ISSN:0975-9646

Koushal Kumar et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2) , 2012,3812-3815

Extracting Explanation from Artificial Neural Networks

Koushal Kumar,Gour Sundar Mitra Thakur
Department of Computer Science & Engineering
Lovely Professional University, Jalandhar (Punjab), India

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.**

Keywords--: Symbolic Interpretation of Neural Networks, Rules Extraction, Decision Trees, If Then Rules

I. INTRODUCTION

Artificial Neural Networks (ANNs) are used in many applications to solve various kinds of 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 [1]. 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 [2]. 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 [16]. Pruning of the tree is used to prevent over-fitting of the data. This pruning mechanism maximizes information gain by getting rid of nodes that do not contribute much to information gain.

II. RULES EXTRACTION METHODS

A) Decompostional 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. The rules extracted from these small networks are combined to form a global relationship. The earliest decompositional rule extraction method is the KT algorithm developed by Fu [3].

B) **Pedagogical Approach:** This approach treats the network as a 'black box' and make no attempt to disassemble its architecture to examine how it works; instead this approach extracts rules by examining the relationship between the inputs and outputs [4]. The pedagogical approach is faster then decompositional approach. One problem with this method is that the size of the search space can grow exponentially with the number of input values.The rule-extraction-as-learning technique of Craven and Shavlik (1994) is example of this technique.

C. **Eclectic Approach:** Eclectic approach combines the previous approaches, analyse 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 [5],[13]

III. DATA AND TOOL USED:

DATA: In this work the data we used is purely real time data. It is combination of primary and secondary data. Data is based upon bank customer account. Data is divided into training set and testing set. We used different proposition of training and testing data to produce better results.

TOOL: 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 [6]. MATLAB is another tool used for completing our research work.

IV. RESEARCH METHODOLOGY

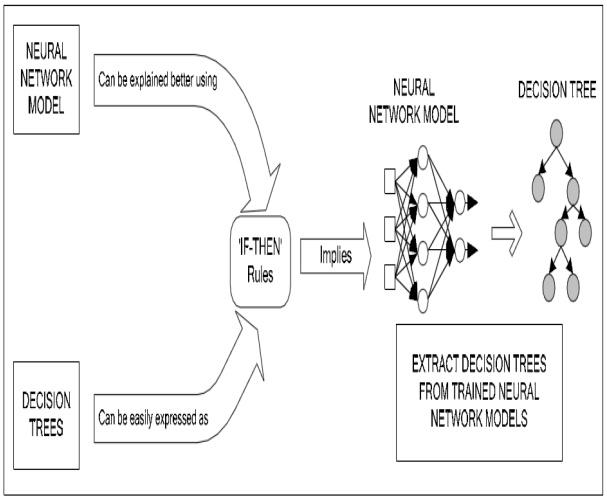

Figure 1. Extracting decision trees from neural networks

As in figure 1 above it can be seen that both decision trees and neural networks can be easily converted into **IF THEN Rules** or we can simply convert neural networks into decision trees. We can use any neural networks architecture like feed forward network, radial basis Function networks, support vector machine, recurrent networks etc[7].

Combining neural networks with decision trees:

The goal of knowledge extraction from ANN's is to find the knowledge stored in the networks weights in symbolic form. One main concern is the fidelity of the extraction process, i.e. how accurately the extracted knowledge corresponds to knowledge stored in networks. There are two main approaches for knowledge extraction from trained neural networks.

A. Extraction of if then rules by clustering the activation values of hidden state neurons.

B. The application of machine learning methods such as decision trees on the observation of input outputs mappings of trained networks when presented with data.

V: Training And Testing Of Feed Forward Neural Network

Multilayer Perceptron:

A multilayer perceptron in weka is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate output. It is a modification of the standard linear perceptron in that it uses three or more layers of neurons (nodes) with nonlinear activation functions, and is more powerful than the perceptron in that it can distinguish data that is not linearly separable, or separable by a hyper-plane [8].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 [9], [10].The error in back propagation algorithm is minimised by using formula.

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

Where n=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[11].

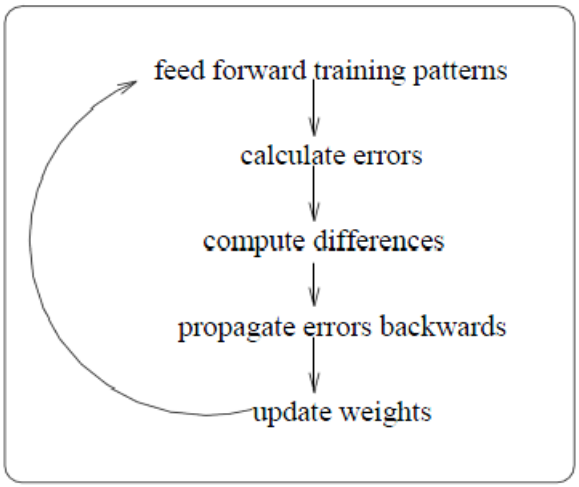

Figure 2: The back-propagation Neural network Epoch [12]

Table1: Defining the Network Parameters

Parameter	Values
Number of Training Data	600
Number of Testing Data	100
Number of Hidden Layers	2
Learning Rate	0.3
Momentum	0.2
Validation Threshold	20
Total no of Epochs	250
Error Per Epoch	0.019
Accuracy	98.6577 %

The above table shows maximum accuracy obtained during training of multilayer perceptron with 10 cross validation. For the cross validation purpose we divide 70% data for training, 15% data for validation and 15% data for testing of networks.

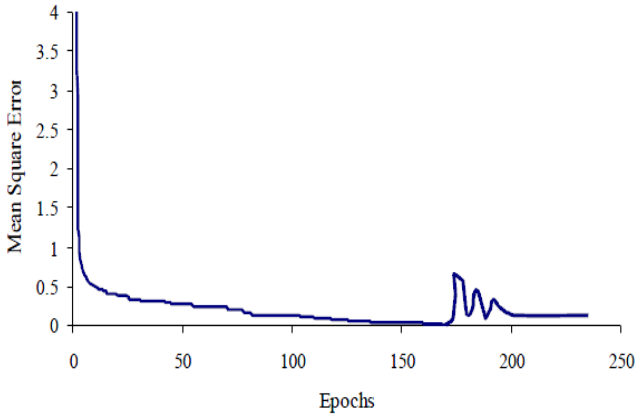

Figure 3: Errors vs. Epochs in 10 folds Cross Validation

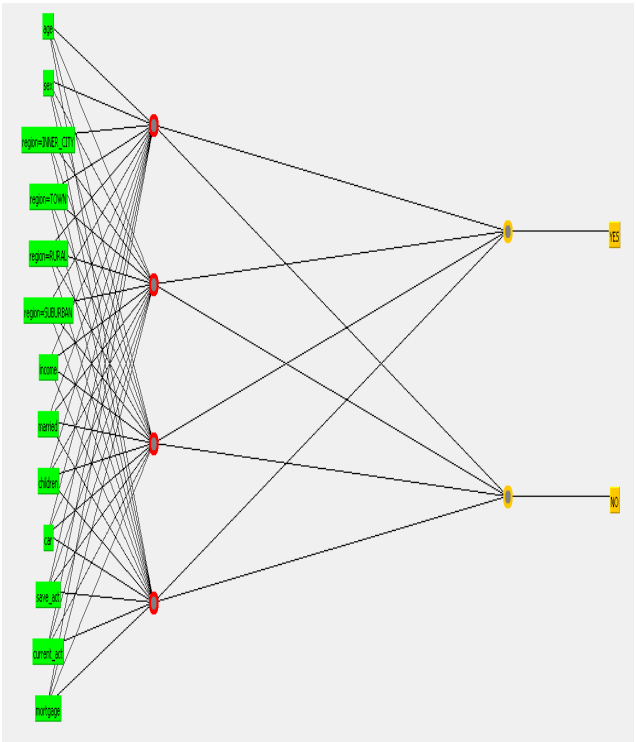

Figure 4: Multilayer Perceptron after classification

VI: EXTRACTION OF KNOWLEDGE FROM NEURAL NETWORKS IN THE FORM OF DECISION TREES

Decision Tree*:* Decision trees are machine learning tool for building a tree structure from a training dataset. A Decision tree learns by starting at the root node and select the best attributes which splits the training data[13]. Compared to neural networks they can explain how they arrive to a particular solution [14,15]. We will use decision trees to extract rules from the trained neural networks. We extracted decision trees from trained neural networks using j48 algorithm. We used the attribute and classification of the 75% training data and the attributes of the remaining 30% test data. A typical decision tree extracted from experiment No. 7 in table 2 as shown in figure 3. We show this particular decision tree because experiment No.7 has the best generalization performance from all experiment in table 2.

Table 2: Decision Trees for classification on the results set o Network

Exp No	Training Performance	Generalization Performance	Time taken to build model in sec	Number of leaves and size of tree
1	90.604%	86.576%	0.2	(7,13)
2	90.70%	86.5%	0.2	(8,15)
3	92.68%	86.5%	0.3	(8,15)
4	92.6%	86.7%	0.3	(9,15)
5	93.4%	86.5%	0.2	(9,15)
6	94.4%	86.8%	0.4	(7,12)
7	95.6%	90.60%	0.2	(7,10)
8	95.6%	88.3%	0.3	(7,9)
9	94.5%	87.6%	0.3	(5,6)
10	95.6%	89.5%	0.4	(6,7)

```
Time taken to build model: 0.02 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances         135               90.604  %
Incorrectly Classified Instances        14                9.396  %
Kappa statistic                          0.8039
Mean absolute error                      0.1603
Root mean squared error                  0.2831
Relative absolute error                 32.9712 %
Root relative squared error             57.4314 %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)      86.2416 %
Total Number of Instances              149

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.839     0.046      0.929     0.839     0.881      0.93     YES
                 0.954     0.161      0.892     0.954     0.922      0.93     NO
Weighted Avg.    0.906     0.113      0.907     0.906     0.905      0.93

=== Confusion Matrix ===

  a  b   <-- classified as
 52 10 |  a = YES
  4 83 |  b = NO
```

Fig 5: Best Performance of decision tree in exp no 7 of table 2 using weka

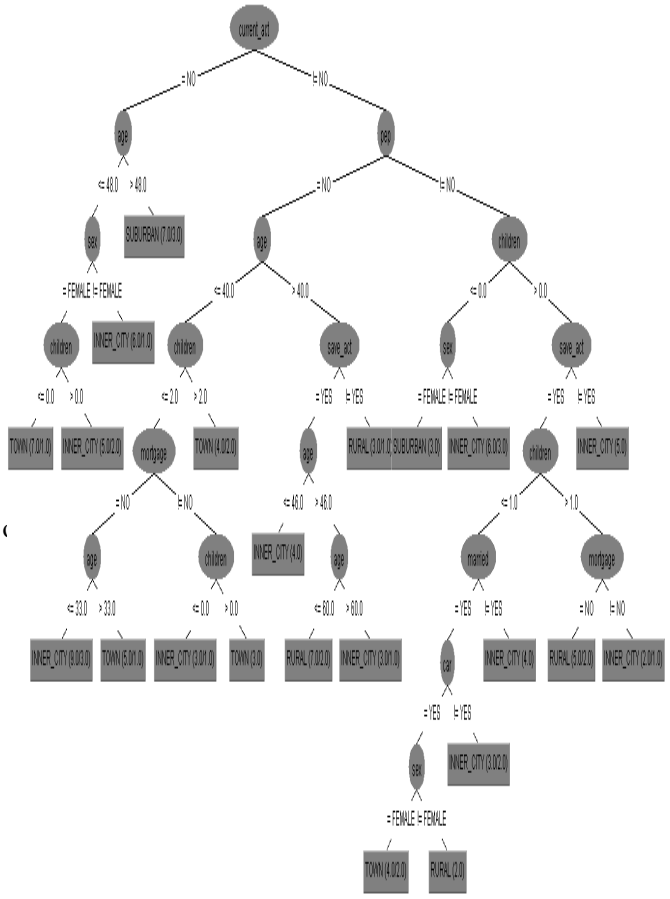

Fig 6**:** The figure show the decision tree extracted from trained neural network in experiment no 7 of table 2.

THE FOLLOWING RULE SET IS OBTAINED FROM THE DECISION TREE OF FIG 6:

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** AND **Children >2**AND **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

IF **Children** ≥1 then **Marital status** = **YES**

Applying similar approach the following set of rules are extracted from fig 5 decision trees

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

Table 3: Performance of decision trees vs. neural networks

Classification Approach	Training Performance	Generalization Performance	Training Time (Epochs)
ANN	93.5%	90.6%	170
Knowledge extract from trained ANN	--	88.9%	--
Decision Tree	91.3%	89.2%	130

CONCLUSION AND FUTURE WORKS:

Neural networks have been successful in the classification of data we have. The prediction accuracy of neural networks can be increased by having more training instances in the dataset. Decision trees have been useful in knowledge extraction from trained neural networks. We have obtained rules for classification of bank customers according to their attributes values. The results show that however neural networks are good in generalization performance of given data set. However they cannot explain how they arrive to a solution. The extracted rules show that three major attributes age, region and mortage has major influence on the data set. The extraction of knowledge from these networks helps us to obtain useful rules which further helps in understanding the results obtain from neural networks. In the future works we can apply soft computing techniques to extract more easily to understand rules.

REFERENCES:

[1] R. Setiono and K. Leow, FERNN: "An algorithm for fast extractin of rules from neural Networks". Appl. Intel, 12 (1-2), pp.15-25, Nov. 2000

[2] I. Taha. And Ghosh., "Three techniques for extracting rules from Feed forward networks" intelligent engineering systems through artificial neural networks vol 6 pp 23-28, in 2002

[3] Li Fu Min. "Rule generation from neural networks", IEEE Transactions on Systems, Man and Cybernetics, Vol. 24 No.8, pp. 1114-1124. 1994

[4] H. Witten LAN; Eibe Frank "Data Mining: Practical machine learning tools and techniques, 2nd Edition". Morgan Kaufmann, San Francisco, 2006

[5] R. Davis, B. G Buchanan, and E. Shortcliff. "Production Rules as a Representation for a Knowledge Based Consultation Program" Artificial Intelligence, vol. 8(1), pp.15-45, in 1998

[6] G.R Weckman, D.F Millie and C. Ganduri, M. Rangwala, W.Young, M. Rinder "Knowledge Extraction from the Neural 'Black Box' in Ecological Monitoring" Journal of Industrial and Systems Engineering Vol. 3, No. 1, pp 38-55 Spring 2009

[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] 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

[9] Quinlan, J.R, C4.5 "Programs for Machine Learning" .California. Morgan Kauffman Publishers, San Mateo, 1993

[10] Lu. H, Setiono.R. "Effective data mining using neural Networks. IEEE Trans Knowledge Data Eng 8 (6), pp. 957-961., Jan 1999

[11] Setiono. Rudy and Liu. Huan "Understanding neural networks via rule extraction," Proceedings of the 14th International Joint Conference on Artificial Intelligence 1995, pp. 480-485.

[12] John G. Carney and P´adraig Cunningham. "The Epoch Interpretation of Learning". Department of Computer Science University of Dublin Trinity College Ireland.1999

[13] S. M Kamruzzaman., REx: "An Efficient Rule Generator" Department of Computer Science and Engineering Manarat International University, Dhaka, Bangladesh.2010

[14] D.Yedjour, H.YEDJOUR and A. Benyettou., "Explaining results of artificial neural networks, Journel of applied scinces vol 2, No 3 2011

[15] Guerreiro, João and Trigueiros, Duarte. "A Unified Approach to the Extraction of Rules from Artificial Neural Networks and Support Vector Machines". Springer publishers In *ADMA 2010* (2010)

[16] Christie M. Fuller and Rick L. Wilson "Extracting Knowledge from Neural Networks" Oklahoma State University, USA 2010