SPEECH RECOGNITION USING NEURAL NETWORKS

T. LALITH KUMAR[1], Dr. T. KISHORE KUMAR[2], Prof. K. SOUNDAR RAJAN[3]

[1]Asst. Prof., [2]Asso.prof, [3]Rector
[1] SVIST, Kadapa [2]NIT Warangal, [3]JNTU, Anantapur
E-mail: lalith_cdp2005@yahoo.co.in, kishore_1571@yahoo.co.in

Abstract: Speech processing has been an active area for several decades with a wide variety of applications ranging from communications to automatic reading machines. There are many speech recognition techniques, which are based on statistical techniques as well as neural networks. The present work investigates the feasibility of two approaches for solving the problem using Neural Networks.

Keywords: Speech Recognition, Neural networks, Multi Layer Perceptron, Recurrent Neural Network, Vector Quantization, Linear Prediction

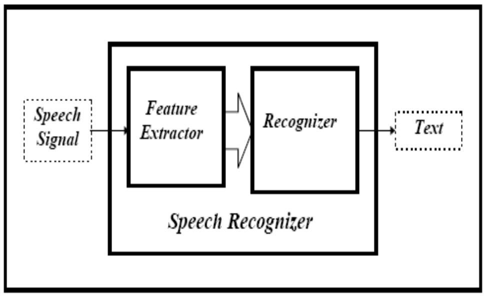
1.0 INTRODUCTION

A speech recognizer system comprises of two distinct blocks, a Feature Extractor and a Recognizer. The Feature extractor block uses a standard LPC Cepstrum coder, which translates the incoming speech into a trajectory in the LPC Cepstrum feature space. The trajectories on such reduced dimension spaces can provide reliable representations of spoken word, while reducing the training complexity and the operation of the recognizer.

The output of the FE block is blind i.e. it does not care about the word that is being represented by that trajectory. The FE block only transduces an incoming pressure wave into a trajectory in some feature space. It is the Recognizer block that discovers the relationships between the trajectories and recognizes the word. The Recognizer is to be designed in two different ways using Neural Networks. The Neural Network architectures used are Recurrent Neural Networks and the Multi Layer Perceptrons.

2.0 PROPOSED WORK

Using neural networks, a simple speaker dependent speech recognition system is implemented that can recognize the spoken digits. Many other methods have been used effectively for voice recognition, such as pattern recognition methods, HMM methods etc. but here the neural networks are used. A speech recognition system, using the pattern recognition capabilities of neural networks, and other mathematical and signal processing tools will be able to correctly identify simple words. The system will recognize samples that it trained with, and will also be able to generalize to other samples of the same word. As larger vocabularies are used, recognition accuracy will decrease. The performance of two different types of neural networks is compared.



Building Blocks of a Speech Recognition System

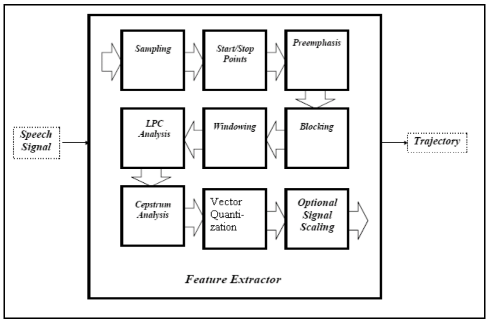
978-0-7695-3654-5/09 $25.00 © 2009 IEEE
DOI 10.1109/ICSPS.2009.51

The first step in developing this speech recognition program is to design a feature extractor. The FE block can be modeled after the stages evidenced in the human biology and development. This is a block that transforms the incoming sound into an internal representation such that it is possible to reconstruct the original signal from it. This stage can be modeled after the hearing organs, which first transduce the incoming air pressure waves into a fluid pressure wave and then converts them into a specific neuronal firing pattern.

The FE block used in speech recognition should aim towards reducing the complexity of the problem before later stages start to work with the data. Furthermore, existing relevant relationships between sequences of points in the input space have to be preserved in the sequence of points in the output space. The rate at which points in the signal space are processed by the FE block does not have to be the same rate at which points in the feature space are produced. This implies that time in the output feature space could occur at a different rate than time in the input signal space.

Once the FE block completes its work, the Recognizer module classifies its output. It integrates the sequences of phonemes into words. This module sees the world as if it where only composed of words and classifies each of the incoming trajectories into one word of a specific vocabulary. The process of correlating utterances to their symbolic expressions, translating spoken language into written language, is called speech recognition. The recognizer is built using the neural networks

3.0 METHODOLOGY ADOPTED



3.1 Feature Extractor

The speech samples from a single speaker are recorded. Five samples for each word are used for training the neural networks. The LPC Cepstrum coefficients of each word are extracted and the K-means vector quantization is applied to get the reduced trajectories.
The feature extraction consists of the following steps:

1 Speech Sampling: The speech was recorded and sampled using an off-the-shelf relatively inexpensive dynamic microphone and a standard PC sound card. The incoming signal was sampled at 22,050 Hertz with 16 bits of precision.

2 Endpoint Detection: A fast and robust technique for accurately locating the endpoints of isolated words has been used. This technique utilizes frame energy to acquire the reference points. The algorithm takes frames of size 100 samples and calculates the energy for each frame and averages it over all the frames to get the reference value of the energy. The energy per frame is calculated as:

$$P[i] = \text{Sum } k=1...j \; (s[k]^2) \qquad (1)$$

where s [k] are the speech data in the frame. Similarly P is calculated for all the frames and an average is taken for the final energy value [E].

$$E= [\text{Sum } k=l....m \; (p[k]\ ^2)]/m \qquad (2)$$

The threshold is set at (constant* E), as the detecting criterion

3 Pre-emphasis: As is common in speech recognizers, a pre-emphasis filter was applied to the digitized speech to spectrally flatten the signal and diminish the effects of finite numeric precision in further calculations. This type of filter boosts the magnitude of the high frequency components, leaving relatively untouched the lower ones.

4 Framing and Windowing: After the signal was sampled, the utterances were isolated, and the spectrum was flattened, each signal was divided into a sequence of data blocks, each block spanning 300 samples, and separated by 100 samples. Next, each block was multiplied by a Hamming window, which had the same width as that of the block, to lessen the leakage effects.

5 LPC Analysis: Then, a vector of 12 Linear Predicting Coding (LPC) Cepstrum coefficients was obtained from each data block using Durbin's method.

6 Vector Quantization: The dimensionality of the LPC Cepstrum vectors is reduced using Vector Quantization Technique. A total of 36 coefficients are obtained after the vector quantization. For the vector quantization the K-means algorithm is used.

The way in which a set of L training vectors can be clustered into a set of M codebook vectors is the following:

1. Initialization: Arbitrarily choose M vectors (initially out of the training set of L vectors) as the initial set of code words in the codebook.
2. Nearest-Neighbor Search: For each training vector, find the code word in the current codebook that is closest (in terms of spectral distance) and assign that vector to the corresponding cell.
3. Centroid Update: Update the code word in each cell using the centroids of the training vectors assigned to that cell.
4. Iteration: Repeat the steps 2 and 3 until the average distance falls below a preset threshold.

After the VQ stage only 3 vectors of size 12 are left. The output of this last stage is the final feature used throughout.

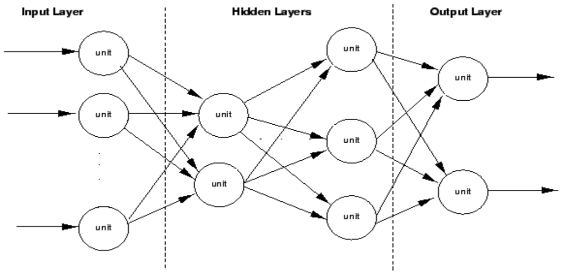
3.2 Recognizer

The recognizer block is built using the neural network approach. The 2 types of Neural networks used are the Multi layer Perceptrons and Recurrent Neural Networks. A neural network is a collection of layers of "neurons," simulating the human brain structure. Each neuron takes input from each neuron in the previous layer (or from the outside world, if it is in the first layer). Then, it adds this input up, and passes it to the next layer. Each connection between layers, however, has a certain weight. Every time the neural network processes some input, it adjusts these weights to make the output closer to a given desired value for the output. After several repetitions of this (each repetition is an iteration), the network can produce the correct output given a loose approximation of the input.

4.0 RESULTS

The 2 different approaches were used for recognition. For each word 5 different training samples were used and the networks are trained. Then the recognition accuracies were calculated by recording more samples of the words.

4.1 MLP Approach



Architecture of a multi-layer perceptron with two hidden layers

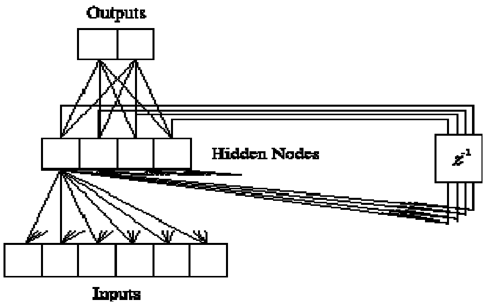
The MLP had 36 input nodes, 36 hidden neurons, and 1 output neuron. The output of the neurons was inside the interval [-1,+1]. The transig was used as the threshold function. Each neuron had an extra connection, whose input was kept constant and equal to one (the literature usually refers to this connection as bias or threshold). The weights were initialized with random values selected within the small interval. The MLP was trained using the Error back propagation method.

Table 1. Testing set results for MLP approach

Digit	% of Errors	Recognition Accuracy
0	10	90%
1	0	100%
2	0	100%
3	0	100%
4	30	70%
5	20	80%
6	10	90%
7	20	80%
8	0	100%
9	20	80%

4.2 Recurrent Neural Network Approach

Strcture of Recurrent neural network



While training an Elman network the following occurs.
At each epoch:

1) The entire input sequence is presented to the network, and its outputs are calculated and compared with the target sequence to generate an error sequence.
2) For each time step, the error is back propagated to find gradients of errors for each weight and bias. This gradient is actually an approximation since the contributions of weights and biases to errors via the delayed recurrent connection are ignored.
3) This gradient is then used to update the weights with the back prop training function chosen by the user.

Table 2. Testing set results for RNN approach

Digit	% of Errors	Recognition Accuracy
0	20	80%
1	20	80%
2	0	100%
3	20	80%
4	30	70%
5	10	90%
6	10	90%
7	30	70%
8	0	100%
9	10	90%

5.0 CONCLUSIONS

Some comments concerning the MLP approach can also be made:

1. Its recognition accuracies were better than the ones obtained with the RNN approach. Even though its performance was better, it is still below the limits required for practical applications.

2. The input layer consists of 36 neurons. The hidden layer was defined by 36 hidden neurons, having a total of 1296 weights and totaling 1296 floating point values. The output layer consisted of only 1 neuron, with 36 weights totaling 36 floating point values. In total, each MLP required 1332 floating point values.

A few comments concerning the RNN can also be made in the light of the items used for comparison above:

1. It achieved only 80% of recognition accuracy.
2. In terms of memory requirement, it is the best. The fully connected RNN with 10 hidden neurons and 1 output neuron requires only 360 floating point values.

REFERENCES

1. Papoulis, “Probability, Random Variables, and Stochastic Processes”, McGraw-Hill, 1991.
2. S. Haykin, “Neural Networks: A Comprehensive Foundation,” Macmillan College.
3. T. Kohonen, “The Neural Phonetic Typewriter,” Computer, Vol.21, No.3, 1988.
4. K. J. Lang and A. H. Waibel, “A Time-Delay Neural network Architecture for Isolated Word Recognition,” Neural networks, Vol.3, 1990.
5. E. Singer and R. P. Lippmann, “A Speech Recognizer using Radial Basis Function Neural Networks in an HMM Framework,” IEEE Proceedings of the ICASSP, 1992.
6. H. Hild and A. Waibel, “Multi-Speaker/Speaker-Independent Architectures for the Multi-State Time Delay Neural Network,” IEEE Proceedings of the ICNN, 1993.
7. R. M. Gray, “Vector Quantization,” IEEE ASSP Magazine, April 1984.
8. J .Tebelskis, “Speech Recognition Using Neural Networks,” PhD Dissertation Carnegie Mellon University, 1995.
9. L. Rabiner and G. Juang, “Fundamentals of Speech Recognition,” Prentice-Hall, 1993.
10. H. Hasegawa, M. Inazumi, “Speech Recognition by Dynamic Recurrent Neural Networks,” Proceedings of 1993 International Joint Conference on Neural Networks.
11. Proakis and Monolakis, “Digital Signal Processing and its Applications” Prentice Hall.