ARABIC SPEECH RECOGNITION USING NEURAL NETWORKS

M. Hamed*, F.W. Zaki, and R. Atta**
*Suez Canal University, Port Said-EGYPT
** Mansoura University, Mansoura-EGYPT

ABSTRACT

Application of neural networks on Arabic speech recognition is investigated. The neural networks used in this study are three-layer back-propagation (BP) network and integrated neural network (INN). The INN consists of a control network and several subnetworks. The control network identifies to which group the input speech belongs, where the recognition targets are partitioned into several groups, and the subnetworks recognize this input within each group. This study is divided into two stages. First, a new approach for the segmentation, that may be defined as a segmental back-propagation network (SBP), is proposed to solve varying length of speech utterances. This new approach is based on the segmentation of the speech utterances into a fixed number of segments throughout the duration of each speech utterance. The SBP- network is then applied to determine the suitable number of segments. Second, a number of experiments is conducted to assess the performance of neural networks. The training algorithms are based on the BP and weight smoothing algorithms. The weight smoothing algorithm is one of several techniques to improve the generalization capability of a neural network. Evaluation experiments are conducted on speaker dependent Arabic digits, alphabet characters, and some Arabic words.

KEY WORDS : Arabic speech recognition, segmentation, neural network, back-propagation, weight smoothing.

1. INTRODUCTION

Artificial neural networks (ANNs), and more particularly Multilayer Perceptrons (MLPs), have recently been recognized as attractive tools for information processing. The main advantage of ANNs is their capability to solve the mapping problems much faster than conventional methods due to massive parallelism. Another advantage lies in the existence of the effective learning algorithm, called back-propagation (BP) (Lippmann 1987). Also, new learning algorithms for neural networks have recently appeared (Lippmann 1987; Freeman and Skapura 1991; Krishnamurthy *et al.* 1990), and have been applied to different problem areas (Krishnamurthy *et al.* 1990; Burr 1988; Matsuoka *et al.* 1990; Waibel *et al.* 1989; Palakal and Zoran 1991).

In speech recognition, MLPs are usually employed as speech pattern discriminators, where output layer units represent class names, and must have a number of properties. First, it may have multiple layers and sufficient interconnections between units in each of these layers. This is to ensure that the network will have the ability to learn complex nonlinear decision surfaces. Second, the network should have the ability to represent relationships between events in time. These events could be LPC (Linear Prediction Coding) coefficients. Third, the number of learning features will be time-invariant.

Fourth, the number of weights in the network can be sufficiently small compared to the amount of training data so that the network may be forced to encode the training data by extracting regularity (Waibel *et al.* 1989).

In case of a small number of patterns, the MLP could achieve higher performances than conventional methods (Burr 1988; Waibel *et al.* 1989). At present, there are practical problems in applying the MLPs as discriminators to large vocabulary, continuous speech recognition. First, the MLP discriminators require a very large training data set to learn complex discrimination hyperplanes for a large number of patterns. Second, if new recognition patterns are added, discriminators in the system must be retrained using a training data set for all patterns. Many attempts to overcome these difficulties involved in using MLP discriminators are now under investigation (Matsuoka *et al.* 1990; Iso and Watanabe 1990).

The ability of layered networks to be generalized is essential in speech processing, particularly in automatic speech recognition by machines. There are several methods to improve the generalization capability of a neural network. One of them is to collect more training data in order to achieve good generalization performance. This could be difficult for some applications and it is not clear specially when the amount of training data is enough. The other approach is to constrain the neural network so that it may fit the considered problem.

According to (Bebis and Georgiopoulos 1994; Hertz *et al.* 1991), a number of techniques attempted to improve the generalization capability of a network by modifying not only the connection weights but also the architecture as training proceeds. These techniques can be divided into two categories. The first category includes methods that start with too many units and discard some units as required. These methods are called *pruning and weight decay* methods. The second category includes methods that start with a small number of nodes and gradually the nodes or connections can be added as needed. These methods are called *constructive* methods. Also, a weight smoothing algorithm would be determined as another technique for improving the generalization capability of neural network (Jean and Wang 1994).

Much work has been done on the recognition of English spoken words. Both either isolated or continuous but there has been a little research on Arabic spoken words. So, this paper implements two types of neural network on Arabic spoken recognition. These two models are back-propagation (BP) network and integrated neural network (INN). The implemented training algorithms are based on back-propagation and weight smoothing algorithms. Also, our work describes the proposed method of segmentation in order to solve the problem of length variation of speech utterances on the basis of BP-network.

2. SEGMENTATION

The speech signal is divided into small segments that can be assumed to be stationary and extracted features for each of these. The typical used values of the segment size range from 25 to 75 msec (Parsons 1987). The variations in the utterance length of the word spoken several times by the same speaker is an important problem. On the other hand, dividing the speech signals with various length provides various number of segments. This problem was solved by applying time normalization approaches (e.g. line time warping, dynamic time warping (DTW),...) (Parsons 1987). Time normalization is frequently done by a process known as "time warping". A warping function provides the neural network input nodes with a fixed-length representation of each utterance.

The proposed approach of segmentation divides the speech signals into a fixed number of-frames throughout the duration of speech utterances. The segmental back-propagation (SBP) network is then used to choose the suitable number of segments. In the process of DTW, a more computational time is taken in computing the distance between the test frames and the reference frames. This time can be significantly reduced when applying the proposed method of speech segmentation which assumes that the number of frames of each utterance is fixed.

2.1. Segmental Back Propagation-Network Structure

It should be mentioned that, several recent approaches have been implemented for time normalization (Bebis and Georgiopoulos 1994), The SBP-network differs from these recent approaches in that it attempts to recognize each utterance individually by using all its frames in an utterance length simultaneously to perform the recognition concept. The SBP-network used in this work is a three layer back-propagation network. The input to the network will be a fixed number of frames each consists of speech features which is defined mathematically as:

$$\text{Input data} = \text{No. of frames} \times \text{No. of features per Frame} \qquad (1)$$

The number of frames is fixed while the utterance length t_u of speech signal depends on the number of samples N_u in one utterance (speech signal), and the sampling rate f_s and it may be calculated by the expression

$$t_u = \frac{N_u}{f_s} \qquad (2)$$

In this moment, the number of samples N in each segment can be deduced according to the multiplication of the number of samples per frame N_f for the utterance length equal one second and the utterance length t_u.

$$N = N_f \times t_u \tag{3}$$

The number of samples per frame N_f and the frame length (msec) t_f can be computed directly as a function of number of frames N_s (would be fixed) through the equations:

$$\begin{aligned} N_f &= t_f \times f_s \\ t_f &= \frac{1000}{N_s} \end{aligned} \tag{4}$$

3. TRAINING ALGORITHMS

Neural networks are highly parallel computing structures consisting of a number of simple processing units, called neural units, and a set of interconnections between these units and the inputs to the network. In order to perform a particular task, neural nets undergo a training process in which the weight vectors associated with the units are modified. This process depends on the network being presented with statistically representative data during the training process. The network modifies the weight vectors based on internally calculated error measure derived from the training data. There is a number of techniques used to modify the weights, but all of the techniques can be classified as either *supervised* or *unsupervised*. The training concepts that will be considered in this work are back-propagation and weight smoothing algorithms as supervised learning rule.

3.1. Back-Propagation Algorithm

Back-propagation is the most widely used tool in the field of ANNs. Currently, it is the most popular method for performing the supervised learning task. The back-propagation algorithm is a generalization of the least mean square (LMS) principle. It uses a gradient search technique to minimize a cost function equal to the mean square difference between the desired and the actual network outputs. The desired output from output nodes is typically "low" (0 or ≤ 0.1) unless that node corresponds to the current input class, in which case it is "high" (1.0 or ≥ 0.9).

The network is trained by initially selecting small random weights and internal thresholds and then preselecting all training data repeatedly. Weights are adjusted after every trial until weights converge and the cost function is reduced to an acceptable value, i.e., prespecified value. An essential component of the algorithm is the iterative method that propagates error terms required to adapt weights back from nodes of the output layer to nodes of the lower layers. The back-propagation algorithm refers to an iterative training process in which an output error signal is propagated back through the network and is used to modify weight values (Lippmann 1987).

3.2. Weight Smoothing Algorithm

A weight smoothing algorithm has been introduced before by Jean 1994 to improve the generalization capability of neural network With this, a smoothing constraint is incorporated into the objective function of back-propagation to seek solutions with smoother connection weights. Experiments were performed on problems of wave form classification, multifont alphanumeric character recognition, and handwritten numeral recognition.

To locate a smoother solution, the back-propagation algorithm can be applied to minimize a new error function which includes not only the normal square error term but an extra error term that measures the unsmoothness of weight vectors. In fact this technique of adding an extra error term is a regularization which is commonly used to stabilize solutions to ill-posed problems (Jean and Wang 1994). The number of input neurons and hidden neurons are I and H, respectively. The connection weight w_{ji} between the j th hidden neuron and the i th input neuron, and the weight vector of the j th hidden neuron is $(w_{j1}, w_{j2}, w_{j3}, \ldots\ldots, w_{jI})$ will be required to the next analysis. A simple measure of the unsmoothness of the weight vector as $\sum_{i=2}^{I}[\ w_{ji} - w_{j(i-1)}\ ]^2$ may lead to the determination of the new error function according to the mathematical form:

$$E_{new} = E_s + \zeta \sum_{j=1}^{H}\sum_{i=2}^{I} [\ w_{ji} - w_{j(i-1)}]^2 \qquad (5)$$

In Eq. (5) the symbol E_s means the normal square error term while ζ is a constant. Therefore, for all i and j,

$$-\frac{\partial E_{new}}{\partial w_{ji}} = -\frac{\partial E_s}{\partial w_{ji}} - 2\zeta\ [2w_{ji} - w_{j(i-1)} - w_{j(i+1)}] \qquad (6)$$

Two variables w_{j0} (= w_{j1}) and $w_{j(I+1)}$ (= w_{jI}) are introduced in Eq. (6) to simplify the boundary condition. It may be note that the term $(2w_{ji} - w_{j(i-1)} - w_{j(i+1)})$ indicates the sum of differences between a weight w_{ji} and its two neighbors. From Eq. (6), the iterative weight updating on the basis of the iteration index t, the weight change $\Delta\ w_{ji}(t+1)$ of w_{ji} at $t+1$, the momentum η, the learning rate δ, and the weight change $\Delta w^s_{ji}(t+1)$ $(= -\frac{\partial E_s}{\partial w_{ji}})$ at t +1 computed from the square error term. This process may be expressed mathematically by

$$\Delta w_{ji}(t+1) = \eta \Delta w_{ji}(t) + \delta \left\{\Delta w^{s}_{ji}(t+1) \right.$$
$$\left. -2\zeta\left[2w_{ji}(t) - w_{j(i-1)}(t) - w_{j(i+1)}(t)\right]\right\} \qquad (7)$$

Eq. (7) indicates that in each weight updating step a compromise between minimizing the square error and reducing the difference among neighboring weights is made. To increase the chance of successful training, two techniques were incorporated into the smoothing algorithm. These are the over-relaxation and annealing smoothing factors. In concern of the over-relaxation technique, Eq. (7) can be reformulated as (Jean and Wang 1994)

$$\Delta w_{ji}(t+1) = \eta \Delta w_{ji}(t) + \delta \Delta w^{s}_{ji}(t+1)$$
$$-2\delta\ \zeta\ \left[2w_{ji}(t)\ -\ w_{j(i-1)}(t)\ -\ w_{j(i+1)}(t)\right] \qquad (8)$$

The first line of Eq. (8) contains a term which is the same as that used for back-propagation with momentum. With this observation, an over-relaxation technique will be adopted in the following procedure where the weight change $\Delta W_{ji}(t+1)$ will be approximated by two operations as:

$$\Delta w^{bp}_{ji}(t+1) = \eta\ \Delta\ w_{ji}(t) + \delta\ \Delta\ w^{s}_{ji}(t+1) \qquad (9\text{-a})$$

$$\Delta w^{sm}_{ji}(t+1) = w^{bp}_{ji}(t+1) - 2\delta\ \zeta\ [2w^{bp}_{ji}(t+1)$$
$$- w^{bp}_{j(i-1)}(t+1) - w^{bp}_{j(i+1)}(t+1)] \qquad (9\text{-b})$$

Eq. (9-a) is a typical back-propagation weight updating and Eq. (9-b) is an extra step to smooth the weights based on the *newer* weights of Eq. (9-a). The technique of using newer weights to perform updating is called over-relaxation and it is used quite frequently in iterative optimization techniques. With Eq. (9-a), the weights are updated and then the errors are back-propagated to the input layer after the presentation of a training sample. Before the presentation of the next training sample, the smoothing operation in Eq. (9-b) is applied to the connection weights of each hidden neuron. The purpose is to gradually smooth the weight vector of each neuron over time. Another perspective of Eq. (9-b) can be observed from the following re-formulation

$$\Delta w_{ji}^{sm}(t+1) = \gamma\, w_{ji}^{bp}(t+1) + \frac{(1-\gamma)}{2} \times [w_{j(i-1)}^{bp}(t+1) + w_{j(i+1)}^{bp}(t+1)] \qquad (10)$$

In Eq. (10) the weighting factor γ can be deduced as $\gamma = 1 - 4\delta\zeta \ (\leq 1)$. Eq. (10) means that the smoothing operation on each weight is equivalent to a *weighted* average of the weight w_{ji} and its two neighboring weights, and that the weighting factor γ determines the effect of smoothing according to the remark that a smaller γ has stronger smoothing effect; if $\gamma = 1$, then the operation has no effect at all. Although the operations are performed only within a two neighbor local area, the averaging effect, or the smoothing effect, is propagated to a larger neighborhood as the training phase continues.

The annealing smoothing factor is another technique used to increase the chance of successful training was to let weighting factor γ be a monotonically increasing function of time (or training iteration), i.e.,

$$\gamma = \gamma\ (t) = 1 - (1-\gamma_o)e^{-t/T} \qquad (11)$$

where $\gamma_0\ (\leq 1)$ and the total number of iteration T are two constants, and t is an iteration counter which is incremented after each data presentation. In this moment, it should be mentioned that $\gamma\ (0) = \gamma_o$ and $\gamma\ (\infty) = 1$. Since the weight changes due to back-propagation decrease as the network training continues, if γ was kept as a constant, the strong smoothing effect at the final stage of training would have damaged the solution developed previously. The function $\gamma\ (t)$ thus enforces the smoothing effect to be monotonically decreased along with the training. In the earlier stage of training, a stronger smoothing effect emphasizes on establishing a coarse yet smoother solution structure; at the later stage of training, weights usually are quite smooth already and a weaker smoothing effect allows more concentration on the development of solution details, i.e., reducing the square error.

For the selection of the both constants, the first constant T determines the rate of decay of smoothing effect and it should be proportional to the other size of the training data set while the other constant γ_o should be chosen to be close to 1 for two reasons. First, a vast amount of averaging operations is performed very frequently (after the presentation of each training sample) so that the smoothing effect may accumulate very fast even especially if each signal operation has very little effect. Therefore a large value of the γ_o will suffice for the purpose of smoothing. Secondly, a very small value of the γ_o will cause too strong smoothing effect to compare with the effect of gradient-decent seeking, and then it will lengthen the training process. The main steps employed in the weight smoothing algorithm are summarized by a flow-chart in Fig. 1. It contains three fundamental steps which can be explained as follows:

- The first step is required to initialize weights and thresholds, where all weights and thresholds are set to small random values when $\gamma = \gamma_0$ and sweep=1.

- The second step presents input and desired outputs for each data sample in the training set where the following three operations should be implemented. Firstly, the training sample is presented and a typical back-propagation weight updating may be performed as in Eq. (9-a). Secondly, for each hidden neuron, the local average will be performed according to Eq. (10) and then finally, updating the generated value of γ according to Eq. (11) would be applied.

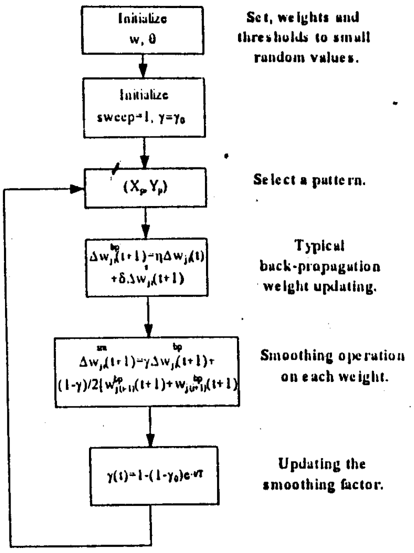
Initialize w, θ
Set, weights and thresholds to small random values.
Initialize sweep=1, γ=γ₀
(X_p, Y_p)
Select a pattern.
Δw_ji^bp(t+1)=ηΔw_ji(t) +δΔw_ji(t+1)
Typical back-propagation weight updating.
Δw_ji^sm(t+1)=γΔw_ji^bp(t+1)+ (1-γ)/2[w_j(i-1)^bp(t+1)+w_j(i+1)^bp(t+1)]
Smoothing operation on each weight.
γ(t)=1-(1-γ₀)e^-νT
Updating the smoothing factor.

Fig. 1 The flow-chart

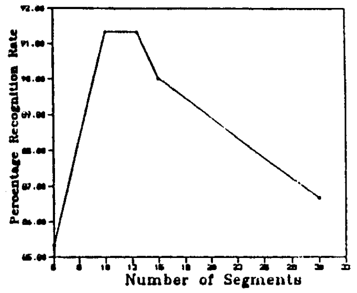

Fig. 2 Recognition rate

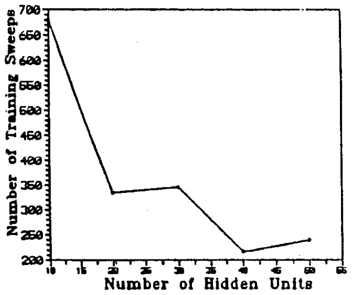

Fig. 3 Number of training sweeps

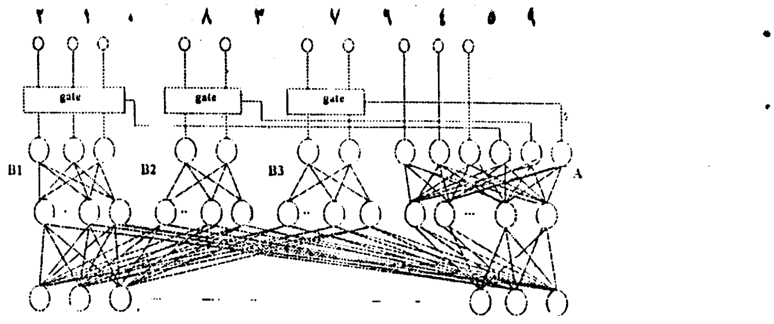

Fig. 4 The integrated neural network for Arabic digits.

- The third step is a repetition by going to the second step and if exactly all training data .can be successfully classified or (sweep > MAX-SWEEP), then stop. Otherwise, increase sweep by 1 and go to the second step.

The smoothing technique provides a simple yet effective design option in building a neural network. The approach is useful for many pattern recognition applications, especially when there are strong spatial correlations among neighboring data points in the input patterns. In smoothing operation, the training is usually slower than that without smoothness by factor of one. The reasons for this condition are twofold. First, the smoothing operation is performed in addition to the weight updating in error back-propagation, and more computation time may be needed for each sweep. Secondly, since the application of the smoothing operation causes the solution seeking in the weight space less gradient-decent, more presentations would be needed in the training.

4. EXPERIMENTS AND RESULTS

In order to examine the validity of the SBP-network and determine the recognition accuracy of the BP and INN networks, a speaker dependent isolated Arabic word recognition experiments must be carried out using the speech data.

4.1. Speech Data

The speech data are obtained in our work by a single female Arabic speaker. For the Arabic digits, 200 utterances in the digit vocabulary. The digit utterances are divided into two sets: a set of 50 utterances (5 utterances for each digit) is required for training, and a set of 150 utterances (15 utterances for each digit) will be needed for recognition. For the Arabic alphabet characters, 644 utterances in the character vocabulary are indicated. Also the character utterances would be divided into two sets: a set of 224 utterances (8 utterances for each character) that will be used for training, and the second set of 420 utterances (15 utterances for each character) which may be implementěd for recognition. For some Arabic words that can be frequently used in communication with the machines, there the 250 utterances in the Arabic word vocabulary, 100 utterances for training and 150 for testing would be introduced for the overall processing.

The speaker generates each vocabulary word to a microphone. The microphone's output is directly connected to a simple 8-bit A/D converter. The utterances are sampled at 10 kHz sampling rate using Sound Blaster Pro's (SBPro) A/D converter and analyzed by variant frame periods. The feature parameters are the 12 th-order LPC PARCOR coefficients as these coefficients may be computed for each frame by performing the Durbin's algorithm on autocorrelation coefficients of speech (Parsons 1987).

4.2. Segmental Back-Propagation Network Configuration

For the examination of the validity of the proposed concept, an isolated Arabic digit recognition has been developed by using SBP-network. The ability of the network to extract the underlying feature of the input speech data is affected by the *frame length* or *number of segments*. If there are insufficient number of segments (i.e frame interval is too long), the network may not be able to extract exactly the features of speech because the speech over this long time interval is a nonstationary process and hence the feature parameters will not reflect the changing properties of the real speech signal. If there is a sufficient number of segment's (i.e sufficiently short-time interval), the network may be able to extract the features of input utterance because the speech over this short interval can be assumed practically as stationary process. So, in order to choose the suitable number of segments, the recognition experiments of 10 Arabic digits are carried out using SBP-network which consists of a variable number of input units, a fixed number of hidden units (50 units used in experiments) and 10 output units to evaluate the best recognition performance of SBP-network.

In all implemented experiments, the number of segments is varied from 5 to 30 (i.e. the number of input units to the network is changed from 60 to 360 according to Eq. 1). For these inputs, several experiments are conducted using a network with 10 output units as well as 50 hidden units. The standard back-propagation algorithm with momentum is used for network training.

The recognition accuracy is obtained as a function of number of segments per word, using 50 utterances for training and 150 utterances for testing. This is illustrated in Fig. 2 where it can be seen that the highest recognition rate is obtained at 10 and 13 segment/word. For convenience of presentation the results are tabulated in Table 1 and this result confirms the conditions that stated above.

4.3. Back-Propagation Network Architecture

The BP-network used in this work for Arabic speech recognition consists of three layers: the input layer with 156 nodes (12 features per frame × 13 frames) derived from the segments via SBP model, hidden layer with a variable number of hidden nodes, and the output layer with 10 nodes for both Arabic digits and chosen Arabic words and 28 nodes for Arabic alphabet characters.

The increase in hidden neurons provides more degree of freedom to classify the training data set. If there are insufficient hidden units, the network will not be able to extract the features of speech exactly and therefore it may not be able to classify the training data set. On the other hand, if there are too many hidden

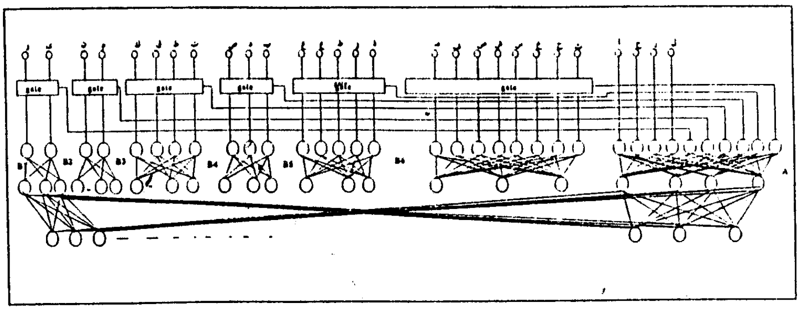

Fig. 5 The integrated neural network for Arabic alphabet characters.

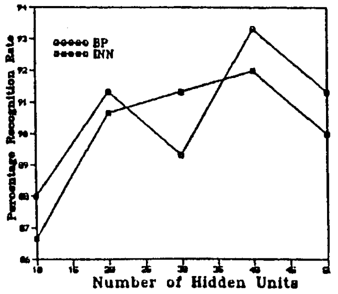

Fig. 6 Recognition rate

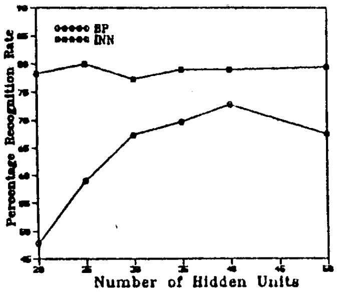

Fig. 7 Recognition rate

Table 1 The recognition results of SBP

No. of Segments	No. of Training Sweeps	Average Recognition rate (on test data)
5	700	85.33
10	360	91.33
13	240	91.33
15	207	90.00
30	150	86.67

Table 2 Arabic digits network

No. of Hidden Units	No. of Training Sweeps	Average Recognition rate (on test data)
10	679	88
20	335	91.33
30	346	89.33
40	217	93.33
50	240	91.33

Table 3 Arabic alphabet characters

No. of Hidden Units	System Error	Average Recognition rate (on test data)
20	3.75E-03	47.86
25	1.93E-03	59.05
30	1.91E-03	67.38
35	1.97E-03	69.76
40	1.51E-03	72.86
50	1.86E-03	67.62

Table 4 Arabic words

No. of Hidden Unit.	No. of Training Sweeps	System Error	Average Recognition rate (on test data)
10	400	2.04 E-04	97.33
20	287	8.32 E-05	96.70
30	320	6.50 E-05	98.00
40	318	9.51 E-05	98.00
50	460	3.36 E-04	96.70

units, a larger amount of training data will be required since there are too many parameters to estimate from the training data. Otherwise it may be difficult to converge this case to the optimal solution. The necessary number of hidden units must be chosen by conducting the recognition experiments using a variable number of hidden units to evaluate the best recognition performance of the BP-network. In the following set of experiments, both the back-propagation and smoothing algorithms are used.

The simulation results using back-propagation algorithm can be tailored as follows:

1- The number of training sweeps decreased monotonically as the number of hidden units was increased. A learning sweep consists of one presentation of (60) training patterns for digits and (50) training patterns for alphabet characters and Arabic words. In this moment, it should be mentioned that Fig. 3 shows the number of training sweeps required for different numbers of hidden units for Arabic digits.

2- The results with no smoothing (back-propagation algorithm) are summarized in Tables 2, 3, and 4. Table 2 shows that the best recognition rate for Arabic digits was 93.33 percent using 40 hidden units. For Arabic alphabet characters, the network with 40 hidden units provides a recognition rate of 72.86 percent as listed in Table 3. Moreover, the best recognition rate for some Arabic words was 98 percent using 30 and 40 hidden units as given in Table 4. Further increase of hidden neurons did not improve the recognition rate due to overfitting.

3- From these experiments, it has been found that a network with 156 input units, 40 hidden units, and a variable number of output units can be used for Arabic spoken recognition. This leads us to concentrate on the proposed system of processing towards the vital applications in the robotics systems for example.

The above set of experiments is repeated but using BP-network with the smoothing algorithm as explained in section 3.2. The smoothing is applied to the network with 40 hidden neurons that gave the best recognition rate for BP-network trained using back-propagation algorithm. The simulation is conducted for various values of γ_0 and *T*. All the values of *T* are,chosen to be integer multiples of the number of training data patterns so as to be proportional to the size of the training data set. The results are summarized in Tables 5 and 6. From these tables, we can see that:

1- For all the experiments of the network with smoothing, the recognition rate of Arabic digits did not improved significantly than that of the network without smoothing. The recognition rate appears to be 94 percent when the constant *T*= 1000, γ_0= 0.997 and 0.999.

2- For all experiments of Arabic alphabet characters, the recognition rate is always lower than that corresponding technique but without smoothing.

3- When there are not enough training data patterns, the average recognition rate is not improved significantly in the case of either smoothing or not. But when the training patterns are increased, the average recognition rate becomes much better than that without smoothing (Jean and Wang 1994).

4.4. Integrated Neural Network Architecture

The INN consists of a control network and several subnetworks. The control network identifies to which group the input speech belongs, and subnetworks recognize the spoken words within each group. Using INN, if there are few

Table 5 Network with 40 hidden neurons trained with 50 training data pattern with smoothing effect (for Arabic digits).

Network training approach		System Error	Average Recognition rate (on test data)
γ_0	T		
0.997	500	4.79E-05	94.00
	1000	4.04E-03	87.33
	15000	4.86E-05	92.67
0.995	1000	1.83E-04	90.00
	2000	6.15E-05	92.67
0.999	1000	6.42E-05	94.00

Table 6 Network with 40 hidden neurons trained with 224 training data pattern under smoothing consideration (for Arabic alphabet characters).

Network training approach		System Error	Average Recognition rate (on test data)
γ_0	T		
0.999	1000	3.10E-02	66.43
	2000	9.12E-03	67.86
	2240	7.72E-03	66.67
0.997	2000	4.39E-02	58.10
	2240	6.07E-02	57.38
0.995	1000	7.07E-03	65.48

Table 7 Recognition results

No. of Hidde n Units	No. of Training Sweeps	Average Recognition rate (on test data)
10	565	86.67
20	303	90
30	232	90.67
40	281	92.67
50	210	90

Table 8 Recognition results

No. of Hidden Units	System Error	Average Recognition rate (on test data)
20	8.29E-04	85.24
25	7.83E-04	87.86
30	2.13E-03	84.52
35	9.72E-04	86.19
40	8.75E-04	86.19
50	1.78E-03	87.38

training data, the network can recognize spoken words with higher recognition accuracy than BP-network. Furthermore, new vocabulary entries can easily be added to an INN by adding new subnetworks corresponding to the new groups. The control network and subnetworks are trained individually using BP-algorithm (Matsuoka *et al.* 1990).

The INN is used to recognize Arabic digits and Arabic alphabet characters. In all experiments, the control and subnetworks were chosen to be three-layer networks. For Arabic digits, the INN architecture is drawn as in Fig. 4. The recognition targets were partitioned into six groups. This assumption was made because a prior knowledge about the recognition targets of Arabic digits is not available.

For Arabic alphabet characters, the INN architecture is given in Fig. 5. Using the manner of articulation, Arabic alphabet characters were partitioned into ten groups (El-Imam 1989). The number of subnetwork hidden units was varied from 5 to 15 to optimize the network performance for each subnetwork. And the number of the control network hidden units was varied from 10 to 50 in steps of 10 for Arabic digits and from 20 to 50 for Arabic alphabet characters. The results can be summarized as follows:

1- For the subnetworks the best recognition rate was at 10 to 15 hidden units.

2- The results are abstracted in Tables 7 and 8. From Table 7, it is seen that the best average recognition accuracy of group identification for Arabic digits was 92.67 percent at 40 hidden units. For Arabic alphabet characters, the results of Table 8 gives us that the average accuracy of group identification was 87.86 percent at 25 hidden units.

Table 9 The recognition results of both INN and BP-network for Arabic alphabet characters (No. of hidden units = 25).

Group	INN				BP-Network
	character	(a) Group Identification	(b) character Recognition In Each Group	Overall	
Unvoice Fractive	ث	100	60	60	20
	ح	100	86.67	86.67	40
	خ	100	86.67	86.67	93.33
	س	100	100	100	86.67
	ش	100	100	100	86.67
	ص	73.33	100	73.33	26.67
	ف	100	46.67	46.67	40
	هـ	100	66.67	66.67	46.67
Voice Fractive	ذ	86.67	86.67	73.33	46.67
	ز	100	100	100	80
	ظ	86.67	100	86.67	60
	ع	100	86.67	86.67	60
	غ	100	73.33	73.33	13.33
Unvoice Stop	ت	100	100	100	46.67
	ط	100	100	100	80
	ق	86.67	100	86.67	73.33
	ك	86.67	93.33	86.67	53.33
Voice Stop	ب	46.67	100	46.67	60
	د	53.33	93.33	46.67	26.67
	ض	73.33	93.33	66.67	46.67
Voicec Nasal	م	93.33	100	93.33	46 67
	ن	93.33	100	93.33	60
Voiced Semi vowel	و	100	100	100	80
	ى	80	100	80	73.33
	أ	86.67	--	86.67	93.33
	ج	66.67	--	66.67	60
	ر	66.67	--	66.67	66.67
	ل	80	--	80	86.67
Average		87.86	90.56	80	59.05

4 5. Comparison

The comparison between INN and BP-Network were carried out for Arabic digits and Arabic alphabet characters. The recognition results of Arabic alphabet characters for the grouping method that was based on the manner of articulaticn is shown in Table 9. for the number of hidden units of 25. From all experiments on Arabic digits and alphabet characters, it was seen that: In an INN, the recognition accuracy of almost 100 percent are obtained for digits and alphabet characters recognition in each group.

For Arabic digits, the best recognition accuracy for both INN and BP-network were 92.67 percent and 93.33 percent based on BP-algorithm (94 percent based on weight smoothing algorithm) respectively These results are obtained at 40 hidden units as illustrated in Fig. 6. Using the grouping method based on the manner of the articulation of Arabic alphabet characters, the recognition accuracy appears to be 80 percent for INN, as compared with 72.86 percent and 67.86 percent for BP-network based on BP-algorithm and weight smoothing algorithm respectively This higher accuracy for the implementations of INN may obtained using 25 hidden units as compared with 40 hidden units for BP-network as drawn in Fig 7 This leads us to give a great attention to the proposed system of analysis to be applied sucessfully in the field of speech recognition and speech translation into text. This is a very important field in the modern fields of computer utilizations

5 CONCLUSIONS

From the presented work, it can be concluded that

1- The proposed approach of segmentation. segmental back-propagation (SBP), to solve the varying length of speech utterances is recommended

2- A small computational time is required to fix the number of frames of each utterance.

3- The values of the average frame length are approximately close to the typical values of average frame size.

4- The BP-network trained with BP-algorithm achieve better recognition rates than that trained with weight smoothing algorithm.

5- The INN achieve higher recognition accuracy than BP-network.

6- The INN training time is reduced compared with BP-network.

6. REFERENCES

Bebis, G. and Georgiopoulos M. (1994). "Feed-Forward Neural Networks", *IEEE Potentials*, pp. 27-31, October/November.

Burr, D. J. (1988). "Experiments on Neural Net Recognition of Spoken and Written Text", *IEEE Trans. on. ASSP*, Vol. 36, No. 7, pp. 1162-1168, July.

El-Imam, Y. A. (1989). "An Unrestricted Vocabulary Arabic Speech Synthesis Systemi", *IEEE Trans. on ASSP*, Vol. 37, No. 12, pp. 1829-1845, December.

Freeman, J. A. and Skapura D. M. (1991). "Neural Networks: Algorithms, Applications, And Programming Techniques", *Addison-Wesley Publishing Company*, New York.

Hertz, J., Krogh A., and Palmer R. G. (1991). "Introduction to the Theory of Neural Computation", *Addison-Wesley Publishing Company*, New York.

Iso, K. and Watanabe T. (1990). "Speaker-Independent Word Recognition Using A Neural Prediction Model", Readings in speech recognition, *Morgan Kaufmann Publishers*, USA..

Jean, J. S. N. and Wang J. (1994). "Weight Smoothing to Improve Network Generalization", *IEEE Trans. on Neural Networks*, Vol. 5, No. 5, pp. 752-763, September.

Krishnamurthy, A. K., Ahalt S. C., Melton D. E., and Chen P. (1990). "Neural Networks for Vector Quantization of Speech and Images", *IEEE Journal. on Selected Areas in Communications*, Vol. 8, No. 8. pp. 1449-1457, October.

Lippmann, R. P. (1987). "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, pp. 4-22, April.

Matsuoka, T., Hamada H., and Nakatsu R. (1990). "Syllable Recognition Using Integrated Neural Networks", *System and Computers in Japan*, Vol. 21, No. 9, pp. 89-98.

Palakal, M. J. and Zoran M. J. (1991). "A Neural Network-Based Learning System for Speech Processing", *Expert System with Applications*, Vol. 2, pp. 59-71, Printed in the USA.

Parsons, T. W. (1987). "Voice and Speech Processing", *McGraw-Hill Book Company*, USA.

Waibel, A., Hanazawa T., Hinton G., Shikano K., and Lang K. J. (1989). "Phoneme Recognition Using Time-Delay Neural Networks", *IEEE Trans. on. ASSP*, Vol. 37, No. 3, pp. 328-339, March.