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

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.

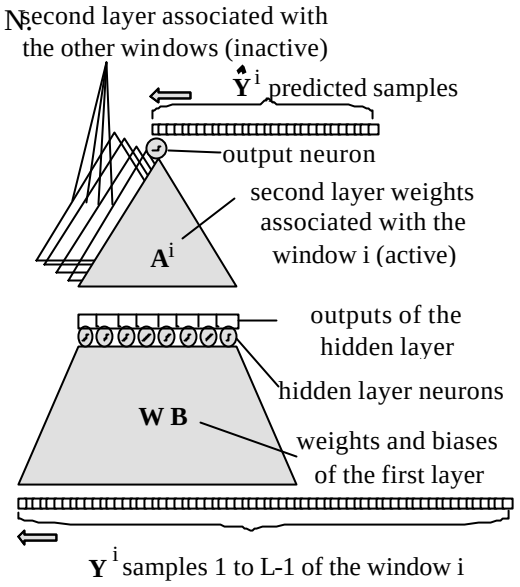
2.2 Coding phase

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.



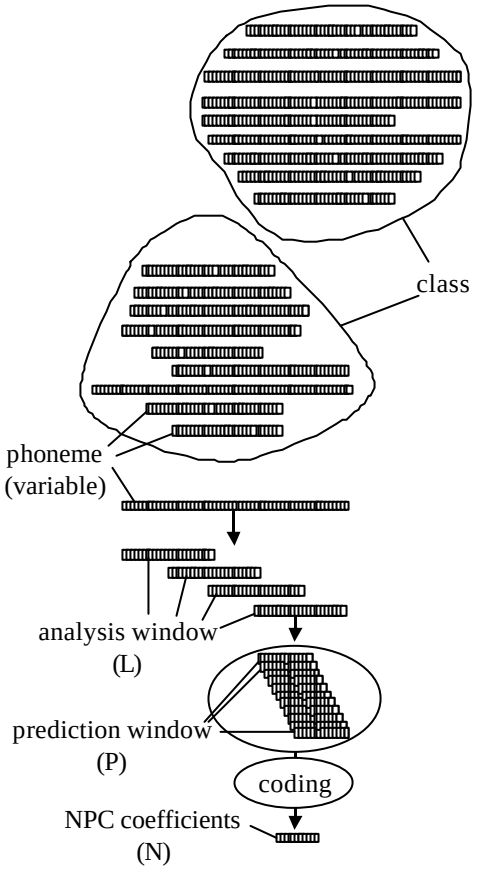


Fig. 1 : Architecture of the NPC Model

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.

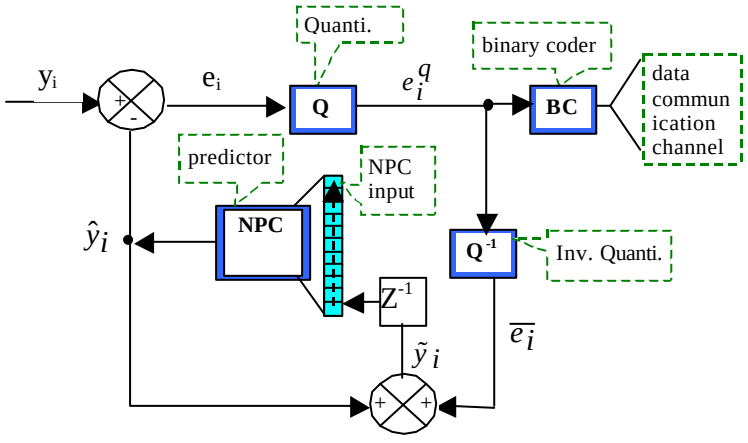


Fig. 3 : Compression block diagram

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.

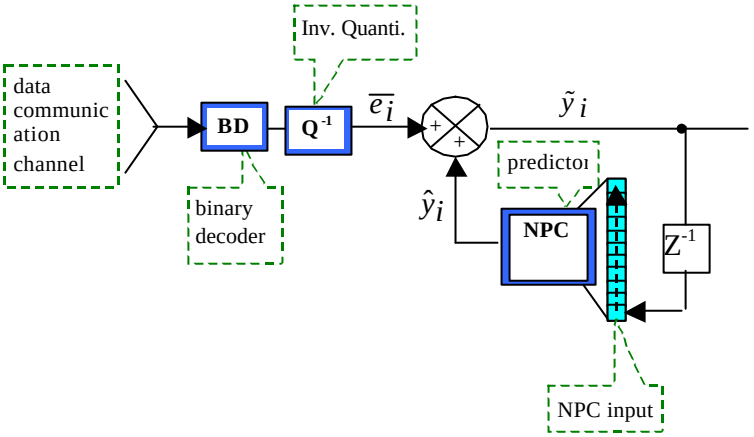


Fig. 4 : Decompression block diagram

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

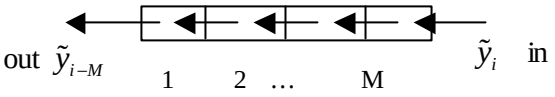
Let N be the number of samples of the speech signal and y_i the i^{th} sample. Let **A** be the second layer of weights and **I** the NPC input vector.
The speech signal compression algorithm is described below :

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

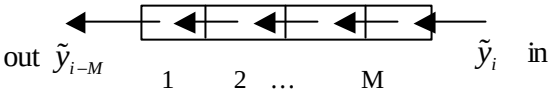
The speech signal decompression algorithm is described below :

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.

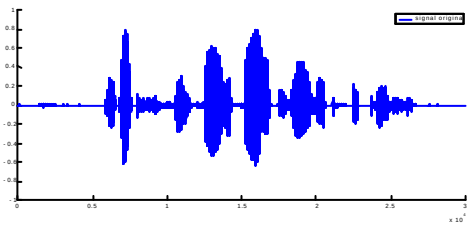


Fig. 5 : Original speech signal y_i

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

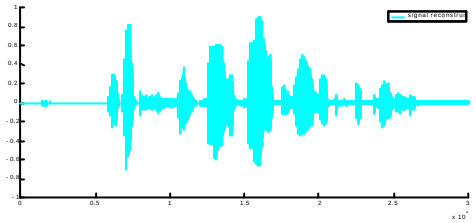


Fig. 6 : Decoded speech signal $\tilde{y}_i$

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 :

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

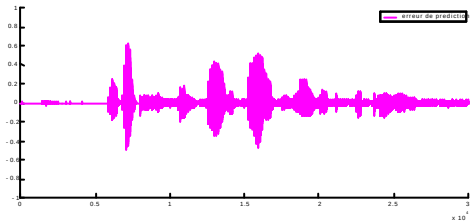


Fig. 7 : Prediction error e_i

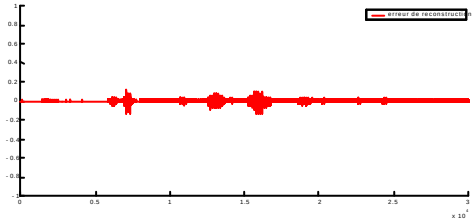


Fig. 8 : Compression error x_i

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.