Journal of Engineering and Applied Sciences 12 (Special Issue 7): 8207-8213, 2017
ISSN: 1816-949X

End-to-End Connectionist Approach for Audio Command Recognition System for Robot Control

Shweta Savdekar and Ramchand Hablani
Department of Computer Science Engineering,
Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India

Abstract: This work studies, the application of feedforward Artificial Neural Network (ANN) to address the task of isolated command recognition system for robot control. The study is based on the end-to-end ANN approach for speech recognition system which is interfaced to the Robot Firebird VI for its movement control through voice command. Initially, the database is created by manually recording different instances for each command. For the front end processing of the recorded signals, Mel Frequency Cepstral Coefficient (MFCC) is used as the feature extraction technique. Extracted features are processed by artificial neural network model which works as a classifier. The recognized commands are then communicated to the Firebird VI robot via. wireless connectivity. As a part of the study undertaken, the study also throws light on ways to enhance the performance of the system. The performance of the system is analyzed in terms of the accuracy of the system to correctly recognize the spoken commands and the robot taking the corresponding action.

Key words: Artificial Neural Network (ANN), Deep Neural Network (DNN), Recurrent Neural Network (RNN), Hidden Markov Model (HMM), Gaussian Mixture Model (GMM) back propagation, Mel Cepstral Frequency Coefficients (MFCCs)

INTRODUCTION

Research in speech processing and communication for the most part, can be said to be fuelled by people's desire to build mechanical models to emulate human verbal communication capabilities. Human Robot Interaction (HRI) is an interesting application of Automatic Speech Recognition (ASR) system. The focus has been to develop an interactive application which can be used to enable speech input control for different applications (Zhou *et al*., 1994). Voice Commander for Windows is an interactive application developed by the researchers to enable speech input control for all applications running under Microsoft's Windows 3.1 environment.

Motivation: Motivated by the recent success of artificial neural network in speech recognition, the study deals with the application of feedforward Artificial Neural Network (ANN) for isolated words identification system. Sharing analogy with biological neural network they are the conceptual stepping stone on the path that power models like Deep Neural Network (DNN), Recurrent Neural Network (RNN) and many other applications. Since last decade, neural network applications have witnessed a tremendous surge and acceptance.

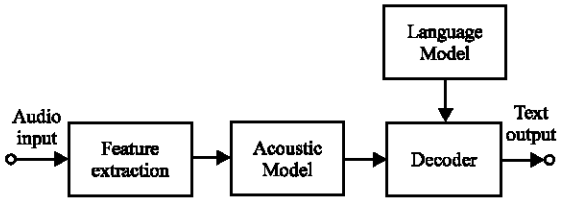

Fig. 1: Block diagram of general Automatic Speech Recognition (ASR) system

Literature review: Automatic Speech Recognition (ASR) or computer speech recognition is the process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a computer program.

General framework for speech recognition system: The task of speech recognition is generally done with combination of different subtasks. Figure 1 illustrates general block diagram of speech recognition system.

Audio input: The user speaks into the microphone. The microphone acts as a transducer that converts analogue speech signal into electrical signal. This captured sound signal is converted into digital signal with the help of

Corresponding Author: Shweta Savdekar, Department of Computer Science Engineering,
Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India

sound card in the computer system. The digitized sequence is thus used as the audio input for further processing.

Feature extraction: The purpose of feature extraction is to provide a compact vector which represents phonetic information while suppressing other unwanted information present in the speech signal.

Acoustic model: The relationship between an audio signal and the phonemes or other linguistic units that make up speech is represented by the acoustic model. It is created by taking audio recording of speech and their text transcriptions and using software to create statistical representations of the sounds that make up each word.

Language model: Language model is a probability distribution over sequences of words. The language model helps to distinguish between different words when the recognition algorithm tries to match the sounds with word sequences.

Decoder: Given some spoken input for recognition, decoder searches for the correct word sequence among all possibilities and gives the recognized output.

Text output: The recognized output is obtained in the form of text and can be applied for different applications.

Models for automatic speech recognition

Hidden Markov Model (HMM): General-purpose speech recognition systems are based on Hidden Markov Models (HMM). It is a statistical method which models the speech utterance sequence for training and also generates the sequence for the process of recognition. The measure in term of maximum likelihood probability is computed and the larger the value of measure, the higher is the chance of similarity between the testing and the training patterns. It is considered as a powerful tool for representing probability distributions over sequence of observation (Baker *et al*., 2009).

Artificial neural network: Artificial Neural Networks (ANNs) also known as connectionist systems are computing systems that borrowed its inspiration from thebiological neural networksof animal brains.

An Artificial Neural Network is an interconnected group of nodes which respond in parallel to a set of input signals given to each input node (Fig. 2).

Each circular node represents anartificial neuronand an arrow represents a connection from the output of oneneuron to the input of another. The leftmost layer in the network is called the input layer and the neurons within the layer are calledinput neurons. The rightmost oroutputlayer contains theoutput neurons or as in this case, a single output neuron. The middle layer is called ahidden layer, since, the neurons in this layer are neither inputs nor outputs (Maind and Wankar, 2014; Metzger *et al*., 2015).

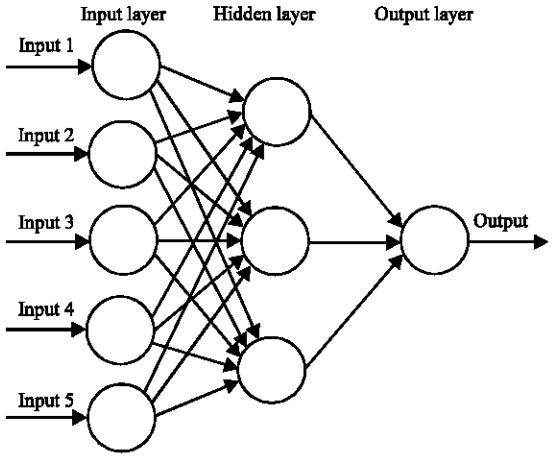

Fig. 2: Artificial neural network

McCulloch-Pitts networks: An early model of an artificial neuron was introduced in 1943 by Warren McCulloch and Walter Pitts. This model became known as a linear binary threshold gate. With a series of given inputs, a weighted sum of all the input and the weight associated with it would be calculated. The weight values which are normalized in the range of either (0, 1) or (-1, 1). Given a certain threshold, the output would be one of two classes in binary based on whether the sum exceeded the threshold or not. The threshold is fixed at 0:

$$\text{Output} = \begin{cases} 0 \text{ if } \sum_j w_j x_j \leq \text{threshold} \\ 0 \text{ if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

Perceptron: In the 1950's, Rosenblatt created a two-layer network, the perceptron. In the McCulloch-Pitts network, the threshold is fixed at 0 whereas in the perceptron implementation, a variable threshold value is used (Fig. 3).

The perceptron was not able to solve the classic XOR (exclusive or) problem led to the decline of the field of neural networks. However, the perceptron had laid foundations for later work in neural computing.

The delta rule: In 1960's, the delta rule was invented. It is also known as the Widrow and Hoff learning rule or the Least Mean Square (LMS) rule. It uses gradient descent

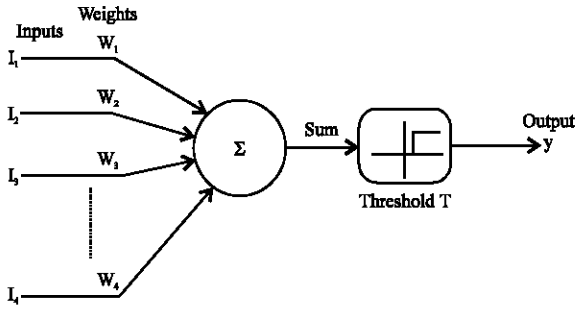

Fig. 3: Artificial Neuron Model

learning to iteratively change network weights to minimize error (difference between target output values and the actual obtained activation). The threshold is replaced by a linear activation function which is used to calculate the activation of the output neuron (Fig. 4).

In linear activation function, the output node's activation is equal to the sum of the product of the network's input and respective weight. Alternative activation functions with respective equations are shown in Fig. 4. They are also used depending on the system demand (Song and Cai, 2015).

Feedforward neural network: It is the simplest type of ANN in which the information only travels forward in the network (no loops), first through the input layer, then through thehidden layer and finally through the output layer. The weighted sum is calculated by the equation:

$$S_j = \sum_j w_{ij} a_i$$

$$a_j = f(S_i)$$

Where:

S_j = The Summation of product of relevant weights and outputs of previous layer i
W_{ij} = The relevant Weights connecting layer i with layer j
a_i = The output of previous nodes
a_j = The output of the node at hand
F = The activation function

For any given set of input data and weights, there will be an associated magnitude of error which is measured by an error function also known as a cost function:

$$E_p = \frac{1}{2}\sum_n (t_{jn} - a_{jn})^2$$

Where:

E_p = The total error over a training pattern
T_{jn} = The Target value for node n in output layer j
a_{jn} = The actual output for the same node
n = All output nodes

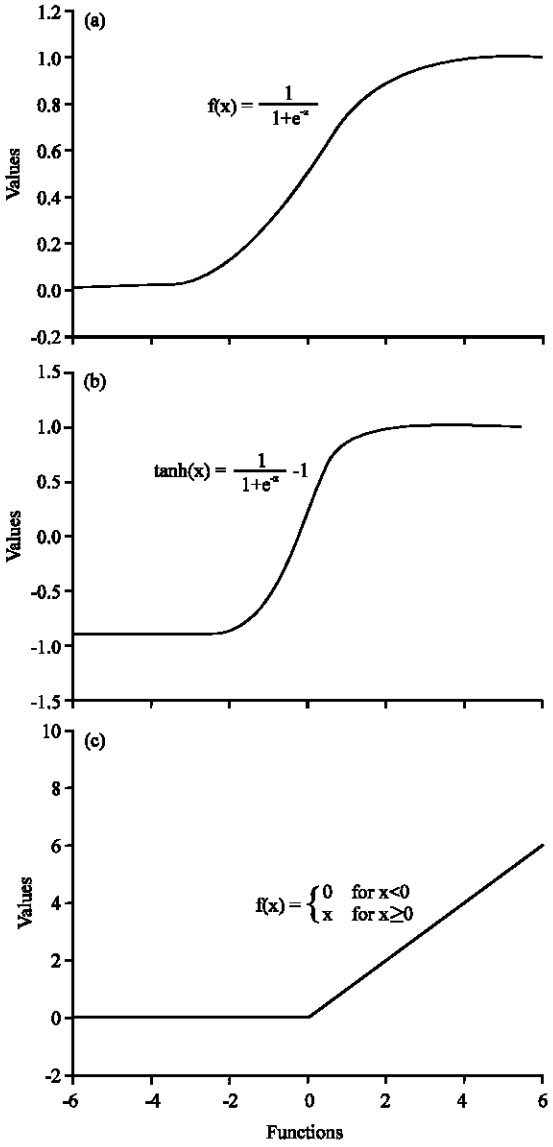

Fig. 4: Different activation functions: a) Sigmoid, b) TanH and c) ReLU

A normalized version of error over an entire set of training patterns (i.e., over one iteration or epoch) is given by the Mean Squared Error (MSE) equation:

$$MSE = \frac{1}{2PN}\sum_p \sum_n (t_{jn} - a_{jn})^2$$

Where:

P and N = Total number of training patterns and output nodes, respectively
p = All training patterns

Gradient descent learning: Gradient descent learning requires the negative of the derivative of the error function. Using the chain rule we can get, the derivative of error equation for a given pattern p with respect to a particular weight as:

$$\frac{\delta E_p}{\delta W_{ijx}} = \frac{\delta E_p}{\delta a_{jz}} \frac{\delta a_{jz}}{\delta W_{ijx}}$$

where, a_{jz} is the output of the node in the output layer that corresponds to the weight. It follows that:

$$\frac{\delta E_p}{\delta a_{jz}} = 2.\frac{1}{2}\left(t_{jn} - a_{jn}\right)(-1) = -\left(t_{jn} - a_{jn}\right)$$

$$\frac{\delta a_{jz}}{\delta W_{ijx}} = \frac{\delta}{\delta W_{ijx}} \sum \left(W_{ijx}\, a_{i_n}\right) = a_{i_x}$$

Thus, the derivative of the error over an individual training pattern is given by:

$$\frac{\delta E_p}{\delta W_{ijx}} = -\left(t_{jn} - a_{jn}\right) a_{i_x}$$

Replacing the difference between the target and actual output of the relevant output and introducing a learning rate Epsilon:

$$\frac{\delta E_p}{\delta W_{ijx}} = -\varepsilon a_{i_x}$$

Re-writing the above equation in the final form of the delta rule:

$$\Delta W_{ijx} = -\varepsilon \frac{\delta E}{\delta W_{ij}} = \varepsilon \delta a_{i_x}$$

Back propagation algorithm: The supervised learning problem of the MLP can be solved with the back-propagation algorithm. The algorithm consists of two steps. In the forward pass, the predicted outputs corresponding to the given inputs are evaluated. In thebackward pass, partial derivatives of the cost function with respect to the different parameters are propagated back through the network.

The whole process is iterated until the weights have converged. The back propagation algorithm searches for weight values that minimize the total error of the network over the set of training examples (training set) (Fig. 5).

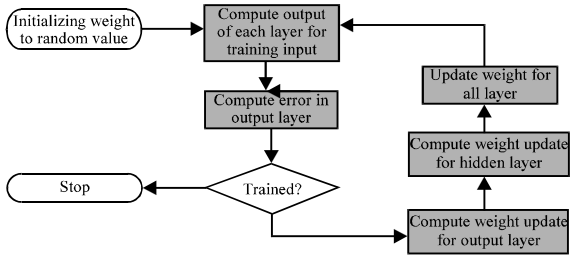

Fig. 5: Back propagation training algorithm

Hybrid model: In hybrid approach, different models for example GMM, HMM, NN are combined to get better results. Training hybrid systems are complex; they have multiple stages and are expertise intensive tasks which need lots of human efforts. By comparison training end-to-end systems are simpler than training hybrid ones, they require fewer stages but doesn't have overwhelming advantage in performance and training speed compared with the hybrid systems. However, end-to-end system is easy to deploy and avoids the inconsistency in hybrid system.

Zied presents a study of isolated word recognition using Hidden Markov Model with Gaussian Mixture (HMM-GM). They have used two parameterization techniques Mel Frequency Cepstral Coefficients (MFCC) and Perceptual Linear Prediction (PLP) and have also introduced the dynamic coefficients and the energy of the signal in order to achieve an improvement in the recognition rate.

Longfei Deep Neural Network Hidden Markov Models or DNN-HMMs are recently very promising acoustic models achieving good speech recognition results over Gaussian Mixture Model based HMMs (GMM-HMMs).

MATERIALS AND METHODS

The proposed system is based on end-to-end framework since, 2014; there has been much research interest in end-to-end ASR (Fig. 6).

Figure 6 illustrates end-to-end framework using artificial neural network. The digitized input sequence obtained from the audio input is converted to compact vector representation using feature extraction technique. The extracted features are directly fed to artificial neural network which acts as classifier and recognizes the spoken word and gives text output. Acoustic and Language models are not required.

The advent of end-to-end speech recognition system using simpler structure with neural networks, made it possible to reduce the need for specific information of the language and speech. Specifically by Graves and Jaitly

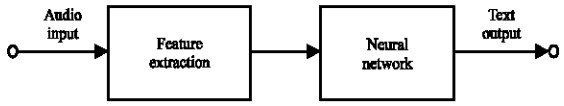

Fig. 6: Block diagram of end-to-end speech recognizer

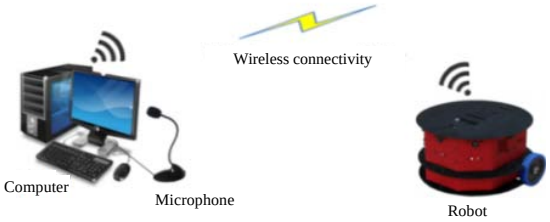

Fig. 7: Structure of the proposed system

(2014), it implements an end-to-end recurrent learning system that utilizes mel-filter bank features to directly output to spoken phonemes without the need of a traditional Hidden Markov Model for decoding.

Voice automated robot: The main motive behind the project was to study the application of Artificial Neural Network (ANN) as an important and basic building block and its application in the domain of speech recognition. Robotics was chosen as an extended application for the Speech recognition system. The Robot Firebird XI is controlled by the computer system which works as an operation platform for the speech recognition and also for the Firebird control and connectivity module. The computer system with sound card and microphone is used. Speech recognition engine is built in MATLAB.

The robot and the speech engine are connected by wireless connectivity. For wireless communication you should have wireless module inside the robot. Another wireless module will plug into USB port of computer from where you control the robot. Firebird VI robot has integrated 2.4 GHz wireless module with unique ID. Once, the connection is established, voice signals are processed and transferred between computer and Firebird robot for further action. The module for controlling the robot is written in Microsoft Visual Studio (Fig. 7).

The process flow: Database creation stands out as the first and very important stage towards building an application. Data set using different utterances for each of the five commands: Front, Back, Left, Right and Stop was built by manually recording the audio samples with .wav extension. The audio samples were recorded of a single speaker and the ambience was varied. Then Voice Activity Detection (VAD) was applied to eliminate the useless portions of each sample. The samples were then exposed to the front end processing for extracting

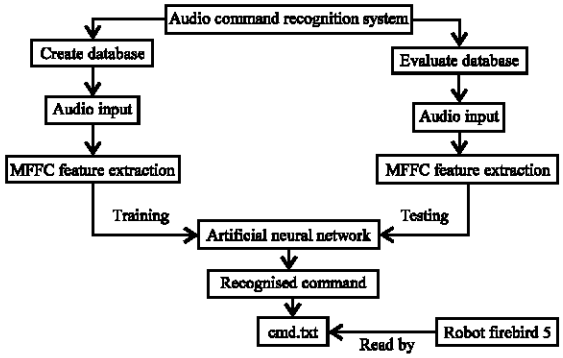

Fig. 8: Process flow diagram

useful features. Mel Frequency Cepstral Coefficients (MFCCs) was used as the feature extraction technique (Fig. 8).

The general practice is to first divide the data into three subsets. The first subset is the training set which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The network weights and biases are saved at the minimum of the validation set error used for improving generalization. The third set is the test set error which is not used during training but it is used to compare different models and to plot the test set error during the training process. Train ratio = 70/100; Validation ratio = 15; Test ratio = 15/100.

In the training mode, the neurons are trained to fire (or not), for particular input patterns. In the testing mode when a taught input pattern is detected at the input its associated output becomes the current output. If the input pattern does not belong in the taught list of input patterns, the firing rule is used to determine whether to fire or not.

The audio command recognition system works as the speech-to-text converter. The recognized command by the neural network is written in a text file cmd.txt which is read by the Firebird robot control module programmed in Visual Studio. After successfully establishing communication with the robot it performs control movements accordingly as per the program.

Ways to enhance system performance

Large dataset: The dataset should have more data spanning range of possible values. Presence of more data results inbetter and accurate models.

Removal of unwanted and outlier values: The presence of unwanted and outlier valuesin the training data often reduces the accuracyof a model or leads toa biased model resulting in inaccuratepredictions.

Compression techniques: The audio compression algorithms affect the performance of a speech recognition system. Richard suggests that if small amounts of white Gaussian noise are added before the speech is compressed, recognition rates can be increased with certain compression algorithms.

Feature extraction: Feature extractionprocess helps infinding out the best subset of attributes which better explains the relationship of independent variables with target variable.

Multiple algorithms: Some algorithms are better suited to a particular type ofdata sets than others. Hence, different relevant models should be applied and checked for the performance.

Parameter tuning: The objective of parametertuning is to find the optimum value for each parameter to improve the accuracy of the model.

Post training analysis (regression): The performance of a trained network can be measured to some extent by the errors on the training, validation and test sets. It is often useful to check the network response.

RESULTS AND DISCUSSION

The trained neural network generated by MATLAB is shown in Fig. 9 along with the different algorithms used and the progress of the network.

Neural network design parameters:

- Input layer-91 Neurons
- Hidden layer-10 Neurons
- Output layer-5 Neurons corresponding to each class
- Training algorithm-Levenberg-Marquardt

Validation performance plot: Best validation performance for mean squared error = 0.022878 for Epoch = 3. Figure 10 shows post-training analysis of the designed neural network. It can be seen from the figure that best validation performance was obtained at Epoch = 3.

Confusion matrix: Figure 11 illustrates the confusion matrix generated by taking 15 samples from the testing dataset. Three samples were taken for each of the five classes.

Rows indicate output and the columns indicate the target. The blue colored box indicates the accuracy. The samples falling on the green colored diagonals are correctly classified samples and the samples lying outside the diagonal are misclassified samples. The speech engine was made flexible enough by providing two options for audio input: audio file from testing dataset; input signal

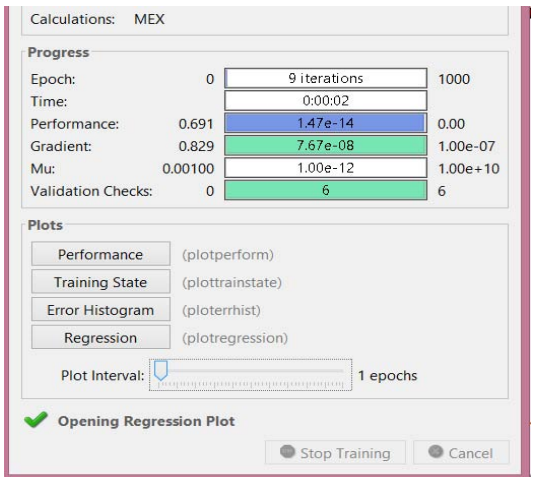

Fig. 9: Trained neural network

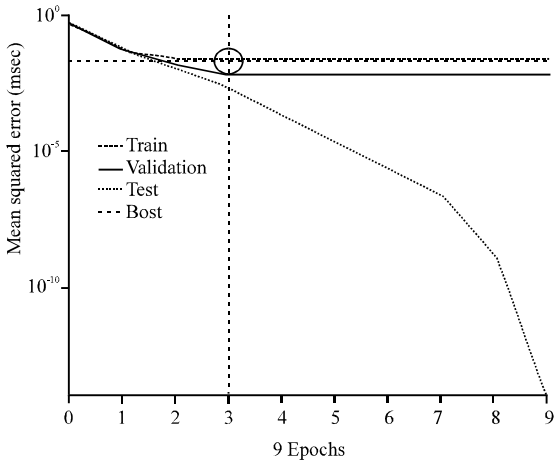

Fig. 10: Graph plotted EPOCH vs. MSE; best validation performance is 0.022878 at Epoch 3

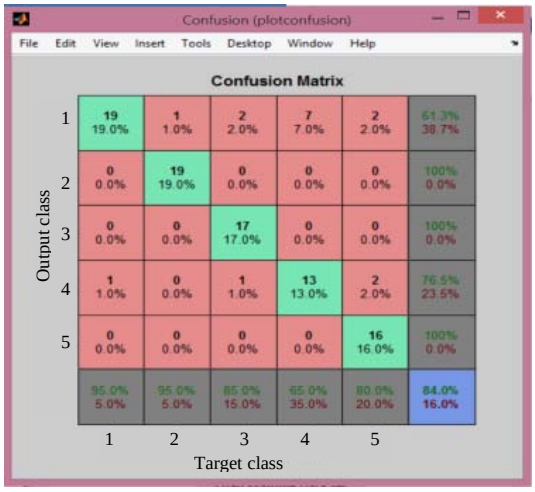

Fig. 11: System generated confusion matrix

from microphone. The engine performed with 100% accuracy when source was taken from testing data set. When audio input was taken from microphone the accuracy was variable depending on the environmental factors.

CONCLUSION

We have attempted to study the application of artificial neural network for isolated command recognition for robot control. Speech recognition speech recognition comes as a challenging problem to deal with and has created a technological impact on society and is expected to flourish further in area of human machine interaction. The engine can be made more robust, generalized by bringing into concepts related to neural network like:

- Deep learning
- Convolutional neural network
- Recurrent neural network

By Graves and Jaitly (2014), the system is based on a combination of the deep bidirectional LSTM recurrent neural network architecture and the connectionist temporal classification objective function.

REFERENCES

Baker, J.M., L. Deng, J. Glass, S. Khudanpur and C.H. Lee *et al*., 2009. Developments and directions in speech recognition and understanding, part 1 (DSP education). IEEE. Signal Process. Mag., 26: 75-80.

Graves, A. and N. Jaitly, 2014. Towards end-to-end speech recognition with recurrent neural networks. Proceedings of the 31st International Conference on Machine Learning (ICML-14), June 21-26, 2014, APSYS SA, Beijing, China, pp: 1764-1772.

Maind, S.B. and P. Wankar, 2014. Research paper on basic of artificial neural network. Intl. J. Recent Innov. Trends Comput. Commun., 2: 96-100.

Metzger, R.A., J.F. Doherty and D.M. Jenkins, 2015. Analysis of compressed speech signals in an automatic speaker recognition system. Proceedings of the 49th Annual Conference on Information Sciences and Systems (CISS), March 18-20, 2015, IEEE, Baltimore, Maryland, ISBN:978-1-4799-8429-9, pp: 1-5.

Song, W. and J. Cai, 2015. End-to-end deep neural network for automatic speech recognition. Allen Institute, Seattle, Washington. https://www.semanticscholar.org/paper/End-to-End-Deep-Neural-Network-for-Automatic-Speec-Song-Cai/528da8ef5cac3b69348b84a50ac2d596ddbee394.

Zhou, R., K.P. Ng and Y.S. Ng, 1994. A voice controlled robot using neural network. Proceedings of the 2nd Australian and New Zealand Conference on Intelligent Information Systems, November 29-December 2, 1994, IEEE, Brisbane, Queensland, ISBN:0-7803-2404-8, pp: 130-134.