An Efficient Noise-Robust Automatic Speech Recognition System using Artificial Neural Networks

Santosh Gupta, Kishor M. Bhurchandi, and Avinash G. Keskar

***Abstract*—The Human Auditory System for speech recognition is highly robust against background noise compared to state-of-the-art Automatic Speech Recognition (ASR) systems. One of the best ways to add robustness to a speech recognition system is to have a compressed and highly robust feature set. In this paper, we present a novel approach for feature compression which makes the proposed noise-robust ASR system simple and very efficient. The other popular approach for feature compression is K-means which is complex and time-consuming. The experiments were performed on the proposed noise-robust ASR system for recognizing 65 different words with very low Signal to Noise Ratio (SNR) of -5 dB. The Mel Frequency Cepstrum Coefficients (MFCCs) were used as features for speech recognition. Back propagation Artificial Neural Network (ANN) was used to design the ASR system. Specialized versions of feedforward neural networks and training algorithms were tested on the speech recognition system and results are presented for each type. Experimental results show that the recognition accuracy of the proposed noise-robust ASR system is very high even with such a low SNR.**

***Index Terms*—Feature Compression, Neural Networks, Noise Robustness, Speech Recognition**

I. Introduction

SPEECH is one of the most efficient and natural forms of human communication. Automatic Speech Recognition (ASR) is the technology in which a machine understands the meaning of the human speech. In today's modern world, speech-based services are gaining popularity and to provide better user experiences in real world applications, noise robustness of ASR systems is becoming more and more important. Generally, there are two approaches to improve noise robustness of ASR systems: Feature Enhancement method and Model Adaptation method [1]. In feature enhancement method, attempts are made to remove the background noise from the recordings prior to speech recognition [2]. In model adaptation method, the recordings are left unchanged and the model parameters of the speech recognizer are updated to be the best representative of the recorded speech [3]. The adaptive training techniques are used to improve the above two approaches. Artificial Neural Networks (ANN) are used for speech recognition in an optimized way [4]. The important benefits of neural networks are that they are robust against interference and are good at comparisons as well as associations [5]. Parallel Processing can also be done by neural networks. For noise robustness, posterior features are generated by training neural networks [6]. Stereo data can also be used to train networks to differentiate between clean and noisy features [7].

In this paper, we propose an efficient noise-robust ASR system in which the important steps include signal processing, feature extraction and compression, and the neural network design using MATLAB. Experiments were performed in two phases. In the first phase, noise-robust ASR system was tested for recognizing 30 different words with SNR of –5 dB in which the size of the input compressed feature vector was varied. The feature vector was computed by our proposed new approach for feature compression. The specialized versions of feedforward neural networks and training algorithms were also compared. The best performing neural network design and training algorithm was then selected and used for the later experiments. In the second phase, noise-robust ASR system was tested for recognizing 65 different words with SNR of –3 dB and –5 dB. Here the number of neurons in the hidden layer of neural network were varied.

The rest of the paper is organized as follows. In Section II, we discuss the signal processing steps involved and the novel approach for feature compression. Section III details the structure of Neural Network. Section IV describes the Experimental setup. Results are presented in Section V and we conclude the paper in Section VI.

Santosh Gupta was with the Electronics and Communication Engineering Department, Visvesvaraya National Institute of Technology, Nagpur 440010 India. He is now with the Centre for Development of Telematics, Bengaluru 560100 India (e-mail: santosh.gupta.ece@gmail.com).

Kishor M. Bhurchandi is with the Electronics and Communication Engineering Department, Visvesvaraya National Institute of Technology, Nagpur 440010 India (e-mail: bhurchandikm@ece.vnit.ac.in).

Avinash G. Keskar is with the Electronics and Communication Engineering Department, Visvesvaraya National Institute of Technology, Nagpur 440010 India (e-mail: agkeskar@ece.vnit.ac.in).

978-1-5090-0396-9/16/$31.00 ©2016 IEEE

II. Signal Processing

The noise-robust ASR system must give excellent accuracy for recognizing clean speech signals (i.e. signal with no noise)

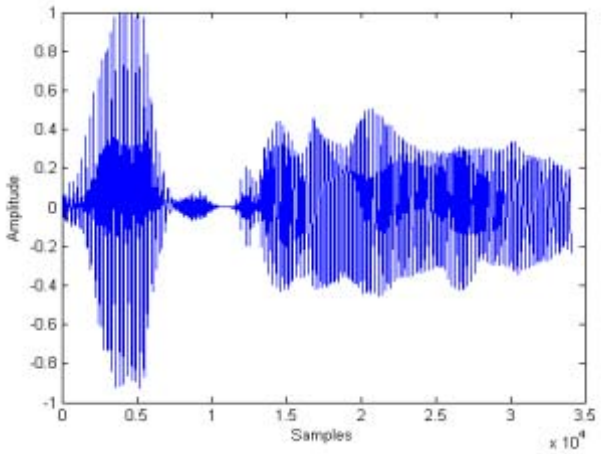

Fig. 1. Time domain representation of the speech signal "AFTERNOON" with no noise.

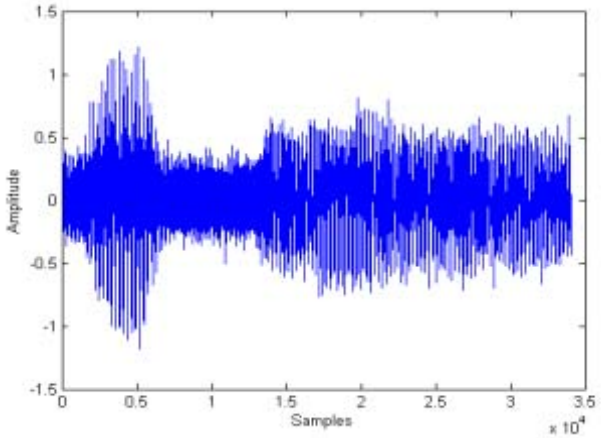

Fig. 2. Time domain representation of the speech signal "AFTERNOON" with SNR of 5 dB.

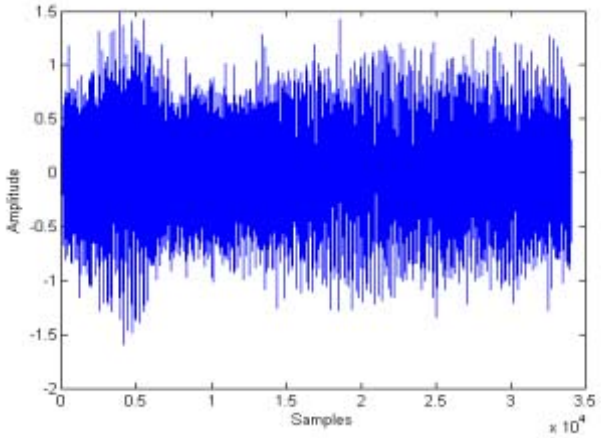

Fig. 3. Time domain representation of the speech signal "AFTERNOON" with SNR of –3 dB.

Thus, experiments were performed for recognizing speech signals with low SNR as well as absolutely clean speech signals. We utilized a model-based speech synthesis (text to speech) approach that enabled the efficient generation of clean speech signals.

A. Speech Synthesis

The Hidden Markov Model (HMM) - based Speech Synthesis System (HTS) [8] was used for synthesizing audio words (clean speech signals). All signals were created using a single US English female speaker model. In total, 65 different words with the sampling rate of 48000 samples per second were synthesized using HTS. These 65 words were very carefully selected so that even non-native English speakers can understand each word. Some of the examples can be considered as name of the countries, fruits, etc.

Using HTS, each word was synthesized in 20 different ways by changing parameters like pitch, voiced/unvoiced speech, and duration. So for 65 words, we had 1300 different speech signals for experimentation. Time domain representation [9] of the synthesized clean speech signal "AFTERNOON" is illustrated in Fig. 1.

After generation of all these 1300 synthesized speech signals, Gaussian noise was added to mimic the effect of background noise. From Fig. 2 and Fig. 3, it can be easily stated, that by decreasing the SNR value from 5 dB to -3 dB, it becomes increasingly difficult to even visually recognize the original waveform. Noise cannot be removed from the signal without any major information loss. These speech signals with such a low SNR value were input to the noise-robust ASR system.

B. Feature Extraction and Compression

The Mel Frequency Cepstrum Coefficients (MFCCs) were used as features for speech recognition. MFCCs represent audio based on human perception. Speech signals have several kind of randomness. It means that even the same words have different duration if spoken again and hence different number of frames. Since MFCC features are calculated for each frame, different sizes of feature vectors are computed. This leads to non-static features. The basic Artificial Neural Networks (ANN) cannot handle these non-static feature vectors. For having proper input to ANN, different sizes of feature vectors need to be compressed into the same size.

A novel approach for MFCC feature compression was implemented i.e. mean and variance of the feature vectors. The other popular approach for compressing MFCC feature vectors (K-means) was not used because of its complexity and time consumption. Computing variance and mean was simple, efficient and gave comparatively more compressed feature vector than K-means algorithm. Variance measures how far a set of numbers is spread out. Let say, the number of frames for input speech signal is 'n'. For each frame, 20 MFCC coefficients are computed. Thus, the size of input feature vector will now be (20 X n). Since the number of frames may be random even for same word sometime, MFCC feature vector is not a constant matrix. As neural network is used, size of input must remain constant. This can be achieved by considering the variance and mean of all the MFCC frames.

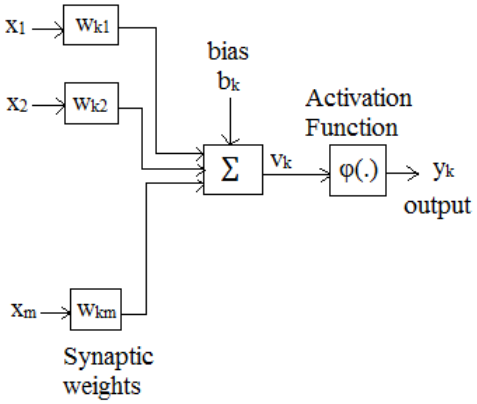

Fig. 4. Model of Neuron

III. Neural Network

A. Node Characteristics

An ANN is an interconnected group of nodes which is similar to the networks of neurons in our brain. Each node has more than one inputs from various connections that have synaptic weights.

From Fig. 4, it can be observed that the neuron model has bias b_k and m number of inputs (x_1, x_2, …., x_m) with corresponding synaptic weights. The output of the summation element is given by:

$$v(k) = \sum_{j=0}^{n} w_{kj}\, x_j + b_k. \qquad (1)$$

The output of the activation function applied becomes:

$$y(k) = \varphi(v(k)) \qquad (2)$$

B. Network Topology and Learning

In an artificial neural network, there is an input layer, zero or multiple hidden layers, and an output layer. The methods of connections among the nodes is determined by the network topology. It also determines the number of nodes in each of the layers.

The Neural network model is trained with supervised learning. In supervised learning, the desired or target output is given to the network corresponding to the example input. The synaptic weights are accordingly adjusted to minimize the error between the network output and the target output. Neural Network data are divided into three subsets for the operation of ANN: Training phase, Validation phase and testing phase.

IV. Experimental Setup

For our proposed noise-robust ASR system, the simulation experiment follows this particular sequence: Speech synthesis with added noise, signal processing, feature extraction and compression, neural network training and recognition. The sequences are illustrated in Fig. 5.

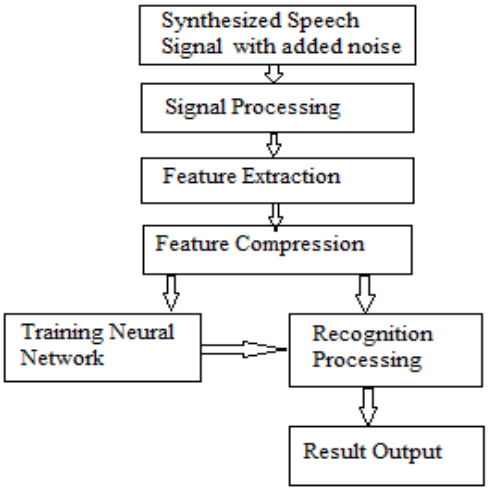

Fig. 5. Neural Network Simulation Experiment Sequence

A. Network Topology Used

The MATLAB Neural Network toolbox [10] was used to train and test the networks. The input to the neural network was the compressed feature vector obtained by taking variance and mean of the extracted MFCC features. We used only one hidden layer. The activation function used for the output layer and hidden layer was softmax and sigmoid respectively.

B. Feedforward Networks and Training Algorithms

Feedforward networks are generally used for any kind of input-output mapping. Specialized versions of the feedforward network include pattern recognition and fitting networks. Another variation on the feedforward network is the cascade forward network in which additional connections are present from the input to every layer.

The network training function updates the bias and weight values of the network depending upon the training algorithms used. Following are the four training algorithms compared for the noise-robust ASR system: Polak-Ribiére Conjugate Gradient [CGP], Resilient Backpropagation [RP], Conjugate Gradient with Powell/Beale Restarts [CGB], Scaled Conjugate Gradient backpropagation [SCG].

V. Results

Experiments were performed in two phases. In first phase, noise-robust ASR system was tested for recognizing 30 different words with SNR of –5 dB in which the size of the input compressed feature vector was varied. Results are presented in Table I and Table II for this experiments in which specialized versions of feedforward neural networks and training algorithms are also compared. The number of neurons in the hidden layer were 10 for this experiments.

From Table I, it is observed that the feedforward network, fitting network and cascade-forward network are giving satisfactory results which are approximately same.

TABLE I
ACCURACY FOR RECOGNIZING 30 DIFFERENT WORDS (SNR -5 DB) WITH INPUT FEATURE VECTOR OF LENGTH 20

Training Algorithms	Feedforward Network	Fitting Network	Pattern Recognition Network	Cascade-forward Network
RP	87.2 %	87.8%	94.3%	86.2%
SCG	86.2%	86.8%	**94.5%**	84.2%
CGB	87.7%	85.5%	77.5%	87.0%
CGP	87.2%	83.3%	93.0%	79.8%

TABLE II
ACCURACY FOR RECOGNIZING 30 DIFFERENT WORDS (SNR -5 DB) WITH INPUT FEATURE VECTOR OF LENGTH 40

Training Algorithms	Feedforward Network	Fitting Network	Pattern Recognition Network	Cascade-forward Network
RP	68.3 %	66.2%	98.2%	95.7%
SCG	55.5%	47.2%	**98.7%**	97%
CGB	54.8%	39.2%	93.3%	97.5%
CGP	42.2%	48%	97.7%	97%

It can be noted that there is not much effect of implementing different training algorithms on feedforward, fitting and cascade-forward network. On the other hand, the pattern recognition network gives much better recognition accuracy than the other three networks. The best result is obtained when SCG training algorithm is used. From Table II, it can be noted that the percent accuracy for Feedforward and Fitting network reduced drastically for all the training algorithms. Reason is that the length of the input feature vector increased but the number of hidden neurons remained same. Thus for improving accuracy for both feedforward and fitting network, number of hidden neurons must be increased. It is also observed that by increasing the number of input coefficients, the accuracy is improved for pattern recognition network and cascade-forward network. Pattern recognition network again gave the best recognition accuracy with SCG Algorithm.

In second phase, noise-robust ASR system was tested for recognizing 65 different words with SNR of –3 dB and –5 dB. Here the number of neurons in the hidden layer of neural network were varied. Pattern Recognition network and SCG algorithm were used as they gave the best recognition accuracy for phase one experiments. Results for this experiment are presented in Table III and Table IV. It can be observed from Table III that the percent accuracy for recognizing 65 different words with SNR of -3 dB increased rapidly when number of neurons in the hidden layer were increased from 5 to 10. Increasing the number of neurons in the hidden layer after certain extent, results in over-training or over-fitting of the neural network. The best recognition accuracy obtained is 99.6%. From Table IV, it is observed that the recognition accuracy obtained is slightly low as compared to that obtained when SNR was -3 dB. The best recognition accuracy obtained in this case is 99.4%.

TABLE III
ACCURACY FOR RECOGNIZING 65 DIFFERENT WORDS (SNR -3 DB) WITH INPUT FEATURE VECTOR OF LENGTH 40

Number of Neurons in Hidden Layer	Percent Accuracy for Recognizing words
5	78.8%
6	82.2%
7	85.5%
8	89.5%
9	96.3%
10	98%
12	98.2%
20	99%
30	99.4%
40	**99.6%**
50	99.6%
100	99.6%
150	99.3%

TABLE IV
ACCURACY FOR RECOGNIZING 65 DIFFERENT WORDS (SNR -5 DB) WITH INPUT FEATURE VECTOR OF LENGTH 40

Number of Neurons in Hidden Layer	Percent Accuracy for Recognizing words
5	59.9%
6	90.5%
7	81.5%
8	77.2%
9	92%
10	94.8%
12	98%
20	99.4%
30	98.8%
40	99.1%
50	98.7%
100	98.5%
150	98.4%

VI. CONCLUSION

The new approach for considering variance and mean as MFCC feature compression gave excellent results for speech recognition. The training algorithm also played very important role for the neural network performance. SCG training algorithm gave the best accuracy for noise-robust speech recognition with pattern recognition network but gave average recognition accuracy with other neural networks. The best recognition accuracy is achieved when number of neurons in the hidden layer is between input feature vector (total number of input coefficients) and the output vector (total number of words to be recognized). The recognition accuracy of our proposed noise-robust ASR system is more than 99% percent which is excellent for recognizing the speech signals with such a low SNR values.

REFERENCES

[1] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero, "Noise Adaptive Training for Robust Automatic Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*., vol. 18, no. 8, pp. 1889-1901, Nov. 2010.

[2] D. Yu, L. Deng, J. Droppo, J. Wu, Y. Gong, and A. Acero, "A minimum-mean-square-error noise reduction algorithm on Mel-frequency cepstra for robust speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, NV, 2008, pp. 4041-4044.

[3] M. L. Seltzer, K. Kalgaonkar, and A. Acero, "Acoustic model adaptation via Linear Spline Interpolation for robust speech recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Dallas, TX, 2010, pp. 4550-4553.

[4] F. Richardson, D. A. Reynolds and N. Dehak, "Deep Neural Network Approaches to Speaker and Language Recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671-1675, Oct. 2015.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice Hall, 1999.

[6] O. Vinyals and S.V. Ravuri, "Comparing multilayer perceptron to Deep Belief Network Tandem features for robust ASR," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, 2011, pp. 4596 –4599.

[7] Andrew L. Maas, Quoc V. Le, Tyler M. ONeil, Oriol Vinyals, Patrick Nguyen, and Andrew Y. Ng, "Recurrent Neural Networks for Noise Reduction in Robust ASR," in *Proc. of 13th Annual Conference of the International Speech Communication Association (Interspeech'12)*, Portland, Oregon, 2012, pp. 22-25.

[8] H. Zen , T. Nose , J. Yamagishi , S. Sako and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proceedings of 6th ISCA Workshop Speech Synthesis*, pp. 294-299, 2007.

[9] DSG Pollock, *A Handbook of Time-series Analysis, Signal Processing and Dynamics*, Academic press London, 1999.

[10] *MATLAB* Version 8, (R2012b), (Computer Software), The MathWorks Inc., Natick, Massachusetts.