Comparative Analysis: Heart Diagnosis Classification using Bp-LVQ Neural Network Models for Analog and Digital Data

D. Rajeswara Rao[1]

[1] KL University

Received: 9 December 2015 Accepted: 1 January 2016 Published: 15 January 2016

Abstract

Decades onwards companies are creating massive data warehouses to store the collected resources. Even though the stored resources are available, only few companies have been able to know that the actual value stored in the database. Procedure used to extract those values is known as data mining. We use so-many technologies to apply this data-mining technique, artificial neural network(ANN) also includes in this data-mining techniques ,ANN is the information processing units which are similar to biological nervous systems. Backpropagation is one of the techniques that used for classification and LVQ (learning Vector Quantization) can be plotted under the competitive learning scheme which is also used for classification. This paper elaborates artificial neural networks, its characteristics and working of backpropagation and LVQ algorithms. In this paper we show the intriguing comparisons between backpropagation and LVQ (Learning Vector Quantization) for both analog and digital data. It also attempts to explain the results between back-propagation and LVQ

Index terms— artificial neural networks (ANN), activation function, multi-layer-feedforwardnetwork, sigmoid, least mean squared error, backpropagation, training, code

1 I. INTRODUCTION

rtificial neural networks (ANN), is often called as "neural networks", is a data processing model based on the biological neural network modeling [5] . Neural networks are widely pre-owned to understand the patterns and the connections in the data. The data may be the outcome of a market research effort, etc. Artificial neural networks have been successfully solved many complex practical issues. The Small processing units present in the network are called as "Artificial Neuron", which operates the information using a connectionist approach to perform complex computations [1] [5] . Basically, neural network have layered architecture with interconnected neurons as from fig-1.1. The neural networks (ANN) can be generally be a either a multiple-layer or a single-layer networks. The multilayer structure of neural networks is shown in fig-1.1.

2 Artificial neural networks had been developed based on the following hypothesis:

? The information is processed among many simple processing units, well known as "neurons". ? The signals are processed among these processing units which are known as neurons over the connection links among them. ? Each and every connection link among these neurons contains an weight, multiples with the transmitted signal. ? Each and every neuron or processing unit applies activation function to its net-input(weight multiplied with its signal input) comes from its previous unit. Let consider a neuron h1 from fig-1.2, which receives inputs from input neurons y1,y2,y3. The weights on the connection from y1,y2,y3 are w1, w2, w3. The net-input N_y from the input nodes with the activations Y1,Y2,Y3 to the neuron h1 is defined as follows:

N_y=w1Y1+w2Y2+w3Y3.

As from the final assumption pass this net input to the activation function given as h1= f(n__y). Some simplifications are necessary to understand the intended properties and to attempt requires mathematical analysis. To implement the above assumptions the whole process of the neural networks are divided in to building blocks. The main building blocks of the neural networks are as follows:

? The Architecture of network.

? Initializing the weights to the nodes.

? Activation Functions.

3 a) Architecture of Neural Networks

The settlement of the neurons into several layers and the arrangement of the connection within and in-between the layers are known as the network architecture. The basic architecture of the simplest possible neural networks that performs classification subsists of a input layer units and a single output layer unit. Number of layers in the neural network can be outlined as the number of layers, which has weighted interconnected links among the neurons. Advanced neural network architecture consists of hidden layer along with the input layers and output layers. If the two layers of interconnected weights are present, then it is found to have hidden layer. The network architecture is divided into different types like Feed-Forward, Feedback, Competitive. For back-propagation algorithm we are using Feed-Forward algorithm, where to LVQ (learning Vector Quantization) uses competitive network.

? Feed-Forward networks: These feed-forward networks have either a single layer of weights, where the neurons in the input layer are directly having connection links to the neurons in the output layer, or multiple layers with an interceding set of hidden neurons. Feed-back networks are also associated in two different ways i) Singlelayer ii) Multilayer. As in the single-layer feed-forward networks the weights from input layer does not influence the output layer. Whereas in multilayer feed forward networks one or many layer of nodes (units) between the input layer and the output layer units, so this network is used to solve the complex problems.

b) Setting the weights to the nodes:

The process of setting the weights enables the learning rules or training process. A neural network focusses on the way in which the weights can be changed. The method of tuning the weights on the connections among the network layers to attain the expected output is known as the network training. The internal process in the network training is called as learning. Basically, the training process is divided into three types i) supervised ii) Unsupervised and iii) Reinforcement training. For both Back-propagation and for LVQ we are using supervised learning to train the data.

Supervised Learning Rule: It is a procedure of contributing the networks with a sample of inputs and collating the output with a target output. Training process continues until we get the target output. The weights must adjust according to the algorithm. The various learning rules that follow the supervised learning are Delta rule, generalized delta rule, Competitive learning rule. Generalized delta rule is used to train the given data set in the back propagation algorithm, where as competitive learning is the process used to train dataset used for LVQ.

? Delta-Rule: This rule is purely based on the least mean squared error (LMS). The Mean squared error is nothing but the average of all the errors calculated between the target and actual values. This rule is used to minimize the error. Let discuss in detail, for a taken input data the output data is equated with a target output. If the difference between target and actual data is zero, no learning process is considered, otherwise the values of weights are adjusted to lessen the error obtained.

The difference between the target output to the actual output value is defined as? (wij) = n* k i *er j ,

where n is the learning rate (?), k i is the activation of unit and er j is the difference between the target value and actual output value. This learning rule not only progress the weight vector nearer to the target weight vector, it does so in the most efficient way. Generalized delta rule: Actually the delta rule uses the local information about the error, where the generalized delta rule deals with error information that is not local. The rule is stated in simple sense as follows for weights updating in a cycle after all the training patterns are presented as W new =w old -n*(E(k)) where n is learning rate and E(k) is the error difference between the target and actual output.

Competitive Learning Rule: In this competitive learning rule, the neurons present in the output-layer of the neural network compete among themselves to be in an active-state. The major idea behind this rule is that to allow the processing units (neurons) to challenge for the authority to answer a taken sets of inputs, such that only a output neuron (processing unit) challenge for the right to respond for a given subset of inputs. So that only a neuron in the output-layer is in an active-state at a time. The neuron which wins in the competition is known as winner-takes-all neuron. Let W kj denotes the weight of input-layer node (unit) j to neuron. The neuron learns by altering the values of weights from inactive input mode to active input mode. If a neuron (processing unit) does not give acknowledgement to a particular input layout, then the learning does not happens in that particular neuron. If any of the neuron wins in the competition, then its weights are adjusted as follows.

Î?"W kj = n (X j -W kj), when neuron k wins the competition. =0 ,when neuron k losses the competition.

4 Global Journal of Computer Science and Technology

Volume XVI Issue V Version I ( )

As from above formulae "n" is well known as the learning-rate(?). The values of the weights are initially set to random values and those weights are being normalized during learning phase (either supervised or unsupervised). The winner-takes-all neuron is selected by using Euclidean distance.

5 c) Activation Function

The activation function is used to calculate the output comeback generated by neurons. Threshold function performs final mapping of activations of network neurons. The outcome of any neuron is a result of thresholding (internal activation). The aggregate of the weighted input signals is pertained with activation function to get the response. There may be linear and non-linear activation functions. Generally, the activation functions are classified into different types [2] : i. Identity Function. (0,1) Hyperbolic tangent S(x)=(e x -e -x)/(e x +e -x) (-1,1) Backpropagation is one of the neural network learning algorithms, delineated to diminish the mean square error. Backpropagation is also well-known as the "error backpropagation", because this algorithm is purely based on the error correction learning rule. This algorithm is used to train the multi-layer artificial neural network. Back propagation uses supervised learning rule, in which it generates error by comparing target output to actual output. The backpropagation algorithm could be broken down into four main steps **??**1][2] : During the first stage, the weights are set-up to some random values (e.g., they ranges from [-1.0,1.0]or[-0.5,0.5]) [2]. Every processing unit in the network is associated with a bias (threshold), which is used to generate the net input. The algorithm used in the back-propagation network to train the network is implemented in four different stages is as follows:Ramp R(x)= x , x>=0 =0 , x<0 R(x)=max(x,0) [-1,+1] Step O, if x<01, x>=1 [0,+1]?

Weights Initialization [2] :

Step-1: Initializing the weights and bias to random values (ranges from [-1.0,+1.0] or [-0.5,0.5]).

Step-2: Checking for the stopping condition, if it is false do the steps from 3 to 10.

Step-3: Foe each and every training set, perform the steps from 4 to 9 as mentioned below.

Feed-Forward of input training patterns [3] :

Step-4: Each and every input unit accepts the input x i and transmits that input signal to hidden layer units.

Step 5: Each hidden unit in the network aggregates its weighted input signals. Activation function to z ij is denoted by Z jz ij = v oj +?x i V ij .i=1 to n Z j = f(z ij)

The result obtained from this activation function is the input to next layer in the network.

Step 6: Each output unit in the network, aggregates its weighted input signals . Activation function applied to y ik is denoted byY k y ik = w ok +?Z j W jk Y k =f (y ik) Backpropagation of the errors:

Step 7: Error is calculated asE(k)= ?[O j (k)-T j (k)]2 j=1 to m E=E(k) f(y ik)

Step 8: Find the mean squared error E t =1/2 ?E k=1 to N

6 Updating of weights and bias

Step 9: For the Output layer the weights and the bias are updated as follows Î?"W jk =?E t z j . Updated weight is as follows W jk (new) = W jk (old) + Î?"W jk Î?"wok=?E . To update bias is w ok (new) =w ok (old) +Î?"w ok Similarly the values of weights and the bias are updated in the networks hidden layer is as follows: Î?"V ij =?E t x i . The new weight is calculated as V ij (new) =V ij (old)+Î?"V ij Î?"v oj =?E. Updated bias is v oj (NEW)=v oj (OLD)+Î?"v oj Step 10: Check the stopping condition.

Based upon the algorithm stated above the terms are defined as x i -Inputs that given to the input units. v oj -Bias used in the hidden layer units. V ij -Weights used in hidden layer units. w ok -Bias used for the outputunits. W jk -Weights that initialized in output layer. ?-Learning rate. Learning Vector Quantization (LVQ) algorithm is the prototype based supervised classification algorithm. It is a particular case of artificial neural network, which implements "winner-take-all" principle [2] . Winner-take-all is the computational principle applied by which neurons in layer compete with each other for activation. The neuron with highest activation stays active while other neurons shut down. LVQ is trained to classify the inputs according to the given targets. Training in LVQ occurs by performing the competition between the neurons. LVQ uses Euclidean distance to perform the competition between neurons. LVQ performs the classification for every target output unit by considering its input pattern i.e, it uses supervised learning technique.

LVQ defines the class boundaries based upon its prototypes. The prototypes are determined during the training procedure using a labeled dataset (the dataset that we take for training).LVQ system is represented by protocols which are defined in future of observed data. The class boundaries are not depends not only on prototypes but also on nearest neighbor rule and winner-takes-it-all. Weight vector for an output unit in a network is known as the "codebook vectors (CV)" or "reference". The architecture of the LVQ algorithm is as shown fig: **??** As from the above diagram the net input to the hidden layer is : n 1 i = || i W 1 -p|| where i W 1 represents training vector i.e,, inputs given to the input layer p represents Weight vector for the units in next layer it is also called as the codebook vector.

Finally the net output of this input layer is passed to the activation function, where we use the competitive activation function for this LVQ algorithm. Competitive Activation Function which represents the input/output relation that purely derives by using the Euclidian rule in which a 1 = compet(n 1) a 1 = 1 neuron which wins the competition =0 for all neurons. Therefore the neuron whose weight vector is nearest to the input vector will gives output as 1, and the remaining neurons will gives the output as 0 as shown above. This states that the

LVQ network purely competitve network . As initially stated that the neurons in input layer are considered as the same class, after this net output generation to the hidden layer the winning neuron represents a subclass. There may be different neurons that may win the competition, they all belongs to the same sub class.

The hidden layer of the LVQ (learning vector quantization) network combines all subclasses into a single class. As shown in the above figure W 2 done the whole process of combining all the sub classes. W 2 is represented in matrix, in which columns represent the subclasses and the rows represents the classes.

Note: W 2 matrix has a value of 1 in each column, eith the other values set to zero (0).The subclass of a particular class is denoted by the value of 1 in the row. Ex: W 2 ij =1 means j sub class is a part of ith class.

The input vector X is selected at random from the inputs given. If the class labels of the input vector x and a codebook vector (weight vector) W agree, the codebook vector W is moved in the direction of the input p W 2 W 1 C n 1 n 2 a 1 a 2 Input Competitive layer n 1 i = || i W 1 -p|| a 1 =compet(n 1)

7 Global Journal of Computer Science and Technology

Volume XVI Issue V Version I ( ) vector X. If the class labels of the input vector X and the codebook vector w is disagreed, the codebook vector W is moved away from the input vector X. I.

Ex: Let {W i } 1 i=1 stand for the set of weighted vectors (codebook vectors), and the {X i } N i=1 stand for the set of input vectors. Suppose, that the codebook vector W c is the nearest to the input vector X i . Let K wc denote the class associated with the codebook vector W c and K xi denote the class label of the input vector X i . The values of K wc and K xi are obtained from the W 2 . The codebook vector W c is regulated as follows:If K wc = K xi ,then W c (New) = W c (Old) + ? n [X i - W c (Old)] where 0< ? n <1. If K wc ? K xi ,then W c (New) = W c (Old) -? n [X i -W c (n)]

,where 0< ? n <1. II .

Remaining codebook Vectors are not modified.

The learning rate (?) is decreased. This whole LVQ process continues until the stopping condition fails.

Learning Vector Quantization Algorithm [2] :

Step-1: Initialize weights vectors (codebook vectors) and learning rate.

Step-2: Check for the stopping condition. If the condition is false, then perform the steps from 3 to 7.

Step-3: For every training input vector p, do the steps from 4-5

Step 4: Figure out J using Squared Euclidean distance E(j) = ? (j W 1 -X i) where X i is input present in the input vector. Find j when E(j)is minimum

Step 5: The value of W j is updated as follows If K wc = K xi ,then W j (New) = W j (Old) + ? n [X i -W j (Old)] where 0< ? n <1. If K wc ? K xi ,then W j (New) = W j (Old)? n [X i -W j (n)] where 0< ? n <1.

Step 6: Reduce the learning rate.

Step 7: Test for the stopping condition.

8 III. COMPARISION BETWEEN BACKPROPAGATION AND LVQ

The practical implementation of backpropagation involves factors like choice of network architecture, momentum factor. While implementing these factors backpropagation algorithm associated with few problems like local minima. A local minimum is the problem that occurs frequently, used to change the weights frequently to minimize the error. As in this local minima, in some cases the error might have to rise part of more general fall. If this is the situation the algorithm will struck and the error will not be decreased further. So, for this drawback LVQ gives best results. In this paper we are comparing the efficiencies obtained for testing the heart disease dataset with both backpropagation and LVQ for the two different ranges (-1,1) and (0,1). The following are the results obtained while comparing the both algorithms. The programming is written for 100 instances of a heart diseases dataset from Cleveland with 14 attributes (13 +class attribute).

9 a) BackPropagation

In our paper we practice backpropagation algorithm with different learning rates and finally conclude, how the efficiency changed based upon the value of alpha (learning rate) . To allow fair comparison between backpropagation and LVQ a wide variety of parameter values are tested for each algorithm. The backpropagation network is trained on our dataset for different alpha values for different ranges and the observed results are mentioned in the below tables as follows: When i)?=0.9 (learning rate) Varying the learning rate alpha from 0.1 to 0.9, it was found that the maximum efficiency is obtained at alpha ?=0.1. The results that obtained for various alpha values are shown in the following tables. Our paper also attempts to check the efficiency for different ranges i,e for analog (0,1) and bipolar (-1,1). The better classification efficiency can be achieved by varying the learning rate. As from the above results , we found that the digital gave better efficiency than analog in vector quantization method. It is also found that maximum efficiency was obtained for alpha value 0.1.

10 IV.

11 Conclusion

In this paper we present a supervised learning based approach to data-mining classification rules for a dataset. The classification is carried out using backpropagation and LVQ. We conclude that LVQ algorithm is one of the best in classification when compared to backpropagation. As from the results obtained for classifying our dataset, we can obtain better classification efficiency by varying the learning rate and it was found that maximum efficiency was obtained for alpha value 0.1 in both algorithms. Comparing the digital results (-1,1) with the analog results, it is found that the digital data gave better efficiency than analog in both back-propagation and LVQ algorithms. Overall comparison between the two algorithms states that the maximum efficiency is obtained in LVQ with high processing time. [1] [2] [3] [4]

12

Figure 1: Fig- 1 . 2 :

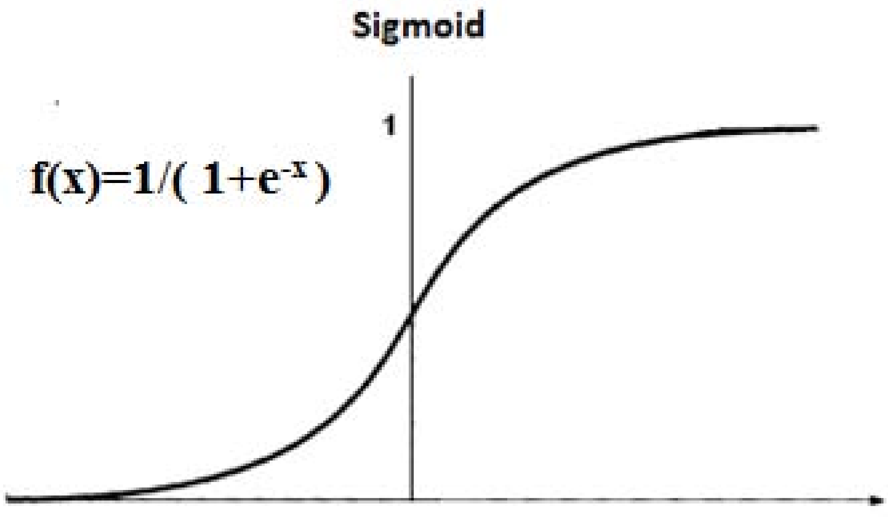


Figure 2:

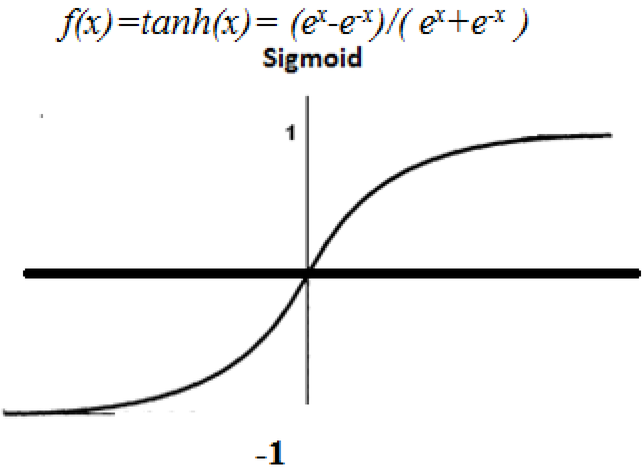


Figure 3:

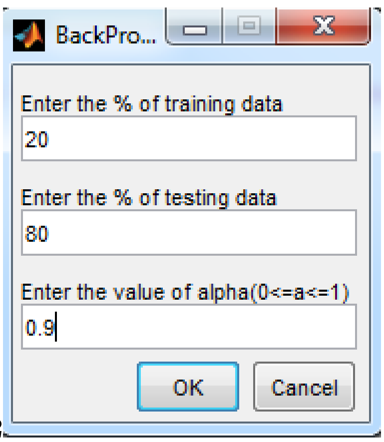
21222


Figure 4: Fig. 2 . 1 : 2 Fig: 2 . 2 :

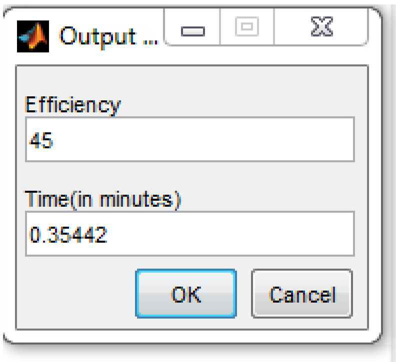


Figure 5:

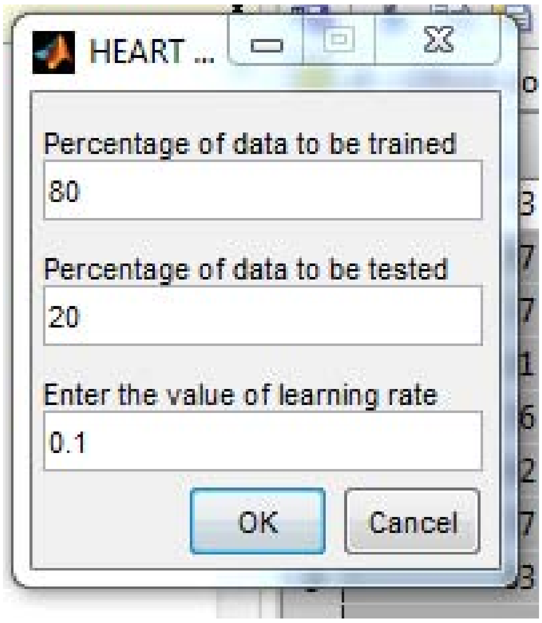
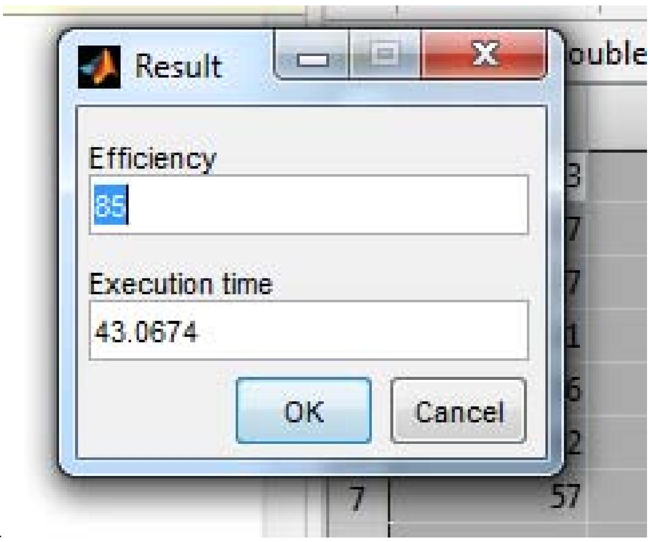


Figure 6:



31

Figure 7: Fig: 3 . 1 :

1

[Note: a) Backpropagation Algorithm]

Figure 8: Table 1 .

.

Sl.No	Training(%)	Testing(%)	Time(in minutes)	Efficiency(in%)
1	20	80	2.2	45
2	40	60	0.35	55
3	60	40	0.007	77.5
4	80	20	0.009	75

ii) ?=0.8 (learning rate)

Table. 3.2 : Efficiency obtained for backpropagation (digital) ?=0.8

Sl.No	Training(%)	Testing(%)	Time(in minutes)	Efficiency (%)
1	20	80	0.003	28.75
2	40	60	0.005	23.333
3	60	40	0.005259	50
4	80	20	0.008362	50

Table. 3.3 : Efficiency obtained for backpropagation (analog) ?=0.1

Sl.No	Training(%)	Testing(%)	Time(min)	Efficiency (%)
1	20	80	0.0032099	38.75
2	40	60	0.006441	43.333
3	60	40	0.075057	40
4	80	20	0.010575	60

Table. 3.3 : Efficiency obtained for backpropagation (digital) ?=0.1

1	20	80	0.0032	62.5
2	40	60	0.00503	63.333
3	60	40	0.0066	60
4	80	20	0.00888	79

[Note: Sl.No Training(%) Testing(%) Time(min) Efficiency(%) b) Learning Vector Quantization Fig.3.3 : Input to LVQ algorithm]

Figure 9: Table . 3

.

Sl.No	Training(%)	Testing(%)	Time(min)	Efficiency (%)
1	20	80	23.7953	54
2	40	60	25.186	57
3	60	40	10.7664	60
4	80	20	10.164	70

Figure 10: Table . 3

.

Sl.No	Training(%)	Testing(%)	Time(min)	Efficiency
1	20	80	6.5829	64
2	40	60	6.2778	70
3	60	40	8.4187	70
4	80	20	7.175	85

Figure 11: Table . 3

:

Sl.No	Training(%)	Testing (%)	Time(inmin)	Efficiency(%)
1	20	80	8.7658	70
2	40	60	9.0779	62
3	60	40	12.1897	80
4	80	20	97.8381	70

Figure 12: Table : 3

[Haykin and Networks ()] , S Haykin , Networks . 1999. Prentice Hall International Inc.

[Mayadevi Somanathan and Kalaichelvi ()] 'An Intelligent Technique for Image Compression'. Athira Mayadevi Somanathan , V Kalaichelvi . *International Journal for Recent Development in engineering and Technology* 2347- 6435. 2014. 2.

[Zurada ()] *An introduction to artificial neural networks systems*, J M Zurada . 1992. st.paul: West Publishing.

[Agrawal et al. (1993)] 'Database Mining : A Performance Persepective'. R Agrawal , T Imielinski , A Swami . *IEEE Transactions on Knowledge and Data Engineering* December 1993. p. .

[Bradely ()] 'Introduction to Neural Networks'. I Bradely . *Multinet Systems Pty Ltd* 1997.

[Sivanandam et al. ()] *Introduction to Neural Networks Using Matlab 6.0*, S N Sivanandam , S N Deepa , S Sumathi . 2006. Noida: Tata MCGraw-Hill.

[Bengio et al.] 'Introduction to the Special issue on neural networks for data mining and knowledge discovery'. Y Bengio , J M Buhmann , M , J M Zurada . *IEEE Trans. Neural Networks*

[Jain et al. ()] A K Jain , J Mao , K M Mohiuddin . *Artificial Neural Networks: A tutorial*, 1996. 29 p. .

[Gaur] 'Neural Networks in Data Mining'. Priyanka Gaur . *International Journal of Electronic and computer Science Engineering* 2277-1956/VIN3-1449-1453. 1. (IJECSE)

[Dr et al. ()] 'NEURAL NETWORKS IN DATA MINING'. Dr , Alok Singh , Singh Chauhan . *Journal of Theortical and Applied Information Technology* 2009.