SPEAKER INDEPENDENT PHONEME RECOGNITION USING NEURAL NETWORKS

N.Uma Maheswari[1],A.P.Kabilan[2] , R.Venkatesh[3]

[1]Senior Lecturer, Dept. of CSE, P.S.N.A College of Engg& Technology, Dindigul-624622,India

[2]Principal, Chettinad college of Engg& Technology, Karur-639114,India

[3]Senior Lecturer, Dept. of CSE, R.V.S College of Engg& Technology, Dindigul-624005,India

E-mail: numamahi@gmail.com , numamahi@yahoo.com

ABSTRACT

Phoneme recognition is important for successful development of speech recognizers in most real world applications. While speaker dependent phoneme recognizers have achieved close to 100% accuracy, the speaker independent phoneme recognition systems have poor accuracy not exceeding 75%.In this paper we describe a two-module speaker independent phoneme recognition system for all-Indian English speech. The first module performs classification of phonemes recognition using Probabilistic neural networks. The second module executes the recognized phonemes from the classified phonemes employing Recurrent Neural Networks. The system was trained by Indian English speech consisting of 1000 words uttered by 50 speakers. The test samples comprised 500 words spoken by a different set of 30 speakers. The recognition accuracy is found to be 98% which is well above the previous results.

Keywords: *Speaker Independent Phoneme Recognition, Probabilistic Neural Networks, Recurrent Neural Networks, Phonemes*.

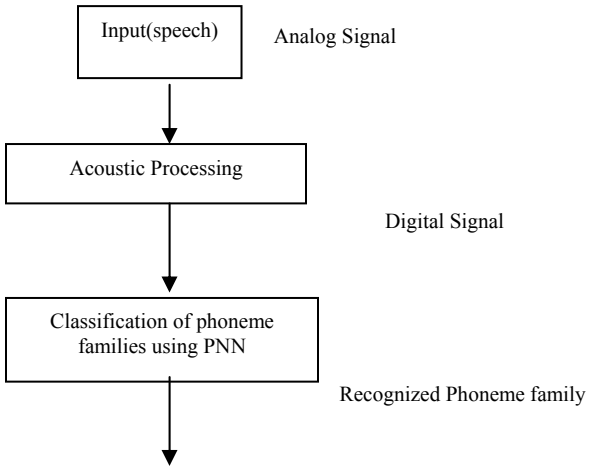
I. INTRODUCTION

Automatic speech recognition is a process by which a machine identifies speech. The machine takes a human utterance as an input and returns a string of words , phrases or continuous speech in the form of text as output. As ASR technology matures, the range of possible applications increases. However, a domain and speaker independent system able to correctly decode all speech found in communication between people into strings of words is not realistic with the current state of technology. The system we have in mind needs to be general, however, which is the main reason for experimenting with phoneme recognition. In our system, the only constraints imposed on the acoustic hypotheses are the phoneme recognition. Apart from this, the system does not incorporate any linguistic knowledge, since it is only intended to output a phoneme string representation of the input speech. Thus, it relies heavily on the human user's language capability. Another motivation for phoneme recognition is speed. The more complex the system, the longer the decoding time. In this work we have adopted a bottom-up approach, in which a speech signal is resolved into a classification of phonemes and identification of phonemes which results in phoneme recognition

II. SYSTEM ARCHITECTURE FOR PHONEME RECOGNIZER

The proposed speech recognition method comprises of three steps: acoustic signal processing, phoneme recognition and word recognition (Figure 1). First, we digitize

the input speech waveform phoneme-by-phoneme. Phonemes are recognized using artificial neural network(high level and
low level) and subsequently words are recognized from the clusters of phonemes using HMM..



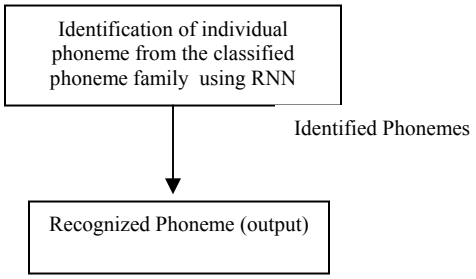


Fig.1 System architecture for Phoneme Recognizer

III. PHONEME RECOGNITION USING NEURAL NETWORKS

Speech is the system of acoustical elements which we use to communicate verbally. Those elements are called phonemes. Phonemes are produced via the human body using the glottis, lungs, vocal cavity, nasal cavity, tongue, and other body parts. Phonemes can be linked together to form syllables, which can be linked together to form words. The phonemes are classified as in the given phoneme table 1

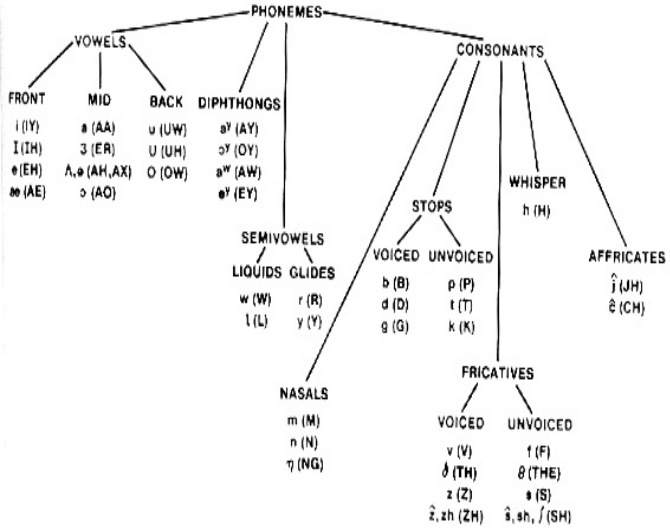


Table 1.Phoneme Classification

This paper proposes a modular-based classifier for the problem of phoneme recognition which can be used for speech recognition. A phoneme is the smallest meaningful distinguishable unit in a language's phonology. Since the total number of phonemes for each language is finite, the goal of phoneme recognition is to classify a speech signal into phonemes with a given set of speech features. We apply a two level classifier method to design the phoneme recognition system. It uses both statistical and neural network based methods in a hierarchical modular system.

We use the concept of phoneme families. To obtain phoneme families, we employ k-mean clustering method[16]. A given unknown phoneme is first classified into a phoneme family at high level classification using Probabilistic Neural Networks(PNN). Due to the powerful characteristics of probabilistic neural networks such as rapid training, convergence to the Bayesian classifier and generalization, we use this statistical-based classifier to construct an initial topology of the proposed hierarchical modular system[3],[4]. Then, the exact label of the phoneme is determined at low level classification using Recurrent neural network(RNN).Multilayer perceptron(MLP) and recurrent neural network (RNN) are employed as local experts to discriminate time-invariant and time-variant phonemes, respectively.

A. Proposed Phoneme Recognition System

We propose a new modular-based approach for phoneme recognition problem. This method consists of two levels of classification: high and low. We define various phoneme families in the high level classification. To obtain the phoneme families, clustering techniques such as *k*-mean clustering are applied. To find an appropriate number of phoneme families, we consider different values of *k*. The value yielding a lower classification error rate is chosen as the best value. Here we have taken the total phoneme families at low level, i.e. $k = 7$.A typical architecture of the proposed system is presented in Fig.2 with k=3

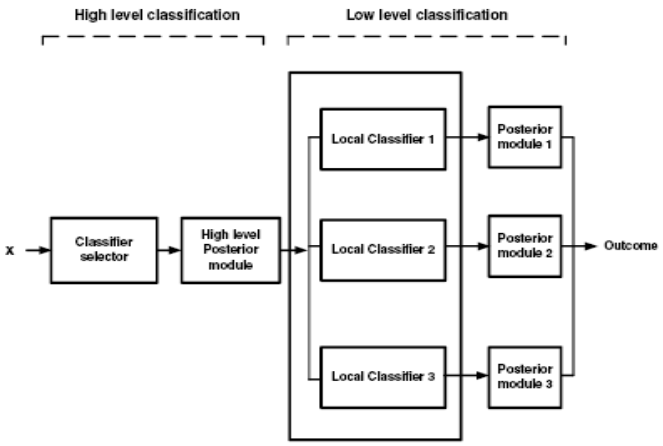


Fig. 2.A typical architecture of the proposed system

There are several components in the system comprising classifier selector, high level posterior module, low level classifiers, and low level posterior modules. An unknown input acoustic waveform is digitized with 38 samples (which was found to be the optimum were fed into the high level classifier. Here we use input layer with 38 nodes, two hidden layers with 12 and 6 nodes each and an output layer with 7 nodes. The high level classifier recognizes an unknown phoneme *X* as phoneme family k^* as follows:

$$k^* = arg\max_k \sum_{l=1}^{\ell} DM_{high}(l,k)\ ;\ k = 1,2,...,K;$$

(1)
where,

$$DM_{high}(l,k) = \begin{cases} 1 & \text{if } k = arg\max_k p(f_l|CL_k) \\ 0 & \text{Otherwise} \end{cases}$$

(2)

where, p(fl/CLk) stands for the posterior probability of the *lth* window of frame-level data given phoneme family *CLk*. Also, l denotes the number of windows of phoneme. In other words, phoneme data is fed to the high level classification on window-by-window basis. Posterior probability of each window of frame-level data given all phoneme families is obtained through high level classifier. Then, label of the pattern is estimated by aggregating the responses over all windows using high level posterior module.

For each phoneme family, we designate a classifier to perform low level classification. Suppose that the output of the high level classification for input pattern *X* is denoted by *CLk*. Hence, the corresponding low level classifier classifies *X* as member j^* of *CLk* if

$$j^* = arg\max_j \sum_{l=1}^{\ell} DM_{low}^k(l,j)\ ;\ j = 1,2,...,m_k;$$

(3)

where,

$$DM_{low}^k(l,j) = \begin{cases} 1 & \text{if } k = arg\max_j p(f_l|CL_{k,j}) \\ 0 & \text{Otherwise} \end{cases}$$

(4)

This approach requires two different algorithms which are learning and classification. The learning algorithm expressed in Algorithm 1, identifies the proper topology of the system. The classification algorithm presented in Algorithm 2,classifies any unknown phoneme.

B. PNN and Neural Networks

PNN as statistical classifier is applied to determine the initial topology of the system. PNN is also used to recognize silence at low level. The Probabilistic Neural Network (PNN) algorithm represents the likelihood function of a given class as the sum of identical, isotropic Gaussians. In practice, PNN is often an excellent pattern classifier, outperforming other classifiers including backpropagation. However, it is not robust with respect to affine transformations of feature space, and this can lead to poor performance on certain data. We have derived an extension of PNN by allowing anisotropic Gaussians, i.e. Gaussians whose covariance is not a multiple of the identity matrix. PNN [16] is a pattern classification algorithm which falls into the broad class of nearest-neighborlike algorithms. It is called a neural network because of its natural mapping onto a two-layer feedforward network.

According to maximum a posteriori (MAP) probability, an unknown pattern *X* is classified as class *Ci,* if

$P(X|C_i)P(C_i) \geq P(X|C_j)P(C_j)$ □j =i
(5)

where, *P*(*X/Ci*) denotes a probability distribution function of class *Ci* and *P*(*Ci*) is a prior probability of class *Ci*. MAP provides a method for optimal classification. It clarifies how to classify a new sample with the maximum probability of success given enough prior knowledge. As can be seen from (5), to classify an unknown pattern *X*, probability distribution function (*pdf*) of all classes should be known. Having enough training data, one is able to estimate such a *pdf*.

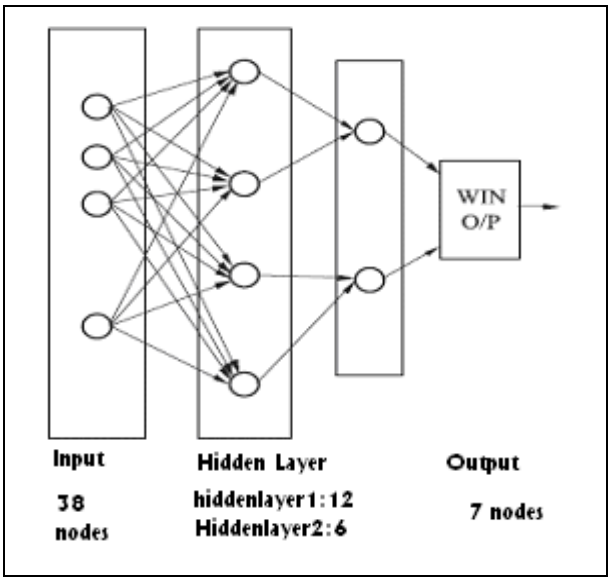


Fig. 3.The architecture of PNN

PNN employs Parzen window density estimation with Gaussian windowing function as a *pdf* estimator given by:

$$P(X|C^c) = \frac{1}{(2\pi)^{\frac{q}{2}}\sigma^q}\frac{1}{n_{C^c}}\left(\sum_{i=1}^{n_{C^c}} exp\left(\frac{-(X-Y_{C^c}^i)^T((X-Y_{C^c}^i))}{2\sigma^2}\right)\right)$$

(6)

where σ is a smoothing parameter which represents smoothness degree of the probability distribution function and *q* shows the dimension of the feature space. $Y_{C^c}^i$ denotes training data i of class *Cc*. Also, *nCc* denotes the number of training data in class *Cc* and *T* is the vector transpose. The probabilistic neural network provides a four-layer network to map an unknown pattern *X* to any number of classes based on (5)[3]. Exemplar is meant for identification of phonemes and class is for classification of phonemes from exemplar.

C. RNN and Neural Networks

MLP and RNNare used as local experts at low level classification in the present modular system[1]. The different phoneme families are considered at low level classification and it identifies the individual phoneme from the identified phoneme families. Each family contains a set of phonemes which are similar in terms of speech features.

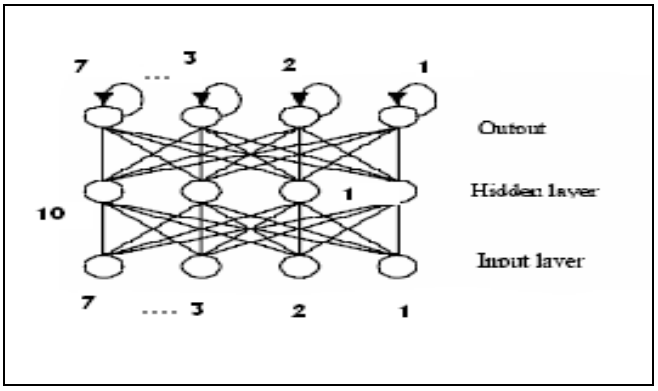


Fig. 4 RNN Architecture

In other words, the phonemes which are very *close* to each other in terms of Euclidean distance are grouped in the same family. Therefore, there is a distributed structure in which each module or family is responsible to learn some specific part of the problem and give its response during the recognition phase. To design such expert modules capable of recognizing any upcoming phoneme pattern, we need more powerful discriminators in low level classification. Multi-layer perceptron is suited for the recognition of phonemes with time invariant input parameters . Also, RNN can learn the temporal relationships of speech data and is capable of modeling time-dependent phonemes . Since both MLP and RNN are trained on other classes' data they are able to discriminate between similar classes. The structure of the used MLP and RNN are very similar, except that RNN has feedbacks on its output layer. The input of the networks is a window of frame level features. Both MLP and RNN have as many outputs as the number of phonemes, *N*.

We can denote the output by

$$O = (o_1, o_2, ..., o_{N-1}, o_N)$$

(7)

where, for a given input *x* belonging to phoneme *k*.

we use this architecture for speech recognition especially for speech recognition by using Backpropagation Through Time (BPTT) as learning algorithm. This architecture also has been proved that this architecture better than MLP in phoneme recognition accuracies by using Backpropagation algorithm. The Backpropagation Through Time (BPTT) algorithm is based on converting the network from a feedback system to purely feedforward system by folding the network

over time. The network can then be trained if it is one large feedforward network with the modified weights being treated as shared weight. The weights are updated either after iteration or after the final iteration of the epoch.

IV. SYSTEM TOPOLOGY'S PARAMETERS

The initial topology of the proposed system is determined using probabilistic neural network. In this regard,

a number of parameters including smoothing parameter (σ), phoneme families number (k) and window size of frame-level data (w) are clarified.

Algorithm 1 Learning algorithm

1: Provide a set of training data for each phoneme

2: Find sample mean of training data belonging to each phoneme C i and denote it as SMC i .

3: for k = 2 to K do

4:Obtain phoneme families

$$CL^k = \{CL_1^k, CL_2^k, ..., CL_k^k\} \quad (8)$$

using (k-mean clustering) on sample mean data.

5:Find the best value of smoothing parameter which leads to minimum error rate at high level classification. Denote this value $\sigma^{k,1}$.

6:Find the best value of smoothing parameter which leads to minimum error rate at low level classification. Denote this value $\sigma^{k,2}$.

7:Obtain the overall error rate of the system, E_k, considering $\sigma^{k,1}$.and $\sigma^{k,2}$ as the smoothing parameters of the high and low level classifiers, respectively.

8: end for k9: k □ ← arg min{E }, where k is the number of the smoothing parameters used.

Algorithm 2 Classification algorithm

1: Provide a testing pattern *X* to the network.

2: Compute the output of the high level classification for a testing pattern .

3: Provide testing pattern *X* to the selected low level classifier.

4: Obtain the output of corresponding low level classifier for testing pattern .

We have examined the accuracy of the system in terms of classification rate considering different values for the smoothing parameter at both high and low level classifications. The value which leads to minimum classification error is for small σ near zero, PNN behaves like nearest neighbor classifier.

V. IMPLEMENTATION

The speaker independent phoneme recognition is implemented by training the system each 50 samples from different speakers consisting of 1000 words each. A test samples taken from a different set of 30 speakers each uttering 500 words. All the samples were of Indian English and taken from TIMIT database. The recognition accuracy is found to be 98% which is well above the previous results.

VI. CONCLUSION

Speech recognition has a big potential in becoming an important factor of interaction between human and computer in the near future. A system has been proposed to combine the advantages of ANN's and HMM's for further speaker independent speech recognition in future. Encouraged by the results of the above described experiment, which indicate that global optimization of ANN system gives some significant performance benefits. We have seen how such a system could integrate multiple ANN modules, which may be recurrent. A Neural Network with trained classifies 98% of the phonemes correctly. A further refined phoneme recognition system can improve the accuracy to near 100%.

REFERENCES:

[1] Noise-robust automatic speech recognition using a discriminative echo state network Mark D. Skowronski and John G. Harris Computational NeuroEngineering Lab Electrical and Computer Engineering University of Florida, Gainesville,FL 32611 Email: {markskow,harris}@cnel.ufl.edu 1-4244-0921-7/07 $25.00 © 2007 IEEE

[2] Medser L. R. and Jain L. C., "Recurrent Neural Network: Design and Applications." London, New York: CRC Press LLC,2001.

[2] D.A.Reynolds,"An Overview of Automatic Speaker Recognition Technology", Proc. ICASSP 2002, Orlando, Florida, pp. 300-304.

[3] Philip D. Wasserman, "Advanced methods in neural computing", Von Nostrand Renhold,1993

[4] D.F. Specht, "Probabilistic neural networks",Neural Networks, vol.3,pp. 109-118,1990.

[5] R.O.Duda, P.E. Hart, and D.G.Stork, "pattern classification", John Wiley and sons,second edition,2001.

[6] L.Deng and D.O'Shaughnessy, "Speech processing: a dyanamic and optimization-oriented approach", Marcel Dekker Inc.,2003

[7] R.Kronland-Martinet, J.Morlet and A.Grossman, "Analysis of sound patterns through wavelet transformation", International Journal of Pattern Recognition and Artificial Intelligence, Vol.1(2)

[8] John Coleman, "Introducing speech and language processing", Cambridge university press,2005

[9] L.Mesbahi and A.Benyetto,"Continuous speech recognition by adaptive temporal radial basis function," in IEEE international conference on systems,Man and Cybernetics,2004,pp.574-579

[10] H. Sakoe and S. Chiba, "Dynamic programming optimization for spokenword recognition", Proceedings of ICASSP-78, vol. 26, no. 1, pp. 43-49, 1997.

[11] "Timit acoustic-phonetic continuous speech corpus", National Institute of Standards and Technology speech Disc 1-1.1, October 1990.

[12] Bourlard, H., Kamp, Y., Ney, H., and Wellekens, C. J. "Speaker-Dependent Connected Speech Recognition Via Dynamic Programming and Statistical Methods," in *Speech and Speaker Recognition*, ed. M. R. Schroeder, Basel, Switzerland: Karger, pp. 115-148,1985.

[13] F. Jelinek ,"Statistical Methods for Speech Recognition", The MIT Press, 1998.

[14] L. R. Rabiner, B. H. Juang, "Fundamentals of Speech Recognition", Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[15] K. Fukunaga. "Statistical Pattern Recognition"' Academic Press.1990

[16] D.F. Specht. (1990) Probabilistic neural networks. Neural Networks, vol. 3, no. 1, pp. 109{118.