International Journal of Information & Network Security (IJINS)
Vol.1, No.4, October 2012, pp. 282~293
ISSN: 2089-3299

Knowledge Extraction From Trained Neural Networks

Koushal Kumar
Department of CSE/IT, Lovely Professional University, Jalandhar, Punjab, India

Article Info

Article history:

Received Jul 16th, 2012
Revised Aug 01th, 2012
Accepted Sept 02th, 2012

Keyword:

Multilayer Perceptron
Decision Trees
IF then Rules
J48 algorithm
Black Box Nature

ABSTRACT

Artificial neural networks (ANN) are very efficient in solving various kinds of problems But Lack of explanation capability (Black box nature of Neural Networks) is one of the most important reasons why artificial neural networks do not get necessary interest in some parts of industry. In this work artificial neural networks first trained and then combined with decision trees in order to fetch knowledge learnt in the training process. After successful training, knowledge is extracted from these trained neural networks using decision trees in the forms of IF THEN Rules which we can easily understand as compare to direct neural network outputs. We use decision trees to train on the results set of trained neural network and compare the performance of neural networks and decision trees in knowledge extraction from neural networks. Weka machine learning simulator with version 3.7.5 is used for research purpose. The experimental study is done on bank customers' data which have 12 attributes and 600 instances. The results study show that although neural networks takes much time in training and testing but are more accurate in classification then decision trees.

Copyright @ 2012 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Koushal Kumar
Department of Computer Science and Engineering,
Lovely Professional University,
Jalandhar, Punjab, India
Email id: kaushal_kumar302@yahoo.com

1. INTRODUCTION

Artificial neural networks (ANN) have been developed as generalizations of mathematical models of biological nervous systems. A first wave of interest in neural networks emerged after the introduction of simplified neurons by McCulloch and Pitts (1943) also known as connectionist models. An Artificial Neural Network is a network of collections of very simple processors ("Neurons") each possibly having a (small amount of) local memory. The units operate only on their local data and on the inputs they receive via the connections or links which are unidirectional [1].A network unit has a rule for summing the signals coming in and a rule for calculating an output signal that is then, sent to other network units. According to Callen the rules for calculating the output is known as the activation function [2]. A neural network has three layers in its structure. First layer is input layer which is directly interact with external worlds; second layer is of hidden unit where computation is done according to function provided, the last layer is output layer from where we get output. Knowledge in neural networks is stored as synaptic weights between neurons. The network propagates the input data from layer to layer until the output data is generated. If the networks is multilayer perceptron with Backpropogation algorithm and the output is different from the desire output, then an error is calculated and propagated backwards through the network. The synaptic weights are modified as the error is propagated [3].

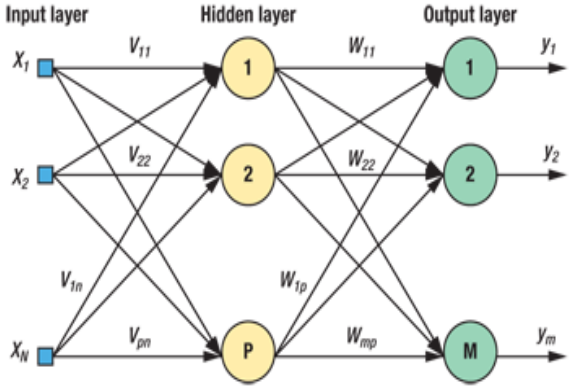

Figure 1 Feed-forward multilayer perceptron

However all we know Artificial Neural networks have many applications in different branches of science and engineering and are used in many problem solving and optimization problems. However the major problem with Neural Networks is that decision given by Neural Networks is Difficult to understand by human being. This is because the knowledge in the Neural Networks is stored as real valued parameters (weights and biases) of the networks [4].Their biggest weakness is that the knowledge they acquire is represented in a form not understandable to humans. Researchers tried to address this problem by extracting rules from trained Neural Networks. Even for an ANN with only single hidden layer, it is generally impossible to explain why a particular pattern is classified as a member of one class and another pattern as a member of another class, due to the complexity of the Network [5]. Decision trees can be easily represented in the form of IF THEN RULES and hence extracting decision trees are probably one of the best methods of interpreting a neural network [6].

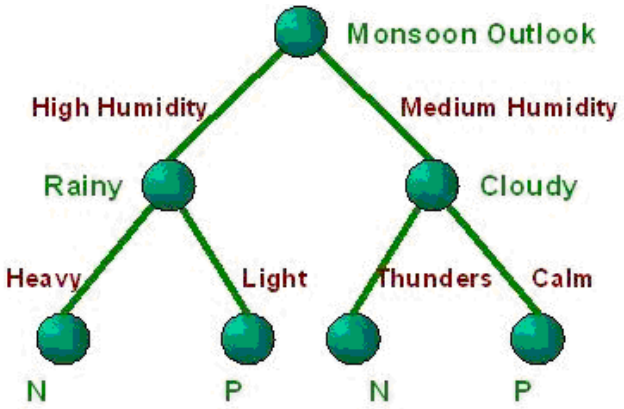

Figure 2 Simple Decision Tree Describing Weather

Figure 2 showing decision trees classifying weather of day at any instant of time. Weather may be of high humidity or it might me of low humidity. Similarly day can be cloudy or rainy etc. As we know lack of explanation capability is one of the most important reasons why artificial neural networks do not get necessary interest in some part of industry. So in this research we work on the problem of overcoming the black box nature of neural networks by extracting simple rules and find the best rule extraction algorithm in accuracy and time parameters. The rules we extract are easy to understand by human beings so provide explanation capability. This paper is thus organized as in section 1 we briefly explained topic of research and problem statement, in section 2 we have explained the rules extraction benefits and its different categories, in section 3 we have provides literature review of our research, section 4 and 5 contain information about data set used ,simulators used in research and training and testing of neural networks, in section 6 we have explained experimental results and discussion, in section 7 we conclude the paper and last section 8 contain our future works.

2. Rule Extraction from Neural Networks

Although neural networks are known as robust classifiers. Artificial Neural Networks (ANNs) are used in many disciplines to solve pattern classification problems. While the accuracy of an ANN's classification of new data is generally accurate, the inherent reasons for that classification are hidden, i.e. a trained neural networks act like black box and difficult to interpret for a human being. That is why ANN's have little use in safety related applications such as medical domain. The black box nature of ANNs is due to the fact that generally ANNs are very complex [7] [8]. So to extract the knowledge behind the neural networks decision making process some techniques are used called Rule extraction. The aim of Rule extraction is to reduce the complexity of an ANN into more easily understood symbolic form. According to the taxonomy presented by Andrews et al, rule extraction algorithms can be roughly categorized into three categories, namely the decompositional, pedagogical eclectic Algorithms [9].

2.1 Rules Extraction Categories

I. Decompositional Approach: This approach is also called local method. Decompositional or local methods extract rules from level of individual, hidden and output units within the trained neural network. Decompositional approaches then typically treat the hidden units as threshold units. The network is decomposed in small networks (one unit). The rules extracted from these small networks are combined to form a global relationship. In decompositional techniques, rules are first extracted at the individual (hidden and output) unit level within the ANN solution. These subsets of rules are then aggregated to form global relationships. The drawback of decompositional approach is more time consuming and computationally more complex [10] [11][12].

II. Pedagogical Approach: The pedagogical or global methods see a trained network as a black box. The methods of this approach extract knowledge that characterize the output classes directly from the inputs. In this approach extraction is done by mapping inputs directly to outputs. The method developed by Saito and Nakano search for combinations of input values which activate each output unit. The pedagogical approach is faster then decompostional approach. One problem with this method is that the size of the search space can grow exponentially with the number of input values. To deal with this problem, the authors used two heuristics that limited the search space. Other limitation is that this method can only be used with discrete-valued features. Because Pedagogical algorithms tend to have exponential complexity and do not scale up to high dimensions of inputs it has very limited and specific use[13].

III. Eclectic Method: Eclectic methods combine the previous approaches. They analyze the ANN at the individual unit level but also extract rules at the global level. One example of this approach is the method proposed by Tickle et al. (called DEDEC.) DEDEC extracts if then rules from MLP networks trained with back propagation algorithm. DEDEC combines the knowledge embedded on the network's weight and rules extracted by a symbolic algorithm [12][14].

3. Review of Literature

There is continuous research in the area of rules extraction from neural networks. There is lot of work has been done on different kinds of rules extraction like fuzzy rules, if then rules etc. M. Fuller Christie et al in (2011) In this paper they describe a process of Knowledge management by extracting IF THEN RULES from Neural Networks. They also describe the techniques of extracting rules. In the end of paper they discus about the parameters of rules extraction. Ex Tree algorithm Dancey Darren et al in (2010) The Ex Tree is an algorithm for extracting Decision trees from trained neural networks. It is of the pedagogical approach of rule extraction. Ex tree uses Craven's querying and sampling method but unlike Craven Trepan which uses M OF N based splits, Ex Tree uses standard splitting tests like CART and C4.5. The standard decision trees algorithms have a disadvantage that the splitting of node is based upon the fewer and fewer instances and as the trees grows downwards. Therefore, the splitting tests that are near the bottom of trees are poorly chosen because there is less data to chosen. Ex trees remove this problem by generating new instances then querying the artificial neural networks (which act as oracle) with newly created instances. G.R Weckman et al in (2009) Presents a paper on knowledge extraction from neural black box in ecological monitoring. They simply try to overcome the black box nature of neural networks on ecological monitoring system. They use the concepts of sensitivity analysis on data and leave the data which is not so much important in giving outputs. Taylor J Brian et al in (2009) they applied a formal technique for rule extraction. The technique consists of discrete math and formal logic for specification and design process of neural networks. In this paper they present different areas where rule extraction from neural networks techniques can be apply.

REANN ALGORITHM: (Kamruzzaman S.M et. al in 2006) developed this algorithm. A three-phase training algorithm REANN is proposed for Backpropogation learning. In the first phase, appropriate network architecture is determined using constructive and pruning algorithm. In the second phase, the continuous activation values of the hidden nodes are discredited by using an efficient heuristic clustering algorithm. In last phase rules are extracted. YAO J.T in (2006) Present an idea of utilizing the rules extracted from neural network in efficient ways. Types of rules that we can extract is also mention in this paper. Descriptive neural networks are obtained from previously trained neural networks. Eclectic et al in 1999 comes with new approach called Eclectic approach [18] [19].

It is hybrid in nature it means it is a combination of decompositional and pedagogical approaches. RULENEG ALGORITHM (E. Pop et al in 1999) developed this algorithm is based upon pedagogical approach of rule extraction. This algorithm can be applied to arbitrary network without bothering about the types of learning used in training the networks. The method extracts conjunctive rules only.

The main drawback of this algorithm is that it is based only on binary values. Here is a simple example for explaining how the algorithm works: If the input is (1 0 0) and is classified as C by NN, the first feature is negated. The new instance will be (0 0 0). The new instance is passed through NN and its class is found. If it is not C, the rule will be if f1 then C. This is repeated for all the features. If the new instance with negated second feature is not classified as C the rule will become if f1 AND not f2 then C. If the new instance with negated third feature is classified as C, the final rule will be if f1 AND not f2 then C. If an instance like (1 0 1) is tested, it will be Classified by the rule found previously. Andrews et. al in 1996 classifies rules extraction approaches into three categories according to view taken by algorithms of the underlying network topology. It can be decompositional (fu, 1991; Towel` and Shavlik, 1993), Pedagogical (Craven and Shavlik, 1996a) [15][16].

4. Data set and Tool Used

Data sources of my research works is Secondary in nature. For the research purpose the data that I am using is based upon bank customers. The data contains 12 attributes of bank customers and 600 instances. Before using the data for our research we have to normalize it into normal forms for getting better results. The tool we used in this research work WEKA. WEKA is abbreviation of Waikato Environment for Knowledge Analysis. It is a popular suite of machine learning software written in Java, developed at the University of Waikato. WEKA is free software available under the GNU General Public License MATLAB is another tool used for completing our research work.

5. Training and Testing of Feed Forward Neural network

We used feed forward networks with tan sigmoid activation function in its first hidden layer and purelin function in second hidden layer. The maximum numbers of neurons in the hidden layers are 10. The error signals are used to calculate the weight updates which represent knowledge learnt in the networks. The performance of Backpropogation algorithm can be improved by adding a momentum term [20]. The error in back propagation algorithm is minimized by using formula.

$$E = \frac{1}{2}\sum_{i=1}^{n}(t_i - y_i)^2$$

Where n is number of epochs, ti is desired target value associated with ith epoch and yi is output of the network. To train the network with minimum possibility of error we adjust the weights of the network [17]

6. Experimental Results and Discussion

For training and testing of neural networks we used feed forward neural networks with back propagation algorithm as a training algorithm. The data set is divided into two parts one part for training and another part for testing. We used cross validation method in which 70% of the whole data is used for training the neural networks, 15% is used for testing and remaining 15% is used for validation.

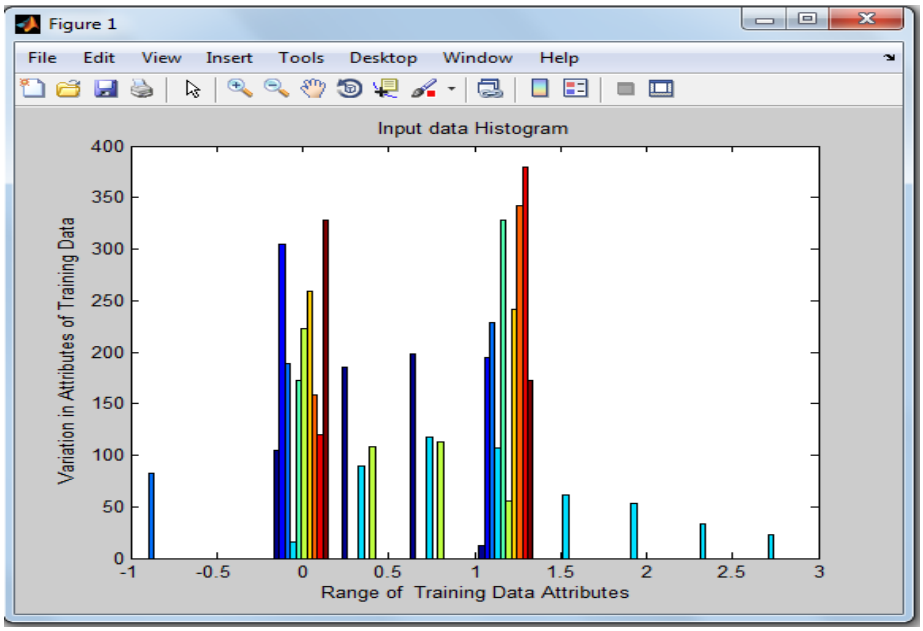

Figure 3: Histogram formation for training data

The Figure 3 shows the histogram formed after loading training set data into Matlab workspace. The different building blocks with different colors showing variation in the range of attributes. The 'X' axis showing range of training data attributes and the 'Y' axis showing variation in attributes of training data set.

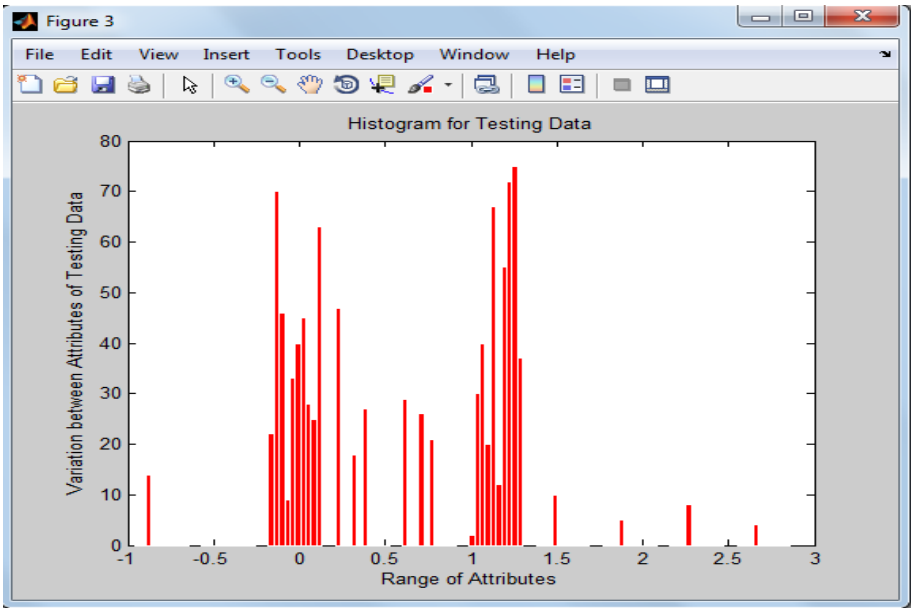

Figure 4: Histogram formation from target data

The Figure 4 showing histogram formed after loading testing set in the workspace of Matlab. The horizontal axis shows range of attributes and vertical axis shows variation between attributes.

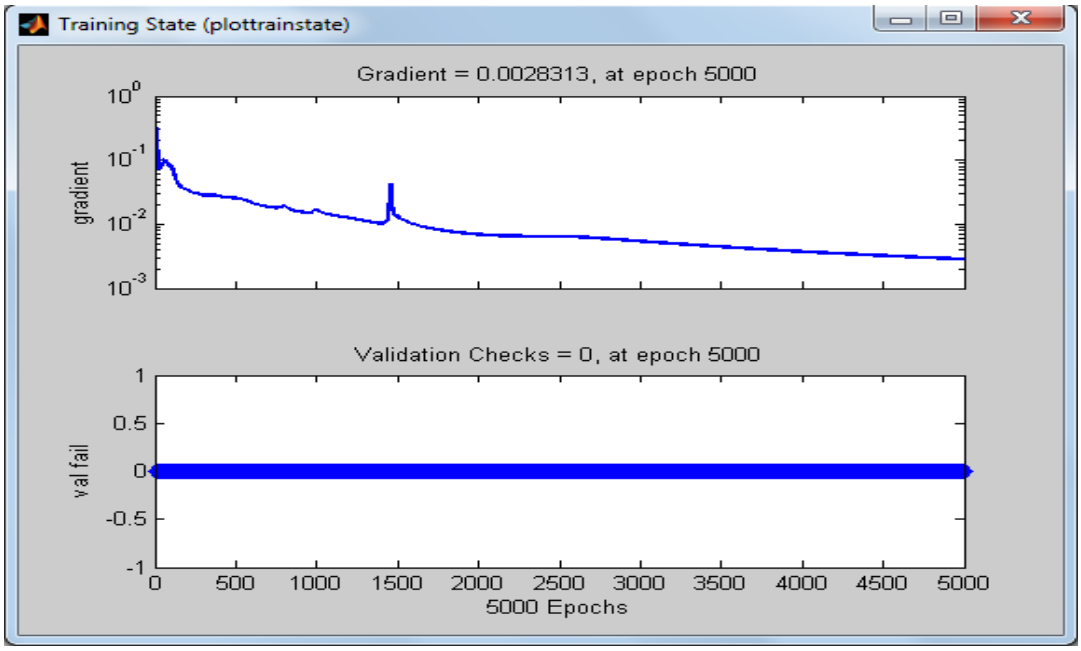

Figure 5: Training state or plot train state diagram

The Figure 5 shows variation in gradient coefficient with respect to number of epochs. The final value of gradient coefficient at epoch number 5000 is 0.0028313 which is approximate near to zero. Minimum the value of gradient coefficient better will be training and testing of networks. From figure it can be seen that gradient value goes on decreasing with increase in number of epochs.

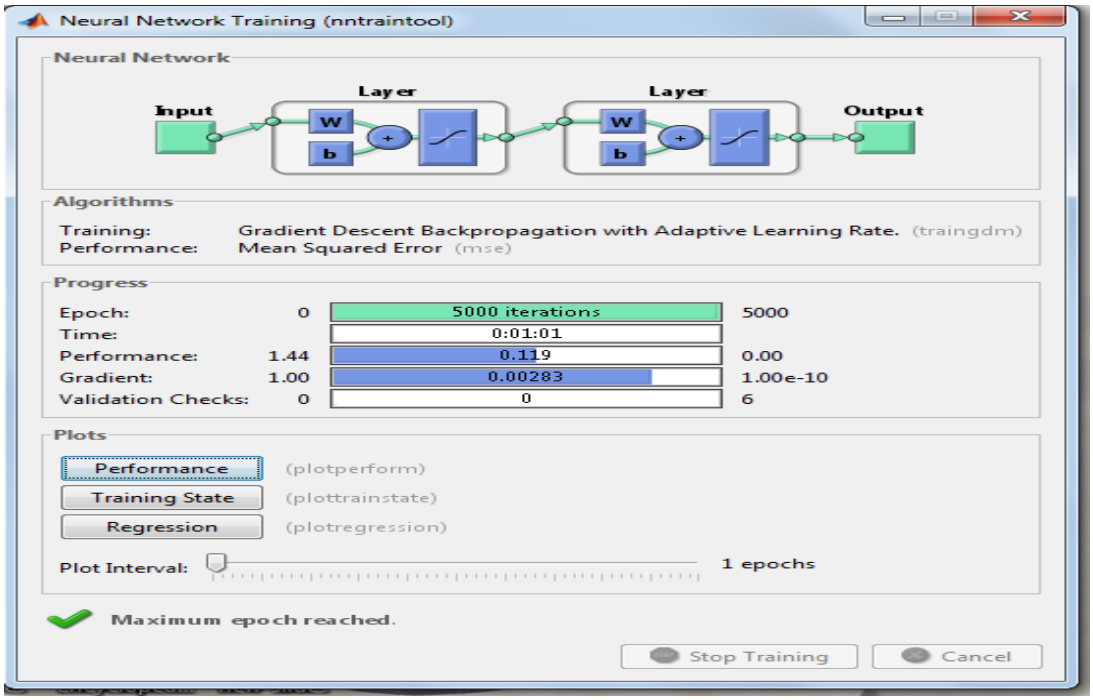

Figure 6: Neural Networks Training & Testing

The Figure 6 shows training of neural networks windows with two hidden layers for training. The total numbers of epochs used are 5000. Further figures shows time, performance, gradient and validation checks parameters used for training and testing of feed forward neural networks

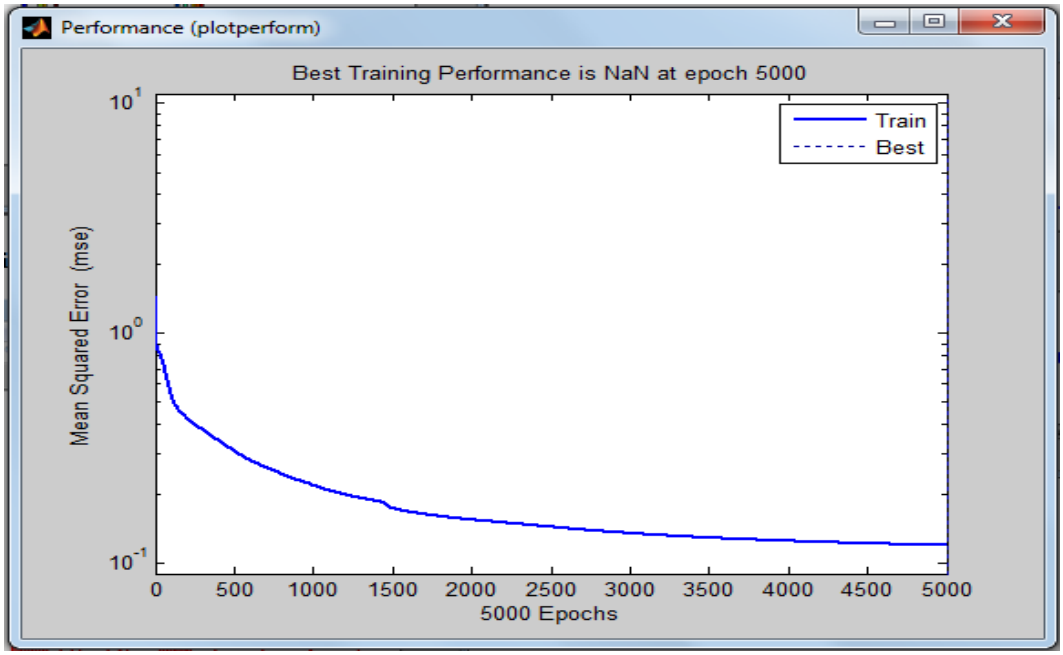

Figure 7: Mean square Error vs. Epochs

The Figure 7 shows variation in mean square error with respect to epochs. It is clear from figure that as we keep on increasing the numbers of epochs for training and testing the error rate keeps on decreasing. After training and testing of the data we used its output values for extracting decision trees for better understanding of results.

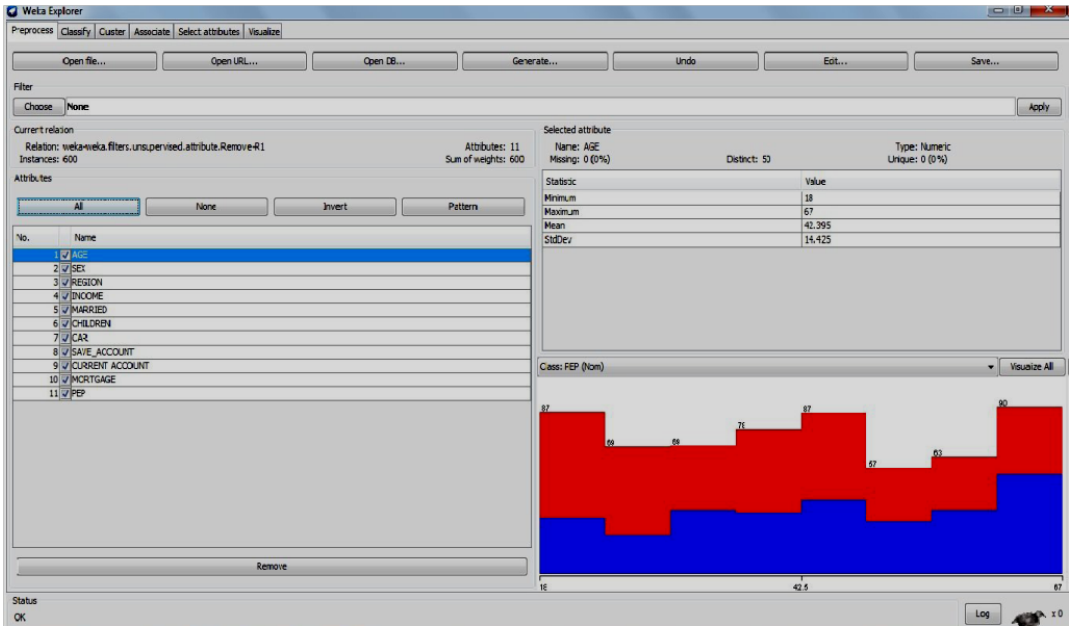

Figure 8: Output data is load in Weka simulator

The Figure 8 shows attributes name which we get after training neural networks. Right side at the bottom of window shows variation in the outputs. For extracting decision trees from neural networks we uses J48 decision trees algorithm and compare its two versions pruned decision trees and unpruned decision trees performance. After applying J48 decision trees algorithm we see that pruned tree output which is approximate 90% is more then unpruned tree which is 86.33%. Similarly it is easily noticeable that unpruned

tree is much more complex then pruned tree. In pruned tree the numbers of leaves of the tree are 15 and size of tree is 29 while in unpruned tree the numbers of leaves of the tress are 64 which are much more then pruned tree and size of tree is 113. The reason for lesser complexity in J48 pruned tree is that pruning helps in reducing complexity by removing those attributes which does not much importance in the data.

Table 1: Results from Unpruned Tree

Parameters	Values
Total instances	600
Correctly classified	518
Incorrectly classified	61
Kappa statistic	0.7942
Mean absolute error	0.167
Root mean square error	0.3471
Time taken to build model	0.1 sec
Numbers of leaves	64
Accuracy	86.33%

Table 2: Pruned Tree Result

Parameters	Values
Total instances	600
Correctly classified	539
Incorrectly classified	82
Kappa statistic	0.724
Mean absolute error	0.161
Root mean square error	0.305
Time taken to build model	0.03 sec
Numbers of leaves	15
Accuracy	89.33%

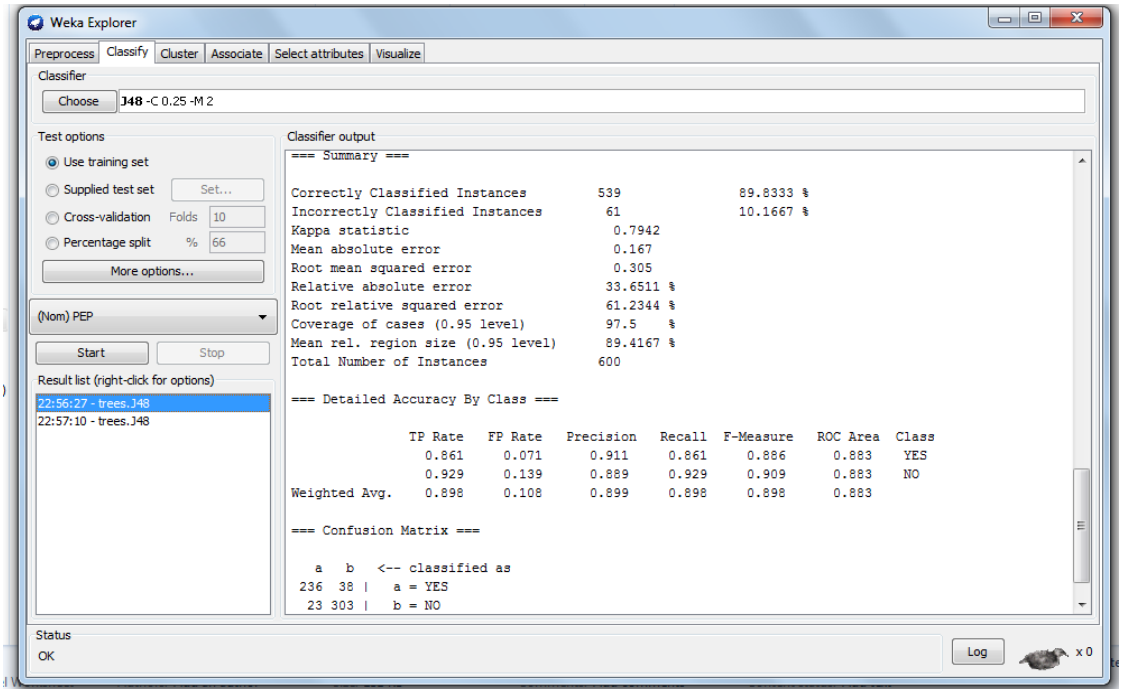

Figure 9: Results of pruned J48 decision trees

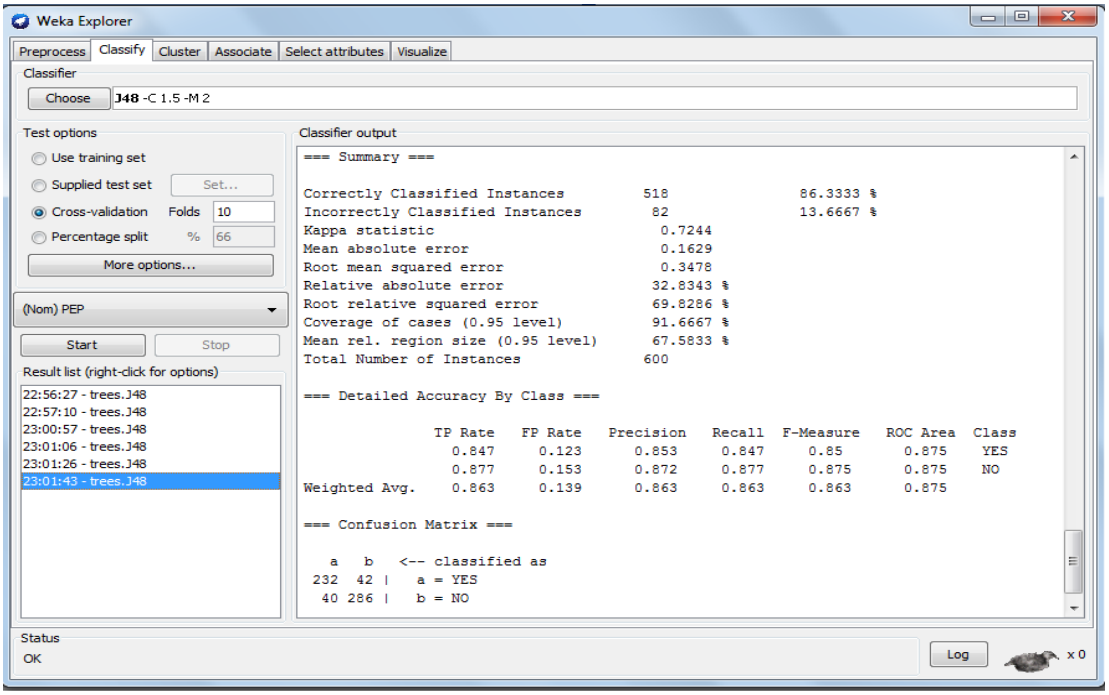

Figure 10: Results of unpruned J48 decision trees

When we compare the results of figure 9 and figure 10 we can see that pruned tree output which is approximate 90% is more then unpruned tree which is 86.33%. Similarly it is easily noticeable that unpruned tree is much more complex then pruned tree. In pruned tree the numbers of leaves of the tree are 15 and size of tree is 29 while in unpruned tree the numbers of leaves of the tress are 64 which are much more then pruned tree and size of tree is 113. The Following Rule Set Is Obtained From the above Decision Tree.

I. Applying Remove redundancy conditions
In this step, we will remove the more general conditions which appear in the same rule with more specific conditions. For example
IF Children ≥1 then Marital status = YES
Applying similar approach the following set of rules are extracted IF Children ≥ 1 AND Children >2AND Children >3 THEN Marital status =YES

We can see that the condition Children ≥ 1 is more Specific than Children >3 and Children > 2. So we Remove all such conditions. The final rule will be
Rule 1:

a) IF Current_act = NO AND Age ≤ 48.0 AND Sex = FEMALE AND Children ≤ 0 THEN Region = Town
b) IF AGE > 48.0 AND Region Suburban AND
Current_act = NO then Pep = NO
c) IF Children ≤ AND Mortgage = NO AND Age ≤THEN Region INNER_CITY
d) IF Age ≤AND Region TOWN AND Mortgage! = NO THEN Children =NO

II. For every pair decision trees Remove redundancy rules. For example
Rule 1: IF Age ≤ AND Salary ≤ 3500 AND Pep =NO THEN Mortage = YES
Rule 2: IF Age ≤ 50 AND Salary ≤ 3500 AND Pep = NO THEN Mortage = YES New Rule: IF Age ≤ 50 AND Salary ≤ 3500 AND Pep = NO THEN Mortage =YES
Rule 3: IF Children > 2 AND Region TOWN AND Age> 40 THEN Save act = YES

III. Remove more specific rules. The rules with a condition set which is a superset of another rule should be removed.
For example

Rule 1: IF Age ≤ 60 AND Region = Rural AND saving act = YES THEN Pep = NO
Rule 2: IF Age <= 60 AND Children <= 1 AND Region =Rural AND saving act = YES THEN Pep= NO
Rule 3: IF Region = Rural AND saving act =YES
THEN Pep = NO
New Rule: IF Region = Rural AND saving _ act =YES THEN Pep = NO
Rule 4: Children = 0 and Sex = FEMALE AND Region = SUBURBAN AND INNER_CITY THEN
Save act = YES

IV. Divide range of conditions. The rules of different branches with the same attribute which has overlapped range should be divided into several parts.
For example:
Rule 1: IF Marital status =Married AND Salary > 20000 THEN Children= YES
Rule 2: IF Marital status=Married AND Salary < 35000 THEN Children = YES
New Rule 1: IF Marital status = Married AND Salary ≤ 35000 THEN Children= YES
New Rule 2: IF Marital status= Married AND Salary ≤ 20000 THEN Children = YES

Comparison of J48 algorithm with others classifiers: For measuring the performance and accuracy of J48 algorithm we compare its output with different classification algorithms among them Best First Decision Tree (BFT), Logistic Model Tree (LMT), J48 Graft Tree (J48GT) and J48 Pruned Tree (J48PT). As different algorithms follow different approaches for solving problems thus the trees obtained by the different methods will be different according to kind of approach it is following. Essentially, this difference becomes apparent in the tree's complexity and its precision. The classification methods were applied to the set of 600 instances and 12 variables. The results of some of the most relevant trees, using 10-fold cross validation, are shown in Table 3

Table 3: Comparison of J48 with others classifiers

Classifier	True positive Rate(TP rate)	False Positive Rate(FP rate)	Correct (%)	Incorrect (%)	Time taken to built model (sec)
BFT	0.883	0.121	85.56	15.44	0.92
LMT	0.882	0.125	86.58	14.42	1.48
J48GT	0.898	0.108	87.22	13.68	0.05
J48PT	0.857	0.122	89.22	11.78	0.02

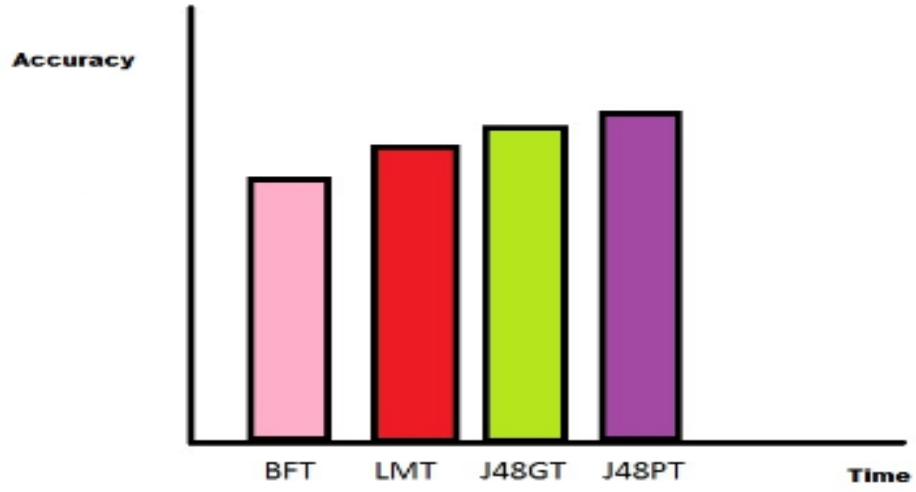

Figure 11: Graphical comparisons of algorithms

Figure 11 show that algorithm J48PT algorithm is best among all others algorithm in point of accuracy and time taken to build model. The pruning algorithm used with simple J48 algorithm gives its more accuracy and less complexity as compare to others algorithms.

7. Conclusion

This work is an attempted to open up these black boxes nature of artificial neural networks by extracting symbolic rules from it in the form of decision trees. As we know neural networks are very efficient in problem solving and has wide acceptance in many areas. Generalization and learning from surrounding are the only two factors which makes artificial neural networks so powerful computation tool. However one major problem with artificial neural networks is its weakness of black box nature, which makes it unable to explain the internal processing. Their most important weakness is that the knowledge they acquire is represented in a form not understandable to humans. Understandability problem of Neural Networks can be solved by extracting Decision Rules or Decision Trees from the trained network. So in this works we extract the symbolic rules in the form of decision trees which are quite easy to understand. Next we compare the decision trees algorithm J48 with some of its competitor algorithms. We used the J48 algorithm and compare it with Best first decision tree algorithm (BFT) J48Graft which is extension of early J48 algorithm, Logistic model tree (LMT). We come with conclusion that j48Graft algorithm is better then all others algorithms and it is more accurate then other algorithms.

8. Future works

As in this works neural network is trained with back propagation algorithm in the future work genetic algorithm can be used for training the neural networks which is more reliable then back propagation algorithm. In this work feed forward neural networks is used for rule extraction in the similar way recurrent neural networks can be used which can be another area of research.

REFERENCES

[1] "Ajith Abraham." Artificial Neural Networks" Oklahoma State University, Stillwater OK, USA 2006
[2] Zhi-Hua Zhou, "Rule Extraction Using Neural Networks or For Neural Networks" National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China 2007
[3] Fiona Nielsen, "Neural Networks algorithms and applications" Neils Brock Business College, Dec 2001.
[4] R. Setiono and K. Leow, FERNN: "An algorithm for fast extraction of rules from neural Networks". Appl. Intel, 12 (1-2), pp.15-25, Nov. 2000
[5] I. Taha et. all "Three techniques for extracting rules from Feed forward networks" intelligent engineering systems through artificial neural networks vol 6 pp 23-28, in 2002
[6] Christie M. Fuller and Rick L. Wilson "Extracting Knowledge from Neural Networks" Oklahoma State University, USA 2010.
[7] W. Duch, R. Adamczak, K. Grabczwski, "A new methodology of extraction, optimization and application of crisp and fuzzy logical rules". IEEE Transactions on Neural Networks, Vol 11, no 2 in 2000
[8] Dancey Darren and Mclean Dave "Decision tree extraction from trained neural networks" Intelligent Systems Group Department of Computing and Mathematics, Manchester Metropolitan University, 2010.
[9] Fu. L.M "Extracting rules from Neural Networks by pruning and hidden units Splitting." Department of information system and computer sciences, University of Singapore, 2000.
[10] G.R Weckman, D.F Millie C Ganduri, Rangwala , W.Young , M. Rinder "Knowledge Extarction from neural network"Black Box In Ecological Monitoring" Journel of industrial and system engineering.vol 3 no 1 pp 38-55 springer 2009.
[11] Guerreiro, Joao and Trigueiros, Duarte. "A Unified Approach to the Extraction of Rules from Artificial Neural Networks and Support Vector Machines". Springer publishers In ADMA 2010
[12] Gallant S.I "Connection expert systems and Communication" department of computer sciences Florida, 1999
[13] J.R Quinlan, (2002) C4.5"Programs in Machine learning.San Mateo, CA: Morgan
[14] Kamruzzaman. S.M "REx: An Efficient Rule Generator" Department of Computer Science and Engineering Manarat International University, Dhaka, Bangladesh, 2010
[15] M. Fuller Christie, Wilson. Rick L"Extracting knowledge from Neural Networks" Oklahoma State University, USA, 2011
[16] R. Andrews, J. Diederich, and A. B Tickle "Survey and critique of techniques for extracting Rules", 2004
[17] Rohitash Chandra, Kaylash Chaudhary and Akshay Kumar. "The Combination and comparison of neural networks with decision trees for wine classification". School of sciences and technology, University of Fiji, in 2007
[18] Setiono R, Leow. K "FERNN an algorithm for fast extraction of rules from neural Networks. Appl .Intel, 12 (1-2), 15-25, 2000
[19] Taylor J. Brain, Darrah A Marjorie "Rule Extraction as a Formal Method for the Verification and Validation of Neural Networks". Institute for Scientific Research, 2009
[20] YAO J.T "Knowledge Extracted from Trained Neural Networks What's next? Department of computer Sciences University of Ragina Canada., 2006

BIBLIOGRAPHY OF AUTHOR

Mr. Koushal Kumar Has done his M.Tech degree in Computer Science and Engineering from Lovely Professional University, Jalandhar, India. He obtained his B.S.C and M.S.C in computer science from D.A.V College Amritsar Punjab. His area of research interests lies in Artificial Neural Networks, Soft computing, Computer Networks, Grid Computing, and data base management systems. He is author of a book with Title "Rules extraction From Trained neural Networks using Decision Trees" with ISBN No 978-3659195754 and he has published numbers of papers in various reputed international computer science journals.