*Proceedings of the 10th Israeli Symposium on Artificial Intelligence Computer Vision and Neural Networks, pp. 177-183, December 27-28, 1993*

# Speaker Independent Vowel Recognition Using Neural Networks.

by
I. Voitovetsky, S. Dahan, Y. Menashe, and H. Guterman
Department of Electrical and Computer Engineering, Ben-Gurion University
of the Negev, P.O. Box 653, Beer-Sheva, 84105, ISRAEL

***Abstract -*** The search for optimal features and classification methods for speech recognition is a "never ending story". Due to the lack of standard criteria for testing and analysis comparison between the different researchers are very difficult. In the present work an effort has been done to obviate the above described problems. Several features for speaker independent vowel recognition were tested as a step to speaker independent phoneme recognition. Features such as linear prediction coefficients (LPC), LPC based Cepstrum, Delta-cepstrum, and Parcor coefficients, extracted from a set of 60 speakers were employed as inputs to multi-layer perceptron (MLP), Kohonen's network and fuzzy-MLP networks. Results of the performance of the different approaches will be presented.

## 1. Introduction

The problems of vowel recognition have been a subject of intensive research. The proposed approaches employ either traditional signal processing techniques, neural network architectures or a combination of both [2]. During the last years the research on the application of NN for speech recognition has significantly increased in terms of both quantity and quality [3,7-9,11,14,16]. However, as most of the proposed architectures were tested under different conditions it is not possible to make performance comparisons.

In order to correct the above describe shortcomings we have employed a relatively large database composed by 60 speakers. The speech database consist of 5 words repeated at least 4 times by the 60 speakers. The extracted features were employed to train and test MLP [18] and Kohonen [4-6] based networks.

## 2. Feature Extraction and Preprocessing

The employed databases consist of 5 words spoken by 60 speakers. Each word contains one of the vowels: /a/, /e/, /i/, /o/ or /u/. The speech signal was sampled after filtering (with a band pass filter, 65Hz-4.7KHz) at 10KHz, with an A/D resolution of 14Bits plus sign bit. The segmentation was manually performed. A high pass filter of the form,

$$HPF(z) = 1 - 0.95z^{-1} \qquad (1)$$

was employed to pre-emphasized the extracted phonemes. The database was divided into a training set of 40 speakers and a test set containing 20 speakers.

The features employed in this work were based on LPC analysis. The LPC and Parcor features were extract using the Levinson-Durbin method [10]. The Cepstrum features were calculated from LPC [1]. The Delta-Cepstrum features [2,15] were calculated as:

$$\Delta c_m(t) = \sum_{k=-2}^{2} k c_m(t+k) \qquad (2)$$

Features for the MLP network were extracted from windows of 25.6ms with 50% overlapping. The orders of the extracted models for the LPC, Parcor and Cepstrum were 12, 16, 18, and 25. A model of order 18 was employed for the Delta-cepstrum features (2).

The feature vectors, for the Kohonen networks, were extracted from 15ms windows with 50% overlapping. The extracted features were: 12th order LPC, 16th order Cepstrum, and 16th order Delta-cepstrum as defined above (2).

## 3. Neural Networks Architecture

Four different NN architectures were employed. A brief description of the relevant issues follows.

### 3.1 Multi-layer perceptron

The proposed classifier consists of a two layer feed forward network [12,13,18]. Training was carried out by an improved Back-Propagation (BP) algorithm [17].

### 3.2 Kohonen Network

Kohonen networks have been extensively used for phoneme classification [3,6-7,11]. Kohonen networks without LVQ or with LVQ1 were employed. Training was achieved with the algorithm described by Kohonen [5]. The performance of the algorithm is greatly affected by the selection of the different training parameters. Therefore, for the sake of consistency we will describe the training strategy. The weight vector $m_i(t)$ is updated as follows,

$$m_i(t+1) = \begin{cases} m_i(t) + h_{ci}(t)\left[x(t) - m_i(t)\right] & \text{if } \; i \in N_c(t) \\ m_i(t) & \text{if } \; i \notin N_c(t) \end{cases} \tag{3}$$

where $N_c(t)$ is the neighborhood region that linearly decreases with time. The parameter $h_{ci}(t)$ might be updated by one of the two options:

$$h_{ci} = \begin{cases} \alpha(t) \\ h_0(t)\exp\left(-\left\|r_i - r_c\right\|^2 / \sigma(t)^2\right) \end{cases} \tag{4}$$

where:

$r_i$ - coordinates of the cell to be update.

$r_c$ - coordinates of the "winner" cell.

$\alpha(t)$, $h_0(t)$, $\sigma(t)$ - linearly decreasing functions.

The training algorithms consist of two phases: the fast learning phase and the slow learning phase. In the fast phase, that took 10% of all training iterations, the neighborhood region $N_c(t)$ starts at 80% of the network, and decrease to 10%, $h_{ci}(t)$ decreases from about 1 to 0.1. During the slow phase all the parameters should arrive to zero. A compendium of the different strategies employed for training can be found in Table 1. The $\alpha(t)$ and $h_0(t)$ employed were for all the cases 0.9 to 0.1 for the fast phase and 0.1 to 0 during the slow phase, while $\sigma(t)$ changes from 20 to 1 during the first stage and from 1 to 0 after it.

| Strategy | Fast Phase $h_{ci}(t)$ | Slow Phase $h_{ci}(t)$ |
|---|---|---|
| 1 | $\alpha(t)$ | $\alpha(t)$ |
| 2 | $h_0(t)\exp\left(-\|r_i - r_c\|^2 / \sigma(t)^2\right)$ | $h_0(t)\exp\left(-\|r_i - r_c\|^2 / \sigma(t)^2\right)$ |
| 3 | $h_0(t)\exp\left(-\|r_i - r_c\|^2 / \sigma(t)^2\right)$ | $\alpha(t)$ |

Table 1. Kohonen's training strategies

### 3.3 Hybrid Kohonen-MLP architecture

The hybrid architecture consists of a Kohonen network (5x5 cells) without LVQ and a conventional two layer feed forward network. The two networks were trained independently. First, a Kohonen network was trained without LVQ. Then, the Kohonen network outputs were employed as inputs for training the MLP. The forward process involves feeding the features' frames through the Kohonen network and the MLP.

### 3.4 Hybrid Fuzzy-MLP architecture

Three gaussian membership functions were defined from the histograms of the feature vectors (Fig. 1). The gaussian functions were manually defined. The training algorithm for the MLP was as above described.

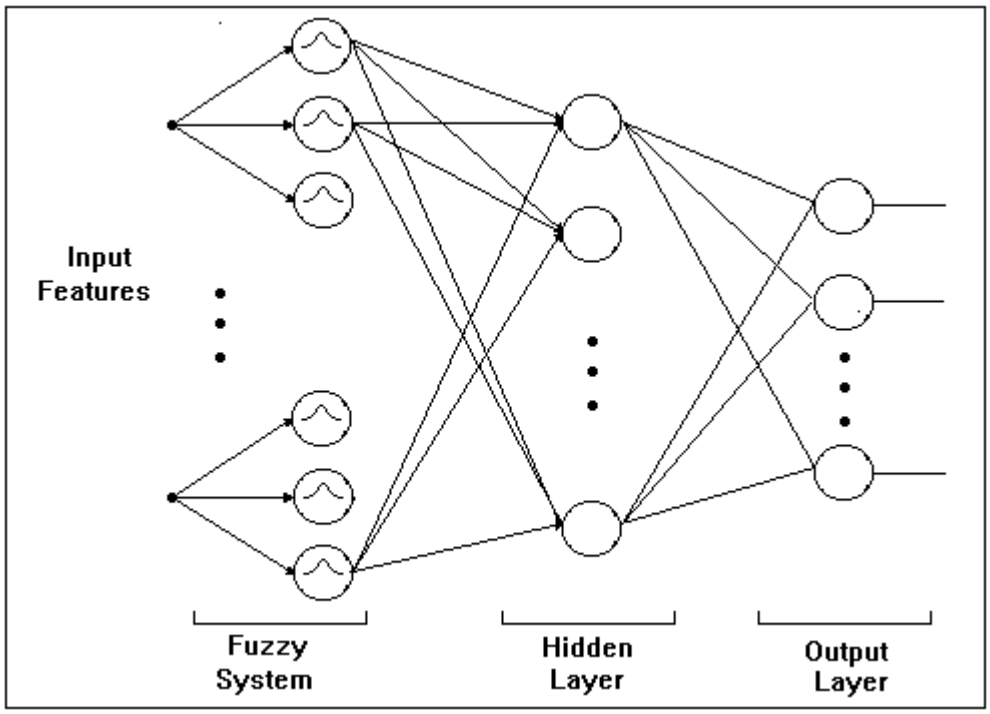


Fig. 1. Hybrid Fuzzy-MLP architecture.

## 4. Results and Discussion

Two criteria were employed to evaluate the performance of the networks:

1. Each frame was independently tested.
2. All the frames obtained from a phoneme were tested and a majority vote decision was taken.

### 4.1 MLP architecture

The classification success as a function of the order of the model can be seen in Fig. 2-3. The results obtained are similar for the two criteria. The LPC model is better for a low order model, while the Cepstrum performance improves for high order models.

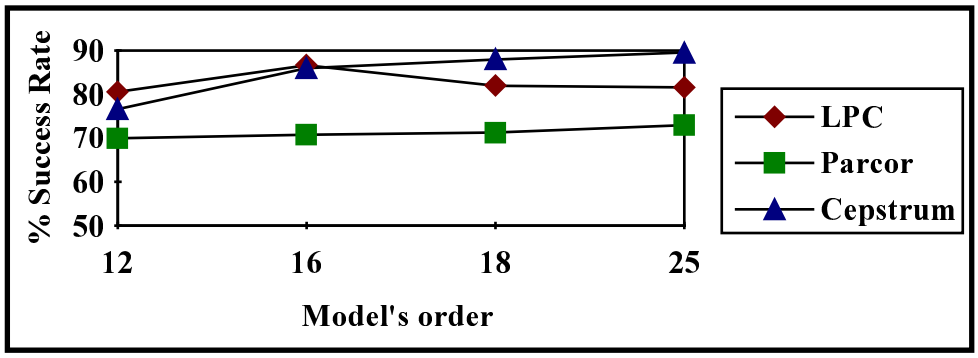


Fig. 2. Success Rate for the first criteria

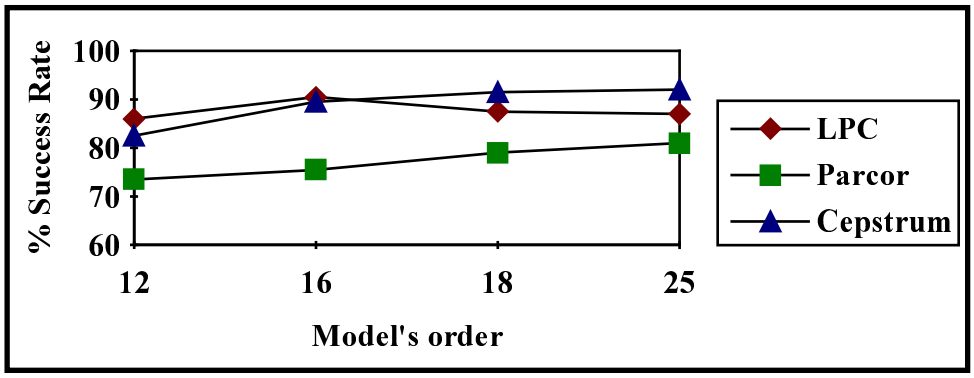


Fig. 3. Success Rate for the second criteria

The results of the generalization performances' tested on 18th order models is shown in Table 2. The Delta-cepstrum was included in this test. The Ceptrum continue to be the best choice.

| Features/Data type | Train data | Test data |
|---|---|---|
| LPC | 95.0 | 87.5 |
| Parcor | 89.3 | 79.0 |
| Cepstrum | 96.8 | 91.5 |
| Delta-cepstrum | 78.0 | 71.0 |

Table 2. MLP generalization. second evaluation criteria

### 4.2 Kohonen Networks

The Kohonen networks were tested with: 12th order LPC, 16th order Cepstrum, 16th order Delta-cepstrum and a combination of Cepstrum and Delta-cepstrum features. The training strategy was 2, as defined in Table 1, for a network of size 10x10 with 200,000 iterations, 20,000 during the fast phase. Tables 3 and 4 summarize the results obtained for the training without LVQ and with LVQ1. The training with LVQ1 required an additional 200,000 iterations.

| Features | Train *Success [%] | Train **Success [%] | Test *Success [%] | Test *Success [%] |
|---|---|---|---|---|
| LPC | 65.91 | 70.13 | 52.64 | 55.00 |
| Cepstrum | 81.71 | 85.38 | 73.68 | 80.50 |
| Delta-cep | 46.43 | 66.38 | 42.11 | 60.50 |
| Both | 68.99 | 78.75 | 61.62 | 70.25 |

Table 3. Kohonen's classification results without LVQ.
* - First Criteria; ** - Second Criteria

| Features | Train *Success [%] | Train **Success [%] | Test *Success [%] | Test *Success [%] |
|---|---|---|---|---|
| LPC | 77.36 | 83.38 | 57.12 | 61.00 |
| Cepstrum | 90.15 | 95.00 | 78.47 | 83.75 |
| Delta-cep | 61.07 | 83.88 | 54.19 | 75.50 |
| Both | 82.84 | 89.50 | 74.26 | 82.00 |

Table 4. Kohonen's classification results with LVQ1.
* - First Criteria; ** - Second Criteria

The obtained results were obviously better for the second criteria and the LVQ1 algorithm. The influence of the size of the network was also investigated (Fig. 4-5). As expected, the influence of LVQ1 decreases as the numbers of cells increase.

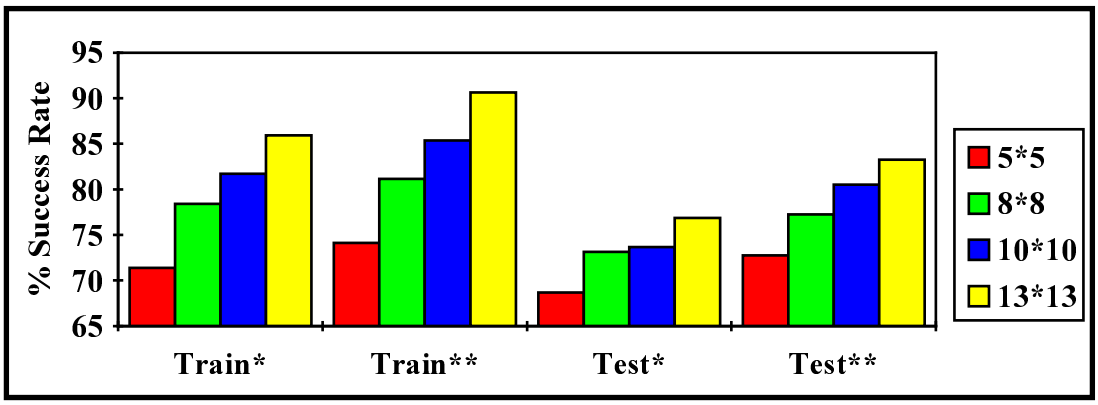


Fig. 4. Kohonen's classifications a function of the network size without LVQ. * - First Criteria; ** - Second Criteria

Finally, the classification performance for LVQ1 was also tested as a function of the number of iterations (Fig. 6). The general conclusion was that there is no significant improvement after 40,000 iterations.

### 4.3 Hybrid Kohonen-MLP architecture

A Cepstrum model of order 16th was employed to train a Kohonen network of size 5x5 without LVQ. The obtained Kohonen network outputs were employed as inputs for training the MLP. The results were 87% for the train set and 77.3% for the test set.

*Proceedings of the 10th Israeli Symposium on Artificial Intelligence Computer Vision and Neural Networks, pp. 177-183, December 27-28, 1993*

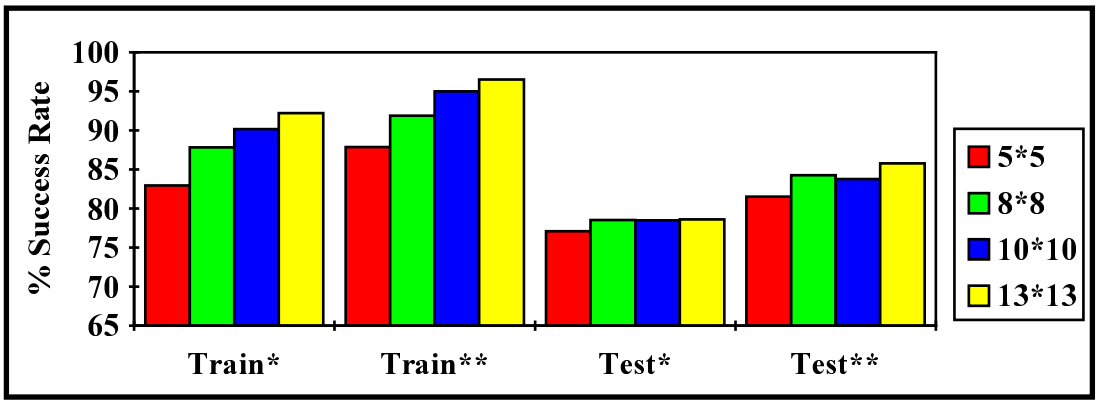


Fig. 5. Kohonen's classifications a function of the network size with LVQ1. * - First Criteria; ** - Second Criteria

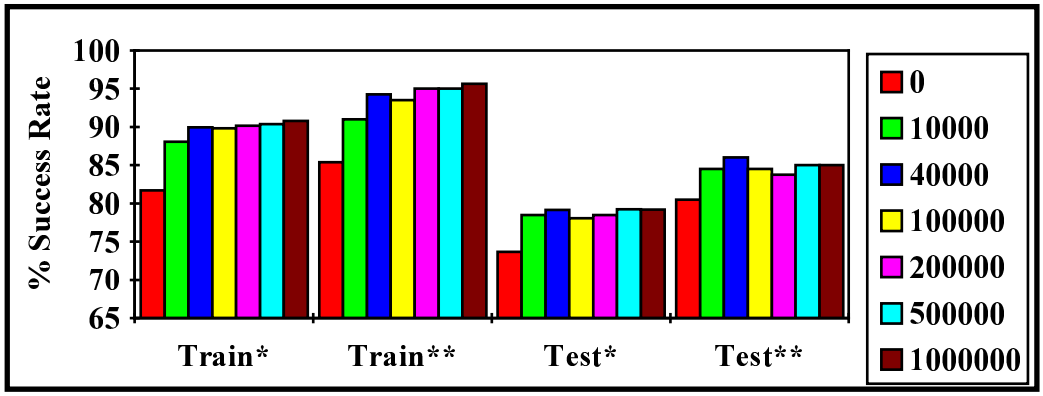


Fig. 6: Kohonen's classification as a function of the training LVQ1 iterations number. * - First Criteria; ** - Second Criteria.

### 4.4. Hybrid Fuzzy-MLP architecture

The input to the fuzzy network classifier was a Cepstrum model of order $18^{th}$. The results achieved with this configuration: 97.75% for the train set and 92.50% for the test set. A general comparison between the architectures is shown in Table 5.

| Architecture | Features | Train [%] | Test [%] |
|---|---|---|---|
| MLP | Cepstrum $25^{th}$ | 97.5 | 92 |
| Kohonen-LVQ1 | Cepstrum $16^{th}$ | 95 | 83.75 |
| Kohonen-MLP | Cepstrum $16^{th}$ | 87 | 77.3 |
| Fuzzy-MLP | Cepstrum $18^{th}$ | 97.75 | 92.5 |

Table 5: The best results obtained.

## 5. Conclusions

The results obtained from this work demonstrated that the performance of the Kohonen's network is inferior to the simple MLP. At the same time, it opens the avenue for further research in the two hybrid architectures. An increase in the number of cells in the Kohonen-MLP network or training with LVQ1 might significantly increase its performance. Also, a hybrid Fuzzy-MLP with adaptation in the Fuzzy layer may achieve better results.

## REFERENCES

[1] B. S. Atal, "Effectiveness of linear prediction characteristics of speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Am.*, vol. 55, no. 6, pp. 1304-1312, June 1974.

[2] J. R. Deller, JR, J. G. Proakis and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*. New York, New York, U.S.A: Macmillian Publishing Company, 1993.

[3] P. Knagenhjelm and P. Brauer, "Classification of vowels in continuous speech using MLP and a hybrid net," *Speech Communication*, vol. 9, pp. 31-34, 1990.

[4] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Berlin, Heidelberg, Germany: Springer-Verlag, 1989.

[5] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp.1464-1480, September 1990.

[6] T. Kohonen, "The "neural" phonetic typewriter," *Computer*, vol. 21, pp. 11-22, March 1988.

[7] T. Kohonen, K. Makisara and T. Saramaki, "Phonotopic maps - insightful representation of phonological features for speech recognition," *Proc. 17th Int. Conf. on Pattern Recognition*, pp. 182-185, 1984.

[8] Y. Komori, K. Hatazaki, T. Tanaka and T. Kawabata, "Combining phoneme identification neural networks into an expert system using spectrogram reading knowledge," *ICASSP-90*, vol. 1, pp. 505-508, 1990.

[9] R. P. Lippmann, "Review of neural networks for speech recognition," *Neural Computation*, vol. 1, pp. 1-38, 1989.

[10] J. Makhoul, "Linear prediction: a tutorial review," *Proc. IEEE*, vol. 63, no. 4, pp. 561-580, April 1975.

[11] E. McDermott and S. Katagiri, "LVQ-based shift-tolerant phoneme recognition," *IEEE Trans. Signal Processing*, vol. 39, no. 6, pp.1398-1411, June 1991.

[12] D. E. Rumelhart, G. E. Hinton and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 9, pp. 533-536, October 1986.

[13] D. E. Rumelhart, J. L. McClelland and the PDP Research Group, *Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations*, MIT Press, Cambridge, Mass., 1986.

[14] H. Sawai, Y. Minami, M. Miyatake, A. Waibel and K. Shikano, "Connectionist approaches to large vocabulary continuous speech recognition," *IEICE Trans.*, vol. E-74, no. 7, pp. 1834-1844, 1991.

[15] F. K. Soong and A. E. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 6, pp. 871-879, June 1988.

[16] K. P. Unnikrishnan, J. J. Hopfield and D. W. Tank, "Connected-digit speaker-dependent speech recognition using a neural network with time-delayed connections," *IEEE Trans, Signal Processing*, vol. 39, no. 3, pp. 698-713, March 1991.

[17] T. P. Volg, J. K. Mangis, A. K. Rigler, W. T. Zink and D. L. Alkon, "Accelerating the convergence of the backpropagation method," *Biological Cybernetics*, vol. 59, pp. 257-263, 1988.

[18] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415-1442, September 1990.