Comparison of different neural and fuzzy models for recognizing vowels within uncontrolled environments

Atul Shrotriya
Department of Mechanical and Aerospace Engineering
The University of Texas at Arlington
Arlington, United States
atul.shrotriya@mavs.uta.edu

***Abstract*—Speech recognition has long been used and implemented in controlled environments. It is quickly becoming mainstream with the advancements in deep learning techniques. This paper compares different basic neural and fuzzy logic models to access which basic techniques are best suited for recognizing vowels which are the core of speech processing in many methodologies. With conventional equipment used for recording in an uncontrolled environment, this paper defines the challenges which can be faced in developing reliable and robust models without using large amounts of pre-recorded data. It is found that fuzzy systems with 3 generalized bell membership functions in two middle layers provide the closest outputs to the expected values.**

I. Introduction

Speech recognition is quickly becoming mainstream in the today's world with a wide variety of devices understanding the user's words for different purposes. An audio interface allows the users to keep their hands and eyes free and focus on other tasks. [8] Although speech recognition has existed and been used for many years in controlled environments, recent developments in the field of deep learning has allowed it to be included in the mainstream use with the help of conventional devices. [9] The availability of large amounts of data allows various organizations to train the deep learning models which can provide a good accuracy even in the presence of noise. These models are regularly updated with edge cases to include different accents and speech patterns used by users. [8,9]

A basic part of speech recognition is vowels. These can be analyzed to check different factors for example, the same vowel may be spoken differently by the same person under different conditions. [10] Vowels can also be accessed for improving speech [11]. Improvement in the basic ASR pipeline is made by replacing existing machine learning models with deep learning models [10]. This paper aims to create different models using feedforward neural networks with 3 and 10 neurons and fuzzy logic systems with triangular and generalized bell membership functions. These methods are chosen as they can encompass a variety of basic techniques. Determining this will help in applying deep learning models in a robust method and provide more accurate outputs.

II. Vowel Recognition Using Uniform Data

Voice recorded in a controlled environment is often different than natural voice recorded by usual microphones. Upon plotting, it can be observed that the amplitudes are very similar with the bin size remaining constant.

A. Determination of Vowel Sequence using Visual Inspection

The audio files can be easily extracted in the form of information which is understandable by MATLAB. This can be achieved through the use of audioread() command. For this step, the data was already available in an excel sheet [1]. The following steps were taken to recognize the vowels:

1. The signal is divided into 8 equal bins of 1 second each. With each bin containing 1000 data points
2. Discrete Fourier transform (DFT) is done on each bin using the fft() command.[2] The number of points used for calculating the Fourier transform should be a power of 2 [3] and is hence taken as 1024 i.e., 2^{11}. The sampling rate was taken as 0.001 seconds or 1 millisecond.
3. Absolute values of the FFT are plotted for half the frequencies as it repeats [4].
4. Peaks can be easily observed on the plots and the top two peaks mark the formant frequencies required to determine the sequence of vowels.

B. Determination of vowel sequence using findpeaks command

MATLAB has a useful findpeaks() command which allows to determine the peaks within a dataset. Due to the presence of noise, there are a lot of peaks however, these can

The University of Texas at Arlington

be separated easily by applying a magnitude threshold of 200 for consideration.

Based on the sequence of peaks, a threshold value of 10 is applied to the frequency to determine the vowel sequence. The findpeaks() function provides peaks that are slightly shifted to higher frequencies. Thus, only the higher end needs to be considered for recognizing the vowel.

III. Vowel Recognition Using Non-Uniform Data

Non-uniform data was recorded for all vowels A,E, I, O and U with 10 samples of each vowel. The samples were varied in pitch and length to observe the effects on detection. The following steps were taken to analyze the samples.

A. Preparation of Training Data for the networks

1. Audio file was converted into a recognizable matrix format for MATLAB
2. The audio recorded was in stereo format. Convolution of audio was done to get a single matrix. Other software and techniques can also be used to convert stereo signal to mono if the sampling rate is different due to any reason.
3. Discrete Fourier Transform (DFT) was done on the new data using the fft() command. Different number of points were tried to calculate the FFT. Some files provided a flat graph when the number of points was below 4096 or 2^{13}. To ensure no loss in data a minimum of 8192 points were required. However, this resulted in some noise being magnified and thus 16384 points were used.
4. Signals of different length, cannot be stored in the same matrix without the addition of zero values. Another method was used to avoid adding synthetic data to the original signal. Here, each matrix was created differently and called through a loop using cell arrays.
5. Comparison of Power Spectral Density (PSD) plots was done between convoluted data and by neglecting the second channel. Both helped in filtering the data but magnified the differences in amplitudes of recordings due to squaring of data.
6. The findpeaks() command, was tried again to separate the peaks. It failed due to the following reasons, the threshold value on magnitudes could not be applied as some magnitudes of some files were so small that they were less than the noise of other files. There were multiple peaks near the formant frequencies. It may be possible to introduce another threshold to neglect values if they are within a certain frequency range. To evade the issue with varying magnitudes, normalization can be applied.
7. Sample rate, number of points were changed to compare their effects on the plots. Reducing the sample rate reduced the difference between formant frequencies and thus, sample rate was kept at 8000 samples to discern the frequencies aptly. Reducing the number of points used for calculating the FFT resulted in loss of data and was thus kept at 16384.
8. The sample size was standardized at 15000 points because it was able to collect most of the relevant information. It is not feasible to do a discrete Fourier transform on the entire data because it identifies the frequency of consistent background noise. While most noise is ideally considered white noise, the region where the non-uniform data was recorded was not controlled. Thus there were consistent frequencies from the exhaust fans which defeats the purpose discrete Fourier transform irrelevan

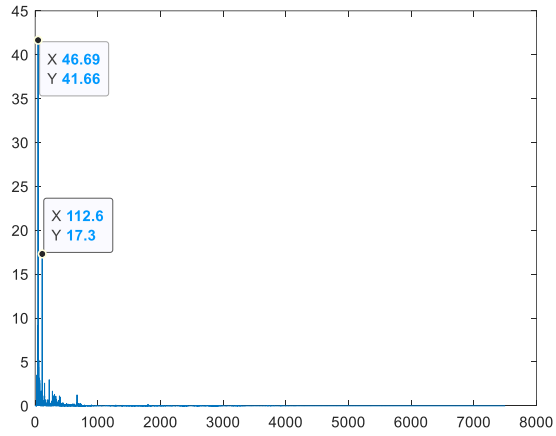

Figure 1. Discrete Fourier Transform (DFT) recognizing exhaust fan frequencies from the background noise

9. The start of points had to be identified visually because the vowels were spoken at different times with different tones and volumes. For comparison, both Fig. 2 and Fig. 3 consist of the same vowel “A” but have vastly varying amplitudes as observed.

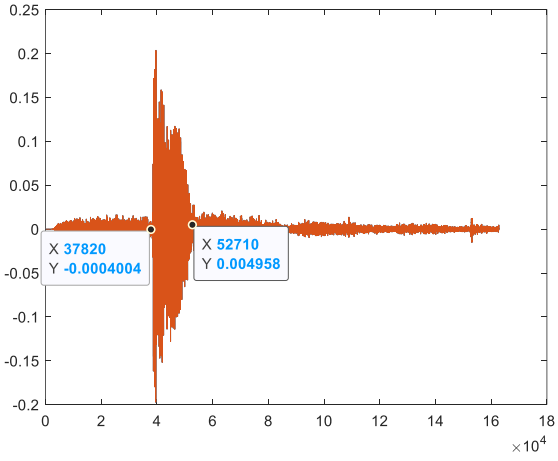

Figure 2. Plot of recording of vowel “A” with amplitudes reaching till 0.2 in magnitude

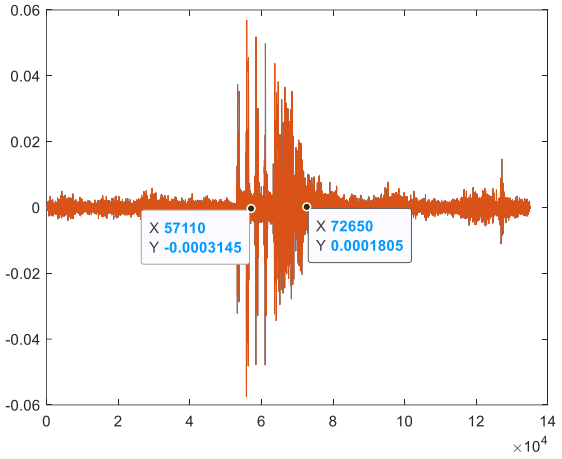

Figure 3. Plot of recording of vowel "A" with amplitudes reaching only till 0.06 in magnitude

10. Data from different channels both channels was compared and it was observed that they overlap closely. Thus, one channel was neglected and only one channel was retained for performing the discrete Fourier transform (DFT). It was also observed that by using a regular laptop microphone, there were some glitches in recording. As seen in Figure 3, the voice was recorded in fragmented intervals. Thus, these were neglected and data was taken once the signal was stable as shown in Figure 4.

11. Formants were identified using visual inspection method. Since the data set was brought to the same size, it was put into the same matrix for ease of processing.

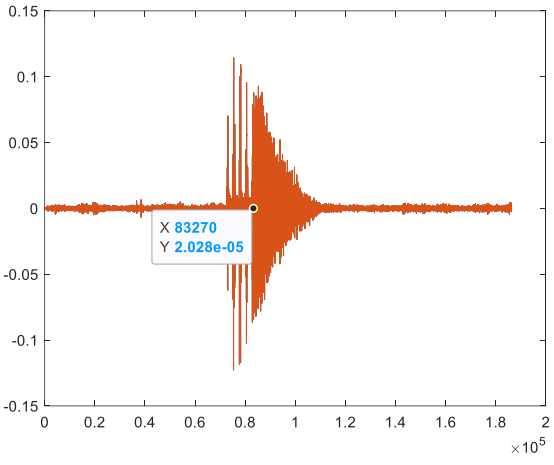

Figure 4. Starting point for considering data in case of fragmented recording

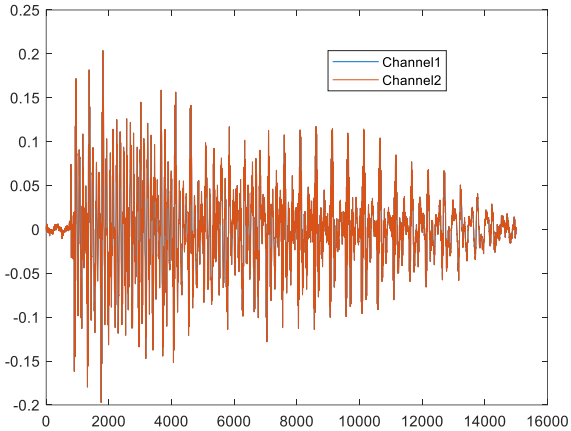

Figure 5. Plot of left and right channels of recording showing a perfect overlap of the signals

12. The peaks of were entered into the training matrix and a corresponding target vector was also formed. These were used for training the neural networks described next in this paper. The middle of the peaks were considered to obtain the average frequency of the vowel formants. The varying pitch, styles and amplitudes of the recorded vowels cause a variation of around 25 percent in the formant frequencies. It was also observed that the comparison of amplitudes at the formant frequencies is reversed depending on how the vowel was spoken or which parts were stressed on. The maximum variation was observed in the first formant of the vowel "I" with the variation of 36.3 percent. It was sometimes recorded as "aye" and at other times as "eye" to encompass different speaking styles.

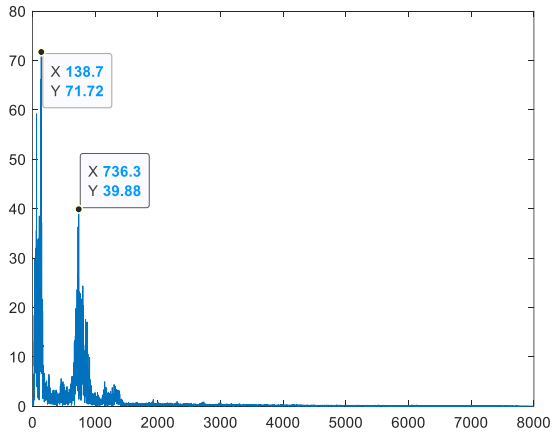

Figure 6. Depiction of visual method used for extracting the formant frequencies from the plots

IV. TRAINING THE MODELS

Four different types of networks were trained to form models and recognize the vowels. These models were based on neural and fuzzy logic.

A. *Neural Network based on 3 perceptrons*

1. The training vector was plotted in the 2D plane and it was observed that the vowel "I" varies the most which can also be predicted by looking at the formant frequencies.

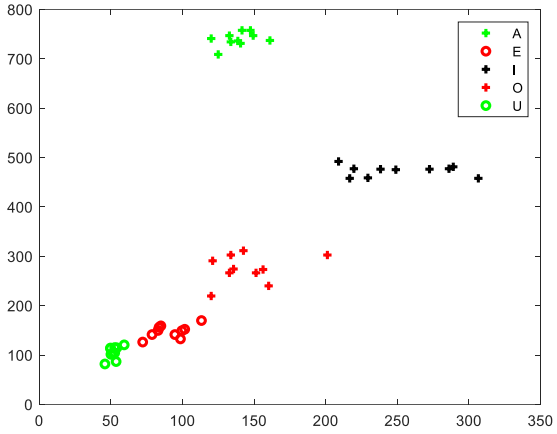

Figure 7. Plot the formants in a 2-dimensional plane

2. The network is trained with 3 perceptrons for 100 epochs and the output is obtained. However, the output is not entirely accurate and only has an accuracy as follows:
    a. 100 percent for the vowel "A"
    b. 50 percent for the vowel "E"
    c. 90 percent for the vowel "I"
    d. 0 percent for the vowel "O"
    e. 60 percent for the vowel "U"
3. The accuracy of output for vowel "O" is zero because the network classified it as one of the other four vowels and didn't assign a unique output for it.
4. The network is trained again for 10000 epochs and the following results are obtained:
    a. 100 percent for the vowel "A"
    b. 60 percent for the vowel "E"
    c. 60 percent for the vowel "I"
    d. 60 percent for the vowel "O"
    e. 90 percent for the vowel "U"
5. The reduction in the accuracy of the vowel "I" may be caused due to overfitting by the neural network over too many epochs.

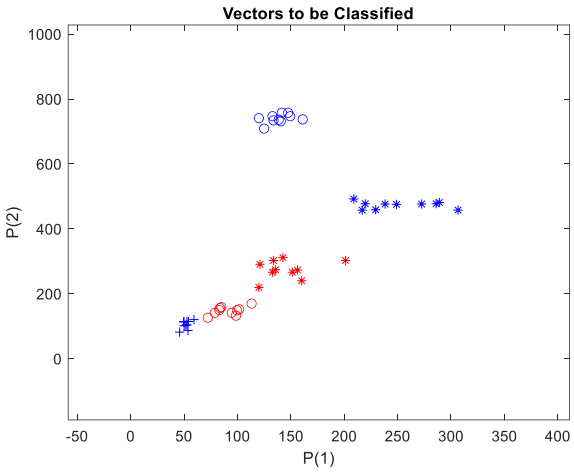

Figure 8. Simulation of the perceptron model with 3 perceptrons

B. *Generating models using the neural network fitting toolbox in MATLAB*

1. The data is first arranged into different input and output matrices as required by the neural network toolbox. For training all fuzzy and neural models, the following convention is used to classify the outputs A=1, E=2, I=3, O=4, U=5.
2. The network is trained using 3 neurons and the training stopped after 34 iterations or epochs with the mean squared error being 0.42.

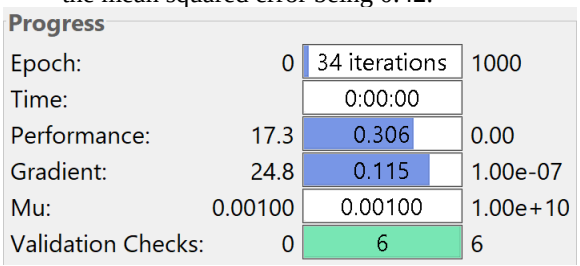

Figure 9. Output provided by the toolbox on completion of training the feed forward neural network with 3 neurons

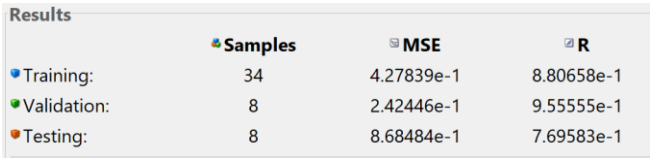

Figure 10. Error data provided by the toolbox on completion of training the feed forward neural network with 3 neurons

3. The training algorithm used is Levenberg-Marquardt as it is the most effective option available.[5] It combines gradient descent and Gauss Newton methods to fit curves.[6]
4. Plots for error histogram and regression are obtained using the toolbox buttons.

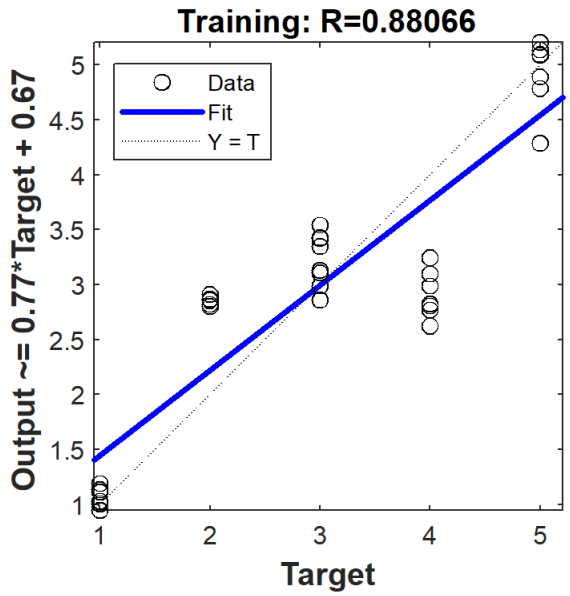

Figure 11. Regression plot for training data of feed forward neural network with 3 neurons

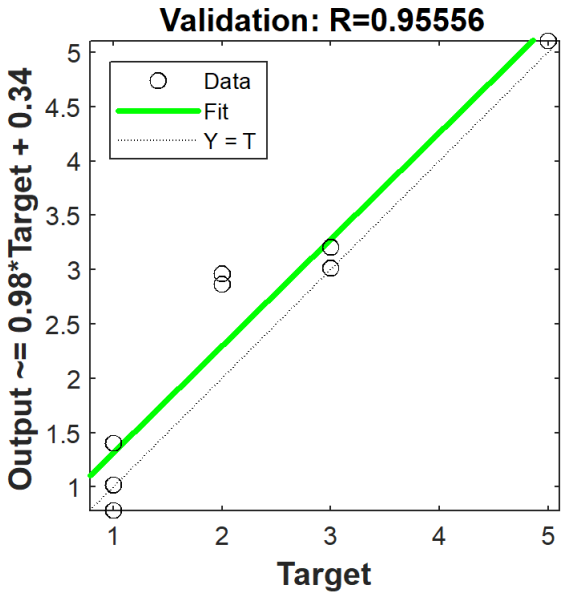

Figure 12. Regression plot for validation data of feed forward neural network with 3 neurons

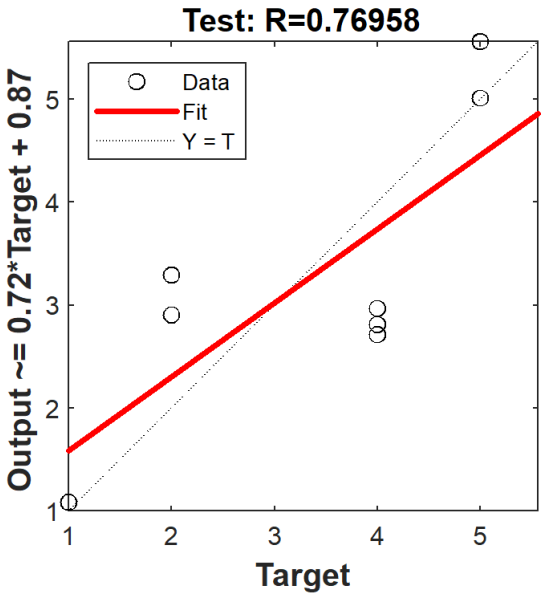

Figure 13. Regression plot for validation data of feed forward neural network with 3 neurons

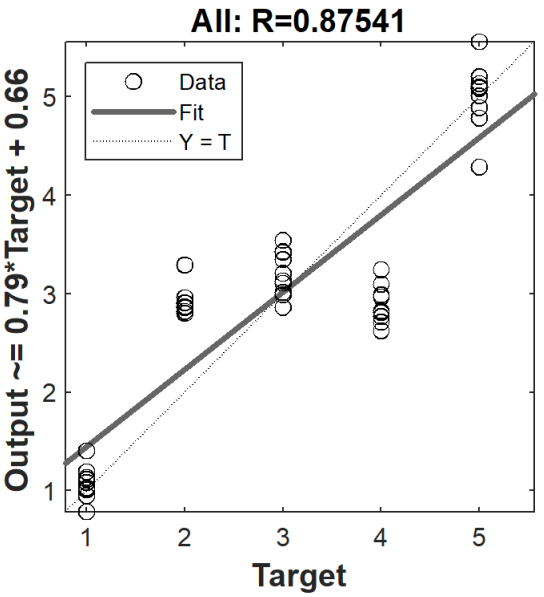

Figure 14. Regression plot for all data of feed forward neural network with 3 neurons

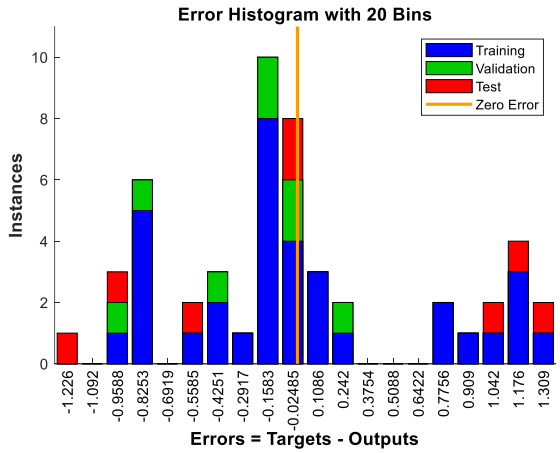

Figure 15. Error histogram for feed forward neural network with 3 neurons

5. The generated model is retrieved in the form of a Simulink block for testing further.

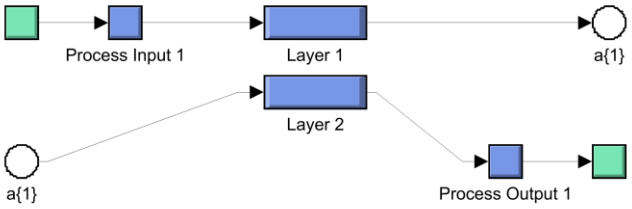

Figure 16. Neural Network Model in Simulink for feed forward neural network with 3 neurons

6. A simple Simulink model is developed for testing the generated neural network as shown below.

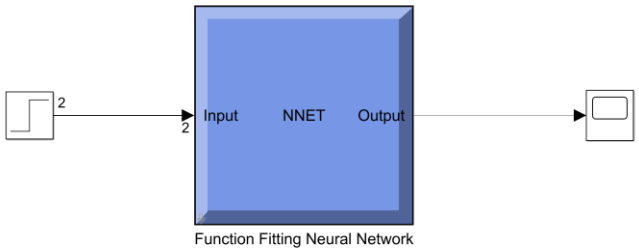

Figure 17. Simulink diagram for testing of feed forward neural network with 3 neurons

7. The system is easily able to recognize the formant frequencies near the vowel "A" but doesn't do an effective job at recognizing the other variables. However, it remains closer to the expected value instead of a wrong value.

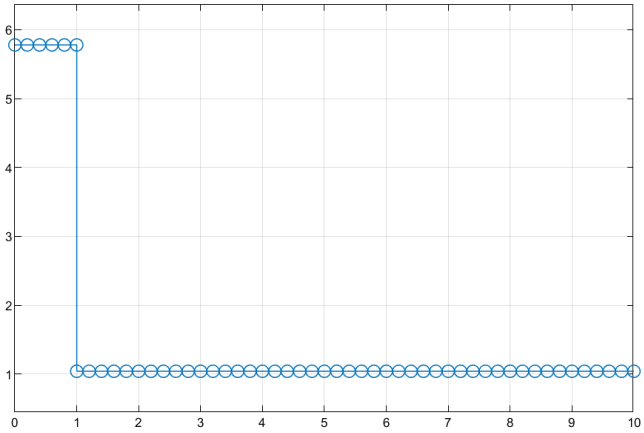

Figure 18. Neural Network output for input values near vowel "A"

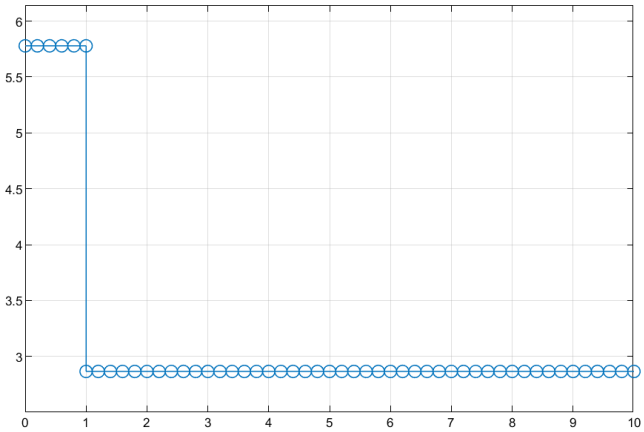

Figure 19. Neural Network output for values near vowel "E"

8. A new model is trained consisting of 10 neurons using the same training dataset. The training stops after 14 iterations or epochs and a mean squared error of 0.3105 is obtained.

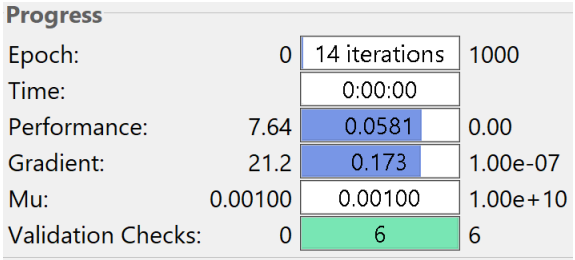

Figure 20. Output provided by the toolbox on completion of training the feed forward neural network with 10 neurons

9. The Simulink block is tested and the output is closer to the expected values than previously for the same input values. The values used for testing the model are not present in the training data.

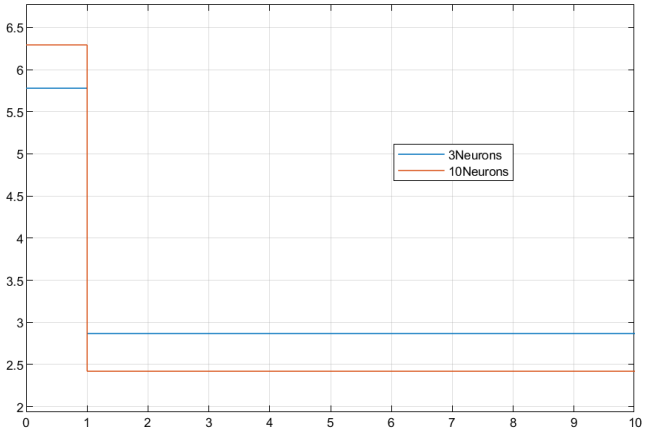

Figure 21. Output comparison for the same values of vowel "E"

C. *Generating models using the Neuro-Fuzzy Toolbox in MATLAB*

1. Data is combined into a single set acceptable by the neuro-fuzzy toolbox with the last column representing the outputs.
2. A Fuzzy Inference System with 3 3 triangular membership functions (TMF) is used to train the model. The membership function type is kept linear throughout all the models.
3. Average testing error of 0.27 is observed in this case.

Figure 22. Output plot of training data for 3 3 TMF

4. A new model is trained using 5 5 triangular membership functions (TMF) and the same dataset.
5. Average testing error is reduced to 0.183 for this case.

Figure 23. Output plot of training data for 5 5 TMF

6. Triangular Membership functions have firm boundaries and are not as good as the generalized bell functions (GBELL) or other membership functions with smooth boundaries. [7] Two more systems are trained with the same dataset using the generalized bell functions as 3 3 and 5 5 respectively.
7. The average testing error for the generalized bell (GBELL) model with 3 3 functions was 0.249

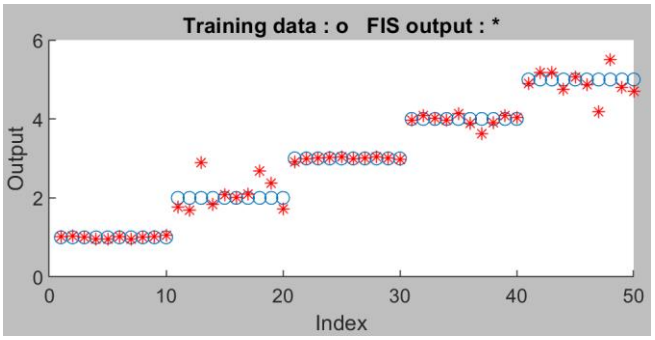

Figure 24. Output plot of training data for GBELL 3 3

8. The average testing error for the generalized bell (GBELL) model with 5 5 functions was 0.045

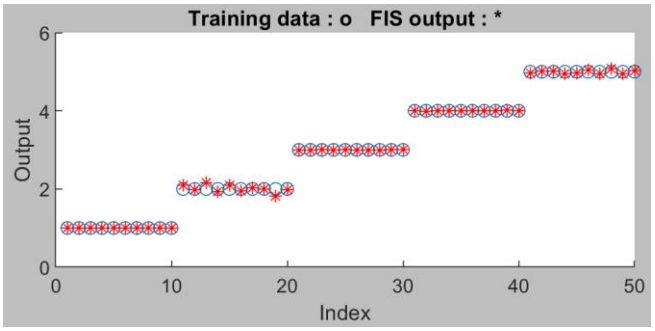

Figure 25. Output plot of training data for GBELL 5 5

9. All the models were exported to workspace and tested with the same testing data used on the neural networks.
10. On checking the output, almost all the fuzzy systems are closer to the output as compared to the neural network models. As expected from the average testing error values, the best output for the testing data is provided by the GBELL 55 model followed by the TMF 55 model. This is due to the larger number of membership functions which help in fitting more data accurately into the model.

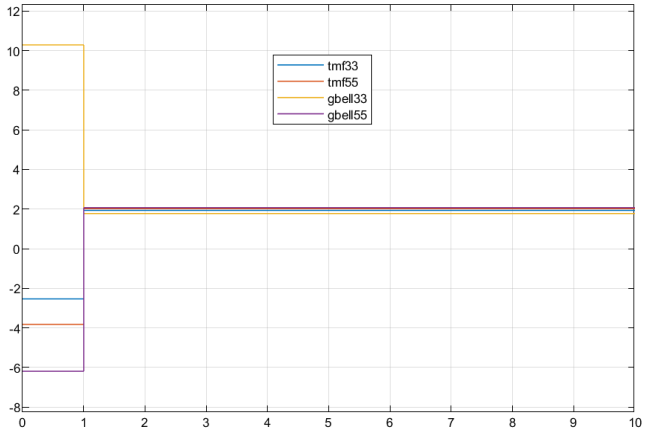

Figure 26. Output comparison for Fuzzy systems

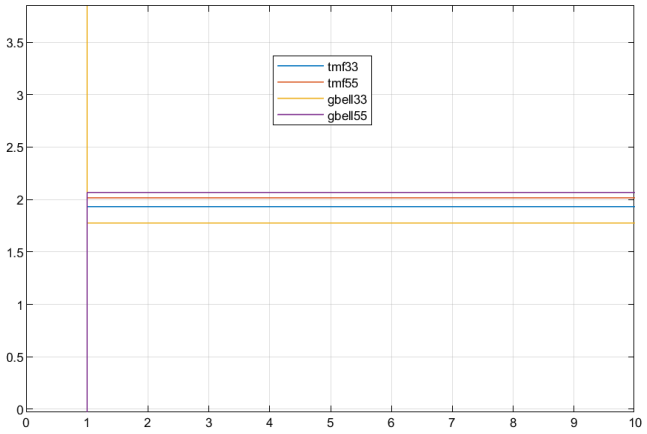

Figure 27. Output comparison for fuzzy systems with improved resolution

V. Comparison of All the Implemented Methods

1. The neural network and fuzzy models are imported into the Simulink diagram and their outputs are plotted against each other. The testing inputs are synthetically designed to be within the range of the expected formant ranges which were observed in the training data.
2. A simple Simulink block with step input was used to compare all the models against each other.
3. 10 different testing values were used to observe the outputs and make comparisons. The testing values were taken randomly between the vowels, “E”, “I”, “O” and “U”. The values near the vowel “A” were not used for testing as it is classified effectively by all methods.
4. The variation in amplitude from the actual expected amplitude was noted to check for the best and least effective models.

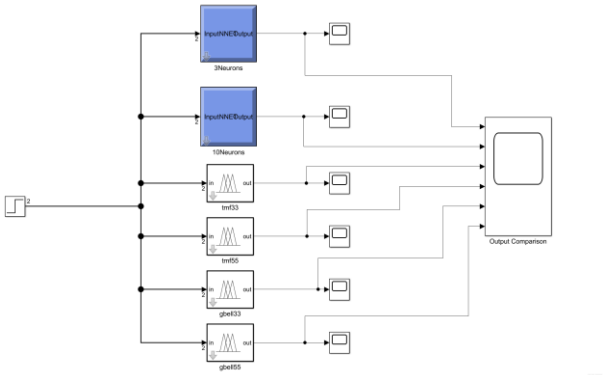

Figure 28. Simulink model for comparison of outputs

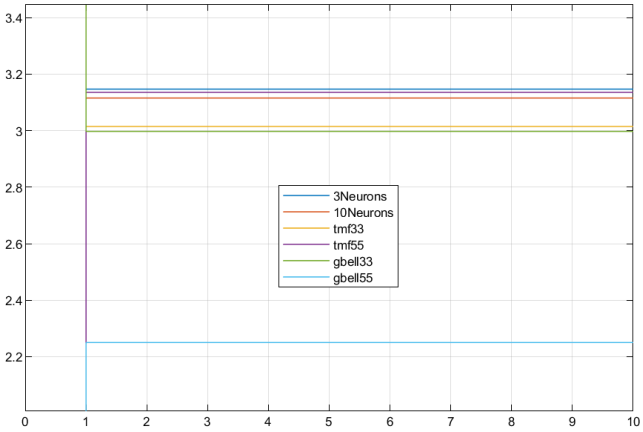

Figure 29. Magnified plot of output comparison using new testing data

VI. Conclusions and Discussion

Based on the comparison plots, it was observed that the overall best model for estimating the outputs was the GBELL 33 closely followed by TMF 55. It should be noted that for a few cases, the TMF 55 provided the closest output so the random selection of testing data may have slightly effected the results. However, both models predicted the values accurately with GBELL 33 remaining within 0.2 of the expected output for all cases.

The values given by models vary a lot before the input is provided. This is due to each model fitting the data in a different manner. The neural networks also provide acceptable outputs remaining within 0.3 of the expected output for all cases.

The GBELL 55 function does not perform well despite having the lowest average mean squared error during training. This occurs due to overfitting the data. This has also been observed in other previous experiments [7].

Based on the results obtained, it is concluded that the generalized bell membership functions with 3 functions in each layer are best suited for recognizing vowels within uncontrolled environments. These functions can be further improved and coupled with deep learning methods to improve the conventional ASR pipelines. [8]

These systems can be implemented with limited resources and can be trained with relatively small amounts of data. A good application of this might be implemented in space systems to save energy on communication and processing hardware.

ACKNOWLEDGMENT

Thanks to the University of Texas at Arlington for providing the necessary resources to perform this research. I would like to thank my research advisor Dr. David A. Hullender and my academic supervisor Dr. Seiichi Nomura for introducing me to the concept of Discrete Fourier Transforms and signal processing in detail which proved to be crucial for this paper. I'm immensely grateful to my professors, Dr. Frank Lewis, Dr. Michael Niestroy, Dr. Kamesh Subbarao and Dr. Robert Woods for explaining the concepts to me in a practical manner which allowed me to implement them in a versatile manner on this project.

REFERENCES

[1] F. L. Lewis and C. He, "Homework 3 - DFT Analysis for speech recognition and Fault diagnosis," Intelligent Control Systems EE 5322, Department of Electrical Engineering, The University of Texas at Arlington, Arlington, Sep., 24, 2020

[2] Mathworks, "FFT Documentation," retrieved on Dec. 5, 2020 from https://www.mathworks.com/help/matlab/ref/fft

[3] D. A. Hullender, "Stochastic systems," Dynamic Systems Modeling ME 5305, Department of Mechanical and Aerospace Engineering, The University of Texas at Arlington, Nov. 14, 2019

[4] S. Nomura – "Discrete Fourier Transforms," Analytic Methods in Engineering ME 5331, Department of Mechanical and Aerospace Engineering, The University of Texas at Arlington, Oct., 2019

[5] M. A. Niestroy – "Neural Networks," System Identification and Estimation EE 5327, Department of Electrical Engineering, The University of Texas at Arlington, Nov., 10, 2020

[6] Marquardt, D., "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," SIAM Journal on Applied Mathematics, Vol. 11, No. 2, June 1963, pp. 431–441

[7] A. Shrotriya and M. A. Niestroy, "Homework 6 – Fuzzy Systems," System Identification and Estimation EE 5327, Department of Electrical Engineering, The University of Texas at Arlington, Arlington, Dec., 01, 2020

[8] A. Geitgey, "Machine Learning is Fun," second edition, United States, 2018, pp. 158-170

[9] A. Coates (Baidu Inc.,), "Speech Recognition", Bay Area Deep Learning School Day 2 at CEMEX auditorium, Stanford University, Stanford, California, Sep., 25, 2016

[10] Adi Y, Keshet J, Goldrick M. VOWEL DURATION MEASUREMENT USING DEEP NEURAL NETWORKS. IEEE Int Workshop Mach Learn Signal Process. 2015; 2015:10.1109/MLSP.2015.7324331.

[11] C. Kim and W. Sung, "Vowel Pronunciation Accuracy Checking System Based on Phoneme Segmentation and Formants Extraction," Seoul National University, Seoul, Korea