AN ANALYSIS OF SPEECH RECOGNITION PERFORMANCE BASED UPON NETWORK LAYERS AND TRANSFER FUNCTIONS

Kuldeep Kumar[1], R. K. Aggarwal[1] and Ankita Jain[2]

[1]Department of Computer Engineering,
National Institute of Technology, Kurukshetra, India.
kuldeepgargkkr@gmail.com,rka15969@gmail.com
[2]Department of Electronics and Communication Engineering,
National Institute of Technology, Kurukshetra, India.
ankitajain.08@gmail.com

ABSTRACT

Speech is the most natural way of information exchange. It provides an efficient means of means of man-machine communication using speech interfacing. Speech interfacing involves speech synthesis and speech recognition. Speech recognition allows a computer to identify the words that a person speaks to a microphone or telephone. The two main components, normally used in speech recognition, are signal processing component at front-end and pattern matching component at back-end. In this paper, a setup that uses Mel frequency cepstral coefficients at front-end and artificial neural networks at back-end has been developed to perform the experiments for analyzing the speech recognition performance. Various experiments have been performed by varying the number of layers and type of network transfer function, which helps in deciding the network architecture to be used for acoustic modelling at back end.

KEYWORDS

Speech recognition, Mel frequency cepstral coefficients, Artificial neural networks, Network layer, Transfer function.

1. INTRODUCTION

Speech recognition is the way of converting the spoken word into the text. Speech recognition system takes an utterance of speech signal as input and converts it into a text sequence similar to information being conveyed by the input data. It has been observed that the success of speech recognition systems requires a combination of various techniques and algorithms, each of which performs a specific task for achieving the main goal of the system. The two main components used in speech recognition are signal processing component at front-end and recognition component at back-end. Signal processing transforms the input speech signal into a form that can be processed by recognizer. To achieve this, firstly the speech input is digitized and then processed through the first-order filters to spectrally flatten the signal. Thereafter, essential features of the speech signal having acoustic correlation with the speech input are extracted. These features can be extracted using various techniques such as linear predictive cepstral coefficient (LPCC) [1], Mel frequency cepstral coefficients (MFCC) [2], perceptual linear prediction (PLP) [3], wavelet [4] and RASTA (relative spectral transform) processing [5] etc. The recognition component is responsible for finding the best match in the knowledge base for the extracted essential feature vectors. The recognizer can be designed using various approaches such as hidden Markov model (HMM) [6], artificial neural networks (ANN) [7], dynamic Bayesian networks (DBN) [8], support vector machine (SVM) [9], hybrid methods (i.e. combination of two or more approaches) and others.

In the paper, firstly, a setup has been built to perform the experiments for analyzing the speech recognition performance. The setup uses MFCC for feature extraction and ANN for acoustic modelling. Using this setup, various experiments have been performed by varying the type of network transfer functions and the number of layers. From the performed experiments, various observations have been taken and conclusions have been drawn.

Apart from the introduction in section 1, the paper has been organized as follows. Section 2 describes the system setup used for our experiments. Section 3 deals with the experiments performed by varying different parameters. In section 4, the observations taken from these experiments have been presented. Finally, conclusion is drawn in section 5.

2. SYSTEM SETUP

This section presents the setup of the system developed for performing the experiments. To build the system setup, firstly, data has been collected. Once the data has been collected, essential features of the speech signal were extracted using signal-processing component as described below.

2.1 Data collection

In this section, the process of data collection used in the developed setup has been described. Various speech-sounds spoken by different speakers were recorded for the words to be recognized by the system. Four main factors considered while collecting the speech-sounds were [10]:

1. Talkers,
2. Speaking conditions,
3. Transducer and transmission systems and
4. Speech units.

First factor concerns with the profile of the talker. The profile of the talker specifies the features such as age, accent, gender, speaking-rate, region, tongue-language etc. A robust speech recognizer should be trained using speakers of various age, sex and regions etc. The setup uses the speech-sounds spoken by various speakers having different features like age, sex and others.

Second factor deals with the recording conditions such as room environment, noisy environment, lab room etc. Room environment was used for the recording. The reason behind it is to represent a real world speech samples collection, because most speech recognition systems are meant to be used in general room environment. Sounds were recorded at sampling rate of 16000 Hz.

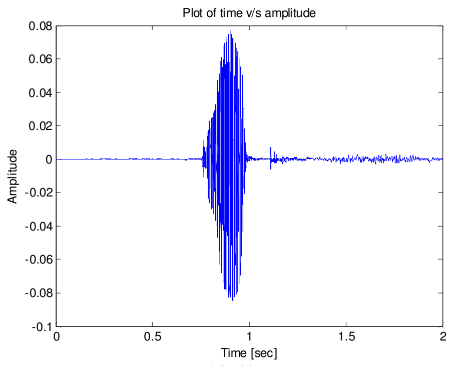
Nature of the transducer and transmission system greatly affects the recognition process. Transducer can be an omni-directional microphone or a unidirectional microphone. For our system setup, the data was recorded using unidirectional microphone. Distance of approximately 5-10 cm was kept between the mouth of the speaker and microphone. Fourth factor, speech units includes specific recognition vocabulary. The speech samples recorded at this stage are then used for signal processing.

2.2 Signal processing component

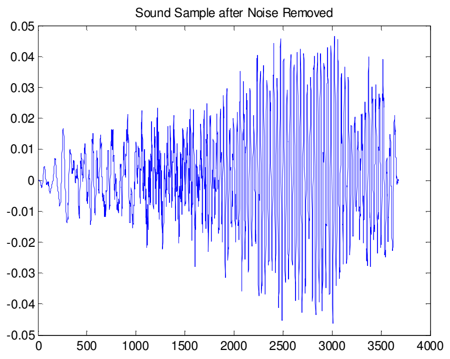
This section describes the process used by the setup for signal processing. Since speech signal is an analog waveform, it cannot be directly processed by digital systems. It has to be represented in a more compact and efficient way. Also, each speech waveform has some characteristics that distinguish it from other speech waveforms. Signal processing is used to achieve this. In signal processing, speech signal is converted into discrete sequence of feature vectors having the relevant information about given utterance that helps in its correct recognition [11]. Signal processing can be divided into two basic steps: Preprocessing and Feature Extraction (cepstrum coefficients generation and normalization). Preprocessing is to pre-process the speech samples to make available them for feature extraction and recognition purpose.

Preprocessing mainly covers A/D conversion, background noise filtering, pre-emphasis, blocking and windowing. During preprocessing, firstly the speech input was digitized. Figure 1.1 displays the digitized waveform for the word 'ek'. Then, fast front-end noise compensation technique, spectral subtraction has been used to remove noise from the speech input. Spectral-subtraction algorithms estimate the power spectrum of additive noise in the absence of speech and then subtract this spectral estimate from the power spectrum of the overall input (which normally includes the sum of speech and noise) [12]. Figure 1.2 shows the speech signal after noise removal. This noise removed speech-signal was then processed through the first-order filters to spectrally flatten the signal (figure 1.3). This process, known as pre-emphasis, increases the magnitude of higher frequencies with respect to the magnitude of lower frequencies. A pre-emphasis coefficient of value 0.97 was used by the designed system setup. In the next step, speech-signal was segmented into small frames having frame shift of 10 milliseconds and an overlap of 50%–70% between consecutive frames. The data in the analysis interval was then multiplied with a hamming window (figure 1.4) of size 25 milliseconds. 512-points fast Fourier transformations (FFTs) were calculated for the windowed data. Finally, feature extraction using MFCC was carried out to extract discriminant and uncorrelated information. Feature extraction computes the feature vectors (figure 1.5) of the speech signal on a frame by frame basis as:

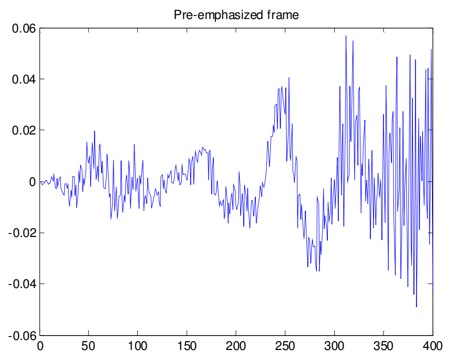
1. Find the fast Fourier transformation (FFT) for frames.
2. Calculate the power spectra.
3. The result of previous operation is being filtered with each k filter (the developed system setup has used the 24 Mel filters) from Mel filter bank and result is aggregated.
4. Take the log of the powers at each of the Mel frequencies.
5. Find the discrete cosine transformation (DCT) of the list of Mel log powers.
6. The MFCCs are the amplitude of resulting spectrum.
7. Append normalized frame energy, producing a 13-dimensional standard feature vector.
8. Compute the first and second order time derivatives of the 13 coefficients using regression formula.



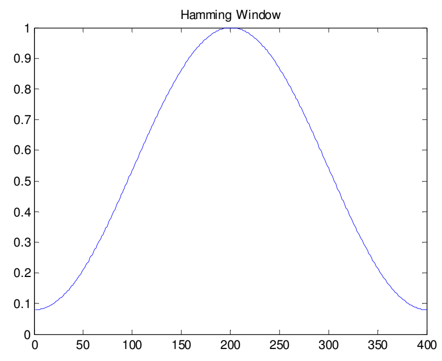
(1.1)



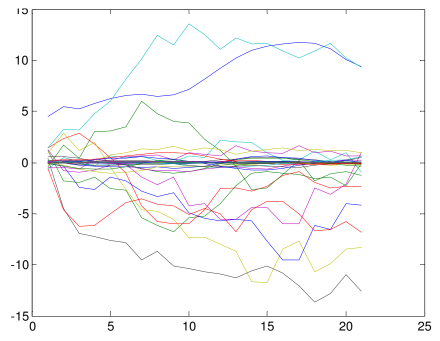
(1.2)



(1.3)



(1.4)



(1.5)

Figure 1. (1) Plot of the time graph for the recorded word 'ek', (2) speech signal after noise removal using spectral-subtraction, (3) pre-emphasized waveform, (4) hamming window, and (5) Mel frequency cepstrum coefficients (MFCC).

2.3 Deciding the neural network architectures for experimental setup

This section deals with the type of the neural network and its architecture to be used for our experiments. The neural network architecture helps in the designing of a robust recognizer.

A neural network is a parallel distributed architecture with large number of nodes called neurons and connections. Each connection points from one node to another and is associated with a weight. Each neuron takes input from the neurons of the previous layer (or from the outside world, if it is in the first layer). Then, it adds up this input and passes it to the next layer. Every time the neural network processes some input, it adjusts its weights to make the observed output closer to the desired output. After several iterations (each iteration is called an epoch), network can produce the correct output.

Here the feed-forward neural networks are used to perform the experiments. The output of the feature-extractor block i.e. MFCC coefficients acts as the input to the neural network. Then, network is designed with single layer as well as multi layers. In designing the network architecture, there is a requirement of transfer function which is applied to each neuron to obtain the output. Three types of transfer functions used for our experiments are: linear transfer function, tangent-sigmoid and log-sigmoid transfer function. In case of feed forward network, if the last layer of a multilayer network has sigmoid neurons, then the outputs of the network are limited to a small range. If linear output neurons are used the network outputs can take on any value.

The first step in training network architecture is to create a network object. Before training a network, the weights and biases must be initialized. When network object is created, it will automatically initialize the weights, but it may want to reinitialize them. After creating the network object, the next step is to simulate the network with the input given to it. Once simulated, the network is ready for training. During training, the weights and biases of the network are iteratively adjusted to minimize the network performance function. The performance function and training algorithm used by feed-forward networks during system setup is mean square error (MSE) and Levenberg-Marquardt back propagation algorithm respectively.

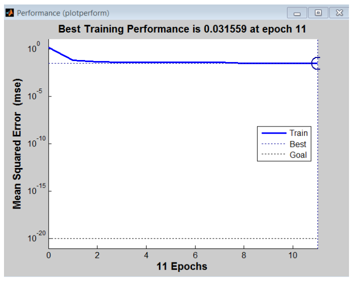
In the presented work, the experiments were performed by taking a particular set of MFCC coefficients and varying the network transfer functions and number of layers. The outputs from the performed experiments and the corresponding observations taken are discussed in the forthcoming sections.

3. EXPERIMENTAL RESULTS

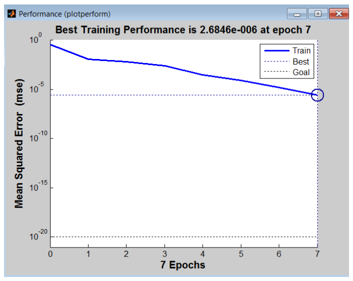
In this section, various experiments have been performed for analyzing the speech recognition performance. For all the experiments performed, the same MFCC coefficients were taken. The experiments also use the same value for the goal of acceptable error in recognition process. Various experiments vary on the use of number of layers of network and further on the type of network transfer functions. The observations taken from these experiments will be discussed in the next section.

3.1 Experiments with single layer networks

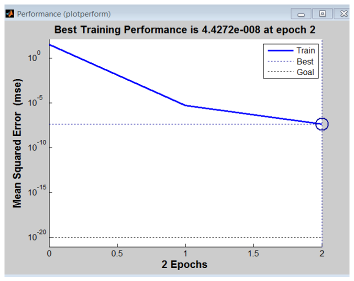
In a single layer network, the transfer functions i.e. linear (purelin), log-sigmoid (logsig) and tangent-sigmoid (tansig), were applied to the network individually for training. The result of training for the network that uses 'tansig' transfer function is shown in the figure 2.1. It can be seen that the performance goal was not met, but the minimum gradient has been achieved in 11 epochs. The training-results for the network that uses 'logsig' and 'purelin' transfer function are shown in the figure 2.2 and figure 2.3 respectively. It can be seen that the performance goal was not met, but the minimum gradient has been achieved in 7 epochs and 2 epochs respectively.



(2.1)



(2.2)



(2.3)

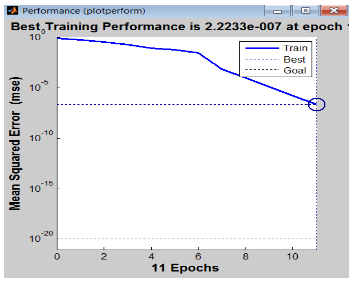
Figure 2. Performance plot with single layer network, (1) for tangent-sigmoid transfer function, (2) for log-sigmoid transfer function, and (3) for linear transfer function.

3.2 Experiments with multi-layer networks

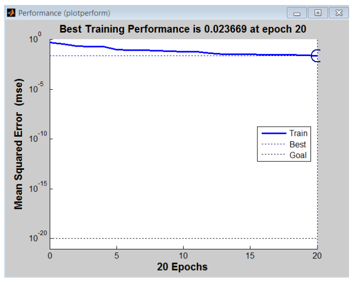
In multi-layer network, two layer and three layer networks were designed by using either same transfer functions or combinations of different transfer functions.

3.2.1 Two layer feed forward networks

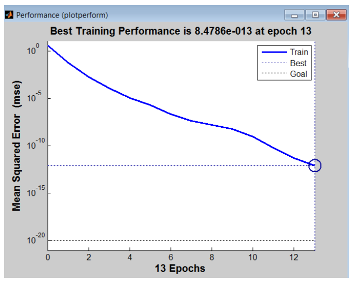
In two layer neural network architecture, two transfer functions are used: one at the hidden-layer, another at the output-layer. In this case, different combinations of transfer functions i.e. tangent-sigmoid, log-sigmoid and linear function were taken. Figure 3 shows the performance plot for different two layer neural networks using any two of these transfer functions. From figures 3.1-3.8, it can be seen that the performance goal was not met, but the minimum gradient has been achieved. In case of figure 3.9, where both hidden-layer as well as output-layer has ‘linear’ transfer function. It can be seen that the performance goal and the minimum gradient has been achieved in 5 epochs.



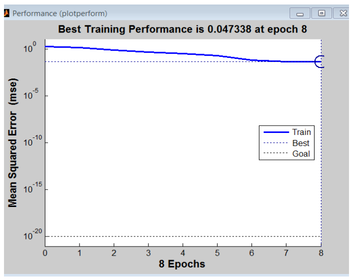
(3.1)



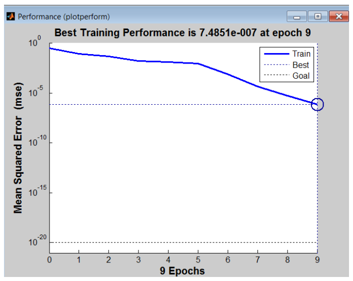
(3.2)



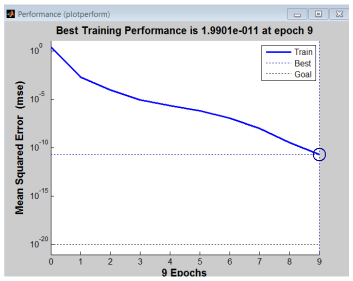
(3.3)



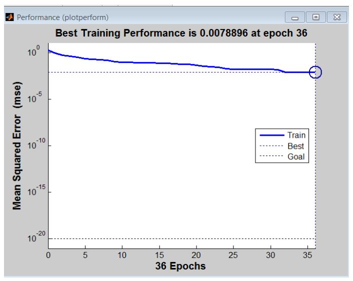
(3.4)



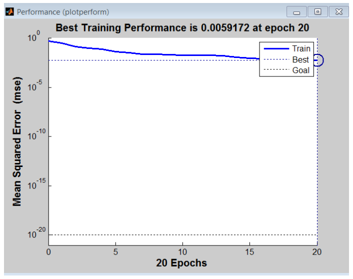
(3.5)



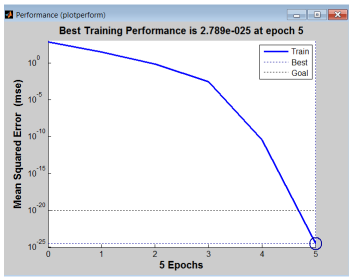
(3.6)



(3.7)



(3.8)

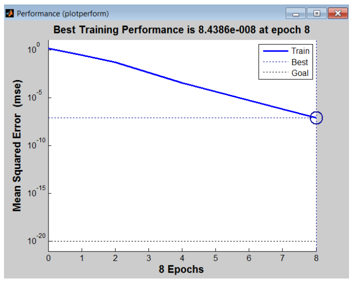


(3.9)

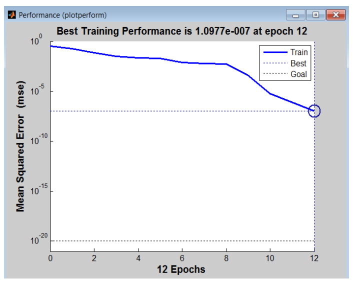
Figure 3. Performance plot with two layer networks, (1) for ‘tangent-sigmoid tangent-sigmoid’ transfer function, (2) for ‘tangent-sigmoid log-sigmoid’ transfer function, (3) for ‘tangent-sigmoid linear’ transfer function, (4) for ‘log-sigmoid tangent-sigmoid’ transfer function, (5) for ‘log-sigmoid log-sigmoid’ transfer function, (6) for ‘log-sigmoid linear’ transfer function, (7) for ‘linear tangent-sigmoid’ transfer function, (8) for ‘linear log-sigmoid’ transfer function, and (9) for ‘linear linear’ transfer function.

3.2.2 Three layer feed forward networks

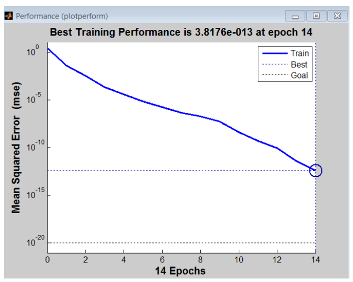
Three layer feed forward neural network has two hidden-layers that uses three transfer functions, two at the hidden-layers and one at the output layer. The output of the first hidden-layer is passed to the second hidden-layer, whose output acts as the input for the output-layer. In this case, different combinations of transfer functions were taken. Figure 4 shows the performance plot for different three layer neural network architectures that uses different combinations of the transfer functions. From figures 4.1-4.26, it can be seen that the performance goal was not met, but the minimum gradient has been achieved. In case of figure 4.27, where all layers have ‘linear’ as transfer function. It can be seen that the performance goal was met and the minimum gradient has been achieved in 7 epochs.



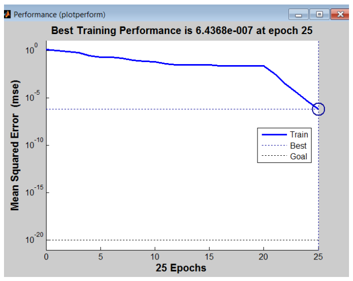
(4.1) tansig tansig tansig



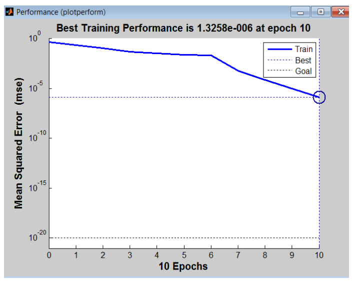
(4.2) tansig tansig logsig



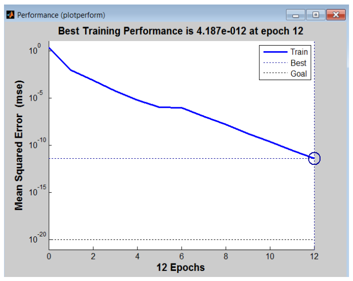
(4.3) tansig tansig linear



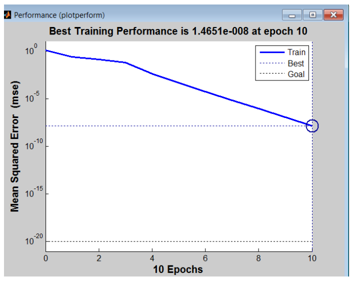
(4.4) tansig logsig tansig



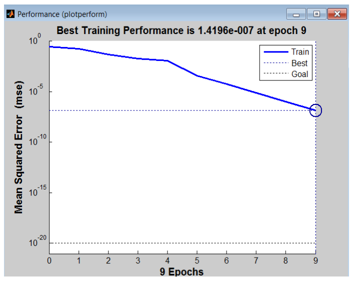
(4.5) tansig logsig logsig



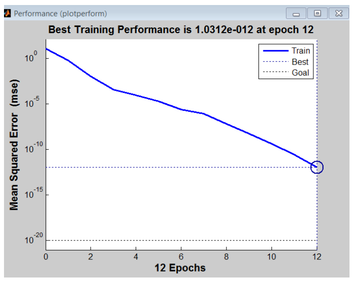
(4.6) tansig logsig linear



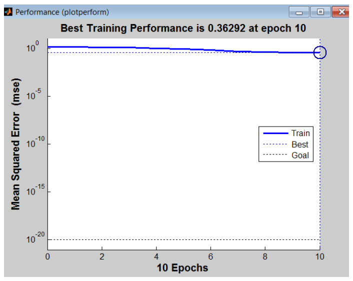
(4.7) tansig linear tansig



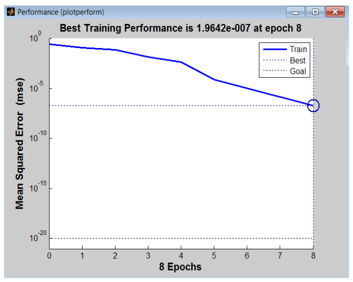
(4.8) tansig linear logsig



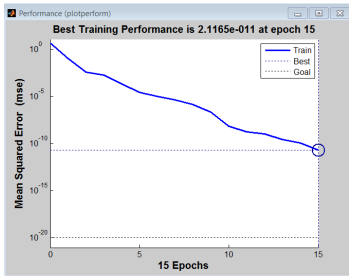
(4.9) tansig linear linear



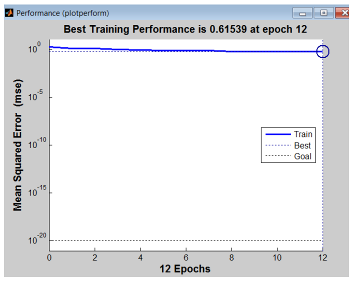
(4.10) logsig tansig tansig



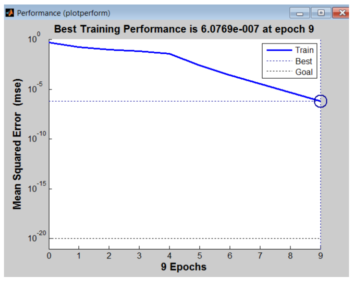
(4.11) logsig tansig logsig



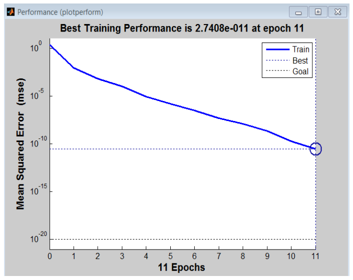
(4.12) logsig tansig linear



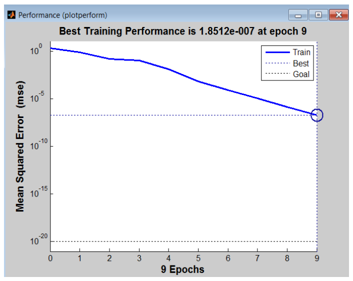
(4.13) logsig logsig tansig



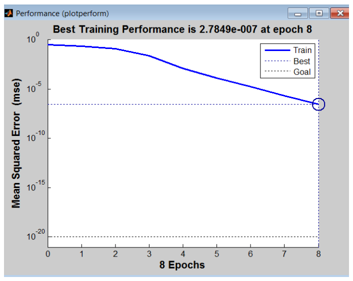
(4.14) logsig logsig logsig



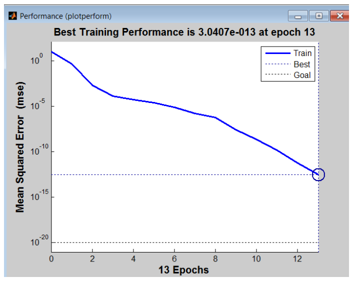
(4.15) logsig logsig linear



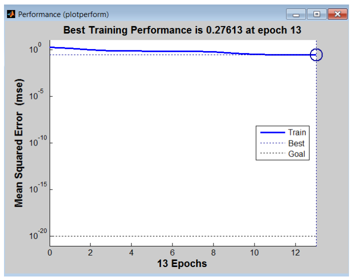
(4.16) logsig linear tansig



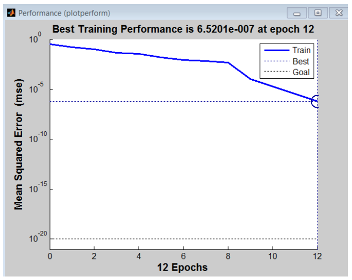
(4.17) logsig linear logsig



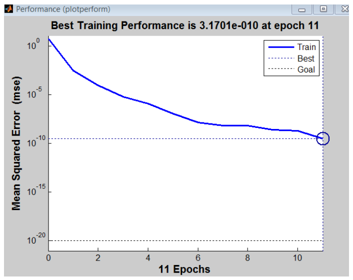
(4.18) logsig linear linear



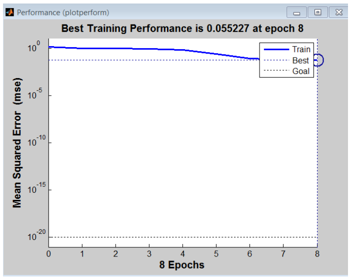
(4.19) linear tansig tansig



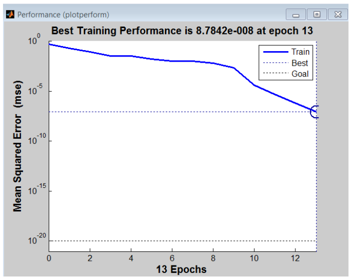
(4.20) linear tansig logsig



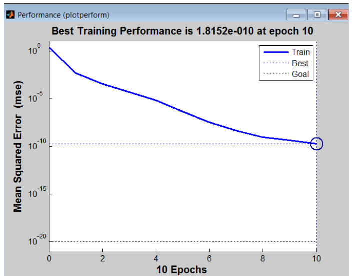
(4.21) linear tansig linear



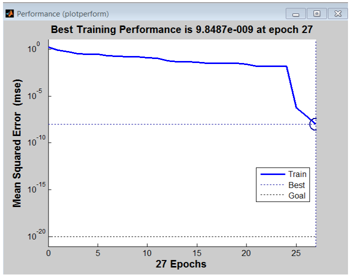
(4.22) linear logsig tansig



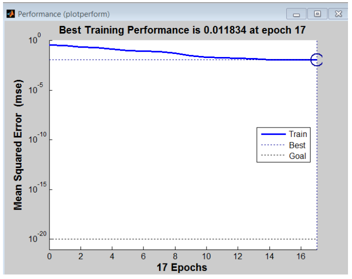
(4.23) linear logsig logsig



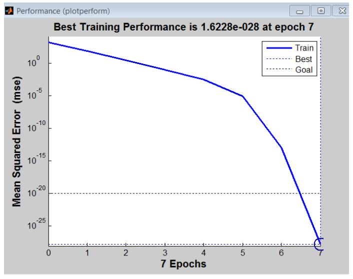
(4.24) linear logsig linear



(4.25) linear linear tansig



(4.26) linear linear logsig



(4.27) linear linear linear

Figure 4. Performance plot with three layer network architecture for different combinations of transfer functions

4. Observations

In all of the performed experiments as explained in the last section, the goal of the acceptable error was taken as a very-very small value i.e. 1e-020. It can be observed that, in all the experiments, the minimum gradient has been achieved, but the performance goal was not achieved in all of them. However, the number of epochs in which the minimum gradient is achieved also varies in different experiments.

It can be observed from figures 2.1, 2.2 and 2.3 that in the case of linear transfer function, the minimum gradient is achieved in less number of epochs than the case of sigmoid transfer functions. Also, in the case of linear transfer function, best training performance is much better than the others two.

From the figures 2.1, 3.1 and 4.1, in the case when all the network layers are using 'tangent-sigmoid' network transfer functions, with increase in the number of layers, a good improvement in the performance of the recognition is observed. Also, the number of epochs in which the minimum gradient reaches has been decreased.

It is observed from the figures 3.1 to 3.9 and 4.1 to 4.27 that best performance is achieved when the linear transfer function has been used for each layer as shown by figures 3.9 and 4.27 in just 5 and 7 epochs respectively which is the minimum epoch value in their respective set.

From the figures 2.2, 3.5 and 4.14, in the case when all the network layers are using 'log-sigmoid' network transfer functions, with increase in the number of layers, performance of the recognition system has been improved. However, with increase in the number of layers, the number of epochs in which the minimum gradient reaches is also increased, but with very-very small value.

From the figures 2.3, 3.9 and 4.27, in the case when all the network layers are using 'linear' network transfer functions, with increase in the number of layers, performance shows sharp improvements. Figures 3.9 and 4.27 shows that performance goal was met on the cost of small increase in the epoch count.

It is observed from the experiments performed in the last section that minimum numbers of epochs are required for achieving the minimum gradient in case of single layer network using 'linear' network transfer function (figure 2.3) but performance goal has not achieved in this case. Similarly, in case of two-layer network that uses 'linear' network transfer functions for hidden layer and 'tangent-sigmoid' network transfer function for the output layer as shown in figure 3.7, maximum number of epochs is required for achieving the minimum gradient.

It is also observed from figure 4.10 that the three layer network that uses 'log-sigmoid ' network transfer functions for the first hidden-layer and the 'tan-sigmoid' network transfer function for the second hidden-layer and output-layer shows weakest performance. The best performance is achieved when all the layers use 'linear' as the transfer function (figure 4.27).

5. CONCLUSION

In this paper, after discussing the system setup, various experiments have been performed by varying the number of network layers and the type of network transfer functions. It is observed that in the case of linear transfer function, minimum gradient is achieved in less number of epochs than the case of sigmoid transfer functions. Experiments also show that with increase in the number of layers of same type either 'linear' or 'sigmoid', the performance of recognition get improved. These experiments help in deciding the architectures of the neural networks to be used for the acoustic modelling of speech signals.

REFERENCES

[1] Markel, J. D. & Gray, A. H. (1976) *Linear Prediction of Speech*, New York: Springer-Verlag.

[2] Davis, S. & Mermelstein, P. (1980) "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 28, No. 4, pp. 357-366.

[3] Hermansky, H. (1990) "Perceptually linear predictive (PLP) analysis of speech", *Journal of Acoustic Society of America*, Vol. 87, No.4, pp. 1738-1752.

[4] Sharma, A., Shrotriya, M. C., Farooq, O. & Abbasi, Z.A. (2008) "Hybrid wavelet based LPC features for Hindi speech recognition", *International Journal of Information and communication Technology, Inderscience publisher,* Vol. 1, No. 3/4, pp. 373-381.

[5] Hermansky, H. & Morgan, N. (1994) "RASTA processing of speech", *IEEE Transaction of Speech and Audio Processing*, Vol. 2, No. 4, pp. 578-589.

[6] Huang X. D, Ariki Y. & Jack M. A. (1990) *Hidden Markov Models for Speech Recognition*. Edinburg University Press.

[7] Gold, B. (1988) "A Neural Network for Isolated Word Recognition", *In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1988.

[8] Deng, Li. (2006) "Dynamic Speech Models: Theory, Applications, and Algorithms", *Synthesis Lectures on Speech and Audio Processing*, Vol. 2, No. 1, pp. 1-118.

[9] Guo, G. & Li, S.Z. (2003) "Content Based Audio Classification and Retrieval by Support Vector Machines", *IEEE Transactions on Neural Networks*. Vol. 14, No. 1, pp. 209-215.

[10] Rabiner, L., Juang, B.H. and Yegnarayana B. (2010) *Fundamentals of Speech Recognition*, Pearson Education, India, pp. 116.

[11] Picone, J. (1993) "Signal Modeling Techniques in Speech Recognition", *Proceedings of the IEEE*, Vol. 81, No. 9, pp. 1215-1247.

[12] Boll, S. F. (1979) "Suppression of Acoustic Noise in Speech using Spectral Subtraction". *IEEE Transaction on Acoustic, Speech and Signal Processing*, Vol. 27, No. 2, pp. 113-120.

Authors

Kuldeep Kumar is with Department of Computer Engineering, National Institute of Technology (NIT), Kurukshetra, Haryana, India. He has done his B.Tech in Computer Engineering (with honours) from University Institute of Engineering and Technology (UIET), Kurukshetra University, Kurukshetra, India in 2009. He has also worked as the summer trainee in the Department of Computer Engineering, Institute of Technology, Banaras Hindu University (IT-BHU), Varanasi, India-221005. His current areas of interest include speech recognition, semantic web, software engineering, automata theory, compiler design, and statistical models.

R.K. Aggarwal is an Associate Professor in the Department of Computer Engineering at National Institute of Technology, Kurukshetra, Haryana, India. He has published more than 20 papers in various conferences and journals. His research interests include automatic speech recognition for Indian languages, pattern classification, statistical modelling, spirituality and Indian culture.

Ankita Jain is an Assistant Professor in the Department of Electronics and Communication Engineering, National Institute of Technology, Kurukshetra, Haryana, India. She did her B.Tech in Electronics and Communication Engineering from Kurukshetra University, Kurukshetra. She has many papers in national/international journal and conferences. Her areas of interest include speech processing, digital systems.