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:

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

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 $\mathbf{x}_k = [y_{k-1}, y_{k-2}, \dots, y_{k-\lambda}]^\top$ 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 $\mathbf{a} = [a_1, \dots, a_N]^\top$ (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 :

$$l^* = \arg \min_{l=1,\dots,L} d_\Omega^{NPC}(l, m) = \arg \min_{l=1,\dots,L} \{\log \frac{Q_m(\mathbf{a}_l)}{Q_m(\mathbf{a}_m)}\} = \arg \min_{l=1,\dots,L} \{Q_m(\mathbf{a}_l)\} \tag{7}$$

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) :

$$Q_m(\mathbf{a}_{1,\dots,L}) = \sum_l^L \sum_{k(m)} (y_k - G_\Omega \circ H_{\mathbf{a}_l}(\mathbf{x}_k))^2 V^\sigma(l^*, l) \tag{8}$$

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 $\mathbf{a}_l$:

$$\begin{aligned} a_{il}(q) &= a_{il}(q-1) - \frac{\partial Q_m}{\partial a_{il}} \qquad (9) \\ &= a_{il}(q-1) + V^\sigma(l^*, l) \sum_{k(m)} (y_k - \phi(V_k))\dot{\phi}(V_k)\phi_i(\mathbf{x}_k) \qquad (10) \end{aligned}$$

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.

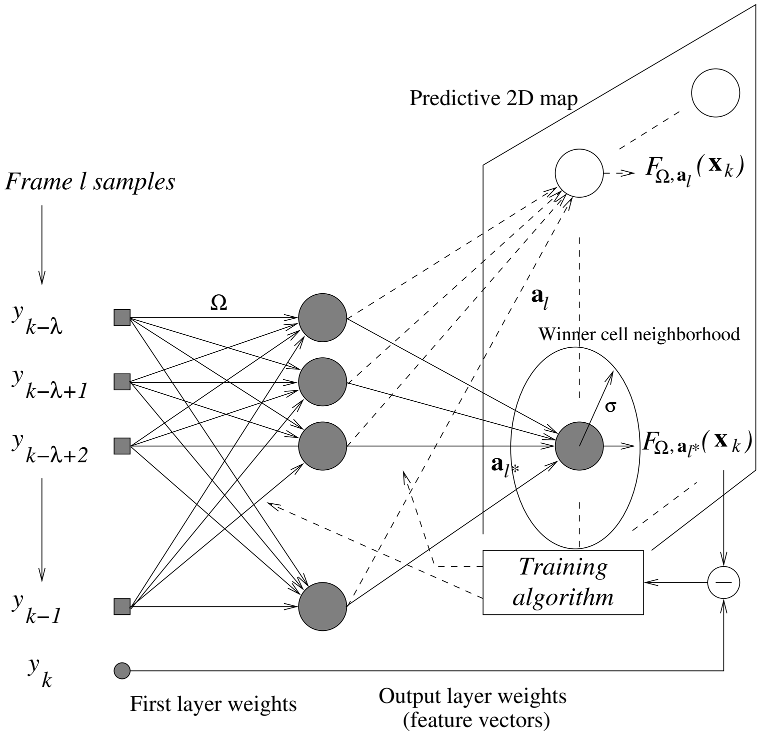

Figure 1: NPC-K coder.

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
d	d	d	d	d	b	g	g
d	d	d	d	d	b	g	g
d	d	d	d	g	b	g	g
g	d	d	d	d	d	g	g
d	d	d	b	d	d	g	g
d	b	d	b	b	b	b	b
d	b	b	b	g	g	g	b

q	q	q	q	q	q	q	t
q	q	q	q	q	q	p	t
q	q	p	t	t	t	t	t
q	p	p	p	t	t	t	t
p	q	p	p	t	t	t	t
p	q	t	t	t	t	t	t
p	q	q	t	t	t	t	t
t	t	t	t	t	t	t	t

ow	ow	aa	ae	ae	ae	ae	ey
ow	aa	aa	aa	ae	ae	ae	ey
ow	aa	aa	aa	ae	ae	ey	ey
ow	ae	aa	aa	ae	ae	ey	ey
aa	ae	ae	ae	ae	ae	ae	ey
aa	aa	ae	ae	ae	ae	ey	ey
ow	ae	ae	ae	ae	ae	ey	ey
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 nonlinear 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