A Novel Approach to Speech Recognition by Using Generalized Regression Neural Networks

Lakshmi Kanaka Venkateswarlu Revada[1], Vasantha Kumari Rambatla[2] and Koti Verra Nagayya Ande[3]

[1] Professor and Head, Department of Information Technology
Sasi Institute of Technology and Engineering, Tadepalligudem, INDIA.

[2] Principal
Perunthalaivar Kamarajar Arts College,PUDUCHERRY 605 107

[3] Lecturer, Department of Information Technology
Sasi Institute of Technology & Engineering, Tadepalligudem, INDIA.

Abstract

Speech recognition has been a subject of active research in the last few decades. In this paper, the applicability of a special model of Generalized Regression Neural Networks as a classifier is studied. A Generalized Regression Neural Network (GRNN) is often used for function approximation. It has a radial basis layer and a special linear layer. This network uses a competitive function for computing final result. The proposed network has been tested on one digit numbers dataset and produced significantly lower recognition error rate in comparison with common pattern classifiers. All of classifiers use Linear Predictive Cepstral Coefficients and Mel - Frequency Cepstral Coefficients. Results for proposed network shows that LPCC features yield better performance when compared to MFCC. It is found that the performance of Generalized Regression Neural Networks is superior to the other classifiers namely Linear and Multilayer Perceptron Neural Networks.

Keywords: *Cepstral Coefficients, Linear Predictive Cpestral Coefficients, Mel –Frequency Cpestral Coefficients Linear Neural Networks, Multilayer Perceptrons, Generalized Regression Neural Networks, Classifiers.*

1. Introduction

A major problem in speech recognition system is the decision of the suitable feature set which can accurately describe in an abstract way the original highly redundant speech signal. In non-metric spectral analysis, Mel-frequency Cepstral Coefficients (MFCC) are one of the most popular spectral features in ASR. In parametric spectral analysis, the LPC Mel- Cepstrum based on an all-pole model is widely used because of its simplicity in computation and high efficiency [6]. Neural networks have seen an explosion of interest over the last few years, and are being successfully applied across an extraordinary range of problem domains, in areas as diverse as finance, medicine, engineering, geology and physics. Indeed, anywhere that there are problems of prediction, classification or control, neural networks are being introduced. Neural networks are very sophisticated modeling techniques, capable of modeling extremely complex functions. For many years linear modeling has been the commonly-used technique in most modeling domains, since linear models had well-known optimization strategies. Neural networks also keep in check the curse of dimensionality problem that bedevils attempts to model non-linear functions with large numbers of variables. Neural networks learn by example. The neural network user gathers representative data, and then invokes training algorithms to automatically learn the structure of the data. Although the user does need to have some heuristic knowledge of how to select and prepare data, how to select an appropriate neural network, and how to interpret the results, the level of user knowledge needed to successfully apply neural networks is much lower than would be the case using some more traditional non-linear statistical methods.

Neural networks are also intuitively appealing, based as they are on a crude low-level model of biological neural systems. In the future, the development of this neuro-biological modeling may lead to genuinely intelligent computers. Meanwhile, the simple neural networks modeled by Trajan already add a significant weapon to the armory of the applied statistician. Neural networks have been used in conjunction with speech recognizers in various ways for automatic speech recognition. Tamura and Waibel [13] applied a neural network to reduce noise in speech signals. Barbier and Chollet [9] used a neural network and a dynamic time warping (DTW) algorithm for speaker-dependent word recognition in cars. Sorensen [12] used two neural networks in tandem for both noise reduction and isolated word recognition under F-16 jet noise. Huang [10] used a set of neural networks to establish a non-linear mapping function between two speakers. This paper describes a method for estimating a continues targets for training patterns of NNs based on the Generalized Regression Neural Networks and the performance is compared with the performance of Linear and Multilayer Perceptrons.

2. System Concept

2.1. Dataset

A sequence of 10 isolated digits (0, 1, 2, …, 9) voices from 35 different speakers (20 Male and 15 Female) were recorded. So there are 350 wave files. We divided them into two separate parts, 20 speakers (200 wave files) for training and 15 remaining speakers (150 wave files) for testing. So the ratio of train to test is 4:3.

2.2 Preprocessing

The speech signals are recorded in a low noise environment with good quality recording equipment. The signals are samples at 11kHz. Reasonable results can be achieved in isolated digit recognition when the input data is surrounded by silence.

2.3 Sampling Rate

150 samples are chosen with sampling rate 11kHz, which is adequate to represent all speech sounds.

2.4 Windowing

In order to avoid discontinuities at the end of speech segments the signal should be tapered to zero or near zero and hence reduce the mismatch.

3. Feature Extraction

The goal of feature extraction is to represent speech signal by a finite number of measures of the signal. This is because the entirety of the information in the acoustic signal is too much to process, and not all of the information is relevant for specific tasks. In present Speech Recognition systems, the approach of feature extraction has generally been to find a representation that is relatively stable for different examples of the same speech sound, despite differences in the speaker or various environmental characteristics, while keeping the part that represents the message in the speech signal relatively intact.

Linear predictive coding (LPC) is a tool used mostly in audio signal processing and speech processing for representing the spectral envelope of a digital signal of speech in compressed form, using the information of a linear predictive mode. It is one of the most powerful speech analysis techniques, and one of the most useful methods for encoding good quality speech at a low bit rate and provides extremely accurate estimates of speech parameters. LPC analyzes the speech signal by estimating the formats, removing their effects from the speech signal, and estimating the intensity and frequency of the remaining buzz. The process of removing the formants is called inverse filtering, and the remaining signal after the subtraction of the filtered modeled signal is called the reside.

The number which describe the intensity and frequency of the buzz, the formants, and the reside signal, can be stored or transmitted somewhere else. LPC synthesizes the speech signal by reversing the process: use the buzz parameters and the residue to create a source signal. Use the formants to create a filter (which represents the tube), and run the sources through the filter, resulting in speech. Because speech signals vary with time, this process is done on short chunks of the speech signal, which are called frames; generally 30 to 50 frames per second give intelligible speech with good compression.

LPC is frequently used for transmitting spectral envelope information, and as such it has to be tolerant of transmission errors. Transmission of the filter coefficients directly is undesirable, since they are very sensitive to errors. In other words, a very small error can distort the whole spectrum, or worse, a small error might make the prediction filter unstable.

LPC is generally used for speech analysis and resynthesis. It is used as a form of voice compression by phone companies, for example in the GSM standard. It is also used for secure wireless, where voice must be digitized, encrypted and sent over a narrow voice channel.

In the LPC analysis one tries to predict x_n on the basis of the p previous samples,

$$x_n^{'} = \sum a_k \, x_{n-k}$$

Then $\{a_1, a_2, ..., a_p\}$ can be chosen to minimize the prediction power Q_p where

$$Q_p = E\left[\left| x_n - x_n^{'} \right|^2 \right]$$

The most popular feature set has been the vector of Mel-frequency cepstral coefficients traditionally used also in speech recognition. MFCC's are cepstral coefficients computed on a warped frequency scale based on known human auditory perception. In a typical MFCC processing, the first step is windowing the speech signal to divide the speech into frames. Since high frequency formants have smaller amplitude than low frequency formants, high frequencies may be emphasized to obtain similar amplitude for all formants. After windowing, FFT is used to find the power spectrum of each frame. Then perform filter bank processing to the power spectrum, which uses Mel-scale. Discrete cosine transformation is applied after converting the power spectrum to log domain in order to compute MFCC coefficients.

Linear Predictive Coding is used to extract the LPCC coefficients from the speech tokens. The LPCC coefficients are then converted to cepstral coefficients. The cepstral coefficients are normalized in between -1 and 1. The speech is blocked into overlapping frames of 20ms every 10ms using Hamming window.

LPCC was implemented using the autocorrelation method. A drawback of LPCC estimates is their high sensitivity to quantization noise. Convert LPCC coefficients into cepstral coefficients where the cepstral order is the LPCC order and to decrease the sensitivity of high and low-order cepstral coefficients to noise, the obtained cepstral coefficients are then weighted.

16 Linear Predictive Cepstral Coefficients are considered for windowing. Linear Predictive Coding analysis of speech is based on human perception experiments. Sample the signal with 11 kHz. Number of frames are obtained for each utterance from LPC coefficients.

Feature extraction consists of computing representations of the speech signal that are robust to acoustic variation but sensitive to linguistic content. The Mel-filter is used to find band filtering in the frequency domain with a bank of filters. The filter functions used are triangular in shape on a curvear frequency scale. The filter function depends on three parameters: the lower frequency, the central frequency and higher frequency. On a Mel scale the distances between the lower and the central frequencies and that of the higher and the central frequencies are equal. The filter functions are

$$H(f)=0 \;\; for \; f \le f_l \; and \; f \ge f_h$$

$$H(f)=(f-f_l)/(f_c-f_l) \;\; for \;\; f_l \le f \le f_c$$

$$H(f)=(f_h-f)/(f_h-f_c) \;\; for \;\; f_c \le f \le f_h$$

Mel frequency cepstral coefficients are found from the Discrete Cosine Transform of the Filter bank spectrum by using the formula given by Davis and Mermelstein [1980].

$$c_i = \sum_{j=1}^{N} P_j \cos(i\pi / N(j-0.5))),$$

Pj denotes the power in dB in the jth filter and N denotes number of samples.

12 Mel frequency coefficients are considered for windowing. Mel-Frequency analysis of speech is based on human perception experiments. Sample the signal with 11 kHz, apply the sample speech data to the Mel-filter and the filtered signal is trained. Number of frames are obtained for each utterance from MFC coefficients.

4. Recognition Methodology

In multi-class mode such as the present case, each classifier tries to identify whether the set of input feature vectors, derived from the current signal, belongs to a specific class of numbers or not, and to which class exactly. For samples that can not be realized as a specific class a random class is selected.

5. Classifiers

Several classifiers are tested for mentioned dataset. The structures of successful classifiers in recognition are described in following subsections.

5.1. Linear Networks:

A general scientific principal is that a simple model should always be chosen in preference to a complex model, if the latter does not fit the data better. In terms of function approximation, the simplest model is the linear model, where the fitted function is a hyperplane. In classification, the hyperplane is positioned to divide the two classes (a linear discriminant function); in regression, it is positioned to pass through the data. A linear model is typically represented using an NxN matrix and an Nx1 bias vector.

A neural network with no hidden layers, and an output with dot product synaptic function and identity activation function, actually implements a linear model. The weights correspond to the matrix, and the thresholds to the bias vector. When the network is executed, it effectively multiplies the input by the weights matrix then adds the bias vector.

5.2. Multi-Layer Perceptron

This is perhaps the most popular network architecture in use today, due originally to Rumelhart and McClelland (1986). The units each performed a biased weighted sum of their inputs and pass this activation level through a transfer function to produce their output, and the units are arranged in a layered feedforward topology. The network thus has a simple interpretation as a form of input-output model, with the weights and thresholds (biases) the free parameters of the model. Such networks can model functions of almost arbitrary complexity, with the number of layers, and the number of units in each layer, determining the function complexity. Important issues in MLP design include specification of the number of hidden layers and the number of units in these layers.

The number of input and output units is defined by the problem (there may be some uncertainty about precisely which inputs to use, a point to which we will return later. However, for the moment we will assume that the input variables are intuitively selected and are all meaningful). The number of hidden units to use is far from clear. As good a starting point as any is to use one hidden layer, with the number of units equal to half the sum of the number of input and output units. Again, we will discuss how to choose a sensible number later.

5.3. Generalized Regression Neural Networks

Generalized regression neural networks (GRNNs) work in a similar fashion to PNNs, but perform regression rather than classification tasks. As with the PNN, Gaussian Kernel functions are located at each training case. Each case can be regarded, in this case, as evidence that the response surface is a given height at that point in input space, with progressively decaying evidence in the immediate vicinity. The GRNN copies the training cases into the network to be used to estimate the response on new points. The output is estimated using a weighted average of the outputs of the training cases, where the weighting is related to the distance of the point from the point being estimated (so that points nearby contribute most heavily to the estimate).

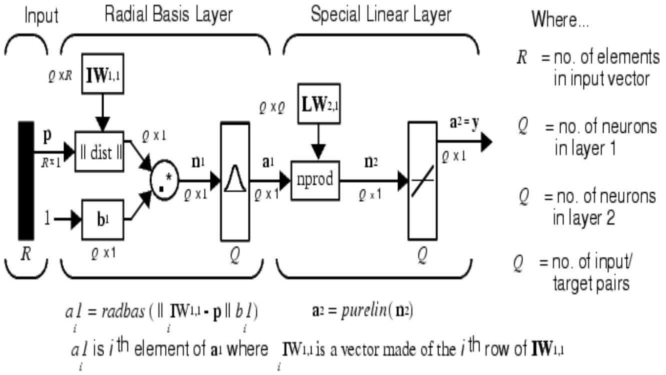

Input
Radial Basis Layer
Special Linear Layer
Where...
R = no. of elements in input vector
Q = no. of neurons in layer 1
Q = no. of neurons in layer 2
Q = no. of input/ target pairs
Q x R
IW1,1
p
R x 1
|| dist ||
Q x 1
.*
n1
Q x 1
1
b1
Q x 1
R
Q
a1
Q x 1
Q x Q
LW2,1
nprod
n2
Q x 1
a2 = y
Q x 1
Q
a1_i = radbas (|| iIW1,1 - p || b1_i)
a2 = purelin(n2)
a1_i is i th element of a1 where iIW1,1 is a vector made of the i th row of IW1,1

Figure 1: Generalized Regression Neural Network Architecture.

The first hidden layer in the GRNN contains the radial units. A second hidden layer contains units that help to estimate the weighted average. This is a specialized procedure. Each output has a special unit assigned in this layer that forms the weighted sum for the corresponding output. To get the weighted average from the weighted sum, the weighted sum must be divided through by the sum of the weighting factors. A single special unit in the second layer calculates the latter value. The output layer then performs the actual divisions (using special division units). Hence, the second hidden layer always has exactly one more unit than the output layer. In regression problems, typically only a single output is estimated, and so the second hidden layer usually has two units.

The GRNN can be modified by assigning radial units which represent clusters rather than each individual training case: this reduces the size of the network and increases execution speed. Centers can be assigned using any appropriate algorithm (i.e., sub-sampling, K-means or Kohonen), and Trajan adjusts the internal weightings to take account.

GRNNs have advantages and disadvantages broadly similar to PNNs - the difference being that GRNNs can only be used for regression problems, whereas PNNs are used for classification problems. A GRNN trains almost instantly, but tends to be large and slow (although, unlike PNNs, it is not necessary to have one radial unit for each training case, the number still needs to be large). Like an RBF network, a GRNN does not extrapolate.

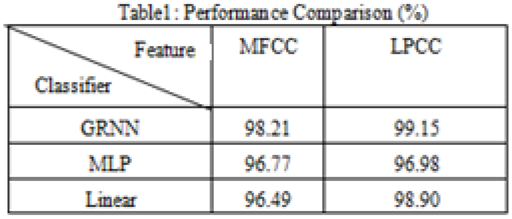
6. TRAINING PHASE

The networks are usually trained to perform tasks such as pattern recognition, decision-making, and motory control. The original idea was to teach them to process speech or vision, similarly to the tasks of the human brain. Nowadays tasks such as optimization and function approximation are common. Training of the units is accomplished by adjusting the weight and threshold to achieve a classification. The adjustment is handled with a learning rule from which a training algorithm for a specific task can be derived. The Linear, Multilayer Perceptron and Generalized Regression Neural Networks are trained for spoken digits for 20 speakers. The learning rate is taken as 0.01, momentum rate is taken as 0.3. Number of epochs are taken as 100. The Random Gaussian Method is chosen for initialization.

6.1 Performance Evaluation

In the training nearly all of described classifiers recognized training patterns performances are obtained with accuracy above or equal to 96.49%. The performance for each classifier against two features have been computed and presented in Table 1.

Table1 : Performance Comparison (%)



Feature / Classifier	MFCC	LPCC
GRNN	98.21	99.15
MLP	96.77	96.98
Linear	96.49	98.90

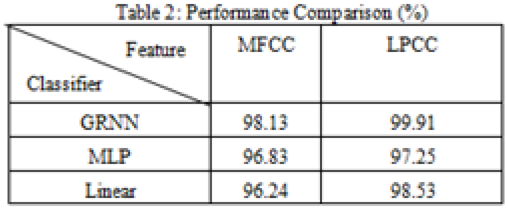
7. TESTING PHASE

The same Linear, Multilayer Perceptron and Generalized Regression Neural Networks are trained for spoken digits for the reaming 15 speakers. The learning rate, momentum rate and the number of epochs chosen are same as in the training case. The initialization chosen is also same as that of training phase.

7.1 Performance Evaluation

In the testing nearly all of described classifiers recognized testing patterns with accuracy above or equal to 96.24%. The performance for each classifier against two features have been computed and presented in Table 2.

Table 2: Performance Comparison (%)



Feature / Classifier	MFCC	LPCC
GRNN	98.13	99.91
MLP	96.83	97.25
Linear	96.24	98.53

8. CONCLUSION

The Generalized Regression Neural Network architecture has been shown to be suitable for the recognition of isolated digits. Recognition of the digits is carried out in speaker dependent mode. In this mode the tested data presented to the network are different from the trained data. The 16 Linear Predictive Cepstral Coefficients with 16 parameters from each frame improves a good feature extraction method for the spoken digits, since the first 16 in the cepstrum represent most of the formant information. It is found that the performance of all classifiers for LPCC features exceeds the performance of all classifiers with MFCC features. The promising results are obtained both in the training and testing phases due to the exploitation of discriminative information with neural networks.

REFRENCES

[1] Edric Gaudard , Guillermo Aradilla ," Speech Recognition based on Template Matching and Phone Posterior probabilities" , 2007, IDIAP-Com 07-02.
[2] Leszek Rutkowski, "Adaptive Probabilistic Neural Networks for Pattern Classification in Time-Varying Environment", IEEE TRANSACTIONS ON NEURAL NETWORKS, 2004,VOL. 15, NO.4, pp. 811-827.
[3] F. Gorunescu, "Architecture of probabilistic neural networks: estimating the adjustable smoothing factor," Research Notes in Artificial Intelligence and Digital Communications, 2004,104, pp. 56-62.
[4] Todor Ganchev, Dimitris K. Tasoulis, Michael N. Vrahatis, "Locally Recurrent Probabilistic Neural Network for Text-Independent Speaker Verification", in Proc. of the Euro Speech, 2003, vol. 3, pp. 762-766.
[5] Todor Ganchev, nastasios,Tsopanoglou, Nikos Fakotakis, George Kokkinakis, "Probabilistic neural networks combined with GMMs for speaker recognition over telephone channels", 14 International Conference on Digital Signal , July 2002, Volume II, pp.1081-1084.
[6] Harosha Matsumoto, Masanora Moroto, "Evaluation of MEL-LPC cepstrum in a large vocabulary continuous speech recognition", Acoustics, Speech, and Signal Processing Proceedings IEEE International Conference on Volume 1, Issue, 2001, pp. 117-120.

[7] Kosaka, T, Omatu, S, “Classification of the Italian Lira using the LVQ method”, Systems, Man, and Cybernetics, 2000 IEEE International Conference on Volume 4, pp. 2769 – 2774.
[8] J. Barry Gomm, Ding Li Yu, “Selecting Radial Basis Function Network Centers with Recursive Orthogonal Least Squares Training”, IEEE Trans. Neural Networks, March 2000, vol.lI,N0.2.
[9] K.K. Yiu, M.W. Mak, C.K. Li, “Gaussian mixture models and probabilistic decision-based neural network for pattern classification: A comparative Study”, Neural Computing and Applications 1999, 8(3): pp. 235- 245.
[10] X.Huang. Speaker normalization for speech recognition. IEEE International Conference on Acoustics, Speech, and Signal Processing, March 1992, 1:465-468.
[11] L.Barbier and G.Chollet. Robust speech parameters extraction for word recognition in noise using neural networks. IEEE International Conference on Acoustics, Speech, and Signal Processing, May 1991, 1:145-148.
[12] H.Sorensen. A cepstral noise reduction multi-layer neural netwok. IEEE International Conference on Acoustics, Speech, and Signal Processing, May 1991, 1:933-936.
[13] S.Tamura and A.Waibel. Nosie reduction using connectionist models. IEEE International Conference on Acoustics, Speech, and Signal Processing, April 1988, 1:553-556.

Dr.R.L.K.Venkateswarlu educational qualifications are M.Tech, Ph.D. He is currently working as Professor & Head in the Department of Information Technology, Sasi Institute of Technology and Engineering, Tadepalligudem. He acted as a coordinator for “ Anveshana” National level Technical Symposium for 2010 and 2011. He got Best paper award in the National Seminar Advances in Mathematical, Statistical and Computational Methods in Science and Technology, November 29-30, 2001 organized by DRDO and ISMU. Dr.R.L.K.Venkateswarlu has attended and presented good number of research papers in national and international conferences. He has also national and international publications to his credit. He is gold medalist and awarded best teacher & researcher from JNTUK & ISM University. His area of interest is Speech recognition, Pattern recognition, Neural networks & Earth quake engineering. He is life member of Indian Society of Technical Education ISTE, Indian Society of Industrial & Applied Mathematics and International Journal of Computational Mathematical Ideas.

Dr. R.Vasanthakumari obtained her Ph.D degree from Pondicherry University, in 2005. She is working as a Principal in Government College, Puducherry, India. She has more then 27 years of teaching experience in P.G. and U.G. colleges. She is guiding many research scholars and has published many papers in national and international conference and in many international journals. Her area of interest is Speech recognition, Pattern recognition, Neural networks & Fluid Dynamics.

Sri A.K.V.Nagayya working as a lecturer in Department of IT, Sasi Institute of Technology And Engineering. He did his MCA from Andhra University in 2008. His area of interest is Data Mining & DataWare Housing and Neural Networks.