IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555
Vol.6, No4, July-August 2016

Hybrid Arabic Speech Recognition System Using FFT, Fuzzy Logic and Neural Network

Salam Hamdan
Department of Computer Science
Princess Sumaya University for Technology
Amman, Jordan
s.hamdan@psut.edu.jo

[1]Adnan Shaout
Department of Computer Science
Princess Sumaya University for Technology
Amman, Jordan
a.shaout@psut.edu.jo

***Abstract*—computers, smartphones and many technologies nowadays are automated. Everybody is using the technology, therefore the need to make computers understand human languages (natural languages) is essential. That has made recognizing speech an important field of research. In this paper, a new model is proposed to recognize Arabic speech using Fast Fourier Transformation (FFT), fuzzy logic and neural network (NN). The reason of using fuzz inference system (FIS) is to help the user in choosing the optimum weight for the Feed Forward neural network. In order to test the proposed speech recognition system, a dataset was generated from six human voices (2 men, 2 women and 2 small girls), each one of them uttered two Arabic words 50 times. The model has been tested using two Arabic words with the same voices. The model gave 98% accuracy.**

***Index Terms*—Arabic speech recognition, fuzzy logic, neural network, Fast Fourier Transform, MATLAB.**

I. INTRODUCTION

Nowadays with the large growth in automated systems and the need to make computers more friendly and easy to deal with, there is a need to make computer understand natural languages [1] so that it can take orders from people through natural languages.

Speech recognition is a very important area thus, it helps to extract features from a speech signal and convert it to a language computers can understand [2].

In order to deal with a speech signal, a number of steps must be done [3]. The first step is to take a speech signal in an analogue form [4] then convert it to a digital form. Many techniques are used to convert an analogue signal to a digital signal [5]. The technique that is used in this paper is the Fast Fourier Transform (FFT) [6]. Afterwards, these modified words are going to be used in the training model. Feed Forward neural network is used to train the model, and according to this model an input target is required and an initial weights are also required to train the model. In MATLAB the initial weights are set randomly. After training the model, a set of weights are gained, thus in neural network to gain the optimal weight there is a need to train the model many times. In this paper, a fuzzy inference system is used to classify these obtained weights from the training times into different classes. Therefore, the FIS is used to choose the optimum weights for the NN.

This model is tested using two Arabic words (شكراً, مرحباً) (Chokran, Marhaman). The first word means hello and the second word means thank you. Each word is uttered 50 times by two males, two females and two children. The total number of generated train data is 500.

This paper is organized as follows, section II presents a literature review of speech recognition models using fuzzy logic and describe the difference among these models. Section III describes the proposed hybrid speech recognition model. Section IV describes the dataset used in this paper. Section V discusses the performance evaluation of the proposed methodology. Section VI concludes the paper. Finally, section VII describes the future work.

II. LITERATURE REVIEW

A number of models have been proposed to recognize speech. In this paper the focusing is on speech recognition using fuzzy logic with other tools. This section presents a brief description of models that uses fuzzy logic and other tools to recognize speech. The section also presents a comparison between these models according to different criteria.

Speech recognition for Arabic language:

Researchers in [7] proposed a methodology to recognize the speech in their model using Hidden Marcov Model (HMM). They used neural network in their work to recognize Arabic isolated speech database and to compute the performance and compare it with 910 discrete HMM, hybrid Hidden Marcov Model / Multi-Layer Perceptron (HMM/MLP). They used the MLP to evaluate the HMM output probabilities and hybrid "*FCM/HMM/MLP*" approaches using the Fuzzy C-Means (FCM) algorithm to chunk the phonetic vectors. They used two kinds of data; the first dataset contained about 30 speakers. The words they uttered were their first name, their last name and were they live. Each word was repeated 10 times. The second database was specific words (view/new, save, save as/ save all) where each speaker repeated the words 10 times, thus the overall dataset consisted of 3900 words (3000 for training and 900 for cross validation to calculate the

[1]Adnan Shaout is a faculty member of the ECE Dept. at the University of Michigan-Dearborn. He is currently on sabbatical at PSUT

learning rate. In their work, the results shows that the hybrid model is better than the discrete model.

Speech recognition for Turkish language:

Researchers Avci and Akpolat [8] conducted a study that recognized speech. They applied an adaptive feature extraction method to extract the words (tokens) and a combination of wavelet packet signal processing and an adaptive network based fuzzy inference system (WPANFIS) to recognize single words. The reasons for using the wavelet signal processing were (1) the window size can be varied, (2) being enormous for slow frequencies and (3) it is used for irregular signals. The WPANFIS system was created from wavelet packet processing, artificial neural network and fuzzy logic. The authors tested their study using 20 individual Turkish speakers. The database contained 25 Turkish words. They used 500 words for training and for testing they used 2000 words with noise. The results of right classifications of words were about 92% for the sample.

Speech recognition for Japanese language:

In [9] a rule-based method was used to recognize the phonemes in the speech. The neural network was used to extract the features from the phonetics, and the learning algorithm that was used to learn the neural network was the back propagation algorithm. The output that was generated from the neural network was then used as an input to the fuzzy logic. The fuzzy logic were used to recognize the phoneme.

Speech recognition for Spanish language:

In [10] they used the feedforward neural network with one hidden layer to analyze the speech for unknown speakers. The learning algorithm they used to learn the neural network was Resilient Backpropagation (trainrp). A set of type-2 fuzzy logic rules was used for decision making to solve uncertainty. They also used genetic algorithm to optimize the number of layers and the nodes in the neural network. The dataset they used was in Spanish. They used 20 different words uttered by three different speakers. They also reduce the computation time by using modularity in their approach.

Speech recognition for German language:

In [11] the authors aimed to study the emotions from the speech signal; the technique they used to recognize the emotions was a neuro-fuzzy network with a weighted fuzzy membership function (NEWFM). In their study, they tried to classify four kinds of emotion signals. The feature extraction was done by the PRAAT software. They used NEWFM to select the feature from the speech signals and to create the fuzzy classifiers (FC). Three types of FCs were acquired at the end of training (FC1, FC2, and FC3) which was classified as high-and-low-arousal emotions, anger-joy, and sorrow-neutral emotion. The dataset was used is the Berlin Emotional-Speech Database, which contains five males speakers and five females speakers. In this work the authors focused on female voices. The dataset also contains seven types of sentiments which were anger, tedium, sickness, panic, joy, sorrow, and a neutral emotion.

A comparison among these different models with respect to many features is illustrated in table 1. Table 1 explains the dataset the authors used in their models to train and test their systems. The features and criteria that was used in the comparison includes the following:

- Gender and number of dataset speakers,
- The number of word repetition if any,
- The total number of words used for training and testing,
- The language of the data set,
- What tools are used to preprocess the words to make it ready to be used in the model, and finally
- The performance of each algorithm.

III. THE PROPOSED MODEL

In this paper, a new model is proposed to recognize speech based on fuzzy logic and Feed Forward neural network.

The first step that must be done in speech recognition is the speech preprocessing. The speech sounds that are going to be used in the model must be transformed from one shape to another [12]. Figure 1 shows the block diagram for a typical speech recognition model. Speech signals are continues [13] which means that the words are in analogue form. In order to deal with these signals we have to transform them into digital form [14]. The size of the sound wave is also large and there is a need to reduce the size of the wave. In this work, Fast Fourier Transform (FFT) is used to transform the signal from analogue to digital form. Figure 2 shows the FFT MATLAB code used for the transformation. The reason for using FFT over the discrete Fourier transform (DFT) is the time efficiency. In FFT the redundant calculations which DFT has are removed [15].

[1]Adnan Shaout is a faculty member of the ECE Dept. at the University of Michigan-Dearborn. He is currently on sabbatical at PSUT

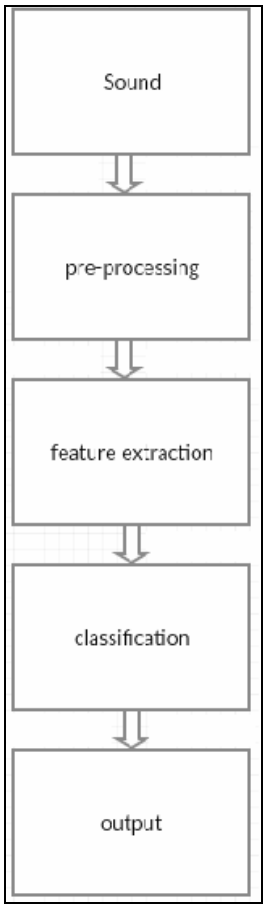

Figure 1: Common speech recognition model

```
1   %% FFT parameters
2   OverlapSize = 0.5;
3   FFTNo = 45;
4   NoOfWindows = 25;
5   NoOfFilters_F = floor(FFTNo/NoOfWindows+1);
6   fft0 = zeros(size(Words,2)*NoOfSamples,FFTNo);
7   %% FFT Training for all words
8
9       for jj = 1:size(Words,2);
10          for kk = 1:NoOfSamples
11              file_name = strcat(Words(jj),'_',int2str(kk));
12              Samples = eval(char(file_name));
13              rr = find(Samples);
14              Speech_Region = Samples(rr)/norm(Samples(rr));
15              WindowSize = floor((size(Speech_Region,1))/(NoOfWindows+1));
16              oo = 0;
17              for ll = 0:OverlapSize:(NoOfWindows-1)/2
18                  bb =
19  Speech_Region(floor(ll*WindowSize)+1:floor(ll*WindowSize)+WindowSize).*hamming(WindowS
20  ize);
21                  ft = log(abs(fft(bb,NoOfFilters_F)));
22                  fft0(kk,oo*NoOfFilters_F+1:oo*NoOfFilters_F+NoOfFilters_F)= ft;
23                  oo = oo + 1;
24              end
25          end
26      end
27
```

Figure 2: The FFT MATLAB code

Figure 2 shows the training sound data set are being preprocessed and the features are extracted from them by using the FFT transformation. Line 6 generates the zero matrixes to save the FFT (preprocess result). Line 9 shows the outer loop to run 2 times because we have two words. Line 11, creates a string as the name of the current sample indexed by the outer loop variable and the inner loop variable. Line 12, get all sample values. Line 13, extract the required sample from all samples. line14, generate the matrix that contains speech region by dividing the norm of the sample vector. In Line 15, moving window size. Lines 17-24: get the values of the moving window with overlap of 0.5 *window size then FFT and save values in the suitable place in the fft0 matrix mentioned above.

The vector size of the extracted information is [1x50] which means that the size of the sound sample is reduced by extracted the most important features which are the frequency elements which are the most 50 frequency elements in the sound wave. All of these samples are saved in a matrix called fft0 as shown in figure 3, which presents a small part of (sample) fft0 output. In neural network we have to feed the inputs column by column and the data existed in fft0 is row by row as shown in figure 3 where each row represents the features of each sound in the sample. Figure 3 shows the size matrixes as 500*50. Rows are the samples and columns are the features extracted for each sample. From this figure we can see that each sample has 50 features extracted in the form of (row vector). Therefore, a transpose will be done on this matrix in order to fit it as an input to the neural network.

Feed forward neural network is used with random initial weights to take the error from the training model. In the proposed methodology, the neural network is trained 1000 times. Figure 4 shows the first neural network training MATLAB code. The neural network will work in parallel with the fuzzy inference system. The first step is the construction of NN and to initialize the NN with suitable parameter values for number of epochs, initial weights, input data and target data. Thereafter training the model as many times as needed to obtain good results.

In the example code explained above, the number of training time is set to 10 for demonstration purposes only. But at the actual training which was done to get accurate results was 100 times. This number could be in thousands to get more accurate results, since it needs the estimation of many parameters such as the NN weights. The FIS is used to simplify finding the best weights suitable for the NN model.

Figure 5 shows the neural network structure. The error generated from the neural network is fed into the FIS each training time.

[1]Adnan Shaout is a faculty member of the ECE Dept. at the University of Michigan-Dearborn. He is currently on sabbatical at PSUT

	1	2	3	4	5	6	7
1	-16.0851	-26.7982	-10.0460	-15.6734	-9.7334	-13.1140	-9.5689
2	-15.7300	-28.3072	-10.9807	-16.4239	-13.7151	-14.4772	-12.2642
3	-15.6082	-27.1975	-14.2219	-14.9151	-13.6623	-14.5096	-12.8048
4	-13.7568	-27.4748	-9.1924	-12.6581	-9.7050	-12.8405	-10.8123
5	-14.6239	-27.9628	-12.0589	-12.4267	-10.4495	-12.6780	-12.4838
6	-16.0851	-26.7982	-10.0460	-15.6734	-9.7334	-13.1140	-9.5689
7	-15.7300	-28.3072	-10.9807	-16.4239	-13.7151	-14.4772	-12.2642
8	-15.6082	-27.1975	-14.2219	-14.9151	-13.6623	-14.5096	-12.8048
9	-13.7568	-27.4748	-9.1924	-12.6581	-9.7050	-12.8405	-10.8123
10	-14.6239	-27.9628	-12.0589	-12.4267	-10.4495	-12.6780	-12.4838
11	-16.0851	-26.7982	-10.0460	-15.6734	-9.7334	-13.1140	-9.5689
12	-15.7300	-28.3072	-10.9807	-16.4239	-13.7151	-14.4772	-12.2642
13	-15.6082	-27.1975	-14.2219	-14.9151	-13.6623	-14.5096	-12.8048
14	-13.7568	-27.4748	-9.1924	-12.6581	-9.7050	-12.8405	-10.8123
15	-14.6239	-27.9628	-12.0589	-12.4267	-10.4495	-12.6780	-12.4838
16	-16.0851	-26.7982	-10.0460	-15.6734	-9.7334	-13.1140	-9.5689
17	-15.7300	-28.3072	-10.9807	-16.4239	-13.7151	-14.4772	-12.2642
18	-15.6082	-27.1975	-14.2219	-14.9151	-13.6623	-14.5096	-12.8048
19	-13.7568	-27.4748	-9.1924	-12.6581	-9.7050	-12.8405	-10.8123
20	-14.6239	-27.9628	-12.0589	-12.4267	-10.4495	-12.6780	-12.4838

Figure 3: Part (sample) of FFT output

```
spread =0.1;
net = newff(input_data,target,[5 5],{'tansig','tansig','tansig'},'trainlm',
'learngdm');
net.trainparam.epochs=6000;
wei= load ('iw.mat');
for i =1:10
net=train(net,input_data,target);
net.IW{1}=wei.iw{1,1};
end
```

Figure 4: Neural network training code

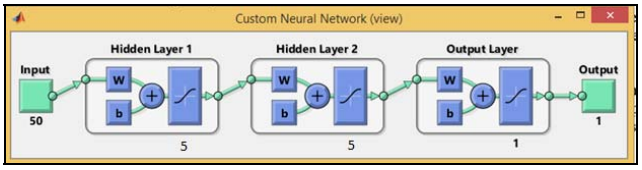

Figure 5: Neural network structure

These errors are entered into the FIS in each loop. The MATLAB fuzzy model is shown in figure 6. The inputs to the fuzzy inference system are the error fuzzy variable and FFT converted dataset. The fuzzy values used for the error variable are six values ordered from very good to very bad as it shown in figure 7. The second input was the dataset after converting it to a fuzzy variable. Figure 8 shows the linguistic values used for the second input. These two inputs are generated from a customized function which is used for classification and feature extraction [16]. The fuzzy inference system uses the set of rules that are shown in figure 9. The reason for using the fuzzy inference system is to choose the optimal weights to the neural network, therefore, the time required to choose the best optimal weight is reduced.

Thereafter, the output of the fuzzy inference system is the initial weights that are going be fed to the neural network. The type of neural network is feed forward. The algorithm that is used to train the neural network is learngdm. The number of hidden layers are two and each layer has 5 neurons, thus the number of iteration required to train the neural network is large. Figure 5 illustrate the structure of the neural network.

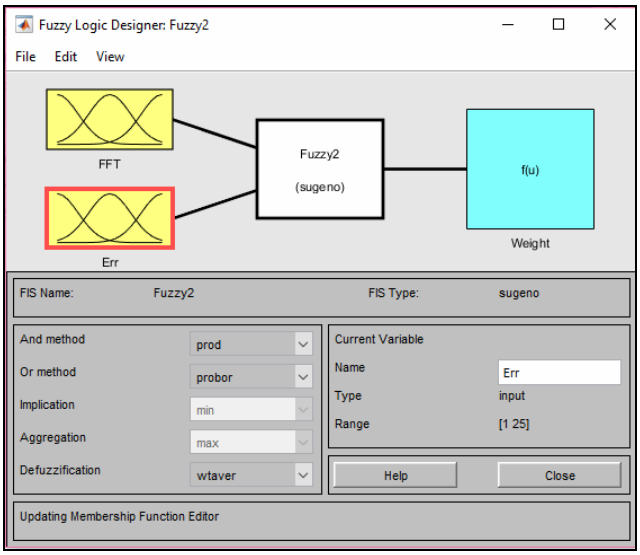

Figure 6: The fuzzy inference system

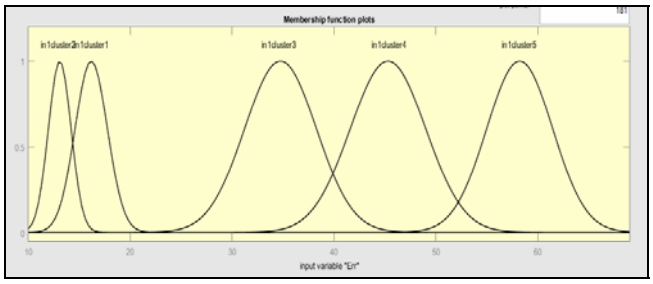

Figure 7: The linguistic values for the first fuzzy input varible

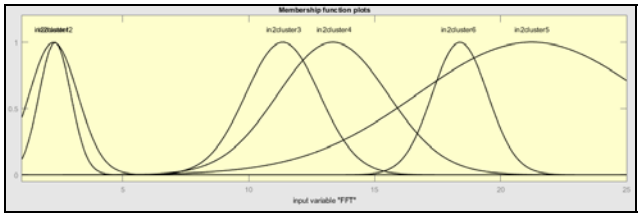

Figure 8: The linguistic values for the second fuzzy input variable

The overall model is shown in figure 10. The first step is the preprocessing step, which is done using the FFT transform, then the neural network is trained 1000 times. The output of the neural network is fed into the fuzzy logic inference system with the training dataset after converting it to a fuzzy variable. The fuzzy inference system will help in

[1]Adnan Shaout is a faculty member of the ECE Dept. at the University of Michigan-Dearborn. He is currently on sabbatical at PSUT

choosing the optimal NN weights instead of choosing it manually.

1. If (FFT is in1cluster1) and (Err is in2cluster1) then (Weight is out1cluster1) (1)
2. If (FFT is in1cluster2) and (Err is in2cluster2) then (Weight is out1cluster2) (1)
3. If (FFT is in1cluster3) and (Err is in2cluster3) then (Weight is out2cluster1) (1)
4. If (FFT is in1cluster4) and (Err is in2cluster4) then (Weight is out2cluster2) (1)
5. If (FFT is in1cluster5) and (Err is in2cluster5) then (Weight is out3cluster1) (1)

Figure 9: The fuzzy rule set

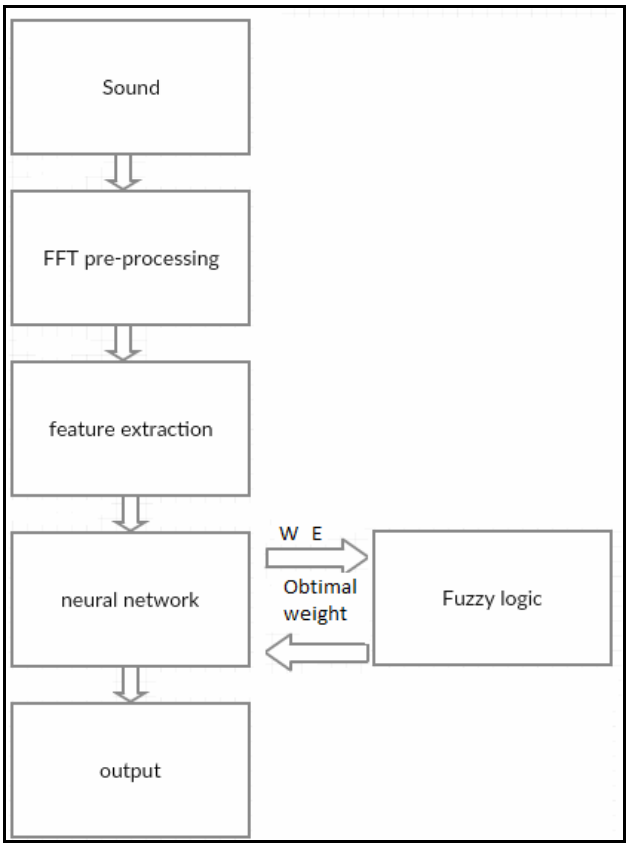

Figure 10: The overall proposed model

IV. DATASET

The dataset is made of 500 words of (Chokran and marhaba) with 250 each. Different speakers (2 men, 2 woman and 2 small girls) with 50 features each where feature are frequency elements. The data input size is 500*50 which is transposed into 50*500 when entered into the NN model.

V. PERFORMANCE EVALUATION

In this paper a new model is being proposed to recognize Arabic speech. This model uses the FFT transform, fuzzy logic and neural network.

The dataset has been used to train the system which consists of two Arabic words (مرحباً , شكراً). Each word is uttered 20 times, which makes the total number of training words equal 500 words. The speakers are a female, a male and a small girl.

Table 1: comparison between speech recognition models

	Dataset 1	**Dataset 2**	**Speakers**	**Word repetition**	**Total words number**	**language**	**Signal preprocessing**	**Tools used with fuzzy logic**	**results**
[7]	Speakers first, last names, and where they live	view/new, save, save as/save all	30 speakers	10 times	3000 for training 900 for testing	Arabic	HMM	neural network	The hybrid model is better than the discredit model
[8]	25 Turkish	-	20		500	Turkish	wavelet packet	an adaptive	92%

[1]Adnan Shaout is a faculty member of the ECE Dept. at the University of Michigan-Dearborn. He is currently on sabbatical at PSUT

	words		speakers		words for training 2000 for testing		signal processing	network based fuzzy inference system	
[9]	100 Japanese city names	-	2 males	twice	First uttered words were used for training the second words for testing	Japanese	Low-pass filter	Multi-layer perceptron neural network	80% of the errors occurring in conventional template matching,
[10]	100 Spanish words	-	Unknown speakers	once	100 words 20 words for training	Spanish	the Sound Forge 6.0 computer program	neural networks and genetic algorithm	96%
[11]	Berlin Emotional-Speech Database	-	Five males and five females	-	493 words 286 female voices 207 were of male voices.	German	the PRAAT software	Neural network	Over all accuracy 86%

The training dataset has been preprocessed and the features of these words has been extracted using FFT filter. The training words are fed into the neural network after the preprocessing step.

This model has been tested using the same two words (Chokran and Marhaban). To test the sound words, the test words must be preprocessed using FFT in the same way as the training dataset.

The trained neural network is used for testing. Figure 12 shows the test data preprocessing code. Line 1 is used to load the saved weights to be inputted to the NN trained model. Line 2 is to set the parameters of the NN. In line 3, X is the output of the trained model with input (fft_test) which contains the test sample values. In line 4, rounding result to get either 1 or 2 describing each word. Lines 5-15 check if the output is 1 then show the word 'chokran' and play sound to view the result, else do the same for the other word 'marhaba'.

```
1  Samples = eval(char(InputIn));
2  zz = find(Samples);
3  Speech_Region = Samples(zz);
4  fft_test = zeros(1,FFTNo);
5  WindowSize = floor((size(Speech_Region,1))/(NoOfWindows+1));
6  oo = 0;
7  for ll = 0:OverlapSize:(NoOfWindows-1)/2
8      bb =
9  Speech_Region(floor(ll*WindowSize)+1:floor(ll*WindowSize)+WindowSize).*hammin
10 g(WindowSize);
11     ft = log(abs(fft(bb,NoOfFilters_F)));
12     fft_test(1,oo*NoOfFilters_F+1:oo*NoOfFilters_F+NoOfFilters_F) = ft;
13     oo = oo + 1;
14 end
```

Figure 11: Code for testing data preprocessing

The proposed model recognize all the testing data, therefore the speakers of testing data are the same speakers as training data, which make the recognition easier for the neural network than making the test with different speakers. Figure 13 shows the result of testing one word of the model. The figure shows that the output matched the target of the neural network. The accuracy of the model is 98%. If we test this model on different speaker sounds then the accuracy may be reduces.

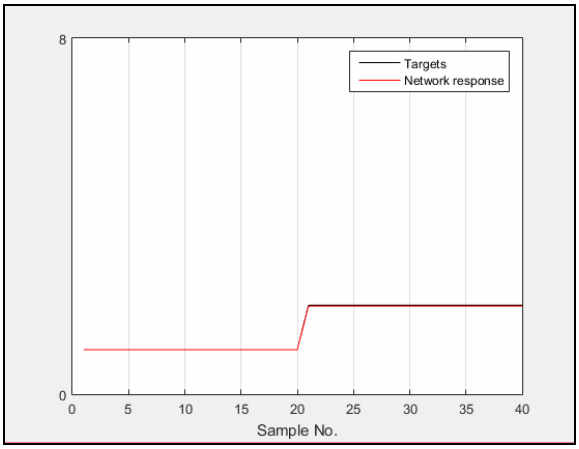

Figure 12: results of one testing word

VI. CONCLUSION

In this paper a new model has been proposed to recognize Arabic speech. The sound words have been preprocessed using FFT transformation, the neural network has been trained 10 times to extract the errors so that these errors can be used as an input to the fuzzy system so that the fuzzy inference system will generate from the fuzzy system an optimized initial weight instead of starting with random initial weights so that these generated weights are used to train the neural network. The model has been tested using two Arabic words from three different persons. Each has uttered the two words 50 times. The results shows that the accuracy was 98%. The test dataset is uttered by the same speakers for the training dataset which made the recognition easier in the neural network.

According to this proposed model, there is still a need to reduce the number of epochs and the training time for the neural network. The future work will be to enhance the neural network initial weights through the use of FIS. The FIS would be applied to generate the optimal initial weights thus the training time could be reduced. Furthermore, the FIS could be used to determine the termination of the training model. The FIS can also be used to classify the NN weights and determine the optimum set of weights.

VII. REFERENCES

1. Hirsch, H.-G. and D. Pearce. *The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions*. in *ASR2000-Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW)*. 2000.
2. Collobert, R. and J. Weston. *A unified architecture for natural language processing: Deep neural networks with multitask learning*. in *Proceedings of the 25th international conference on Machine learning*. 2008. ACM.
3. Rabiner, L.R., *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE, 1989. **77**(2): p. 257-286.
4. Paul, A.M., et al., *Speech Controlled Automatic Slide Change.* IJRCCT, 2016. **5**(3): p. 084-087.
5. Steensgaard-Madsen, J., *Analog-to-digital converter*. 2016, Google Patents.
6. Cochran, W.T., et al., *What is the fast Fourier transform?* Proceedings of the IEEE, 1967. **55**(10): p. 1664-1674.
7. Lazli, L. and M. Sellami, *Connectionist probability estimators in HMM arabic speech recognition using fuzzy logic*, in *Machine Learning and Data Mining in Pattern Recognition*. 2003, Springer. p. 379-388.
8. Avci, E. and Z.H. Akpolat, *Speech recognition using a wavelet packet adaptive network based fuzzy inference system.* Expert Systems with Applications, 2006. **31**(3): p. 495-503.
9. Amano, A., et al. *On the use of neural networks and fuzzy logic in speech recognition*. in *Neural Networks, 1989. IJCNN., International Joint Conference on*. 1989. IEEE.
10. Melin, P. and O. Castillo, *Voice recognition with neural networks, fuzzy logic and genetic algorithms*, in *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing*. 2005, Springer. p. 223-240.
11. Viswanathan, M., et al. *Emotional-speech recognition using the neuro-fuzzy network*. in *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*. 2012. ACM.
12. Lippmann, R.P., *Review of neural networks for speech recognition.* Neural computation, 1989. **1**(1): p. 1-38.
13. Andre-Obrecht, R., *A new statistical approach for the automatic segmentation of continuous speech signals.* Acoustics, Speech and Signal Processing, IEEE Transactions on, 1988. **36**(1): p. 29-40.
14. Picone, J.W., *Signal modeling techniques in speech recognition.* Proceedings of the IEEE, 1993. **81**(9): p. 1215-1247.
15. Johnson, S.G. and M. Frigo, *A modified split-radix FFT with fewer arithmetic operations.* Signal Processing, IEEE Transactions on, 2007. **55**(1): p. 111-119.
16. Mathworks. *Neuro-fuzzy classifier*. 2010 [cited 2016 3 May]; Available from: http://www.mathworks.com/matlabcentral/fileexchange/29043-neuro-fuzzy-classifier/content/custmf1.m.

[1]Adnan Shaout is a faculty member of the ECE Dept. at the University of Michigan-Dearborn. He is currently on sabbatical at PSUT