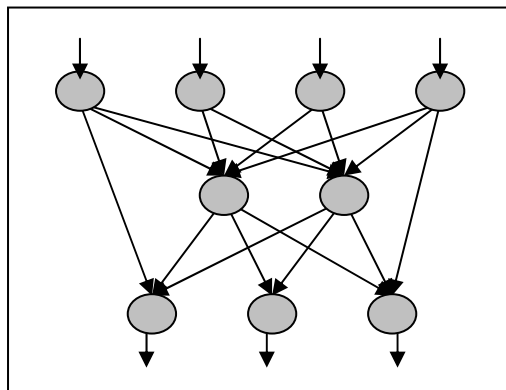


Figure 2.10 : Réseau avec raccourcis

Les réseaux à connexions locales constituent ainsi une autre structure multicouche, mais qui à l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale (figure 2.11). Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.

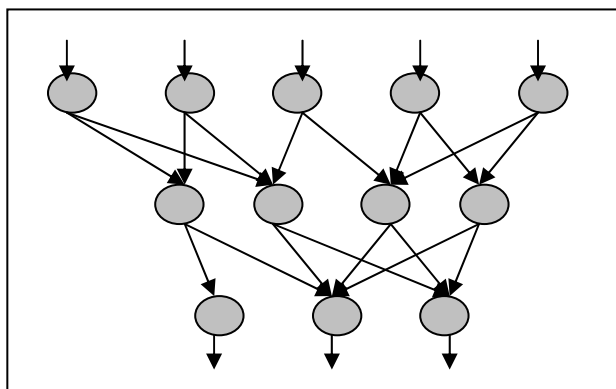


Figure 2.11 : Réseau à connexions locales

Notons que le temps ne joue aucun rôle fonctionnel dans un réseau de neurones non bouclé : si les entrées sont constantes, les sorties le sont également. Le temps nécessaire pour le calcul de la fonction réalisée par chaque neurone est négligeable et, fonctionnellement, on peut considérer ce calcul comme instantané. Pour cette raison, les réseaux non bouclés sont souvent appelés « réseaux statiques », par opposition aux réseaux bouclés ou « dynamiques » que nous introduirons plus bas [36].

4.1.2. Les réseaux de neurones bouclés

Les réseaux bouclés, appelés aussi réseaux récurrents [133, 73, 83] peuvent avoir une ou plusieurs couches, mais leur particularité est la présence d'interconnexions depuis la sortie d'une unité vers une autre unité de la même couche ou d'une couche inférieure (voir figure 2.12). Ce

type d'interconnexions permet de modéliser des aspects temporeux et des comportements dynamiques, où la sortie d'une unité dépend de son état antérieur. Les boucles internes rendent ce type de réseaux instable, ce qui nous oblige à utiliser des algorithmes plus spécifiques (et usuellement plus complexes) pour l'apprentissage.

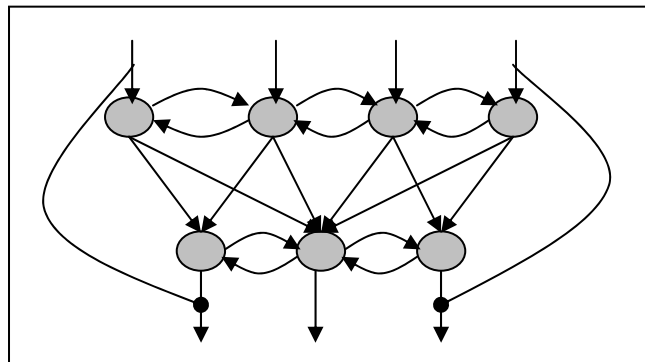


Figure 2.12 : Réseau à connexions récurrentes

Un type particulier de réseau récurrent sont les réseaux totalement connectés, c'est la structure la plus générale. Chaque neurone est connecté à tous les neurones du réseau (et à lui-même), comme représenté dans la figure 2.13.

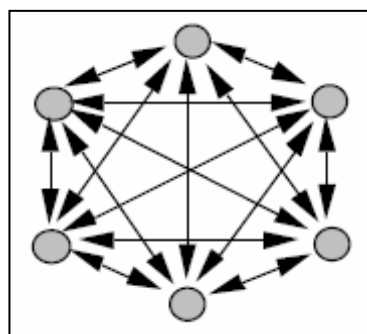


Figure 2.13 : Réseau à connexions complète

L'architecture d'un réseau peut être aussi classée selon son évolution au cours de l'apprentissage et de son utilisation. En fonction de ce critère, les réseaux sont divisés dans les groupes suivants :

- *Réseaux statique* : le réseau a sa structure définie avant l'apprentissage. La quantité de neurones ainsi que sa structure d'interconnexions ne changent pas. Les seuls changements subis par le réseau se situent au niveau des poids synaptiques, modifiés au cours de l'apprentissage. Ce type de modèle pose une difficulté : la détermination du bon nombre de neurones et interconnexions à utiliser. Un réseau avec peu d'unités et d'interconnexions risque de ne pas pouvoir apprendre à résoudre correctement un problème, et un réseau avec trop d'unités peut

avoir des problèmes de convergence, de complexité et de généralisation [72, 43]. Il n'existe pas de méthode optimale qui permet de spécifier automatiquement le nombre exact d'unités à employer pour bien résoudre un problème donné. Les réseaux PMC à Rétro-Propagation classiques sont des réseaux statiques.

- *Réseaux dynamique* : les réseaux ayant une structure dynamique sont des réseaux dont le nombre d'unités et d'interconnexions peut varier. Ils sont aussi appelés réseaux ontogéniques [41]. Les changements dans la structure du réseau peuvent être de type *génératif (incrémental)* ou de type *destructif (élagage)*. Le choix entre ces deux méthodes est controversé : faut-il commencer petit et agrandir ou faut-il commencer grand et réduire ensuite ? [39]. D'un point de vue calculatoire, commencer petit et ajouter des unités et des interconnexions au fur et à mesure de l'apprentissage est plus performant que faire le travail d'apprentissage sur un grand réseau pour en détruire certaines parties ensuite. Une seule chose semble être acceptée par la plupart des chercheurs : les réseaux ontogéniques sont une des meilleures méthodes dont on dispose pour choisir une bonne architecture pour un réseau et pour bien résoudre un problème donné.

4.2. Apprentissage

On appelle « apprentissage » des réseaux de neurones la procédure qui consiste à estimer les paramètres des neurones du réseau, afin que celui-ci remplisse au mieux la tâche qui lui est affectée. C'est une phase du développement durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel en général à deux types d'exemples de comportement :

- Un premier pour le calcul des poids synaptiques ;
- Un deuxième dit de test ou de généralisation, différent du précédent, pour évaluer la qualité de l'apprentissage, et la capacité du réseau à traiter des données inconnues.

Dans le cadre de cette définition, on peut distinguer deux types d'apprentissage qui sont :

4.2.1. Apprentissage supervisé

L'utilisateur dispose d'un comportement de référence précis qu'il désire faire apprendre au réseau. Le réseau est donc capable de mesurer la différence entre son comportement actuel et le comportement de référence, et de corriger ses poids de façon à réduire cette erreur. L'apprentissage supervisé utilise des connaissances empiriques, habituellement représentées par des ensembles d'exemples étiquetés par la classe à laquelle ils appartiennent.

4.2.2. Apprentissage non supervisé

Appelé aussi auto-organisation, ce type d'apprentissage modifie les poids du réseau en fonction de critères internes comme la coactivation des neurones. Les comportements résultants de ces apprentissages sont en général comparables à des techniques d'analyse de données.

5. Le modèle du Perceptron Multi-Couches

Les Perceptrons Multi-Couches (PMC) sont les réseaux à la base des méthodes connexionnistes. Ils sont, en effet, les plus employés et les plus étudiés. Deux abréviations anglaises sont utilisées dans la littérature pour les nommer : *MLP* pour *Multi Layer Perceptrons* et, de manière un peu abusive, *ANN*, *Artificial Neural Networks*. Nous allons faire ici une brève description du PMC et des processus d'activation et d'apprentissage couramment utilisés avec ce type de modèle.

5.1. Architecture

Comme ses prédécesseurs à deux couches, le PMC (voir figure 2.14) se structure en couches où l'activité de la première (couche d'entrée), est fixée à l'entrée du système et celle de la dernière (la couche de sortie) est interprétée comme la réponse du réseau. De plus, une ou plusieurs couches intermédiaires (dites couches cachées) s'interposent entre l'entrée et la sortie. La connectivité de ce réseau est restreinte : un neurone d'une couche inférieure ne peut être relié qu'à des neurones des couches suivantes [73].

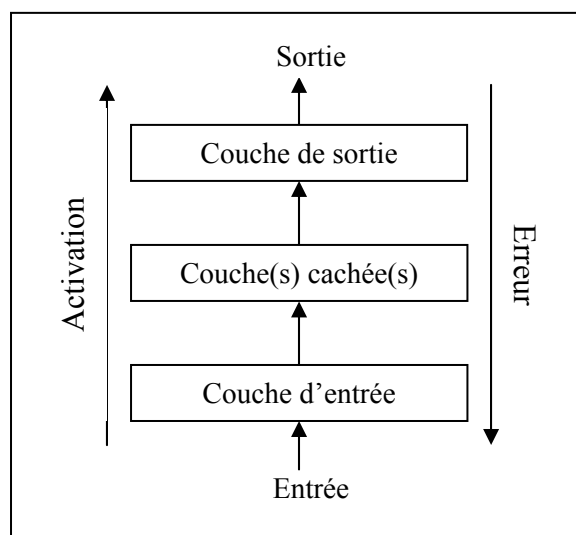


Figure 2.14 : Un Perceptron Multi-Couches

Le calcul de l'activation dans le réseau s'effectue en propageant l'activation initiale de la couche d'entrée jusqu'à la couche de sortie. L'erreur, elle, est calculée dans le sens inverse. Puisque le réseau ne possède pas de boucles, ces deux calculs peuvent être réalisés chacun en une passe unique dans le réseau, et ne requièrent donc qu'un temps proportionnel au nombre de liens synaptiques.

5.2. Apprentissage

Les réseaux PMC utilisent une méthode d'apprentissage de type supervisé. L'apprentissage est réalisé par un algorithme de minimisation sur l'erreur estimée à partir des sorties du réseau. Cet algorithme réalise une descente du gradient de la surface d'erreur (voir figure 2.15). L'emploi de cette méthode permet d'entraîner les unités de sortie du réseau, où l'erreur peut être obtenu directement par simple différence entre la valeur de sortie désirée et la valeur obtenue par le réseau ; mais il permet aussi d'entraîner les unités cachées grâce à une technique de propagation en arrière de l'erreur à travers les couches du réseau (figure 2.14), d'où le nom de Rétro-Propagation (*Back-Propagation Learning*), c'est la méthode la plus utilisée aujourd'hui dans les algorithmes d'apprentissage des réseaux multi-couches.

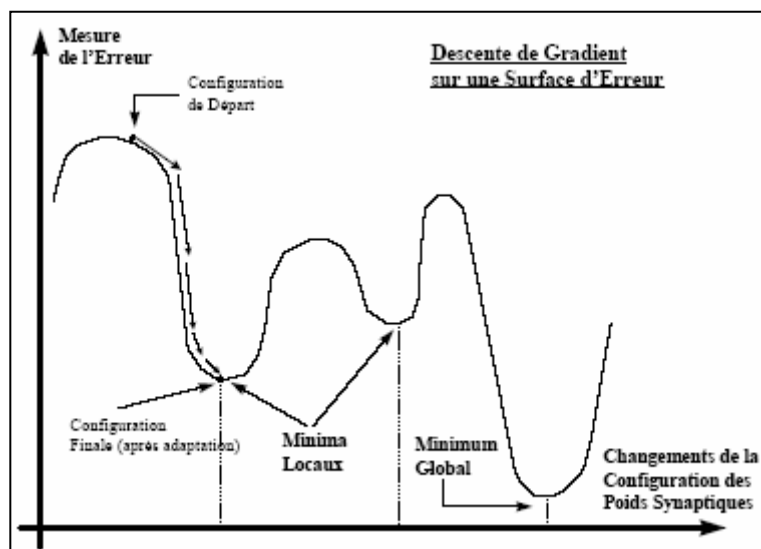


Figure 2.15 : Descente de gradient sur une surface d'erreur et minima locaux

Le principe de cette méthode est basé sur la minimisation de l'erreur quadratique E calculée en fonction des n sorties désirées yd_i et des n sorties effectivement données y_i par le réseau :

$$E = \sum_{i=1}^n (y_i - yd_i)^2. \quad (2.4)$$

Minimiser cette énergie revient alors à modifier les poids des connexions de la manière suivante :

$$\Delta W_{ij} = -\alpha \cdot \delta_i \cdot X_j \quad (2.5)$$

Avec :

α : valeur constante qui sert à contrôler la vitesse de convergence de l'apprentissage (pas d'apprentissage – *learning rate*). Cette valeur est comprise dans l'intervalle [0..1].

δ_i : représente l'erreur de l'unité i .

X_j : valeur de l'entrée j de l'unité en question.

Pour les neurones de la dernière couche :

$$\delta_i = y_i - yd_i \quad (2.6)$$

Pour les neurones des couches internes :

$$\delta_i = Fct'(Net_i) \cdot \sum_k \delta_k \cdot W_{ki} \quad (2.7)$$

avec :

δ_k : représente l'erreur de l'unité k qui se trouve dans la couche suivante après i .

Net_i : valeur cumulée des entrées pondérées par les poids synaptiques de l'unité i .

$Fct'(x)$: dérivée de la fonction d'activation *sigmoïde*.

La rétropropagation de l'erreur est une technique très générale pour calculer de façon efficace le gradient de l'erreur, et son utilité dépasse le cadre des réseaux de neurones. En connexionnisme, cette technique peut être adaptée à des réseaux très différents, par exemple en variant la fonction d'erreur du réseau, les fonctions d'activation de ses neurones et même sa connectivité.

6. Revue des différentes applications des réseaux de neurones

L'intérêt porté aujourd'hui aux réseaux de neurones tient sa justification dans les quelques propriétés fascinantes qu'ils possèdent et qui devraient permettre de dépasser les limitations de l'informatique traditionnelle. Nous rappelons dans ce qui suit les principaux domaines d'application pour lesquels les réseaux de neurones sont les mieux adaptés.

- *La reconnaissance de forme* : dans ce type de problème, les données représentent l'information recueillie par un ou plusieurs capteurs (camera, sonar, micro). Le but ici est de reconnaître le ou les objets perçus par le capteur.

- *La classification* : classer un ensemble d'objets, c'est attribuer à chacun une classe (ou catégorie) parmi plusieurs classes *définies à l'avance*. Cette tâche est appelée « classification » ou « discrimination ». Un algorithme qui réalise automatiquement une classification est appelé *classifieur*. Les applications des classifieurs sont très nombreuses, par exemple dans le domaine de la reconnaissance des formes (chiffres et caractères manuscrits ou imprimés, images, parole, signaux temporels, ...). Plusieurs auteurs considèrent les réseaux de neurones comme des classifieurs sophistiqués par opposition aux méthodes classiques de classification et d'analyse de données.

- *La transformation* : plusieurs applications ont été proposées où le réseau doit apprendre une transformation mathématique entre l'entrée et la sortie. En traitement d'image, par exemple, il existe des méthodes connexionnistes capables de calculer les contours des objets, la correspondance temporelle d'objets en mouvement et la compression de données.

- *La prédiction et le contrôle de processus* : le problème ici consiste à estimer l'état futur d'un processus à partir de son comportement et des variables environnementales attenantes. Un domaine de prédiction où les réseaux ont un succès incontournable est celui de la finance.

Dans ces domaines d'application, les possibilités d'apprentissage automatique, de généralisation et de traitement d'informations incomplètes ou bruitées sont souvent citées comme les avantages prépondérants du connexionnisme.

7. Avantages et limites d'utilisation des réseaux de neurones

Les réseaux connexionnistes, en particulier ceux employés pour la construction de systèmes intelligents, possèdent les avantages suivants :

- *L'apprentissage* : il existe des algorithmes très puissants qui permettent de déterminer les poids des connexions à partir d'exemples. Cet apprentissage qui se fait d'une façon assez simple et permet d'obtenir de bons résultats par rapport aux autres techniques d'apprentissage automatique [30, 72] ;

- *Parallélisme massif* : les réseaux sont composés d'un ensemble d'unités de traitement de l'information qui peuvent opérer en parallèle. Bien que la plupart des implémentations des réseaux connexionnistes soient réalisées sur des simulateurs séquentiels, il est possible de faire des implémentations (logicielles ou matérielles) exploitant la possibilité d'activer simultanément les unités. La plupart des implémentations des réseaux de neurones peuvent être facilement converties d'une version séquentielle vers une version parallèle.

Bien que les réseaux connexionnistes sont capables d'effectuer un grands nombre de tâches utiles, cependant, ils présentent aussi un certain nombre d'inconvénients, tel que :

- *Architecture et paramètres* : il n'existe pas de méthode automatique pour choisir la meilleure architecture possible pour un problème donné. Il est assez difficile de trouver la bonne topologie du réseau ainsi que les bons paramètres de réglage de l'algorithme d'apprentissage. L'évolution du processus d'apprentissage est très influencée par ces deux éléments (l'architecture du réseau et les paramètres de réglage) et dépend beaucoup du type de problème traité. Le simple fait de changer la base d'apprentissage utilisée, peut nous obliger à reconfigurer le réseau en entier ;
- *Initialisation et codage* : les algorithmes d'apprentissage connexionniste sont en général très dépendants de l'état initial du réseau (initialisation aléatoire des poids) et de la configuration de la base d'apprentissage. Un mauvais choix des poids employés pour initialiser le réseau, de la méthode de codage des données, ou même de l'ordre des données, peut bloquer l'apprentissage ou poser des problèmes pour la convergence du réseau vers une bonne solution ;
- *Boîte noire* : les connaissances acquises par le réseau sont codées par l'ensemble des valeurs des poids synaptiques ainsi que par la façon dont les unités sont interconnectées. Il est très difficile pour un être humain de les interpréter directement. Les réseaux connexionnistes sont des boîtes noires, où les connaissances restent enfermés et sont inintelligibles pour l'utilisateur ou pour l'expert. Un réseau ne peut pas expliquer le raisonnement qui l'a amené à une solution spécifique.

8. Conclusion

Le PMC (dont l'algorithme d'apprentissage est la rétropropagation du gradient), semble un réseau adéquat, puissant et prometteur dans le domaine de la reconnaissance de formes. En effet, plusieurs variantes de ce réseau appliquées à la reconnaissance avec tous ses sous-domaines (imagerie, vision, parole) ont été proposées dans la littérature [68, 90]. Cependant, l'algorithme de rétropropagation semble poser certains problèmes : convergence incertaine vers la solution optimale, apprentissage lent, ...etc.

Les algorithmes génétiques (AG), algorithmes itératifs de recherche globale, servent essentiellement à résoudre des problèmes d'optimisation. Ils risquent d'être moins piégés dans des minima locaux que les algorithmes classiques, parce qu'ils explorent en parallèle un ensemble de solutions possibles au problème posé. Ils ont été utilisés pour l'apprentissage du

PMC et ils ont donné des résultats meilleurs que la rétropropagation [70, 140]. Il nous semble alors intéressant de choisir cette alternative pour notre système de reconnaissance. La théorie des algorithmes génétiques est abordée dans le chapitre suivant.

Chapitre 3

Les Algorithmes Génétiques

" J'ai donné le nom de sélection naturelle ou de persistance du plus apte à cette conservation des différences et des variations individuelles favorables à cette élimination des variations nuisibles "

Charles Darwin, L'origine des espèces

1. Introduction

Les systèmes évolutionnistes utilisent une métaphore de l'évolution darwinienne comme méthode de résolution de problèmes. On peut en effet considérer l'évolution des formes de vie comme la réponse au problème de la survie. En se positionnant au niveau d'une espèce, la survie se traduit par des modifications physiques ou comportementales qui entraînent une plus grande efficacité dans la reproduction. En se plaçant au niveau de l'écosystème, la prolifération de formes de vie différentes en compétition produit une grande variété de solutions. Le support de l'évolution naturelle est la chaîne d'ADN. Les systèmes évolutionnaires vont donc recréer un support similaire et la dualité génotype, phénotype utilisée dans la nature.

Ces principes, présentés pour la première fois par Darwin, ont inspiré dans les années 50, les chercheurs en informatique. Mais la faible puissance des machines de l'époque et des connaissances de la génétique naturelle n'a pas permis d'avoir de résultats concluants. Plus tard, dans les années 60 et 70, trois écoles utilisant le principe de l'évolution de façon différente apparurent et qui ont donné naissance à une classe d'algorithmes regroupés sous le nom générique d'Algorithmes Evolutionnaires¹ (EA pour *Evolutionary Algorithms*) (figure 3.1):

- Les algorithmes génétiques (GA pour *Genetic Algorithm*) développés par J. H. Holland [65], ils ont été imaginés comme outils de modélisation de l'adaptation. Ils travaillent dans l'espace des chaînes de bits $[0,1]^n$. Ce sont les plus connus des algorithmes évolutionnaires, et souvent les seules variantes connues des chercheurs des autres disciplines.
- La programmation évolutionnaire (EP pour *Evolutionary Programming*), imaginée par Larry Fogel [47] pour la découverte d'automates à états finis pour l'approximation de séries temporelles. La EP a rapidement travaillé sur des espaces de recherche très variés.
- Les stratégies d'évolution (ES pour *Evolutionary Strategies*), ont été mises au points par deux jeunes ingénieurs H. P. Schwefel et I. Rechenberg travaillant sur des problèmes numériques [123]. Le contexte était l'optimisation paramétrique. Un énorme progrès a été apporté par les techniques adaptatives d'ajustement des paramètres de mutation, et ce sont sans contestation les meilleurs algorithmes pour les problèmes purement numériques.

¹ Les Algorithmes Evolutionnaires sont une classe des algorithmes de recherche (et sous certaine conditions des algorithmes d'optimisation), basés sur le modèle de l'évolution naturelle.

Plus tard, une autre classe d'algorithmes évolutionnaires vit le jour, la programmation génétique (GP pour *Genetic Programming*) de J. Koza [80]. Apparue initialement comme sous-domaine des algorithmes génétiques, et est devenue une branche à part entière. La spécificité de la programmation génétique est l'espace de recherche, fait d'arbres représentant des programmes complets. La GP cherche à atteindre un des vieux rêves des programmeurs, faire écrire un programme par un autre programme.

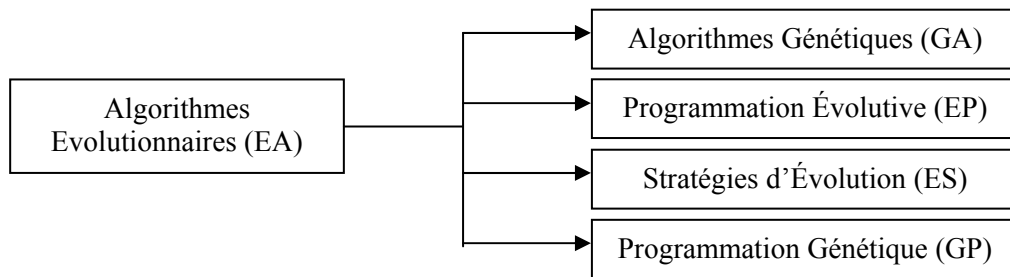


Figure 3.1 : Différentes branches des algorithmes évolutionnaires

Ces différentes classes d'algorithmes évolutionnaires ne diffèrent que sur les détails d'implantation des opérateurs et sur les procédures de sélection et de remplacement de la population. Plus précisément, au niveau de la représentation des individus, des mécanismes d'adaptation, de la fonction d'évaluation, du degré d'utilisation des opérateurs de mutation et de recombinaison et du mécanisme de sélection [8]. Malgré que leur but soit différent à l'origine, ils sont maintenant surtout utilisés pour résoudre des problèmes d'optimisations. Dans la suite, nous verrons plus en détail les algorithmes génétiques (AG) qui sont les plus connus et les plus utilisés des algorithmes évolutionnaires.

2. Principes généraux

Développés dans les années 70 avec le travail de John Holland [65] puis approfondis par Goldberg [52]. Les AGs sont certainement la branche des algorithmes évolutionnaires la plus connue et la plus utilisée. Ils représentent une famille assez riche et très intéressante d'algorithmes d'optimisation stochastique fondés sur les mécanismes de la sélection naturelle et de la génétique [52]. Les AGs s'attachent à simuler le processus de sélection naturelle dans un environnement hostile lié au problème à résoudre [32]. Ils utilisent un vocabulaire similaire à celui de la génétique naturelle. On parlera ainsi d'individu dans une population et bien souvent, l'individu sera résumé par un seul chromosome. Les chromosomes sont eux-mêmes constitués de gènes qui contiennent les caractères héréditaires de l'individu. On retrouvera aussi les principes de sélection, de croisement, de mutation...etc.

La particularité de ces algorithmes est le fait qu'ils font évoluer des populations d'individus codés par une chaîne binaire. Ils utilisent les opérateurs de mutation binaire et de recombinaison de différents types.

Les opérateurs génétiques fonctionnent au niveau génotypique tandis que le mécanisme de sélection opère sur le niveau phénotypique². D'après Charles Darwin, les individus les mieux adaptés à leur environnement sont les plus favorisés : ils ont les plus grandes chances de survie et de reproduction. L'algorithme essaye ainsi de faire évoluer une population initiale afin d'obtenir de meilleurs individus (voir figure 3.2).

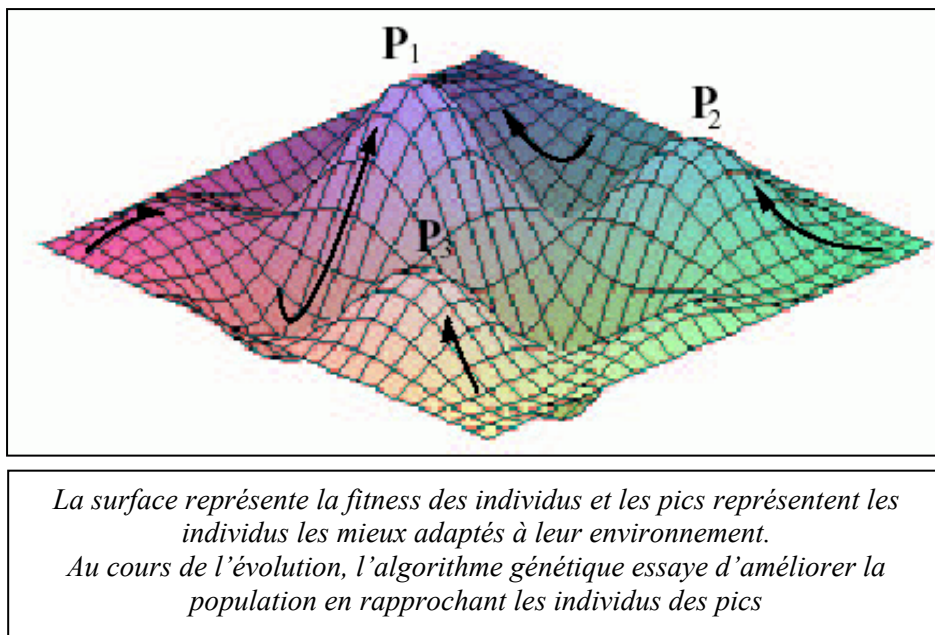


Figure 3.2 : Amélioration de l'adaptation des individus au cours de l'évolution

Les AGs ont été appliqués avec succès à un grand nombre de problèmes réputés difficiles dans des domaines d'applications très divers. On les retrouve aussi bien en théorie des graphes qu'en compression d'images numériques ou encore en programmation automatique et en reconnaissance de formes [75], ou même pour l'apprentissage des réseaux de neurones, ...etc.

Les algorithmes génétiques diffèrent des algorithmes classiques d'optimisation et de recherche essentiellement en quatre points fondamentaux [52] :

- Les AGs utilisent un codage des paramètres et non les paramètres eux-mêmes ;
- Les AGs travaillent sur une population de points, au lieu d'un point unique ;

² Le phénotypique d'un individu est l'ensemble de ses propriétés physiques alors que le génotype est le codage de ces caractéristiques.

- Les AGs n'utilisent que les valeurs de la fonction étudiée, pas sa dérivée ou une autre connaissance auxiliaire ;
- Les AGs utilisent des règles de transition probabilistes, et non déterministes.

De plus, les algorithmes génétiques utilisent deux stratégies importantes pour trouver une solution ou un ensemble de solutions. Ces stratégies sont : l'*exploration* et l'*exploitation*. Elles permettent de trouver le maximum global (solution du problème) du fait qu'elles sont complémentaires [13]. Si l'exploration investigue l'ensemble des solutions de l'espace de recherche, la phase d'exploitation quant à elle se sert de la connaissance trouvée aux solutions précédemment visitées pour aider à trouver de meilleures solutions. La combinaison de ces deux stratégies peut être tout à fait efficace mais la difficulté est de savoir où se trouve la meilleure solution.

3. Présentation des algorithmes génétiques

Un algorithme génétique implémente une version très simplifiée et très schématique des mécanismes de l'évolution biologique. A partir d'un problème qu'on cherche à résoudre, un algorithme génétique est défini, selon [87], par la donnée des éléments de base suivants :

- *Individu, chromosome ou séquence* : les chaînes de système génétique artificielle sont analogues aux chromosomes du système biologique. Un individu représente une solution potentielle du problème qui correspond à une valeur codée de la variable (ou des variables) en considération. Il se compose de gènes indiquant des informations génétiques ;
- *Gène* : les chromosomes se composent de gènes qui peuvent prendre des valeurs différentes (Allèle), appartenant à un alphabet qui dépend du codage adopté au problème à résoudre. La position d'un gène dans un chromosome est identifiée par son locus. Un gène est une caractéristique génétique d'un individu ;
- *Population* : les algorithmes génétiques ne travaillent pas sur un individu unique, mais plutôt sur une population de chromosomes ou de points afin d'effectuer des opérations de recherche sur un domaine de possibilité plus important. En général, la taille de la population est constante, c'est-à-dire que le nombre d'individus est fixe ;
- *Génération* : une génération est une population à un instant donné t , les algorithmes génétiques faisant évoluer les populations, cette évolution est effectuée par des opérateurs de sélection, de croisement et de mutation ;

- *Environnement* : représente l'espace de recherche, caractérisé en terme de performance correspondant à chaque individu possible ;
- *Fonction d'adaptation* : c'est une fonction – positive – que nous cherchons à maximiser car elle mesure l'adaptation d'un individu donné à son environnement. Elle est la clé de voûte des algorithmes génétiques. Il est donc très important de définir une bonne fonction d'adaptation pour le problème traité.

Nature	Algorithme génétique
Chromosome	Chaîne
Gène	Trait ou caractéristique
Allèle	Valeur de la caractéristique
Locus	Position dans la chaîne
Génotype	Structure
Phénotype	Ensemble de paramètre (structure décodée)

Tableau 3.1 : Analogie entre algorithme génétique et nature

On aboutit donc à une structure présentant quatre niveaux d'organisation (figure 3.3):

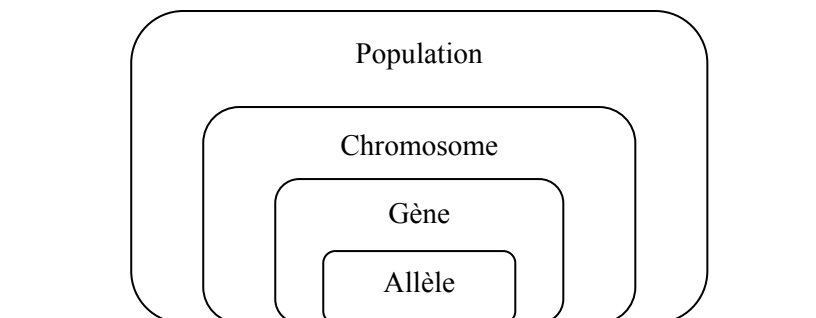


Figure 3.3 : Niveaux d'organisation de l'algorithme génétique

4. Mécanisme d'un algorithme génétique

Il s'agit d'un algorithme itératif de recherche globale dont le but est d'optimiser une fonction définie par l'utilisateur, appelée *fonction d'adaptation* (ou *fitness*, en anglais). Pour cela, les algorithmes génétiques fonctionnent avec une population regroupant un ensemble d'individus appelés *chromosomes* distribués sur l'entièreté de l'espace de recherche. Chaque chromosome est constitué d'un ensemble de *gènes*. Pour chaque individu on attribue une valeur calculée par la fonction d'adaptation (*fitness*). En pratique, à partir d'une population, des

chromosomes sont générés d'une façon aléatoire lors de l'initialisation. Pour définir la taille de la population, Man et al [94] ont mentionné que cette taille varie d'un problème à un autre. Dans chaque cycle d'opérations génétiques, une nouvelle population appelée *génération* est créée à partir des chromosomes de la population courante. Pour cela certains chromosomes appelés '*parents*' sont sélectionnés afin d'élaborer les opérations génétiques. Les gènes de ces parents sont mixés et recombinaés pour la production d'autres chromosomes appelés '*enfants*' constituant la nouvelle génération possédant, dans son ensemble, de meilleures solutions que la précédente.

Les algorithmes génétiques mettent en oeuvre des mécanismes assez simples pour l'évolution vers la solution. Ils ne font que des copies et des échanges de morceaux de chaînes, en plus des opérations d'évaluation et de sélection des éléments d'une population. Les manipulations des chaînes sont faites en utilisant trois opérateurs basiques : la *reproduction*, le *croisement* et la *mutation*. Les étapes de l'algorithme génétique sont répétées durant t cycles, l'arrêt de l'algorithme est fixé d'après un *critère d'arrêt*. On peut avoir plusieurs critères d'arrêt :

- Le nombre de génération fixé initialement a été atteint ;
- La valeur de la fonction d'adaptation a atteint une valeur fixée *a priori* ;
- L'absence d'évolution de la valeur de la fonction d'adaptation des individus d'une population à une autre ;
- Les chromosomes ont atteint un certain degré d'homogénéité.

La figure 3.4 illustre les différentes étapes d'un algorithme génétique simple [52, 94] :

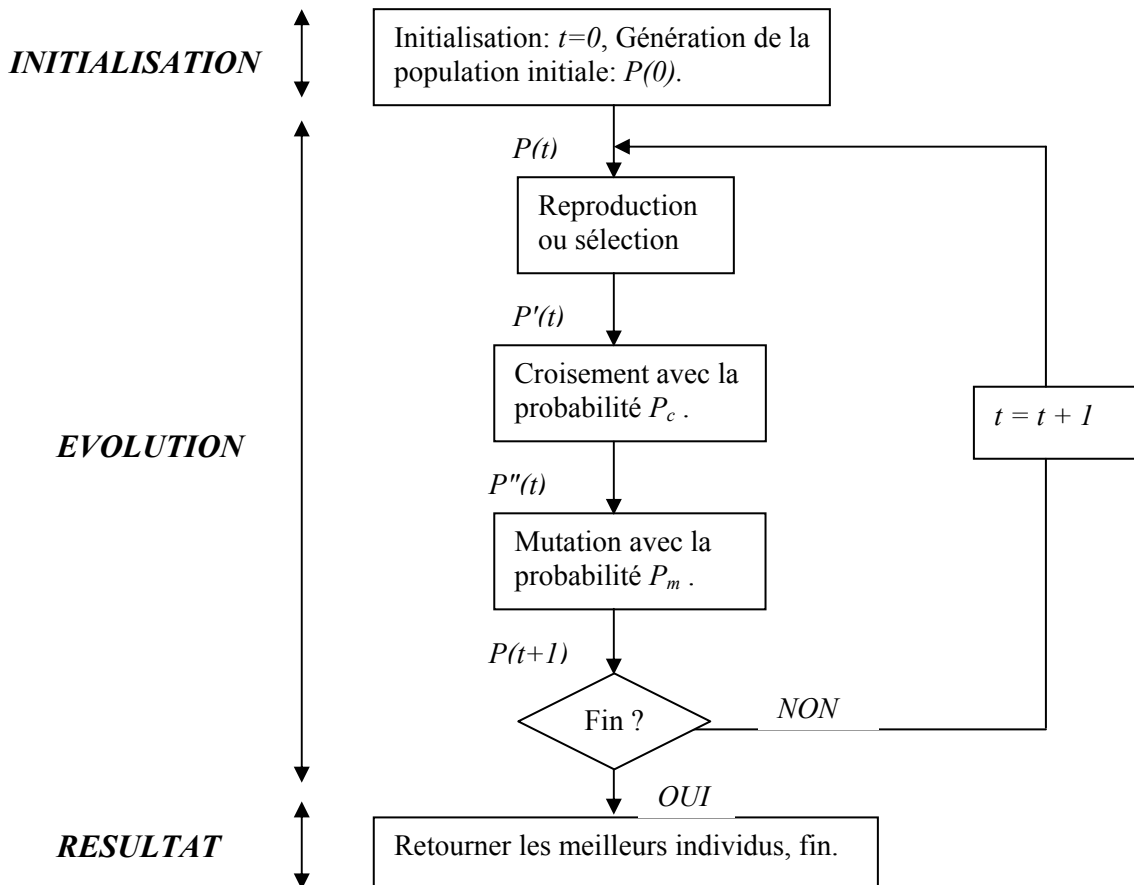


Figure 3.4 : Principe général d'un algorithme génétique simple

De cette structure, on peut conclure une construction d'un AG comme suit :

- Une représentation chromosomique des solutions du problème ;
- Une méthode de création de la population initiale des solutions. Donc une population de N individus est tirée aléatoirement ;
- Une fonction d'évaluation qui permet d'évaluer les solutions plus ou moins "adaptés" (*fitness*);
- Des opérateurs génétiques qui modifient la composition des chromosomes des parents au cours de la reproduction ;
- Les valeurs des paramètres que l'algorithme génétique emploie (taille de la population, probabilité d'application des opérateurs génétiques,...etc) qui peuvent influencer la vitesse de convergence de l'algorithme.

Dans ce qui suit, nous allons détailler les différents mécanismes des éléments d'un algorithme génétique.

5. Codage (représentation) d'un individu

Le premier problème rencontré lors de l'utilisation des algorithmes génétiques est le codage des individus (i.e. élément de l'espace de recherche). Cette représentation est l'un des points fondamentaux des algorithmes génétiques. Le chromosome peut selon différentes façons, contenir l'information concernant la solution qu'il représente.

Dans sa première analyse des algorithmes génétiques, John Holland a proposé la représentation binaire des solutions du problème à résoudre, et souvent, c'est le codage le plus utilisé. Ceci est du, d'après Holland, au fait que la représentation binaire conserve la granularité de l'information nécessaire pour l'algorithme [33]. Dans ce type de codage, le chromosome est simplement une concaténation des codes binaires de toutes les composantes du vecteur de l'espace de recherche qui doit être borné. Le codage binaire lui-même présente plusieurs choix. Prenons, par exemple le cas d'un espace de recherche à éléments entiers dans $\langle 0, n \rangle$. Un premier codage traditionnel consiste à remplacer chaque élément par son expression binaire. On peut également utiliser un codage de Gray³.

<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0:</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1:</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>2:</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>3:</td><td>0</td><td>1</td><td>1</td></tr> </table>	0:	0	0	0	1:	0	0	1	2:	0	1	0	3:	0	1	1	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>4:</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>5:</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>6:</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>7:</td><td>1</td><td>1</td><td>1</td></tr> </table>	4:	1	0	0	5:	1	0	1	6:	1	1	0	7:	1	1	1	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0:</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1:</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>2:</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>3:</td><td>0</td><td>1</td><td>0</td></tr> </table>	0:	0	0	0	1:	0	0	1	2:	0	1	1	3:	0	1	0	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>4:</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>5:</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>6:</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>7:</td><td>1</td><td>1</td><td>1</td></tr> </table>	4:	1	1	0	5:	1	0	0	6:	1	0	1	7:	1	1	1
0:	0	0	0																																																																
1:	0	0	1																																																																
2:	0	1	0																																																																
3:	0	1	1																																																																
4:	1	0	0																																																																
5:	1	0	1																																																																
6:	1	1	0																																																																
7:	1	1	1																																																																
0:	0	0	0																																																																
1:	0	0	1																																																																
2:	0	1	1																																																																
3:	0	1	0																																																																
4:	1	1	0																																																																
5:	1	0	0																																																																
6:	1	0	1																																																																
7:	1	1	1																																																																
<i>Codage classique</i>		<i>Codage de Gray</i>																																																																	

Tableau 3.2 : Codage des entiers

Cependant, des nouvelles versions modifiées [115, 33, 122] des algorithmes génétiques originaux ne se basant plus sur le codage binaire des paramètres à optimiser, mais travaillent directement sur les paramètres eux-mêmes, on parle alors d'algorithmes génétiques codés-réels. Ces versions offrent généralement l'avantage d'être mieux adaptées aux problèmes d'optimisation numériques continus, d'accélérer la recherche et de rendre plus aisé le développement de méthodes hybrides avec d'autres plus classiques.

D'autres auteurs tels que Man et al [94], Janikow et Michalewicz [69], Herrera et al [64], et Davis [32] ont effectué une comparaison entre la représentation binaire et la représentation

³ Le principe d'un codage de Gray est de coder deux éléments consécutifs par deux chaînes de bits qui ne présentent qu'un seul bit de différence.

réelle. Ces auteurs ont trouvé que la représentation réelle donne de meilleurs résultats d'après leur problème à résoudre. Citons quelques exemples de codage non binaire :

- Les algorithmes génétiques *codés-réels* [140, 101] ont été utilisés dans des systèmes de commande et pour l'apprentissage des réseaux de neurones. Le chromosome, dans ce cas est une chaîne de réels.
- Les algorithmes génétiques codés *en règles de production* [53] ont été utilisés dans des applications industrielles. Le chromosome, dans ce cas est une chaîne de règles (chromosome = (Règle A, Règle B, Règle C, ..., Règle P)).

En général, il n'existe pas de consensus sur la représentation idéale des individus. Le choix du codage est laissé à l'utilisateur car l'algorithme n'en a aucune connaissance. Il faut choisir un codage approprié, adapté au problème traité.

6. Génération de la population initiale

Ce mécanisme doit être capable de produire une population d'individus uniformément répartis sur l'espace de recherche, qui servira de base pour les générations futures. Suivant la position de la population initiale dans l'espace d'état, on favorise la recherche de l'optimum. Si l'on n'a aucune connaissance de la position de l'optimum, on génère aléatoirement des individus en faisant des tirages uniformes dans l'espace de recherche. Si par contre, on dispose d'informations *a priori* sur le problème, on indique un sous-espace où l'on est sûr de trouver l'optimum, il faut générer les individus dans ce sous-espace afin d'accélérer la convergence. A partir de cette population initiale, il faut être capable de maintenir la diversité de la population au cours des générations afin d'entretenir le processus d'exploration de l'espace d'état : c'est le rôle des opérateurs génétiques. Mais auparavant, la performance de chacun des individus doit d'abord être évaluée.

Le nombre d'individus d'une population ou la taille de la population constitue un paramètre important pour l'AG qu'il faudra déterminer. La représentation de la population P est:

$$P = (C_1, C_2, \dots, C_i, \dots, C_{taille-pop}) \quad (3.1)$$

où C_i représente le $i^{ème}$ chromosome dans la population et *taille-pop* représente le nombre de chromosomes dans la population.

7. Fonction d'adaptation (*fitness*)

Chaque chromosome apporte une solution potentielle au problème à résoudre. Néanmoins, ces solutions n'ont pas toutes le même degré de pertinence. C'est à la fonction de performance, dite aussi fonction d'adaptation ou *fitness*, de mesurer cette efficacité pour permettre à l'algorithme génétique de faire évoluer la population dans un sens bénéfique pour la recherche de la meilleure solution.

Autrement dit, la fonction d'adaptation doit pouvoir attribuer à chaque individu un indicateur positif représentant sa pertinence pour le problème qu'on cherche à résoudre [33]. La fonction d'évaluation ou d'adaptation (*fitness*) associe donc un coût à chaque chromosome. Et c'est grâce à la *fitness* que l'algorithme évalue l'adaptation des individus à leur environnement et calcule leurs chances de survivre et de se reproduire en favorisant la sélection d'individus dans la direction de l'optimum qui est, *a priori*, inconnue.

Elle constitue donc le critère à base duquel l'individu serait ou pas sélectionné pour être reproduit dans la génération suivante. La qualité de cette fonction conditionne, pour une grande part, l'efficacité d'un algorithme génétique. Il est, par conséquent, important de tenir compte de sa complexité. En effet, dans le cas où la fonction d'adaptation apparaît excessivement complexe, consommant une importante puissance de calcul, il est souhaitable de lui rechercher une approximation plus simple.

Il faut distinguer entre *la fonction objective* et *la fonction d'adaptation*. Dans certains cas, elles peuvent être identiques mais en général la fonction d'adaptation dépend de la fonction objective, laquelle dépend de la nature du problème à résoudre. La solution optimale du problème est obtenue à partir de la fonction d'évaluation du chromosome. Dans le cas d'un problème de minimisation, la solution est associée à la plus petite valeur trouvée de la fonction d'adaptation calculée pour chaque individu de la population. Dans le cas d'une maximisation, alors la valeur la plus grande de la fonction sera prise en compte.

8. Opérateurs génétiques

Trois opérateurs jouent un rôle prépondérant dans la possible réussite d'un algorithme génétique : *l'opérateur de sélection*, *l'opérateur de croisement* et *l'opérateur de mutation*. Si le principe de chacun de ces opérateurs est facilement compréhensible, il est toutefois difficile d'expliquer l'importance isolée de chacun dans la réussite de l'algorithme génétique. Cela

tient pour partie au fait que chacun de ces opérateurs agit selon divers critères qui lui sont propres (valeur sélective des individus, probabilité d'activation de l'opérateur,... etc).

8.1. Opérateur de sélection (ou reproduction)

Cet opérateur est peut-être le plus important puisqu'il permet aux individus d'une population de survivre, de se reproduire ou de mourir, en fonction de leur adaptation en donnant plus de chance aux meilleurs. Ceci suppose alors que l'évaluation des individus a été faite [48]. Cet opérateur détermine donc la capacité de chaque individu à persister dans la population et à se diffuser. En règle générale, la probabilité de survie d'un individu sera directement reliée à sa performance relative au sein de la population. Cela traduit bien l'idée de la sélection naturelle : les gènes les plus performants ont tendance à se diffuser dans la population tandis que ceux qui ont une performance relative plus faible ont tendance à disparaître.

Pour sélectionner les meilleurs individus (chromosomes) dans une population, il existe plusieurs types de sélection, chacune étant adaptée à un type particulier de problèmes. On en cite les plus classiques qui sont: la sélection *ordonnée* (*rank selection*), *roulette biaisée* (*Roulette wheel selection*), la sélection *par tournoi* (*tournament selection*) et la sélection *uniforme* (*uniform selection*).

8.1.1. Sélection ordonnée (Rank selection)

Chaque chromosome C_i d'une population P est évalué par la fonction d'adaptation f_i . Les valeurs de la fonction d'adaptation obtenues pour l'ensemble des chromosomes seront classées dans un ordre croissant ou décroissant. Les meilleurs chromosomes seront donc sélectionnés.

8.1.2. Roulette biaisée (Roulette wheel selection)

Appelée aussi la roue de loterie biaisée. C'est la méthode la plus connue et la plus utilisée dans la littérature [52]. Selon cette méthode, chaque chromosome sera dupliqué dans une nouvelle population proportionnellement à sa valeur d'adaptation. On effectue en quelque sorte, autant de tirages avec remise qu'il y a d'éléments dans la population. Ainsi, la performance d'un individu particulier, C_i , étant f_i , la probabilité avec laquelle il sera réintroduit dans la nouvelle population P de taille *taille-pop* est :

$$P_i = \frac{f_i}{\sum_{j=1}^{taille_pop} f_j} \quad (3.2)$$

où $\sum_{j=1}^{taille_pop} f_j$ représente la somme de toutes les valeurs des fonctions d'adaptation de chaque chromosome C_i de la population P .

Plus la performance d'un individu est élevée par rapport à celles des autres individus de la même population, plus il a une chance d'être reproduit dans la population. Les individus ayant une grande performance relative ont donc plus de chance d'être sélectionnés. On parle alors de *sélection proportionnelle*.

La figure 3.5 représente un exemple de modèle de la roulette. Si l'on considère la somme des performances de la population (le disque entier), la performance de l'individu 1 représente 10% de la somme, l'individu 2, 20%, ...etc. Ce qui équivaut à dire que l'individu 1 a 10% de chance d'être sélectionné, l'individu 2, 20%, ...etc.

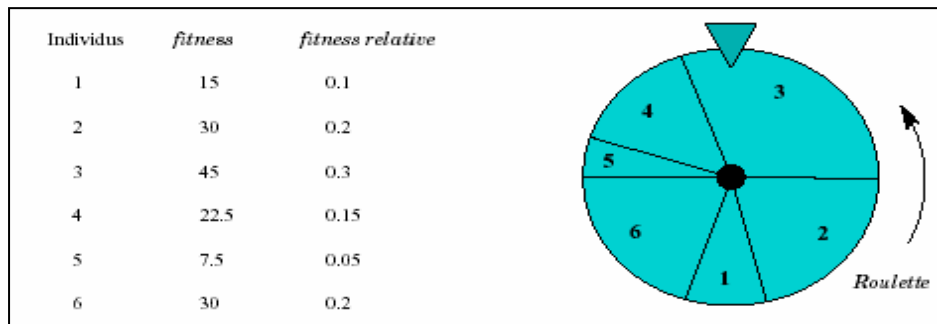


Figure 3.5 : Exemple d'application de la *Roulette Wheel Selection*

8.1.3. Sélection par tournoi (Tournament selection)

Cette technique utilise la méthode de la roulette biaisée pour sélectionner deux individus. On récupère celui dont la valeur de la fonction d'adaptation est la plus grande. Cette méthode choisit toujours une valeur de la fonction d'adaptation plus élevée par rapport à la technique de la roulette biaisée.

8.1.4. Sélection uniforme (Uniform selection)

C'est une technique très simple qui consiste à sélectionner un individu C_i aléatoirement de la population P . La probabilité p_i pour qu'un individu soit sélectionné est définie par :

$$p_i = \frac{1}{taille_pop} \quad (3.3)$$

8.2. Opérateur de croisement (recombinaison ou crossover)

Le croisement est l'opérateur de recherche essentiel d'un algorithme génétique. Il a lieu après la sélection. Il permet l'échange d'informations génétiques entre les individus après sélection, de façon aléatoire en une (ou plusieurs) position(s) de césure (*locus*) [52]. Dans ce cas, deux des individus retenus de la sélection (parents) échangent une ou plusieurs parties de leurs génotypes, selon une probabilité P_c (souvent supérieure à 60%) [38], pour former deux individus différents de ceux d'origine.

Toutefois, un individu sélectionné lors de la reproduction ne subit pas nécessairement l'action d'un croisement. Ce dernier ne s'effectue qu'avec une certaine probabilité. Plus cette probabilité est élevée, plus la population subira de changement.

Il existe différentes manières pour croiser deux génotypes, les trois les plus connues : le croisement "à un point" (one-point), "à deux points" (two-points), et le croisement uniforme (uniform crossover).

8.2.1. Croisement à "1 point" (one-point)

Le croisement "à un point" peut être illustré comme dans la figure 3.6. Il consiste à sélectionner aléatoirement un emplacement (une position) de césure et de permuter les parties droites des deux parents.

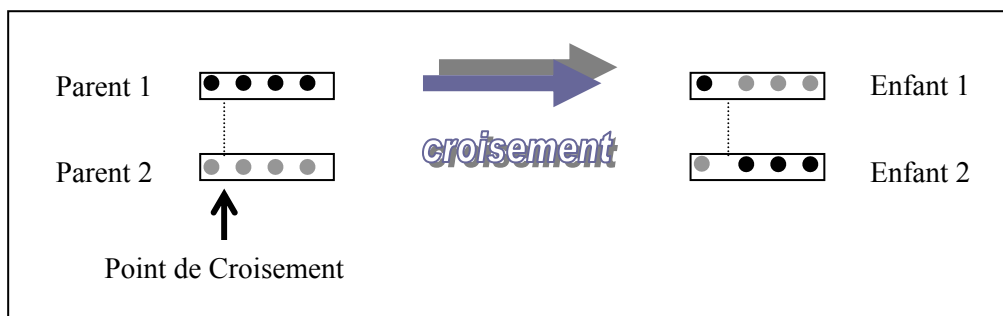


Figure 3.6 : Croisement à 1 point (one point crossover)

Ce mécanisme présente l'inconvénient qu'il privilégie les extrémités des individus. Et, selon le codage choisi, il peut générer des fils qui sont plus ou moins proches de leurs parents. Il existe d'autres techniques qui permettent de gérer le croisement afin de mieux l'adapter au codage et à l'heuristique de recherche. Les plus classiques de ces techniques sont les croisements à 2 points et les *k-point slicing crossover* ($k > 2$).

8.2.2. Croisement à "2 points" (two-points)

Ce croisement diffère du premier, tout simplement, du fait qu'il s'effectue en deux points de coupure, comme le montre la figure 3.7.

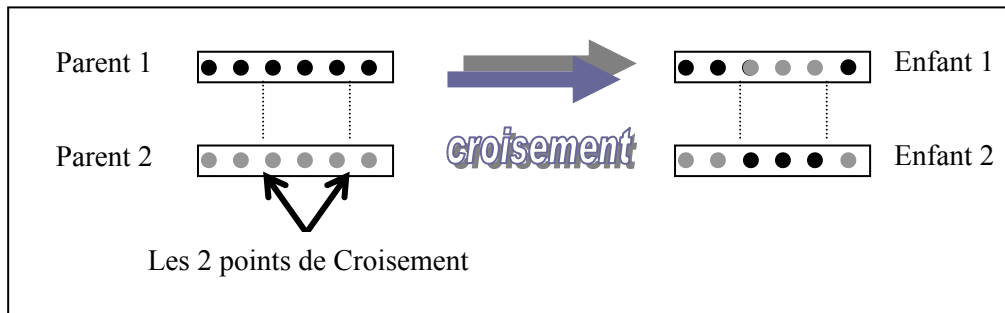


Figure 3.7 : Croisement à 2 points (two points crossover)

Le croisement multi-points (ou k-points pour $k > 2$) est une généralisation du croisement à 2 points. Cette technique s'applique autant pour une codification binaire que réelle des chromosomes. C'est une technique très utilisée dans différentes applications du fait que les résultats obtenus sont satisfaisants [38].

8.2.3. Croisement uniforme (uniform crossover)

Ce croisement a été défini par Syswerda [132]. C'est aussi un croisement multi-points dont le nombre de coupures est indéterminé *a priori*. Cette technique est complètement différente des deux précédentes. Un masque de croisement est généré aléatoirement pour chaque couple d'individus ou pour chaque génération. Les valeurs de ce masque sont binaires. Sa taille est identique à celle du chromosome. Son fonctionnement est illustré par la figure 3.8.

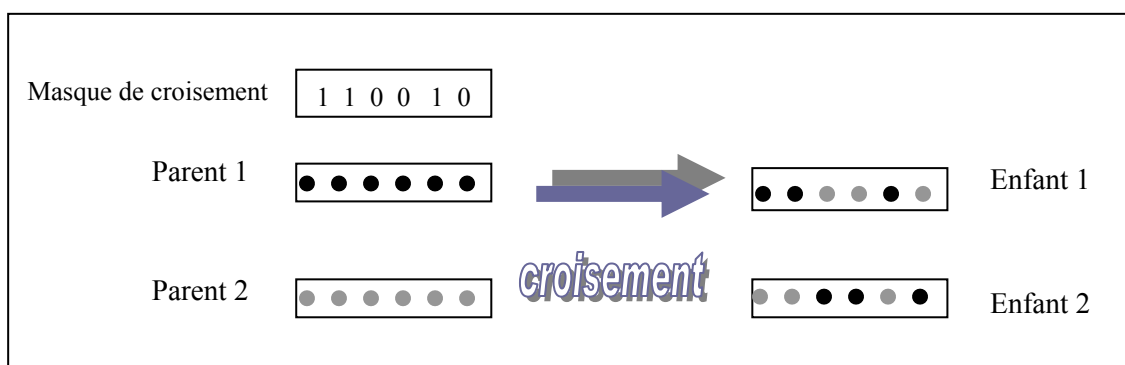


Figure 3.8 : Croisement uniforme (uniform crossover)

Le fonctionnement du croisement uniforme est le suivant : si la valeur du bit du masque est égale à 1 alors la valeur du gène du parent1 est copiée chez l'enfant1 et si la valeur du bit du

masque est égale à 0 alors la valeur du gène du parent2 est transmise à l'enfant1. Les valeurs des gènes de l'enfant2 sont les suivantes : les valeurs des gènes du parents1 lorsque la valeur du bit du masque est égale à 0 et les valeurs des gènes du parents2 lorsque la valeur du bit du masque est égale à 1.

Malgré que cette technique soit différente des deux autres au niveau conceptuel, on remarque qu'il existe une ressemblance entre ces techniques. Le croisement uniforme peut être un cas général des deux techniques.

8.3. Opérateur de mutation

Si c'est la reproduction associée au croisement qui donne aux algorithmes génétiques l'essentiel de leur puissance, quel est donc l'intérêt de l'opérateur de mutation ? Le rôle de la mutation reste très flou en génétique (aussi bien naturelle qu'artificielle). Mais quelle que soit la cause réelle, il apparaît que la mutation joue bien un rôle secondaire dans la mise en œuvre des algorithmes génétiques. Elle est cependant nécessaire parce que, bien que la reproduction et le croisement explorent et recombinent efficacement les notions existantes, ils peuvent parfois devenir trop zélés et perdre de la matière génétique potentiellement utile (des 1 ou des 0 à des endroits précis, dans le cas du codage binaire). Dans les systèmes génétiques artificiels, l'opérateur de mutation protège contre de telles pertes irrécupérables.

La mutation est traditionnellement considérée comme un opérateur intervenant à la marge dans la mesure où sa probabilité P_m est en général assez faible (de l'ordre de 1%). Mais elle confère aux algorithmes génétiques une propriété très importante : l'ergodicité (i.e. tous les points de l'espace de recherche peuvent être atteints). Elle permet donc d'explorer l'espace de recherche en déplaçant, avec une faible probabilité P_m , des individus dans leurs voisinages. Cet opérateur est donc d'une grande importance et il est loin d'être marginal. Il a de fait un double rôle : celui d'effectuer une recherche locale et/ou de sortir d'une trappe (optima locaux).

La mutation consiste à modifier aléatoirement, avec une certaine probabilité P_m , la valeur d'un composant de l'individu. En fonction du codage choisi, on peut définir une technique de mutation adaptée.

Dans le codage binaire, chaque bit $a_i \in \{0, 1\}$ est remplacé selon la probabilité P_m par son inversé $a'_i = 1 - a_i$.

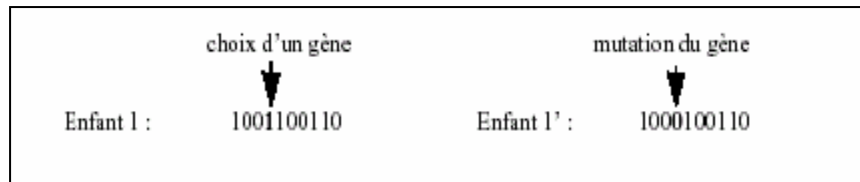


Figure 3.9 : Opération de Mutation

Pour des chaînes de réelles, on imitera ce comportement en ajoutant un bruit, par exemple gaussien, à la valeur du gène choisi aléatoirement.

$$x_i = x_i \pm \text{norm}(\sigma). \quad (3.4)$$

9. Principaux paramètres d'un algorithme génétique

Les opérateurs de l'algorithme génétique sont guidés par un certain nombre de paramètres structurels donnés. La valeur de ces paramètres influence la réussite ou non et la rapidité d'un algorithme génétique. Nous allons maintenant discuter rapidement le rôle de ces paramètres.

- La taille de la population, N , et la longueur du codage de chaque individu, l . Si N est trop grand, le temps de calcul de l'algorithme peut s'avérer très important ; si N est trop petit, il peut converger trop rapidement vers un mauvais chromosome. Cette importance de la taille est essentiellement due à la notion de *parallélisme implicite* : plus N est grand, plus élevé est le nombre de solutions potentielles évaluées en parallèle par l'algorithme génétique.
- La probabilité de croisement, p_c . Elle dépend en général, de la forme de la fonction de performance. Son choix est bien souvent heuristique (tout comme pour p_m). Plus elle est élevée, plus la population subit de changements importants. Les valeurs généralement admises sont comprises entre 0.5 et 0.9.
- La probabilité de mutation, p_m . Ce taux est généralement faible puisqu'un taux élevé risque de conduire à une solution sous-optimale en perturbant celle qui est optimale.
- Il existe plusieurs stratégies qui peuvent être adoptées dans la génération d'une nouvelle population. Dans un cycle de génération, un individu de la population courante est remplacé par un nouvel enfant. Il est fort probable que le meilleur individu de la génération courante ne sera pas reproduit dans la nouvelle génération. Pour éviter que les meilleurs individus soient altérés, il est donc intéressant d'introduire la stratégie de l'*élitisme* [93]. Elle consiste à copier le meilleur ou un sous-ensemble des meilleurs chromosomes de la génération courante à la prochaine génération. Ainsi, on peut choisir, par exemple, de reproduire à l'identique les 5%

meilleurs de la population à chaque génération, l'opérateur de reproduction ne jouant alors que sur les 95% restant⁴. Cette stratégie d'élitisme permet d'augmenter la dominance des meilleurs individus dans une population ce qui permet une amélioration des solutions à obtenir.

10. Discussion sur les algorithmes génétiques

La robustesse des algorithmes génétiques est sans aucun doute régulée par des facteurs liés à leur modélisation. En outre, l'intérêt majeur qu'on leur accorde est nécessairement justifié par le fait qu'ils présentent des propriétés fortes intéressantes pour l'optimisation d'un problème donné.

Les algorithmes génétiques se présentent comme une approche très intéressante. Ils sont capables d'exploiter l'espace de recherche sans avoir à imposer de restrictions liées au type d'algorithme choisi, ce qui n'est pas toujours le cas des autres algorithmes d'apprentissage automatique. Par contre, les choix faits au niveau du codage employé, ou du type de méthodes de sélection et de transformation utilisées, vont beaucoup influencer l'apprentissage. Une des tâches les plus difficiles va être la définition de ces choix d'implémentation des algorithmes génétiques. Ce type de problème est d'autant plus délicat qu'il n'existe pas de méthodologie générale à suivre.

La difficulté du choix d'un codage et d'une bonne fonction de sélection et d'adaptation fait des algorithmes génétiques une technique très particulière qui doit être bien maîtrisée pour donner de bons résultats. Dans le cas contraire, ces algorithmes n'arriveront pas à trouver de bonnes solutions aux problèmes posés. Les algorithmes génétiques sont très lourds à exécuter, d'où l'intérêt d'exploiter le parallélisme dans ce type d'approche. De plus, le manque d'une garantie de convergence vers une bonne solution s'agit de l'un des principaux désavantages des algorithmes génétiques. Depuis longtemps, ils ont été très largement mis à contribution sans qu'aucune étude ne soit venue pour valider le fonctionnement. C'est d'abord Koza [80] qui en valida les fondements mathématiques. L'algorithme génétique a été modélisé par les chaînes de Markov [4]. Ce premier pas a permis d'enlever aux algorithmes génétiques la métaphore biologique qui les entourent, les mettant en relation avec les processus stochastiques, qui sont des modèles mathématiques.

⁴ On parle alors parfois d'*AG augmenté* (augmented GA) pour qualifier ce type d'AG.

Cette modélisation a permis de démontrer mathématiquement qu'un algorithme génétique converge vers l'optimum [55]. Cette convergence ne vient pas des algorithmes génétiques de manière intrinsèque, mais de l'astuce algorithmique visant à conserver systématiquement le meilleur individu dans la population. En effet, au terme de la modélisation, la représentation stochastique fournit la preuve mathématique de la convergence d'un algorithme génétique utilisant la politique *Elitiste*. La clé de la convergence réside donc dans l'opérateur de sélection [55], car le modèle utilisé ne démontre le résultat majeur de la convergence que si on tient compte d'une caractéristique essentielle qui est la conservation du meilleur individu dans la population. Cette caractéristique étant réalisée par l'opérateur de sélection.

Comme d'autres algorithmes d'optimisation, ils connaissent les problèmes liés à la convergence et au blocage dans des *minima locaux*. Par contre, si les fonctions de mutation, de reproduction et de croisement sont bien définies, les algorithmes génétiques peuvent constituer une solution assez intéressante pour s'échapper des minima locaux. Un autre avantage des algorithmes génétiques est le fait qu'ils admettent l'introduction de connaissances théoriques et de solutions préexistantes dans la population initiale à traiter. Donc, cette population initiale nous permet d'avoir un point de départ pour l'apprentissage qui peut être plus proche de la solution finale recherchée.

11. Conclusion

Nous avons présenté dans ce chapitre les principes de base d'un outil de résolution qui a fait ses preuves en intelligence artificielle (IA), à savoir l'algorithme génétique. Nous retenons principalement qu'un algorithme génétique est une reproduction artificielle de mécanismes naturels liés à la génétique. Son principe, basé sur une description fidèle d'une évaluation naturelle, est un processus cyclique manipulant une population et assurant une recherche efficace dans un espace de solutions candidates d'un problème donné, par application des trois opérateurs qui sont la sélection, le croisement et la mutation, dans le but d'optimiser la fonction d'adaptation associée, définie sur un espace éventuellement complexe.

L'algorithme génétique n'exige aucune connaissance de la manière dont le problème est résolu ; il est seulement nécessaire de pouvoir estimer la qualité d'une solution potentielle. Cette approche présente l'immense avantage pratique de fournir des solutions pas trop éloignées de l'optimal, même si l'on ne connaît pas de résolution algorithmique.

Il est clair que pour pouvoir construire un système de reconnaissance de la parole, robuste, fiable et qui peut rapprocher les performances humaines en terme de reconnaissance, une

combinaison de différents paradigmes et approches est nécessaire. C'est à ce niveau que se situe notre contribution et qui consiste à utiliser une des nouvelles orientations de l'IA ; l'intégration des algorithmes génétiques pour augmenter les performances d'un classifieur neuronal. Ces modèles sont appelés modèles hybrides dont le principe sera expliqué dans le chapitre 4, ainsi qu'une présentation détaillée des différents modules du système proposé.

Chapitre 4

Proposition d'un système hybride AG/PMC pour la reconnaissance de la parole

1. Introduction

Dans ces dernières années, beaucoup de recherches ont été entreprises dans le domaine de la reconnaissance de la parole, bien que les études en cours soient nombreuses, il n'existe encore aucun système capable de traiter de façon fiable la reconnaissance.

Malgré l'efficacité des réseaux de neurones, qui s'avèrent être des outils performants de reconnaissance des formes en générale et de la parole en particulier, grâce à leur capacité d'apprentissage, de généralisation et de classification, il existe encore des problèmes liés à leur style d'apprentissage : le temps d'apprentissage est lent, les paramètres initiaux peuvent avoir des effets considérables sur les concepts appris, il n'existe pas de méthodologies pour le choix d'une topologie de réseau adéquate au problème donné et leur incapacité à expliquer les résultats qu'ils fournissent.

Ces limitations ont conduit à proposer des hybridations avec d'autres techniques telle que les algorithmes génétiques afin de remédier aux inconvénients et aux limites de ce paradigme, tout en améliorant le taux de reconnaissance.

L'interaction entre les processus d'apprentissage et d'évolution est beaucoup étudiée. Elle a montré son intérêt dans le domaine de l'optimisation sous la forme d'algorithmes dits « hybrides » combinant un algorithme génétique et un réseau de neurone. Plusieurs recherches consistent à étudier la dynamique de cette interaction [136, 2, 25, 44, 46, 91, 100, 108].

Nous proposons dans ce chapitre un système de reconnaissance vocale basé sur une nouvelle technique qui combine efficacement les algorithmes génétiques et les PMCs, pour reconnaître des chiffres isolés indépendamment du locuteur. Cet algorithme génétique qui propose de faire évoluer une population d'individus (*MLP – Multi-Layer Perceptron*) dans un environnement, tout en favorisant la survie et la reproduction des individus les mieux adaptés à cet environnement, en vue de trouver un système de reconnaissance vocale optimal de meilleurs performances.

2. Orientation vers les approches hybrides

Chacun des paradigmes d'intelligence artificielle (IA) présentés précédemment : les réseaux de neurones et les algorithmes génétiques, possèdent des caractéristiques propres à leur fonctionnement, offre des avantages et souffre en même temps de plusieurs limites. Pour dépasser ces limites, les travaux actuels d'IA se sont orientés vers les systèmes hybrides, et ils

ont rapporté beaucoup de connaissances au sujet des meilleures manières de combiner les différentes techniques d'IA.

Le concept de système hybride ou de méthodes hybrides est très large. Ce groupe d'applications inclut toute méthode qui intègre deux ou plusieurs approches dans le même système pour la résolution d'un problème, éliminant les limites d'un paradigme en exploitant les avantages d'un autre. Cette hybridation se base sur la complémentarité qui peut exister entre deux paradigmes et tente aussi de les faire coopérer pour résoudre un problème complexe, où chacun d'entre eux prendra à sa charge le traitement d'une tâche qui s'accommode le mieux avec son style de raisonnement et où chacun d'entre eux viendrait pallier les inconvénients de l'autre sans pour autant en renforcer les faiblesses.

3. Pourquoi les algorithmes génétiques

Une question peut se poser : *pourquoi les algorithmes génétiques ?*

La plupart des algorithmes d'apprentissage employés pour les réseaux de neurones sont déterministes, de type gradient. La dépendance entre les caractéristiques modifiables du réseau et le critère d'évaluation est une fonction différentiable. Optimiser de façon itérative le critère d'évaluation revient alors à apporter des modifications de faible amplitude aux paramètres dans la direction du gradient du critère d'évaluation par rapport à ces paramètres. Cette optimisation est réalisée par un algorithme d'apprentissage, de type supervisé, dans le sens du gradient pour une maximisation sur l'erreur estimée à partir des sorties du réseau et dans le sens contraire (descente de gradient) pour une minimisation. Cet algorithme réalise une descente de gradient de la surface d'erreur.

Ce type de problème d'apprentissage d'un réseau de neurones par rétro-propagation peut donc être résolu par une méthode de type recherche locale. Partant d'une configuration de poids initiale, la méthode va chercher la meilleure solution dans le voisinage de cette configuration. Cette solution est optimale localement mais peut ne pas correspondre à un optimal global, car il est possible qu'il existe une meilleure solution qui n'est pas dans le voisinage de la configuration initiale. La rétro-propagation met plus de temps pour atteindre le voisinage d'une solution optimale (voir figure 4.1). Sauf que, elle peut atteindre cette solution avec une meilleure précision [79].

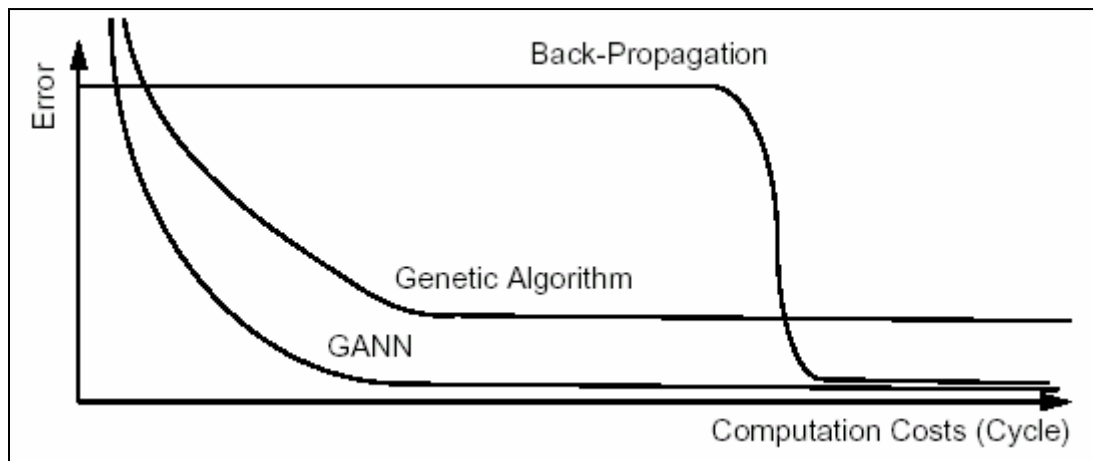


Figure 4.1 : Convergence des stratégies de recherche de [78]

L'utilisation d'algorithmes d'apprentissage déterministes pour les réseaux de neurones impose aussi une autre exigence, l'absence d'optima locaux du critère d'évaluation du réseau par rapport aux paramètres. L'expérience montre que cette exigence est rarement satisfaite dans la pratique. En dehors de l'existence de solutions optimales multiples, peu gênante, nous rencontrons souvent un nombre important de solutions sous-optimales qui correspondent à des optima locaux du critère d'évaluation du réseau. Les algorithmes de recherche déterministes sont incapables de quitter un optimum local atteint et ne permettent donc pas d'assurer la complétude⁵ de la recherche.

Grâce à leur caractère stochastique, les algorithmes génétiques peuvent surmonter ces problèmes et, sous certaines conditions, atteindre avec une probabilité élevée un optimum global quelque soit l'ensemble des solutions initiales [99]. D'autre part, les algorithmes génétiques permettent aussi d'assurer la complétude de la recherche.

Si l'on se base sur le principe que les algorithmes génétiques effectuent une recherche dans l'espace global des solutions potentielles, et que les réseaux de neurones se concentrent plutôt vers une recherche locale dans l'espace du voisinage (dont la solution trouvée, supposée optimale, ne l'est peut-être pas si l'on considère l'espace global), on peut supposer que ces deux méthodes sont complémentaires [131].

En conclusion, les algorithmes génétiques ont montré leur capacité à éviter la convergence des solutions vers des optima locaux, aussi bien lorsqu'ils sont combinés avec des méthodes de recherche locale comme la rétro-propagation du gradient [116] que lorsqu'ils sont seuls [52] (voir figure 4.1).

⁵ La probabilité d'atteindre un état cible (état recherché) à partir de n'importe quel autre état est non nulle.

4. Intégration Algorithme Génétique/Réseaux de Neurones

L'étude des réseaux de neurones et l'étude des algorithmes génétiques se sont développés en parallèle et ont souvent été en interaction durant cette dernière décennie [99]. Toutes les deux sont des méthodes de calcul autonomes pour les systèmes de traitement des informations biologiques et sont devenues des domaines de recherche très répandues à travers le monde. En général, les réseaux de neurones sont utilisés comme des systèmes d'apprentissage et les algorithmes génétiques comme des systèmes d'optimisation. Mais, de nombreux chercheurs ont découvert qu'elles peuvent être combinées selon différentes manières [66], le plus souvent, selon l'un des deux types général suivants :

- Un premier type de combinaison où les algorithmes génétiques et les réseaux de neurones sont appliqués séquentiellement : dans ce cas, les algorithmes génétiques et les réseaux de neurones sont appliqués à deux phases différentes du problème. La combinaison la plus simple est d'utiliser les algorithmes génétiques pour le traitement de l'ensemble des données d'apprentissage des réseaux de neurones. Par exemple, une application des algorithmes génétiques peut porter sur l'optimisation de l'espace de représentation des données toutes en éliminant (débarassant) les caractéristiques redondantes ou inutiles du système de reconnaissance de formes [66]. Cette sélection des primitives est considérée comme un problème d'optimisation combinatoire et a fait l'objet de recherche dans de nombreuses disciplines.

N. Benahmed [16] a conçu un système pour la reconnaissance de chiffres manuscrits isolés où un algorithme génétique a été utilisé pour résoudre ce type de problème de sélection des primitives dans la reconnaissance. Le second objectif de cette sélection est aussi de maintenir et/ou améliorer la performance du classifieur neuronal utilisé par le système de reconnaissance. Les résultats obtenus lors de la sélection des primitives ont permis de réduire la complexité du réseau de neurones utilisé. Le nombre de primitives a été réduit de 25% par rapport à l'ensemble des primitives extraites du système tout en maintenant la performance en taux de reconnaissance du système.

Les combinaisons séquentielles ne sont pas très intéressantes, du fait que les algorithmes génétiques et les réseaux de neurones sont utilisés bien indépendamment et l'une ou l'autre peut facilement être remplacée par une technique alternative.

- Un deuxième type, où elles sont appliquées simultanément. Ce type de combinaison est connu sous le nom de : *combinaison collaborative*. L'évolution des réseaux de neurones par les algorithmes génétiques peut se faire de différentes manières :

a- *Recherche directe des poids de connexions* : dans ce cas, un individu consiste en un ensemble de poids codés d'un réseau et l'algorithme génétique est utilisé pour optimiser ces poids avec les avantages qu'il offre, à savoir, l'indépendance par rapport à l'architecture, les propriétés de recherches globale, ...etc [114]. Cette approche semble donner en général d'excellents résultats même si le temps de convergence des algorithmes reste fort élevé [101, 140, 3, 124, 135].

b- *Recherche des paramètres d'apprentissage optimaux* : un individu manipulé par les algorithmes génétiques consiste en un codage des paramètres de la fonction d'apprentissage : du taux d'apprentissage, et éventuellement du moment, intervenant dans l'algorithme de la descente du gradient pour l'ajustement des poids [140, 115, 120].

c- *Recherche des poids initiaux d'un réseau de neurones* : suivie d'une descente de gradient traditionnelle pour l'ajustement « plus fin » de ces poids : il s'agit d'un couplage direct entre deux méthodes d'optimisation [115].

A. Yas Abbas et M.A. Musbah [1] ont développé un modèle hybride neuro-génétique pour la reconnaissance de formes. Ce modèle se compose de deux étapes, une première où un algorithme génétique est utilisé pour la recherche d'une configuration initiale de poids pour la deuxième phase. Pour cette dernière, le meilleur vecteur de poids initiaux sera considéré pour l'apprentissage du réseau neuronal par l'algorithme de la rétro-propagation du gradient. Les résultats sont satisfaisants car ils permettent d'accélérer la vitesse de convergence et de réduire le nombre d'époques à moins de 50%.

d- *Recherche d'une topologie optimale* : les réseaux qui relèvent de ce type d'approche sont aussi appelés réseaux évolutifs (*evolutionary ANN*) [113, 110, 10, 45, 107, 79]. Un individu des algorithmes génétiques est une description architecturale du réseau spécifiant, entre autres, le nombre de couches cachées et le nombre de nœuds dans ces couches ; les algorithmes génétiques sont alors utilisés pour faire évoluer favorablement une population de topologies et sélectionner les architectures de RN plus performants par rapport à une application donnée.

Alternativement, dans l'approche collaborative, les algorithmes génétiques et les réseaux de neurones sont utilisés conjointement dans un même système, dans lequel, une population de réseaux de neurones est évoluée, pour la résolution d'un même problème.

Dans le cadre de cet mémoire, notre intérêt s'est porté sur l'hybridation AG/PMC pour la recherche de poids optimaux.

5. Notre choix

Les réseaux de neurones de type PMC ont été fortement utilisés dans différentes applications et particulièrement en reconnaissance de formes. L'emploi de ces réseaux de neurones plutôt que des techniques classiques pour la classification peut se justifier par les arguments suivants :

- Par rapport à l'ensemble des systèmes connexionnistes, ils ont l'avantage d'être basés sur des principes simples et relativement maîtrisables, et donc une simplicité de mise en œuvre (peu d'analyse mathématique préliminaire) ;
- Capacité de classification et de généralisation ;
- Capacité d'approximation universelle prouvée ;
- Il est possible de détecter le moment où l'algorithme d'apprentissage n'est plus capable d'améliorer les performances, ce qui permet d'optimiser le temps d'apprentissage ;
- Robustesse par rapport à des défaillances internes du réseau (caractère distribué de la représentation).

La plupart des applications courantes utilisent cette architecture de PMC, dont les fonctions d'activation à chaque nœud sont de type sigmoïdal. Il existe un certain nombre de théorèmes garantissant les capacités d'approximation universelle sous certaines conditions (continuité de la fonction à approcher, nombre suffisant d'unités dans les couches cachées), même si le réseau ne comporte qu'une seule couche cachée. Cependant, ces théorèmes ne fournissent pas de méthodes systématiques constructives guidant le choix de la topologie à adopter pour une application particulière [68, 115].

Notre choix est alors porté sur une architecture de type perceptron multi-couches (PMC), qui est généralement entraîné par l'algorithme de rétro-propagation. Ces réseaux ont été appliqués avec succès à une multitude de problèmes de classification. Les PMCs sont les plus employés en reconnaissance de la parole [102, 103] et ont montré leurs aptitudes, notamment pour les

mots isolés [23, 111]. Le PMC est alors un excellent candidat pour une application de reconnaissance acoustique. Une seule couche cachée est généralement utilisée. L'addition de cette couche cachée permet au réseau de modéliser des fonctions de décision complexes et non linéaires entre n'importe quel espace d'entrée et de sortie.

Cependant, l'algorithme d'apprentissage pour l'entraînement des PMCs habituellement utilisé présente certaines limites, comme cité précédemment. Il est donc évident plus intéressant d'envisager l'utilisation d'autres techniques d'optimisation. En effet, les algorithmes génétiques conçus pour imiter un processus observé dans l'évolution naturelle est proposé en tant qu'algorithmes d'optimisation puissants, pouvant être une alternative plus prometteuse pour l'apprentissage des réseaux de neurones multi-couches, et semblent améliorer les performances de ces derniers.

6. Architecture générale du système de reconnaissance proposé

L'objectif de notre projet est d'effectuer, à partir d'une population d'individus entraînés par algorithme génétique, une sélection du meilleur classifieur neuronal de type PMC, spécialisé pour la reconnaissance des chiffres isolés indépendamment du locuteur, en appliquant une stratégie de sélection basée sur les algorithmes évolutionnaires. Les algorithmes génétiques qui sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et de l'évolution naturelle : croisement, mutation, sélection, ...etc. Pour les utiliser, nous devons suivre les étapes suivantes :

- choisir un codage des éléments de la population, Le codage binaire a été très utilisé à l'origine. Les codages réels sont désormais largement utilisés, notamment pour l'optimisation de problèmes à variables réelles [65, 20] ;
- La mise en place d'un mécanisme de génération de la population initiale ;
- La définition de la fonction à optimiser. Celle-ci est appelée fonction d'adaptation (*fitness*) ou fonction d'évaluation de l'individu. La solution optimale du problème est obtenue à partir de la fonction d'évaluation du chromosome [67] ;
- L'utilisation des opérateurs de reproduction permettant de diversifier la population au cours des générations et d'explorer l'espace d'état [20] ;
- La définition des paramètres de dimensionnement : taille de la population, nombre total de générations ou critère d'arrêts, probabilités d'application des opérateurs de croisement et de mutation.

Dans ce qui suit, nous allons décrire l'architecture générale du système proposé (figure 4.2).

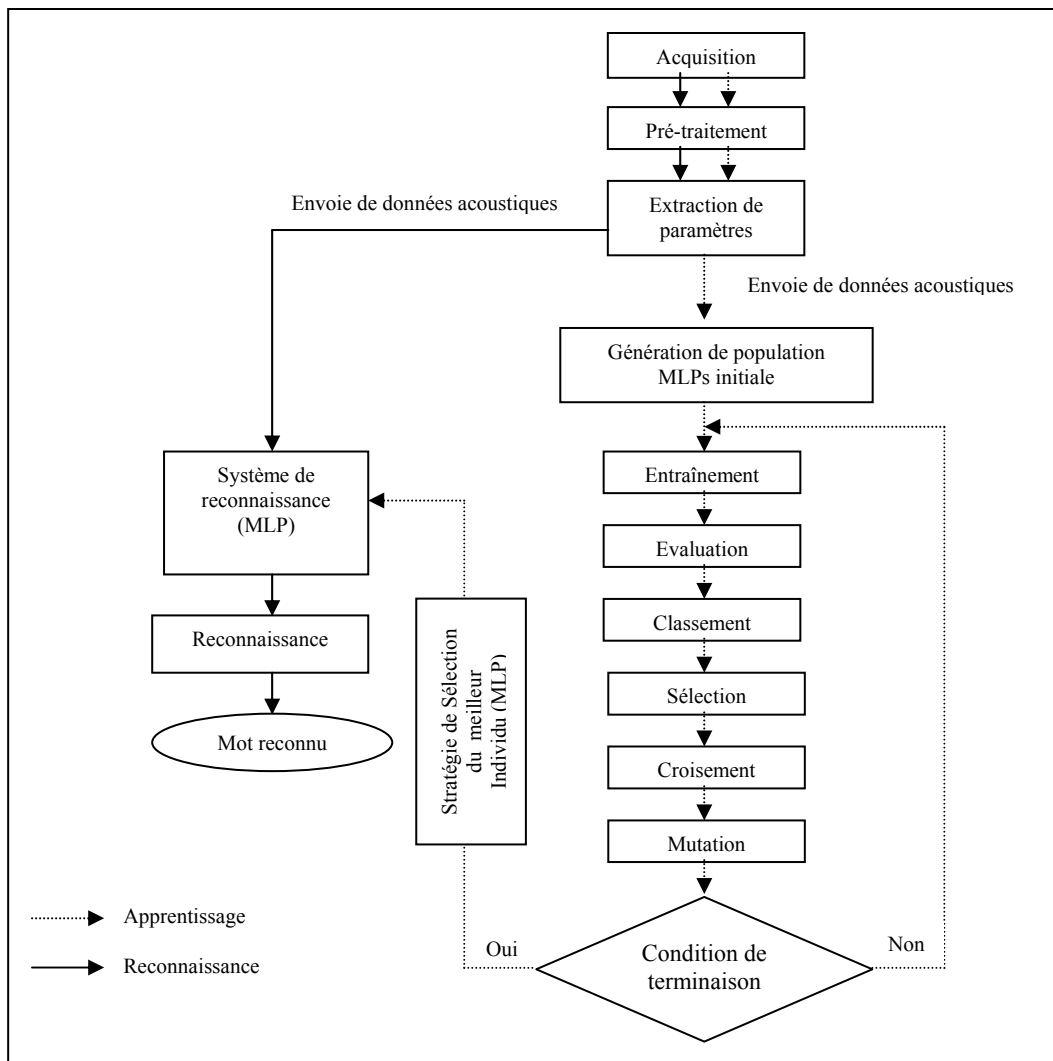


Figure 4.2 : Schéma général du système de reconnaissance vocal

7. Paramétrisation du signal vocal

Les premiers modules de traitement dans un système de reconnaissance de la parole sont les suivants :

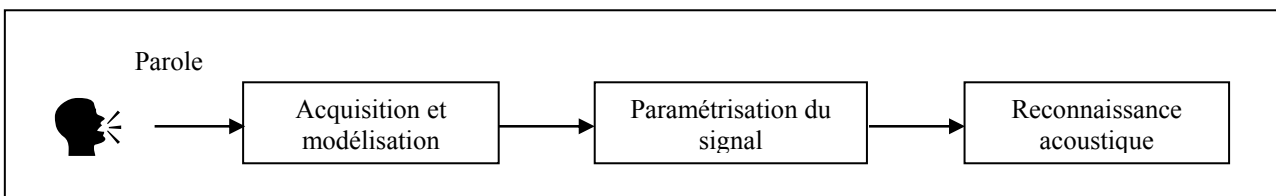


Figure 4.3 : Chaîne de traitement acoustique d'un système de reconnaissance de la parole

7.1. Base de données

Dans tout système de reconnaissance de la parole, il est nécessaire d'avoir une base de données afin d'effectuer tous les traitements. Plusieurs bases de données ont été construites

pour différentes langues (BRIEF-120 et BDSONS pour la langue française, TIMIT pour la langue anglaise, ...etc), mais pour la langue arabe on remarque une absence presque totale et c'est le grand problème posé lors du développement des système de reconnaissance de la parole arabe.

Dans notre processus de reconnaissance, les expériences seront effectuées sur une base de données personnelle constituée de dix (10) chiffres prononcés en arabe. Le nombre de classes est donc égale à 10 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9). Cette base de données est partagée en 3 corpus :

- Le premier corpus (Base de Données 1 – BD1) prononcée par 50 locuteurs (25 de sexe masculin et 25 de sexe féminin) de différents âges. Chaque chiffre est prononcé 5 fois par chaque locuteur, et ceci pour la phase d'apprentissage.
- Le deuxième corpus (Base de Données 2 – BD2) prononcée par 40 locuteurs (20 de sexe masculin et 20 de sexe féminin) de différents âges. Chaque chiffre est prononcé 5 fois par chaque locuteur, et ceci pour la phase de test.
- Et le troisième corpus (Base de Données 3 – BD3) prononcée par 20 locuteurs (10 de sexe masculin et 10 de sexe féminin) de différents âges. Chaque chiffre est prononcé 5 fois par chaque locuteur, et ceci pour la validation du système.

Les données ont été enregistrées dans une salle sourde en utilisant un microphone.

7.2. Acquisition du signal vocal

Comme le montre la figure 4.3, le signal de parole après son acquisition via un dispositif d'entrée, usuellement un microphone, est d'abord numérisé pour être utilisable par ordinateur. Cette opération tend à transformer un phénomène temporel analogique en une suite d'éléments discrets, les échantillons. Pourtant, avant d'obtenir ces mesures, le signal a subi des modifications dues à l'environnement dans lequel se trouve le locuteur, et à l'influence du système d'acquisition. La numérisation sonore repose sur deux paramètres : *la quantification* et *la fréquence d'échantillonnage*.

La quantification définit le nombre de bits sur lesquels on veut réaliser la numérisation. Elle permet de mesurer l'amplitude de l'onde sonore à chaque pas de l'échantillonnage. Le choix de la fréquence d'échantillonnage (f_e) est aussi déterminant pour la définition de la bande passante représentée dans le signal numérisé.

Le module suivant, dans la chaîne de traitement acoustique est celui qui extrait des paramètres pertinents pour la reconnaissance de la parole.

7.3. Extraction des paramètres acoustiques

L'extraction de caractéristiques est une des plus importante étape dans le processus de reconnaissance de la parole. C'est la première étape du processus de reconnaissance de la parole : étape de codage et d'extraction d'informations nécessaire à la reconnaissance. Ces dernières doivent représenter au mieux le signal qu'elles sont censées modéliser et extraire le maximum d'informations utiles pour la reconnaissance.

7.3.1. Mise en forme du signal vocal

Avant l'extraction des paramètres acoustiques du signal vocal, il est nécessaire de mettre en forme le signal de parole (figure 4.4). Pour cela, un ensemble d'opérations est pris en considération avant le traitement.

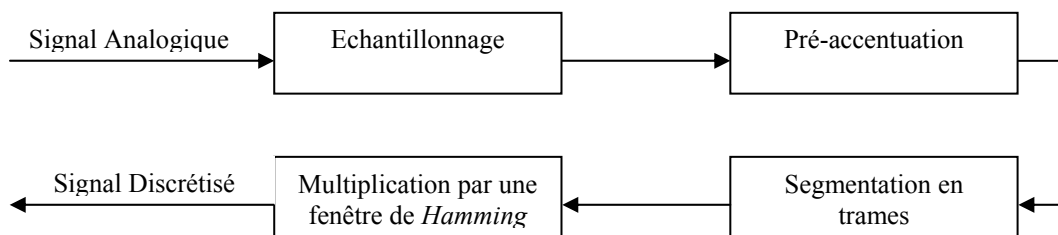


Figure 4.4 : Mise en forme du signal vocal

Le choix de la fréquence d'échantillonnage (f_e) est indépendant de la méthode d'analyse choisie. L'information acoustique pertinente du signal de parole se situe principalement dans la bande passante [50Hz – 8Khz], la f_e devrait donc au moins être égale à 16Khz, selon le théorème de *Schannon*, mais elle peut varier en fonction du domaine d'application ou des besoins ou contraintes matériels. Dans notre système, on choisit de travailler avec une f_e de valeur moyenne égale à 11025Hz.

La pré-accentuation du signal vocal consiste en un passage dans un filtre de transmittance : $1 - \mu Z^{-1}$ (μ est compris entre 0.9 et 1), ce qui a pour effet d'accentuer les parties hautes fréquences des spectres, qui sont moins énergétique que les basses fréquences. Ce pré-traitement assure un bon conditionnement des algorithmes de résolution. En pratique, la valeur de μ n'est pas critique et on choisit une valeur fixe, par exemple 0.95.

La trame acoustique est un ensemble de coefficients ou paramètres, calculés sur un bloc d'échantillons. Dans la plupart des applications, ce bloc d'analyse est de taille fixe, il

correspond à un temps de parole de 20 à 40 ms. La suite de vecteurs d'analyse est obtenue en déplaçant ce bloc de 10 à 20 ms ; il y a donc recouvrement (chevauchement) de blocs, ce qui apparente cette analyse à une analyse de type fenêtre glissante.

Pour assurer une bonne résolution spectrale, cela impose un temps d'analyse suffisant. On utilise très couramment des fenêtres de 40 ms décalées de 20 ms. On choisit donc pour la fréquence d'échantillonnage précédente, $N=512$ comme nombre d'échantillons par tranche d'analyse, et un recouvrement de $L=256$ échantillons. Ce découpage du signal en trames produit des discontinuités aux frontières des trames, qui se manifestent par des lobes secondaires dans le spectre ; ces effets parasites sont réduites en appliquant aux échantillons de la trame une fenêtre de pondération dite : *fenêtre de Hamming*. La fonction de *Hamming* est définie par :

$$w(n) = \begin{cases} 0.54 - 0.46 \left(\frac{2\pi n}{N-1} \right) & , 0 \leq n \leq N-1 \\ 0 & \text{sin on} \end{cases} \quad (4.1)$$

7.3.2. Paramètres acoustiques

Après cette mise en forme du signal (commune à la plupart des méthodes d'analyse de la parole), la phase suivante est de déterminer les paramètres pertinents pour la reconnaissance de la parole.

Il existe de nombreux algorithmes pour calculer des vecteurs acoustiques mais ils visent tous à obtenir des vecteurs acoustiques représentatifs de l'information linguistique contenue dans le signal de parole et aussi insensibles que possible aux causes non-linguistiques de variabilité tels que l'identité du locuteur, l'environnement acoustique (par exemple, le bruit d'ambiance) ou le canal de transmission (par exemple, la distorsion induite par une ligne téléphonique ou un microphone inadapté).

La technique d'analyse PLP (Perceptual Linear Predictive) [59] est basée sur le spectre de parole court-terme. Quoique ce dernier soit ensuite modifier par des transformations spectrales psychophysiques, la technique PLP est altérable lorsque les valeurs spectrales court-terme sont modifiées par la réponse fréquentielle du canal de transmission. Pour remédier à ce problème, on utilise l'analyse RASTA-PLP (RelAtive SpecTrAl processing-Perceptual Linear Predictive) qui rend la technique PLP plus robuste aux déformations spectrales linéaires. C'est une technique de paramétrisation acoustique robuste au bruit, qui a pour objectif d'extraire des paramètres acoustiques qui donneraient des performances aussi

bonnes que les paramètres usuels. On choisit d'utiliser le filtrage RASTA pour supprimer les composantes qui varient plus lentement ou plus rapidement que le signal de parole et donc leur influence sur le système de reconnaissance.

Chaque segment de parole contient un nombre variable de trames ce qui complique la gestion de la dynamique temporelle par les réseaux de neurones du type PMC. Pour lever cette difficulté, un alignement temporel est effectué après la phase d'analyse afin de garder une taille fixe pour le vecteur spectral, quelle que soit la longueur de son analyse. Pour cela une procédure particulière est utilisée. Celle-ci consiste à segmenter les données sur les zones stables de chaque mot, puis en divisant chaque segment en trois intervalles sur lesquels nous effectuons une moyenne des vecteurs acoustiques. Le nombre de paramètres présentés à l'entrée de notre système est toujours fixe quelle que soit la longueur du segment [76].

L'ensemble de nos travaux est basé sur l'utilisation des vecteurs acoustiques de type RASTA-PLP. Chaque segment de parole est représentée donc par 12 composantes plus l'énergie (RASTA-PLP + E). Les valeurs des 13 coefficients sont normalisées par leur écart-type. Ainsi le vecteur acoustique fournit à l'entrée de notre PMC est composé de 13 paramètres.

8. Présentation du modèle de population

Une population de SRAPs basés sur les réseaux de neurones de type feed-forward (PMC), est donc créée dans un environnement acoustique (figure 4.5). L'initialisation de la population se fait en créant un certain nombre de SRAPs ayant une architecture identique mais des poids de valeur différents [66, 131].

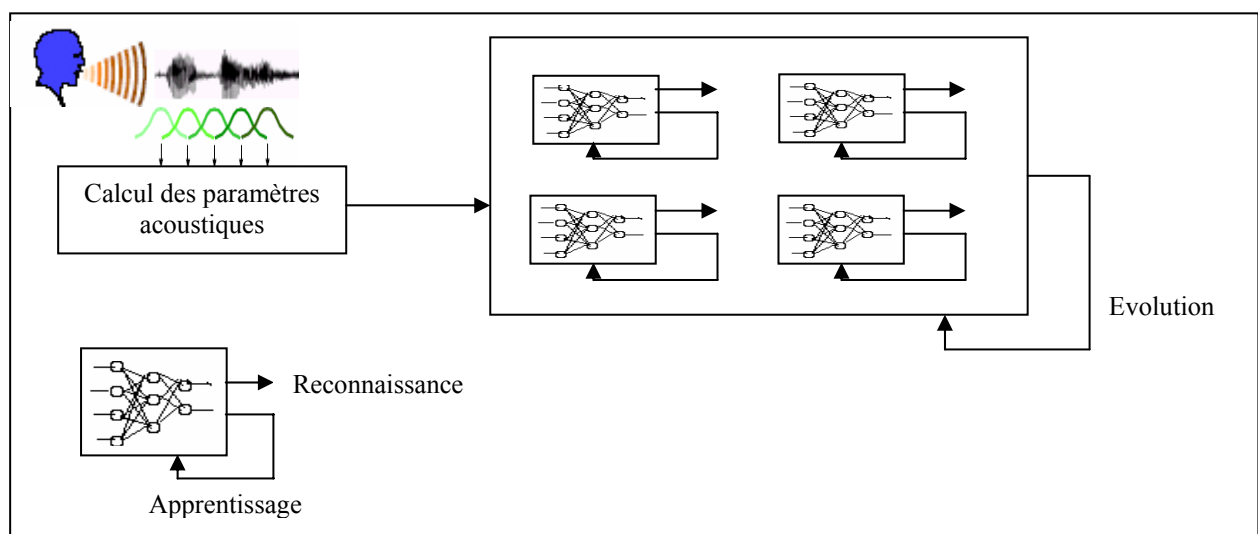


Figure 4.5 : Modèles de population de SRAPs

8.1. Architecture du réseau neuronal

Le réseau de neurones est composé de trois couches dont les caractéristiques sont les suivantes :

- *Couche d'entrée* : avec $n=13$ entrées, représentant les vecteurs caractéristique des coefficients RASTA-PLPs plus l'énergie E ;
- *Couche cachée* : dont le nombres d'unités est indéterminé ;
- Et une *couche de sortie* avec dix neurones ($s=10$) représentant les différents mots à reconnaître.

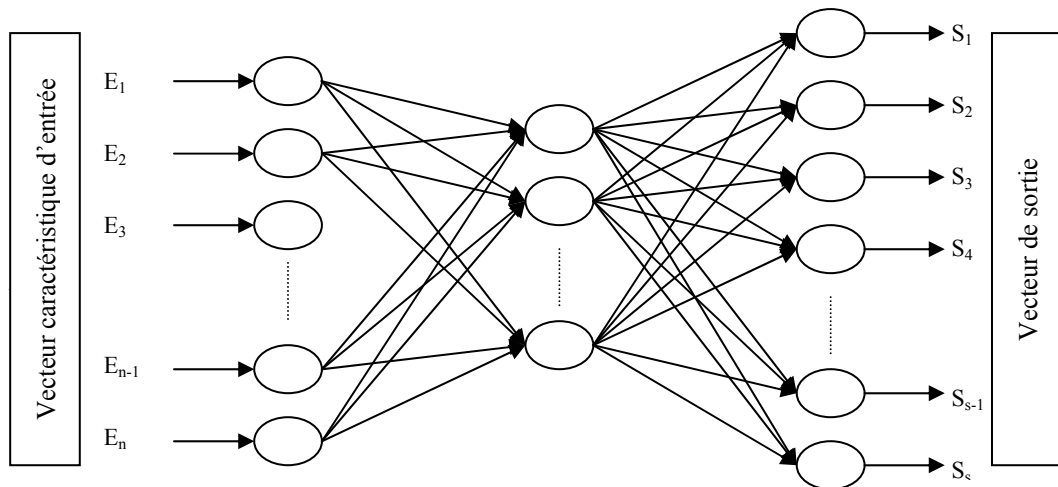


Figure 4.6 : Structure générale d'un individu (SRAP)

Afin de déterminer une architecture adéquate, nous devons effectuer une série d'apprentissages en faisant varier le nombre d'unités (neurones) de la couche cachée. Ceci nous permettra d'éviter tout effet de sur-apprentissage et d'architecture non adéquate. Tous les apprentissages sont effectués jusqu'à la convergence du réseau. L'algorithme d'apprentissage est un algorithme adapté : la représentation des solutions, les opérateurs génétiques sont adaptés au problème.

8.2. Codage des chromosomes (Représentation des solutions)

Du fait qu'il est nécessaire de décoder les poids en valeurs réelles avant chaque opération de reconnaissance ou d'apprentissage des réseaux de neurones, et afin de limiter le nombre d'opérations, déjà coûteuse, à effectuer lors de l'apprentissage, la représentation adoptée dans le système proposé est une représentation basée réels, qui consiste alors à se servir de nombres réels directement.

Soit w_{ij}^k le poids d'une connexion du neurone i vers le neurone j de la couche k , et soit θ_i^k le seuil du neurone i de la couche k . Si l'on considère le PMC de la figure 4.6, sa représentation génotypique est la suivante :

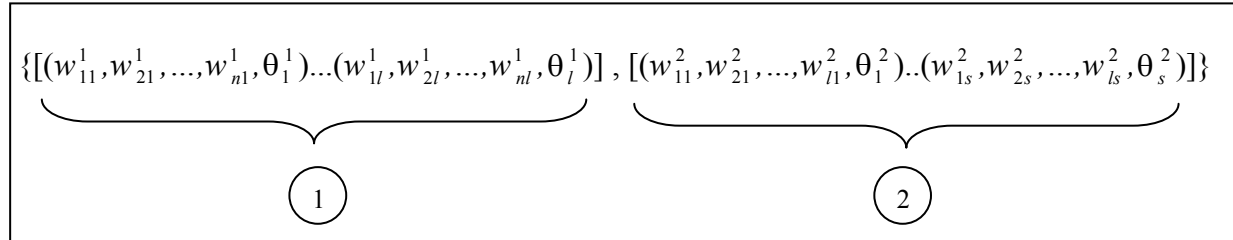


Figure 4.7 : Représentation génotypique du réseau

Chaque chromosome codifie un réseau en sa totalité et il est formé de deux listes :

1. Couche cachée ;
2. Couche de sortie.

Chaque liste est la concaténation des neurones de la couche correspondante. Chaque neurone est représenté par la concaténation des poids de ses entrées et de son seuil. Le réseau est alors entièrement codé dans les chromosomes.

8.3. Génération et initialisation de population

Généralement, la population est générée aléatoirement de manière à répartir les individus uniformément sur l'espace de recherche. Le nombre d'individus d'une population constitue un paramètre important pour l'algorithme génétique qu'il faudra déterminer.

L'algorithme d'apprentissage utilisé pour les individus PMCs de la population est un algorithme supervisé de type rétro-propagation du gradient. Dans cet algorithme, [22, 40] proposent de choisir des poids de manière uniforme dans un intervalle dépendant des données à apprendre. [40] propose des intervalles variant de $[-4.0, 4.0]$ à $[-0.5, 0.5]$ selon les données d'apprentissage. [101, 70] utilisent en général une probabilité initiale de distribution des poids dans l'intervalle $[-1.0, +1.0]$, ceci est dû au fait que, généralement, les solutions optimales tendent à avoir des poids dont la valeur absolue est assez petite. Pour les algorithmes génétiques, il serait plus intéressant d'élargir un peu cet intervalle afin d'éviter leur concentration sur une partie de l'espace de recherche. L'intervalle d'initialisation utilisé dans le système proposé est donc $[-2.0, +2.0]$. Tous les poids et les seuils sont initialisés aléatoirement sur cet intervalle.

8.4. Evaluation

La performance de chaque PMC, dans le cadre d'un algorithme génétique, est exprimée par un scalaire qui permet d'obtenir un classement suffisamment discriminant des individus. L'évaluation consiste généralement en un calcul de l'erreur quadratique du réseau codé par le chromosome à évaluer. Pour chaque individu i , sa valeur de *fitness* est donnée par :

$$f_i = \frac{1}{e_r} \quad (4.2)$$

où $e_r = \sum (y_j - d_j)^2$; y_j : sortie calculée du neurone de sortie j ; d_j : sortie désirée.

Un bon chromosome est alors un chromosome dont la fonction *fitness* f_i tend vers 1.

9. Evolution

Il est nécessaire d'effectuer des choix concernant la manière de faire évoluer les individus. Partant d'une méthode basée sur les algorithmes génétiques, nous devons déterminer quelle méthode de sélection, quel croisement et quelle mutation choisir, quels taux (fréquences d'agissement) leur associer.

9.1. Sélection ou reproduction

La sélection suppose que l'évaluation des individus a été faite. Elle permet d'identifier les individus les plus adaptés à leur environnement et d'éliminer, par conséquent, les moins adaptés selon les critères d'un mode de sélection donné. La plus connue et la plus utilisée dans la littérature est la méthode de sélection par roulette biaisée (ou *roulette wheel selection*).

9.2. Croisement

Cet opérateur consiste en l'obtention de deux nouveaux individus en combinant deux chromosomes parents. A cet opérateur est assigné une probabilité de croisement p_c . Il existe différentes techniques de croisement. Chacune des techniques s'applique sur des chromosomes dont la représentation est soit binaire ou réelle. Dans le système proposé, un croisement uniforme agissant sur les neurones directement semble être adéquat pour l'apprentissage des réseaux de neurones. L'idée derrière cette combinaison est que le succès d'un réseau est lié à la synergie entre les différents poids, et cette synergie est d'autant plus grande entre les poids des connexions afférents au même nœud. Il serait alors important de ne pas détruire ce bloc.

9.3. Mutation

L'opérateur de mutation inverse ou modifie certains sites des chromosomes de la nouvelle génération pour éviter l'uniformité des solutions. A cet opérateur est assigné une probabilité p_m . A chaque site de chaque individu est associé une probabilité aléatoire. Le site subit une mutation si sa probabilité est supérieure à celle de p_m .

La mutation peut être soit sur les poids, qui seront donc considérés comme sites indifféremment et sont modifiés, ou bien, la mutation agit sur tous les poids des connexions afférents au même nœud, qui seront donc tous modifiés ou aucun. Afin de conserver la cohérence de chaque nœud, comme dans le cas du croisement, on choisit que la mutation agit sur tous les poids d'un même nœud au lieu de les considérer indépendants.

Parmi la multitude de mutations existantes, trois mutations sont proposées. Il est possible de multiplier un gène par une valeur aléatoire (mutation multiplicative), d'ajouter une valeur aléatoire (mutation additive) ou d'effectuer une mutation gaussienne.

- *La mutation multiplicative* consiste à réduire la valeur du gène en la multipliant par une valeur aléatoire comprise entre 0 et d, soit pour un gène x_i :

$$x_i = x_i * \text{rand}(0, d) \quad (4.3)$$

- *La mutation Additive* consiste à muter les gènes en ajoutant une valeur aléatoire comprise entre -d et d, soit pour un gène x_i :

$$x_i = x_i + \text{rand}(-d, +d) \quad (4.4)$$

où d est une valeur entière que nous avons choisi égale à 2.

- *La mutation Normale* consiste à ajouter une valeur calculée selon une loi normale centrée sur 0 et d'écart-type σ :

$$x_i = x_i + \text{norm}(\sigma) \quad (4.5)$$

Pour faire un bon choix des paramètres d'évolution, il suffit d'effectuer plusieurs expériences. C'est un problème de réglage qui doit être optimisé pour chaque type de problème traité. Cela constitue une part importante du travail de l'expérimentateur. Dans la littérature, la définition de ces paramètres diffère d'une application à une autre.

10. Stratégie de sélection du meilleur individu PMC

Etant donné un grand nombre de différents classifieurs (PMCs), il existe un certain nombre de stratégies de sélection possibles. Habituellement, il n'est souvent pas facile de choisir le classifieur le plus performant (SRAP). La stratégie la plus simple pourrait être de choisir le meilleur à partir d'une population d'individus PMCs entraînée par un algorithme génétique sur un corpus d'apprentissage. Par suite, un corpus de test est utilisé pour mesurer la validité du système de reconnaissance proposé [109].

Une telle approche, bien qu'elle soit plus simple, le résultat de la sélection peut ne pas avoir la capacité de généralisation quand on lui présente des entrées qui n'ont pas été vues lors de l'apprentissage. Cette stratégie ne garantit donc pas des performances optimales de sélection [117].

Afin de remédier à ce problème et assurer des performances optimales de sélection, au lieu d'obtenir le meilleur classifieur unique, un sélecteur multiple devrait pouvoir choisir un sous-ensemble de classifieurs qui soit optimal dans le sens qu'il produit de meilleures performances. Ensuite on procède dans une deuxième phase de sélection, pour choisir le plus performant parmi les classifieurs contenus dans ce sous-ensemble.

10.1. Critère de sélection

Dans la plupart des cas, le critère de sélection est utilisé directement dans les algorithmes de sélection pour guider la recherche d'un sous-ensemble optimal de classifieurs.

Plusieurs scientifiques ont essayé d'appliquer différentes mesures de sélection pour choisir le meilleur sous-ensemble de classifieurs [126, 81, 118]. Comme il a été montré dans [118], une sélection par vote majoritaire (*MV*, *Majority Voting*) est particulièrement meilleure pour la réduction de la complexité du système de recherche.

Pour un nombre restreint de classifieurs, le choix peut être fait exhaustivement [127].

Etant donné un grand nombre de différents classifieurs, les algorithmes génétiques [32] sont avérés appropriés pour explorer un espace de recherche très vaste [82, 26, 119] et ont prouvé leur supériorité par rapport à d'autres techniques heuristiques de sélection [119]. D'autres approches évolutionnaires basées sur la sélection comprenant *la recherche tabou* (*tabu search*) [50, 117, 119], *l'étude par accroissement basé sur la population* (*PBIL*, *Population Based Incremental Learning*) [11, 119], ont montré une performance comparable tout en offrant une convergence plus rapide de l'algorithme.

10.2. Représentation

Etant donné un système de M classifieurs : $D = \{D_1, D_2, \dots, D_M\}$, soit $y_i = (y_{1i}, y_{2i}, \dots, y_{Mi})$ le résultat commun du système pour la $i^{\text{ème}}$ entrée multidimensionnelle x_i , où $y_{ji} = y_j(x_i)$ ($i = 1, \dots, N; j = 1, \dots, M$) désigne la sortie du $j^{\text{ème}}$ classifieur pour la $i^{\text{ème}}$ entrée. Dans ce travail, on suppose que les sorties y_{ji} sont binaires, telles que $y_{ji} = 1$ pour une classification incorrecte et $y_{ji} = 0$ pour une classification correcte.

Soit $v_k = (v_{k1}, v_{k2}, \dots, v_{kM})$ représentant un sous-ensemble de classifieurs, où $v_{kj} = \{0,1\}$ indique l'inclusion (1) ou bien l'exclusion (0) du $j^{\text{ème}}$ classifieur dans la décision de sélection.

Le vecteur solution, c'est à dire le chromosome peut être représenté comme suit (figure 4.7).

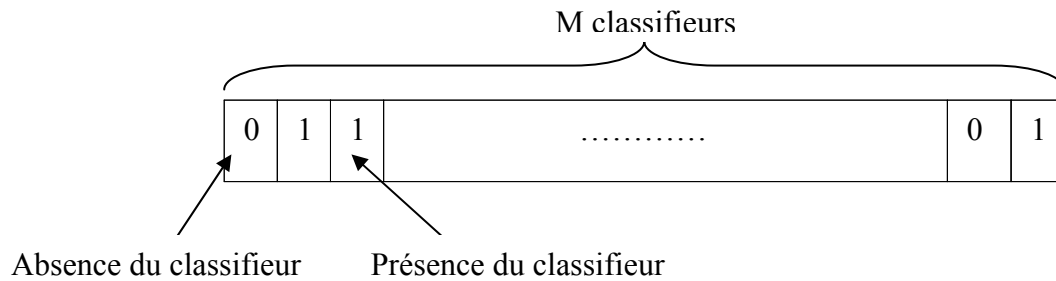


Figure 4.8 : Schéma représentant la structure du chromosome.

Chaque ensemble de classifieurs est représenté par une chaîne binaire de taille égale au nombre de classifieurs disponibles.

Etant donnée une combinaison v_k , la décision de sélection du sous-ensemble produite par le sélecteur par vote majoritaire peut être obtenue comme suit :

$$y_i^{MV}(v_k) = \begin{cases} 0 & \text{si } \sum_{j=1}^M y_{ji} v_{kj} \leq \left\lceil \frac{\sum_{j=1}^M v_{kj}}{2} \right\rceil \\ 1 & \text{si } \sum_{j=1}^M y_{ji} v_{kj} > \left\lceil \frac{\sum_{j=1}^M v_{kj}}{2} \right\rceil \end{cases} \quad (4.6)$$

Etant donné un ensemble de validation $X_{VA} = (x_1, x_2, \dots, x_N)$, la sélection peut être reformulée comme un processus simple d'optimisation dont l'objet est v_k , et la fonction d'évaluation (*fonction fitness*) que nous avons employée dans cette étude est représentée par la formule suivante :

$$y^{MV}(v_k) = \left[\sum_{i=1}^N y_i^{MV}(v_k) \right] / N. \quad (4.7)$$

10.3. Algorithmes de recherche

Un certain nombre d'algorithmes de recherche ont été considérés par rapport au sélecteur de classifieurs par vote majoritaire (MV).

Dans la sélection de classifieurs, le problème des minima locaux peut être posé dans le cas d'un espace de recherche très vaste. Des algorithmes évolutionnaires fonctionnant sur la base de population de solutions ont été montrés pour le traitement de tels cas. Les trois algorithmes présentés dans ce domaine sont : les algorithmes génétiques, recherche tabou et recherche par accroissement basé sur la population (*PBIL*). Ils utilisent tous la représentation binaire des solutions sélectionnées présentée dans le paragraphe 10.2 et produisent le meilleur sous-ensemble de classifieurs comme résultat.

10.3.1. Les algorithmes génétiques

Au départ, nous utilisons un ensemble large de différents classifieurs de type PMC, tous entraînés par le même ensemble d'apprentissage et le même ensemble de caractéristiques. Parmi ces classifieurs nous cherchons le sous-ensemble manifestant des comportements indépendants et en même temps fournissant une grande performance. A cette fin, une technique basée sur l'algorithme génétique est utilisée pour sélectionner l'ensemble de classifieurs le plus efficace.

Chaque chromosome est doté d'une fonction d'adaptation (*fitness*) f_i , mesurant son adaptation qui est dans notre cas la performance par vote majoritaire (*MV performance*).

10.3.2. La recherche Tabou (Tabu search)

La recherche Tabou est une méthode itérative à solution unique basée sur un algorithme de recherche de voisinage qui commence avec une solution initiale pour l'améliorer pas à pas en choisissant une nouvelle solution dans son voisinage [7]. Elle a été introduite par F. Glover [49] et a montré sa performance sur de nombreux problèmes d'optimisation. Elle n'a aucun caractère stochastique et utilise la notion de mémoire pour éviter de tomber dans un optimum local.

Le principe de l'algorithme est le suivant : à chaque itération, le voisinage (complet ou sous-ensemble de voisinage) de la solution courante est examiné et la meilleure solution est sélectionnée. En appliquant ce principe, la méthode autorise de remonter vers des solutions qui semblent moins intéressantes mais qui ont peut être un meilleur voisinage.

Le risque est de cycler entre deux solutions. Pour éviter les phénomènes de cyclage, la méthode a l'interdiction de visiter une solution récemment visitée. Pour cela, une liste taboue contenant les attributs des dernières solutions visitées est tenue à jour. Chaque nouvelle solution considérée enlève de cette liste la solution la plus anciennement visitée.

Ainsi, la recherche de la solution courante suivante se fait dans le voisinage de la solution courante actuelle sans considérer les solutions appartenant à la liste taboue.

Cette méthode ne s'arrête pas d'elle-même et il faut déterminer un critère d'arrêt en fonction du temps de recherche que l'on s'octroie. Ce critère peut être, par exemple, l'exécution d'un certain nombre d'itérations ou la non-amélioration de la meilleure solution pendant un certain nombre d'itérations. Ainsi, tout au long de l'algorithme, la meilleure solution doit être conservée car l'arrêt se fait rarement sur la meilleure solution. Les différentes étapes de l'algorithme peuvent être décrites comme suit :

Algorithme Recherche Tabou

Procédure : φ fonction *fitness*

Variables locales : S solution courante, L liste tabou, M meilleure solution, K itération courante, N nombre d'itération

Paramétrages : Taille de la liste, Critère d'aspiration

Choix d'une solution initiale S_0 ;

Solution courante $S \leftarrow S_0$;

Meilleure solution $M \leftarrow S$;

$K \leftarrow 0$;

while $K < N$ **do**

$K \leftarrow K + 1$

 Muter le chromosome en un ou plusieurs points choisis aléatoirement

 Appliquer l'opérateur " *pairwise exchange* " une ou plusieurs fois

 Calculer la fonction *fitness* $\varphi(S)$

 Mise à jour de L

 Génération des candidats E par opération de voisinage

$C \leftarrow \text{best}(E)$

If ($\varphi(S) < \varphi(M)$) **OU** C n'est pas tabou **OU** C vérifie l'aspiration **then**

$S \leftarrow C$

else

$E \leftarrow E \setminus C$

end if

end while

return S

L'opérateur " *pairwise exchange* " accroît la capacité d'explorer l'espace de recherche. Il permet simplement d'échanger une paire de bits dans le chromosome tout en préservant le même nombre de classifieurs.

10.3.3. La recherche par accroissement basé sur la population (*PBIL*)

La méthode PBIL [11] est une méthode d'optimisation à population de solutions, prélevée d'un vecteur spécial de probabilités, qui est mis à jour à chaque étape selon les chromosomes les plus convenables. La mise à jour du vecteur de probabilités est effectuée selon une méthode appropriée d'apprentissage supervisé.

Etant donné le vecteur de probabilités $p = (p_1, p_2, \dots, p_M)$, et une population de chromosomes $G = (v_1, v_2, \dots, v_k, \dots, v_c)$, où $v_k = (v_{k1}, v_{k2}, \dots, v_{kM})$, chaque probabilité est mise à jour selon l'expression suivante :

$$p_j^{new} = p_j^{old} + \Delta p_j \quad (4.8)$$

avec

$$\Delta p_j = \eta \left(\frac{\sum_{k=1}^C v_{kj}}{C} - p_j \right) \quad (4.9)$$

où η est un paramètre de contrôle de la vitesse de convergence. L'algorithme converge si le vecteur de probabilités ne contient que des 0 et des 1. Dans ce cas, p devient la meilleure combinaison de classifieurs. L'algorithme PBIL peut être décrit par les étapes suivantes :

1. Créer le vecteur de probabilité de même longueur que les chromosomes de la population et l'initialiser avec des valeurs de 0.5 à chaque bit ;
2. Prélever un certain nombre de chromosomes selon le vecteur de probabilités ;
3. Evaluer la performance de chaque chromosome ;
4. Mettre à jour le vecteur de probabilités selon la formule (4.8) ;
5. Mettre à jours les chromosomes de la population ;
6. Si toutes les valeurs du vecteur de probabilités sont à 0 ou à 1 alors *fin*,

Sinon aller à l'étape 2.

L'algorithme PBIL n'utilise aucun opérateur génétique comme dans les algorithmes génétiques. Cependant, il contient un mécanisme spécifique permettant une exploitation bénéfique des informations d'une génération à une autre. Pour plus de détails, voir [11].

11. Conclusion

Dans ce chapitre, nous avons décrit un modèle hybride AG/PMC pour la reconnaissance de la parole. Le terme des modèles hybrides est couramment employé de nos jours, pour nommer des systèmes qui tentent d'allier les avantages de deux paradigmes différents. Nous avons décrit la manière de combiner les avantages des réseaux de neurones (approche locale par rétro-propagation du gradient) et des algorithmes génétiques (approche globale par algorithmes évolutionnaires), en vue de trouver un SRAP optimal par sélection basée sur le critère du vote majoritaire (*MV, Majority Voting*).

Conclusion et perspectives

Les modèles hybrides sont de plus en plus utilisés en reconnaissance de la parole, ils sont couramment employés de nos jours pour nommer des systèmes qui tentent d'allier les avantages de plusieurs méthodes différentes pour augmenter le taux de reconnaissance et arriver à des systèmes avec des performances comparables à celles des humains.

Dans le cadre de notre étude, nous nous sommes intéressés aux méthodes de classification connexionnistes de type PMC qui sont largement utilisées dans le domaine de la reconnaissance de la parole et ceci pour leurs capacités particulières d'apprentissage et de généralisation. Cependant, elles présentent certaines limites.

Notre contribution consiste donc à montrer l'apport des développements actuels que connaît l'IA pour remédier à ces limites. Nous avons pensé à l'utilisation d'une approche à hybridation neuro-génétique pour concevoir un système de reconnaissance de chiffres isolés parlés de l'arabe. Pour ce faire, nous avons entraînés une population de PMCs par algorithme génétique. D'autre part, un critère de sélection de type vote majoritaire (*Majority Voting*) est utilisé directement dans les algorithmes de sélection pour guider la recherche d'un classifieur optimal (un SRAP) de type PMC.

Ce travail ne présente qu'un point de départ, nous lui prévoyons des possibilités d'évolution :

- Implémenter le modèle hybride AG/PMC proposé et comparer les résultats de classification avec celles d'un classifieur neuronal classique traitant le même problème de reconnaissance.
- Afin de comparer les performances du système de reconnaissance proposé pour les différentes techniques d'analyse acoustique, les mêmes conditions d'expériences doivent être utilisées pour les différents tests (corpus d'apprentissage, nombre d'itérations, etc.), en utilisant les coefficients LPC, MFCC, PLP, ...etc, ainsi que des combinaisons éventuelles de ces derniers avec l'énergie (E) et le TPZ (taux de passage par zéro), ou même des combinaisons entre ces coefficients.
- Implémenter d'autres méthodes évolutionnaires, comme les stratégies d'évolution (*ES* pour *Evolutionary Strategies*), et la programmation génétique (*GP* pour *Genetic Programming*), d'une part dans l'objectif de trouver des taux de classification plus élevés

- et d'autre part, de bien cerner toutes les approches en vogue actuellement relevant des algorithmes évolutionnaires.
- Un problème essentiel relatif à l'architecture du réseau neuronal se pose dans la pratique, car aucune procédure formelle n'est disponible pour répondre aux questions (dans le cas du PMC) : combien faut-il mettre de couches de neurones, et combien de neurones par couches ? Il semble donc intéressant d'utiliser des algorithmes évolutionnaires (AE) pour le développement de réseaux de neurones en vue de trouver une architecture optimale et adéquate.
- D'autre part pour un vocabulaire étendu, il est intéressant d'utiliser des modèles de phonèmes au lieu de mots, du fait que le nombre de phonèmes qui permet la construction de n'importe quel mot présent dans la plupart des langages est faible, ce qui facilite l'entraînement avec des bases relativement petites.

Références bibliographiques

- [1] Yas Abbas Alsultanny, Musbah M. Aqel, "Pattern recognition using multilayer neural-genetic algorithm", *Neruocomputing* 51, pp. 237-247, 2003.
- [2] ACKLEY D., LITTMAN M., "Interactions between learning and evolution", pp. 487-509, 1992.
- [3] Enrique Alba, J. Francisco Chicano, "Training Neural Networks with GA Hybrid Algorithms", ed. GECCO 2004, Springer-Verlag, 2004.
- [4] E. N. Allen, M.D. Jose, "Annals of mathematics and artificial intelligence 5", 1992.
- [5] J. Anderson, E. Rosenfeld, "Neurocomputing: Foundations of research", MIT Press, Cambridge, Second printing, ISBN 0-262-01097-6, 1988.
- [6] B.S. Atal, S.L. Hanauer, "Speech analysis and synthesis by linear prediction of speech wave", *J.Acoust. Soc. Am.*, vol. 50, no. 2, pp. 637-655, 1971.
- [7] V. Bachelet. "Métaheuristiques parallèles hybrides : Application au QAP". PhD thesis, USTL LIFL France, 1999.
- [8] T. Back, Schwefel, "An overview of evolutionary algorithms for parameter optimization", *Evolutionary computation*, Vol. 1, pp. 1-23, 1993.
- [9] J. K. Baker, "The Dragon systems, an overview", IEEE, 1975.
- [10] Balakrishnan, Karthik, Honavar, Vasant. "Evolutionary Design of Neural Architectures - A Preliminary Taxonomy and Guide to Literature". Technical Report: ISU CSTR-95-01, AI group - Dept. Computer Science, Iowa State University. 1995.
- [11] S. Baluja. "Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning". Technical Report No. 163, Carnegie Mellon University, Pittsburgh PA, 1994.
- [12] Claude Barras, "Reconnaissance automatique de la parole : vers l'indexation automatique d'archives sonores", LIMSI-CNRS, article paru dans Orsay-Info n° 62, janvier 2001.
- [13] Beasley D., Bull D.R., Martin R.R., "An overview of genetic algorithms: Part1", *Research topics. University computing, UCISA*, (15)2, 58-69, 1993.
- [14] A. Belaïd, Y. Belaïd, "Reconnaissance des formes : méthodes et applications", Inter Edition, 1992.
- [15] Bellanger M., "Traitement numérique du signal, Théorie et pratique", éditions Masson, 1ère édition en 1980. Actuellement en 5ème édition, ISBN 2-225-84997-8, 1995.
- [16] N.Benahmed, "Optimisation de réseaux de neurones pour la reconnaissance de chiffres manuscrits isolés : sélection et pondération des primitives par algorithmes génétiques", Thèse de Maîtrise, Mars 2002.
- [17] C. M. Bishop, "Neural networks for pattern recognition", Clarendon Press, Oxford, 1995.
- [18] R. Boite, M. Kunt, "Traitement de la parole", Presses Polytechniques Romandes, 1987.
- [19] R. Boite, H. Boulard, T. Dutoit, J. Hancq, H. Leich, "Traitement de la Parole", Presses

Polytechniques et Universitaires Romandes, 2000.

- [20] C.L Bridges, D.E. Goldberg. "An analysis of multipoint crossover". In Proceedings of the Foundation of Genetic Algorithms. Foga, 1991.
- [21] L. Buniet. "Contribution à la reconnaissance des consonnes sonantes du français", Mémoire de DEA informatique, 74 pp, Université de Nancy-1, Nancy (France), 1991.
- [22] Burel G., "Réseaux de neurones en traitement d'images : des modèles théoriques aux applications industrielles". Thèse de doctorat de l'Université de Bretagne Occidentale, 1991.
- [23] Burr D.J., "Speech Recognition Experiments with Perceptrons". Neural Information Processing Systems, ed. D. Anderson, pp. 144-153, New-York : American Institute of Physics, 1988.
- [24] Caliope, J. P. Tibch, "La parole et son traitement automatique", Collection Technique et Scientifique des Télécommunications, Paris, Masson, 1989.
- [25] CHALMERS D. J., "The Evolution of Learning : An Experiment in Genetic Connectionism", TOURETZKY D., ELMAN J., SEJNOWSKI T., HINTON G., Eds., *Proc. of the 1990 Connectionist Models Summer School*, Morgan Kaufmann, San Mateo, CA, USA, 1990, also available as report 48 of the CRCC, Indiana University, Bloomington, IN 47405, USA.
- [26] S.-B. Cho. "Pattern recognition with neural networks combined by genetic algorithms". *Fuzzy Sets and Systems*, 103:339–347, 1999.
- [27] Claude Barras, "Reconnaissance de la parole continue : adaptation au locuteur et contrôle temporel dans les modèles de Markov cachés", Thèse, Mai 1996.
- [28] Pierre Combescure, Christel Sorin, "Le traitement du signal vocal", Rapport de Recherche, ENST, 1994.
- [29] Cooley J.W., Tukey J.W., "An algorithm for the machine calculation of complex fourier series", *Mathematics of Computation*, numéro 19, pp. 297-301, 1965.
- [30] E. Davalo, P. Naïm, "Des réseaux de neurones", Eyrolles, 2ème édition, Paris, 1993.
- [31] S. B. Davis et P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no 4, pp 357-366, 1980.
- [32] L. Davis, "Handbook of Genetic Algorithms", Van Nostrand Reinhold, New-York, 1991.
- [33] L. Davis, "Handbook of genetic algorithms", Ed. VNR, New York, 1992.
- [34] Decoux Benoît, "Reconnaissance de formes", 2001.
- [35] Stéphane DEKETELAERE, "Reconnaissance Automatique de la Parole", Rapport de recherche, MULTITEL, 2001.
- [36] G. Dreyfus, J. -M. Mez, M. Samuelides, M. B. Gordon, F. Badran, S. Thiria, L. Hérault, "Réseaux de neurones: méthodologie et applications", Groupe Eyrolles, ISBN : 2-212-11464-8, 2^{ème} édition, 2004.
- [37] T. Dutoit, "Je parle, donc je suis ?", Un bilan des développement récents en traitement automatique de la parole, Faculté Polytechnique de Mons, 2000.
- [38] Eiben A.E., Hinterding R., & Michalewicz Z., "Parameter Control in Evolutionary Algorithms". *Evolutionary Computation, IEEE Transactions on*, 3(2), pp. 124-141, 1999.

- [39] Elman, Jeffrey L. "Learning and Development in Neural Networks: The Importance of Starting Small". *Cognition*, 48(1993), pp.71-99. 1993.
- [40] Falhman S.E., "An Empirical Study of Learning Speed in Backpropagation Networks". Technical report CMU-CS-88-162, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1988.
- [41] Fiesler, Emile. "Comparative Bibliography of Ontogenic Neural Networks". Proceedings of the International Conference on Artificial Neural Nets - ICANN'94. Sorrento, Italy, May 1994.
- [42] Fiesler, E. & Beale, R. "Handbook of Neural Computation". Institute of Physics and Oxford University Press. New York, NY - U.S.A., 1997.
- [43] Fletcher, Justin Barrows S. "A Constructive Approach to Hybrid Architectures for Machine Learning". Ph.D. Thesis. Washington State University, August 1994.
- [44] FLOREANO D., "Emergence of Nest-Based Foraging strategies in Ecosystems of Neural Networks", MEYER J., ROITBLATT H., WILSON S., Eds., *Proc. SAB 2*, MIT Press, p. 410-416, 1993.
- [45] Floreano, Dario & Mondada, Francesco. "Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot". In: *From Animals to Animats III - Proceedings of the 3rd. International Conference on Simulation of Adaptive Behaviour*. MIT Press - Bradford Books, Cambridge - MA. 1994.
- [46] FLOREANO D., NOLFI S., "Learning and evolution", *Autonomous robots*, vol. 7, n 1, p. 89-113, 1999
- [47] Fogel. L., Owens. A., Walsh. M., "Artificial intelligence through simulated Evolutionary", John Wiley, 1966.
- [48] David B. Fogel, "Evolutionary Optimization", Orincon Corporation, IEEE, 1992.
- [49] F. Glover. "Future paths for integer programming and links to artificial intelligence". *Computers and Operations Research*, 13 : 533-549, 1986.
- [50] F. Glover, M. Laguna, "Tabu search", Kluwer Academic Publishers, Boston, 1997.
- [51] B. Gold, N. Morgan, "Speech and Audio Signal Processing", John Willey & Sons, 2000.
- [52] D. E. Goldberg, "Genetic Algorithm in search, optimization and machine learning", Addison-Wesley, 1989.
- [53] J. Greenstette, "Genetic Algorithms", IEEE, pp. 5-8, Oct., 1993.
- [54] M. Guivarch, "Les réseaux de neurones: techniques de base et apprentissage", 3ème université d'été. Réseaux connexionnistes en Informatique. Méthodes et Applications. Pp. II.2-II.31, Lyon, 1989.
- [55] R. Günter, "Convergence Analysis of canonical Genetic Algorithms", IEEE, Transaction on Neural Networks, Vol. 5, N°1, January, 1994.
- [56] Haeb-Umbach R., Geller D., Ney H., "Linear discriminant analysis for improved large vocabulary continuous speech recognition", ICASSP'92, pp.13-16, volume I, 1992.
- [57] J. P. Haton, J. M. Pierrel, G. Perennou, J. Caelen, J. L. Gauvain, "Reconnaissance automatique de la parole", Afcet Dunod, Bordas, Paris, 1991.

- [58] H. Hermansky, B. A. Hanson et H. Wakita. "Low-dimensional representations of vowels based on all-pole modeling in the psychophysical domain". *Speech Communication*, vol. 4, pp 181-187, 1985.
- [59] H. Hermansky. "Perceptual linear predictive analysis of speech". *Journal of the Acoustical Society of America*, vol. 87, no 4, pp 1738-1752, 1990.
- [60] H. Hermansky, N. Morgan, A. Bayya et P. Kohn. "Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP)". *Proceedings of the European Conference on Speech Communication and Technology*, pp 1367-1370, 1991.
- [61] H. Hermansky, N. Morgan, A. Bayya et P. Kohn. "RASTA-PLP speech analysis". Rapport technique TR-91-069, 6 pp, International Computer Science Institute, Berkeley (CA, États-Unis), 1991.
- [62] H. Hermansky, N. Morgan, A. Bayya et P. Kohn. "RASTA-PLP speech analysis technique". *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp 121-124, 1992.
- [63] H. Hermansky et N. Morgan. "RASTA processing of speech". *IEEE Transactions on Speech and Audio Processing*, vol. 2, no 4, pp 578-589, 1994.
- [64] Herrera F., Lozano M., & Verdegay J.L., "Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis". *Artificial Intelligence Review*. 12(4). 265-319, 1998.
- [65] Holland, John H. "Adaptation in Natural and Artificial Systems". University of Michigan Press. 1975.
- [66] Talib Hussain, "Methods of combining Neural Networks and genetic Algorithms", itrc97, tutorial paper, Queen's University, 1997.
- [67] L. Ingber, B. Rosen. "Genetic Algorithms and very fast simulated re-annealing. Mathematical computer Modelling", 16 (11):87--100, 1992.
- [68] N. Intrator, D. Reisfeld, Y. Yeshurun, "Face recognition using a hybrid supervised/unsupervised neural network", *IEEE*, pp. 3461-3466, 1996.
- [69] Janikow C.Z., & Michalewicz Z. , "An experimental comparison of binary and floating point representations in genetic algorithms". *Proceeding of the fourth international conference on Genetic Algorithms*, Morgan Kaufmann, pp. 31-36, 1991.
- [70] D. J. Janson, F. Frenzel, "Training product unit neural networks with genetic algorithm", *IEEE*, pp. 26-32, October, 1993.
- [71] F. Jelinek, "Continuous speech recognition by statistical methods", *Proc. of the IEEE*, vol. 64, no. 4, pp. 532-556, 1976.
- [72] Jodouin, Jean-François. "Les Réseaux de neurones : Principes et définitions". Editions Hermès, Paris, 1994.
- [73] Jodouin, Jean-François. "Les Réseaux Neuromimétiques : Modèles et applications". Editions Hermès, Paris, 1994.
- [74] Denis Jovet, "Reconnaissance de mots connectés indépendamment du locuteur par des méthodes statistiques", Thèse de Doctorat, ENST, 1988.
- [75] Juro Gottweis, "Pattern recognition using genetic algorithms", Technical Report, Comenius University, Bratislava / Slovakia, 2000.

- [76] M. Kabache, M. Guerti, "Application e neurones à la reconnaissance des phonèmes spécifiques à l'Arabe standard", IEEE SETIT 2005 Sciences Electroniques, Technologies de l'Information et des Télécommunications, Sousse, Tunisie, Mars 2005.
- [77] Christof Kech, "Computation and neural systems series", CNS, 1992.
- [78] Kitano Hiroaki: "Empirical Studies on the Speed of Convergence of Neural Network Training using Genetic Algorithms", in; *Eighth National Conference on Artificial Intelligence*, Vol. II, pp 789-795, AAAI, MIT Press, 1990.
- [79] Philipp Koehn, "Combining Algorithms and Neural Networks : The encoding Problem", Thesis, Dec 1994 .
- [80] Koza, John R. "Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)". MIT Press, 1992.
- [81] Kuncheva L.I., Whitaker C.J.: "Measures of Diversity in Classifier Ensembles". Submitted to Machine Learning.
- [82] L.I. Kuncheva and L.C. Jain. "Designing classifier fusion systems by genetic algorithms". *IEEE Ttansactions on Evolutionary Computation*, 4(4):327–336, 2000.
- [83] Labbi, Abderrahim. "Sur l'Approximation et les Systèmes Dynamiques dans les Réseaux Neuronaux". Thèse de Doctorat en Mathématiques Appliquées, Laboratoire LIFIA - IMAG, Grenoble - France, Décembre 1993.
- [84] R. LAMY, "Adaptation de modèles acoustiques et traitement des vecteurs acoustiques pour la reconnaissance automatique de la parole téléphonique", DEA ISC de l'Université Joseph Fourier de Grenoble, 2001.
- [85] K. F. Lee, "Large vocabulary speaker independant continious speech recognition: the SPHINX system", IEEE, 1990.
- [86] K.-F. Lee, H.-W. Hon & R. Reddy, "An overview of the SPHINX speech recognition system", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 1, pp. 35-45, 1990.
- [87] Lerman et Ngouenet, "Algorithmes génétiques séquentiels et parallèles pour une représentation affine des proximités", Rapport de Recherche de l'INRIA Rennes - Projet REPCO 2570, INRIA, 1995.
- [88] A. Leroi-Gourhan. "Le geste et la parole ; tome 1 : technique et langage". Collection sciences d'aujourd'hui, 324 pp, Éditions Albin Michel, 1992.
- [89] Liénard J.S., "Les processus de la communication parlée" – Masson, 1977.
- [90] P. Lippman, "Pattern classification using neural networks", IEEE, pp. 47-64, November, 1989.
- [91] LITTMAN M., "Simulations combining evolution and learning", in [17], p. 465–477, 1996.
- [92] J. Makhoul, "Linear prediction: a tutorial review", *Proc. of the IEEE*, vol. 63, no. 4, pp. 561-580, 1975.
- [93] Man K.F, & Tang K.S., "Genetic Algorithms for Control and Signal Processing". *Industrial Electronics, Control and Instrumentation IECON. 23rd International Conference on*, 4, 1541-1555, 1997.
- [94] Man K. F., Tang K. S., Kwong S., "Genetic algorithms: Concepts and designs." Springer, 2000.

- [95] J. D. Markel et A. H. Gray Jr. "Linear prediction of speech". 288 pp, Springer-Verlag, 1976.
- [96] W. S. McCulloch et W. Pitts. "A logical calculus of the ideas imminent in the nervous activity". *Bulletin of Mathematical Biophysics*, no 5, pp 115-133, 1943.
- [97] O. Mella, "Contribution à l'identification automatique du locuteur sur des critères acoustiques et phonétiques". Thèse de doctorat mention informatique, 244 pp, Université de Nancy 1, Nancy (France), 1993.
- [98] P. J. Melse, "Neural Networks: a conceptual overview", Technical Report, TCR-89-08, pp. 1-41, Aout, 1989.
- [99] Mihail CRUCIANU, "Algorithmes d'évolution pour les Réseaux de Neurones", Rapport de recherche 187, Fév 1997.
- [100] MITCHELLM., BELEW R., Eds., "Adaptive Individuals In Evolving Population Models", SFI Studies in the Sciences of Complexity, Addison-Wesley, 1996.
- [101] D-J-Montana, L. Davis, "Training Feed-Forward Neural Networks using genetic algorithms", PROCEEDINGS OF THE International Join Conference on Artificial Intelligent, San Mateo, Calif.:Morgan Kaufmann Publishers, 1989.
- [102] N. Morgan, H. Boulard, "Continuous speech recognition using multi-layer percpetrons with hidden Markov models", *Proceedings of the international conference on acoustics, Speech and signal processing*, pp. 413-416, April 1990.
- [103] N. Morgan, H. Hermansky, H. Boulard, P. Kohn et C. Wooters. "Continuous speech recognition using PLP analysis with multilayer perceptrons". *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp 49-52, May 1991.
- [104] N. Morgan et H. Hermansky. "RASTA extensions: robustness to additive and convolutional noise". *ESCA Technical Research Workshop: Speech processing in adverse conditions*, pp 115-118, 1992.
- [105] N. Morgan, M. A. Boulard, "Neural networks for statistical recognition of continous speech", *Processing of the IEEE*, Vol.83, N°5, Mai 1995.
- [106] A. Newell, J. Barnett, J. W. Forgie, C. C. Green, D. H. Klatt, C. R. Licklider, J. Munson, D. R. Reddy, W. A. Woods, "Speech Understanding Systems: Final Report of a study Group", North-Holland/American Elsevier, Amsterdam, 1973.
- [107] Nolfi, Stefano; Parisi, Domenico; Elman, Jeffrey L. "Learning and Evolution in Neural Networks". *Adaptive Behavior*, 3(1), pp.5-28. 1994.
- [108] PARISI D., NOLFI S., "The influence of learning on evolution", *in [17]*, p. 419-428, 1996.
- [109] D. Partridge and W.B. Yates. "Engineering multiversion neural-net systems". *Neural Computation*, 8:869- 893, 1996.
- [110] Patel, M. & Honavar, V. (Eds.) "Advances in Evolutionary Synthesis of Neural Systems". Boston, MA - U.S.A. MIT Press, 1997.
- [111] Peeling S.M. et Moore R.K., "Experiments in Isolated digit Recognition Using the Multi-layer Perceptron". Technical Report 4073, Royal Speech and Radar Establishment, Malvern, Worcester, Angleterre, 1987.
- [112] J.W. Picone, "Signal Modelling Techniques in Speech Recognition", *Proc. IEEE*, Vol 81,

N°9, pp 1215-1247, 1993

- [113] Porto, V. W. "Neural-Evolutionary Systems". In: Handbook of Neural Computation (section D2). E. Fiesler and R. Beale (Eds.) Institute of Physics and Oxford University Press. New York, NY - U.S.A., 1997.
- [114] Potter, Michell. "A Genetic Cascade-Correlation Learning Algorithm". In: Proceedings of the COGANN-92 International Workshop on Combinations of Genetic Algorithms and Neural Networks, pp.123-133, IEEE Computer Society Press, 1992.
- [115] J-M. Renders, "Algorithmes génétiques et réseaux de neurones: applications à la commande de processus". Ed. Hermes, Bruxelles, 1995.
- [116] Richard K.Belew, Jhon McInerney, Nicol N.Schraudolph, "Evolving Networks: using the Genetic Algorithm with connectionist Learning", Cognitive Computer Science, Research Group, California at San Diego, June 1990
- [117] F. Roli and G. Giacinto. "Hybrid Methods in Pattern Recognition", chapter Design of multiple classifier systems, pages 199–226. World Scientific Publishing, 2002.
- [118] Ruta D., Gabrys B.: "Analysis of the Correlation Between Majority Voting Errors and the Diversity Measures in Multiple Classifier Systems". Accepted for the International Symposium on Soft Computing SOCO'2001.
- [119] D. Ruta and B. Gabrys. "Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting". In *Proceedings of the 2nd International Workshop on Multiple Classifier Systems*, number LNCS 2096 in Lecture Notes in Computer Science, pages 399–408, Cambridge, UK, 2001. Springer Verlag.
- [120] Salomon, Ralf & van Hemmen, Leo. "Accelerating Backpropagation through Dynamic Self-Adaptation". *Neural Networks*, vol.9(4), pp.589-601, Pergamon Press. 1996.
- [121] Sarle, Warren S. *Comp.ai.neural.nets FAQ* (Usenet Frequently Asked Questions -Neural Nets). Web: <http://www.cis.ohio-state.edu/hypertext/faq/usenet/ai-faq/neural-nets/top.html> . Ftp: <ftp://ftp.sas.com/pub/neural/FAQ.html>
- [122] J. Schultz, J. Greenstelle, "Test and evaluation by genetic algorithms", IEEE, pp. 9-14, Oct, 1993.
- [123] Schwefel. H. P., "Numerical Optimization of computer Models", Wiley, (1981).
- [124] Udo Seiffert, "Multiple Layer Perceptron Training Using Genetic Algorithm", ESANN'2001 proceedings – European Symposium on Artificial Neural Networks, pp. 159-164, Bruges (Belgium), April 2001.
- [125] S. Seneff, "A joint/mean-rate model of auditory speech processing", *Journal of Phonetics*, vol. 16, no. 1, pp. 55-76, 1988.
- [126] Sharkey A.J.C., Sharkey N.E.: "Combining Diverse Neural Nets". *The Knowledge Engineering Review* 12(3) (1997) 231-247
- [127] A.J.C. Sharkey, N.E. Sharkey, U. Gerecke, and G.O. Chandroth. "The 'test and select' approach to ensemble combination". In J. Kittler and F. Roli, editors, *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, number LNCS 1857 in Lecture Notes in Computer Science, pages 30–44, Cagliari, Italy, 2000. Springer-Verlag.
- [128] O'Shaughnessy D., "Speech Communication, human and machine"-Addison Wesley, 1987.
- [129] Singer H., "Utilisation de dissyllabes pour la reconnaissance de la parole" - Rapport LIMSI, 88-4, 1988.

- [130] Siohan O., "On the robustness of linear discriminant analysis as preprocessing for noisy speech recognition", ICASSP'95, pp. 125-128, volume I, Detroit (USA), 1995.
- [131] Anne Spalanzani, "Algorithmes évolutionnaires pour l'étude de la robustesse des systèmes de RAP", Thèse, 1999
- [132] G. Syswerda, "Uniform cross over in génétic algorithms", Third Conference on genetic algorithms, pp. 2-9, Morgan Kauffmann, 1989.
- [133] Szilas, Nicolas. "Apprentissage dans les Réseaux Récurrents pour la Modélisation Mécanique et Etude de leurs Interactions avec l'Environnement". Thèse de Doctorat en Sciences Cognitives, Laboratoire LIFIA/LEIBNIZ - IMAG, Grenoble - France, Décembre 1995.
- [134] Taboada J., Feijoo S., Balsa R., Hernandez C., "Explicit estimation of speech boundaries", IEEE Proc. Sci. Meas. Technol., vol. 141, pp. 153-159, 1994.
- [135] R. Tlemsani, N. Neggaz, A. Benyettou, "Amélioration de l'Apprentissage des Réseaux Neuronaux par les Algorithmes Evolutionnaires : application à la classification phonétique", IEEE SETIT 2005 Sciences Electroniques, Technologies de l'Information et des Télécommunications, Sousse, Tunisie, Mars 2005.
- [136] URZELAI J., "Evolutionary Adaptive Robots : artificial evolution of adaptation mechanisms for autonomous systems", PhD thesis, EPFL, Lausanne, Suisse, 2000.
- [137] Dominique Vaufreydaz, "Modélisation statistique du langage à partir d'Internet pour la RAPC", Thèse, 2002.
- [138] D. Wang, "Pattern recognition: neural networks in perspective", pp. 52-60, IEEE, August, 1993.
- [139] Widrow, Bernard & Lehr, M. "30 Years of Adaptive Neural Networks : Perceptron, Madaline, and Back-Propagation". Proceedings of the IEEE, New York, Vol.78, N.9, pp.1415-1441. September 1990.
- [140] B. Yoon, D-J. Holmes, G. Langholz, A. Kandel, "Efficient genetic algorithms for training layerd feedforward neural networks", Information Sciences76, pp. 67-85, 1994.

NOTES

A large rectangular area containing numerous horizontal dotted lines, intended for writing notes.