Vowel Recognition using Concurrent Neural Networks

Angel Cațaron
Dept. of Electronics and Computers, TRANSILVANIA University of Brasov, Romania
e-mail: cataron@vega.unitbv.ro

***Abstract* – Vowel recognition represents the detection of a vowel in a speech stream and assigning it the correct class label. In this paper we use the *Concurrent Neural Networks (CNN)* for the task of romanian vowel recognition. This model is a *winner-takes-all* collection of individually trained neural networks. Each network of the system provide best results for one class of input partterns only. As basic components of such a system, we used the Kohonen Self-Organizing Map (SOM). We also performed similar tests training SOM as single neural network. We used a speech database with spoken words collected from different persons, males and females. They pronounced the romanian words representing the 10 digits, from zero to nine and each word was pronounced 10 times by each speaker. The experiments proved that our SOM-CNN neural model performed better than SOM for similar tasks, with an increase of more that 10% of correctly recognized vowels.**

***Index Terms* – Concurrent Neural Networks, Kohonen Self-Organizing Map, vowel recognition.**

I. INTRODUCTION

Recent work [2] proved that CNN increase speaker recognition accuracy comparing to the scores of the well known neural models like the Multi-Layer Perceptron (MLP), the Time Delay Neural Network (TDNN) or the Self-Organizing Map (SOM).

We will study in this paper the behaviour of the CNN with SOM as basic component in the problem of vowel recognition.

In section II we will briefly present the Concurrent Neural Networks recognition model. In section III we will describe the speech database that we used during the experiments. Section IV will depict the tests that we performed as well as the results that we obtained.

II. CONCURRENT NEURAL NETWORKS

Let us consider the set $X \subset \Re^p$ which is formed by M feature vectors from a p-dimensional Euclidian space, that is, $X=\{\mathbf{x}_1, \mathbf{x}_2 \ldots \mathbf{x}_M\}$, $\mathbf{x}_i \in \Re^p$, $1 \le i \le M$. CNN are a collection of neural networks. They are trained individually and the recognition decision is based on a *winner-takes-all* strategy.

We consider each of the feature vectors from the set X are *a priori* known to belong to one of the n classes, that is

$$X = X_1 \cup X_2 \cup X_n$$

and

$$X_1 \cap X_2 \cap X_n = \Phi,$$

where X_1, X_2 ... X_n are subsets of X and can be used as training pattern sets for each of the n neural networks (fig. 1).

The CNN global training technique is a supervised one, and for the individual networks can be used their own training algorithms.

In figure 1, n is the number of parallel performing neural networks, that is the total number of the pattern classes.

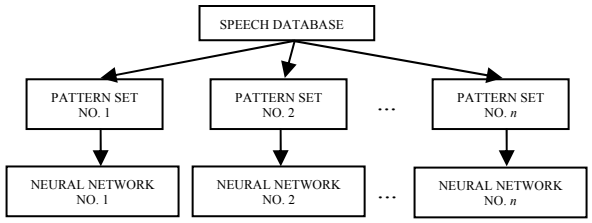

Fig. 1. The CNN model used in the training phase.

The database consists of the pre-processed speech signal, that is the set X. The pattern sets extracted from this database, that is the subsets X_1, X_2 ... X_n, are inputs of the n networks in the training phase.

In the recall phase, each network should be activated by the patterns from its corresponding class only. The CNN performs a selection of the outputs from the individual networks (fig. 2). The selection means finding the strongest response. The selected network is declared as *winner* and its index is the class index associated to the test pattern. This classifying method suppose that the number of classes is known before the training phase and for each class is available a sufficiently number of patterns.

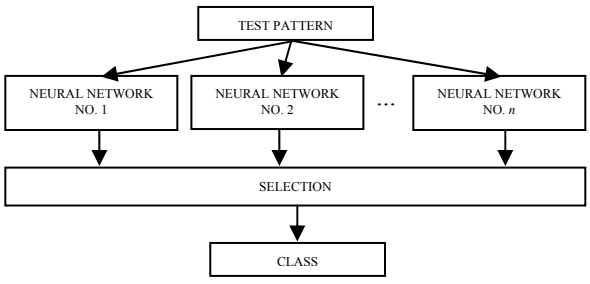

Fig. 2. The CNN recall model.

SOM use an unsupervised training algorithm. Each component network is trained with its dedicated pattern set. The classification decision is based on the minimum quantization error. The network which generate the minimum is selected as winner.

III. THE SPEECH DATABASE

Our approach of the problem of vowel recognition was based on the study of the similarities between the patterns obtained by cepstral analysis of the vocalized regions from the target words and a set of prototypes.

The cepstrum analysis realise a separation of the signal components, producing a linear combination of them. Therefore, the inferior part of the cepstrum is the contribution of the periodic excitation and the superior part corresponds to the transfer function of the vocal tract.

The speech pattern extraction was based on the Mel-scale cepstral analysis [3]. The signal was first transformed using a Fast Fourier Transform (FFT), then was applied to a Mel-scale filterbank. The Mel-scale is a non-linear frequency scale reflecting the human auditory system perception capabilities and is related to the normal frequency scale using the relation:

$$F_{MEL} = 2595 \log_{10}\left(1 + \frac{F_{HZ}}{700}\right). \qquad (1)$$

The speech signal was split into 20 ms frames using overlapped Hamming windows and a standard radix-2 decimation-in-time FFT algorithm was used in order to compute the short-time spectrum. The spectral output from the filterbank was transformed to cepstral domain using a discrete cosine transform (DCT). The Mel-scale filterbank outputs Y_j were computed by composing the short-time magnitude spectrum using triangular Mel-scale filterbank and the weighted filterbank components falling within each band. The Mel-frequency Cepstral Coefficients C_i were computed using the following DCT:

$$C_i = \sum_{j=1}^{N}\left(\log|Y_j|\cos\left(\frac{\pi i}{N}(j-0.5)\right)\right) \qquad (2)$$

with the condition $1 \le i \le M$, where N is the number of filters in the filterbank and M is the number of desired cepstral coefficients.

As input data, we used romanian pronounciations of the 10 digits. Each of the 4 persons pronounced these words 10 times. Every time, we used the first 5 repetitions to train our system and the remaining 5 repetitions for the recognition tests. We used, therefore, 200 repetitions for training and 200 repetitions for testing. Each repetition from the training set was manually segmented to extract the signal regions corresponding to the vowels and the semivowels.

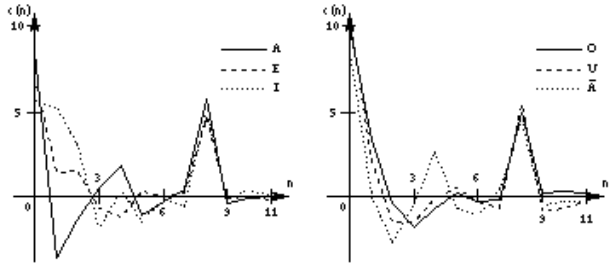

Fig 3. Typical sequences of cepstral coefficients determined for phonemes *a*, *e*, *i*, *o*, *u*, *ă*.

The compute of the feature vectors was done by mel-scale cepstral analysis of these vocalized regions in windows of 256 samples, with a step of 50 samples. The feature vectors have 12 components, computed as cepstral coefficients. In (2), we used M=12 and N=8.

The number of windows and, therefore, the number of patterns extracted from the signal associated with a phoneme is influenced by the signal temporal length and by the phoneme issue frequency. The words were recorded under normal conditions, without any special preparation, with a general purpose microphone. Figure 3 presents a typical cepstral coefficients sequence for each of the 6 vowels to be recognized by our system.

IV. EXPERIMENTS

The analyse of a signal is done in a finite time. In order to achieve this goal, we need to know the signal characteristics on the entire time interval. The vocal signal is not stationary for an infinite period. The stationarity property keeps only for few milliseconds, this is the reason why the "long term" signal analyse methods can be used, for speech analysis, only on independent, temporal windows. The speech is a dynamic process, and we need more than one window for a complete wiew. We use the "short term" analysis to study the vocal signal on frames, with a sliding window.

First of all, we need to know if a speech sequence, endig at the moment $n = m$, is vocalized or not. The vocalized signal is characterized by a higher energy. Let us consider we have two signals, $s_1(n)$ vocalized and $s_2(n)$ unvocalized, of infinite duration. Their energies will be in the following relation:

$$E_{s_1} > E_{s_2}, \qquad (3)$$

where

$$E_s = \sum_{n=-\infty}^{\infty} s^2(n). \qquad (4)$$

The equivalent formula, applied to a signal frame that contains the points around *m* is:

$$E_{S_1}(m) > E_{S_2}(m), \quad (5)$$

where

$$E_S(m) = \sum_{n=m-N+1}^{m} s^2(n) \quad (6)$$

and *N* is the number of speech samples falling in the frame. Based on (5), we can decide if the speech sequence is vocalized or not comparing the energy level with a threshold: if the energy value exceeds the threshold, the sequence is vocalized. A very important element, useful to extract any information about the signal based on the energy is *N*, the window width. For a large window, the energy will be smooth, and the decision around the threshold is not influenced by energy spikes. On the other side, a too large window can lead us to a very smooth energy and is difficult to determine the real width of the vocalized region or to determine the very short unvocalized regions. In figure 4 we present the energy values for the same sequence, using *N*=20 and *N*=2000. In our experiments, we used *N*=300 at a sample frequency of 8000 Hz, that is a window selects 37.5 ms from the signal. We used a rectangular window, allowing us a recurrent computation of the energy:

$$E_S(m) = E_S(m-1) + s^2(m) - s^2(m-N). \quad (7)$$

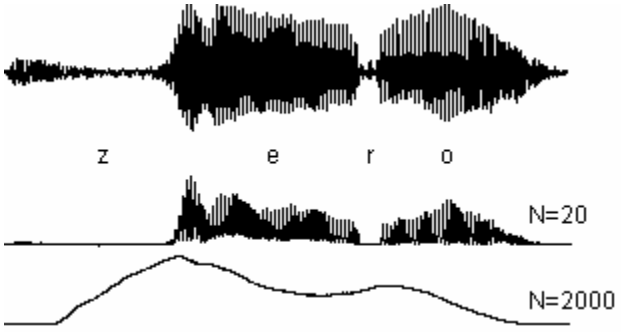

Fig. 4. The energy function of the voice signal corresponding to the romanian pronounciation of the word *zero* for *N*=20 and *N*=2000.

In order to take a more realistic decision about the vocalized or the unvocalized character of the signal, we established two thresholds. When the ascending energy exceeds the superior threshold, we consider a vocalized region begins. When, in a vocalized region, the energy becomes less that the inferior threshold, we consider an unvocalized region begin. The advantages of using two thresholds are a more accurate detection of the beginning of the vocalized regions and the avoid of considering unvocalized the lower amplitude vocalized regions from the end of the signal sequences.

In figure 5 we present the detection of the vocalized and unvocalized regions when we use one threshold and two thresholds for a pronounciation of the romanian equivalent of 'nine'. This word, phonetically transcripted as *nou≠*, contains three vowels: *o*, *u* and *≠*. For the last vowel, we used as simbol the romanian letter *ă*. We set the amplitude thresholds as percents of the highest amplitude of the signal. We can see that using two thresholds we obtain best discrimination betweend the two types of regions. We also avoid the short oscillations that could occur at the beginning and at the end of the vocalized regions.

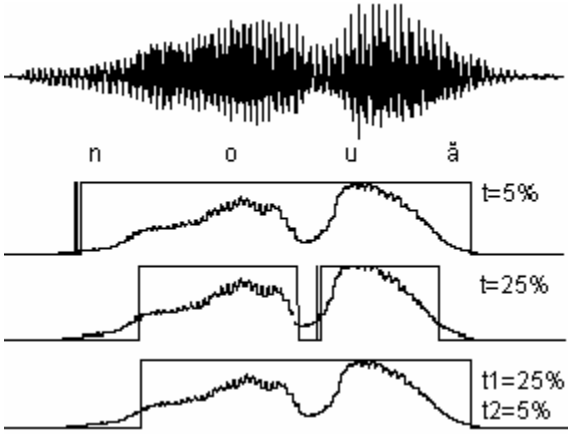

Fig. 5. Finding the vocalized and unvocalized regions of a romanian pronounciation of the word *nine*, phonetically transcripted as *nou≠*. The threshold are computed according to the maximum energy level.

The signal analyse based on the short-time energy gives us an idea about the regions where we could find the vowels, significantly decreasing the search domain.

The next step is an exact find of a vowel and determining their identity. We used an individual SOM to recognize all the six vowels from our words and we compared then these results with the performances of the SOM-CNN.

We first trained a 10x15 nodes Kohonen self-organizing map in 10000 training steps. After the calibration, we obtained the prototypes set as presented in table I. We had only 2 prototypes representing the vowel *ă* because the available training data for this vowel was reduced, determined by its low frequency issue in our words collection.

The training of SOM-CNN was done after a prevoius, supervised separation of the patterns in the six classes. Each component neural network became specialized to recognize the patterns from one class only. We placed in the same class both the vowels and the semivowels realisations of the phonemes. We used six 5x5 SOM and the global number of nodes was the same as used for the individual SOM: 150.

TABLE I
THE TRAINING PATTERNS NUMBER OF SOM AND SOM-CNN AND THE PROTOTYPES NUMBER OBTAINED AFTER TRAINING SOM AND SOM-CNN.

	A	E	I	O	U	Ă
Training	372	516	339	525	453	52
SOM	42	37	16	21	21	2
SOM-CNN	25	25	25	25	25	25

TABLE II
THE RESULTS OF THE EXPERIMENTS OF VOWEL AND SEMIVOWEL RECOGNITION USING SOM AND SOM-CNN.

	SOM	SOM-CNN
E1	73.68%	84.21%
E2	62.47%	73.93%
E3	50%	56.25%
E4	5.88%	5.48%
E5	13.07%	8.61%

The recognition was done as following. The vocalized sequences from the test signal were preprocessed using the mel-scale cepstral analysis, as prevoiusly described. The resulting vectors were compared to the prototypes and we selected the prototype that generated the minimum quantization error, but not exceeding an acceptable limit. We obtained sequences of class labels, as depicted in figure 6 using as well the romanian equivalent of the word 'nine'. The SOM does not recognize the final *ă*. Both models find an *o* after *u* because in the current speech, a short *o* is inserted on this position, though this vowel does not appear in the phonetical transcription.

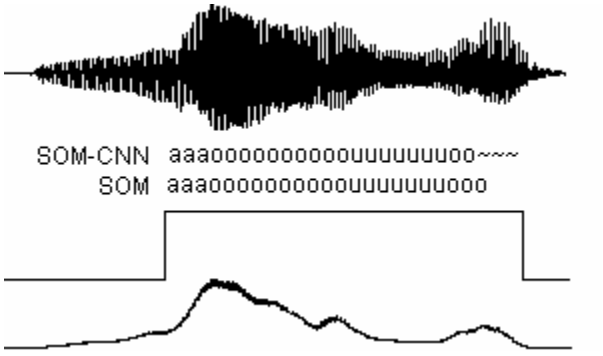

Fig. 6. The way that was interpreted the romanian pronounciation of the word nine by SOM and SOM-CNN. We used ~ as symbol of the phoneme ≠.

To compare the performances of SOM and SOM-CNN we presented in table II the results of the following experiments: the percent of the vowels and semivowels correctly recognized by the two models (E1), the proportion that the real dimension of the vowel or semivowel was detected (E2), the number of sequences vowel-semivowel correctly recognized (E3), the percent of false alarms, meaning the detection of a vowel or semivowel instead of a consonant (E4) and the percent of confusions between vowels (E5).

V. CONCLUSIONS

We presented in this paper the experiments of behaviour of the SOM-CNN for a problem of vowel recognition. We used as database the romanian pronounciations of the 10 digits. Each of them was repeated 10 times by 4 persons, males and females. We used half of the repetitions for training and the remaining of them for testing. The training data set was created by manually segmentation of the words by extracting the regions corresponding to the vowels. These segments were processed by Mel-scale based cepstral analysis. The resulting vectors were used as input patterns for the maps training algorithm.

We tested SOM and SOM-CNN and the rate of the correctly recognized vowels and semivowels of CNN was superior with more than 10% than the individual SOM.

REFERENCES

[1] C.M. Bishop, *Neural networks for pattern recognition*. New York: Oxford University Press, 1995.

[2] A. Cataron and V.-E. Neagoe, "Concurrent Neural Networks for Speaker Recognition". Proceedings of IEEE International Conference on Telecommunications ICT2001, Special Sessions, Bucharest, Romania, pp. 252-257, 2001.

[3] J.R. Deller, J.G. Proakis and J.H.L. Hansen, *Discrete-time processing of speech signals*. Upper Saddle River, New Jersey: Prentice Hall, 1987.

[4] S. Haykin, *Neural networks - A comprehensive Foundation*. New York: Macmillan College Publishing Company, 1994.

[5] T. Kohonen, "The self-organizing map". *Proceedings of the IEEE* 78, pp. 1464-1480, 1990.

[6] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Englewood Cliffs, New Jersey: Prentice Hall, 1993.

[7] A. Waibel., T. Hanazawa, G. Hinton, K. Shikano and K.J. Lang, "Phoneme recognition using time-delay neural networks". *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-37, pp. 328-339, 1989.

[8] J.M. Zurada, *Introduction to Artificial Neural Systems*. St. Paul, Minessotta: West Publishing Company, 1992.