UNIVERSITE PARIS VI

Habilitation à Diriger des Recherches

Spécialité Génie Informatique, Automatique et Traitement du Signal

Méthodes neuronales pour l'extraction de caractéristiques non linéaires et discriminantes : application aux signaux de parole

présentée par

Bruno GAS

le 23 novembre 2005

devant le jury composé de :

Régine ANDRE-OBRECHT	Raporteur
Gérard CHOLLET	Examinateur
Christian JUTTEN	Raporteur
Gernot KUBIN	Examinateur
Jean-Sylvain LIENARD	Raporteur
Maurice MILGRAM	Examinateur
Jean-Luc ZARADER	Examinateur

Laboratoire des Instruments et Systèmes d'Ile de France

.

Table des matières

.

.

Résumé

L'extraction de caractéristiques du signal de parole est un domaine de l'analyse du signal peu exploré par la communauté de recherche en parole. La raison principale en est que nous disposons aujourd'hui d'outils performants : des outils essentiellement fondés sur l'analyse fréquentielle des signaux pour leur paramétrisation et des outils d'analyse statistique pour leur classification.
Les applications en reconnaissance de la parole les plus évoluées se heurtent cependant à de grandes difficultés lorsqu'il s'agit de traiter de signaux en environnement fortement perturbé comme la téléphonie cellulaire par exemple. Des auteurs ont souligné récemment l'importance de revenir sur certains aspects de la chaîne de traitement, l'extraction de caractéristiques en particulier, quitte à renoncer, pour un temps au moins, à la course aux performances en terme de scores de reconnaissance.
Le travail présenté dans ce mémoire propose de reprendre les toutes premières étapes de traitement des systèmes de reconnaissance de la parole, à savoir l'extraction de caractéristiques et la classification phonétique. Une nouvelle modélisation permettant de prendre en compte les caractéristiques non linéaires du processus de production de la parole est proposée. Fondée sur l'utilisation d'un perceptron multicouches, elle permet de sursoir aux limites bien connues des systèmes connexionnistes appliqués à la modélisation non linéaire des signaux :

- la complexité des calculs requis, difficilement compatible avec les applications temps réel ;
- la non unicité des solutions obtenues ;
- la multiplicité des paramètres libres requis.

Un ensemble de validations expérimentales est proposé et un projet de recherche portant sur l'utilisation conjointe de cartes auto-organisantes prédictives et de perceptrons multicouches pour l'extraction non supervisée de caractéristiques est présenté.

Mots clés

Traitement non linéaire du signal, réseaux de neurones, extraction de caractéristiques, reconnaissance de phonèmes, perceptrons multicouches, cartes auto-organisantes

.

Première partie

Contributions

.

Chapitre 1

Introduction

1.1 Parcours de recherche

Mon premier contact avec la recherche remonte au stage de DEA que j'ai effectué à la Sagem en 1990 [58]. Ce stage avait pour principal objet la mise en oeuvre d'un modèle de Hopfield pour l'estimation de la vitesse d'objets en mouvement dans des séquences d'images infrarouges. Je souhaitais alors, avant tout, travailler sur les modèles connexionnistes, découverts avec grand intérêt lors de mon DEA. Intéressé par les sciences du vivant d'une part, attiré par la physique d'autre part, je trouvais là l'occasion de renouer avec la perspective d'une recherche plus fondamentale, à l'issue de mon cursus en EEA. J'entreprends donc une thèse de doctorat dans ce domaine à l'École Supérieure d'Electronique et d'Electrotechnique (ESIEE) de Marne-la-Vallée, sous la direction de René Natowicz et Gilles Bertrand du département informatique de l'ESIEE. Je souhaitais tout particulièrement traiter de la plausibilité biologique des modèles neuronaux en tirant parti des avancées récentes dans le domaine de la biologie (travaux de Changeux et Danchin), tout en me plaçant dans la lignée des travaux du courant cybernétique (conférences Macy) ainsi que du plus récent mouvement néo-connexionniste. Les réflexions induites de mes lectures m'ont conduit à suivre un certain nombre d'hypothèses comme la nécessité d'une approche *spike*, c'est à dire d'une activités binaires des cellules sous forme de trains d'impulsions, la nécessité d'un apprentissage non supervisé, ou tout du moins non guidé par un objectif extérieur, la nécessité de traiter des signaux évoluant dans le temps (en contexte), ou encore d'avoir une structure de réseau obligatoirement récurrente, etc.
Cette recherche m'a donc conduit à proposer un nouveau modèle de réseau que j'ai appliqué à la reconnaissance de séquences temporelles [59]. Les deux particularités du modèle étaient, pour une part, la propriété de chaque cellule de pouvoir fonctionner selon deux modes d'activité : *évoqué* (neurone de MacCullogh et Pitts) et *spontané* (émission aléatoire d'activité, en accord avec des expériences reportées par Changeux sur l'émission spontanée de cellules neuronales du cerveau), et pour une autre part de définir les liaisons synaptiques comme des intégrateurs de l'activité pré-synaptique (filtres moyenneurs) et dont seul le temps d'intégration faisait objet de l'apprentissage. Je me suis plus particulièrement penché sur l'étude de l'entrée en résonance d'un tel réseau en contact avec des séquences extérieures déterministes, mais également sur l'observation de l'émergence de séquences déterministes dues au seul fait des interactions inter-cellulaires dans un réseau

fermé (ni entrées, ni sorties) et initialement aléatoire. L'ensemble de ces travaux on fait l'objet de 7 publications dans des conférences internationales avec comité de lecture.

Au mois de septembre 1994, je suis recruté en qualité d'Attaché Temporaire d'Enseignement et de Recherche à l'ENSEA (Ecole Nationale Supérieure de l'Electronique et de ses Applications). J'y effectue une année de recherche au laboratoire ETIS (Equipe Traitement des Images et du Signal) sous la direction de Jean-Pierre Coquerez, au sein de l'équipe de Philippe Gaussier, dans le domaine de la robotique autonome (approche *animats*). C'était donc un changement de thématique offrant une perspective de recherche très intéressante relativement à mes centres d'intérêt et travaux antérieurs : la prise en compte de la dimension de l'*action* en *contexte*. Les centres d'intérêts de Philippe Gaussier étaient très proches des miens : contribution à une *nouvelle robotique*, dont les maîtres mots étaient l'*autonomie* des agents, leur interaction réciproque, à partir de modèles neuronaux, biologiquement plausibles, comme par exemple les cartes de Kohonen probabilistes, et formalisée autour de la structure PER-AC : *PERception-ACtion*. Concernant les données de travail, je passais ainsi des séquences temporelles à une dimension aux séquences d'images, réelles ou de synthèse, prises à l'aide d'une caméra placée sur un robot mobile.

1.2 Thématique actuelle

Changement de signaux, mais également changement de thématique : recruté à l'Université Pierre et Marie Curie en 1995, je rejoins le laboratoire PARC (Perception, Automatique et Réseaux Connexionnistes) dirigé par Maurice Milgram pour travailler dans le domaine de l'analyse du signal appliquée à la reconnaissance du locuteur et de la parole. Les méthodologies à partir desquelles je travaillerai pendant ces 10 années seront plus "conventionnelles", me permettant d'approfondir différents domaines du traitement du signal, essentiellement le filtrage adaptatif, mais également l'analyse statistique des données et la reconnaissance des formes (méthodes globales et structurelles, inférence bayesienne, chaînes de Markov cachées, modèles connexionnistes, etc.). En 1997 le laboratoire PARC devient le groupe PARC du Laboratoire des Instruments et Systèmes d'Ile de France (LISIF), laboratoire nouvellement créé par le rapprochement de trois laboratoires de l'UFR 924 de Physiques et de ses applications : le PARC, le LEAM et une équipe de l'ESPCI. L'activité de notre équipe tournera autour de plusieurs thématiques liées aux méthodes adaptatives de traitement du signal (traitement de signaux LIDAR pour l'estimation de la vitesse des vents atmosphériques) et aux méthodes connexionnistes de type perceptron multicouches (détection de la présence de passagers pour l'activation/inhibition d'un airbag automobile), rééducation neuro-orthopédique par interfaces robotiques (projet RENOIR en collaboration avec le Laboratoire de Robotique de Paris 6). Cependant, mon implication restera essentiellement dans le domaine de l'extraction de caractéristiques du signal de parole.

L'analyse du signal de parole, première étape de traitement dans les systèmes de reconnaissance automatique, que ce soit de parole, de locuteur ou de langue, est une étape fondamentale relativement à la robustesse du système global. Cependant, elle reste peu traitée par la communauté internationale. Les systèmes de l'état de l'art fonctionnent, pour la plupart, à l'aide d'une modélisation court-terme du spectre basée sur l'estimation

des paramètres MFCC (Mel Frequency Cepstrum Coding) qui offrent de bons résultats en terme de robustesse et de distribution. Du fait de l'extrême variabilité du signal de parole, les acteurs de la recherche considèrent comme acquis l'insuffisance d'une approche qui resterait strictement bottum-up : l'ensemble des sons distinctifs, en termes de signification pour une langue particulière, c'est à dire l'ensemble des phonèmes, permet de décrire exactement le discours parlé, mais une décision basée sur la reconnaissance de telles entités à partir du signal ne peut pas être obtenue sans un minimum d'erreur. Par ailleurs, il n'est pas nécessaire de connaître avec exactitude la catégorie phonétique des trames, prises individuellement, puisqu'il existe des contraintes "contextuelles" à différents niveaux, phonétique, syntaxique, sémantique et pragmatique permettant de réduire les ambiguïtés, et finalement de conduire à une décision globale. Aussi, les méthodes statistiques ont-elles été considérablement mises en avant depuis le début des années 80. Des méthodes qui excellent dans l'apprentissage de grandes bases de données représentant des situations opérationnelles bien modélisées, mais qui généralisent mal aux nouveaux environnements. La robustesse reste donc un thème de recherche majeur, comme peuvent l'illustrer les thématiques de certaines conférences internationales en parole. Elle est alors définie comme la capacité d'un système à maintenir ses performances lorsqu'il est exposé à des conditions mal représentées dans les données d'apprentissage. Les sources de dégradation peuvent être dues aux caractéristiques du locuteur (accent, dialecte, état, ...), à la chaîne de transmission (téléphone filaire, téléphone cellulaire, voie sur IP, audiovisuel), aux conditions d'acquisition (bruits d'environnement, systèmes d'acquisition utilisés, etc.). A titre d'exemple, en téléphonie cellulaire on trouve de nombreuses formes de bruits non stationnaires, et des variations significatives du type et de la position du microphone. De même, les données audiovisuelles comportent de nombreuses variations en termes de contenu, de locuteur, de dialecte et d'accent, de canal utilisé, etc.

Des auteurs influents comme Hermansky et Bourlard ont posé le problème de revoir les méthodologies utilisées, allant jusqu'à proposer de cesser la « course aux performances » qui a tendance à s'afficher au gré des campagnes d'évaluation des systèmes. Ne vaudrait-il pas mieux mettre en œuvre de nouvelles méthodologies qui, au prix d'une baisse possible des scores, permettraient tout au moins d'explorer de nouvelles voies, potentiellement prometteuses ? Depuis 10 ans, les travaux en parole de notre équipe s'inscrivent dans ce contexte. Ils ont pour objet la définition et l'analyse de nouvelles méthodes d'extraction de caractéristiques du signal. Les applications envisagées sont la reconnaissance de phonèmes, du locuteur, de la langue, mais également, dans une moindre mesure, la compression/décompression. Ces recherches restent suffisamment génériques pour pouvoir être appliquées à d'autres domaines du traitement du signal comme par exemple l'estimation de fréquences doppler sur des signaux LIDAR (§4.3.5, p. 86).

Deux voies de recherche ont été privilégiées : l'analyse non linéaire du signal de parole d'une part et l'adaptation de l'analyse à la tâche de classification d'autre part. Cela nous a conduit à proposer une nouvelle méthode d'extraction de caractéristiques (les paramètres *NPC* ou le *codage prédictif neuronal*). La méthodologie employée est celle de réseaux de neurones de type Perceptron Multicouche (MLP), exploités en tant que filtres adaptatifs non linéaires. Le mémoire présente donc le modèle original ainsi qu'un ensemble d'extensions visant à améliorer le caractère discriminant des paramètres générés. La co-

opération des deux étapes que sont l'analyse du signal et la classification est ainsi étudiée sous différents angles, mettant en exergue la problématique essentielle qu'est le rapport modélisation/discrimination.
Notre démarche ne prétend cependant pas être aussi radicale que la position d'Hermansky le laisse penser dans «Should recognizers have ears ?» [76]. Le principe de la segmentation du signal de parole en trames *stationnaires* de longueur fixe (approche court-terme) n'est par exemple jamais remis en cause. J'esquisserai dans les conclusions et perspectives de ce mémoire (§4.4, p. 88) des orientations de recherche remettant plus fondamentalement en cause les approches dominantes actuelles.

Enfin, l'ensemble des résultats obtenus, les réflexions menées, les programmes écrits et les articles rédigés, et dont je retrace l'historique ici, sont le fruit du travail collectif d'une équipe composée de deux permanents, Jean-Luc Zarader, Professeur, et moi-même, Maître de Conférences, mais également de quatre étudiants en thèse, Cyril Chavy, Mohamed Chetouani, Sébastien Herry et Christophe Charbuillet et de pas moins de 17 jeunes stagiaires de DEA ou de troisième année d'école d'ingénieur que j'ai encadrés, avec qui j'ai eu la chance de pouvoir travailler tout au long de ces dix années.

1.3 Plan du mémoire

Le rapport d'Habilitation se présente sous la forme de quatre chapitres : le présent chapitre 1 d'introduction suivi des trois chapitres 2, 3 et 4.
Le chapitre suivant est consacré aux paramètres NPC et à la modélisation non linéaire. Différentes problématiques propres à l'extraction de caractéristiques sont exposées avant d'être placées dans le contexte de l'analyse non linéaire du signal de parole. L'évidence de la présence de non linéarités dans le signal de parole est ensuite discutée. J'expose alors le principe du codage prédictif neuronal (NPC) que nous avons developpé ainsi que les premières validations expérimentales menées. Le chapitre s'achève sur une étude du sur-apprentissage inhérent à ce type d'approche. L'ensemble de ces travaux s'articulent autour de la thèse de Cyril Chavy, entre 1996 et 2000.

Le chapitre 3 traite des algorithmes d'adaptation NPC-C, NPC-2, DFE-NPC et LVQ-NPC permettant l'adaptation de l'extracteur à la tâche de classification. Un critère équivalent au critère de maximisation de l'information mutuelle, mais fondé sur un rapport d'erreurs de prédiction, le critère MER, est proposé ainsi qu'une nouvelle mesure de dissemblance appelée *distance NPC*. La maximisation du MER conduit aux algorithmes DFE-NPC et les gains attendus de cette démarche sont l'obtention de paramètres NPC à pouvoir discriminant plus élevé, relativement au problème posé. La démarche suivie est guidée par le soucis d'établir une coopération explicite entre l'extracteur et le classifieur. L'erreur de prédiction et de classification sont alors simultanément minimisées, au sein d'une même architecture d'apprentissage : c'est l'algorithme LVQ-NPC. Ces travaux ont été réalisés dans le cadre de la thèse de Mohamed Chetouani de 2001 à 2004.

Le chapitre 4 présente une analyse théorique du modèle NPC suivie de la description du dernier modèle d'adaptation SOM-NPC proposé. Ce dernier incorpore au sein d'un même

module l'étape d'extraction et l'étape de classification et reprend le principe des cartes auto-organisantes de Kohonen, autorisant ainsi une adaptation éventuellement non supervisée de l'extracteur. Ce modèle est analysé en tant que nouvelle perspective de recherche dans le domaine de l'extraction de caractéristiques. Des projets réalisés, ou en cours de réalisation, ainsi que diverses collaborations, sont ensuite présentés, qui s'articulent autour de deux étudiants actuellement en thèse, Sébastien Herry jusqu'en décembre 2005 et Christophe Charbuillet de 2004 à 2007. Le chapitre se termine par un ensemble de questions ouvertes que j'expose en forme de propositions pour un nouveau projet de recherche.

.

Chapitre 2

Les paramètres NPC (Neural Predictive Coding)

Encadrement doctoral : Cyril Chavy (1997-2004)

L'analyse du signal de parole se résume aujourd'hui essentiellement en la modélisation, paramétrique ou non, du spectre court-terme. Peu de travaux de recherche ont été dédiés à l'amélioration des qualités discriminantes des paramètres. J'expose dans ce premier chapitre les raisons qui m'ont conduit à traiter ce thème particulier ainsi que le contexte historique et scientifique dans lequel il se place. J'y introduis les paramètres NPC proposés en 1997 et je développe la problématique du traitement non linéaire des données de parole telle qu'elle s'est posée à nous, mais aussi à une certaine communauté, très réduite, de chercheurs. Le travail de thèse de Cyril Chavy est central dans cette approche puisqu'il a étudié en détail, mis en oeuvre et validé expérimentalement la première version de l'extracteur NPC. Il proposera aux cours de ses travaux une deuxième version des paramètres (les paramètres NPC-2) ouvrant la voie vers la coopération des deux étages que sont l'extraction de caractéristiques et la classification.

2.1 Un nouveau sujet de recherche

Encadrement : Cyril Chavy (DEA d'électronique 1996), Bruno Breton (DEA d'électronique 1997)
Publications : [22], [61]

Démarrer un nouveau thème de recherche nécessite un certain temps, le temps de trouver ses repères dans la communauté, d'analyser les différents travaux et d'en appréhender les directions porteuses. L'année de mon recrutement (1995) a été pour moi l'année de la prise de contact avec ce nouveau domaine de recherche qu'est le traitement du signal de parole dans toute sa généralité. Ce n'est qu'en 1996 que je me suis posé le problème plus particulier de l'extraction de caractéristiques. L'explication de cette démarche nécessite de rappeler le contexte qui était celui du laboratoire de l'époque.

L'équipe *Signal* dans laquelle je me suis inséré à mon arrivée était forte d'une solide expérience en traitement du signal, en méthodes connexionnistes et en traitement de la

parole. L'ensemble de ces disciplines étaient également enseignées dans les différentes formations liées aux activités du laboratoire, notamment au sein du DEA d'Electronique et du DEA de Robotique de Paris 6. Jean Luc Zarader, responsable de notre équipe depuis 1997, avait auparavant travaillé sur la mise en oeuvre de modèles de filtres et d'algorithmes adaptatifs pour l'estimation de la fréquence Doppler de signaux Lidar [146]. Puis, en collaboration avec Maurice Milgram, il avait étudié l'application des modèles connexionnistes au traitement du signal. Il avait en particulier débuté une activité de recherche concernant la reconnaissance du locuteur à l'aide de modèles de réseaux dynamiques à poids fonction du temps [23] ainsi que par des méthodes connexionnistes prédictives [22].
Les systèmes non linéaires connexionnistes pour l'analyse du signal de la parole étaient donc une méthodologie choisie déjà quelques années avant mon arrivée au laboratoire. Ayant précédemment moi-même travaillé à l'élaboration de modèles connexionnistes appliqués à la reconnaissance de séquences temporelles, c'est tout naturellement que je me suis intégré dans l'équipe. Mon expérience en traitement du signal de parole étant alors faible, j'ai pu m'appuyer sur celles acquises par mes collègues pendant ces années.

Le principe de la classification par des réseaux prédictifs est fondé sur une approche dite par *modélisation*. Cette méthodologie de la Reconnaissance Des Formes (RDF), consistant à associer à une forme inconnue le modèle le plus vraisemblable, a été étudiée en détail par Maurice Milgram et son équipe en reconnaissance de caractères (réseaux *diabolos*). Elle s'inscrivait donc dans la "culture" du laboratoire.
Un réseau prédictif est un réseau dont la tâche consiste à prédire les composantes d'un vecteur acoustique à l'instant t on fonction de ses valeurs échantillonnées aux instants précédents sur un horizon fini $t-1$, $t-2$,..., $t-\lambda$. Le plus souvent $\lambda = 2$. Pour un problème donné de classification de signaux, on affecte un ou plusieurs réseaux prédictifs à chacune des classes de signaux à modéliser. Artières [5] a utilisé cette approche en reconnaissance du locuteur à l'aide de perceptrons multicouches (PMC). Les résultats obtenus à partir de modèles à poids fonction du temps, de réseaux RBF et de réseaux PMC ont montré l'intérêt de l'approche prédictive. Ils ont également mis en évidence une forte dépendance des performances obtenues en fonction des paramètres utilisés (MFCC, LPC, BF, etc.) [145]. Cet aspect à été essentiel dans la suite puisqu'il nous à conduit à revoir l'étape d'extraction des caractéristiques qui allait devenir le thème central de mes recherches.

2.1.1 De la validité des comparaisons.

Forts du succès obtenu par les méthodes de classification prédictive, nous avons souhaité les appliquer au problème de la reconnaissance de phonèmes. La base TIMIT récemment acquise par le laboratoire nous en offrait l'opportunité puisqu'elle était intégralement segmentée. C'était l'objet du stage de DEA de Cyril Chavy que j'ai co-encadré avec Jean Luc Zarader.
Le système proposé comprenait 7 réseaux de type PMC (Multi-Layer-Perceptron) avec une règle d'apprentissage du type rétropropagation du gradient [105],[127]. Nous avons obtenu des scores en généralisation compris entre 60% et 65% que nous avons publiés à l'issue du stage [61]. Ces résultats restaient malgré tout inférieurs aux taux de reconnaissance trouvés dans la littérature. Waibel et Lang [138], [103] obtenaient par exemple des scores de 98% avec le TDNN (Time Delay Neural Network) contre 93% obtenus à l'aide de modèles HMM

(Hidden Markov Models). Les conditions expérimentales étaient assurément différentes. Waibel présentait des résultats sur un nombre de classes très restreint (3 phonèmes voisés acoustiquement proches /b/, /d/, /g/) et les bases de données utilisées n'étaient pas la base TIMIT. En 1992, S. Young [142] obtenait "seulement" 61.7% de phonèmes reconnus à l'aide de modèles HMM à trois états pour 48 modèles et en mode indépendant du contexte sur la base TIMIT. Les premiers travaux en la matière (juste après la mise à jour de la base TIMIT en 1988) revenaient cependant à Kai-Fu Lee [88] qui regroupait en 39 classes les 64 labels disponibles de la base. En mode indépendant du contexte, il obtenait 64,07% de bonne reconnaissance pour 48 modèles HMM appris. Ce taux pouvait monter à 73.80% en mode dépendant du contexte (1450 modèles HMM). On peut également citer en France Frédéric Bimbot [11] qui obtenait sur une base mono-locuteur, qui n'était pas non plus TIMIT mais comportait 34 classes phonétiques, 78% de reconnaissance à l'aide de réseaux TDNNs.

Ces résultats étaient donc bien inférieurs à ceux obtenus par Waibel, mais supérieurs aux nôtres. La question de savoir qu'elle validité on peut accorder à ce type de comparaison est donc posée mais la réponse loin d'être simple, du fait de conditions expérimentales souvent très différentes. La communauté, comme dans beaucoup d'autres domaines, s'en remet aux campagnes d'évaluation, les campagnes NIST étant les plus connues. Or il n'existe pas de campagne spécifiquement dédiée à la reconnaissance de phonèmes, même si ce thème peut apparaître au sein d'une campagne plus vaste. La reconnaissance de phonèmes est en fait une étape intermédiaire dans un processus de traitement beaucoup plus large. Par ailleurs, avec l'avènement des méthodes statistiques, très largement employées aujourd'hui avec les chaînes de Markov cachées, le niveau phonétique ne donne plus lieu à une étape de décision, mais seulement à l'estimation du *treillis phonétique*, c'est à dire à l'estimation de probabilités a posteriori d'appartenance des vecteurs caractéristiques à des classes phonétiques. La décision s'opère à des niveaux de traitement plus élevés, prenant en compte d'autres informations de nature contextuelle, voire syntaxique puis sémantique.
La réalisation d'un système complet de reconnaissance automatique de la parole (RAP) demeurait, et demeure toujours, hors de portée d'une équipe de la taille de la notre. Nous avons donc envisagé d'incorporer nos algorithmes dans des systèmes plus vastes en établissant des collaborations avec d'autres laboratoires (cf. chapitre 4).

Cette incursion dans le domaine de la reconnaissance de phonèmes ne nous a pas conduit à abandonner la reconnaissance du locuteur. En 1996, Bruno Breton a en effet effectué son stage de DEA au sein de notre équipe sur cette thématique. Son objectif était de tester le modèle connexionniste récemment proposé par Sastry [128], en fait un réseau de neurones comportant des *cellules mémoires* permettant de mémoriser des états passés, tant en classification directe qu'en classification par prédiction. Les résultats obtenus ne se sont pas avérés pertinents et par ailleurs, l'apprentissage était extrêmement lent. Cela nous alors conduit à abandonner provisoirement la thématique de la reconnaissance du locuteur, thématique que nous avons repris plus tard à l'occasion de la thèse de Mohamed Chetouani (cf. chapitre 4, §4.3.2, p. 82).
Un point commun à la reconnaissance du locuteur et à la reconnaissance de phonèmes est l'étape d'extraction des caractéristiques. Bien entendu, les caractéristiques pertinentes ne sont pas les mêmes lorsque l'on passe du phonème au locuteur. C'est sur l'extraction de

caractéristiques appliquée à la reconnaissance de phonèmes que, du point de vue méthodologique, je me suis plus particulièrement orienté.

2.2 L'extraction de caractéristiques : un enjeu

Encadrement : Jean-Charles Didiot (DEA d'Electronique 1997)
Publications : [69]

Les systèmes de reconnaissance automatique de la parole utilisent pour la plupart les caractéristiques du signal estimées sur de courtes périodes de temps, typiquement 10 à 20 ms, pendant lesquelles le signal est supposé stationnaire (figures 2.1 et 2.2). Certaines des méthodes d'extraction utilisées aujourd'hui comme les paramètres LPC (Linear Predictive Coding) héritent historiquement des applications de codage de la parole [84], [6]. L'*extraction de caractéristiques* pour la reconnaissance de la parole (ou du locuteur) et le *codage* pour la compression/décompression ou la synthèse de la parole ne pousuivent pourtant pas les même objectifs. L'extraction de caractéristiques vise à réduire autant que faire se peut la forte redondance du signal (ce qui se traduit par une réduction de la dimension de l'espace des données) tout en conservant le maximum d'informations pertinentes pour les niveaux supérieurs de traitement, c'est à dire des informations *discriminantes*. Il n'y a pas nécessité de reconstruire le signal comme c'est le cas pour le codage puisqu'il s'agit de tâches de reconnaissance.

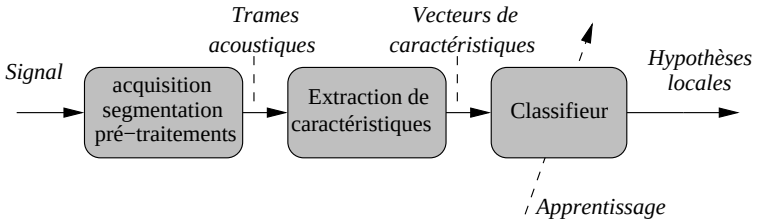


FIG. 2.1 – Schéma de principe des premiers étages de traitement d'un système de RAP

Comme je le souligne plus haut, les motivations qui nous ont conduit à travailler sur le calcul des paramètres sont, entre autre, la disparité des résultats obtenus en reconnaissance du locuteur selon que l'on utilise tel ou tel jeu de coefficients. J'ai donc souhaité effectuer une étude des différentes méthodes d'extraction des caractéristiques proposées dans la litérature, étude qu'à conduite Jean-Charles Didiot dans le cadre de son stage de DEA.

2.2.1 Une foison de paramètres

Le travail de Jean-Charles Didiot était donc d'explorer les différentes techniques d'extraction de caractéristiques sur une tâche de reconnaissance de phonèmes. Il a sélectionné pour ce faire un sous ensemble de 8 classes de phonèmes parmi les plus utilisées dans la langue anglaise. Ces phonèmes étaient extraits de la base Darpa-TIMIT : /*a*/, /*ae*/, /*ey*/, /*ix*/, /*iy*/, /*ow*/, /*s*/ et /*z*/. Tester les méthodes d'extraction de caractéristiques nécessitait

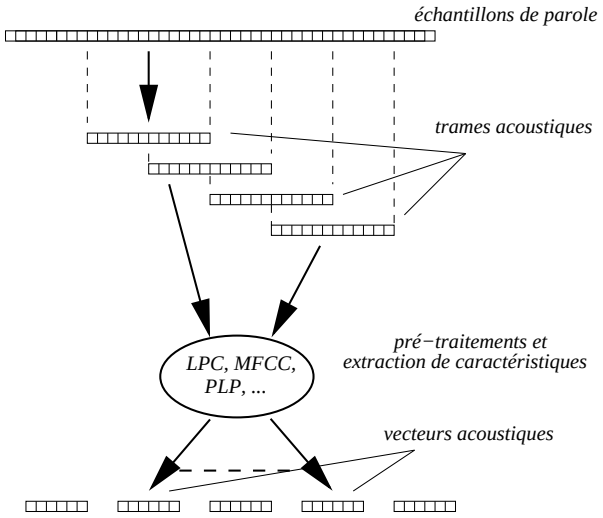


FIG. 2.2 – Principe de la segmentation en trames acoustiques

de disposer d'un classifieur. Nous avons effectué deux études de classification : l'une plus quantitative, portait sur l'estimation des scores en reconnaissance par une méthode de type k-ppv. L'autre, plus qualitative portait sur l'analyse statistique des paramètres (analyse discriminante). La figure 2.3 montre les scores obtenus sur une base de 6400 trames phonétiques (800 trames par classe). Il s'agit de résultats obtenus sur une base de test comportant 100 phonèmes par classes. De même que cela avait été noté en reconnaissance

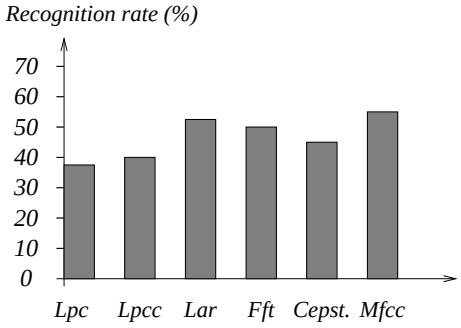


FIG. 2.3 – Scores obtenus sur une base de 6400 trames phonétiques pour 6 jeux de paramètres : LPC, LPCC, LAR, Cepstre, FFT, et MFCC (extrait de [66]).

du locuteur, on trouve des scores relativement différents selon l'analyse du signal effectuée. Par ailleurs les scores obtenus sont faibles. Une raison en est (hormis la simplicité du classifieur mis en oeuvre) la forte confusion entre les classes phonétiques dans l'espace des paramètres. Ces confusions sont mises en évidence par les analyses discriminantes effectuées sur les mêmes données. A titre d'exemple, la figure 2.4 représente une analyse discriminante en 2 dimensions effectuée sur les paramètres LPC et MFCC : Cette analyse reste imprécise mais corrobore tout de même les résultats du classifieur puisque les confu-

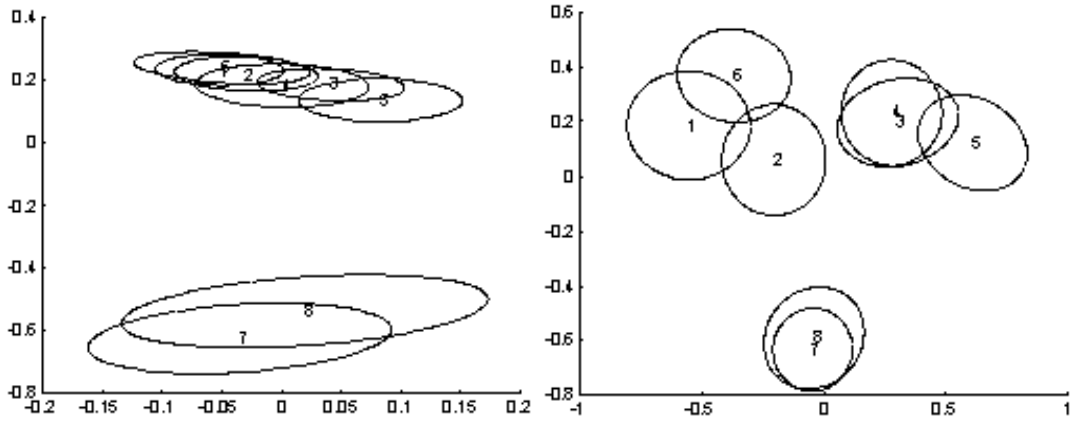


FIG. 2.4 – Analyses discriminantes effectuée sur les codes LPC à gauche et MFCC à droite (extrait de [66], voir aussi [149] et [69]).

sions inter-classes des paramètres MFCC sont moins importantes (on peut en particulier distinguer 3 sous-groupes) que celles des paramètres LPC.

2.2.2 Des problèmes posés

Cette première étude appelle un certain nombre de commentaires concernant : **1)** La faiblesse des scores obtenus, **2)** La comparaison des résultats avec ceux de la communauté, **3)** Les outils d'évaluation utilisés.

La faiblesse des scores obtenus

Les scores obtenus par Jean-Charles Didiot sont plus faibles que ceux obtenus par la communauté (cf. §2.1.1). La simplicité du classifieur utilisé n'explique pas tout. La classification mise en oeuvre ici s'effectue au niveau de la trame, donc sur des segments de phonèmes ou *quasi-phonèmes* plutôt que sur le phonème en entier. Il est intéressant ici de citer les travaux d'Hermansky et de son équipe à ce sujet [76], [89], [90]. Les auteurs montrent en effet que la trame ne suffit pas à caractériser complètement le phonème. Par ailleurs, les parties périphériques (début et fin) d'un phonème introduisent une variabilité très importante due aux phénomènes de co-articulation (inertie naturelle de l'organe vocal) : l'enveloppe spectrale d'un phonème dépend dans de nombreux cas du phonème qui le précède. Il en résulte du point de vue statistique une variance intra-classes pouvant atteindre des proportions du même ordre de grandeur que la variance inter-classes.

La comparaison des résultats avec ceux de la communauté

Il est souvent difficile d'établir des comparaisons exactes avec les résultats obtenus par d'autres laboratoires. Ainsi que je le décris plus haut, les bases de données diffèrent les unes des autres. Les conditions expérimentales également. Par exemple, les auteurs, lorsqu'ils souhaitent s'affranchir des phénomènes de co-articulation, ne considèrent souvent que les trames des parties centrales des phonèmes. Nous n'avons pas suivi cette démarche

puisque nous travaillons sur toutes les trames sans distinction. Ce dernier point contribue notoirement à la faiblesse des scores.
Nous participons à la campagne ESTER depuis 2004 dans le but de valider nos méthodes de codage et de pouvoir établir des comparaisons dans des conditions strictement définies (cf. §4.3.3).

Les outils d'évaluation utilisés

La validation des méthodes d'extraction de caractéristiques nécessite l'emploi de classifieurs élaborés dont une petite équipe ne dispose pas obligatoirement l'expertise (Classifieurs HMM, GMM, etc.). Une solution à ce problème est la collaboration avec d'autres équipes disposant des outils en question. On ne peut cependant proposer d'insérer de nouveaux algorithmes dans des systèmes existants (tâche qui demande un travail non négligeable, en particulier le réapprentissage de l'ensemble des paramètres adaptables du système), sans apporter également des éléments objectifs prouvant la supériorité des algorithmes étudiés. C'est ce que je me suis efforcé de faire en proposant continuellement au cours des stages et thèses encadrées d'évaluer les modèles concurrents, dans des conditions expérimentales rigoureusement identiques, à l'aide de classifieurs simples à mettre en œuvre.

2.2.3 La reconnaissance de phonèmes aujourd'hui

Les remarques faites au paragraphe précédent ne remettent pas en cause la thématique de recherche visant à améliorer l'extraction de caractéristiques au niveau de la trame. Les systèmes actuels de RAP fonctionnent toujours sur ce principe de la modélisation court-terme du spectre [18]. En revanche, la décision n'a plus lieu au niveau de la trame mais à des niveaux supérieurs de traitement. Le contexte est alors pris en compte, permettant d'améliorer significativement les scores. Notre but n'a pas été de construire un système de reconnaissance complet mais d'utiliser des classifieurs simples pour permettre de comparer différentes méthodes d'extraction de caractéristiques. Cependant, la notion d'appartenance des signaux à des catégories prendra de plus en plus d'importance dans notre réflexion. A titre d'exemple, le dernier modèle que je décris (§4.2, p. 76) est un extracteur de caractéristiques utilisable également comme classifieur phonétique. Ainsi, les deux étapes d'extraction et de classification sont devenues une seule et même étape.

Le contexte historique

Les motivations qui m'ont conduit à travailler sur le calcul des paramètres sont donc la disparité des résultats obtenus en reconnaissance du locuteur selon que l'on utilise tel ou tel jeu de paramètres. Ce constat a déjà été fait, notamment par Leung et al. dans [106] lorsqu'ils ont testé une dizaine de représentations acoustiques sur des phonèmes TIMIT et NTIMIT à l'aide de plusieurs types de classifieurs. Pour chaque classifieur, les performances obtenues dépendaient fortement du type de codage utilisé. Une autre motivation, plus générale, est le constat des faibles progrès effectués par les systèmes de reconnaissance automatique dans le courant des années 90. Les systèmes en question sont capables de dépasser les performances de l'oreille humaine dans des cas très précis. L'identification de locuteurs parmi un grand nombre, ou bien sur des segments très courts [5], en sont des exemples. En revanche, les conditions d'acquisition sont très restrictives (elles doivent être

identiques entre l'apprentissage et l'exploitation et toutes les sources de bruit doivent être modélisées, sinon évitées). Or ces paramètres n'affectent au contraire que très peu l'oreille humaine. Ainsi que le suggèrent des auteurs influents comme Bourlard [20] et Hermansky [75, 76], ne fallait-il pas revisiter certaines des étapes de traitement, pourtant considérées comme quasi-définitivement acquises ?
A l'heure actuelle, les paramètres les plus utilisés restent les paramètres cepstraux MFCC, en raison de leur robustesse notamment. Un important travail consistant à exploiter les propriétés auditives de l'oreille humaine a cependant été réalisé par Hermansky lorsqu'il a proposé les coefficients PLP [74], [77]. La stagnation des systèmes de reconnaissance automatique dans les années 90 a cependant relancé l'étude des paramètres, mais pas tant du point de vue du signal que du point de vue de la reconnaissance des formes. Il s'est agit en effet de renforcer le caractère robuste et discriminant des paramètres. Ce mouvement s'est traduit par l'augmentation de la dimension du vecteur caractéristique en adjoignant les coefficients dérivés Δ et deux fois dérivés $\Delta\Delta$ (paramètres dynamiques). Confronté au problème des grandes dimensions (lorsque l'on augmente la dimension des données pour un même ensemble d'apprentissage, on améliore les scores en apprentissage mais les performances en généralisation se dégradent) l'emploi de méthodes de réduction de la dimension s'est ensuite avéré nécessaire (analyse discriminante linéaire, LDA) [53].

Des voies de recherche

Un axe de recherche emprunté par certains chercheurs de la communauté a été la prise en compte des phénomènes non linéaires pouvant survenir aux différents niveaux de traitement. Le premier de ces niveaux aura été bien entendu celui de la classification (utilisation de classifieurs PMC [118], [5] et TDNN [138] par exemple) mais également celui de la réduction de dimension et du rehaussement des caractéristiques par des méthodes d'analyse discriminante non linéaires [126]. En revanche, à des niveaux plus proches du signal, on trouve peu de travaux de référence. Les paramètres LPC, dont Kohda et Nakano avaient montré les faiblesses lorsqu'on les utilise pour des tâches de classification [97], [122], mais pour lesquels Itakura avait cependant trouvé un intérêt en utilisant la distance qui porte son nom [83], trouvent un nouvel échos ici. Les réseaux de neurones non bouclés peuvent en effet être considérés comme une extension non linéaire des filtres AR [104], [52]. Dès lors, une extraction de paramètres *non linéaire* peut-être envisagée. C'est sur ce terrain que je me suis placé durant mes dix dernières années de recherche en proposant les paramètres NPC que je présente plus loin dans ce chapitre (§2.4).
Enfin, l'amélioration du caractère discriminant des paramètres conduit naturellement à penser l'étape d'extraction de caractéristiques en fonction du problème de classification posé. J'emprunterai cette voie en travaillant sur des modèles faisant intervenir progressivement la coopération du classifieur avec l'extracteur. Cette progression trouve un cadre formel dans les travaux de Katagiri [91] concernant les méthodes MCE dont je parle au chapitre 3 de ce mémoire.

2.3 Linéarité versus non linéarité

La question de la non linéarité est centrale puisqu'à l'origine du modèle NPC présenté dans ce mémoire. L'un des objectifs fixés était de savoir s'il est possible d'améliorer les scores

en reconnaissance de phonèmes par le simple fait de prendre en compte les non linéarités présentes dans le signal de parole. Mais cela pose plusieurs questions parmi lesquelles 1) qu'entend t-on par non linéarité dans le signal de parole ? 2) existent-elles et si oui dans quelles proportion ? 3) recèlent-elles un caractère discriminant relativement à l'application envisagée ?

2.3.1 Questions sur la non linéarité des systèmes et des signaux

Stricto sensu, un système f est linéaire si le principe de superposition et la propriété d'échelle s'appliquent, à savoir $f(ax+by) = af(x)+bf(y)$. Il est de plus invariant s'il produit les mêmes sorties pour des entrées identiques à des instants différents. Si un système viole la propriété de superposition, il est généralement considéré comme non linéaire. Par ailleurs, un signal x est dit *linéaire* s'il est généré par un système linéaire invariant, le système le plus fréquemment invoqué étant le filtre linéaire attaqué par un bruit blanc gaussien. On accepte souvent une définition plus souple tolérant une distribution non gaussienne des valeurs du signal. Tout signal ne pouvant donc être généré de cette manière est considéré comme non linéaire.
Pour ce qui concerne le signal de parole, ce dernier a de multiples raisons d'être non linéaire : le système de production est en effet non linéaire à différents niveaux. Le canal de transmission peut l'être également, le système d'acquisition et de traitement nécessaire à sa mesure aussi. En considérant que les éventuelles non linéarités du canal de transmission et du système de mesure sont négligeables, il reste à étudier le système de production. Ceci peut être fait de deux façons : par une étude directe du système s'il est connu [2] ou par une étude du signal si le système n'est pas connu. Dans ce dernier cas, on fait appel à des méthodes de détection de non linéarité dont la dernière utilisée est la méthode des *données de remplacement* ou *surrogate data* proposée par Theiler *et al* [134]. Par ces deux voies, on montre que le signal de parole est, par certains aspects, non linéaire.

2.3.2 Production non linéaire de la parole

Un état de l'art approfondi concernant l'éventuelle non linéarité du système de production de la parole, mais aussi la détection de non linéarités à partir du signal de parole, a été proposée par Gernot Kubin en 1995 [101]. Une synthèse par Faundez *et al.* [55] existe également et est à l'origine de l'action COST277 (cf. §4.3.6, p. 87). Ces études confirment l'hypothèse d'un système de production non linéaire, notamment en ce qui concerne la production des sons voisés.
Loin du modèle source filtre souvent utilisé, l'organe de production vocal est un système complexe dans lequel les phénomènes non linéaires tiennent une place essentielle [2]. Par exemple, avec la vibration des cordes vocales, la forme d'onde change avec l'amplitude du signal [133] [129] [130]. Des phénomènes de bifurcation existent au point que l'on caractérise les cordes vocales par leur diagramme de bifurcations [109]. Des *sous harmoniques* sont présentes, entraînant des phénomènes de quasi-périodicité car les deux cordes vocales ne vibrent pas exactement à la même fréquence. Des phénomènes de turbulence sont également à la source de bruits dans la génération de certains sons [113] (les fricatives par exemple). Ils surviennent dès qu'il y a une forte interaction entre la source d'air et les différents obstacles présents dans le conduit vocal (palais, lèvres, etc.). Ces phénomènes, typiquement non linéaires, produisent des formes d'ondes fortement irrégulières.

2.3.3 Sur l'évidence de la présence de non linéarités dans le signal de parole

L'étude établie par Kubin sur la détection de non linéarités à partir du signal de parole montre clairement que les sons voisés sont non linéaires, contrairement aux sons non voisés. Ainsi, la modélisation source/filtre où la source est un bruit blanc gaussien et le filtre un système linéaire invariant est bien adaptée à la modélisation des parties non voisées du signal de parole. En revanche elle ne l'est plus en ce qui concerne les parties voisées. Ceci entre en contradiction avec la réalité des systèmes de compression qui utilisent avec succes ce modèle pour les sons voisés (codeurs CELP). Gernot Kubin [101] explique cette contradiction en affirmant qu'elle n'est qu'apparente : le système mis en oeuvre est en fait un filtre *adaptatif* excité par un signal variable (sélection des signaux d'excitation par *codebook* par exemple), le tout formant donc un système fortement non linéaire.
D'autres études vont dans le même sens, à savoir que le signal de parole qui est constitué d'environ 40% de sons non voisés, est linéaire dans les parties non voisées et non linéaire dans les parties voisées. Thyssen et al. [135] ont expérimentalement montré la présence de non linéarités dans le signal de parole. Dans leur approche, les auteurs considèrent une trame de signal de parole de $25ms$ sur laquelle ils estiment les paramètres d'un filtre tous pôles à l'aide d'une analyse LPC. L'erreur de prédiction permet d'estimer le gain de prédiction qui est le rapport de la variance du signal sur la variance de l'erreur. On observe par ailleurs que cette erreur n'est pas totalement blanche, signe que toutes les informations ne sont pas pour autant extraites ou que le modèle sous-jacent n'est pas linéaire. Bien entendu, une étude sur l'ordre du filtre doit être menée [70] afin d'estimer l'ordre optimal. Pour s'assurer qu'il ne reste plus aucune information de type linéaire dans le signal d'erreur, Thyssen a effectué plusieurs analyses LPC en cascades. Il a ainsi montré que pour un signal voisé, le gain de prédiction du 5ème filtre LPC arrive à $0dB$. Il estime ensuite les paramètres d'un filtre non linéaire (typiquement un réseau de neurone prédicteur ou un filtre de Volterra d'ordre de prédiction identique) sur la dernière erreur de prédiction. Le gain de prédiction résultant confirme la présence de non linéarités, au sens d'informations "non captées" par les filtres linéaires.
Cet argument donné par Thyssen a été repris à plusieurs reprises dans la littérature puisqu'on le retrouve cité dans de nombreuses publications [101], [55], [14], [51], [111], [110] mais d'autres auteurs comme Townshend [137] avaient déjà montré quelques années auparavant, et à l'aide de méthodes similaires, qu'un prédicteur non linéaire peut extraire des informations complémentaires (de l'ordre de 2,8dB à 3dB de plus sur les mesures effectuées), y compris à partir de séries chaotiques [95]. D'autres prédicteurs non linéaires ont également été proposés comme les réseaux à fonctions radiales de base (RBF) [51], [14] pour lesquels Weigend [141] a mis en évidence leur supériorité dans le cas des entrées de dimension élevée ou encore les réseaux de neurones récurrents [82]. Enfin une autre approche consiste à estimer des approximations localement linéaires du prédicteur non linéaire [131].
Gernot Kubin fait remarquer que ces mesures sont effectuées sur des systèmes de prédiction à court-terme, c'est à dire sur des intervalles de temps d'environ 30ms. L'hypothèse de stationnarité du signal de parole est donc respectée et les sources de non linéarité mesurées ne sont pas en rapport avec l'évolution fortement non linéaire qui intervient sur des échelles de temps plus grandes et où l'hypothèse de stationnarité n'est plus respectée. Des systèmes neuronaux prédictifs ont été développés pour prendre en compte l'évolution long terme du

signal de parole [136], [96]. C'est alors l'évolution dynamique des vecteurs caractéristiques qui est modélisée comme dans [4], [119], [7], [5]. Les réseaux peuvent être également du type RBF [22] ou encore à poids dynamiques fonction du temps [23]. Levin [107], [108] a également proposé un modèle de réseau perceptron multicouches (PMC) à commande cachée (Hidden Control Neural Network, HCNN) pouvant être élégamment couplé avec des HMMs ou étendu à l'estimation de l'erreur quadratique de prédiction [124].

Il ressort de ces études que des phénomènes non linéaires ont lieu dans le mécanisme de production de la parole et qu'ils sont, au moins en partie, décelables au travers de l'analyse du signal de la parole. Les sons voisés et d'énergie élevée sont plus particulièrement concernés.

2.4 Les paramètres NPC : un modèle intuitif

Encadrement : Philippe Sellem (DEA de Robotique 1997)
Publications : [149]

J'ai proposé le modèle NPC (Neural Predictive Coding) en 1996 et publié les premiers résultats le concernant en 1997 [69], [149]. Il est le résultat de près de deux ans de réflexions menées sur la problématique de la modélisation du signal de parole. J'en expose le principe dans le paragraphe suivant (§2.4.1) ainsi que les premières expérimentations menées par Philippe Sellem durant son stage de DEA (§2.4.2). L'année suivante, en 1998, Cyril Chavy entreprendra dans le cadre de sa thèse une étude plus approfondie que je résumerai au paragraphe 2.5, p. 33.

2.4.1 Le principe

Le modèle NPC, tout comme les paramètres LPC, est basé sur la représentation source-conduit de l'organe vocal à ceci près que le filtre est remplacé par un réseau de neurones. Les travaux de Lapedes et Farber [104] concernant la modélisation non linéaire de signaux ont été une inspiration importante dans la proposition du nouveau modèle NPC. Les auteurs montrent en effet que l'on peut considérer les réseaux de type PMC comme une extension naturelle au domaine non linéaire des méthodes linéaires de traitement adaptatif du signal. En France, Gérard Dreyfus et son équipe ont proposé une synthèse sous le même angle du problème de la prédiction non linéaire [52].
Pour une trame de signal donnée, la méthode d'extraction des paramètres LPC suit l'hypothèse selon laquelle le signal a été émis par un modèle source/filtre selon :

$$y_k = \sum_{i=1}^{\lambda} a_i y_{k-i} + \varepsilon_k \tag{2.1}$$

où ε_k désigne la source (un bruit blanc centré) et les a_i les coefficients du filtre. Cette modélisation convient tout particulièrement aux signaux de parole non voisés si l'on considère que le filtre représente la forme du conduit vocal. Elle est étendue au cas des signaux voisés mais les limitations de ce modèle sont évidentes et ont déjà été largement décrites dans la littérature [18]. En premier lieu, la source n'est plus un bruit blanc mais un signal quasi-périodique. Il s'agit de l'onde glottale, le plus souvent représentée par un peigne

de Dirac, et non modélisable par un filtre court-terme. L'onde glottale est explicitement éliminée dans les systèmes de RAP (filtrage cepstral avec les paramètres MFCC ou filtrage court-terme avec les paramètres LPC), mais elle recèle des informations propres au locuteur exploitables dans les systèmes de reconnaissance automatique du locuteur (RAL) [125]. Ensuite, la transmittance est *tous-pôles*, ce qui n'est pas le cas dans la réalité puisque au moins deux conduits fonctionnent en parallèle (le conduit oral et les fosses nasales). Il devrait donc y avoir des zéros dans la fonction de transfert du filtre qui n'y sont pas pour des raisons évidentes de simplification de l'estimation des coefficients. On préfère en l'occurrence surestimer le degré du dénominateur pour pouvoir assimiler le numérateur à une constante. Par ailleurs ce modèle suppose que les sons appartiennent à deux catégories distinctes *voisés* ou *non voisés*. Or il existe des sons comme les sons *fricatifs voisés* qui sont les deux à la fois. Enfin, cette modélisation néglige les interactions qui existent entre la source et le conduit vocal.

Ce qui nous intéresse ici, c'est l'extension de ce modèle proposée par Lapedes et Farber : le filtre linéaire est remplacé par un réseau PMC à une couche cachée et une cellule de sortie. Cela permet une représentation plus fidèle des sons voisés, plus fréquents dans les langues latines et anglo-saxonnes. Nous faisons donc l'hypothèse que les signaux de parole satisfont :

$$y_k = \sigma(\sum_j a_j \sigma[\sum_{i=1}^{\lambda} \omega_{ij} y_{k-i}]) + \varepsilon_k \tag{2.2}$$

où σ désigne la fonction sigmoïde et les a_j et ω_{ij} les paramètres ajustables du modèle. Le problème de la caractérisation du signal devient celui de l'estimation des paramètres ω_{ij} et a_j que l'on identifie aux poids d'un réseau de neurones (respectivement les poids de la couche cachée et de la cellule de sortie). Cette estimation est conduite par minimisation de l'erreur quadratique *de prédiction* sur la trame considérée :

$$Q(\Omega, \mathbf{a}) = \sum_{k=\lambda}^{D-1} \|\hat{y}_k - y_k\|^2 \tag{2.3}$$

dans laquelle Ω représente la matrice des poids de la couche cachée et $\mathbf{a}$ le vecteur des poids de la cellule de sortie. $\hat{y}_k$ désigne l'estimation de y_k par le réseau, notée $\hat{y}_k = F_{\Omega,\mathbf{a}}(y_k)$ où $F_{\Omega,\mathbf{a}}$ est la fonction réalisée par le réseau. λ désigne l'horizon de prédiction et D le nombre d'échantillons d'une trame.
Comme le souligne Thyssen dans [135], l'inconvénient majeur de cette approche est le grand nombre de paramètres libres du modèle. Par exemple, pour un horizon de prédiction égal à 12 et 8 cellules sur la couche cachée, on arrive à un total de $(12+1)\times 8 + (8+1)\times 1 = 113$ poids et seuils, soit un vecteur caractéristique de 113 coefficients auxquels il convient d'ajouter les coefficients dynamiques Δ et $\Delta\Delta$, conduisant à 339 coefficients.

L'idée centrale du modèle NPC est de ne considérer comme coefficients caractéristiques que les poids de la couche de sortie. Dans l'exemple précédent, cela revient à construire un vecteur de caractéristiques à 8 composantes. La dimension de l'horizon de prédiction devient alors indépendante de la dimension du vecteur de caractéristiques puisque seul le nombre de cellules cachées fixe la dimension de ce dernier. Cette formulation répond à

l'idée selon laquelle seuls les poids de la couche de sortie sont représentatifs du signal traité et donc du modèle sous-jacent dont on attend une représentation.
Si au départ, aucun argument théorique ne permettait d'accréditer une telle hypothèse, les premières expérimentations que je présente dans les paragraphes suivants ont montré qu'elle n'était pas infondée. Ce n'est que plus tard que je remarquerai que l'idée NPC constitue en fait une parfaite illustration du théorème de superposition de Kolmogorov [63] (cf. chapitre 4, §4.1.3, p. 75).

2.4.2 Les paramètres NPC appliqués à la reconnaissance de phonèmes

Avec le stage de DEA de Philippe Sellem commence un travail très captivant : celui de soumettre à l'épreuve de l'expérimentation un nouveau modèle dont rien ne nous indique, à première vue, qu'il fonctionnera correctement. En l'occurrence, les paramètres extraits seront-ils suffisamment *discriminants* pour permettre d'obtenir ensuite des scores en classification au moins comparables aux méthodes concurrentes ?
A la suite d'un travail remarquable, Philippe Sellem nous livre des résultats très encourageants. Le modèle, qui s'appelle encore le NLPC (Non Linear Predictive Coding), permet d'obtenir des scores honorables de l'ordre de 60 à 70 % de reconnaissance. Je reviendrai sur ce point plus bas. Deux modes de fonctionnement sont alors envisagés :
– le codage *en bloc* d'un ensemble de trames
– le codage *à la volée* ou *en ligne* d'une succession quelconque de trames.
Le codage *en bloc* ne sera pas retenu pour des raisons évidentes d'inadéquation avec des applications temps réel. Il s'agit comme le montre la figure 2.5 à gauche de regrouper l'ensemble des trames phonétiques que l'on souhaite coder (des trames extraites sur plusieurs minutes de signal par exemple) en une unique base et d'entraîner l'ensemble des poids de l'extracteur sur cette base.

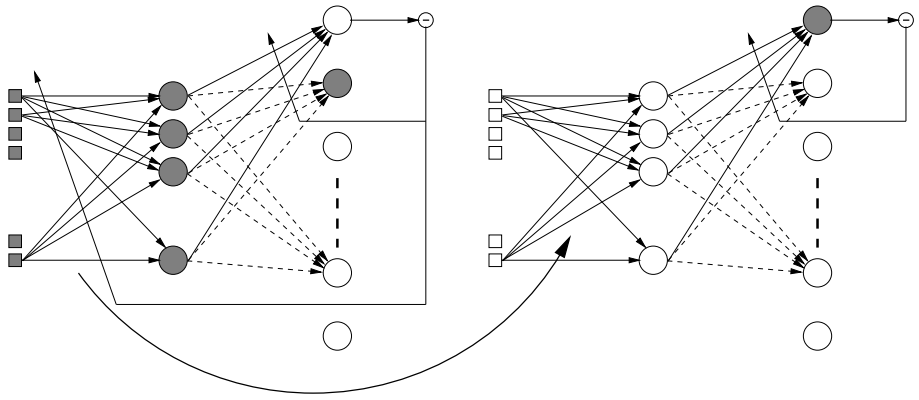


FIG. 2.5 – Schéma simplifié des deux modèles NLPC "en bloc" à gauche et "à la volée" à droite

A chaque trame correspond une cellule de la couche de sortie. On dispose donc d'autant de cellules de sortie qu'il y a de trames à coder. A l'issue de l'apprentissage, les poids des différentes cellules de sortie constituent les paramètres caractéristiques de la trame correspondante. A l'aide d'un classifieur rudimentaire de type PMC à une couche cachée

et 5000 itérations d'apprentissage nous avons obtenu sur une base de 8 phonèmes /*aa*/, /*ae*/, /*ey*/, /*ix*/, /*iy*/, /*ow*/, /*s*/ et /*z*/ les scores de 73% de bonne reconnaissance en apprentissage et 64% en généralisation. La matrice de confusion (table 2.1) obtenue sur les données de test donne une première idée des scores obtenus par classe. Si l'on compare ces

Trames reconnues / Trames présentées	/*a*/	/*ae*/	/*ey*/	/*ix*/	/*iy*/	/*ow*/	/*s*/	/*z*/
/*a*/	89	1	0	0	7	3	0	0
/*ae*/	23	63	3	2	0	8	0	0
/*ey*/	0	5	50	22	17	5	0	1
/*ix*/	9	5	25	40	18	2	0	0
/*iy*/	0	0	23	19	57	0	0	1
/*ow*/	16	23	12	9	4	35	0	0
/*s*/	0	0	0	1	0	0	57	42
/*z*/	0	0	3	3	1	0	41	52

TAB. 2.1 – Matrice de confusion obtenue en test après codage NLPC "en bloc" de trames phonétiques (extrait du rapport de stage de P. Sellem).

premiers résultats avec les scores obtenus sur les mêmes données avec les paramètres LPC et MFCC (il est vrai avec une méthode de classification différente puisqu'il s'agissait des k-ppv) on obtient des résultats comparables. Le fait remarquable ici est que limiter les codes aux seuls jeux de poids des couches de sortie (il aurait fallu en toute rigueur considérer les poids de la première couche également) ne réduit pas les propriétés discriminantes des paramètres NLPC générés. En revanche, comme pour tout problème d'apprentissage, le calcul des poids-codes nécessite de disposer d'une base conséquente de trames et demande un temps de calcul qui rend l'exploitation du modèle délicate.

La deuxième version du NLPC apporte une solution au problème de la complexité de l'estimation des paramètres. Ainsi que l'indique la figure 2.5 à droite, seuls les poids des deuxièmes couches sont estimés lorsque l'on présente successivement les trames en entrée. Le calcul revient ainsi à estimer les poids d'un réseau sans couche cachée ce qui présente les avantages suivants :

1) L'estimation des paramètres NLPC pour une trame de parole donnée est indépendante de celle des trames voisines, ce qui autorise le codage *à la volée* des données ;

2) Les temps de calcul sont considérablement réduits car les poids de la première couche ne sont plus estimés ;

3) Dans l'hypothèse où les fonctions de transition des cellules de sortie sont linéaires, l'extraction des codes devient un problème de régression linéaire.

La question qui se pose est de savoir quelles valeurs donner aux poids de la première couche puisqu'ils ne sont pas estimés lors de l'extraction. La poursuite des simulations a montré que les performances n'étaient que faiblement entamées si l'on se contentait d'utiliser la

méthode NLPC *en bloc* pour estimer ces poids puis la méthode *à la volée* pour estimer les codes, y compris lorsque les données d'apprentissage ne sont pas identiques. Le procédé devient donc :

- 1) *Estimation des poids de la première couche :* on constitue une base de trames caractéristique des signaux de parole à traiter et l'on effectue sur cette base un apprentissage en bloc de l'ensemble des poids du réseau.

- 2) *Extraction des paramètres NLPC :* on récupère les poids de la première couche initialement calculés (une fois pour toute) et l'on estime pour chaque trame de signal à paramétrer, les poids d'une deuxième couche initialisée aléatoirement.

Dans les mêmes conditions que précédemment, nous avons obtenu un score moyen de 62,5% de reconnaissance en généralisation et la matrice de confusion présentée table 2.2. Une brève comparaison des deux tables de confusion montre clairement une augmentation

Trames reconnues / Trames présentées	/a/	/ae/	/ey/	/ix/	/iy/	/ow/	/s/	/z/
/a/	71	7	3	2	3	13	0	0
/ae/	8	45	16	8	2	22	0	0
/ey/	0	18	33	22	13	14	0	1
/ix/	3	8	17	22	23	21	0	6
/iy/	0	0	10	14	67	5	0	5
/ow/	27	18	12	12	4	28	0	0
/s/	0	0	1	0	0	0	70	28
/z/	0	0	3	0	2	0	45	49

TAB. 2.2 – Matrice de confusion obtenue en généralisation après codage NLPC *à la volée* de trames phonétiques (extrait du rapport de stage de P. Sellem).

des confusions avec la deuxième méthode de codage *à la volée*. On pouvait s'attendre à cette diminution des performances, notamment parce que les signaux codés n'ont pas été utilisés pour l'apprentissage des poids de la première couche. Néanmoins, la dégradation est faible et ce score reste comparable à ceux obtenus par d'autres techniques. Enfin, et surtout, on tient là une méthode d'extraction dont la complexité est compatible avec des opérations d'extraction de caractéristiques en temps réel du flux de parole.

2.4.3 Des questions posées

Dès la fin du stage de Philippe Sellem, je propose un sujet de thèse portant sur la mise en œuvre de l'extracteur NPC. Il s'agit d'explorer en détail la voie ouverte et de répondre à tout un ensemble de questions que j'énonce ci-dessous. Une majorité des réponses seront trouvées dans le cadre des travaux de thèse de Cyril Chavy exposés dans ce chapitre puis de Mohamed Chetouani exposés dans le chapitre suivant (chapitre 3). Certaines sont d'ordre pratique et concernent la mise en oeuvre de l'extracteur, d'autres sont d'ordre plus théoriques et concernent essentiellement la justification et la validation de la méthode

d'extraction.

Complexité de mise en œuvre

Un défaut des paramètres NPC est la nécessité d'utiliser une méthode d'apprentissage de type descente de gradient pour l'estimation en bloc des poids de la première couche (phase que nous avons appelée *phase d'adaptation* de l'extracteur). Des questions propres à ce genre d'approche se posent telles que la présence de minima locaux ou l'apparition de problèmes de sur-apprentissage. Elles ont été traitées par Cyril Chavy (§2.5.7, p. 39). En comparaison, les systèmes concurrents sont de faible complexité et simples à mettre en oeuvre. En particulier, ils ne sont pas adaptatifs.

Robustesse

Les conditions d'acquisition du signal audio au sens le plus large (qualité du micro, éloignement du ou des locuteur(s), bruits de l'environnement, acoustique des lieux), les variabilités inter et intra-locuteur, le rythme de l'élocution, le vocabulaire, sont autant de défis auxquels doivent faire face les systèmes de reconnaissance automatique. On accorde donc une grande importance à la robustesse des paramètres acoustiques générés et cela constitue un sujet d'étude à part entière. Les signaux de parole utilisés pour tester les paramètres NPC ont été principalement extraits de la base de parole NTIMIT de signaux téléphonés. Il s'agit donc de signaux bruités et dont la bande passante est considérablement réduite.

Mise en oeuvre

Les scores obtenus pour un nombre limité de classes phonétiques se retrouvent-ils sur l'ensemble des phonèmes d'une langue donnée dans le cadre d'une application de RAP ? On constate que la question se pause au niveau de l'étage de classification. Pour un système de classification discriminant comme le perceptron multicouches, les scores en classification de phonèmes baissent lorsque le nombre de classes augmente. Une réponse à ce problème est la conception d'un extracteur de caractéristiques modulaire procédant par sous-groupes phonétiques. Cette solution a été expérimentée par Mohamed Chetouani (chapitre 3, §3.4.4, p. 60).

Coopération paramétrisation/classification

Un reproche habituellement fait aux paramètres linéaires LPC, et dont héritent naturellement les paramètres NPC, est leur caractère insuffisamment discriminant qui les rend moins bien adaptés aux applications de reconnaissance. La nécessité de prendre en compte dès l'extraction de caractéristiques la problématique de la classification nous est donc très vite apparue. Faut-il concevoir un extracteur simple ou un ensemble extracteur-classifieur plus facilement adaptable à un système de RAP ? Dans cette dernière hypothèse doit-on envisager une interaction réciproque entre l'extracteur et le classifieur ? Trouve t-on là un terrain pour la problématique modélisation/discrimination ? En réponse, les paramètres NPC sont les premiers d'une liste faisant intervenir tour à tour différentes méthodes de coopération paramétrisation/classification que je présente au chapitre suivant.

Validation expérimentale

Comme je l'ai souligné plus haut dans ce mémoire (chapitre 2, §2.1.1, p. 18), une validation expérimentale complète des paramètres NPC nécessite leur insertion dans un dispositif complet de RAP. Ceci peut être réalisé à l'occasion de participations à des campagnes d'évaluation et de coopération avec d'autres équipes (cf. Chapitre 4, §4.3.3, p. 83).

Validation théorique

Peut-on expliquer du point de vue méthodologique la raison pour laquelle les poids de la couche de sortie portent suffisamment d'informations sur le signal pour envisager leur utilisation comme vecteur acoustique ? S'agit-il d'une propriété universelle des réseaux PMC ? Je donne des éléments de réponse au chapitre 4 (§4.1, p. 73).

2.5 Première validation expérimentale

Encadrement : Mathieu Györffy (DEA de Robotique 1999)
Publications : [26], [27], [28], **[66]**, [25]

Un travail important de Cyril Chavy a été de mener les premières validations expérimentales du modèle NPC. Il a pour cela étudié l'influence de différents paramètres, notamment la taille de la fenêtre de prédiction (horizon de prédiction de l'extracteur). Il a également mis en place un protocole d'expérimentation et effectué un certain nombre de mesures comparatives de scores des paramètres NPC avec d'autres paramètres de la littérature comme les paramètres LPC et MFCC. J'aborde également dans cette section l'étude des gains de prédiction obtenus par le filtrage non linéaire NPC.

2.5.1 Les signaux et la constitution des bases de données.

La présentation des scores nécessite de préciser avant tout les conditions dans lesquelles ils ont été obtenus. Les conditions expérimentales sont celles décrites dans les différents articles. Elles sont restées sensiblement les mêmes tout au long des expérimentations et s'axent selon deux problématiques :

1. Le choix des signaux et la constitution des bases d'apprentissage ;
2. Le choix du classifieur.

Il est souvent difficile de comparer les travaux obtenus par différentes équipes sur un problème donné car les conditions expérimentales, et en particulier les signaux utilisés, diffèrent sensiblement de l'une à l'autre. Dans ce cadre, les signaux de parole proposés par le consortium NIST constituent une base de référence reconnue internationalement. Mais plus encore que les signaux eux-mêmes, la participation aux campagnes d'évaluation garantie l'harmonisation des résultats obtenus par les différentes équipes participantes et le bien fondé des comparaisons effectuées entre les différents systèmes.
Nous devions dans une première étape valider expérimentalement les modèles NPC et nous avons utilisé pour cela les signaux de parole issus de deux grandes bases de données NIST utilisées dans la majorité des laboratoires : la base TIMIT et sa version téléphonique NTIMIT [1].

La base TIMIT a été développée par le Massachusetts Institute of Technology (MIT), le Standford Research Institute (SRI) et Texas Instrument (TI). Il s'agit d'une base de parole continue multilocuteurs composée de 10 phrases prononcées par 630 locuteurs (192 hommes et 438 femmes) de 8 régions : les régions dialectales (DR) des Etats-Unis. Chacune de ces régions présente deux sous ensembles TRAIN et TEST dédiés respectivement à l'apprentissage des systèmes et à leur test (généralisation). Les enregistrements ont été effectués à l'aide de microphones de qualité et à une fréquence d'échantillonnage de 16kHz (16bits). La base NTIMIT a été obtenue par le passage de la base TIMIT à travers un réseau téléphonique (appels locaux et longue distance). La bande passante est donc réduite entre 330Hz et 3400Hz mais la fréquence d'échantillonnage reste toujours à 16kHz. Nous avons beaucoup utilisé la base NTIMIT car les performances des systèmes de reconnaissance sont très dégradés sur cette base [120],[143]. Elle constitue donc un outil intéressant d'évaluation des codeurs NPC et de leur robustesse.

Dans le cadre de ses travaux de thèse, Cyril Chavy a construit une base regroupant six classes de phonèmes extraits de la base NTIMIT et qu'il a ensuite utilisée pour l'ensemble de ses expérimentations. La table 2.3 en donne les caractéristiques principales. Les phonèmes

phonèmes :	/*s*/, /*z*/, /*ah*/, /*ih*/, /*aa*/, /*iy*/
base :	NTIMIT, région DR1 (New England)
nombre de trames :	6000
répartition / classes :	1000 par classes de phonèmes
ratio apprentissage / test :	50% apprentissage 50 % test
répartition / locuteur :	certains locuteurs présents dans les 2 bases
répartition / trames :	pas de trames d'un même phonème dans les 2 bases

TAB. 2.3 – Caractéristiques de la base de phonèmes utilisée pour la mise au point des codeurs NPC.

choisis sont parmi les plus fréquents et permettent d'évaluer les codages dans différents cas de figure :
– voisé / non voisé ({/*ah*/, /*aa*/, /*ih*/, /*iy*/} et {/*s*/, /*z*/})
– couples facilement séparables (/*s*/ et /*ah*/ par exemple)
– couples difficilement séparables (/*s*/ et /*z*/ ou /*ah*/ et /*aa*/ par exemple).

2.5.2 Principales hypothèses et contexte expérimental

L'évaluation d'un extracteur de caractéristiques en parole n'est pas une affaire aisée. Nous avons dû procéder à certaines simplifications ci-dessous dont la principale conséquence aura été de dégrader les scores obtenus. Bien entendu, l'ensemble des méthodes concurrentes ont été testées dans des conditions strictement analogues, subissant donc des dégradations équivalentes.

1) **Le mode d'évaluation :** La meilleur solution d'évaluation consiste en l'incorporation de l'extracteur dans un système complet de RAP et en l'analyse comparative des performances obtenues. Comme je l'ai expliqué au chapitre précédent (§2.1.1, p. 18), notre

équipe ne disposait pas d'un tel système ni même d'ailleurs du savoir faire ni du personnel nécessaires à sa réalisation. Une alternative consistait à envisager une collaboration avec un ou plusieurs laboratoires disposant d'un tel matériel. Pour les raisons que j'ai données plus haut (cf. §2.2.2, p. 22) cette solution ne pouvait être mise en oeuvre, du moins dans un premier temps. Nous avons donc opté pour l'évaluation des paramètres NPC sur des problèmes de reconnaissance de phonèmes et de reconnaissance de locuteurs. Il devient alors possible d'utiliser des systèmes de classification simples (typiquement des modèles connexionnistes de type perceptron multicouches ou des modèles statistiques de type GMM).

2) **La segmentation phonétique :** Dans les expériences menées, nous avons fait l'hypothèse que l'étape de segmentation phonétique était résolue. Mais cela suppose de disposer de bases d'apprentissage segmentées comme le sont les bases TIMIT et NTIMIT. Or ces segmentations comportent des erreurs qui reviennent à introduire un bruit supplémentaire dans les données.

3) **Le contexte :** Nous n'avons pas pris en compte le problème du contexte phonétique et nous nous sommes donc systématiquement placés en reconnaissance indépendante du contexte.

4) **La stationnarité :** Le phonème ne peut être considéré comme stationnaire. On admet le plus souvent qu'il se décompose en trois parties quasi-stationnaires, *début*, *milieu* et *fin*. Par soucis de simplification, nous avons calculé les scores à partir des décisions locales, au niveau de la trame, et non au niveau du phonème entier, sans tenir compte non plus de l'appartenance de ces trames au début, au milieu ou à la fin du phonème. Décider au niveau de la trame augmente la probabilité d'erreurs de classification mais permet toutefois de disposer d'un plus grand nombre d'exemples dans les bases d'apprentissage pour un temps global de parole identique.

5) **Les co-articulations :** Ne pas tirer parti ni du contexte phonétique, ni de la position des trames phonétiques en début, milieu ou fin de phonème conduit à ne pas considérer également les problèmes de co-articulations (les début et fin de phonèmes dépendent dans certains cas de la classe des phonèmes précédents et suivants). Afin de s'abstraire des problèmes de co-articulation, et donc de réduire la variance des vecteurs caractéristiques générés, certaines équipes calculent leurs scores en sélectionnant uniquement les trames centrales des phonèmes.

L'ensemble des paramètres testés dans les études que nous avons publiées (LPC, LPCC, FB, MFCC, PLP, NPC, etc.) l'ont été dans des conditions expérimentales rigoureusement identiques. Ceci permet de valider notre démarche mais témoigne de la difficulté qu'il y a à comparer les scores bruts émanents de différentes équipes.

2.5.3 Les classifieurs utilisés

La classification des vecteurs caractéristiques a été effectuée le plus souvent à l'aide d'un perceptron multicouches. Le modèle connexionniste utilisé était du type PMC (12-10-6),

comportant 12 entrées pour la classification de vecteurs de caractéristiques de dimension 12, une couche cachée comportant 10 cellules et 6 cellules de sortie (une par classe phonétique). Les algorithmes d'apprentissage utilisés ont été tour-à-tour l'algorithme de rétropropagation du gradient, sans puis avec pas adaptatif, avec critère d'arrêt par cross validation et enfin l'algorithme de Levenberg-Marquadt.
Nous avons également utilisé l'analyse discriminante, surtout dans les premiers temps de l'étude, ainsi qu'en témoignent le rapport de stage de DEA de Jean Charles Didiot, et l'article [66] en annexe de ce mémoire p. 109. La méthode directe des K-ppv a également été utilisée, et enfin, plus récemment avec Mohamed Chetouani, les GMM dans le cadre de la reconnaissance de locuteurs [31] et de la campagne d'évaluation ESTER [60].

Le choix du classifieur PMC a fait l'objet d'un débat lors de la soutenance de thèse de Mohamed Chetouani. En utilisant un classifieur de type neuronal, ne donnait-on pas un avantage certain à notre méthode de codage, également neuronale, relativement aux autres méthodes de codage, non neuronales ? Nous n'avons pas directement traité cette question, bien qu'importante dans le cadre d'une insertion de notre système dans une machine RAP. Ces systèmes fonctionnent d'autant mieux et plus rapidement que les vecteurs caractéristiques ont des distributions localement gaussiennes à covariances diagonales, et l'on sait que les vecteurs MFCC satisfont souvent le mieux cette contrainte. Il est certain que la distribution éventuellement non gaussienne des paramètres NPC peut favoriser notre méthode de classification si l'on considère que les méthodes neuronales accommodent mieux des distributions non gaussiennes que les méthodes GMM et HMM. Je reviens sur ce point dans les conclusions et perspectives §4.4, p. 88.
Cette question ne se posera plus pour les systèmes NPC postérieurs (NPC-2, DFE-NPC et suivants) dans la mesure où la coopération entre l'extraction de caractéristiques et la classification est explicitement envisagée.

2.5.4 Scores et comparaisons

La figure 2.6, extraite du mémoire de thèse de Cyril Chavy [25], donne un exemple d'apprentissage d'un réseau PMC à une couche cachée après calcul des paramètres NPC sur deux structures légèrement différentes du codeur. La différence concerne la fonction d'activation du neurone de sortie, linéaire dans un cas et sigmoïdale dans le deuxième. Ces résultats montrent que l'on peut s'affranchir de la fonction sigmoïde et ne conserver qu'une fonction linéaire, avec les avantages que cela procure du point de vue de l'estimation des paramètres. La table 2.4 reporte les premiers scores comparatifs obtenus et publiés dans [27]. Ces résultats constituent une deuxième validation expérimentale après celle obtenue par Philippe Sellem durant son stage de DEA [69], [149] puisque les scores obtenus par les paramètres NPC indiquent en moyenne deux points de plus que les codages les plus courament utilisés.

2.5.5 Gains de prédiction mesurés sur le prédicteur NPC

Dans le cadre du stage de DEA de Mathieu Györffy, nous avons effectué des mesures du gain de prédiction d'un codeur NPC après adaptation de la couche cachée sur une base comportant les 6 classes de phonèmes suivantes : /*s*/, /*z*/, /*ah*/, /*ih*/, /*aa*/ et /*iy*/, à

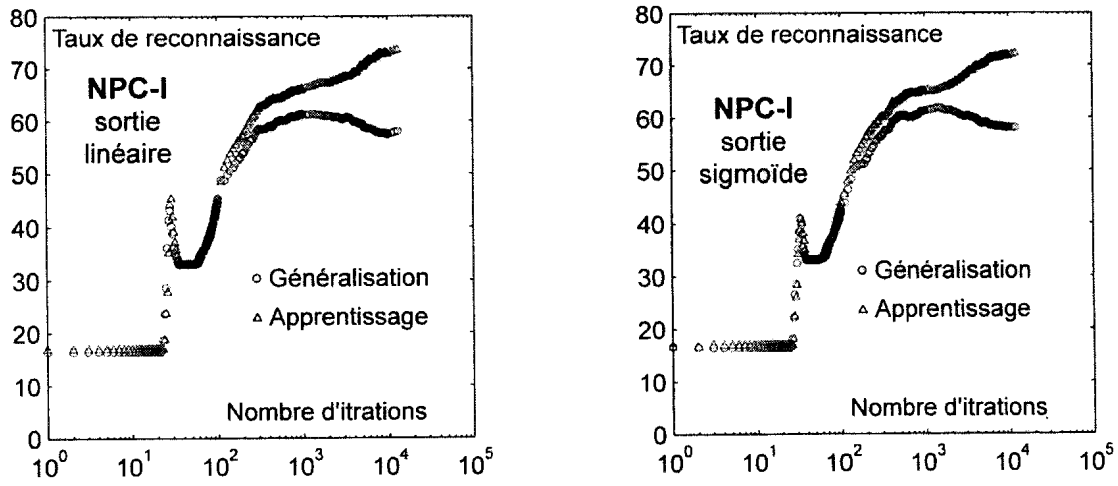


FIG. 2.6 – Scores en cours d'apprentissage avec fonction de transition liéaire à gauche et sigmoïde à droite (extrait de [25]).

paramètres	LPC	LPCC	LAR	MFCC	FB	NPC (linéaire)	NPC (sigmoïde)
scores moyens (%)	56.7	57.9	58.2	58.3	59.1	61.2	61.4

TAB. 2.4 – Taux de reconnaissance obtenus en généralisation pour différents paramètres (extrait de [27]).

raison de 1000 trames de 256 échantillons extraits de NTIMIT. La comparaison avec un filtre LPC à 12 coefficients montre que le codeur NPC a un gain de prédiction supérieur en moyenne sur l'ensemble des phonèmes testés, ainsi que l'indique la table 2.5. Ces gains

phonèmes	LPC(12)	NPC (12)	amélioration
/s/	3.90	4.04	+3.59%
/z/	3.68	3.88	+5.43%
/ah/	8.38	8.84	+5.49%
/ih/	5.97	6.22	+4.18%
/aa/	10.64	11.44	+7.52%
/iy/	5.93	6.26	+5.56%
app.	6.45	6.79	+5.27%
test	6.39	6.77	+5.90%

TAB. 2.5 – Gains de prédiction moyens calculés pour les 6 classes de phonèmes extraits de la base NTIMIT (tiré de [28]).

de prédiction apparaissent moins élevés que ceux trouvés dans la litérature (Thyssen dans [135] reporte des gains allant de $14.1dB$ pour un prédicteur LPC jusqu'à $19.3dB$ pour un

prédicteur non linéaire PMC, Diaz dans [51] reporte des gains passant de $13.6dB$ à $15.1dB$ et Townshend dans [137] une augmentation moyenne de $2.8dB$). La raison principale est qu'il ont été calculés sur des signaux de parole bruités et à bande limitée puisqu'il s'agit de signaux téléphoniques. Comme déjà précisé au paragraphe §2.3.3, p. 26, ces mesures ne permettent pas de conclure que les paramètres NPC modélisent des caractéristiques spécifiquement non linéaires. En effet, à ordre de prédiction égal (12 dans le cas qui nous intéresse), le nombre de paramètres libres n'est pas identique d'un codeur à l'autre : 12 dans le cas du filtre LPC mais $12 \times 12 + 12 = 156$ dans le cas du filtre NPC. L'amélioration du gain de prédiction apportée par le codeur NPC peut être tout autant due à sa complexité, nettement supérieure que celle du LPC. Un avantage certain de la structure NPC est précisément de pouvoir augmenter l'ordre de prédiction du filtre, sans pour autant modifier la dimension du vecteur des paramètres. A titre d'exemple, la table 2.6 indique les résultats obtenus et publiés dans [26] pour différentes largeurs de la fenêtre de prédiction et un même nombre de coefficients du vecteur caractéristique égal à 12. On observe bien une sensible

mémoire de prédiction	16	20	25	40
Paramètres libres	204	252	312	492
Gain de prédiction (dB)	6.63	6.78	6.82	6.92

TAB. 2.6 – Gains de prédiction moyens calculés sur les six phonèmes extraits de la base NTIMIT pour différentes largeurs de la fenêtre de prédiction (extrait de [26]).

augmentation du gain de prédiction avec le nombre de paramètres libres du filtre NPC. En revanche, l'analyse faite par Kubin (voir le paragraphe §2.3.3, p. 26) selon laquelle les phénomènes non linéaires sont plus à chercher du côté des signaux voisés que non voisés est corroborée par le tableau 2.5 : l'augmentation du gain de prédiction apportée par le modèle NPC est plus faible pour le seul phonème non voisé /s/ que pour l'ensemble des autres phonèmes voisés.

D'autres mesures, démontrant la présence de caractéristiques liées aux aspects non linéaires, ont été proposées dans la litérature [141], basées également sur des rapports d'erreur de prédiction de filtres non linéaires/linéaires. Mohamed Chetouani reporte dans [33] des résultats montrant que les zones non linéaires des fonctions sigmoïdes des cellules d'un réseau NPC sont également exploitées. Ces études nécessiteraient d'être poussées plus avant, surtout dans le cadre de l'extraction de caractéristiques, car jusqu'à présent elles sont surtout le fait de recherches en codage pour la compression/synthèse de la parole.

2.5.6 Horizon de prédiction

Comme indiqué précédemment, la taille de la fenêtre de prédiction a une influence non négligeable sur les gains de prédiction obtenus. Cyril Chavy a mis en évidence les contributions respectives des différents retards sur la prédiction. Pour une entrée i correspondant à un retard i, il définit sa contribution comme la somme des valeurs absolues des poids liant cette entrée à l'une quelconque des cellules de la couche cachée [26] (voir figure 3.1, p. 47) :

$$S_i = \sum_j |\omega_{ij}| \tag{2.4}$$

La figure 2.7 montre une diminution progressive des contributions en fonction du retard, compatible avec la tâche de prédiction à court-terme du réseau NPC. A noter cependant un rehaussement autour du retard 38, pouvant correspondre à une trace de la source glottale à environ $430Hz$. La plupart des expérimentations ont été réalisées avec une valeur de

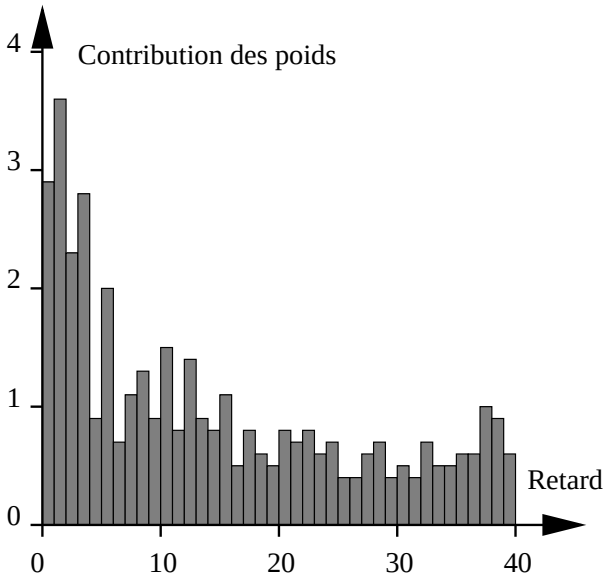


FIG. 2.7 – Contribution des différents retards à la prédiction, d'après [25] et [26].

$\lambda = 20$.
La valeur de λ a également une conséquence importante sur l'estimation des paramètres NPC puisqu'elle conditionne le nombre d'exemples d'apprentissage dont on dispose pour effectuer l'estimation d'un jeu de paramètres NPC sur une trame (voir à ce sujet le paragraphe suivant).

2.5.7 De l'influence du nombre d'itérations

J'aborde dans cette section un point important qui concerne les phénomènes de sur-apprentissage que l'on peut observer lors de l'adaptation de l'extracteur de caractéristiques NPC ou lors du calcul des paramètres caractéristiques. La question particulière du nombre d'itérations en phase d'extraction nous a amené à poser le problème du rapport qu'il y a entre la modélisation et la discrimination, et donc à discuter du caractère discriminant du code généré. La proposition du modèle NPC-2 par Cyril Chavy, et que j'expose au chapitre suivant, en est la conséquence directe.

Les expérimentations menées par Cyril Chavy durant sa thèse et de Mathieu Györffy au cours de son stage de DEA ont mis en évidence un phénomène particulier de sur-apprentissage, que nous avons appelé *sur-modélisation*, lors de la phase d'extraction des caractéristiques (phase d'utilisation de l'extracteur NPC).
Lorsque l'on reporte les scores obtenus en classification de phonèmes en fonction du nombre d'itérations d'apprentissage durant l'extraction de caractéristiques on obtient une courbe

localement "en cloche" qui croise la courbe croissante du gain de prédiction. Je reporte sur la table 2.7 quelques unes de ces mesures qui ont été publiées dans [27]. Les meilleurs

Nombre d'itérations	1	5	10	20	40	100	250	500
Scores (%)	60.1	**61.2**	**61.2**	60	59.6	59.8	59.7	59.7
Gains de prédiction (dB)	1.59	4.62	5.54	6.09	6.45	6.69	6.78	6.81

Nombre d'itérations	1	5	10	20	40	100
Scores (%) sur $\{s, z\}$	64.3	**64.7**	64.1	64.3	63.3	62.6
Scores (%) sur $\{ah, aa, ih, iy\}$	67.1	67.2	68	**68.3**	67.9	67.4

TAB. 2.7 – Scores obtenus en classification pour différentes valeurs du nombre d'itérations d'apprentissage en phase d'extraction de caractéristiques. Pour les 6 phonèmes en haut, et pour deux sous groupes classés séparément en bas (extrait de [27]).

scores en reconnaissance sont obtenus pour un nombre très faible d'itérations (à noter que l'algorithme d'apprentissage est une descente de gradient *stochastique* : une itération représente ici l'apprentissage d'une trame complète, c'est à dire l'apprentissage de $D - \lambda = 256 - 20 = 236$ fenêtres de signal). Passé un certain nombre d'itérations, les scores obtenus commencent à décroître. On observe un comportement différent des gains de prédiction qui sont en croissance continue avec le nombre d'itérations. Le gain de prédiction mesure la qualité de la *modélisation* des phonèmes (minimisation de l'erreur de prédiction) et, d'après ces résultats, s'oppose après un certain nombre d'itérations à la qualité des scores obtenus en reconnaissance. Comme l'indique la table 2.7 le nombre optimal d'itérations

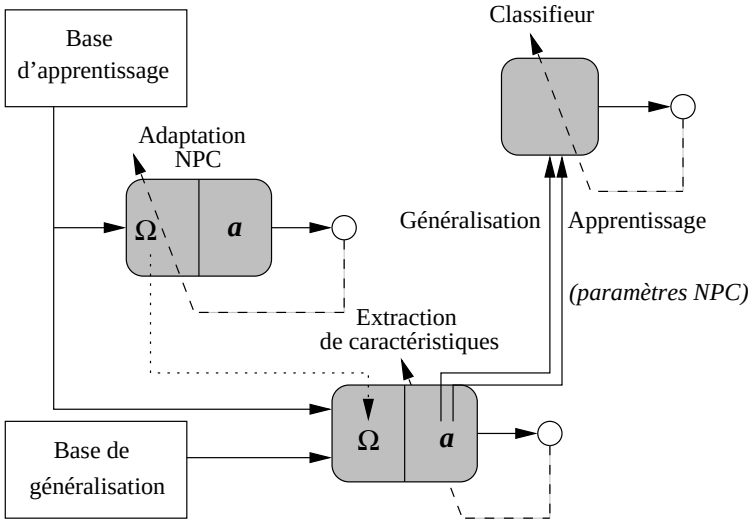


FIG. 2.8 – Schéma bloc représentant l'ensemble du processus : adaptation NPC, extraction des paramètres NPC, apprentissage et test du classifieur.

peut varier d'une classe de phonèmes à l'autre. Dans la pratique cependant, le nombre moyen de 10 itérations permet d'obtenir les meilleurs scores en reconnaissance et c'est ce

nombre que nous avons retenu pour une grande majorité des expérimentations. A noter qu'aucun phénomène de sur-apprentissage n'est observé pendant la phase d'adaptation de l'extracteur NPC : les gains moyens de prédiction augmentent sans occasionner de baisse des scores en classification [26]. Ce résultat s'explique par le très grand nombre de trames utilisées en apprentissage et "vues" par la couche cachée lors de son adaptation. Tel n'est pas le cas en phase d'extraction où le nombre d'exemples $\mathbf{y}_k$ utilisés pour extraire les caractéristiques d'une trame est faible.
En rester là n'était cependant pas satisfaisant. Nous avons souhaité bien entendu identifier les causes de la sur-modélisation et, au mieux, mettre au point un *critère d'arrêt* comme on en trouve pour l'apprentissage des modèles connexionnistes en reconnaissance des formes. La majeur partie du stage de M. Györffy était consacré à cela et son travail, d'une qualité remarquable, nous a apporté des réponses.

Dans le cas qui nous intéresse, l'étude des causes pouvant donner lieu à la sur-modélisation est complexe car le système mis en oeuvre traite simultanément de l'analyse des signaux (paramétrique en l'occurrence) et de leur catégorisation. On peut ainsi trouver plusieurs origines au sur-apprentissage, liées aux différentes étapes de calculs ainsi qu'aux données elles-mêmes. En particulier :

1. Le calcul des paramètres NPC nécessite un algorithme d'apprentissage (tant en phase d'adaptation qu'en phase d'extraction) qui peut donc donner lieu à des problèmes de sur-apprentissage ;
2. Le classifieur utilisé nécessite également un algorithme d'apprentissage donnant lieu aux problèmes de généralisation connus en reconnaissance des formes et liés à la minimisation du *risque empirique* ;
3. Il existe une très forte variabilité "naturelle" entre différentes occurrences d'un même phonème dont j'ai énoncé les différentes causes (cf. §2.5.2, p. 34 et §2.4.3, p. 31). Cette variabilité se traduit par une grande disparité des distributions des paramètres. Or l'extracteur de caractéristiques a une influence directe sur la qualité de ces distributions.

Le travail de stage de M. Györffy a consisté à traiter la première cause énoncée ci-dessus. La deuxième a été écartée car elle concerne exclusivement le classifieur. Enfin, le traitement de la troisième conduira à proposer une deuxième version du codeur NPC.

Plusieurs techniques habituellement utilisées pour limiter les problèmes de sur-apprentissage, comme la cross-validation, les techniques de régularisation ou l'ajout d'exemples "artificiels" ont été testées. Seul l'ajout d'un terme de régularisation à la fonction de coût a permis d'obtenir un résultat intéressant.
La cross-validation consistait à séparer l'ensemble des exemples d'apprentissage disponibles pour l'extraction de caractéristiques d'une trame donnée en deux sous-ensembles, l'un d'apprentissage et l'autre de test. Les problèmes de généralisation que l'on pouvait ainsi mettre en évidence n'apparaissent qu'au delà d'un grand nombre d'itérations, supérieur à 50 le plus souvent, et ne sont donc pas corrélés à la sur-modélisation observée.
La technique de régularisation consistait à ajouter à la fonction de coût un deuxième terme visant à minimiser l'amplitude des poids en valeur absolue. Comme le montre la figure 2.9, la sur-modélisation est nettement diminuée mais au dépend des scores obtenus puisque l'on

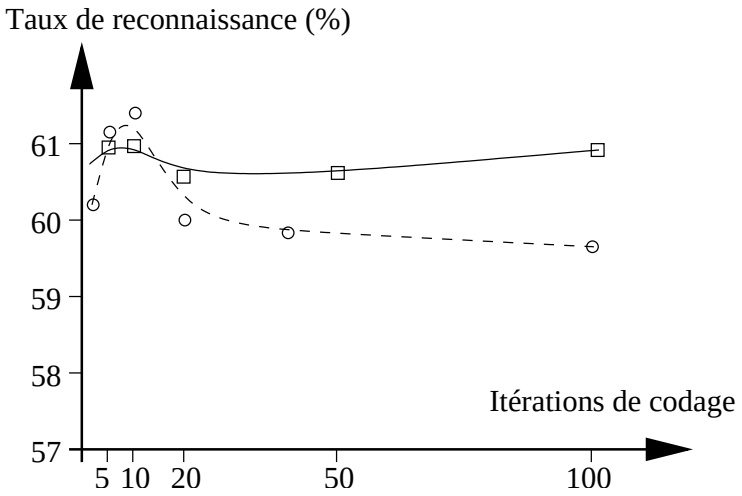


FIG. 2.9 – Scores moyens obtenus avec et sans terme de régulation (tiré du rapport de stage de DEA de M. Györffy).

perd environ 1 point en moyenne (la figure 2.9 représente des taux moyens obtenus sur une dizaine de simulations).

Dans l'ensemble, les différentes expérimentations réalisées n'ont pas permis de mettre en évidence un moyen satisfaisant de s'affranchir du problème de sur-modélisation constaté en phase d'extraction. J'ai pensé avoir atteint là une limite du modèle proposé, non pas due aux algorithmes mis en oeuvre (les méthodes précédentes auraient alors permis de s'affranchir de la sur-modélisation) mais à la nature des données très variables et à la méthodologie de segmentation en trames. A noter que la validité de cette expérimentation est limitée par le faible nombre d'échantillons ($D - \lambda$) dont on dispose pour un apprentissage. On touche là au dilemme lié à la stationnarité du signal : D faible n'autorise pas une bonne analyse, mais D élevé conduit à l'analyse d'un signal non stationnaire (voie de recherche sur laquelle je reviens dans les perspectives (§4.4, p. 88)).
On dispose néanmoins d'un moyen d'agir sur la représentation des caractéristiques et leur distribution dans l'espace des entrées du classifieur puisque l'extracteur NPC est adaptatif. Cet aspect guidera la suite de mes recherches dans ce domaine.

2.6 Conclusion

Dans ce chapitre, j'ai exposé qualitativement le principe de fonctionnement d'un nouveau modèle d'extraction de caractéristiques : le modèle NPC. Ce travail a été l'occasion de co-encadrer cinq étudiants en stage de DEA et un étudiant en thèse, Cyril Chavy. Les expérimentations qu'ils ont menées ont montré que les paramètres NPC permettent d'obtenir une amélioration des scores en reconnaissance de phonèmes.
Contrairement aux modèles concurrents, l'extracteur NPC est non linéaire d'une part, et adaptatif d'autre part. J'ai surtout insisté sur l'aspect non linéaire en exposant différentes justifications de cette approche. L'aspect adaptatif n'est pas exempt d'inconvénients que

j'ai soulignés, comme la complexité en temps de calcul ou les problèmes de sur-apprentissage inhérents à ce genre d'approche. En revanche, les qualités attendues sont la possibilité d'une coopération entre les deux étages d'extraction de caractéristiques et de classification, objet du chapitre suivant de ce mémoire.

.

Chapitre 3

Les modèles NPC-2 et DFE-NPC ou la coopération analyse/classification

Encadrement doctoral : Mohamed Chetouani (2001-2004)

Ainsi que l'expliquent Zahorian et al. dans [144], l'extraction de caractéristiques doit être telle que les vecteurs acoustiques appartenant à la même classe soient proches au sens d'une certaine distance définie dans leur espace de représentation, tandis que des vecteurs de classes différentes soient éloignés dans ce même espace. Si l'on fait l'hypothèse d'une distribution gaussienne des vecteurs caractéristiques, cela signifie qu'un objectif supplémentaire est donné à l'extraction : celui de minimiser la covariance intra-classe et de maximiser conjointement la covariance inter-classes.

Le travail de recherche que je présente dans ce chapitre est centré principalement sur l'étude de cet objectif. J'y présente tour à tour 4 versions du modèle NPC qui sont les paramètres NPC-C (ou paramètres contraints), les paramètres NPC-2 proposés par Cyril Chavy, les paramètres DFE-NPC (Discriminant Feature Extraction) que j'ai proposés moi-même et les paramètres LVQ-NPC proposés par Mohamed Chetouani.
Ces modèles, nombreux, constituent autant d'étapes dans la réflexion que j'ai menée concernant la problématique modélisation/discrimination. Avec le modèle NPC-C, je tente d'imposer explicitement des contraintes lors de l'adaptation du codeur, l'objectif étant d'obtenir des paramètres dont la distribution, supposée gaussienne, minimise la covariance intra-classe et maximise la covariance inter-classes. Le modèle NPC-2 poursuit cette logique en imposant une cellule prédictive par classe phonétique plutôt qu'une cellule par trame. Avec le modèle DFE-NPC, la possibilité est donnée de contrôler les deux caractéristiques antagonistes que sont la qualité de la modélisation et la qualité de la classification. L'adaptation vise alors la maximisation du critère MMI (Maximum Mutual Information). Enfin, les paramètres LVQ-NPC sont obtenus à l'aide de contraintes explicites guidées par le classifieur et non plus par un critère extérieur comme le critère MMI. C'est un modèle coopératif extracteur/classifieur empruntant le formalisme MCE-GPD proposé par Juang et Katagiri [91].

3.1 La coopération extracteur/classifieur

La minimisation de la covariance intra-classe et la maximisation conjointe de la covariance inter-classes nécessitent de disposer d'informations de classe d'appartenance des signaux que l'on souhaite paramétrer. Par hypothèse, ceci ne peut pas être satisfait lors de la phase d'extraction de caractéristiques puisque l'on se situe précisément avant l'étape de classification. En revanche, lors de l'étape d'adaptation du codeur NPC, on suppose que l'on connaît la classe d'appartenance des signaux utilisés. En ce sens, les modèles NPC-C, NPC-2 et DFE-NPC sont trois algorithmes d'adaptation de la première couche du codeur NPC, l'extraction de caractéristiques restant inchangée par ailleurs.

Disposer des classes d'appartenance des signaux dès l'étape d'analyse conduit à envisager la coopération des deux étages de traitement que sont l'extraction de caractéristiques et la classification. Différents modèles ont été proposés dans la litérature sous le vocable d'*extraction de caractéristiques discriminantes* (DFE). Les techniques DFE consistent à extraire du signal un vecteur de caractéristiques dont les composantes sont rehaussées en regard de la classification qui doit suivre. Ainsi, l'analyse DFE d'un signal pour la reconnaissance de phonèmes ne donnera pas les mêmes caractéristiques que la même analyse pour une reconnaissance de locuteurs.
La sélection d'un espace de représentation des données adapté à la classification a été formalisée par Juang et Katagiri [92], [87], [91] dans un cadre plus général : celui de la *minimisation de l'erreur de classification* ou MCE (Minimum Classification Error). Katagiri est parti dans le début des années 90 sur le constat que les critères d'apprentissage les plus souvent utilisés, le critère MSE (*Mean Squared Error*) de l'erreur quadratique moyenne par exemple, ne minimisent pas exactement l'erreur de classification, même si dans certains cas idéaux, ils permettent d'estimer des probabilités a posteriori et donc d'appliquer parfaitement bien la règle de décision bayesienne. Reprenant une approche plus ancienne d'Amari [3], Katagiri et al. [92] ont proposé une fonction erreur qui reflète directement les erreurs de classification. Elle est continue et différentiable de sorte à pouvoir utiliser une technique de descente de gradient pour l'apprentissage : en occurrence une règle de descente probabiliste (*Generalized Probabilistic Descent* ou GPD). C'est la méthode MCE/GPD, déjà appliquée aux modèles de type perceptron multicouches [93] puis aux modèles LVQ [112] ou encore aux modèles HMM [93].

Dans le cas qui nous préoccupe, à savoir adapter l'extracteur de caractéristiques à la tâche de classification, la méthode MCE/GPD a déjà été utilisée. Biem et Katagiri [10] ont par exemple proposé d'estimer les fréquences centrales et largeurs des bancs de filtres d'un codeur fréquentiel. Les mêmes auteurs ont proposé d'appliquer cette méthode à l'estimation des paramètres d'un filtrage cepstral [9]. Dans ces exemples, les paramètres du filtre sont déterminés par l'algorithme de rétropropagation de l'erreur mis en œuvre lors de l'apprentissage du classifieur neuronal. L'apprentissage du classifieur et l'adaptation des paramètres du codeur sont donc effectués en une seule et même phase. Mohamed Chetouani a proposé durant sa thèse une version du codeur fonctionnant selon ce principe, l'extracteur LVQ-NPC, mais bâti sur un classifieur LVQ et s'apparentant plus aux travaux de McDermott sur la question [112], [117]. Je le présente au paragraphe 3.5, p. 65 de ce chapitre.

3.2 L'adaptation sous contraintes : un succès relatif

Encadrement : Serges Fornesi (DEA d'Electronique 1999), Julien Soleilhac (DEA de Robotique 2000)
Publications : [65], **[68]**, [25]

Avant de décrire le principe de l'adaptation avec contraintes, je reviens sur la description des deux phases d'adaptation et d'extraction du modèle NPC. Je la complète en donnant l'expression des fonctions de coût minimisées lors de ces deux phases d'apprentissage. L'idée sous-jacente aux paramètres NPC-C est d'ajouter un ou plusieurs termes de contrainte durant la phase d'adaptation afin d'obtenir un extracteur plus discriminant au sens de la minimisation de la variance intra-classe et de la maximisation de la variance inter-classes. Ainsi que les expérimentations menées par Serges Fornesi durant son stage de DEA le montreront, il est difficile d'obtenir un tel résultat.

3.2.1 Adaptation de l'extracteur NPC

J'ai expliqué au chapitre précédent que la conception du prédicteur NPC partait de l'idée que seuls les poids liant les cellules de la couche cachée à la cellule de sortie, encore appelée *cellule de prédiction*, sont extraits pour constituer le vecteur de code ou *paramètres NPC*. Nous avons vu que cela conduisait au problème de l'estimation des poids de la couche cachée, lequel est résolu par la méthode de l'apprentissage "en bloc" d'une structure du prédicteur comportant une cellule prédictive par trame présente dans la base. Nous avons appelé cette étape *phase d'adaptation* de l'extracteur (figure 3.1). Cela nous conduit à

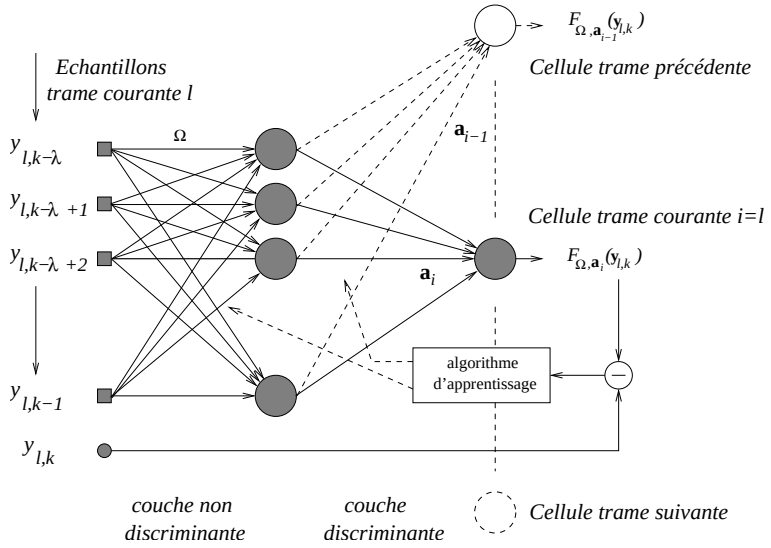


FIG. 3.1 – Structure du réseau NPC en phase d'adaptation (extrait de [66]).

envisager la formulation la plus générale pour laquelle le prédicteur neuronal réalise un opérateur $F_{\Omega,\mathbf{a_1},\ldots,\mathbf{a_L}}$ de $[-1,+1]^\lambda$ dans $[-1,+1]^L$ tel que :

$$F_{\Omega,\mathbf{a_1},\ldots,\mathbf{a_L}} : \mathbf{y}_k = [y_{k-1},\ldots,y_{k-\lambda}] \longrightarrow [\hat{y}_k^1,\ldots,\hat{y}_k^L] = F_{\Omega,\mathbf{a_1},\ldots,\mathbf{a_L}}(\mathbf{y}_k) \qquad (3.1)$$

où l'on a : $\hat{y}_k^i = F_{\Omega,\mathbf{a_i}}(\mathbf{y}_k)$ et L désigne le nombre total de cellules de sortie et D la longueur d'une trame en nombre d'échantillons. Les poids de la couche cachée sont donc obtenus par minimisation de l'erreur quadratique moyenne :

$$Q^{NPC}(\Omega, \mathbf{a}_{1,\ldots,L}) = \frac{1}{2} \sum_{l=1}^{L} \sum_{i=1}^{L} \sum_{k=\lambda}^{D-1} \left(y_{l,k} - F_{\Omega,\mathbf{a}_i}(\mathbf{y}_{l,k})\right)^2 \delta_{i-l} \tag{3.2}$$

où $\delta_{i-l} = 1$ lorsque que $i = l$ et 0 sinon. L'indice l parcours toutes les trames de la base tandis que l'indice i parcours toutes les cellules de sortie du réseau. δ_{i-l} permet pour une trame l donnée de ne comptabiliser que la sortie $i = l$ du réseau correspondant à la trame.

3.2.2 Extraction des paramètres NPC

L'extraction des paramètres NPC peut avoir lieu lorsque que l'on dispose d'un extracteur *adapté*, c'est à dire lorsque les poids de la première couche ont été estimés comme indiqué dans le paragraphe précédent. La première couche agit alors comme un opérateur G_Ω de transformation. Ainsi, pour une trame quelconque l dont on souhaite extraire les paramètres NPC, on cherche le modèle $F_{\Omega,\mathbf{a}} = G_\Omega \circ H_\mathbf{a}$ le plus proche du processus ayant produit la trame. L'opérateur $H_\mathbf{a}$ représente la fonction réalisée par la couche de sortie du prédicteur et $\mathbf{a}$ les poids de cette couche que l'on obtient par minimisation de l'erreur quadratique de prédiction :

$$Q^{NPC}(\mathbf{a}) = \frac{1}{2} \sum_{k=\lambda}^{D-1} \left(y_{l,k} - H_\mathbf{a}(\mathbf{z}_{l,k})\right)^2 \text{ où } \mathbf{z}_{l,k} = G_\Omega(\mathbf{y}_{l,k}) \tag{3.3}$$

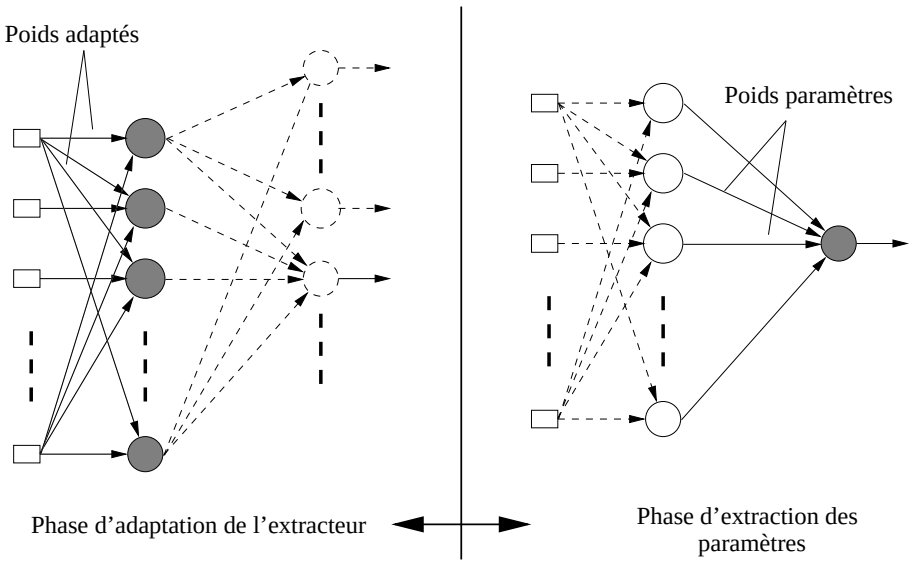


FIG. 3.2 – Phase d'adaptation de l'extracteur à gauche puis phase d'extraction des paramètres NPC, à droite (extrait de [68]).

3.2.3 Adaptation sous contraintes : les paramètres NPC-C

Le principal objet du stage de DEA de Serge Fornesi était d'expérimenter une méthode d'adaptation des poids de la première couche en ajoutant une contrainte sur les poids de la deuxième couche de sorte à minimiser la variance intra-classe de ces poids. Nous avons pour cela ajouté au coût quadratique moyen un terme de pénalité, pondéré par un coefficient α :

$$Q^{NPC-C}(\Omega, \Theta) = \alpha Q^M(\Omega, \Theta) + (1 - \alpha) Q^D(\Omega, \Theta) \tag{3.4}$$

avec $\Theta = [\mathbf{a}_1, \ldots, \mathbf{a}_L]$ représente l'ensemble des poids de la couche de sortie. Q^M est le terme de coût lié à l'erreur de prédiction, appelé coût de *modélisation*, définit plus haut (eq. 3.2) et Q^D le coût de *discrimination* défini à partir d'une estimation des covariances inter et intra-classe.
Disposant de l'ensemble des L trames de la base d'apprentissage, ces trames doivent être maintenant groupées par classes d'appartenance. On estime à toute itération q la variance empirique de chacune des classes en supposant la distribution normale sphérique :

$$\sigma_c^2(q) = \frac{1}{NL_c} \sum_{l=1}^{L_c} \|\mathbf{a}_l - \overline{\mathbf{a}_c}\|^2 \tag{3.5}$$

où $\overline{\mathbf{a}_c}(q)$ est la moyenne empirique de la classe estimée par

$$\overline{\mathbf{a}_c} = \frac{1}{L_c} \sum_{l=1}^{L_c} \mathbf{a}_l \tag{3.6}$$

à l'itération q de l'apprentissage. L_c est le nombre de trames de la classe c et N la dimension des données (le nombre de coefficients des vecteurs paramètres). J'ai volontairement omis les indices q par soucis de lisibilité des équations.
La figure 3.3 représente le résultat d'une analyse discriminante effectuée sur les poids codes à plusieurs moments de l'adaptation de la première couche. Elle met clairement en évidence l'impact des contraintes appliquées et visant ici à minimiser la variance intra-classe des paramètres. Mais l'on peut observer également que les distributions ne sont pas à proprement parler sphériques. Ces résultats ont été obtenus en procédant à l'adaptation en deux étapes. Jusqu'à l'itération 4800, seule l'erreur de prédiction (modélisation) était minimisée avec $\alpha = 1$. Nous avons ensuite appliqué la contrainte de discrimination avec un poids très faible $\alpha = 0.99$. La figure 3.4 reporte l'évolution de l'erreur de prédiction en cours d'adaptation et l'on peut ainsi observer l'impact de l'application de la contrainte discriminante dès l'itération 4800.

Contrairement à ce que pourraient laisser penser ces figures, les quatre mois d'expérimentations menées par Serges Fornesi ont débouché sur des conclusions fort peu encourageantes. Je ne reporterai ici qu'un seul résultat, celui du meilleur score obtenu en généralisation sur les mêmes bases de données (NTIMIT) que précédemment, à savoir 61.5%. Très généralement, on peut dire que la minimisation des variances intra-classe obtenues lors de l'adaptation du codeur ne se retrouvaient pas sur les poids-codes générés en phase d'extraction, même en tenant compte du problème de sur-modélisation. La contribution de ces contraintes au calcul des poids-paramètres restait donc très faible.

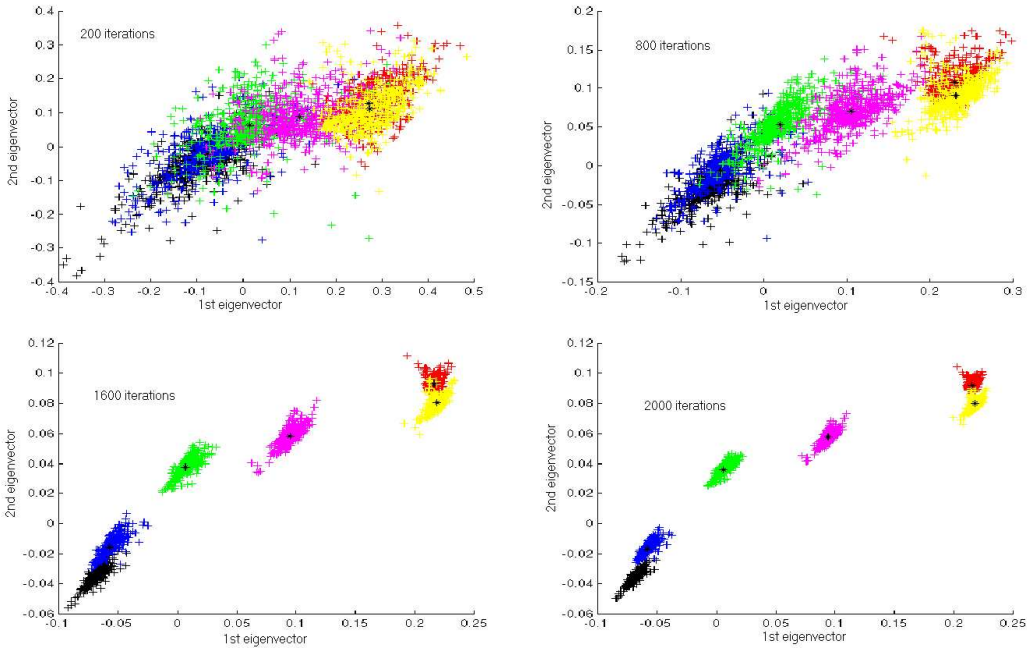


FIG. 3.3 – Analyses discriminantes calculées sur les poids-codes de la deuxième couche au cours de la phase d'adaptation (tiré de l'article [68]). Les analyses sont présentées à partir de l'itération 4800.

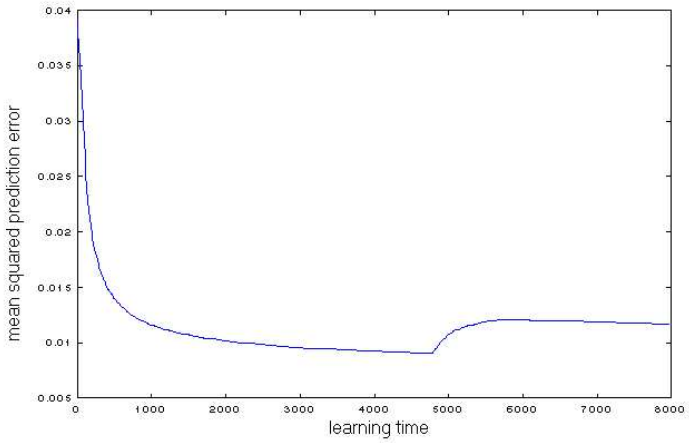


FIG. 3.4 – Évolution de l'erreur de prédiction au cours de la phase d'adaptation (tiré de l'article [68]).

Beaucoup de mesures sur les données ont été effectuées pour s'assurer de la justesse de ces résultats. La maximisation conjointe de la covariance inter-classes a également été testée sous la forme d'un troisième terme dans la fonction de coût à minimiser sans résultat convainquant. La généralisation de ces contraintes au cas de données de distribution nor-

male mais non sphérique a également été testée, mais sans plus de succès. Je ne reporte pas ici tous les détails de ces résultats que l'on peut trouver dans le rapport de stage de Serge Fornesi.
C'est plus tard, en tentant de valider le modèle NPC et en proposant d'autres contraintes (modèles DFE-NPC et LVQ-NPC), que je comprendrai les raisons de cet échec relatif (cf. §4.4.2, p. 89). Relatif car les recherches effectuées dans le sens d'accroître les performances discriminantes des paramètres NPC a conduit Cyril Chavy a proposer un modèle plus simple dont on peut dire qu'il est une limite du modèle NPC-C au cas des contraintes dites *maximales*.

3.3 Le modèle NPC-2 ou le modèle des contraintes maximales

Encadrement : Mathieu Györffy (DEA d'Electronique 1999), Julien Soleilhac (DEA d'Electronique 2000)
Publications : [26], [65], [67], **[68]**

Le modèle NPC-2 est une extension du modèle NPC qui permet la prise en compte d'informations de classe d'appartenance des trames durant la phase d'adaptation. La phase d'extraction reste identique à celle du codeur NPC. Après la définition des paramètres NPC-2, je présente dans cette section une analyse comparative des résultats obtenus et je reprend la discussion entamée au chapitre précédent (cf. §2.5.7, p. 39) et concernant la sur-modélisation. Je termine en introduisant la problématique de la coopération entre l'extraction de caractéristiques et la classification.

3.3.1 Définition du modèle

Si l'on reprend le coût quadratique défini pour le modèle NPC-C (eq. 3.4, p. 49), le coût de discrimination Q^D atteint son minimum pour une valeur particulière des poids-codes de la deuxième couche :

$$\forall l,\ \mathbf{a}_{i=l} = \overline{\mathbf{a}}_{c(l)} \Longrightarrow \sigma_c^2 = 0 \text{ et } Q^D = 0 \tag{3.7}$$

Si donc l'on contraint l'ensemble des poids-codes $\mathbf{a}_i$ correspondant aux trames l d'une même classe phonétique c à être identiques à leur valeur moyenne $\overline{\mathbf{a}_c}$, la fonction de coût Q^D atteint d'emblée son minimum. Ce résultat revient à modifier la structure du codeur lors de la phase d'adaptation : la contrainte étant continuellement imposée, on obtient un réseau à une seconde couche par classe plutôt qu'une seconde couche par trame et la fonction de coût devient :

$$Q^{NPC-2}(\Omega,\Theta) = \frac{1}{2}\sum_{l=1}^{L}\sum_{i=1}^{N_C}\sum_{k=\lambda}^{D-1}\left(y_{l,k} - F_{\Omega,\mathbf{a}_i}(\mathbf{y}_{l,k})\right)^2 \delta_{i-c(l)} \tag{3.8}$$

où $F_{\Omega,\mathbf{a}_i}(\mathbf{y}_{l,k})$ représente la sortie de la cellule de prédiction i associée à la classe $c(l)$ lorsque l'on présente un vecteur $\mathbf{y}_{l,k}$ de la trame l en entrée et N_C le nombre total de classes. Cette idée a été proposée par Cyril Chavy pendant sa thèse et publiée la première fois dans [26].

3.3.2 Les premiers résultats

Une amélioration sensible des scores en classification (de l'ordre de un point, toutes expériences confondues, relativement aux paramètres NPC et de deux à trois points relativement aux paramètres MFCC et LPC) a été obtenue avec les paramètres NPC-2. Les premiers résultats ont été publiés au GRETSI en 1999 [65]. La figure 3.5 présente une comparaison des scores obtenus à l'aide de différents codeurs (LPC, MFCC, NPC et NPC-2). Il est

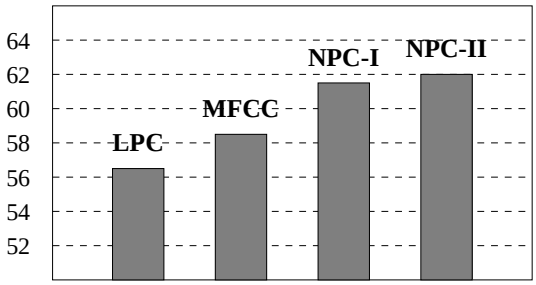


FIG. 3.5 – Scores obtenus par différents paramètres à l'aide d'un classifieur PMC (tiré de l'article [65]).

intéressant de comparer l'évolution de l'erreur de prédiction durant la phase d'adaptation pour les deux modèles NPC et NPC-2 dans les mêmes conditions. C'est ce que présente la figure 3.6 sur près de 4000 itérations d'apprentissage. On note l'"effort" supplémentaire demandé au codeur NPC-2 en terme de discrimination et qui se traduit par une erreur de prédiction résiduelle plus élevée. Ces résultats ont été complétés par des expérimentations

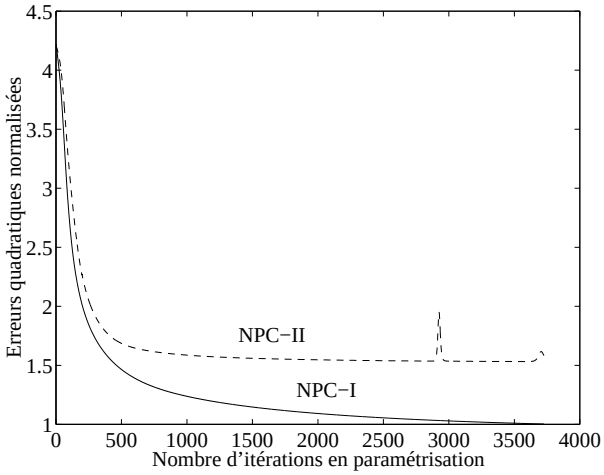


FIG. 3.6 – Evolution des coûts Q^{NPC} et Q^{NPC-2} durant la phase d'adaptation sur des données identiques (tiré de l'article [65]).

effectuées sur d'autres phonèmes de la base NTIMIT, en particulier un sous groupe de phonèmes voisés : /*aa*/, /*ae*/, /*ey*/, /*ow*/ et deux sous groupes régulièrement testés dans la littérature car fréquents et réputés difficiles. Ce sont les plosives voisées /*b*/, /*d*/, /*g*/ et

non voisées /p/, /t/, /k/, testées notamment par Waibel et Lang [138], [103] pour valider leur modèle (Le TDNN). Le tableau 3.1 rassemble les différents scores obtenus sur chacune de ces bases. Ces scores sont des scores "moyens" estimés sur une dizaine de simulations.

Phonèmes	LPC	MFCC	NPC	NPC-2
/aa/, /ae/, /ey/, /ow/	55.5%	58%	61.2%	**63%**
/b/, /d/, /g/	57.5%	62.3%	65%	**70.2%**
/p/, /t/, /k/	61%	**68%**	61.5%	65%

TAB. 3.1 – Scores comparatifs obtenus en généralisation sur trois groupes représentatifs de phonèmes extraits de la base NTIMIT (tiré des articles [67] et [68])

Ils mettent en évidence la supériorité des taux de reconnaissance obtenus à partir des paramètres NPC-2.
On peut noter que les résultats sur les phonèmes /p/,/t/,/k/ sont pratiquement tous équivalents pour toutes les méthodes de codage utilisées (LPC, MFCC, NPC, NPC-2). Par ailleurs, la méthode NPC-2 donne de meilleurs scores pour les phonèmes /aa/, /ae/, /ey/, /ow/ et /b/,/d/,/g/ : plus de 60 à 65% de reconnaissance contre 58 à 62% avec les paramètres MFCC. Ces phonèmes sont voisés tandis que les phonèmes /p/,/t/,/k/ ne le sont pas. Ainsi que nous l'avons souligné au paragraphe 2.3.3 du chapitre précédent, les phonèmes non voisés sont correctement représentés par le modèle linéaire source-filtre. L'extracteur non linéaire n'apporte donc pas d'informations supplémentaires significatives, ce que l'on retrouve dans cette comparaison des scores où les paramètres NPC-2 restent inférieurs à ceux obtenus par les paramètres MFCC.

Ces résultats encourageants m'ont incité à poursuivre dans la voie des extracteurs de caractéristiques "discriminants", c'est à dire faisant explicitement appel aux classes d'appartenance des signaux dès l'étape d'analyse. La suite de mes recherches sera donc essentiellement tournée vers l'étude des différents modes de coopération entre l'extraction de caractéristiques et la classification.

3.3.3 Sur-modélisation NPC-2

La question de la sur-modélisation (cf. §2.5.7, p. 39) reste posée avec les paramètres NPC-2. La figure 3.6 montre que la contrainte consistant à imposer une seconde couche par classe conduit à une erreur de prédiction résiduelle plus élevée. Cette contrainte n'a plus lieu lors de l'extraction de caractéristiques (les classes d'appartenance n'étant pas connues, l'algorithme mis en oeuvre est exactement celui du modèle NPC). L'erreur de prédiction est donc minimisée au delà de ce qu'elle peut l'être lors de la phase d'adaptation. Sans critère d'arrêt, la sur-modélisation semble donc inévitable. La table 3.2 reporte des mesures des scores de classification obtenus sur les deux groupes de phonèmes NTIMIT /aa/, /ae/, /ey/, /ow/ et /b/, /d/, /g/ pour différentes valeurs du nombre d'itérations d'apprentissage lors de la phase d'extraction de caractéristiques. On retrouve à nouveau un maximum avant ou autour des 20 premières itérations, selon les phonèmes traités : augmenter le nombre d'itérations d'apprentissage lors de la phase d'extraction conduit ainsi à une baisse de performance en classification.

Itérations	phonèmes	5	10	20	30	100	150	200	500
Scores(%)	/aa/, /ae/, /ey/, /ow/	59.2	62	**62.6**	61.8	59.8	59.5	58.5	58
Scores(%)	/b/, /d/, /g/	**70**	64	56	55	49.6	48	48.1	48

TAB. 3.2 – Evolution des scores en reconnaissance en fonction du nombre d'itérations de codage lors de l'extraction de caractéristiques (tiré de l'article [68]).

Un autre point mis en évidence est que le nombre optimal d'itérations varie d'un groupe de phonèmes à l'autre. La nécessité de ne retenir qu'une valeur entraîne qu'elle sera optimale pour certaines catégories de phonèmes, mais sous-optimale pour d'autres. Cet argument ira dans le sens de concevoir un extracteur *modulaire*, composé d'un ensemble de systèmes NPC dédiés chacun à des sous groupes particuliers de phonèmes, thème que j'aborde au paragraphe 3.4.4, p. 60 de ce mémoire.

3.3.4 Peut-on réaliser un classifieur NPC-2 ?

C'est relativement tôt que nous nous sommes posé la question de réaliser un système permettant d'intégrer l'extraction de caractéristiques et la classification. La solution consistant à faire disparaître l'extracteur n'est certainement pas appropriée dans le cas des signaux de paroles. La redondance du signal de parole est telle qu'il apparaît hasardeux d'envisager une classification qui ne soit pas précédée d'une étape de réduction de la dimension des données. Des auteurs comme Bojer et Hammer [19] se sont également posés la question de la réduction de la dimension des données (des images satellitaires en l'occurence), simultanément à leur catégorisation. Ils utilisent pour cela le modèle des cartes auto-organisantes (SOM) proposées initialement par Kohonen. L'évolution actuelle du codeur NPC va dans ce sens, comme je l'explique à la fin de ce mémoire (4.2, p. 76).

Ouvrant la voie de cette démarche, Cyril Chavy et Mathieu Györffy ont tenté de réaliser une *classification prédictive* à l'aide du modèle NPC-2. L'on dispose en effet, après la phase d'adaptation, d'une seconde couche par classe. Chacune de ces secondes couches minimise l'erreur de prédiction sur l'ensemble des phonèmes appartenant à la catégorie phonétique qu'elle représente. La décision est réalisée en affectant à une trame de classe inconnue le modèle le plus vraisemblable : celui qui minimise l'erreur de prédiction :

$$\hat{c}(l) = \arg \min_{i=1,\ldots,N_c} \left\{ \sum_{k=\lambda}^{D-1} (y_{l,k} - F_{\Omega,\mathbf{a}_i}(\mathbf{y}_{l,k}))^2 \right\} = \arg \min_{i=1,\ldots,N_c} \{Q_l(\Omega, \mathbf{a}_i)\} \qquad (3.9)$$

où $Q_l(\Omega, \mathbf{a}_i)$ représente l'erreur de prédiction calculée sur la trame courante l en utilisant la cellule de prédiction i et D la longueur des trames en nombre d'échantillons.

Les scores obtenus par cette méthode de classification ne sont pas probants, inférieurs à ceux que l'on peut obtenir avec un classifieur séparé de type PMC comme ceux que nous utilisons. En général, les classifieurs qui procèdent par *modélisation* des données sont insuffisament discriminants (cf. les réseaux diabolos en reconnaissance de caractères, ou bien les HMM et GMM en parole) et nécessitent d'être renforcés par des critères explicitement

discriminants, comme pour le modèle DFE-NPC que je présente ci-dessous. Mais dans le cas présent, la raison principale provient de l'insuffisance de représentation de chacune des classes. Une cellule prédictive par classe s'avère largement insuffisant étant donnée la variabilité du signal de parole. J'aurai l'occasion de revenir sur ce problème de la classification prédictive à plusieurs reprises jusqu'à la définition du modèle SOM-NPC (§4.2, p. 76).

3.4 Les modèles DFE-NPC

Encadrement : Mohamed Chetouani (DEA d'Electronique 2001)
Publications : Voir les sous-paragraphes.

Ainsi que je l'ai indiqué plus haut, les méthodes DFE (Discriminant Feature Extraction) introduites dans la littérature par Juang et Katagiri [87], [91], caractérisent les dispositifs d'extraction de caractéristiques explicitement guidés par la tâche de classification. Les auteurs ont formalisé cette approche autour de la notion de descente probabiliste du gradient (GPD) en utilisant un critère spécifique appelé *critère de l'erreur de classification minimum* (MCE) calculé sur l'ensemble du système extracteur et classifieur. Ils l'ont appliqué notamment à la réalisation de bancs de filtres adaptables [9], [10]. Cependant, lorsque les deux étapes font appel à des algorithmes de nature différente, il devient difficile de minimiser un critère commun. De la Torre et al. [49] ont proposé des extensions de ces méthodes permettant un apprentissage séparé des deux modules. Nous avons également travaillé dans cette voie en définissant un critère (le critère du rapport d'erreur de modélisation LMER ou *Log Modelization Error Ratio*) permettant de s'affranchir de tout classifieur pendant la phase d'adaptation.
Les travaux que je présente dans cette section ont été réalisés dans le cadre du stage de DEA de Mohamed Chetouani, pour une part, puis dans le cadre de sa thèse lorsqu'il s'est agit d'étendre l'extraction de caractéristiques à l'ensemble des phonèmes. Mohamed Chetouani a en effet proposé deux architectures permettant d'étendre l'extracteur NPC, jusqu'alors testé sur des sous groupes phonétiques restreints, à l'ensemble des classes phonétiques. Je présente ces deux architectures au paragraphe 3.4.4, p. 60. Avant cela, j'introduis la *distance NPC* et le critère MER, utilisés pour établir une nouvelle fonction de coût discriminante et dont Mohamed Chetouani a montré que la minimisation est équivalente à la maximisation de l'information mutuelle (critère MMI, §3.4.3, p. 60).

3.4.1 La distance NPC et le rapport de modélisation LMER

Publications : [67], **[68]**

Jusqu'à présent, je n'ai pas abordé le problème de la validation expérimentale et théorique des modèles NPC. La principale question posée est celle de la validité de l'hypothèse des poids-codes : les poids de la couche de sortie codent-ils les informations permettant de séparer les signaux en classes distinctes ? Le succès des premières expérimentations mentionnées au chapitre précédent a apporté une confirmation expérimentale à ce qui n'était au départ qu'une intuition. J'ai souhaité aller plus avant, tant sur le plan expérimental que sur le plan théorique.
La distance NPC est une mesure de similarité calquée sur la distance d'Itakura [83] que j'ai introduite en 2000 [67], précisément pour évaluer les capacités discriminantes d'un

codeur NPC. Je l'utiliserai également un peu plus tard pour la proposition du modèle non supervisé SOM-NPC.

Au début des années 70, les coefficients LPC sont proposés pour le codage efficient du signal de parole [84] [6], mais les premiers résultats en reconnaissance ne sont pas jugés probants. Itakura considère que c'est la complexité de l'espace des paramètres LPC qui empêche d'établir une rêgle de décision performante. Il propose d'utiliser une nouvelle distance [83] et montre que, d'un point de vue statistique, le critère de décision (en l'occurence le logarithme du rapport des erreurs de prédiction) est équivalent au logarithme du rapport de vraisemblance. La distance d'*Itakura* sera utilisée comme distance locale, permettant de comparer des segments de parole au sein d'algorithmes de programmation dynamique (reconnaissance de mots isolés).
Suivant le même principe, la distance NPC est également définie comme le rapport des erreurs de prédiction :

$$d^{NPC}(l,m) = \log \frac{Q_m(\mathbf{a}_l)}{Q_m(\mathbf{a}_m)} \tag{3.10}$$

où $Q_m(\mathbf{a}_m)$ désigne l'erreur de prédiction calculée sur une trame m lorsque l'on utilise les paramètres NPC $\mathbf{a}_m$ estimés à partir de cette même trame et $Q_m(\mathbf{a}_l)$ l'erreur de prédiction calculée sur une trame m lorsque l'on utilise les paramètres NPC $\mathbf{a}_l$ estimés à partir de la trame l. Ainsi, lorsque $l = m$, on obtient $d^{NPC}(l,l) = 0$. Par ailleurs, lorsque la trame m s'éloigne de la trame l, elle est moins bien prédite par les paramètres NPC $\mathbf{a}_l$ et l'on a donc $Q_m(\mathbf{a}_l) > Q_m(\mathbf{a}_m)$, d'où $d^{NPC}(l,m) > 0$.
d^{NPC} est en fait une mesure de dissemblance car, de même que la distance d'Itakura, elle n'est pas symétrique. En revanche, à la différence cette fois de la distance d'Itakura, la distance NPC dépend des poids de la (ou des) couche(s) cachée(s) Ω. La bonne réalisation de la propriété $m \neq l \Rightarrow d_\Omega^{NPC}(m,l) > 0$ n'est pas assurée pour toute valeur de ces poids.
Afin de mesurer l'évolution du caractère discriminant du codeur NPC-2 au cours de la phase d'adaptation, j'ai proposé la définition d'un rapport des erreurs de prédiction que nous avons appelé *Modelization Error Ratio* (MER), ou LMER selon que l'on considère ou non le logarithme. Le MER est défini comme suit :

$$\Gamma = \frac{\sum_l \sum_{i,i\neq c(l)} Q_l(\mathbf{a}_i)}{\sum_l Q_l(\mathbf{a}_{c(l)})} \tag{3.11}$$

Au numérateur, la somme des erreurs de prédiction obtenues lorsque l'on présente les trames de parole l de la base d'adaptation en utilisant les poids-codes $\mathbf{a}_i$ qui ne correspondent pas à leur classe d'appartenance $c(l)$. Au dénominateur, la somme des erreurs de prédiction obtenues lorsque l'on présente les trames de parole l en utilisant les poids-codes $\mathbf{a}_{i,i=c(l)}$ correspondant à leur classe d'appartenance $c(l)$. Plus l'amplitude du MER est élevée, plus l'extracteur s'avère discriminant au sens d'une classification qui aurait lieu par minimisation de l'erreur de prédiction, ainsi que nous l'avons vu au paragraphe §3.3.4, p.54.
Les figures 3.7 p. 58, représentent l'évolution du MER pendant la phase d'adaptation des deux bases /b/, /d/, /g/ et /p/, /t/, /k/. A $t = 0$, le MER démarre en 1 (donc le

LMER en 0) : l'extracteur, lorsque ses paramètres Ω sont initialisés aléatoirement, n'est pas discriminant, les erreurs de prédiction restant semblables quelque soient les jeux de poids-codes utilisés.

Les courbes présentent une forte croissance dès les premières itérations et atteignent un maximum dès les 50 premières itérations (ce fait a été observé quelque soient les phonèmes étudiés). Passé ce maximum, les propriétés discriminantes du codeur sont amoindries au fur et à mesure des itérations d'apprentissage. Dans le même temps la qualité de la modélisation est bien entendue améliorée (les mesures du gain de prédiction l'attestent).
Au paragraphe 2.5.7 p. 39, lors de l'étude de la sur-modélisation, j'ai écrit « aucun phénomène de sur-apprentissage n'est observé pendant la phase d'adaptation de l'extracteur NPC ». Les figures 3.7 et 3.8 montrant l'évolution du MER en cours d'adaptation indiquent donc le contraire.
Cette contradiction s'explique car, en fait, l'on ne considère pas le même système. Avec le MER, on obtient un indice de ce que serait l'erreur de classification *par modélisation*, c'est à dire en effectuant une classification prédictive (décision par minimisation de l'erreur de prédiction), comme indiqué au paragraphe «*Peut-on réaliser un classifieur NPC-2 ?*» p. 54. Lorsque l'on classe les paramètres NPC à l'aide d'un système indépendant, le problème de sur-apprentissage se retrouve lors de la phase d'extraction des paramètres, c'est à dire lors du calcul des poids-codes de la couche de sortie, et non pas lors de l'adaptation, comme cela a été mis en évidence.

3.4.2 Contraintes MER

Publications : [33], [43], **[68]**

Le stage de Mohamed Chetouani a été essentiellement consacré à l'analyse du MER et à la mise en oeuvre d'un algorithme d'adaptation basé sur la maximisation du MER. En effet, l'objectif étant d'augmenter les caractéristiques discriminantes du codeur NPC, il semblait naturel de définir une nouvelle fonction de coût basée sur la minimisation de l'inverse du MER. Le troisième modèle NPC ainsi obtenu a été baptisé *DFE-NPC* et publié pour la première fois dans [33].
Reprenons la définition du MER (eq.3.11, p.56) en notant Q^D le *coût de discrimination* et Q^M le *coût de modélisation* :

$$\Gamma = \frac{Q^D}{Q^M} \tag{3.12}$$

La nouvelle fonction à minimiser est $Q^{DFE-NPC} = 1/\log\Gamma$, dont le gradient relativement à un poids quelquonque est :

$$\frac{\partial Q}{\partial \omega} = \frac{\partial}{\partial \omega}\left(\frac{1}{\log\Gamma}\right) = -\frac{1}{\log^2\Gamma}\frac{\partial}{\partial\omega}\left(\log\Gamma\right) = -\frac{1}{\log^2\Gamma}\left(\frac{1}{\Gamma}\frac{\partial\Gamma}{\partial\omega}\right) \tag{3.13}$$

Il vient :

$$\frac{\partial Q}{\partial \omega} = \frac{1}{\log^2\Gamma}\left(\frac{1}{Q^M}\frac{\partial Q^M}{\partial\omega} - \frac{1}{Q^D}\frac{\partial Q^D}{\partial\omega}\right) \tag{3.14}$$

Le premier terme n'est autre que la modification due à la minimisation de l'erreur de prédiction, c'est à dire la règle d'adaptation NPC-2. Le deuxième terme, de signe opposé,

correspond à la maximisation de l'erreur de discrimination. Ces deux termes sont pondérés respectivement par l'amplitude de l'erreur de prédiction et de discrimination. Afin de conserver une influence sur la pondération prédiction/discrimination, nous avons finalement retenu :

$$\frac{\partial Q}{\partial \omega} = \alpha \frac{\partial Q^M}{\partial \omega} - (1-\alpha)\frac{\partial Q^D}{\partial \omega} \tag{3.15}$$

où, pour $\alpha = 1$, on retrouve le modèle purement prédictif NPC-2. La figure 3.8 représente l'évolution du MER durant la phase d'adaptation pour deux valeurs de α. Pour $\alpha = 1$ on

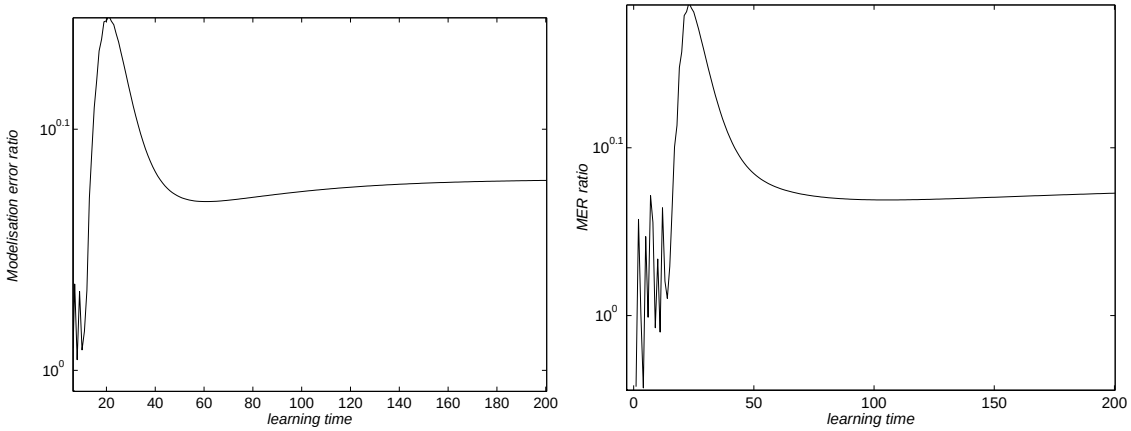


FIG. 3.7 – Evolution du MER lors de la phase d'adaptation pour les phonemes /b/, /d/, /g/ à gauche et /p/, /t/, /k/ à droite (extrait des articles [67],[68]).

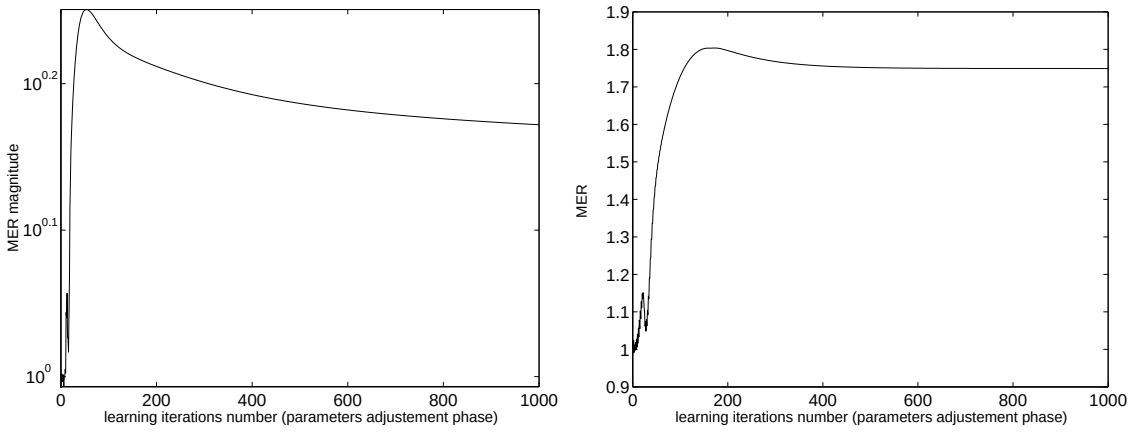


FIG. 3.8 – Evolution du MER lors de la phase d'adaptation sans contrainte ($\alpha = 1$, modèle NPC-2, à gauche), avec contrainte ($\alpha = 0.5$, modèle DFE-NPC, à droite). (tiré des articles [67], [68]).

retrouve l'évolution du MER dans le cas d'un modèle NPC-2 avec un maximum suivi d'une décroissance progressive. Pour $\alpha = 0.5$, on note l'influence de la contrainte discriminante

appliquée sur les poids : le MER atteint une valeur maximum plus tradivement et la décroissance qui s'en suit est beaucoup moins marquée.

La maximisation du MER a permis d'améliorer sensiblement les scores en reconnaissance, ainsi qu'en témoigne le tableau 3.3 où je récapitule les taux de reconnaissance obtenus sur les bases de phonèmes déjà utilisées précédemment. Pour ce qui concerne les phonèmes

phonèmes	LPC	MFCC	PLP	NPC	NPC-2	DFE-NPC
/*aa*/, /*ae*/, /*ey*/, /*ow*/	56.23%	58.25%	57%	61%	62.95%	**65.25%**
/*b*/, /*d*/, /*g*/	57.28%	62%	64%	65%	70.4%	**73%**
/*p*/, /*t*/, /*k*/	62.3%	66.6%	66%	63.33%	65%	**70.3%**

TAB. 3.3 – Scores comparatifs obtenus en reconnaissance à l'aide d'un réseau PMC sur les trois bases {/*aa*/, /*ae*/, /*ey*/, /*ow*/}, {/*b*/, /*d*/, /*g*/} et {/*p*/, /*t*/, /*k*/} pour plusieurs codeurs de la littérature et les trois versions du codeur NPC (tiré de [68] et de [43] pour le codeur PLP).

voisés (les voyelles et les plosives), les paramètres DFE-NPC confortent les résultats déjà obtenus avec les paramètres NPC-2 et décrits au paragraphe 3.3.2. En revanche, en ce qui concerne les phonèmes non voisés (/*p*/, /*t*/, /*k*/), ils améliorent les scores obtenus par les paramètres NPC-2 et MFCC. Le modèle DFE-NPC permet donc, grâce aux contraintes discriminantes, d'obtenir une meilleure extraction de caractéristiques, y compris dans le cas de phonèmes non voisés.

Mohamed Chetouani a fait une étude expérimentale exhaustive du paramètre α, notant qu'il était difficile à déterminer car sa valeur optimale diffère d'un groupe de phonèmes à l'autre. A titre d'exemple, le tableau 3.4 indique la valeur optimale de α obtenue pour les trois bases de phonèmes étudiées et les scores obtenus en classification. Il a ensuite

phonèmes	/*aa*/, /*ae*/, /*ey*/, /*ow*/	/*b*/, /*d*/, /*g*/	/*p*/, /*t*/, /*k*/
α	0.5	0.7	0.5
scores	63.5%	67.66%	66.33%
scores (α adaptatif)	65.25%	70.1%	69.67%

TAB. 3.4 – Valeurs optimales du paramètre α et scores obtenus en généralisation pour trois groupes phonétiques et scores obtenus avec α adaptatif (tiré de [33]).

proposé une loi d'adaptation du paramètre permettant son estimation automatique en cours d'adaptation, le principe étant de privilégier la minimisation de l'erreur de prédiction pour commencer (α proche de 1), puis de diminuer progressivement sa valeur jusqu'à 0. Le même tableau 3.4 indique en dernière ligne les scores alors obtenus tels qu'ils ont été publiés dans [33].
Ce que l'on peut dire du point de vue de la problématique *modélisation/discrimination*, c'est qu'il est nécessaire de favoriser la modélisation dans un premier temps avant que d'introduire des contraintes discriminantes. Cet aspect n'avait été que très peu abordé lors de notre étude des paramètres contraints NPC-C.

3.4.3 Le LMER est équivalent au critère MMI

Publications : [36]

De même qu'Itakura dans [83] montrait que le logarithme du rapport des erreurs de prédiction est équivalent, d'un point de vue statistique, au logarithme du rapport de vraisemblance (le meilleur critère pour le test d'hypothèses), Mohamed Chetouani s'est attaché à montrer [36] que le LMER est équivalent au critère MMI (Maximisation de l'information mutuelle), concluant ainsi que la minimisation de $1/\Gamma$ conduit à l'élaboration de l'extracteur permettant la meilleur classification lorsque la décision est obtenue par maximisation du MER :

$$\hat{c}(l) = \arg\max_{m} \log \frac{\sum_{i,i\neq m} Q_l(\mathbf{a}_i)}{Q_l(\mathbf{a}_m)} \tag{3.16}$$

A noter la différence avec le classifieur proposé au paragraphe 3.3.4 p. 54 (eq. 3.9) et dont on montre qu'il revient à minimiser la distance NPC :

$$\hat{c}(l) = \arg\min_{i} d^{NPC}(i,l) = \arg\min_{i} \left\{ \frac{Q_l(\mathbf{a}_i)}{Q_l(\mathbf{a}_l)} \right\} = \arg\min_{i} Q_l(\mathbf{a}_i) \tag{3.17}$$

L'approximation du MER (eq. 3.11) par :

$$\Gamma \simeq \frac{\sum_{l}\sum_{i} Q_l(\mathbf{a}_i)}{\sum_{l} Q_l(\mathbf{a}_{c(l)})} \tag{3.18}$$

conduit à l'équivalence des deux règles de décision : maximisation du MER et minimisation de la distance d^{NPC}. J'exploiterai ce résultat lors de la définition des paramètres SOM-NPC.
La démonstration de l'équivalence du LMER avec le critère MMI a été obtenue sous l'hypothèse que le conduit vocal pouvait être modélisé par un processus statistique du type $y_k = f(\mathbf{y}_k) + \varepsilon_k$ où ε_k est un bruit blanc gaussien centré, et que les poids solution correspondent à un minimum global de la fonction de coût. Le lecteur trouvera le détail de cette démonstration dans l'article [36], en annexe de ce mémoire p. 171.

3.4.4 Une architecture pour le codage

Publications : [34], [35], **[37]**, [38], [40]

Les expérimentations menées sur les différents extracteurs de caractéristiques l'ont été jusqu'à présent sur des groupes restreints de phonèmes. La principale raison en était la simplicité des modèles ainsi mis en oeuvre. En effet, la mise en œuvre des paramètres NPC nécessite d'adapter l'extracteur (calcul des poids de la couche cachée). Les extracteurs les plus utilisés dans les systèmes de RAP (LPC, MFCC et PLP) ne présentent pas cet inconvénient. Or les temps de calcul nécessaires à l'adaptation sont particulièrement élevés. Réduire le nombre de classes permettait donc d'accélerer sensiblement les expérimentations. Cyril Chavy a cependant eu l'occasion de mettre en oeuvre un extracteur NPC sur l'ensemble des phonèmes de la base NTIMIT. Il a constaté une baisse sensible des scores en classification qu'il a interprété comme étant due à la structure de l'extracteur devenue trop simple relativement à la complexité du problème posé.

Mohamed Chetouani a repris ce travail dans le cadre de sa thèse et fait le même constat. Lorsque l'on ajoute des classes, les méthodes d'extraction traditionnelles permettent d'obtenir des résultats comparables à ceux obtenus lorsque l'on traite un nombre réduit de classes mais les paramètres NPC ne le permettent pas, ainsi que le montre le tableau 3.5. Il

paramètres	LPC	MFCC	PLP	NPC	NPC-2	DFE-NPC
scores (%)	48.5	**56.78**	54.44	47.23	50.55	53.89

TAB. 3.5 – Scores obtenus en classification sur l'ensemble des phonèmes {/*aa*/, /*ae*/, /*ey*/, /*ow*/, /*b*/, /*d*/, /*g*/, /*p*/, /*t*/, /*k*/} pour différentes méthodes d'extraction de caractéristiques (tiré de [29]).

a donc proposé deux architectures, l'une hiérarchique et l'autre coopérative. L'idée étant de décomposer l'ensemble des phonèmes en sous-groupes phonétiques et de dédier un codeur NPC par sous groupe.

L'approche consistant à procéder par décomposition a déjà été expérimentée dans la littérature, mais au niveau de la classification. On peut citer à titre d'exemple l'architecture hiérarchique de classifieurs PMC proposée par Sivadas et al. dans [132]. Un premier classifieur permet de séparer la parole des zones de silence, vient ensuite un deuxième séparant les signaux voisés des signaux non voisés, etc. La conception d'une architecture modulaire requiert de partitionner l'ensemble des phonèmes en groupes de classes ou *macro-classes*. Cette partition ne peut être arbitraire. On regroupe ensemble des catégories phonétiques proches, les critères pouvant être la proximité accoustique ou la proximité des configurations du conduit vocal nécessaires à leur production. Mohamed Chetouani a choisi d'utiliser la décomposition proposée dans l'Alphabet Phonétique International (API) par soucis de rapidité et parce que le problème du regroupement n'était pas au coeur de notre préoccupation (tableau 3.6). Des auteurs comme Jordan et Jacobs ont cependant proposé une

Ω_1	voyelles antérieures	/*ih*/, /*ey*/, /*eh*/, /*ae*/
Ω_2	voyelles centrales	/*ah*/, /*er*/
Ω_3	voyelles postérieures	/*uw*/, /*uh*/, /*ow*/, /*aa*/
Ω_4	diphtongues	/*ay*/, /*ow*/, /*oy*/
Ω_5	semi-voyelles	/*y*/, /*w*/
Ω_6	liquides	/*l*/, *r*
Ω_7	nasales	/*m*/, /*n*/, *ng*
Ω_8	occlusives voisées	/*b*/, /*d*/, /*g*/
Ω_9	occlusives non voisées	/*p*/, /*t*/, /*k*/
Ω_{10}	fricatives voisées	/*v*/, /*z*/, /*jh*/
Ω_{11}	fricatives non voisées	/*f*/, /*s*/, /*ch*/

TAB. 3.6 – Regroupement des classes phonétiques en macro-classes, selon la décomposition proposée par l'API.

structure HME (*mélange hiérarchique d'experts*) [85], [86] permettant de réaliser simultanément le regroupement et l'apprentissage des experts (en l'occurence des réseaux PMC).

Chong et al. [47] sont ainsi arrivés à proposer des macro-classes permettant d'obtenir de meilleurs scores en reconnaissance que ceux obtenus avec la décomposition proposée dans l'Alphabet Phonétique International.

Architecture : le modèle hiérarchique

Dans le modèle proposé par Mohamed Chetouani, l'extraction de caractéristiques d'une trame particulière nécessite sa redirection vers l'un des extracteurs *expert* : celui correspondant à sa macro-classe d'appartenance. Les *macro-classifieurs* établissant cette redirection sont des modèles NPC-2 utilisés en *classifieurs prédictifs* comme indiqué au paragraphe 3.3.4, p.54. Ici, les classes sont les macro-classes d'appartenance des trames et l'organisation des macro-classifieurs se fait selon une architecture en arbre (figure 3.9). Chaque

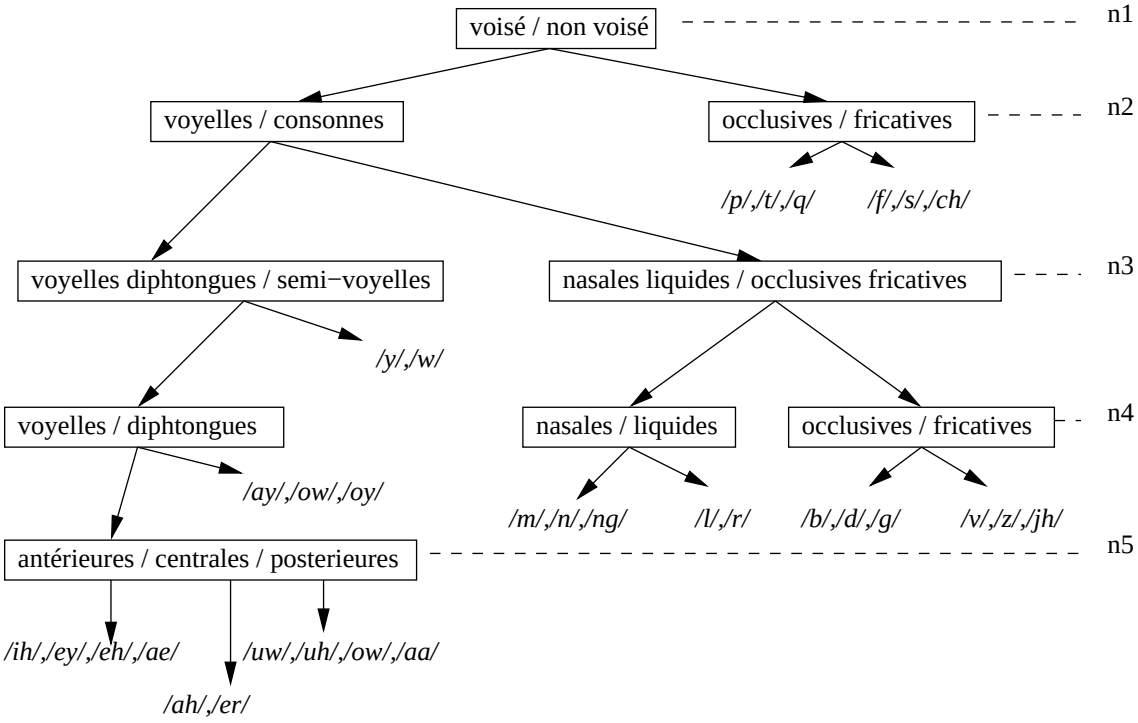


FIG. 3.9 – Architecture en arbre de l'extracteur hiérarchique (tiré des articles [34] et [37]).

extracteur *expert* permet de générer des paramètres DFE-NPC calculés après une première étape d'adaptation sur une base de phonèmes appartenant à la macro-classe correspondante.
Le tableau 3.7 récapitule les scores obtenus en reconnaissance sur l'ensemble des phonèmes, pour chacune des macro-classes. Ces scores sont variables d'une macro-classe à l'autre et le niveau de profondeur de l'arbre où se situe l'extracteur influe nettement sur le résultat. Les voyelles donnent par exemple de moins bons résultats. Les erreurs d'orientation qui peuvent être faites à chaque noeud conduisent à extraire pour les trames concernées des paramètres avec l'extracteur non approprié. Le tableau 3.8 indique les scores de macro-classification obtenus sur chacun des noeuds d'orientation. La comparaison des tableaux 3.7 et 3.8 montre

Ω	macro-classes	phonèmes	scores (%)	profondeur
Ω_1	voyelles antérieures	/*ih*/, /*ey*/, /*eh*/, /*ae*/	39.72	5
Ω_2	voyelles centrales	/*ah*/, /*er*/	41.45	5
Ω_3	voyelles postérieures	/*uw*/, /*uh*/, /*ow*/, /*aa*/	36.08	5
Ω_4	diphtongues	/*ay*/, /*ow*/, /*oy*/	56.64	4
Ω_5	semi-voyelles	/*y*/, /*w*/	64.65	3
Ω_6	liquides	/*l*/, /*r*/	75.46	4
Ω_7	nasales	/*m*/, /*n*/, *ng*	57.61	4
Ω_8	occlusives voisées	/*b*/, /*d*/, /*g*/	72.94	4
Ω_9	occlusives non voisées	/*p*/, /*t*/, /*k*/	88.99	2
Ω_{10}	fricatives voisées	/*v*/, /*z*/, /*jh*/	70.65	4
Ω_{11}	fricatives non voisées	/*f*/, /*s*/, /*ch*/	74.43	2

TAB. 3.7 – Scores obtenus pour chacune des macro-classes traitées (extrait de [37]).

voisés / non voisés	98.74%
voyelles / consonnes	83.3%
occlusives / fricatives (non voisés)	98.33%
voyelles diphtongues / semi-voyelles	82.3%
nasales liquides / occlusives fricatives (voisés)	93.03%
voyelles / diphtongues	88.4%
nasales /liquides	96.14%
occlusives / fricatives (voisés)	95.28%
antérieures / centrales /postérieures	77.3%

TAB. 3.8 – Scores obtenus en macro-classification (tiré de [37]).

que les scores moins bons obtenus pour les voyelles sont liés aux scores obtenus par les macro-classifieurs correspondant, moins bons également. Le taux de reconnaissance global calculé sur l'ensemble des phonèmes traités est de 61.65%. Si l'on compare ce résultat avec les scores obtenus avec les paramètres traditionnels (LPC, MFCC et PLP) dans les mêmes conditions expérimentales (même arbre de décision utilisé et mêmes macro-classifieurs), on constate une amélioration d'environ 10 points pour le DFE-NPC (tableau 3.9).

LPC	MFCC	PLP	DFE-NPC modulaire
48.3%	51.25%	52.3%	**61.65%**

TAB. 3.9 – Taux de reconnaissance sur l'ensemble des phonèmes pour différents paramètres (extrait de [37]).

La structure hiérarchique permet donc de traiter l'ensemble des phonèmes tout en spécialisant les différents extracteurs utilisés sur des sous-groupes de classes phonétiques. Les expérimentations montrent que l'on conserve ainsi les scores initialement obtenus sur des bases comportant un faible nombre de phonèmes de caractéristiques voisines. On retrouve

cependant un défaut inhérent aux arbres de décision, à savoir la démultiplication des erreurs lorsque le niveau de profondeur de l'arbre devient conséquent : une trame mal dirigée est traitée par un extracteur non adapté à cette catégorie de trame et les paramètres extraits ne sont pas optimaux. La qualité de la redirection des trames, et donc les performances des macro-classifieurs, apparaît cruciale pour un tel système.

Architecture : le modèle coopératif

Une autre structure consistant à arranger les macro-classifieurs, non plus selon une organisation arborescente, mais selon une organisation parallèle a conduit à de meilleurs résultats. Les 11 macro-classifieurs traitent identiquement chacune des trames et l'extracteur choisi est celui dont le macro-classifieur présente l'erreur de prédiction la plus faible.
Mohamed Chetouani a cependant choisi de traiter ce type d'architecture en faisant appel à des systèmes de macro-classification discriminants (réseaux PMC) plutôt que prédictifs. Ceci en accord avec la conclusion tirée au paragraphe 3.3.4 p. 54 selon laquelle, même s'il est possible d'utiliser un extracteur NPC-2 comme classifieur, ce dernier s'avère être peu performant. Le modèle qu'il propose consiste donc à faire fonctionner l'ensemble des extracteurs de caractéristiques en parallèle puis à sélectionner le code le plus vraisemblable : chaque jeu de paramètre "candidat" est acheminé en entrée d'un classifieur PMC dont les fonctions de transition sont de type *softmax*. Une probabilité d'appartenance du code à la macro-classe considérée est alors estimée et une décision selon le principe du maximum de vraisemblance effectuée (figure 3.10).

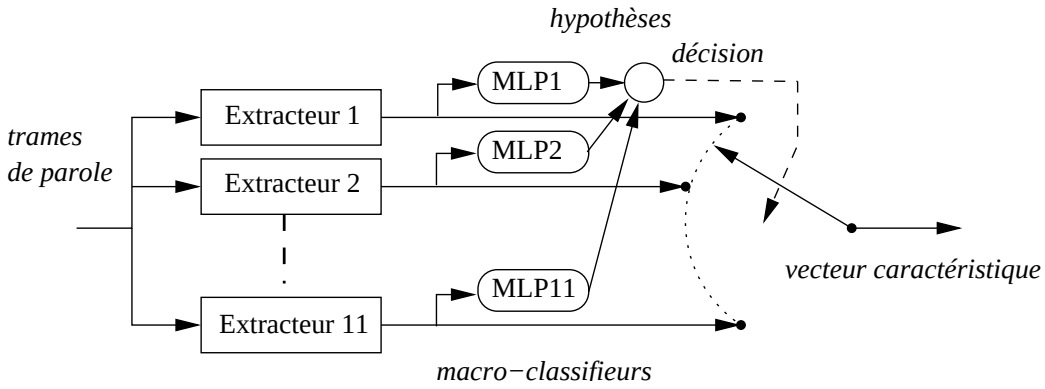


FIG. 3.10 – Architecture coopérative (extrait des articles [35] et [40]).

L'originalité du travail est ici de tirer pleinement partie du principe d'adaptation des extracteurs par maximisation du MER. Il propose en effet une extension consistant à ajouter au numérateur du MER l'erreur de prédiction $\Omega_{\overline{\tau}}$ de la trame courante calculée sur les autres extracteurs :

$$\Gamma^{\Omega_\tau} = \frac{\sum\limits_{i,i\neq c(l)} Q_l^{\Omega_\tau}(\mathbf{a}_i) + Q_l^{\Omega_{\overline{\tau}}}}{Q_l^{\Omega_\tau}(\mathbf{a}_{c(l)})} \tag{3.19}$$

où Γ^{Ω_τ} désigne le MER calculé sur l'extrateur Ω_τ. Un principe de coopération est ainsi introduit entre les différents extracteurs de caractéristiques (une coopération *inter-macro-*

classes) permettant de renforcer la discrimination au sein d'une même macro-classe. Les

$\mathbf{\Omega}$	macro-classes	phonèmes	scores (%) macro-classes	scores (%) classes
Ω_1	voyelles antérieures	/*ih*/, /*ey*/, /*eh*/, /*ae*/	82.85	55.69
Ω_2	voyelles centrales	/*ah*/, /*er*/	96.33	90.7
Ω_3	voyelles postérieures	/*uw*/, /*uh*/, /*ow*/, /*aa*/	78.33	45.23
Ω_4	diphtongues	/*ay*/, /*ow*/, /*oy*/	83.54	58.37
Ω_5	semi-voyelles	/*y*/, /*w*/	89.66	84.28
Ω_6	liquides	/*l*/, /*r*/	89.82	75.14
Ω_7	nasales	/*m*/, /*n*/, *ng*	80.42	54.05
Ω_8	occlusives voisées	/*b*/, /*d*/, /*g*/	82.85	65.45
Ω_9	occlusives non voisées	/*p*/, /*t*/, /*k*/	78.21	58.21
Ω_{10}	fricatives voisées	/*v*/, /*z*/, /*jh*/	78.17	61.57
Ω_{11}	fricatives non voisées	/*f*/, /*s*/, /*ch*/	77.46	59.89
$\mathbf{\Omega}$	moyenne	tous les phonèmes	**83.42**	**64.41**

TAB. 3.10 – Scores obtenus en macro-classification et classification (extrait de [40]).

performances globales de l'architecture coopérative sont de 64% comme l'indique le tableau 3.10, ce qui représente une amélioration de près de 12 points relativement aux autres méthodes de codage (cf. tableau 3.9 p.63) et de 3 points relativement à l'architecture hiérarchique présentée plus haut. Le concept d'architecture parallèle permet de ne pas handicaper un groupe de phonèmes par rapport à un autre, du simple fait de sa position dans l'arbre de décision.

Les modèles que j'ai présentés dans cette section représentent un effort important de recherche dont l'objectif était de valider la modélisation NPC sur l'ensemble des catégories phonétiques. Deux structures ont été proposées par Mohamed Chetouani dans le cadre de ses travaux de thèse. L'une de type hiérarchique à arbre de décision et l'autre de type parallèle avec coopération. La deuxième architecture a montré sa supériorité d'une part, et une certaine élégance d'autre part puisque toujours fondée sur la maximisation du critère MER étendue de façon très naturelle à la notion de macro-classes.

3.5 Le modèle LVQ-NPC ou la coopération directe

Publications : [41], [39], [29], **[31]**, **soumise : [42]**

J'aborde ici l'avant dernière version du modèle NPC, appelée LVQ-NPC ou la *coopération directe extracteur-classifieur*. La cinquième et dernière version SOM-NPC sera présentée quant à elle au chapitre suivant. Je rappelle que lorsque l'on met en œuvre la contrainte MER (modèle DFE-NPC), on adapte l'extraction de caractéristiques à une tâche de classification donnée, indépendamment du classifieur utilisé en amont. Le codage est alors optimal au sens de la minimisation du critère MMI, seulement dans le cadre restreint où la classification est opérée par minimisation de l'erreur de prédiction (eq. 3.17), ou mieux, par maximisation du MER (eq. 3.16).

Avec le modèle LVQ-NPC, on cherche à rendre optimal le codage relativement à l'erreur de classification réalisée par un classifieur "extérieur". En l'occurence un classifieur LVQ qui doit être intégré dans le processus d'adaptation, comme le montre la figure 3.11.

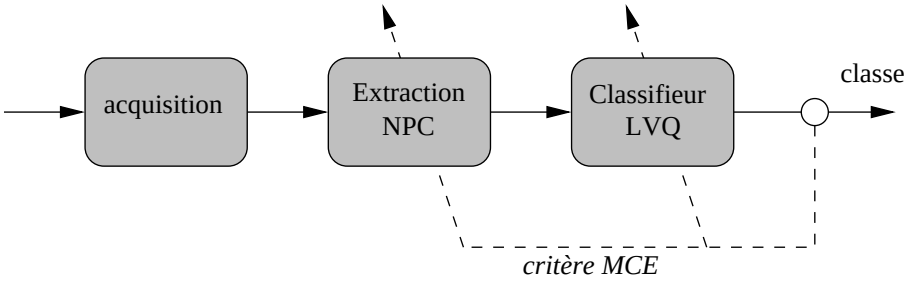


FIG. 3.11 – Schéma type de l'extraction de caractéristiques LVQ-NPC.

Comme je l'ai écrit plus haut (§3.4, p.55 et §3.1, p. 46), Biem et Katagiri [8] ont proposé une approche semblable où ils utilisent un banc de filtres comme extracteur de caractéristiques et un réseau PMC comme classifieur. Cette approche pose une difficulté liée à la complexité très différente des deux systèmes. Lors de l'adaptation, l'évolution des paramètres de l'extracteur est beaucoup plus lente que celle du classifieur. Il est même possible d'atteindre un minimum local sans que les paramètres de l'extracteur ne soient réellement modifiés. De la Torre et al. [50] ont proposé de remédier à ce problème en proposant un apprentissage indépendant de l'extracteur et du classifieur, et dont le schéma rappel celui du DFE-NPC (figure 3.12). Le classifieur est cependant beaucoup plus simple et permet

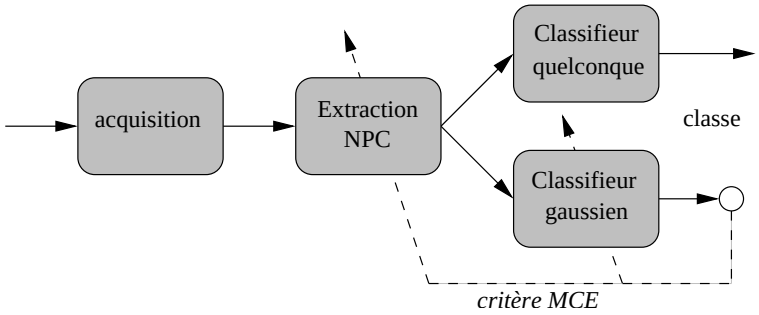


FIG. 3.12 – Schéma proposé par De la Torre et al. dans [50].

ainsi de contraindre l'évolution des paramètres de l'extracteur de caractéristiques. A l'issue de l'adaptation de l'extracteur, un classifieur plus sophistiqué (réseau PMC, GMM ou HMM) est utilisé en lieu et place du classifieur gaussien initial.
Le choix d'un classifieur à prototypes de type LVQ a été guidé par la nécessité d'avoir un classifieur de complexité équivalente à celle de l'extracteur de caractéristiques et par la simplicité de mise en oeuvre de la coopération extracteur-classifieur. Katagiri [91] et MacDermott [117] ont déjà proposé une méthode d'apprentissage d'un modèle LVQ par descente de gradient GPD. En définissant une fonction de perte adéquate, ils ont montré

que l'on obtenait ainsi une règle d'apprentissage très proche de la règle LVQ2. Mohamed Chetouani a donc repris les travaux de McDermott pour proposer un ensemble extracteur-classifieur constitué d'un modèle LVQ avec règle d'apprentissage MCE/GPD, appelé pour l'occasion classifieur MCE-LVQ, et un extracteur de caractéristiques NPC. L'extracteur est un modèle à deux couches cachées et autant de cellules de sortie que de prototypes représentés dans le classifieur LVQ. Les poids de la couche de sortie sont donc appris par minimisation de l'erreur de prédiction, mais également sous les contraintes de modification des prototypes, imposées par l'apprentissage du classifieur LVQ.

3.5.1 le principe

La méthode DFE consiste à considérer l'extracteur de caractéristiques (NPC) et le classifieur (LVQ) comme un seul module $\Phi = (\Omega, \mathbf{m})$ où Ω représente l'ensemble des paramètres de l'extracteur NPC et $\mathbf{m}$ l'ensemble des paramètres du classifieur LVQ. Mettre en œuvre la méthode consiste à déterminer les trois éléments que sont 1) les fonctions discriminantes, 2) la mesure de mauvaise classification (*misclassification measure*) et 3) la fonction de perte.

Fonctions discriminantes. Pour un problème à C classes, les C fonctions discriminantes reflètent chacune dans quelle proportion un échantillon inconnu appartient à la classe correspondante. C'est le choix de la structure du classifieur qui détermine la nature des fonctions discriminantes.
Pour un classifieur LVQ, la fonction discriminante g_i associée à la classe i pourra être la distance au vecteur représentant le plus proche $\mathbf{m}_{ij}$ associé à cette même catégorie, ce qui correspond à la règle du LVQ :

$$g_i(\mathbf{a}, \Lambda) = \min_{j=1,\ldots,N_i} d(\mathbf{a}, \mathbf{m}_{ij}) \tag{3.20}$$

où $\mathbf{a}$ est le vecteur de code (poids de la couche de sortie de l'extracteur) associé à la trame de parole traitée, où $\mathbf{m}_{ij}$ est le j^{eme} vecteur de référence (adaptable) associé à la catégorie c_i, et où N_i est le nombre de vecteurs de référence de cette catégorie.

Mesure de mauvaise classification. Idéalement, le critère global doit incorporer des informations relatives au processus de comparaison inter-catégories. La mesure de mauvaise classification suivante, appelée *critère MCE*, a été proposée pour dépasser les limitations d'une précédente mesure (non continue) proposée par Amari [3], et dont Katagiri et Dermott proposent une version particulière :

$$d_i(\mathbf{a}, \Lambda) = -g_i(\mathbf{a}, \Lambda) + \overline{g}_i(\mathbf{a}, \Lambda) \tag{3.21}$$

où $\overline{g}_i$ désigne la fonction discriminante associée à la catégorie incorrecte la plus proche. Le signe de cette mesure reflète la qualité de la classification : négatif lorsque la classification est mauvaise, positif dans le cas contraire. Une forme plus générale de cette mesure se rapproche beaucoup du critère MMI dont j'ai parlé plus haut à propos du DFE-NPC (§3.4.3, p. 60). Cette similarité est discutée en détail dans [117].

Fonction de perte. Une fonction de perte doit ensuite être définie en sachant qu'elle doit s'approcher de la fonction idéale *zero-un* qui associe la valeur 0 lorsqu'il n'y a pas d'erreur de classification et la valeur 1 dans le cas contraire. Dans le cadre MCE/GPD, cette fonction de perte est approximée par une fonction continue permettant de mettre en oeuvre une descente de gradient :

$$L(d) = \frac{1}{1 + e^{-\zeta d}} \tag{3.22}$$

Pour une valeur élevée de ζ, elle se rapproche de la fonction binaire idéale, tout en restant continue.

Lois d'apprentissage et d'adaptation. L'algorithme de descente du gradient conduit aux équations de modification suivantes des vecteurs représentants $\mathbf{m}$ du classifieur LVQ et des poids-codes $\mathbf{a}$ de l'extracteur de caractéristiques pour une trame l :

$$\mathbf{m}_i = \mathbf{m}_i + 2\alpha(t)L(l,\Lambda)(1 - L(l,\Lambda))(\mathbf{a}_l - \mathbf{m}_i) \tag{3.23}$$

$$\mathbf{m}_j = \mathbf{m}_i - 2\alpha(t)L(l,\Lambda)(1 - L(l,\Lambda))(\mathbf{a}_l - \mathbf{m}_j) \tag{3.24}$$

$$\Delta\mathbf{a}_l = -2\beta(t)L(l,\Lambda)(1 - L(l,\Lambda))(\mathbf{m}_i - \mathbf{m}_j) \tag{3.25}$$

On note que les règles de modification des vecteurs représentants du classifieur sont proches de la règle d'apprentissage LVQ2, à ceci près que dans le cas de l'algorithme LVQ2, le deuxième représentant $\mathbf{m}_j$ doit appartenir à la bonne catégorie. Par ailleurs, l'adaptation $\Delta\mathbf{a}_l$ des poids-code est proportionnelle à la différence $\mathbf{m}_i - \mathbf{m}_j$ des deux représentants les plus proches mais de catégories différentes, montrant ainsi que la contrainte du classifieur LVQ sur l'extracteur de caractéristique se traduit par une augmentation explicite de la variance inter-classes.
Bien entendu, la règle de modification complète des poids-codes de l'extracteur doit prendre en compte également la minimisation de l'erreur de prédiction. On reprend donc la loi globale de modification permettant de pondérer la modélisation et la discrimination :

$$\Delta\mathbf{a}_l = \theta\Delta\mathbf{a}_l^M + (1 - \theta)\Delta\mathbf{a}_l^{MCE} \tag{3.26}$$

où $\Delta\mathbf{a}_l^M$ est obtenue par minimisation de l'erreur de prédiction selon l'algorithme NPC déjà exposé. La modification des poids des couches cachées suit le même principe que pour les modèles précédents NPC, NPC-2 et DFE-NPC : les couches cachées (2 dans le cas du modèle étudié) sont estimées par rétropropagation de l'erreur de prédiction et de discrimination respectivement, dans les proportions θ et $1 - \theta$.

3.5.2 Evaluation

Le modèle a été testé expérimentalement sur des données proches de celles utilisées dans les expérimentations précédentes. Extraits de la base NTIMIT de signaux téléphoniques dans des proportions dont le lecteur trouvera le détail dans l'article [42] inséré en annexe de ce mémoire, trois sous groupes de phonèmes ont été sélectionnés ($\{/ih/,/ey/,/eh/,/ae/\}$, $\{/b/,/d/,/g/\}$ et $\{/p/,/t/,/k/\}$). La table 3.11 indique l'ensemble des scores comparés obtenus pour différentes paramètres (LPC, MFCC, PLP, NPC et LVQ-NPC), et ce à l'aide de trois classifieurs distincts. Le classifieur par mélange de gaussiennes (GMM) a été entraîné

(%)	LPC	MFCC	PLP	NPC	LVQ-NPC
	/ih/,/ey/,/eh/,/ae/				
GMM	35.22	48.12	45.12	40.03	**54.81**
LVQ1	36.53	43.21	40.33	46.66	53.12
PMC	41.21	44.23	43.22	47.31	52.23
	/b/,/d/,/g/				
GMM	54.13	59.23	57.21	62.24	**66.33**
LVQ1	50.22	57.82	57.03	60.33	64.55
PMC	55.31	58.12	57.63	61.86	63.89
	/p/,/t/,/k/				
GMM	44.10	51.45	46.98	49.36	**53.22**
LVQ1	43.03	50.12	46.22	48.07	50.12
PMC	45.12	50.56	45.88	48.66	50.98

TAB. 3.11 – Scores obtenus en classification par trois classifieurs (GMM, LVQ1 et PMC) sur les trois familles de phonèmes (voyelles, occlusives voisées et occlusives non voisées) et tirés de [42].

à l'aide de l'algorithme EM (Estimation-Maximisation) avec l'hypothèse de matrices de covariance diagonales (décision selon le critère du maximum de vraisemblance). Le classifieur LVQ (algorithme LVQ-1) a été initialisé à l'aide de l'algorithme des k-moyennes ($k = 16$). il est différent de celui utilisé lors de l'adaptation de l'extracteur LVQ-NPC. Enfin, le classifieur neuronal PMC (perceptron multicouches) comportait une couche cachée de 10 neurones et a été entraîné à l'aide de l'algorithme de Levenberg-Marquardt.

3.5.3 Interprétation

Parmi tous les paramètres accoustiques testés, les paramètres NPC tiennent une place spéciale. Ils ont en effet été générés par la première version de l'extracteur. Ce dernier n'incorpore aucune information discriminante explicite, ce qui permet donc d'analyser l'apport de la discrimination pendant la phase d'adaptation du LVQ-NPC. Concernant les phonèmes voisés (voyelles et plosives), le tableau 3.11 montre que les paramètres NPC et LVQ-NPC permettent une amélioration des scores quelque soient les classifieurs utilisés. Ainsi, l'introduction d'une modélisation non linéaire (NPC) et l'apport d'informations discriminantes (LVQ-NPC) permet un gain de 6 points relativement au codage MFCC. A noter une cohérence certaine des résultats lorsque l'on passe d'un classifieur à l'autre, les meilleurs scores étant obtenus par les GMM.
L'étude des occlusives non voisées (phonèmes /p/,/t/,/k/) est intéressante car ces phonèmes pénalisent les méthodes prédictives, même dans le cas des modèles non linéaires comme le NPC (score de 49.36%, j'en donne les raisons plus haut, cf. §3.4.2, p.57). Ce sont les paramètres MFCC qui permettent en effet d'obtenir les meilleurs résultats (51.45%), sauf lorsque l'on prend en compte des informations discriminantes comme avec le LVQ-NPC (53.22%), quelque soit le classifieur utilisé. Ceci corrobore les résultats déjà obtenus dans ce sens avec les paramètres DFE-NPC.

Un paramètre libre important du modèle LVQ-NPC est le nombre de prototypes $\mathbf{m}_i$ par classe. Nous avons réalisé plusieurs adaptations avec un nombre différent de prototypes pour chacune d'entre elles. Lorsque le nombre de prototypes par classe augmente, on remarque une amélioration des performances en apprentissage, mais simultanément une dégradation en généralisation, ainsi que le montre la figure 3.13. Cependant, entre 25 et 50 prototypes, on ne note pas de réelle différence. Pour des raisons de temps de calcul, le nombre de 25 prototypes a donc été retenu dans toutes les expérimentations.

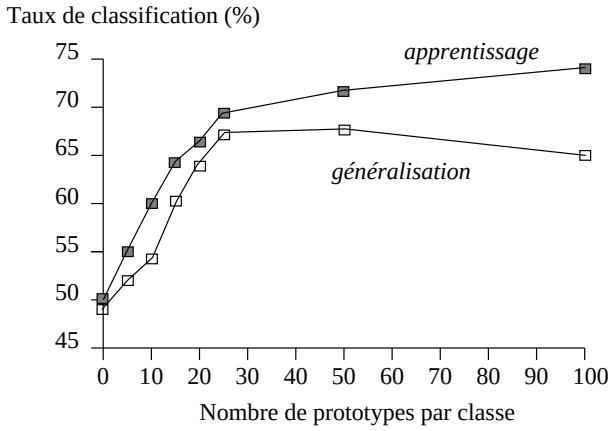


FIG. 3.13 – Scores obtenus en classification en fonction du nombre de prototypes par classe en apprentissage et en généralisation (extrait de [29]).

3.5.4 Question sur la dimension

Beaucoup d'auteurs ont travaillé sur la transformation de caractéristiques (linéaire mais également non linéaire) dans le but de diminuer la dimension des vecteurs caractéristiques en produisant des coefficients discriminants et décorrélés. Or, comme l'affirme Wang dans [139], un extracteur de caractéristiques conçu au mieux ne devrait pas nécessiter une étape de transformation de caractéristiques.
Dans une série de simulations, Mohamed Chetouani s'est placé dans des conditions plus proches de celles employées dans les systèmes de RAP en considérant des vecteurs caractéristiques comportant les coefficients statiques (12) mais également les coefficients dérivés Δ (12) et $\Delta\Delta$ (12). Je synthétise sur la figure 3.14 les résultats obtenus après une analyse discriminante non linéaire (NLDA) effectuée sur ces vecteurs de 36 paramètres (LPC, MFCC, PLP, LVQ-NPC). Un résultat remarquable est l'augmentation significative des scores que l'on peut obtenir en considérant les paramètres dynamiques Δ et $\Delta\Delta$. Ceci s'observe quelque soient les phonèmes traités et quelque soient les paramètres utilisés. Mais ce qui nous intéresse ici est le comportement des scores en dimension élevée.
Il apparaît clairement que l'on peut diminuer cette dimension de 36 à 25 sans affecter les scores pour les paramètres LPC, MFCC et PLP. Le comportement des paramètres LVQ-NPC est quelque peu différent puisque lorsque l'on diminue la dimension du vecteur des caractéristiques, les performances diminuent également. Les coefficients LVQ-NPC modélisent donc des données dont la dimension intrinsèque apparaît légèrement supérieure à

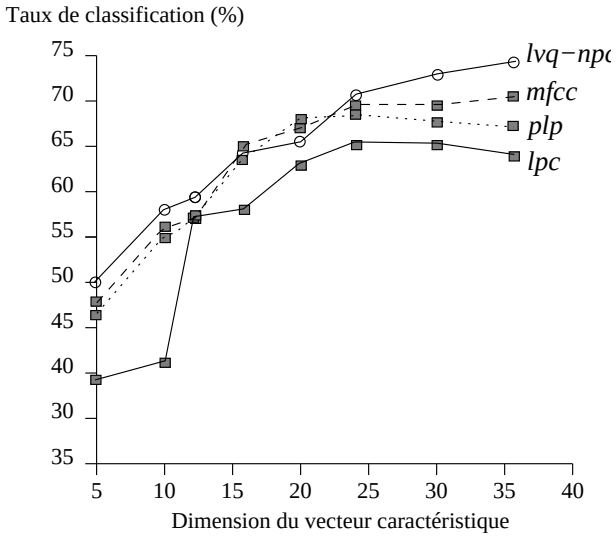


FIG. 3.14 – Scores obtenus en reconnaissance sur les plosives voisées avec les paramètres LPC, MFCC, PLP et LVQ-NPC puis réduction de dimension par analyse NLDA (extrait de [41]).

celle donnée par les autres systèmes d'extraction, ce qui peut être une illustration de la prise en compte de caractéristiques supplémentaires, en l'occurence non linéaires.

3.6 Conclusion

J'ai présenté dans ce chapitre les différentes extensions du modèle NPC qui, des paramètres NPC-2 aux paramètres LVQ-NPC, nous ont conduit à affirmer que la modélisation non linéaire et l'utilisation d'informations discriminantes permet de mieux appréhender la complexité des données et d'obtenir en conséquence une amélioration significative des scores en reconnaissance de phonèmes. Les contraintes NPC-C n'ont pas donné les résultats attendus. La raison principale semble en être que la distribution des paramètres n'est pas gaussienne. Les contraintes orientées, non pas sur la nature des distributions, mais sur les objectifs de la classification (modèles DFE-NPC et LVQ-NPC), donnent au contraire des résultats intéressants. Le modèle LVQ-NPC est une version très aboutie de l'extracteur de caractéristiques NPC en ce qu'il dirige la modélisation des phonèmes en incorporant des informations discriminantes directement issues du classifieur. L'opération d'adaptation s'effectue simultanément avec celle de l'apprentissage du classifieur. Un formalisme commun a été utilisé pour cela : l'apprentissage MCE/GPD. Enfin, les extracteurs NPC-2 et DFE-NPC permettent de réaliser une classification directe des trames, sans faire appel à un classifieur externe. L'algorithme de décision consistant à choisir le modèle minimisant l'erreur de prédiction est, à une approximation près, équivalent au critère MMI. Cette voie consistant à réduire les deux étapes d'analyse et de classification en une seule étape est en cours d'investigation actuellement et fait l'objet d'une section du chapitre «*Synthèse et perspectives*» suivant.

.

Chapitre 4

Synthèse et perspectives

Agencé en trois parties, ce chapitre débute par une analyse concernant quelques aspects plus théoriques des paramètres NPC et rédigée en forme de synthèse du modèle. J'y traite essentiellement de la validation du modèle NPC que j'appuie sur les travaux de la communauté connexionniste concernant les propriétés d'approximation universelle des réseaux PMC à partir du théorème de Kolmogorov.
Je décris ensuite des travaux de recherche qui sont actuellement en cours et concernent la classification par modélisation prédictive (le modèle SOM-NPC). Je les présente dans ce mémoire car des résultats intéressants ont déjà été obtenus et acceptés pour publication. Par ailleurs, ils constituent un aboutissement de l'axe de recherche suivi pendant près de dix ans, mais également une ouverture vers de nouveaux champs de recherche à venir que j'expose ensuite.

4.1 Le théorème de Kolmogorov : vers une validation théorique des paramètres NPC

Publications : [68], soumises : [62]

4.1.1 Des réseaux à poids partagés

Les paramètres NPC, dans leur première version, nécessitent l'apprentissage d'un réseau PMC dont la caractéristique est de comporter autant de cellules de sortie que de trames dans la base d'apprentissage et une seule couche cachée. La structure mise en œuvre est équivalente à L réseaux parallèles apprenant chacun une trame parmi L et dont la première couche est une couche dite à *poids partagés* au sens des réseaux TDNN proposés par Lang et Waibel [138], [103]. Le modèle NPC-2 reprend la même structure, à ceci près que l'on ne considère plus qu'une cellule de sortie par classe de signaux.
Le modèle DFE-NPC n'est plus assimilable à un réseau PMC à poids partagés car pour une trame quelconque de la base d'apprentissage, toutes les cellules de sortie sont adaptées. La cellule qui représente la classe de la trame courante minimise l'erreur de prédiction sur cette trame, tandis que les autres cellules, représentant des classes différentes, maximisent l'erreur de prédiction sur cette même trame. Ainsi, le réseau DFE-NPC se rapproche plus par sa structure d'un réseau PMC classifieur (une sortie par classe) à ceci près que l'on

n'établit pas une décision de type Winner Takes All (WTA) en sortie mais une régression sur des données.
Une variante au DFE-NPC consiste à représenter chacune des classes par plusieurs cellules de sortie plutôt qu'une seule. Une telle structure permet de mieux représenter les signaux présentant une grande variabilité comme les signaux de parole. Le modèle SOM-NPC que je présente au paragraphe 4.2 reprend cette idée.

4.1.2 Première validation théorique proposée

Une question posée depuis le début de mes travaux de recherche sur le NPC, et restée longtemps sans réponse, était de savoir ce qui d'un point de vue théorique justifiait d'utiliser seulement les poids de la couche de sortie comme paramètres représentatifs. Autrement dit, était-il possible de valider théoriquement le fait que ces poids soient discriminants, donc candidats à devenir vecteurs caractéristiques, et que les poids des couches cachées ne le soient pas. J'ai proposé dans [68] des éléments de réponse que j'étoffe dans [62] mais nécessitant des hypothèses peu réalistes concernant le processus de production de la parole.

Le formalisme

Selon le formalisme déjà présenté dans les chapitres précédents, un prédicteur NPC modélise un ensemble de fonctions : autant de fonctions de $\Re^\lambda$ dans $\Re$ qu'il y a de cellules de sorties, donc de trames. Cette formalisation repose ainsi sur le principe qu'une trame de signal quelconque l est produite par un processus du type :

$$y_k = f_l(\mathbf{y}_k) + \varepsilon_k \tag{4.1}$$

Dans le cas idéal, ε_k désigne un bruit blanc gaussien. Mais nous avons vu qu'une telle supposition n'est, en général, pas réaliste avec le signal de parole. Lorsque la trame est non voisée, il existe f_l linéaire satisfaisant (4.1). En revanche, lorsque la trame est voisée, nous avons vu que f_l non linéaire est une meilleure approximation, ce qui justifie notre approche NPC. Nous ne pouvons considérer pour autant que ε_k est blanche. Il reste en effet une erreur résiduelle quasi-périodique, souvent idéalisée par un peigne de Dirac à la fréquence du pitch. Ce signal résiduel correspond à la source gothique dans le modèle source-filtre du conduit vocal. La raison de sa présence est due au fait que (4.1) est une modélisation prédictive *court-terme* du signal de parole. Dans la démonstration qui suit, nous négligeons cette composante en assimilant ε_k à un bruit blanc. Nous considérons ainsi que toute l'information concernant le processus générateur est présente dans f_l, ce qui est une approximation de la réalité. On y trouve en particulier des informations concernant le locuteur [125], même si elles sont, la plupart du temps il est vrai, ignorées par les analyseurs (MFCC notamment).

Signaux non bruités

Nous nous plaçons dans le cadre théorique de l'approximation de fonctions et considérons dans un premier temps des données d'apprentissage non bruitées. La modélisation du processus générateur d'un signal $y_k = f(y_{k-1}, \dots y_{k-\lambda}) + \varepsilon_k$ est vue comme un problème d'approximation d'une fonction f de $[-1,+1]^\lambda$ dans $[-1,+1]$ qui à $\mathbf{y}_k$ associe $y_k = f(\mathbf{y}_k)$ et pour lequel on suppose dans un premier temps $\varepsilon_k = 0$.

Soient deux phonèmes de classes différentes i et j, $j \neq i$ et un ensemble d'apprentissage $\mathcal{D} = \mathcal{D}_i \cup \mathcal{D}_j$ constitué de trames de classe i et j. On a pour chacun des couples d'apprentissage $(y_k, \mathbf{y}_k)$:

$$\begin{cases} (\mathbf{y}_k, y_k) \in \mathcal{D}_i \Longrightarrow y_k = f_i(\mathbf{y}_k) \\ (\mathbf{y}_k, y_k) \in \mathcal{D}_j \Longrightarrow y_k = f_j(\mathbf{y}_k) \end{cases} \tag{4.2}$$

construire un analyseur NPC, $F_{\Omega,\mathbf{a}_i,\mathbf{a}_j}$, minimisant le coût quadratique calculé sur $\mathcal{D}$ conduit à obtenir deux opérateurs $F_{\Omega,\mathbf{a}_i}$ $F_{\Omega,\mathbf{a}_j}$, vérifiant sur les données d'apprentissage [16] :

$$\begin{cases} (\mathbf{y}_k, y_k) \in \mathcal{D}_i \Longrightarrow y_k = F_{\Omega,\mathbf{a}_i}(\mathbf{y}_k) \\ (\mathbf{y}_k, y_k) \in \mathcal{D}_j \Longrightarrow y_k = F_{\Omega,\mathbf{a}_j}(\mathbf{y}_k) \end{cases} \tag{4.3}$$

Sur les couples de $\mathcal{D}_i$ et de $\mathcal{D}_j$ uniquement, on obtient l'égalité des prédicteurs avec les processus :

$$\begin{cases} H_{\mathbf{a}_i} \circ G_\Omega = f_i \\ H_{\mathbf{a}_j} \circ G_\Omega = f_j \end{cases} \tag{4.4}$$

où $H_{\mathbf{a}_i}$ et $H_{\mathbf{a}_j}$ représentent l'opérateur réalisé par les deux couches de sortie du réseau et G_Ω l'opérateur réalisé par la couche cachée. L'opérateur global F réalisé par le réseau est donné par la composition des fonctions : $F_{\Omega,\mathbf{a}_i} = H_{\mathbf{a}_i} \circ G_\Omega$ et $F_{\Omega,\mathbf{a}_j} = H_{\mathbf{a}_j} \circ G_\Omega$. La validation proposée (le lecteur en trouvera le détail dans [62], soumise) consiste à montrer que :

$$i \neq j \Longrightarrow H_{\mathbf{a}_i} \neq H_{\mathbf{a}_j} \Longrightarrow \mathbf{a}_i \neq \mathbf{a_j}. \tag{4.5}$$

On montre que cette condition de différenciation des poids de la couche de sortie est d'autant mieux obtenue que la fonction de coût minimisée pendant l'apprentissage est du type MER^{-1}.

Bruit gaussien

Les résultats obtenus précédemment peuvent être généralisés au cas d'échantillons $(\mathbf{y}_k, y_k)$ bruités additivement par un bruit blanc gaussien centré. Il existe en effet un résultat important concernant l'interprétation des sorties d'un réseau entraîné par minimisation d'une fonction erreur quadratique : la sortie $F(\mathbf{y})$ approxime l'espérance conditionnelle de la donnée cible y : $F(\mathbf{y}) =< y|\mathbf{y} >$ [15] [16].
En supposant que les données cibles sont générées à partir d'une fonction déterministe $f(\mathbf{y})$ avec un bruit additif gaussien centré ε, alors les données cibles sont données par $y = f(\mathbf{y}) + \varepsilon$. La sortie du réseau, lorsque le minimum de la fonction de coût est atteint, devient $F(\mathbf{y}) =< y|\mathbf{y} >=< f(\mathbf{y}) + \varepsilon|\mathbf{y} >= f(\mathbf{y}) + < \varepsilon >= f(\mathbf{y})$ puisque ε est centré. Ce résultat est généralisable au cas des données d'entrées $\mathbf{y}_k$ également perturbées par un bruit gaussien centré. Webb [140] a en effet montré que dans ce cas, la solution optimale (minimum de la fonction de coût) est à nouveau donnée par l'espérance conditionnelle de la donnée cible $< y_k|\mathbf{y}_k >$.

4.1.3 Il existe des couches cachées «*universelles*»

Des résultats issus de la théorie de l'approximation de fonctions permettent d'étayer les travaux que je viens de présenter au paragraphe précédent. En effet, en 1957, Kolmogorov

prouve avec son théorème de superposition (réfutation du 13ème problème de Hilbert) que toute fonction continue f de $\Re^\lambda$ dans $\Re$ peut être représentée par une somme de fonctions continues de $\Re$ dans $\Re$:

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2\lambda+1} \phi_q(\sum_{p=1}^{\lambda} \psi_{pq}(x_p)) \tag{4.6}$$

Hecht-Nielsen [73] a montré que l'on pouvait interpréter cette superposition de Kolmogorov comme un perceptron multicouches avec une couche cachée calculant les variables :

$$y_q = \sum_{p=1}^{\lambda} \psi_{pq}(x_p) \tag{4.7}$$

Cette proposition a été critiquée par Poggio et Girosi [71] pour plusieurs raisons, l'une étant qu'appliquer rigoureusement le théorème de Kolmogorov requérait l'apprentissage de fonctions d'activation non paramétriques, certes continues, mais pouvant être fortement irrégulières (les fonctions de transition implémentées dans les réseaux PMC sont des fonctions sigmoïdes strictement croissantes). Depuis, d'autres résultats exploitant des théorèmes de l'analyse fonctionnelle ont permis de démontrer la propriété d'approximateur universelle des perceptrons multicouches [81]. Mais ce qui nous intéresse plus particulièrement dans les travaux de Hecht-Nielsen, c'est que dans sa démonstration, les couches cachées sont estimées indépendamment de la fonction f approximée, de sorte que cette partie du réseau de neurones est apprise une fois pour toute pour une valeur donnée de λ. Kurkova [102], Sprecher et Katsuura [94] et d'autres encore, ont ainsi montré qu'il existait des couches cachées *universelles*, indépendantes même de λ. Le modèle d'extraction des paramètres NPC est construit selon ce principe : seuls les poids de la couche de sortie dépendent de la fonction f approximée.

Tous ces arguments nécessitent d'être développés plus en détail. Il serait intéressant notamment d'étudier la conservation de cette propriété en fonction de la structure du réseau (nombre de couches et nombre de cellules). Par exemple, les différents modèles NPC que j'ai présentés dans ce mémoire n'extraient pas correctement les caractéristiques des signaux lorsque le nombre de classes augmente mais que la structure reste inchangée (voir à ce sujet le paragraphe §3.4.4, p. 60 concernant les structures modulaires).
Des auteurs [123] ont déjà tiré parti du théorème de Kolmogorov pour réaliser des filtres non-linéaires. Le modèle SOM-NPC que je présente au paragraphe suivant illustre expérimentalement les résultats de Hecht-Nielsen, Kurkova, Sprecher et Katsuura. Il permet de montrer en effet que les poids des cellules de sortie portent suffisamment d'informations concernant les fonctions modélisées pour s'organiser de façon non supervisée en zones quasi-homogènes de classes identiques. Il intègre l'extraction de caractéristiques et la classification au sein d'un seul et unique module et doit son nom de SOM-NPC au fait qu'il est fondé sur l'utilisation des cartes auto-organisantes de Kohonen.

4.2 Finalisation : le modèle SOM-NPC

Encadrement : Farid Feiz (Master SDI 2005)
Publications : [60], [63], **[64]**

L'algorithme SOM-NPC (Self-Organizing Map) est une nouvelle méthode d'adaptation *non supervisée* du modèle NPC. L'extracteur SOM-NPC est utilisable également, au même titre que les modèles NPC-2 et DFE-NPC, en classifieur de caractéristiques (voir à ce sujet le paragraphe 3.3.4, p. 54). Il permet néanmoins d'obtenir des scores en reconnaissance plus satisfaisants et peut donc être considéré comme un système pour lequel l'extraction et la classification de caractéristiques font partie d'un seul et même module (figure 4.1). Les

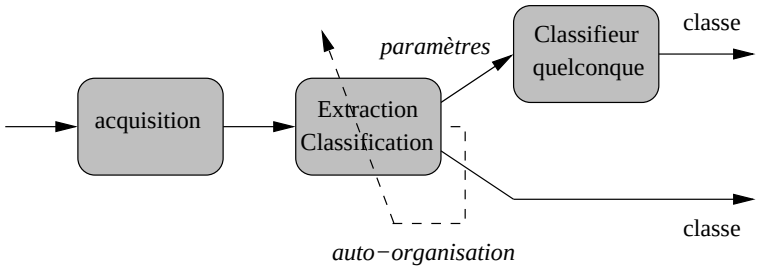


FIG. 4.1 – Schéma d'extraction/classification pour le modèle SOM-NPC avec deux voies possibles d'utilisation : l'extraction de paramètres suivie de leur classification ou la classification immédiate.

premières simulations concernant ce modèle constituent une nouvelle illustration expérimentale de l'organisation des données discriminantes sur les poids-codes de la couche de sortie du prédicteur. J'en présente quelques unes dans les paragraphes suivants, après une brève présentation des cartes auto-organisantes de Kohonen et des paramètres SOM-NPC.

4.2.1 Les cartes auto-organisantes de Kohonen

Le modèle des cartes auto-organisantes de Kohonen [99], [98], [100], est peut-être le modèle de réseau de neurones non supervisé le plus connu, bien qu'encore mal maîtrisé sur le plan théorique [17], [48]. Une carte de Kohonen est un réseau de cellules, non hiérarchiquement organisées, simples et latéralement interconnectées selon une topologie bien définie (une carte en 2 dimensions le plus souvent). Les neurones agissent en tant qu'unité d'entrée et reçoivent donc tous le même signal d'entrée de dimension λ. Apprendre un réseau à réaliser une tâche donnée se résume au problème de trouver le bon ensemble de vecteurs poids représentant au mieux l'ensemble des vecteurs d'entrée. Un vecteur d'entrée quelconque est ainsi comparé à l'ensemble des poids selon une certaine métrique (euclidienne la plupart du temps). Kohonen a mené de nombreuses expérimentations [100] dans lesquelles il utilise l'algorithme SOM (Self-organizing Map) pour créer des cartes de phonèmes. Les vecteurs d'entrée sont des spectres court-terme de signaux de parole et les cellules des représentations de phonèmes (carte phonotopique).
Le modèle SOM-NPC est également une carte de représentation phonétique, mais pour laquelle le signal est directement appliqué en entrée, sans calcul du spectre court-terme, puisque l'extracteur de caractéristiques fait partie intégrante du système.

4.2.2 Retour sur la distance NPC

Les paramètres NPC-C, NPC-2, DFE-NPC et LVQ-NPC sont tous fondés sur le principe d'incorporation des contraintes de classification lors de la phase d'adaptation. En restreignant le nombre de cellules de prédiction à une par classe, les modèles NPC-2 et DFE-NPC sont les modèles les plus "contraints", entraînant une augmentation substantielle de l'erreur de prédiction. Le modèle SOM-NPC apporte une souplesse supplémentaire en ce qu'il autorise un nombre plus élevé, mais néanmoins limité, de cellules par classe. Les cellules en question sont organisées selon une carte en deux dimensions appelée *carte prédictive* dans laquelle on introduit une notion de voisinage (au sens du plus court chemin) entre les cellules (figure 4.2). Chaque cellule de sortie représente ainsi un ensemble de trames pho-

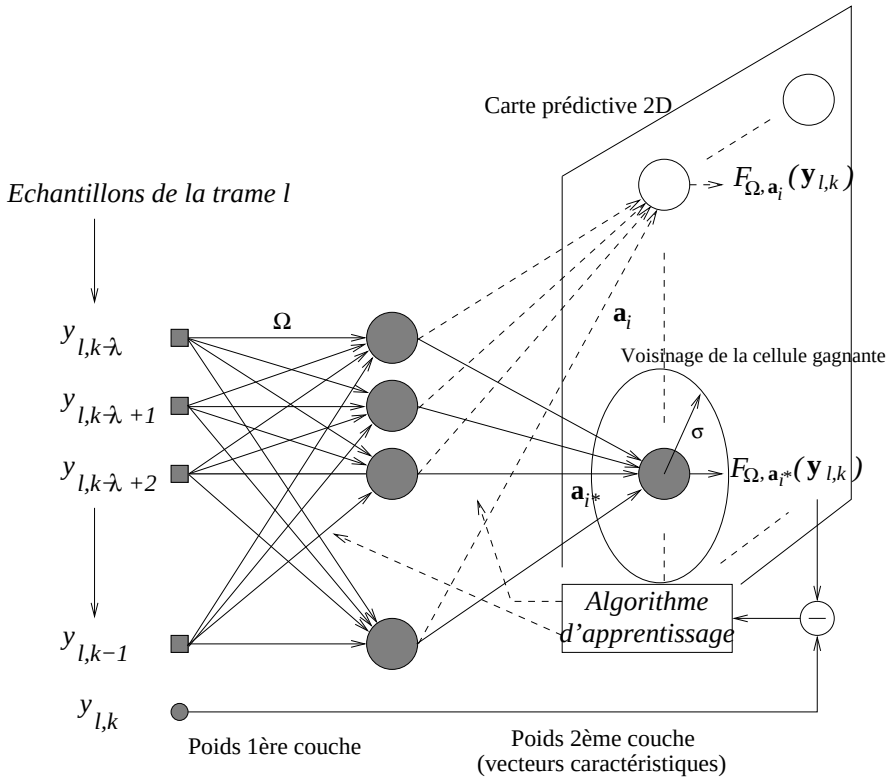


FIG. 4.2 – Structure de l'extracteur SOM-NPC (extrait de [64]).

nétiques : les trames pour lesquelles l'erreur de prédiction est la plus faible. On souhaite, après adaptation de la carte, obtenir la propriété de *tonotopie* que l'on retrouve dans les cartes auto-organisantes de Kohonen : deux trames proches dans l'espace des signaux sont codées par les poids de deux cellules proches au sens de la distance définie sur la carte. Pour obtenir ceci, j'ai proposé d'utiliser un algorithme d'apprentissage, en tous points identique à l'algorithme de Kohonen, dans lequel j'ai remplacé la distance euclidienne dans l'espace des entrées par la distance NPC dans l'espace des signaux.
L'algorithme se résume aux trois étapes suivantes :

Pour toute les trames d'apprentissage l :

1) Trouver le neurone gagnant i^* sur la carte de tel que :

$$i^* = \arg \min_{i=1,\ldots,N} d_\Omega^{NPC}(i,l) \tag{4.8}$$

où $d^{NPC}(i,l)$ représente la distance entre la trame modélisée par i et la trame l présentée en entrée. N est le nombre de cellules de la carte.

2) Adapter les poids du neurone gagnant et de ses voisins sur la carte de sorte à minimiser la distance d^{NPC} entre ces représentants et la trame en entrée (on montre très simplement que ceci est équivalent à minimiser l'erreur de prédiction) :

$$Q_l(\mathbf{a}_{1,\ldots,N}) = \sum_i^N \sum_k (y_{l,k} - G_\Omega \circ H_{\mathbf{a}_i}(\mathbf{y}_{l,k}))^2 V^\sigma(i^*, i) \tag{4.9}$$

où $V^\sigma(i,i^*) = e^{-\frac{d(i,i^*)}{2\sigma}}$ est la fonction de voisinage (une loi gaussienne dans notre cas, $d(i,i^*)$ étant la longueur du chemin le plus court entre i et i^* sur la carte prédictive et σ écart type). σ est une fonction décroissante du temps d'apprentissage de sorte que $\sigma(q) = [\frac{\sigma_f}{\sigma_d}]^{\frac{1}{N_q}} \sigma(q-1)$ où σ_d et σ_f sont les valeurs initiales et finales imposées à écart type et N_q le nombre d'itérations d'apprentissage.

3) Mettre à jour les poids de la première couche par rétropropagation de l'erreur.

Bien entendu, chercher le neurone gagnant se résume au choix de la cellule minimisant l'erreur de prédiction :

$$\arg \min_{i=1,\ldots,N} d_\Omega^{NPC}(i,l) = \arg \min_{i=1,\ldots,N} \{\log \frac{Q_l(\mathbf{a}_i)}{Q_l(\mathbf{a}_l)}\} = \arg \min_{i=1,\ldots,N} \{Q_l(\mathbf{a}_i)\} \tag{4.10}$$

Les expressions de mise à jour des poids sont dérivées du traditionnel algorithme de rétropropagation de l'erreur (descente de gradient). Le lecteur trouvera une description plus détaillée du modèle dans l'article [64] donné en annexe.

4.2.3 Premiers résultats obtenus à partir de signaux extraits de la base TIMIT

Les premières expérimentations, réalisées dans le cadre du stage de Master de Farid Feiz et publiées dans [64] et [63], ont porté sur trois sous-classes de phonèmes extraits de la base TIMIT : {/*aa*/, /*ae*/, /*ey*/, /*ow*/}, {/*b*/, /*d*/, /*g*/} et {/*p*/, /*t*/, /*k*/}. La labellisation de la carte selon le principe d'un étiquetage des cellules par la classe de fréquence la plus élevée a donné les résultats présentés figure 4.1. On observe clairement une différenciation des cellules en régions, même si ces régions ne sont pas parfaitement homogènes. Cette organisation illustre de façon intéressante les résultats de Hecht-Nielsen, Kurkova, Sprecher et Katsuura dont j'ai parlé plus haut, et selon lesquels seuls les poids des cellules de sortie dépendent des fonctions approximées par le réseau.

Par ailleurs, et c'est également ce qui nous intéresse ici, l'étiquetage réalisé à partir des trames de la base d'apprentissage nous permet d'utiliser l'extracteur de caractéristiques en classifieur. La règle de décision consiste à choisir la cellule de sortie, et donc la classe,

d d d d d g g g	k k k k k k k t	ow ow aa ae ae ae ae ey
d d d d d d b g g	k k k k k k p t	ow aa aa aa ae ae ae ey
d d d d d d b g g	k k p t t t t t	ow aa aa aa ae ae ey ey
d d d d g b g g	k p p p t t t t	ow ae aa aa ae ae ey ey
g d d d d d g g	p k p p t t t t	aa ae ae ae ae ae ae ey
d d d b d d g g	p k t t t t t t	aa aa ae ae ae ae ey ey
d b d b b b b b	p k k t t t t t	ow ae ae ae ae ae ey ey
d b b b g g g b	t t t t t t t t	aa ey ey ae ae ae ae ae

TAB. 4.1 – Labellisation des cellules de la carte prédictive pour trois bases phonétiques extraites de la base TIMIT.

Paramètres	Classifieur	Scores en généralisation		
		Voyelles	Plosives voisées	Plosives non voisées
SOM-NPC	SOM-NPC	59%	63%	69%
SOM-NPC	PMC	56%	64%	77%
LPC	PMC	70%	63%	76 %
MFCC	PMC	76%	64%	79%

TAB. 4.2 – Scores obtenus en classification par l'extracteur/classifieur SOM-NPC (résultats extraits de [64].

qui minimise l'erreur de prédiction lorsque l'on présente une trame de parole en entrée. Le tableau 4.2 présente les taux de reconnaissance obtenus par l'extracteur-classifieur SOM-NPC précédent. Cette étude comparative montre les résultats obtenus lorsque l'on exploite le SOM-NPC en tant qu'extracteur de caractéristiques (la classification est alors opérée par un réseau PMC), mais également en tant qu'extracteur/classifieur. Il s'agit de résultats obtenus en généralisation. Les exemples classés n'ont donc pas été utilisés, ni lors de la phase d'adaptation ni lors de l'étiquetage de la carte.

Des simulations en cours montrent qu'une augmentation de la dimension de la carte permet d'augmenter ces scores. En conséquence, même si l'on n'approche pas encore les scores obtenus par un système MFCC-PMC, comme le montre le tableau 4.2, ils n'en restent pas moins honorables et suffisent pour conclure que le SOM-NPC peut être étudié en tant que système complet d'extraction-classification. On ne pouvait pas tirer une telle conclusion à partir des travaux menés sur le modèle NPC-2 utilisé en classifieur (cf. §3.3.4, p. 54). Ci-dessous, au paragraphe 4.3.3, je présente d'autres résultats obtenus sur un problème de segmentation et regroupement de locuteurs. Enfin, je complète l'analyse du modèle dans le cadre des conclusions et perspectives, §4.4.3, p. 90.

4.3 Des projets réalisés, en cours et à venir...

Encadrement doctoral : Sébastien Herry (DEA de Robotique 2002), Christophe Charbuillet (DEA de Robotique 2004)

Les activités de recherche menées par notre équipe en extraction de caractéristiques ont atteint un niveau de maturité nous permettant aujourd'hui d'envisager de participer aux campagnes d'évaluation nationales et internationales. Ainsi que je l'ai souligné au chapitre 2 (cf. §2.1.1, p. 18), ces campagnes restent le meilleur moyen d'évaluer expérimentalement un nouveau modèle. Le problème posé pour ce qui concerne les paramètres NPC est celui de leur intégration dans un système complet de reconnaissance. Nous avons envisagé quatre grands domaines d'application : la compression/décompression audio, la reconnaissance de la parole, la reconnaissance du locuteur et la reconnaissance de la langue. Le premier domaine ouvre un champ d'applications relativement simples et rapides à mettre en oeuvre que nous avons exploré succinctement (§4.3.1).
Le deuxième est indéniablement le plus complexe, sauf à considérer des sous-domaines comme par exemple la reconnaissance de mots isolés. Le troisième est plus directement accessible, et c'est pourquoi nous avons décidé de participer à la campagne d'évaluation nationale ESTER [60] dont une partie est consacrée à la reconnaissance du locuteur (§4.3.3).
Enfin, le quatrième domaine rejoint la reconnaissance du locuteur du point de vue de la difficulté, ne serait-ce que parce que certaines des méthodologies utilisées y sont transposables. Sébastien Herry, en thèse à Thales dans le cadre d'un contrat CIFRE, a réalisé un système d'identification de la langue et notre équipe, en partenariat avec Thales, participe à la campagne d'évaluation NIST du mois d'octobre 2005. La poursuite de cette collaboration après la thèse de Sébastien Herry devrait être l'occasion de tester les paramètres NPC dans le cadre de l'identification de la langue (§4.3.4).
Enfin, ces dernières années ont vu l'émergence d'une communauté Européenne du traitement non linéaire de la parole au sein de laquelle nous nous sommes intégrés et qui permet aujourd'hui d'envisager de nouvelles collaborations (cf. §4.3.6).

4.3.1 Compression/décompression de la parole

Encadrement : David Charles Elie Nelson (DEA RESIN 2001), Michel Sendis (DEA RESIN 2003)
Publications : **[148]**

Bon nombre d'études sur les aspects non linéaires du signal de parole ont été réalisées dans le cadre de la compression/décompression (cf. chapitre 2). Bien que ne s'inscrivant pas directement dans les thèmes de notre équipe, Jean-Luc Zarader a cependant souhaité appliquer l'extracteur NPC au domaine du codage. Il a ainsi montré, à partir d'expérimentations, que le codeur NPC pouvait être porteur dans ce domaine d'applications qui connaît aujourd'hui un essort considérable.
Dans le cadre de son stage de DEA, David Charles Elie Nelson a réalisé une base d'apprentissage comportant 100 occurrences de 39 phonèmes segmentés en trames de 20ms. Après 40000 itérations d'apprentissage (adaptation "en bloc" des poids de la couche cachée du codeur), il disposait d'un codeur adapté pour une tâche de compression/décompression. A titre d'exemple, la table 4.3 montre les gains de prédiction et de quantification obtenus à partir d'expérimentations menées sur 5 phrases de la base Timit et n'ayant pas servi à l'apprentissage.
Cette étude a permis de montrer que l'on pouvait diviser d'un facteur 4 le débit du signal audio ($16kb/s$) sans altérer la qualité à l'audition du signal. Les meilleurs gains de pré-

	Compression		
Phrases	Quantification 3 bits		Quantification 4 bits
	$G_p(dB)$	$G_q(dB)$	$G_q(dB)$
1	12.1	20.8	27.6
2	13.5	21.9	28.4
3	14.0	22.7	29.8
4	14.2	23.2	30.1
5	14.3	24.7	31.1

TAB. 4.3 – Gains de prédiction et de quantification mesurés sur un codeur/décodeur NPC pour 5 phrases de la base Timit et deux valeurs de quantification (extrait de [148]).

diction étaient obtenus pour les phrases comportant le plus grand nombre de phonèmes voisés.
Au delà de ces résultats préliminaires, une voie de recherche prometteuse est envisageable, consistant à mettre au point un système de compression/décompression exploitant la présence de non linéarités mais tirant également parti des résultats obtenus avec les modèles DFE-NPC, LVQ-NPC et SOM-NPC dans la réalisation d'une quantification vectorielle adaptée.

4.3.2 Reconnaissance du locuteur

Collaborations : Université Polytechnique de Mataro (Marcos Faundez)
Publications : [32], [30]

La thématique de la reconnaissance de locuteurs, laissée de côté pendant quelques années (cf. chapitre 2, §2.1.1, p. 18), a été reprise par Mohamed Chetouani dans le cadre de sa thèse. Un certain nombre de stages de DEA ont également été consacrés à ce thème, avec en particulier pour objectif la mise en oeuvre des paramètres NPC dans le cadre de la participation à la campagne nationale ESTER dont je parle au paragraphe suivant.
Les travaux menés ont été réalisés dans le cadre d'une collaboration avec Marcos Faundez, dirigeant l'équipe de traitement de la parole de l'Université Polytechnique de Mataro et responsable du COST277. Cette étude a été financée par le COST277 (voir à ce sujet le paragraphe §4.3.6). Les tests d'évaluation ont été menés sur une base de locuteurs de langue espagnole (49 locuteurs) réalisée par le laboratoire de Marcos Faundez. Cette étude préliminaire a permis d'obtenir une première évaluation des paramètres NPC sur un problème de reconnaissance de locuteurs.
Le principe du système proposé est d'adapter un extracteur NPC par locuteur de la base. A la suite de l'extracteur se trouve un modèle de référence (un réseau de neurones de type diabolo ou prédictif, un mélange de gaussiennes GMM, etc.). Lors du test, une décision est prise par comparaison des hypothèses issues de chacun des modèles de locuteur.
Afin d'évaluer au mieux les performances, Mohamed Chetouani a utilisé deux algorithmes de modélisation : la méthode des matrices de covariance (la mesure de similarité AHS (ARithmetic-Harmonic Sphericity) permet de comparer les matrices entre-elles [12]) et

la méthode des modèles auto-régressif vectoriels [13]. La table 4.4 présente les résultats obtenus pour plusieurs extracteurs utilisés dans les mêmes conditions.

Extracteurs	scores AHS (%)	scores MAV (%)
LPC	90.61	90.61
LPCC	96.73	93.06
MFCC	97.55	95.69
PLP	86.12	78.36
NPC	100	100

TAB. 4.4 – Scores comparatifs obtenus en reconnaissance de locuteurs pour plusieurs extracteurs et deux méthodes de modélisation des locuteurs (AHS et MAV). Extrait de [32] et [30].

Ces scores sont d'autant plus intéressants qu'ils apportent une validation expérimentale supplémentaire des paramètres NPC en complétant les résultats déjà obtenus en reconnaissance de phonèmes. En revanche, le niveau élevé des scores obtenus (100%) provient du fait qu'il s'agit d'une base fermée comportant peu de locuteurs enregistrés dans des conditions d'acquisition optimales.

4.3.3 Regroupement de locuteurs : l'évaluation ESTER

Encadrement : Mourad Djedour (DEA RESIN 2003), Corina Iovan (Stage Erasmus 2004), Christophe Charbuillet (Stage DEA RESIN 2004)
Collaborations : LIA-Avignon (J. F. Bonastre), IRISA-Rennes (G. Gravier)
Publications : [60], [24]

La campagne ESTER (Evaluation des Systèmes de Transcription Enrichie des émissions Radiophoniques) [72], [57] est organisée dans le cadre du projet EVALDA, financé par le Ministère de la Recherche (appel à projet TECHNOLANGUE), sous l'égide scientifique de l'Association Francophone de la Communication Parlée (AFCP) avec le concours de la Délégation Générale de l'Armement (DGA) et de ELRA (European Language Ressources Association). Elle a pour objectif l'évaluation des performances des systèmes de transcription d'émissions radiophoniques, et plus généralement de promouvoir une dynamique de l'évaluation en France, autour du traitement de la parole de langue française (mise en place d'une structure pérenne d'évaluation et de diffusion des informations et des ressources concernées par ces évaluations). Les protocoles et les métriques utilisées s'appuient autant que possible sur les protocoles existants au sein des évaluations organisées par DARPA et NIST, de manière à profiter de l'expérience acquise mais également à faciliter les comparaisons entre les campagnes d'évaluation.

Ainsi que je l'ai souligné à plusieurs reprises dans ce mémoire, l'évaluation des paramètres NPC devait se faire, au mieux, dans le cadre d'une campagne d'évaluation. C'est pourquoi j'ai souhaité que nous participions à la campagne ESTER. Cette dernière comporte plusieurs tâches d'évaluation, dont la tâche dite *SRL* de Segmentation et Regroupement de Locuteurs, à laquelle nous avons participé à partir de la deuxième phase d'évaluation.

Notre inscription après le démarrage officiel de la campagne (nous n'avons donc pu participer à la première phase d'évaluation) nous a conduit à ne présenter de scores que sur les paramètres SOM-NPC. La difficulté de participer à une telle campagne (mise au point du système, temps d'apprentissage, etc.) et le peu de temps imparti, nous ont conduit à présenter des résultats que l'on peut qualifier de préliminaires [60]. Lors de l'atelier de clôture du mois de mars 2005, les participants ont cependant plaidé pour la poursuite de la campagne avec une phase 3 d'évaluation qui devrait nous permettre ainsi de réaliser une évaluation complète.
Par ailleurs, la nécessité de réaliser un système complet de regroupement de locuteur sur des signaux issus d'émissions radiophoniques a conduit à des collaborations informelles avec, d'une part Jean-François Bonastre du Laboratoire d'Informatique d'Avignon (LIA) pour le système ALIZE (plateforme logiciel libre en vérification automatique du locuteur) [115], [116] et d'autre part Guillaume Gravier de l'IRISA/Rennes pour l'utilisation du programme *audioseg* de segmentation et regroupement de locuteurs.

Phase d'adaptation de l'extracteur

50 itérations d'apprentissage ont été effectuées sur un signal du corpus 50h phase 1 (programme de France Inter représentant une heure de parole), soit 23 heures de simulation sur un PC pentium cadencé à 3.2 GHz. Les trames étaient de 20ms, extraites toutes les 10ms. L'extracteur comprenait une première couche de 16 neurones (horizon de prédiction de 20 échantillons) et une carte prédictive 2D comprenant 10x10 = 100 neurones prédictifs. Cette structure permettait d'estimer ensuite des vecteurs caractéristiques de dimension 16. Les poids initiaux étant choisis selon une distribution aléatoire.

Extraction des paramètres acoustiques

Je reporte sur le tableau 4.5 les résultats obtenus par les paramètres SOM-NPC estimés sur l'ensemble des signaux de la base ESTER d'évaluation (phase 2) (environ 8 heures de parole). Pour toutes les trames à traiter, l'extraction des vecteurs caractéristiques NPC a nécessité 10 itérations d'apprentissage (descente de gradient stochastique). Les temps de calcul étaient de l'ordre du temps réel et peuvent être sensiblement améliorés par une optimisation du code. Afin de permettre une comparaison des paramètres NPC avec d'autres méthodes d'extraction de caractéristiques, j'ai également reporté les résultats obtenus avec les paramètres LPC et LFCC (MFCC avec échelle linéaire) sur les mêmes signaux.
Dans notre étude, la mise en oeuvre de l'algorithme de segmentation à été effectuée par le système *audioseg* dans lequel une segmentation BIC en trois passes est implémentée. Un algorithme de regroupement hiérarchique a ensuite été appliqué sur les segments, basé sur la distance BIC, et toujours réalisé par l'outil *audioseg*.

Au vu de ces premiers résultats, les scores obtenus par les paramètres NPC se situent entre ceux obtenus par les paramètres LPC et ceux obtenus par les paramètres LFCC. Ils sont encourageants en ce qu'ils montrent que le modèle non linéaire NPC apporte des informations supplémentaires utiles à la segmentation locuteur, relativement au modèle linéaire de la même famille (modèle prédictif LPC).

Fichiers	LPC	LFCC	SOM-NPC
INTER_1_DGA	43.70	15.42	22.54
INTER_2_DGA	51.29	46.19	23.51
FINFO_1_DGA	35.17	27.58	38.52
FINFO_2_DGA	41.01	27.71	44.28
RFI_1_ELDA	27.85	14.67	22.59
RFI_2_ELDA	26.59	23.25	21.89
RTM_3_ELDA	33.31	4.8	22.47
RTM_4_ELDA	21.15	1.82	11.07
RTM_5_ELDA	8.28	6.42	8.33
RTM_6_ELDA	25.28	23.62	24.02
RTM_7_ELDA	19.11	42.98	15.17
RTM_8_ELDA	12.76	1.41	4.92
RTM_9_ELDA	21.08	14.71	23.09
RTM_10_ELDA	9.92	0.63	10.64
Moyenne Pondérée :	33.08%	22.50%	25.22%

TAB. 4.5 – Taux d'erreurs obtenus en reconnaissance en regroupement de locuteurs et présentés lors de l'atelier de cloture ESTER [60]

Christophe Charbuillet a débuté sa thèse de doctorat en octobre 2004 sur le thème de l'extraction de caractéristiques appliquée à la segmentation et au regroupement de locuteurs. A ce titre, il poursuit la campagne d'évaluation ESTER dans le cadre de la tâche SRL. Invité par Jean François Bonastre, il a eu l'occasion d'effectuer un séjour au LIA à Avignon pour travailler sur le système ALIZE. L'objectif de son travail est aujourd'hui de mettre en œuvre un algorithme génétique pour l'estimation non supervisée des poids de la couche cachée de l'extracteur NPC, donnant ainsi naissance à une nouvelle méthode d'adaptation appelée GA-NPC. Christophe Charbuillet a validé dans un premier temps son approche sur un problème d'adaptation des paramètres LFCC consistant à adapter la fréquence centrale et la largeur des bancs de filtres. Les premiers résultats obtenus sur la base d'évaluation ESTER sont encourageants puisqu'ils permettent de situer l'extracteur au même niveau que les paramètres MFCC. Ils ont fait l'objet d'une première publication acceptée au workshop *Non Linear Speech Processing* [24], école d'été des jeunes chercheurs en traitement non linéaire de la parole associée à l'action Européenne COST 277.

4.3.4 Reconnaissance de la langue

Encadrement : Sébastien Herry (DEA RESIN 2002 et Thèse CIFRE), Tanguy Evain (DEA, Université de Rennes-1, 2004)
Collaboration contractuelle : Thales (Celestin Sedogbo)
Publications : [79], [78]
Brevet : [80]

Un domaine en phase active de recherche à l'heure actuelle est celui de la reconnaissance (identification, détection et vérification) de langues. Les domaines d'application potentiels

vont de la surveillance civile et militaire aux centres d'appels téléphoniques (call center). Sébastien Herry effectue sa thèse sur ce sujet depuis novembre 2002, thèse que je co-encadre avec Jean-Luc Zarader et qui fait l'objet d'un contrat CIFRE avec Thales Communications ainsi que d'un contrat de collaboration sur trois ans entre Thales et le LISIF. Le contrat porte sur l'étude des méthodes neuronales prédictives appliquées à l'identification de la langue.
Les systèmes actuels de détection ou d'identification de la langue sont le plus souvent basés sur la modélisation statistique de la distribution des paramètres spectraux et/ou la modélisation du langage à partir des suites phonétiques. Ce dernier aspect nécessite cependant de disposer de modèles de langage pour toutes les langues étudiées, ce qui représente une réelle difficulté. Sébastien Herry a choisi une voie permettant de s'affranchir de l'étape de classification phonétique. Celle-ci consiste à classer directement les séries de paramètres spectraux à l'aide d'un ensemble de classifieurs neuronaux en coopération réciproque et par fusion de données.
Le système qu'il propose permet d'obtenir des scores de 77,4% sur un problème de reconnaissance d'une langue parmi 11. Ces mesures ont été effectuées sur le corpus multi-langues OGI [121] (base de données téléphoniques), sur des phrases d'une durée de 3 secondes. A titre de comparaison, les meilleures équipes ont obtenu des scores d'environ 80% sur la dernière évaluation NIST 2003 [114]. Ce système à fait l'objet d'un dépôt de brevet par Thales [80] dont les co-inventeurs sont Sébastien Herry, moi-même, Célestin Sedogbo et Jean-Luc Zarader.
Les mêmes questions concernant la comparaison des scores obtenus par des équipes différentes sur des systèmes différents se posent (cf. §2.1.1, p. 18). Nous participons donc, en collaboration avec Thales, à la campagne d'évaluation NIST 2005 qui aura lieu du 7 octobre au 7 décembre 2005. Le système proposé par Sébastien Herry pourra ainsi être validé dans le cadre d'une campagne internationale d'évaluation.

4.3.5 Estimation de la fréquence Doppler : le projet ADM

Encadrement : Serge Denis (DEA de robotique 2001)
Collaboration contractuelle : Laboratoire de Météorologie Dynamique (Philippe Drobinsky, Pierre Flamand)
Publications : [147], [150], [56]

Le stage de DEA de Serge Denis, effectué en 2001, avait pour sujet le *codage neuronal pour l'estimation de fréquence doppler*. Je le cite dans ce mémoire car il est une illustration des applications potentielles de l'extracteur NPC. En l'occurrence, il s'agissait d'utiliser un extracteur NPC pour extraire des paramètres caractéristiques d'un signal LIDAR. Serge Denis a avant tout réalisé une pré-étude consistant à estimer la fréquence centrale d'un signal de spectre gaussien et bruité. Il a pour cela constitué une base de signaux bruités comportant une quinzaine de *classes fréquentielles*, c'est à dire des signaux construits à partir de 15 fréquences centrales différentes. En dotant l'extracteur NPC de 2 cellules cachées, il obtenait des paramètres NPC à deux composantes, et pouvait ainsi représenter, dans le plan en 2 dimensions, les vecteurs caractéristiques estimés sur des signaux de fréquence centrale quelconque. Les résultats qu'il a obtenu on montré que les paramètres NPC estimés sur une succession de signaux de fréquence centrale strictement croissante

trouvaient une représentation linéaire dans le plan. Il était ainsi possible d'estimer, par une simple régression linéaire, la fréquence centrale de signaux non représentés dans la base d'adaptation.
Cette pré-étude a été réalisée dans le cadre d'une collaboration scientifique contractuelle avec le Laboratoire de Météorologie Dynamique (LMD) du C.N.R.S.(UMR 8539) avec P. Drobinski, A. Dabas et P.H. Flamant. L'objet était d'estimer par des méthodes connexionnistes la fréquence doppler des signaux Lidar rétrodiffusés par les aérosols, l'information recherchée étant la vitesse des vents, proportionnelle à la fréquence Doppler. Cette étude faisait suite au projet ALADIN lancé par l'Agence Spatiale Européenne (ESA). Ces travaux ont été à l'origine de deux publications (une conférence [150] et une revue IEEE [147]). Ils se sont poursuivis dans le cadre d'un nouveau projet, le projet ADM, auquel Jean-Luc Zarader et moi-même avons également participé. Je ne m'étend pas sur les méthodologies (connexionnistes) que nous avons utilisées et qui sont décrites dans le rapport final [56] car elles ne faisaient pas appel aux paramètres NPC, sujet central de ce mémoire.

4.3.6 L'action Européenne COST 277

Collaborations : Université Polytechnique de Mataro (Marcos Faundez), Université de Stirling (Amir Hussain)
Publications : **[31]**, [44], [63], [46]

Ainsi que je l'ai souligné dans l'introduction de ce mémoire, la thématique de l'extraction de caractéristiques en reconnaissance de la parole est restée peu traitée par la communauté internationale. En revanche, une communauté européenne s'est constituée autour du traitement non linéaire de la parole au début des années 2000, que nous avons pu rejoindre en 2003.
Fondé en 1971, le COST (European CO-operation in the field of Scientific and Technical research) est une plateforme intergouvernementale pour la coopération européenne dans le domaine de la recherche scientifique et technique. Son objectif est de contribuer au renforcement de la position européenne dans le domaine de la recherche. La flexibilité accordée aux actions COST permet en particulier l'exploration de nouveaux champs de recherche. Environ 200 actions sont actives aujourd'hui et engagent près de 30000 scientifiques de 34 états membres européens.

En juin 2001, Marcos Faundez de l'Université Polytechnique de Mataro (Espagne) a proposé et pris en charge une nouvelle action COST-TIST (Telecommunications, Information Science and Technologie) : l'action COST 277, *Non Linear Speech Processing* [54], dédiée au traitement non linéaire de la parole. Cette dernière a pris fin tout récemment au mois de juin 2005 (la durée des actions est de 4 années). Son principal objectif était l'amélioration des services vocaux dans les systèmes de télécommunication par le développement de nouvelles techniques non linéaires de traitement de la parole. C'est lors de notre participation au congrès associé à cette action, NOLISP 2003 [36], que nous avons rencontré Marcos Faundez et que des collaborations ont pu ainsi voir le jour. En particulier :
- une collaboration en reconnaissance du locuteur avec Marcos Faundez (§4.3.2) [32], [30] ;
- une collaboration en extraction de caractéristiques à partir d'une analyse en sous-bandes avec Amir Hussain [46] ;

– l'organisation d'une session spéciale au congrès ICANN 2005 avec Marcos Faundez et Amir Hussain sur le thème *Non Linear Predictive Models for Speech Processing* [44], [63],
– l'organisation d'une session "table ronde" de discussion au congrès ICANN 2005.

Suite à notre engagement dans cette thématique du traitement non linéaire de la parole, notre équipe s'est portée candidate pour l'organisation de la conférence NOLISP 2007. Cette candidature a été acceptée lors de la dernière conférence NOLISP qui s'est tenue à Barcelonne au mois d'avril 2005. Le prochain Workshop sur le traitement non linéaire de la parole, NOLIPS 2007, se tiendra donc à Paris et se distinguera des précédents puisqu'il ne sera plus supporté financièrement par l'action 277 du COST. L'objectif affiché devient celui de fédérer une communauté naissante, devenue autonome, ainsi que son extension au delà des frontières européennes.

4.4 Conclusion et perspectives

J'ai présenté dans ce mémoire un ensemble de travaux de recherche ayant principalement trait à l'extraction de caractéristiques du signal de parole. Ces travaux avaient pour objectif la mise au point de nouvelles méthodes d'analyse du signal visant à améliorer les systèmes de reconnaissance, les applications envisagées allant de la reconnaissance de la parole à l'identification de la langue en passant par la reconnaissance et le regroupement non supervisé de locuteurs.

Deux voies de recherches ont été étudiées qui sont, d'une part la modélisation non linéaire des processus de production de la parole et d'autre part la coopération des deux étapes que forment l'extraction de caractéristiques et la classification des traits acoustiques.
Un nouveau modèle d'analyse non linéaire, le modèle NPC, ainsi que cinq algorithmes permettant une adaptation discriminante des paramètres ont été proposés. Les différents modèles ont été évalués sur une tâche de classification dite *locale* et indépendante du contexte : la reconnaissance de phonèmes principalement. La nécessité d'une évaluation globale a été prise en compte et s'est traduite par la participation à une campagne d'évaluation nationale en reconnaissance du locuteur.

Les problématiques étudiées durant ces années, et portant sur la caractérisation non linéaire des signaux de parole ainsi que sur la coopération de l'analyse et de la classification, nous ont conduit à proposer des réponses que j'ai détaillées dans le présent document. Des questions, anciennes pour certaines, mais nouvelles pour d'autres sont également régulièrement apparues au cours de ce travail.
Elles constituent des paramètres à prendre en compte dans l'éventualité d'une poursuite de cette recherche. Mais elles sont également des indications pour de nouvelles thématiques de recherche et justifient donc d'être citées dans une rubrique portant sur les perspectives.

4.4.1 L'utilité d'une recherche en extraction de caractéristiques

De notre point de vue, ainsi que l'illustrent les perspectives présentées dans les paragraphes suivants, la recherche en extraction de caractéristiques mérite tout l'intérêt que nous lui avons porté et reste porteuse de résultats à venir. La question nécessite cependant d'être posée car, de toute évidence, la réponse apportée par une grande partie de la communauté

de recherche en parole est plus nuancée. Le principal grief opposé est qu'elle conduit à remettre en cause les méthodes d'analyse actuellement employées, en l'occurrence essentiellement les paramètres MFCC, lesquelles présentent des propriétés intéressantes. J'en citerai deux : la robustesse, en moyenne plus élevée que les méthodes d'analyse concurrentes, et la distribution statistique plus «gaussienne», ou «multi-gaussiennes» mais à covariances plus diagonales que d'autres. Ces caractéristiques permettent d'allier robustesse et efficacité, propriétés largement exploitées dans les systèmes actuels de reconnaissance.
Il est par ailleurs admis aujourd'hui que la prise en compte du contexte à tous les niveaux de traitement, allant du niveau phonétique au niveau sémantique, voire pragmatique, permet de retrouver l'essentiel de l'information perdue localement lors de l'extraction de caractéristiques. Quel intérêt aurait-on, dans ces conditions, à chercher une information qu'il est possible de retrouver par ailleurs ? Cette argumentation est une argumentation forte que l'on ne peut laisser de côté dans le cadre d'une analyse des perspectives.

La réponse à nos yeux réside dans ce que l'on entend par *possible de retrouver par ailleurs*. Plus on est proche des conditions réelles de la communication humaine, à savoir la parole spontanée, indépendante du locuteur, à grand vocabulaire, en environnement perturbé, en présence de bruits non stationnaires et non pas simplement additifs, plus les hypothèses locales sont incorrectement établies et plus élevé est le niveau de modélisation contextuel requis pour lever les ambiguïtés. Ce fait explique la dégradation des performances des systèmes automatiques actuels lorsqu'ils sont placés dans ces conditions de fonctionnement, certes dégradées, mais courantes. C'est la raison pour laquelle il nous a semblé important d'axer notre effort de recherche sur l'analyse du signal, étape dont l'influence sur le reste du système n'est plus à démontrer.
L'obtention de scores élevés en reconnaissance de phonèmes par une approche qui serait uniquement de type *bottom-up* ne saurait pour autant être une direction de recherche valable à nos yeux, dès lors que l'on sait que la perception humaine n'atteint que difficilement le niveau des 80% de reconnaissance indépendante du contexte, se satisfaisant d'un signal naturellement ambigu. D'autres arguments que je présente dans les paragraphes suivants renforceront cette idée qu'une poursuite de la recherche en extraction de caractéristiques, si elle est utile, ne peut se faire sans élargir le cadre dans lequel nous nous sommes placés jusqu'à présent.

4.4.2 La problématique de l'évaluation

Les contraintes NPC-C (cf. §3.2.3, 49), consistant à minimiser la variance intra-classe et maximiser la variance inter-classes des paramètres, n'ont pas donné les résultats attendus. Pourtant, d'autres méthodes comme les contraintes MER ou les contraintes LVQ ont permis d'améliorer significativement les performances. Cela est dû, entre autre, à la distribution des paramètres NPC qui ne satisfait pas l'hypothèse gaussienne suivie. Les contraintes imposées directement par le système de classification, qu'il soit par modélisation (choix du modèle le plus vraisemblable : NPC-2 et DFE-NPC) ou par discrimination (LVQ-NPC), ne font pas cette hypothèse.
Ceci nous conduit à une question d'importance lorsque l'on songe à intégrer les paramètres NPC dans un système de RAP ou de RAL. Ces systèmes fonctionnent en effet d'autant mieux que la distribution des paramètres est gaussienne et plus efficacement également

lorsque les matrices de covariance peuvent être approchées par des matrices diagonales. Les recherches portant sur l'analyse du signal de parole doivent-elles donc être guidées par le soucis d'améliorer la distribution des paramètres ? Au vu des résultats obtenus avec le NPC-C, l'extraction de caractéristiques non linéaires ne permet pas cela.
Bien qu'une motivation de recherche seulement guidée par l'adéquation de l'étape d'extraction de caractéristiques avec les modèles de classification existants soit intellectuellement moins satisfaisante, elle n'en constitue pas moins une direction possible, en tout cas un complément certain aux validations déjà réalisées. Certains systèmes de RAP peuvent par exemple accueillir avec succès les paramètres NPC. Citons par exemple les systèmes *mixtes* de type HMM/ANN [21] où les modèles connexionnistes mis en œuvre permettent l'estimation de distributions quelconques, ou encore les réseaux à commande cachée proposés par Ester Levin au début des années 90 [107],[108].

4.4.3 L'approche *deux en un* de l'analyse et de la classification simultanées

Nous avons concrètement traduit la coopération extraction/classification par l'adaptation de l'extraction de caractéristiques à la tâche de classification, et ce par apprentissage. Il s'agit donc d'une adaptation à la base de donnée, qui peut également s'accompagner d'une adaptation aux caractéristiques de l'environnement d'acquisition, aux caractéristiques du bruit ou encore aux caractéristiques du canal.
Or, de plus en plus aujourd'hui, la robustesse des systèmes de reconnaissance automatique est définie comme leur habilité à maintenir au mieux leurs performances lorsque les conditions environnementales ne sont plus celles présentes dans les bases utilisées lors du développement du système. Cela peut aller du changement de dialecte, à l'introduction de nouveaux mots en passant par le changement des propriétés du canal ou du bruit (citons le cas typique de la téléphonie cellulaire où les bruits sont fortement non stationnaires et à variation abrupte et où la position du micro est continuellement changeante).
Le modèle d'adaptation proposé pour les paramètres NPC, opéré lors de son développement, ne serait ainsi pas approprié pour de telles applications. Cette conclusion conduit cependant à une direction de recherche pertinente qui serait l'adaptation continue *en ligne* et non supervisée de l'extracteur NPC, à défaut de pouvoir concevoir un système définitivement *universel*.

L'adaptation SOM-NPC définie plus haut au paragraphe 4.2 de ce chapitre est une voie de recherche que j'ai ouverte en 2004 permettant de prendre en considération quelque uns des arguments développés ici dans le cadre des perspectives. Si l'on s'en tient au cadre habituel imposé par les systèmes de reconnaissance actuels, essentiellement la modélisation court-terme du signal, elle apporte la possibilité d'une approche non supervisée comme celle que nous avons utilisée pour la segmentation et le regroupement de locuteurs dans un flux audio (cf. §4.3.3).
Plus loin cependant, il devient possible, ainsi que les premiers résultats expérimentaux l'ont confirmé, de confondre les deux étapes de traitement que sont l'extraction de caractéristiques et la classification locale. Les scores obtenus lorsque l'on utilise l'extracteur en tant que classifieur sont en effet comparables aux scores obtenus par les paramètres LPC dans les mêmes conditions expérimentales. L'intérêt de cette approche est que les conditions de l'extraction se rapprochent ainsi de celles qui ont présidées lors de l'adaptation de la

carte, sous la forme d'une distribution *tonotopique* des modèles sur la carte. Ceci se traduit mathématiquement par le fait que la minimisation de la distance NPC lors de l'extraction est une approximation de la maximisation du MER, d'autant plus valable que le nombre de cellules sur la carte est élevé.
Si l'on souhaite cependant utiliser le classifieur SOM-NPC comme générateur d'hypothèses locales au sein d'un système conventionnel de reconnaissance, il reste nécessaire de pouvoir estimer les probabilités a posteriori dont a besoin le système HMM amont. Les cartes de Kohonen fournissent là un domaine de recherche particulièrement intéressant, en témoignent les analyses mathématiques qui en sont faites actuellement et qui tendent à interpréter l'auto-organisation des cartes comme une estimation des densités de probabilités, (suivant là une approche par modélisation) et à interpréter l'exploitation d'une carte apprise en terme d'estimation des probabilités a posteriori (approche discriminante). Nous sommes donc avec cette approche très précisément au cœur de la problématique modélisation/discrimination, ce qui lui confère un aspect particulièrement attractif et prometteur en terme de recherche.

Bien entendu, il convient de ne pas occulter le fait que le processus même d'auto-organisation et de son contrôle lors de l'apprentissage restent mal appréhendés du point de vue mathématique. Ainsi, les éventuelles pistes de recherche qui pourraient être mises en œuvre pour une adaptation en ligne visant à plus de robustesse au sens défini au paragraphe précédent (par le biais d'une adaptation en ligne de la vitesse d'apprentissage et de la dimension du voisinage en cours d'apprentissage par exemple) adressent des problèmes certainement très difficiles.
Enfin, de nombreux algorithmes ont déjà été proposés permettant de rendre les cartes auto-organisantes sensibles au contexte : ils pourraient être avantageusement repris dans le cadre du modèle SOM-NPC appliqué dans le cadre d'une prise en compte du contexte dès l'étape d'analyse du signal.

4.4.4 Vers un programme de recherche

Au delà des perspectives de recherche déjà dessinées autour du SOM-NPC, se pose la question de la remise en cause du schéma actuel auquel obéissent les systèmes de RAP de l'état de l'art. Conçus autour d'approches statistiques dans les étapes de décision (mixtures de gaussiennes et chaînes de Markov cachées pour l'essentiel), les premières étapes de traitement sont, elles, rarement considérées, bien que reconnues de cruciale importance. Les expérimentations menées, et que j'ai présentées dans ce mémoire, montrent que les paramètres NPC permettent d'obtenir des améliorations significatives des scores, tant en reconnaissance de phonèmes qu'en reconnaissance de locuteurs. Cependant, même s'il reste important, et intéressant, de valider cette approche sur de plus grands systèmes, la démarche que j'ai suivie et que je propose de continuer de suivre, s'inscrit plus vers la recherche de nouvelles méthodes de traitement pouvant conduire, dans un premier temps, à de moins bonnes performances lors de campagnes d'évaluation.

Ainsi, le programme de recherche que je propose est axé sur l'étude de nouvelles méthodologies d'analyse du signal et de reconnaissance des formes en vue d'améliorer la robustesse des systèmes actuels. Il trouve de multiples points d'ancrage et inclue les aspects suivants :

– la modélisation non linéaire des signaux
– la modélisation des signaux non stationnaires
– la prise en compte du contexte phonétique
– l'adaptation en ligne de l'analyse des signaux

J'ai déjà insisté sur l'adaptation en ligne lors de l'analyse des perspectives du modèle SOM-NPC ci-dessus. Je propose donc de compléter ici les trois premiers points.

Concernant la modélisation non linéaire de signaux, les travaux considérés dans ce mémoire donnent un aperçu de ce qui a déjà été fait, tant par notre équipe que par d'autres, sans toutefois clore définitivement la thématique. En effet, si les signaux de parole dits *non voisés* peuvent se satisfaire d'une modélisation toute à la fois linéaire et gaussienne, il n'en va pas de même des signaux voisés qui s'apparentent plus à des signaux déterministes, éventuellement chaotiques, qu'à des signaux aléatoires. De ce point de vue, l'approche source/filtre n'apparaît pas comme la mieux adaptée, d'autant qu'elle ne caractérise que les dépendances court-terme du signal. Elle pourrait être remplacée par des modèles à base d'oscillateurs non linéaires comme par exemple des réseaux de neurones récurrents.

Le deuxième point concerne la modélisation de signaux non stationnaires et remet en cause plus fondamentalement l'approche classique. Le signal de parole, fortement non stationnaire sur le moyen et long terme (au delà de 30 ms), est supposé stationnaire sur de courts intervals de temps. L'analyse de trames consécutives de 10 à 20 ms permet ainsi de générer des hypothèses dites locales, lesquelles sont réexaminées, en séquence, à un niveau de traitement supérieur (chaînes de markov) pour déterminer le treillis phonétique. Changer d'échelle de temps en s'accordant la possibilité de modéliser le phonème entier, voire la syllabe, requiert une analyse non stationnaire du signal, analyse que l'on peut également mettre en relation avec l'analyse non linéaire vue plus haut. La complexité des modèles trouve alors une compensation dans le nombre d'échantillons disponibles pour procéder à leur estimation.

Il est une source de variabilité dont on aimerait bien s'abstraire et qui est due aux phénomènes de co-articulation. Les phonèmes ont un début, un milieu et une fin et pour une même classe phonétique, début et fin peuvent différer du simple fait de la juxtaposition de tel phonème ou de tel autre. A cette dépendance contextuelle s'en ajoute une autre, favorable cette fois : toutes les séquences phonétiques n'existent pas dans un langage donné. La question posée est donc de savoir si la prise en compte du contexte doit être considérée dès l'étape d'analyse des signaux, dans le but d'en permettre une meilleure modélisation, ou non, comme c'est le cas dans les systèmes actuels. Cette question du contexte est bien entendu en relation étroite avec l'analyse non stationnaire des signaux, si bien que la non linéarité, la non stationarité et le contexte forment un ensemble cohérent qu'il serait particulièrement intéressant d'étudier.

Ces questions que je pose ainsi en forme de conclusion de ce mémoire sont autant de voies de recherche, voire d'une action fédérative pour un ensemble d'équipes, comme cela a pu être le cas pour le traitement non linéaire de la parole autour de l'action Européenne COST 277. Elles posent des problèmes d'ordre pratique, comme la réalisation concrète

de systèmes de traitement ayant pour objectif d'être efficaces et robustes, mais également théoriques et méthodologiques pour tout ce qui concerne les questions de linéarité, gaussianité, stationnarité, et plus généralement la modélisation des processus dynamiques non linéaires. Leur champ d'application est large, en particulier dans le domaine des signaux bio-médicaux, comme en témoigne le projet *Renoir* de rééducation neuro-orthopédique auquel nous participons. Elles se situent enfin au carrefour de plusieurs branches des sciences de l'ingénieur, comme le traitement du signal, la reconnaissance de formes, et l'intelligence artificielle, mais également des sciences du vivant lorsque l'on cherche à modéliser le fonctionnement du conduit vocal ou de l'organe auditif, ou encore l'ensemble de la chaîne de communication que forment l'organe vocal, l'environnement, l'organe auditif et le cerveau.

.

Bibliographie

[1] University of Pennsylvania Linguistic Data Consortium. The nist darpa-timit acoustic-phonetic continuous speech corpus : a multi speakers data base, 1990.

[2] H. M. Teager and S. M. Teager. Evidence for non linear sound production mechanisms in the vocal tract. *Speech Production and Speech Modeling*, 55 :241–261, July 1989.

[3] S. Amari. A theory of adaptive pattern classifiers. *IEEE Transactions on Elec. Comput.*, 16 :299–307, june 1967.

[4] T. Arthieres and P. Gallinari. Neural models for extracting speaker characteristics in speech modelization systems. In *EuroSpeech*, pages 2263–2266, 1993.

[5] T. Artières. *Méthodes prédictives neuronales : application à l'identification du locuteur*. PhD thesis, Université Pierre et Marie Curie, 1996.

[6] B.S. Atal and S. L. Hanauer. Speech analysis and synthesis by linear prediction of the peech wave. *Journal of the Acoustical Society of America*, 50 :637–655, 1971.

[7] Y. Bennani and P. Gallinari. Neural networks for discrimination and modelization of speakers. *Speech communication*, 1995.

[8] A. Biem. *Neural models for extracting speaker characteristics in speech modelization system*. PhD thesis, Paris VI, 1997.

[9] A. Biem and S. Katagiri. Feature extraction based on minimum classification error/ generalized probabilistic descent method. In *Proceedings of International Conference on Signal and Speech Processing*, volume 2, pages 275–278, 1993.

[10] A. Biem and S. Katagiri. Filter bank design based on discriminative feature extraction. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 485–488, 1994.

[11] F. Bimbot, G. Chollet, and J.P. Tubach. Tdnns for phonetic features extraction : a visual exploration. In *International Conference on Speech and Signal Processing (ICASSP)*, volume 1, pages 73–76, 1991.

[12] F. Bimbot and L. Mathan. Text-free speaker recognition using an arithmetic-harmonic sphericity measure. In *Eurospeech*, pages 169–172, 1991.

[13] F. Bimbot, L. Mathan, A. De Lima, and G. Chollet. Standrad and target driven ar-vector models for speech analysis and speaker recognition. In *International Conference on Speech and Signal Processing (ICASSP)*, volume 2, pages 5–8, 1992.

[14] M. Birgmeier. Nonlinear prediction of speech signals using radial basis function networks. In *Proceedings of EUSIPCO96*, pages 459–462, September 1996.

[15] C. M. Bishop. Novelty detection and neural network validation. In *IEE proceedings : Vision, Image and Signal Processing. Special Issue on applications of neural networks*, volume 141, pages 217–222, 1994.

[16] C. M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press - Oxford, 1995.

[17] C. M. Bishop, M. Svensen, and C. K. I. Williams. Developments of the generative topographic mapping. *neurocomputing*, 21 :203–224, 1998.

[18] R. Boite, H. Bourlard, T. Dutoit, J. Hancq, and H. Leich. *Traitement de la parole*. Presses polytechniques et universitaires romandes, 2000.

[19] T. Bojer, B. Hammer, M. Strickert, and T. Villmann. Determining relevant input dimensions for the self-organizing map. In *Proc. of Neural Networks and Soft Computing (ICNNSC 2002)*, pages 388–393, 2003.

[20] H. Bourlard, H. Hermansky, and N. Morgan. Towards increasing speech recognition errors. *Speech Communication*, 18 :205–231, 1996.

[21] H. Bourlard and N. Morgan. Hybrid hmm/ann systems for speech recognition : Overview and new research directions. *Lecture Notes In Computer Science*, 1387 :389–417, 1997.

[22] I. Callot, J. L. Zarader, and B. Gas. Predictive radial basis functions networks for speaker identification. In *ISMIP'96 (International Symposium on Multi-Technology Information Processing)*, pages 435–439, 1996.

[23] I. Callot, J. L. Zarader, and M. Milgram. Dynamic neural network using weights transformation, application to speaker identification. In *International Conference on Artificial Neural Networks (ICANN'95)*, pages 9–13, 1995.

[24] C. Charbuillet, B. Gas, M. Chetouani, and J. L. Zarader. A new approach for speech feature extraction based on genetic algorithms. In *WNLSP'05 (Non Linear Speech Processing Workshop)*, 2005.

[25] C. Chavy. *Codeur Neuronal Prédictif : application au codage de phonèmes*. PhD thesis, Université Paris VI, 2004.

[26] C. Chavy, B. Gas, and J. L. Zarader. Discriminative coding with predictive neural networks. In *ICANN'99 (International Conference on Artificial Neural Network)*, pages 219–220, 1999.

[27] C. Chavy, B. Gas, and J. L. Zarader. Neural predictive coding applied to noisy phonemes. In *IJCNN'99 (International Joint Conference on Neural Networks)*, pages 219–220, 1999.

[28] C. Chavy, B. Gas, and J. L. Zarader. Neural predictive coding for speech signal. In *IWANN'99 (International workshop on Artificial Neural Networks)*, volume 2, pages 824–832, 1999.

[29] M. Chetouani. *Codage neuro-prédictif pour l'extraction de caractéristiques de signaux de parole*. PhD thesis, Université Paris VI, 2004.

[30] M. Chetouani, M. Faundez, B. Gas, and J. L. Zarader. A new nonlinear feature extraction algorithm for speaker verification. In *ICSLP'04 (International Conference on Spoken and Langage Processing)*, 2004.

[31] M. Chetouani, M. Faundez, B. Gas, and J. L. Zarader. *Nonlinear speech processing : Algorithms and Analysis*, chapter Non-Linear Speech Feature Extraction for Phoneme Classification and Speaker Recognition. G. Chollet and A. Esposito and M. Faundez and M. Marinaro, Springer-Verlag, 2005.

[32] M. Chetouani, M. Faundez-Zanuy, B. Gas, and J. L. Zarader. A new nonlinear speaker parameterization algorithm for speaker identification. In *Odyssey'04 (Proc. of ISCA Tutorial and Research Workshop on Speaker and Recognition Langage Workshop)*, pages 309–314, 2004.

[33] M. Chetouani, B. Gas, and J. L. Zarader. Discriminative training for neural for neural predictive coding applied to speech feature extraction. In *IJCNN'02 (International Joint Conference on Neural Networks)*, volume 1, pages 852–857, 2002.

[34] M. Chetouani, B. Gas, and J. L. Zarader. The modular neural predictive coding architecture. In *ICONIP'02 (International Conference On Neural Information Processing)*, pages 452–456, 2002.

[35] M. Chetouani, B. Gas, and J. L. Zarader. Cooperative modular neural predictive coding. In *NNSP'03 (IEEE International Workshop on Neural Networks for Signal Processing)*, pages 637–646, 2003.

[36] M. Chetouani, B. Gas, and J. L. Zarader. Maximisation of the modelisation error ratio for neural predictive coding. In *NOLISP'03 (ISCA Tutorial and Research Workshop on NOn-Linear Speech Processing)*, pages 77–80, 2003.

[37] M. Chetouani, B. Gas, and J. L. Zarader. Modular neural predictive coding for discriminative feature extraction. In *ICASSP'03 (IEEE International Conference on Acoustic Speech and Signal Processing)*, volume 2, pages 33–36, 2003.

[38] M. Chetouani, B. Gas, and J. L. Zarader. Une architecture modulaire pour l'extraction de caractéristiques en reconnaissance de phonèmes. In *GRETSI'03 (Colloque du Groupe d'Etude du Traitement du Signal et des Images)*, 2003.

[39] M. Chetouani, B. Gas, and J. L. Zarader. Classifieur à prototypes et codage neuro-prédictif pour l'extraction non linéaire de caractéristiques en classification de phonèmes. In *JEP'04 (Journées d'Etudes sur la Parole)*, pages 125–128, 2004.

[40] M. Chetouani, B. Gas, and J. L. Zarader. Coopération entre codeurs neuro-prédictifs pour l'extraction de caractéristiques en reconnaissance de phonèmes. In *RFIA'04 (Reconnaissance des Formes et Intelligence Artificielle)*, pages 667–676, 2004.

[41] M. Chetouani, B. Gas, and J. L. Zarader. Learning vector quantization and neural predictive coding for nonlinear speech feature extraction. In *EUSIPCO'04 (European Signal Processing Conference)*, 2004.

[42] M. Chetouani, B. Gas, and J. L. Zarader. Simultaneous non-linear prediction and discrimination for improved speech feature extraction. *Soumis à Neurocomputing*, 2005.

[43] M. Chetouani, B. Gas, J. L. Zarader, and C. Chavy. Extraction de caractéristiques par codage neuro-prédictif. In *JEP'02 (Journées d'Etudes sur la Parole)*, pages 85–88, 2002.

[44] M. Chetouani, A. Hussain, M. Faundez-Zanuy, and B. Gas. Non linear predictive models for speech processing. In *International Congress on Artificial Neural Networks (ICANN'05)*, pages 779–785, september 2005.

[45] M. Chetouani, A. Hussain, B. Gas, and J. L. Zarader. New sub-band processing framework using non-linear predictive models for speech feature extraction. In *NOLIPS'05 3th International Conference on NOn-Linear Speech Processing*, pages 269–274, 2005.

[46] M. Chetouani, A. Hussain, B. Gas, and J. L. Zarader. Functionally expanded neural networks applied to speech feature extraction. *Soumis à IEEE Trans. on Speech and Audio Processing*, 2006.

[47] C. F. Chong, L. W. Chan, and P. C. Ching. Hierarchical mixtures of experts for phonetic classification. In *International Symposium on Multi-Technology Information Processing (ISMIP'96)*, pages 461–466, 1996.

[48] M. Cottrell, J. C. Fort, and G. Pagès. Theoretical aspects of the som algorithm. *neurocomputing*, 21 :119–138, 1998.

[49] A. de la Torre, A. M. Peinado, A. J. Rubio, and V. Sánchez. A dfe-based algorithm for feature selection in speech recognition. In *International Conference on Speech and Signal Processing (ICASSP)*, volume 2, pages 1519–1522, 1997.

[50] A. de la Torre, A. M. Peinado, A. J. Rubio, V. E. Sánchez, and J. E. Díaz. An application of minimum classification error to feature space transformations for speech recognition. *Speech Communication*, 20 :273–290, 1996.

[51] F. Díaz-de-María and A. R. Figueiras-Vidal. Nonlinear prediction for speech coding using radial basis functions. In *International Conference on Speech and Signal Processing (ICASSP)*, volume 1, pages 788–791, 1995.

[52] G. Dreyfus, O. Macchi, S. Marcos, O. Nerrand, L. Personnaz, Roussel-Ragot, D. Urbani, and C. Vignat. Adaptive training of feedback neural networks for non linear filtering. *Neural Networks for signal processing*, 2 :550–559, 1992.

[53] T. Eisele, R. Haeb-Umbach, and D. Langmann. A comparative study of linear feature transformation techniques for automatic speech recognition. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, volume 1, pages 252–255, 1996.

[54] M. Faundez-Zanuy. European cooperation in the field of scientific and technical research, action 277 : Non linear speech processing.

[55] M. Faundez-Zanuy and A. Esposito. Nonlinear speech processing applied to speaker recognition. In *Conf. on the Advent of Biometrics on the Internet*, 2002.

[56] P. Flamant, A. Garnier, D. Bruneau, A. Hertsog, C. Loth, A. Dolfi, A. Dabas, J. L. Zarader, B. Gas, C. Werner, J. Streicher, I. Leike, and F. Jochim. Evaluation of direct detection doppler wind lidar signal processing, final report. Technical report, IPSL, ONERA, Meteo France, UPMC, DLR (Contrat ESTEC N.14442/NL/SF), september 2002.

[57] S. Galliano, E. Geoffrois, D. Mostefa, K. Choukri, J. F. Bonastre, and G. Gravier. The ester ase ii evaluation campaign for the rich transcription of french broadcast news. In *Eurospeech/Interspeech*, 2005.

[58] B. Gas. Application des réseaux neuronaux au traitement d'images infrarouges *(mémoire de stage)*, 1990.

[59] B. Gas. *Un modèle connexionniste non supervisé pour l'apprentissage et la reconnaissance de séquences temporelles.* PhD thesis, Université Parix XI, mai 1994.

[60] B. Gas, C. Charbuillet, M. Chetouani, and J. L. Zarader. Paramètres npc pour la segmentation et le regroupement de locuteurs dans un flux audio. In *Workshop ESTER (Evaluation de Systèmes de Transcription enrichie d'Emissions Radiophoniques)*, mars 2005.

[61] B. Gas, C. Chavy, J. L. Zarader, and I. Callot. Cooperative criterion for predictive neural networks : application to phonemes recognition. In *ISMIP'96 (International Symposium on Multi-Technology Information Processing)*, pages 449–454, 1996.

[62] B. Gas, M. Chetouani, and J. L. Zarader. Extraction de caractéristiques non linéaire et discriminante : application à la reconnaissance de phonèmes. *Soumis à Traitement du Signal*, 2006.

[63] B. Gas, M. Chetouani, J. L. Zarader, and C. Charbuillet. Predictive kohonen map for speech feature extraction. In *ICANN'05 (International Congress on Artificial Neural Networks)*, pages 793–799, september 2005.

[64] B. Gas, M. Chetouani, J. L. Zarader, and F. Feiz. The predictive self organizing map : application to speech features extraction. In *WSOM'05 (Workshop on Self Organizing Maps)*, pages 497–504, september 2005.

[65] B. Gas, J. L. zarader, and C. Chavy. Codage discriminant appliqué à la reconnaissance de phonèmes. In *GRETSI'99 (Colloque du Groupe d'Etude du Traitement du Signal et des Images)*, volume 3, pages 873–876, 1999.

[66] B. Gas, J. L. zarader, and C. Chavy. A new approach to speech coding : The neural predictive coding. *Journal of Advanced Computational Intelligence*, 4(1) :120–127, 2000.

[67] B. Gas, J. L. zarader, C. Chavy, and M. Chetouani. Discriminant features extraction by predictive neural networks. In *SSIP'2001 (International Conference on Signal, Speech and Image Processing)*, pages 1831–1835, 2001.

[68] B. Gas, J. L. zarader, C. Chavy, and M. Chetouani. Discriminant neural predictive coding applied to phoneme recognition. *Neurocomputing*, 56 :141–166, 2004.

[69] B. Gas, J. L. zarader, P. Sellem, and J.C. Didiot. Speech coding by limited weights neural network. In *ICSMC'97 (IEEE International Conference on Systems Man and Cybernetics)*, 1997.

[70] T. Gautama, D.P. Mandic, and M.M Van Hulle. On the characterisation of the deterministic/stochastic and linear/nonlinear nature of time series. Technical Report DPM-04-5, Imperial College London, 2004.

[71] F. Girosi and T. Poggio. Representation properties of networks : Kolmogorov's theorem is irrelevant. *Neural Computation*, 1(4) :465–469, 1989.

[72] G. Gravier, J.F. Bonastre, S. Galliano, E. Geoffrois, K. Tait, and K. Choukri. Ester, une campagne d'évaluation des systèmes d'indexation d'émission radiophoniques. In *Journées d'Etude sur la Parole (JEP'04)*, 2004.

[73] R. Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. In *Proceedings of the International Conference on Neural Networks*, pages 11–13, 1987.

[74] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustical Society of America*, 87(4) :1738–1752, 1990.

[75] H. Hermansky. Should recogizers have ears ? In *Proc. ESCA Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*, pages 1–10, 1997.

[76] H. Hermansky. Should recognizers have ears ? *Speech Communication*, 25 :3–27, 1998.

[77] H. Hermansky and N. Morgan. Rasta processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2 :587–589, 1994.

[78] S. Herry. Improvement in language detection by neural discrimination comparaison with predictive model. In *International Conference on Artificial Neural Networks (ICANN)*, 2005.

[79] S. Herry, B. Gas, C. Sedogbo, and J. L. Zarader. Langage detection by neural discrimination. In *ICSLP'04 (International Conference on Spoken and Langage Processing)*, 2004.

[80] S. Herry, B. Gas, C. Sedogbo, and J. L. Zarader. Méthode d'identification automatique de la langue temps réel par discrimination sans annotation de corpus(brevet, nř d'enregistrement national : 04 02597), 2004.

[81] K. Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2 :359–366, 1989.

[82] A. Hussain. locally-recurrent neural-networks for real-time adaptive nonlinear prediction of non-stationary signals. *Control and Intelligent Systems*, 28(2) :65–71, 2000.

[83] F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 23 :67–72, 1975.

[84] F. Itakura and S. Saito. Analysis synthesis telephony based on the maximum likelihood method. In *Proceedings of the 6th International Congress on Acoustic*, pages 17–20, 1968.

[85] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3 :79–87, 1991.

[86] M. I. Jordan and R. A. Jacobs. Hierarchies of adaptive experts. *Advances in neural Information Processing System*, 4 :985–993, 1992.

[87] B. H. Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, 40(12) :3043–3054, december 1992.

[88] A. Kai-Fu Lee and Hsiao-wuen Hon. Speaker independent phone recognition using hidden markov models. *IEEE Transactions on Acoustic Speech and Signal Processing*, 1989.

[89] S. Kajarekar and H. Hermansky. Analysis of source of variability in speech. In *Proceedings of Eurospeech99*, pages 343–346, 1999.

[90] S. KARAJEKAR. *Analysis of variability in Speech with Applications to Speech and Speaker Recognition.* PhD thesis, OGI, Portland, USA, 2002.

[91] S. KATAGIRI. *Handbook of Neural Networks for Speech Processing.* Artech House, 2000.

[92] S. KATAGIRI, C. H. LEE, and B. H. JUANG. A generalized probabilistic descent method. *Proceedings of the Acoustical Society of Japan*, pages 141–142, 1990.

[93] S. KATAGIRI, C. H. LEE, and B. H. JUANG. Discriminative multilayer feed-forward networks. In *IEEE Workshop on Neural Networks for Signal Processing*, pages 309–318, 1991.

[94] H. KATSUURA and D. A. SPRECHER. Computational aspects of kolmogorov's superposition theorem. *Neural Networks*, 7(3) :455–461, 1994.

[95] G. I. KECHRIOTIS and E. S. MANOLAKOS. Using neural networks for nonlinear and chaotic signal processing. In *International Conference on Speech and Signal Processing (ICASSP'93)*, volume 1, pages 465–468, 1993.

[96] A. KEHAGIAS and V. PETRIDIS. Time series segmentation using predictive modular neural networks. *Neural Computation*, 9 :1691–1710, 1997.

[97] M. KOHDA, S. HASHIMOTO, and S. SAITO. Spoken digit mechanical recognition. *Trans. Inst. Electron. Commun. Eng. (Japan)*, 55(3), 1972.

[98] T. KOHONEN. Analysis of a simple self-organizing process. *Biol. Cybernet.*, 44 :135–140, 1982.

[99] T. KOHONEN. Self-organized formation of topologically correct features maps. *Biol. Cybernet.*, 43 :59–69, 1982.

[100] T. KOHONEN. *Self-Organizing Maps, 3rd Edition.* Springer, Berlin, 2001.

[101] G. KUBIN. *Speech coding and synthesis*, chapter Nonlinear processing of Speech, pages 557–609. W.B. Kleijn and K.K. Paliwal Editors, Elsevier Science, 1995.

[102] V. KURKOVA. Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, 5 :501–506, 1992.

[103] K. J. LANG, A. H. WAIBEL, and G.E. HINTON. A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3 :23–43, 1990.

[104] A. LAPEDES and R. FARBER. Nonlinear signal processing using neural networks : Prediction and system modelling. *Internal Report, Los Alamos National Laboratory*, july 1987.

[105] Y. LECUN. *Modèles connexionnistes de l'apprentisssages.* PhD thesis, Paris XI, 1987.

[106] H. C. LEUNG, B. CHIGIER, and J. R. GLASS. A comparative study of signal representations and classification techniques for speech recognition. In *International Conference on Acoustic Speech and Signal Processing*, pages 680–683, 1993.

[107] E. LEVIN. Word recognition using hidden control neural architecture. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 433–436, 1990.

[108] E. LEVIN. Hidden control neural architecture modeling of nonlinear time varying systems and its applications. *IEEE Trans. on Neural Networks*, 4(1) :109–116, 1993.

[109] J. C. Lucero. A theoretical study of the hysteresis phenomenon at vocal fold oscillation onset-offset. *Journal of Acoustic Society of America*, 1 :423–431, 1999.

[110] N. Ma, T. Nishi, and G. Wei. On a code-excited nonlinear predictive speech coding (cenlp) by means of recurrent neural networks. *IEICE Transactions fundamentals, spec. issue on digital signal processing*, E81-A(8) :1628–1634, 1998.

[111] N. Ma and G. Wei. Speech coding with nonlinear local prediction model. In *Proceedings of International Conference on Acoustic, Speech and Signal Processing*, volume 2, pages 1101–1104, 1998.

[112] E. MacDermott and S. Katagiri. Prototype based discriminative training for various speech units. In *IEEE International Conference on Acoustic Speech and Signal Processing*, volume 1, pages 417–420, 1992.

[113] P. M. Maragos and A. Potamianos. Fractal dimensions of speech sounds : computation and application to automatic speech recognition. *Journal of Acoustic Society of America*, 3 :1925–1933, 1999.

[114] A. Martin and M. Przybocki. Language recognition evaluation. In *Proc. of Eurospeech*, september 2004.

[115] D. Mauraru, S. Meignier, L. Besacier, and J. F. Bonastre. The elisa consortium approaches in speaker segmentation during the nist 2002 speaker recognition evaluation. In *International Conference on Speech and Signal Processing (ICASSP)*, 2003.

[116] D. Mauraru, S. Meignier, C. Fredouille, L. Besacier, and J. F. Bonastre. The elisa consortium approaches in broadcast news speaker segmentation during the nist 2003 rich transcription evaluation. In *International Conference on Speech and Signal Processing (ICASSP)*, 2004.

[117] E. McDermott. *Discriminative Training for Speech Recognition*. PhD thesis, Waseda University, Japan, 1997.

[118] A. Mellouk. *Un système Neuro-prédictif pour la reconnaissance automatique de la parole continue*. PhD thesis, Université Pierre et Marie Curie, 1994.

[119] A. Mellouk and P. Gallinari. Discriminative neural prediction system for speech recognition. In *Proceedings of International Conference on Acoustic, Speech and Signal Processing*, pages 1533–1537, 1993.

[120] P. J. Moreno and R. N. Stern. Sources of degradation of speech recognition in the telephone network. In *International Conference on Acoustic Speech and Signal Processing*, volume 1, pages 109–112, 1994.

[121] Y. K. Muthusamy, R. A. Colde, and B. T. Oshika. The ogi multilingual telephone speech corpus. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 895–898, 1992.

[122] Y. Nakano, A. Ichikawa, and K. Nakata. Evaluation of various parameters in spoken digits recognition. In *IEEE conf. Speech Communication and Processing*, page C4, april 1972.

[123] A. Pagès-Zamora, M. A. Lagunas, M. Nájar, and A.I. Pérez-Neira. The k-filter : A new architecture to model and design non-linear systems from kolmogorov's theorem. *Signal Processing*, 44 :249–267, 1995.

[124] B. Petek and A. Ferligoj. Exploiting prediction error in a predictive-based connectionnist speech recognition system. In *International Conference on Speech and SIgnal Processing*, pages 267–270, 1993.

[125] V. C. Raycar, B. Yegnanarayana, and S. R. Duraiswamy. Speaker localization using excitation source information in speech. *IEEE Transactions on Speech and AUdio Processing*, 2004.

[126] W. Reichl, S. Harengel, F. Wolferstetter, and G. Ruske. Neural networks for nonlinear discriminant analysis in continuous speech recognition. In *Eurospeech*, pages 537–540, 1995.

[127] D. E. Rumelhart, G. E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323 :533–536, 1986.

[128] P.S. Sastry, G. Santharam, and K.P. Unnikrishnan. Memory neuron networks for identification and control of dynamical systems. *IEEE Trans. on Neural Networks*, 5 :305–319, march 1994.

[129] J. Schoentgen. Non-linear signal representation and its application to the modelling of the glottal waveform. *Speech communication*, 9 :189–201, 1990.

[130] J. Schoentgen. On the bandwidth of a shaping function model of the phonatory excitation signal. In *No Linear Speech Processing Workshop (NOLISP'03)*, 2003.

[131] A. C. Singer, G. W. Wornell, and A. V. Oppenheim. Codebook prediction : a nonlinear signal modeling paradigm. In *Proceedings of International Conference on Signal and Speech Processing*, volume 5, pages 325–328, 1992.

[132] S. Sivadas and H. Hermansky. Hierarchical tandem feature extraction. In *Proceedings of International Conference on Signal and Speech Processing (ICASSP)*, volume 1, pages 809–812, 2002.

[133] H. Strik. Automatic parametrization of differentied glottal flow : comparing methods by means of synthetic flow pulses. *Journal of Acoustic American society*, 5 :2659–2669, may 1998.

[134] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. Farmer. Testing for nonlinearity in time series : the method of surrogate data. *Physica D*, 58 :77–94, 1992.

[135] J. Thyssen, H. Nielsen, and S. D. Hansen. Non-linear short-term prediction in speech coding. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 185–188, 1994.

[136] N. Tishby. A dynamical system approach to speech processing. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 365–368, 1990.

[137] B. Townshend. Non linear prediction of speech. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 425–428, 1991.

[138] A. H. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustic, Speech, and Signal processing*, 37(3) :328–339, march 1989.

[139] X. Wang and K. D. O'Shaughnessy. Improving the efficiency of automatic speech recognition by feature transformation and dimensionality reduction. In *EUROSPEECH*, 2003.

[140] A. R. Webb. Functional approximation by feed-forward networks : a least square approach to generalisation. *IEEE Transactions on Neural Networks*, 5(3) :363–371, 1994.

[141] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart. Predicting the future : a connectionist approach. *International journal of neural systems*, 1(3) :193–209, 1990.

[142] S.J. Young. The general use of tying in phoneme-based hmm speech recognisers. In *IEEE International Conference on Speech and Signal Processing (ICASSP'92)*, volume 1, pages 569–572, 1992.

[143] D. Yuk and J. Flanagan. Telephone speech recognition using neural networks and hidden markov models. In *IEEE International Conference on Acoustic Speech and Signal Processing*, volume 1, pages 157–160, 1999.

[144] S. A. Zahorian, D. Qian, and A.J. Jagharghi. Acoustic-phonetic transformations for improved speaker-independent isolated word recognition. In *Preceedings of International Conference on Signal and Speech Processing*, volume 1, pages 561–564, 1991.

[145] J. L. Zarader. *Méthodes adaptatives et neuronales pour le traitement du signal : Application aux signaux lidar et de parole (HDR)*. PhD thesis, Université Paris VI, 2000.

[146] J. L. Zarader, G. Ancellet, A. Dabas, N. K. M'Sirdi, and P. H. Flamant. Performance of an adaptive notch filter for spectral analysis of coherent lidar signals. *International Journal of Atmospheric and Oceanic Technology (IJAOT)*, 13(2) :16–28, 1996.

[147] J. L. Zarader, A. Dabas, P. H. Flamant, B. Gas, and O. Adam. Adaptive parametric algorithmes for processing coherent doppler lidar signal. *IEEE Transaction on Geosciences and Remote Sensors*, 37(6) :2678–2691, 1999.

[148] J. L. zarader, B. Gas, D. Charlelie-Nelson, and C. Chavy. New compression and decompression of speech signals by npc. In *SSIP'01 (International Conference on Signal, Speech and Image Processing)*, pages 1521–1526, 2001.

[149] J. L. zarader, B. Gas, J. C. Didiot, and P. Sellem. Neural predictive coding : application to phoneme recognition. In *ICONIP'97 (International Conference On Neural Information Processing)*, 1997.

[150] J. L. zarader, B. Gas, P. Drobinski, and P. Dabas. Doppler frequency estimation and quality control by neural networks. In *CLRC'99 (Coherent Laser Radar Conference)*, 1999.

Deuxième partie

Recueil de publications

.

Liste des publications en annexe

.

A.1 Article de revue [66]

A New Approach to Speech Coding : The Neural Predictive Coding

B. GAS and J.L. ZARADER and C. CHAVY

Journal of Advanced Computational Intelligence, Vol 4, p. 120–127 (2000)

.

A new approach to speech coding: the Neural Predictive Coding

B. Gas, J.L. Zarader, C. Chavy
Laboratoire des Instruments et Systèmes, Université Paris VI
4 place Jussieu, 75252 Paris cedex 05, FRANCE

October 12, 2000

Abstract

In this article we propose a new speech signal coding model applied to the recognition of phonemes. This model is an extension to the non linear area of adaptive coding systems used in speech processing. For this purpose, we use predictive connectionist methods. We show that it is possible to take into account class membership information of the phonemes from the stage of coding. To evaluate the NPC encoder, a study of a database of phonemes by discriminant analysis and an application to phonemes recognition are carried out. Simulations presented here show that classification has obviously been improved, compared to currently used types of coding.

1 Introduction

1.1 Speech coding

In applications such as speech recognition, the main goal of the coding process is to extract a maximum of interesting signal features while reducing the amount of data to a minimum. The selection of the representation of signals non only depends on its intrinsic qualities but also on the recogniser system (HMM [9], [19], and neural networks [15], [14], [22], [16] for instance) and on the application goal (speaker, speech or langage recognition, ...). Two main classes of coding methods can be identified. Those obtained in the temporal domain from a voice-producing model and those issued from the frequential domain corresponding to a voice-listening model.

A large number of representations in the temporal domain have been proposed. One of the most commonly used is issued from the LPC model (Linear Predictive Coding) [12], [20]. Other representations also exist, but they are mostly derived from the computation of LPC coefficients, such as the PARCOR ones which afford the advantage of being within -1 and +1, when the filter is stable. One also finds the LAR coefficients (Log Aera Ratio) and the LPCC coefficients (Linear Predictive Cepstrum Coding), which are in fact a cepstral representation of signal but are computed using LPC coefficients.

In the frequential domain, most features extraction methods are based on a speech physiological audition model. They focus on improving low frequencies $[0, 1500Hz]$ much more than higher frequencies. The most frequent representation uses the Mel scale like the MFCC Coding (Mel Frequency Cepstral Coefficients) [3], [28].

We carried out several phonemes recognition tests using these various methods. We present them further in this paper. Results given hereunder show a very moderate performance. In particular, recognition rates measured on our test set (generalisation scores) never exceed 65 %. We know from linguistic type experiences that the human ear is able to reach recognition rates close to 85 % [10] without taking into account contextual information of superior levels. The weak discriminative capacities and the insufficient robustness of the present encoders justify that we should improve coding techniques, or even create new coding methods. That is what we propose by presenting the NPC coding in this article. We focus on enhancing speech features extraction by using information coming from the training databases: that is to adapt the encoder to the application task.

The improvement of the coding discriminant capacities has already been the subject of research work ([24], [6],[5]). The idea to contribute to the decision (most often a pattern recognition task) from the coding stage is not new. Other research work carried out along the same lines can be found. For the most part it is inspired from the MCE/GPD theory proposed by Juang and Katagiri [13] in 1992. The authors formalized the selection of the representation space problem, adapted to the classification task. They called their method "Discriminant Feature Extraction" (DFE). Biem and Katagiri [1] applied it to the filter bank parametric representation and also to the cepstral filtering [2]. Bachianni *and al.* have also proposed to use DFE techniques for the time-frequency masking filters optimisation.

In general, DFE methods lead to bring together features extractor and classifiers into a same global module. For such approaches, the training algorithms of the classifier and of the encoder are dependent or even merged. De la Torre *and all* [4] have proposed variations of this method which allow independent training of both the classifier and the encoder. Our NPC encoder is based on this idea. This is why we present in this paper the coding problem in details while the classifier used is only presently briefly: this allows us to make comparisons between several different coding methods.

New processing tools have appeared since the first encoders based on signal processing were proposed. In non linear processing area, one finds connectionist techniques. In the eighties, research teams [8], [16] underlined the natural contribution of neurone systems to the non linear adaptive filtering. New methods of data coding can be easily inferred from this [25], [26],[27]. Authors as Lapedes [17] and Widrow [23] have considered in this way a generalisation of the linear adaptive filtering to neural systems. As systems of neurones are also used as classifiers, they have the dual capacity of achieving prediction and classification simultaneously. With such models it becomes possible to bring together the decision and coding stages. Advantages of neural networks are clear: they allow taking into account the non-linear features of the signal. However, their application to speech coding may create problems as underlined by Thyssen [25]. One of the main problems is the very large quantity of parameters (the network synaptic weights). The method we propose here brings a solution to this problem and allows consequently simple use of

connectionist methods applied to speech coding.

1.2 Scope of the article

In a first part, we will discuss in details both the problem of the speech signal coding and the NPC encoder proposed. We will justify our approach and some points of discussion are presented. In a second part, experiments on usual speech coding methods are conducted ; this allows comparisons with NPC. Finally, the last part is devoted to classification rates obtained with NPC coding compared to the most commonly used coding methods.

2 The Neural Predictive Coding model

In the present systems, the low scores obtained in phoneme classification are most often compensated by the use of contextual information coming from the superior levels of decision. One can however note that today, the current systems give good results but in limited environments, such as for instance the mono-speaker mode, or in unionised environment conditions or for the non spontaneous speech.

Let us consider for our purpose the problem of the adaptive filtering method applied to the coding of phonemes. The goal of the well-known LPC coding consists in determining the coefficients of a linear predictive filter. For this, an optimisation criterion based on minimising the prediction error needs to be defined. Linear filters cannot be well adapted to modelling of non linear processes or for the estimation of non Gaussian signals. Neural networks can be considered as a new family of non linear filters. Learning algorithms allowing the minimisation of an error criterion have been proposed as the back-propagation algorithm [18], [21]. They adapt therefore very well to the problem of the speech signal coding. It has been shown that neural networks with one hidden layer are universal approximators [11]. This leads to a more sophisticated encoder structure, as shown in the following section. This is however interesting for the non linear process modelling.

2.1 Neural network weights as coding parameters

One very simple neural coding solution is to imagine a neural network used as a non-linear predictive filter (Non Linear Auto-Regressive filter, NLAR) for any phoneme to be coded. The number of neurones of the hidden layer and the size (number of samples) of the input prediction window will determine the total number of weights of the system. At the end of learning, that is when a prediction error threshold has occurred, one can consider that network weights are representative of the learned phoneme. We can then consider them as components of a features vector. We are confronted with the problem of the explosion of the code dimension [25]. Let us suppose a structure comprising 6 hidden cells and 12 inputs. By counting weights and thresholds, one obtains $12 \times 6 + 6 + 6 \times 1 + 1 = 85$ coding coefficients of the signal. This is considerable compared to the number of parameters commonly used (about 40 to 50 parameters). The answer we propose is to consider only the second layer weights as the phoneme coding parameters. This is the main central idea

of our research work. This leads to an important modification in the network structure but the computing method (quadratic criterion minimisation) remains the same.

2.2 NPC Learning and coding

Now, let us consider a neural network with two layers, as shown in figure 1. The first layer (the hidden layer) is composed of a given number of cells. This number corresponds to the number of coding coefficients we want to generate. The second layer is composed of one single cell: the *prediction cell*. The key idea is that the first layer input weights are shared by all the phonemes, while the second layer weights are specific to each phonemes. During learning, the first layer weights are updated so as to minimise the prediction error on all the database phonemes. The second layer weights are particular to a given phoneme: they are updated to minimise the prediction error on this phoneme. As a consequence, there are as many second weight layers as there are phonemes belonging to the learning database. The expected behaviour of such a structure is the coding of the common prediction information by the first weight layer only, while the coding of prediction information specific to each phonemes (we will call them *discriminative features*) is achieved by the second layer weights.

Such a coding process requires the following two computing phases:

- A first stage which is called the *parameters learning phase* or *parameter setting phase*: it consists in learning the first layer weights. We thus obtain the *encoder parameters*;
- A second stage which is called the *coding phase*: the first layer weights are initialised with the weights value drawn from the first learning phase and they will remain fixed. The only weights to be learnt are the second layer weights, which become the *phonemes parameters* or *coding parameters* of the phoneme.

All network weights need to be computed in the first learning phase: the first layer weights and every phoneme second layer weights. The criterion to be minimised is the quadratic prediction error computed on the set of examples of the basis. Obtaining a common weights layer for all these examples leads to a greater difficulty for convergence, that is costly in terms of computation time. Thus the total prediction error, although continuously decreasing, never tends to 0 (see figure 8 and figure 14).

The coding phase is faster than the learning phase (first layer weights are no longer modified). The first layer acts as a multi-dimensional non-linear filter. The prediction error depends on each phoneme (the phonemes being taken separately). As a result, convergence is far more rapid. The learning rule acts as a modified Madaline Rule III (MRIII) [23].

2.3 Normalisation constraint

A simple extension of the model allows to obtain a set of normalized codes. Indeed, in absolute terms, learning does not impose limit values to computed weights. To avoid a great variance of the generated coefficients, we propose to add a constraint on weights

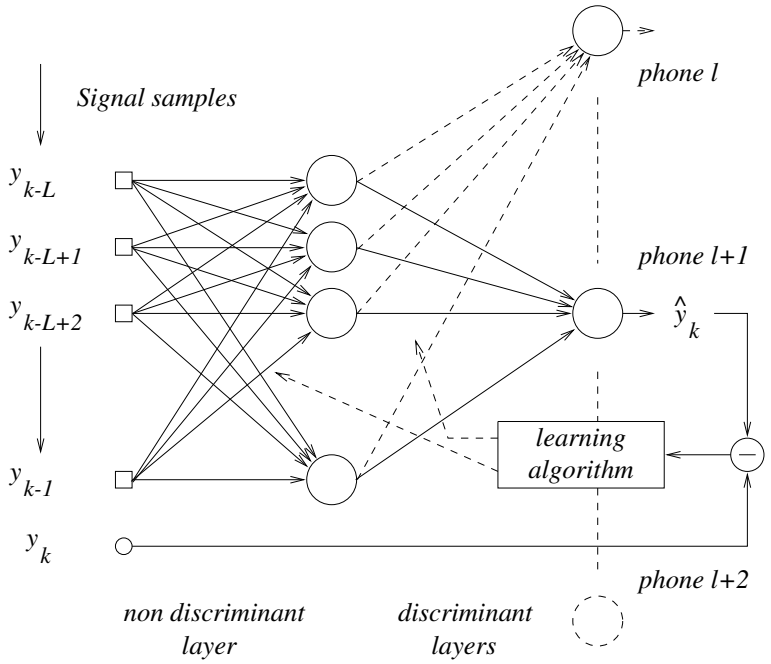


Figure 1: Structure of the NPC encoder.

guaranteeing the evolution of their values into a predefined interval. The two advantages of this approach are, on one hand to avoid a possible explosion of the weight values and on the other hand, to generate a set of normalized coefficients between -1 and 1 for example.

3 Formal description of the NPC encoder

We propose to formally describe our model in this section. Let us recall that a transversal filter performs a weighted sum of its inputs: $\hat{y}_k = \mathbf{a}^\top \mathbf{y}_k$ with $\mathbf{y}_k = [y_{k-1}, y_{k-2}, \ldots, y_{k-L}]^\top$ and $\mathbf{a} = [\ldots, a_i, \ldots]^\top$. $\hat{y}_k$ is the signal prediction and $\mathbf{y}_k$ the vector of the L last samples of the signal y_k. The vector $\mathbf{a}$ components are the filter parameters to be determined.

A formal *neurone* computes an *activation function* of the weighted sum $y_k = \sigma(\mathbf{a}^\top \mathbf{y}_k)$. Therefore, it can be compared with a transversal filter, except that the activation function is usually not linear. Let us now extend this to a three layer MLP model (one hidden layer) with one output cell (the prediction cell). It performs the following output function: $y_k = \sigma(\mathbf{a}^\top \mathbf{z})$ where $\mathbf{z} = [.., \sigma(\omega_i \mathbf{y}_k), ..]^\top$ denotes the output of the first layer cells vector. Identifying the model consists in estimating the non linear filters parameters, i.e. the set of the first layer weights $\Omega = [\ldots, \omega_i, \ldots]^\top$ and the set of the output cell weights $\mathbf{a} = [\ldots, a_i, \ldots]^\top$.

In the NPC model case, the Ω weights are the encoder parameters. They must be computed once and once only: this is the *parameter setting phase*. Coding an unknown phoneme consists in identifying the model by estimating the second layer weights while the first layer weight acts as a multi-dimensional filter.

The network architecture is described in figure 1. Let M be the size of the coding

vector. The hidden layer is composed of $M-1$ cells. The weights between this hidden layer and the prediction cell are the first $M-1$ vector components, the output cell bias being th M^{th}. Each phoneme frame of size N (N samples) is composed of $N-L+1$ training examples. L denotes the predictive window width: the number of the previous samples used to predict the next sample. Let S_ϕ be the examples set belonging to the phoneme ϕ:

$$S_\phi = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{N-L+1}\} \quad \text{with} \quad \mathbf{y}_k = [y_{k-1}, y_{k-2}, \ldots, y_{k-L}]^\top \tag{1}$$

For any example $\mathbf{y}_k$, the neural network is trained to predict the next signal value y_k, denoted $\hat{y}_k$. The global quadratic criterion $\mathcal{L}$ to minimise is given by:

$$\mathcal{L} = \sum_{k=1}^{N-L+1} \epsilon_k^2, \tag{2}$$

where $\epsilon_k = y_k - \hat{y}_k$ is the prediction error. This quadratic form must be extended to all the examples in the whole database. Let us now remember that the signal prediction corresponding to the phoneme ϕ is given by:

$$\hat{y}_k = \sigma(\mathcal{V}_k) = \sigma\left(\sum_{i=1}^{M-1} \psi(a_i^\phi) z_{i,k} + \theta\right) \tag{3}$$

where σ is the sigmoid activation function. ψ is the weights limitation function: it can be the sigmoid function itself, for example, which leads the weights to remain within the $[-1,+1]$ interval. The a_i^ϕ are the coding vector components of the phone ϕ. Outputs of the first layer cells are:

$$z_{i,k} = \sigma(\mathcal{V}_{i,k}) = \sigma\left(\sum_{j=1}^{L} \omega_{ij} y_{k-j} + \theta_i\right) \tag{4}$$

The modification rule of the weight a_i^ϕ, which is the i^{th} phoneme ϕ coding parameter, is simply derived from the usual one (see the backpropagation algorithm [18]):

$$\Delta a_i^\phi = 2\lambda \frac{\partial \psi}{\partial a}(a_i^\phi) \sum_{k=1}^{N-L+1} \epsilon_k \frac{\partial \sigma}{\partial \mathcal{V}}(\mathcal{V}_k) z_{i,k} \tag{5}$$

where λ is the learning rate. Back-propagation algorithm leads to the following first layer weights modification rule:

$$\Delta \omega_{ij}^\phi = 2\lambda \psi(a_i^\phi) \sum_{k=1}^{N-D+1} \epsilon_k \frac{\partial \sigma}{\partial \mathcal{V}}(\mathcal{V}_k) \frac{\partial \sigma}{\partial \mathcal{V}}(\mathcal{V}_{i,k}) y_{k-j} \tag{6}$$

The global modification rules for a set of phonemes $\{\phi\}$ are:

$$\begin{cases} a_i^\phi \longleftarrow a_i^\phi + \Delta a_i^\phi \\ \omega_{ij} \longleftarrow \omega_{ij} + \frac{1}{|\{\phi\}|} \sum_{\{\phi\}} \Delta \omega_{ij}^\phi \end{cases} \tag{7}$$

After the learning phase, the first weights layer are memorised as the encoder parameters but the phonemes parameters a_i^ϕ are no longer used. They have to be computed again during the coding phase. This coding phase is a special case of the learning phase previously described: the first layer weights (the encoder parameters) remain fixed. The only weights to be learnt are the phonemes coefficients a_i^ϕ. Their final values are the phoneme coding.

4 Traditional coding methods

In this paragraph we will present simulations resulting from traditional coding methods. These results will enable us to make comparisons with the NPC coding.

We built an 8-phoneme base using the Darpa-Timit [7] database. This database is composed of speakers speaking 10 different dialects of the United States. We took into account all speakers speaking the same dialect. The phonemes chosen are among the most used: /*a*/, /*ae*/, /*ey*/, /*ow*/, /*ix*/, /*iy*/ (vowels) and /*s*/,/*z*/ (fricatives). This base is constituted by 100 examples per phoneme class. To select phonemes for each particular class, we checked the following conditions:

- Every phoneme, according to its duration, is divided into windows of a fixed length (256 samples), each of them being a phoneme example. All the examples chosen for our learning and test basis belong to distinct phonemes .
- Examples are chosen randomly among all speakers of the same dialect so as to model a multi speaker environment.

Among temporal coding methods, we considered the *LPCC* coding (Linear Predictive Cepstrum Coding) and the *LAR* coding (Log Aera Ratio). These coding methods are derived from the well known *LPC* coding (Linear Predictive Coding), also presented.

Concerning spectral coding, we considered the *FFT* coding (Fast Fourier Transform): average power measured on filters banks, then the *Cepstrum* coding and the *MFCC* coding (Mel Frequency Cepstrum Coding). This last coding reproduces the signal spectrum with a scale of frequencies based on the human ear scale, also called Mel frequency scale. It is among the most commonly used methods for speech processing because of its robustness.

Figure 2 gives the diagram used to test these different types of codings. The encoders used provide a set of representative coefficients for each basis phonemes (12 coefficients).

4.1 Comparison by discriminant analysis

The discriminant analysis allows, by a judicious projection of examples on a 2D subspace, to obtain a visual representation of data clustering. Thus one can have a first idea of the confusions that occur between classes. Ellipses drawn on figures are centred on the centroids of classes. Surfaces circumscribed measure the display of examples around their centroids. Figures 3 to 5 represent the projection of the base examples after phoneme coding using six different algoritms. On figure 3 we can see temporal coding methods: LPC

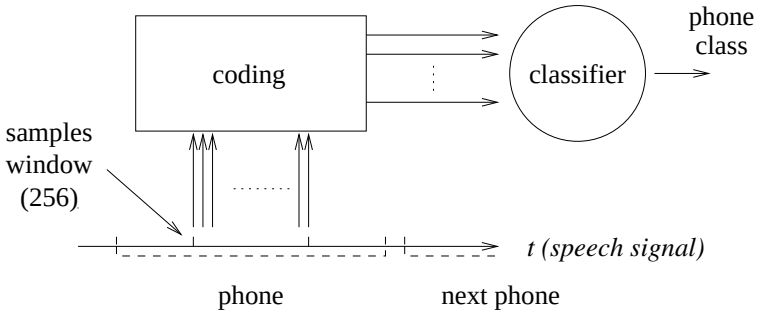


Figure 2: Diagram of the phoneme classifier.

coding and LAR coding. On figures 4 and 5 are presented frequential coding methods: LPCC coding, FFT (filters banks), MFCC and Cepstral coding.

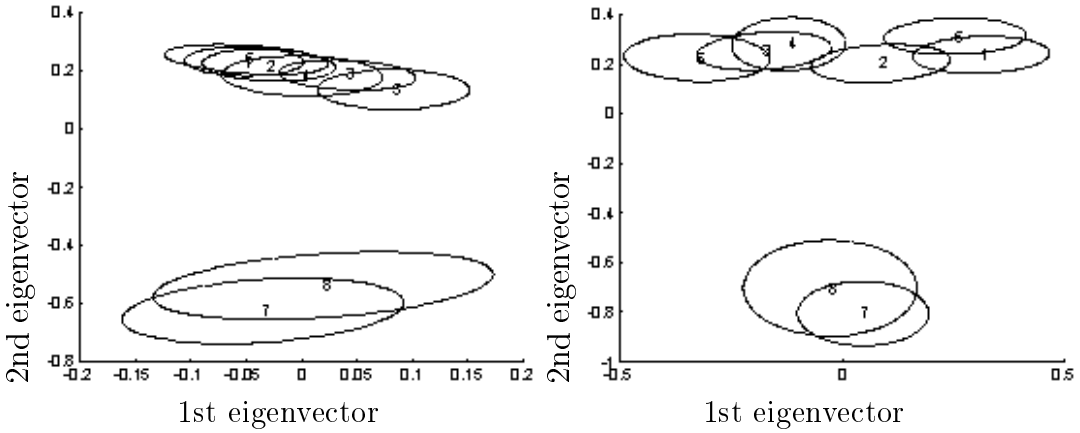


Figure 3: LPC coding (LH side) and LAR coding (RH side).

One can note that for nearly all of the coding methods a separation of the phonemes in two different clusters is made. On the one hand, phonemes 1 to 6 (/*a*/, /*ae*/, /*ey*/, /*ow*/, /*ix*/, /*iy*/) and on the other hand phonemes 7 and 8 (/*s*/ and /*z*/). This separation seems natural due to the fact that these two subsets of phonemes have very different acoustic features. Spectral studies show that the *s* and *z* phonemes (fricatives) hold the high frequency part of the spectrum while the other six phonemes (vowels) occupy a much lower frequency part of the spectrum.

The separation made by MFCC coding appears clearly here. We distinguish three clusters of classes. The set of the vowel phonemes is made of two subclasses phonemes: /*a*/, /*ae*/, /*iy*/ on one side and /*ey*/, /*ow*/, /*ix*/ on the other side.

These results are qualitative. But they already show the nature of the confusions that occur between phoneme classes when coding them. They have to be completed by more quantitative measures as we propose to do in the following paragraph.

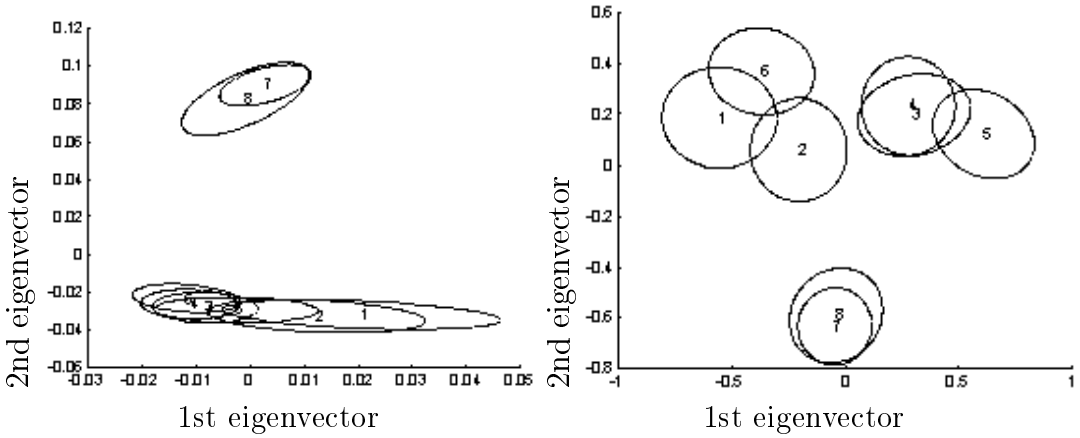


Figure 4: LPCC coding (LH side) and MFCC (RH side).

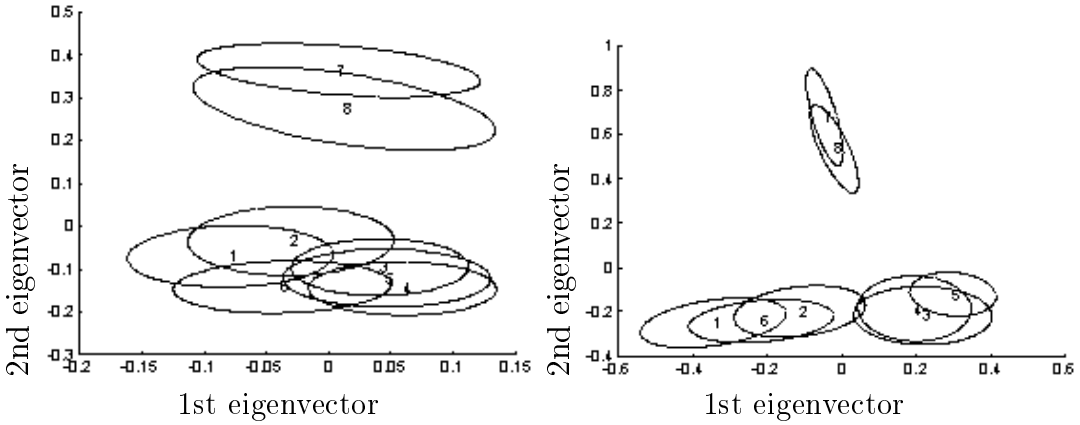


Figure 5: Cepstrum coding (LH side) and FFT (filters banks) coding (RH side).

4.2 Classification by computation of distances

We present here more quantitative results obtained from the discriminant analysis previously described. Co-ordinates of the 2D subspace represent now the phonemes. The eight classes are represented by their centroids and the classification of an unknown example is obtained by computing its distance to these centroids. Such a classification acts as a 1-ppv classifier on the 2D projection space. figures 6 and 7 show results obtained by using respectively the Euclidean distance and the Mahalanobis distance.

One can note on these figures a gap between the different coding methods. Therefore these gaps show the influence of coding on the classification results. The temporal LAR coding and the frequential MFCC coding give the best results with, in general, a superiority of the MFCC coding. On the other hand, changing the distance does not modify the scores noticeably, nor the relative differences between coding methods. This can be explained by the fact that the within-class covariance used by the Mahalanobis distance is already taking into account by the discriminant projection and therefore by the Euclidean distance. Of course, we cannot generalise these results to the totality of

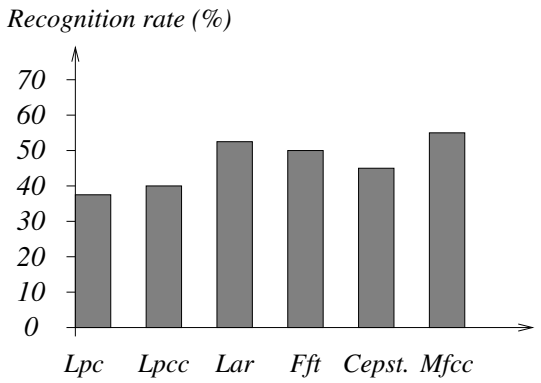


Figure 6: Phonemes classification by Euclidean distance using different types of data coding.

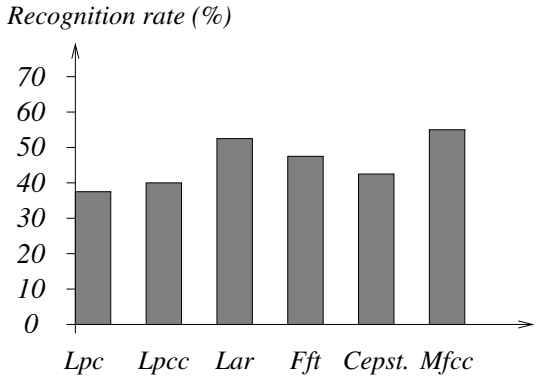


Figure 7: Phonemes classification by Mahalanobis distance for types of data coding.

the English language phonemes. On a small basis like ours, they simply give interesting indications.

The classification implemented remains very simple. One can appreciably improve results obtained by using undeterminist classifiers based on hidden Markov models or connectionist classifiers as neural networks. Let us recall that our purpose is not the classification problem. We focus on comparing NPC coding with the mainly used coding methods by using a particular application: the phonemes recognition.

5 NPC encoder evaluation

The aim of the application proposed here is to identify one phoneme out of eight, which are /*aa*/, /*ae*/, /*ix*/, /*iy*/, /*ow*/, /*z*/, /*s*/. The use of the NPC encoder for applications as phonemes recognition implies the necessary computation of NPC parameters, which are the first layer weights. Then comes the data coding stage itself which is the second computation step of the coding.

Concerning the tests described in this paper, the encoder parameters training is per-

formed with 800 phonemes from the constituted database, i.e. 100 phonemes per class. The selection of the 100 phonemes is of great importance: the more the phonemes selection is representative of the phonemes variance, the more the data coding performed will be discriminant. Previously selected phonemes will no longer be used for the coding phase. So, we built two databases, one for the first computation step (100 phonemes per class), and one for the coding step (100 phonemes among the other 700 phonemes). This last basis enables to evaluate the NPC performances. Only these last phonemes are used for classification tests and discriminant analysis.

The size of the input layer is $L = 40$. Therefore the number of signal samples used $(y_{k-1}, y_{k-2}, \ldots, y_{k-40})$ to predict the next signal value $(\hat{y}_k)$ is 40. Each phoneme carries $N - L = 216$ learning examples. The hidden layer has 11 cells so as to perform 12 coding coefficients (comprising the 11 weights and the threshold). Phonemes were successively presented and their corresponding codes (output cell weights) were computed. At the same time, backpropagation rule was used and the hidden layer weights were updated.

The figure 8 gives the example of a prediction signal obtained with a phone */aa/* example of the Darpa-Timit basis, after coding. It shows a relatively faithful tracking of the original signal.

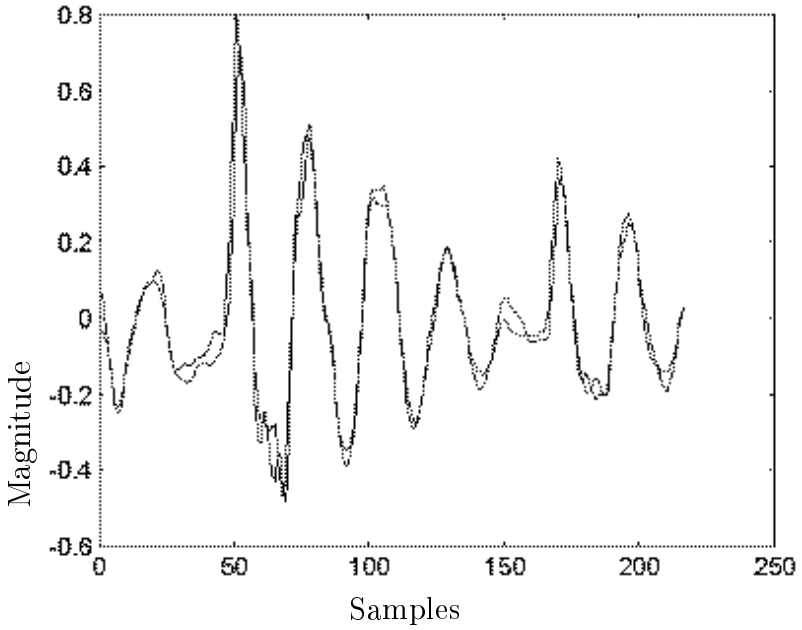


Figure 8: Real signal and predicted signal for one Darpa-Timit phone example (*/aa/* phone.)

Concerning convergence times, it is necessary to count between 800 and 2000 iterations for the parameter setting phase of the model (first layer weights learning). Studies carried out in our laboratory showed that an excess number of learning iterations could be prejudicial, leading to a decay in performance. One can suppose, indeed, that the first layer would begin to retain discriminative information, which would not therefore be

captured by the second layer. For example, the figure 9 gives results obtained by a neural classifier after the eight phonemes coding by a NPC encoder and configured for different learning durations of the first layer. On the contrary, the learning time needed for the

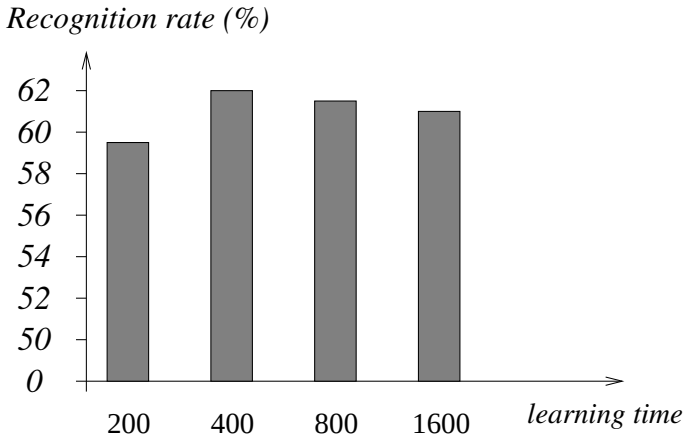


Figure 9: Classification results for different learning times of the first layer.

coding phase (second layer weights learning) is very short. Convergence is obtained between the tenth and the fiftieth learning iterations. In fact, on one hand this is due to the very small number of weights to process and on the other hand to the fact that the learning task concerns each phoneme separately. It can be noted that using stochastic gradient method would make the convergence process even faster.

5.1 NPC evaluation by discriminant analysis

We will show in this section a global comparison of different methods of coding by using discriminant analysis. This was carried out exactly in the same conditions as for the classical coding methods. As stated in previous section, NPC phonemes examples used for projection are not the same as those used for NPC parameters learning.

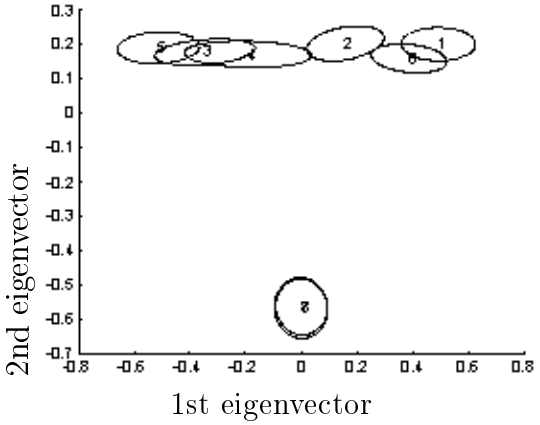


Figure 10: Discriminant analysis for NPC coding.

Classes numbers 1 to 8 respectively correspond to /*aa*/, /*ae*/, /*ey*/, /*ix*/, /*iy*/, /*ow*/, /*s*/, /*z*/ phonemes. Comparison between temporal codings (figs. 3 and 4) and NPC coding (fig. 10) shows a more efficient separation between clusters of NPC coded data, more particularly between 1,2,6 and 3,4,5 vowels. The comparison with MFCC coding projection shows that phonemes 7 and 8 (/*s*/ and /*z*/) are more dissociated than in the NPC projection. Conversely, NPC projection increases separation between classes 3 and 4 (/*ey*/ and /*iy*/). In fact, according to this first analysis, NPC and MFCC give comparable scores. We therefore completed this study with more efficient methods like the MLP classification.

5.2 NPC evaluation using MLP classifiers

In this paragraph we resume the study of the phonemes coding. The classifier used is a network of formal neurones. It is a feedforward Multi-Layer perceptron (MLP) with six cells on the hidden layer and eight cells on the ouput layer. The learning rule applied is the gradient descent using the error backpropagation algorithm.

For the learning phase it is necessary to divide the initial set of examples into two bases. One is called the *learning set* (80% of the initial base examples) and the other the *test set* (the last 20% examples of the initial base). This second test base allows us to measure the classification performance on examples that have not been learnt. We can then obtain a measure of the classifier generalisation capabilities. Of course, our aim is to test the possible impacts of the NPC encoder on the data classification and not the modification of the classifier itself. One can see on figure 10 the 800 phonemes

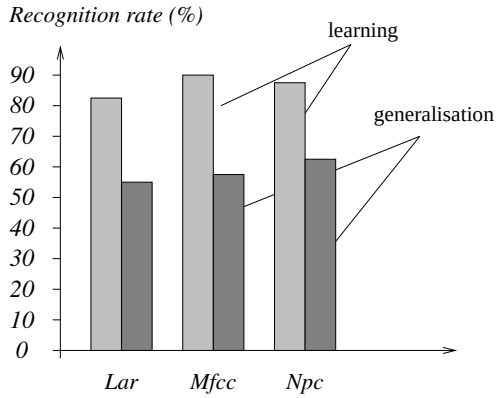


Figure 11: Recognition rates obtained with MLP classifier for /*aa*/, /*ae*/, /*ey*/, /*ow*/, /*ix*/, /*iy*/, /*s*/ and /*z*/ phonemes (learning and generalisation).

projection on the first two eigen vectors subspace (the discriminant analysis previously presented). NPC phonemes used for projection are not the same as those used for the parameters learning. One can see on figure 11 comparisons between recognition rates obtained with a MLP classifier, grey bars representing the training set and the dark grey bars the generalisation set. Recognition rates have been obtained after 30000 learning iterations. NPC coded data give better results in generalisation (61% for NPC, 56% for MFCC and near 54% for LAR) and so corroborate discriminant analysis study.

5.3 Evaluation of the NPC coding on *b-d-g* and *p-t-q* phonemes

Phonemes /*b*/, /*d*/, /*g*/ and /*p*/, /*t*/, /*q*/ are of particular interest for the valuation of coding methods in speech recognition. They frequently appear in the English language and their identification is considered to be difficult. In particular they are composed of non linear features because they belong to the set of the *occlusive* phonemes.

For example, those phonemes have been used by Lang and Waibel in [22],[16] to validate their model (the Time Delay Neural Network, TDNN) applied to speech signal treatment. We present in this paper a set of simulations for the recognition on the one hand of the /*p*/,/*t*/,/*q*/ phonemes and on the other hand of the /*b*/,/*d*/,/*g*/ phonemes. Figures 12 and 13 show recognition results obtained for these two subsets of phonemes.

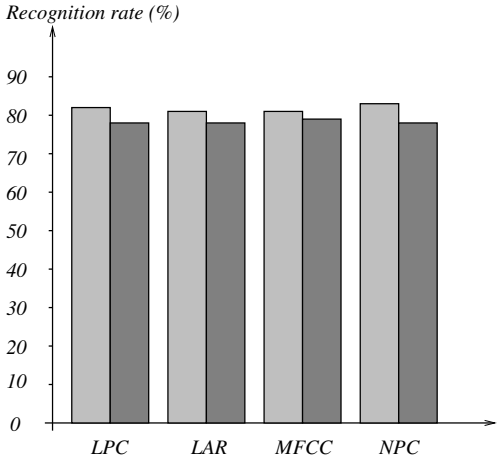


Figure 12: Recognition rates obtained with MLP classifier for /*p*/,/*t*/,/*q*/ phonemes.

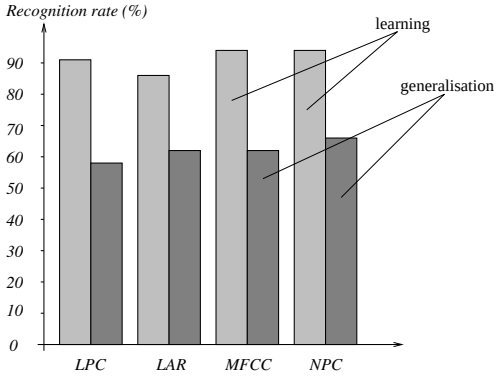


Figure 13: Recognition rates obtained with MLP classifier for /*b*/,/*d*/,/*g*/ phonemes.

One can note that the results are roughly equivalent for all coding methods used (LPC, LAR, MFCC et NPC) as regards the /*p*/,/*t*/,/*q*/ phonemes. On the other hand, the NPC coding method gives better scores for the /*b*/,/*d*/,/*g*/ phonemes (over 65% recognition rate against 62% for the MFCC coding). This results are consistent with the fact that

phonemes /b/,/d/,/g/ are *voiced* phonemes while /p/,/t/,/q/ phonemes are not. The voiced feature emphasises the predictible nature of the phonemes and so gives temporal coding methods (LPC, LAR et NPC) an advantage. The non linear features present in the speech signal are then better taking into account by the NPC coding. The figure 14 puts in evidence the consequences of the voiced nature of phonemes /b/,/d/,/g/. The prediction error computed during the NPC parameters learning phase is lower in the case of /b/,/d/,/g/ phonemes than in the /p/,/t/,/q/ phoneme case.

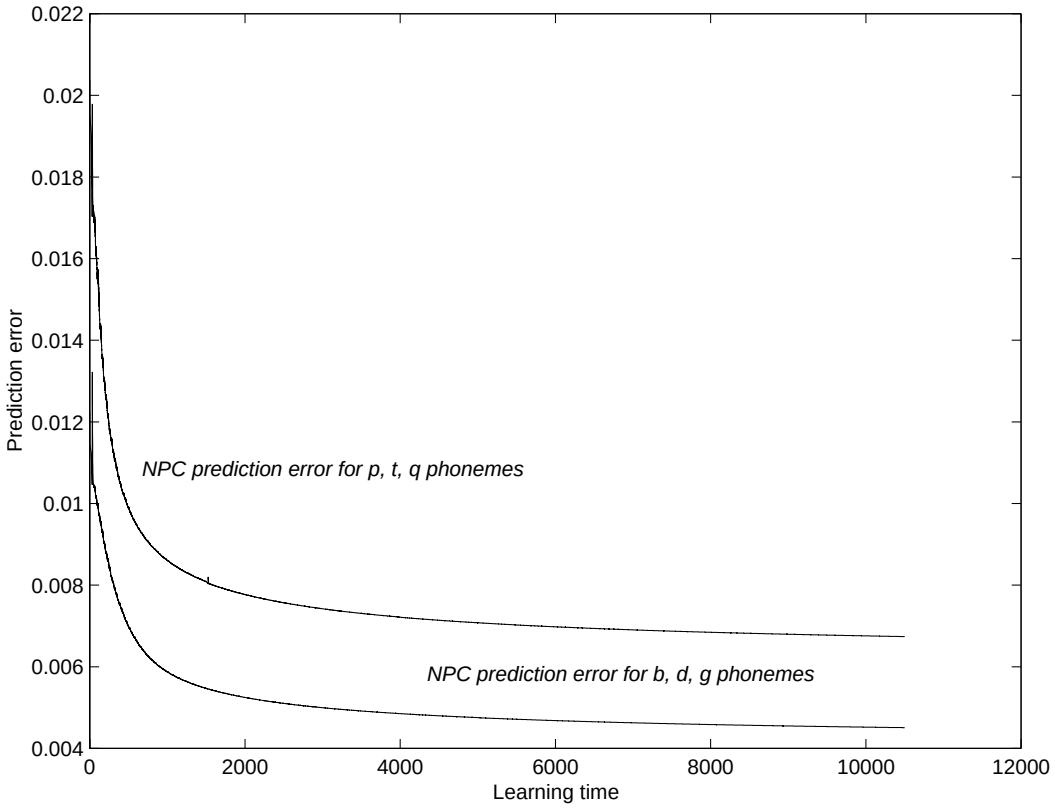


Figure 14: Prediction error during NPC parameter learning phase for /p/,/t/,/q/ and /b/,/d/,/g/ phonemes.

6 Conclusions and future prospects

We presented in this article the first results obtained using the Neural Predictive Coding method. The main advantage of the proposed NPC coding is to integrate database information since the encoder training stage is performed from the application examples. This leads to an encoder adapted to the application task.

We showed that we obtain better results for classification by using a qualitative analysis (Discriminant analysis) and a simple classification approach (MLP classifiers). Another advantage is the possibility we have to produce normalised coding coefficients so that it

will no longer be necessary to standardise the coded data at the next step of the speech processing.

We have seen that the number of learning iterations of the parameters model (first weights layer learning) does not seem to have a great impact on performance. This point will certainly require further research work, particularly from the theoretical point of view. Studies are presently under way.

References

[1] A. Biem and S. Katagiri. Feature extraction based on minimum classification error/ generalized probabilistic descent method. In *Proceedings of International Conference on Signal and Speech Processing*, volume 2, pages 275–278, 1993.

[2] A. Biem and S. Katagiri. Filter bank design based on discriminative feature extraction. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 485–488, 1994.

[3] S. B. Davis and P. Melmerstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 28(4):357–376, 1980.

[4] A. de la Torre, A. M. Peinado, A. J. Rubio, V. E. Sánchez, and J. E. Díaz. An application of minimum classification error to feature space transformations for speech recognition. *Speech Communication*, 20:273–290, 1996.

[5] T. Kawahara and S. Doshita. Phoneme recognition by combining discriminant analysis and hmm. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 557–560, 1991.

[6] T. Kawahara, T. Ogawa, S. Kitazawa, and S. Doshita. Phoneme recognition by combining bayesian linear discriminations of selected pairs of classes. In *Proceedings of International Conference on Signal and Speech Processing*, page 78, 1990.

[7] University of Pennsylvania Linguistic Data Consortium. Darpa-timit : a multi speakers data base.

[8] G. Dreyfus, O. Macchi, S. Marcos, O. Nerrand, L. Personnaz, Roussel-Ragot, D. Urbani, and C. Vignat. Adaptive training of feedback neural networks for non linear filtering. *Neural Networks for signal processing*, 2:550–559, 1992.

[9] K. S. Fu. *Syntactic Methods in Pattern Recognition.* Academic Press, 1974.

[10] J. P. Haton. *Knowledge-based Approaches in Acoustic-Phonetic Decoding of Speech.* Springer Verlag, 1987.

[11] K. Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[12] F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 23:67–72, 1975.

[13] B. H. Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Transactions on Signal Processing*, 40(12):3043–3054, december 1992.

[14] T. Kohonen. The 'neural' phonetic typewriter. *IEEE Computer*, 21(3), 1988.

[15] T. Kohonen and al. Phonotopic maps insightful representation of phonological features for speech recognition. In *Proceedings of International Conference on Pattern Recognition*, volume 7, 1984.

[16] K. J. Lang, A. H. Waibel, and G.E. Hinton. A time-delay neural network architecture for isolated word recognition. *Neural Networks*, 3:23–43, 1990.

[17] A. Lapedes and R. Farber. Nonlinear signal processing using neural networks: Prediction and system modelling. *Internal Report, Los Alamos National Laboratory*, july 1987.

[18] Y. Lecun. *Modèles connexionnistes de l'apprentisssages.* PhD thesis, Paris XI, 1987.

[19] S. E. Levinson. Structural methods in automatic speech recognition. *Procedings of the IEEE*, 73(11):1625–1650, july 1987.

[20] J. D. Markel. *Linear prediction of speech.* Springer Verlag, Berlin, Heidelberg, NewYork, 1976.

[21] D. E. Rumelhart, G. E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[22] A. H. Waibel, T. Hanazawa, G.E. Hinton, K. Shikano, and K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustic, Speech, and Signal processing*, 37(3):328–339, march 1989.

[23] B. Widrow. 30 years of adaptative neural networks: Perceptron, madaline and backpropagation. *Proc. of the IEEE*, 78:1415–1442, 1990.

[24] S. A. Zahorian, D. Qian, and A.J. Jagharghi. Acoustic-phonetic transformations for improved speaker-independent isolated word recognition. In *Preceedings of International Conference on Signal and Speech Processing*, volume 1, pages 561–564, 1991.

[25] J. Thyssen, H. Nielsen, and S. D. Hansen. Non-linear short-term prediction in speech coding. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 185–188, 1994.

[26] N. Tishby. A dynamical system approach to speech processing. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 365–368, 1990.

[27] B. Townshend. Non linear prediction os speech. In *Proceedings of International Conference on Signal and Speech Processing*, volume 1, pages 425–428, 1991.

[28] H. Wassner and G. Chollet. New time-frequency derived cepstral coefficients for automatic speech recognition. In *EUSIPCO'96*, 1996.

A.2 Article de conférence [148]

New Compression and Decompression of Speech Signals by a Neural Predictive Coding (NPC)

J. L. ZARADER and B. GAS and D. CHARLELIE-NELSON and C. CHAVY

International Conference on Signal, Speech and Image Processing (SSIP'01), p. 1521–1526 (2001)

.

New compression and decompression of speech signals by a Neural Predictive Coding (NPC)

J.L. ZARADER, B. GAS, C. CHAVY, D. CHARLES ELIE NELSON
Laboratoire des Instruments et Systèmes d'Ile de France (LISIF)
Université Pierre et Marie CURIE
BP 164, Tour 22-12, 2ème étage,
4 Place Jussieu, 75252 Paris Cedex 05
FRANCE
zarader@ccr.jussieu.fr, gas@ccr.jussieu.fr, chavy@ccr.jussieu.fr, dcharl01@hotmail.com

Abstract: - In this paper, we present a new method of speech compression and decompression based on a Neural Predictive Coding of speech signals. The NPC system is designed to predict the samples of a speech signal window from previous ones. In the coder/decoder that we proposed the transmitted data is computed from the prediction error of the NPC (difference between the sample and its corresponding prediction calculated by the NPC).

The initial goal of the NPC is to extract the signal discriminative features relative to the database which it is extracted. After a precise description of NPC coding, we discuss about the first phase of the algorithm: the adjustment of the parameters of the coder. Then we explain the compression and decompression algorithms. To finish, we present an example and some results on this technic of compression.

Key-Words: - Speech compression, Speech decompression, Neural Networks, Speech Coding, Speech Prediction.

1 Introduction

Speech transmission and storage constitute an important field of research. In these applications, the first stage consists in compressing the speech signal and, more generally, the audio signal. The main goal of this compression is to reduce the rate of the transmission. Of course the decompressed speech signal must be of the same quality that the original speech signal.

In the first part, we will describe the coder NPC. In the second part, we will present the compression and decompression algorithms. Finally, we will discuss about the results of this coding.

2 NPC Presentation

The function of the NPC is to compute a vector of parameters [1] extrated from a frame (20ms) of the speech signal. This vector is a discriminant feature of signal and can be used in an application of speech recognition [2].

The NPC is a non linear predictive coding, so it preserves the non linearities of the signal [3,4]. One problem that occurs with most of the non linear predictive models is that they generate a great number of parameters. So another aim of this NPC coding is to limit this number.

NPC is based on a two layers perceptron. It is trained to predict a signal sample from the previous ones. The key idea is that weights of the second layer are proper to each window, and constitute the coding coefficients, while the weights of the first layer are common to all the windows, and constitute the fixed part of the system : the NPC is a discriminant coder. The pocessing is decomposed in two phases :

- the training phase : is intended to adjust the first layer weights (computation of the fixed part of the coder)
- the coding phase : determination of the parameters representing the speech signal.

2.1 Training phase

We extract a great number of windows of L samples each. Let P be the inputs neuron number, N the neuron number in the hidden layer and $y_i(k)$ the k^{th} sample of the i^{th} window. P is also called the predictor memory. The samples k-P to k-1 of the i^{th} window form the vector :

$$\mathbf{Y}_{k,P}^{i} = \left[y_i(k-P), y_i(k-P+1), ..., y_i(k-1) \right]$$

which is also the prediction window.

A second layer is associated with each window. Let $\mathbf{A}^i$ be the vector of the N weights of the second layer associated with the window i. So there is one first layer

and there are as many second layers as there are windows (see figure 1).

We present the first P samples of a window to the MLP (Multi Layers Perceptron) constituted by the common first layer, and the second layer associated with this window. The neuron outputs of the hidden layer are :

$$\mathbf{X}^i(k) = f\left(\mathbf{W} \cdot Y^i_{k,P} + \mathbf{B}\right) \qquad (1)$$

where f is the activation function, $\mathbf{W}$ the matrix $P \times N$ of the first layer weights, and $\mathbf{B}$ the vector of the N first layer biases.

Then, the prediction of the k th sample of the window i is:

$$\hat{y}_i(k) = f\left(\mathbf{A}^i \cdot \mathbf{X}^i(k)\right) \qquad (2)$$

The MLP is trained to predict the next sample, so the prediction error is :

$$e_i(k) = y_i(k) - \hat{y}_i(k) \qquad (3)$$

The criterion to minimize for the second layer associated with the window i is:

$$J_2^i = \sum_k e_i(k)^2 \qquad (4)$$

I.e. the sum of (3) on all the samples of the window i.

On the other hand, the first common layer is optimized for the prediction of all the samples of all the windows. So for the first layer the criterion to minimize is:

$$J_1 = \sum_i \sum_k e_i(k)^2 \qquad (5)$$

I.e. the sum of (3) on all the samples of all the windows.

We modify weights to minimize these criteria by using the backpropagation algorithm.

k varies from P+1 to L, so each analysis window provides L-P pairs (input vector-target output) for the predictive neural network.

Once this weights optimization is done, we obtain a first layer ($\mathbf{W}$ and $\mathbf{B}$) that constitutes the fixed part of the coding system. $\mathbf{W}$ and $\mathbf{B}$ will be no longer updated. Then, the system is ready to code.

From a connectionist viewpoint, the first layer captures the common information. From a signal viewpoint, the first layer does non linear optimal transformations for the prediction of all the windows.

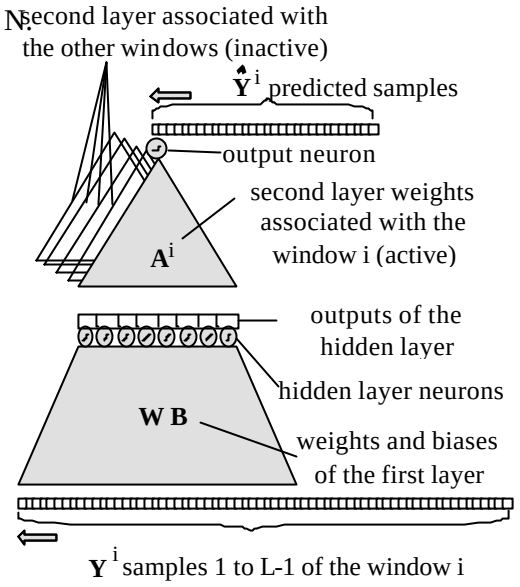
2.2 Coding phase

The $\mathbf{A}^i$ previously computed are not used for the coding phase, they are only used for the first layer adjustment (see figure 2). For each window to code we use $\mathbf{W}$ and $\mathbf{B}$ previously computed, and we initialize at random $\mathbf{A}^i$. Then we minimize the criterion :

$$J_2^i = \sum_k e_i(k)^2 \qquad (6)$$

This is done by modifying the weights of the second layer ($\mathbf{A}^i$) with the Madaline rule I.

$\mathbf{A}^i$ constitute then the coding coefficients of the window i. So the number of coefficients generated is the same as the number of neurons in the hidden layer which is N.



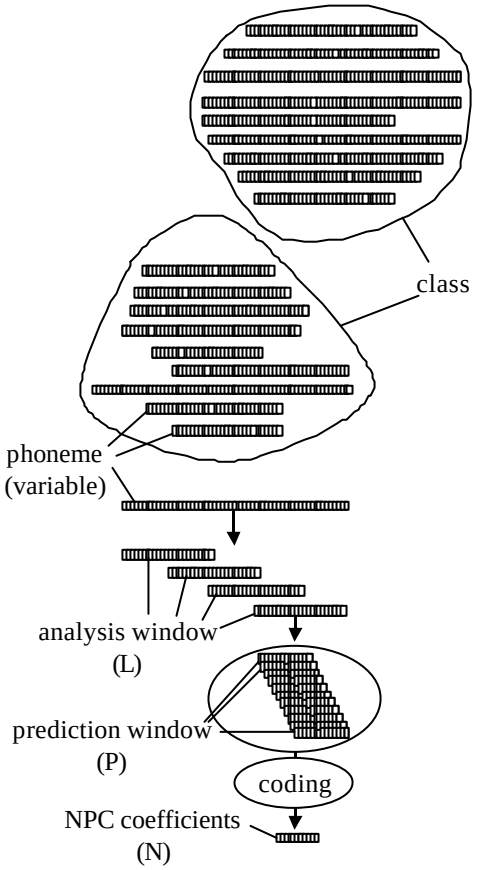


Fig. 1 : Architecture of the NPC Model

Fig. 2 : several kinds of window for the NPC algorithm, the numbers in brackets are the width of each window.

3 Compression and decompression of the speech signal

Speech signals compression brings the following advantages :

- the reduction of the rate during the data transmission from the transmitter to the receiver.
- a saving place in data storage over physical supports (such as : hard drives, cdrom, ...).

In this chapter, we are going to shortly describe the speech compression and decompression methods. Then we will present the algorithms and, finally, we will give the results obtained.

3.1 Principle

The method that we propose is based on the property that two consecutive samples of a speech signal are correlated.

According to this observation, this compression method is close to the others methods based on the prediction of the speech signal (ADPCM, CELP,…) [5,6]. The aim is to quantify and to transmit (or to store) the following prediction error : $e_i = y_i - \hat{y}_i$, where $\hat{y}_i$ is a prediction of y_i (i^{th} sample of the speech signal). By this way, we reduce the amplitude of the data to transmit and thus we reduce the flow.

We have represented, on the figure 3, the compression block diagram with the predictor NPC.

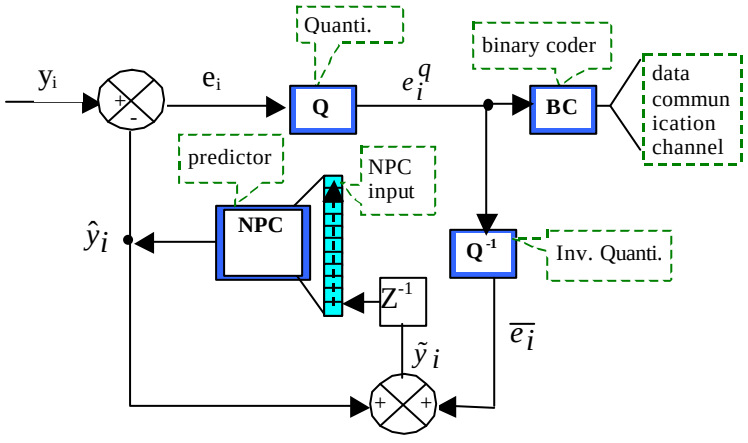


Fig. 3 : Compression block diagram

where y_i is the i^{th} sample of the signal and Z^{-1} represents the time delay.

Q and BC are respectively the quantifier and the binary coder used before the transmission of the error prediction, e_i, to the receiver.

As we can observe, on this figure 3, we have introduced the decoded signal, $\tilde{y}_i$. In fact it is necesssary, at the reception, to excited the coder NPC with datas deduced from the quantified error. It is why the "NPC input block" is a vector of M samples of decoded speech signal. By this way, we are sure that the predictor is robust and the prediction is optimize for a good reception and hearing.

Here, is an important difference between the original version of NPC : during the coding phase, the samples of the initial speech signal y_i are repaced by their "estimates" $\tilde{y}_i$.

So, the algorithm which has been used during the coding phase is close to the NLOE (Non-Linear Output Error) algorithm. The main difference with the NLOE is that the predicted signal $\hat{y}_i$ (input of the networks) is replaced by the decoded signal $\tilde{y}_i$.

On the figure 4, we have represented the decompression block diagram.

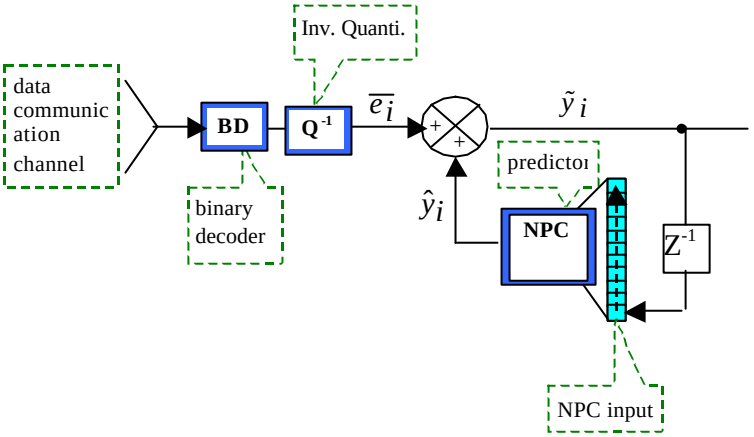


Fig. 4 : Decompression block diagram

At the reception, the error is decoded by a binary decoder (BD) then we applied an inverse quantization with Q^{-1}. Finally, the signal wich will be heard is the $\tilde{y}_i$.

About the quantization we have used the european A-law. For a given signal x, the output of the A-law compression is :

$$y = \begin{cases} \dfrac{A.|x|}{1+\log(A)}\operatorname{sgn}(x) & , \ 0 \le \dfrac{|x|}{x_{max}} \le A^{-1} \\ x_{max}\dfrac{1+\log\left(A.|x|/x_{max}\right)}{1+\log(A)}\operatorname{sgn}(x) & , \ A^{-1} \le \dfrac{|x|}{x_{max}} \le 1 \end{cases}$$

Where A is the A-law parameter of the compander, x_{max} is the maximum value of the signal x, log is the natural logarithm and sgn is the signum function. We easily notice that the more we have quantization levels, more the coded signal is close to the original signal. The error produced during the quantization is called quantization noise.

3.2 Algorithms

3.2.1 First layer of the NPC

In a first time, it is necessary to compute the first layer of weights of the NPC. For that, we have used the DARPA-TIMIT database. This base contains 8 American dialects (New-england, Northern,…). There are 630 men and women speakers. Each speaker says 10 sentences. For each sentence, a segmentation by phoneme is given.
Using the segmentation file, we have extracted 100 examples for each reduced phoneme (39). As presented in part 2, each example is divided into several windows (20ms) with an overlapping of 50%. So, we have realized the learning database for the NPC.
The training is stopped after 40000 epochs because the backpropagation error don't significantly decrease. Then the weights of the first layer are fixed and could be used in the second stage : the compression/decompression.

3.2.2 Compression

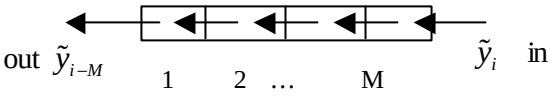
Let N be the number of samples of the speech signal and y_i the i^{th} sample. Let **A** be the second layer of weights and **I** the NPC input vector.
The speech signal compression algorithm is described below :

Initialization

- **A** with zeros.
- **I** with zeros

For each sample of the signal

- Prediction of $y_i \rightarrow \hat{y}_i$,from **I** with NPC
- Calculation of prediction error $e_i = y_i - \hat{y}_i$
- Quantization of the prediction error : $e_i^q = Q(e_i)$
- ***Transmission of*** e_i^q ***to the receiver***.
- Inverse quantization : $\overline{e}_i = Q^{-1}(e_i^q)$.
- Modification of the second layer of NPC by backpropagation of $\overline{e}_i$.
- Calculation of the decoded signal : $\tilde{y}_i = \overline{e}_i + \hat{y}_i$
- Introduction of $\tilde{y}_i$ in **I** :



End

3.2.3 Decompression

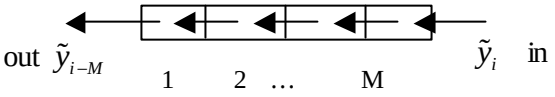
The speech signal decompression algorithm is described below :

Initialization

- **A** with zeros.
- **I** with zeros

For each sample of the signal

- Prediction of $y_i \rightarrow \hat{y}_i$ from **I** with NPC
- ***Reception of*** e_i^q ***to the receiver***.
- Inverse quantization : $\overline{e}_i = Q^{-1}(e_i^q)$.
- Modification of the second layer of NPC by backpropagation of $\overline{e}_i$.
- Calculation of the decoded signal : $\tilde{y}_i = \overline{e}_i + \hat{y}_i$
- Introduction of $\tilde{y}_i$ in **I** :



End

The mean difference between these two algorithms is that the steps of calculation and quantization of the prediction error are not necessary in decompression.
Of course, the performance of this compression strongly depends of the ability of NPC to predict the speech signal.
After this presentation of the NPC compression, we are going to present some results obtained on sentences extracted from the DARPA-TIMIT database.

3.3 Results

3.3.1 Signals and errors

We have represented on figure 5 a speech signal to compress.

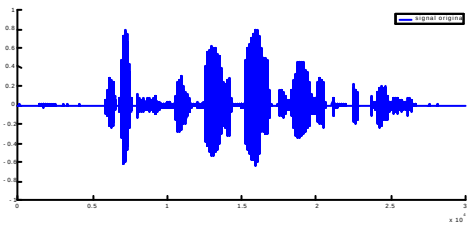


Fig. 5 : Original speech signal y_i

The decoded signal is represented on the figure 6, with the same scale.

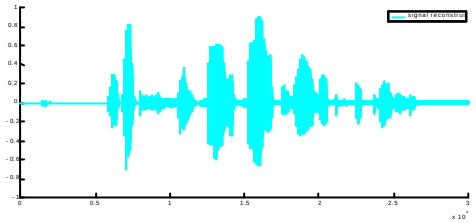


Fig. 6 : Decoded speech signal $\tilde{y}_i$

For this example, the error prediction is quantified with 4 bits whereas each sample of the original signal is coded with 16 bits. So the rate is reduced by a factor 4 and, as we can see on the figure 6, with a minimum of degradation for the decoded signal.

The figures 7 and 8 respectively represent the prediction error e_i and the compression error x_i which are defined by :

$$e_i = y_i - \hat{y}_i \quad \text{and} \quad x_i = y_i - \tilde{y}_i$$

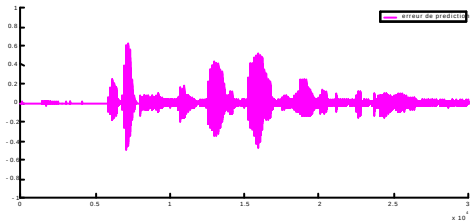


Fig. 7 : Prediction error e_i

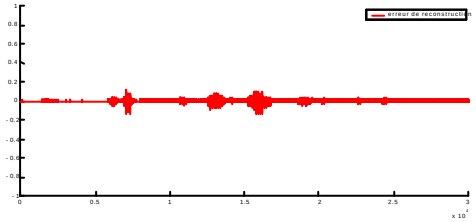


Fig. 8 : Compression error x_i

It can be noticed, on these two last figures, that the compression error x_i is smaller than the prediction error e_i. This result is due to the correction of the predicted signal given by the quantization error $\overline{e_i}$.

In order to evaluate the performances of the compression and decompression of the signals, we propose to study the two following criterions :

The prediction gain : G_p

The prediction gain is calculated from the signal to prediction error ratio. This ratio is calculated in dB.

$$G_p = 10\log_{10}\left(\frac{\sum_i y_i^2}{\sum_i e_i^2}\right) \qquad (7)$$

This criterion is appropriate to test the NPC performances, i.e to check the ability of the NPC to predict precisely a speech signal.

The quantization gain : G_q

This criterion evaluates the deterioration inflicted to the decoded signal compared to the original by the quantization. G_q is calculated from the ratio between the power of the speech signal and the power of the compression error x_i.

G_q is writen (in dB) :

$$G_q = 10\log_{10}\left(\frac{\sum_i y_i^2}{\sum_i x_i^2}\right) \qquad (8)$$

3.3.2 Performances

In the table 1, we present the results obtained for the two criterions G_p and G_q and for two quantizations on 3 and 4 bits.

For these tests we have used five sentences extracted from DARPA-TIMIT. Of course, these sentences has not been introduced in the training phonemes database. These sentences has been classified by levels of prediction gain.

Sentence	Compression		
	Quantization with 3 bits		Quantization with 4 bits
	G_p (dB)	G_q (dB)	G_q (dB)
1	12.1	20.8	27.6
2	13.5	21.9	28.4
3	14.0	22.7	29.8
4	14.2	23.2	30.1
5	14.3	24.7	31.1

Table 1 : Results obtained for G_p and G_q for two quantizations and five sentences

About these results we can do several observations:

- The prediction gain is as much more important than the number of voiced phonemes is greater than the number of unvoiced phonemes.
- According to audio tests realized, we can noticed that the decoded signal is very close

to the initial signal (for the human ear) when G_q is at least equal to 20 dB.

4 Conclusion and futures works

We have presented in this paper a new speech coder/decoder based on the a Neural Pedictive Coding. This coder is interesting for differents reasons :

- The weak complexity of the algorithms allow to use the coder/decoder in a real time application (after computation of the first layer of the NPC).
- The flow of transmission of data is divided by a factor 4. So the rate is of 16 kb/s, with a good restitution of the decoded signal for the human ear.

About our future works we will interest to the structure of the NPC. As we have discussed in chapter 3.3.2, the power of the prediction error depends on the kind of phoneme. So the prediction error of the NPC is more important for unvoiced sound than for voiced sound. This result is linked to the fact that a voiced sound is more adapted to a predictible model.
Consequently, in the future coder/decoder, we will increase the number of NPC predictor. Each NPC will be trained on a particulary class of phoneme (voiced and unvoiced):

- fricative, liquid, nasal, occlusive and vowels.

After that, each windows (about 20ms) of speech signal will be coded by all NPC predictors. Then the quantization error, associated to the most effective NPC, will be transmitted. This method need to introduce a delay time of one window (20ms) and it will be neccessary to transmit, at the beginning of each window, the NPC coder which must be active in reception.

References:

[1] Gas B, Zarader J.L and Chavy C., "A new approach to speech coding: the Neural Predictive Coding"; *International Journal of Advanced Computational Intelligence*, Vol 3, n°6, Novembre 2000 pp 19-28.

[2] Chavy C, Gas B, Zarader J.L. "Neural predictive coding applied to noisy phoneme recognition"; *International Joint Conference on Neural Networks* (IJCNN 99'). July 1999, Washington, USA, pp 220-223.

[3] B. Townshend "*Nonlinear prediction of speech*", ICASSP'91, pp 425-428.

[4] J. Thyssen, H. Nielsen, S. Duus Hansen "*Non-linear short-term prediction in speech coding*", ICASSP'94, I-185-188.

[5] Ramachandran R.P and Richard M., *Modern Methods of speech processing*, Kluwer Academic Publishers, 1995.

[6] Boite R, Boulard H, Dutoit T, Hancq J and Leich H, *Traitement de la parole*, Presses Polytechniques et Universitaires Romandes, 1999.

A.3 Article de revue [68]

Discriminant Neural Predictive Coding Applied to Phoneme Recognition

B. GAS and J.L. ZARADER and C. CHAVY and M. CHETOUANI
Neurocomputing, Vol. 56, p. 141–166 (2004)

.

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®
Neurocomputing 56 (2004) 141–166

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Discriminant neural predictive coding applied to phoneme recognition

B. Gas*, J.L. Zarader, C. Chavy, M. Chetouani

Laboratoire des Instruments et Systèmes d'Ile de France, Groupe Perception Automatique Reseaux Connexionnistes, Université Paris VI, 4 place Jussieu, BP 164, 75005 Paris, France

Accepted 5 August 2002

Abstract

In this article, we propose to study a speech coding method applied to the recognition of phonemes. The model proposed (the neural predictive coding (NPC) and its three declinations NPC-1, NPC-2 and DFE–NPC) is a connectionist model (multilayer perceptron) based on the nonlinear prediction of the speech signal. We show that it is possible to improve the discriminant capacities of such an encoder with the introduction of signal membership class information as from the coding stage. As such, it fits in with the category of discriminant features extraction (DFE) encoders already proposed in literature. In this study we present a theoretical validation of the model in the hypothesis of unnoised signals and Gaussian noised signals. We also define a new distance, the NPC distance, that will allow experimental validation of the model. NPC performances are compared to that obtained with traditional methods used to process speech on the Darpa Timit phoneme base. Simulations presented here show that the classification rates have clearly improved compared to usual methods, in particular regarding phonemes considered difficult to process (/b/,/d/,/g/ and /p/,/t/,/k/ phonemes).

Keywords: Speech coding; Neural networks; Nonlinear signal processing; Discriminant feature extraction

1. Introduction

Although applications already exist (let us mention for instance the vocal dictation for speech recognition), speech processing still remains a research field that is broadly opened. In fact, spontaneous speech, speech in a voiced environment, multispeaker

* Corresponding author. Tel.: +33-1-44-27-54-78; fax: +33-1-44-27-62-14.
E-mail address: gas@ccr.jussieu.fr (B. Gas).

0925-2312/$ - see front matter
doi:10.1016/j.neucom.2002.08.001

recognition including a large vocabulary, are so many issues still requiring proper solving.

In applications such as speech recognition, the classification process is divided into two consecutive steps: coding and classification. The main goal of the coding process is to extract a maximum of interesting signal features while reducing the amount of data to a minimum. The selection of the representation of signals depends on its intrinsic qualities but also on the recogniser system (HMM [15,30], and neural networks [25–27,41] for instance) and on the application goal (speaker, speech or language recognition, ...). Two main classes of coding methods can be identified. Those obtained in the temporal domain from a voice-producing model and those issued from the frequential domain corresponding to a voice-listening model.

A large number of representations in the temporal domain have been proposed. One of the most commonly used is issued from the linear predictive coding (LPC) model [20,32]. Other representations also exist, but they are mostly derived from the computation of LPC coefficients, such as the PARCOR ones which afford the advantage of being within -1 and $+1$, when the filter is stable. One also finds the log area ratio (LAR) coefficients and the linear predictive cepstrum coding (LPCC) coefficients, which are in fact a cepstral representation of signal but are computed using LPC coefficients.

In the frequential domain, most features extraction methods are based on a speech physiological audition model. The most frequent representation uses the Mel scale like the mel frequency cepstral coefficients (MFCC) coding [10,42] and also the PLP and Rasta-PLP [18,19].

The small discriminant capacities and the insufficient robustness of present coding are a justification for improving coding techniques, and even for perfecting new methods. We will in particular focus on two large axes:

- the use of nonlinear coding techniques to exceed the limits of the present coding methods,
- the taking into account of the information of signal categorisation into classes, as from the coding stage, to improve the discriminant capacities of the encoder.

The encoder model we will present in this article is a nonlinear model (multilayer perceptron) that also exploits the information of signal class membership.

1.1. Nonlinear extension

Nonlinear dynamic systems present a richer behaviour than linear systems, making it possible to model features of the signal that are absent in the traditional linear approach. Tishby [32] applied a neural network approach to the speech signal prediction problem. The resulting predictors have had a few additional interesting features such as the fact that the spectrum of the residue was whiter than that of the linear predictor, indicating that the structure of the speech signal was better captured by the nonlinear predictor. Tishby has shown [38] that the two issues, the reduction in dimension and the nonlinear temporal variability, can be addressed using geometrical methods from

nonlinear dynamics. Ma et al. [31] used this type of results to propose a signal process based on a nonlinear local prediction model (NLLP) applied to speech coding. With the same implementation, the speech coding based on the NLLP gave improved performance compared to linear local scheme (LLP). A description of nonlinear modelisation with nonlinear auto-regressive models (NLAR models) can be found in [35]. Cleveland et al. [9] also proposed a local linear approximation of the interpolation functions leading to a new interpretation of the signal in the form of a code-book for its own prediction, e.g. a predictive counterpart to vector quantization. An alternative proposed by Diaz-de-Maria [12] using radial basis function (RBF) networks imply building a parametric model to obtain a global approximation. Thownshend [39] took measurements of the correlation dimension of normally spoken speech from a single speaker, showing that one should be able to build a nonlinear predictor for speech that would significantly outperforms linear predictors.

Now that we have pointed out the interest of modelling nonlinear features of the speech signal, the problem is how can we do it. Volterra's filters are a first type of such nonlinear filters [33]. Major drawbacks of this model lie in the fact that the number of the model parameters grows fast with the prediction window and that, for a signal compression task, the inverse filter is not always stable. Thyssen et al. [37] made experiments showing that Volterra filters are superior to the LPC-technique, regarding the prediction gain (18.2 dB compared to 14.1 dB for LPC). But a linear predictor with a 10 prediction window width only requires 10 coefficients compared to 65 with a second-order Volterra's filter. Knowing that neural networks can also be used as adaptive filters [13], authors also proposed a Predictive Time Delay Neural Network that allows a substantial growth of the prediction gain (19.3 dB).

1.2. Discriminative feature extraction

Discriminative features extraction (DFE) consists in extracting an enhanced features vector from the signal, in relation to the classification task.

The improvement of the discriminative capacities of speech coding algorithms has already been the subject of research works [1,23,24,29,45]. The main methods proposed these last years have been inspired by the MCE/GPD theory proposed by Juang and Katigiri [21] in 1992. Before this date, Zahorian et al. [45] proposed to improve HMMs discriminant capabilities by achieving a linear *discriminant* transformation of the features vector. Authors show that from a set of 30 words of one syllable, one can obtain an improvement of near 25% of the recognition rates. The transformation is such that feature vectors belonging to the same class are well-clustered and feature vectors belonging to separate classes are well separated. The within-class covariance is minimised and the between-class covariance is maximised. The within-class covariance computing needs to cluster data by classes, so that category information are introduced from the coding stage. Linear discriminant analysis (LDA) is a standard technique in statistical pattern classification for dimensionality reduction with a minimal loss in discrimination [14]. Its application to speech recognition has shown consistent gains for small vocabulary and mixed results for large vocabulary. Known to be inappropriate for the case of classes with unequal sample covariances, Saon et al. [34] used instead

the heteroscedastic discriminant analysis (HDA) and proposed an extension of it allowing a 10–13% relative improvement in the word error rate over standard cepstral processing.

In the last years, Thomae et al. [36] have proposed a new approach called the "extended linear discriminant analysis with model transformation (ELDA-MT)" and using the MCE/GPD algorithm with a HMM classifier. They obtained a significant reduction of word error rate of relatively 6.2%.

Juang et al. [21,22] have proposed a method, usually called *discriminant features extraction* (DFE), and based on the use of the *minimum error criterion* (MCE) and the *gradient probabilistic descent* (GPD). Several works have been made from this method to realise varied discriminant coding applications. A front-end feature extractor and a back-end classifier are usually investigated separately due to the lack of a theoretical framework to formalise this situation. De la Torre et al. [11] proposed variations of the DFE method allowing an independent learning of the classifier and the encoder. Unlike this, the main idea of DFE is that the feature representation for classification purposes should be designed while taking into account the entire pattern recogniser. Biem and Katagiri [3–5] applied this method to the filter bank parametric representation and also to the cepstral filtering. Bachianni et al. [2] have also proposed to use DFE techniques for the time-frequency masking filter optimisation. Since DFE methods lead to bringing together the extraction of features and the features classification into a single global module, the respective training algorithms become interdependent or even merge together.

1.3. Scope of the article

Encoders NPC-1, NPC-2 and DFE–NPC proposed by Gas and Zarader [16,17,46] will respectively be presented in Sections 2, 5 and 6. In Section 3, you will find a summary of experimental conditions used, and also the first results obtained with the NPC-1 encoder. In Section 4, we will tackle the issue of the discrimination capability of an encoder. The theoretical and experimental validation of NPC-2 encoder will be presented in Section 5. Finally, we will present in Section 6, the latest version DFE–NPC of the encoder giving the best classification rates for the recognition of phonemes.

2. Neural predictive coding: the NPC model

The neural predictive coding (NPC) model is an extension of the LPC to the non-linear area. The model is derived from the Lapedes and Farber [28] nonlinear predictive model. Let us consider for our purpose the problem of the adaptive filtering method applied to the coding of phonemes. The goal of the well-known LPC coding consists in determining the coefficients of a linear predictive filter. Speech signal is divided into consecutive fixed length frames and coefficients are computed so as to minimise the mean square prediction of any given signal frame: hence, they are representative of their corresponding signal frame (Fig. 1).

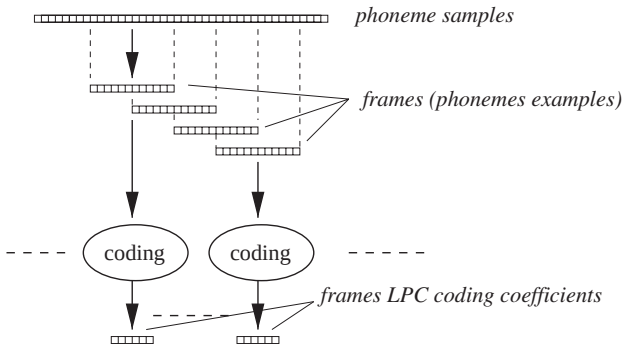


Fig. 1. Speech signal decomposition for LPC coding.

By sampling the continuous scalar signal with a fixed sample rate, a time series $y_1, y_2, \ldots, y_k, y_{k+1}, \ldots$ can be obtained and grouped into overlapping vectors $\mathbf{x}_k = [y_{k-1}, y_{k-2}, \ldots, y_{k-\lambda}]^\top \in \Re^\lambda$ where λ is the sequence length called the prediction window width.

Let us now consider a simple neural network used as a nonlinear predictive filter (NLAR model), e.g. with λ inputs (λ is the prediction window width) and one output cell. One can also adapt the weights so as to minimise the mean square prediction error. Since there is no analytical solution to this problem (unlike the linear case), solution weights have to be found recursively, e.g. with the backpropagation learning algorithm. The representative coefficients of the framed signal are the final network weights. Nonlinear signal features can then be extracted but, unlike in the LPC case, the number of weights grows fast with prediction window width λ. For instance, a $20 \times 8 \times 1$ network structure leads to 177 parameters (the network weights and bias) against 20 for the corresponding LPC encoder. For comparison, the coefficients number remains between 16 and 32 in traditional speech coding systems.

The NPC model provides a very simple solution to this problem. Considering the same network, one can decide to use only the second layer weights as coding parameters. In such a case, window width λ (e.g. the network input number) has no more effect on the code dimension which only remains proportional to the number of hidden layer cells. This is the main idea of the NPC coding. In a phoneme coding task, the hidden layer weights must be representative of the coded phoneme features while the first layer weights must not. Such a property can be obtained by adapting the first layer weights to all the phonemes in the database, while the hidden layer weights are only updated on the samples of the phoneme that needs coding. This leads us to consider the same number of hidden layers and phonemes that need coding, as opposed to only one first layer common to all phonemes. The learning process is broken down into two computing stages (Fig. 2):

- The first stage is the *parameters adjustment phase* or *parameterisation phase*. It consists in learning the weights of the first layer. We thus obtain the NCP encoder parameters.

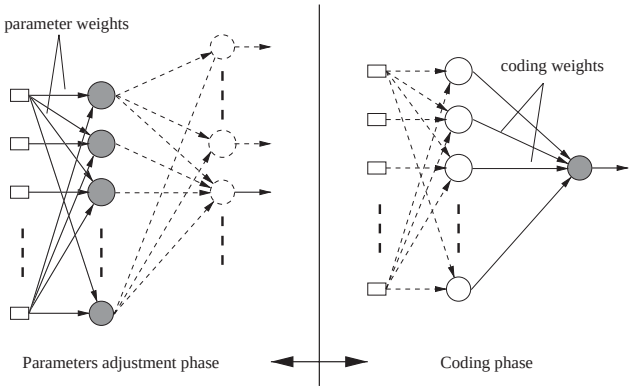


Fig. 2. NPC parameters adjustment phase and coding phase.

- The second stage is the *coding phase*: the first layer weights are initialised with the weights drawn from the parameter adjustment phase and they will remain fixed. The only weights to be learnt are the hidden layers weights, that will become the phoneme coding parameters.

Maintaining a common weights layer leads to a greater convergence difficulty which is time consuming for computing. The second learning phase (the coding phase) can be done faster than the first one since the encoder parameters no longer require updating. Both for the parameters setting phase and the coding phase, the learning rule is of the modified Madaline Rule III (MRIII) [44] type.

The NPC distance and the modelling error ratio (MER) will be defined further below in this article: they are useful tools which will permit us to make a clear experimental demonstration to show that, indeed, the second weights layer acts as a phoneme feature extractor. Before, we propose to formalise the NPC model.

1. Following now the Lapedes and Farber [28] model, one can see the NPC encoder as a layered neural network trained to predict time series. It is trained from examples of pairs of $\mathbf{x}_k = [y_{k-1}, y_{k-2}, \ldots, y_{k-\lambda}]^\top$ input vectors and y_k output samples, while minimising the mean square error:

$$Q(\mathbf{w}, \mathbf{a}) = \frac{1}{2} \sum_{k}^{K} (y_k - f_{\mathbf{w},\mathbf{a}}(\mathbf{x}_k))^2, \tag{1}$$

where $f_{\mathbf{w},\mathbf{a}}$ is the nonlinear n dimensional function with parameters noted $\mathbf{w}$ (first layer weights) and $\mathbf{a} = [a_1, \ldots, a_N]^\top$ (hidden layer weights) including sigmoidal node functions. The error is minimised by modifying parameters $\mathbf{w}$ and $\mathbf{a}$ using a standard multidimensional optimisation method, e.g. steepest descent (error back propagation). More precisely, $f_{\mathbf{w},\mathbf{a}}$ can be viewed as the composition of two functions $g_{\mathbf{w}}$ (corresponding to the network first layer) and $h_{\mathbf{a}}$ (corresponding to the network output layer) such that:

$$f_{\mathbf{w},\mathbf{a}} = h_{\mathbf{a}} \circ g_{\mathbf{w}} \quad \text{with } f_{\mathbf{w},\mathbf{a}}(\mathbf{x}_k) = h_{\mathbf{a}}(\mathbf{z}_k) \text{ and } \mathbf{z}_k = g_{\mathbf{w}}(\mathbf{x}_k). \tag{2}$$

NPC coding allows an arbitrary number of coding coefficients by creating a hidden layer for each phoneme, while the first layer remains the same for all phonemes.

Considering one phoneme l, the cost function (1) becomes:

$$Q(\mathbf{w}, \mathbf{a}_l) = \frac{1}{2} \sum_{k}^{K(l)} (y_k - f_{\mathbf{w},\mathbf{a}_l}(\mathbf{x}_k))^2, \tag{3}$$

where k represents the samples composing the phoneme l and $K(l)$ the samples number. For a set of $l = 1, \ldots, L$ phonemes, the NPC structure leads to the following cost function:

$$Q(\mathbf{w}, \mathbf{a}_1, \ldots, \mathbf{a}_L) = \frac{1}{2} \sum_{k=1}^{K} \sum_{l=1}^{L} (y_k - h_{\mathbf{a}_l} \circ g_{\mathbf{w}}(\mathbf{x}_k))^2 \delta_{L(\mathbf{x}_k)-l}, \tag{4}$$

$h_{\mathbf{a}_l}$ being the output layer weights linked to the phoneme l, K the total number of samples (over all the phonemes) and $L(\mathbf{x}_k)$ the phoneme number of the sample $\mathbf{x}_k$ (which means that $\mathbf{x}_k$ is a sample of phoneme $L(\mathbf{x}_k)$). δ is the Kronecker symbol: $\delta_{L(\mathbf{x}_k)-j}$ has value one every time the phoneme $L(\mathbf{x}_k)$ is j, zero otherwise. Output layer weights are proper to each phoneme: they are the coding coefficients vector, while the first layer weights are common to all the phonemes and constitute the NPC encoder parameters. Fig. 3 shows the network structure.

2. *The parameters adjustment phase*: The purpose of this computing phase is to estimate the $g_{\mathbf{w}}$ function, e.g. the encoder parameters. This must only be done once, on

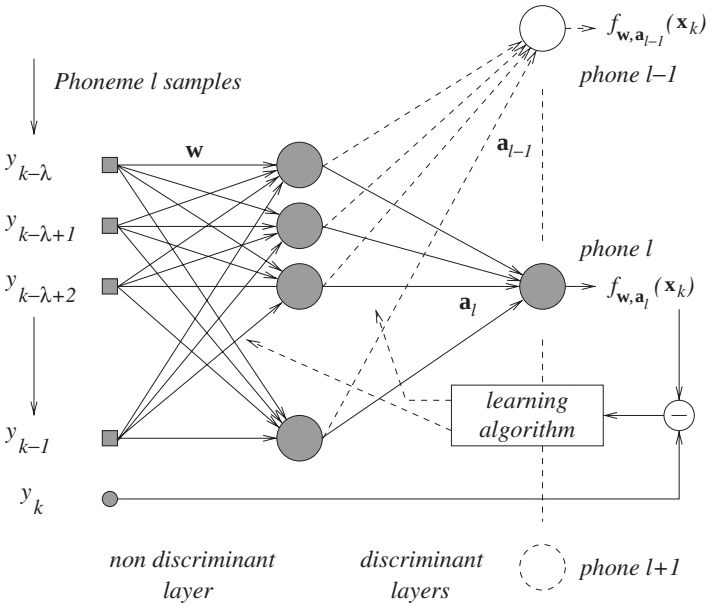


Fig. 3. The NPC network structure.

a database representing at best the coding problem. Eq. (4) gives the cost function to be minimised by the traditional back propagation algorithm. During the learning process, the coding coefficients $\mathbf{a}_l$ must also be computed to ensure a correct estimation of $g_{\mathbf{w}}$. Once the learning process has been completed, the phonemes coding coefficients are no longer used. The encoder is then ready to code data.

3. *The coding phase*: This is the code generation phase. For one given phoneme l and the signal samples y_k composing it, the first network layer $g_{\mathbf{w}}$ acts as an n-dimensional filter, or a space transformation operator: $\mathbf{z}_k = g_{\mathbf{w}}(\mathbf{x}_k)$. One obtains phoneme l coding coefficients by minimising the following cost function:

$$Q(\mathbf{a}_l) = \frac{1}{2}\sum_k (y_k - h_{\mathbf{a}_l}(\mathbf{z}_k))^2. \tag{5}$$

3. Experimental validation of NPC models

With a view to evaluating the performance of the various NPC encoders we present in this article, we describe in this section the experimental conditions of our simulations.

3.1. The TIMIT phoneme database

We built several phoneme bases each extracted from the Darpa-TIMIT ant NTIMIT [40] speech database. The second Ntimit base groups all the Timit base signals transmitted by telephone. Timit and Ntimit databases group speakers classified in categories for 10 regions in the United States. We limited ourselves to the speakers in the first region (New England).

We called B1–B4 the four bases we exploit here. The first B1 base groups four classes of voiced phonemes (vowels) very commonly used: /*aa*/, /*ae*/, /*ey*/ and /*ow*/. They have been extracted from the Timit base. B2 and B3 bases group two series of phonemes extracted from the Timit base: /*b*/, /*d*/, /*g*/ (voiced plosives) and /*p*/, /*t*/, /*k*/ (unvoiced plosives). These two databases are of a particular interest to evaluate coding and classifications methods for speech processing. In fact, they are frequently used and simultaneously difficult to process. They were used by Waibel and Lang [27,41] to validate their model (the time delay neural network, (TDNN)). B4, the fourth database brings together six classes of phonemes /*s*/, /*z*/, /*aa*/, /*ah*/, /*iy*/, /*ih*/ extracted from the telephoned Ntimit database. We used it to test the NPC encoder on signals with a limited pass-band. As it is composed of three sub-classes of phonemes, this base enables an encoder discriminant capacities to be evidenced: the three sub-classes are well separated while phonemes are acoustically very close to each other within each sub-class. The choice of phonemes is ruled by the following conditions (Fig. 4):

- Depending on its duration, each phoneme is split into a number of frames with a fixed length (analysis windows) with a given interlace factor. Each frame constitutes a phoneme example.

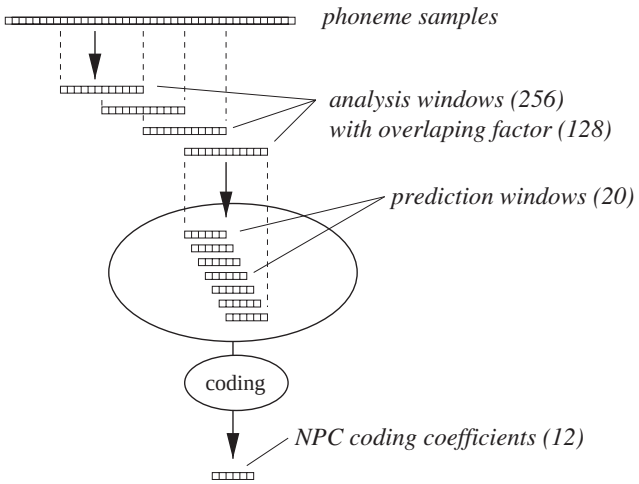


Fig. 4. The NPC coding scheme (numbers in brackets are the width of each window).

Table 1
Used base main features

Base	B1		B2		B3		B4	
Base type	Timit		Timit		Timit		Ntimit	
Sampling frequency	16 KHz		16 KHz		16 KHz		8 KHz	
Base name	B1L	B1T	B2L	B2T	B3L	B3T	B4L	B4T
Examples number	500	500	500	500	500	500	500	500
Frame size	256	256	256	256	256	256	128	128
Overlaping factor	128	128	128	128	128	128	64	64

- Examples are chosen at random among all available speakers to produce a multi-speaker model.

We subdivided each B1–B4 base into two sub-classes, one for encoder parameterisation and classifier learning, and the other for generalisation tests. Table 1 sums up the characteristics of the four bases thus constituted.

3.2. Traditional coding methods

We made comparisons between NPC coding and the traditional coding methods. Among temporal coding methods, we considered LPC coding. For spectral coding, we considered the MFCC coding. Due to its robustness, MFCC is among the most commonly used methods in speech processing. We set to 12 the number of coding coefficients per phoneme, so the dimension of the coding vectors was 12.

Table 2
Recognition rates comparison between LPC, MFCC and NPC coding

Coding method	LPC	MFCC	NPC
Recognition rate (%)	55.5	58	**61.2**

Table 3
NPC performance for several iterations number of the coding process

Iteration number	1	5	10	20	40	100	250	500
Recognition rate (%)	60.1	**61.2**	**61.2**	60	59.6	59.8	59.7	59.7

3.3. Classification by MLP neural network

The classifier used to estimate the performance of all the encoders was a basic back-end multilayer perceptron (MLP) with 12 inputs (equal to the coding vectors dimension), 10 hidden neurones and as many outputs as there were phoneme classes. The learning rule was a gradient descent using the backpropagation algorithm.

3.4. Experimental results

Experimental results obtained with NPC encoder showed improved scores for the classification of phonemes [8] compared to the major competitive encoding methods. Table 2 illustrates scores obtained with B4T generalisation database compared to LPC and MFCC encoding.

The same experiments revealed the existence of an over-fitting problem during the phoneme encoding phase. In fact, when codes are recovered for several values of the iteration number, and then each of these sets of codes are classified, a reduction of scores can be noted after passing through a maximum. Table 3 is an illustration of this for a number of iterations ranging from 1 to 500. The good result obtained for classification can be explained by the fact that the optimisation method is of the stochastic gradient type. This means that for a 128-sample frame and a 20-sample prediction window, $128 - 20 - 1 = 107$ learning stages are obtained for one iteration.

We will deal with this over-fitting issue when we study the NPC-2 encoder (see Section 5.2.5).

4. Constraint coding by NPC model

We applied here Zahorian's idea [45] quoted in the introduction, according to which a discriminant linear transformation of vectors makes it possible to increase the classifier performance. Transformation is such that vectors for characteristics belonging to the same class must be close to each other within their representation space while vectors of

a different class must be very distant. This means that (in the hypothesis of a Gaussian distribution of data into classes) the within-class covariance will be minimised while the inter-class variance is maximised. Such a processing can be introduced in NPC encoder in the form of constraints on the code weights during learning. These constraints are expressed in the form of two additional terms of the cost function to be minimised. However, it is difficult to obtain a noticeable improvement of coding by this means, except if we take into account a strong constraint that will consist of imposing a unique set of code per class during the parameterisation phase.

4.1. Constraint obtained with the minimisation of within-class covariances

Let us consider a $\mathbf{a}_{l=1,\ldots,L_c} = [a_l^1, a_l^2, \ldots, a_l^N]^\top$ set of data (vectors of phoneme codes in this case) belonging to the same c class (N is the code dimension: $N = 12$ in our case). In the hypothesis of a spherical Gaussian distribution of datas, we obtain the following expression of the within-class covariance:

$$\sigma_c^2 = \frac{1}{NL_c} \sum_{i=1}^{N} \sum_{l=1}^{L_c} (a_l^i - \bar{a}^i)^2 \quad \text{where } \bar{a}^i \simeq \frac{1}{L_c} \sum_{l=1}^{L_c} a_l^i \text{ is the average vector.} \tag{6}$$

The joint minimisation of the prediction error and of the within-class covariance gives the cost function (P being the total number of classes):

$$Q = \alpha Q^m + (1 - \alpha) Q^d \quad \text{with } Q^d = \frac{1}{P} \sum_{c=1}^{P} \sigma_c^2, \tag{7}$$

where α is a weighting factor. The first term is of the same family as the prediction error, that we will call the *modelling error*; while the second term is of the *discrimination error* type. Therefore, the α factor (which may be adaptive) weights modelling and discrimination. This process is only implemented during the encoder parameters adjustment phase. In fact, during the coding phase, classes are unknown. The purpose of the discrimination constraint is therefore to cause a rearrangement of the first layer weights so as to build a more discriminant filter during the coding stage.

Fig. 5 illustrates the evolution of the Q^m modelling quadratic error (prediction error only) during the encoder parameter learning. Until iteration 4800, learning was carried out without constraint ($\alpha = 1$), but then, for the following iterations, we had to impose a constraint ($\alpha = 0.99$). For this experiment and the following ones, we worked on the NTIMIT phoneme B4 database (six classes /*s*/, /*z*/, /*aa*/, /*ah*/, /*iy*/, /*ih*/ extracted from the NTIMIT database, New England area), including 1000 samples per class. Of which 500 were allotted to the parameters adjustment phase and to the classifier learning (B4L base), while the other 500 samples (B4T base) were allotted to generalisation tests (see above Section 3 on experimental conditions in this article for more details). Therefore this figure clearly illustrates the impact of the discriminant constraint on the prediction error (from iteration 4800), mainly regarding the antagonist aspect of the two Q^m and Q^d contributions. With this result we foresee difficulties to determine the α weighting. Scores obtained for generalisation on the B4T test base (coding followed by a classification of phonemes) after this parameterisation phase confirm this difficulty as they do not exceed 61.5%.

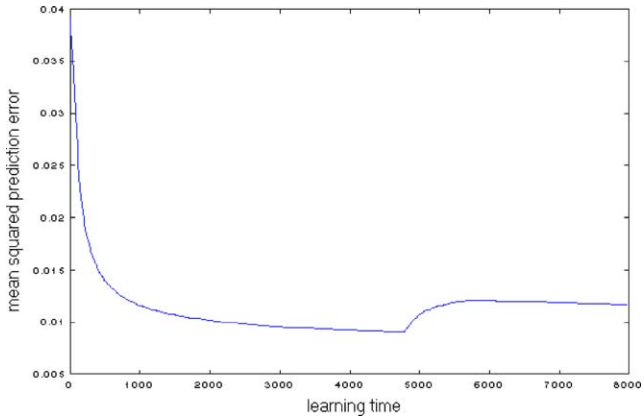


Fig. 5. Mean squared prediction error with constraint.

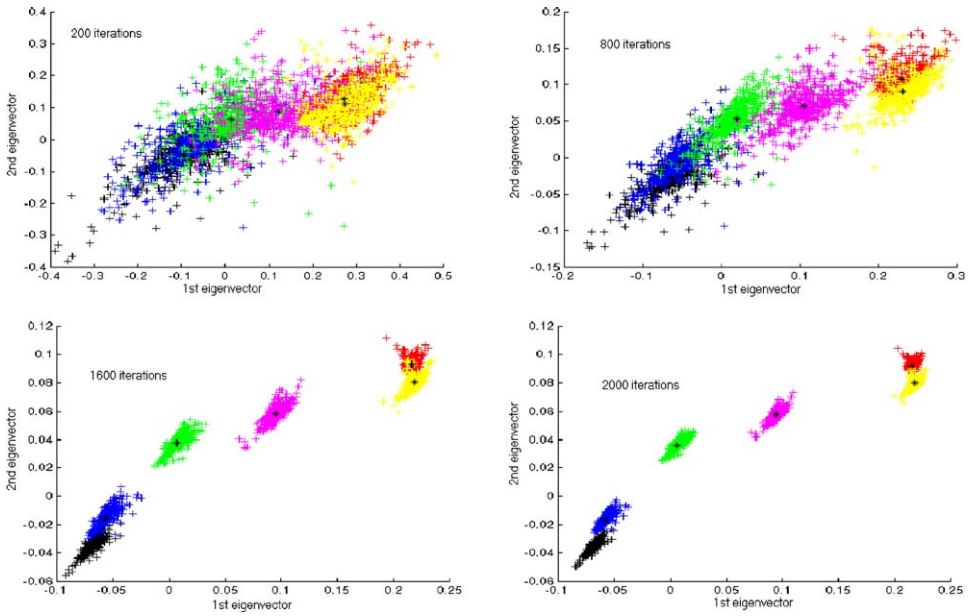


Fig. 6. Discriminant analysis for several iterations of the parameterisation phase.

Fig. 6 illustrates four discriminant analyses for all learning data for iterations 200, 800, 1600 and 2000 of parameterisation. It clearly shows the impact of constraints on the scatter of points, the respective variances of which strongly decrease. The performance is not what one could have expected when looking at Fig. 6. In fact the noticeable decrease of within-classes covariances obtained during parameterisation does not

occur or only very slightly during coding. This would tend to mean that the first layer of the encoder parameter weights is only very slightly modified by this processing.

4.2. From NPC-1 to NPC-2

The *maximal constraint* is a particular case of constraint where we impose:

$$\mathbf{a}_l \leftarrow \bar{\mathbf{a}}_{c_l}, \tag{8}$$

where $\mathbf{a}_l$ means the code vector of an l phoneme with a $c_l \in \{1,\ldots,P\}$ membership class. $\bar{\mathbf{a}}_1,\ldots,\bar{\mathbf{a}}_P$ represent P class centres. Thus only one set of codes remains possible. The particular case of the maximal constraint is interesting as it allows a noticeable improvement of results to be obtained (exceeding 62%). We will study it in details in the following paragraphs under the name of NPC-2 encoder.

5. Discriminant features extraction: the NPC-2 model

The NPC-2 encoder is an extension of NPC model that allows class membership information to be taken into account during the parameters adjustment phase. This can be obtained by limiting the set of weights of the second layer, e.g. the coding coefficients will be limited to one coefficients vector by class of phonemes instead of one by phoneme as in the case of NPC-1. Assuming c_l is the class membership of phoneme l among a set of P possible classes, the cost function given by Eq. (4) must be rewritten as

$$Q(\mathbf{w},\mathbf{a}_1,\ldots,\mathbf{a}_P) = \frac{1}{2}\sum_k \sum_j (y_k - f_{\mathbf{w},\mathbf{a}_j}(\mathbf{x}_k))^2 \delta_{c(\mathbf{x}_k)-j}, \tag{9}$$

so that every sample $\mathbf{x}_k$ belonging to the same phoneme class $c(\mathbf{x}_k)$ will share the same coding vector $\mathbf{a}_{c(\mathbf{x}_k)}$. Once the error prediction has been minimised over all the phoneme sample sequences, the encoder parameters are ready to be used. The coding phase can then be performed, in a similar manner as for NPC-1 coding.

5.1. NPC-2 experimental results

On a phoneme recognition task (basis B1–B3), we compared four encoders which were the LPC, the MFCC, the NPC-1 and the NPC-2 in the same experimental conditions. Recognition rates were obtained after 30,000 classifier learning iterations. We then carried over the results in Table 4 (voiced vowel phonemes, voiced plosive phonemes and unvoiced plosive phonemes).

On can note that the results are roughly equivalent for all coding methods used (LPC, MFCC, NPC-1 and NPC-2) as regards the /*p*/, /*t*/, /*k*/ phonemes. On the other hand, the NPC-2 coding method gives better scores for the /*aa*/, /*ae*/, /*ey*/, /*ow*/ and /*b*/, /*d*/, /*g*/ phonemes (over 60% and 65% recognition rates against 58% and 62% for the MFCC coding). This results are consistent with the fact that phonemes /*aa*/, /*ae*/, /*ey*/, /*ow*/ and /*b*/, /*d*/, /*g*/ are voiced phonemes while /*p*/, /*t*/, /*k*/ phonemes

Table 4
Recognition rates (generalisation set) obtained with MLP classifier for several classes of phonemes

Phonemes	LPC	MFCC	NPC-1	NPC-2
/aa/, /ae/, /ey/, /ow/	55.5%	58%	61.2%	**63%**
/b/, /d/, /g/	57.5%	62.3%	65%	**70.2%**
/p/, /t/, /q/	61%	68%	61.5%	**65%**

are not. The voiced feature emphasises the predictable nature of the phonemes and so gives temporal coding methods (LPC, NPC-1 and NPC-2) an advantage. The nonlinear features present in the speech signal are then better taken into account by the NPC coding. We will see next that the NPC-3 overcomes this difficulty by giving more importance to the classification task against the prediction task.

5.2. NPC-2 validation

We address here the problem of demonstrating how the NPC-2 encoder second layer weights model discriminant informations.

5.2.1. NPC-2 theoretical validation for unoised phonemes

First there is an evidence that the codes extracted from the coding phase carry discriminant phonetic features. Let us suppose that we have two phonemes i and j belonging to two different classes c_i and c_j. If F_i and F_j are the respective predictive models of phonemes i and j such that

$$y_k = \begin{cases} F_i(\mathbf{x}_k) \text{ for all samples } \mathbf{x}_k \text{ of phoneme } i, \\ F_j(\mathbf{x}_k) \text{ for all samples } \mathbf{x}_k \text{ of phoneme } j, \end{cases} \tag{10}$$

then the absolute minimum of the error function is reached when (5) vanishes; assuming that $\mathbf{x}_k$ and y_k are unnoised data, this corresponds to the following result on the network mapping:

$$\forall \mathbf{x}_k \begin{cases} c(\mathbf{x}_k) = i \Rightarrow h_{\mathbf{a}_i} \circ g_{\mathbf{w}}(\mathbf{x}_k) = y_k = F_i(\mathbf{x}_k), \\ c(\mathbf{x}_k) = j \Rightarrow h_{\mathbf{a}_j} \circ g_{\mathbf{w}}(\mathbf{x}_k) = y_k = F_j(\mathbf{x}_k) \end{cases} \tag{11}$$

and this means that for the training data we obtain at least:

$$\begin{aligned} h_{\mathbf{a}_i} \circ g_{\mathbf{w}} &= F_i, \\ h_{\mathbf{a}_j} \circ g_{\mathbf{w}} &= F_j. \end{aligned} \tag{12}$$

Since $g_{\mathbf{w}}$ remains the same for phonemes i and j, it is clear that $F_i \neq F_j$ leads to $h_{\mathbf{a}_i} \neq h_{\mathbf{a}_j}$ and as a consequence, the second layer weights carry the discriminant features. Similar arguments based on the modelling of classes of phonemes lead to the same results for the second layer weights obtained after the parameter setting phase. Assuming that there is one function per phoneme class $F_1, \ldots, F_P$, Eq. (4) will tend

to zero when the absolute minimum has been reached. One has (assuming again that data are unnoised):

$$\forall c \in \{1,\ldots,P\}, \quad h_{\mathbf{a}_c} \circ g_{\mathbf{w}} = F_c, \tag{13}$$

therefore $\mathbf{a}_c$ will carry the discriminant features of classes.

5.2.2. Gaussian noised signals

There exists an important result for the interpretation of the outputs of a network trained by minimising a sum-of-square error function: the output $f(\mathbf{x})$ approximates the conditional averages of the target data y: $f(\mathbf{x}) = \langle y|\mathbf{x}\rangle$ [7,6].

The absolute minimum of the error function occurs for the network mapping

$$f_{\mathbf{w}^\star,\mathbf{a}_j^\star}(\mathbf{x}) = \langle y|\mathbf{x}\rangle_j. \tag{14}$$

Assuming the data is generated from a set of deterministic functions $F_j(\mathbf{x})$ with superimposed zero-mean Gaussian noise b then the target data is given by

$$y = F_j(\mathbf{x}) + b \tag{15}$$

the network output takes the form

$$f_{\mathbf{w}^\star,\mathbf{a}_j^\star}(\mathbf{x}) = \langle y|\mathbf{x}\rangle_j = \langle F_j(\mathbf{x}) + b|\mathbf{x}\rangle_j = F_j(\mathbf{x}) + \langle b\rangle = F_j(\mathbf{x}) \tag{16}$$

since $\langle b\rangle = 0$. Thus the network has averaged over the noise on the data and the result of Section 5.2.1 can be then applied to show that $\mathbf{a}_j$ carries the discriminant features. In this demonstration, we have considered data in which the input vectors $\mathbf{x}_k$ are known exactly, but the target scalar y_k are noisy. This is not the exact situation since we know that $\mathbf{x}_k = [y_{k-1},\ldots,y_{k-\lambda}]$, leading the fact that if y_k is noisy, then $y_{k-1},\ldots,y_{k-\lambda}$ are noisy and then $\mathbf{x}_k$ is noisy. Webb [43] showed that in the situation in which the target data is generated from a smooth function $F_i(\mathbf{x})$ but where the input vector $\mathbf{x}$ is corrupted by additive noise, the optimum solution is again given by the conditional expectation of the target data. Again, theoretical validation of Section 5.2.1 can be applied to show that the $\mathbf{a}_j$'s carry the discriminant features.

In addition to this validation, there are experimental evidences of the discriminant properties of the NPC encoder output weights, as we will see in the coming sections.

5.2.3. NPC distance definition

Let us define $Q(\mathbf{w},\mathbf{a}_j,i)$ as the prediction error computed on the phoneme i samples $\mathbf{x}_k$ using the $\mathbf{a}_j$ class model parameters. According to this definition, $Q(\mathbf{w},\mathbf{a}_i,i)$ is the prediction error of phoneme i using the $\mathbf{a}_i$ parameters, i.e the mean squared error $Q(\mathbf{w},\mathbf{a}_i)$ given by Eq. (5).

Following Itakura's definition of the distance [20], one can define a new distance called the NPC distance between two phonemes i and j as

$$d_{\mathrm{NPC}}(i,j) = \log \frac{Q(\mathbf{w},\mathbf{a}_j,i)}{Q(\mathbf{w},\mathbf{a}_i,i)}. \tag{17}$$

Eq. (17) gives the ratio of the phoneme i prediction error using the c_j class model parameters $\mathbf{a}_{c_j}$ and the same phoneme prediction error, but using the c_i class model

parameters. When applying the i phoneme signal to the NPC with its adapted coding coefficients $\mathbf{a}_i$, the output residual error $Q(\mathbf{w}, \mathbf{a}_j, i)$ is minimal. On the other hand, when applying the same signal to the NPC with the adapted coding coefficients $\mathbf{a}_j$ of the j phoneme signal, the residual error $Q(\mathbf{w}, \mathbf{a}_j, i)$ is not minimal and one obtains $Q(\mathbf{w}, \mathbf{a}_j, i) \geqslant Q(\mathbf{w}, \mathbf{a}_i, i)$ and for $i = j$, $d_{\mathrm{NPC}}(i, j) = 0$. This ratio $Q(\mathbf{w}, \mathbf{a}_j, i)/Q(\mathbf{w}, \mathbf{a}_i, i)$ is called likelihood ratio and $d_{\mathrm{NPC}}(i, j)$ the *log likelihood ratio distortion measure* (this is not a true distance since it is not symmetrical).

5.2.4. Evaluation of the NPC log modelling error ratio

Now the question is to see whether the inequality $Q(\mathbf{w}, \mathbf{a}_j, i) \geqslant Q(\mathbf{w}, \mathbf{a}_i, i)$ is well verified, e.g. whether the NPC coding coefficients model the discriminant features of coded phonemes. Focusing on NPC-2 discriminant properties, we generalised the NPC distance to define the NPC *log modelling error ratio* (log MER) as

$$\Gamma_{\mathrm{NPC}} = \log \frac{\sum_{i=1}^{P} \sum_{j=1, j \neq i}^{P} Q(\mathbf{a}_j, i)}{(P-1) \sum_{i=1}^{P} Q(\mathbf{w}, \mathbf{a}_i, i)}. \tag{18}$$

One should obtain $\Gamma_{\mathrm{NPC}} \geqslant 0$, once the parameter setting phase has been completed. We carried out experiments on the tree different bases of phonemes B1–B3 which results are illustrated in Figs. 7–9. We computed the NPC modelisation error ratio at every learning iteration. It can been seen from figures that the initial values are close to 1 (the log has been omitted). This indicates that the initial status of NPC encoder (initial weights are randomly chosen) corresponds to an encoder without any features extraction capabilities. While the prediction error decreases, the NPC modelisation error ratio simultaneously increases showing a rise in the discriminant capabilities of the encoder.

5.2.5. Phonemes over-modelisation

One can also note in those figures (Figs. 7–9) that the discrimination performance first reached a maximal value and then it started a progressive degradation. This could indicate that an excess number of learning iterations could be prejudicial, leading to a decay in performance. In fact, we have well observed this loss of performance on classification rates (see Section 3.4). During the coding phase, we stopped the encoder optimisation for several iterations numbers. We computed the recognition rate for each of these numbers with the MLP classifier. Tables 5 and 6 show the recognition rates (generalisation set) versus the coding iterations number. They show that there exists an optimal iteration number between 5 and 30. Others simulations showed the stability of this number. Over fitting is a well-known problem in neural networks. To circumvent the difficulty, one can use different traditional strategies as the early stopping, or adding a regularisation term to the error criterion. In our case, we propose to use the MER measure on a small data set, which belonging classes are known, to evaluate the optimal coding iteration number (this solution is currently studied).

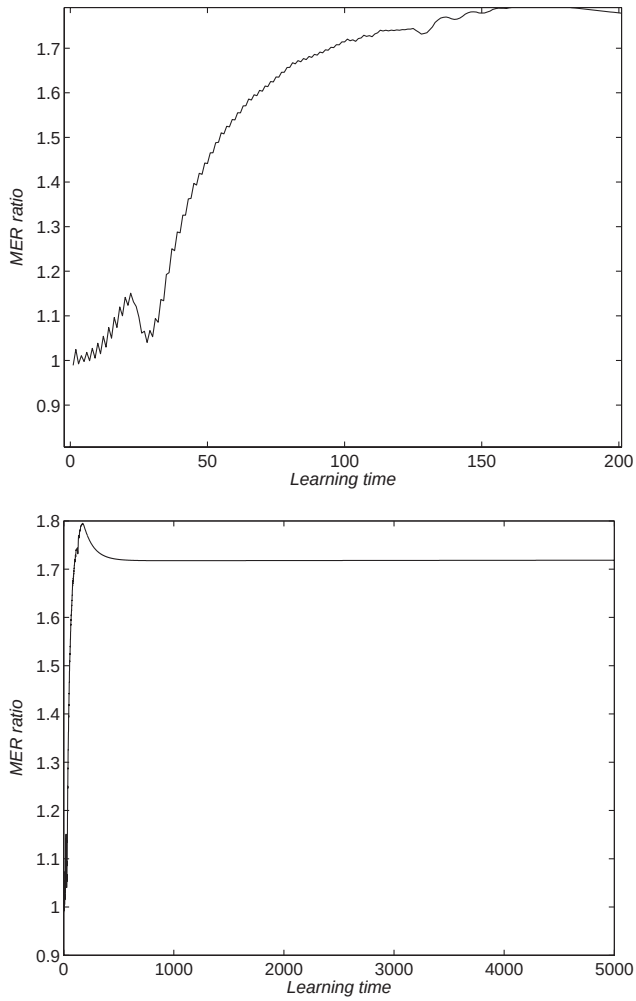


Fig. 7. NPC MER during the parameter learning phase (phonemes /*aa*/, /*ae*/, /*ey*/, /*ow*/), top: the first 200 learning iterations, bottom: 5000 learning iterations.

6. The DFE–NPC model

In Section 4, we tried to improve the discriminant capacities of encoder NPC-1 by minimising the within-class covariance of generated code vectors. We obtained the definition of NPC-2 encoder given in Section 5 that allows a better integration of this constraint to be made. We will examine in this section the second option consisting in maximising the between-class covariance of the generated codes to improve the discriminant capacities of the encoder. We will use the NPC metric previously defined

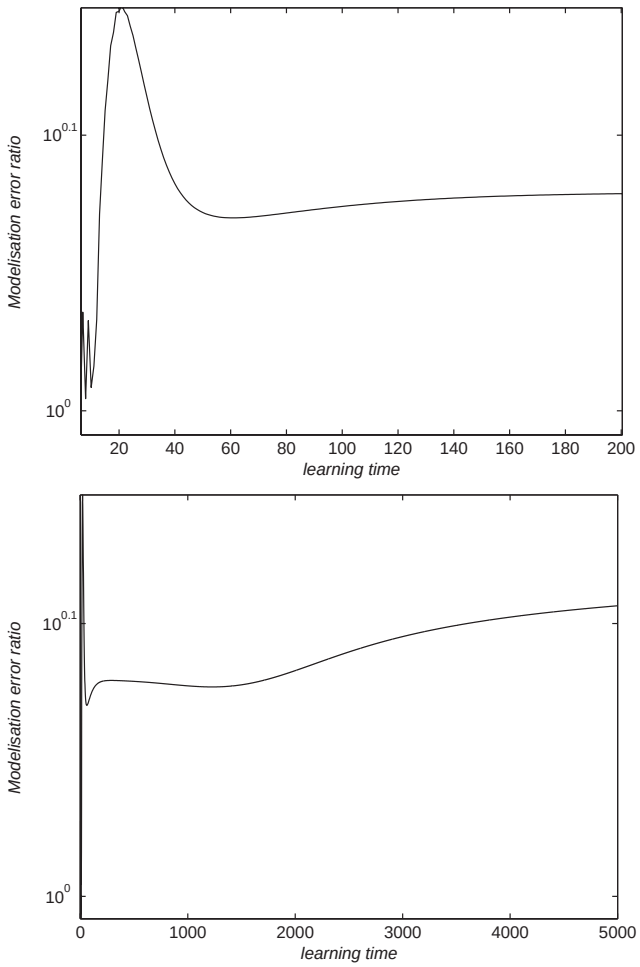


Fig. 8. NPC MER during the parameter learning phase (phonemes /*b*/, /*d*/, /*g*/), top: the first 200 learning iterations, bottom: 5000 learning iterations.

in Section 5 instead of the euclidian metric between the generated codes, as used in Section 4: experiments based on this last method did not make it possible to improve results significantly for classification. This might lead us to think that although constraints applied to codes during the parameters adjustment phase are tending to improve the between-class covariance, conversely they are in contradiction with adaptations issued from the minimisation of the prediction error. So the discriminant capacities of the encoder are not improved after the coding phase. However, we will see in this section that it is possible to optimise discriminant constraints by privileging those which also tend to minimise the prediction error.

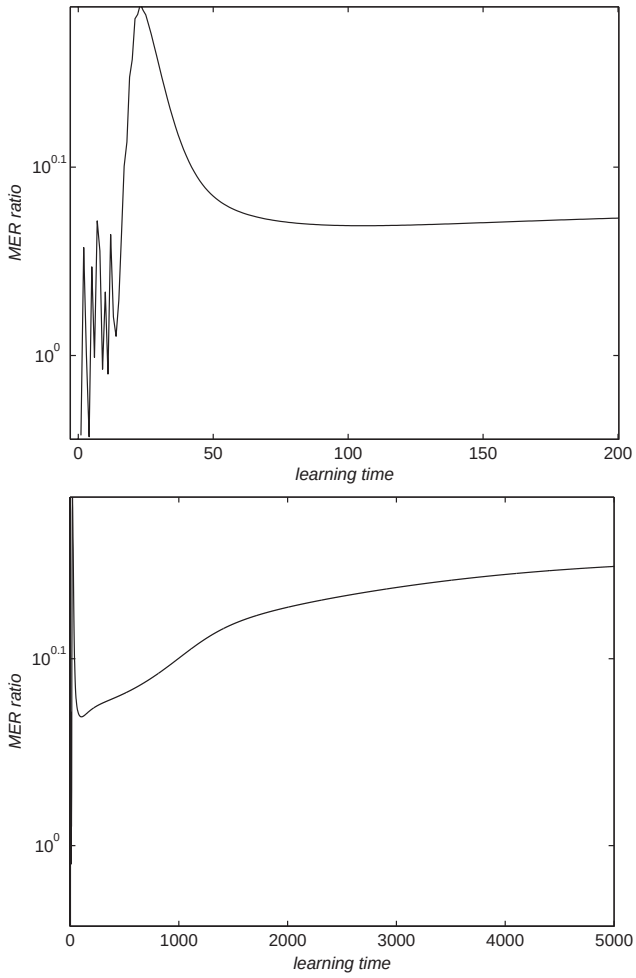


Fig. 9. NPC MER during the parameter learning phase (phonemes /*p*/, /*t*/, /*q*/), top: the first 200 learning iterations, bottom: 5000 learning iterations.

6.1. Maximisation of log modelisation error ratio (LMER)

We can consider the MER defined by Eq. (18) as an estimate of the between-class covariance of the aid for a new metrics, the d_{NPC} metrics, instead of the Euclidean metrics. Thus, maximisation of this variance leads to impose a variation to the code weights which will maximise the difference between the NPC prediction models of the various phonemes.

Let us consider the definition we gave for the LMER (Eq. (18)) and write it as

$$\Gamma_{\mathrm{NPC}} = \log \frac{Q^d}{(P-1)Q^m}. \tag{19}$$

Table 5
NPC-2 performance for several iterations number of the coding process (/*aa*/, /*ae*/, /*ey*/, /*ow*/ phonemes)

Iteration number	5	10	20	30	50	100	150	200	500
Recognition rate (%)	59.2	62	**62.6**	61.8	60.3	59.8	59.5	58.5	58

Table 6
NPC-2 performance for several iterations number of the coding process (/*b*/, /*d*/, /*g*/ phonemes)

Iteration number	5	10	20	30	50	100	150	200	500
Recognition rate (%)	**70**	64	56	55		49.6	48	48.1	48

The new function to minimise $Q_{\mathrm{DFE-NPC}}$ is now $\Gamma_{\mathrm{NPC}}^{-1}$. The law for the modification of any a weight in the second layer (code weight) or ω for the first layer (parameter weight) in parameterisation is proportional to the opposite of the cost function gradient, reverse of LMER:

$$a^{(q+1)} = a^{(q)} - \mu \frac{\partial}{\partial a}\left(\frac{1}{\Gamma_{\mathrm{NPC}}}\right) \tag{20}$$

with

$$\frac{\partial}{\partial a}\left(\frac{1}{\Gamma_{\mathrm{NPC}}}\right) = \frac{P-1}{\Gamma_{\mathrm{NPC}}^2 Q^d Q^m}\left[Q^d \frac{\partial Q^m}{\partial a} - Q^m \frac{\partial Q^d}{\partial a}\right]. \tag{21}$$

We simplified the $1/\Gamma_{\mathrm{NPC}}$ cost function to be minimised. By minimising MER rather than LMER, we do not change the principle of the discrimination error for minimising, but we obtain a simpler expression:

$$\frac{\partial}{\partial a}\left(\frac{1}{\Gamma_{\mathrm{NPC}}}\right) = \frac{P-1}{Q^d}\left[\frac{\partial Q^m}{\partial a} - \frac{1}{\Gamma}\frac{\partial Q^d}{\partial a}\right]. \tag{22}$$

We obtain a modification law composed of two terms. With the first one we return to the principle of prediction error minimisation. The second one, of opposite sign, corresponds to the maximisation of the discrimination error. For coding, we use the NPC-1 model adaptation law that remains unchanged for the computation of code weights.

So as to retain an influence on the prediction/discrimination weighting, we finally retained

$$\frac{\partial}{\partial a}\left(\frac{1}{\Gamma_{\mathrm{NPC}}}\right) = (P-1)\left[\alpha \frac{\partial Q^m}{\partial a} - (1-\alpha)\frac{\partial Q^d}{\partial a}\right]. \tag{23}$$

6.2. Experimental validation of DFE–NPC encoder

Fig. 10 illustrates the evolution of MER during the encoder parameterisation phase on B1L base, for voiced phonemes. If we compare this evolution for two respective

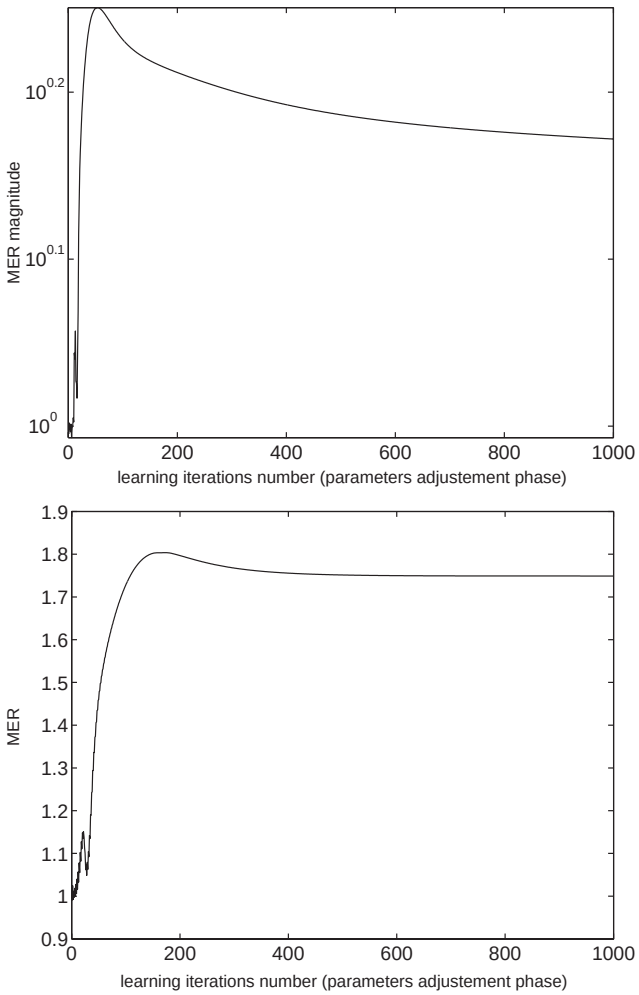


Fig. 10. MER evolution during the parameters adjustment phase: $\alpha = 1$ (top, NPC-2) and $\alpha = 0.5$ (bottom, NPC-3).

values of α weighting: with $\alpha = 1$, we find again the quadratic cost function of NPC-2 encoder; with $\alpha = 0.5$, we obtain a DFE–NPC encoder with a contribution evenly distributed between the prediction error and the discrimination error. We note that in the same experimentation conditions, the MER is higher for NPC-3 encoder than for NPC-2 encoder. It would be interesting to see whether this result can be found again for "Euclidean" between-class covariances estimated on both encoders. Fig. 11 reports the result of simulations carried out to this effect. The between-class covariance is estimated in this result for each iteration of parameterisation. We can see that indeed, in DFE–NPC case the between-class covariance converges to a higher value than in NPC-2

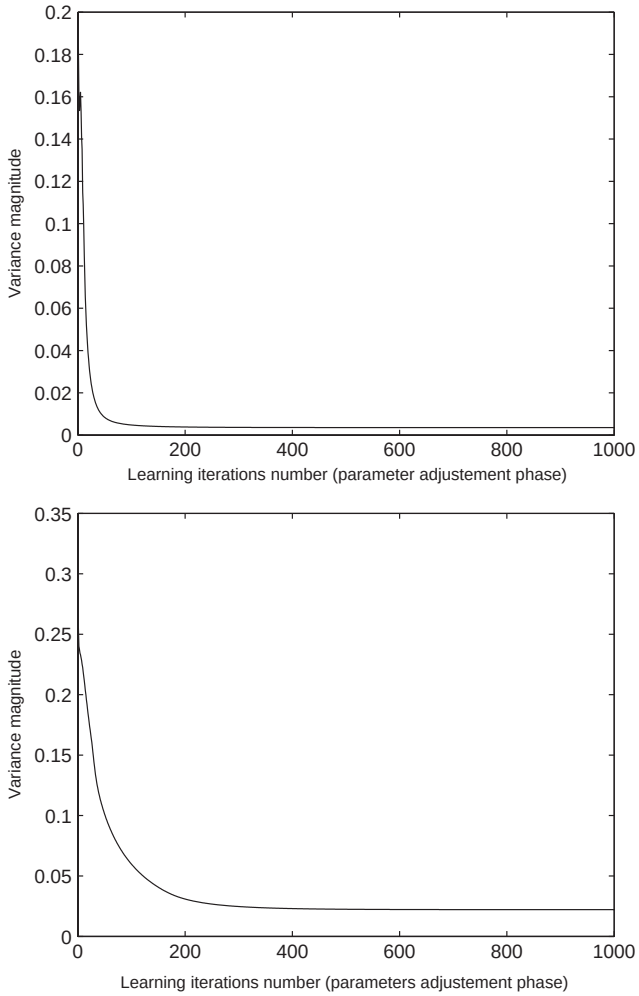


Fig. 11. Inter-classes variance evolution during the parameters adjustment phase: $\alpha = 1$ (top, NPC-2) and $\alpha = 0.5$ (bottom, DFE–NPC).

case. We thus show experimentally that maximising the MER leads to maximising the between-class covariance and therefore to a better representation space for classification codes.

This new representation space for codes, linked to DFE–NPC encoder leads to a noticeable improvement of performances for classification. To this end we renewed the previous experiments for classification on bases B1–B3 and we obtained the scores shown in Fig. 12.

Results obtained on bases B1 and B2 (voiced phonemes) are better than for all previous coding. However NPC-2 encoder already gave good results for the same data.

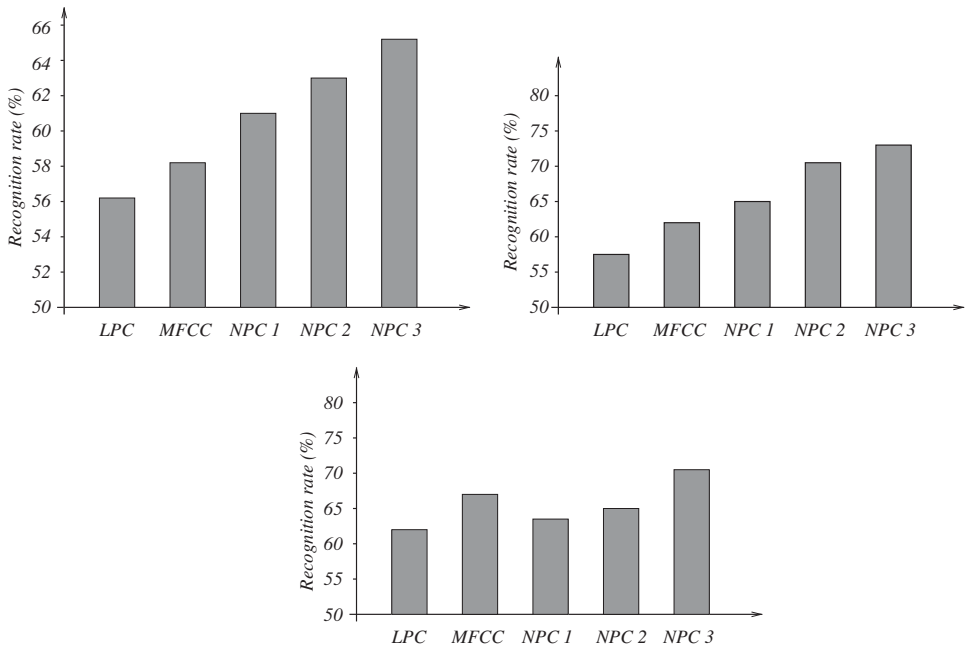


Fig. 12. Recognition rates (generalisation set) obtained with MLP classifier for /*aa*/, /*ae*/, /*ey*/, /*ow*/ phonemes, /*b*/, /*d*/, /*g*/ phonemes and /*p*/, /*t*/, /*k*/ phonemes (NPC-3 coding).

Conversely, the improvement brought by DFE–NPC encoder on base B3 (/*p*/, /*t*/, /*k*/ phonemes) is remarkable, as it is specific to this encoder. In fact, let us recall that NPC-2 encoder does not give a good performance on this same base (cf. Section 5). These phonemes are unvoiced and their less predictable nature put in default temporal encoders. The DFE–NPC model is better adapted as the weak minimisation of the prediction error can be compensated by the strengthening of the discrimination error (since the cost function is the ratio of the prediction cost over the discrimination cost).

The optimal value of α weighting is not easy to define as it varies according to phoneme classes to be studied. Within the context of these simulations, we therefore chose an adaptive weighting. The adaptation law concerned obeys the rule according to which the minimisation of the prediction error (α close to 1) prevails over the maximisation of the discrimination error (α close to 0), at least as long as the prediction error is sharply decreasing. Fig. 13 gives an example of the evolution of α adaptive parameter during the encoder parameterisation phase of the encoder on base B1.

7. Conclusion

We showed in this article the interest of using nonlinear coding methods to code the speech signal (improvement of scores obtained for phoneme recognition). We have

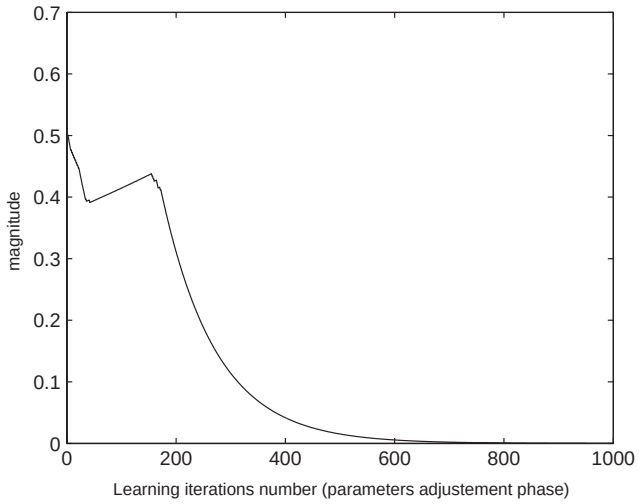


Fig. 13. Evolution of the α parameter during the parameters adjustment phase (phonemes /*aa*/, /*ae*/, /*ey*/, /*ow*/).

presented a new nonlinear coding model to code the speech signal, the NPC model, and also three of its declinations. The latter enables us to classify the encoder in the DFE coding methods category (taking into account information on membership class as from the coding stage). Results of the experiments described in this article have shown that recognition rates have clearly improved. We also proposed a theoretical validation (for unnoised signals and Gaussian noised signals) that allowed us to demonstrate the appropriateness of the model. By combining these results with the NPC encoder, we defined a new nonlinear distance between phonemes, the NPC distance. We then extended this definition to that of the distance between several points called modelisation Error Ratio (MER). As MER permits to analyse the discriminant properties of the encoder, we were able to support our study of the model with an experimental validation. Finally, we have proposed the MER as a new cost function (the DFE-NPC model). Results obtained put in obviousness a noticeable improvement of performances for classification.

Our current studies are devoted to the coding of a larger number of phoneme classes. Lastly, we now focus on a study dealing with the application of NPC coding to the compression/decompression of the speech signal.

References

[1] K. Aikawa, H. Singer, H. Kawahara, Y. Tohkura, A dynamic cepstrum incorporating time-frequency masking and its application to continuous speech recognition, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 2, Minneapolis, MN, USA, 1993, pp. 668–671.

[2] M. Bacchiani, K. Aikawa, Optimization of time-frequency masking filters using the minimum classification error criterion, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 2, Adelaide, South Australia, 1994, pp. 197–200.
[3] A. Biem, Neural models for extracting speaker characteristics in speech modelization system, Ph.D. Thesis, Paris VI, 1997.
[4] A. Biem, S. Katagiri, Feature extraction based on minimum classification error/generalized probabilistic descent method, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 2, Minneapolis, MN, USA, 1993, pp. 275–278.
[5] A. Biem, S. Katagiri, Filter bank design based on discriminative feature extraction, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 1, Adelaide, South Australia, 1994, pp. 485–488.
[6] C.M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.
[7] C.M. Bishop, Novelty detection and neural network validation, in: IEE Proceedings: Vision, Image and Signal Processing (Special Issue on Applications of Neural Networks) Vol. 141, 1994, pp. 217–222.
[8] C. Chavy, B. Gas, J.L. Zarader, Discriminative coding with predictive neural networks, in: International Conference on Artificial Neural Network, Edinburgh, UK, 1999.
[9] W.S. Cleveland, S.J. Devlin, Locally weighted regression: an approach to regression analysis by local fitting, Am. Stat. Assoc. 83 (9) (1988) 596–610.
[10] S.B. Davis, P. Melmerstein, Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. Acoust. Speech Signal Process. 28 (4) (1980) 357–376.
[11] A. de la Torre, A.M. Peinado, A.J. Rubio, V.E. Sánchez, J.E. Díaz, An application of minimum classification error to feature space transformations for speech recognition, Speech Commun. 20 (1996) 273–290.
[12] F. Díaz-de-María, A.R. Figueiras-Vidal, Nonlinear prediction for speech coding using radial basis functions, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 1, Detroit, MI, USA, 1995, pp. 788–791.
[13] G. Dreyfus, O. Macchi, S. Marcos, O. Nerrand, L. Personnaz, Roussel-Ragot, D. Urbani, C. Vignat, Adaptive training of feedback neural networks for nonlinear filtering, Neural Networks Signal Process. 2 (1992) 550–559.
[14] R.O. Duda, P.B. Hart, Pattern Classification and Scene Analysis, 2nd Edition, Wiley, New York, 2000.
[15] K.S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, New York, 1974.
[16] B. Gas, J.L. Zarader, C. Chavy, A new approach to speech coding: The neural predictive coding, J. Adv. Comput. Intell. 4 (1) (2000) 120–127.
[17] B. Gas, J.L. Zarader, P. Sellem, J.C. Didiot, Speech coding by limited weights neural network, in: IEEE International Conference on Systems Man and Cybernetics, Orlando, FL, USA, 1997.
[18] H. Hermansky, Perceptual linear predictive (plp) analysis of speech, J. Acoust. Soc. Am. 4 (1990) 1738–1752.
[19] H. Hermansky, N. Morgan, Rasta processing of speech, IEEE Trans. Speech Audio Process. 2 (1994) 587–589.
[20] F. Itakura, Minimum prediction residual principle applied to speech recognition, IEEE Trans. Acoust. Speech and Signal Process. 23 (1975) 67–72.
[21] B.H. Juang, S. Katagiri, Discriminative learning for minimum error classification, IEEE Trans. Signal Process. 40 (12) (1992) 3043–3054.
[22] S. Katagiri, Handbook of Neural Networks sor Speech Processing, Artech House, Norwood, MA, 2000.
[23] T. Kawahara, S. Doshita, Phoneme recognition by combining discriminant analysis and HMM. in: Proceedings of International Conference on Signal and Speech Processing, Vol. 1, Toronto, Ont., Canada, 1991, pp. 557–560.
[24] T. Kawahara, T. Ogawa, S. Kitazawa, S. Doshita, Phoneme recognition by combining bayesian linear discriminations of selected pairs of classes, in: Proceedings of International Conference on Signal and Speech Processing, Albuquerque, NM, USA, 1990, p. 78.
[25] T. Kohonen, The 'neural' phonetic typewriter, IEEE Comput. 21 (3) (1988).
[26] T. Kohonen, et al., Phonotopic maps insightful representation of phonological features for speech recognition, in: Proceedings of International Conference on Pattern Recognition, Vol. 7, 1984.

[27] K.J. Lang, A.H. Waibel, G.E. Hinton, A time-delay neural network architecture for isolated word recognition, Neural Networks 3 (1990) 23–43.
[28] A. Lapedes, R. Farber, Nonlinear signal processing using neural networks: prediction and system modelling, Internal Report, Los Alamos National Laboratory, 1987.
[29] J.H. Lee, H.Y. Jung, T.W. Lee, S.Y. Lee, Speech feature extraction using independent component analysis, in: International Conference on Signal and Speech Processing, Vol. 3, Istanbul, Turkey, 2000, pp. 1631–1634.
[30] S.E. Levinson, Structural methods in automatic speech recognition, Proc. IEEE 73 (11) (1987) 1625–1650.
[31] N. Ma, G. Wei, Speech coding with nonlinear local prediction model, in: Proceedings of International Conference on Acoustic, Speech and Signal Processing, Vol. 2, Seattle, WA, USA, 1998, pp. 1101–1104.
[32] J.D. Markel, Linear Prediction of Speech, Springer, Berlin, Heidelberg, New York, 1976.
[33] B. Picinbono P. Chevalier, P. Duvaut, Le filtrage de volterra transverse rel et complexe en traitement du signal, Traitement du Signal 7 (5) (1990) 451–476.
[34] G. Saon, M. Padmanabhan, R. Gopinath, S. Chen, Maximum likelihood discriminant feature spaces, in: International Conference on Signal and Speech Processing, Vol. 2, Istanbul, Turkey, 2000, pp. 1229–1132.
[35] A.C. Singer, G.W. Wornell, A.V. Oppenheim, Codebook prediction: a nonlinear signal modeling paradigm, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 5, San Francisco, CA, USA, 1992, pp. 325–328.
[36] M. Thomae, G. Ruske, T. Pfau, A new approach to discriminative feature extraction using model transformation, in: International Conference on Signal and Speech Processing, Vol. 3, Istanbul, Turkey, 2000, pp. 1615–1618.
[37] J. Thyssen, H. Nielsen, S.D. Hansen, Non-linear short-term prediction in speech coding, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 1, Adelaide, South Australia, 1994, pp. 185–188.
[38] N. Tishby, A dynamical system approach to speech processing, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 1, Albuquerque, NM, USA, 1990, pp. 365–368.
[39] B. Townshend, Nonlinear prediction on speech, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 1, Toronto, Ont., Canada, 1991, pp. 425–428.
[40] The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus (TIMIT), Speech Disc 1-1.1/NTIS PB91-505065, October 1990.
[41] A.H. Waibel, T. Hanazawa, G.E. Hinton, K. Shikano, K.J. Lang, Phoneme recognition using time-delay neural networks, IEEE Trans. Acoustic, Speech, and Signal Processing 37 (3) (1989) 328–339.
[42] H. Wassner, G. Chollet, New time-frequency derived cepstral coefficients for automatic speech recognition, in: EUSIPCO'96, Trieste, Italy, 1996.
[43] A.R. Webb, Functional approximation by feed-forward networks: a least square approach to generalisation, IEEE Trans. Neural Networks 5 (3) (1994) 363–371.
[44] B. Widrow, 30 years of adaptative neural networks: perceptron, madaline and backpropagation, Proceedings IEEE 78 (1990) 1415–1442.
[45] S.A. Zahorian, D. Qian, A.J. Jagharghi, Acoustic-phonetic transformations for improved speaker-independent isolated word recognition, in: Proceedings of International Conference on Signal and Speech Processing, Vol. 1, Toronto, Ont., Canada, 1991, pp. 561–564.
[46] J.L. Zarader, B. Gas, J.C. Didiot, P. Sellem, Neural predictive coding: application to phoneme recognition, in: International Conference on Neural Information Processing, Dunedin, Otago, New Zealand, 1997.

A.4 Article de conférence [37]

Modular Neural Predictive Coding for Discriminative Features Extraction

M. Chetouani and B. Gas and J.L. Zarader

IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP'03), Vol. 2, p. 33–36 (2003)

.

MODULAR NEURAL PREDICTIVE CODING FOR DISCRIMINATIVE FEATURES EXTRACTION

M. Chetouani, B. Gas, J.L. Zarader

Laboratoire des Instruments et Systèmes d'Ile-De-France
Université Paris VI
BP 164, Tour 22-12 2ème étage
4 Place Jussieu, 75252 Paris Cedex 05
France
mohamed.chetouani@lis.jussieu.fr gas@ccr.jussieu.fr zarader@ccr.jussieu.fr

ABSTRACT

In this paper, we present an architecture called the Modular Neural Predictive Coding Architecture (Modular NPC). The Modular NPC is used for Discriminative Feature Extraction (DFE). It provides an architecture based on phonetics knowledge applied to phoneme recognition. The phonemes are extracted from the Darpa-Timit speech database. Comparisons with coding methods (LPC, MFCC, PLP) are presented: they put in obviousness an improvement of the recognition rates.

1. INTRODUCTION

In the aim of improving the speech recognition task, several ways can be chosen. One of them is to improve the feature extraction stage. In fact, recent works shown the importance of this stage [1], [2],[3]. The feature extraction is commonly made by temporal methods like Linear Predictive Coding (LPC) or cepstral methods like Mel Frequency Cepstral Coding (MFCC). Human auditory knowledge like Perceptual Linear Predictive coding (PLP) [4] are also often used. The problem with these classical methods is the lack of discrimination. Indeed, there is no explicit mechanism which discourages the models from resembling each other.

The principal method for introducing discrimination is called the Discriminative Feature Extraction (DFE) based on the Minimum Classification Error (MCE) criterion [1]. The key idea of DFE method is that the feature extraction and the classification stage can be simultaneously trained in order to improve the pattern recognition system.

There is another strategy for DFE implementation. It consists in the independent training of both the feature extractor and the classifier [5] [2]. This method is more adapted for complex problems. Indeed, during the simultaneously training of the two stages, the evolution of the feature extractor parameters is small compared to the classifier parameters [2]. The feature extractor has to be trained with a criterion which measure the discrimination power of selected features. For example, the criterion can be the Maximization of the Mutual Information (MMI) between the the features and the class labels [3].

In this paper, we present a Modular Neural Predictive Coding (NPC) model for speech Discriminative Feature Extraction (DFE). First, The DFE-NPC model is introduced. The section 3 describes the Modular NPC architecture. The experimental setup are given in the section 4 and the results on phonemes recognition task are given in the section 5. Finally, we give conclusions from the proposed work.

2. THE DFE-NEURAL PREDICTIVE CODING

The Neural Predictive Coding (NPC) [6] is an extension of the Linear Predictive Coding (LPC) to the nonlinear area. The NPC model is based on a feedforward multi-layer perceptron used as a nonlinear predictor (cf. Fig.1). This strategy is consistent with the fact that speech production is known to be nonlinear [7].

2.1. The NPC model

Let L being the length of the prediction window. The Non Linear Auto-Regressive (NLAR) model computed by the NPC model is the follow:

$$\hat{y}_k = F(\mathbf{y}_k) \tag{1}$$

Where k is the index of samples and $\mathbf{y}_k$ is the prediction context: $\mathbf{y}_k = [y_{k-1}, y_{k-2}, ..., y_{k-L}]^T$.

F is a nonlinear function composed by two functions $G_{\mathbf{w}}$ ($\mathbf{w}$ first layer weights) and $H_{\mathbf{a}}$ ($\mathbf{a}$ output layer weights):

$$F_{\mathbf{w},\mathbf{a}}(\mathbf{y}_k) = H_{\mathbf{a}} \circ G_{\mathbf{w}}(\mathbf{y}_k) \tag{2}$$

With $\hat{y}_k = H_{\mathbf{a}}(\mathbf{z}_k)$ and $\mathbf{z}_k = G_{\mathbf{w}}(\mathbf{y}_k)$.

The NPC model has the major advantage to allow a nonlinear modelisation with an arbitrary limited number of coding coefficients. The key idea of the NPC-2 [6], an extension of the NPC model, is to allow an arbitrary number of coding coefficients by creating a second layer for each phoneme class. The first layer remaining the same for all the classes. The cost function is defined as:

$$Q_{NPC-2} = \sum_i \sum_k \sum_l (y_{i,k} - F_{\mathbf{w},\mathbf{a}_l}(\mathbf{y}_{i,k}))^2 \delta_{c_i - l} \quad (3)$$

C_i is the class membership of the phoneme i among a set of M classes. $F_{\mathbf{w},\mathbf{a}_l}$ is one of the M functions corresponding to the $\mathbf{a}_l$ output layer weights. The Kronecker symbol δ associates the class C_i to the output layer l. Output layers are proper to each phoneme, they are the coding coefficients.

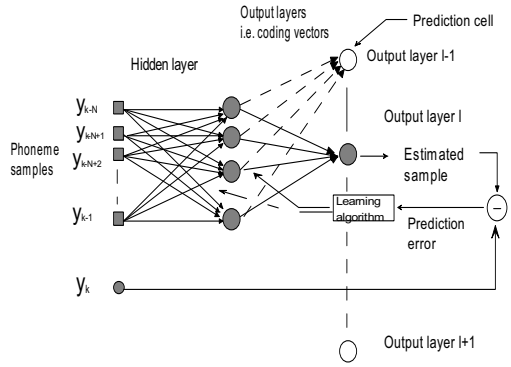


Fig. 1. Architecture of the Neural Predictive Coding model

The learning process needs to be realized in two phases: the *parameters adjustment phase* and the *coding phase*. During the first phase, all the network weights are estimated from a learning set composed of phonemes belonging to the M classes. Next, the output layers weights are non longer used while the hidden layer weights become the encoder *parameters*. Then, during the *coding phase*, the network works as a two layers perceptron composed of the hidden layer previously computed and one output cell. The *coding phase* consists in the estimation of the output weights which are the NPC coding coefficients.

2.2. NPC-2 feature extraction principle

The aim of the NPC model is to manage to compute discriminant output layer weights. These weights $\mathbf{a}_l$, the coding vectors, have to carry discriminant phonetic features. The first layer $\mathbf{w}$ weights are common to all the phonemes.

Considering two phonemes i and j belonging to two different classes C_i and C_j, the NPC-2 models associated to the two phonemes are the following:

$$\begin{cases} F_{\mathbf{w},\mathbf{a}_i} = H_{\mathbf{a}_i} \circ G_{\mathbf{w}} \\ F_{\mathbf{w},\mathbf{a}_j} = H_{\mathbf{a}_j} \circ G_{\mathbf{w}} \end{cases} \quad (4)$$

The NPC-2 models $F_{\mathbf{w},\mathbf{a}_i}$ and $F_{\mathbf{w},\mathbf{a}_j}$ are different whereas $G_{\mathbf{w}}$ is common to the two phonemes i and j. This function remains common features and the discrimminant features are carried by the discriminant functions $H_{\mathbf{a}_i}$ and $H_{\mathbf{a}_j}$.

After the computation of all the phonemes belonging to the M classes, one could make the same conclusion for the features extracted from each classes. The coding vectors associated to each classes carry discriminant features while the first layer weights carry common features.

2.3. Maximization of the Modelisation Error Ratio

The principal problem with predictive approach is the lack of discrimination: predictive models are trained independently of each other. As a result, there is no explicit discrimination between the models. In order to solve this problem we developed a measure of discrimination between NPC-2 models: the Modelisation Error Ratio (MER)[6]. L_j^i is the prediction error computed on the phoneme i using the NPC-2 model $H_{\mathbf{a}_j}$ associated to the phoneme j:

$$L_j^i = \sum_k (y_{i,k} - H_{\mathbf{a}_j} \circ G_{\mathbf{w}}(\mathbf{y}_{i,k}))^2 \quad (5)$$

The MER is the inverse ratio of the prediction error of the phoneme i predicted by the correct NPC-2 model to the prediction errors of the phoneme i predicted by the others NPC-2 models:

$$\Gamma = \frac{Q^d}{(M-1)Q^m} \quad (6)$$

With $Q^d = \sum_{i=1}^M \sum_{j=1, j\neq i}^M L_j^i$ and $Q^m = \sum_{i=1}^M L_i^i$.

The DFE-NPC [6] optimization is based on the maximization of the MER:

$$Q_{DFE-NPC} = \frac{1}{\Gamma} \quad (7)$$

The modification law of any a or w weights is proportional to the gradient of $Q_{DFE-NPC}$ (7):

$$\frac{\partial}{\partial a}(\frac{1}{\Gamma}) = \frac{M-1}{Q^d}(\frac{\partial Q^m}{\partial a} - \frac{1}{\Gamma}\frac{\partial Q^d}{\partial a}) \quad (8)$$

The maximization of the Modelisation Error Ratio allows Discriminative Feature Extraction (DFE), this optimization is called the DFE-NPC.

3. MODULAR NPC

The DFE can be improved by opting for a "divide and conquer" method: a hard problem is broken up into a set of easier problems. This principle is not new in speech processing. The Hierarchical Mixtures of Experts (HME) [8]

is one of the examples of the implementation of the principle "divide and conquer". The HME is a set of "expert networks" which are trained on different parts of the input space. The outputs are combined by a "gating network" trained to select the expert which is adapted to the part of the problem.

Commonly, the feature extraction is carried out in the same way for all the phonemes in spite of the differences. Indeed, there is many kinds of differences like the voicing for example. The key idea of the Modular NPC is to provide an architecture which allows to better process the phonemes. This is done by grouping the phonemes which have closed features. Then an expert in the feature extraction process of each group provides discriminant features.

3.1. Description

The method used for the feature extraction decomposition used is known as the "soft split" method [8]. It consists in dividing the phoneme recognition task into sub-problems which have common elements. This division is guided by phonetics knowledge. The phonemes which have common elements are grouped in the same macro-class. The group classification is similar to phoneme classification in phonetic.

The principle of the Modular NPC architecture is to guide the phoneme by "gating networks" (based on macro-classifiers) to the "expert", a DFE-NPC encoder expert in the feature extraction of this phoneme. The Modular NPC architecture is organized as a tree (see table 1).

Macro-Classifier	Node	Classes
Level 1	1	Voiced / Unvoiced
Level 2	1	Vowels / Consonants
	2	Plosives / Fricatives (Unvoiced)
Level 3	1	Vowels-Diphthongs / Semi-Vowels
	2	Nasals-Liquids / Plosives-Fricatives (Voiced)
Level 4	1	Vowels/Diphthongs
	2	Nasals /Liquids
	3	Plosives/Fricatives (Voiced)
Level 5	1	Front/ Central /Back Vowels

Table 1. Description of the Modular Architecture

3.2. Macro-classification

Instead of training a DFE-NPC by incorporating class information, we trained it by incorporating macro-class informations. Note that the same discriminant algorithm (maximization of the MER) is used. Considering a macro-classifier τ which the function is to discriminate between Ω macro-classes, the cost function is defined as:

$$L = \sum_i \sum_k \sum_l (y_{i,k} - \Phi_{\mathbf{w},\mathbf{a}_{\Omega_l}}(\mathbf{y}_{i,k}))^2 \delta_{\Omega_i - l} \qquad (9)$$

Ω_i is the macro-class membership of the phoneme i. $\Phi_{\mathbf{w},\mathbf{a}_l}$ is one of the Ω functions.

Unlike the NPC model, the codes resulting from the *parameters adjustment phase* are used for the classification. The macro-classification is done by a predictive classification method:

$$\Omega_i = \arg\min_{\Omega} \sum_k \sum_l (y_{i,k} - \Phi_{\mathbf{w},\mathbf{a}_{\Omega_l}}(\mathbf{y}_{i,k}))^2 \qquad (10)$$

Once the macro-classification is achieved, the phoneme is directed towards an "DFE-NPC expert" which provides a vector code representing the phoneme.

4. EXPERIMENTAL CONDITIONS

In order to evaluate the DFE power of the Modular NPC, phoneme recognition experiments are performed on this architecture. The different phonemes are extracted belonging to the Darpa-Timit database. The phonemes are extracted from all the speakers from the first region (New England) in order to produce a multi-speaker environment. Depending on their duration, each phoneme is split into a number of frames: the length of analysis windows is 256 samples with an overlapping factor of 128 samples. For each class the number of frames is set to 300.

We made comparisons between the Modular NPC and traditional coding methods: LPC, MFCC and PLP coding methods. The dimension of the coding vectors is set to 12.

The classifier used to estimate the performance of all the encoders is a multi-layer perceptron (MLP) with 12 inputs (the coding vectors dimension), 10 neurons and as many outputs as there are phoneme classes. The learning rule is a gradient descent using the backpropagation algorithm.

The phoneme recognition process is broken up on several stages. First, the phoneme is divided into fixed frames. Then, the frames are coded with the different encoders. The classification provides a label. Finally, by the help of the number of frames, a majority voting method allows to obtain the overall decision (for the whole phoneme).

5. PHONEME RECOGNITION RESULTS

In this paragraph, we present the results on phoneme recognition. The recognition rates presented are all on a test base (300 frames for each class) and the classifier is trained in the same conditions for the different coding methods.

Phoneme recognition rates for test database are summarized in table 2. The results of the DFE-NPC experts are presented in table 3.

Voiced/Unvoiced	98.74%
Vowels/Consonnants	83.3%
Plosives/Fricatives (Unvoiced)	98.33%
Vowels-Diphthongs/Semi-Vowels	82.3%
Nasals-Liquids/Plosives-Fricatives (Voiced)	93.03%
Vowels/Diphthongs	88.4%
Nasals/ Liquids	96.14%
Plosives/ Fricatives (Voiced)	95.28%
Front/ Central/ Back Vowels	77.3%

Table 2. Recognition rates for the Macro-classification

Front vowels: ih ey eh ae	39.32%
Central vowels: ah er	41.45%
Back vowels: uw uh ow aa	36.08%
Diphthongs: ay aw oy	56.64%
Semi-Vowels: y w	64.65%
Liquids: l r	75.46%
Nasals: m n ng	57.61%
Plosives (Voiced): b d g	72.94%
Plosives (Unvoiced): p t k	88.99%
Fricatives (Voiced): v z jh	70.65%
Fricatives (Unvoiced): f s ch	74.43%

Table 3. Recognition rates for Modular NPC for each base

The overall phoneme recognition is about 61.65% (see table 4). One have to note that the classifier, based on a MLP, is a basic classifier which can explain the performances of the Modular NPC. Indeed, the real objective of this work is the feature extraction stage and not the classification stage.

LPC	MFCC	PLP	Modular NPC
48.3%	51.25%	52.3%	61.65%

Table 4. Recognition rates for all the phonemes

6. CONCLUSIONS

We have presented an architecture for discriminative feature extraction: the Modular NPC. This architecture is based on ”gating networks” and ”expert networks”. The ”gating networks” allow to redirect the phoneme to an ”expert” in the feature extraction of this phoneme. The ”gating networks” are macro-classifiers based on predictive classification and the ”expert” are based on DFE-NPC. The DFE-NPC provide the discrimination needed for the task by the maximization of the Modelisation Error Ratio (MER). In addition, the architecture is organized by phonetics knowledge. Results of the experiments described in this article have shown that the recognition rates have been clearly improved: approximately 10% than traditional methods used in a great number of applications. The Modular NPC has also the advantage to be based on same modules since the ”gating” and the ”expert” networks are based on DFE-NPC. The principle of discrimination in the different modules is the same, and it is based on the maximization of the MER.

7. REFERENCES

[1] S. Katigiri, *Handbook of Neural Networks for Speech Processing*, Artech House eds., 2000.

[2] A. de la Torre, Antonio Peinado, Antonio J. Rubio, José C. Segura, and C. Benitéz, “Discriminative feature weighting for hmm-based continous speech recognizers,” *Speech Communication*, vol. 38, pp. 267–286, 2002.

[3] K. Torkkola, “On feature extraction by mutual information maximization,” *ICASSP*, vol. 1, pp. 821–825, 2002.

[4] H. Hermansky, “Perceptual linear predictive (plp) analysis of speech,” *The Journal of the Acoustical Society of America*, pp. 1738–1752, 1990.

[5] A. de la Torre, Antonio Peinado, Antonio J. Rubio, Victoria E. Sánchez, and Jesús E. Diaz, “An application of minimum classification error to feature space transformations for speec recognition,” *Speech Communication*, vol. 20, pp. 273–290, 1996.

[6] M. Chetouani, B. Gas, J.L. Zarader, and C. Chavy, “Neural predictive coding for speech discriminant feature extraction: The dfe-npc,” *Proc. of ESANN*, pp. 275–280, 2002.

[7] H. Teager and S. Teager, “Evidence for nonlinear sound production mechanisms in the vocal tract,” *Proc. NATO ASI on Speech production and Speech Modeling*, pp. 241–261, 1990.

[8] S.R. Waterhouse, *Divide and Conquer: Pattern Recognition using Mixtures of Experts*, Ph.D. thesis, University of Cambridge, 1997.

A.5 Article de conférence [36]

Maximisation of the Modelization Error Ratio for Neural Predictive Coding

M. Chetouani and B. Gas and J.L. Zarader

ISCA Tutorial and Research Workshop on NOn-Linear Speech Processing, p. 77–80 (2003)

.

MAXIMIZATION OF THE MODELISATION ERROR RATIO FOR NEURAL PREDICTIVE CODING

M. Chetouani, B. Gas, J.L. Zarader

Laboratoire des Instruments et Systèmes d'Ile-De-France
Université Paris VI
BP 164, Tour 22-12 2ème étage
4 Place Jussieu, 75252 Paris Cedex 05
France
mohamed.chetouani@lis.jussieu.fr gas@ccr.jussieu.fr zarader@ccr.jussieu.fr

ABSTRACT

In this paper, we introduce a model for Discrimant Feature Extraction (DFE): the Neural Predictive Coding (NPC). It is an extension of the Linear Predictive Coding (LPC). The Modelisation Error Ratio (MER), a discriminant criterion adapted for predictive models, is introduced. We propose a theoretical validation of the discriminant properties of the MER. The experimental validation consists on phoneme recognition task. The phonemes are extracted from the Darpa-Timit speech database. The performances are compared with traditional methods: LPC, MFCC, PLP.

1. INTRODUCTION

Speech recognition is not a really solved problem since the systems can not lead with difficult environment like noise, multi-speaker, etc. ... One of the problem pointed out by several authors [1], [2], [3] seems to be the feature extraction stage. Indeed, the comparison of the evolution of both classification and feature extraction techniques puts in obviousness a great difference. The most used feature extraction techniques are based on Linear Predictive Coding (LPC), Mel Frequency Cepstral Coding (MFCC) and the Perceptual Predictive Linear (PLP) technique.

In a discrimination point of view, the Discriminant Feature Extraction (DFE) based on the Minimum Classification Error (MCE) is an successful approach for many applications [2]. But, there is another approach for DFE framework which consists in the the independent training of both the feature extractor and the classifier [3]. This method is more adapted for complex problems. Indeed, during the simultaneously training of the two stages, the evolution of the feature extractor parameters is small compared to the classifier parameters [3]. The feature extractor has to be trained with a criterion which measures the discrimination power of selected features. For example, the criterion can be the Maximization of the Mutual Information (MMI) between the features and the class labels [4].

In this paper, we focused on DFE paradigm for Neural Predictive Coding (NPC), the DFE-NPC model [5] in phoneme recognition task. The NPC model is a non linear extension of LPC model. The proposed work is a theoretical and experimental validation of discriminant properties the Modelisation Error Ratio (MER) for the DFE-NPC model.

The paper is organized as follow: we first introduce the DFE-NPC model and the MER. In a second time, a statistical validation of discriminant properties the MER is proposed. And, finally we test the model in phoneme recognition task and we make some conclusions.

2. THE DFE-NEURAL PREDICTIVE CODING

2.1. The NPC model

The Non Linear Auto-Regressive (NLAR) model computed by the NPC model is the follow:

$$\hat{y}_k = F(\mathbf{y}_k) \tag{1}$$

Where k is the index of speech samples and $\mathbf{y}_k$ is the prediction context: $\mathbf{y}_k = [y_{k-1}, y_{k-2}, ..., y_{k-\lambda}]^T$ and λ the length of the prediction window.

F is a nonlinear function composed by two functions $G_{\mathbf{w}}$ ($\mathbf{w}$ first layer weights) and $H_{\mathbf{a}}$ ($\mathbf{a}$ output layer weights):

$$F_{\mathbf{w},\mathbf{a}}(\mathbf{y}_k) = H_{\mathbf{a}} \circ G_{\mathbf{w}}(\mathbf{y}_k) \tag{2}$$

With $\hat{y}_k = H_{\mathbf{a}}(\mathbf{z}_k)$ and $\mathbf{z}_k = G_{\mathbf{w}}(\mathbf{y}_k)$.

The NPC model has the major advantage to allow a nonlinear modelisation with an arbitrary limited number of coding coefficients. The key idea of the NPC model is that the coding vectors (output layers weights $\mathbf{a}_l$) have to carry discriminant phonetic features. The first layer $\mathbf{w}$ weights are common to all the phonemes. This strategy is consistent

with the fact that the speech production model can be separated a common zone (the vocal tract) and specific zones (contributions of the glottal flow and the radiation) [6].

Let us consider two phonemes i and j belonging to two different classes C_i and C_j, the NPC models associated to the two phonemes are the following:

$$\begin{cases} F_{\mathbf{w},\mathbf{a}_i} = H_{\mathbf{a}_i} \circ G_{\mathbf{w}} \\ F_{\mathbf{w},\mathbf{a}_j} = H_{\mathbf{a}_j} \circ G_{\mathbf{w}} \end{cases} \quad (3)$$

The NPC models $F_{\mathbf{w},\mathbf{a}_i}$ and $F_{\mathbf{w},\mathbf{a}_j}$ are different whereas $G_{\mathbf{w}}$ is common to the two phonemes i and j. This function remains common features and the discrimminant features are carried by the discriminant functions $H_{\mathbf{a}_i}$ and $H_{\mathbf{a}_j}$.

After the computation of all the phonemes belonging to the M classes, one could make the same conclusion for the features extracted from each classes. The coding vectors associated to each classes carry discriminant features while the first layer weights carry common features (cf. figure 1).

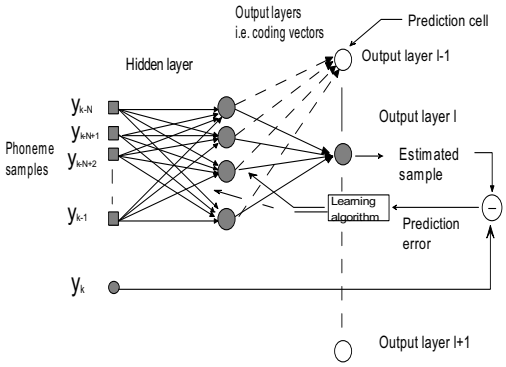


Fig. 1. Architecture of the Neural Predictive Coding model

The learning process needs to be realized in two phases: the *parameters adjustment phase* and the *coding phase*. During the first phase, all the network weights are estimated from a learning set composed of phonemes belonging to the M classes. Next, the output layers weights are non longer used while the hidden layer weights become the encoder *parameters*. Then, during the *coding phase*, the network works as a two layers perceptron composed of the hidden layer previously computed and one output cell. The *coding phase* consists in the estimation of the output weights which are the NPC coding coefficients.

2.2. The Modelisation Error Ratio

The principal problem with predictive approach is the lack of discrimination: predictive models are trained independently of each other. As a result, there is no explicit discrimination between the models. In order to solve this problem we developed a measure of discrimination between NPC models: the Modelisation Error Ratio (MER)[5].

Let us define L_j^i as the prediction error computed on the phoneme i using the NPC-2 model $H_{\mathbf{a}_j}$ associated to the phoneme j:

$$L_j^i = \sum_k (y_{i,k} - H_{\mathbf{a}_j} \circ G_{\mathbf{w}}(\mathbf{y}_{i,k}))^2 \quad (4)$$

The i-MER is the inverse ratio of the prediction error of the phoneme i predicted by the correct NPC-2 model to the prediction errors of the phoneme i predicted by the others NPC-2 models:

$$\Gamma_i = \frac{\sum_{j=1, j\neq i}^{M} L_j^i}{L_i^i} \quad (5)$$

Where M is the number of classes.

2.3. Statistical validation of the MER discriminant properties

In this section, we show the discriminant properties of the MER by a comparaison with the MMI criterion.

First, let us suppose that the model of the vocal tract can be approximated by the statistical process:

$$y_{i,k} = F_{\mathbf{w},\mathbf{a}_i}(\mathbf{y}_{i,k}) + \epsilon_{i,k} \quad (6)$$

With $\mathbf{y}_{i,k} = [y_{i,k-1}, y_{i,k-2}, ..., y_{i,k-\lambda}]^T$ is the prediction context of the phoneme i and $\epsilon_{i,k}$ are signal noises independently and identically distributed (i.i.d.). The $\epsilon_{i,k}$ obey to a normal distributions $\mathcal{N}(0, \Sigma)$ with zero means and $\Sigma = \sigma \mathrm{I}$ covariances. According to this model, the conditional likelihood of observation $y_{i,k}$ is given by:

$$\begin{aligned} p(y_{i,k}|\mathbf{y}_{i,k}, \mathbf{w}, \mathbf{a}_i) &= p(\epsilon_{i,k}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\epsilon_{i,k}^2}{2\sigma^2}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(F_{\mathbf{w},\mathbf{a}_i}(\mathbf{y}_{i,k}) - y_{i,k}\right)^2}{2\sigma^2}} \end{aligned} \quad (7)$$

And the joint likelihood of the frame speech signal $\mathcal{Y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,K}]$ of the phoneme i composed by K samples (in the hypothesis of equiprobable distribution) is the follow:

$$p(\mathcal{Y}_i|\mathbf{Y}_i, \mathbf{w}, \mathbf{a}_i) = \frac{1}{(\sqrt{2\pi\sigma^2})^K} e^{-\frac{\sum_{k=1}^{K}\left(F_{\mathbf{w},\mathbf{a}_i}(\mathbf{y}_{i,k}) - y_{i,k}\right)^2}{2\sigma^2}} \quad (8)$$

$\mathbf{Y}_i = \{\mathbf{y}_{i,1} \ldots \mathbf{y}_{i,K}\}$ is the prediction window sequence of the phoneme i.

Equation 8 shows that minimization of the prediction error L_i^i of the phoneme i by the NPC-2 model $F_{\mathbf{w},\mathbf{a}_i}$ is

equivalent to maximize the joint likelihood (under the previous assumptions):

$$\min_{\mathbf{w},\mathbf{a}_i} L_i^i = \min_{\mathbf{w},\mathbf{a}_i} \sum_{k}^{K} \left(y_{i,k} - F_{\mathbf{w},\mathbf{a}_i}(\mathbf{y}_{i,k})\right)^2 \iff \max_{\mathbf{w},\mathbf{a}_i} p(\mathcal{Y}_i|\mathbf{Y}_i, \mathbf{w}, \mathbf{a}_i) \quad (9)$$

The Minimization of the inverse MER consists in the minimization of the ratio between the prediction error of the correct model (L_i^i) and prediction errors of other models ($\sum_{j=1,j\neq i}^{M} L_j^i$). According to this definition and the previous result (9), the minimization of the inverse MER (5) can be seen as the follow statistical process:

$$\min_{\mathbf{w},\mathbf{a}_i,\mathbf{a}_j} \frac{1}{\Gamma_i} = \min_{\mathbf{w},\mathbf{a}_i,\mathbf{a}_j} \frac{L_i^i}{\sum_{j=1,j\neq i}^{M} L_j^i} \iff \max_{\mathbf{w},\mathbf{a}_i,\mathbf{a}_j} \frac{p(\mathcal{Y}_i|\mathbf{Y}_i, \mathbf{w}, \mathbf{a}_i)}{\sum_{j=1,j\neq i}^{M} p(\mathcal{Y}_i|\mathbf{Y}_i, \mathbf{w}, \mathbf{a}_j)} \quad (10)$$

The statistical interpretation of the Maximization of the MER (10) can be related to discriminant criterion like the Maximization of the Mutual Information (MMI). The MMI criterion has been successfully applied to predictive models [7]. The Maximization of the Mutual Information (MMI) consists in the maximisation of the correct model probability to the the probability of all models:

$$\max_{\rho_i} \mathrm{I} = \max_{\rho_i} \frac{p(\mathcal{Y}_i|F_i)}{\sum_{j=1}^{M} p(\mathcal{Y}_i|F_j)p(F_j)} \quad (11)$$

Where $\mathcal{Y}_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,k}]$ is the speech signal of the phoneme i and F_i is the associated predictor among M predictors (M classes). ρ_i is the set of parameters of the phoneme's model i.

The MMI criterion is equivalent to:

$$\begin{aligned} &\equiv \min_{\rho_i} \frac{\sum_{j=1}^{M} p(\mathcal{Y}_i|F_j)p(F_j)}{p(\mathcal{Y}_i|F_i)} \\ &\equiv \min_{\rho_i} \left\{ \frac{\sum_{j=1,j\neq i}^{M} p(\mathcal{Y}_i|F_j)p(F_j)}{p(\mathcal{Y}_i|F_i)} + p(F_i) \right\} \end{aligned} \quad (12)$$

Since $p(F_i)$ is independent of the model parameters ρ_i, the MMI consists in:

$$\min_{\rho_i} \left\{ \frac{\sum_{j=1,j\neq i}^{M} p(\mathcal{Y}_i|F_j)p(F_j)}{p(\mathcal{Y}_i|F_i)} \right\} \quad (13)$$

As we can see in the equations (13) and (10), the two criteria are equivalent. In (13) the numerator is a mean likelihood which is associated to an anti-model measure, whereas the denominator is a modelisation term. The MER is also defined in the same conditions (10), the numerator is the set of prediction errors computed by anti-models, and the denominator is the prediction error computed by the correct model.

2.4. Discriminant Feature Extraction

Following the definition of the i-MER (5), the MER is an extension to all the M classes and is defined as:

$$\Gamma = \frac{Q^d}{(M-1)Q^m} \quad (14)$$

With $Q^d = \sum_{i=1}^{M} \sum_{j=1,j\neq i}^{M} L_j^i$ and $Q^m = \sum_{i=1}^{M} L_i^i$.

The DFE-NPC [5] optimization is based on the minimization of the inverse MER:

$$Q_{DFE-NPC} = \frac{1}{\Gamma} \quad (15)$$

The MER is a measure of discrimination between NPC models. Indeed, the maximization of the MER can be seen as a maximization of the between-class covariance. We compare a DFE-NPC model with the NPC-2 model (a model without explicit discrimination). The figure 2 represents the evolution of the MERs during the parameters adjustment phase, and in a same time we estimate the between-class covariances (cf. figure 3).

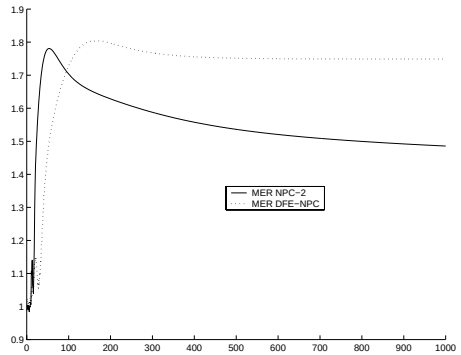


Fig. 2. Evolution of the MER during the parameters adjustment phase

3. PHONEME RECOGNITION EXPERIMENTS

The aim of this work is to validate experimentally the discriminant properties of the DFE-NPC based on the MER. We have proposed in [8] a Modular NPC architecture. It allows to combine DFE-NPC models by macro-classification stages for phoneme recognition task. So, the experience

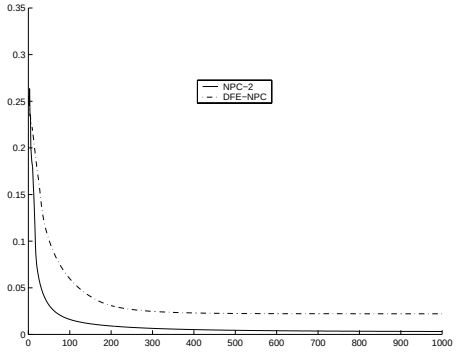


Fig. 3. Evolution of the between-class covariance during the parameters adjustment phase

consists in a comparison of this architecture with traditional coding methods: LPC, MFCC and PLP.

The phonemes belong to the Darpa-Timit database. They are extracted from all the speakers in order to produce a multi-speaker environment. Depending on their duration, each phoneme is split into a number of frames: the length of analysis windows is set to 256 samples with an overlapping factor of 128 samples. For each class the number of frames is set to 300 and the coding vector dimension is set to 12. The classifier used to estimate the performance of the DFE-NPC encoder is a simple multi-layer perceptron (MLP) with 12 inputs, 10 neurons and as many outputs as there are phoneme classes.

The phoneme recognition process is broken up on several stages. First, the phoneme is divided into fixed frames. Then, the frames are coded by the different encoders and the classification provides a label. Finally, by the help of the number of frames, a majority voting method allows to obtain the overall decision (for the whole phoneme).

Front vowels: ih ey eh ae	39.32%
Central vowels: ah er	41.45%
Back vowels: uw uh ow aa	36.08%
Diphthongs: ay aw oy	56.64%
Semi-Vowels: y w	64.65%
Liquids: l r	75.46%
Nasals: m n ng	57.61%
Plosives (Voiced): b d g	72.94%
Plosives (Unvoiced): p t k	88.99%
Fricatives (Voiced): v z jh	70.65%
Fricatives (Unvoiced): f s ch	74.43%

Table 1. Recognition rates for Modular NPC for each base

The recognition rates for the different classes are presented on table 1, the recognition rates of the macro classification stages are take into account. The overall is about 61.65% (see table 2). The results put in obviousness the improvement of the DFE-NPC. Indeed, the discrimination introduced allows to increase the recognition rates.

LPC	MFCC	PLP	Modular NPC
48.3%	51.25%	52.3%	61.65%

Table 2. Recognition rates for all the phonemes

4. CONCLUSIONS

Discriminant Feature Extraction is an essential requirement for speech recognition. The DFE-NPC model has main characteristics for feature extraction. First, it is a non linear model but with a limited coding coefficients dimension. We have proposed a theoretical validation of the DFE-NPC discriminant properties (based on the MER) by a link with the MMI. The experimental validation with the help of the Modular NPC shows that the recognition rates are clearly improved (approximately 10%).

5. REFERENCES

[1] H. Boulard, H. Hermansky, and N. Morgan, "Towards increasing speech recognition error rates," *Speech Communication*, vol. 18, pp. 205–231, 1996.

[2] S. Katagiri, *Handbook of Neural Networks for Speech Processing*, Artech House eds., 2000.

[3] A. de la Torre, Antonio Peinado, Antonio J. Rubio, José C. Segura, and C. Benitéz, "Discriminative feature weighting for hmm-based continous speech recognizers," *Speech Communication*, vol. 38, pp. 267–286, 2002.

[4] K. Torkkola, "On feature extraction by mutual information maximization," *ICASSP*, vol. 1, pp. 821–824, 2002.

[5] M. Chetouani, B. Gas, J.L. Zarader, and C. Chavy, "Neural predictive coding for speech discriminant feature extraction: The dfe-npc," *Proc. of ESANN*, pp. 275–280, 2002.

[6] L. Rabiner and B.J. Juang, *Fundamentals of speech recognition*, Prentice-Hall, 1993.

[7] A. Mellouk, P. Gallinari, and F. Rauscher, "Prediction and discrimination in neural networks for continous speech recognition," *Eurospeech*, pp. 1603–1607, 1993.

[8] M. Chetouani, B. Gas, and J.L. Zarader, "Modular neural predictive coding for discriminative feature extraction," *To appear in Proc. of ICASSP*, 2003.

A.6 Article de conférence [45]

New Sub-band Processing Framework using Non-linear Predictive Models for Speech Feature Extraction

M. Chetouani and A. Hussain and B. Gas and J.L. Zarader

3th International Conference on NOn-Linear Speech Processing (NOLISP'05), p. 269–274 (2005)

.

Sub-band Processing and Predictive Models for Speech Feature Extraction

Mohamed Chetouani[1], Amir Hussain[2], Bruno Gas[1], Jean-Luc Zarader[1]

[1] Laboratoire des Instruments et Systèmes d'Ile-De-France
Université Paris VI, Paris, FRANCE
[2] Dept. of Computing Science and Mathematics
University of Sitrling, Scotland, U.K.

Abstract. Speech feature extraction methods are commonly based on time and frequency processing approaches. In this paper, we propose a new framework based sub-band processing an non-linear prediction. The key idea is to pre-process the speech signal by a filter bank. From the resulting signals, non-linear predictors are computed. The feature extraction consists in the association of different Neural Predictive Coding (NPC) models. We apply this new framework to phoneme classification. The experiments carried out with the NTIMIT database show an improvement of the classification rates in comparison to the full-band approach. The new method gives also better performances than the traditional ones (LPC, MFCC and PLP).

1 Introduction

Speech feature extraction stages are commonly based on time and frequency processing methods. Indeed, the most used method is the Mel Frequency Cepstral Coding (MFCC) but the Linear Predictive Coding (LPC) has been also used. The success of the MFCC is partly due to the sub-band processing used since it is based on Mel-scale filter bank.

The filter banks are usually designed according to so some model of the auditory system [1]. Several models have been proposed for speech feature extraction [14], [7], [9], [5]. Among the advantages of sub-band processing approaches, the robustness is a significant point. For instance in speech coding [6], [16] and in speech enhancement [11], the filter banks clearly improve the performances in noisy environnements.

In a classification point of view, the division of the whole frequency domain into sub-bands and the application of different strategies is way for error rates reduction. In speaker recognition [2], it is known that some sub-bands remain more speaker dependent features. In speech recognition, similar ideas have been followed. Works on human being recognition [1] show that the linguistic message is decoded in different sub-bands and the final decision consists in merging the information from these sub-bands [10].

In this paper, we are interested in the combination of sub-band processing and non-linear predictive methods. The key idea is to divide the whole frequency

domain into sub-bands and then non-linear predictors are used for feature extraction. The combination of filter banks and predictors is intensively used in speech coding. The purpose of this paper is to compare full-band non-linear predictors for feature extraction and sub-band based approaches.

The paper is organized as follows. Section 2 is dedicated to the description of the new feature extractor. Then, section 3 presents the non-linear predictor used: The Neural Predictive Coding (NPC). After, we present the experimental conditions and the performances of the system for different phoneme groups. Finally, we give some conclusions and prospects.

2 Non-linear Predictive Sub-band Feature Extractor

Traditionally, linear and non-linear predictors are computed directly from speech signal samples. In speech coding and in speech enhancement, it has been shown that such approach is not adapted for noisy environments. Several methods have been proposed [6] and among them the combination of filter banks and predictors. In feature extraction, similar ideas have been investigated. The Perceptual Linear Prediction (PLP) [8] is an example. It consists in the modification of the spectrum by auditory knowledge and then using the all-pole modelling like in the LPC.

The principle of the proposed feature extraction method is described in figure 1. The first stage aims to pre-process the speech signal by a filter bank. The following stage consists to the extraction of the features. Instead of using energy from these sub-bands, the Neural Predictive Coding (NPC) model is used. The features are extracted from signals resulting from the pre-processing.

This approach is different from traditional feature extraction methods. We propose to pre-process the signal by a filter bank. The first stage can be designed to reflect auditory knowledge. Feature extractors based on auditory models usually compute the energy of the sub-bands. In our model, the second stage is based on a non-linear feature extractor: The NPC model which is a predictive model. The next section is dedicated to the description of this model.

3 Neural Predictive Coding

The Neural Predictive Coding (NPC) model [4] [3] is a non-linear extension of the well-known LPC encoder. Like in the LPC framework with the AR model, the vector code is estimated by prediction error minimization. The main difference lies in the fact that the model is non-linear and it is a connectionnist model:

$$\hat{y}_k = F(\mathbf{y_k}) = \sum_{\mathbf{j}} \mathbf{a_j} \sigma(\mathbf{w^T y_k}) \tag{1}$$

Where F is the prediction function realized by the neural model. $\hat{y}_k$ is the predicted sample. $\mathbf{y_k}$ the prediction context: $\mathbf{y_k} = [\mathbf{y_{k-1}}, \mathbf{y_{k-2}}, ..., \mathbf{y_{k-\lambda}}]^\mathbf{T}$ and λ

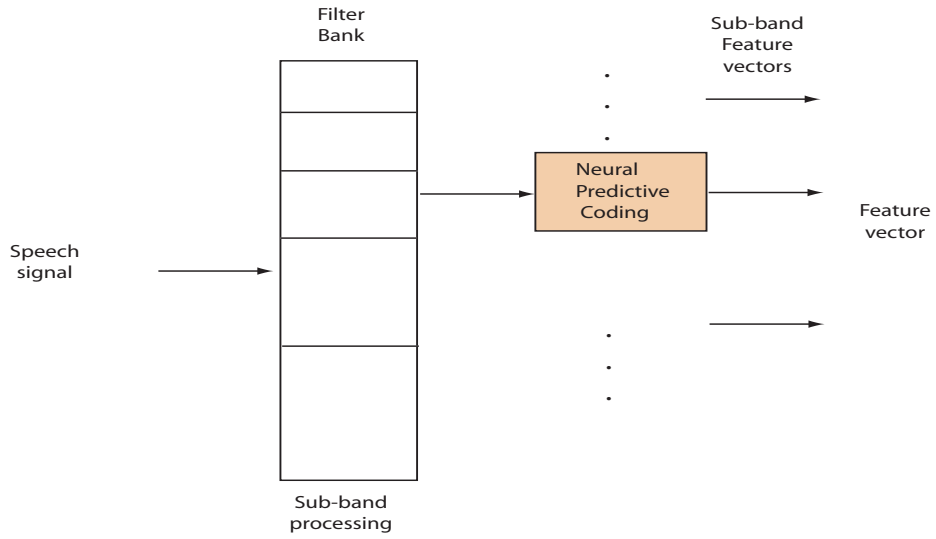


Fig. 1. Non-Linear Predictive Sub-band Based Feature Extraction.

the length of the prediction window. $\mathbf{w}$ and $\mathbf{a}$ represent the first and the output layer weights. σ is a non-linear activation function, the sigmoid function in our case.

The key idea is to use the NPC model as a non-linear auto-regressive model. As in the LPC framework for the predictor coefficients, the NPC weights are the vector code. It is well-known that the weights can be consider as a representation of the input vector. A drawback of this method is that non-linear models have no clear physical meanings [13]. The solution weights can be very different for a same minimum of the prediction error. In our approach, we impose constraints on weights.

3.1 Description

The NPC model is a Multi-Layer Perceptron (MLP) with one hidden layer. Only the output layer weights are used as coding vector instead of all the neural weights. For that we consider that the function F realized by the model, under convergence assumptions, can be decomposed into two functions: $G_{\mathbf{w}}$ ($\mathbf{w}$ first layer weights) and $H_{\mathbf{a}}$ ($\mathbf{a}$ output layer weights):

$$F_{\mathbf{w},\mathbf{a}}(\mathbf{y_k}) = \mathbf{H_a} \circ \mathbf{G_w}(\mathbf{y_k}) \tag{2}$$

With $\hat{y}_k = H_{\mathbf{a}}(\mathbf{z_k})$ and $\mathbf{z_k} = \mathbf{G_w}(\mathbf{y_k})$.

As one can note the NPC structure allows a different prediction window's length independently to the coding vector size contrary to the LPC structure.

For the layers specialization, the learning phase is realized in two times. First, the *parameterization phase* consists in the learning of all the weights by the prediction error minimization criterion:

$$Q = \sum_{k=1}^{K} (y_k - \hat{y}_k)^2 = \sum_{k=1}^{K} \left(y_k - F(\mathbf{y_k})\right)^2 \tag{3}$$

With y the speech signal, $\hat{y}$ the predicted speech signal, k the samples index and K the number of samples.

In this phase, only the first layer weights $\mathbf{w}$ which are the NPC encoder parameters are kept. Since the NPC encoder is set up by the parameters defined in the previous phase, the second phase, called the *coding phase*, consists in the computation of the output layer weights $\mathbf{a}$ (vector code). This is done also by prediction error minimization but only the output layer weights are updated. One can note that the output function is linear (cf. equation 1), so it can be done by the Levinson algorithm as for the LPC model. Here, for consistency with the *parameterization phase*, it is done by the backpropagation algorithm.

The result from the layers specialization, the first layer weights $\mathbf{w}$ are common to all the speech signal frames while the second layer weights $\mathbf{a}$ are specific to each frame. For each frame, a feature vector $\mathbf{a}$ is computed by prediction error minimization.

4 Evaluation and discussion

4.1 Experimental conditions

The NTIMIT database [12] is used in this experiment and more especially the two first regions (DR1, DR2). By using this database, we carry out speech recognition in telephone quality. One can note that the telephone bandwidth is approximately limited to 300-3400Hz. We focus on the processing of front vowels (/ih/, /ey/, /eh/, /ae/), voiced plosives (/b/, /d/, /g/) and unvoiced plosives (/p/, /t/, /k/). This choice can be justified by the fact that the classification of these phonemes is known to be difficult and they are also often used. For training and test databases, we use the division proposed by the database.

The classification is carried out by GMM (16 centers, diagonal assumption) and it is a frame by frame classification (32ms with 16ms of overlapping). The dimension of the features is set to 12.

The proposed feature extractor is based on sub-band processing but the optimal number of sub-bands is still an open issue. In Mel-scale filter banks, this number is about 20. We evaluate the performances of our model with 2 sub-bands (300-1140Hz, 1046-3400Hz) and 4 sub-bands (300-765Hz, 700-1640Hz, 1515-2700Hz, 2100-3400Hz) [15]. For the first case, we extract 6 NPC coefficients by sub-band and for the second one, we extract 3 NPC coefficients in order to keep a dimension 12 for the feature vector.

We make comparisons with traditional methods: LPC, MFCC and PLP. And in order to evaluate the performances of the sub-band approach, we also test a full-band NPC model.

4.2 Results

The classification rates for the NPC model in the different sub-band are grouped in 1. The performances of the 2 sub-bands model are better than the full-band model. This result shows that the sub-band approach, by dividing the whole frequency domain, is effective for phoneme classification. Indeed, the phoneme dependent features are distributed among different sub-bands [1], [10]. However, for the 4 sub-bands model, the performances decrease. This is partly due to the lack of data for each sub-band.

Table 1. Classification rates for different sub-bands (vowels).

Phoneme	Full-band	2 sub-bands	4 sub-bands
/ih/, /ey/, /eh, /ae/	49.03	52.4	50.89

The classification rates for the different methods are presented table 2. For voiced phonemes, the performances are improved by non-linear methods even in the case of the plosives (NPC: 62.24%, Sub-band NPC:63.87 %).

Table 2. Classification rates: significative improvements by non-linear and sub-band methods.

Phoneme	LPC	MFCC	PLP	NPC	Sub-Band NPC
/ih/, /ey/, /eh, /ae/	35.22	48.12	45.12	49.03	**52.4**
/b/, /d/, /g/	54.13	59.23	57.21	62.24	**63.87**
/p/, /t/, /k/	44.10	51.45	46.98	49.36	**52.56**

In the case of unvoiced plosives, the performances are appreciatively equivalent to the state-of-art method (MFCC:51.45%, Sub-band NPC: 52.56%). Despite of this result, the performances are improved in comparison to the full-band approach (Full-band: 49.36%, Sub-band: 52.56%).

5 Conclusion

In this paper, we propose a new framework for speech feature extraction. It consists in the combination of filter banks and non-linear predictors. The filter banks act to pre-process the signal in different sub-bands. These sub-bands remain phoneme dependent features. They are extracted by the help of a non-linear feature extractor: The Neural Predictive Coding (NPC) model. The obtained features are decorrelated and are efficient for classification. The results are better than the full-band approach and moreover better than traditional methods: MFCC, LPC and PLP.

In our approach, the number of sub-bands is still an open issue. Our future works are dedicated to the introduction of explicit discriminant criteria between the NPC models. We are also interested in different applications like speaker recognition and speech/music discrimination.

References

1. J.B. Allen. How do humans process and recognize speech? *IEEE Trans. on Speech and Audio Processing*, 2(4):567–577, 1994.
2. L. Besacier and J.F. Bonastre. *Lecture Notes in Computer Science, Audio and Video-based Biometric Person Authentification*, chapter Subband approach for automatic speaker recognition: Optimal division of the frequency, pages 195–202. Springer, 1997.
3. M. Chetouani. *Codage neuro-prédictif pour l'extraction de caractéristiques de signaux de signaux de parole*. PhD thesis, Université Paris VI, 2004.
4. B. Gas, J.L. Zarader, C. Chavy, and M. Chetouani. Discriminant neural predictive coding applied to phoneme recognition. *Neurocomputing*, 56:141–166, 2004.
5. O. Ghitza. Auditory models and human performance in tasks related to speech coding and speech recognition. *IEEE Trans. on Speech and Audio Processing*, 2(1):115–132, 1994.
6. B. Gold and N. Nelson. *Speech and Audio Signal Processing : Processing and Perception of Speech and Music*. John Wiley and Sons, INC, 2000.
7. S. Greenberg. Representation of speech in the auditory periphery. *Journal of Phonetics, Special Issue*, 16(1), January 1998.
8. H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *The Journal of the Acoustical Society of America*, pages 1738–1752, 1990.
9. H. Hermansky. Auditory modeling in automatic recognition of speech. *Proc. Keele Workshop*, 1996.
10. H. Hermansky, S. Tibrewala, and M. Pavel. Towards asr on partially corrupted speech. *Proc. ICSLP*, 1996.
11. A. Hussain and D.R. Campbell. Binaural sub-band adaptive speech enhancement using artificial neural networks. *Speech Communication*, 25:177–186, 1998.
12. C. Jankowski, A. Kalyanswamy, S. Basson, and J. Spitz. Ntimit: A phonetically balanced, continous speech, telephone bandwidth speech database. *ICASSP*, 1:109–112, 1990.
13. W. B. Kleijn. Signal processing representations of speech. *IEICE Trans. Inf. and Syst.*, E86-D(3):359–376, March 2003.
14. K.K. Paliwal. Spectral subband centroid features for speech recognition. *ICASSP*, 2:617–620, 1998.
15. S. Tibrewala and H. Hermansky. Sub-band based recognition of noisy speech. *Proc. ICASSP*, 2:1255–1258, 1997.
16. Rongshan Yu and C. C. Ko. A warped linear-prediction-based subband audio coding algorithm. *IEEE Trans. on Speech and Audio Processing*, 10(1):1–8, 2002.

A.7 Chapitre de livre [31]

Non-linear Speech Feature Extraction for Phoneme Classification and Speaker Recognition

M. Chetouani and M. Faundez and B. Gas and J.L. Zarader

Nonlinear speech processing : Algorithms and Analysis, p. 344–350 (2005)

Editeurs : "G. Chollet and A. Esposito and M. Faundez and M. Marinaro, Springer-Verlag",

.

Non-linear Speech Feature Extraction for Phoneme Classification and Speaker Recognition

Mohamed Chetouani[1], Marcos Faundez-Zanuy[2], Bruno Gas[1], and Jean-Luc Zarader[1]

[1] Laboratoire des Instruments et Systèmes d'Ile-De-France
Université Paris VI, Paris, France
[2] Escola Universitària Politècnica de Mataró, Barcelona, Spain

Abstract. In this paper we propose a new feature extraction algorithm based on non-linear prediction: the Neural Predictive Coding (NPC) model which is an extension of the classical LPC one. We apply this model to two significant tasks: phoneme classification and speaker identification. For the first one, the NPC model is trained with a Minimum Classification Error (MCE) criterion. The experiments carried out with the NTIMIT database show an improvement of the classification rates. For speaker identification, we propose a new feature extraction principle based on the NPC model. We also investigate different initialization methods. The new method gives better performances than the traditional ones (LPC, MFCC and PLP).

1 Introduction

The design of speech and speaker recognition systems is commonly based on three stages: signal acquisition, feature extraction and pattern classification. Feature extraction is an important stage in recognition systems. The main objective of this stage is the extraction of relevant characteristics for the next stage which is the classification stage. It is usually done in a same way for phoneme classification and for speaker recognition whereas the final purpose is different. Linear Predictive Coding (LPC) or the Mel Frequency Cepstral Coding (MFCC) are the most used methods.

However, some limits seem to be reached and it is difficult to overcome them with conventional methods. One of the reasons is the great variability of the speech signal. Variations are due to several elements. There are obviously inter-speakers and intra-speakers variabilities. The sound environment as well as the phonetic context (coarticulation) are also elements prone to introduce variability into speech signals. Several authors pointed out the need of specific features for each task [1], [2]: speech transmission, speech, speaker recognition or even language identification.

A good feature extractor must extract the features necessary to the recognition process. The problem is that these features are not really known. A solution consists in the exploitation of knowledge about the human operation. This way is limited by the efficiency of the knowledge for the objective [1]. Some of the

G. Chollet et al. (Eds.): Nonlinear Speech Modeling, LNAI 3445, pp. 344–350, 2005.

phenomena are useful like critical bands (filter banks) or non-linear modelization (Mel and Bark scales) which have been implemented in the MFCC or the PLP. Designing feature extractor by exploiting these knowledge is called data-driven methods [1].

In this paper, we focus on feature extraction for two significant tasks in speech processing: phoneme classification and speaker identification. The proposed work is an investigation on non-linear speech processing in feature extraction. For that, we examine non-linear modelization but also discriminant criteria.

The paper is organized as following. First, we describe our non-linear model called the Neural Predictive Coding (NPC). Then, we discuss on the two different tasks: phoneme classification and speaker identification. For both experiences are carried out and they give significative improvements of the recognition rates.

2 Neural Predictive Coding

The Neural Predictive Coding (NPC) model [3] is a non-linear extension of the well-known LPC encoder. Like in the LPC framework with the AR model, the vector code is estimated by prediction error minimization. The main difference lies in the fact that the model is non-linear and it is a connectionnist model:

$$\hat{y}_k = F(\mathbf{y}_k) = \sum_j a_j \sigma(\mathbf{w}^T \mathbf{y}_k) \tag{1}$$

Where F is the prediction function realized by the neural model. $\hat{y}_k$ is the predicted sample. $\mathbf{y}_k$ the prediction context: $\mathbf{y}_k = [y_{k-1}, y_{k-2}, ..., y_{k-\lambda}]^T$ and λ the length of the prediction window. $\mathbf{w}$ and $\mathbf{a}$ represent the first and the output layer weights. σ is a non-linear activation function, the sigmoid function in our case.

The key idea is to use the NPC model as a non-linear auto-regressive model. As in the LPC framework for the predictor coefficients, the NPC weights are the vector code. It is well-known that the weights can be consider as a representation of the input vector. A drawback of this method is that non-linear models have no clear physical meanings [4]. The solution weights can be very different for a same minimum of the prediction error. In our approach, we impose constraints on weights.

2.1 Description

The NPC model is a Multi-Layer Perceptron (MLP) with one hidden layer. Only the output layer weights are used as coding vector instead of all the neural weights. For that we consider that the function F realized by the model, under convergence assumptions, can be decomposed into two functions: $G_{\mathbf{w}}$ ($\mathbf{w}$ first layer weights) and $H_{\mathbf{a}}$ ($\mathbf{a}$ output layer weights):

$$F_{\mathbf{w},\mathbf{a}}(\mathbf{y}_k) = H_{\mathbf{a}} \circ G_{\mathbf{w}}(\mathbf{y}_k) \tag{2}$$

With $\hat{y}_k = H_{\mathbf{a}}(\mathbf{z}_k)$ and $\mathbf{z}_k = G_{\mathbf{w}}(\mathbf{y}_k)$.

As one can note the NPC structure allows a different prediction window's length independently to the coding vector size contrary to the LPC structure.

For the layers specialization, the learning phase is realized in two times. First, the *parameterization phase* consists in the learning of all the weights by the prediction error minimization criterion:

$$Q = \sum_{k=1}^{K} (y_k - \hat{y}_k)^2 = \sum_{k=1}^{K} (y_k - F(\mathbf{y}_k))^2 \tag{3}$$

With y the speech signal, $\hat{y}$ the predicted speech signal, k the samples index and K the number of samples.

In this phase, only the first layer weights $\mathbf{w}$ which are the NPC encoder parameters are kept. Since the NPC encoder is set up by the parameters defined in the previous phase, the second phase, called the *coding phase*, consists in the computation of the output layer weights $\mathbf{a}$ (vector code). This is done also by prediction error minimization but only the output layer weights are updated. One can note that the output function is linear (cf. equation 1), so it can be done by the Levinson algorithm as for the LPC model. Here, for consistency with the *parameterization phase*, it is done by the backpropagation algorithm.

The result from the layers specialization, the first layer weights $\mathbf{w}$ are common to all the speech signal frames while the second layer weights $\mathbf{a}$ are specific to each frame. For each frame, a feature vector $\mathbf{a}$ is computed by prediction error minimization.

3 Feature Extraction in Phoneme Classification

The purpose of this task is to extract phonetic information from the speech signal. In this section, we investigate the importance of non-linear modelization. First, we compare our non-linear feature extractor (NPC) to traditional methods in order to validate the importance of non-linear modelization. Secondly, we propose a discriminant model with non-linear discrimination.

The discriminant model is based on the simultaneous training of a classifier and the NPC model [5] (see figure 1). For classification, we use the LVQ model (Learning Vector Quantization). The LVQ and the NPC are optimized through the Minimum Classification Error (MCE) criterion and the new model is called the LVQ-NPC model.

3.1 Evaluation and Discussion

The NTIMIT database [6] is used in this experiment and more especially the two first regions (DR1, DR2). By using this database, we carry out speech recognition in telephone quality. We focus on the processing of front vowels (/ih/, /ey/, /eh/, /ae/), voiced plosives (/b/, /d/, /g/) and unvoiced plosives (/p/, /t/, /k/). This choice can be justified by the fact that the classification of these phonemes is known to be difficult and they are also often used. For training and test

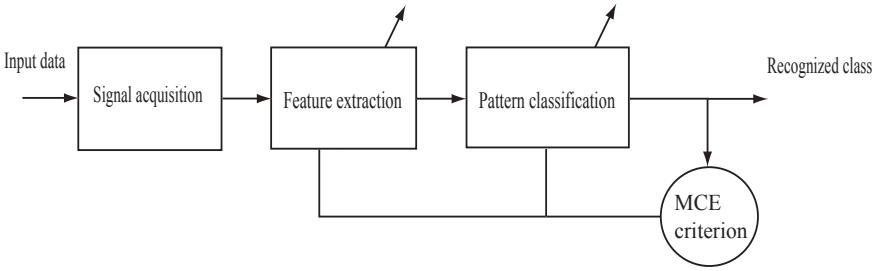


Fig. 1. Discriminative Feature Extraction based on the Minimum Classification Error

databases, we use the division proposed by the database. A part of the training set is used as a cross-validation base, in order to stop the classifiers training but also the LVQ-NPC *parameterization phase*. The classification is carried out by GMM (16 centers, diagonal assumption) and it is a frame by frame classification (32ms with 16ms of overlapping). The dimension of the features is set to 12.

The classification rates for the different methods are presented table 1. For voiced phonemes, the performances are improved by non-linear methods (NPC: 49.03%, LVQ-NPC: 54.81%) even in the case of the plosives.

Table 1. Classification rates: significative improvements by non-linear and discriminant methods

Phoneme	LPC	MFCC	PLP	NPC	LVQ-NPC
/ih/, /ey/, /eh, /ae/	35.22	48.12	45.12	49.03	**54.81**
/b/, /d/, /g/	54.13	59.23	57.21	62.24	**66.33**
/p/, /t/, /k/	44.10	51.45	46.98	49.36	**53.22**

For unvoiced phonemes, the performances are degraded for predictive methods (LPC: 44.10%, NPC: 49.36%) compared to the MFCC (51.45%). However, discrimination makes it possible to overcome this problem and to obtain better results LVQ-NPC: 53.22% (cf. table 1). This result is important because it shows that non-linear approach is obviously needed for modelization but also for discrimination.

4 Feature Extraction in Speaker Identification

This section is devoted to another task: feature extraction for speaker identification. This task consists in the extraction of speaker dependent features and they have to be independent to the phonetic context. As for phoneme classification, these features are not really known even if some efforts have been done [2].

Currently in speaker recognition, feature extraction is carried out in a same way for all the speakers. Most of the efforts have been made in the second

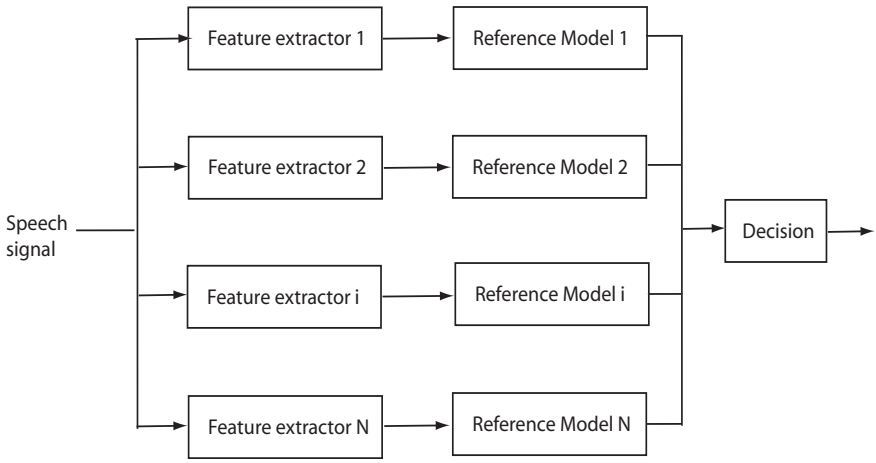


Fig. 2. Speaker dependent feature extraction principle

stage which is the model of the speaker. In our system, the speaker is obviously modelized by the second stage (reference model) but the first stage (feature extractor) is also specialized in the processing of this speaker. The speaker-dependent features are extracted by the NPC model (see figure 2). Each NPC model is specialized in the processing of only one speaker.

4.1 Linear Initialization of Non-linear Models

We also investigate initialization of non-linear models [7]. This initialization is referred as a linear initialization but it differs from conventional neural networks initializations. The proposed initialization exploits speech knowledge. We used the LPC coding method for the initialization of the non-linear coding model. By this way, one can see the NPC as a non-linear feature extractor initialized by a linear model. By neglecting biaises and removing the non-linear activation functions (for linear approximation), one obtains the following equivalence:

$$\Theta = \mathbf{w} \bullet \mathbf{a} \tag{4}$$

Where Θ are the LPC parameters of the speech signal, $\mathbf{w}$ first layer weights (determined in the previous phase: the *parameterization phase*) and $\mathbf{a}$ second layer weights.

If the NPC vector code dimension is ρ and the prediction window is λ then the LPC vector code dimension has to be set to λ. The second layer weights $\mathbf{a}$ are given by:

$$\mathbf{a} = \mathbf{w}^{+} \bullet \Theta \tag{5}$$

Where $\mathbf{w}^{+}$ is the pseudo-inverse of $\mathbf{w}$.

The initialized weights are determined by a simple LPC analysis with an order λ. Once the initialization is accomplished, the coding process (prediction

error minimization) proceeds as well as the original NPC *coding phase* by the backpropagation algorithm.

Initialization of non-linear models by linear ones has been already investigated [8] but with matrices decomposition methods (SVD, QR,...). The main limitation relies in the multiplicity of the solution. In this paper, we propose a method which guarantees a single solution.

4.2 Evaluation and Discussion

Our experiments have been computed over 49 speakers from the Gaudi database [9] that has been obtained with a microphone connected to a PC. One minute of read text is used for training, and 5 sentences for testing (each sentence is about 2-3 seconds long). The feature vector dimension is set to 16. A covariance matrix (CM) is computed for each speaker, and an Arithmetic-Harmonic Sphericity (AHS) measure is used in order to compare matrices [10].

Table 2 presents the experimental results. One can see that for the traditional methods, the best performances are obtained for the MFCC (97.55%) and the LPCC (96.73%) coding methods. These methods try to model the phonetic context but also the speaker characteristics. The LPC model has a better score (90.61%) than the PLP (86.12%). This is due to the fact that the PLP method suppresses speaker dependent characteristics. It is why the PLP allows comparable performances with the MFCC in speech recognition task (cf. Section 3).

Table 2. Speaker identification rates

Speech coding method	Identification rate (%)
LPC	90.61
LPCC	96.73
MFCC	97.55
PLP	86.12
NPC (random initialization)	61.63
NPC (linear initialization)	100

Depending on the initialization, the NPC behavior is very different. We obtain 61.63% for the random initialization while for the linear initialization we obtain 100%. This last initialization gives the best results. One of the reasons is that it allows the unicity of the initialization. Another reason is that the LPC is appropriated to the modelization of voiced part which contains more speaker-dependent characteristics.

5 Conclusions

Usually, speech and speaker recognition systems are improved by statistical methods. In this paper, we study the feature extraction stage. This stage can be

improved by several methods and among them non-linear processing. Feature extraction in phoneme classification has shown that non-linear processing is needed for modelization but also for discrimination. We proposed another method for feature extraction in speaker identification. In this task, we show that non-linear models can be initialized by linear models and it can be useful.

The proposed work is an investigation on feature extractors. We show that non-linear speech processing can be effective if we use it in a suitable way.

References

1. H. Hermansky: Should Recognizers Have Ears?. Speech Communication. **25** (1998) 3–27
2. L. Mary, K.S. Rama Murty, S.R. Mahadeva Prasanna, B. Yegnanarayana. Features for Speaker and Language Identification. Proc. of ISCA Tutorial and Research Workshop on Speaker and Language Recognition (Odyssey'04). (2004), 323–328
3. B. Gas, J.L. Zarader, C. Chavy, M. Chetouani. Discriminant neural predictive coding applied to phoneme recognition. Neurocomputing. **56**, (2004), 141–166
4. W. B. Kleijn. Signal Processing Representations of Speech. IEICE Trans. Inf. and Syst. **E86-D**, 3, March, (2003), 359–376
5. M. Chetouani, B. Gas, J.L. Zarader. Learning vector quantization and neural predictive coding for nonlinear speech feature extraction. EUSIPCO (2004).
6. C. Jankowski and A. Kalyanswamy and S. Basson and J. Spitz. NTIMIT: A Phonetically Balanced, Continous Speech, Telephone Bandwidth Speech Database. ICASSP, 1, (1990), 109–112.
7. M. Chetouani, M. Faundez-Zanuy, B. Gas, J.L. Zarader. A new nonlinear speaker parameterization algorithm for speaker identification. Proc. of ISCA Tutorial and Research Workshop on Speaker and Language Recognition (Odyssey'04). (2004), 309–314.
8. T. L. Burrows. Speech Processing with Linear and Neural Networks Models. PhD Cambridge, 1996.
9. J. Ortega-Garcia and al. Ahumada: a large speech corpus in Spanish for speaker identification and verification. ICASSP, 2, (1998), 773–776.
10. F. Bimbot and L. Mathan. Text-free speaker recognition using an arithmetic-harmonic sphericity measure. EUROSPEECH, (1991), 169–172

.

A.8 Article de conférence [64]

The Predictive Self-Organizing Map : Application to Speech Features Extraction

B. Gas and M. Chetouani and J.L. Zarader and F. Feiz

Workshop on Self Organizing Maps (WSOM'05)

.

The Predictive Self-Organizing Map : application to speech features extraction

B. Gas, M. Chetouani, J.L. Zarader, F. Feiz
LISIF / Université Paris VI
Paris. France
Bruno.Gas@upmc.fr

Abstract - *Some well known theoretical results concerning the universal approximation property of MLP neural networks with one hidden layer have shown that for any function f from $[0,1]^n$ to $\Re$, only the output layer weights depend on f. We use this result to propose a network architecture called the predictive Kohonen map allowing to design a new speech features extractor. We give experimental results of this approach on a phonemes recognition task.*

Key words - speech features extraction, function approximation, signal prediction

1 Introduction

Most of the speech recognition systems require in the very first stage to model the short-term spectrum of the signal (typically windows from 10 to 20 ms). MFCC parameters (Mel Frequency Cepstrum Coding) are for a long time used because of their robustness and of the quality of their statistical distribution. Authors as Hermansky [3] however pointed out the importance to revisit the feature extraction stage. He proposed to use the more recent perceptual auditive models such as the PLP and RASTA-PLP [1],[2]. One also find parametric approximation methods of the short-term spectrum. Instead of using directly the short-term spectrum as for MFCC, one can approximate it by parametric approaches like it is done in the well-known LPC (Linear Predictive Coding). Usually these approximations are based on linear assumptions of the speech production model (i.e. vocal tract).

1.1 Non linear models

Gas and Zarader [7] proposed a new feature extraction method based on a neural network approach (MLP) : The Neural Predictive Coding (NPC). This model is a non-linear extension of the LPC. Consequently, the NPC parameters are the coefficients of the non-linear auto-regressive model estimated by prediction error minimization [4]. They can be seen as a nonlinear parametric modeling of the short-term spectrum. The main drawback of neural networks approach is the feature vector dimension which can be very high [5]. Traditionaly used approach consists in reducing the representation space by the means of a discriminant analysis (LDA) [8], possibly nonlinear (NLDA) [6]. The NPC model aims to solve this problem by using the output layer weights as a signal representation or features. The generated acoustic vector thus sees its dimension depending only on the arbitrary number of hidden

cells and not on the input size (i.e. prediction context). It is not necessary any more to change the representation space.

1.2 Discriminative models

One drawback of the NPC parameters, inherited from LPC parameters, is their lack of discrimination. In fact, they are more adapted to speech coding and synthesis applications [11]. Juang and Katigiri [9] showed that a reinforcement of the discriminant property can be obtained by adapting the features extraction to the classification task. For example, Biem and Katagiri [10] proposed to estimate the optimal spectral width of the MFCC filters bank during the classifier training stage. Similar ideas have been used to make improvements of the NPC coder. Two new versions of the coder were thus proposed (DFE-NPC and LVQ-NPC), [13]. They were tested on phonemes recognition [14] and speaker recognition [15].

1.3 Unsupervised models

Some applications (for example the segmentation of unknown speakers in radio broadcast news) do not provide classes membership information (the speakers). An alternative consists in using unsupervised algorithms. We propose in this article a new unsupervised version of the coder called NPC-K (K for Kohonen) which could be also called SOM-NPC. The output layer cells are organized according to a topological map called the *topological predictive map*. We show by experiments that a specialization of the output layer weights is obtained by self-organization, according to the membership class of the input signals.

2 NPC-K parameters

In 1957, Kolmogorov proved with its superposition theorem (13th Hilbert problem refutation) that every continuous function f from $\mathcal{E}^n$ to $\Re$ defined on the n-dimensional Euclidean unit cube $\mathcal{E}^n$ and with range on the real line $\Re$ can be represented as a sum of continuous functions:

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \phi_q \left(\sum_{p=1}^{n} \psi_{pq}(x_p) \right) \tag{1}$$

Hecht-Nielsen [16] recognized that this specific format of Kolmogorov's superpositions can be interpreted as a feedforward neural network with a hidden layer that computes the variables

$$y_q = \sum_{p=1}^{n} \psi_{pq}(x_p) \tag{2}$$

This suggestion has been criticized by Poggio and Girosi [17] for several reasons, one being that applying Kolmogorov's theorem would require the learning of nonparametric activation functions. However, other similar results have been obtained by the use of functional analysis theorems [18]. What makes Hecht-Nielsen's network particularly attractive for us is that the hidden layers are fixed independently of any function f, so that in theory this part of the neural network is trained once for n (It was demonstrated by Kurkova [19], Sprecher and Katsuura [20] and others that there are universal hidden layers that are independant even

of n). The NPC features extractor is builded from this principle : only the output layer weights are the feature vector. The remaining problem is then to estimate the hidden layer weights. Four estimation methods have been already proposed (NPC, NPC-2, DFE-NPC and LVQ-NPC). The proposed one here has the advantage of being unsupervised and clearly puts in obviousness the output weights specialization.

2.1 NPC-K coder definition

Following the Lapedes and Farber [4] model, one can see the NPC encoder as a layered neural network trained to predict time series. For a given signal frame m generated by an unknown non linear operator f, it is trained from examples of pairs of $\mathbf{x}_k = [y_{k-1}, y_{k-2}, \ldots, y_{k-\lambda}]^\top$ input vectors and y_k output samples, while minimizing the mean square error:

$$Q_m(\Omega, \mathbf{a}) = \frac{1}{2}\sum_{k}^{K}(y_k - F_{\Omega,\mathbf{a}}(\mathbf{x}_k))^2 \tag{3}$$

where $F_{\Omega,\mathbf{a}}$ is the non linear function realized by the neural network with parameters noted Ω (first layer weights) and $\mathbf{a} = [a_1, \ldots, a_N]^\top$ (hidden layer weights) including sigmoidal node functions. More precisely, $F_{\Omega,\mathbf{a}}$ can be viewed as the composition of two functions G_Ω (corresponding to the network first layer) and $H_\mathbf{a}$ (corresponding to the network output layer) such that:

$$F_{\Omega,\mathbf{a}}(\mathbf{x}_k) = \sum_i a_i \sigma[\sum_j \omega_{ij} y_{k-j}] = G_\Omega \circ H_\mathbf{a}(\mathbf{x}_k) \tag{4}$$

The NPC coding needs two computing stages. 1) the *parameters adjustment stage* which consists in the learning of the weights of the first layer Ω once a time; 2) the *features extraction stage* which occurs at every signal frame coding: only the $\mathbf{a}$ weights are learned while the hidden layer weights (issued from the first stage) remain fixed. The prediction error which must be minimized over all the sample vectors $\mathbf{x}_k$ of the frame m is then given by :

$$Q_m(\mathbf{a}) = \sum_k (y_k - H_\mathbf{a}(\mathbf{z}_k))^2 \text{ with } \mathbf{z}_k = G_\Omega(\mathbf{x}_k), \tag{5}$$

using a standard multidimensional optimisation method, e.g. steepest descent (error back propagation).

2.2 NPC distance

The first stage (first layer weights learning), which is unsupervised in our case, is done by defining a set of predictive output cells organized on a 2 dimension map. Because the comparison between patterns from the input signals space and vectors from the second layer weights space is not immediate, we need to define a specific distance. The *NPC distance* [14] between two signal frames l and m is defined as the Itakura's distance measure was in the framework of linear prediction techniques [22]:

$$d_\Omega^{NPC}(l, m) = \log \frac{Q_m(\Omega, \mathbf{a}_l)}{Q_m(\Omega, \mathbf{a}_m)} \tag{6}$$

(6) gives the ratio of the frame m prediction error using the frame l NPC parameters $\mathbf{a}_l$ and the same frame prediction error, but using the frame m NPC parameters $\mathbf{a}_m$. When applying

the m signal frame to the NPC (for a given Ω) with its adapted coding coefficients $\mathbf{a}_m$, the output residual error $Q_m(\Omega, \mathbf{a}_m)$ is minimal. On the other hand, when applying the same signal to the NPC with the adapted coding coefficients $\mathbf{a}_l$ of the l signal frame, the residual error $Q_m(\Omega, \mathbf{a}_l)$ is not minimal and one obtains $Q_m(\Omega, \mathbf{a}_l) \geq Q_m(\Omega, \mathbf{a}_m)$. For $l = m$, one has $d_\Omega^{NPC}(l, m) = 0$. Let us note that $d_\Omega^{NPC}(l, m)$ is a not a true distance since it is not symetrical.

2.3 First layer weight and predictive map trainning

We define a network structure with L output cells on a 2D map (see fig. 1) with a local neighborhood function V^σ. In traditional Kohonen map, the algorithm is based on the Euclidean distance in the input space. However in this new predictive map, we use, for consistency, the previously defined NPC distance in the signal space. The obtained algorithm is described as follows:
For all the training frames m :

1) finding the winner neuron l^* of the map such that :

$$l^* = \arg \min_{l=1,\ldots,L} d_\Omega^{NPC}(l, m) = \arg \min_{l=1,\ldots,L} \{\log \frac{Q_m(\mathbf{a}_l)}{Q_m(\mathbf{a}_m)}\} = \arg \min_{l=1,\ldots,L} \{Q_m(\mathbf{a}_l)\} \quad (7)$$

2) updating the winner neuron and its neighbors weights such as to minimize the d^{NPC} distance (this is equivalent to minimize the square prediction error) :

$$Q_m(\mathbf{a}_{1,\ldots,L}) = \sum_l^L \sum_{k(m)} (y_k - G_\Omega \circ H_{\mathbf{a}_l}(\mathbf{x}_k))^2 V^\sigma(l^*, l) \quad (8)$$

were $V^\sigma(l, l^*) = e^{-\frac{d(l,l^*)}{2\sigma}}$ is the neighborhood function (a gaussian low in our case, $d(l, l^*)$ being the length of the shortest way between l and l^* in the map and σ the standard deviation). σ is a decreasing function of the learning time such that $\sigma(q) = [\frac{\sigma_f}{\sigma_i}]^{\frac{1}{N}} \sigma(q-1)$ where σ_i and σ_f are the initial and the final imposed values of the standard deviation and N the learning iteration number.

3) updating the first layer weights by error backpropagation

The expressions that permit to adapt the vector weights are derived from the traditionnal MLP backpropagation algorithm (gradient descent) as follows :

1) output layer weigths $\mathbf{a}_l$:

$$a_{il}(q) = a_{il}(q-1) - \frac{\partial Q_m}{\partial a_{il}} \quad (9)$$

$$= a_{il}(q-1) + V^\sigma(l^*, l) \sum_{k(m)} (y_k - \phi(V_k)) \dot{\phi}(V_k) \phi_i(\mathbf{x}_k) \quad (10)$$

ϕ being the sigmoid function, V_k the l map cell potential : $V_k = \sum_j a_{jl} \phi_j(\mathbf{x}_k)$ and $\phi_i(\mathbf{x}_k)$ the output of the i^{th} first layer cell.

2) first layer weigths ω_{ji} :

$$\omega_{ji}(q) = \omega_{ji}(q-1) - \frac{\partial Q_m}{\partial \omega_{ji}} \quad (11)$$

$$= \omega_{ji}(q-1) + \sum_{l=1}^{L} a_{il} V^{\sigma}(l^*, l) \sum_{k(m)} (y_k - \phi_l^2(\mathbf{x}_k)) \dot{\phi}_l^2(\mathbf{x}_k) \dot{\phi}_i^1(\mathbf{x}_k) y_{k-j} \quad (12)$$

where $\phi_i^1(\mathbf{x}_k)$ is the activity of the i^{th} first layer cell and $\phi_l^2(\mathbf{x}_k)$ the activity of the l output map cell. $\dot{\phi}$ denotes the derivative sigmoid function.

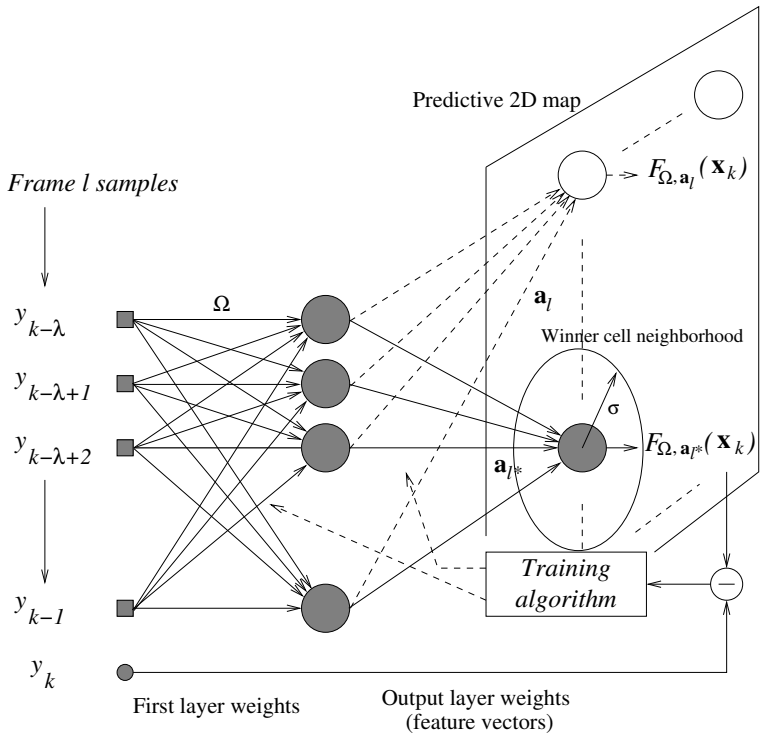


Figure 1: NPC-K coder.

2.4 NPC-K feature vector computing

There are at least two ways of using the NPC-K predictive map. One can uses it as a feature extractor or both as a feature extractor and a feature classifier. As forth-mentioned in paragraph 2.1), the first way consists in estimating the **a** weight vector while presenting the signal frames. The second way consists in using the predictive map by 1) labelling it in an adequate manner and 2) choosing the map cell which minimize the NPC distance when presenting a signal as input. This last way is of a greater interest for us : combining data modelization and data classification is one of the research interests on which we are focused.

	vowels				voiced plosives			unvoiced plosives		
frames	11701				883			3223		
phones	/aa/	/ae/	/ey/	/ow/	/b/	/d/	/g/	/p/	/t/	/k/
frames	2924	4600	2161	2016	258	312	313	623	1100	1510
%	24%	39%	18%	17%	29%	35%	35%	19%	34%	46%
cells (/64)	13	32	13	6	14	32	18	19	34	46
%	20.3%	50%	20.3%	9.3%	22%	50%	28%	15.6%	53.1%	31.2%

Table 1: Phoneme training bases

2.5 Experimental results

We built three phoneme bases extracted from the Darpa-TIMIT speech database. The first base groups four classes of voiced phonemes (vowels) very commonly used: /*aa*/, /*ae*/, /*ey*/ and /*ow*/. the second and the third bases group two series of phonemes : /*b*/,/*d*/,/*g*/ (voiced plosives) and /*p*/,/*t*/,/*k*/ (unvoiced plosives). Those phonemes are frequently used and simultaneously difficult to process. We used the two first Dialect Regions : *DR1* (see table 1) for the training set of both the NPC-K first layer estimation and both the MLP classifier training. *DR2* for the test set.

d d d d d g g g	q q q q q q q t	ow ow aa ae ae ae ae ey
d d d d d d b g g	q q q q q q p t	ow aa aa aa ae ae ae ey
d d d d d d b g g	q q p t t t t t	ow aa aa aa ae ae ey ey
d d d d g b g g	q p p p t t t t	ow ae aa aa ae ae ey ey
g d d d d d g g	p q p p t t t t	aa ae ae ae ae ae ae ey
d d d b d d g g	p q t t t t t t	aa aa ae ae ae ae ey ey
d b d b b b b b	p q q t t t t t	ow ae ae ae ae ae ey ey
d b b b g g g b	t t t t t t t t	aa ey ey ae ae ae ae ae

Table 2: Map cells labelling for the 3 phoneme bases

We trained three NPC-K coders of 16 inputs, 16 hidden cells, $8 \times 8 = 64$ predictive cells and σ varying from 8 to 0.1. After 50 training epochs (for example each epoch means 11701 frames presented to the network for the first vowels base) we then obtained the map cells labelling in table 2. A map cell is labelled according to the most frequently winner class. The coder can be then used as a phonemes classifier. The number of cells sharing the same label depends

features extractor	classifier	data set	recognition rate		
			vowels	voiced plosives	unvoiced plosives
NPC-K	NPC-K map	training set	64%	66%	76%
NPC-K	NPC-K map	test set	59%	63%	69%
NPC-K	MLP	training set	64%	88%	86%
NPC-K	MLP	test set	56%	64%	77%
LPC	MLP	training set	75%	88%	87 %
LPC	MLP	test set	70%	63%	76 %

Table 3: Phonemes recognition rates obtained from 2 layers MLP and NPC-K classifiers

on the signals class complexity but also on the ratio of the corresponding frames used for the training (see the table 1). Once the first stage is finalized, we compute the NPC-K parameters of the *DR1* and *DR2* speech frames. The *DR1* features were used to train a two layers MLP ($16 \times 10 \times 3$ cells, for the /p/, /t/, /k/ phonemes experiment for example) as a phoneme classifier (60000 training iterations). We reported on table 3 the recognition rates obtained on the three bases from both the coder and both the MLP classifier. For comparaison, we added the scores obtained using the LPC features extractor (Linear Predictive Coding) on the same data set. The visible organization of the output cells on the 2D map shows that the output layer weights carry really important features related to the modelized short-term spectrum. However comparaison with the LPC coding shows that vowels features are extracted better with LPC than with NPC parameters. On the contrary, all of the plosives features are well extracted with NPC as well as with LPC.

3 Conclusions

We have proposed a predictive self-organizing map architecture which ensure the unsupervised training of a NPC coder under the assumption that only the second layer weights carry the modelized signal features. Phoneme feature extraction experiments given in this article have shown an interesting self-organizing process of the output cells which seems to confirm the initial assumptions. Our current works are devoted to the study of an adaptative neighborhood function. We are also focusing on a non deterministic reading of the predictive map mainly because the higher levels of speech systems usually need class probability estimation.

References

[1] H. Hermansky (1990) Perceptual Linear Predictive (PLP) analysis of speech. *J. of the Acoustical Society of America* **vol. 4** p. 1738-1752

[2] H. Hermansky (1994) RASTA processing of speech. *IEEE Trans. on Speech and Audio Processing* **vol. 2** p. 587-589

[3] H. Hermansky (1998) Should recogizers have ears ? *Speech Communication* **vol. 25** p. 3-27

[4] A. Lapedes, R. Farber (1987) Nonlinear signal processing using neural networks: Prediction and system modelling. *Internal Report, Los Alamos National Laboratory*

[5] J. Thyssen, H. Nielsen, S.D. Hansen (1994) Non-linear short-term prediction in speech coding. *Proc. of Int. Conf. on Signal and Speech Processing* **vol. 1** p. 185-188

[6] W. Reichl W, S. Harengel, F. Wolferstetter, G. Ruske (1995) Neural networks for non-linear discriminant analysis in continuous speech recognition. *Eurospeech* p. 537-540

[7] B. Gas, J.L. Zarader, C. Chavy (2000) A New Approach to Speech Coding : The Neural Predictive Coding. *J. of Advanced Computational Intelligence* **vol. 4**(1) p. 120-127

[8] M. J. Hunt, C. Lefebvre (1989) A comparison of several acoustic representations for speech recognition with degraded and undegraded speech. *Int. Conf. on Speech and Signal Processing* **vol. 2** p. 262-265"

[9] B.H. Juang, S. Katagiri (1992) Discriminative Learning for Minimum Error Classification. *IEEE Trans. on Signal Processing* **vol. 40**(12) p. 3043-3054

[10] A. Biem, S. Katagiri (1994) Filter bank design based on Discriminative Feature Extraction. *Proc. of Int. Conf. on Signal and Speech Processing* **vol. 1** p. 485-488

[11] J.L. Zarader, B. Gas, D. Charlelie-Nelson, C. Chavy (2001) New compression and decompression of speech signals by NPC. *Inter. Conf. on Signal, Speech and Image Processing* p. 119-125

[12] C. Chavy, B. Gas, J.L. Zarader (1999) Discriminative coding with predictive neural networks. *Inter. Conf. on Artificial Neural Network* **vol. 4**(1) p. 219-222

[13] M. Chetouani, B. Gas, J.L. Zarader (2003) Modular neural predictive coding for discriminative feature extraction. *IEEE Inter. Conf. on Acoustic Speech and Signal Processing* **vol. 2** p. 33-36

[14] B. Gas, J.L Zarader, C. Chavy, M. Chetouani (2004) Discriminant neural predictive coding applied to phoneme recognition. *Neurocomputing* **vol. 56** p. 141-166

[15] M. Chetouani, M. Faundez-Zanuy, B. Gas, J.L. Zarader (2004) A New Nonlinear speaker parameterization algorithm for speaker identification. *Proc. of ISCA Tutorial and Research Workshop on Speaker and Recognition Langage Workshop* p. 309-314

[16] R. Hecht-Nielsen (1987) Kolmogorov's mapping neural network existence theorem. *Proc. of Int. Conf. on Neural Networks* p. 11-13

[17] F. Girosi F, T. Poggio (1989) Representation properties of networks: Kolmogorov's theorem is irrelevant. *Neural Computation* **vol. 1**(4) p. 465-469

[18] K. Hornik (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* **vol. 2** p. 359-366

[19] V. Kurkova (1992) Kolmogorov's theorem and multilayer neural networks. **vol. 5** p. 501-506

[20] H. Katsuura, D.A. Sprecher (1994) Computational aspects of Kolmogorov's superposition theorem. *Neural Networks* **vol. 7**(3) p. 455-461

[21] D.A. Sprecher (1996) A numerical implementation of Kolmogorov's superposition. *Neural Networks* **vol. 9**(5) p. 765-772

[22] F. Itakura (1975) Minimum Prediction Residual Principle Applied to Speech Recognition. *IEEE Transactions on Acoustic, Speech and Signal Processing* **vol. 23** p. 67-72

A.9 Article de revue [42]

Simultaneous Non-linear Prediction and Discrimination for Improved Speech Feature Extraction

M. CHETOUANI and B. GAS and J.L. ZARADER

article soumis à Neurocomputing (juin 2005)

.

SIMULTANEOUS NON-LINEAR PREDICTION AND DISCRIMINATION FOR IMPROVED SPEECH FEATURE EXTRACTION

M. Chetouani, B. Gas, J.L. Zarader

Laboratoire des Instruments et Systèmes d'Ile-De-France
Université Paris VI,
4 Place Jussieu, 75252 Paris Cedex 05, France
mohamed.chetouani@lis.jussieu.fr gas@ccr.jussieu.fr zarader@ccr.jussieu.fr

Abstract

Speech recognition systems are composed of different modules involving signal processing, pattern recognition and so on. In this paper we focus on the design of the feature extractor stage which aims to compute optimal vectors for the next pattern classification stage. We propose a new non-linear feature extraction method based on the Neural Predictive Coding (NPC) scheme integrated with a Learning Vector Quantization (LVQ) stage. The main idea is to design an improved feature extractor based on the NPC by the introduction of additional discriminant constraints provided by the LVQ classifier. This cooperation between the NPC and the LVQ is achieved with the help of the well-known Discriminative Feature Extraction (DFE) framework. The feature extractor and the classifier are designed for simultaneuos optimisation of the same objective: the Minimization of the Classification Error (MCE). The effectiveness of the proposed method is estimated on phoneme classification tasks. We evaluate the performances on the NTIMIT database (telephone quality), focussing the investigations on high confusable phonemes: front vowels (/ih/, /ey/, /eh/, /ae/), voiced plosives (/b/, /d/, /g/) and unvoiced plosives (/p/, /t/, /k/). The performance is compared with other widely used coding methods namely, the Linear Predictive Coding (LPC), Mel Frequency Cepstral Coding (MFCC) and the Perceptual Linear Predictive (PLP) schemes. The experiments show an improvement in the rates by the use of our proposed non-linear discriminant method.

Key words: Data-driven Feature Extraction, Neural Predictive Models, Non-linear Speech Processing.
PACS:

1 Introduction

In pattern recognition, there are two main ways to improve the performance of the recognizer: namely by using a feature extractor and a classifier (see figure 1). For research purposes, one could aim to investigate a sophisticated feature extractor which allows a better description of the pattern. On the other hand, one can choose to use a basic descriptor or feature extractor and develop a sophisticated classifier. As pointed by Duda et al.[1], the distinction between feature extraction and classifier stages is made only for practical reasons. In speech processing, the distinction is more accentuated since the methods for classification and feature extraction are very different. Most of the classification methods at present are statistical based while the feature extraction ones are frequential and/or temporal methods. One of the reasons is that by examining the pattern shape (speech signal) one cannot directly or easily find appropriate features and this is in contrast with other pattern classification problems such as handwritten or object recognition problems. Some methods were initially proposed for spectrogram reading[2]) about two decades ago, whilst more recently, methods related to speech signal description such as energy or zero crossing rate have been proposed.

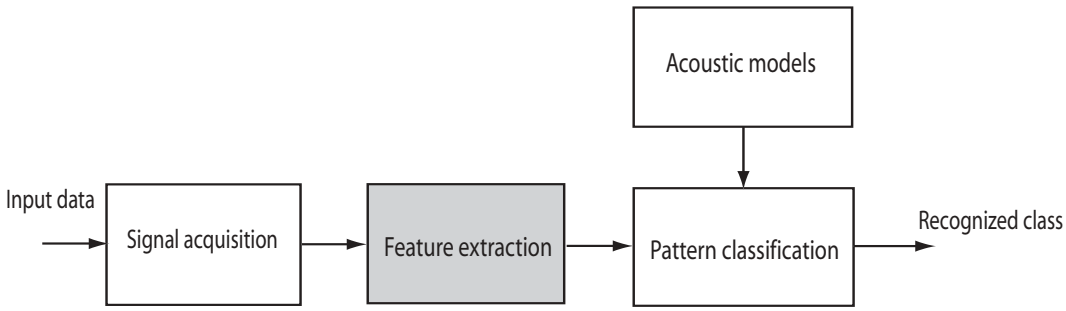


Fig. 1. Automatic speech recognition system: an association of specialized modules.

2 Problem dimensions

2.1 Traditional methods

The most commonly used speech coding methods are the Mel Frequency Cepstral Coding (MFCC) and the Perceptual Linear Predictive (PLP) methods. These methods are examples of successful auditory knowledge integration. Indeed, they are respectively based on the Mel and the Bark scales and both of them are inspired by the human auditory process. Various models which try to mimic this process have been proposed[3,4]. However, as pointed out by Hermansky [5], the human properties are not all implementable and useful

for recognition. Some auditory properties such as non-linear scale or spectral compression seem to be efficient.

In contrast with the MFCC or PLP, Linear Predictive Coding (LPC) is motivated from a speech production perspective. It is based on an Auto-Regressive (AR) model of the vocal tract and has been intensely used in transmission but less in recognition tasks.

All these methods have been successfully applied to speech processing but they have some limitations. First, speech signals result from non-linear phenomena due to several physiological reasons (vocal tract, air flow turbulences, ...)[6–8]. Secondly, there is no *a priori* class information during the coding process: the features are extracted in the same way for all the sounds in spite of their obvious differences (vowels, voiced, unvoiced, nasals, ...). In this contribution, we investigate introduction of class membership information and non-linear modelling during the coding phase in contrast with what is usually done.

2.2 Related works

2.2.1 Non-linear feature extraction

In order to overcome these limitations, several feature extraction methods have been considered. Lee et al.[9] have proposed a feature extractor based on the Independent Component Analysis (ICA). The ICA provides the possibility of Higher-Order Statistics (HOS) extraction which is useful for non-linear characterization. It has been applied to word recognition and it has shown a relative error reduction of 47.4% compared to the MFCC. Another non-linear feature extraction method is the reconstructed phase space[10]. This method is derived from non-linear/chaotic signal processing techniques and it highlights the dynamic characteristics of the process (i.e. sound production dynamics). In a phoneme classification task from the TIMIT database, the reconstructed phase space scores (31.23%) were outperformed by the MFCC (50.31%) but it was found to give promising results when used in fusion[11]. Other methods based on signal processing approaches have also been proposed such as the Discrete Wavelet Transform[12] or Modulations[13].

2.2.2 Knowledge integration

Knowledge integration during the feature extractor design is one way for achieving global improvement. Methods related to this integration are called Data-driven methods[14]. These methods can simplify the classification's task by decreasing the required complexity. Ideally, an optimal feature extractor should suppress the unwanted characteristics from the speech signal and

keep only the appropriate ones. However this optimal feature extractor is not known.

For this purpose, Malayath et al.[14] proposed to modify the traditional spectral methods used in speech processing by the introduction of class membership informations (*a priori* informations). Their work involved the optimization of spectral basis functions by Linear Discriminant Analysis (LDA). The obtained basis functions are an alternative to the traditional Discrete Cosine Transform (DCT) and are more adapted for the classification purpose. This feature extractor has been applied to a connected digit recognition task, and it showed a relative error reduction of 28.8%.

The previous works, proposed in the literature, demonstrate the need of effective feature extractors for speech recognition. One can argue that the optimal feature extractor design has to be carried out while taking into account the final purpose. For instance, a given method can extract relevant features for speech recognition without being efficient for speech transmission. Several authors[5,15] have pointed out the need of using an adequate representation for each speech application.

3 Scope of the paper

In this paper, we focus on two key elements for speech feature extraction: non-linear modelling and data-driven methods. The first point has been highlighted by the previous works (cf. §2.2). Non-linear processing has been successfully applied to speech synthesis[16], speech coding[17], prediction[18], speaker recognition[19,20] and language identification[21]. On the other hand, data-driven methods are usually introduced by a discriminative criteria based on statistical approaches (data analysis, maximization of the mutual information MMI etc.).

Section 4 presents the approach used for non-linear extension of a well-known method namely, the Linear Predictive Coding. Then, we describe the proposed model termed the Neural Predictive Coding. Section 6 examines the methods proposed in the literature for discriminative feature extraction. The next section proposes a new scheme involving non-linear predictive models and discriminative criteria. Section 8 is devoted to the evaluation of the new model for phoneme classification. Finally, we give some concluding remarks and future work recommendations.

4 Non-linear extension of the LPC model

As mentioned earlier, the Linear Predictive Coding (LPC) method is motivated by the speech production model. The LPC coefficients are related to the vocal tract model[22]. This coding method has been intensely applied in speech processing[15] and offers numerous advantages. For one, it is a rigorous methematically defined model and at the same time, it is very simple to implement. However, it also has some limitations, namely:

- it is not well adapted for unvoiced speech signals,
- linear based: it does not take into account non-linear phenomena present in speech signals,
- high variability within recognition classes[15],
- it is not optimized for recognition.

Several authors have investigated some solutions (see Ref.[15] for a good review). In this paper, we propose a non-linear extension of the LPC using neural networks: termed the Neural Predictive Coding (NPC)[23].

The goal of the NPC model is to extract relevant features for speech recognition and is not normally used as a classifier or a non-linear predictor. The objective here is to encode the speech signal by prediction error minimization with discriminant constraints.

5 The Neural Predictive Coding: a Non-Linear Auto-Regressive model

The Neural Predictive Coding (NPC) model (cf. figure 2) is basically a non-linear extension of the well-known LPC encoder. As in the LPC framework with the Auto-Regressive (AR) model, the vector code is estimated by prediction error minimization. The main difference lies in the fact that the model is non-linear (connectionist based):

$$\hat{y}_k = F(\mathbf{y}_k) = \sum_j a_j \sigma(\mathbf{w}^T \mathbf{y}_k) \tag{1}$$

Where F is the prediction function realized by the neural model. $\hat{y}_k$ is the predicted sample. $\mathbf{y}_k$ the prediction context: $\mathbf{y}_k = [y_{k-1}, y_{k-2}, ..., y_{k-\lambda}]^T$ and λ the length of the prediction window. $\mathbf{w}$ and $\mathbf{a}$ represent the first and the last (output) layer weights. σ is a non-linear activation function (the sigmoid function in our case).

Neural networks have been used in speech prediction due to their efficient approximation capabilities[24]. Multi-Layer Perceptron (MLP)[17], Radial Basis Functions (RBF)[25] and recurrent networks[17,18] have been successfully applied to non-linear speech prediction to-date. However other models can also be used such as Volterra filters[26], quadratic model[27], locally linear methods[28] or non-parametric methods[29].

The key idea of the NPC model is to use it as a non-linear auto-regressive model. As in the LPC framework, the NPC weights (predictor coefficients) represent the vector code. It is well-known that the neural weights can be considered as a representation of the input vector. A drawback of this method is that non-linear models have no clear physical meaning[15]. The solution weights can be very different for a same minimum of the prediction error. To overcome this limitation, we impose constraints on the weights.

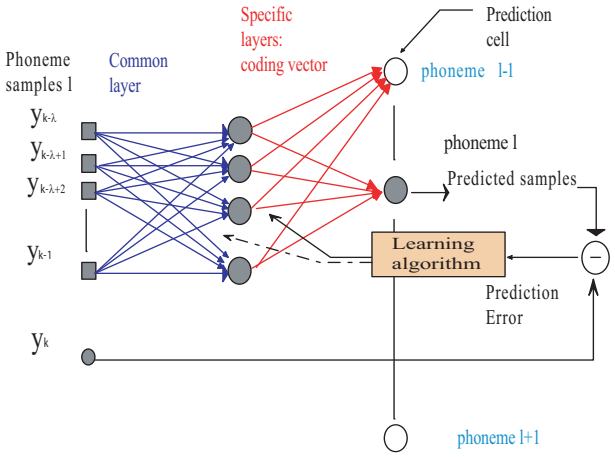


Fig. 2. Neural Predictive Coding (NPC) architecture: a connectionist model used as a non-linear predictor.

5.1 Description

The original NPC model is a Multi-Layer Perceptron (MLP) with one hidden layer. Here we consider a model with two hidden layers. Only the output layer weights are used as the coding vector instead of all the neural weights. For this, we assume that the function F realized by the model, under convergence assumptions, can be decomposed into two functions: $G_{\mathbf{w}_{1,2}}$ ($\mathbf{w}_{1,2}$ first and second layers weights) and $H_{\mathbf{a}}$ ($\mathbf{a}$ output layer weights):

$$F_{\mathbf{w}_{1,2},\mathbf{a}}(\mathbf{y}_k) = H_{\mathbf{a}} \circ G_{\mathbf{w}_{1,2}}(\mathbf{y}_k) \tag{2}$$

With $\hat{y}_k = H_{\mathbf{a}}(\mathbf{z}_k)$ and $\mathbf{z}_k = G_{\mathbf{w}_{1,2}}(\mathbf{y}_k)$.

The learning phase is realized in two stages. First, the *parameterization phase* involves the learning of all the weights by the prediction error minimization criterion:

$$Q = \sum_{k=1}^{K} (y_k - \hat{y}_k)^2 = \sum_{k=1}^{K} \left(y_k - F(\mathbf{y}_k)\right)^2 \tag{3}$$

With y the speech signal, $\hat{y}$ the predicted speech signal, k the samples index and K the number of samples.

In this phase, only the first layer weights $\mathbf{w}_{1,2}$ (which are the NPC encoder parameters) are kept. Since the NPC encoder is set up by the parameters defined in the previous phase, the second phase, called the *coding phase*, involves computation of the output layer weights $\mathbf{a}$: representing the phoneme coding vector. This is done also by prediction error minimization but only the output layer weights are updated. One can notice that the output function is linear (cf. equation 1), so it can use simple LMS-like algorithms. Here, for consistency with the *parameterization phase*, it is done by the backpropagation algorithm.

The NPC weights modification law $\Delta\mathbf{a}^{Pred}$ and $\Delta\mathbf{w}_{1,2}^{Pred}$ are proportional to the gradient of the prediction errors $Q_{NPC} = \sum_i^M Q_i$, with M representing the number of classes.

6 Discriminative Feature Extraction: a new way for feature extraction improvement

The NPC model has been successfully applied to feature extraction for phoneme classification[23]. Previous works have shown that efficient performances are obtained by the introduction of discriminative criteria during the *parameterization phase*. We have proposed a criterion based on prediction errors ratio called the Modelization Error Ratio (MER)[23]. This criterion is related to the Maximization of the Mutual Information (MMI)[30] and the principle allows the design of a discriminant feature extractor (called the DFE-NPC) in independently of the classifier. Indeed, during the DFE-NPC *parameterization phase* there is no cooperation with the next classification stage. Here, the proposed work involves a different scheme: simultaneous optimization of the feature extractor and the classifier.

6.1 Classification guided discrimination

The Discriminative Feature Extraction (DFE) framework is an example of constraints imposed by the classifier for achieveing the same goal, namely the Minimum Classification Error (MCE). The key idea of the MCE criterion is to update the parameters of the classifier and/or the feature extractor for minimum classification error (see figure 3). As has been shown in (Ref.[31]), the traditional discriminant methods (i.e. Minimum Square Error (MSE), Maximization of the Mutual Information (MMI) and so on) do not necessarily minimize the classification error.

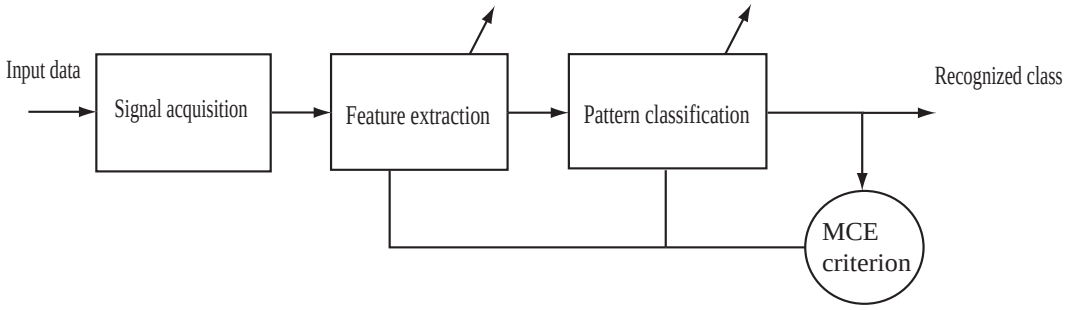


Fig. 3. Discriminative Feature Extraction based on the Minimum Classification Error.

The minimisation of the MCE criterion (see figure 3) is realized by an adequate update rule: the Generalized Probabilistic Descent (GPD)[31]. The final purpose needs several steps. First, a loss function that reflects the classification performances of the recognizer is defined. A misclassification measure is extended from the loss function. This measure aims to evaluate the distance between one specific class and the others. Then, the GPD is used to minimize the misclassification distance which is passed through a non-linear and differentiable function such as the sigmoid function. The DFE framework based on the MCE criterion has been successfully applied in both speech recognition[32,33] and speaker recognition[34].

6.2 Application to the NPC model

One of the major drawbacks of simultaneous training is the eventual suboptimality caused by local minima[33]. To overcome this limitation, Torre et al.[33] proposed independent training of both the feature extractor and the classifier.

In this contribution, we propose simultaneous training of both the NPC and the classifier. The learning problem is overcome by choosing an adequate classifier in terms of complexity. Due to the NPC complexity (MLP-based predic-

tor) and the NPC objective (feature extraction by the weights), an appropriate classifier family of prototype-based classifiers is obtained. The learning procedure involves prototypes' (representatives') adjustment in order to describe optimal class boundaries.

The simultaneous training of a prototype-based classifier and the NPC makes it possible to impose discriminant constraints on both the NPC weights and, equally importantly, on the feature vectors (output layer weights) while minimizing the prediction error.

6.3 Prototype-based classifiers

In order to implement prototype-based classifiers, one can use different methods: unsupervised (K-means, clustering methods, Self Organization Maps and so on) or supervised (k-NN, LVQ). As mentioned previously, we need to impose discriminant constraints so we use supervised methods and more precisely the Learning Vector Quantization (LVQ) classifier.

The LVQ classifier has been successfully applied to different domains such as handwritten recognition[35] or speech recognition[36]. Several versions have been proposed: LVQ1, LVQ2 or LVQ3. The MCE/GPD framework has been applied to the LVQ classifier resulting in a model close to the LVQ2[31]. In the next sections, we use this model for achieving the LVQ-NPC cooperation.

7 Simultaneous training of the LVQ classifier and the NPC feature extractor

In the DFE framework, the feature extractor and the classifier are considered as a single module described by $\mathbf{\Phi} = (\mathbf{a}, \mathbf{m})$ ($\mathbf{m}$ being the LVQ classifier prototypes).

7.1 DFE framework

The DFE framework needs the definition of a discriminant function for each class. In the case of the LVQ classifier, the discriminant function is the negative of the minimum distance from the input pattern (the feature vector) to the genuine prototype:

$$g_i(\mathbf{a}) = -\min_{\tau} \|\mathbf{a} - \mathbf{m}_{i,\tau}\|^2 = -\min_{\tau} \ \mathrm{d}(\mathbf{a}, \mathbf{m}_{i,\tau}) \tag{4}$$

where $\mathrm{d}(\mathbf{a}, \mathbf{m}_{i,\tau})$ is the Euclidean distance between the input pattern $\mathbf{a}$ (feature vector) and $\mathbf{m}_{i,\tau}$ is the τth prototype of the class C_i.

According to the DFE framework, we define the misclassification measure as:

$$\mu_i(\mathbf{a}) = -g_i(\mathbf{a}) + \left[\frac{1}{M-1} \sum_{j \neq i} g_j(\mathbf{a})^{-\psi} \right]^{-\frac{1}{\psi}} \tag{5}$$

where ψ is a positive number. For a large ψ, the misclassification measure becomes:

$$\mu_i(\mathbf{a}) = -g_i(\mathbf{a}) + \overline{g}_i(\mathbf{a}) \tag{6}$$

$\overline{g}_i(\mathbf{a})$ is the competing discriminant function (anti-discriminant function) to the class C_i. This leads to only consider the first incorrect prototype[36]:

$$\overline{g}_i(\mathbf{a}) = \max_{j \neq i} \mathrm{g}_j(\mathbf{a}) \tag{7}$$

The misclassification measure $\mu_i(\mathbf{a})$ (5) has to be positive when $\mathbf{a}$ is misclassified and negative if it is not the case:

$$\mu_i(\mathbf{a}) = \mathrm{d}(\mathbf{a}, \mathbf{m}_{i,\tau}) - \mathrm{d}(\mathbf{a}, \mathbf{m}_{j,\upsilon}) \tag{8}$$

where $\mathbf{m}_{i,\tau}$ is the closest prototype of the genuine class while $\mathbf{m}_{j,\upsilon}$ is the closest prototype of the incorrect class.

The next step involves defining the MCE loss function which reflects the classification errors:

$$l_i(\mathbf{a}) = l_i(\mu_i(\mathbf{a})) = \frac{1}{1 + e^{-\zeta \mu_i(\mathbf{a})}} \tag{9}$$

The MCE objective function is the following empirical loss:

$$L(\mathbf{a}, \mu) = \sum_{n=1}^{N} \sum_{i=1}^{M} l_i(\mathbf{a}_n) \delta_{C(\mathbf{a}_n) - i} \tag{10}$$

where $C(\mathbf{a}_n)$ is the class membership of the feature vector $\mathbf{a}_n$ and δ is the Kronecker symbol which equals 1 when $C(\mathbf{a}_n) = i$. N is the number of frames and M the number of classes.

The Generalized Probabilistic Descent (GPD) is applied for updating the parameters $\mathbf{\Phi} = (\mathbf{a}, \mathbf{m})$:

$$
\begin{aligned}
\mathbf{a}_n &= \mathbf{a}_n - \beta(t)\frac{\partial l_i(\mathbf{a}_n)}{\partial \mathbf{a}_n} \\
\mathbf{m}_{i,\tau} &= \mathbf{m}_{i,\tau} - \alpha(t)\frac{\partial l_i(\mathbf{a}_n)}{\partial \mathbf{m}_{i,\tau}} \\
\mathbf{m}_{j,\upsilon} &= \mathbf{m}_{j,\upsilon} - \alpha(t)\frac{\partial l_i(\mathbf{a}_n)}{\partial \mathbf{m}_{j,\upsilon}}
\end{aligned}
\tag{11}
$$

where $\alpha(t)$ and $\beta(t)$ are the learning rates of the LVQ classifier and the NPC model respectively. The learning rates are a decreasing function of the epoch index t.

According to the MCE loss function (9), the updating rules for the LVQ parameters can be written as:

$$
\begin{aligned}
\mathbf{m}_{i,\tau} &= \mathbf{m}_{i,\tau} + 2\alpha(t) l_i(\mathbf{a}_n)(1 - l_i(\mathbf{a}_n))(\mathbf{a}_n - \mathbf{m}_{i,\tau}) \\
\mathbf{m}_{j,\upsilon} &= \mathbf{m}_{j,\upsilon} - 2\alpha(t) l_i(\mathbf{a}_n)(1 - l_i(\mathbf{a}_n))(\mathbf{a}_n - \mathbf{m}_{j,\upsilon})
\end{aligned}
\tag{12}
$$

And for the NPC model, the updating rule for the feature vector $\mathbf{a}_n$ is:

$$
\Delta\mathbf{a}_n^{MCE} = -2\beta(t) l_i(\mathbf{a}_n)(1 - l_i(\mathbf{a}_n))(\mathbf{m}_{i,\tau} - \mathbf{m}_{j,\upsilon}) \tag{13}
$$

One can notice that the feature vectors are updated in a function of the distance between the two prototypes: the genuine and the incorrect. They are updated in the direction of the maximum separability between these two classes.

These discriminant contributions also need to be associated with the prediction modifications: $\Delta\mathbf{w}_{1,2}^{Pred}$ and $\Delta\mathbf{a}^{Pred}$ defined in 5.1.

7.2 Cooperation

The objective of this cooperation is to introduce discriminant constraints on the NPC *parameterization phase*. Several solutions can be used, for instance one can use constraint minimization like in (Ref.[37]) where the simultaneous training of classifiers is processed with a Lagrangian formalism. We opt for another approach. The two optimizations are moderated with the help of a coefficient θ.

Consequently, the modification for the feature vector $\mathbf{a}_n$ combines prediction

and discrimination:

$$\Delta \mathbf{a}_n = \theta \Delta \mathbf{a}_n^{Pred} + (1-\theta)\Delta \mathbf{a}_n^{MCE} \tag{14}$$

The second phase of the cooperation involves the modification of the first two layers in the maximum class separability direction. However, the relation between the first two layers' weights $\mathbf{w}_{1,2}$ and the MCE criterion (9) is not direct as for the output layer weights. Considering the NPC objective in the cooperation, that reverts to bringing the features closer to their adequate prototypes and to move away from the incorrect prototypes. In other words, the feature vector $\mathbf{a}_{i,n}$ produced by the NPC model for the analysis window $\mathbf{y}_{i,n}$ (belonging to the class C_i) must be close to one of the prototypes $\mathbf{m}_{i,\tau}$.

In order to achieve the above, we introduce a new stage into the NPC model. For the window $\mathbf{y}_{i,n}$, we determine the two modifications necessary for:

- Bringing the feature closer to the prototype $\mathbf{m}_{i,\tau}$: minimization of the prediction error under the constraint: the output layer is fixed to $\mathbf{m}_{i,\tau}$. One obtains the modification of the first two layers $\Delta \mathbf{w}_{1,2}^{mod}$.
- Move away from the prototype $\mathbf{m}_{j,\upsilon}$: maximization of the prediction error under the constraint: the output layer is fixed to $\mathbf{m}_{j,\upsilon}$. One obtains the modification of the first two layers $\Delta \mathbf{w}_{1,2}^{disc}$.

During the above two processes, one estimates the modifications necessary to maximize the separability of the classes from the first layers' ($\mathbf{w}_{1,2}$) point of view.

The modification law of the first layers is a moderation of these two effects:

$$\Delta \mathbf{w}_{1,2} = \theta \Delta \mathbf{w}_{1,2}^{mod} + (1-\theta)\Delta \mathbf{w}_{1,2}^{disc} \tag{15}$$

One can notice that this modification law does not integrate the modification of the NPC model $\Delta \mathbf{w}_{1,2}^{Pred}$. Indeed, this modification is not useful any more because the contribution $\Delta \mathbf{w}_{1,2}^{mod}$ makes it possible to take into account the modelling part required by the LVQ-NPC process.

After the *parameterization phase*, the LVQ is no longer used. The *coding phase* is identical to all other models (cf. §5.1)

8 Experiments in feature extraction for phoneme classification

In general, the evaluation can be carried out on different tasks. For instance, we applied successfully the NPC model to speaker identification[38]. Here, we

focus on a different task: phoneme classification. This task is adapted for the comparison of the discriminant capabilities of feature extractors.

8.1 Experimental conditions

The NTIMIT database[39] is used in the experimental part of this work. This database is composed of TIMIT utterances which are transmitted over telephone lines. Generally, the performances of speech recognition systems are degraded using this kind of a database. For instance, in (Ref.[39]), several tests were carried out for better understanding the sources of degradation. It was shown that the phoneme error rates using the TIMIT/NTIMIT tasks increase by about 10%.

The NTIMIT database is composed of 10 sentences pronounced by 630 speakers from 8 areas of the United States. For each speaker, the sentences are grouped on three types. There are 2 dialect calibration sentences (SA), 5 phonetically-compact sentences (SX) and 3 phonetically-diverse sentences (SI). The two SA sentences are identical for all the speakers and are usually not used for training and testing because they might produce a bias.

In this work, we use the two first dialect regions DR1 (New England) and DR2 (Northern) with a predefined configuration for the training set comprising 114 speakers (DR1: 24 male and 14 female, DR2: 53 male and 23 female) and the test set comprising 37 speakers (DR1: 7 male and 4 female, DR2: 18 male and 8 female).

We focus on subset composed by front vowels (/ih/, /ey/, /eh/, /ae/), voiced plosives (/b/, /d/, /g/) and unvoiced plosives (/p/, /t/, /k/). This choice can be justified by the fact that the classification of these phonemes is known to be difficult. The number of phonemes for each class is described in table 1. A part of the training set is used as a cross-validation base, both in order to stop the classifiers' training and also the LVQ-NPC *parameterization phase*.

We carry out phoneme context-independent classification., and as a result, the performances shown represent frame-by-frame classification rates. Depending on their duration, each phoneme is split into a number of frames. The analysis window size is fixed to 32ms (8kHz for NTIMIT) with an overlapping of 16ms.

8.2 Classifiers

The proposed work is an evaluation of a new feature extractor: the LVQ-NPC. Consequently, we make comparisons with the most commonly used methods:

Table 1
Database composition: training, cross-validation and test sets.

Phoneme	ih	ey	eh	ae	b	d	g	p	t	k
Train	911	438	712	494	316	575	236	561	848	837
Cross-validation	100	100	100	100	50	50	50	100	100	100
Test	316	167	292	153	199	192	100	201	310	273

the Mel Frequency Cepstral Coding (MFCC) and the Perceptual Linear Predictive (PLP) speech coding methods. We also compare with the Linear Predictive Coding (LPC) since the proposed model NPC is a non-linear extension of the LPC. For all these methods, the feature vector dimension is set to 12 without dynamic parameters like Δ and $\Delta\Delta$.

The classification is done on a frame-by-frame basis without context dependency. Moreover, it must be carried out by several kinds of classifiers in order to measure the discriminant power of each feature extraction method. For each group of phonemes (i.e. vowels, voiced and unvoiced phonemes), we train a complete system: feature extractor plus classifier.

8.2.1 Gaussians Mixture Models (GMM)

This model is based on densities estimation of each class. GMMs are trained by the help of the Expectation-Maximization (EM) algorithm with diagonal covariance matrices assumption. As this classifier is sensitive to initialization, the parameters are initialized by the k-means algorithm (10 iterations) with $k = 16$. Classification is done according to the maximum likelihood (ML) criterion.

8.2.2 Prototypes classification (LVQ)

The LVQ model (Learning Vector Quantization) is a prototype-based classifier. The training and the testing are carried out by the consideration of the Euclidean distance. As this method is also sensitive to initialization, we use the k-means algorithm with $k = 50$. The LVQ model used for classification is different from the model used for LVQ-NPC cooperation. Here, we use the LVQ1 algorithm first in order to measure LVQ-NPC generalization capabilities and also to compare the coding methods with prototype classifiers.

8.2.3 Multi-Layer-Perception (MLP)

The MLP is based on non-linear discriminant functions. The model has one hidden layer of 10 neurons and the input size is 12 (feature vector size). The

training is done by the Levenberg-Marquardt algorithm.

8.2.4 LVQ-NPC

Like for all prototype-based classifiers, a key parameter is the optimal number of prototypes by class. We carried out different classification experiments with the LVQ-MCE classifier. We present results for voiced plosives but similar behavior is obtained for other classes. Figure 4 shows the classification rates versus the number of prototypes. We can notice that an optimal number of prototypes by class seems to be 25. Indeed, this number gives efficient performances on both training (69.52%) and test (67.11%). For 50 prototypes by class, equivalent results are obtained (training: 71.65%, test: 67.48%) but for implementation reasons we set the number of prototypes by class to 25. Figure 4 shows also that a greater number of prototypes obviously increase the classification rates for the training part but penalize the generalization capability.

Another important parameter is the moderating one (cf. equations 14 and 15). We use a traditional decreasing law:

$$\theta(t) = \theta_0 \left(1 - \frac{t}{T}\right) \tag{16}$$

where T is the iteration number. The value of θ_0 differs according to the phonetic group: 0.6 for vowels and 0.7 for plosives (voiced and unvoiced). Such evolution law shows that it is necessary to start with modelling classes and then to progressively increase the discrimination.

As mentioned earlier (cf. §5.1), explicit feature extraction is carried out during the *coding phase* using the backpropagation algorithm. We set different iterations for each kind of phoneme: 5 for vowels and 10 for plosives. The coding vector dimension is set to 12 and the NPC structure to 20-16-12-1 (representing size of the prediction window, first hidden layer size, second hidden layer size and output layer size respectively).

8.3 Results

In this section, we present the results for phoneme classification. The classification rates shown are those obtained on the test set. The usefulness of discrimination is evaluated by comparison with the NPC model without LVQ cooperation (i.e. the original NPC model with two hidden layers). In this way, we evaluate the importance of both discrimination and non-linear modelling.

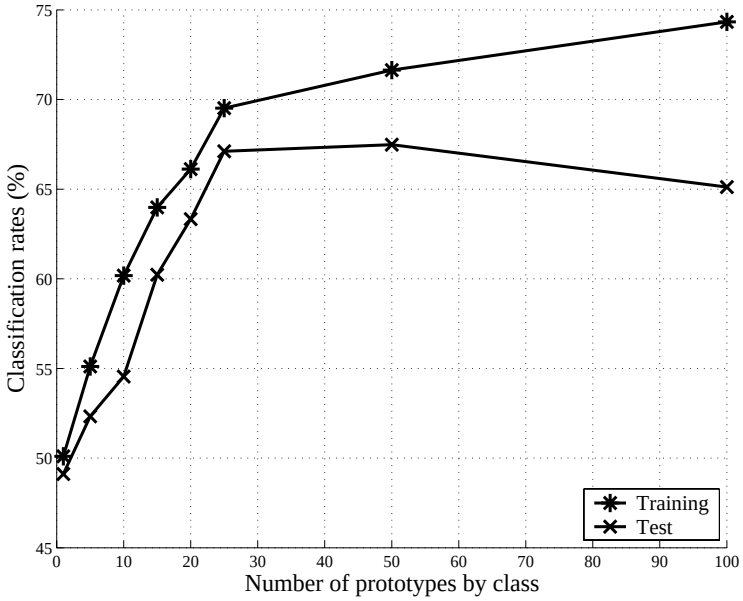


Fig. 4. Classification rates versus the number of prototypes by class.

Table 2 shows the classification rates for vowels with various coding and classification methods. One can notice that the proposed methods (NPC and LVQ-NPC) improve the classification rates. The non-linear modelling by neural networks (NPC: 49.03%) and the discrimination (LVQ-NPC: 54.81%) allow an improvement of more than 6% compared to the MFCC coding. LPC features are not used in speech recognition tasks but a comparison with their non-linear extension (NPC) is very important and it shows the efficiency of non-linear processing (cf. 2).

A key point of the results is that the proposed simultaneous learning during the *parameterization phase* between the LVQ-MCE classifier and the NPC feature extractor shows good generalization capabilities. Indeed during the *coding phase* with other classifiers, we obtain 54.81% for GMM, 53.12% for LVQ1 and 52.23% for the MLP. The best performances for all the feature extractors are obtained by GMM which is the most widely used classifier in speech recognition. One can also observe that the MLP gives better results for MFCC, PLP and NPC than the LVQ1 classifier. However it is not the case for the LVQ-NPC feature extractor since the classification rates for LVQ1 (53.12%) are better than the MLP (52.23%). It seems to be due to the LVQ-NPC cooperation during the *parameterization phase* which introduces prototype-like discrimination.

The /b/,/d/,g/ task has been used for the validation of the Time Delay Neural Network (TDNN)[41]. Those plosives are also voiced phonemes and we find a similar behavior for the coding methods (cf. table 3). The non-linear methods also give an improvement (NPC: 62.24%) and the discriminant model gives the best results with 66.33% for the LVQ-NPC. One can also notice the general-

Table 2
Classification rates for the vowels: /ih/, /ey/, /eh/, /ae/: significative improvements by non-linear and discriminant methods.

	LPC	MFCC	PLP	NPC	LVQ-NPC
GMM	35.22	48.12	45.12	49.03	54.81
LVQ1	36.53	43.21	40.33	46.66	53.12
MLP	41.21	44.23	43.22	47.31	52.23

ization capabilities of our approach by comparing the result of the LVQ-NPC with the LVQ-MCE classifier during the *parameterization phase* (67.11%) in figure 4 and those obtained by other classifiers (cf. table 3).

Table 3
Classification rates for the voiced plosives: /b/, /d/, /g/.

	LPC	MFCC	PLP	NPC	LVQ-NPC
GMM	54.13	59.23	57.21	62.24	66.33
LVQ1	50.22	57.82	57.03	60.33	64.55
MLP	55.31	58.12	57.63	61.86	63.89

The classification of unvoiced plosives (/p/,/t/,k/) is interesting because they are regarded as less predictable, and hence less attractive for predictive models. Indeed, it can be seen that the NPC results (49.36%) are lower than the MFCC ones (51.45%) for this case. However, proposed use of discrimination makes it possible to overcome this problem and to obtain better results independently to the classification method (cf. table 4). Once again, the best classification rates are obtained by the GMMs for this case.

Table 4
Classification rates for the unvoiced plosives: : /p/, /t/, /k/.

	LPC	MFCC	PLP	NPC	LVQ-NPC
GMM	44.10	51.45	46.98	49.36	53.22
LVQ1	43.03	50.12	46.22	48.07	50.12
MLP	45.12	50.56	45.88	48.66	50.98

The classification rates on the NTIMIT database not only show the need but also the efficiency of non-linear feature extractors. For all the voiced phonemes, the rates are improved by the introduction of the NPC model. For unvoiced phonemes, the predictive models' performances are penalized even if they are non-linear. In this task, the discrimination is shown to increase the classification rates. One can therefore conclude that we need to exploit the non-linear domain not only for modelization but also for discrimination.

9 Conclusions

In this paper, we have presented a new feature extractor method which makes use of non-linear modelling with neural networks and simultaneous discrimination capability provided by a classifier. This feature extractor is termed the LVQ-NPC since it is based on simultaneous learning of the Learning Vector Quantization and the Neural Predictive Coding models. The proposed method has many advantages. Firstly, the feature extraction aims to take into account the non-linear phenomena usually observed in speech production. Secondly, we are able to extract features with discriminant constraints in order to keep the relevant characteristics of the speech signal for the application (phoneme classification).

We proposed a simple and original learning procedure for the NPC and the LVQ within the Discriminative Feature Extraction (DFE) framework based on the Minimum Classification Error (MCE). The cooperation shows that the NPC weights are updated with two criteria: minimization of the prediction error and minimization of the classification error. In this way, we benefit from the advantages of both criteria. The complexity during the *parameterization phase* introduced by LVQ-NPC cooperation is different from the complexity during the *coding phase*. Indeed, this phase can be done almost as simply as for the LPC coding method.

The LVQ-NPC has been applied to the phoneme classification task using the NTIMIT database. It has demonstrated good performance compared to the most widely used coding methods MFCC and PLP. The model has also been shown increase the classification rates even if the phonemes are unvoiced and have a high confusability (/p/,/t/,/k/ phonemes). The generalization capabilities have been estimated by the use of different classification methods (GMM, LVQ1 and MLP).

Our current studies are devoted to the extension of the proposed framework to achieve simultaneous learning with other classifiers such as MLP, GMM and HMM.

References

[1] Richard O. Duda and Peter E. Hart and David G. Stork, "Pattern Classification," *Wiley-Interscience Publication*, (2001).

[2] R. De Mori and M. Palakal, "On the use of computer vision techniques for automatic speech recognition," *Proc. of ICVPR*, 691–6931 (1985).

[3] O. Ghitza, "Auditory Models and Human Performance in Tasks Related to Speech Coding and Speech Recognition," *Proc. IEEE Trans. on Speech and Audio Processing*, **2**,1, 115–132 (1994).

[4] S. Greenberg, "Representation of speech in the auditory periphery," *Journal of Phonetics, Special Issue*, **16**,1, January, (1998).

[5] H. Hermansky, "Should Recognizers Have Ears?," *Proc. ESCA Tutorial and Research Workshop on Robust Speech Recognition for Unknown Communication Channels*,1–10 (1997).

[6] M. Faundez-Zanuy, G. Kubin, W.B. Kleijn, P. Maragos, S. McLaughlin, A. Esposito, A. Hussain, J. Schoentgen, "Nonlinear Speech Processing: Overview and Applications," *Control and Intelligent Systems ACTA Press*, **30**,1, 1–10 (2002).

[7] S. McLaughlin, S. Hovell and A. Lowry, "Identification of nonlinearities in vowel generation," *Proc. Eusipco*, 1133–1136 (1988).

[8] H. Teager and S. Teager, "Evidence for nonlinear sound production mechanisms in the vocal tract," *Proc. NATO ASI on Speech production and Speech Modeling*, **II**, 241–261 (1989).

[9] J.H. Lee, H.Y. Jung, T.W. Lee, S.Y. Lee, "Speech feature extraction using Independent Component Analysis," *Proc. IEEE ICASSP*, **3**, 341–348 (1988).

[10] A. C. Lindgren, M. T. Johnson and R. J. Povinelli, "Speech Recognition using Reconstructed Phase Space Features," *Proc. IEEE Intl. Conf. on Neural Networks*, **1**, 61–63 (2003).

[11] A. C. Lindgren, M. T. Johnson and R. J. Povinelli, "Third-Order Moments of Filteref Speech Signals For Robust Speech Recognition," *Proc. of ISCA Tutorial and Research Workshop on Non-Linear Speech Processing (NOLISP)*, (2005).

[12] O. Farooq and S. Datta, "Phoneme Recognition using wavelet based features," *Information Sciences*, **150**, 5–15, March (2003).

[13] D. Dimitriadis and P. Maragos, "Continous-time models for AM-FM and their application to speech recognition," *Proc. of ISCA Tutorial and Research Workshop on Non-Linear Speech Processing (NOLISP)*, (2003).

[14] N. Malayath and H. Hermansky, "Data-driven spectral basis functions for automatic speech recognition," *Speech Communication*, **40**, 449–466 (2003).

[15] W. B. Kleijn, "Signal Processing Representations of Speech," *IEICE Trans. Inf. and Syst.*, **E86-D**, 3, 359–376, March (2003).

[16] E. Rank and G. Kubin, "Nonlinear synthesis of vowels in the LP residual domain with a regularized RBF network," *Proc. IWANN*, **2085**, II, 746–753 (2001).

[17] M. Faundez-Zanuy, "Non-linear speech coding with MLP, RBF and Elman based prediction," *Proc. IWANN*, **II**, 671–678 (2003).

[18] A. Hussain, "Locally-Recurrent Neural Networks for Real Time Adaptative Nonlinear Prediction of Non-Stationary Signals," *Control and Intelligent Systems*, **28**, 64–71 (2000).

[19] M. Faundez and D. Rodriguez, "Speaker recognition by means of a combination of linear and nonlinear predictive models," *Proc. ICASSP'99*, (1999).

[20] G.-.J. Jang and T.-W. Lee and Y.-H. Oh, "Learning statistically efficient features for speaker recognition," *Neurocomputing*, **49**, 329–348 (2002).

[21] D. Petrovska-Delacrétaz, M. Abalo, A. El Hannani and G. Chollet, "Data-Driven speech segmentation for speaker verification and language identification," *Proc. of ISCA Tutorial and Research Workshop on Non-Linear Speech Processing (NOLISP)*, (2003).

[22] B. Gold and N. Nelson, "Speech and Audio Signal Processing : Processing and Perception of Speech and Music," *John Wiley and Sons, INC*, (2000).

[23] B. Gas, J.L. Zarader, C. Chavy and M. Chetouani, "Discriminant neural predictive coding applied to phoneme recognition," *Neurocomputing*, **56**, 141–166 (2004).

[24] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, **2**, (1989).

[25] M. Birgmeier, "Nonlinear prediction of speech signals using radial basis function network," *EUSIPCO*,(1996).

[26] J. Thyssen, H. Nielsen and S.D. Hansen, "Non-linearities short-term prediction in speech coding," *Proc. ICASSP*,**1**, 185–188 (1994).

[27] M. Vandana, "M-ary predictive coding," *Proc. of ISCA Tutorial and Research Workshop on Non-Linear Speech Processing (NOLISP)*,(2003).

[28] A. Kumar and A. Gersho, "LD-CELP speech coding with nonlinear prediction," *IEEE Signal Processing Letters*,**4**, 89–91, April (1997).

[29] S. Wang and E. Paksoy and A. Gersho, "Performance of nonlinear prediction of speech," *ICSLP*,**1**, 29–32 (1990).

[30] M. Chetouani, B. Gas and J.L. Zarader, "Maximization of the Modelisation Error Ratio for Neural Predictive Coding," *Proc. of ISCA Tutorial and Research Workshop on Non-Linear Speech Processing (NOLISP)* (2003).

[31] S. Katagiri, "Handbook of Neural Networks for Speech Processing," *Artech House eds*,**1** (2000).

[32] A. Biem, "Optimizing Features and Models Using the Minimum Classification Error Criterion," *ICASSP*,**1**, 916–919 (2003).

[33] A. de la Torre, Antonio Peinado, Antonio J. Rubio and José C. Segura and C. Benitéz, "Discriminative feature weighting for HMM-based continous speech recognizers," *Speech Communication*,**38**, 267–286 (2002).

[34] F. Valente and C. Wellekens, "Minimum classification error / Eigenvoices training for speaker identification," *ICASSP*,**2**, 213–216 (2003).

[35] C.-L. Liu and M. Nakagawa, "Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition," *Pattern Recognition*,**34**, 601–615 (2001).

[36] E. McDermott, "Discriminative Training for Speech Recognition," *Waseda University (Japan)* (1997).

[37] X. Driancourt, "Optimisation par descente de gradient stochastique de systèmes modulaires combinant réseaux de neurones et programmation dynamique," *Université Paris XI Orsay* (1994).

[38] M. Chetouani, M. Faundez-Zanuy, B. Gas and J.L. Zarader, "A new nonlinear speaker parameterization algorithm for speaker identification," *Proc. of ISCA Tutorial and Research Workshop on Speaker and Language Recognition (Odyssey'04)*, 309–314 (2004).

[39] C. Jankowski and A. Kalyanswamy and S. Basson and J. Spitz, "NTIMIT: A Phonetically Balanced, Continous Speech, Telephone Bandwidth Speech Database," *ICASSP*, **1**, 109–112 (1990).

[40] P. J. Moreno and R. M. Stern, "Sources of degradation of speech recognition in the telephone network," *ICASSP*,**1**, 109–112 (1994).

[41] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang, "Phoneme recognition using time delay neural networks," *IEEE Trans. Acoustics, Speech and Signal Processsing*,**37**, 3, 328–339 (1994).