MODULAR NEURAL PREDICTIVE CODING FOR DISCRIMINATIVE FEATURE 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 measures 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.

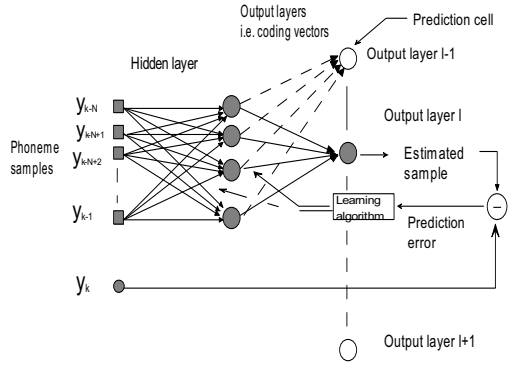

Fig. 1. Architecture of the Neural Predictive Coding model

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 speech 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.