CLASSIFICATION WITH DEEP BELIEF NETWORK FOR LARGE DATASET

*Azhagammal Algarsamy[1],[1,] Ruba Soundar Kathavarayan[2]
[1] Department of Computer Science, Research Scholar , Bharathiyar University,
Tamilnadu, India
[2] Department of Computer Science Engineering, P. S. R Engineering College, Sivakasi-626140, Tamilnadu, India

Abstract: Deep Belief Network is an algorithm among deep learning. It is an effective method of solving the problems from neural network with deep layers, such as low velocity and the overfitting phenomenon in learning. The learning of a DBN starts with pretraining a series of the RBMs followed by ne-tuning the whole net using backpropagation. Generally, the sequential implementation of both RBMs and backpropagation algorithm takes signicant amount of computational time to process massive data sets. Available methods for classifications are support vector machines, quadratic classifiers, k-nearest neighbor, boosting, decision trees, neural network, learning vector quantization. From the above list, we use Neural Network for classification in this paper, Neural network is one of the most effective method for classification which are relatively rude electronic networks of neuron based on the neural structure of the brain. Neural network technique is either supervised or unsupervised. The supervised method contains algorithms like Back Propagation, Feed Forward, and so many. Deep neural network comprises more than one hidden layers which actually improves the accuracy in classification. The Deep belief network(DBN) is an unsupervised learning. It includes more than one trained RBMs. In this paper we implement DBN in large dataset (Readmission rate of Diabetic patients in medical data) and observe the accuracy based on different test cases.

***IndexTerms* - Deep Belief Network, DBN, Classification, large Dataset**

I. INTRODUCTION

Classification is a data mining approach to prognosticate group membership for record instances. The goal of classification is to foretell the target class for each case in the data. Many classification methods are used based on applications. There are two main classification techniques supervised and unsupervised. In supervised classification set of possible class is known in advance. In unsupervised classification technique, the set of possible class is unknown, after classification we can assign name to the class. Some classification algorithms are Logistic Regression, Navie Bayies classifier, SVM, Decision tree, Boosted Trees, Random Forest, Neural Network, Nearest Neighbor. The classification block diagram showed in figure1.

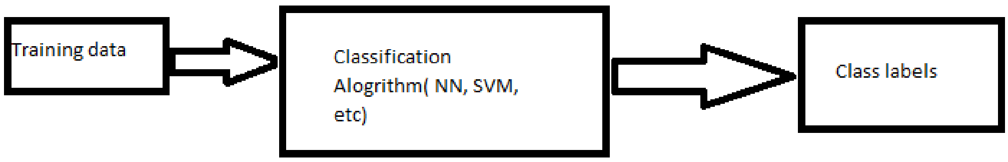


figure 1: Classification Block Diagram

From the above methods we use neural network for classifying larger dataset. Neural network is composed of nodes, arranged in layers, which convert an input vector into some output. Each unit takes an input, applies a function to it and then passes the output on to the next layer. The standard neural network structure contains three layers. First one is input layer which is used to read the user dataset. The second layer is hidden layer, the activation function is implemented. The third layer is output layer which

*Assistant Professor, Department of Computer Science, Ayya Nadar Janaki Ammal College, Sivakasi denoted the class. Each neuron in each layer is connected and weighted. Weightings are applied to the signals passing from one unit to another and during the training phase; the weights are adjusted based on the output. The standard neural network architecture demonstrated in figure 2.

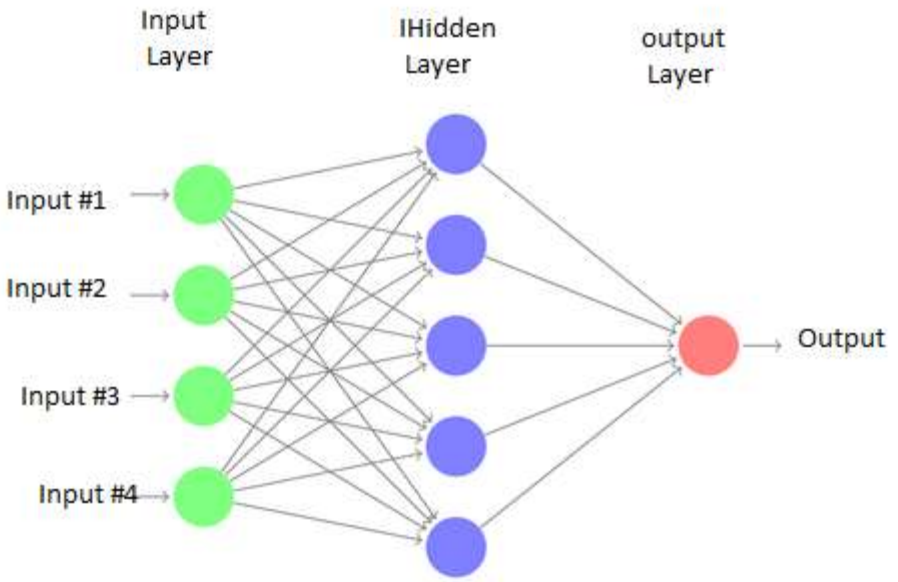


figure 2: Neural Network Architecture

Modification in neural network structure known as deep learning is introduced by many researcher to improve the classification accuracy. Deep learning referred to as " Stacked Neural Network", that is the network is composed of more than one hidden layer, the number of node layers through which data passes in a multistep process of pattern recognition. Deep learning networks capable of handling very large, high dimensional datasets with billions of parameters that pass through non-linear functions. The deep learning architecture is presented in figure 3.

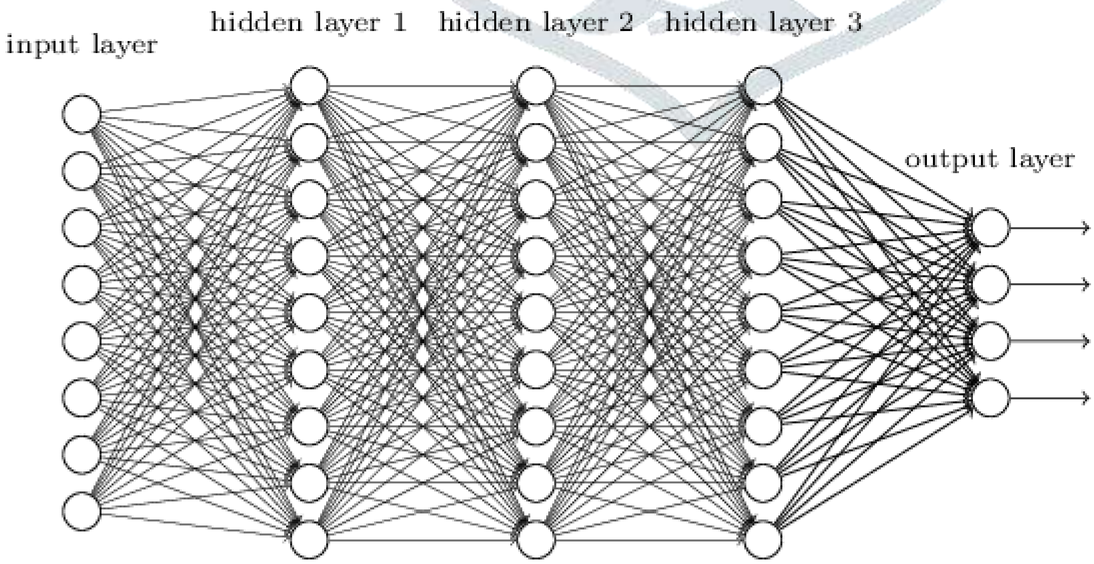
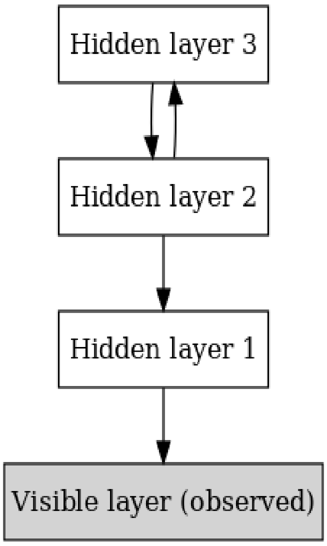


figure 3: Deep Learning Architecture

One of the most common deep learning architecture is Deep Belief Network (DBN). DBN is a generative graphical model, composed of multiple hidden units. When DBN trained on a set of training sets with output class labels, it can learn to probabilistically reconstruct its input. The layers then act as feature detectors. After this learning step, a DBN can further trained with supervision by updating the weights to perform classification. In weight updating process the learning rate is used to diminish function of time, it is relevant to the error rate in each epoch, and also determines how much epoch is required to train the network based on the current values of the weights.

II. METHODOLOGY

Several Restricted Boltzmann Machines (RBM) can be stacked and trained in a greedy manner to form Deep Belief Network architecture. RBM is an unsupervised learning method which contains two layers: input and hidden layers. Each and every layer was constructed with nodes. The input nodes process the user dataset. The hidden nodes extract the multilevel features of the dataset. The connection between the input and hidden layers weighted parameters denoted by W [2]. DBN are graphical models which gain to extract a deep hierarchical representation of the training data. The graphical DBN is showed in figure. 4. The standard RBM is presented in figure. 5[2].



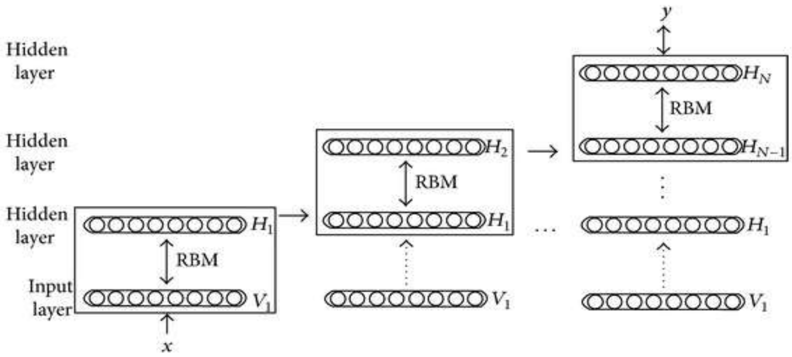


figure 4: DBN Graphical Model **figure 5: RBM Structure**

Let v_i(0≤i≤N) be a binary variable of input neurons, N is the number of input neurons, h_i(0≤j≤M) be a binary variable of hidden neuron, where M is the number of hidden neurons. The energy function $E(i, j)$ for input vector $i \in \{0,1\}^N$ and hidden vector $j \in \{0,1\}^M$ [2] is represented as

$$\boldsymbol{E}(\boldsymbol{v},\boldsymbol{h}) = \sum_i \boldsymbol{b}_i \boldsymbol{v}_i - \sum_i \boldsymbol{c}_j \boldsymbol{h}_j - \sum_i \sum_j \boldsymbol{v}_i \boldsymbol{W}_{ij} \boldsymbol{h}_j \quad \textbf{(1)}$$

The joint probability distribution of v and h is represented as

$$\boldsymbol{p}(\boldsymbol{v},\boldsymbol{h}) = \frac{\mathbf{1}}{\boldsymbol{Z}}\mathbf{exp(-E(v,h)),\ Z=}\sum_v \sum_h \mathbf{exp(-E(v,h))} \quad \textbf{(2)}$$

Where

b_i *and* c_i are the coefficients for v_i and h_j respectively,

Weight w_{ij} is the parameter between v_i and h_j,

Z is the partition function,

P(v,h) is a probability function, calculates sum over all possible pairs of visible and hidden vectors.

DBN copy the joint distribution between input vector v and the h hidden layers h^kas follows

$$\boldsymbol{P(a, h^1,....,h^h) = (\pi_{k=0}^{h-2}\ p(h^k(h^{k+1}))\ \ P(1^{h-1},...,h^h)} \qquad \textbf{(3)}$$

Where $a = h^0$,

$$\boldsymbol{P(h^{k-1}|\ h^k)} \qquad \textbf{(4)}$$

is a conditional distribution for the visible units conditioned on the hidden units of the RBM at level k, and $\boldsymbol{P(h^{h-1},h^b)}$ is the visible hidden joint distribution in the RBM[2].

Algorithm for Training DBN

Input: Dataset

Output: Trained network

Step 1: Train the first layer as RBM that models the input $a = h^{(0)}$ as its visible layer.

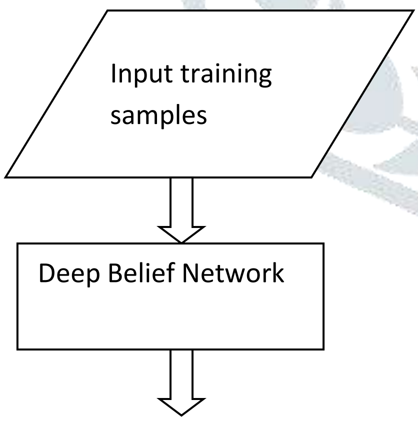
Step 2: By using the first layer obtain representation of the input that will be used as input for the next layer

$p(h^{(1)}{=}1|h^{(0)})$ or $p(h^{(1)}|h^{(0)})$

Step 3: Train the second layer as an RBM

Step 4: Repeat step 2 and step 3 for all the number of layers

figure 6: Algorithm for Training Deep Belief Network



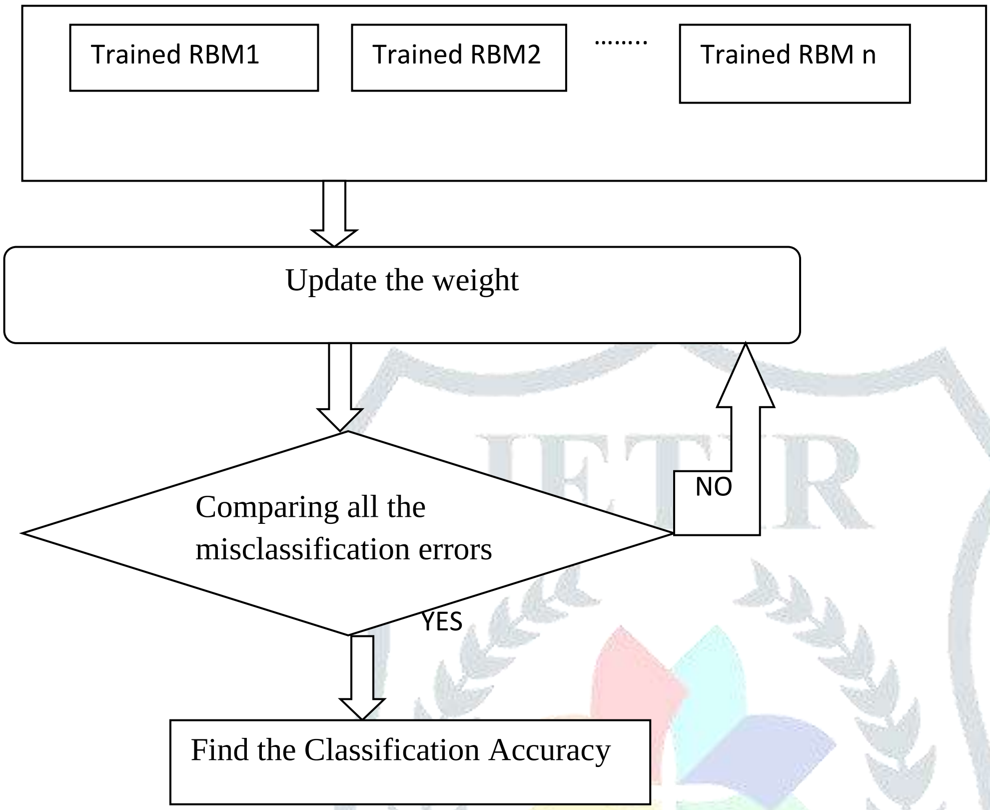


figure 7: Block diagram of DBN algorithm

The figure 7 presents the block diagram of DBN in which the input samples are trained based on the DBN in which the weights are updated and the results are compared to the misclassification errors and finally display the classification accuracy. Training and Testing: Basically individual DBN is pre-trained using contrastive divergence. The pre-training process used to get faster convergence at the fine tuning stage. Testing process is used to determine how the network is trained based on the testing samples.

III. EXPERIMENTAL RESULTS

The proposed method has been implemented by R tool .This section will demonstrate the performance of DBN on medical dataset (Readmission rate of diabetic patients) for different test cases.

Readmission rate of diabetic patients in medical dataset: This dataset used to classify the diabetic patients readmission rate in the hospital. It classifies the patients readmission as NO,>30 and <30. It includes 100000 instance and 51 attributes. The datasets are trained and tested in various aspects of depth with the three test cases. In the first case, the training process includes approximately 60000 instance (60%) and testing process includes 40000 instance (40%). In the second case, the training process includes approximately 70000 instance (70%) and testing process includes 30000 instance (30%). In the third case, the training process includes approximately 80000 instance (80%) and testing process includes 20000 instance (20%). The accuracy for three test case are discussed on the Table 1. figure 8 demonstrates the accuracy chart for medical dataset with three test cases. The test cases with 4 or more hidden layers indicate that too deep architecture brings opposite effect. The error rate for the training datasets are visualized in figure.9 with learning rate of 0.04 and depth =3.

Table 1: Accuracy for medical dataset

Number of depth	TEST CASE 1			TEST CASE 2			TEST CASE 3		
	Accuracy in DBN		Training Time	Accuracy in DBN		Training Time	Accuracy in DBN		Training Time
1	59.12		1007.8	62.09		1019.7	63.7		1027.5
2	61.72		1021.61	64.57		1024.1	65.98		1031.7
3	63.87		1029.4	66.71		1028.4	68.36		1039.5
4	62.75		1038.4	64.8		1032.48	66.8		1041.8
5	61.69		1045.91	63.7		1038.7	65.14		1057.9

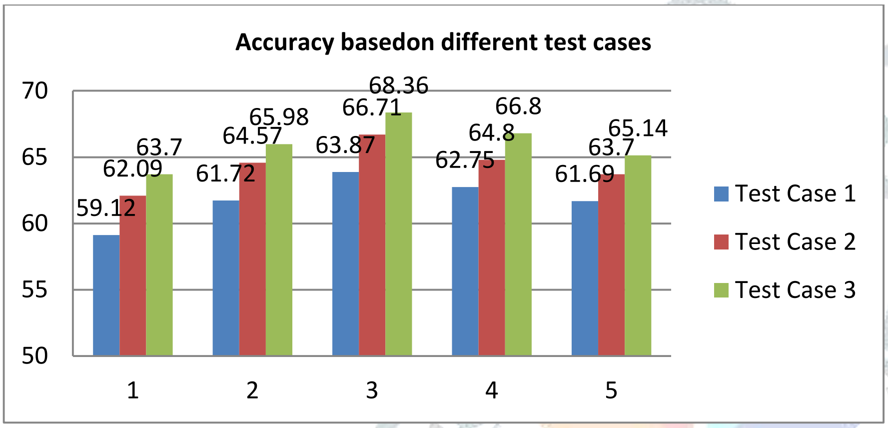


figure 8: Accuracy based on different test cases with learning rate=0.04

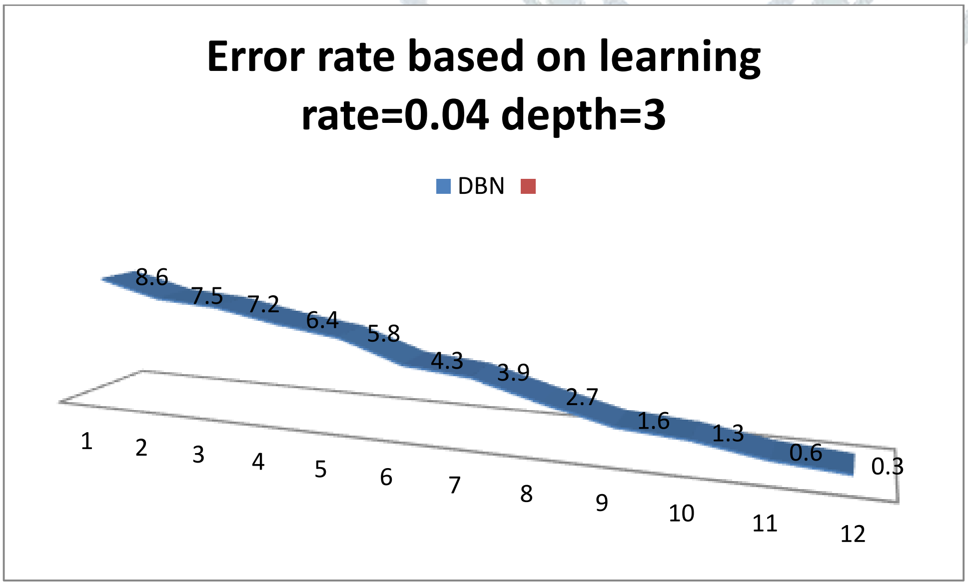


Fig. 9: Error rate based on learning rate=0.04 depth=3

IV. CONCLUSION

In the above experimental results the Deep Belief Network algorithm is implemented on medical dataset and obtain the accuracy based on different test cases. In future we add the learning rate decay to the weight updating process to obtain high accuracy and less training time.

References

[1] Shin Kamada, Takumi Ichimura.: An adaptive learning method of Deep Belief Network by layer generation algorithm. IEEE Xplore. Nov 2016.

[2] Hinton. G. E.: A Practical Guide to Training Restricted Boltzmann Machines. *Neural Networks Tricks of the Trade Lecture Notes in Computer Science*. vol. 7700,.pp. 599-619, 2012.

[3] Mohamed A,Dahl G E, Hinton. G.: Acoustic modeling using deep belief networks[J]. *Audio Speech and Language Processing IEEE Transactions an*, vol. 20, no. 1, pp. 14-22, 2012.

[4] Boureau Y, Cun L. Y.: *Sparse feature learning for deep belief networks [C] //Advances in neural information processing systems*. pp. 1185-1192, 2008.

[5] Roux N Le, Y Bengio.: Representational power of restricted Boltzmann machines and deep belief networks[J]. *Neural Computation*, vol. 20, no. 6, pp. 1631-1649, 2008.

[6] UCI Machine Learning Repository: https://archive.ics.uci.edu.

[7] Hinton. G.: A practical guide to training restricted Boltzmann machines[J]. *Momentum*, vol. 9, no. 1, pp. 926, 2010.

[8] Bengio Y, Lamblin P, Popovici Det al.:Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, vol. 19, pp. 153, 2007.

[9] Ranzato A M, Szummer M.: Semi-supervised learning of compact document representations with deep networks[C]//Proceedings of the 25th international conference on Machine learning. *ACM*, pp. 792-799, 2008.

[10] Neal R M, Hinton G. E.: A view of the EM algorithm that justifies incremental sparse and other variants[M]//Learning in graphical models. pp. 355-368, 1998.

[11] Hinton G E, Salakhutdinov R. R.: Reducing the dimensionality of data with neural networks[J]. *Science*, vol. 313, no. 5786, pp. 504-507, 2006.