www.ijiarec.com

ISSN:2348-2079

Volume-5 Issue-1

International Journal of Intellectual Advancements and Research in Engineering Computations

An enhanced hybrid DBN/HMM for Tamil language speech recognition system

Sundarapandiyan S[#] Research Scholar, Dr.Shanthi N[*] Professor,

Department of Computer Science and Engineering, Nandha Engineering College, Erode, Tamilnadu, India

[#]s_sundarapandiyan@yahoo.com [*]shanthimoorthi@yahoo.com

Abstract **- As the use of Recognition system for language has increased, so has the means and the incentive to create methods to formulate designs that incorporate Automatic speech recognition system for various languages that are widely used for communication. Accordingly, there is a great need in building a system that would help in understanding the language Tamil and a design that can automatically recognize the language. This paper presents a systematic new technique known as Hybrid DBN/HMM to improve the accuracy of automatic speech recognition system for the language Tamil. In this method, we use DBN as the posterior probability estimator. DBN is recently proved to be an effective classification technique for different machine learning problems. We apply the Hybrid DBN/HMM Technique on Tamil language speech corpus provided by LDC-IL and the results are compared with other techniques such as GMM/HMM and MLP/HMM. The Hybrid DBN/HMM appears to pay off better results than the other Techniques.**

Key words---Deep Belief Network, Hidden Markov Model, Tamil Language, Speech Recognition.

I.INTRODUCTION

One of the primary goals of natural language processing is the speech recognition. Automatic Speech Recognition (ASR) system transcribes the input speech into a composed of four modules which are Feature Extraction, Acoustic model, Language model and Decoding.

In the Feature Extraction phase, a series of acoustic features like Mel Frequency Cepstral Coefficients (MFCC), Linear Predictive Cepstral Coefficients (LPCC) Perceptual Linear Perceptron Coefficients (PLP) etc. are extracted from the speech signal. An acoustic model provides a mapping between a unit of speech such as phonemes and syllable and series of an acoustic feature provided by Feature Extraction. An acoustic model estimates the probability of a word given an acoustic feature, i.e., posterior probability. This posterior probability converted into the emission probability that is given in section 4.4.An acoustic model is generated using various approaches such as Artificial Neural Networks [1], Gaussian Markov Model(GMM) [2], Hidden Markov Model (HMM) [3][4] and hybrid methods (i.e. combination of two or more approaches).

Language modeling is estimating the probability distribution of speech units such as words and sentence, i.e., prior probability. Most widely used Language modeling techniques are n-gram language model and neural network based language model. The language model is expressed as (1)

$$P(w_{1,}w_{2,\ldots\ldots,}w_m) = \prod_{i=1}^{m} P(w_i|w_{1,}w_{2,\ldots\ldots,}w_{i-1})P(w_i|w_{i-(n-1)},\ldots,w_{i-1}) \quad (1)$$

The decoder uses the acoustic and the language models to generate the word sequence W=w1,w2,...,w_m with the maximum posterior probability given the input feature vectors X=x1,x2,...,x_n.

$$P(W|X) = \frac{P(X|W)P(W)}{P(X)} \quad (2)$$

In (2) $P(W|X)$-Maximum Posterior Probability $P(X|W)P(W)$-Emission probability estimated from Acoustic model, P(X) - Prior probability calculated from language model.

In recent days Speech recognition is integrated with numerous real-world applications. Speech Recognition engine is used in any mobile application, for example, voice-control of the device, accessing menus, voice dialers, access to content, navigation - address search and entry by voice. Speech Recognition in computer empowers users to interact with their computers by voice. It was designed for people who want significantly to limit their use of the mouse and keyboard while maintaining or increasing their overall productivity. You can dictate documents and emails in mainstream applications use voice commands to start and switch between applications, control the operating system, and even fill out forms on the Web.

II. RELATED WORK

Many Techniques are implemented to improve the performance of speech recognition system. The HMM/MLP, hybrid approach speech recognition system, uses MLP to estimate phone class probabilities, that are utilized in an observation probabilities for a phone- based HMM [5]. Within the hybrid approach, context-dependent phonetic modeling with MLPs has been proposed by Bourlard [6]. Then the speech technology migrates to the GMM/HMM concepts, in which all Hidden Markov Model states share the same Gaussian Mixture Model structure with the same number of Gaussian in each state [7].Deep Belief Networks are MLP with many Hidden layers. Recently, Deep Belief Networks (DBNs) are introduced as a newly powerful machine learning technique. In a Hybrid HMM/DBN, automatic speech recognition system DBNs are used as an HMM posterior estimator for phone recognition system [8].

III. PROBLEM DEFINITION

In general Standard statistical recognizer requires a strong assumption about the statistical character of the input. This type of assumption is not necessary for DBN estimator. DBNs are discriminative rather than the generative model in GMM. Weights of MLPs are initialized randomly, but DBNs use a greedy layer-by-layer pre- training algorithm to initialize the network weights. DBNs are capable of modeling the types of variability present in the speech recognition process. DBN's are unsupervised learning algorithm, but MLP's are supervised learning algorithm [9][10][11]. This paper describes the concepts behind "Hybrid DBN/HMM in Tamil Language word recognition" and overcomes the problems in Hybrid MLP/HMM and G M M /HMM.

IV. HYBRID DBN/HMM RECOGNITION SYSTEM

A. Deep Belief Network

A DBN is made by stacking several bipartite undirected graphical models called the restricted Boltzmann machines (RBMs) in Figure 2. An RBM is a bipartite graph with two layer architecture they are visible stochastic units xi are connected to hidden stochastic units hj in which binary stochastic visible units that represent observations are compared to binary stochastic hidden units using undirected weighted connections. DBNs are the probabilistic generative model [9]. They are restricted to represent that there are no visible-visible or hidden-hidden connections shown in Figure 1. Typically all visible units are connected to all hidden units. The units need not be Bernoulli's variable and can, in fact, have distribution in the exponential family [12].

The Weights on the connections and the biases of the individual units define a probability distribution over a binary state vector v, h of the visible units and hidden units. The joint configuration (v,h) has an energy and is given by

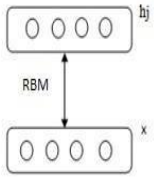

Figure 1. RBM

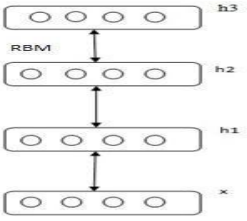

Figure 2. DBN

$$E(v,h|\Theta) = -\sum_{i=1}^{v} a_i v_i - \sum_{j=1}^{H} b_j h_j - \sum_{i=1}^{v}\sum_{j=1}^{H} v_i h_j w_{ij} \quad (3)$$

In (3) Θ is model parameter of the Deep Belief Network it contains weight between visible and hidden layers biases of the visible and hidden layers.

Where $\Theta - \{w, b, a\}$

w_{ij} -Weight between Visible unit i and hidden unit j

a_i, b_j -Bias Terms for visible and hidden units

The network assigns a probability to every possible pair of visible and hidden vector via energy function

$$P(v.h|\Theta) = \frac{1^{-E(v,h|\Theta)}}{Z(\Theta)} \quad (4)$$

The probability that the network assigns to a visible vector v is given by summing over all

possible hidden vector in (4)

$$P(v) = \frac{1}{Z(\Theta)} \sum_H e^{-E(v,h|\Theta)} \quad (5)$$

In (5) $z(\Theta)$ - Normalizing constant

$$Z(\Theta) = \sum_v \sum_H e^{-E(v,h|\Theta)} \quad (6)$$

Since there are no hidden-hidden or visible-visible connections, the conditional distributions $P(h|v)$ and $P(v|h)$ are given by

$$P(h_{j=1}|v;\ \Theta) = \sigma(\sum_{i=1}^{v} w_{ij}\, v_i + \alpha_i) \quad (7)$$

$$P(v_{i=1}|h;\ \Theta) = \sigma(\sum_{j=1}^{H} w_{ij}\, h_j + b_i) \quad (8)$$

$$\sigma(x) = \frac{1}{1+\exp(-z)} \quad (9)$$

In fact the visible and hidden units Gaussian and Bernoulli respectively, the energy function is defined as

$$E(v, h|\Theta) = \frac{1}{2}\sum_{j=1}^{v}(v_{i-}\, b_i)^2 - \sum_{i=1}^{H} a_j h_j \sum_{i=1}^{v}\sum_{j=1}^{H} v_i h_j w_{ij} \quad (10)$$

And the conditional probabilities

$$P(h_{j=1}|v;\Theta) = \sigma(\sum_{i=1}^{v} w_{ij} v_i + a_i) \quad (11)$$

$$P(v_j|h;\Theta) = \sigma(\sum_{j=1}^{H} w_{ij} h_j + b_i, 1) \quad (12)$$

The aim of learning in an RBM is to maximize the marginal probability of the data $P(v|\Theta)$.It can be done very efficiently called "Contrastive Divergence"[13].

B. System Architecture

Figure 3 shows the architecture of hybrid DBN/HMM speech recognition system. In the hybrid DBN/HMM, speech recognition system DBN is used as HMM state posterior estimator. Feature files were computed from the audio training data, and it is necessary for training the Deep Neural Network. Each recording is transformed into a sequence of feature vectors consisting of Mel-Frequency Cepstral Coefficients (MFCCs). The input to the DBN is feature vector x1,x2...xt. DBN was trained by greedy layer-by-layer technique. The output layer of the DBN uses the softmax activation function to produce the posterior probability. Baye's theorem converts this posterior probability to emission probability explained in this section. Viterbi decoding uses this Emission probability for finding the word sequences.

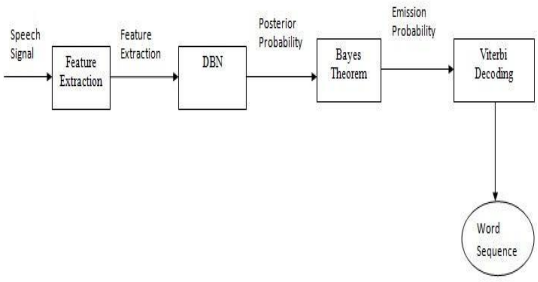

Figure 3: Architecture of Hybrid DBN/HMM Tamil speech recognition system

C. DBN TrainingAlgorithm

Training an RBM is performed by the algorithm known as "Contrastive Divergence Learning"[13][14] shown in Figure 4. Let W be the matrix of IxJ (I - the number of visible neurons, J - the number of hidden neurons) that represents weights between neurons. Multiplication of each input forms current neuron state S by weight

$$s_j = F\left(\sum(s_{iij} + N(0,1))\right) \quad (13)$$

Where
s_j -Current neuron state.

s_i -Adding the set of neurons given in the input layer, one bias neuron.

N (0, 1) - Random number from normal distribution with mean 0.0 and variance 1.0.

Algorithm 1: Reconstruction of Deep Belief Neural Network

Step 1: Get Training data point from the dataset.

Step 2: Use values of this data point to set the state of visible neurons s_i.

Step 3: Compute s_j for each hidden neuron by using (15) and states of visible neurons s_i

Step 4: Now s_i and s_j values can be used to calculate $(s_i . s_j)$ 0 $. s_i . s_j$ is obtained by multiplication of current activation state of neuron i and neuron j. Where s_i is the state of a visible neuron, and s_j is the state of a hidden neuron.

Step 5: On visible neurons compute s_i using the s_j computed in step 3, this is known as "reconstruction." Without using Normal distribution.

Step 6: Compute state of hidden neurons s_j again using s_i from 5 step.

Step 7: Now use s_i and sj to compute $(s_i . s_j)$.

Step 8: Repeating steps 5,6 and 7 multiple times we calculate $(s_i . s_j)$*n.

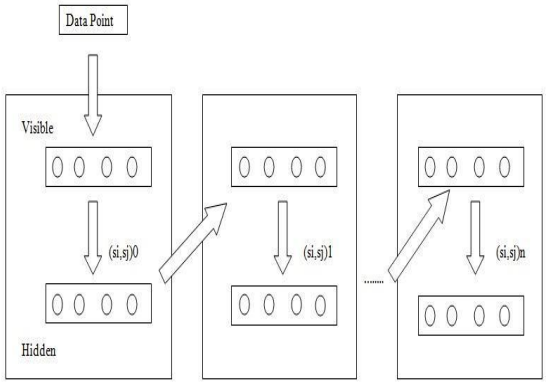

Figure 4: Contrastive Divergence Learning for Deep Belief Neural Network

Algorithm 2: Learning of Deep Belief Neural Network

For each iteration and data point in data set

Step1: Initialize the matrix CDpos =0, CDneg=0. Perform steps 1...8 in reconstruction algorithm.

Step2: Compute $CDpos=CDpos + (s_i . s_j)0, CDneg = CDneg + (s_i . s_j)n$.

Step3: Estimate the value of Contrastive Divergence.
$CD = < s_i . s_j > 0 - < s_i . s_j > n = CDpos - Cdneg$

Step 4: Update weights and Bias $W = W + \alpha \times CD$ (Biases are just weights to neurons that stay always 1.0)

Step 5: Compute"error function" between data and reconstruction.

Repeat for the next epoch until the error is small or some fixed number of the epochs.

Where
Indexes after <... > mean that average is taken after 0 or n[th] reconstruction step performed.
CD - Contrastive Divergence Reconstruction Variable.
α- Learning rate.

D. Estimating HMM Emission Probability with DBN

From the training of DBN, the posterior probabilities are determined. Viterbi decoding algorithm uses the emission probability so emission probability can be computed by applying posterior probability to the Baye's rule [15].

$$\frac{P(X_n|w_k)}{P(X_n)} = \frac{P(w_k|X_n)}{P(w_k)} \quad (14)$$

Where
$P(w_k|X_n)$-Posterior probability.
$P(w_k$ - Prior probability.

That is dividing the posterior estimates from DBN outputs by class prior probability. The scaling factor p(xn) is a constant for all classes will not change the classification.

E. Viterbi Decoding

To perform the Viterbi decoding, the emission probability and transition probability are necessary. Estimation of calculating the emission probability bj(k) is given in section 4.4.

$b_j(k)$-P(Observation k at t | State q_i at t)
Set the transition probability a_{ij} which possesses the self-loop likelihood.

a_{ij}-P(State q_i at t+1 | state q_i at t)
Apply this emission probability and the transition probability to the Viterbi decoding for the given acoustic features to find the maximum probability Viterbi sequence.

F. Viterbi Algorithm

To find single best state sequence, $W = w1, w2, w3, ..., wm$ for a given observation sequence $X = x1, x2, x3, ..., xn$.

$$\delta_t(i) = \max_{w_1 w_2 w_3 \ldots w_m} P(w_1 w_2 w_3 \ldots w_m = i, x_1\, x_2\, x_3 \ldots . x_n|\boldsymbol{\lambda}) \quad (15)$$

Where
$\delta t\,(i)$ -Best score along a single path, at time t.

$$\delta_{t+1}(j) = \max(\delta_t\,(i) a_{ij} b_j(x_t + 1) \quad (16)$$

Where
λ- HMM model parameter which contains Transition probability, Emission probability,

and Initial State Distribution.
To find the state sequence we need to keep track of state which maximizes the above (20). This Viterbi sequence considered as recognized word.

V. EXPERIMENTAL SETUP

A. Speech Corpus

The LDC-IL Tamil Speech corpus is used in our experiment.80% of the data used for training and remaining 20% of the data used for testing. The parameter of the LDC-IL speech corpus is, sample rate 16 kHz and 16 bit, wave format-mono.

B. Feature Extraction

The feature used in this experiment is 12th order Mel Frequency Cepstral Coefficients (MFCC) and an energy Coefficient along with their first and second temporal derivatives [16]. Here frame length is 25ms, and a frame shift is 10ms for extracting the feature. After the feature is extracted from the training data we need to convert into matrix format to provide inputs to the D N N .

C. Neural Network Parameter

RBM weights are initialized randomly using a normal distribution with a mean is 0, and the standard deviation is 0.1. Number of hidden layers are 100,150,200,250 for each experiment. In the training process, we set the learning rate as 0.1 and momentum as 0.5.All RBMs are trained with 50 epochs. A weight decay of 0.0002 is used to penalize the large weights. Parameter selection for DBN explained in [17][18].

D. Viterbi decoding parameter

Here the decoding process uses the Viterbi decoding algorithm. The parameter for the Viterbi decoding algorithm is emission probability and transition probability. Estimation of this emission probability and transition probability is given in section 4.4.

VI. EXPERIMENTAL R E S U L T S

In our research, three experiments can be carried out to show the effectiveness of Hybrid DBN/HMM. All these three experiments are done with LDC-IL speech corpus. The first experiment is done using Hybrid MLP/HMM; the word error rate is obtained as 19%.The second experiment uses GMM/HMM technology compare to MLP/HMM it reduces the word error rate from 3 to 5 %. The third test is based on Hybrid DBN/HMM and its word error rate only. The table shows the comparison of three experiments.

TABLE 1
COMPARISON OF DBN/HMM WITH VARIOUS ACOUSTIC MODEL TECHNIQUES

Techniques	Speech Corpus	WER
MLP/HMM	LDC-IL corpus	15.4
GMM/HMM	LDC-IL corpus	27.4
DBN/HMM	LDC-IL corpus	10.1

Then the next experiment is based on the various number of hidden layer in DBN/HMM. The number of hidden layers used in DBN/HMM is 100,150,200,250 respectively. Table 2 shows the WER for the different number of hidden layer size.

TABLE 2
COMPARISON OF DBN/HMM WITH VARIOUS NUMBER OF HIDDEN LAYER SIZE

Number Of Hidden Layers	WER
100	15.4
150	14.6
200	14.0
250	13.8

VII.CONCLUSIONS

To conclude, a Tamil speech recognition system is designed to investigate the process of automatic speech recognition. In this paper, we describe a new acoustical model approach Hybrid DBN/HMM for Tamil speech recognition. Several experiments are carried out in order to show the effectiveness of the hybrid approach. Compared to GMM/HMM and MLP/HMM, DBN/HMM pay off high accuracy.

ACKNOWLEDGMENT

We wish to express our gratitude to the Linguistic Data Consortium – Indian Languages for providing the standard LDC-IL text Tamil corpus with transcription.

REFERENCES

[1] Deuth H, Beale M, Neural Network Toolbox. For Use with MATLAB, MathWork Inc, 2005.

[2] Reynolds, D.A,Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Mod-els, IEEE Transactions On Speech Audio Processing 1995.

[3] Lawrence R. Rabiner,A Tutorial on Hidden Markov Model and Selected Applicaiton in Speech Recog- nition, IEEE, 1989.

[4] Steve Young, Gunnar Evermannl, The HTK Book Microsoft Corporation, 2005.

[5] H.Bourlard and N.Morgan, "Merging Multilayer Perceptrons and Hidden Markov Models , Some experiment in Continuous Speech Recognition,Neural Network: Advances and Apllications,E.Gelenbe,ed 1990.

[6] R.M Schwartz ,Y.L Chow ,O.A Kimball ,S.Roucos ,M.Krasner , and J.Makhoul ,"Context - Dependent Modeling for Acoustic -phonetic Recognition of Continuous Speech ", ICASSP 85,pp.1205-1208,1985.

[7] Daniel Poveya , Luk ´Burgetb , Mohit Agarwalc , Pinar Akyazid , Feng Kaie ,"The Subspace Gaussian Mixture Model – a Structured Model for Speech Recognition", Microsoft Research, Redmond, WA.

[8] Van Hai Do , Xiong Xiao, Eng Siong Chng,"Comparison and Combination of Multilayer Perceptrons and Deep Belief Networks in Hybrid Automatic Speech Recognition Systems " APSIPA ASC 2011 Xi'an.

[9] A. Mohamed, G. Dahl, and G. Hinton, "Deep Belief Networks for phone recognition," in Proc. of NIPS 2009 Workshop on Deep Learning for Speech Recognition and Related Applications, 2009.

[10] A. Mohamed, D. Yu, and L. Deng, "Investigation of full-sequence training of deep belief networks for speech recognition," in Eleventh Annual Conference of the International Speech Communication Association, 2010.

[11] G. Dahl, A. MarcAurelio Ranzato, and G. Hinton, "Phone Recognition with the Mean-Covariance Restricted Boltzmann Machine," Advances in Neural Information Processing Systems, vol. 24, 2010.

[12] M. Welling, M. Rosen-Zvi, and G. E. Hinton, "Exponential family harmoniums with an application to information retrieval," in Proc. NIPS, 2005.

[13] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural computation, vol. 14, no. 8, pp. 1771-1800, 2002.

[14] H.Chen and A.F Murray "Continuous restricted Boltzmann machine with an implementable training algorithm" image signal process June 2003.

[15] H.Bourlard and N.Morgan "connectionist speech recognition hybrid approach" Kluwer Academic press 1994.

[16] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi, '' Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques", JOUR- NAL OF COMPUTING, VOLUME 2, ISSUE 3, MARCH 2010.

[17] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," Tech. Rep., Univ.Toronto, Aug. 2010.

[18]http://www.cs.toronto.edu/hinton/MatlabForSciencePaper.html