A Neural Network Based Speech Recognition System For Isolated Tamil Words

B. Bharathi[1], V. Deepalakshmi[2], I. Nelson[3]

1) PG Student,SSN College of Engineering , Department of Computer Science and Engg.,
SSN Nagar - 603 110 Tamil Nadu, India. E-mail : b_bharathi@ssnce.ac.in

2)Lecturer SSN College of Engineering, Department of Computer Science and Engg.,
SSN Nagar – 603 110 Tamil Nadu, India.

3)Lecturer SSN College of Engineering, Department of Electronics and Communication Engg.,
SSN Nagar- 603 110 Tamil Nadu, India.

***Abstract** - Speech recognition is always looked upon as a fascinating field in human computer interaction. It is one of the fundamental steps towards understanding human cognition and their behavior. While most of the literature on speech recognition is based on Hidden Markov Models (HMM). This paper presents a neural network approach for speech recognition in Tamil language.This paper proposes a neural network approach to build a speaker independent isolated word recognition system for Tamil language. The proposed system includes six steps. First, preprocessing step is to denoise the input speech signal using wavelet transform. Second, the unvoiced part is removed by using the energy values and number of zero crossings. Thirdly, to do feature extraction based on Mel Frequency Cepstral Coefficients (MFCC). Fourthly, these feature vectors are normalized to reduce speaking rate specific variations of the features of the phonetic classes using Cepstral Mean Normalization. Next, Self Organizing Map (SOM) neural network makes each variable length MFCC trajectory of an isolated word into a fixed length MFCC trajectory and thereby making the fixed length feature vector. Finally, the resulting fixed number of feature vector is submitted to a feed forward neural network in order to recognize the spoken words.*

Keywords-Tamil speech recognition, noise removal, feature extraction, cepstral mean normalization, Self Organizing map, feed forward neural network.

I INTRODUCTION

The speech recognition problem may be interpreted as a speech-to-text conversion problem. A speaker wants his/her voice to be transcribed into text by a computer. Automatic speech recognition has been an active research topic for more than four decades. With the advent of digital computing and signal processing, the problem of speech recognition was clearly posed and thoroughly studied. These developments were complemented with an increased awareness of the advantages of conversational systems. The range of the possible applications is wide and includes: voice-controlled appliances fully featured speech-to-text software, automation of operator-assisted services, and voice recognition aids for the handicapped.

There are many distinctive features in our speech recognition system.

The system:

- is tolerant to noisy environment
- is designed for Tamil language recognition
- is implemented using neural networks
- is speaker independent speech recognition system.

In the following sections, we present the implementation stages of our system. Section 2 of the paper describes wavelet based denoising method to remove the noise in the speech signal. In section 3, we describe the silence removal algorithm. In section 4, we describe the feature extraction based on Mel Frequency Cepstral Coefficients. Section 5, describes normalization procedure using cepstral mean normalization. In section 6, describes converting variable length MFCC trajectory of an isolated word into a fixed length MFCC trajectory using Self Organizing Map. The next stage of the design to train the system for different utterances of the words in the vocabulary set. These utterances should constitute a good sample set of the various conditions and situations in which the word may be pronounced. This training was implemented on feed forward networks using back propagation algorithm. This is discussed in Section 7. Conclusion and future works are drawn in Section 8.

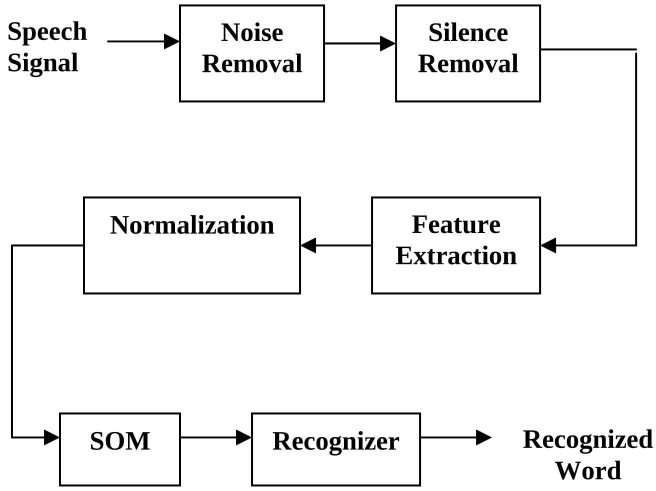

Fig. 1 The Proposed Speech Recognition System

II WAVELET BASED DENOISING

In Fourier based signal processing, the out of band noise can be removed by applying a linear time invariant filtering approach. However, it cannot be removed from the portions where it overlaps the signal spectrum. The denoising technique used in the wavelet analysis is based on an entirely different idea and assumes the amplitude rather than the location of the spectrum of the signal to be different form the noise.[2] The localising property of the wavelet is helpful in thresholding and shrinking the wavelet coefficients that helps in separating the signal from noise.

The denoising by wavelet is quite different from traditional filtering approaches because it is non-linear, due to thersholding step.

The denoising by thresholding involves the following steps:

1. Calculate a wavelet transform and order the coefficients by increasing frequency.
2. Calculate the median absolute deviation (MAD) on the largest coefficient spectrum.

$$\text{MAD} = \frac{\text{Median}(\,|c_0|,|c_1|,\ldots\ldots\ldots\ldots,|c_{2^{n-1}-1}|\,)}{0.6745}$$

Here c_0, c_1 … are the coefficients. The factor 0.6745 in the denominator rescales the numerator so that MAD is also a suitable estimator for the standard deviation for the Gaussian white noise.

3. Noise threshold will be calculated using the equation given below:

$$\text{Thresh} = \text{MAD}\,\sqrt{\ln(N)}$$

N – size of the time series.

4. Perform softthresholding of the wavelet coefficients.

```
If(coef[i] < = thresh)
    Coef[i]=0
Else
    Coef[i]=coef[i]-thresh
```

5. The coefficients obtained from step 4 are then padded with zeros to produce a legitimate wavelet transform and this is inverted to obtain the signal estimate.

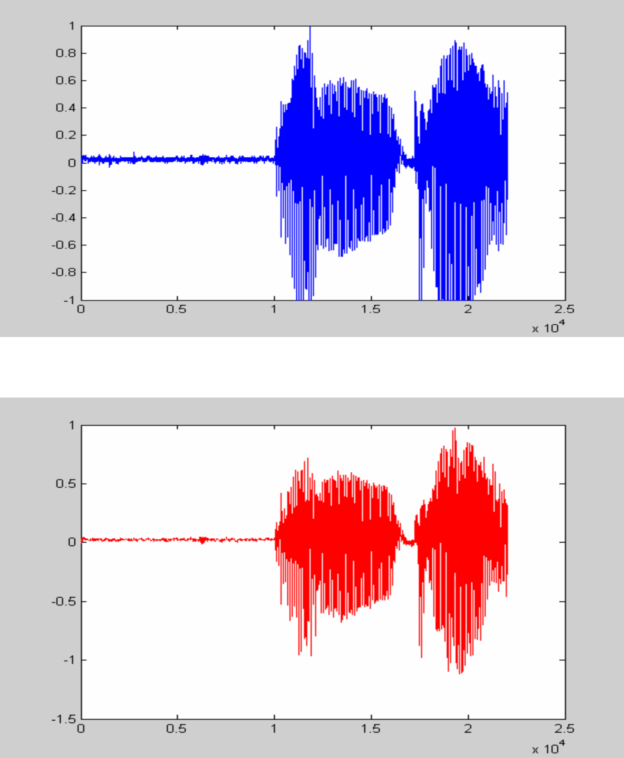

Fig.2 Speech Waveform of the word "Onnru" before and after noise removal

III SILENCE REMOVAL

The next task is to identify the presence of a speech signal. Two criteria were used to identify the presence of a spoken word. First, the total energy is measured, and second the numbers of zero crossings are counted. Both of these were found to be necessary, as voiced sounds tend to have a high volume (and thus a high total energy), but a low overall frequency (and thus a low number of zero crossings), while unvoiced sounds were found to have a high frequency, but a low volume. Only background noise was found to have both low energy and low frequency. The method was found to successfully detect the beginning and end of the several words tested.

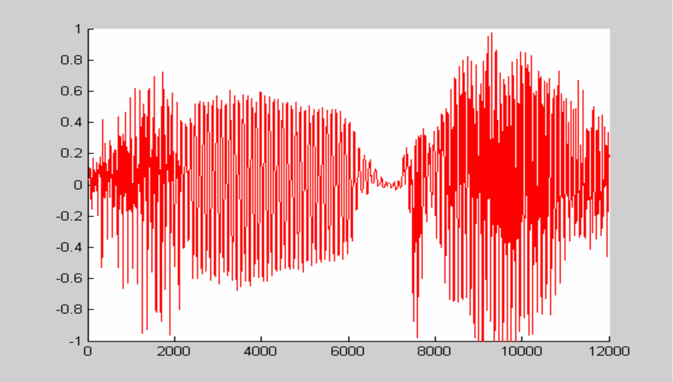

fig.3 Speech Waveform of the word "Onnru" after Silence Removal

IV FEATURE EXTRACTION

This module converts the speech waveform to parametric representation for further analysis and processing. A wide range of possibilities exist for parametrically representing the speech signal for the speech recognition task, such as Linear Predictive Coding (LPC) and Mel- Frequency Cepstrum Coefficients(MFCC).[3] MFCC's are based on the known variation of the human ear's critical bandwidths with frequency, filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is expressed in the mel-frequency scale, which is a linear frequency spacing below 1000 Hz and a logarithmic space above 1000Hz. The speech waveform is passed as input to MFCC processor that generates the MFCC co-efficient of the speech signal.

A. *Mel-Frequency Cepstrum Coefficients Processor*

A block diagram of the structure of an MFCC processor is given in fig. The speech input is typically recorded at a sampling rate above 10000Hz. The sampling frequency was chosen to minimize the effects of aliasing in the analog-to-digital conversion. The sampled signals capture all frequencies upto 5 kHz, which cover most energy of sounds that are generated by humans. The main purpose of the MFCC processor is to mimic the behavior of the human ears and MFCC's are less susceptible to variations. The following steps are involved in MFCC processor for generating MFCC.

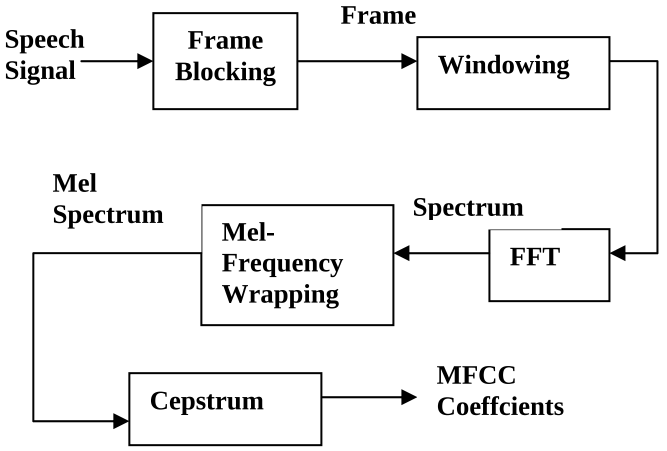

Figure 4: Block diagram of MFCC Processor

B. Frame Blocking

In this step the speech signal is blocked into frames of N samples, with adjacent frames being separated by M
(M < N). The first frame consists of the first N samples. The second frame begins M samples after the first frame, and overlaps it by N – M samples. This process continues until all the speech is accounted for within one or more frames. Typical values for N =256 (which is equivalent to ~ 30 msec windowing and facilitate the fast radix-2 FFT) and M=100'

C. Windowing

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. This minimizes the spectral distortion by using window to taper the signal to zero at the beginning and end of each frame. We define the window as w(n), 0≤ n ≤ N-1, where N is the number of samples in each frame. The result of windowing is the signal

$$Y_1 (n)=x_1(n)w(n), 0\le n \le N\text{-}1$$

Typically the Hamming window is used, which has the form:

$$W (n)=0.54 - 0.56\cos(2\Pi n / N - 1), 0\le n \le N\text{-}1$$

D. Fast Fourier Transform (FFT)

The next processing step is the Fast Fourier Transform, which converts each frame of N samples from the time domain into frequency domain. The FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT), which is defined on the set on N samples $\{x_n\}$ as follows:

$$x_{n=}\sum_{k=0}^{N-1}{}_{k} e^{-2\prod jkn/N}, \quad 0,1,2,\ldots..,N -1$$

j denotes the imaginary unit, i.e. j = √-1, x_n's are complex numbers. The resulting sequence $\{x_n\}$ is interpreted as follows:

- The zero frequency corresponds to n= 0, positive frequencies 0 < f < Fs/2 corresponds to values 1≤ n ≤ N /2-1
- Negative frequencies –Fs/2 < f < 0 correspond to N / 2 + 1 ≤ n ≤ N-1

Fs denote the sampling frequency. The result after this step is often referred to as spectrum or periodogram.

E. Mel-Frequency Wrapping

The human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency f, measured in Hz, a subjective pitch is measured on a scale called the 'mel' scale. The mel-frequency scale is linear frequency spacing below 1000Hz and a logarithmic spacing above 1000Hz. As a reference point, the pitch of 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 mels. We use the following approximate formula to compute the mels for a given frequency f in Hz:

$$\text{Mel (f)}=2595 * \log_{10}(1 + f / 700)$$

F. Cepstrum

In this final step, we convert the log mel spectrum back to time. The result is called the mel frequency cepstrum coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. As the mel spectrum coefficients are real numbers, we convert them to time domain using the Discrete Cosine Transform (DCT). $\check{S}_k$. k=1.2…K, denotes those mel power spectrum coefficients then the MFCC's, $\check{C}_n$, are calculated as follows:

$$\check{C}_{n\,=}\sum_{k=1}^{K} (\log \check{S}_k) \cos [n (k - \tfrac{1}{2}) \prod / K],$$

n = 1,2.. … K

We exclude the first component, $\check{C}_0$, from the DCT since it represents the mean value of the input signal, which carries little speaker specific information.

V CEPSTRAL MEAN NORMALIZATION

It was noticed that recognizer performance degraded because of variability in the acoustic realization of the utterance, which could come from various sources. First, it may be due to change in the environment as well as position and characteristics of the microphone. Second with in speaker, variability could result from change in speaker's physiological state, speaking rate, voice quality, socio-linguistic background, dialect, vocal tract size, shape etc. These factors contribute to change in amplitude, duration and SNR ratio for a given utterance. Hence in order to make the system more robust to above said distortions we implemented a normalization technique by which cepstral coefficients were normalized to have zero mean and unit variance within the given frame.[8]

The normalization coefficients were calculated over a relatively short sliding window (frame). The feature vectors were normalized as follows:

$$^\wedge C_{t-T} (j)= \frac{C_{t-D} (j) - \mu_t(j)}{\sigma_t(j)}$$

Where

- C_{t-D} (j) is the jth component of the original feature vector at time t – T
- $\hat{}C_{t-T}$ (j) is the normalized version
- T denotes the delay in terms of feature vectors.

The normalization coefficients, mean $\mu_t(j)$ and standard deviation $\sigma_t(j)$. For each feature vector component j were calculated over the sliding finite length normalization window as shown below

$$\mu_t(j) = 1 / N \sum_{n=1}^{N} C_n(j)$$

Standard deviation

$$\sigma_t(j) = 1 / N \sum_{n=1}^{N} (C_n(j) - \mu_t(j))^2$$

Where N denotes the normalization segment length in terms of the feature vectors. Here the mean removal can be regarded as the linear high pass filter and division by standard deviation act as an automatic gain control.

VI NEURAL NETWORK

An artificial neural network consists of a potentially large number of simple processing elements (*neurons*), which influence each other's behavior via a network of excitatory or inhibitory weights. Each unit simply computes a nonlinear weighted sum of its inputs, and broadcasts the result over its outgoing connections to other units. A training set consists of pattern of values that are assigned to designated input and/or output units. As patterns are presented from the training set, a learning rule modifies the strengths of the weights so that the network gradually learns the training set. Neural networks are usually used to perform static pattern recognition, that is, to statically map complex inputs to simple outputs, such as an N-ary classification of the input patterns. Moreover, the most common way to train a neural network for this task is via a procedure called *backpropagation* (Rumelhart 1986), whereby the network's weights are modified in proportion to their contribution to the observed error in the output unit activations (relative to desired outputs)

A. Self Organizing Map (SOM)

Since the recognizer Neural Network must have fixed number of input, here it addresses the problem of solving the variable size of the feature vector of an isolated word into a constant size. The SOM (Self Organizing Map **)** Neural Network[16] makes each variable length MFCC trajectory of an isolated word into a fixed length MFCC trajectory and thereby making the fixed length feature vector, to be fed into to the recognizer.

This neural network mainly is transforming an n-dimensional input vector space into a discretized m-dimensional space while preserving the topology of the input data. The structure of this neural network is two layered i.e. input space and output space. The training procedure is unsupervised and it is called the competitive learning and expressed as winner takes all. Compared to biological neural network here it is totally a statistical approach.

B Design of constant trajectory mapping module

Using the Self Organizing Map the variable length each and every MFCC trajectory is mapped to a constant trajectory of 6 clusters while preserving the input space. The implemented algorithm is consisting of three parts.[9]

B.1 Competitive process

Let *x* be the *m*-dimensional input vector then,

$$x = [x_1, x_2, ..., x_m]^T$$

Let w_j be the synaptic weight vector of neuron *j* then,

$$w_j = \quad [w_{j1}, w_{j2}, ..., w_{jm}]^T \qquad j = 1, 2 ..., l$$

The index of the best match neuron $i(x)$ is

$$i(x) = arg \min_j \left\| x - w_j \right\|, \quad j = 1, 2..., l$$

Where l is the total number of neurons in the network.

B.2 Cooperative process

The lateral distance excited neuron *j* and winning neuron *i* $d_{j,i}^2$ is,

$$d_{j,i}^2 = \left\| r_j - r_i \right\|^2$$

Where r_j *is the position of neuron j and* r_i *is the position of the neuron i.*

The width σ of the topological neighborhood shrinks with the time as follows:

$$\sigma(n) = \sigma_0 \exp\left(\frac{- \ n}{\tau_1} \right), \qquad n = 0, 1, 2...$$

The variation of the topological neighborhood $h_{j,i(x)}(n)$ is,

$$h_{j,i(x)}(n) = \exp\left(- \frac{d_{j,i}^2}{2\sigma^2(n)} \right), \quad n = 0, 1, 1...$$

B.3 Adaptation process

The changing of the learning rate is as follows:

$$\eta(n) = \eta_0 \exp\left(\frac{- \ n}{\tau_2} \right), \qquad n = 0, 1, 2...$$

The adaptation of weights is as follows:

$$w_j(n+1) \quad = \quad w_j(n) \quad + \quad \eta(n) h_{j,i(x)}(n) \big(x - w_j(n) \big)$$

$$where\ \eta_0 = 0.1, \tau_2 = 1000, and\ \tau_1 = \frac{1000}{\log \sigma_0}$$

The size of the center is changed dynamically with number of frames. Since it has 6 clustered centered all are initialized to random weight initially and allow the variable length trajectory of each MFCC coefficient to arrange to the size of six unique shape preserving time domain feature sequence.

VII DESIGN OF RECOGNIZER

The recognizer was designed to recognize the 10 digits and each digit input to the recognizer of size of 78 features, feature vector.

A.BackPropagation

Back propagation is the most widely used supervised training algorithm for NeuralNetworks [7]. The training of a network by backpropagation involves three stages; the feedforward of the input training pattern, the calculation and backpropagation of the associated error, and the adjustment of the weights. After training, application of the net involves only the computations of the feedforward phase. Even if the training is slow, a trained net can produce its output very rapidly.

Train the multilayer feedforward network by gradient descent to approximate an unknown function, based on some training data consisting of pairs (x,t). The vector x represents a pattern of input to the network which are the feature vectors of the signals obtained from SOM, and the vector t the corresponding target, the Tamil words corresponding to the vector passed. The overall gradient with respect to the entire training set is just the sum of the gradients for each pattern. We number the units, and denote the weight from unit j to unit i by wij.

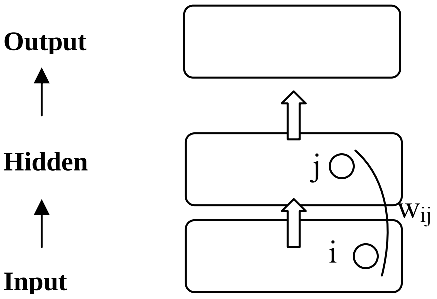

Fig. 5 A feedforward neural network, highlighting the connection from unit i to j

The backpropagation algorithm is implemented as follows:

```
1 . Initialize the input layer: $y_0 = x$
2. Propagate activity forward:
     for l = 1, 2, ..., L,
   $y_l = f_l( w_l y_{l-1} + b_l )$, where $b_l$ is the vector of bias weights.
3. Calculate the error in the output layer:
             $\delta_L = t - y_L$
4.Backpropagate the error:
   for l = L-1, L-2,.. 1,
        $\delta_L = (w_{l+1}^T \delta_{l+1} ). f_l^1$ (net l)
     where T is the matrix transposition
     operator.
5.Update the weights and biases:
     $\Delta W_l = \delta_l y_{l-1}^T$ ;
      $\Delta b_l = \delta_l$
```

For speech recognition, the acoustic observation vectors with 13 MFCC coefficients where extracted from a window of 20ms. When the words were tested with 10 speakers then 90% words were recognized correctly. The experimental results indicate that the, new approach developed for training the neural network's architecture proved to be simple and very efficient. It reduced considerably the amount of calculations needed for finding the correct set of parameters.

VIII CONCLUSION AND FUTURE WORK

The experiments made with dynamic programming and neural network learning process for distinguishing the exemplars in frequency and discriminatory template patterns for each word in the vocabulary, provided the basis for an effective Tamil speech recognition system.

The future scope of the problem is to broaden to larger vocabularies continuous speech, and different speakers and to perform word recognition in noisy environment basically words uttered over the telephone network.

REFERENCES

[1] Tebelskis, "Speech Recognition Using Neural Networks," PhD Dissertation, Carnegie Mellon University, 1995.

[2] O. Farooq, S. Datta " A Novel Wavelet based pre-processing for robust features inASR"

[3] T.Pfau,R.Falthauser,G. Ruske " A combination of speaker Normalization and speech rate Normalization for ASR"

[4] L. Rabiner and B.H. Huang. "Fundamentals of Speech Recognition "Prentice-Hall, Englewood Cliffs, NJ, 1993.

[5] Xuedong Huang, Alex Acerd,Hsiao- Wuen "Spoken Language Processing" PH PTR 2001.

[6] Chen. Audiovisual speech processing. "*Signal Processing Magazine*" 18:9–21, January 2001.

[7] Casser, M., Eck, D., and Port, R. Meter "A neural network model that learns metrical patterns" Connection Science,11(2):187–216., 1999.

[8] Amita Dev, S.S Agrawal and D Roy Choudhary " On the performance of Front-ends for Hindi Speech recognition with Degraded and Normal Speech"

[9] K.M. Peshan Sampath, P.W.D.C Jayathilake, R. Ramanan, S. Fernando, Suthrjan Dr. Chatura De Silva- "Speech Recognition using Neural Networks" .

[10] Christopher M. Bishop "Neural Networks for Pattern Recognition" Oxford University 1995.

[11] C. R. Jankowski Jr., H. H. Vo, and R. P. Lippmann, "A Comparison of Signal Processing Front Ends for Automatic Word Recognition," IEEE Transactions on Speech and Audio processing, vol. 3, no. 4, July 1995.

[12] S. Furui, "Digital Speech Processing, Synthesis and Recognition," Marcel Dekker Inc., 1989.

[13] K-F Lee, H-W Hon, and R. Reddy, "An Overview of the SPHINX Speech Recognition System," IEEE Transactions on Acoustic, Speech, and Signal Processing, vol. 38, no. 1, January 1990.

[14] H. Hasegawa, M. Inazumi, "Speech Recognition by Dynamic Recurrent Neural Networks," Proceedings of 1993 International Joint Conference on Neural Networks.