Codage discriminant appliqué à la reconnaissance de phonèmes

Bruno Gas, Jean-Luc Zarader, Cyril Chavy

LIS, Université Pierre et Marie Curie
4 place Jussieu, BP 164,75252 cedex 5, France
gas@ccr.jussieu.fr, zarader@ccr.jussieu.fr
chavy@ccr.jussieu.fr

Résumé – Nous proposons dans cet article une nouvelle méthode de codage appliquée à la reconnaissance de phonèmes. Le modèle en question est une extension au domaine non linéaire des méthodes de codage adaptatives habituellement utilisées en reconnaissance de la parole. Il est basé sur l'utilisation d'un réseau de neurones *perceptron multicouches* en prédiction. Nous montrons qu'il est possible d'introduire des informations de classe d'appartenance des signaux dès l'étape de codage, ce qui permet d'améliorer significativement les résultats en reconnaissance. Afin d'évaluer les performances du codeur *NPC* (Codeur Predictif Neuronal), nous présentons une étude expérimentale à partir de phonèmes issus de la base Darpa-Ntimit. Les simulations présentées mettent en évidence une amélioration des taux de classification relativement aux codages classiques.

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 adaptative coding systems used in speech processing. We use for this purpose predictive connectionnists methods. We show that it is possible to take into account class membership information of the phonemes from the stage of coding. In order to evaluate the *NPC* coder (Neural Predictive Coding), a study of Darpa-Ntimit phonemes recognition is done. Presented simulations put in obviousness an improvement of the classification, relatively to currently used coding methods.

1 Introduction

Depuis quelques années, le codage de la parole connaît un regain d'intérêt au sein de la communauté scientifique. En effet, les applications actuelles en parole donnent de bons résultats mais dans des environnements limités, comme par exemple en mode mono-locuteur, en milieu non bruité ou encore en parole non spontanée.

Dans la chaîne de traitement, le codage occupe une place fondamentale. Il effectue l'extraction des traits utilisés pour la reconnaissance des formes. Actuellement, on trouve deux grandes familles de codage qui sont les codages temporels (LPC, LPCC, LAR, etc.) et les codages paramétriques ou fréquentiels (Bancs de Filtres, Cepstre, MFCC, etc.). la représentation cepstrale utilisant l'échelle de Mel (codage MFCC) est la méthode de codage la plus employée [5] parce que la plus robuste. Deux points fondamentaux nous intéressent particulièrement dans cette étude. Ce sont la modélisation non linéaire du signal de parole d'une part et l'incorporation de connaissances de plus haut niveau dès l'étape de codage d'autre part.

1.1 Modélisation non linéaire

Les modèles de traîtement prédédents ont pour point commun d'être linéaires mais la famille des codages temporel s'est vue récemment étendue au domaine non linéaire [7], [8]. Comme l'ont montré plusieurs auteurs (Lapedes *et al.* [6], les réseaux de neurones formels se prettent bien à la modélisation de filtres adaptatifs non linéaires. Thyssen et al. ont proposé deux modèles. L'un est basé sur un filtre du second ordre de Volterra et l'autre sur un réseau TDNN (Time Delay Neural Network) utilisé en prédiction. En comparant les différents gains de prédiction obtenus, les auteurs montrent clairement l'intérêt apporté par les réseaux de neurones. Utiliser un réseau neuronal pour le codage consiste à exploiter les poids synaptiques calculés pour minimiser l'erreur de prédiction. Ces poids sont alors représentatifs du segment de parole prédit et constituent le vecteur de code. Un inconvénient posé par ce type d'approche est l'explosion du nombre de paramètres lorsque l'on augmente le nombre d'entrées et/ou le nombre de cellules cachées. Le modèle que nous proposons apporte une réponse à ce problème [3]. Il permet de séparer l'ensemble des poids en deux catégories : les poids codant des informations *discriminantes* et les poids codant des informations *non discriminantes*. Le vecteur de code regroupant uniquement les poids de la première catégorie, et sa dimension étant arbitraire, le modèle permet de définir aisément une dimension du code appropriée à l'application.

1.2 Extraction de caractéristiques discriminantes

Dans le cadre d'une application en reconnaissance de phonèmes, l'information discriminante regroupera les caractéristiques du signal permettant de distinguer deux phonèmes différents. Cela nous amène naturellement au deuxième volet de cet article : l'apport d'informations de plus haut niveau, en l'occurence des informations de catégorisation des signaux en classes.

L'objectif d'un codage adapté à la classification est de rehausser les caractéristiques séparant des signaux appartenant à des classes différentes tout en diminuant la contri-

bution de leurs caractéristiques communes. Réciproquement, il est d'augmenter les caractéristiques communes des signaux appartenant à une même classe tout en diminuant l'influence des caractéristiques qui les séparent. De telles méthodes de codage perdent en généralité ce qu'elles apportent en étant adaptées à la tâche de classification. On peut les inclure au sein d'une nouvelle et troisième famille : la famille des codages DFE (Discriminant Feature Extraction) introduite par Juang et Katagiri [4]. Ces travaux ont permis à Biem et Katagiri [1] d'élaborer des transformations du cepstre et du spectre de puissance afin d'obtenir des codages rehaussant les caractéristiques discriminantes du signal. Ces approches conduisent souvent à considérer le codeur et le classifieur comme un seul et même système. De la Torre et al. [2] ont proposé des variantes permettant de séparer le codeur du classifieur. Le modèle que nous proposons s'inscrit dans cette dernière catégorie. Afin de mettre en évidence ses capacités discriminantes, nous en proposons deux versions. La première permet le rehaussement des caractéristiques discriminantes des signaux de parole sans considérations de classe. La deuxième prend en compte la catégorisation de ces signaux en classes phonétiques. D'autres types de classifications pourraient être envisagées comme par exemple l'identification du locuteur ou encore de la langue.

2 Le Codeur NPC *(Neural Predictive Coding)*

Le Codeur Predictif Neuronal est une extension du codage LPC (Linear Predictive Coding) à la modélisation de signaux non linéaires. Etant donnée une séquence d'échantillons $\{y_{k-i,i=1,\dots n}\}$ extraite d'un phonème ϕ quelconque, le réseau effectue la prédiction de l'échantillon suivant y_k^ϕ en fonction des n précédents. Soit F de $\Re^n \rightarrow \Re$ la fonction réalisée par le réseau. La prédiction $\hat{y}_k^\phi$ s'écrit :

$$\hat{y}_k^\phi = F([y_{k-1}^\phi, y_{k-2}^\phi, \dots, y_{k-n}^\phi]^\top)$$

Nous désignerons par $\mathbf{x}_k^\phi$ le vecteur des échantillons précédents, soit :

$$\hat{y}_k^\phi = F(\mathbf{x}_k^\phi)$$

Soit $\Omega = [\omega_{ij}]$ le vecteur des poids du réseau. Ces poids sont calculés de sorte à minimiser l'erreur de prédiction $\epsilon_k = y_k - \hat{y}_k$ pour toutes les séquences $\mathbf{x}_k^\phi$ d'échantillons appartenant au phonème ϕ. L'erreur quadratique de prédiction s'écrit :

$$\mathcal{L}^\phi = \sum_k (y_k^\phi - F_\Omega(\mathbf{x}_k^\phi))^2$$

Après minimisation de $\mathcal{L}^\phi$ par l'algorithme de rétropropagation du gradient, F_Ω constitue une *modélisation* NLAR (Non Linear Auto-Regressive) du phonème ϕ et Ω peut être considéré comme *caractéristique* de ce phonème. L'inconvénient d'une telle approche est, comme nous l'avons dit, le très grand nombre de paramètres générés.

Le codeur NPC permet de limiter arbitrairement ce nombre en spécialisant les connexions. Pour cela, nous décidons que les poids liant la fenêtre de prédiction à la première couche codent des informations *non discriminantes*, c'est à dire communes à tous les phonèmes, tandis que les poids liant la couche cachée à la couche de sortie (une cellule de prédiction) codent les informations *discriminantes*, c'est à dire propres à chaque phonème. Le vecteur de caractéristiques est donc constitué uniquement de ces derniers.

Sur le plan formel, cela nous conduit à définir la fonction F_Ω réalisée par le réseau comme la composition de deux fonctions G_Ω et $H_{\mathbf{a}^\phi}$, l'une associée à la première couche (Ω représente maintenant les poids de la première couche), et l'autre à la deuxième couche ($\mathbf{a}^\phi$ est le vecteur des caractéristiques du phonème ϕ) :

$$F_\Omega = H_{\mathbf{a}^\phi} \circ G_\Omega \quad \text{avec} \quad \hat{y}_k^\phi = H_{\mathbf{a}^\phi}(\mathbf{z}_k^\phi) \quad et \quad \mathbf{z}_k^\phi = G_\Omega(\mathbf{x}_k^\phi)$$

La répartition des informations discriminantes et non discriminantes sur les poids du réseau s'obtient en définissant deux fonctions de coût pour les deux ensembles de poids en question. La première, $\mathcal{L}^\phi(\mathbf{a}^\phi)$, est calculée à partir de l'erreur de prédiction moyennée sur les séquences composant un même phonème ϕ. La deuxième, $\mathcal{L}(\Omega, \mathbf{a}^\phi, ...)$, est calculée sur l'ensemble des séquences composant tous les phonèmes de la base.

Le fonctionnement du codeur s'effectue en deux phases : 1) le calcul des poids Ω ou *phase de paramétrisation du codeur*, 2) le calcul des codes ou *phase de codage* proprement dite.

2.1 Paramétrisation du codeur

La phase de paramétrisation est l'estimation de la fonction G_Ω. Elle s'obtient par minimisation du critère quadratique suivant :

$$\mathcal{L} = \frac{1}{|\{\phi\}|} \sum_\phi \mathcal{L}^\phi(\mathbf{a}^\phi) = \frac{1}{|\{\phi\}|} \sum_\phi \sum_k (y_k^\phi - H_{\mathbf{a}^\phi} \circ G_\Omega(\mathbf{x}_k^\phi))^2$$

Lors de cette phase, les paramètres $\mathbf{a}^\phi$ doivent également être estimés. En effet, le choix d'une valeur arbitraire et unique pour tous les phonèmes $\mathbf{a}^\phi = \mathbf{a}^0, \forall\phi$ conduirait à reporter l'information discriminante uniquement sur l'erreur de prédiction et ainsi à répartir l'information non discriminante sur l'ensemble des paramètres du réseau et non pas seulement sur les poids Ω. L'estimation des coefficients $\mathbf{a}^\phi$ s'obtient par minimisation de l'erreur de prédiction sur les séquences composant le phonème :

$$\mathcal{L}^\phi(\mathbf{a}^\phi) = \sum_k (y_k^\phi - H_{\mathbf{a}^\phi} \circ G_\Omega(\mathbf{x}_k^\phi))^2$$

2.2 Codage

La phase de codage est la phase de génération des codes. Pour un phonème quelconque ϕ et l'ensemble des séquences $\mathbf{x}_k^\phi$ qui le composent, la première couche du réseau est utilisée comme un opérateur de changement de représentation : $\mathbf{z}_k^\phi = G_\Omega(\mathbf{x}_k^\phi)$. Le vecteur des poids Ω étant celui obtenu par la phase de paramétrisation. L'estimation du vecteur caractéristique $\mathbf{a}^\phi$ s'obtient par minimisation de l'erreur de prédiction sur l'ensemble des vecteurs $\mathbf{z}_k^\phi$ calculés sur les séquences $\mathbf{x}_k^\phi$:

$$\mathcal{L}^\phi = \sum_k (y_k^\phi - H_{\mathbf{a}^\phi}(\mathbf{z}_k^\phi))^2$$

2.3 Codage discriminant

Une modification adéquate des fonctions de coût précédentes permet d'introduire très simplement des informations de classe d'appartenance lors de la paramétrisation du codeur. C'est ce que nous proposons d'établir dans ce paragraphe pour la classification de phonèmes.

Reprenons la phase de paramétrisation. Nous ne considérons plus un vecteur de code par phonème mais un vecteur de code par classe de phonèmes. Tout se passe comme si nous devions coder les classes de phonème plutôt que les phonèmes eux-mêmes. Soient $C_1, \ldots, C_M$ les M classes d'appartenance des phonèmes et $\mathbf{a}^{C_i}$ les vecteurs de code associés. Ces M vecteurs constituent les M jeux de poids de la deuxième couche du réseau qu'il convient d'estimer. Le coût quadratique à minimiser devient naturellement (pour les poids de la deuxième couche) :

$$\mathcal{L}^{C_i}(\mathbf{a}^{C_i}) = \sum_{\phi \in C_i} \sum_{k} (y_k^{\phi} - H_{\mathbf{a}^{C_i}} \circ G_{\Omega}(\mathbf{x}_k^{\phi}))$$

et pour les paramètres de la première couche :

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}^{C_i}$$

L'étape de codage reste inchangée : on reprend à nouveau le codage des phonèmes les uns après les autres en minimisant la fonction de coût définie au paragraphe 2.2.

3 Application au codage de phonèmes

Nous proposons une application de notre modèle, dans ses deux versions, au codage de la parole pour la classification de phonèmes. Dans ce qui suit, nous appellerons NPC-I le codeur sans discrimination de classes et NPC-II le codeur avec discrimination.

Les bases de données : Nous avons extrait 6000 séquences de 256 échantillons de parole appartenant à 6 classes différentes de phonèmes (base Darpa N-Timit de parole par le téléphone échantillonnée à $16KHz$). Après filtrage puis sous échantillonnage à $8KHz$, nous avons séparé cette base en deux sous-bases, l'une d'apprentissage et l'autre de test, afin d'entraîner puis de tester en généralisation le classifieur. Les phonèmes étant de longueurs différentes, le tableau 1 donne le nombre exact de phonèmes entiers présents dans la base. Nous avons également appliqué une préaccentuation des signaux suivie d'une normalisation à $[-0.8 + 0.8]$ calculée au niveau du phonème entier.

/aa/	/ah/	/ih/	/iy/	/s/	/z/
70	129	175	111	75	109

TAB. 1: Répartition des phonèmes (~ 600) dans la base

Paramétrisation : Le codeur que nous présentons est un modèle $20 \times 12 \times 1$, c'est à dire qu'il comporte 20 entrées (la fenêtre de prédiction est donc de largeur 20), 12 cellules cachées (le vecteur de code généré comporte 12 paramètres) et une cellule de prédiction (prédiction à un pas de l'échantillon suivant). La figure 1 montre le comportement de l'erreur quadratique durant près de 4000 itérations d'apprentissage pour les deux versions du codeur, NPC-I et NPC-II. On note l'«effort» supplémentaire demandé au NPC-II (minimisation de l'erreur de prédiction sur un jeu de code par classe plutôt que par séquences de phonème). Ceci se traduit par une erreur résiduelle plus élevée que pour NPC-I.

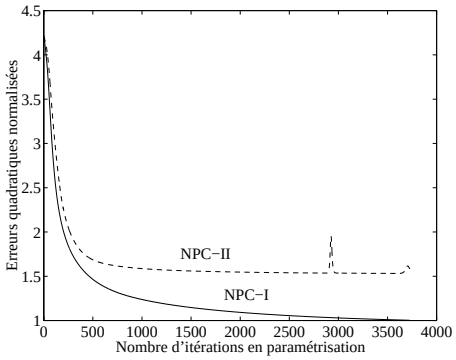

FIG. 1: Erreur quadratique pour la paramétrisation de NPC-I et NPC-II

Codage : Sur les figures 2 et 3 sont reportées les mêmes erreurs de prédiction obtenues lors du codage des séquences de phonèmes. Il s'agit ici de moyennes effectuées sur l'ensemble des phonèmes codés. La convergence est plus rapide qu'en paramétrisation et ceci est principalement dû à une moindre complexité du codage (la première couche de poids est figée et l'on minimise l'erreur de prédiction sur la séquence phonétique à coder et non sur l'ensemble de la base). Ajoutons également que l'algorithme utilisé en phase de codage est de type *gradient stochastique*, contrairement à la paramétrisation.

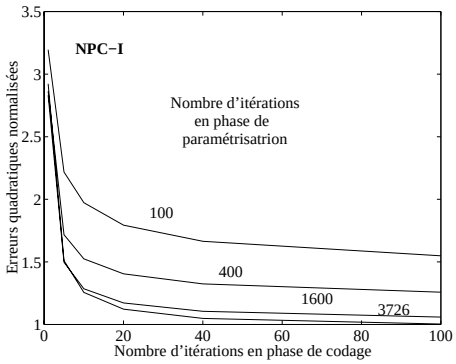

FIG. 2: Moyenne des erreurs quadratiques en codage avec NPC-I

Classification : Nous avons utilisé pour la classification des séquences phonétiques un perceptron à une couche cachée de structure $(12 \times 10 \times 6)$ et l'algorithme de rétropropagation (gradient total). Nous avons effectué un ensemble de simulations dans le but d'observer le com-

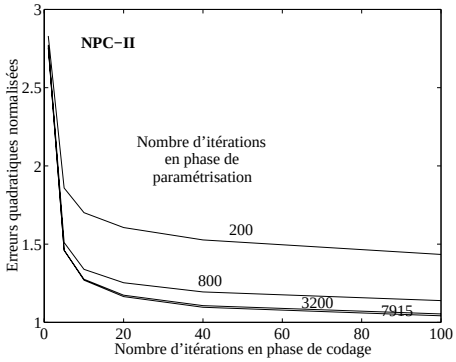

FIG. 3: Moyenne des erreurs quadratiques en codage avec NPC-II

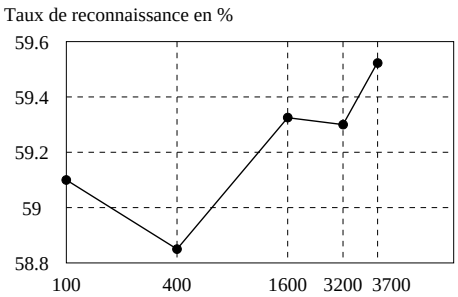

FIG. 4: Scores en classification (généralisation), fonction du nombre d'itérations effectuées en paramétrisation

portement du codeur NPC-II en apprentissage. La figure 4 montre les scores obtenus en fonction du nombre d'itérations effectuées en paramétrisation tandis que la figure 5 représente ceux obtenus en fonction du nombre d'itérations pour le codage des séquences. Il s'agit de scores en *généralisation*. Contrairement à la paramétrisation, le codage présente un phénomène de sur-apprentissage d'où la nécessité de mettre au point un critère d'arrêt.

Tests comparatifs Nous plaçant dans les mêmes conditions expérimentales, nous avons réalisé un codage des séquences de phonèmes par les méthodes les plus couramment utilisées, c'est à dire LPC et MFCC. La figure 6 montre des résultats obtenus. Il s'agit de moyennes effectuées sur une dizaine de simulations. On remarque clairement l'amélioration apportée par les deux codages NPC-I et NPC-II, et notament la supérioté de NPC-II sur NPC-I.

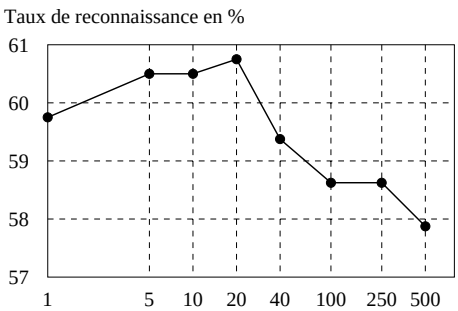

FIG. 5: Scores en classification (généralisation), fonction du nombre d'itérations effectuées lors du codage

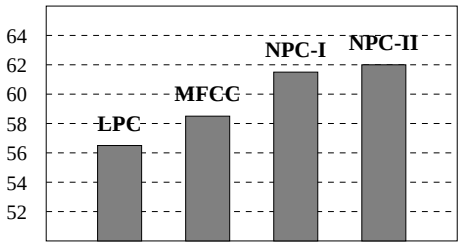

FIG. 6: Scores en généralisation obtenus avec les différents codeurs

4 Conclusion

L'amélioration des performances a été remarquée sur toutes les expériences menées. Elle montre que l'information permettant de discriminer les phonèmes a bien été reportée sur les poids de la deuxième couche utilisés pour le codage. Ces résultats justifient, nous semble-t-il, la nécessité de mettre au point de nouvelles méthodes de codage adaptées à l'application et prenant en compte des informations issues d'étapes de traitement de plus haut niveau, ici de catégorisation des signaux en classes.

Références

[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. 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.

[3] B. Gas, J. L. zarader, P. Sellem, and J.C. Didiot. Speech coding by limited weights neural network. In *IEEE International Conference on Systems Man and Cybernetics*, pages 4081–4085, 1997.

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

[5] B. H. Juang, L. R. Rabiner, and J. G. Wilpon. On the use of bandpass liftering in speech recognition. *IEEE Transactions on Acoustic and Speech Signal Processing*, 35(7):947–954, 1987.

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

[7] 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.

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