Recognition of Spoken Arabic Digits Using Neural Predictive Hidden Markov Models

Rafik Djemili, Mouldi Bedda, and Hocine Bourouba
Automatic and Signals Laboratory of Annaba, Badji Mokhtar University, Algeria

Abstract: *In this study, we propose an algorithm for Arabic isolated digit recognition. The algorithm is based on extracting acoustical features from the speech signal and using them as input to multi-layer perceptrons neural networks. Each word in the vocabulary digits (0 to 9) is associated with a network. The networks are implemented as predictors for the speech samples for a certain duration of time. The back-propagation algorithm is used to train the networks. The hidden markov model (HMM) is implemented to extract temporal features (states) for the speech signal. The input vector to the networks consists of twelve mel frequency cepstral coefficients, log of the energy, and five elements representing the state. Our results show that we are able to reduce the word error rate comparing with an HMM word recognition system.*

Keywords*: Speech recognition, hidden Markov models, artificial neural networks, hybrid HMM/MLP.*

Received September 15, 2003; accepted January 19, 2004

1. Introduction

Hidden Markov models (HMMs) [1, 32] are widely used for automatic speech recognition and have proved useful in dealing with the statictical aspects of the speech signal. These models are optimal in the sense that the distributions of probability of the studied pattern clusters are known, their classification by a bayesian method will give a minimal error rate. In practice, these distributions must be estimated from a large number of training data. But, when the volume of these data is limited, it would be interesting to turn to the connectionist methods, which have the ability of generalization from incomplete data. In addition, the stochastic models present some limitations, in particular due to the restrictive assumptions in general introduced into the associated algorithms of optimization. On the other hand, artificial neural networks appear useful for the classification of static forms, while being weak with regard to the treatment of the temporality of speech signals. An example of connectionist architectures and more Particularly Multilayer Perceptrons (MLP) used in pattern classification are [22, 26]. Thus, it seems interesting to try to combine the respective capacities of hidden Markov models and neural networks to produce new powerful hybrid models which draw their source in the two formalisms, several authors have proposed original architectures of this type [10, 20, 21, 28, 33, 37].

Our study investigates such combination inspired from [7] and applied for a specified task concerned with Arabic isolated digit recognition.

The organization of this paper is as follows, in section 2 we review the acoustic modeling used in our experiments and the theory of hidden Markov models. In section 3, we discuss some aspects of artificial neural networks, we also remind the implementation of back propagation algorithm, the mostly used in training multilayer perceptrons. Following this, section 4 explains existing hybrid systems based on neural networks and involving hidden Markov models, and the manner with which they were succesfully applied to problems in speech recognition. In section 5, we will discuss more amply the hybrid algorithm used, the choice of certain crucial parameters of our work, as well as procedure suggested and included in the algorithm of training neural networks, we will also give the results obtained on data bases built for this purpose. Finally, the last section will give a summary of our work and possible perspectives.

2. Hidden Markov Models

2.1. Acoustic Modeling

The goal of acoustic modeling is to derive some convenient representation of speech signals before their use in a speech recognition system. Hence each speech signal in the current study, is sampled at 22050 Hz decimated at a rate of 11025 Hz, passes through a high frequency preemphasis filter with a transfer function $H(z)= 1-az^{-1}$. The preemphasized data is blocked into overlapping frames. Each frame is 23.2 ms duration, with 11.6 ms spacing. Spectral analysis is performed to get twelve Mel Frequency Cepstral Coefficients (MFCC) [13] and the log of the energy calculated in the temporal domain. The first twelve MFCC are obtained from the energies of F bank filters directly using the DCT transform:

$$c_k = \sum_{i=1}^{F} \log E_i \cos\left[\frac{pk}{F}(i-\frac{1}{2})\right] \qquad 1 \le k \le d \qquad (1)$$

The overall system is depicted in Figure1.

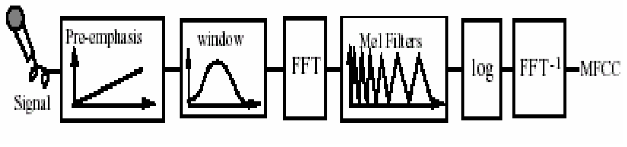


Figure 1. Analysis of speech frames.

2.2. Presentation of Hidden Markov Models

In this subsection, we remind the basic definition of an HMM, we formalize the assumptions that are made and describe the basic elements of algorithms for HMMs, we use the notation as in [31].

A hidden Markov model can be defined as a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden) but can only be observed through another set of stochastic processes that produce the sequence of observations.

A Markov model of order k is a probability distribution over a sequence of variables $q_1^t = \{q_1, q_2, ..., q_t\}$ with the following conditional independence proprety:

$$p\left(q_t \middle| q_1^{t-1}\right) = p\left(q_t \middle| q_{t-k}^{t-1}\right) \qquad (2)$$

Since q_{t-k}^{t-1} summarizes all the relevant past information, q_t is generally called a state variable. Because of the above conditional independence property, the joint distribution of a whole sequence can be decomposed into the product:

$$p\left(q_1^T\right) = p\left(q_1^k\right) \prod_{t=k+1}^{T} p\left(q_t \middle| q_{t-k}^{t-1}\right) \qquad (3)$$

The special case of a Markov model of order 1 is the one used in our study. In this case, the distribution is even simpler:

$$p\left(q_1^T\right) = p(q_1) \prod_{t=2}^{T} p\left(q_t \middle| q_{t-1}\right) \qquad (4)$$

and it is completly specified by the so-called initial state probabilities $p(q_1)$ and transition probabilities $p(q_t|q_{t-1})$.

An HMM is characterized by the following five elements:

1. N: The number of states in the model.
2. M: The number of distinct observation symbols per state, we denote an observation sequence by: $o = \{o_1, o_2, ..., o_T\}$
3. The state transition probability distribution $A = \{a_{ij}\}$ where:

$$a_{ij} = prob[q_{t+1} = S_j / q_t = S_i] \quad 1 \le i, j \le N \qquad (5)$$

i. e: The probability of being in state S_j at time t+1 and in state S_i at time t.

4. The observation symbol probability distribution in state j, $B = \{b_j(k)\}$ where:

$$b_j(k) = prob[v_k \ at \quad t / q_t = S_j] \quad 1 \le k \le M \qquad (6)$$

i. e: The probability of observing the symbol v_k at time t in the state S_j.

5. The initial state distribution p= {p$_i$} where

$$p_i = prob(q_1 = S_i) \quad 1 \le i \le N \qquad (7)$$

i. e: The probability of being in state S_i at time t= 1.

For convenience, we use the compact notation $l = (A, B, p)$ to indicate the complete parameter set of the model.

2.3. Viterbi Algorithm

To find the single best state sequence *Q= {q$_1$, q$_2$,…., q$_T$}* for a given observation sequence *o= {o$_1$, o$_2$,…, o$_T$}*, we use a formal technique based on dynamic programming methods, and called the Viterbi algorithm. We first define the quantity:

$$d_t(i) = \underset{q_1 \cdots q_T}{Max} \quad p[q_1....q_t = i, o_1....o_t | l] \qquad (8)$$

i. e: $d_t(i)$ is the best score (highest probability along a single path, at time t, which accounts for the first t observations and ends in state S$_i$. By induction we have:

$$d_{t+1}(j) = \left[\underset{i}{max}\, d_t(i) a_{ij}\right] b_j(o_{t+1}) \qquad (9)$$

To actually retrieve the state sequence, we need to keep track of the argument which maximized (9) for each t and j. We do this via the array $y_t(j)$. The complete procedure for finding the best sequence can now be stated as follows:

Initialization

$$d_1(i) = p_i b_i(o_1) \qquad 1 \le i \le N \qquad (10)$$
$$y_1(i) = 0$$

Recursion

$$d_t(j) = \underset{1 \le i \le N}{Max}\left|d_{t-1}(i) a_{ij}\right| b_j(o_t) \qquad 2 \le t \le T, \quad 1 \le j \le N$$
$$y_t(j) = \underset{1 \le i \le N}{Arg\,max}\left|d_{t-1}(i) a_{ij}\right| \qquad 2 \le t \le T, \quad 1 \le j \le N \qquad (11)$$

Termination

$$p = \underset{1 \le i \le N}{Max}[d_T(i)]$$
$$q_T = \underset{1 \le i \le N}{Arg\,\max}[d_T(i)] \qquad (12)$$

Path (state sequence) backtracking

$$q_t = y_{t+1}(q_{t+1}) \qquad t = T-1, T-2, ..., 1 \qquad (13)$$

2.4. HMM Training Algorithm

The training procedure (Figure 2) is a variant of a well known K-means iterative procedure for clustering data. We assume we have a training set of observations, and an initial estimate of all model parameters. However, unlike the one required for reestimation, the initial estimate can be chosen randomly, or on the basis of any available model which is appropriate to the data.

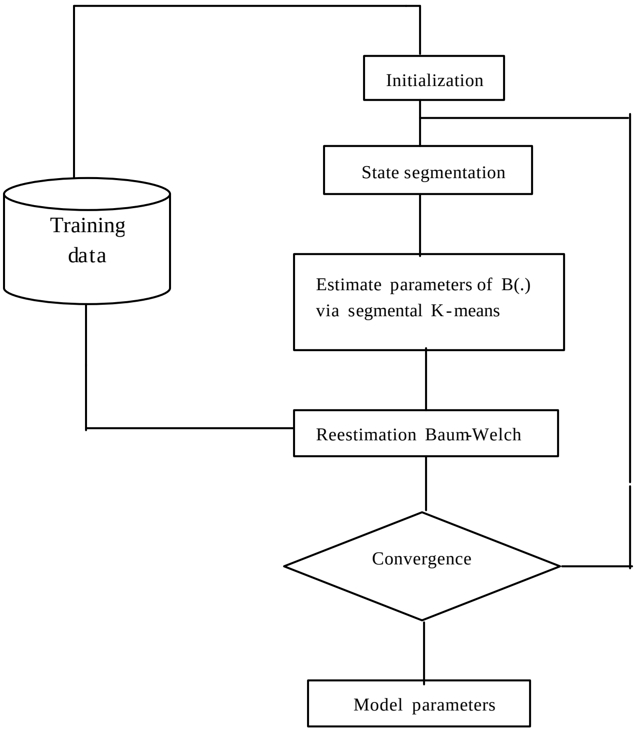


Figure2. The segmental k-means training procedure to estimate reliable HMM parameters.

Following model initialization, the set of training observations sequences, is segmented into states, based on the current model l . This segmentation is achieved by finding the optimum state sequence via the Viterbi algorithm, and then backtracking along the optimal path. The result of segmenting each of the training sequences is, for each of the N states, a maximum likelihood estimate of the set of the observations that occur within each state S_i according to the current model. An updated estimate of the $b_j(k)$ parameters is: $b_j(\hat{k})$= number of vectors with codebook index k in the state j divided by the number of vectors in state j.

Based on this segmentation, updated estimates of the a_{ij} coefficients can be obtained by counting the number of transitions from state i to j and dividing it by the number of transitions from state i to any state (including itself). An updated model $\hat{l}$ is obtained from the new model parameters and the Baum-Welch equations are used to estimate all model parameters. The resulting model is then compared to the previous model (by computing a distance score that reflects the statistical similarity of the HMMs). If the model distance score exceeds a threshold, the old model l is replaced by the new model $\overline{l}$, and the overall training loop is repeated. If the model distance score falls below the threshold, then the model convergence is assumed and the final model parameters are saved.

3. Artificial Neural Networks

3.1. Neural Networks Basic Definition

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. An illustration of one element is shown below in Figure 3.

The essential element is called neuron or node for its operation founded on that of an automat proposed as an approximation of the operation of a biological neuron [11]. The output of this cell is a nonlinear function of the weighted sum of its entries. A very current analytical form for the decision is the sigmoid function, but other functions can also be used. The topology of the network, i.e. the way in which the cells are inter-connected, is an essential characteristic of such a network.

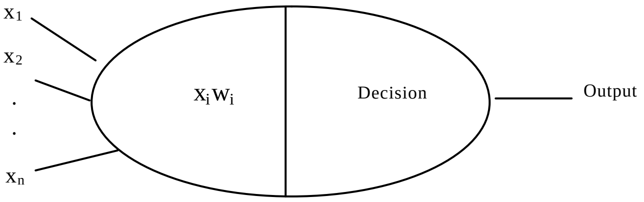


Figure 3. Artificial neuron.

3.2. The Multilayer Perceptron

The architecture of neural networks employed in our work and widely used elsewhere is the MultiLayer Perceptron (MLP). As indicated by its name, MLP consist of several layers of neurons (nodes) connected to each other. In this kind of topology, the input vector feeds into each of the first layer perceptrons, the outputs of this layer feed into each of the second layer perceptrons and so on (Figure 4). Often the nodes are fully connected between layers, i.e. every node in layer l is connected to every node in layer l+1.

Thus we refer to the architecture in Figure 4 as a two layer network. It is also common to specify an architecture by referring to the number of hidden layers, that is layers that are neither inputs nor outputs. Thus, the network in the figure is also referred to as a one hidden layer network.

When dealing with multilayer perceptrons, a problem we are faced with is how to choose the number of layers, for this purpose, Lippman demonstrated that a two layer perceptron can implement arbitrary convex decision boundaries [26], later it was shown that a two layer network can perform an arbitrarily close approximation to any nonlinear decision boundary [27].

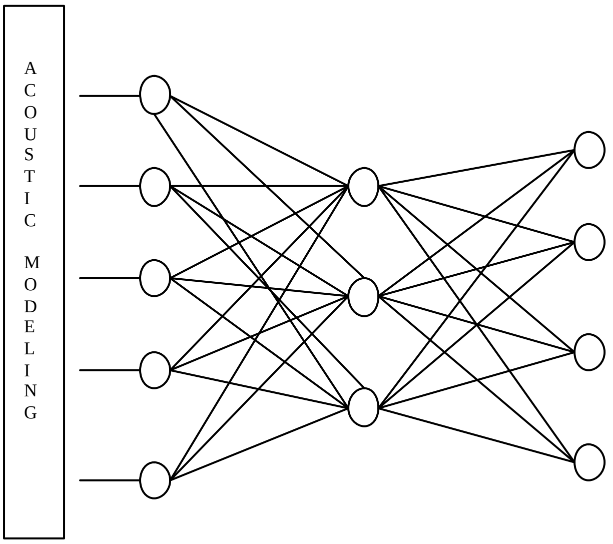


Figure 4. Architecture of a typical multilayer perceptron.

It has also been shown that a two layer perceptron is capable of forming an arbitrarily close approximation to any continuous nonlinear mapping [12]. Following this discussion, we used in our work a two layer perceptrons.

Commonly neural networks are adjusted or trained, so that a particular input leads to a specific target output. Such a situtation in which the network is adjusted based on comparison of the output and the target until the network output matches the target, is called supervised learning, we will see an algorithm for training supervised multilayer networks.

3.3. MLP Learning Algorithm (Backpropagation)

Backpropagation was created by generalizing the Widrow-Hoff learning rule to multilayer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding output vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an approximate way defined by the user.

Networks with biases, a sigmoid layer and a linear output layer are capable of approximating any function with a finite number of discontinuities. Standard backpropagation is a gradient descent algorithm [34], the term backpropagation refers to the manner in which the gradient is computed for nonlinear multilayer networks.

The simplest implementation of backpropagation learning, updates the network weights and biases in the direction in which the criterion function decreases most rapidly. Before we give main formulas used in the implementation of the algorithm, let us introduce the following notations [19]:

$u_{l,j}$: Output of the j^{th} node in layer l.

$w_{l,j,i}$: Weight which connects the i^{th} node in layer l-1 to the j^{th} node in layer l.

x_p: p^{th} training sample.

$u_{0,i}$: i^{th} component of the input vector.

$D_j(x_p)$: Desired response of the j^{th} output node for p^{th} training sample.

N_l: Number of nodes in layer l.
L: Number of layers.
P: Number of training patterns.
The output of a node in layer l is given by:

$$u_{l,j} = f\left(\sum_{i=0}^{N_{l-1}} w_{l,j,i}u_{l-1,i}\right) \quad (14)$$

Where f(.) is the sigmoid nonlinearity.

The most common learning algorithm for the MLP uses a gradient search technique to find the network weights and biases that minimize a criterion function. The criterion function to be minimized is the sum of squared error criterion function:

$$J(w) = \sum_{p=1}^{P} J_p(w) \quad (15)$$

$J_p(w)$ is the total squared error for the pth pattern:

$$J_p(w) = \tfrac{1}{2}\sum_{q=1}^{N_L}(u_{L,q}(x_p) - d_q(x_p))^2 \quad (16)$$

The weights of the network are determined iteratively according to:

$$w_{l,j,i}(k+1) = w_{l,j,i}(k) - m\sum_{p=1}^{P}\frac{\partial J_p(w)}{\partial w_{l,j,i}}\Big| w(l) \quad (17)$$

where μ is a positive constant called the learning rate. equations 14-17 summarize one epoch of the backpropagation learning algorithm, these equations have to be repeated until termination condition is reached, we will discuss conditions applied in our study in section 5.

4. Overview of Hybrid Systems in Speech Recognition

In order to overcome the unsatisfying performance of speech recognition systems based on neural networks formalism, researchers have attempted to combine connectionist models with non connectionist tools, especially hidden Markov models very useful in speech recognition [23, 25].

Hence, several researchers have explored such hybrids, the majority of which are constructed by sending the output of a NN to a HMM post-processor [3, 8, 16]. Several others propose a NN architecture that can emulate a HMM [9, 29], alternatively [36] uses the NN to rescore the N best hypothesis produced with a HMM.

In [18, 36] the outputs of the NN are not interpreted as probabilities, but rather are used as scores and generally combined with dynamic programming. On the other hand, [8, 17, 28, 29] interpret the outputs of the NN as posterior probabilities and use the Viterbi algorithm.

The systems proposed in [2-4] combine a NN with a HMM. In [2] the outputs of the NN are quantized, hence no global optimization of the hybrid can be performed. In the system introduced in [3, 4] a global optimization of all the parameters of the hybrid is performed by backtracking the derivative of the training criterion from the HMM to the NN.

Another kind of hybrids is in [24], a network per class or per state is trained to predict the next input frame given only a few previous frames, the difference between predicted and actual input is used to compute a state conditionnal observation likelihood, the Viterbi algorithm is used to obtain a segmentation and train the network. Hence, the system is trained to maximize the likelihood of the observation given the predictive network model.

The system in our study, described more in next section similar in part to [24], uses a network per unit HMM, and since in most isolated word recognition systems, the unit is a word model then we built a network for each word in the vocabulary.

5. Hybrid HMM/MLP Algorithm

5.1. Choice of Model Parameters

Before introducing the hybrid algorithm used, it seems useful to bring back here certain details inherent in the application of technologies of HMMs and NNs in a real process. Thus for the HMM, the number of states N and the size of the codebook were fixed respectively at 5 and 64, continuing our own investigation on the application of hidden Markov models in recognition of Arabic isolated digits [14].

For more details on the HMM, we invite the reader towards the following references [5, 6, 31, 35]. Concerning neural networks, we have fixed the learning rate m at 0.01 and at 9 the number of neurones in the hidden layer for computational considerations as in [15].

5.2. Hybrid Algorithm

In this algorithm, the neural network is implemented as a predictor [7]. It predicts the observation vector of the next frame given the observation vector of the current frame and the HMM state to which this current frame belongs to. The Viterbi algorithm is used to determine this state. The multilayer perceptron implemented in this study, is constitued with eighteen input nodes, thirteen nodes to represent the observation vector and five nodes to represent the five states. The state is indicated by entring a value of "1" at the corresponding node and entring zeros at other nodes. For example, state 2 is represented as 01000 while state 4 is represented as 00010. The network has nine hidden nodes and thirteen output nodes. The back - propagation training procedure is implemented to train the network. For each trained word there is a network. To recognize a word, its observation vector and the state of each frame are introduced to every networks. A prediction error is calculated for each frame using the mean square error. This error represents the difference between the predicted values of the observation vector of the frame and the actual ones. This error is summed over all frames of the word to obtain the total error for each network. The word is recognized as that of the network that gives the lowest error.

5.3. Results and Discussion

To evaluate the algorithms described above, a limited size database was recorded. The task we are dealing with is of recognizing isolated digits (0-9) in a speaker independent manner.

Hence a training set consisting of twenty occurrences of each digit by 20 talkers (i.e. a single occurrence of each digit per talker) was used. Half the talkers were male, half female.

For testing the algorithm, we used two other independent test sets with the following characteristics:

- TS-1: The same 20 talkers as were used in the training, 300 occurrences of digits (0-9).
- TS-2: A new set of 6 talkers, five occurrences per digit per talker were used, giving 300 occurrences of digits (0-9).

Also for purposes of the cross validation, we built another set of occurrences called CVS (Cross Validation Set), it consists of three occurrences per digit. A difficult aspect in implementing backpropagation learning algorithm is to automate the termination of the algorithm. Although the algorithm could be terminated either when the magnitude of the gradient is sufficiently small since by definition the gradient will be zero at the minimum, or when J falls below a fixed threshold. However, this requires some knowledge of behavior of the training step, and would yield to different values for each network model, which is not desirable for our system. So, we proposed as a stopping criterion relative error between two successive epochs, we believe this can bring a guaranted convergence of the algorithm.

$$relative \quad error = \frac{Old\ MSE - New\ MSE}{Old\ MSE} * 100\% \qquad (18)$$

We there after fixed this Relative Error (RE) successively to 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, and 10^{-6}.

Tables 1 and 2 respectively show a comparison of the word error rates for TS-1 and TS-2.

Table 1. Word error rate in percent for TS-1.

Arabic Digits	**HMM System**	**Hybrid HMM/MLP System**						
		RE= 10^{-1}	**R.E= 10^{-2}**	**RE= 10^{-3}**	**RE= 10^{-4}**	**RE= 10^{-5}**	**RE= 10^{-6}**	**C.V.**
0	0	23.34	23.34	13.34	10	13.34	13.34	13.34
1	16.67	16.67	16.67	13.34	6.67	6.67	6.67	6.67
2	3.34	3.34	3.34	6.67	6.67	6.67	6.67	6.67
3	23.34	33.34	23.34	13.34	10	10	10	10
4	6.67	23.34	23.34	6.67	0	3.34	0	3.34
5	10	10	10	10	0	3.34	0	3.34
6	3.34	3.34	3.34	3.34	3.34	3.34	3.34	3.34
7	0	73.34	23.34	6.67	3.34	6.67	6.67	6.67
8	10	23.34	23.34	10	10	10	10	10
9	3.34	16.67	16.67	16.67	13.34	13.34	13.34	13.34
Total	7.67	22.67	16.67	10	6.34	7.67	7.00	7.67

Table 2. Word error rate in percent for TS-2.

Arabic Digits	**HMM System**	**Hybrid HMM/MLP System**						
		RE= 10^{-1}	**R.E= 10^{-2}**	**RE= 10^{-3}**	**RE= 10^{-4}**	**RE= 10^{-5}**	**RE= 10^{-6}**	**C.V.**
0	13.34	23.34	36.67	26.67	10	10	10	10
1	0	16.67	0	0	0	0	0	0
2	26.67	3.34	3.34	3.34	26.67	26.67	26.67	26.67
3	40	33.34	26.67	26.67	40	40	40	23.34
4	16.67	23.34	43.34	23.34	20	16.67	20	16.67
5	3.34	10	20	20	6.67	3.34	10	3.34
6	10	3.34	36.67	26.67	10	10	10	10
7	6.67	73.34	13.34	13.34	6.67	6.67	3.34	6.67
8	3.34	23.34	13.34	13.34	6.67	3.34	10	3.34
9	10	16.67	3.34	3.34	10	10	10	10
Total	13	22.67	19.67	15.67	13.67	12.67	14	11

We notice that our best result is reached when the relative error coincides with value 10^{-4} for TS-1 and 10^{-5} for TS-2. However the phase of training using such a stopping criterion is very expensive in computational time, this is why, we tried out the method known as Cross Validation.

As depicted in Figure 5, the algorithm stops for only 49 epochs in the training of digit zero and 24 epochs in that of the digit nine.

Furthermore, this last method provides a reduction in the error rate of 1.67% concerning TS-2, which shows the good generalization of the algorithm since this test set is built from speakers who were not used for the training.

Now considering a comparison of the results obtained for systems HMM alone and the hybrid one, thus as Figure 6 in top for TS-1 shows it, the hybrid system introduces a reduction in the error rate of 1.34% into the method of the relative error but guard the same result with the method of cross validation.

Figure 6 in bottom shows the comparison for TS-2, one can note a reduction in the error rate of 0.34% when we used the criterion of the relative error and also a more significant reduction equals to 2% compared to system HMM alone, when the method of cross validation is put into consideration.

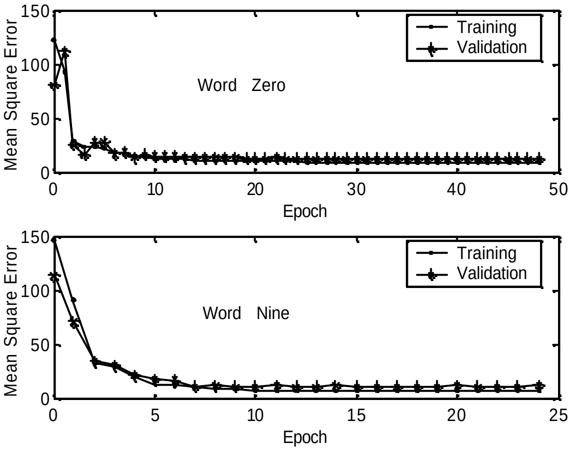


Figure 5. Plots representing the convergence of performance function when cross-validation is used as stopping criterion in training phase: In top the digit zero and bottom for the digit nine.

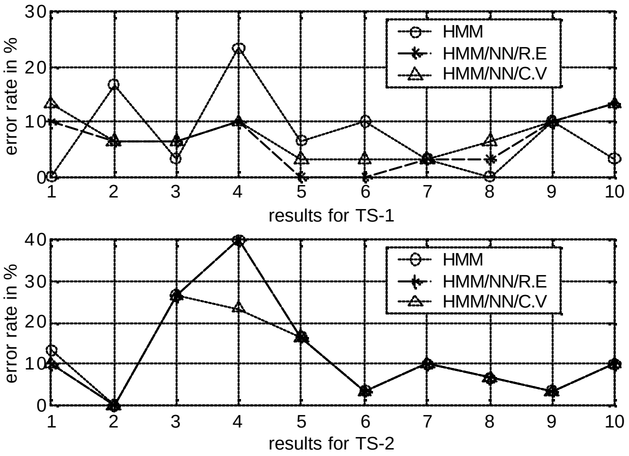


Figure 6. Plots of comparison results of the WER in (%) for systems HMM alone and hybrid HMM/NN with relative error and cross validation stopping criterion.

6. Conclusion

The work from which we have seen the details relates to a specific task in speech recognition, that is of isolated words, we treated Arabic isolated digits (0-9) forming a limited vocabulary of ten words, but all the study can be extended if one wants to use a vocabulary going until a hundred word or even a little more without modification of the algorithms.

Initially, we built hidden Markov models for each word of the vocabulary, then we use these models to segment into states all the occurrences of training data by means of Viterbi algorithm. Also we built a neural network for each word, each was implemented as pattern predictor instead of pattern discriminator.

This technique which puts neural networks in post treatment of hidden Markov models, provides satisfying results compared to the use of HMMs alone, this is indeed shown using tables 1 and 2 in which we note a reduction of the error rates for the two test sets.

In our hybrid system, one could also incorporate contextual information especially in the first layer, such as for example providing the vectors preceding and following the current vector, that will have as a preliminary consequence an increase in number of neurons of the hidden layer.

It is also interesting to study the behavior of our system in a task even more difficult, that of the recognition of connected words and to use the neural networks to rescore best hypotheses.

These two last consequences will constitute our future investigations.

References

[1] Bahl L. R., Jelinek F., and Mercer R. L., "A Maximum Likelihood Approach to Continuous Speech Recognition," *IEEE Transaction Pattern Analysis Machine Intelligence*, vol. *PAMI*- 5, no.2, pp. 179-190, 1983.

[2] Bengio Y., Cardin R., De Mori R., and Normandin Y., "A Hybrid Coder for Hidden Markov Models Using A Recurrent Neural Network," *in Proceedings International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 537-540, Albuquerque, NM, 1990.

[3] Bengio Y., "Artificial Neural Networks and their Application to Sequence Recognition," *PhD Thesis*, McGill University, Montreal, Canada, 1991.

[4] Bengio Y., "A Connectionist Approach to Speech Recognition," *to appear in the special issue of IJPRAI on Neural Nets*, 2003.

[5] Bengio Y., "Markovian Models for Sequential Data," *Neural Computing Surveys 2*, pp. 129-162, 1999.

[6] Bilmes J., "What HMMs Can Do," *Technical Report*, University of Washington, February 2002.

[7] Botros N. M., Siddiqi M., and Deiri M. Z., "Automatic Speech Recognition Using Hidden Markov Models and Artificial Neural Networks," *in Proceedings of IEEE*, pp. 1770-1775, 1993.

[8] Bourlard H. and Wellekens C. J., "Speech Pattern Discrimination and Multilayer Perceptrons," *Computer Speech and Language*, vol. 3, pp. 1-19, 1989.

[9] Bridle J. S., "Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters," *Advances in Neural Information Processing Systems 2*, in Touretsky D. S. (Ed), Morgan Kaufmann, pp. 211-217, 1990.

[10] Cohen M., Franco H., Morgan N., Rumelhart D., and Abrash V., "Hybrid Neural Network/Hidden Markov Model Continuous Speech Recognition," *in Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Banff, Canada, pp. 915-918, 1992.

[11] Cullough W. Mc. and Pitts W., "A Logical Calculus of Ideas Immanent In Nervous Activity," *Bull. Math. Biophysics*, vol. 5, pp. 115-133, 1943.

[12] Cybenko G., "Approximation by Superpositions of a Sigmoidal Functions," *Mathematics of Control Signals and Systems*, vol. 2, no. 4, pp. 303-314, 1989.

[13] Davis S. B. and Mermelstein P., "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *in Proceedings of ICASSP*, pp. 357-366, August 1980.

[14] Djemili R., "Reconnaissance de Mots Arabes Isolés Par Dynamic Time Warping & Hidden Markov Models," *Magister Thesis*, Université Badji Mokhtar Annaba, June 2001.

[15] Djemili R., Bedda M., and Bourouba H., "On Combining Artificial Neural Networks into an HMM Arabic Speech Word Recognizer," *in Proceedings of the International Arab Conference on Information Technology (ACIT'2002)*, vol. 1, pp. 349-355, Doha, Qatar, 2002.

[16] Driancourt X., Bottou L., and Gallinari P., "Learning Vector Quantization Multilayer Perceptron and Dynamic Programming: Comparison and Cooperation," *in Proceedings of the International Joint Conference on Neural Networks, IJCNN*, vol. 2, pp. 815-819, 1991.

[17] Franzini M., Lee K. F., and Waibel A., "Connectionist Viterbi Training: A New Hybrid Method for Continuous Speech Recognition," *in Proceedings of ICASSP*, Albuquerque, NM, pp. 425-428, 1990.

[18] Haffner P., Franzini M., and Waibel A., "Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition," *in Proceedings of ICASSP*, Torento, pp. 105-108, 1991.

[19] Hush D. R. and Horne B. G., "Progress in Supervised Neural Networks," *IEEE Signal Processing Magazine*, vol. 1, pp. 8-39, January 1993.

[20] Iwanida H., Katagiri S., and McDermott E., "Speaker Independent Large Vocabulary Word Recognition Using LVQ/HMM Hybrid Algorithm," *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP*), Torento, pp. 553-556, 1991.

[21] Le Cerf P., and Weiye Ma Van Compernolle D., "Multilayer Perceptrons as labelers for Hidden Markov Models," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 185-193, 1994.

[22] Le Cun Y., "Modèles Connexionistes de l'apprentissage," *PhD Thesis*, Paris VI University, 1987.

[23] Lee K. F., *Automatic Speech Recognition: The Development of the SPHINX System*, Kluwer Academic Publication, 1989.

[24] Levin E., "Word Recognition Using Hidden Control Neural Architecture," *in Proceedings ICASSP*, Albuquerque, NM, pp. 433-436, 1990.

[25] Levinson S. E., Rabiner L. R., and Sondhi M. M., "An Introduction to the Application of The Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *Bell System Technical Journal*, vol. 64, no.4, pp. 1035-1074, 1983.

[26] Lippman R. P., "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, vol. 4, pp. 4-22, 1987.

[27] Makhoul J., El-Jaroudi A., and Schwartz R., "Formation of Disconnected Decision Regions with a Single Hidden Layer," *in Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 455-460, 1989.

[28] Morgan N. and Bourlard H., "Continuous Speech Recognition Using Multilayer Perceptrons with Hidden Markov Models," *in Proceedings of IEEE ICASSP*, vol. 2, pp. 26-30, Albuquerque, 1990.

[29] Morgan N. and Bourlard H., "Neural Networks for Statistical Recognition of Continuous Speech," *in Proceedings of IEEE*, vol. 83, no. 5, pp. 742-770, 1995.

[30] Niles L. T. and Silverman H. F., "Combining Hidden Markov Models and Neural Networks classifiers," *in Proceedings ICASSP*, pp. 417-420, Albuquerque, NM, 1990.

[31] Rabiner L. R., "A Tutorial in Hidden Markov Models and Selected Applications in Speech Recognition," *in Proceedings IEEE*, vol. 7, no. 2, pp. 257-286, 1989.

[32] Rabiner L. R. and Juang B. H., *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.

[33] Renals S., Morgan N., Cohen M., and Franco H., "Connectionist Probability Estimation in the Decipher Speech Recognition System," *in Proceedings IEEE ICASSP*, San Francisco, pp. 601-604, 1992.

[34] Rumelhart D. E., Hinton G. E., and Williams R. J., *Learning Internal Representations by Error Propagation*, *Parallel Distributed Processing Exportation of the Microstructure of Cognition*, MIT-Press, vol. 1, pp. 318-362, 1986.

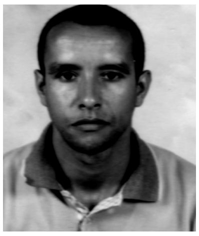
[35] Smyth P., Heckerman D., and Jordan M., "Probabilistic Independence Networks for Hidden Markov Models," *Neural Computation*, vol. 9, no. 2, 227-269, 1997.

[36] Tebelskis J. and al., "Continuous Speech Recognition Using Linked Predictive Networks," *Advances in Neural Information Processing Systems 4*, in Hanson M. and Lippman (Eds), Morgan Kaufman, pp. 977-984, 1992.

[37] Zavaliagkos G., Zhao Y., Schwartz R., and Makhoul J., "A Hybrid Segmental Neural Net/Hidden Markov Model System for Continuous Speech Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 1, pp. 151-160, 1994.

Rafik Djemili received the engineering and the MSc degrees, respectively, in 1993 and 2001, both from Badji Mokhtar Annaba University. In 2001, he joined the Automatic and Signals Laboratory of Annaba, where he worked on Arabic speech recognition, statistical methods and neural networks. He has been an assistant professor at Djelfa Univeristy, Algeria since December 2002.

Mouldi Bedda obtained the high studies degree in physics in 1981 from Houari Boumediene Algiers University, and the PhD in electrical engineering in 1985 from Nancy University, France. In 1990 he was a professor at Badji Mokhtar Annaba University. His interests are in the areas of signal processing, speech recognition, text to speech conversion and character recognition. He has been the director of the Automatic and Signals Laboratory of Annaba, since 2001.



Hocine Bourouba received the engineering and the MSc degrees, from Badji Mokhtar Annaba University in 1998 and 2001, respectively. Since 2001, he has joined the Automatic and Signals Laboratory of Annaba in research work in speech recognition and signal processing algorithms.