DISCRIMINATIVE TRAINING FOR NEURAL PREDICTIVE CODING
APPLIED TO SPEECH FEATURES EXTRACTION

M. CHETOUANI, B. GAS, J.L. ZARADER, C. CHAVY
Laboratoire des Instruments et Systèmes d’Île de France (LISIF)
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 chavy@ccr.jussieu.fr

ABSTRACT
In this paper, we present a predictive neural network called Neural Predictive Coding (NPC). This model is used for non linear discriminant features extraction (DFE) applied to phoneme recognition. We validate the nonlinear prediction improvement of the NPC model. We also, present a new extension of the NPC model : NPC-3. In order to evaluate the performances of the NPC-3 model, we carried out a study of Darpa-Timit phonemes (in particular /b/, /d/, /g/ and /p/, /t/, /q/ phonemes) recognition. Comparisons with traditionnal coding methods are presented: they put in obviousness an improvement of the classification. We also show how an adaptative constraint allows improvements on recognition task.

1 INTRODUCTION

Speech production is commonly modelised as a linear predictive filter. In fact, the technique of linear prediction has been used in many speech processing systems : transmission, recognition, etc. … For recognition task, the modellisation of the vocal tract by a linear filter allows a feature extraction from the signals : the Linear Predictive Coding (LPC). But, it is known that the underlying speech production model is nonlinear [1,2], so one can get an improvement of the recognition rate by extracting nonlinearities. The implementation of nonlinear predictors is essentially based on two techniques : Volterra filters [3] and neural networks [4,5]. The major advantage of Volterra filters is that, like in linear predictors, the least mean square solution for the filter coefficients can be expressed analytically but the main drawback lie in the fact that the number of coefficients grows fast with the prediction window. Predictive neural networks have already been successfully applied to speech [6]. The neural networks weights can estimate the vocal tract model, as in linear predictive coding. A large number of speech representations have been proposed ; in the temporal domain (LPC, LPCC, LAR, etc. …), and also in the frequential domain (FFT filter banks, Cepstre, MFCC, etc. …). The most often used is the cepstral representation using the Mel scale (MFCC) because of its robustness. In the recent years works have been made to improve the discriminant features extraction from the speech signals. Most of them are frequency-based model [7, 8]. In the temporal domain, predictive networks are trained independently of each other. As a result, there is no explicit discrimination between the models. We recently proposed a new neural predictive coding (NPC)[9] designed to model short-term nonlinearities. It has the major advantage to allow an arbitrary limited number of coding coefficients. The NPC model has already been extended by incorporating class informations to improve the next pattern recognition stage, the NPC-2 model [10,11]. Moreover, we proposed, in this paper, the new NPC-3 model. It is a discriminative training based on a mechanism which discourages phoneme models from resembling each other.

The paper is organized as follows : In section 2, we describe the NPC model, then we validate it on nonlinear prediction task. We present in section 4 the NPC-2 model. We then present an extension of the NPC-2 model for discriminative features extraction (DFE) : the NPC-3 model. We also compare the NPC-3 model with traditionnal coding methods on phoneme recognition task. In section 7, we look at an adaptative method which to control the influence between prediction and discrimination.

2 THE NPC MODEL

The Neural Predictive Coding model is an extension of the LPC traditional coding (Linear Predictive Coding) to the modelling of non linear speech signals. It is based on a two layers perceptrons which is composed of one hidden layer followed by an output cell; the prediction cell. For such a task, the speech signal is divided into fixed length frames and the current speech sample is predicted from a combination of finite past samples. L being the
length of the prediction window, one has:

\[ \hat{y}_k = \hat{F}(y_k) \text{ with } y_k = \begin{bmatrix} y_{k-1}, y_{k-2}, \ldots, y_{k-L} \end{bmatrix}^T \]  

(1)

\( \hat{F} \) is a non-linear function which is composed of two functions \( G_w \) (corresponding to the hidden layer) and \( H_a \) (corresponding to the output layer):

\[ \hat{F}, a = H_a \circ G_w \]  

(2)

with \( \hat{y}_k = H_a(z_k) \) and \( z_k = G_w(y_k) \).

\( w \) denotes the hidden layer weights vector and \( a \) the output layer weights vector. All network weights are usually computed by minimizing a prediction cost function as the quadratic error criterion:

\[ L = \sum_k (y_k - \hat{y}_k)^2 \]  

(3)

Where \( k \) is the index of the samples.

For instance, over all the samples composing a signal, one can obtain after learning a function \( F \) which is a nonlinear auto-regressive model (NLAR) of the signal. One problem that occurs with this approach is that it generates a great number of parameters. The aim is to limit this number, and the key idea of the NPC coding is to allow an arbitrary number of coding coefficients by creating a second layer for each phoneme, the first layer remaining the same for all phonemes. The cost function previously defined becomes:

\[ L = \sum_{i,k,l} (y_{i,k} - F_{w,a}(y_{i,k})) \delta_{i-1} \]  

(4)

where \( F_{w,a} \) is one of the \( M \) functions corresponding to \( a_i \) output layers weights. \( \delta \) is the Kronecker symbol which associates the phoneme \( i \) to the output layer \( l \).

Output layers weights are proper to each phoneme, and they are the coding coefficient (see figure 1).

The learning process needs to be broken down in the following two phases:

- The parameters adjustment 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 no longer used while the hidden layer weights become the encoder parameters.
- The coding phase. The network works as a two layers perceptron composed of the hidden layer previously computed and of one output cell. These weights are the only ones requiring updating. They are the NPC coding coefficients.

3 NONLINEAR PREDICTION

To evaluate the NPC performances we tested it on a phoneme prediction task. We extracted voiced and unvoiced phonemes from the Darpa-Timit [12]; /aa/ voiced phoneme and /p/ unvoiced phoneme. The non-linearities provided by NPC-model allow a better fitting of speech signals than linear models like LPC. The performances of extraction of non-linearities are measured by two measures R-measure and Q-measure [13]. R-measure is defined as follow:

\[ R = \frac{\text{Residual variance (non linear predictor)}}{\text{Residual variance (linear predictor)}} \]  

(5)

The ratio \( R \) measures the improvement of the non-linear predictor to capture non-linearities of the signal (\( R<1 \)). We extented the R-measure to speech processing, the linear prediction is known as the linear predictive coding (LPC), so the R-measure become:

\[ R = \frac{\text{Residual variance (NPC)}}{\text{Residual variance (LPC)}} \]  

(6)

The R ratio measures improvements of the nonlinear prediction with respect to the linear prediction. A R ratio inferior to one, signifies that the nonlinearities introduced by the nonlinear predictor are positive for the prediction. As we can see on table 1, the NPC model has extracted the non-linearities of the phonemes:

<table>
<thead>
<tr>
<th>Voiced phonemes</th>
<th>Unvoiced phonemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.13</td>
<td>0.236</td>
</tr>
</tbody>
</table>

Table 1 R measure for voiced and unvoiced phonemes

One can see on figures 2 and 3 the prediction of /aa/ and /p/ phonemes. Linear model can not capture the nonlinearities of the phoneme /p/. For unvoiced phonemes, the introduction of nonlinear features extraction allows an improvement of the prediction.
One can also measure the nonlinear features extraction, by investigating the distribution of the hidden units over the quadratic part of the Taylor expansion with respect to the linear part [13]:

\[ Q = \frac{\text{Quadratic contribution}}{\text{Linear contribution}} = \frac{f''(x)}{f'(x)} \]  

(7)

with \( f'(x) \) and \( f''(x) \) the first and second derivates of the activation with respect to the input \( x \). According to the Q measure, one can estimate the nonlinear features extraction by taking into account the signal input and the network characteristics. The sigmoid used is defined between \(-1;+1\) \], so the nonlinearities are present near the values domain: \(-1; -0.7 \cup [0.7; 1] \). We measured the Q distribution for /b/ voiced occlusive phoneme. One can see on figure 4, the Q distribution for /b/ phonemes. It shows that the NPC model extracts nonlinear features of the phonemes.

4 THE NPC-2 MODEL

The NPC-2 model [10, 11] is an extension of NPC model which allows class membership information to be taken into account during the parameters adjustment phase. This is done by limiting the set of the output layer weights (the coding coefficients) to one coefficient vector by phoneme class instead of one by phonemes. The cost function previously defined for NPC model becomes:

\[ L = \sum \sum \delta C_i \]  

(8)

\( C_i \) is the class membership of phoneme \( i \) among a set of \( M \) possible classes. \( F_{w,j} \) is one of the \( M \) functions corresponding to the \( a_j \) output layer weights. \( \delta \) is the Kronecker symbol which associates the class \( C_i \) to the output layer \( l \).

5 NPC-3 MODEL

To guarantee a class-discriminant features extraction, one can add constraints to the weights evolution during the learning process. One possible mechanism is to introduce explicit discrimination between models. After the parameters adjustment phase of a phoneme \( i \), we obtain the \( F_{w,j} \) NPC model. To get a discrimination, one can estimate the prediction error of the phoneme \( i \) using another model which is the \( F_{w,a_j} \) model. As a result, the prediction error is: \( L_j = \sum (y_{i,k} - F_{w,a_j}(y_{i,k}))^2 \).
The mechanism which discourage the models from resembling each other is obtained by the maximisation of the modelisation error ratio (MER) [11], $\Gamma_{NPC}$:

$$\Gamma_{NPC} = \frac{Q^d}{(M-1)Q^m}$$ \quad (9)

with $Q^d = \sum_{i=1}^{M} \sum_{j=1}^{M} L^i_j$ and $Q^m = \sum_{i=1}^{M} L^i$.

Where $Q^m$ is the NPC-2 model prediction cost function which has to be minimized, and the $Q^d$ discriminant cost function has to be maximized.

So it is possible to optimise the discrimination by applying the constraint which consist in the minimisation of the reverse modelling error ratio:

$$Q_{NPC3} = \frac{1}{\Gamma_{NPC}}$$ \quad (10)

The modification law of any a or w weights is proportional to the gradient of $Q_{NPC3}$:

$$\frac{\partial}{\partial a} \left( \frac{1}{\Gamma_{NPC}} \right) = M -1 \left( \frac{\partial Q^m}{\partial a} - \frac{1}{\Gamma} \frac{\partial Q^d}{\partial a} \right)$$ \quad (11)

It is composed of two terms, the first term corresponding to the prediction error minimisation (The NPC-2 cost function) and the second to the discrimination measure maximisation.

To set an influence on the prediction / discrimination ratio, we used the following expression:

$$\frac{\partial}{\partial a} \left( \frac{1}{\Gamma_{NPC}} \right) = (M-1) \left( \alpha \frac{\partial Q^m}{\partial a} - (1-\alpha) \frac{\partial Q^d}{\partial a} \right)$$ \quad (12)

Where $\alpha$ represents the intensity of the constraint. One can note, that if $\alpha = 1$, we find the cost function of the NPC2-model.

\section{EXPERIMENTAL CONDITIONS}

To evaluate the NPC performances we tested it in a phoneme recognition task. We will describe in this part the experimental conditions.

\subsection{The database}

We built several phonemes bases each extracted from the Darpa-Timit [12] database. This database is composed of speakers speaking 10 different dialects of the United States. The first base groups four classes of voiced phonemes (vowels) very commonly used: /aa/, /ae/, /ey/, /ow/. This base is constituted of 500 examples per phoneme class. We also evaluated the NPC coding on two other bases: /b/, /d/, /g/ (voiced plosives) and /p/, /t/, /k/ (unvoiced plosives). They are particularly interesting, because they frequently appear in the English language and their identification is considered to be difficult. Those phonemes have been used by Lang and Waibel [14] to validate the Time Delay Neural Network (TDNN).

To select phonemes for each 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.
- Examples are chosen randomly among all speakers so as to model a multi speaker environment.

\subsection{Traditional coding methods}

Our aim is to test the efficiency of the speech features extraction of the NPC-3 encoder. The performance will be estimate by classification. We made comparisons between NPC coding and traditional coding methods; LPC coding (Linear Predictive Coding), MFCC coding (Mel Frequency Cepstrum Coding). This latter method reproduces the signal spectrum with a scale of frequencies based on the human ear scale; the Mel frequency scale. The number of coding coefficients is set to 12.

\subsection{Classification with MLP}

The classifier used to estimate performances of coding method is a basic MLP with 12 inputs (coding vectors dimension), 10 hidden neurons and as many outputs as there are phoneme classes. The learning rule is the gradient descent using error back propagation algorithm.

\subsection{NPC evaluation using MLP classifier}

In this paragraph, we present the results of classification of the different phoneme bases using the different coding methods; NPC-1, NPC-2 and NPC-3, and the traditional methods; LPC and MFCC. We measure the generalisation score.

On figure 5, one can see comparisons between recognition rates obtained by MLP classifier. Recognition rates have been obtained after 30000 learning iterations. The NPC coders give better results in generalisation: 63.5% for NPC-3, 62.95% for NPC-2, 61% for NPC and 58.25% for MFCC and 56.33% for LPC. Moreover, one can see the better performance of NPC-3.
One can note, on the figure 6, that the results for /b/, /d/, /g/ phonemes are consistent with the fact that they are voiced phonemes like vowels. Phonemes /p/, /t/, /q/ are unvoiced phonemes, so spectral methods like MFCC have better performances than predictive methods like LPC, NPC-1 and NPC-2. But, one can note that the discrimination introduced in the NPC-3 encoder, compensates the default of temporal encoders for unvoiced phonemes.

Unlike the MFCC or LPC coding methods, the underlying model nonlinearities present in speech signals are taken into account by the NPC-1, NPC-2 and NPC-3 models. Moreover, the discriminant optimisation of NPC-3 model, gives better results for unvoiced phonemes. A comparison between the NPC-2 model and NPC-3 model, shows the better discrimination of the NPC-3 model. In fact, as one can note on the figure 7, the NPC-3 MER is higher than the NPC2 MER. In addition, a study of the between-class covariance (see figure 8) shows that the NPC-3 between class covariance converges to a higher value than NPC-2.

7 Intensity of the constraint $\alpha$

According to the equation 11, the NPC-3 model optimisation introduce a constraint in order to set an influence on the prediction / discrimination ratio. One can see on figure 9, the effect of $\alpha$ on the recognition performance.
discriminant features extraction (DFE). Consequently, we showed a significant improvement of the recognition rate specially in the case of unvoiced phonemes. And, the analyse of the MER makes it possible to know the discriminant properties of the encoder. Thanks to the adaptative adjustment of the constraint, the discrimination is accentuated.

So our prospects are to optimise the discriminant algorithm to speech recognition task. And it is necessary to define a new criterion which will make it possible to stop the parameters adjustment phase when the discriminant properties are the most significant.

References: