A HYBRID LEARNING VECTOR QUANTIZATION/TIME-DELAY NEURAL NETWORKS SYSTEM FOR THE RECOGNITION OF ARABIC SPEECH

S.A. Selouani[1], *J. Caelen*[2]
[1]LCP Institute of' electronics, USTHB, BP 32 El Alia-Algiers-Algeria
[2]CLIPS/IMAG, BP 53, 38041 Grenoble cedex 9 France

ABSTRACT

In this paper, we present an approach which significantly improves the performances of automatic speech recognition systems (ASRSs) dedicated to Arabic language. We propose to combine a version of Learning Vector Quantization (LVQ) and Time Delay Neural Networks (TDNNs) using an autoregressive version (AR) of the backpropagation algorithm. The underlying idea of this approach consists in the incorporation of AR-TDNNs in a hybrid structure in order to give the LVQ-based system the ability to overcome failures due to the language particularities such as emphasis, gemination and vowel lengthening. The test corpus is composed of subsets taken from an Arabic database. The results show that the proposed LVQ/AR-TDNN system achieves a highly recognition rate compared to the baseline LVQ-based system.

1. PROBLEMATIC OF COMPLEX ARABIC PHONEMES RECOGNITION

The present systems of automatic speech recognition (ASR) dedicated to Arabic remain confronted to the problems of the strong inflexion of the language. This syntactic particularity is complicated by the poverty of the Arabic vocalic system which is partially compensated by the semantic relevance of the vowels lengthening. In the consonantal system, another phonetic complexity resides in the presence of features as subtle as emphasis and gemination [1]. Unfortunately, the developed ASR systems do not take into account these phonetic properties in order to limit the drop of their performances. This has as a consequence the quasi absence of any commercial product dedicated to Arabic language while we are observing a boom of which is actually called *'language industries'*.

Therefore, in the case of an emphatic vs. non-emphatic opposition, an efficient ASR system must be capable to distinguish, for example, between the two words: /sa:ra/ (to walk) and /s̲a:ra/ (to become), where an emphasis is observed over /s̲/ fricative. The present ASR systems cannot easily raise this ambiguity. In the following example illustrating the gemination case, we require the ASR to discriminate between the two words: /nafaδa/ (to escape) and /naf:aδa/ (to execute), where the /f/ fricative is geminated. A similar problem is encountered in the vocalic system. For instance, the two words: /suru:ru/ (happiness) and /sururu/ (umbilical cord) differ only by the lengthening of the second vowel /u/. We require the recognition system to detect this vowel without altering its temporal property. It is proposed to be done by an original combination of Waibel's TDNN [12] and a modified version of the Kohonen's LVQ algorithm [7].

2. AUTOREGRESSIVE TIME DELAY NEURAL NETWORKS (AR-TDNN)

Contrarily to feedforward networks, recurrent networks are generally trickier to work with, but they are theoretically more powerful, having the ability to represent temporal sequences of unbounded length. Because speech is a temporarily unstable phenomenon, we consider recurrent networks to be more adequate than feedforward networks. Another consideration related to phonetic context influence leads us to use an autoregressive version of backpropagation algorithm (AR-back propagation) proposed by Russel [9]. This type of networks can in principle captures naturally the co-articulation phenomenon of speech. Some studies show that they are very performing in the context-dependent labeling. However, this power turns out to be source of disappointment in the case of phoneme time shifting. The approach we are investigating proposes to integrate in addition to the AR component, a delay component similar to the one used by Waibel's TDNN [12]. Through this combination, we expect that the ability of the system to discern the phonological length even in a strong coarticulation context will be increased.

The model described by Russel et al includes an autoregressive memory which constitutes a form of self-feedback where the output depends on the current output plus a weighted sum of previous outputs. Then, the classical AR node equation is given by:

$$y_i(t) = f\left(bias_i + \sum_{j=1}^{P} w_{i,j} x_j(t) \right) + \sum_{n=1}^{M} a_{i,n} y_i(t-n) \qquad (1)$$

$y_i(t)$ is the output of node i at time t. $f(x)$ is the $tanh(x)$ bipolar activation function, P is the number of input units. M is the order of autoregressive prediction. Weights $w_{i,j}$ biases and coefficients $a_{i,n}$ are adaptive and are optimized in order to minimize the output error. Our proposition consists in incorporating a time delay component on the inputs nodes of each layer and then equation (1) becomes:

$$y_i(t) = f\left(bias_i + \sum_{m=0}^{L} \sum_{j=1}^{P} w_{i,j,m} x_j(t-m) \right) + \sum_{n=1}^{M} a_{i,n} y_i(t-n) \qquad (2)$$

Where L is the delay order at the input.

Feedforward and feedback weights were initialized from a uniform distribution in the range [-0.8, 0.8]. A neuron of the AR-TDNN configuration is shown in Figure 1.

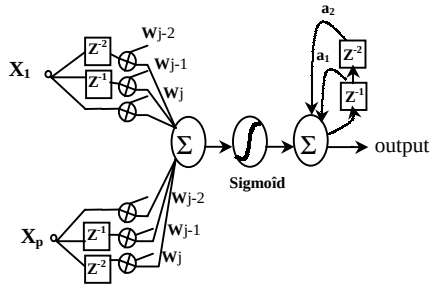

Figure 1. A neuron from a hidden layer of an autoregressive time delay neural network

AR backpropagation learning algorithm performs the optimization of feedback coefficients in order to minimize the mean squared error noted *E(t)* and defined as:

$$E(t) = \frac{1}{2}\sum_{i}\left(d_i(t) - y_i(t)\right)^2 \qquad (3)$$

Where d_i is the desired value of the ith output node.

The weight and feedback coefficient changes noted respectively $\Delta w_{j,i,m}$ and $\Delta a_{i,n}$ are accumulated between an update interval $[T_0,T_1]$. The calculation of this variations is detailed in [9]. In the proposed AR-TDNN version, the update interval $[T_0,T_1]$ is fixed such as it corresponds to the time delay of the inputs. So, it T if the frame duration, then the updated feedback coefficient is written as follows:

$$a_{i,n}^{new} = a_{i,n}^{old} + \frac{1}{L.T}\sum_{t=T_0}^{T_1}\Delta a_{i,n}(t) \qquad (4)$$

The weights are also written :

$$w_{i,n}^{new} = w_{i,n}^{old} + \frac{1}{L.T}\sum_{t=T_0}^{T_1}\Delta w_{i,n}(t) \qquad (5)$$

Hence, this type of networks which combines both input delays and feedback of outputs can "remember" past situations and performs context sensitive decisions. We must recall that AR-TDNN system is trained by using the Nguyen-Widrow initialization conditions [8].

The TDNN part of the system consists of three layers. Each neuron in the first hidden layer receives input from the coefficients in the three-frame window of the input layer. The input is centered around the hand-labeled phones.

Experiments using approximately 6000 phonemes uttered by six speakers are carried out in order to compare the performances between the AR-TDNN configuration and the monolithic (simple) connectionist structure. The monolithic architecture performs recognition of all macro classes and features by a simple neural network using the standard backpropagation learning procedure.

The results given in Table 1 show that in the case of complex Arabic macro-classes the AR-TDNN with a recognition rate average of 86% overpasses significantly the standard backpropagation-based system with globally 70% recognition rate.

Class / System	Long/brief Vowels	Plos.	Fri.	Nas.	Liq.	Emp.	Gem.
Simple NN	61.3	75.7	77.2	73.4	71.8	70.0	60.9
AR-TDNN	91.9	83.2	89.1	83.6	79.1	84.2	88.8

Table 1. Recognition rate (%) of simple and AR-TDNN structures. (Vow: Vowel, Plos: Plosives, Fri: Fricatives, Nas: Nasals, Liq: Liquids, Emp: Emphatic, Gem: Geminate).

A significant difference of accuracy in favor of AR-TDNN is observed in the case of semantically relevant lengthening of phoneme. This experiment confirms the capability of the AR-TDNN configuration to deal with features as subtle as emphasis, gemination and vowel extension.

3. OPTIMAL USE OF LVQ TRAINING DATA

LVQ is a nearest neighbor pattern classifier based on competitive learning and it provides an important gain in learning speed in comparison with neural networks.

In a speech application, the principle is that each phoneme category to be learned is assigned a number of reference vectors having the same dimension as the input vector. In the learning phase, LVQ attempts to adapt the positions of the reference vectors such as each input vector has a reference vector of the right category as its closest reference vector. In the recognition step, the unknown input vector will be categorized by finding the reference vector that is closest to that input vector.

Let X(t) be a sample of speech sequence and let V(t) represents a number of codebook vectors which approximate the input vector X(t) by its quantized values. Many codebook vectors can be assigned to each class of input vectors. X(t) belongs to the same class to which belongs its nearest V(t) noted $V_i(t)$. The *i* index represents the class.

Starting with defined initial values by using the K-means clustering, the following equations define the optimized LVQ1 (OLVQ1):

- $V_i(t+1) = V_i(t) + \alpha_i(t)\,[X(t) - V_i(t)]$ if X and $V_i \in$ same class
- $V_i(t+1) = V_i(t) - \alpha_i(t)\,[X(t) - V_i(t)]$ if X and $V_i \notin$ same class
- $\alpha_i(t) = \alpha_i(t-1)/(1 + z(t)\alpha_i(t-1))$

z(t)=+1 if the classification is correct and –1 otherwise.

In the case of the optimized learning vector quantization (OLVQ1), the basic LVQ1 algorithm is modified in such a way that an individual learning rate $\alpha_i(t)$ is assigned to each free parameter input vectors [7]. The basic LVQ1 is optimized in order to determine $\alpha_i(t)$ for fastest possible convergence of the above equations. It must be warned that $\alpha_i(t)$ does not rise above the value 1.

- *Optimized rules for OLVQ1 learning*

In order to involve better use of training data, we propose to proceed OLVQ1 algorithm with the optimal rules given in Figure 2. This approach is supported by the principle consisting in extracting the maximum amount of information with only a small sample of task-specific data. This principle is widely used by recent work in cache-based language modeling [3].

This process consists to split the initial learning database in a defined number of subsets. A random procedure performs the distribution of hand-labeled sequences over the subsets. The number of subsets is chosen in relation with the number and quality of sequences of the initial learning database. For each subset, one would select a representative sample of patterns from each class. The OLVQ1 is consecutively applied over the subsets. Then, at each step, a temporary codebook is created through the use of a test database which permits to measure the efficiency of reference vectors. Only the reference vectors of phonemes recognized with higher scores are retained and stored in the final codebook. The phoneme sequences having obtained relatively bad results are mixed with remaining learning sequences belonging to the next subset. In this way, another attempt to find more representative codebook is carried out with more training data.

1. **Proceed** splitting the training database into ***N*** subsets
2. **Init** by using K-means the codebook vectors
3. **Select** one subset S_i and put it in a temporal training set
4. **If** number of subsets not elapsed **then**
   apply OLVQ1 over temporal training set
   test ASRS on subset S_j $(j \neq i)$
   feed M-best code vectors in the final code book
   add remaining elements of S_i to the temporal training set
5. **If** number of final codebook vectors reached
   then goto 6 **else** increment *i* **and goto** 3
6. **test** data using optimal codebook, **End**

Figure 2. Optimized rules for the application of OLVQ1 algorithm to speech data

Learning and test phases are alternated, and Figure 3 shows results at consecutive stages of training corresponding to the successive use of subsets. Notice that the recognition accuracy is first improved significantly until an optimum is reached; after that, when learning is continued, the accuracy starts to stabilize slowly. We believe that the ability of the algorithm to generalize for new data is increased compared to the classical way which consists to learn in one pass (dash line in Figure 3). Thus, The main advantage of this approach resides in the fact that learning process is tuned towards complex phonemes which are generally the root of ASR performances drop. It is worthy of note that the use of more cluster units improves performance. In the presented case where 10 subsets are used, a difference of 5% over approximately 6000 phonemes is observed in favor of the learning using optimized rules.

- *Selection of M-best reference vectors*

The selection criterion of reference vectors to put into the temporal codebook depends on the recognition rate and on the medians of the shortest distances between codebook vectors in each class. Empirical thresholds related to these two parameters are determined. These thresholds permit us to perform a pruning procedure in order to select the best reference vectors. Each class whose the codebook vectors have obtained a recognition rate which overpasses a given empirical threshold (95%-98%) and having its global distance under the average median is retained for the composition of the final codebook. The size of the final codebook is about 200.

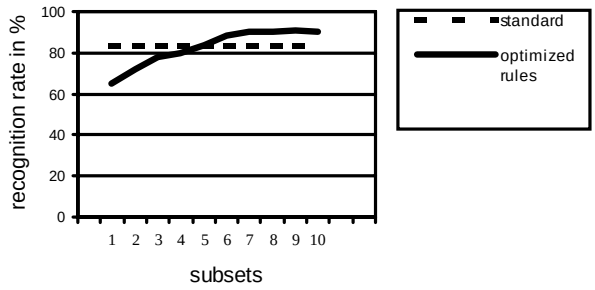

Figure 3.The subset number effect over OLVQ1 recognition rate

- *Overlearning and stopping rule*

The classical problem of 'overlearning' i.e., when The codebook vectors become very specifically tuned to the training data, is avoided by stopping the learning process after some 'optimal' number of steps. Such a stopping rule can only be found by experience, and it also depends on the input data. Let us recall here that Kohonen [7] proposes that OLVQ1 may generally be stopped after a number of steps that is of 50 times the number of codebook vectors.

4. HYBRID LVQ/TDNN SYSTEM

Mc Dermott and Katagiri [5] performed an interesting comparison between Waibel's TDNN and Kohonen's LVQ algorithm using the same database and similar conditions. The LVQ system achieves roughly the same error rate as the TDNN, but LVQ was much faster during training, slower during testing and requires more memory than the TDNN.

In hybrid system that we present, illustrated in Figure 4, a combination of the two algorithms is proposed. This combination is justified by the fact that designers of systems dedicated to the Arabic language are unanimously observing that the well known ASR systems have not the ability to deal with emphasis, gemination and relevant vowel lengthening [4][11]. Hence, The drawback of traditional classifiers applied individually, leads us

to develop such a hybrid system, more adapted to the Arabic language particularities. The OLVQ1 algorithm applying optimal rules is used but we it will be noted LVQ for simplicity considerations.

4.1. Description of the Hybrid System

Training the LVQ/AR-TDNN on an utterance proceeds in two steps. The first step performs optimal alignment between the acoustic models of phones and the speech signal. In the second step the AR-TDNN system acts as post-processor to OLVQ1 and refines its recognition results. The global task is then divided between the main system constituted by OLVQ1 and the "booster" system composed of AR-TDNN. We require OLVQ1 to achieve phone identification without discriminating between long and short vowels and between emphatic and non emphatic consonants. The gemination detection is also not required. The hand-labeled data set presented to OLVQ1 presents a single label for phonemes belonging to these macro-classes. For instance, in the case of /a/ short vowel and /a:/ long vowel, a unique /A/ label is given. The /A/ sequence of phones is presented to the AR-TDNN system which makes final and finer decision related to the long/short vowel discrimination.

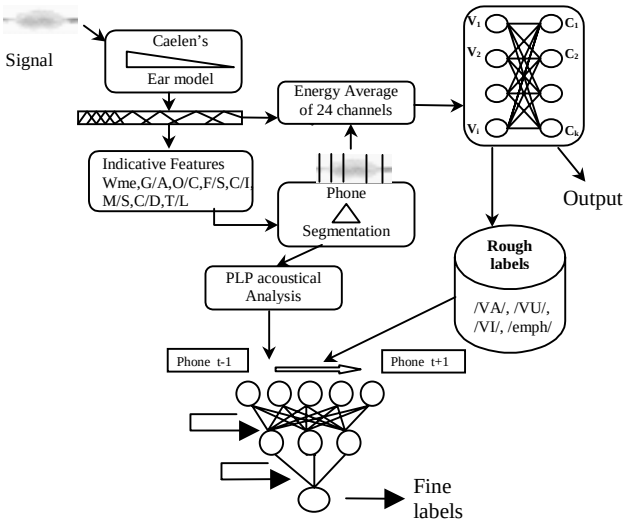

Figure 4. LVQ/TDNN hybrid structure for Arabic speech recognition.

Because of the importance of the phonetic context for performing phoneme identification, a careful analysis must be done for selecting the learning set. The supervision of this learning considers phones as entire items. The co-articulation effect makes this supervision difficult. The adopted solution consists in executing the learning phase such as if a phone of the target-phoneme appears in the speech continuum, the AR-TDNN activation arises gradually in the output. In the example of geminated consonants detection/classification, the task consists in learning to recognize the following sequence: LCG-GEM-RCG: LCG is the left phonetic context of the geminated consonant (noted GEM) and RCG is its right phonetic context.

GEMI_NET (gemination expert network) receives three input token at a time t and it must detect a geminated sequence from any other sequence combination. The learning proceeds in the setting at the high level (+1) the output when the end of the LCG-GEM-RCG sequence is attained. Low level (-1) is set otherwise i.e. if a scrolling (stream) of non-geminated phone sequences is observed. An autoregressive order of 2 is chosen and a delay of 2 phones is also fixed. These lower values of delay and order are justified by the fact that phones are used instead frames. Consequently the stability of AR nodes is ensured. Besides of GEMI-NET system, two other AR-TDNN-based expert systems are provided: DURA_NET and EMPHA_NET. They respectively perform long/short vowel discrimination and Emphatic/Non-Emphatic opposition detection. These tasks are accomplished according to the same protocol conducted by GEMI_NET.

4.2. Acoustic Attributes and Segmentation Strategy

As it is shown in Figure 4, two auditory models are used: Caelen ear model [2] for homogenous phone segmentation and PLP (perceptual linear predictive) [6] technique as acoustical analyzer for AR-TDNN. The choice of auditory models is justified by the robustness they involved to ASR systems [3].

Cues derived from hearing phenomena studies are extracted thanks to the Caelen ear-model [2]. In this model, the internal ear is represented by a coupled filter bank where each filter is centered on a specific frequency. The filters number can be limited to 24 covering a 16Hz-12000Hz frequency range. The 24-channel spectrum obtained in the output of the 24 coupled filters can be used directly as input data.

Furthermore, from a particular linear combination of the outputs of these channels, 7 cues are derived: acute/grave (AG), open/close (OC), diffuse/compact (DC), sharp/flat (SF), mat/strident (MS), continuous/discontinuous (CD) and tense/lax (TL). We showed in [10] that these indicative features are very relevant to characterize the Arabic phonemes. According to the procedure described in Figure 5, a delta coding of the acoustic indicative features is done in order to find out their variation. A function which is the sum of absolute outputs of delta coders is evaluated. It quantifies in such a way the discontinuity between two successive frames. If this amount is over a threshold, which is variable in time, a mark is attached to the current frames. The frames between two successive marks make an homogeneous phone. For each phone, the log-energy of the 24 channels outputs are used as parameters by OLVQ1 system.

Δ coding procedure
R=0 and Delta empiric value
X_n: indicative feature
Z_n :encoded indicative feature
n: frame sample

For all frames Do
$Z_n=X_n-R/Delta$
If $(X_{n-1}-X_{n-2})*(X_{n-1}\ X_n)\geq 0$ then
$R=X_{n-1}$
If $(Z_n <> 0)$ then $R=X_n$
EndDo

Figure 5. The principle of phone segmentation

Over each phone, an average of PLP coefficients is calculated and used as input to the AR-TDNN experts. This type of acoustical parameters have been retained because it gives the best cross-validation results as it is shown in [10]. This preference is also justified by a gain in the learning time because of the fact that the PLP analysis does not require a great number of vector components.

5. EXPERIMENTAL RESULTS

We compare hybrid system performances to a baseline LVQ-based system. The results obtained by the two systems are presented in Figure 6. These results concern 60 VCV utterances and 50 phrases. This corpus has been pronounced by six Algerian native speakers (3 men and 3 women). As a whole, the test concerns 3724 vowels (1348 long), 1197 fricatives (182 geminated, 193 emphatics), 1089 plosives (215 geminated, 273 emphatics), 573 nasals and 413 liquids. The semi-vowels are assimilated to their corresponding vowels.

The analysis of the results revealed that hybrid configuration is more accurate in all cases of complex phonemes. We found that this system achieved 87% accuracy, which represents 8% fewer errors than the OLVQ1 baseline system which obtained 79% mean correct rate. Concerning the standard OLVQ1, we noticed that even if it was relatively effective to detect short vowels, it failed dramatically in the detection of long vowels. An unbalance of performances which can reach 20% for the case of /a:/ vowel is observed. The same phenomenon is observed in the emphasis detection case where the difference for the /s̲/ consonant is about 15 % in favor of hybrid system. The redundancy of this trend leads us to conclude, as it was expected, that standard OLVQ1 is not capable to perceive relevant phoneme duration changes and emphasis feature. In contrast, the hybrid system achieved successfully this task.

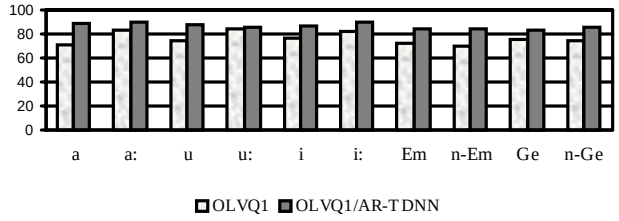

Figure 6. Results of LVQ-based baseline system and LVQ/AR-TDNN hybrid system for long and short vowels, emphatic and geminated consonants.

Either monolithic or hybrid architectures realize mediocre scores in the particularly case of plosives. The omission percentage is the highest for the case of glottal and pharyngeal consonant (/h/, /ħ/, /q/, /γ/, /ε/, /?/). It is certainly due to their shortness and their sensibility to the utterance speed. The co-articulation effects make them merged into the vocalic context. We have noticed the failing of the two systems in the identification of the emphatic feature for the /d̲/ consonant. The explanation does not reside in the difficulty inherent to this consonant acoustical proprieties, but rather in the capability of the speaker to pronounce it correctly. In fact, in a VCV context, it is very difficult to keep the emphatic character of /d̲/ and more often, it is its opposite by this feature (/d/) which is achieved. This defect is mainly due to the characteristic of the Algiers regional accent. In the case of the /ð/ and /ð̲/, We must notice that although these two consonants are considered as fricatives, in the case where they are geminated, they are detected as plosives. We have also remarked that when a conjunction of gemination and emphasis is realized, the hybrid system totally succeeds to make feature discrimination while the standard system fails in all cases.

6. SUMMARY

A completely connectionist hybrid approach for Arabic speech recognition is presented. Our objective was to test on Arabic language, the ability of a system combining original versions of Linear Vector Quantization algorithm and Time Delay Neural Networks to detect features as subtle as gemination, emphasis and relevant lengthening of vowels. This hybrid system has been confronted to a baseline LVQ-based system. Regarding obtained results, it seems clear that the proposed approach improves significantly performances in all cases and its generalization can easily be considered to the all known ASR systems. The split of the global speech recognition task into subtasks assigned to more adapted systems i.e. a specific task assigned to a specific system, constitutes from our point of view a powerful and promising way to radically solve problems encountered by the automatic speech recognition systems dedicated to Arabic.

7. REFERENCES

[1] El-Ani S.H, *Arabic phonology: an acoustical and physiological investigation*, Mouton ed., the Hague, 1970.

[2] Caelen J., *un modèle d'oreille, analyse de la parole continue*, doctorat ès sciences thesis, Toulouse, 1979.

[3] R.Cole et al, "The Challenge of Spoken Languages Systems: Research Directions for the Nineties", *IEEE trans. on speech and audio processing* 3(1): 1-20, 1995.

[4] Djoudi M., Fohr D., Haton J.P., "Phonetic study for automatic recognition of Arabic", *European Conference on speech and technology*: 268-271, 1989.

[5] Mc Dermott E. and Katagiri S., "LVQ-Based Shift-Tolerant Phoneme Recognition", *IEEE trans. on signal processing*, 39(6):1398-1411, June 1991.

[6] Hermansky H., "Perceptual linear predictive (PLP) analysis of speech", *Jour. Ac. Soc. Amer*. 87 (4): 1738-1752, 1990.

[7] Kohonen T., *Self-Organizing Maps*, Springer-Verlag, Heidelberg, 1995.

[8] Nguyen D., Widrow B., "Improving the learning speed of two-layer neural networks by choosing initial values of the adaptative weights", *IJCNN, III*: 21-26, San Diego, 1990.

[9] Russel R.L. and Bartley C., "The autoregressive backpropagation algorithm", *IJCNN, Vol II*:369-377, 1991.

[10] Selouani S.A. and Caelen J., "Recognition of phonetic features using neural networks and knowledge-based system", *3rd IEEE Symposium on Intelligence and systems*: 404-411, Washington D.C., May 1998.

[11] Selouani S.A., and Caelen J., "Experiment in automatic speech recognition of standard Arabic", *Proceedings of KFUPM workshop* :161-171, Dahran, 1996.

[12] Waibel A., Hanazawa T., Hinton G. and Shikano K., "Phoneme recognition using time-delay neural networks", *IEEE trans. ASSP* 37: 328-339, 1989.