Afaster Training Algorithm and Genetic Algorithm to Recognize Some of Arabic Phonemes

Dr. Sarah B. Aziz, Saba Abdul-Wahed S., Maysaa Abdul-Kareem N.

Basrah Univ., collage Since, computer dep.

Abstract

Neural network and Genetic algorithm are used in this work to recognize some of Arabic phonemes recorded by more than person in various age. This work implemented in two basic stage, first one include speech signal process such as segment, analyses and determined the best samples which represent the basic feature to sound signal. The second stage contain recognize operation which done by Levenberg-Marquardt(LM) Algorithm which is one of the faster training algorithm in the Neural network used to determine number of hidden neuron with Genetic Algorithm. A research tool has been implemented, using the Matlab 6.5, sound forge5 and programming language pascal7

*1.*Introduction:

In the past decade, two areas of research which have become very popular are the fields of neural networks and genetic algorithms. Both are computational abstractions of biological information processing systems, and both have captured the imaginations of researchers all over the world [1].

Neural Network and Genetic algorithm have received great acclaim in the computer science research community since the 1980s.For the most part, this results from successful applications of these new computing models, but also, because the concepts share the spirit of a movement that goes beyond science[2,3].

In general, Neural Network are one of the promises for the future in computing. They offer an ability to perform tasks outside the scope of traditional processors. They can recognize patterns, images, speech, characters and language with vast data set and generalize those into recommended courses of action[4,5] .

According to Elaine & Knight multiple layer networks can represent any function, which implies that the design process has to focus on the definition of the number of neurons & the learning strategy. Therefore Genetic algorithm will be used in this work to find the suitable number of hidden neurons in the Neural network [1]. There are many types of learning algorithms some of them are known as heuristic and the others as stochastic, LM algorithm which is a stochastic, is widely accepted as the most efficient one in the sense of realization accuracy[6]. It gives a good compromise between the speed of the Newton algorithm and the stability of the steepest descent method, and consequently it constitutes a good transition between these methods.

Speech recognition is a hard problem, requiring a combination of many techniques, however modern methods have been able to achieve an impressive degree of accuracy [4,6].Many approaches have been tried during the long history of research in this speech signal recognition field, such as HMM, FFT and DWT to analysis the speech signal, NN, filter bank to recognize it . In this work we use FFT and zero-crossing for segmentation and analysis and neural networks with Genetic algorithm for recognize the signal [4,5,6].

The outline of this paper is as follow: the speech recognition will be discussed in section 2. Section 3 will describe the Neural network structure & the learning algorithm. The basic steps of Genetic algorithm will be given in section 4. Section 5 will describe the data set used in training & testing and the results will be shown in section 6. finally section 7 contains some remarks and conclusion.

2. Speech Recognition

The process of speech recognition is typically divided into several well defined steps. Different systems vary on the nature of these steps, as well as how each step is implemented, but the most successful systems follow a similar methodology[7,8,9].

1. Divide the speech signal into evenly spaced frames .

2. Process each frame for important characteristics

This step typically involves performing a spectrum analysis of the frame. This can be done with a Linear Precidive Coding, or with a bank of frequency filters, but the most successful technique to date has been that of **Fast Fourier Transform** (FFT). The **zero-**

crossing method is used to segment the phoneme in each word contain in the speech signal.

2-1 segmentation signal

The first task is to identify the precense of a speech signal. This task is easy if the signal is clear, however frequently the signal contains background noise, resulting from a noisy microphone, a fan running in the room, etc. in this task we assume the signal recorded in best environment (the percentage of noise from microphone is smaller and no fan running in the room) ,and use the zero crossing method to determine the begging and the end of each phoneme in each recorded signal speech .There are several published researches of using zero-crossing for Segmentation speech signal [6], [9], [10]. In [6] are described possibilities of the spectral analysis based on zero-crossing.The formant frequency estimation on the base of zero-crossing is presented in [9]. Experiments on speech recognition when frequency information of the signal is obtained from zero-crossing intervals are described in [10].

In this work, we determine the cutting reigned to each phoneme in the recorded signal through calculate the number of change the sign signal from positive to negative or inverse round the zero axis to each frame(frame containe128 points). After that the absolute deferential between each two sequential frames calculate and the maximum from them will be the bester frame to segment the signal.

2-2 obtaining signal speech samples

The next important step in the processing of the signal is to obtain a frequency spectrum of each frame. The purpose of the frequency spectrum is to identify the formants, which are the peaks in the frequency spectrum.

One method to obtain a frequency spectrum is to apply an FFT to each frame. The resulting information can be examined manually to find the best features, and we can obtained them from the signal by done this steps :-

1. Run the FFT on frame (the frame content 128 point from signal wave)

2. Calculate the increment value in the frequency($Freq._{inc}$) for each samples using this equation

$$Freq._{inc} = \text{Sampling Freq. / FFT points} \quad \ldots\ldots\ldots(1)$$

3. Calculate the frequency at each point ($Freq._{P}$) according to this equation

$$Freq._{P} = Freq._{inc} * \text{Point.seq} \quad \ldots\ldots\ldots..(2)$$

4. Sort the all values (Freq and its location in the Frame).
5. Select first three values from freq and its location only and save there in files .
6. Repate the steps from step 1 to step 5 until reach process to end of file wave.

3. Neural Network

Neural networks with back propagation learning showed results by searching for various kinds of functions. However, the choice of the basic parameter (network topology, learning rate, initial weights) often already determines the success of the training process. The selection of these parameter follow in practical use rules of thumb, but their value is at most arguable[11].

Very important feature of neural networks is their adaptive nature, where "learning by example" replaces "programming" in solving problems. This feature makes such computational models very appealing in application domains where one has little or incomplete understanding of the problem to be solved but where training data is readily available. There are many different kind of learning rules, the most common one is called Back Propagation Algorithm (BPA)[12].

Although the (BPA) has been a significant illestone in neural network research area of interest, it has been known as an algorithm with a very poor convergence rate. Many attempts have been made to speed up the BPA. Commonly known heuristic approaches such as momentum, variable learning rate, or stochastic learning lead only to a slight improvement. Better results have been obtained with the artificial enlarging of errors for neurons operating in the saturation region [11,12,13].

A significant improvement on realization performance can be observed by using various second order approaches namely Newton’s method, conjugate gradient’s, or the Levenberg-Marquardt(LM) technique[12,14].

3-1 network architecture :-

The network in this work contain from three layers shown as :-

Input layer, have a six neurons represent the one frame output from privies stage .

Hidden layer, the number of its neurons determined it the genetic algorithm in the next section.

Output layer, have a three neurons its value in the binary code to represent seven phonames, table(1) show these value.

phoneme	code
س	100
ش	010
ا	110
و	001
ي	101

Table (1): the code of output

3-2 Learning Algorithm

Learning is a fundamental and essential characteristic of NNs. There are two types of learning; Supervised and Unsupervised. In supervised learning, the set of training data consists of previous cases with known input and output values. The neural network system receives the actual output, computes the error and adjusts the weights according to the error [15,16]. The most common supervised learning algorithm is the Back Propagation (BP).

The BP learning is based on the gradient descent along the error surface [11, 13, 15,16]. That is, weight adjustment is proportional to the negative gradient of the error with respect to the weights. In mathematical words [17]

$$w_{k+1} = w_k + \alpha d_k \quad \text{........} \qquad (3)$$

Where, w_k denotes the weight matrix at epoch k, and α is appositive constant, generated randomly in the range between 0 and 1, is called learning rate.

The direction vector d_k is negative of gradient of the output error function E,

$$d_k = -\nabla E(w_k) \quad \text{........} \qquad (4)$$

There are two standard learning schemes for the BP algorithm: **incremental** (also called online) mode and **batch** (also called offline) mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode, all of the inputs are applied to the network before the weights are updated. In either case the vector w_k contains the weights computed during k^{th} iteration, and the output error function E is a multivariate function of the weights in the network [17,18]:

$$E(w_k) = \begin{bmatrix} E_p(w_k) & [\mathit{online}] \\ \sum_p (w_k) & [\mathit{offline}] \end{bmatrix} \quad \text{........} \qquad (5)$$

Where, $E_p(w_k)$ denotes the haf-sum-of-squares error functions of the network output for a certain input pattern P.

Normally, the BP learning uses the weight change propotional to the negative gradient of the instantaneous error with respect to the weight. A more effective method[11-13] can be derived by starting with the following Taylor series expansion of the error as a function of the weight vector,

$$E(w + \Delta w) = E(w) + g^T \Delta w + \frac{1}{2} \Delta w^T H \Delta w + \text{....} \quad \text{........} \qquad (6)$$

Where , $g = \frac{\Delta E}{\Delta w}$ is the gradient vector, and $H = \frac{\partial^2 w}{\partial w^2}$ is the Hessian matrix. Newton's method of BP learning uses this Hessian matrix as a component of weight update.

Unfortunately, it is complex and expensive to compute the Hessian matrix for feed forward NN. Levenberg -Marquardt (LM) was designed to approach second-order training speed without having to compute the Hessian Matrix [14,16]. When the error function has the form of a sum of squares (as is typical in training feed forward networks), then the Hessian matrix can be approximated as

$$H = J^T J \quad \text{.........(7)}$$

And the gradient can be computed as,

$$g = J^T E \quad \text{.........(8)}$$

Where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights, and E is the vector of network errors. The Jacobian matrix can b computed through a standard BP technique that is much less complex than computing the Hessian matrix [14,15].

LM algorithm uses this approximation to the Hessian matrix in the following Newton-like update:

$$w_{k+1} = w_k - \left[J^T J + mI^{-1} \right] * J^T E \quad \text{........(9)}$$

Where, I is the identity matrix and m is a scalar initialize randomly. When the scalar m is zero, this is just Newton's method, using the approximate Hessian matrix. When m is large, this becomes gradient descent with a small step size.

Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, m is decreased (by mu-dec) after each success ful step (reduction in error function) and is increased(by mu_inc) only when a tentative step would increase the error function. In this way the error function will always be reduced at each iteration of the algorithm[14-15].

4 Genetic Algorithm

NNs are used as learning systems and GAs as optimization systems, but as many researchers have discovered, they may be combined in a number of different ways resulting in highly successful adaptive systems.

Genetic Algorithms (GA) are a method of "breeding" computer programs and solutions to optimization or search problems by means of simulated evolution. Processes loosely based on natural selection, crossover, and mutation are repeatedly applied to a population of binary strings which represent potential solutions. Over time, the number of above-average individuals increases, and better fit individuals are created, until a good solution to the problem at hand is found [3].

Genetic algorithms are global search methods, that are based on principles like selection, crossover and mutation. This work examines how genetic algorithms can be used to optimize the network topology etc. of neural networks. It investigates, how various encoding strategies influence the GA/NN synergy. They are evaluated according to their performance on academic and practical problems of different complexity.

This section, GA will be used to get the suitable number of hidden neurons in the network . The main steps are illustrated bellow:

4-1. Coding [1,2,3]

Thus, each individual (chromosome) consists of four genes.

4-2.Population Initialization

The population of the individuals will be initialized randomly, the values of the chromosome genes are generated randomly in some range as shown in the table (2)

Gene-Name	Init. value	Fina. value
Hidden no.	6	16
Init.mu	0.001	0.9
Mu_inc	2	10
Mu_dec	0.1	0.9

Table (2): values of chromosome

4-3. Fitness Function

Each individual in the population of the GA should be evaluated to determine its ability to produce new individual through the reproduction operations. The formula of the fitness function is as:

$$Fitness = perf + hidd / \max\_hidd \quad \textbf{(10)}$$

Where perf is error function in neural network

4-4. Reproduction

The time of the reproduction phase is get after the population initialization and evaluation of each individual by the fitness function. The reproduction phase includes four different processes.

A. Selection [1,2,11]

Selection is an important operation. A combination between two selected techniques; Ranking and Tournament that were explained will be used to select the two parents. Where, first the individuals of the population is ranked then it is breakdown into a finest part and normal part. The size of each part is determined as follows:-

$$Finest\_size = round\left(Population\_size / 3\right) \quad .....(11)$$

$$Normal\_size = Population\_size - Finest\_size \quad ......(12)$$

Then, the tournament technique is used to select the two parents for crossover.

Thus, three individuals will be selected randomly from the finest part and the best one of them will be selected as a first parent, while the second parent will be selected randomly from the rest individuals of finest part.

B. Crossover

For each genetic cycle, the two selected parents will be recombined by using the uniform crossover to produce one child with probability PC = 0.8 .

C. Mutation

Each gene in the chromosome that obtained by the crossover will be muted by adding a value generated randomly in the range [-2, +2] for the first gene, in the range [-0.002, +0.002] for the second gene, in the range [-1,+1] to the third gene and in the range [-0.002,+0.002] to the last gene. This mutation will not be effected just if the new value of the gene is in the right range.

D. Replacement

After evaluating the new individual produced from the crossover and mutation, a selected individual of worse fitness will be replaced by the new individual under some condition. Then, among the three individuals that are selected randomly from the normal part the worse one will be selected to be replaced by the new individual if it has fitness better than the worse one.

4-5. Convergence

The proposed GA is iterated until either the number of the genetic cycles reaches to the predetermined maximum cycle's number which in this work is set to 25 cycles or the finest part of the population is not changed for some cycles that set for 5 in this work.

4-6. Result

After we checking the recognition result to all of recording sound segments for each run in the GA we find the best architecture to the suggested network is **the run 6** . in this way the NN finished training in less than 500 epochs, some of other training algorithm such as GDA need to more than 5000 epochs .

Table (4) : the GA result

5. conclusion and recognition result

Speech synthesis is a complex task that aims to produce naturally sounding speech. While working systems that produce intelligible speech have existed since the 1970s, the final aim of producing a synthesis that is indistinguishable from a human speaker has still to be realized [6].

We choice two groups of person from both sex in different age to implement this work through recording some of segments sound , these segment recorded in one environment ,and in same

Some of the results run to GA						
Run	Hid. no	Init-mu	Mu-inc	Mu-dec	Prf	Fit.
1	6	0.010196	5	0.1957	1.1075e-012	0.4
2	6	0.042581	2	0.2309	1.1952e-013	0.4
3	7	0.019648	4	0.3299	1.8102e-012	0.5
4	6	0.032135	5	0.1265	6.1194e-014	0.4
5	6	0.054364	5	0.1895	8.0505e-014	0.4

recorded program and parameters ,the segment contain from two or three phoneme sort as constant –vowel or constant-vowel-constant.Some of the recorded segments are("سـا", "شا","سـاس","شـاس","سـوش","شـيش","شو" and "شي"). After implement the network on this segment we find the percentage which shown in the table (5) to each phoneme contain in the recorded segment. And the recognize percentage to each segment is more than 90% .

phoneme	Recognize percentage
السين	95%
الشين	90%
الالف	100%
الواو	90%
الياء	90%

6	6	0.01 5406	4	0.1 543	3.1551 e-014	0. 4
7	6	0.06 9041	4	0.6 268	4.558e -012	0. 4

Table (6):the recognize percentage to each phoneme.

References

1. Udo Seiffert, "Multiple Layer Perceptron Trainig Using Genetic Algorithms", ESANN' 2001, Bruges(Belgium),25-27April 2001, D-Facto public.,pp-159-164.
2. A.J.F. van Rooij, L.C. Jain and R.P. Johnson, "Neural Network training using genetic algorithms", Singapore : World Scientific,1996.
3. P. Robbins, A. Soper and K. Rennolls, "Use of genetic algorithms for optimal topology determination in back propagation neural networks", Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, Vienna, Springer, 1993, 726-730.
4. Joe Frankel, Korin Richmond, Simon King and Paul Taylor, "An Automatic Speech Recognition System using Neural Networks And Linear Dynamic Models To Recovre And Model Articultory Traces", Centre for Speech Technology Research, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, UK.
5. C. M. Bishop, "Neural Networks for Pattern Recognition", Oxford University Press, Oxford, UK,

1995.
6. G. Kubin, "Synthesis and coding of continuous speech with the nonlinear oscillator model," in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, pp. 267–270, Atlanta, Ga, USA, May 1996.
7. D.S. Kim, S.Y. Lee, R.M. Kil, "Auditory processing of speech signal for robust speech recognition in real-world noise environment ", IEEE Transaction on Speech and Audio Processing, Vol. 7, No.1, pp.59-69, 1999.
8. Stephen McLaughlin," A Brief introduction to non linear speech analysis and synthesis ", Newsletter, volume 14, Number 2, June 2003.
9. R. De Mori, L. Moisa, R. Gemello, F. Mana, D. Albesano, "Augmenting standard speech recognition features with energy gravity centres", Computer Speech and Language 15,pp.341-354, 2001.
10. Konstantin Biatov, "The Method of an Audio Data Classification and Segmentation", Fraunhofer Institute for Media Communication, Schloss Birlinghoven , 53754,Sankt Augustin, Germany.

11. Torvik, L. and Wilamowski, B. M., "Modification of the Backpropagation Algorithm for Faster Convergence", presented at 1993 International Simulation Technology Multi conference November 7-10, San Francisco; also in proceedings of Workshop on Neural Networks WNN93 pp. 191-194, 1993.
12. Battiti, R., "First- and second-order methods for learning: between steepest descent and Newton's method", Neural Computation, vol. **4**, no. 2, pp. 141-166, 1992.
13. Sperduti, A. and Starita, A. (1993). "Speed up learning and network optimization with extended back-propagation", Neural Networks, **6,** 365-383.
14. Bogdan M. Wilamowski, Serdor Iplikei, and others, "An Algorithm for Fast Convergence in Training Neural Networks" ,IEEE, PP. 1776-1782, 2001
15. H. Demuth, and M. Beal, "Neural network toolbox", User Guide, version 4, for use with Matlab, Massachusetts: The math works Inc., 1998.
16. F. Zahedi," Intelligent system for business", Expert systems with neural networks, woods worth. Puplishing Inc., 1993.
17. S. Kamarthi, and S. Pittner, "Accelerating neural training using weight extrapolation", Neural Networks, 12,1285-1299.
18. L. Fausett, "Fundamentals of Neural Networks algorithms, Applications, and Programming Techniques" ,Addision-Wesley,1991.
19. Hagan, M. T. and Menhaj, M., "Training feed forward networks with the Marquardt algorithm", IEEE Transactions on Neural Networks, vol. **5**, no. 6, pp. 989- 993, 1994.