www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 1 January 2016, Page No. 15624-15630

Modular Implementation of Neural network Structures in Marketing Domain Using Data Mining Technique

Dr.S.P.Victor* C.RajKumar
Associate Professor-CS-St.Xaviers College
drspvictor@gmail.com
Research Scholar, MS University, Tirunelveli
crajkumarmsu@rediffmail.com

Abstract-
The process of implementing neural networks comprises several strategies in the field of data mining. The proper application of supervised and unsupervised learning concepts play a vital role in the successful implementation of Neural network methods which are not easily used for data mining tasks. Since they often produce incomprehensible models and require long training times. In this paper we implement neural network learning algorithms that are able to produce comprehensible models and that do not require excessive training times. The real time implementation of marketing domain is taken into account for the incorporation of data mining rule extraction approach which involves extracting symbolic models from trained neural networks. In near future we will implement the descriptive and predictive approaches of data mining implementation in real time using neural network conceptual schema.
Keywords: Data mining, neural networks, rule extraction, Marketing, predictive data

I.INTRODUCTION

The central focus of the data mining enterprise is to gain insight into large collections of data Often achieving this goal involves applying machine learning methods to inductively construct models of the data at hand. Neural networks have been applied to a wide variety of problem domains to learn models that are able to perform such interesting tasks as steering a motor vehicle recognizing genes in uncharacterized DNA sequences, scheduling payloads for the space shuttle and predicting exchange rates. Although neural network learning algorithms have been successfully applied to a wide range of supervised and unsupervised learning problems, they have not often been applied in data mining settings in which two fundamental considerations are the comprehensibility of learned models and the time required to induce models from large data sets [1]. We discuss new developments in neural network learning that actively address the comprehensibility and speed issues which often are of prime importance in the data mining community. Specifically we describe algorithms that are able to extract symbolic rules from trained neural networks and algorithms that are able to directly learn comprehensible models .Inductive learning is a central task in data mining since building descriptive models of a collection of data provides one way of gaining insight into it. Such models can be learned by either supervised or unsupervised methods depending on the nature of the task. In supervised learning the learner is given a set of instances of the form [2].

H(x) = yi where y represents the variable that we want the system to predict and x is a vector of values that represent features thought to be relevant to determining y[3].

The goal in supervised learning is to induce a general mapping from x vectors to y values that is the learner must build a model to predict y values for previously unseen examples. In unsupervised learning the learner is also given a set of training examples but each instance consists only of the x part it does not include the y value. The goal in unsupervised learning is to build a model that accounts for regularities in the training set. In both the supervised and unsupervised case learning algorithms define considerably in how they represent their induced models [4]. Many learning methods represent their models using languages that are based on or closely related to logical formulae Neural network learning methods on the other hand represent their learned solutions using real valued parameters in a network of simple processing units We do not provide an introduction to neural network models in this article but instead refer the interested reader to one of the good textbooks . Specifically we discuss why one might want to consider using neural networks for such tasks and we discuss why trained neural networks are usually hard to understand [5]. The Suitability of Neural Networks for Data Mining Before describing particular methods for data mining with neural networks we rest make an argument for why one might want to consider using neural networks for the task The essence of the argument is that for some problems neural networks provide a more suitable inductive bias than competing algorithms[6].

DOI: 10.18535/ijecs/v5i1.28

Let us discuss the meaning of the term inductive bias Given a set of training examples there are in nightly many models that could account for the data and every learning algorithm has an inductive bias that determines the models that it is likely to return There are two aspects to the inductive bias of an algorithm its restricted hypothesis space bias and its preference bias [8]. The restricted hypothesis space bias refers to the constraints that a learning algorithm places on the hypotheses that it is able to construct For example the hypothesis space of a perception is limited to linear discriminate functions The preference bias of a learning algorithm refers to the preference ordering it places on the models that are within its hypothesis space For example most learning algorithms initially try to t a simple hypothesis to a given training set and then explore progressively more complex hypotheses until they nod an acceptable. In some cases neural networks have a more appropriate restricted hypothesis space bias than other learning algorithms For example sequential and temporal prediction tasks represent a class of problems for which neural networks often provide the most appropriate hypothesis space [7]. Recurrent networks which are often applied to these problems are able to maintain state information from one time step to the next This means that recurrent networks can use their hidden units to learn derived features relevant to the task at hand and they can use the state of these derived features at one instant to help make a prediction for the next instance. In other cases neural networks are the preferred learning method not because of the class of hypotheses that they are able to represent but simply because they induce hypotheses that generalize better than those of competing algorithms Several empirical studies have pointed out that there are some problem domains in which neural networks provide superior predictive accuracy to commonly used symbolic learning algorithms [9]. Although neural networks have an appropriate inductive bias for a wide range of problems they are not commonly used for data mining tasks As stated previously there are two primary explanations for this fact trained neural networks are usually not comprehensible and many neural network learning methods are slow making them impractical for very large data sets [10].

II.PROPOSED METHODOLOGY

This proposed methodology focuses on the implementation of a neural network strategy to search the requested data mining details by implementing the marketing computations. .

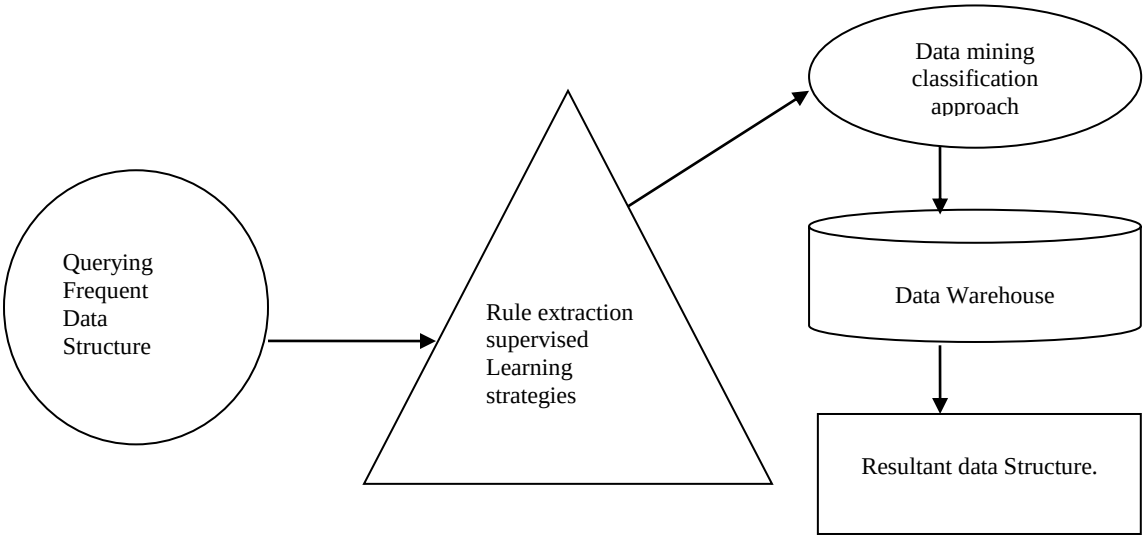

Fig 1: Proposed Neural network based data mining structure

III. IMPLEMENTATION

The most common action in data mining is classification. It recognizes patterns that describe the group to which an item belongs. It does this by examining existing items that already have been classified and inferring a set of rules. Similar to classification is clustering. The major difference being that no groups have been predefined. Prediction is the construction and use of a model to assess the class of an unlabeled object or to assess the value or value ranges of a given object is likely to have. The next application is forecasting. This is different from predictions because it estimates the future value of continuous variables based on patterns within the data. Neural networks, depending on the architecture, provide associations,
Classifications, clusters, prediction and forecasting to the data mining industry.

DOI: 10.18535/ijecs/v5i1.28

Artificial neural networks are relatively crude electronic networks of neurons based on the neural structure of the brain. They process records one at a time, and learn by comparing their classification of the record (i.e., largely arbitrary) with the known actual classification of the record. The errors from the initial classification of the first record is fed back into the network, and used to modify the networks algorithm for further iterations.

Roughly speaking, a neuron in an artificial neural network is

1. A set of input values (xi) and associated weights (wi).

2. A function (g) that sums the weights and maps the results to an output (y).

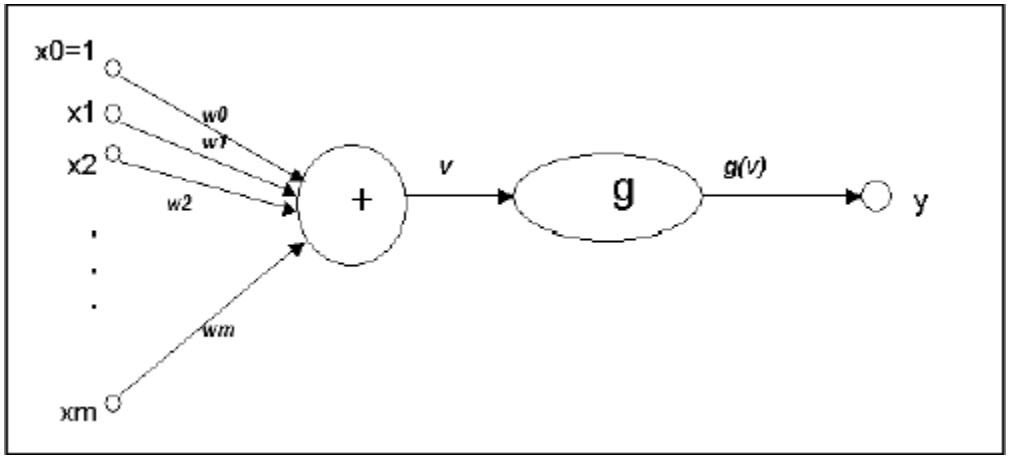

Fig-2: Neuron in Neural network

Neurons are organized into layers: input, hidden and output. The input layer is composed not of full neurons, but rather consists simply of the record's values that are inputs to the next layer of neurons. The next layer is the hidden layer. Several hidden layers can exist in one neural network. The final layer is the output layer, where there is one node for each class. A single sweep forward through the network results in the assignment of a value to each output node, and the record is assigned to the class node with the highest value.

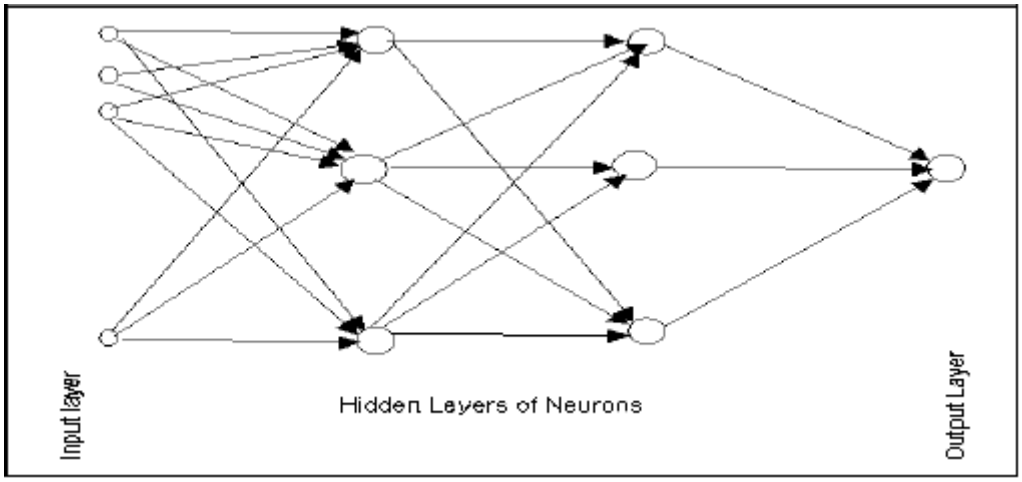

Fig-3: Sample space Neuron mining structures

In the training phase, the correct class for each record is known (this is termed supervised training), and the output nodes can be assigned correct values -- 1 for the node corresponding to the correct class, and 0 for the others. (In practice, better results have been found using values of 0.9 and 0.1, respectively.) It is thus possible to compare the network's calculated values for the output nodes to these "correct" values, and calculate an error

term for each node (the Delta rule). These error terms are then used to adjust the weights in the hidden layers so that, hopefully, during the next iteration the output values will be closer to the correct values.

The Iterative Learning Process

A key feature of neural networks is an iterative learning process in which records (rows) are presented to the network one at a time, and the weights associated with the input values are adjusted each time. After all cases are presented, the process is often repeated. During this learning phase, the network "trains" by adjusting the weights to predict the correct class label of input samples. Advantages of neural networks include their high tolerance to noisy data, as well as their ability to classify patterns on which they have not been trained. The most popular neural network algorithm is the back-propagation algorithm proposed in the 1980's.

Once a network has been structured for a particular application, that network is ready to be trained. To start this process, the initial weights are chosen randomly. Then the training, or learning, begins.

The network processes the records in the Training Set one at a time, using the weights and functions in the hidden layers, and then compares the resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights for application to the next record. This process occurs over and over as the weights are continually tweaked. During the training of a network the same set of data is processed many times as the connection weights are continually refined.

Note that some networks never learn. This could be because the input data does not contain the specific information from which the desired output is derived. Networks also will not converge if there is not enough data to enable complete learning. Ideally, there should be enough data available to create a Validation Set.

Delta rule:

The training process normally uses some variant of the Delta Rule, which starts with the calculated difference between the actual outputs and the desired outputs. Using this error, connection weights are increased in proportion to the error times, which are a scaling factor for global accuracy. This means that the inputs, the output, and the desired output all must be present at the same processing element. The most complex part of this algorithm is determining which input contributed the most to an incorrect output and how the input must be modified to correct the error. (An inactive node would not contribute to the error and would have no need to change its weights.) To solve this problem, training inputs are applied to the input layer of the network, and desired outputs are compared at the output layer. During the learning process, a forward sweep is made through the network, and the output of each element is computed layer by layer. The difference between the output of the final layer and the desired output is back-propagated to the previous layer(s), usually modified by the derivative of the transfer function. The connection weights are normally adjusted using the Delta Rule. This process proceeds for the previous layer(s) until the input layer is reached.

Table-1: Organic Vegetable Market analysis report

Resident/Vegetable market-Xi/Yi	Retailers	Veg-Stores	Super Market-Poly	Home Suppliers	Dedicated attachments	Hifi-Organic suppliers
Apartments	8 %	17 %	25 %	30 %	6 %	14 %
Individual house	11 %	18 %	27 %	33 %	5 %	6 %
Line Houses	14 %	20 %	18 %	40 %	7 %	1 %
Remote Residents	16 %	14 %	45 %	6 %	16 %	3 %
Gated Community	1 %	2 %	14 %	29 %	21 %	33 %
Non Roof Concrete	47 %	46 %	4 %	1 %	1 %	1 %
Market value	91	117	133	139	56	58
Weights	3	4	5	6	1	2

In this table the horizontal values are the primary computation sector whereas the vertical values are meaningless due to the non individual retailer or veg stores or super market etc.

Table-2: Vertical representation of Organic Vegetable Market analysis report

Resident/Vegetable market-Xi/Yi	Dedicated attachments	Hifi-Organic suppliers	Retailers	Veg-Stores	Super Market-Poly	Home Suppliers
Apartments	6 %	14 %	8 %	17 %	25 %	30 %
Individual house	5 %	6 %	11 %	18 %	27 %	33 %

Line Houses	7 %	1 %	14 %	20 %	18 %	40 %
Remote Residents	16 %	3 %	20 %	14 %	41 %	6 %
Gated Community	21 %	33 %	1 %	2 %	14 %	29 %
Non Roof Concrete	1 %	1 %	47 %	46 %	4 %	1 %
Market value	56	58	91	117	133	139
Weights	1	2	3	4	5	6

Goal: Particular non pesticide vegetable product (Optimized minimal cost) to reach all part of the Tirunelveli city to capture the entire Tirunelveli market.

Swatch Case (Xi)
Case X1: Yi = Y6 || Yi= Y5
Case X2: Yi = Y6 || Yi= Y5
Case X3: Yi = Y6 || Yi= Y4
Case X4: Yi = Y5 || Yi= Y3
Case X5: Yi = Y2 || Yi= Y6
Case X6: Yi = Y3 || Yi= Y4
End Case

Table-3: Neuron weight Adjustment Table

w/A	W11	W12	W13	W14	W21	W22	W23	W24	W31	W32	W33	W34	W41	W42	W43	W44	W51	W52	W53	W54	W61	W62	W63	W64
A1	.2	.5	.7	1	.3	.6	.8	1	1	.8	.6	.3	1	.8	.6	.3	1	.7	.5	.2	1	**.7**	.5	.2
A2	.2	.5	.7	1	.3	.6	.8	1	1	.8	.6	.3	1	.8	.6	.3	1	.7	.5	.2	1	**.7**	.5	.2
A3	.1	.4	.6	1	.2	.5	.7	1	1	.7	.5	.2	1	.7	.5	.2	1	.6	.4	.1	1	.7	**.5**	.2
A4	.1	.3	.5	1	.1	.4	.6	1	1	.6	.4	.1	1	.6	.4	.1	1	.5	**.2**	.1	1	.5	.3	.1
A5	.1	.2	.4	1	.2	.5	**.7**	1	1	.5	.3	.1	1	.5	.3	.1	1	.4	.2	.1	1	.4	.2	.1
A6	.1	.2	.3	1	.1	.2	.3	1	1	.2	.3	**.1**	1	.2	.3	.1	1	.3	.2	.1	1	.3	.2	.1

The weight corrections are computed as follows,

START
Step 1: Goal: Particular non pesticide vegetable product (Optimized minimal cost) to reach all part of the Tirunelveli city to capture the entire Tirunelveli market.

Step 2: The selection of the market is based on the higher level of consumption rates.
Swatch Case (Xi)
Case X1: Yi = Y6 || Yi= Y5
Case X2: Yi = Y6 || Yi= Y5
Case X3: Yi = Y6 || Yi= Y4
Case X4: Yi = Y5 || Yi= Y3
Case X5: Yi = Y2 || Yi= Y6
Case X6: Yi = Y3 || Yi= Y4
End Case

Step 3: Iterate the weight values based on the following criteria

a1=1 for yi-y6 or yi=y5 otherwise a1=0
a2=1 for yi-y6 or yi=y5 otherwise a1=0
a3=1 for yi-y6 or yi=y4 otherwise a1=0
a4=1 for yi-y5 or yi=y3 otherwise a1=0
a5=1 for yi-y2 or yi=y6 otherwise a1=0
a6=1 for yi-y3 or yi=y4 otherwise a1=0

Step 4: Check the criteria for Ai=1(To reach all the resident types)

Compute the total value for all the resident types and market.
∑aiwj.Yk=100

Step 5: Iterate the weights as follows,
If ∑aiwj.Yk > 100 reduce the weights in the order w6iw5i,w4i,w3i,w2i,w1i
Else
If ∑aiwj.Yk < 100 increase the weights in the order w1iw2i,w3i,w4i,w5i,w6i
Recompute ∑aiwj.Yk

Step 6: If ∑aiwj.Yk=100(approximate value with lesser significance rate) then STOP GOTO Step 5:

STOP

IV.RESULTS AND DISCUSSION:

Now the proposed algorithmic computational values for fig 1.2 we obtain the following tabulation,

∑aiwj.Yk=100

a1=1 for yi-y6 or yi=y5 otherwise a1=0
a2=1 for yi-y6 or yi=y5 otherwise a1=0
a3=1 for yi-y6 or yi=y4 otherwise a1=0
a4=1 for yi-y5 or yi=y3 otherwise a1=0
a5=1 for yi-y2 or yi=y6 otherwise a1=0
a6=1 for yi-y3 or yi=y4 otherwise a1=0

Performing the weight corrections we obtain the following result
(a1.w62).Y1+(a2.w62).Y2+(a3.w63).Y3+(a4.w53).Y4+(a5.w23).Y5+(a6.w34).Y6
=0.7*30+0.7*33+0.5*40+0.2*41+0.7*33+0.1*47
=100

Table-4: Weight Correction Table

Resident type	Marketers	Supply Value
Apartments	Home Suppliers	21 %
Individual house	Home Suppliers	23 %
Line Houses	Home Suppliers	20 %
Remote Residents	Super Market-Poly	8 %
Gated Community	Hifi-Organic suppliers	23 %
Non Roof Concrete	Retailers	5 %

The following resultant graph shows the optimized solution for the neural network based data mining in vegetable marketing using our proposed model.

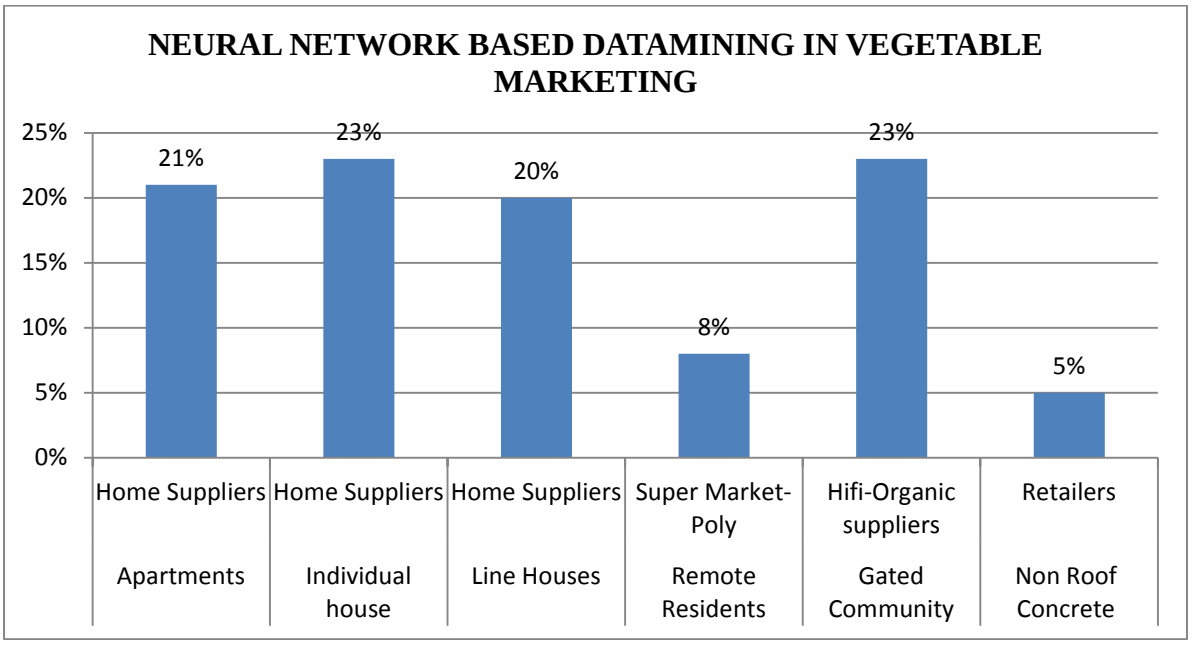

Fig-4: Neural network based data mining in Vegetable Marketing Chart

V.CONCLUSION:

In this paper, we implemented the neural network based rule extraction schema for the data mining classification technique in the non pesticide vegetable marketing domain with our proposed algorithmic strategy, The advantage of such simplifications is that the complexity of identifying the optimal business strategy is reduced when tackling the original problem, and this allows the use of techniques that require evaluating a lot of individuals through the search for the best solution. This approach may include many relationships that can be decisive when searching for a satisfactory decision making. The overall method proves to be highly efficient compared to random probability theory based approach, dramatically reducing running time and number of features required for the optimization issues. Moreover, the experimental results revealed that the expressiveness of weight adjustment towards the impact influence optimization representatives is significantly higher than that of normal data mining classification procedures, because a lower number of features are associated with better accuracy, mainly due to higher specificity, reducing false alarms in fixing the market tasks. In our future work, we have planned to propose a new neural network based prediction method based on data mining technique, provide its implementation and compare its results with the different existing prediction based data mining algorithms.

References

[1] T. Ash, "Dynamic node creation in back propagation networks", *Connection Sci*., vol. 1, pp. 365–375, 1989.
[2] L. Prechelt, "Proben1-A Set of Neural Network Benchmark Problems and Benchmarking Rules", *University of Karlsruhe*, Germany, 1994.
[3] R. Reed, "Pruning algorithms-A survey," *IEEE Trans. Neural Networks*, vol. 4, pp. 740-747, 1994.
[4] R. Setiono and L.C.K. Hui, "Use of quasi-Newton method in a feedforward neural network construction algorithm", *IEEE Trans. Neural Networks*, vol. 6, no.1, pp. 273-277, Jan. 1995.
[5] R. Setiono, Huan Liu, "Understanding Neural networks via Rule Extraction", *In Proceedings of the International Joint conference on Artificial Intelligence*, pp. 480-485, 1995.
[6] Simon Haykin, "Neural Networks- A Comprehensive Foundation", Second Edition, *Pearson Edition Asia*, Third Indian Reprint, 2002.
[7] T. Y. Kwok and D. Y. Yeung, "Constructive Algorithm for Structure Learning in feed- forward neural network for regression problems," *IEEE Trans. Neural Networks*, vol. 8, pp. 630-645, 2007.
[8] M. Monirul. Islam and K. Murase, "A new algorithm to design compact two hidden-layer artificial neural networks", *Neural Networks*, vol. 4, pp. 1265–1278, 2007.
[9] M. Monirul Islam, M. A. H. Akhand, M. Abdur Rahman and K. Murase, "Weight Freezing to Reduce Training Time in Designing Artificial neural Networks", *Proceedings of 5th ICCIT*, EWU, pp. 132-136, 27-28 December 2008.
[10] R. Parekh, J.Yang, and V. Honavar, "Constructive Neural Network Learning Algorithms for Pattern

Classification", *IEEE Trans. Neural Networks,* vol. 11, no. 2, March 2008.