www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 2 February , 2014 Page No. 3852-3863

EFFICIENT DATA MINING FOR MINING CLASSIFICATION USING NEURAL NETWORK

Mrs.Maruthaveni.R, Mrs.Renuka Devi.S.V
Assistant Professor., M.Phil Scholar,
Department of Computer Science, Dr.SNS RCAS, Coimbatore.
dhanuvene1@gmail.com, renukadevisv@gmail.com

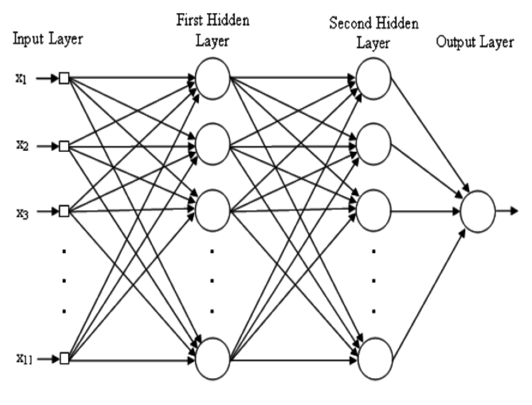
Abstract

One of the data mining problems receiving enormous attention in the database community is classification. Although Artificial Neural Networks (ANNs) have been successfully applied in a wide range of machine learning applications, they are often regarded as "black box", that means predictions cannot be explained. To enhance the explanation of neural network, a novel algorithm is to extract symbolic rules from neural network has been proposed. With the proposed approach, concise symbolic rules with high accuracy can be extracted from the trained neural network. Extracted rules are comparable with other methods in terms of number of rules. The network is first trained to attain the desired accuracy rate. Redundant connection of the network are then removed by a network pruning rule. The effectiveness of the proposed approach is clearly demonstrated by the experimental results on a set of benchmark data mining classification problems.

Index Terms: Data mining, neural network, pruning, symbolic rules, classification, rule extraction, clustering.

1. Introduction

Data mining refers to extracting or mining knowledge from large amounts of data. If we see the data that have been stored in our data warehouses and databases like a mountain, the gems are buried within the mountain. Data mining helps end users to extract useful business information from large databases. Data mining is the process of finding and interpreting the valuable information by using the knowledge of multidisciplinary fields such as artificial intelligence, statistics, machine learning, database management & so on. While the predictive accuracy obtained by artificial neural networks (ANNs) is often higher than that of other methods, it is generally difficult to understand how ANNs arrive at a particular conclusion due to the complexity of the ANN architectures. It is often said that an ANN is practically a "black box". Even for an ANN with only single hidden layer is generally difficult 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. This may cause problems in some cases. To solve this downside, researchers are interested in developing a humanly understandable representation for ANNs.



Architectural graph of the network.

Neural networks learn the classification rules by many passes over the training dataset, so that the learning time of a neural network is usually long. A neural network is usually a layered graph with the output of one node feeding into one or many other nodes in the next layer. The classification process is buried in both the structure of the graph and the weights assigned to the links between the nodes. The use of neural networks in classification is not uncommon in machine learning community. Neural networks give a lower classification error rate than the decision trees but require longer learning time. Our results from applying neural networks to mine classification rule for large databases with the focus on articulating the classification rules represented by neural networks.

ANNs have the ability of distributed information storage, reasoning, parallel processing, and self-organization. It conjointly has the capability of rapid fitting of nonlinear data, thus it will solve several issues which are difficult for other methods. It is high affordability to the noise information with low error rate, and also the ceaselessly advancing and improvement of varied network pruning, training, and rule extraction algorithms, built the application of the ANNs in the data mining increasingly favoured by the overwhelming majority of users. In machine learning and data mining analysis, rule extraction has become progressively vital topic, and a growing variety of researchers and practitioners have applied ANNs for machine learning in a variety of real world applications. An inherent defect of ANNs is that the learned knowledge is masked in a large quantity of connections, that result in the poor transparency of knowledge and poor rationalization ability.

In several applications, it is extremely fascinating to extract symbolic rules from these networks. Not like a set of weights, symbolic rules are often simply taken and verified by human specialist. They will conjointly give new insights into the application problems and the corresponding data. A number of works are available in the literature to explain the functionality of ANNs by extracting rules from trained ANNs. The most downside of existing works is that they confirm the quantity of hidden neurons in neural network manually. Thus, the prediction accuracy and rules extracted from trained network might not be optimum since the performance of neural network is greatly obsessed on their architecture. what is more, rules extracted by existing algorithms do not seem to be easy as a result it is difficult to understand by the users.

The network is 1st trained to attain some needed accuracy rate. Redundant connections of the network square measure then removed by a network pruning rule. The activation values of the hidden nodes within the network square measure analyzed, and classification rules square measure generated mistreatment the results of this analysis. 2 categories of approaches for data processing with ANNs are projected. the primary approach, typically referred to as extraction of rule that involves symbolic models from trained ANNs. The secondary approach is to directly learn straightforward, easy-to-understand networks. data processing mistreatment cropped artificial neural network tree (ANNT). This approach consists of 3 phases: coaching, pruning and rule extraction. It improved the generalization ability of the network and also the variety of rules extracted is reduced. The key technology and ways in which to attain the info mining supported neural networks is researched. the mix of knowledge mining technique and neural network model will greatly improve the potency of knowledge mining techniques, and has been wide used.

2. Analysis Of Existing Work

A number of algorithms for extracting rules from trained ANNs have been developed in the last two decades. Saito and Nakano proposed a medical diagnosis expert system based on a multilayer ANN. They treated the network as a black box and used it only to observe the effects on the network output caused by change of inputs. Two methods for extracting rules from neural network are described by Towell and Shavlik . The first method is the subset algorithm, which searches for subsets of connections to a node whose summed weight exceeds the bias of that node. The most important downside with subset algorithms is that the price of finding all subsets increases as the size of the ANNs increases. The second method, the MofN algorithm, is an improvement of the set methodology that's designed to expressly seek for M-of-N rules from information based mostly ANNs. rather than considering associate degree ANN association, teams of connections square measure checked for his or her contribution to the activation of a node, that is completed by agglomeration the ANN connections.

Liu and Tan planned X2R in, an easy and quick algorithmic rule which will be applied to each numeric and separate knowledge, and generate rules from datasets. It will generate good rules within the sense that the error rate of the principles isn't worse than the inconsistency rate found within the original knowledge. the matter of the principles generated by X2R, square measure order sensitive, i.e., the principles ought to be dismissed in sequence. Liu represented a family of rule generators therein will be wont to extract rules in varied applications. It includes versions which will handle noise in knowledge, manufacture good rules, and might induce order freelance or dependent rules. The basic idea of the algorithm is using first order information in the data to determine shortest sufficient conditions in a pattern that can differentiate the pattern from patterns belonging to other classes.

Setiono presented MofN3, a new method for extracting M-of-N rules from ANNs. The topology of the ANN is the standard three-layered feedforward network. Nodes within the input layer ar connected solely to the nodes within the hidden layer, whereas nodes within the hidden layer are connected to nodes within the output layer. Given a hidden node of a trained neural network with N incoming connections, show however the worth of M will be simply computed. so as to facilitate the method of extracting M-of-N rules, the attributes of the set have binary values –1 or 1. Kamruzzaman and Islam proposed an algorithm, REANN to extract rules from trained ANNs for medical diagnosing issues. This paper investigates the rule extraction method for less than three medical datasets.

Jin and Sendhoff provide an up-to-date yet not necessarily complete review of the existing research on Pareto-based multi-objective machine learning (PMML) algorithms. They illustrate, on three benchmark problems (breast cancer, diabetes, and iris), how will address necessary topics in machine learning, like generating explainable models, model choice for generalization, and ensemble extraction, mistreatment the Pareto-based multi-objective approach. They compare three Pareto-based approaches to the extraction of neural ensembles and indicate that the method by commerce off accuracy and complexness will offer reliable results. Finally, Wang et al. projected a unique formula of regression rules extraction from ANN, which relies on linear intelligent insertion. The linear operate and symbolic rules area unit accustomed the neural network, and also the rules area unit generated by the decision tree.

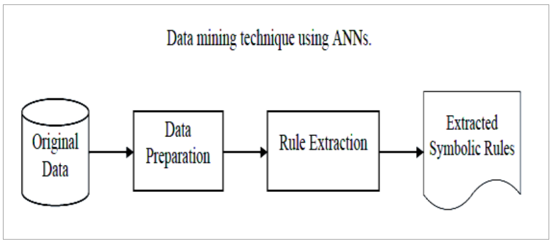
3. Limitation of the existing rule-extraction algorithm

- ✓ Use predefined and stuck variety of hidden nodes that require human experience and previous knowledge of the issues to be resolved,
- ✓ Clustering algorithmic formula used to discretize the output values of hidden nodes are inefficient,
- ✓ Computationally expensive,
- ✓ Could not produce concise rules, and
- ✓ Extracted rules are order sensitive.

To overcome these limitations we've projected a theme for data processing by extracting symbolic rules from trained ANNs.

4. Proposed Data Mining Using ANNs

The data mining method exploitation ANNs with the stress on symbolic rule extraction, the planned data processing theme consists of two steps: data preparation and rule extraction.



4.1 Data Preparation

One must prepare quality information by pre-processing the data. The input to the data mining algorithms is assumed to be distributed, containing incorrect values or no missing wherever all options square measure vital. The real-world data could also be noisy, incomplete, and inconsistent, which might disguise helpful patterns. data preparation could be a method of the first information to form it acceptable a particular data mining technique. data preparation is that the first vital step within the data mining and plays a crucial role within the entire data mining process.

The data mining using ANNs can only handle numerical data. There may be several different kinds of attributes that must be representing input and output attributes. We will now discuss each attribute kind and some common methods to represent such an attribute.

- **Real-valued attributes** square measure sometimes rescaled by some function that maps the value into the range 0…1 or −1…1 in a way that makes a roughly even distribution within that range.

- **Integer-valued attributes** square measure most often handled as if they were real-valued. If the amount of various values is only small, one among the representations used for ordinal attributes may additionally be applicable. Note that often attributes whose values are integer numbers are not extremely integer-valued however are ordinal or cardinal instead. We tent to take into account all integer-valued attributes as real-valued.

- **Ordinal attributes** with *m* different prices are either mapped onto an equidistant scale creating them pseudo-real-valued or are represented by $m - 1$ inputs of that the leftmost *k* have value 1 to represent the *k*-th attribute value whereas all others are 0. A binary code exploitation solely [log2 *m*] inputs may also be used.

- **Nominal attributes** with *m*, an Different values area unit typically either diagrammatic employing a 1-of-m code or a binary code.

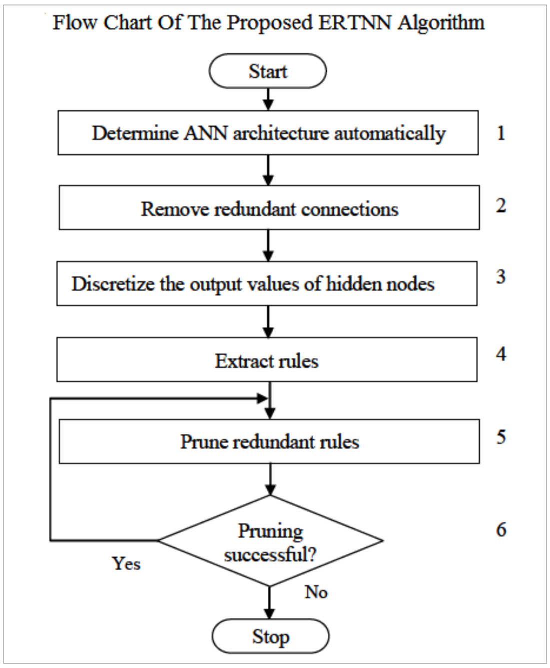
- **Missing attribute** values can be replaced by a fixed value (e.g., the mean of the non-missing values of this attribute) or can be represented explicitly by adding another input for the attribute that is 1 if the attribute value is missing.

4.2 Extracting Rules From A Trained Neural Network (ERTNN- Algorithm)

Extracting symbolic rules from trained ANN is one of the promising areas that are commonly used to explain the functionality of neural network. The aim of this subsection is to introduce a new algorithm, referred to as ERTNN (extraction of symbolic rules from trained ANNs), to extract symbolic rules from trained ANNs. It is difficult to find the explicit relationship between the input tuples and the output tuples. A number of reasons contribute to the difficulty of extracting rules from a pruned network.

First, even with a pruned network, the links may be still too many to express the relationship between an input tuples and its class label in the form of *if . . . then* ... rules. If a network still has *n* input links with binary values, there could be as many as 2, distinct input patterns. The rules could be quite lengthy or complex even for a small *n.*

Second, a standard three-layer feedforward ANN is the basis of the proposed ERTNN algorithm. The hyperbolic tangent function, which may take any worth in the interval [−1, 1] is used as the hidden node activation function. Rules are extracted from near optimal neural network by using a new rule extraction algorithm. The aim of ERTNN is to search for simple rules with high predictive accuracy. The major steps of ERTNN are summarized in this Figure.



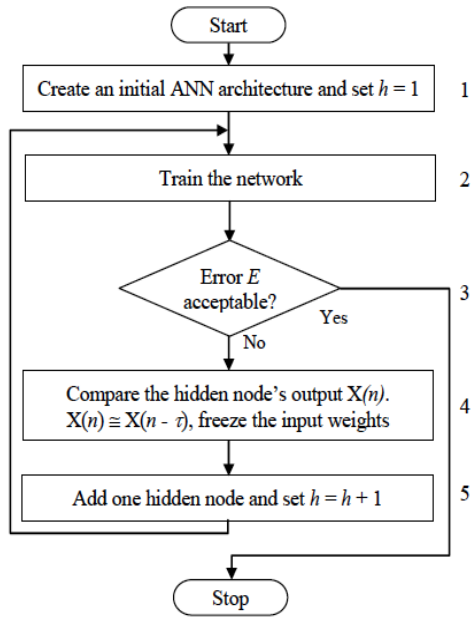
The rules extracted by ERTNN are compact and understandable, and do not involve any weight values. The accuracy of the principles from pruned networks is as high because the accuracy of the original networks. The important features of the ERTNN algorithm are the principles extracted by rule extraction algorithm is recursive in nature and is order insensitive, that is the rules need not to be required to fire sequentially.

4.3 Weight Freezing Based Constructive Algorithm

One drawback of the traditional backpropagation algorithm is the need to determine the quantity of nodes within the hidden layer prior to training. To beat this issue, several algorithms that construct a network dynamically have been proposed. The most well known constructive algorithms are dynamic node creation (DNC), feedforward neural network construction (FNNC) algorithm, and the cascade correlation (CC) algorithm. The constructive algorithm used in the ERTNN algorithm is based on the FNNC algorithm. In FNNC algorithm, the training process is stopped when the classification accuracy on the training set is 100%. However, it is impractical to urge 100% classification accuracy for many of the benchmark classification issues. Additionally, higher classification accuracy on the coaching set does not guarantee the higher generalization ability that is classification accuracy on the testing set.

The training time is an important issue in designing neural network. One approach for reducing the quantity of weights to be trained is to train few weights rather than all weights during a network and keep remaining weights mounted, commonly referred to as weight freezing. The thought behind the weight freezing-based constructive algorithm is to freeze input weights of a hidden node once its output does not modification abundant within the consecutive few training epochs. Theoretical associate degree experimental studies reveal that some hidden nodes of an neural network maintain nearly constant output after some training epochs, whereas others continuously change during the whole training period.

This weight freezing method should be considered as combination of the two extremes: for training all the weights of neural network and for training the weights of only the newly added hidden node of ANNs. The major steps of our weight freezing based constructive algorithm are summarized, Flow chart of the weight freezing based constructive algorithm.



4.4 Pruning Algorithm

The pruning algorithm aims at removing redundant links and units without increasing the classification error rate of the network. A small quantity of units and links left in the network after pruning enable us to extract concise and comprehensible rules. Pruning offers an approach for dynamically determinant associate degree acceptable constellation. Pruning techniques begin by training a larger than necessary network and then eliminate weights and nodes that are deemed redundant.

The nodes of the hidden layer are determined mechanically by weight freezing based constructive algorithm in ERTNN, the aim of this pruning algorithm used here is to get rid of as several supernumerary nodes and connections as potential. A node is pruned if all the connections to and from the node are pruned. Typically, ways for removing weights from the network involve adding a penalty term to the error function. It is hoped that by add a penalty term to the error function, supernumerary connections can have small weights, and thus pruning will reduce the complexity of the network considerably.

The simplest and most commonly used penalty term is the sum of the squared weights. Given a set of input patterns $ni\ x \in \Re$, i =1, 2, . . ., k , let wm is a p-dimensional vector weights for the arcs connecting the input layer and the m-th hidden node, m =1, 2, . . ., h . The weight of the connection from the l-th input node to the m-th hidden node is denoted by wml, vm is a C-dimensional vector of weights for the arcs connecting the m-th hidden node and the output layer. The weight of the connection from the mth hidden node to the p-th output node is denoted by vpm. It has been suggested that faster convergence can be achieved by minimizing the cross entropy function instead of squared error function.

This pruning algorithm removes the connections of the ANN according to the magnitudes of their weights. As the eventual goal of the ERTNN algorithm is to get a set of simple rules that describe the classification method, it is it's vital that every one uncalled-for nodes and connections should be removed. In order to get rid of several connections as possible, the weights of the network should be prevented from taking values that are too large. At an equivalent time, weights of irrelevant connections ought to be inspired to converge to zero. The penalty function is found to be notably appropriate for these purposes. The steps of the pruning algorithm are explained as follows:

- **Step 1** Train the network to meet a

Pre-specified accuracy level with the condition satisfied by all correctly classified input patterns.

$$\max_p |e_{pi}| = \max_p |S_{pi} - t_{pi}| \le \eta_1,\ p = 1, 2, \ldots, C. \qquad (1)$$

Let n1 and n2 be positive scalars such that (n1 + n2) < 0.5 (n1 is the error tolerance, n2 is a threshold that determines if a weight can be removed), where n1 [0, 0.5). Let (*w*, *v*) be the weights of this network.

- **Step 2** Remove connection between input

nodes and hidden nodes, and also remove connection between hidden nodes and output nodes. The task is accomplished in two phases. In first phase, connection between input nodes and hidden nodes are removed. For each $_{ml}\ w$ in the network, if

$$\max_p |v_{pm} w_{ml}| \le 4\eta_2, \qquad (2)$$

then remove $_{ml}\ w$ from the network. In the second phase, connections between hidden nodes and output nodes are removed. For each $_{pm}\ v$ in the network, if

$$|v_{pm}| \le 4\eta_2 \le 4\eta_2, \qquad (3)$$

then remove $_{pm}\ v$ from the network.

- **Step 3** Remove connections

between input nodes and hidden nodes further. If no weight satisfies condition (2) or condition (3), then for each $_{ml}\ w$ in the network,

$$w_{ml} = \max_p |v_{pm} w_{ml}| \qquad (4)$$

Remove $_{ml}\ w$ with smallest $_{ml}\ w$. Continue, otherwise stop.

- **Step 4** Train again the network and

calculating accuracy of the network in classification.

- **Step 5** If classification accuracy of

the network falls below an appropriate level, then stop and use the previous setting of the network weights. Otherwise, head to **Step 2**.

4.5 (RE) Rule Extraction Algorithm

Classification rules are sought in several areas from automatic knowledge acquisition to data mining and neural network rule extraction because some of their attractive options. They are understandable, explicit and verifiable by domain consultants, and may be modified, extended and passed on as standard knowledge. The projected rule extraction algorithm, will be applied to each numeric and discrete data, consist of three major functions:

a) Rule Extraction(RE): This function initialize

the extracted rule list to be empty and sorts the examples according to example frequency. Then it picks the frequent occurring example as the base to generate a rule then it will add the rule to the list of extracted rules. Then it find all the example, that are covered by the rule and remove from the example space. It will repeats the above process iteratively and continuously adds the extracted rules to the rule list until the example space becomes empty, then total data examples have been covered by the rule extraction and they have all been removed.

b) Rule Clustering: The rules are clustered in terms of their category levels. Rules of the same category are clustered together as one group of rules.

c) Rule Pruning: Redundant(repeat) or more specific rules in each cluster are removed. In every clusters, more than one rule may cover the same example.

For examples, the rule “if (color = green) and (height < 4) then grass” is already contained in a more general rule “if (color = green) then grass”, and thus the rule “if (color = green) and (height < 4) then grass” is redundant. Rule extraction eliminates these redundant rules in each cluster to further reduce the size of the best rule list.

A default rule should be selected to accommodate possible unclassifiable patterns. If the rules are clustered, then the choice of the default rule is based on cluster of rules. The steps of the rule extraction(RE) algorithm are explained as follows:

- **Step 1** Extract Rule:

Sort on frequency (data without duplicates)

i = 0;

while (data without duplicates is NOT(!) empty)

{extract *R i* to cover the pattern happened more frequently;

remove all the patterns covered by *R i* ;

i = *i*+1; }

The core of this step contains greedy algorithm that finds the shortest rule based on the primary order information, which may differentiate the pattern into consideration from the patterns of alternative classes. It then iteratively extracts shortest rules and take away the patterns covered by every rule until all patterns are coated by the rules.

- **Step 2** Cluster Rule: Cluster rules

according to their category levels. Rules extracted in Step one are grouped in terms of their class levels.

- **Step 3** Prune Rule: Replace

specific rules with more general ones; Remove noise rules; Eliminate redundant rules;

- **Step 4** Check whether all patterns are coated

by any principle on extraction. If affirmative then stop, otherwise continue.

- **Step 5** Determine a default rule on

extraction. A default rule is chosen if no rule can be applied to a pattern.

5. Performance Evaluation

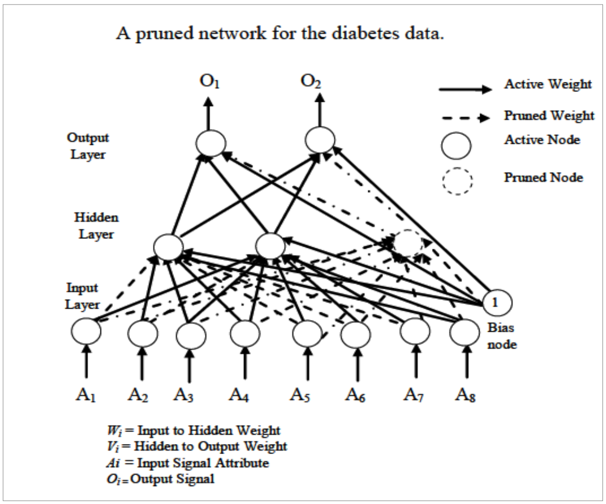
This section evaluates the performance of the ERTNN algorithm on a set of well-known benchmark classification problems including diabetes, wine, iris, golf playing, season, and lenses that are widely used in data mining research and machine learning. The datasets representing all the issues were real world data.

5.1 Dataset Description

This section briefly describes the datasets utilized in this study. The datasets are summarized in Table.1

Table 1. Characteristics of datasets.

Sl No.	Datasets	No. of Examples	Input Attributes	Output Classes
1	Diabetes	768	8	2
2	Iris	150	4	3
3	Wine	178	13	3
4	Season	11	3	4
5	Golf Playing	14	4	2
6	Lenses	24	4	3



The diabetes dataset: The Pima Indians Diabetes information consists of 768 data pairs with eight attributes normalized between zero and one. The eight attributes are number of pregnancies (*A*1), plasma glucose concentration (*A*2), blood pressure (*A*3), triceps skin fold thickness (*A*4), Two hour serum insulin (*A*5), body mass index (*A*6), diabetes pedigree function (*A*7), and age (*A*8). In this database, 268 instances are positive (output equals 1) and 500 instances are negative (output equals 0).

The iris dataset: This is perhaps the best-known database to be found within the pattern recognition literature. The set contains three classes of fifty instances each, where every class refers to a type of Iris plant. 4 attributes are used to predict the iris class, *i.e.*, sepal length (*A*1), sepal width (*A*2), petal length (*A*3), and petal width (*A*4), all in centimetres. Among the 3 classes, class one is linearly separable from the other two classes, and classes two and three are not linearly separable from one another. To ease data extraction, we reformulate the data with three outputs, where class 1 is represented by {1, 0, 0}, class 2 by {0, 1, 0}, and class 3 by {0, 0, 1}.

The wine dataset: In classification context, this is often a well-posed downside with well behaved class structures, a honest dataset for first testing of a new classifier, however not terribly difficult. These knowledge are the results of a chemical analysis of wines grown within the same region in Italia however derived from 3 completely different cultivars. The analysis determined the amount of thirteen constituents found in every of the three types of wines, range of instances 178, range of attributes thirteen. All attributes are continuous. This was a three-class downside.

The season data: The season dataset contains separate data only. There are eleven examples within the dataset, every of that consisted of three-elements. These are tree, weather and temperature. This was a four-class downside.

6. Extracted Rules

The number of rules extracted by the ERTNN algorithm and the accuracy of the rules is presented. This subsection discusses the rules extracted by ERTNN in terms of the original attributes. The amount of conditions per rule conjointly the number of rules extracted have also visualized here.

Number of extracted rules and rules accuracies.

Sl No.	Datasets	No. of Extracted Rules	Accuracy
1	Diabetes	2	76.56%
2	Iris	3	98.67%
3	Wine	3	91.01%
4	Season	4	100%
5	Golf Playing	3	100%
6	Lenses	8	100%

The diabetes data

Rule 1: If Plasma glucose concentration (*A*2) <= 0.64 and Age (*A*8) <= 0.69 then tested negative. Default Rule: tested positive.

The iris data

Rule 1: If Petal-length (*A*3) <= 1.9 then iris setosa

Rule 2: If Petal-length (*A*3) <= 4.9 and Petal-width (*A*4) <= 1.6 then iris versicolor

Default Rule: iris virginica.

The wine data

Rule 1: If Input 10 (*A*10) <= 3.8 then class 2

Rule 2: If Input is 13 (*A*13) > = 845 then class one

Default Rule: class three.

The season data

Rule 1:If Tree (A2) = yellow then autumn

Rule 2:If Tree (A2) = leafless then autumn

Rule 3:If Temp(A3) = low then winter

Rule 4: If Temp(A3) = high then summer

Default Rule: spring.

The golf playing data

Rule 1: If Outlook (A1) = sunny and Humidity >=85 then don't play

Rule 2: Outlook (A1) = rainy and Wind= strong then don't play

Default Rule: play.

7. Conclusions

In this paper we have got conferred a neural network based data mining theme to mining classification rules from given databases. This work is an effort to apply the connectionist approach to mining by extracting symbolic rules an equivalent as that of decision trees. A vital feature of the proposed rule extraction rule is its recursive in nature. A group of experiments was conducted to check the proposed approach employing a well outlined set of data mining issues. The results indicate that using the planned approach top quality rules can be discovered from the given datasets. The extracted rules are comprehensible, concise, order insensitive, and do not involve any weight values. The accuracy of the principle from the pruned network is as high because the accuracy of the absolutely connected networks. Experiments showed that this methodology helped plenty to scale back the number of rules significantly without sacrificing classification accuracy. Almost all cases ERTNN outperformed the others. With the rules extracted by the strategy introduced here, ANNs ought to now not be regarded as black boxes.

References

1. Darbari, A. *Rule Extraction from Trained ANN: A Survey*; Technical Report; Department of Computer Science, Dresden University of Technology, Dresden, Germany, 2000.
2. Wang, L.; Sui, T.Z. Application of data mining technology based on neural network in the engineering, In *Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing*, Shanghai, China, 21–25 September 2007.
3. Ni, X. Research of data mining based on neural networks. *World Academy Sci. Eng. Tech*. **2008**, *39*, 381-384.
4. Vesely, A. Neural networks in data

mining. *Agric. Econ.* **2003**, *49*, 427-431.

5. Baesens, B.; Setiono, R.; Mues, C.; Vanthienen, J. Using neural network rule extraction and decision tables for credit-risk evaluation. *Manage. Sci.* **2003**, *49*, 312-329.

6. Jacobsson, H. Rule extraction from recurrent neural networks: A taxonomy and review. *Neural Comput.* **2005**.

7. Setiono, R.; Baesens, B.; Mues, C. A note on knowledge discovery using neural networks and its application to credit screening, *Eur. J. Operation. Res.* **2009**.

8. Setiono, R.; Leow, W.K. FERNN: An algorithm for fast extraction of rules from neural networks *Appl. Intell.* **2000.**

9. Setiono, R.; Liu, H. Symbolic representation of neural networks. *IEEE Comput.* **1996.**

10. Setiono, R.; Liu, H.; Lu, H. Effective data mining using neural networks. *IEEE Trans. Knowl. Data En.* **1996.**

11. Towell, G.G.; Shavlik, J.W. Extracting refined rules from knowledge-based neural networks. *Mach. Learn.* **1993.**

12. Saito, K.; Nakano, R. Medical diagnosis expert system based on PDP model. In *Proceedings of IEEE International Conference on Neural Networks*, San Diego, CA, USA, 24–27 July 1988.

13. Fu, L. Rule learning by searching on adapted nets. In *Proceedings of National Conference on Artificial Intelligence*, Anaheim, CA, USA, 1991.

14. Setiono, R. Extracting M-of-N rules from trained neural networks. *IEEE Trans. Neural Netw.***2000.**

15. Zurada J.M., “An introduction to artificial neural networks systems”, St. Paul: West Publishing (1992).

Authors

First Author:
Mrs.RENUKA DEVI.S.V
M.Phil Scholar,
Department of Computer Science,
Dr.SNS RCAS, Coimbatore.

Author Guide:
Mrs.MARUTHAVENI.R
Assistant Professor.,
Department of Computer Science,
Dr.SNS RCAS, Coimbatore.