

DÉPARTEMENT DE MATHÉMATIQUES

MÉMOIRE DE FIN D'ÉTUDES

Pour l'obtention du diplôme de MASTER MATHÉMATIQUES ET APPLICATIONS AU CALCUL SCIENTIFIQUE

Présenté par

SOUAD DAKIR

sous la direction du Pr. Mohamed ETTAOUIL

Thème:

Sélection des modèles neuronales par test croisé, Application au problème de la reconnaissance des locuteurs.

soutenu le 17 juin 2017

Devant le Jury:

Pr. Fatima EZZAKI	Faculté des Sciences et Techniques Fès	Examinateur
Pr. Mohamed ETTAOUIL	Faculté des Sciences et Techniques Fès	Encadrant
Pr. Rachid EL AYADI	Faculté des Sciences et Techniques Fès	Examinateur
Pr. Rachid EL KHAOULANI	Faculté des Sciences et Techniques Fès	Président

DÉDICACE

À mes chers parents

Aucun mot ne saurait exprimer tout mon amour et toute ma gratitude. Merci pour vos sacrifices le long de ces années.

Merci pour votre présence rassurante et pour tout l'amour que vous procurez à notre petite famille. Vous avez toujours été pour moi les parents idéals, la lumière qui me guide dans les moments les plus obscures. Vous m'avez toujours conseillé et orienté dans la bonne voie du travail et de l'honneur. Votre droiture, conscience et amour pour nos familles me serviront d'exemple dans la vie.

En témoignage de profonds liens qui nous unissent, veuillez chers parents trouver à travers ce travail l'expression de mon grand amour, mon attachement et ma profonde reconnaissance.

Puisse votre existence pleine de sagesse, d'amour me servir d'exemple dans ma vie et dans l'exercice de ma profession.

Puisse Dieu vous prêter longue vie et bonne santé afin que je puisse vous combler à mon tour. **Je vous aime très fort.**

À mes chers frères et sœurs et leurs petites familles

Qui ont toujours été là pour moi, avec leurs précieux conseils et encouragements. Qu'ils trouvent dans ce travail, toute ma reconnaissance et tout mon grand amour envers eux.

Je vous aime beaucoup.

À mes chers amis

Je vous dis que c'est grâce à vous que je suis à ce niveau, un merci spécial à vous, **Je vous aime beaucoup.**

REMERCIEMENTS

Au terme de ce travail, je voudrais exprimer mes remerciements et ma profonde reconnaissance à tout ceux qui ont contribué de prêt ou de loin à sa réalisation.

Je voudrais, en tout premier lieu, exprimer ma profonde reconnaissance à mon encadrant, professeur **ETTAOUIL MOHAMED**, d'avoir accepté de m'initier à la recherche et de diriger ce travail. Comme tous ceux qui ont eu la chance d'être sous sa direction, j'ai pu constater à quel point il est attachant, attentionné et profondément gentil. Il restera pour moi un modèle, pour ses grandes compétences scientifiques, pour son exigence de clarté, sa franchise et ses qualités humaines. Je lui adresse mes remerciements les plus chaleureux, d'avoir accepté de me prendre sous son ail, pour tout ce qu'il m'a appris et pour ses encouragements, et j'espère dans l'avenir pouvoir transmettre ne serait-ce-qu'une partie de tout ce qu'il m'a appris, la meilleure façon selon moi de lui rendre hommage.

j'aimerais bien adresser des chaleureux remerciements à tous les membres du jury : Madame **Fatima EZZAKI**, professeur directrice de laboratoire LMCS à la faculté des sciences et techniques de Fès, Monsieur **EL KHAOULANI EL IDRISSI Rachid** professeur à la faculté des sciences et techniques de Fès et Monsieur **Rachid EL AYADI** professeur à la faculté des sciences et techniques de Fès, qui m'ont fait l'honneur de bien vouloir juger ce travail et de l'enrichir par leurs remarques et leurs critiques.

Je je tiens à remercier également mon cher collègue M. **Elhoussaine HSSAYNI**, et les doctorants, M.**Hassan RAMCHOUN**, et M.**Zakariae EN-NAIMAN** qui n'ont épargné ni temps ni effort pour m'aider à concrétiser ce travail.

Je tiens également à remercier le corps professoral de ma formation à la Faculté des Sciences et Techniques de Fès pour leur contribution à ma formation et pour leurs enseignements précieux et multidisciplinaires.

TABLE DES MATIÈRES

Ta	able des figures		8		
Lis	ste de	es abrév	iations		9
ln	trodu	ction gé	énérale		10
1	Rés	eaux de	neurones	s artificiels et apprentissage	13
	Intro	duction			13
	1.1	De neu	ırone biolo	ogique vers le neurone formel	13
	1.2	Topolo	gies des p	principaux types des réseaux de neurones	16
		1.2.1	Topologi	e des réseaux de neurones	
		1.2.2	Principau	ux types de réseaux de neurones	17
			1.2.2.1	Adaline et Madaline	17
			1.2.2.2	Perceptron multi-couches	19
			1.2.2.3	Réseau de Hopfield	
			1.2.2.4	Réseau de Kohonen	21
	1.3	Proprié	été fondan	mentale du Perceptron multi-couches	22
			1.3.0.5	Approximation universelle	23
			1.3.0.6	Parcimonie	23
			1.3.0.7	De l'approximation de fonction à la modélisation statistique .	23
	1.4	Apprer	itissage de	e Perceptron multi-couches	24
		1.4.1		ı de coût	
		1.4.2	Algorithr	ne d'apprentissage	
			1.4.2.1	Rétro-propagation du gradient (back-propagation)	
		1.4.3	Surajuste	ement	29
	Con	clusion .			29
2			ge statist	- ·	30
	Intro				
	2.1			pervisé et non supervisé	
	2.2	Régres	sion et cla	assification	31

	2.3 2.4 2.5	Perform	gme d'apprentissage supervisé	31 32 32
	2.6		tion des risques et choix du modèle	34
		2.6.1	Sélection du modèle	34
			2.6.1.1 sélection des variables	34
			2.6.1.2 Validation croisée	35
			2.6.1.3 Régularisation	36
			2.6.1.3.1 arrêt prématuré (ou early-stopping)	36
			2.6.1.3.2 modération des poids (ou weight decay)	37
		2.6.2	Procédure standarde de conception d'un modèle	37
	Cond			38
3			Estimation des performances du modèle neuronale	39 39
	3.1		nination du nombre de neurones cachés d'un réseau de neurones par vali-	39
	0.1		croisée	39
	3.2		tion des performances d'un modèle – Méthode du test croisé	40
	3.3		nentation de la méthode de test croisé pour le problème de classification	
		Iris Fis		41
		3.3.1	Description de la base des données	41
		3.3.2	Conception du modèle	42
		3.3.3	Normalisation et pré-traitement des données	42
		3.3.4	Implémentations et résultats numériques	43
			3.3.4.1 Apprentissage sans régularisation	43
			3.3.4.2 Apprentissage avec régularisation	44
			3.3.4.3 Résultats obtenus en testant le modèle sélectionné	44
	Cond	clusion .		45
4	C	2		46
4 Système de reconnaissance automatique du locuteur théorie et application		reconnaissance automatique du locuteur theorie et application	40	
	4.1		onnaissance du locuteur théorie et principes	46
		4.1.1	Introduction à la reconnaissance automatique du locuteur	46
			4.1.1.1 Identification et vérification du locuteur	47
			4.1.1.2 Mode dépendant et indépendant du texte	48
			4.1.1.3 Sources d'erreurs	48
			4.1.1.4 Système de reconnaissance automatique du locuteur	48
		4.1.2	Analyse acoustique du signal de parole	49
			4.1.2.1 Production et perception du signal vocal	49
			4.1.2.2 Coefficients cepstraux	50
		4.1.3	Modélisation des locuteurs	51
		•	4.1.3.1 Approche vectorielle	51
			4.1.3.2 Approche statistique	52
			4.1.3.3 Approche connexionniste	53
			4.1.3.4 Approche relative	53
		4.1.4	Décision et mesure des performances	53

4.2 Reco	nnaissance du locuteur par les réseaux de neurones, protocole expérimental		
et résultats			
4.2.1	LPCC ou Linear Prediction Cepstral Coefficients	54 54	
4.2.2	Description de la base des données	54	
4.2.3	Paramétrisation des données	55	
4.2.4	Prédiction par PMC	55	
4.2.5	Implémentations du test croisé et quelques résultats numériques	57	
	4.2.5.1 Apprentissage sans régularisation	57	
	4.2.5.2 Apprentissage avec régularisation	58	
	4.2.5.3 Commentaires	59	
Conclusion		59	
Conclusion et	Perspectives	61	
Bibliographie		62	

TABLE DES FIGURES

1.1 1.2 1.3 1.4 1.5 1.6 1.7	Neurone biologique(a) et sa formalisation mathématique(b). Quelques fonctions d'activation Classification des types de réseaux de neurones selon leur topologie. Réseau Adaline (ADaptive LInear NEuron). Schéma général du perceptron. Un réseau de neurones à une couche cachée. Réseau entièrement connecté : modèle de Hopfield Réseau entièrement connecté par couches : modèle de Kohonen Règle de Hebb.	14 16 18 19 20 20 21 22 26
2.1	Modèles ayant : un grand biais et une faible variance (en haut), un faible biais et une forte variance (sur-ajustement) (au milieu), un bon compromis biais/variance	
2.2	(en bas). Principe de la validation croisée : les plis bleus sont utilisés pour la validation, les autres pour l'apprentissage	33 36
2.3	Principe de la méthode d'arrêt prématuré.	37
3.1 3.2 3.3	Schéma résumant les différentes étapes de la construction d'un modèle. Iris de Ficher. Scores de validation croisée des réseaux, présentés pour trois ensembles de test,	40 42
3.4	en fonction du nombre de neurones cachés qu'ils comportent. Scores de validation croisée des réseaux, présentés pour trois ensembles de test, en fonction du nombre de neurones cachés qu'ils comportent.	43 44
3.5	Erreur d'apprentissage et celle de test effectué sur le modèle choisi en fonction des itérations.	45
4.1	Traitement de la parole	47
4.2	Schéma typique d'un système de vérification du locuteur.	49
4.3	L'appareil phonatoire	49
4.4	Calcul des coefficients cepstraux avec une échelle Mel.	50
4.5	Caractères de Tifinagh adoptés.	55
4.6	Le prédicteur PMC pour la reconnaissance du locuteur.	56

4.7	Les étapes principales de la reconnaissance du locuteur.	57
4.8	Scores de validation croisée des réseaux, présentés pour trois ensembles de test,	
	en fonction du nombre de neurones cachés qu'ils comportent.	58
4.9	Scores de validation croisée des réseaux, présentés pour trois ensembles de test,	
	en fonction du nombre de neurones cachés qu'ils comportent.	59

LISTE DES ABRÉVIATIONS

DTW: DYNAMIC TIME WARPING

GMM: GAUSSIAN MIXTURE MODELS

HMM: HIDDEN MARCOV MODELS

LPC: LINEAR PREDICTION COEFFICIENTS

LPCC: LINEAR PREDICTION CEPSTRAL COEFFICIENTS

MFCC: MEL FREQUENCY CEPSTRAL COEFFICIENTS

RAL: RECONNAISSANCE AUTOMATIQUE DES LOCUTEURS

RNA: RESEAUX DE NEURONES ARTIFICIELS

PMC: PERCEPTRON MULTI-COUCHES

INTRODUCTION GÉENÉRALE

Durant la dernière décennie, on a observé l'apparition de plus en plus de domaines et d'applications nécessitant l'authentification des personnes : contrôle d'accès à des bâtiments sécurisés, transmission de données personnelles, recherche et surveillance d'individus, contrôle aux frontières, justice, domotique... Ces applications utilisent généralement des technologies d'authentification basées sur la biométrie. On parle dans ce cas là d'une authentification biométrique. L'authentification biométrique est la reconnaissance automatique d'une personne en utilisant des traits distinctifs, i.e., des caractéristiques physiques (biologiques) ou traits comportementaux personnels automatiquement mesurables, robustes et distinctifs qui peuvent être utilisés pour identifier ou vérifier l'identité prétendue d'un individu. Certaines technologies biométriques constituent une solution efficace relativement simple et pas chère qui assure de bonnes performances en reconnaissance. Elles offrent beaucoup plus d'avantages que les méthodes classiques d'authentification telles que l'utilisation de mots de passe ou de clés et badges d'accès, qui sont très vulnérable au vol et à la falsification. Les technologies biométriques fournissent en effet encore plus de sureté et de confort d'utilisation.

On trouve deux grandes tâches en reconnaissance : l'identification et la vérification. Dans la première, le système biométrique pose et essaye de répondre à la question suivante : qui est la personne X? Le système requiert une information biométrique et la compare avec chaque information stockée dans la base de données des utilisateurs autorisés; c'est une comparaison parmi d'autres. Une des applications d'identification est la recherche des criminels et de terroristes en utilisant des données de surveillance. Dans la vérification, le système biométrique demande à l'utilisateur son identité et essaye de répondre à la question : est ce la personne X?. Dans cette tâche, l'utilisateur annonce son identité par l'intermédiaire, par exemple, d'un numéro d'identification ou d'un nom d'utilisateur, puis le système sollicite une information biométrique provenant de cette personne pour la comparaître avec la donnée préalablement enregistrée qui correspond à l'identité prétendue. C'est une comparaison vraie/fausse. Le système trouvera finalement ou non une correspondance entre les deux. La vérification est communément employée dans des applications de contrôle d'accès et de paiement par authentification.

Plusieurs informations biométriques ont été utilisées dans diverses applications et secteurs, à savoir : l'empreinte digitale, la main, l'iris, la rétine, le visage, la voix, les veines, la signature ... Ces modalités se distinguent par leur fiabilité, le niveau d'acceptation par l'utilisateur, le cout et l'effort de mise en œuvre. Des modalités comme l'iris ou la rétine donnent les meilleurs taux de

reconnaissance en comparaison avec les autres informations biométriques. Cependant elles sont peu acceptées par les utilisateurs et sont lourdes à mettre en place. Selon le niveau de sécurité requis, les ressources financières et les contraintes techniques d'utilisation, une modalité peut être privilégiée par rapport à une autre.

Les réseaux de Neurones Artificiels constituent un outil très puissant pour traiter de nombreuses applications. Ils ont prouvé à de nombreuses reprises leurs efficacités dans plusieurs domaines de recherche tels que : l'analyse et compression d'images, problème de classification, reconnaissance de formes, reconnaissance de la parole, reconnaissance de l'écriture, analyse de signal,commande de processus,robotique, et recherche sur le Web. Parmi les propriétés attractives pour un réseau de neurones, la capacité d'apprendre de son environnement, d'améliorer sa performance à travers un processus d'apprentissage.

Dans les années 40, les travaux de McCullch et Pitts sont l'origine des réseaux de neurones artificiels[80],ces travaux montraient qu'assembler des entités de calcul simples permettent de réaliser des opérations complexes. Dans les années 50, les premières machines ont été apparues, ces machines sont capables d'auto-régler le calcul qu'elles réalisesnt à partir des exemples qu'on leur présente. C'est l'apparition de perceptron et de l'Adaptive Liner Neuron ADALINE. Dans les années 80, le net regain aux réseaux de neurones artificiels grâce au modèle de Hopfield et de ses applications : mémoires auto associatives. Puis la machine de Boltzmann et au perceptron multi-couches. Dans les années 90, Kohonen a proposé diverses variantes pour la classification et quantification vectorielle dont les algorithmes de quantification vectorielle à l'apprentissage. Les propriétés importantes d'un réseau de neurones ont sûrement la capacité d'apprendre de son environnement, d'améliorer sa performance à travers un processus d'apprentissage. Le problème d'ajustement d'une fonction à un nombre fini de point est le risque que le réseau apprenne le bruit sur la fonction et non plus la fonction et nous nous trouvons face à un problème de surapprentissage ou surajustement. Pour résoudre ce problème, plusieurs méthodes ont été présentées comme celles de régularisation telle que l'arrêt prématuré (early stopping) ou les méthodes de pénalisation.

La détermination de l'architecture neuronale influence la convergence des algorithmes d'apprentissage : le choix du nombre des couches cachées et des neurones artificiels en chaque couche, la fonction d'activation...etc. Les paramètres optimaux de l'architecture des réseaux de neurones artificiels sont ceux qui donnent une erreur minimale en considération du temps de convergence de processus d'apprentissage. Dans ce travail, on va étudier la méthode de validation croisée comme étant une méthode statistique pour la sélection de modèle neuronal.

Dans notre projet on va s'intéresser à l'application des réseaux de neurones à la reconnaissance du locuteur.

Ce manuscrit s'articule principalement autour de 5 chapitres :

Le chapitre 1 est consacré à un tour d'horizon de la littérature sur les travaux concernant les réseaux de neurones artificiels. Ce tour d'horizon révèle toute la richesse et la variété des travaux consacrés à ce domaine, qu'il a permis de répertorier de multiples domaines d'applications ainsi que plusieurs types de méthodes de résolution. Le modèle de base présenté est commun aux

différentes architectures. Ces architectures définissent les réseaux de neurones artificiels. Nous présentons l'algorithme d'apprentissage pour la correction des erreurs produites.

Le chapitres 2 concerne la présentation des méthodes statistiques mises en œuvre au cours de cette étude, et quelques notions d'apprentissage statistique.

Dans le chapitre 3, nous avons bien traité les méthodes statistiques de la validation croisée et du test croisé pour la sélection de nombre de neurones cachés et l'hyper-paramètre de régularisation, nous avons les testé par le problème de classification d'IRIS dont on a présenté les résultats obtenus.

Le chapitre 4 est consacré à un état de l'art sur la reconnaissance du locuteur (RAL) en l'abordant dans ses aspects acoustique et phonétique. Cela dit, nous allons faire un traitement d'extraction des paramètres d'un signal parole qui nous permettra de faire la reconnaissance de ces paramètres en utilisant un réseau de neurones de type perceptron multi-couches.

Enfin, le dernier chapitre est consacré aux résultats obtenus en appliquant la méthode de test croisé sur le problème de reconnaissance du locuteur.

Afin de traiter ce problème, nous avons crée notre propre base en enregistrant quelques lettres amazighes pour faire la classification selon le genre du locuteur.

CHAPITRE 1

RÉSEAUX DE NEURONES ARTIFICIELS ET APPRENTISSAGE

Introduction

La mise au point des systèmes intelligents capables de modéliser et de copier le fonctionnement du cerveau humain suscite un engouement général dans le monde scientifique depuis des années. Il n'est donc pas surprenant qu'une technologie basée sur des fondements biologiques tels que les réseaux de neurones retienne particulièrement l'attention. Cependant, avant d'appliquer les réseaux de neurones artificiels (ou formels) pour résoudre un problème complexe, il est indispensable de se familiariser avec les principales notions de base et ce qui est le but de ce chapitre.

À la suite d'un rappel sur le système nerveux et les neurones biologiques, les différents types de neurones formels et les modes d'apprentissage sont présentés et ce par ordre chronologique de parution dans la littérature scientifique. Cette classification nous permet d'une part de mieux comprendre l'évolution considérable du domaine des réseaux de neurones formels ces dernières années, et d'autre part de présenter les exemples de réseaux de neurones les plus pertinents en présentant leurs structures, leurs algorithmes de fonctionnements, leurs qualités et de ce fait leurs domaines d'application.

1.1 De neurone biologique vers le neurone formel

Un bref rappel sur les neurones biologiques nous permettra de mieux comprendre l'origine des réseaux de neurones formels. Le système nerveux central est composé d'un ensemble de cellules nerveuses fortement inter-connectées, leur nombre est estimé à une centaine de milliards. Les neurones sont des cellules particulières capables de recevoir, de traiter et de transmettre des messages sous forme d'impulsions électriques. Chaque neurone est constitué d'un corps cellulaire, d'une multitude de petites ramifications, appelées dendrites, et d'une ramification unique beaucoup plus longue appelée axone, ce qui est représenté à la figure 1.1(a). La connexion inter-neuronale se fait entre l'axone et les dendrites, par l'intermédiaire d'une synapse, c'est le

nombre de neurones et les connexions inter-neuronales qui définissent le réseau de neurones.

Le mode de fonctionnement d'un neurone biologique est le suivant : il reçoit des impulsions électriques provenant des neurones auxquels il est connecté; à partir de tous ces signaux, le neurone peut ou non émettre un message, dans l'affirmative, il le fait par l'intermédiaire de son axone et la synapse qui est le lieu de passage des impulsions. Une synapse peut donc être excitatrice ou inhibitrice. Le niveau d'excitation ou d'inhibition correspond à une pondération des influx nerveux arrivant des dendrites. On considère que le corps cellulaire du neurone réalise la somme pondérée de tous les influx nerveux, un signal sera émis si cette somme est supérieure à un seuil.

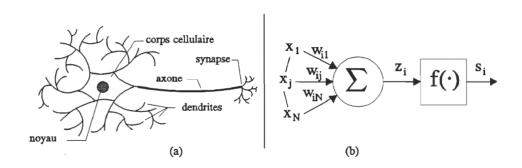


Figure 1.1 – Neurone biologique(a) et sa formalisation mathématique(b).

En se basant sur les fondements biologiques du système nerveux, il est possible d'envisager un réseau artificiel qui serait lui aussi composé de cellules de calcul, appelées neurones artificiels ou formels et de connexions entre ces cellules. Nous intéressons dans un premier temps aux cellules de traitement. Par analogie avec les neurones biologiques, ces cellules doivent être aptes à réaliser les trois opérations suivantes : collecter, traiter et transmettre des messages. Pour satisfaire ces exigences, le neurone formel doit posséder plusieurs entrées pour collecter les messages provenant des autres neurones formels, pour que ceux-ci soient pondérés le neurone formel doit posséder des poids de pondération au niveau de chacune de ses entrées ; une partie de traitement modélisée par une fonction mathématique appelée fonction de transition ; une sortie unique qui transmet le message résultant du traitement des signaux d'entrée aux autres neurones formels. Nous avons alors le parallèle suivant entre le neurones formel et le neurone biologique : les dendrites sont des entrées, les synapses des poids de connexion et l'axone la sortie unique. Ceci est montré dans la figure 3.1 où l'on retrouve un neurone biologique (figure 1.1(a)) et son homologue formel (figure 1.1(b)).

Mathématiquement on peut transcrire cette modélisation sous la forme de l'équation suivante, les connexions se font dans le sens des neurones d'indice j vers le neurone d'indice i :

$$s_i = f(z_i) = f(\sum_{i=1}^{N_j} w_{ij} x_j - \theta_i)$$
 (1.1)

οù

```
-x_i: Entrée élémentaire du neurone j;
```

- $-z_i$: Entrée totale du neurone i;
- $-s_i$: Sortie du neurone i;
- $-\theta_i$: Seuil du neurone *i*
- w_{ii} : Poids de connexion entre deux neurones i et j;
- f(.): Fonction de décision non linéaire;
- $-N_i$: le nombre de neurones j connectés au neurone i;

Dans un modèle plus général de neurone formel, on définit la fonction de décision comme étant constituée par une fonction d'entrée totale et une fonction d'activation. Le premier modèle de neurone formel a été présenté en 1943 par Mc Culloch et Pitts, c'est un automate à seuil dont l'unité de sortie est binaire, la fonction de décision utilisée est une fonction à seuil. Ce neurone ne peut donc qu'être actif (sortie à 1) ou inactif (sortie à 0). D'autres modèles ont été obtenus en modifiant la fonction de décision, en général ces fonctions sont plus complexes et elles permettent d'obtenir des neurones plus performants. Les modèles les plus courants sont au nombre de six ([8]) :

- Automates booléens : la sortie est une fonction booléenne des entrées elles-mêmes booléennes.
- Automates à seuil : la sortie est binaire, les entrées binaires ou réelles, la fonction d'entrée est affine, la fonction d'activation est la fonction signe ou de Heavyside.
- Automates linéaires : la sortie et les entrées sont réelles, la fonction d'entrée est linéaire,
 la fonction d'activation est l'identité.
- Automates à saturation : la sortie et les entrées sont comprises dans un intervalle [u,v],
 les fonctions d'entrée et d'activation sont linéaires.
- Automates continus : la sortie et les entrées sont réelles, la fonction d'entrée est linéaire ou affine, la fonction d'activation est sigmoïde.
- Automates probabilistes : la sortie est binaire, les entrées quelconques, la fonction d'entrée est linéaire ou affine, la fonction d'activation est stochastique

Les différents modèles sont donc caractérisés par leur fonction de décision (fonctions d'entrée et d'activation), les cinq fonctions d'activation les plus fréquemment employées par les réseaux de neurones sont les suivantes [34] :

- 1. fonction linéaire : $f(x) = \alpha x$ où α est un nombre positif.
- 2 fonction à seuils :

$$f(x) = \begin{cases} \beta & \text{si } x \ge \theta \\ -\delta & \text{si } x < \theta \end{cases}$$

où les valeurs de β et δ définissent les niveaux de seuil, la variable θ est le seuil prédéfini.

3 fonction à rampe

$$f(x) = \begin{cases} \gamma & \text{si } x \ge \gamma \\ x & \text{si } |x| < \gamma \\ -\gamma & \text{si } x \ge -\gamma \end{cases}$$

où γ est la valeur de saturation de la fonction et les points $x=\gamma$ et $x=-\gamma$ sont les points de discontinuité de la fonction.

4 fonction sigmoïde

$$f(x) = \frac{1}{1 + \exp \alpha x} \tag{1.2}$$

où $\alpha > 0$, en général $\alpha = 1$, ce qui maintient la sortie dans l'intervalle [0,1].

5 fonction gaussienne

$$f(x) = \exp^{\frac{-x^2}{\nu}} \tag{1.3}$$

où x est la moyenne et ν est la variance prédéfinie.

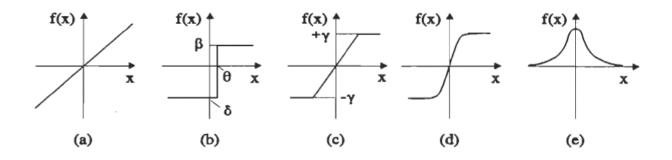


Figure 1.2 – Quelques fonctions d'activation

1.2 Topologies des principaux types des réseaux de neurones

Nous verrons dans cette section l'un des cruciaux principes des réseaux de neurones tels que la topologie. Ensuite, nous présentons les différents types de ces réseaux qui existent dans la littérature.

1.2.1 Topologie des réseaux de neurones

La topologie des réseaux de neurones est définie par les trois points suivants : les données traitées par le réseau [34], l'arrangement dans l'espace des cellules de traitement (neurones formels) et l'interconnexions entre ces neurones. Dans la section précédente nous avons défini les neurones, cellules élémentaires des réseaux de neurones, il convient donc d'étudier maintenant les modes d'arrangements et d'interconnexions possibles entre eux. Tout comme le neurone formel est une modélisation mathématiques du neurones biologique, le réseau formel d'interconnexions est basé sur le réseau de neurones biologique formant le système nerveux central. S'il existe un parallèle entre une arborisation de neurones biologiques et un schéma d'interconnexions entre neurones formels, il est à noter que la densité en neurones est infiniment plus grande dans le cas biologique que dans le cas artificiel. Nous avons donc introduit la notion de topologie des réseaux de neurones, définissons maintenant la terminologie associée à cette dernière :

1. Couche:

Les réseaux de neurones sont organisés en couches de neurones formels qui correspondent

à l'arrangement dans l'espace de ces derniers. D'un point de vue théorique des graphes, les réseaux à couches sont des graphes sans circuit.

2. Connexions:

On distingue deux types de connexions dans les réseaux de neurones : les connexions intra-couches relient les neurones d'une même couche, voire un neurone sur lui même ; les connexions inter-couches relient les neurones de couches différentes. Les deux types sont compatibles et il est possible de voir des réseaux possédant les deux types de connexions.

3. Boucles:

Une boucle est formée par un neurone connecté sur lui-même directement ou par l'intermédiaire de plusieurs neurones. D'un point de vue théorique des graphes, les réseaux bouclés constituent des graphes ayant au moins un circuit.

4. Propagation et rétro-propagation :

On distingue deux types de réseaux de neurones selon leur aptitude à propager une information dans un sens unique entre les neurones ou leur aptitude à propager une information dans un sens et rétro propager une information complémentaire dans l'autre sens. Dans ce dernier cas les neurones sont appelés à fonctionner dans les deux sens.

Les définitions précédentes nous permettent de classifier les réseaux de neurones selon leur topologie, cependant d'autres critères de classification sont intéressants pour parfaire la classification, tel que :

- Le nombre de couches,
- Le mode de connexion des réseaux de neurones à boucles : réseaux entièrement connectés et réseaux entièrement connectés par couches,
- Le type de couches simples ou non : les réseaux à couches simples possèdent plusieurs couches et les connexions se font uniquement d'une couche i vers la couche i+1. S'ils sont non simples, les réseaux ont des connexions de n'importe quelle couche vers n'importe quelle autre.

L'arbre de la figure 1.3 permet de faire une synthèse des classifications envisagées sur les réseaux de neurones selon leur topologie. Le choix de classement a été fait d'une part sur l'existence ou non des boucles dans le réseau et d'autres part sur le nombre de couches, celles-ci pouvant-être cachées ou non et les connexions entre couches simples ou non. Certaines classes de réseaux sont complétées par des exemples connus de réseaux de neurones, ceux-ci seront étudiés dans la suite du chapitre.

1.2.2 Principaux types de réseaux de neurones

Dans cette section, nous donnons un survols sur les différents réseaux de neurones tels que,les Adaline et Madaline, le Perceptron multi-couchesles, les réseaux de Hopfield et les réseaux de Kohonen, et

1.2.2.1 Adaline et Madaline

Proposés par B. Widrow en 1960, les modèles de réseaux de neurones adaline (ADAptive Linear NEuron) et madaline (multi-adaline) sont des réseaux à deux couches, de structure ana-

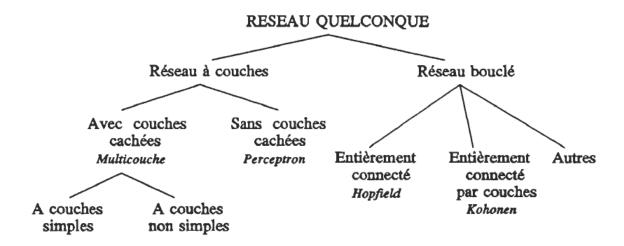


Figure 1.3 – Classification des types de réseaux de neurones selon leur topologie.

logue au perceptron, ayant respectivement un ou plusieurs neurones sur la deuxième couche (couche de sortie), un réseau adaline est proposé à la figure 1.4. Contrairement au perceptron qui délivre un signal binaire en sortie et pour lequel la correction d'erreur se fait par comparaison entre sorties, obtenue et voulue, binaires, l'adaline et le madaline comparent des sorties analogues, ce qui affine beaucoup la correction des poids.

L'équation de sortie d'un réseau adaline, résultant de la propagation d'un signal d'entrée dans les deux couches du réseau, est donnée par :

$$s_i = f(\sum_j w_{ij} x_j - \theta_i) \tag{1.4}$$

οù

 $-s_i$: la sortie binaire du réseau

z_i : la sortie analogue du réseau

− x_i : une entrée élémentaire

 $-\theta_i$: le seuil du neurone i

 $-w_{ij}$: le poids de connexion

f(.): la fonction de transition (fonction seuil)

La règle de correction des poids, appelée règle de Widrow-Hoff du nom de ses inventeurs ou encore règle du Delta, se base sur un signal d'erreur E_i qui est la différence entre la sortie désirée et la somme pondérée des entrées non seuillées $E_i = d_i - z_i$ où d_i est la sortie désirée, la règle de modification des poids est alors : $\Delta w_{ii} = kx_i E_i$ où k une constante.

Ce choix de l'erreur permet d'obtenir d'une part une solution approchée aux problèmes non linéairement séparables et d'autre part une solution plus robuste car elle tend vers un minimum local de la fonction d'erreur.

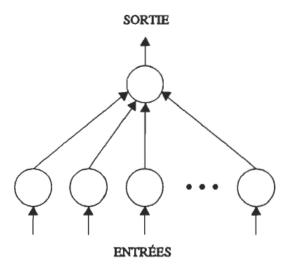


Figure 1.4 – Réseau Adaline (ADaptive Linear NEuron).

1.2.2.2 Perceptron multi-couches

Le perceptron est le premier modèle de réseau de neurones, il a été présenté à la fin des années 1950 par Rosenblatt. Ce réseau est constitué de neurones formels de Mc Culloch et Pitts il met en application la règle de Hebb, c'est un réseau à une seule couche de poids adaptatifs.

Le perceptron multi-couches est un réseau orienté de neurones artificiels organisés en couches, où l'information voyage dans un seul sens da la couche d'entrée vers la couche de sortie. Chaque couche du réseau regroupe un ensemble de neurones artificiels dont le nombre peut varier d'une couche à une autre. L'organisation générale de perceptron est composée de trois éléments :

- une retine : elle comprend les cellules sur lesquelles s'inscrit le stimulus.
- une ou plusieurs couches de cellules d'association : chacune des cellules est connectée à des cellules de la rétine (dans le sens rétine vers couche d'association), ou à d'autres cellules d'association ou à des cellules de décision.
- une couche de cellules de décision : chacune des cellules est connectée à des cellules d'association (dans le sens couche d'association vers couche de décision), ou à d'autres cellules de décision.

Les cellules d'association ou de décision réalisent la somme des impulsions reçues et répondent suivant une loi du tout-ou-rien après une comparaison avec un seuil, ou elles donnent une réponse modulée. La figure 1.5 propose un schéma général du perceptron, par la suite nous nous limiterons à l'étude d'un perceptron simplifié. Dans ce cas les sorties de toutes les cellules sont binaires, les fonctions des cellules d'association sont booléennes,

les cellules d'association sont des automates à seuil et les connexions se font uniquement d'une cellule d'association vers une cellule de décision.

Dans un tel réseau, le flux de l'information circule des entrées vers les sorties sans "retour en arrière". Ainsi, si l'on représente le réseau comme un graphe dont les sont les neurones nœuds et les arêtes les connexions entre ceux-ci, le graphe d'un réseau non bouclé est acyclique.

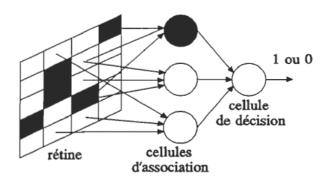


Figure 1.5 – Schéma général du perceptron.

Nous utilisons dans ce travail ce type des réseaux, qui réalisent une fonction (linéaire ou non) d'une combinaison linéaire de fonctions neurones (appelées « neurones cachés ») des entrées du réseau :

$$y(x_0,...,x_n) = \varphi[\sum_{i=0}^{j} w_{j+1,i}\varphi_i(\sum_{k=0}^{n} w_{ik}x_k)]$$

où $\{x_0,...,x_n\}$ sont les entrées du réseau (avec $x_0=1$), $\{\varphi_i,i=1,...,j\}$ sont les sorties des neurones cachés (avec $\varphi_0=1$), w_{kl} est le paramètre associé à la variable du neurone . Si la

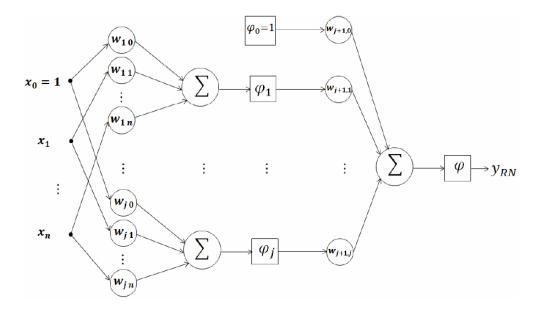


Figure 1.6 – Un réseau de neurones à une couche cachée.

fonction $\varphi(.)$ est la fonction identité, et si les fonctions $\varphi_i(.)$ sont des fonctions sigmoïdes, le réseau est un approximateur universel parcimonieux comme on va définir dans les parties suivantes.

1.2.2.3 Réseau de Hopfield

La grande limitation d'utilisation des réseaux à deux couches, en partie due à l'incapacité de corriger les poids de connexion pour des couches cachées supplémentaires, est responsable d'une baisse d'intérêt général pour le domaine des réseaux de neurones dans les années 70. Au début des années 80, des modèles de réseaux bouclés constituent le renouveau dans le domaine des réseaux de neurones, ce sont respectivement le modèle de Hopfield qui est une réseau entièrement connecté et le modèle de Kohonen qui est un réseau entièrement connecté par couches.

Le principe de base du modèle de Hopfield consiste à faire évoluer le réseau vers un état stable, à partir d'un état initial quelconque. Le réseau est susceptible de posséder de nombreux états stables et de créer des bassins attracteurs autour de ceux-ci. L'approche de Hopfield est basée sur le fait que le système nerveux recherche des états stables, attracteurs, dans son espace d'états. A ce moment, un état voisin tend à se rapprocher de l'état stable qui lui est le plus proche, ce qui permet la correction des erreurs et la capacité à compléter des informations manquantes. Les réseaux de Hopfield sont des mémoires adressables par leur contenu, ou mémoires auto-associatives, c'est-à-dire qu'un état mémorisé est retrouvé par une stabilisation du réseau. De plus si on supprime une fraction des neurones, les attracteurs sont peu modifiés. Cette robustesse rappelle celle du cerveau et s'explique par le fait que l'information est répartie dans tout le réseau. D'un point de vue structural, ce réseau est constitué de neurones formels de Mc Culloch et Pitts

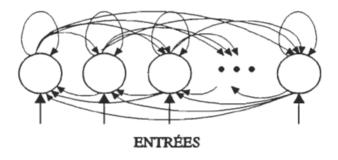


Figure 1.7 – Réseau entièrement connecté : modèle de Hopfield

totalement interconnectés. Les poids de connexion w_{ij} entre un neurone i et un neurone j se présentent sous forme d'une matrice W. Dans le cas d'un modèle simple, on considère que $w_{ij} = w_{ji}etw_{ii} = 0$ (il n'y a pas de boucles sur le neurone lui-même). La fonction d'activation utilisée est la fonction signe, une modélisation du réseau est proposée à la figure.

1.2.2.4 Réseau de Kohonen

Ce modèle se distingue par l'utilisation d'un neurone formel spécifique, une structure de réseau relativement complexe et surtout un mode d'apprentissage de type supervisé. Le neurone formel utilisé est caractérisé par le fait qu'il évolue d'un état à un autre en passant par une série de transitions rapides. La stabilisation de l'état de sortie nécessite donc un temps plus important que le temps de variation des entrées. On a alors une limitation sur la rapidité de variation des états du réseau. La loi de fonctionnement du neurone formel est la suivante : $s = f(\sum_i w_i x_i)$ où

- s : la sortie

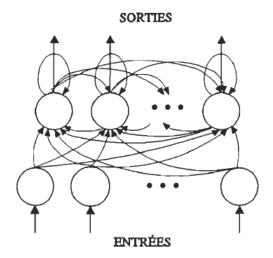


Figure 1.8 – Réseau entièrement connecté par couches : modèle de Kohonen

 $- w_i$: un poids synaptique

x_i: une entrée élémentaire

-f(.): une fonction sigmoïde

La structure du réseau est représentée à la figure 1.8, c'est un réseau entièrement connecté par couches à une seule couche. On peut considérer ce réseau comme un réseau entièrement connecté, relié en amont avec une couche de cellules d'entrée. On distingue dans ce modèle des connexions internes et externes, les poids de connexions sont stockés dans deux matrices distinctes. Le mécanisme d'apprentissage est le suivant :

- si le neurone i reçoit un potentiel d'action de la part d'un neurone j et qu'il s'active, alors la connexion entre ces deux neurones est renforcée.
- si le neurone i ne reçoit pas de potentiel d'action, ou qu'il ne s'active pas en ayant reçu un potentiel d'action, alors le poids de la connexion diminue.

L'équation d'apprentissage proposée par Kohonen est :

$$\frac{dw_i}{dt} = ksx_i - \beta(s)w_i \tag{1.5}$$

où le terme $\beta(s)w_i$ est un terme d'oubli, $\beta(s) \geq 0$ est la fonction d'oubli et k un paramètre en général supérieur à zéro. Ceci permet d'avoir un mécanisme d'apprentissage qui tient compte des phénomènes d'oubli qui sont proportionnels à l'inactivité des neurones.

1.3 Propriété fondamentale du Perceptron multi-couches

La proprété fondamentale des réseaux de neurones est *l'approximation parcimonieuse*. Cette expression traduit deux propriétés distinctes : d'une part, les réseaux de neurones sont des approxiamateurs universels, et d'autre part, une approximation à l'aide de réseau de neurones nécessite, moins de paramètres ajustables que les approximateurs usuels.

1.3.0.5 Approximation universelle

La propriété d'approximation universelle [7] [15] peut s'énoncer de la façon suivante :

Toute fonction bornée suffisamment régulière peut être approchée uniformément, avec une précision arbitraire, dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche cachée en nombre fini, possédant tous la même fonction d'activation, et un neurone de sortie linéaire.

C'est cette propriété qui justifie notre choix de l'architecture de réseaux de neurones à une couche cachée. De plus, le seul degré de liberté qui subsiste pour la détermination de l'architecture du réseau est alors le nombre de neurones cachés, ce qui simplifie l'optimisation de l'architecture de réseaux, comme nous le verrons plus loin.

1.3.0.6 Parcimonie

Lorsque l'on veut modéliser un processus à partir des données, on cherche toujours à obtenir les résultats les plus satisfaisants possibles avec un nombre minimum de paramètres ajustables. Dans cette optique, [23] a montré que :

Si le résultat de l'approximation (c'est-à-dire la sortie du réseau de neurones) est une fonction non linéaire des paramètres ajustables, elle est plus parcimonieuse que si elle est une fonction linéaire de ces paramètres. De plus, pour des réseaux de neurones à fonction d'activation sigmoidale, l'erreur commise dans l'approximation varie comme l'inverse du nombre de variables de la fonction à approcher. Par conséquent, pour une précision donnée, donc pour un nombre de neurones cachés donné, le nombre de paramètres du réseau est proportionnel au nombre de variables de la fonction à approcher.

Ces résultats s'appliquent aux réseaux de neurones à fonction d'activation sigmoidale, puisque la sortie de ces neurones n'est pas linéaire par rapport à leurs coefficients. Ainsi, l'avantage des réseaux de neurones par rapport aux approximateurs usuels (tels que les ploynômes) est d'autant plus sensible que le nombre de variables de la fonction à approcher est grand : pour des problèmes faisant intervenir une ou deux variables, on pourra utiliser indifféremment des réseaux de neurones, des polynômes, des réseaux d'ondelettes, etc. En revanche, pour des problèmes présentant trois variables ou plus, il est généralement avantageux d'utiliser des réseaux de neurones.

Bien entendu, cette propriété est démontrée d'une manière générale, et peut se révéler inexacte pour un problème particulier. Elle consiste néanmoins une justification fondamentale de l'utilisation des réseaux de neurones, et elle s'est avérée dans la très grande majorité des problèmes pratiques.

1.3.0.7 De l'approximation de fonction à la modélisation statistique

Les problèmes que l'on rencontre en pratique ne sont que très rarement des problèmes d'approximation de fonction *connue*. Dans la très grande majorité des cas, on cherche à établir un

modèle à partir de mesures, ou, en d'autres termes, à trouver la fonction qui passe "au plus près" (en un sens qui sera précisé plus loin) d'un nombre fini de points expérimentaux, généralement entachés, c'est-à-dire la fonction que l'on obtiendrait en calculant la moyenne d'une infinité de mesures effectuées en chaque point du domaine de validité du modèle. Le nombre de points de ce domaine étant lui-même infini, la connaissance de la fonction de régression nécessiterait donc une infnité de mesure en un nombre infini de points.

Les réseaux de neurones, en raison de leur propriété fondamentale, sont de bons candidats pour réaliser une approximation de la fonction de régression. C'est ce qui justifie l'utilisation réelle des réseaux de neurones : la recherche d'une approximation de la fonction de régression à partir d'un nombre fini de points.

L'utilisation des réseaux de neurones entre donc complétement dans le cadre des méthodes statistiques : les méthodes de recherche d'une approximation de la fonction de régression. De telles méthodes de recherche d'une approximation de la fonction régression *linéaires*. L'apport des réseaux de neurones réside donc dans leur capacité à approcher des fonctions *non linéaires*.

1.4 Apprentissage de Perceptron multi-couches

La principale qualité des réseaux de neurones est leur capacité d'apprentissage, dans ce contexte, l'apprentissage est défini comme la capacité d'emmagasiner une information qui peut être rappelée par la suite. Les connaissances d'un réseau connexionniste sont mémorisées dans les poids de connexion, ces poids sont fixés lors d'un processus d'apprentissage. Il est important de comprendre que les poids de connexion sont totalement dépendants les uns des autres, en effet, pris séparément ils n'ont aucune signification. Ce sont les poids de connexion et l'ensemble du réseau qui définissent les capacités de ce dernier. L'apprentissage est fait en fonction des trois mécanismes suivants :

- 1. Développement de nouvelles connexions inter-neuronales.
- 2. Suppression de connexions inter-neuronales inactives.
- 3. Modification des poids de connexion inter-neuronaux existants.

À l'heure actuelle les points (1) et (2) ont été peu développés et peuvent être considérés comme des cas particuliers du cas (3) selon le changement apporté à la valeur des connexions, pour le cas (1) une connexion initialement à "0" devient positive ou négative et dans le cas (2) une connexion initialement positive ou négative devient "0". La phase d'apprentissage est capitale dans le fonctionnement d'un réseau de neurones puisqu'elle consiste à fixer les poids de connexions afin que le réseau de neurones devienne apte à réaliser la tâche qui lui est assignée. On distingue différents types d'apprentissage qui sont directement liés à la topologie du réseau de neurones :

- Supervisé: des associations entrée/sortie sont présentées au réseau par un professeur, celui-ci doit déterminer, selon la nature du problème, la règle d'apprentissage la plus appropriée et le nombre de modèles à présenter.
- Non supervisé : cet apprentissage ne concerne que les réseaux ne disposant que des informations d'entrée et qui les regroupent suivant un certain critère dans le but de créer des classes.

- Avec prototypes ou de nombreux exemples : cet apprentissage concerne les réseaux qui nécessitent des algorithmes d'apprentissage itératifs. Les modifications des poids sont successives et dépendent du choix de présentation des modèles.
- Par empreinte ou par cœur : dans ce cas la mémorisation est parfaite et définitive dés la première présentation de l'association à apprendre.
- Hors ligne : le réseau est entrai né durant la phase d'apprentissage à l'aide d'un ensemble de données qui permettent de fixer les poids de connexion. Une fois la phase d'apprentissage achevée, le réseau est fonctionnel et il peut être utilisé sans modifications pour mener à bien la tâche qui lui a été assignée.
- En ligne : contrairement à l'apprentissage hors ligne, le réseau est en phase d'apprentissage continuelle, c'est-à-dire que toute action faite par le réseau est prise comme un élément d'apprentissage. Le réseau apprend et est apte à s'ajuster tout en réalisant la tâche qui lui est assignée.

En fonction de ces caractéristiques, de nombreuses règles d'apprentissage ont été proposées, chaque type de réseau possède une ou plusieures règles d'apprentissage et compte tenu du nombre de types de réseaux de neurones et de la diversité des algorithmes d'apprentissage proposés dans la littérature, il est impensable de tous les étudier. Aussi nous allons nous restreindre aux réseaux et algorithmes les plus représentatifs des dernières décennies. Cependant la grande majorité des règles d'apprentissage sont basées sur la règle de Hebb proposée en 1949 par [22], cette règle est basée sur des études biologiques, selon le principe suivant : Lorsqu'une connexion entre deux cellules est très forte, si la cellule émettrice est activée alors la cellule réceptrice l'est aussi. Pour lui permettre de jouer un rôle déterminant lors du mécanisme d'apprentissage, il faut donc augmenter le poids de cette connexion. En revanche, si la cellule émettrice est activée sans que la cellule réceptrice le soit, la connexion n'est pas prépondérante et on peut laisser un poids faible à cette connexion.

La règle formelle de Hebb est alors : Si deux neurones connectés sont activés en même temps, la connexion qui les relie doit être renforcée. Dans un cas contraire elle n'est pas modifiée. Ce principe est illustré à la figure 4.4. La règle de Hebb peut être traduite mathématiquement de la façon suivante :

$$w_{ij}(t+1) = w_{ij}(t) + \mu A_t A_j \tag{1.6}$$

Оù

- w_{ij} le poids de connexion entre les neurones i et j,
- $-A_i$ l'activation du neurone i,
- $-\mu$ un paramètre d'apprentissage.

D'autre règles d'apprentissage ont été proposées par la suite, on peut considérer qu'il existe deux grandes familles qui diffèrent par leur source d'inspiration :

- source biologique: dans ce cas l'étude des modèles biologiques est à l'origine des modèles formels, par exemple les premières étapes des systèmes visuels animaux permettent de proposer des machines aptes à la reconnaissance des formes.
- source mathématique: des réseaux connexionnistes déjà existants sont considérés comme des fonctions de transfert des entrées, il faut alors calculer les paramètres de cette fonction. Les algorithmes qui en résultent sont basés sur des minimisations de fonctions de coûts ou des techniques d'algèbre linéaire.

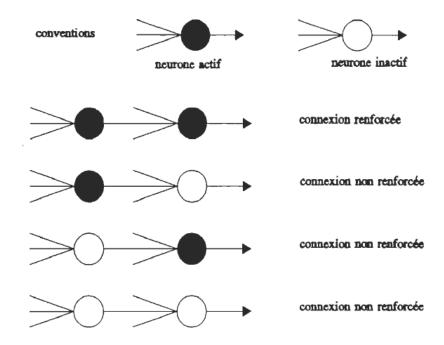


Figure 1.9 – Règle de Hebb.

Il est nécessaire de n'utiliser qu'une partie de la base de donnés, préalablement centrées réduites, pour effectuer l'apprentissage du réeau et sélectionner le nombre de neurones cachés l'ensemble d'apprentissage/validation. Les autres exemples constituent l'ensemble de test : ils sont utilisés pour estimer les capacités de généralisation du modèle.

l'apprentissage du réseau est réalisé par minimisation d'une fonction de coût.

1.4.1 Fonction de coût

L'erreur quadratique moyenne (EQM) est la moyenne, sur tous les exemples de l'ensemble d'apprentissage, des carrés des erreurs de modélisation, c'est-à-dire des carrés des différences entre les valeurs de la grandeur d'intérêt prédites par le modèle $\varphi(x_i)$ et ses valeurs observées expérimentalement y_i :

$$J = \frac{1}{N} \sum_{i=1}^{N} (y_i - \varphi(x_i))^2$$
 (1.7)

L'erreur quadratique moyenne est le plus souvent utilisée lorsque le modèle doit prédire une grandeur continue. Dans le cadre d'un problème de classification à deux classes, la grandeur à prédire est une variable binaire : 0 si l'exemple appartient à la classe C_0 ou 1 si l'exemple appartient à la classe C_1 . Néanmoins, si l'on estime les paramètres d'un réseau de neurones à partir de ces données, celui-ci fournit une estimation de la probabilité à posteriori de la classe C_1 [21],[25],[32].

1.4.2 Algorithme d'apprentissage

Les algorithmes d'apprentissage ont pour objectif de permettre à la machine d'acquérir des connaissances opérationnelles à partir d'exemples. Ils cherchent à découvrir la structure cachée d'un ensemble de signaux à partir des valeurs observées, les facteurs inhérents qui expliquent les variations et les dépendances observées. Ils sont utiles pour la prédiction, la prise de décision, et l'analyse des données. Les réseaux de neurones multi-couches sont habituellement bâtis selon le modèle normalisé et comprennent 3 ou 4 couches en tout (donc 1 ou 2 couches cachées). S'il est théoriquement possible de construire des réseaux avec un très grand nombre de couches cachées, les réseaux comprenant plus de couches cachées sont très rares, étant donné que chaque nouvelle couche augmente la quantité de calculs d'une manière exponentielle. La plupart des réseaux de neurones multi-couches sont, dans la pratique, des perceptrons multi-couches (PMC).

1.4.2.1 Rétro-propagation du gradient (back-propagation)

L'algorithme de la rétro-propagation du gradient est une généralisation de la règle de Widrow-Hoff (voir section 1.2.2.1) pour les réseaux multi-couches. L'idée est de trouver une fonction d'erreur qui ne travaille pas en tout-ou-rien. La solution consiste à changer le neurone formel en utilisant une fonction d'activation dérivable (en général sigmoïde) en remplacement de la fonction à seuil. Il faut alors trouver un mécanisme simple de calcul des dérivées composées. Le principe de fonctionnement du réseau est qu'un signal d'entrée est propagé vers la sortie et l'erreur obtenue entre la sortie désirée et la sortie obtenue est rétro-propagée en suivant le chemin inverse, ce mécanisme est répété jusqu'à l'obtention d'un résultat satisfaisant. Pour réaliser la phase d'apprentissage complète, on dispose d'un ensemble d'exemples qui sont des couples entrée/sortie désirée, la rétro-propagation est appliquée à chaque modèle, à la fin du processus lorsque l'erreur totale est inférieure au seuil choisi, le réseau a convergé.

Le neurone applique une fonction sigmoïde à la somme pondérée de ses entrées, un exemple de fonction est présenté à la figure 4.2(d). Ce choix d'une fonction sigmoïde est motivé par la nécessité de pouvoir calculer la dérivée de l'erreur quadratique par rapport à chaque poids de pondération. Ce principe fonde les méthodes de type algorithme du gradient, qui sont efficacement utilisées dans des réseaux de neurones multi-couches comme les perceptrons multi-couches. L'algorithme du gradient a pour but de converger de manière itérative vers une configuration optimisée des poids synaptiques. Cet état peut être un minimum local de la fonction à optimiser et idéalement, un minimum global de cette fonction (dite fonction de coût).

Normalement, la fonction de coût est non linéaire au regard des poids synaptiques. Elle dispose également d'une borne inférieure et moyennant quelques précautions lors de l'apprentissage, les procédures d'optimisation finissent par aboutir à une configuration stable au sein du réseau de neurones. Voici les principales lignes de la méthode de rétro-propagation :

Algorithme : Les poids dans le réseau de neurones sont au préalable initialisés avec des valeurs aléatoires. On considère ensuite un ensemble de données qui vont servir à l'apprentissage. Chaque échantillon possède ses valeurs cibles qui sont celles que le réseau de neurones doit à terme prédire lorsqu'on lui présente le même échantillon. L'algorithme se présente comme ceci :

- Soit un échantillon \vec{x} que l'on met à l'entrée du réseau de neurones et la sortie recherchée pour cet échantillon \vec{t}
- On propage le signal en avant dans les couches du réseau de neurones : $x_k^{(n-1)} \mapsto x_i^{(n)}$
- ullet La propagation vers l'avant se calcule à l'aide de la fonction d'activation g, de la fonction d'agrégation h (souvent un produit scalaire entre les poids et les entrées du neurone) et des poids synaptiques \vec{w}_{jk} entre le neurone $x_k^{(n-1)}$ et le neurone $x_j^{(n)}$. Attention au passage à cette notation qui est inversée, \vec{w}_{jk} indique bien un poids de kvers i.

$$x_j^{(n)} = g^{(n)}(h_j^{(n)}) = g^{(n)}(\sum_k w_{jk}^{(n)} x_k^{(n-1)})$$

- Lorsque la propagation vers l'avant est terminée, on obtient à la sortie le résultat \vec{y}
- On calcule alors l'erreur entre la sortie donnée par le réseau \vec{v} et le vecteur \vec{t} désiré à la sortie pour cet échantillon. Pour chaque neurone i dans la couche de sortie, on calcule:

$$e_i^{sortie} = g'(h_i^{sortie})[t_i - y_i]$$

• On propage l'erreur vers l'arrière $e_i^{(n)} \mapsto e_j^{(n-1)}$ grâce à la formule suivante : $e_j^{(n-1)} = g'^{(n-1)}(h_j^{(n-1)}) \sum_i w_{ij} e_i^{(n)}$

$$e_j^{(n)} = \sum_i [t_i - y_i] \frac{\partial y_i}{\partial h_i^{(n)}}$$

 $e_j^{(n)} = \sum_i [t_i - y_i] \frac{\partial y_i}{\partial h_j^{(n)}}$ • On met à jour les poids dans toutes les couches : $w_{ij}^{(n)} = w_{ij}^{(n)} + \lambda e_i^{(n)} x_j^{(n-1)} \text{ où } \lambda \text{ représente le taux d'apprentissage (de faible magnitude et infériour à 1)}$ et inférieur à 1)

La grande majorité des PMC sont des réseaux avec apprentissage supervisé, mais il en existe quelques-uns qui peuvent apprendre sans professeur pour les guider. Une classe de ces réseaux non-supervisés est celle des réseaux auto-associatifs. Par défaut, un PMC est un réseau hétéroassociatif, ce qui veut dire que la sortie est différente de l'entrée. Dans le cas particulier où l'entrée doit être la même que la sortie, ces réseaux étant appelés auto-associatifs, on n'a pas besoin des valeurs de sortie que devraient produire le PMC, celles-ci étant les mêmes que l'entrée. Ces PMC auto-associatifs ont ceci d'intéressants que la première partie du réseau constitue un codage des données, alors que la seconde partie est le décodage, pour retrouver l'information entrée.

Quelques améliorations ont été proposées pour augmenter la vitesse d'apprentissage des PMC:

- 1. L'utilisation de la fonction tangente hyperbolique, $y=\frac{2a}{1+exp(-bx)}-a$ où a et b sont constants. Les valeurs de a=1.716 et b=0.667 sont efficaces selon Guyon. L'avantage de cette fonction par rapport à la fonction sigmoïde de base a été testé empiriquement.
- 2. L'utilisation d'un terme de Momentum dans l'équation de la loi delta :

$$\Delta w_{ij}(p) = \beta \times \Delta w_{ij}(p-1) + \alpha \times y_i(p) \times \delta_k(p)$$
 (1.8)

avec $0 < \beta < 1$ (une bonne valeur de β est 0.95). Cette nouvelle équation porte le nom de loi delta généralisée. L'ajout du terme de momentum permet de stabiliser l'apprentissage, tout en augmentant la descente de la courbe de la somme des erreurs carrées selon les itérations.

3. L'utilisation d'un taux d'apprentissage variable plutôt que constant. Si le changement dans la somme des erreurs carrées porte le même signe algébrique, le taux d'apprentissage devrait être augmenté. Si le signe algébrique du changement de la somme des erreurs carrées alterne d'une itération à l'autre, le taux d'apprentissage devrait être décrémenté.

1.4.3 Surajustement

Comme nous l'avons indiqué dans le paragraphe précedent, toute fonction suffisamment régulière peut être approchée avec une erreur arbitraire par un réseau de neurones décrit plus haut, possédant un nombre fini de neurones cachés.

Cependant, le problème du surajustement présenté dans le chapitre qui suit se pose, il est étudié dans le cas des réseaux de neurones, par [17].

Dans cette étude, contraints par le petit nombre d'exemples disponibles, nous avons adopté la méthode de régularisation par la modération des poids dont le paramètre de régularisation λ a été déterminé par validation croisée.

Conclusion

Dans ce chapitre nous avons défini les différentes topologies et modèles des réseaux de neurones, nous avons introduit la notion d'apprentissage par l'algorithme de rétro-propagation du gradient.

CHAPITRE 2

APPRENTISSAGE STATISTIQUE

Introduction

L'apprentissage statistique est un ensemble d'algorithmes dont le but est de construire des modèles à partir d'un ensemble de données expérimentales (ou, dans certains cas, de données simulées). Ces modèles sont uniquement *prédictifs* et non *explicatifs* : ils s'opposent en cela aux modèles de connaissances, qui, reposent sur l'utilisation d'équations fondées sur les principes de la physique, de la chimie, de la biologie,... sont à la fois prédictifs et explicatifs. On attend des modèles construits par apprentissage statistique qu'ils rendent compte des observations à partir desquelles ils ont été conçus, mais surtout qu'ils soient capables de généraliser, c'est-à-dire de prédire des observations qui n'ont pas été utilisées pour leur conception.

2.1 Apprentissage supervisé et non supervisé

On distingue deux types de situations selon qu'une grandeur à expliquer ait été, ou non, observée conjointement aux variables descriptives.

Le premier cas est un problème d'apprentissage supervisé : à l'aide d'un ensemble de données X, on cherche une fonction capable de reproduire au mieux la grandeur à expliquer observée Y, selon un critère à définir. Autrement dit : $Y = \varphi(X)$

Le second cas, dans lequel on ne dispose pas de variable à expliquer, est un problème d'apprentissage non supervisé. La finalité recherchée est alors la taxinomie des observations : les regrouper en classes homogènes les plus dissemblables possibles entre elles. C'est un problème d'agrégation (clustering).

N'ayant mis en œuvre que l'apprentissage supervisé dans notre travail, nous développons ce sujet dans les paragraphes qui suivent.

2.2 Régression et classification

L'apprentissage supervisé peut, lui aussi, être divisé en deux catégories, selon le type de la variable à expliquer s'elle est continue ou discrète.

Lorsque la fonction, éventuellement non linéaire, à construire relie des variables d'entrées, ou descripteurs, à une grandeur de sortie continue, on parle d'un problème de régression; lorsque la grandeur de sortie est discrète, c'est un problème de classification.

2.3 Paradigme d'apprentissage supervisé

Pour formaliser le problème d'apprentissage supervisé, nous nous donnons :

- Un générateur aléatoire de vecteurs $x \in \mathbb{X}$, échantillonnés (généralement indépendamment) selon une distribution $\mathbb{P}(x)$, fixe mais inconnues. Typiquement, \mathbb{X} est ici un sousespace de \mathbb{R}^d :
- Un superviseur (ou oracle) qui pour chaque entrée x associe une sortie $y \in \mathcal{Y}$, échantillonnée selon la distribution conditionnelle $\mathbb{P}(y/x)$, également fixe mais inconnue. Typiquement, \mathcal{Y} sera identifiable à un sous-ensemble fini de \mathbb{N} pour la classification ou à un sous-ensemble compact de \mathbb{R} pour la régression. Nous ne considérons pas de sorties vectorielles, mais les résultats présentés s'y adaptent;
- Une machine capable d'implémenter un ensemble de fonctions, c'est-à-dire que nous définissons un espace d'hypothèses $\mathcal{H} \subset \mathcal{Y}^{\mathbb{X}}$.

L'objectif est donc de trouver la fonction $\varphi \in \mathcal{H}$ qui prédise "au mieux" les réponses de l'oracle.

Pour quantifier cela, nous introduisons une fonction de perte $L \in \mathbb{R}^{\mathcal{Y} \times \mathcal{Y}}$ qui mesure l'erreur $L(y, \varphi(x))$ entre la réponse y de l'oracle pour une entrée x et la prédiction $\varphi(x)$ pour cette même entrée. Le risque réel est alors l'intégration de la fonction de perte sur l'espace des possibles :

$$R(\varphi) = \int \int_{\mathbb{X} \times \mathcal{Y}} L(y, \varphi(x)) d\mathbb{P}(x, y) = \mathbb{E}[L(y, \varphi(x))]$$
 (2.1)

L'objectif de l'apprentissage supervisé est alors de déterminer la fonction φ de $\mathcal H$ qui minimise le risque réel associé, c'est-à-dire résoudre

$$\varphi_0 \in arg\min_{\varphi \in \mathcal{H}} [\mathbb{E}[L(y, \varphi(x))]$$
 (2.2)

Toutefois, même en faisant l'abstraction de la difficulté potentielle du problème d'optimisation sous-jacent, ce problème nécessite de connaître la distribution jointe $\mathbb{P}(x,y) = \mathbb{P}(x)\mathbb{P}(y/x)$, qui est inconnue. La seule information disponible se trouve dans les données. On suppose disposer d'une base d'entraînement composée de couples entrée-sortie, $\{(x_i,y_i)_{1\leqslant i\leqslant N}\}$. Nous noterons parfois z_i le couple (x_i,y_i) . On peut alors estimer φ_0 par φ_N , minimiseur du risque empirique associé à la base d'entraînement :

$$\varphi_0 \in \underset{\varphi \in \mathcal{H}}{arg \min} R_N(\varphi) = \frac{1}{N} \sum_{i=1}^N L(y_i, \varphi(x_i))$$
 (2.3)

On peut remarquer que pour une fonction φ donnée, le risque empirique est un estimateur non biaisé du risque réel, c'est-à-dire que $\mathbb{E}[L(y,\varphi(x))]=R(\varphi)$. Toutefois, dans le cas général on ne peut pas écrire que le risque empirique du minimiseur φ_N est un estimateur sans biais du risque réel du minimiseur φ_0 , cela en raison de la dépendance implicite de φ_N en les données (qui sont les quantités aléatoires). Dès lors, on peut se demander si l'estimateur φ_N converge vers la solution φ_0 , à quel point $R_N(\varphi_N)$ est une bonne approximation de $R(\varphi_N)$, etc. [37] répond à ses questions, parmi d'autres. Sans trop entrer dans les détails, il montre par exemple qu'avec probabilité d'au moins $1-\delta$ on a

$$\forall \varphi \in \mathcal{H}, R(\varphi) \leqslant R_N(\varphi) + O\left(\sqrt{\frac{d_{vc}}{N}} \ln \frac{1}{N}\right)$$
 (2.4)

où d_{vc} est la dimension de Vapink-Chervonenkis de l'espace d'hypothèse \mathcal{H} (grossièrement, le nombre de degrés de liberté, soit le nombre de paramètres libres pour une architecture linéaire). On peut donc calculer une borne supérieure du risque réel, qui montre par ailleurs que le principe de minimisation du risque empirique est pertinent et donne même sa vitesse de convergence. Cela pourrait être utilisé pour évaluer un estimateur φ , et même sélectionner le modèle (entendre l'espace d'hypothèse, voir le principe de minimisation structurelle du risque). Notons toutefois que nous avons ici accès à une majoration du risque, pas au risque lui-même. La borne est vraie pour toute distribution $\mathbb{P}(x,y)$. C'est sa grande force, de par le caractère général qu'elle apporte, mais c'est aussi sa faiblesse, dans la mesure où la borne peut souvent être large. Pour cette raison notamment, nous ne développons pas plus avant cette approche, le lecteur intéressé pour se référer à [16] pour une introduction au sujet.

2.4 Performance de généralisation d'un modèle

Les performances de généralisation d'un modèle mesurent les capacités du modèle à traiter de nouveaux exemples, objectif premier de l'apprentissage.

Afin d'estimer ces capacités, la base de données, amputée de la base de test, est initialement scindée en deux sous-ensembles : *l'ensemble d'apprentissage* utilisé pour ajuster les paramètres du modèle et *l'ensemble de validation* sur lequel les performances de généralisation du modèle sont estimès par l'erreur de généralisation calculée sur la base de validation.

L'obtention de bonnes performances en apprentissage (risque empirique minimum en apprentissage) ne garantit cependant pas de bonnes performances de généralisation. Cette limitation s'appelle le sur-ajustement.

2.5 Sur-ajustement et compromis Biais/Variance

Nous allons illustrer le problème du sur-ajustement dans le cadre d'un problème de régression.

Considérons un processus ψ engendrant un ensemble d'observations $Y=\{y_i=\psi(x_i)+\nu,x_i\in X,i=1,...,n\}$ avec ν un bruit additif de moyenne nulle et de variance σ^2 et $X=\{x_i,i=1...,n\}$. Soit $\varphi(x_i,X_j)$ le modèle approchant ψ ayant appris sur le sous-ensemble X_j de X.

L'erreur de prédiction, au point x_0 , peut être mesuré par :

$$\begin{split} \mathbb{P}_{\varphi}^{2}(x_{0},X) &= \mathbb{E}[Y - \varphi(x_{0},X)] \\ &= var[Y - \varphi(x_{0},X)] + [\mathbb{E}[Y - \varphi(x_{0},X)]]^{2} \\ &= var[Y - \psi(x_{0}) + \psi(x_{0}) - \varphi(x_{0},X)] + [\mathbb{E}[Y - \psi(x_{0}) + \psi(x_{0})\varphi(x_{0},X)]]^{2} \\ &= var[\nu + \psi(x_{0}) - \varphi(x_{0},X)] + [\mathbb{E}(\nu) + \mathbb{E}[\psi(x_{0}) - \varphi(x_{0},X)]]^{2} \\ &= \sigma^{2} + var[\varphi(x_{0},X)] + [\mathbb{E}[\psi(x_{0}) - \varphi(x_{0},X)]]^{2} \end{split}$$

Cette relation donne l'erreur de prédiction en un point. L'erreur de prédiction ou erreur de généralisation est donc l'espérance mathématique de la quantité précédente :

$$\begin{split} \mathbb{P}_{\varphi}^{2}(X) &= \mathbb{E}[\mathbb{P}_{\varphi}^{2}(x, X)] \\ &= \int \mathbb{P}_{\varphi}^{2}(x, X) p_{x} d_{x} \\ &= \sigma^{2} + \mathbb{E}[var[\varphi(x, X)]] + \mathbb{E}[\mathbb{E}[\psi(x) - \varphi(x, X)]]^{2} \end{split}$$

Soit:

 σ^2 : le bruit,

 $\mathbb{E}[var[\varphi(x, X)]]$: la variance, $\mathbb{E}[\mathbb{E}[\psi(x) - \varphi(x, X)]]^2$: le biais.

Le bruit de mesure, qui est irréductible, correspond à l'erreur commise lors de l'enregistrement des observations; la variance représente la sensibilité du modèle aux particularités de l'ensemble d'apprentissage et le biais mesure la distance entre les prédictions et la fonction de régression.

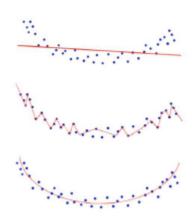


Figure 2.1 – Modèles ayant : un grand biais et une faible variance (en haut), un faible biais et une forte variance (sur-ajustement) (au milieu), un bon compromis biais/variance (en bas).

Le biais et la variance varient en sens inverse, en fonction de la complexité du modèle : quand la complexité du modèle (le nombre de paramètres du modèle) augmente, le biais diminue car le modèle s'ajuste aux exemples, mais la variance augmente car le modèle peut grandement varier en fonction des exemples utilisés pour l'apprentissage (Figure 2.1); lorsque le modèle est

trop simple, il est moins sensible aux particularités des exemples d'apprentissage impliquant une variance faible, mais l'apprentissage risque alors d'être de mauvaise qualité, avec un fort biais (Figure 2.1).

Lorsque l'erreur de généralisation estimée à l'aide des exemples de l'ensemble de validation est grande devant l'erreur commise par le modèle sur les exemples d'apprentissage, le modèle est dit *sur-ajusté*.

Il est donc nécessaire de déterminer la complexité optimale du modèle à apprendre, autrement dit, le nombre optimal des paramètres du modèle, afin d'être en mesure d'apprendre le phénomène sous-jacent aux exemples d'apprentissage sans s'ajuster trop finement aux particularités de ceux-ci, notamment au bruit de mesure de ces exemples.

2.6 Réduction des risques et choix du modèle

2.6.1 Sélection du modèle

Le choix de l'architecture d'un réseau détermine la classe des fonctions calculables par celui-ci, ou encore sa complexité potentielle. C'est évidemment le premier paramètre sur lequel les utilisateurs de réseaux ont joué pour contrôler les performances d'un système. La démarche la plus évidente pour choisir la meilleure architecture est bien entendu de tester plusieurs modèles différents, changeants les types de neurones, le nombre de couches, le nombre de neurones cachés. Cependant, l'évaluation comparative des réseaux ainsi créé pose problème, de nombreuses méthodes existant mais étant beaucoup trop lourdes en calculs. Pour cette raison, la communauté de réseaux de neurones a adopté des procédures sous-optimales. La plus courante consiste en l'utilisation d'un ensemble de validation, le réseau offrant les erreurs les moindres sur cet ensemble étant considéré le meilleur. Cette méthode est également coûteuse en temps de calculs et soumise à de nombreux aléas.

On présente ici quelques méthodes statistiques :

2.6.1.1 sélection des variables

Lorsqu'on modélise un processus physique ou chimique bien connu, on détermine généralement, par une analyse préalable du problème, les variables qui ont une influence sur le phénomène étudié; dans ce cas, une étape de sélection des variables n'est pas toujours nécessaire. En revanche, ce n'est pas le cas lorsqu'on cherche à modéliser un processus économique, social ou financier, ou encore un processus physico-chimique complexe ou mal connu : les experts du domaine peuvent donner des indications sur les facteurs qu'ils estiment pertinents, mais il s'agit souvent de jugements subjectifs qu'il faut mettre à l'épreuve des faits. On est alors conduit à retenir un grand nombre de variables candidates (appelées également facteurs ou descripteurs), potentiellement pertinentes. Néanmoins, la complexité du modèle croît avec le nombre de variables et par suite au nombre de paramètres à estimer qu'il comporte. Par ailleurs, le risque de surajustement d'un modèle est d'autant plus faible que le nombre de paramètres à estimer qu'il comporte est petit devant le nombre d'exemples que contient l'ensemble d'apprentissage [28] [9].

Le nombre de paramètres étant lié au nombre de variables du modèle, la réduction du nombre de variables doit être réalisée.

Une fois les entrées du modèle sont sélectionnées, la complexité optimale du modèle peut être déterminée par *validation croisée* [24].

2.6.1.2 Validation croisée

Cette méthode consiste en l'estimation des performances de généralisation de modèles de différentes complexités et en la sélection du modèle dont les performances de généralisation ainsi estimées sont les meilleures. Pour ce faire, l'ensemble d'apprentissage/validation est partitionné en N_{vc} sous-ensembles disjoints, appelés plis; pour les problèmes de classification, il faut s'assurer que les classes sont représentées de manière homogène dans chaque sous-ensemble. L'apprentissage d'un modèle de complexité donnée est effectué avec les exemples de $N_{vc}-1$ plis, et l'erreur quadratique moyenne commise par ce modèle sur les exemples du pli restant (pli de validation) est calculée. Cette opération est répétée N_{vc} fois, afin que chaque exemple se trouve une fois et une seule dans le pli de validation (Figure 2.2). Le score de validation croisée, calculé à l'issue des apprentissages, est la moyenne des erreurs quadratiques commises sur les plis de validation; il constitue une estimation de la capacité de généralisation des modèles de la complexité considérée. Après avoir ainsi évalué les capacités de généralisation des modèles de différentes complexités, on considère que le modèle optimal est celui pour lequel le plus petit score de validation croisée a été obtenu.

En utilisant la fonction de coût des moindres carrés, on procède généralement comme suit :

- pour chaque partie laissée de côté, on calcule l'erreur quadratique moyenne de validation (EQMV),
- à la fin, la performance de généralisation du modèle (ou encore le score de validation croisée) est estimée en réalisant la moyenne quadratique des N_{vc} erreurs (EQMV) précédentes.

Dans le contexte de réseaux de neurones, la recherche de l'architecture optimale s'effectue souvent en partant d'un modèle linéaire et en augmentant progressivement le nombre de neurones cachés. Le modèle optimal est alors défini comme étant celui qui présente le meilleur score de validation croisée.

La limite naturelle de la validation croisée correspond au cas où N_{vc} est égal au nombre d'exemples dans la base d'apprentissage. Cette méthode est connue sous le nom de "leave-one- out" (voir [30]) car chaque apprentissage n'est validé que sur un seul exemple. Les difficultés de cette méthode sont de deux ordres :

- le temps de calcul nécessaire, qui-pour une même base d'apprentissage est d'autant plus grand que N_{VC} est élevé (il est donc maximum dans le cas du leave-one-out),
- des performances contrastées en termes de taille de l'architecture sélectionnée et d'estimation des performances. À ce niveau, deux cas sont à distinguer :
 - le nombre d'exemples est grand au regard de la complexité de la fonction à approcher (nombre d'entrées, non-linéarité): dans ce cas, le phénomène de sur-ajustement est difficile à mettre en évidence. La méthode donne certes de bons résultats - même avec un petit nombre de partitions - mais sans grand mérite car il y a peu de risque de sur-ajustement.
 - le nombre d'exemples est petit au regard de la complexité de la fonction à approcher :
 on est obligé d'augmenter le nombre de partitions de façon à garder un nombre suffi-

sant d'exemples pour réaliser l'apprentissage des N_{vc} modèles. Les résultats montrent alors une tendance à la surestimation de la taille des modèles nécessaires et à la sous-estimation des scores de validation croisée. Ceci traduit un phénomène mis en évidence par [5] : une petite modification des données d'apprentissage peut entraîner de grandes différences dans les modèles sélectionnés. Autrement dit, si l'on raisonne en termes de fonction de coût, les exemples dont on se sert pour estimer les paramètres d'un modèle peuvent grandement influencer les minima vers lesquels convergent les différents apprentissages. On parle alors d'instabilité vis-à-vis des données d'apprentissage : les EQMVs calculées à partir des différentes partitions ne peuvent donc pas raisonnablement êtres moyennées pour estimer la performance de généralisation du modèle.

La littérature conseille généralement d'utiliser $N_{vc}=10$. Cependant, ne sachant pas à priori s'il dispose de "peu" ou de "beaucoup" d'exemples (au sens défini ci-dessus), le concepteur essaiera souvent différentes valeurs de N_{vc} . Si l'on se rappelle qu'à partir d'une base d'apprentissage, il est recommandé de procéder à plusieurs initialisations des poids de façon à diminuer le risque de minima locaux, on arrive très vite à un nombre d'apprentissages très élevé. En soi, ceci n'est pas grave si les résultats de ces différents essais sont cohérents. Dans le cas contraire, le découragement peut rapidement intervenir.

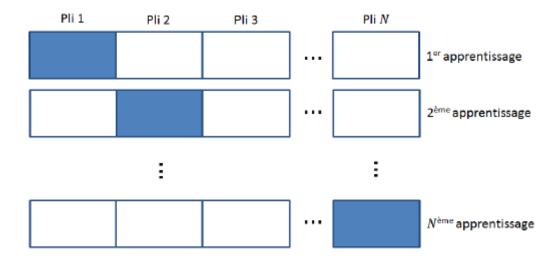


Figure 2.2 — Principe de la validation croisée : les plis bleus sont utilisés pour la validation, les autres pour l'apprentissage

2.6.1.3 Régularisation

Différentes méthodes de régularisation existent pour réduire les risques de sur-ajustement, comme, par exemple : l'arrêt prématuré et la modération des poids.

2.6.1.3.1 arrêt prématuré (ou early-stopping) Cette méthode consiste en l'arrêt de l'apprentissage avant que le minimum de la fonction de coût ne soit atteint, empêchant ainsi le modèle de s'ajuster trop finement aux exemples [38]. Dans cette méthode, l'EQM sur les exemples

d'un ensemble de données disjoint de l'ensemble d'apprentissage/validation est calculée et l'apprentissage du modéle est arrêté dès que le minimum de cette EQM est atteint, comme illustré dans la Figure 2.3. Ceci a pour conséquence de réduire la dimension de l'ensemble d'apprentissage/validation.

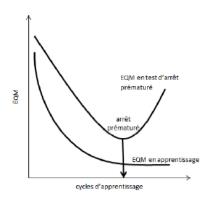


Figure 2.3 – Principe de la méthode d'arrêt prématuré.

2.6.1.3.2 modération des poids (ou weight decay) La régularisation par modération des poids consiste en l'ajout d'un terme dans la fonction de coût à minimiser [20]. Ce terme additionnel est proportionnel à la norme du vecteur des paramètres à ajuster ; l'algorithme d'apprentissage minimise une combinaison linèaire de l'erreur quadratique et de la norme des paramètres, de la forme :

$$J = (1 - \lambda)J + \frac{\lambda}{N_w} \sum_i w_i^2$$

où J est la fonction de coût initiale définie précédemment, N_w le nombre de paramètres w_i du réseau de neurones et λ est un hyper-paramètre qui détermine les influences respectives de l'erreur de modélisation et du terme de régularisation. Plus λ est grand, plus l'apprentissage minimise la variance, ce qui augmente le biais; plus λ est petit, plus l'apprentissage minimise le biais, ce qui augmente la variance. Le choix de λ détermine donc le compromis biais-variance défini auparavant.

2.6.2 Procédure standarde de conception d'un modèle

Pour chaque valeur des hyper-paramètres à comparer

Obtenir un modèle entrainé à partir de D_{train}

Evaluer la performance du modèle sur D_{valid} ,

Retourner l'hyper-paramètre où les hyper-paramètres ayant donné le modèle avec la meilleure performance sur D_{valid} ,

Fin

Conclusion

Dans ce chapitre nous avons traité le problème d'apprentissage statistique. Dans un premier lieu nous avons parlé de problème de généralisation et celui de sur-ajustement, après nous avons défini quelques méthodes statistique pour l'éviter.

CHAPITRE 3

SÉLECTION ET ESTIMATION DES PERFORMANCES DU MODÈLE NEURONALE

Introduction

Le nombre de neurones cachés présents dans un réseau des neurones détermine la complexité du modèle, donc sa capacitée à généraliser.

Dans ce chapitre on va s'intéresser au problème de déterminer ce nombre, ceci par application des méthodes statistiques celles de la validation croisée et du test croisé.

3.1 Détermination du nombre de neurones cachés d'un réseau de neurones par validation croisée

Comme nous l'avons présenté dans le paragraphe 2.6 du Chapitre 2, la complexité optimale du modèle est déterminée par validation croisée. Dans notre étude, la sélection du nombre de neurones cachés a été réalisée à l'aide de la méthode de validation croisée à 10 plis.

Le nombre optimal de neurones cachés déterminé, l'apprentissage du modèle final est effectué à l'aide de tous les exemples de l'ensemble d'apprentissage/validation. Comme nous l'avons vu précédemment, l'apprentissage dépendant de l'initialisation des paramètres du réseau de neurones,on effectue l'apprentissage de plusieurs réseaux possédant le nombre optimal de neurones cachés et l'on calcule l'erreur quadratique moyenne réalisée sur la base d'apprentissage/validation. Le modèle obtenant les meilleures performances d'apprentissage est alors sélectionné. Ensuite, les performances de ce modèle doivent être estimées. Ces différentes étapes sont schématisées sur la Figure 3.1.

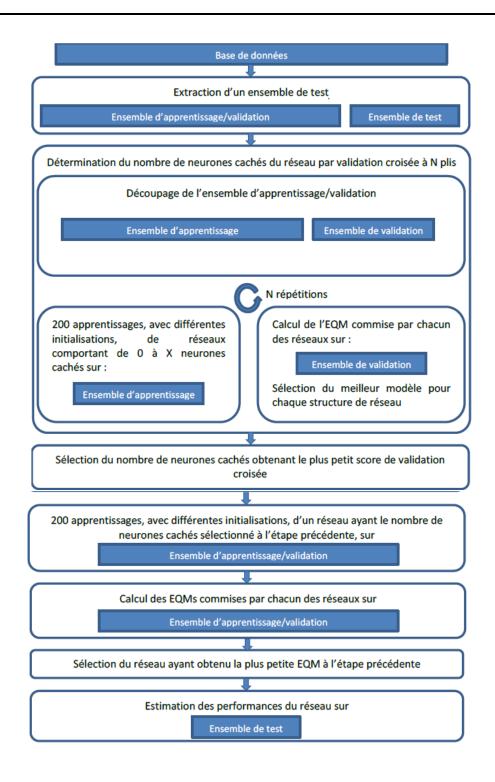


Figure 3.1 – Schéma résumant les différentes étapes de la construction d'un modèle.

3.2 Estimation des performances d'un modèle – Méthode du test croisé

Comme nous l'avons vu, la base des données est généralement scindée en deux : l'ensemble d'apprentissage/validation (qui, dans le cadre de la validation croisée, est lui-même partitionné

en N_{vc} plis) et l'ensemble de test, qui est utilisé pour estimer les performances de généralisation du modèle.

Compte tenu du petit nombre de données disponibles pour nos études; de classification d'IRIS et celle de reconnaissance du locuteur et malgré l'utilisation de la validation croisée, les risques d'ajustement du modèle aux spécificités des exemples d'apprentissage/validation, entrainant une mauvaise généralisation du modèle construit, sont grands. Afin de mesurer la variabilité des performances de nos classifieurs, leur estimation a été effectuée à l'aide d'une méthode originale, dite de « test croisé ».

Selon cette méthode, qui combine conception et test de modèles, l'ensemble de tous les exemples disponibles est partitionné en N_t plis disjoints; dans le même esprit que la validation croisée, mais avec un objectif différent, un modèle est conçu par validation croisée en utilisant N_t-1 plis comme ensemble d'apprentissage/validation (lui-même partitionné en N_{vc} plis de validation croisée), et il est testé sur le pli restant. La procédure est itérée N_t fois de manière à ce que tous les exemples disponibles soient utilisés en test une fois et une seule; cette méthode consiste donc en itérations de la procédure présentée Figure 3.1 et peut être résumée par le pseudo-code suivant :

```
Partitionner l'ensemble E de tous les exemples en sous-ensembles disjoints Tant que i \leq N_t Retirer les exemples du sous-ensemble i de l'ensemble des exemples Partitionner l'ensemble des autres exemples en N_{cv} ensembles disjoints Concevoir un modèle par validation croisée Estimer ses performances de test sur les exemples du sous-ensemble i Réintégrer les exemples du sous-ensemble i dans l'ensemble \mathbb E Fin
```

On dispose ainsi de modèles différents, qui peuvent avoir des nombres de neurones cachés différents puisqu'ils ont été conçus à l'aide d'ensembles d'apprentissage/validation puisqu'ils ont été conçus à l'aide d'ensembles d'apprentissage/validation et l'on dispose donc des estimations du nombre optimal de neurones cachés et de l'ensemble d'indicateurs de performances. On peut estimer la moyenne et l'écart-type de chacune de ces grandeurs, donc connaître leur ordre de grandeur et leur variabilité.

3.3 Implémentation de la méthode de test croisé pour le problème de classification lris Fisher

Notre but dans cette section est d'appliquer la procédure de test croisé dans l'apprentissage du PMC permettant de classifier les Iris de Fisher et de donner les résultats numériques trouvés.

3.3.1 Description de la base des données

Les iris de Fisher sont des données proposées en 1933 par le statisticien Ronald Aylmer Fisher comme données de référence pour l'analyse discriminante et la classification. Il est parfois

aussi appelé Iris d'Anderson du nom d'Edgar Anderson qui a collecté ces données afin de quantifier les variations de morphologie des fleurs d'iris de trois espèces. Le jeu de données comprend 50 échantillons de chacune des trois espèces d'iris (Iris setosa, Iris virginica et Iris versicolor). Quatre caractéristiques ont été mesurées à partir de chaque échantillon : la longueur et la largeur des sépales et des pétales, en centimètres.



Figure 3.2 – Iris de Ficher.

Généralement, dans un problème de classification à k classes, la couche de sortie a souvent contenu le nombre de classes, chaque neurone représente une classe. Dans ce cas de classification IRIS, la couche de sortie contient trois neurones. Le codage des classes utilisé est :

La sortie désirée correspondante à la classa a) (Setosa) est : $(1;0;0)^t$ La sortie désirée correspondante à la classa b) (Virginica) est : $(0;1;0)^t$ La sortie désirée correspondante à la classa c) (Versicolor) est : $(1;0;1)^t$

3.3.2 Conception du modèle

Comme on a s'est intéressé par quatre caractéristiques, alors chaque exemple est un vecteur de quatre composantes, et par la suite la couche d'entrée doit contenir quatre neurones. Concernant la couche cachée plusieurs travaux ont été effectués dans le cadre d'optimisation d'architecture, ils ont trouvé que le nombre optimal de neurones dans la couche cachée pour ce problème est 10 neurones dont on va vérifier par le test croisé.

Le modèle utilisé avec une seule couche cachée dont on va tester avec un nombre de neurones entre zéro et dix-huit, 4 neurones dans la couche d'entrée et 3 neurones en sortie. Pour tout le réseau la fonction d'activation utilisée est la fonction sigmoïde.

3.3.3 Normalisation et pré-traitement des données

Les données collectées sont mesurées en centimètres, alors ne sont pas forcément comprises entre 0 et 1, donc pour les entrer au réseau il faut tout d'abord les normaliser. La normalisation des données se fait suivant la formule :

$$x_{i \text{ new}}^{k} = \frac{|x_{i}^{k} - \min(x_{i})|}{\max(x_{i}) - \min(x_{i})}$$
(3.1)

3.3.4 Implémentations et résultats numériques

Les résultats que nous présentons ici ont été obtenus à l'aide de la base des données étiquetées, présentées dans le paragraphe précédant. Pour rappel, cette base est composée de 150 mesures de la longueur et la largeur des sépales et des pétales, en centimètres.

L'apprentissage est effectué par l'algorithme de rétro-propagation avec un pas d'apprentissage $\epsilon=0.25$.

3.3.4.1 Apprentissage sans régularisation

Les réseaux de neurones testés comportent de 0 à 18 neurones cachés. Nous n'avons pas mis en oeuvre de réseaux comportant plus de 18 neurones cachés en raison du nombre de paramètres à ajuster que cela implique. En effet, notre base de données totale ne comporte que cent-cinquante exemples, alors qu'un réseau de neurones à quatre entrées, dix-huit neurones cachés et trois neurones de sortie, comporte $(4+1)\times 18+(18+1)\times 3=147$ paramètres à ajuster. Ce nombre est du même ordre de grandeur que le nombre d'exemples, ce qui crée un risque de sur-ajustement très important pour des modèles dont l'apprentissage est effectué sans régularisation.

Les réseaux de neurones avec une couche cachée obtenant le meilleur score de validation croisée, pour chaque ensemble de test croisé, sont ceux comportant dix neurones cachés, comme illustré Figure 3.3 pour trois tests pris au hasard.

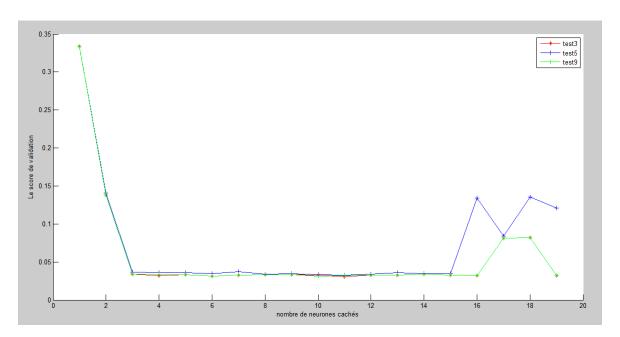


Figure 3.3 — Scores de validation croisée des réseaux, présentés pour trois ensembles de test, en fonction du nombre de neurones cachés qu'ils comportent.

Afin de pouvoir ultérieurement évaluer l'impact de la méthode de régularisation sur les performances des classifieurs, la moyenne des performances obtenues en test par les dix classifieurs sans régularisation est égale à 1.02.

3.3.4.2 Apprentissage avec régularisation

La construction des classifieurs conventionnels, avec régularisation par modération des poids, nécessite non seulement la détermination du nombre de neurones cachés (N_{NC}) du réseau de neurones mais également celle du paramètre de régularisation λ . Ces deux paramètres ont été conjointement déterminés par validation croisée. Nous avons testé les valeurs suivantes pour le paramètre de régularisation : $\lambda = 0$, 05; 0, 10 et 0, 15 avec un N_{NC} variant de 0 à 18.

À l'issue des procédures de validation croisée, le paramètre de régularisation, sélectionné pour chacun des dix modèles construits, était : 0 c'est-à-dire sans régularisation. Le nombre de neurones cachés sélectionné est égale à 10 (Figure 3.4). À titre informatif, nous avons aussi représenté le score de validation croisée de réseaux ayant appris sans régularisation.

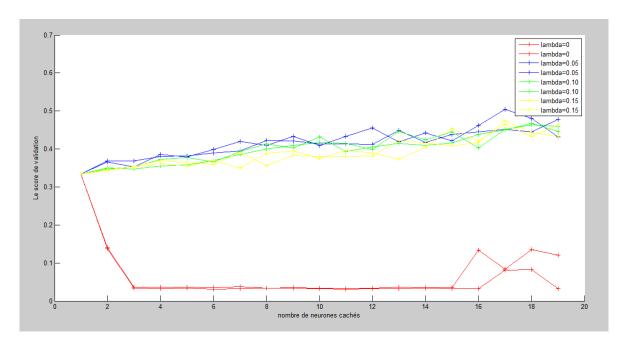


Figure 3.4 — Scores de validation croisée des réseaux, présentés pour trois ensembles de test, en fonction du nombre de neurones cachés qu'ils comportent.

La moyenne des performances obtenues par les dix classifieurs conventionnels, construits selon la méthode de test croisé avec régularisation, sur les plis de test correspondants est égale à 1.89.

3.3.4.3 Résultats obtenus en testant le modèle sélectionné

Le tableau suivant montre la base utilisée pour l'apprentissage et pour le test,

Les résultats illustrés dans la figure 3.5 sont obtenus en effectuant l'apprentissage sur 75 exemples avec $\lambda=0$

	Base d'apprentissage	Base du test
Setosa	25	25
Virginica	25	25
Versicolor	25	25
Total	75	75

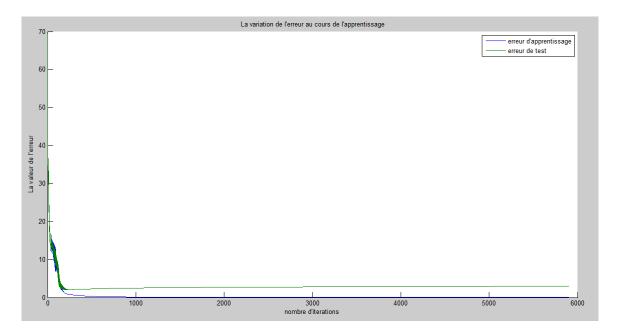


Figure 3.5 — Erreur d'apprentissage et celle de test effectué sur le modèle choisi en fonction des itérations.

Conclusion

Dans ce chapitre nous avons traité la méthode du test croisé pour déterminer le nombre du neurones cachés, nous avons la testé par le problème de la classification d'IRIS, ainsi nous avons affiché les résultats obtenus.

CHAPITRE 4

SYSTÈME DE RECONNAISSANCE AUTOMATIQUE DU LOCUTEUR THÉORIE ET APPLICATION

Introduction

La reconnaissance automatique du locuteur (RAL) est un terme génétique regroupant les problèmes relatifs à l'identification ou à la vérification du locuteur sur la base de l'information contenue dans le signal acoustique : il s'agit de reconnaître une personne à partir de sa voix. Un système de reconnaissance de locuteur procède en trois ètapes : l'analyse acoustique du signal parole, la modélisation du locuteur et une dernière étape de décision.

Le signal sonore, tel qu'il est utilisé par les systèmes automatiques, est aussi porteur d'une information relative au matériel qui compose la chaîne d'enregistrement et de transmission ou à l'environnement du locuteur. Ces informations sont généralement perçues comme nuisibles, car elles dégradent fortement les performances dans les différentes tâches des systèmes automatiques.

Dans ce chapitre, nous allons introduire le principe de la reconnaissance automatique du locuteur et nous présentons les différentes étapes du système, après nous allons présenter le protocole expérimental de notre approche par les réseaux de neurones et les résultats obtenus.

4.1 La reconnaissance du locuteur théorie et principes

4.1.1 Introduction à la reconnaissance automatique du locuteur

Comme on peut le constater sur la figure 4.1, la reconnaissance automatique du locuteur s'inscrit dans le domaine plus général du traitement de la parole. Elle exploite la variabilité interlocuteurs et s'intéresse aux informations extra-linguistiques du signal vocal.

Les variations individuelles entre locuteurs ont deux origines essentielles. D'abord, les caractéristiques morphologiques de l'appareil de phonation sont différentes pour chaque locuteur, indépendamment de la phrase prononcée. Ensuite, une même phrase n'est pas prononcée de la

même façon par deux locuteurs; on observe des différences dans les débits d'élocution, dans l'étendue des variations du pitch ou encore des différences liées à leur milieu socioculturel. Cette variabilité est l'essence même de la reconnaissance automatique du locuteur.

La reconnaissance automatique du locuteur est probablement la méthode la plus ergonomique pour résoudre résoudre les problèmes d'accès notamment dans le cas des transactions téléphoniques. Cependant, la voix ne peut être considérée comme une caractéristique biométrique d'une personne compte tenu de la variabilité intra-locuteur. Ainsi, on préfère la qualifier comme une signature vocale plutôt qu'une empreinte vocale.

Les applications potentielles des systèmes de reconnaissance sont nombreuses, incluant notamment la sécurisation accrue des cartes d'accès (par exemple, cartes de crédit et cartes téléphoniques), le contrôle d'accès à des bases de données et bâtiments protégés, commerce électronique, services d'information et de réservation, etc.

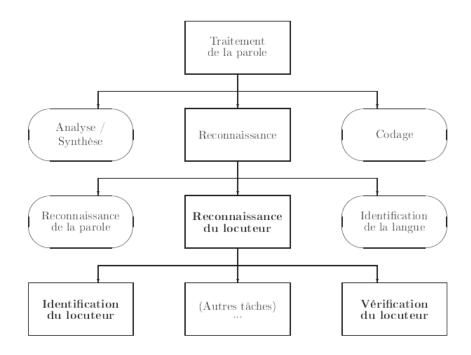


Figure 4.1 – Traitement de la parole.

4.1.1.1 Identification et vérification du locuteur

On distingue deux tâches principales en reconnaissance du locuteur : identification et vérification. l'identification du locuteur consiste à reconnaître une personne parmi un ensemble de locuteurs en comparant son expression vocale à des références connues. Deux modes d'identification sont possibles : identification en ensemble fermé pour lequel le locuteur est identifié parmi un nombre connu de locuteurs ou bien identification en ensemble ouvert pour lequel le locuteur à identifier n'appartient pas forcément à cet ensemble.

La vérification (ou l'authentification) du locuteur consiste, après que le locuteur a décliné son identité, à vérifier l'adéquation de son message vocal avec la référence acoustique du locuteur qu'il prétend être. C'est une décision en tout ou rien.

4.1.1.2 Mode dépendant et indépendant du texte

On distingue également la reconnaissance du locuteur indépendante du contenu de la phrase prononcée (mode indépendant au texte) et la reconnaissance du locuteur qui prononce un mot ou une phrase clef(mode dépendant du texte). Les niveaux de dépendance au texte sont classés suivant les applications :

- Systèmes à texte libre (ou free-text) : le locuteur est libre de prononcer ce qu'il veut.
 Dans ce mode, les phrases d'apprentissage et de test sont différentes.
- Systèmes à texte suggéré (ou text-prompted : un texte, différent à chaque session et pour chaque personne, est imposé au locuteur et déterminé par la machine. Les phases d'apprentissage et de test peuvent être différentes.
- Systèmes dépendants du vocabulaire (ou vocabulary-dependent): le locuteur prononce une séquence de mots issus d'un vocabulaire limité. Dans ce mode, l'apprentissage et le test sont réalisés sur des textes constitués à partir du même vocabulaire.
- Systèmes personnalisés dépendants du texte (ou user-specific text dependent : chaque locuteur a son propre mot de passe. Dans ce mode, l'apprentissage et le test sont réalisés sur le même texte.

D'évidence, la connaissance à priori du message vocal rend la tâche des systèmes de RAL plus facile et les performances sont meilleures. La reconnaissance en mode indépendant du texte nécessite plus de durée de parole que le mode dépendant du texte.

4.1.1.3 Sources d'erreurs

Le signal acoustique de la parole présente des caractéristiques qui rendent complexe son interprétation. L'information portée par ce signal peut être analysée de bien des façons et à plusieurs niveaux (acoustique, phonologique, morphologique, syntaxique. Plus particulièrement, on a vu que la variabilité inter-locuteurs est l'essence même de la reconnaissance.ll existe, cependant, plusieurs facteurs qui peuvent augmenter la variabilité inter-locuteur comme par exemple :

- L'état pathologique du locuteur (maladie, émotions,...).
- Vieillissement (la voix d'une personne change au fur et à mesure de son vieillissement).
- Facteurs socioculturels (le locuteur peut changer d'accent).
- Locuteurs non coopératifs (notamment dans des applications judiciaires).
- Conditions de prise de son, bruit ambiant, ...

4.1.1.4 Système de reconnaissance automatique du locuteur

La reconnaissance automatique du locuteur peut être interprétée comme une tâche particulière de reconnaissance de formes. Différents modules sont présents dans ce système (figure 4.2). Tout d'abord, le message vocal, capté par un microphone, est converti en signal numérique. Il est ensuite analysé dans un étage d'analyse acoustique. À l'issue de cette étape d'apprentissage, on crée un modèle du locuteur. A la reconnaissance, un modèle de classification va mesurer la similarité entre les paramètres acoustiques du signal prononcé et les modèles de locuteurs présents dans la base. En dernier lieu, un module de décision, basé sur une stratégie de décision donnée, fournit la réponse du système. On peut également introduire un module d'adaptation pour augmenter les performances du système de reconnaissance.

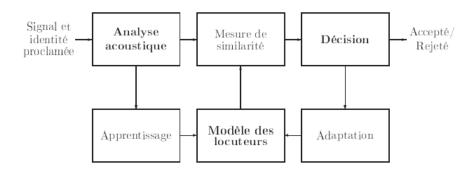


Figure 4.2 – Schéma typique d'un système de vérification du locuteur.

4.1.2 Analyse acoustique du signal de parole

La mise en œuvre d'une tâche de reconnaissance de locuteur (ou de parole) est loin d'être facile, et ce pour deux raisons majeurs. La première teint au fait que l'on ne maîtrise pas l'espace acoustique et en particulier la fonction de production d'un signal de parole correspondant à l'émission d'un symbole donné par un locuteur particulier. La seconde, qui n'est qu'un corollaire de la première, est que la concrétisation acoustique d'un symbole donné n'est pas unique.

4.1.2.1 Production et perception du signal vocal

La phonation est réalisée au moyen d'un appareil qui n'est pas spécifique à la parole. Les organes qu'elle met en jeu sont d'abord affectés aux fonctions vitales de respiration et de nutrition. Le processus de phonation comporte trois étapes essentielles [1] :

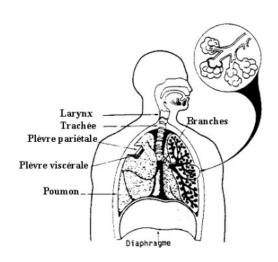


Figure 4.3 – L'appareil phonatoire.

- La génération d'une énergie ventilatoire qui va servir à mettre en mouvement oscillatoire les cordes vocales ou à les écarter afin de générer un bruit.
- La vibration des cordes vocales donnant naissance à tous les sons voisés, suit 80% du temps de phonation.

 La réalisation dune disposition articulatoire dans ce qu'il est commode de désigner sous le nom de cavités surpra-glottiques.

Le système vocal se compose d'une soufflerie (poumons et conduit trachéo-bronchique), du larynx et du conduit vocal, lui même formé par le pharynx et les cavités orales et nasales (figure 4.3).

Dans le cadre du traitement de la parole, une bonne connaissance des mécanismes de l'audition et des propriétés perceptuelles de l'oreille est aussi importante qu'une maîtrise des mécanismes de production. L'appareil auditif se divise en deux parties : le système auditif périphérique correspondant à ce que l'on nomme communément l'oreille (qui se décompose en oreille externe, moyenne et interne) et le système auditif central.

Cependant, tout ce qui peut être acoustique-ment mesuré ou observé par la phonétique articulatoire n'est pas nécessairement perçu. Les psychoacoustiques tentent de comprendre comment l'information auditive est traitée par le cerveau. En effet, au delà des caractéristiques mesurables (comme l'intensité et la fréquence), le son a deux qualités subjectives, la force et la hauteur, qui s'apprécient différemment. Les qualités subjectives relèvent des sensations éprouvées par un sujet qui écoute, et ne peuvent pas se mesurer sans lui. Ainsi, l'intensité perçue d'un son est égale à l'intensité physique (mesurée en décibels). Quant à sa hauteur, elle dépend de l'intensité avec laquelle ce son est transmis à l'auditeur.

Par ailleurs, la prosodie assure la fonction de segmentation syntaxique de l'énoncé [40] [?].

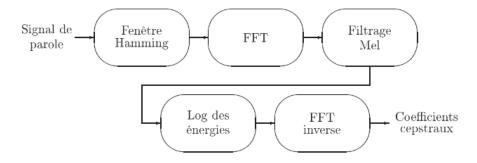


Figure 4.4 – Calcul des coefficients cepstraux avec une échelle Mel.

4.1.2.2 Coefficients cepstraux

L'analyse acoustique du signal de parole consiste à extraire l'information pertinente et à réduire au maximum la redondance. Généralement, on calcule un jeu de coefficients acoustiques à des intervalles de temps réguliers. Les techniques de paramétrisation acoustique sont nombreuses néanmoins on peut les regrouper en trois grandes familles :

- Analyse par bancs de filtres.
- Analyse par transformée de Fourier.
- Analyse par prédiction linéaire.

Les coefficients cepstraux issus d'une analyse par transformée de Fourier caractérisent bien la forme du spectre et permettent de séparer l'influence de la source de celle du conduit vocal. Ils peuvent aussi être calculés à partir d'une analyse de prédiction linéaire.

Le cepstre du signal de parole est défini comme la transformée de Fourier inverse du logarithme

de la densité spectrale de puissance. Pour ce signal, la source d'excitation glottique est convoluée avec la réponse non pulsionnelle du conduit vocal considéré comme un filtre linéaire :

$$s(t) = e(t) * h(t) \tag{4.1}$$

où s(t) est le signal de la parole, e(t) est la source d'excitation glottique et h(t) est la réponse impulsionnelle du conduit vocal.

L'application du logarithme du module de la transformée de Fourier à l'équation ?? donne :

$$log|S(f)| = log|E(f)| + log|H(f)|$$
(4.2)

Par une transformée de Fourier inverse, on obtient :

$$s'(cef) = e'(cef) + h'(cef)$$
(4.3)

La dimension du nouveau domaine est homogène à un temps et s'appelle la *quéfrence* (cef), le nouveau domaine s'appelle donc le domaine *quéfrentiel*. Un filtrage dans ce domaine s'appelle *lifrage*.

Les coefficients cepstraux les plus répandus sont les MFCC (Mel Frequency Cepstral Coefficients). Ils présentent l'avantage d'être faiblement corrélés entre eux, et qu'on peut donc approximer leur matrice de covariance par une matrice diagonale. L'utilisation ici d'un filtrage Mel est justifiée par le fait qu'il reproduit la sélectivité de l'oreille qui diminue avec l'accroissement de la fréquence.

4.1.3 Modélisation des locuteurs

Ce paragraphe parcourt brièvement les techniques les plus couramment utilisées en reconnaissance du locuteur. Comme dans le cas de la reconnaissance de la parole, le problème de reconnaissance du locuteur peut se formuler selon un problème de classification. Différentes approches ont été développées, néanmoins on peut les classer en quatre grandes familles :

- L'approche vectorielle : le locuteur est représenté par un ensemble de vecteurs de paramètres dans l'espace acoustique. Ses principales techniques sont la reconnaissance à la base de DTW et par quantification vectorielle.
- L'approche statistique : consiste à représenter chaque locuteur par une densité de probabilité dans l'espace des paramètres acoustiques. Elle couvre les techniques de modélisation par les modèles de Marcov cachés, par les mélanges de gaussiennes et par des mesures statistiques du second ordre.
- L'approche connexionniste : consiste principalement à modéliser les locuteurs par des réseaux de neurones.
- L'approche relative : il s'agit de modéliser un locuteur relativement par rapport à d'autres locuteurs de référence dont les modèles sont bien appris.

4.1.3.1 Approche vectorielle

Reconnaissance du locuteur à base DTW La reconnaissance par DTW (Dynamique Time Warping) repose sur le principe que chaque mot est représenté par une prononciation d'un même mot, l'algorithme met en correspondance des séquences de paramètres par distorsion temporelle (Time Warping). La programmation dynamique permet d'aligner temporellement une phrase de

test avec une phase d'apprentissage ce qui signifie que c'est une technique exclusivement utilisée en mode dépendant du texte.

Cette approche est facile à mettre en œuvre et donne des performances relativement bonnes [19][4].

Quantification vectorielle Il s'agit de représenter l'espace acoustique par un nombre fini de vecteurs acoustiques. Cela consiste à faire un partitionnement de cet espace en régions, qui seront représentées par leur vecteur centroïde. Pour déterminer la distance d'un vecteur acoustique à cet espace, on effectue une mesure minimale. Si le vecteur acoustique provient du même locuteur pour lequel on établit le dictionnaire de quantification, la distorsion sera en général moins grande que si ce vecteur provient d'un autre. Ainsi, on va représenter un locuteur par son dictionnaire de quantification. De nombreux articles proposent l'emploi de la quantification vectorielle en reconnaissance du locuteur.

4.1.3.2 Approche statistique

Modèles de Markov cachés Les modèles de Markov (ou HMM pour Hidden Marcov Models) ont été initialement introduits en reconnaissance de la parole. Puis leur utilisation s'est étendue peu à peu au domaine de la reconnaissance du locuteur. Dans cette approche, il ne s'agit plus d'une mesure de distance d'une forme acoustique à une référence, mais de la probabilité que la forme acoustique ait été engendrée par le modèle de référence du locuteur. Le modèle d'un locuteur est constitué de l'association d'une chaîne de Markov, une succession d'états avec des probabilités de transition d'un état à l'autre, et de lois de probabilités (probabilités d'observation d'un vecteur acoustique dans un état).

Les mélanges de gaussiennes La reconnaissance du locuteur par mélanges de gaussiennes (ou GMM pour Gaussian Mixture Models) consiste à modéliser un locuteur par une somme pondérée de composantes gaussiennes [31]. Ainsi une large gamme de distributions peut être parfaitement représentée. Chaque composante des gaussiennes est supposée modéliser un ensemble de classes acoustiques. L'utilisation de ce type de modèle semble être bien prometteuse. Il semble bien modéliser les caractéristiques spectrales des voix des locuteurs, et il est relativement simple à mettre en œuvre. Les mélanges de gaussiennes sont considérés comme un cas particulier des HMM et une extension de la quantification vectorielle.

Mesures statistiques du second ordre Cette partie présente une famille de mesures de similarité entre locuteurs. Ces mesures reposent sur les caractéristiques du second ordre d'une séquence. Plusieurs mesures de distance ont été utilisées, on peut citer : Le rapport de vraisemblance, la distance de *Kullbak-Leibler*, maximum de vraisemblance, test de sphéricité, dérivation absolue des valeurs propres. Ces mesures donnent des résultats très encourageants sur la parole propre, et, naturellement, voient leurs performances se dégrader sur la parole téléphonique. De part leur relative simplicité, ces mesures peuvent également servir de références pour évaluer la qualité d'une base de données [2] et [26].

4.1.3.3 Approche connexionniste

Les réseaux de neurones ont été assez largement utilisés en reconnaissance du locuteur. Ils offrent en effet une bonne alternative au problème de la discrimination entre les locuteurs. Ces outils de classification permettent de séparer des classes, dans un espace de représentation donné, de façon non linéaire. Pour une bonne description de cette technique, on peut lire [29]. L'inconvénient de cette technique en identification du locuteur est le coût important lié à l'ajout d'un nouveau locuteur dans la base de référence (ce n'est pas le cas en vérification du locuteur). On peut aussi utiliser les réseaux de neurones en les couplant à d'autres techniques, comme par exemple les modèles de Markov cachés. On parle alors de méthodes hybrides.

4.1.3.4 Approche relative

Cette nouvelle technique consiste à modéliser un locuteur non plus de façon absolue mais relativement à un ensemble de locuteurs bien appris. L'état de l'art de cette technique sera abordé avec plus de détails à [39]

4.1.4 Décision et mesure des performances

Comme on l'a déjà vu, on distingue deux tâches principales en reconnaissances du locuteur : la vérification du locuteur et l'identification du locuteur. Cependant, un système de RAL peut aussi servir à identifier les segments de chaque locuteur dans un document audio, à partir du locuteur ou à faire l'indexation des documents audio :

L'identification du locuteur consiste à reconnaître un locuteur en comparant son identité vocale à des références connues. Les performances du système d'identification sont données en termes de taux d'identification correcte I_c ou incorrecte I_i , soit :

$$I_c = \frac{\text{Nombre de tests correctement identifiés}}{\text{Nombre total de tentatives}}$$

et

$$I_i = \frac{\text{Nombre de tests mal identifiés}}{\text{Nombre total de tentatives}}$$

avec $I_c + I_i = 100\%$

La vérification du locuteur consiste, après que le locuteur a décliné son identité, à vérifier l'adéquation du message vocal avec la référence acoustique qu'il prétend être. C'est une précision en tout. Les performances de vérification de locuteur sont données en termes des faux rejets f_r et de fausses acceptations f_a :

$$f_r = \frac{\text{Nombre de tentatives d'abonnés rejetées}}{\text{Nombre total de tentatives d'abonnés}}$$

$$f_a = \frac{\text{Nombre de tentatives d'imposteurs acceptées}}{\text{Nombre total de tentatives d'imposteurs}}$$

La segmentation en locuteurs consiste à découper un flux sonore avec ou sans modèle généralement en deux phases : la détection de changements de locuteurs et le regroupement de locuteurs segments (appartenant au même locuteur). La première phase repose sur le principe

de calcul d'une distance entre deux portions de signal consécutives. Si elle excède un seuil, on décide qu'il y a eu changement de locuteur. La deuxième phase consiste à regrouper les segments de parole par locuteur selon un critère de distance. Une bonne segmentation fournit les changements de locuteurs corrects et des segments ne contenant qu'un seul locuteur. Nous distinguons deux types des erreurs pour la détection des changements de locuteurs :

- Une fausse alarme (FA) a lieu lorsqu'un changement de locuteur est détecté alors qu'il n'existe pas.
- Une détection manquée (DM) a lieu quand un changement de locuteur existant n'est pas détecté.

Une valeur élevée de FA est significative d'une sur-segmentation et une valeur élevée de DM indique une sous-segmentation.

La poursuite du locuteur (Speaker tracking se fait avec un modèle du locuteur, contrairement à la segmentation qui peut se faire sans modèle. Elle consiste à déterminer quand une personne parle dans une conversation.

Indexation des documents audio : Grâce au développement des techniques numériques, les besoins en outils d'indexation se font cruellement ressentir. Il devient donc indispensable d'indexer automatiquement les documents audio pour être exploitables.

La clé d'indexation qui nous intéresse ici est l'identité du locuteur : nous voudrions savoir qui parle et quand. En pratique, on dispose des documents audio représentés par leurs modèles respectifs. La phase de recherche du système d'indexation consiste, généralement à évaluer des mesures de similarité entre la requête et ces différents modèles. Par ailleurs le système d'indexation par locuteurs peut servir également comme étape préliminaire pour des tâches de transcription ou pour le suivi de locuteurs.

4.2 Reconnaissance du locuteur par les réseaux de neurones, protocole expérimental et résultats

4.2.1 LPCC ou Linear Prediction Cepstral Coefficients

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

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

4.2.2 Description de la base des données

Dans le but d'acquérir un ensemble adéquat de parole d'entrainement et de test pour la reconnaissance automatique du locuteur, une base de données a été demandée. Par conséquent, plusieurs bases de données peuvent aider les chercheurs pour comparer l'efficacité de différentes

approches sur des données communes et de sélectionner la plus prometteuse. Pour cette raison, on peut trouver plusieurs bases de données comme [36] qui est utilisées largement en reconnaissance du locuteur et qui comporte un grand nombre de locuteurs, elle contient 630 locuteurs (438 Hommes/ 192 Femmes). D'autres bases de données sont utilisées en reconnaissance automatique du locuteur comme [27], [33], [18], etc.

Mais ces bases de données sont très coûteuses, pour cette raison nous avons décidé de réaliser notre propre base de données. Ce travail a été réalisé en prononçons quelques lettres amazighs comme illustré sur la figure 4.5. La base contient 2 locuteurs, dans chaque locuteur prononce



Figure 4.5 – Caractères de Tifinagh adoptés.

33 lettres (figure 4.5) comme illustré dans le tableau suivant : Le but d'apprentissage est de classifier les locuteurs selon le genre.

Genre	Base d'apprentissage	Base de test
Féminin	17	16
Masculin	16	17
Total	33	33

4.2.3 Paramétrisation des données

Nous utilisons une paramétrisation standarde de 13 coefficients LPCC. Cependant, en suivant les travaux de [6] qui montrent que les derniers coefficients LPCC et les coefficients dynamiques suffisent à caractériser un locuteur dans une tâche de vérification indépendante du texte en souhaitant la même chose pour l'identification du locuteur dépendante du texte.

4.2.4 Prédiction par PMC

Les réseau de neurones artificiels (ANN) est une architecture informatique non algorithmique inspirée par une recherche sur la façon dont le cerveau encode et traite l'information. Les tâches comme la vision artificielle et la reconnaissance de la parole peuvent être automatisées

si notre ordinateur fonctionne plus comme un cerveau humain. Comme les humains peuvent reconnaître facilement les voix individuelles, l'application des RNA dans la reconnaissance des locuteurs semble être une approche prometteuse. La meilleure façon de comparer les différentes méthodes de reconnaissance des haut-parleurs consiste à comparer leur performance de reconnaissance, leur temps de calcul et leur taille de stockage en fonction de la même base des données.

La figure 4.6 montre un prédicteur MLP de modèle de parole. Il produit un vecteur caractéristique prédit en utilisant l'entrée précédente, les vecteurs de caractéristiques vocales $a_{t-1},...,a_{t-\tau}$ en tant qu'entrées. Les indices représentent des nombres de trames de vecteurs de fonctions de parole. Le symbole τ indique le nombre de vecteurs de caractéristiques de la parole d'entrée utilisés pour la prédiction.

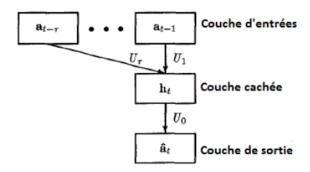


Figure 4.6 – Le prédicteur PMC pour la reconnaissance du locuteur.

La figure 4.7 suivante présente les différentes étapes de la reconnaissance du locuteur par le perceptron multi-couches,

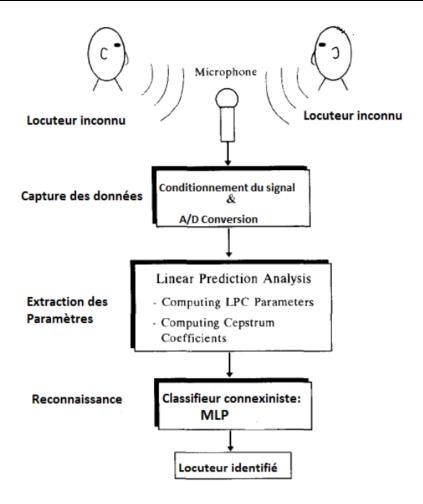


Figure 4.7 – Les étapes principales de la reconnaissance du locuteur.

4.2.5 Implémentations du test croisé et quelques résultats numériques

Les résultats que nous présentons ici ont été obtenus à l'aide de la base de données étiquetées, présentée dans le paragraphe précédant. Pour rappel, cette base est composée de 66 enregistrements.

L'apprentissage est effectué par l'algorithme de rétro-propagation avec un pas d'apprentissage $\epsilon=0.25$.

4.2.5.1 Apprentissage sans régularisation

Les réseaux de neurones testés comportent de 1 à 4 neurones cachés. Nous n'avons pas mis en oeuvre de réseaux comportant plus de 4 neurones cachés en raison du nombre de paramètres à ajuster que cela implique. En effet, notre base de données totale ne comporte que soixante exemples, alors qu'un réseau de neurones à treize entrées, quatre neurones cachés et deux neurones de sortie, comporte $(13+1)\times 4+(4+1)\times 2=66$ paramètres à ajuster. Ce nombre est exactement égale au nombre d'exemples, ce qui crée un risque de sur-ajustement très important pour des modèles dont l'apprentissage est effectué sans régularisation.

Les réseaux de neurones à une couche cachée obtenant le meilleur score de validation croisée, pour chaque ensemble de test croisé, sont ceux comportant quatre neurones cachés, comme illustré Figure 4.8 pour trois tests pris au hasard.

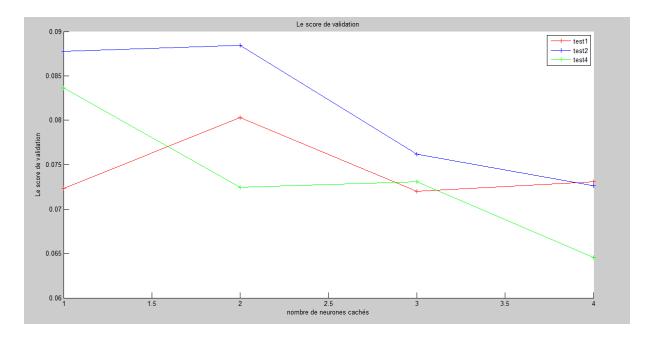


Figure 4.8 – Scores de validation croisée des réseaux, présentés pour trois ensembles de test, en fonction du nombre de neurones cachés qu'ils comportent.

Afin de pouvoir ultérieurement évaluer l'impact de la méthode de régularisation sur les performances des classifieurs, la moyenne des performances obtenues en test par les dix classifieurs sans régularisation égale à 1.88.

4.2.5.2 Apprentissage avec régularisation

La construction des classifieurs conventionnels, avec régularisation par modération des poids, nécessite non seulement la détermination du nombre de neurones cachés (N_{NC}) du réseau de neurones mais également celle du paramètre de régularisation λ . Ces deux paramètres ont été conjointement déterminés par validation croisée de 5 plis. Nous avons testé les valeurs suivantes pour le paramètre de régularisation : $\lambda = 0$, 05; 0, 10 et 0, 15 avec un N_{NC} variant de 1 à 4.

À l'issue des procédures de validation croisée, le paramètre de régularisation, sélectionné pour chacun des dix modèles construits, était : 0.15. Le nombre de neurones cachés sélectionné est égale à 4 (Figure 4.9).

La moyenne des performances obtenues par les dix classifieurs conventionnels, construits selon la méthode de test croisé avec régularisation, sur les plis de test correspondants est égale à 1.09.

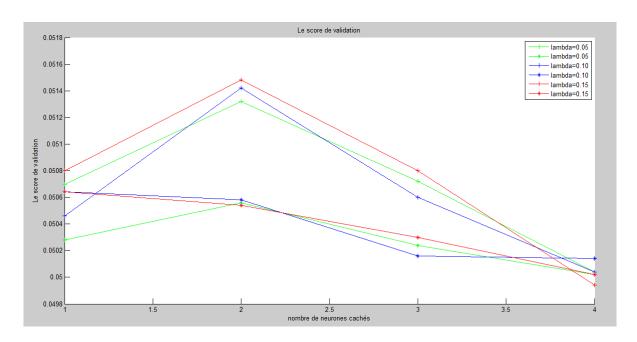


Figure 4.9 – Scores de validation croisée des réseaux, présentés pour trois ensembles de test, en fonction du nombre de neurones cachés qu'ils comportent.

4.2.5.3 Commentaires

En comparant la performance de deux modèles à partir de la moyenne des performances obtenues en test on voit que le modèle régularisé est plus performant que celui sans régularisation. les paramètres optimaux choisis par la méthode sont $N_{NC}=4$ et $\lambda=0.15$.

Le tableau suivant montre le taux de reconnaissance obtenu par le modèle sélectionné en fonction de la taille de la base du test :

Taille d'ensemble de test en %	50	30	20	10
Taux de reconnaissance en %	68	70	65	80

Conclusion

Dans ce chapitre, nous avons introduit le principe de la reconnaissance automatique du locuteur ainsi que les différentes étapes du système. La reconnaissance automatique du locuteur est probablement la méthode la plus ergonomique pour résoudre les problèmes d'accès. Cependant, la voix ne peut être considérée comme une caractéristique biométrique d'une personne compte tenu de la variabilité intra-locuteur. Un système de reconnaissance de locuteur procède généralement en trois étapes : l'analyse acoustique du signal de la parole, la modélisation du

locuteur et une dernière étape de décision. En analyse acoustique, les MFCC sont les coefficients acoustiques les plus répandues. Quant à la modélisation l'approche GMM constitue l'état de l'art en RAL, en mode indépendant du texte. La décision d'un système de reconnaissance automatique du locuteur est basée sur les deux processus d'identification et/ou de vérification de locuteur, et cela quelle que soit l'application ou la tâche visée.

Nous avons ainsi présenté notre approche basée sur l'extraction des caractéristiques d'un signal parole, ensuite nous avons appliqué la méthode du test croisé pour sélectionner le nombre du neurones cachés dans un perceptron multi-couches, nous avons remarqué que la reconnaissance à l'aide d'un réseau de neurones artificiel donne les meilleurs taux de reconnaissance.

CONCLUSION ET PERSPECTIVES

Au cours de ce travail, nous avons traité le problème de la reconnaissance du locuteur par les réseaux de neurones, plus précisément par le perceptron multi-couches.

Nous avons crée une base des données en prononçant quelques lettres Amazighs , à l'aide de la méthode LPC nous avons fait l'extraction des paramètres afin de faire la classification de deux locuteurs selon le genre s'il s'agit d'un homme où d'une femme.

À partir de la base obtenue, une nouvelle méthode de conception et d'évaluation d'un modèle statistique a été appliquée : la méthode de test croisé, imposée par la petite taille de la base disponible. À l'aide de cette méthode, un classifieur conventionnel de réseaux de neurones à une couche cachée a été conçu et évalué.

Comme perspectives de notre travail, nous souhaitons de réaliser une autre base de données mais cette fois avec un nombre de locuteurs supérieur à 2 et en la testant par différentes architectures des réseaux de neurones, et d'appliquer d'autres méthodes de la sélection des modèles neuronales et en les comparant par notre approche de test croisé.

BIBLIOGRAPHIE

- [1] K.Barktova, Production, decription et perception du signal vocal. Rapport technique, France Télécommunication R and D, 2002.
- [2] F.Bimbot, I.Magrin-Chagnolleau, et L.Mathan, Second-order statistical measures for text-independent speaker identification. Speech Communication, pages: 177-192, 1995.
- [3] M. Ben, Approches robustes pour la vérification du locuteur par normalisation et adaptation hiérarchique. Thèse de Doctorat, Université de Rennes 1,2004.
- [4] Booth,I., Barlow,M., et Watson, B.,Enhancements to DTW and VQ decision algorithms for speaker recognition. Speech Communication, 1993.
- [5] L. BREIMAN, "Heuristics of Instability and Stabilization in Model Selection" Annal of Statistics 24, pp. 2350-2383, 1996.
- [6] D. Charlet et D. Jouvet, Optimisation du paramétrage acoustique pour la vérification du locuteur. In Acte JEP 96, pages 399–402, 1996.
- [7] G.CYBENKO, "Approximation by superpositions of a sigmoidal function", Mathematics of Controls. Signals and Systems, Vol.,pp.303-314. 1989.
- [8] Davalo, E., Naïm, P., Des réseaux de neurones, Eyrolles, Paris, 1989.
- [9] G. Dreyfus, J.M. Martinez, M. Samuelides, M.B. Gordon, F. Badran, S. Thiria, Apprentissage statistique: Réseaux de neurones Cartes topologiques Machines à vecteurs supports: Eyrolles, 2011.
- [10] G. Dreyfus.L.Personnaz, "Perceptrons, Past and Presnt", Enciclopedia Italiana, in press. TOULOUSE, 1997.

- [11] . Ettaouil, M. Lazaar, "Compression of Medical Images using Improved Kohonen Algorithm", International Journal of Computer Applications (IJCA) Special Issue on Software Engineering, Databases and Expert Systems SEDEX, September 2012, (1): 41-45.
- [12] . Ettaouil, K. El Moutaouakil, "Improved Optimal Competitive Hopfield Network for the Maximum Stable Set Problem", International Journal on Computer Science and Engineering, 2010, 2(6): 2071-2077.
- [13] . Ettaouil, Y. Ghanou, K. Elmoutaouakil, M. Lazaar, "Medical Image Compression by A new Architecture Optimization Model for the Kohonen Networks", International Journal of Computer Theory and Engineering, 2011, 3(2): 204-210.
- [14] . Ettaouil, Y. Ghanou, K. Elmoutaouakil, M. Lazaar, "A New Architecture Optimization Model for the Kohonen Networks and Clustering", Journal of Advanced Research in Computer Science (JARCS), 2011, 3(1): 14-32.
- [15] K.FUNAHASHI,"On the approximate realization of continous mappings by neural networks", Neural Networks. Vol.2.pp.183-192,1989.
- [16] M. Geist, Précis introductif à l'apprentissage statistique. Support de cours, Supélc, 2011.
- [17] S. Geman, E. Bienenstock, R. Doursat, "Neural Networks and the Bias/Variance Dilemma," Neural Computation, vol. 4, pp. 1-58, 1992.
- [18] M. Falcone et A. Gallo, « The Idquo; SIVA rdquo; speech database for speaker verification: description and evaluation », in , Fourth International Conference on Spoken Language, 1996. ICSLP 96. Proceedings, 1996, vol. 3, p. 1902-1905 vol. 3.
- [19] S.Furui, Cepstral analysis technique for automatic speaker verification, Dans IEEE Transactions Acoustics, Speech, and Signal Processing, Volume 29, pages 254-272, 1981.
- [20] S. Geman, E. Bienenstock, R. Doursat, "Neural networks and the bias/variance dilemma," Neural Computation, vol. 4(1), pp. 1-58, 1992.
- [21] H. Gish, "A probabilistic approach to the understanding and training of neural network classifiers," Acoustics, Speech, and Signal Processing, vol. 3, pp. 1361-1364.
- [22] D.Hebb, Organization of Behavior, New York: Wiley, 1949.
- [23] K.HORNIK, M.Stinchcome, H.White and P.Auer, "Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives", Neural Computation, Vol.6,n 6,pp.1262-1275,1994.

- [24] PA. Lachenbruch, MA. Mickey, "Estimation of Error Rates in Discriminant Analysis", Technometrics, vol. 10(1), pp. 1-11, 1968.
- [25] R.P. Lippmann, "Neural Networks, Bayesian a posteriori Probabilities, and Pattern Classification," From Statistics to Neural Networks. vol. 136, pp. 83-104, 1994.
- [26] I.Magrin-Chagnolleau, J.Bonaqtere, et F.Bimbot, Effect of utterance duration and phonetic content on speaker identification using second-order satistical methods. Dans European Conference on Speech Communication and Technology (EUROSPEECH), volume 1, pages 337-340 (1995).
- [27] H.Melin. « Gandalf-a Swedish telephone speaker verification database », in , Fourth International Conference on Spoken Language, 1996. ICSLP 96. Proceedings, 1996, vol. 3,
- [28] J.E. Moody, "The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems", in Moody, J.E., Hanson, S.J., and Lippmann, R.P., Advances in Neural Information Processing Systems 4, pp. 847-854,1992.
- [29] J.Oglesby et S.Mason, Speaker recognition with a neural classifier. Dans First IEE International Conference, pages :306-309,1989.
- [30] M. PLUTOWSKI, S. SAKATA and H. WHITE "Cross-Validation Estimates IMSE" Advances in Neural Information Processing Systems 6, San Mateo, CA: Morgan Kaufmann Publishers, 1994.
- [31] D.Reynolds, Speaker identification and verification using gaussian mixture speaker models. Speech Communication, page 91-108, 1995.
- [32] M. D. Richard and R. P. Lippmann, "Neural Network Classifiers Estimate Bayesian a posteriori Probabilities," Neural Computation, vol. 3, pp. 461-483, 1991.
- [33] M. N. Ronald Cole, « The Cslu Speaker Recognition Corpus », 1999.
- [34] Sànchez-Sinencio, E., Lau, C., Artificial Neural Networks, IEEE Press, 1992.
- [35] M. Stone, "Cross-validation choice and assessment of statistical predictions." Journal of the Royal Statistical Society. Series B, vol. 36(2), pp. 111-148, 1974.
- [36] « TIMIT Acoustic-Phonetic Continuous Speech Corpus Linguistic Data Consortium ». [Enligne]. Disponible sur : https ://catalog.ldc.upenn.edu/LDC93S1. [Consulté le : 16-mai-2016]. [29] H. Melin, « Gandalf-a Swedish telephone speaker verification database », in , Fourth International Conference on Spoken Language, 1996. ICSLP 96. Proceedings,

- 1996, vol. 3, p. 1954-1957 vol.3.
- [37] V.N. Vapnik, Statistical Learning Theory. Wiely-Interscience, September 1994.
- [38] C. Wang, S.S. Venkatesh, J.S. Judd, "Optimal Stopping and Effective Machine Complexity in Learning," Advances in Neural Information Processing Systems 6, pp.303-310, 1994.
- [39] Yassine Mami, "Reconnaissance de locuteurs par localisation dans un espace de locuteurs de référence", Thèse doctorale de l'Ecole nationale supérieure de télécommunication, 2004.
- [40] E.Zwicker et R.Feldtkeller Psychoacoustique CNET/ENST. Cllection technique et scientifique des télécommunicatios, Masson Paris, 1981