5. DEEP BELIEF NETWORKS FOR PHONEME RECOGNITION

Application of deep neural networks for solving various complex problems successfully [84] has motivated its application in speech recognition. The objective of this chapter is to elucidate on the proposed models to build DBNs for phoneme recognition in continuous Tamil speech and show its competence by comparing with state of art models. Two types of DBN, namely Bernoulli – Bernoulli DBN (BBDBN) and Gaussian - Bernoulli DBN (GBDBN) are applied on Tamil continuous speech data to handle the greater variability characteristics of phonemes observed in the previous chapter. The models built so to identify the spoken phonemes are analyzed for their competences.

5.1 TAMIL PHONEME RECOGNITION MODEL USING DEEP BELIEF NETWORKS

The proposed methodology to build Tamil phoneme recognition model is shown in Fig. 5.1. One of the challenges faced while modelling DBNs is formulating an appropriate training strategy to train the network. Greedy method and random method are methods generally used to initialize the parameters of the network.

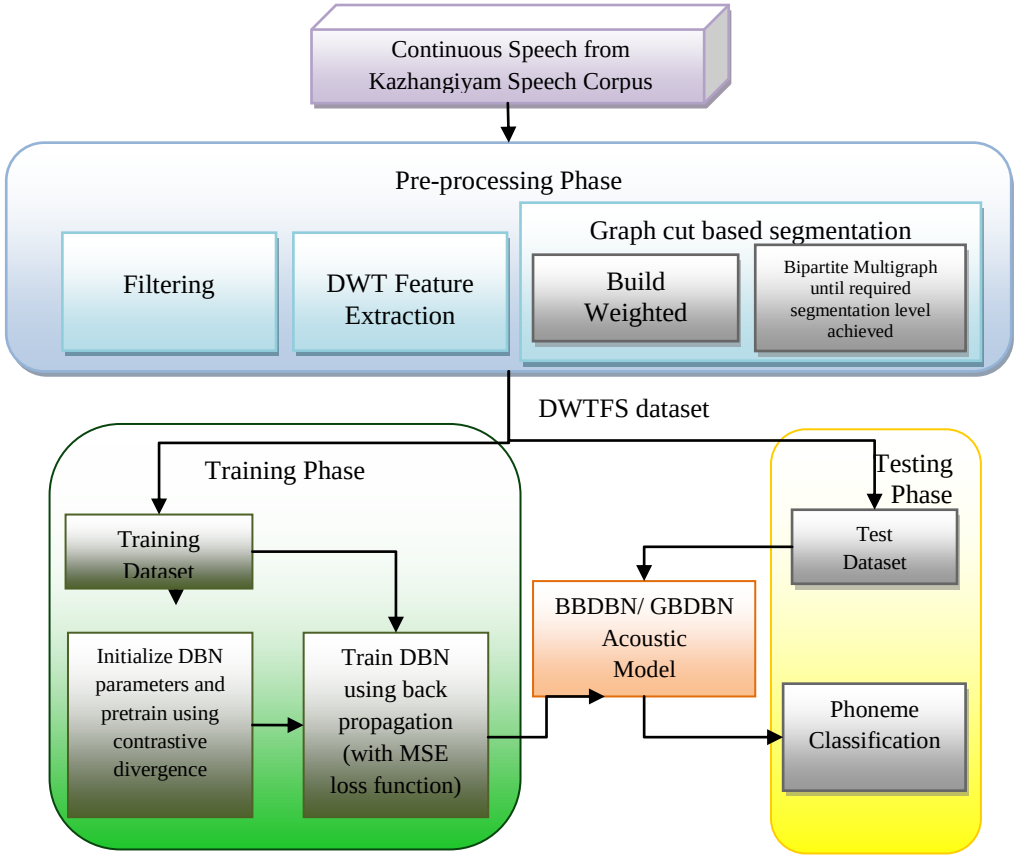

Fig. 5.1 Architecture of BBDBN/GBDBN Based Acoustic Model to Recognize Phonetic Units of Tamil Continuous Speech

The model building process in DBN is a general discriminative training method which considers each pair of layers that is bipartite in nature, as a restricted boltzmann machine. This method initializes the DBN parameters randomly, pre-trains using contrastive divergence and further trains using back propagation technique. The steps to build BBDBN or GBDBN acoustic model is given algorithm 5.1. The continuous Tamil speech is the input to the algorithm which is initially subjected to segmentation process as explained in chapter 3 to build the dataset. The architectural parameters like number of layers or RBMs and number of neurons in each layer of the model under consideration either BBDBN or GBDBN are decided. The model parameters namely bias of individual neurons, connection weights of the links in the DBN are initialized with random number between 0 and 1. The training dataset is used first to pre-train the BBDBN or GBDBN using the contrastive divergence technique as explained below. It thus identifies the initial values for neuron biases and connection weights of the DBNs.

Algorithm 5.1 Steps to build BBDBN/GBDBN based acoustic model

Step 1: Segment the continuous Tamil speech data into phonetic segments using Graphcut based segmentation algorithm.

Step 2: Build the monophone training dataset and test dataset.

Step 3: Decide the DBN architectural parameters number of layers in DBN and number of neurons in each layer and design the BBDBN/GBDBN.

Step 4: Initialize the weight and bias parameters of BBDBN/GBDBN with random values in the range (0, 1).

Step 5: Pre-train the DBN using contrastive divergence.

Step 6: Train DBN with back propagation technique with monophone train dataset.

Step 7: Test the acoustic model build with monophone test dataset.

Pre-Training DBN Using Contrastive Divergence

Contrastive divergence [85] is an efficient training procedure performing an approximate training for RBMs. The procedure repeatedly tries to reconstruct the visible vector from the hidden vector generated from the visible vector thus updating the weight parameters of the RBM. The training dataset is sent as input to the first RBM. Each RBM in DBN is trained one by one, where the output of one RBM turns as the input to the next RBM in the stack. The weight parameters w_{ij} are updated during the training process as follows,

$$\Delta w_{ij} = \langle v_i h_j \rangle_{td} - \langle v_i h_j \rangle_{rd} \quad (5.1)$$

In the above equation, the change in weight parameter Δw_{ij} is calculated. The first term $\langle v_i h_j \rangle_{td}$ refers the measured frequency between the visible units *i* and the hidden units *j* where visible units are initially populated with training data samples and hidden units the posterior probabilities determined using equation 2.12. the second term $\langle v_i h_j \rangle_{rd}$ refers the measured frequency for visible units, the reconstructed data constructed using equation 2.13 when the hidden units were constructed using equation 2.12. Similarly, the posterior probabilities and reconstructed data for visible units are evaluated for pre-training GBDBN using equation 2.14 and 2.15. Training RBM using contrastive divergence is depicted in the Fig. 5.2 and is repeated for all the RBMs of DBN in sequence.

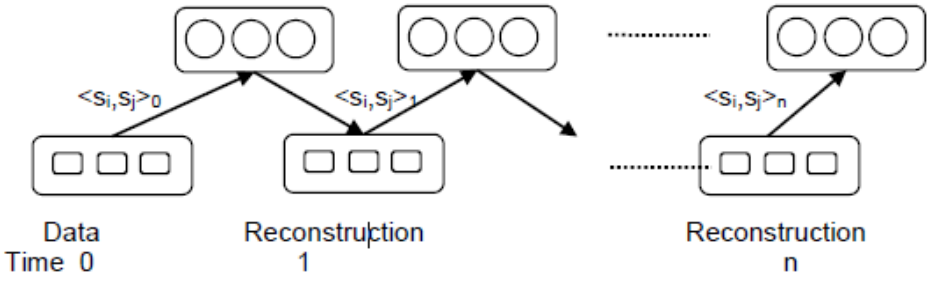

Fig. 5.2 Training RBM using contrastive divergence

Training DBN using Back Propagation

The pre-training phase used to initialize the DBN parameters is followed by the backpropagation training. This phase uses back propagation technique to fine tune the parameter values obtained in the previous phase. The training data is initially passed through the input layer of DBN and forwarded to the output layer as explained in the learning of BBDBN and for GBDBN in chapter 2 respectively. The input data is forwarded to the consecutive layers by evaluating the conditional probabilities of the hidden units defined in terms of the sigmoid activation function of the neurons. At the output layer, the error is calculated and propagated to all the preceding layers. The MSE loss function used to calculate the error at the output layer. The equations involved to backpropagate the error observed at the output layer to fine tune the model parameters are listed below. The error at the output layer is calculated using equation 5.2.

$$\delta_j^{(l)} = y_j^{(l)}(1-y_j^{(l)})(y_j^{(l)} - o_j) \quad (5.2)$$

and the error for hidden layers is calculated by,

$$\delta_j^{(l)} = y_j^{(l)}(1-y_j^{(l)}) \sum_{q=1}^{m} w_{jq}^{(l+1)} \delta_q^{(l+1)} \tag{5.3}$$

The error is back propagated to the layers using their partial derivatives for each layer $l=L-1, L-2,...2$, given by

$$\frac{\partial E}{\partial w_{ij}}(l) = \delta_j^{(l)} y_j^{(l-1)} \tag{5.4}$$

and the network weights are updated using negative gradients given by,

$$\triangle w_{ij}^{(l)} = -\gamma y_i^{(l)} \delta_j^{(l+1)} \tag{5.5}$$

and the bias of each layer is updated using

$$\triangle b_j^{(l)} = -\gamma \delta_j^{(l+1)} \tag{5.6}$$

where $1<l<L$ denotes the hidden and output layers, $y^{(l)}$ denotes the output layer of l^{th} RBM, $w_{ij}^{(l+1)}$, connection weight of i^{th} neuron in $(l)^{th}$ layer to j^{th} neuron in $(l+1)^{th}$ layer, γ, the learning rate.

The trained DBN acoustic model is then tested for its efficiency with the test dataset using various performance metrics.

5.2 EXPERIMENT AND RESULTS

Both DBNs BBDBN and GBDBN, are pre-trained by applying contrastive divergence learning procedure on RBM for 1000 epochs each. The pre-training process trains one RBM at a time in the stack and proceeds to the next one in the stack. Once the pre-training is complete for all the RBMs in the DBN, the whole DBN is trained using back propagation learning is conducted for 1000 epochs with a batch size of 100 data points, step size 0.1, initial momentum 0.5, final momentum 0.9 and weight cost 0.0002.

Experiments are conducted on BBDBN and GBDBN with DWTFS dataset of Kazhangiyam corpus. The RMSE values observed during the contrastive divergence pre-training of each of the six RBMs in the stack of DBN with 5 hidden layers is shown in Fig. 5.3. The RMSE values tend to decrease with the pretraining of deeper RBMs. The RMSEs witnessed during pretraining RBM6 is lesser than RMSE recorded for RBM5, which is lesser than for RBM4, and soon.

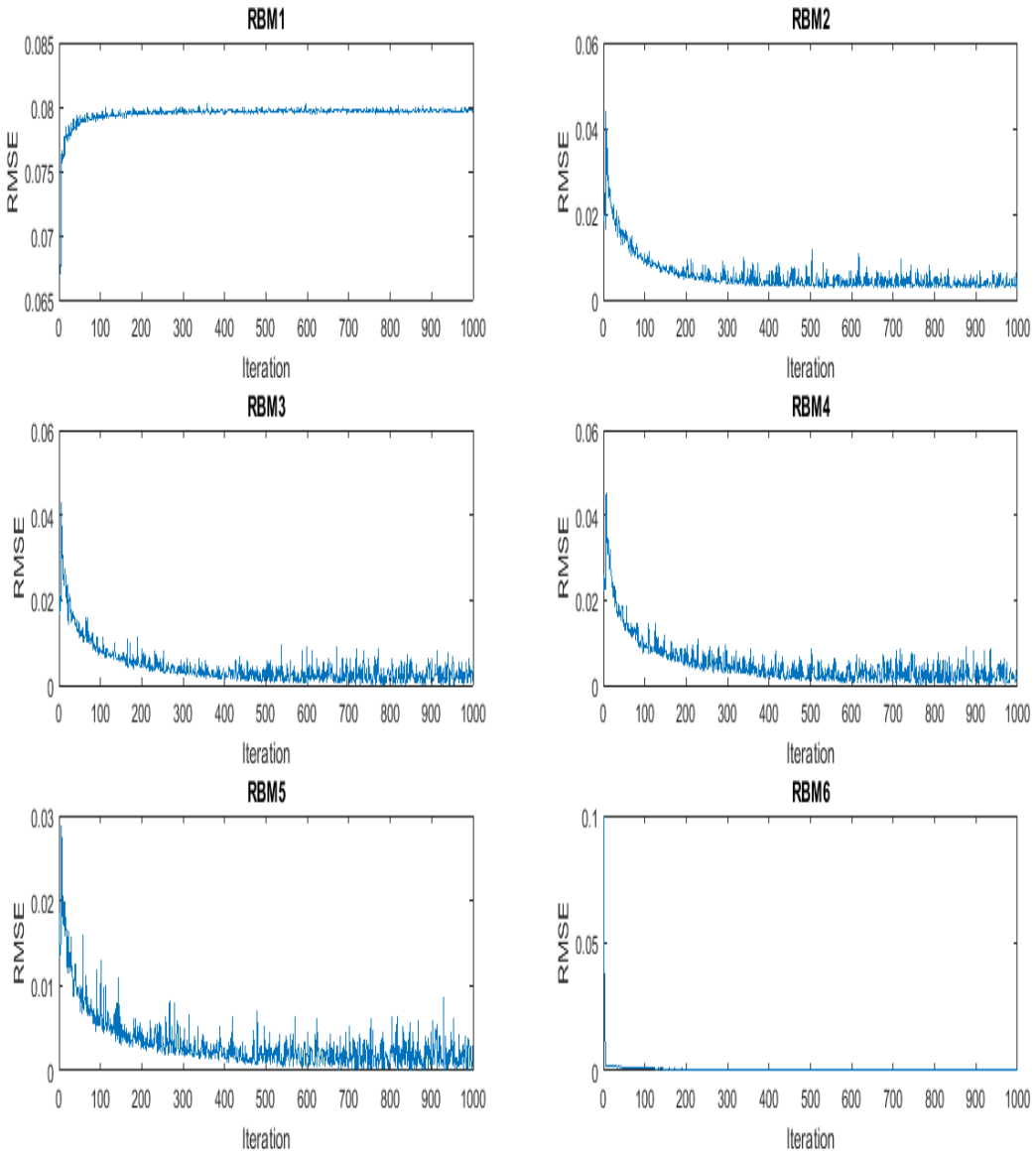

Fig. 5.3 RMSE Observed during Contrastive Divergence Pretraining of RBMs Constituting GBDBN

The efficiency of the models is experimented to analyze the impact of deeper networks. This is done through implementing DBNs by increasing the number of hidden layers from 4 to 6 and evaluating their performances. The architecture of DBNs with 3 hidden layers is 90x100x120x70x39, DBNs with 4 hidden layers is 90x100x120x100x70x39, DBNs with 5 hidden layers is 90x100x120x120x100x70x39 and DBNs with 6 hidden layers is 90x100x100x120x120x100x70x39. The sigmoid function is used as the activation function of neurons. The time taken to build various models in an i7 processor machine is shown in Table XIV. It shows the increase in the number of layers of any DBN increases the time taken to build the model due to the fact that each RBM in the stack is pretrained in isolation and one after the other in their order in the stack.

Table XIV Time Taken (in mins) for Building 3, 4, 5 and 6 – hidden Layer BBDBN and GBDBN Models

Models	3	4	5	6
BBDBN	135	189	234	277
GBDBN	148	178	246	289

The DBNs pretrained with contrastive divergence are then observed during their backpropagation training. The RMSE values recorded in their backpropagation training phase is portrayed in Fig. 5.4. The RMSE values for BBDBN is observed to be much better when compared to the RMSE observed for GBDBN. Screenshots showing the pretraining and training phases of DBN are given in Appendix C.

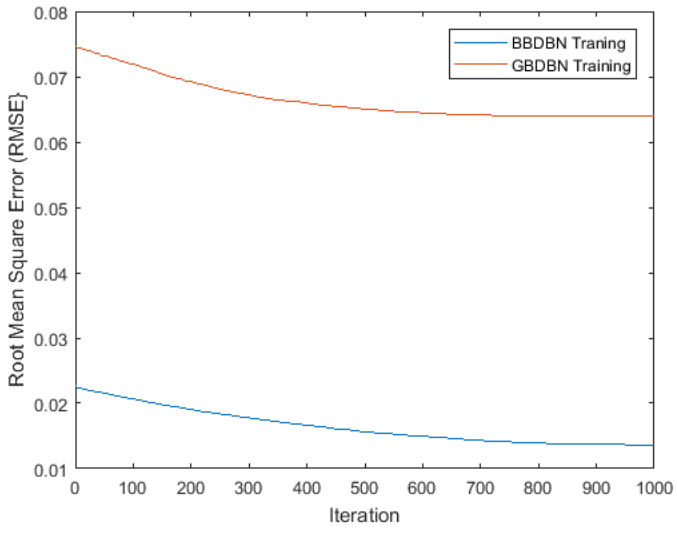

Fig. 5.4 RMSE Observed during BBDBN and GBDBN Training

The results of the experiments in terms of RMSE values are shown in Table XV. The RMSE values while training or testing BBDBN seems to be much lower than training or testing GBDBN for all models with diverse hidden layers. The RMSE values observed while testing BBDBN with 3, 4, 5 and 6 hidden layers are 0.01708, 0.14913, 0.01529 and 0.15281 respectively whereas the RMSE values observed while testing GBDBN with 3, 4, 5 and 6 hidden layers are 0.06933, 0.059303, 0.062027and 0.064215 respectively. The RMSE values of 4 hidden layers architecture performs better when compared to the other architectures for both BBDBN and GBDBN acoustic models.

Table XV Performance Comparison with respect to RMSE Values of BBDBN and GBDBN Models for Different Number of Layers

Model/No. of Layers	**3**	**4**	**5**	**6**
BBDBN-Training	0.01743	0.015124	0.015266	0.015232
GBDBN-Training	0.06921	0.05973	0.061828	0.063849
BBDBN-Testing	0.01708	0.014913	0.01529	0.015281
GBDBN-Testing	0.06933	0.059303	0.062027	0.064215

The Phoneme Error Rate (PER) recorded for the train and test dataset of DWTFS on BBDBN and GBDBN is shown in Table XVI. The best PER observed is 14.62 % for 6-hidden layer BBDBN and 20.53 for 5 and 6 hidden layer GBDBN. It is seen that the phoneme error rate is better in BBDBN with various depth of the network when compare to their equivalents in GBDBN. The observation also shows that increasing the number of layers in the network reduces the error rate by increasing the performance the network classification except very few situations. From Fig. 5.5 it is clear that 6-hidden layer BBDBN model has produced lower PER and outstanding to the other BBDBN models experimented here with similar PER during both training and testing phases. Similarly, when observing the various models of GBDBN, the 5 and 6-hidden layer models of GBDBN is better to 3 and 4-hidden layer GBDBN models with lower PER observed in testing phase that in training phase.

Table XVI PER during Training and Testing 3, 4, 5 and 6-hidden Layer BBDBNs and GBDBNs

Model/No. of Layers	3	4	5	6
BBDBN-Training	19.05	18.42	18.78	14.62
GBDBN-Training	28.51	27.56	22.34	22.34
BBDBN-Testing	20.32	17.28	19.94	14.97
GBDBN-Testing	30.74	29.24	20.53	20.53

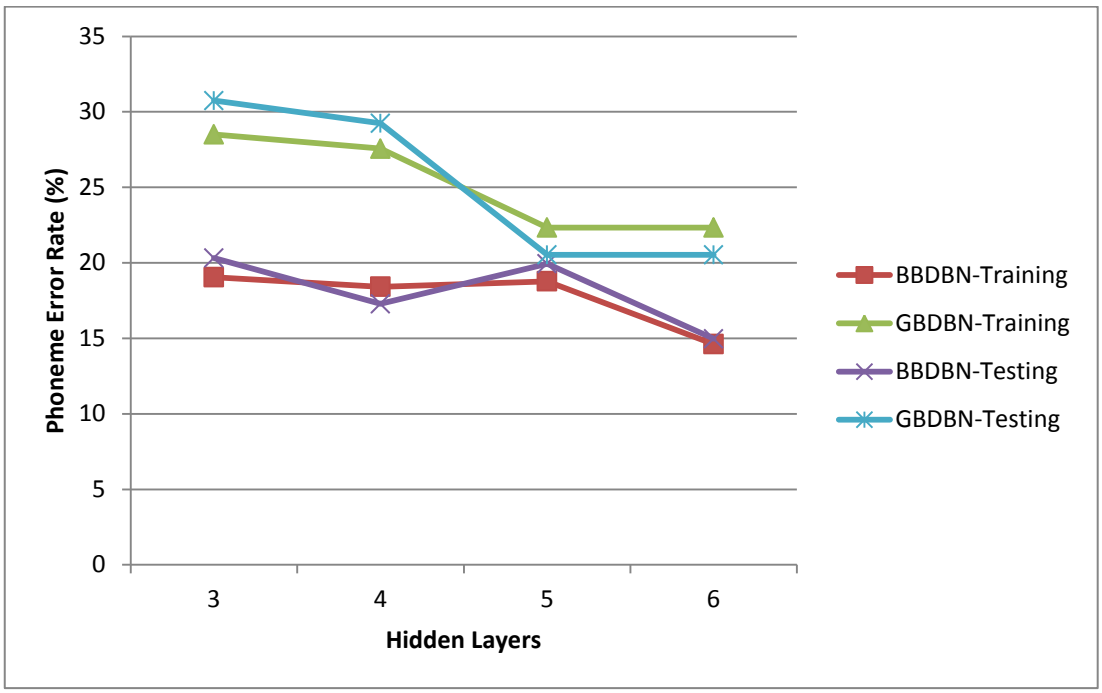

Fig. 5.5 Phoneme Error Rate (PER) During Training and Testing 3, 4, 5 and 6-hidden Layer BBDBNs and GBDBNs

Further evaluations with respect to the precision, recall, F-measure and accuracy of the models experimented is portrayed in Table XVII and Fig. 5.6. The F-measures observed for various BBDBNs are 0.218716, 0.274306, 0.26776 and 0.285486 for 3, 4, 5 and 6-hidden layer BBDBNs respectively. Similarly, the F-measures recorded for various GBDBNs are 0.172116, 0.191223, 0.196193 and 0.201997 for 3, 4, 5 and 6-hidden layer GBDBNs respectively. The 4-layer BBDBN stands out with the highest precision of 0.299081, whereas 6-hidden layer GBDBN stands out in its group with a highest precision of 0.196765.

Table XVII Comparative Performance Results of Various Models

Performance Measure / Model	Precision	Recall	F-Measure	Accuracy(%)
3-hidden layer BBDBN	0.32461	0.31312	0.318763	80.32
4- hidden layer BBDBN	0.39908	0.35332	0.37481	82.15
5- hidden layer BBDBN	0.37449	0.36136	0.367803	80.64
6- hidden layer BBDBN	0.38768	0.38332	0.38549	85.21
3- hidden layer GBDBN	0.27439	0.2699	0.272126	70.38
4- hidden layer GBDBN	0.29563	0.28701	0.291257	71.6
5- hidden layer GBDBN	0.29434	0.29808	0.296199	78.57
6- hidden layer GBDBN	0.29677	0.30751	0.302044	78.57

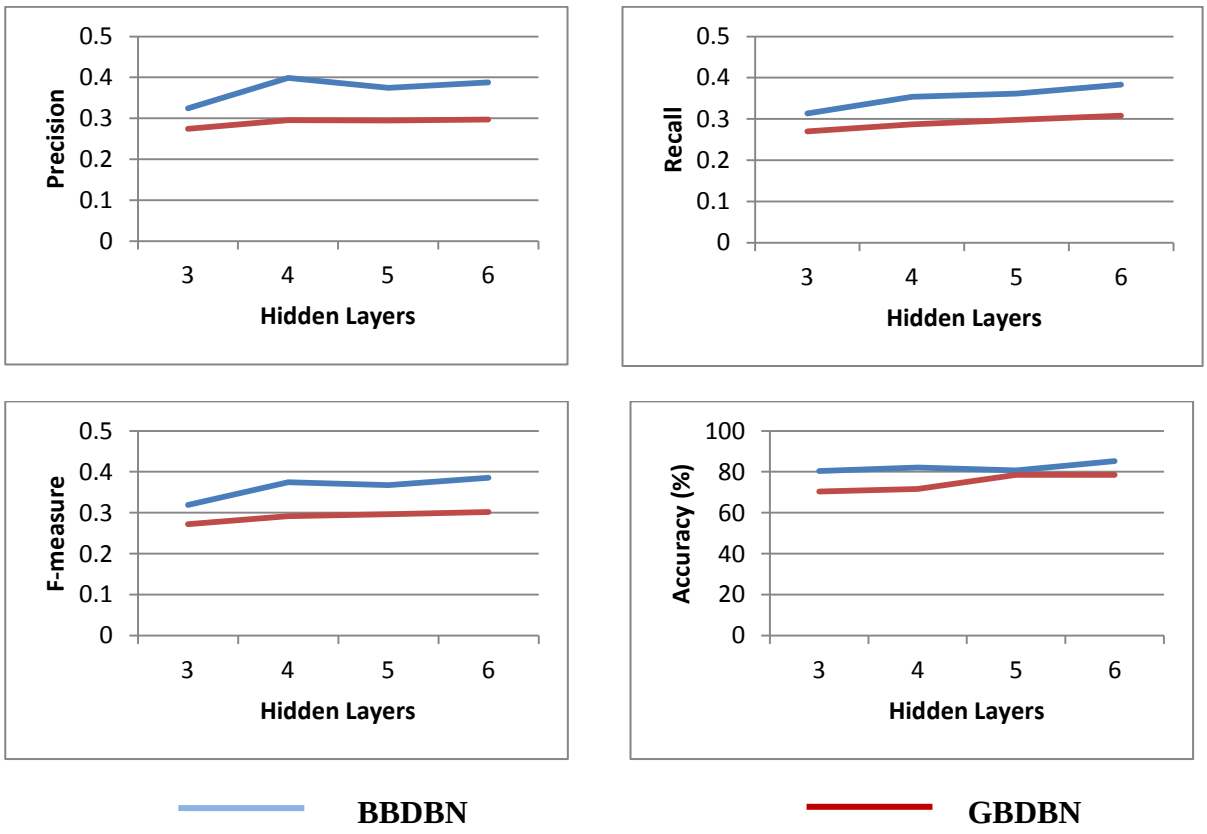

Fig. 5.6 Performance Comparison of Various Models (a) Precision (b) Recall (c) F-measure (d) Accuracy

When analyzing the recall, it is observed to be more for 6-hidden layer architectures of both BBDBN and GBDBN models with values 0.283322 and 0.207514 respectively. The accuracies recorded for 3, 4, 5 and 6 –hidden layer BBDBNs are 80.32%, 82.15%, 80.64% and 85.21% respectively marking greater accuracy for 6-hidden layer model. Similarly, the accuracies observed for GBDBNs of 3, 4, 5 and 6 –hidden layer models are 70.38%, 71.6%, 78.57% and 78.57% respectively marking greater accuracy for 5 and 6 –hidden layer models.

The best result of the current experiments is compared with the results of ANN and ANFIS models of chapter 4. The comparative results with respect to PER is given in Table XVIII to prove the strength of DBN.

Table XVIII Comparative Performance of BBDBN and GBDBN with ANN and ANFIS

Method	PER
Artificial Neural network	35.5%
Adaptive Neuro-Fuzzy Inference System	38%
BBDBN	14.97%
GBDBN	20.53%

Using BBDBN for Phoneme recognition of Tamil speech turned out with 14.97% PER, where as it was 14.97% for GBDBN. Thus BBDBN out performs ANN, ANFIS and GBDBN for the DWTFS dataset and is shown in Fig. 5.7.

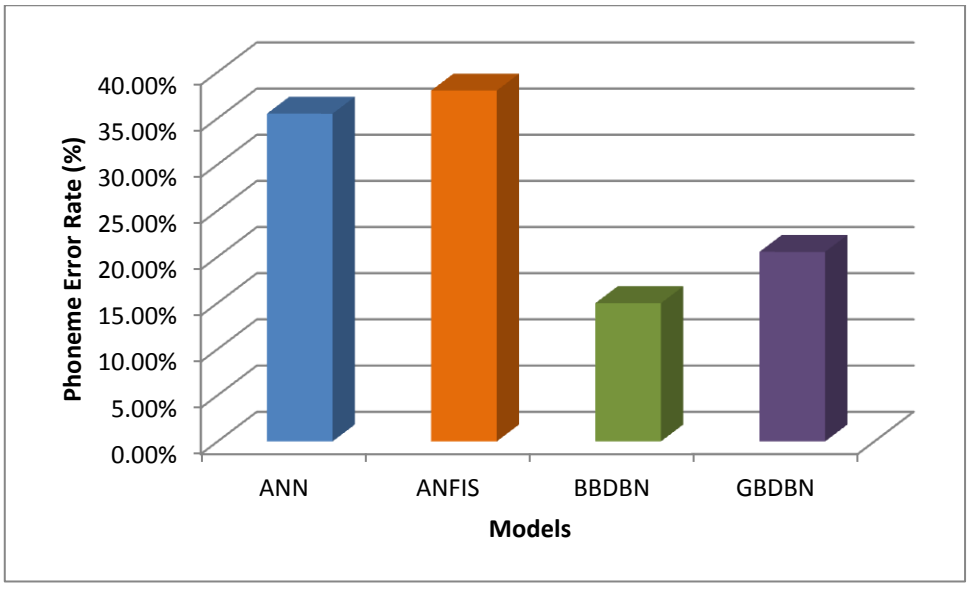

Fig. 5.7 PER Comparison of Various Models

Findings

The notion of using DBN for capturing the greater variability in the speech characteristics of different speakers is met to an extent. This is inferred from the fact that these models give more efficient models than the ones built with ANN or ANFIS providing better phoneme recognition accuracies. It is also seen that deeper networks capture more variability in data to build better models. The results show the efficiency of the network stagnates or degrades when the network depth crosses a threshold depth which is dependent on the problem under consideration. In addition, it is also observed the models built with BBDBN outperform the equivalent models built with GBDBN for this phoneme recognition problem. It is inferred that DBNs are appropriate to build prominent phoneme recognition models. In spite of the efficiency observed in DBN, this model building process suffered from an intolerable increase in time overhead with the increase in the depth of the network. This is evitable due to the use of contrastive divergence for pretraining the network that works on a single RBM at a time which also needs further attention in this research. In addition the model also risks the solution to be trapped in local minima due to the use of gradient descent based backpropagation.

SUMMARY

This chapter discussed the implementation of deep belief networks for the phoneme recognition problem. It elucidated the pretraining phase of DBNs with contrastive divergence and intensive learning through backpropagation. Varied phoneme recognition models developed with 3, 4, 5 and 6-hidden layers BBDBNs and GBDBNs were described. The performances of models built were compared and the comparative analysis is presented with tables and charts. The challenges of time complexity and the risk of solution being trapped to local mimimum are handled by proposing better pretraining techniques which will be elaborated in the forthcoming chapters.

Remarks

- The article titled ‘Deep Belief Networks for Phoneme Recognition in Continuous Tamil Speech- An Analysis’ is published in Traitement Du Signal , ISSN: 0765-0019, Vol.34, No. 3-4, pp. 137-151, 2017. Lavoisier. DOI: 10.3166/ts.34.137-151 (Published- SCIE, Scopus Indexed)