Concurrent Neural Networks for Speaker Recognition

Angel Cataron

Dept. of Electronics and Computers,
TRANSILVANIA University of Brasov, Romania
email: Cataron@vega.unitbv.ro

Victor-Emil Neagoe

University POLITEHNICA of Bucharest,
Faculty of Electronics and Telecommunications, Bucharest, Romania
Tel: +40 92 302998, email: Vic@gitprai.pub.ro

ABSTRACT

We propose a new recognition model called *Concurrent Neural Networks (CNN)*, representing a *winner-takes-all* collection of neural networks. Each network of the system is trained individually to provide best results for one class only. We have applied the above model for the task of speaker recognition. We performed distinct speaker recognition experiments using three variants of basic components of the CNN system: the Multi-Layer Perceptron (MLP), the Time-Delay Neural Network (TDNN) and the Kohonen Self-Organizing Map (SOM). We have used two databases: a clean speech database called SPEECHDATA and a telephone database called TELEPHDATA. The experiments proved a significant increase of the recognition score using the proposed CNN model by comparation to the use of a single neural network for the whole speaker recognition task. The SOM best has performed in our experiments proving an increase of about 38% for SPEECHDATA as well as an increase of about 30% for TELEPHDATA.

I. INTRODUCTION

The speech and speaker recognition experiments proved that the classical neural networks models do not perform very satisfying. MLP, TDNN and SOM provide acceptable recognition accuracies for isolated word recognition, but the performances decrease dramatically for speaker recognition tasks [2]. On the other side, reducing the number of speakers leads to better results, and we tried to exploit this result. We proposed and we tested a complex network model, the *concurrent neural network* which consisted in a collection of specialized small neural networks.

II. CONCURRENT NEURAL NETWORKS

Concurrent neural networks (CNN) are a collection of neural networks which use a global *winner-takes-all* strategy. Each network is used to correctly classify the patterns of one class and the number of networks equals the classes number. The CNN training technique is a supervised one, but for the individual networks their particular training algorithms are used. The CNN model is depicted in figure 1.

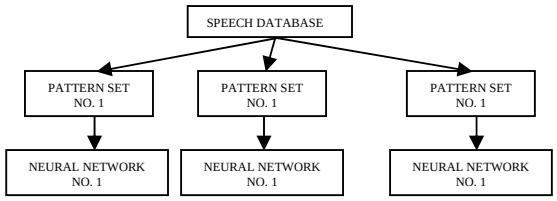

Figure 1: The CNN model used in the training phase.

In this figure, *n* is the number of parallel performing neural networks and, also, the classes number. The database consists of the pre-processed speech signal and the pattern sets extracted from this database are inputs of the networks in the training phase. Each network shold be activated by the patterns in the training class only. The training algorithm of the CNN is the following:

Step 1. Create the database by pre-processing the voice signal.

Step 2. Extract the training pattern sets from the database. If necessary,add the desired outputs.

Step 3. Use the specific training algorithm for each individual neural network from the CNN using the training sets from step 2.

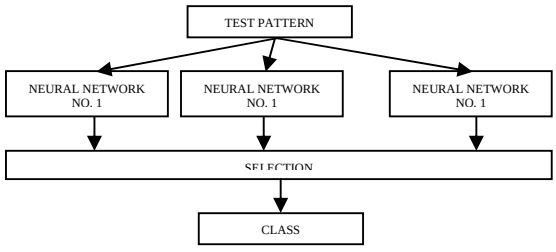

Figure 2: The CNN recall model.

In the recall phase, the CNN use an outputs selection of the individual networks (fig. 2). The selection consists of finding the best response net. The selected network is declared the *winner* and its index is the class index. This classifying method suppose that the classes number is *a priori* known and for each class sufficiently patterns number are available. The recall algorithm is the following:

Step 1. Create the input pattern by pre-processing the voice signal.

Step 2. Apply the pattern in parallel to the *n* trained networks.

Step 3. Find the best response network using the *winner-takes-all* selection strategy. The network index is the patterns' class index.

In our experiments we used MLP, TDNN and SOM as individual neural networks. We used the backpropagation of error training algorithm for MLP and TDNN. For SOM we used the Kohonen training procedure.

MLP-CNN and TDNN-CNN use pairs of training patterns and desired responses. The *positive patterns* describe the class with the same index as the network. A *negative pattern* is associated to the other classes. For an efficient training, we balanced the number of positive and negative patterns.

Because SOM use an unsupervised training algorithm, SOM-CNN use positive training patterns only. As previously, each component network is trained with its dedicated pattern set. The classification decision is based on the minimum quantization error. The map which generate this minimum is selected as winner.

III. SPEECH DATABASES AND FRONT-END PROCESSING

To test different types of CNN, two speech databases were used in the speaker recognition experiments. They consist of 20 repetitions of 12 words ("one"-"nine", "zero", "nought" and "oh") spoken by 25 talkers, giving a total size of 6000 utterances. The first database, SPEECHDATA, contains data collected under controlled conditions, with a minimum amount of noise interference. The talker's speech was recorded on a proffesional cassette tape recorder using a high-quality microphone. The recorded voice signal was then digitised using a 12-bit,

analog/digital converter, at a 7.5 KHz sampling rate. The second database, TELEPHDATA, is more colosely to the conditions in a real-world application of speaker recognition using the telephone network. The acquired speech was processed with the same system as presented above.

The speech pattern extraction was based on the 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 X_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:

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

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. For each frame P a delta coefficient d_P computed with the following relation were used:

$$d_P = \frac{\sum_{i=1}^{M} i\left(C_{P+i} - C_{P-i}\right)}{2\sum_{i=1}^{M} i^2} \qquad (3)$$

For the beginning and the end of the word, the delta coefficients were computed using simple first-order differences:

$$d_p = C_{P+1} - C_P, \quad P < M \qquad (4)$$

$$d_P = C_P - C_{P-1}, \quad P \ge N_F - M \qquad (5)$$

where N_F is the frames number in the utterance.

In the SPEECHDATA and TELEPHDATA, the values of N and M were 16 and 8. A frame pattern consisted of 8 cepstral coefficients, 1 coefficient representing X_j and 9 delta coefficients. The speech signal of each word was processed in 15 overlapped Hamming windows and resulted 270 feature coefficients per word.

IV. EXPERIMENTS

These experiments compare the recognition capabilities of MLP, TDNN, SOM and the CNN which use these basic models as components in a speaker recognition task. We used the two databases, SPEECHDATA and TELEPHDATA, which consisted of words spoken by 25 persons, therefore we had 25 classes.

We first used the single, basic networks (MLP, TDNN and SOM) in distinct experiments to test them for the whole speaker recognition task.

The input layers of these neural networks consisted of 18x15 units. The training sequence consisting of 270-components arrays was applied to this layer. The 12 words were pronounced 20 times by each of the 25 talkers. We used the first 8 repetitions set in the training phase and the remaining 12 repetitions set for the recognition tests.

MLP and TDNN were trained supervised with the backpropagation of error trainig procedure. Therefore, we associated to each training pattern a desired response of the neural network. We designed the MLP and the TDNN to consist of two hidden neurons layersand an output layer with 25 units. The words spoken by the first talker should activate the first output unit and let the other units inactive, the words spoken by the second talker should activate the second output unit and let the other units inactive, etc. In the ideal case, the response of an active output unit was 1 and the response of the inactive output was 0. The desired response associated to the words from the first class was (1,0,0,...,0), the desired response associated to the words from the second class was (0,1,0,...,0), etc. In the

recognition phase, we accepted a threshold of 0.3 to decide if an output unit becameactive or remained inactive when a test pattern was applied to the neural network's input layer.

We trained the SOM using the unsupervised Kohonen Self-Organized Map training algorithm. After the training phase, each output unit was calibrated using the training patterns set. The classification decision of a test pattern was drawn by finding the best matching unit according to the Euclidian distance.

The recognition scores obtained to the tests of the basic units are presented in the table 1 on the column marked with the letter **B** (basic architecture) for both the SPEECHDATA and TELEPHDATA.

Table1: The recognition rates in the speaker recognition experiments using MLP, TDNN and SOM individually and as basic components of a CNN.

	SPEECHDATA		**TELEPHDATA**	
	B	**CNN**	**B**	**CNN**
MLP	**71.25%**	**86.22%**	**6.75%**	**72.86%**
TDNN	**57.28%**	**89.31%**	**32.33%**	**72.72%**
SOM	**52.86%**	**90.42%**	**44.72%**	**75.25%**

In the second set of experiments, we tested the proposed neural model, theCNN in the same speaker recognition task.

We first used MLP and then TDNN as basic components of CNN. We decided touse 25 neural networks in each case, one for each class. The neural networks consisted of a 270-units input layer, two hidden layers and a 25-units output layer. We also used the *backpropagation of error* training algorithm for these neural networks, but we created distinct training patterns sets for each of them. We defined two types of input patterns. We named *positive examples* the input patterns which belonged to the class with the same index as the component neural network that they were applied to. The input patterns from the other classes to the same neural network we named *negative examples*. According to this definition, the patterns from class 1, extracted from the words spoken by speaker 1, were positive examples for the neural network with index 1 (figure 1) and were negative examples for the other neural networks. The desired responses of the positive examples had the same format as described above, when we trained MLP and TDNN as recognisers for the whole speaker recognition task. We associated the desired response (1,0,0,...,0) to the positive examples of the neural network 1 and the desired responses (0,1,1,...,1) to the negative examples of neural network 1. In order to balance the number of positive and negative examples, we copied 24 times each positive example in each training patterns set. Based on these ideas, we created complete training sets for each basic neural networks of MLP-CNN and TDNN-CNN. In the recognition phase, a test pattern should activate only one neural network, that is this activated neural network provides an output close to the response of a positive pattern. Every other neural network from the CNN should provide outputs close to the response of their negative example. We draw a decision only when exactly one of the 25 neural networks provided a positive responses and all the other provided negative responses. Else, the response was undecided and the pattern was not recognised.

In figure 3 we presented the average recognition rates for the 25 speakers in two typical experiments using TDNN and TDNN-CNN. It can be seen that TDNN-CNN increase the average recognition rate, though it is possible that TDNN provide better results for some individual speakers.

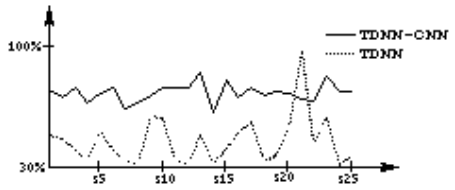

Figure 3: The average recognition scores for the 25 speakers in two tipical recognition experiments using TDNN and TDNN-CNN.

The training of the SOM-CNN was much simpler. We created 25 training patterns sets and we used the Kohonen Self-Organising Map training algorithm independently for each of the 25 SOMs. For the recognition, the test pattern was aplied in parallel to every

SOM. The map providing the least quantization error were decided to be the winner and its index were the class index that the pattern belonged to.

In the table 1 we presented the recognition rates for the MLP-CNN, TDNN-CNN and SOM-CNN in the columns marked with **CNN**.

V. CONCLUSIONS

We tested the CNN with MLP, TDNN and SOM as basic components in order to study their recognition capabilities for speaker recognition tasks. The speech data consisted in a clean database, SPEECHDATA, acquired in a relatively noise-free room environment, and a telephone database, TELEPHDATA, acquired over conventional dial-up lines. Each database comprised 20 repetitions of 12 isolated words (the digits 0-9 plus "nought" and "oh") each spoken by 25 talkers. Each word was parametrized into a time sequence of 15 frames of an 18-dimension feature vector, based on the Mel-scale cepstral analysis. The recognition rates obtained in the speaker recognition experiments using the CNN improved the results obtained using the classical neural models. MLP performed better than TDNN and SOM with SPEECHDATA, but the results with TELEPHDATA were very weak. On the other side, SOM-CNN provided the best performances, their results being around 4% better than MLP-CNN and superior to the TDNN-CNN. We also obtained an increase of approximately 15% of the scores obtained with SPEECHDATA comparing to the results obtained using TELEPHDATA for MLP-CNN, TDNN-CNN and SOM-CNN.

These experiments proved that the use of specialized neural networks for each class leads us to better classification rates, though the training phase takes a longer time.

ACKNOWLEDGMENT

The authors would like to thank Dr. F.J. Owens of the Universiry od Ulster at Jordanstown for providance of the two speech databases.

REFERENCES

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

[2] Cataron, A. "Speech recognition using time delay neural networks". MSc Thesis, POLITEHNICA University of Bucharest, 1996.

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

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

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

[6] Neagoe, V.-E, O. Cula "A fuzzy connectionist approach to vowel recognition". *In Real World Applications of Intelligent Technologies (part II)*, eds. B. Reusch and D. Dascalu, printed by National Institute for Research and Development in Microtechnologies, Bucharest, 1998.

[7] Neagoe, V.-E, O. Stanasila "Recunoasterea formelor si retele neurale - algoritmi fundamentali". Ed. Matrix Rom, Bucuresti, 1999.

[8] Owens, F.J., R. Andonie, G.H. Zheng, A. Cataron, S. Manciulea "A comparative study of the multy-layer perceptron, the multi-output layer perceptron, the time-delay neural network and the Kohonen self-organizing map in an automatic speech recognition task". Proceedings of EIS'98 International ICSC Symposium on Engineering of Intelligent Systems, ICSC Academic Press, 624-629, Tenerife, Spain, February 11-13, 1998.

[9] Rabiner, L., B.-H. Juang "Fundamentals of speech recognition". Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[10] Waibel, A., 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, 328-339, 1989.

[11] Zurada, J.M. "Introduction to Artificial Neural Systems". West Publishing Company, St. Paul, Minessotta, 1992.