ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

EFFECTIVE DATA MINING USING NEURAL NETWORKS

Shreyash Tambe*, Bhushan Thakur, Avnish Vishwakarma
* Department of Computer Engineering, Vidyavardhini's College Of Engineering and Technology, India

DOI: 10.5281/zenodo.48819

ABSTRACT

Classification is one of the data mining problems receiving great attention recently in the database community. This paper presents an approach to discover symbolic classification rules using neural networks. Neural networks have not been thought suited for data mining because how the classifications were made is not explicitly stated as symbolic rules that are suitable for verification or interpretation by humans. With the proposed approach, concise symbolic rules with high accuracy can be extracted from a neural network. The network is first trained to achieve the required accuracy rate. Redundant connections of the network are then removed by a network pruning algorithm. The activation values of the hidden units in the network are analyzed, and classification rules are generated using the result of this analysis. The effectiveness of the proposed approach is clearly demonstrated by the experimental results on a set of standard data mining test problems.

KEYWORDS: Classification, Data Mining, Network Pruning, Neural Networks, Rule Extraction.

INTRODUCTION

One of the data mining problems is classification. Various classification algorithms have been designed to tackle the problem by researchers in different fields such as mathematical programming, machine learning, and statistics. Recently, there is a surge of data mining research in the database community. The classification problem is re-examined in the context of large databases. Unlike researchers in other fields, database researchers pay more attention to the issues related to the volume of data. They are also concerned with the effective use of the available database techniques, such as efficient data retrieval mechanisms. With such concerns, most algorithms proposed are basically based on decision trees. The general impression is that the neural networks are not well suited for data mining. The major criticisms include the following:

1) Neural networks learn the classification rules by many passes over the training data set so that the learning time of a neural network is usually long.

2) 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. Articulating the classification rules becomes a difficult problem.

3) For the same reason, available domain knowledge is rather difficult to be incorporated to a neural network. On the other hand, the use of neural networks in classification is not uncommon in machine learning community. In some cases, neural networks give a lower classification error rate than the decision trees but require longer learning time. In this project, we present our results

from applying neural networks to mine classification rules for large databases with the focus on articulating the classification rules represented by neural networks

REVIEW OF LITERATURE

Data mining (the analysis step of the "Knowledge Discovery in Databases" process, or KDD), an interdisciplinary subfield of computer science, is the computational process of discovering patterns in large data sets ("big data") involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems.

The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use. Aside from the raw analysis step, it involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating. [7]

The term is a misnomer, because the goal is the extraction of patterns and knowledge from large amount of data, not the extraction of data itself. It also is a buzzword and is frequently applied to any form of large-scale data or information processing (collection, extraction, warehousing, analysis, and statistics) as well as any application of computer decision support system, including artificial intelligence, machine learning, and business intelligence. The popular book "Data mining: Practical machine learning tools and techniques with Java" (which covers mostly machine learning material) was originally to be named just "Practical machine learning", and the term "data mining" was only added for marketing reasons. Often the more general terms "(large scale) data analysis", or "analytics" – or when referring to actual methods, artificial intelligence and machine learning – are more appropriate. [7]

The actual data mining task is the automatic or semi-automatic analysis of large quantities of data to extract previously unknown, interesting patterns such as groups of data records (cluster analysis), unusual records (anomaly detection), and dependencies (association rule mining). This usually involves using database techniques such as spatial indices. These patterns can then be seen as a kind of summary of the input data, and may be used in further analysis or, for example, in machine learning and predictive analytics. For example, the data mining step might identify multiple groups in the data, which can then be used to obtain more accurate prediction results by a decision support system. Neither the data collection, data preparation, nor result interpretation and reporting are part of the data mining step, but do belong to the overall KDD process as additional steps. [7]

The related terms data dredging, data fishing, and data snooping refer to the use of data mining methods to sample parts of a larger population data set that are (or may be) too small for reliable statistical inferences to be made about the validity of any patterns discovered. These methods can, however, be used in creating new hypotheses to test against the larger data populations. [7]

EXISTING SOLUTION

Systems that construct classifiers are one of the commonly used tools in data mining. Such systems take as input a collection of cases, each belonging to one of a small number of classes and described by its values for a fixed set of attributes, and output a classifier that can accurately predict the class to which a new case belongs.

These notes describe C4.5, a descendant of CLS and ID3. Like CLS and ID3, C4.5 generates classifiers expressed as decision trees, but it can also construct classifiers in more comprehensible ruleset form [3].

A. Decision trees

B. Rule set classifiers

PROPOSED SYSTEM

The proposed system is basically a research based system which classifies the given database using rules that are extracted from neural network. At present, data mining is a new and important area of research, and neural network itself is very suitable for solving the problems of data mining because its characteristics of good robustness, self-organizing adaptive, parallel processing, distributed storage and high degree of fault tolerance. The combination of data mining method and neural network model can greatly improve the efficiency of data mining methods, and it has been widely used.

Neural Networks (Backpropagation)

In machine learning and cognitive science, artificial neural networks (ANNs) are a family of statistical learning models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning. [5]

Backpropagation, an abbreviation for "backward propagation of errors", is a common method of training artificial neural networks used in conjunction with an optimization method such as gradient descent. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function. [6]

Backpropagation requires a known, desired output for each input value in order to calculate the loss function gradient. It is therefore usually considered to be a supervised learning method, although it is also used in some unsupervised networks such as auto encoders. It is a generalization of the delta rule to multi-layered feedforward networks, made possible by using the chain rule to iteratively compute gradients for each layer. Backpropagation requires that the activation function used by the artificial neurons (or "nodes") be differentiable. [6]

The backpropagation learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

Each propagation involves the following steps:

1.Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.

2.Backward propagation of the propagation's output activations through the neural network using the training pattern target in order to generate the deltas (the difference between the input and output values) of all output and hidden neurons.

Phase 2: Weight update

For each weight-synapse follow the following steps:

1.Multiply its output delta and input activation to get the gradient of the weight.

2.Subtract a ratio (percentage) of the gradient from the weight.

This ratio (percentage) influences the speed and quality of learning; it is called the learning rate. The greater the ratio, the faster the neuron trains; the lower the ratio, the more accurate the training is. The sign of the gradient of a weight indicates where the error is increasing.

This is why the weight must be updated in the opposite direction. [6] Repeat phase 1 and 2 until the performance of the network is satisfactory.

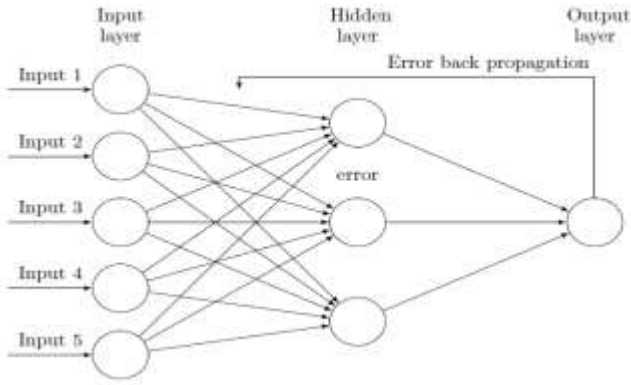
IMPLEMENTATION METHODOLOGY

Here we will study how the entire project idea was implemented into a run able code which produces a tangible output, i.e. a running version of software. Hence in this chapter we will learn implementation of the project. Implementation is vital part in software building and designing. Our project is implemented in Java. We have used Eclipse to build our Project. We are going to use MySQL database to store data. DBC will serve as an interface between the program and database.

A. Designing a Neural Network

A neural network has been designed using Backpropagation algorithm which is a supervised learning algorithm. The Backpropagation neural network is a multilayered, feedforward neural network and is by far the most extensively used. It is also considered one of the simplest and most general methods used for supervised training of multilayered neural networks. Backpropagation works by approximating the non-linear relationship between the input and the output by adjusting the weight values internally. [1]

Figure:



BPN Network

B. Creating/Gathering training Data sets

We create/gather the suitable training data sets for which the neural network is to be trained. The training dataset should be large enough to train the network for almost all the probable combinations of data.

C. Thermometer Encoding

Before we feed the data to the network for its training, the data needs to be encoded in a format which the neural network can easily work with. For this, we use thermometer encoding technique. To solve the problem, a neural network was first constructed. Each attribute value was coded as a binary string for use as input. to the network The e strings used to represent the attribute values are in The thermometer coding scheme was used for binary representations of the continuous attributes Each bit of a string was either 0 or 1 depending on which subinterval the original value was located For example, a salary value of 140k would be coded as (1, 1, 1, 1, 1, 1} and a value of 100k as (0, 1, 1, 1, 1, l}. For the discrete attribute, elevel, for example, an elevel of 0 would be coded as (0, 0, 0, 0}, 1 as {0, 0, 0, 1}, etc. [1]

Table:

Table 1:Coding of the attributes for neural network input

Attribute	No. of inputs	Subintervals
Salary	6	[20k,25k],[25k,50k],[50k,75k], [75k,100k],[100k,125k]
Commission	3	[0k,25k],[25k,50k],[50k,75k]
Age	6	[20,30],[30,40],[40,50],[50,60], [60,70],[70,80]
Elevel	4	[0],[1],[2],[3],[4]

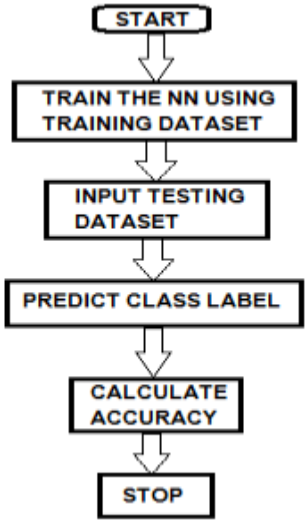
D. Training the Neural Network

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually tweaked. A limiting value is set so that the training stops on reaching the limit. A learning rate is initialized which is set to either low (if the dataset is variable and large) or high (if the dataset is small). [6]

E. Testing the Neural Network against Random data

Once the training part is completed, the network is given random sets of data as input. The network will classify the given input into the appropriate group/class based on how effectively the network is trained. The network will be 80-90% accurate in classifying the random sets of data. [1]

G. Flowchart

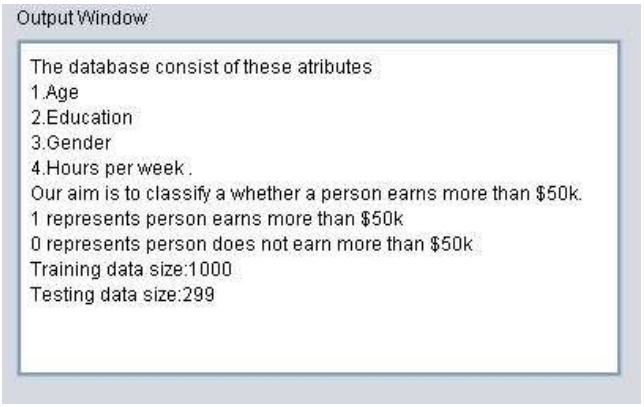


Flowchart

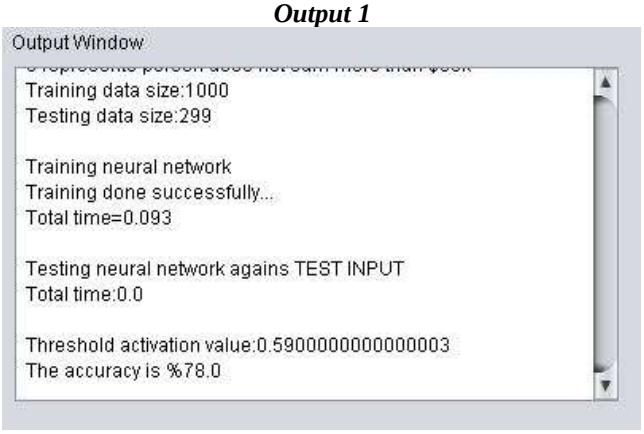
RESULTS

The following figure shows the output of the algorithm for a test database of 300 inputs.

Snapshots:



Output 1



Output 2

CONCLUSION

In this paper we present a neural network based approach to mining classification rules from given databases. The approach consists of three phases:

In semester VII we have implemented following modules.

1. Design of Neural Network
2. Suitable Machine Learning Database

We have implemented following modules for the Semester VIII

1. Database Attribute Encoding
2. Integration of ANN module, Encoding module, Classification module
3. GUI module

REFERENCES

[1] Bekir Karlik and A. Vehbi Olgac International Journal of Artificial Intelligence And Expert Systems (IJAE), Volume (1): Issue (4) 111 Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks J. D. Poston and W. D. Horne, "Discontiguous OFDM considerations for dynamic spectrum access in idel TV channels," in Proc. IEEE DySPAN, 2005.

[2] A comparative study of decision trees ID3 and C4.5, Badr HSSINA, Abdelkarim MERBOUHA V. Chakravarthy, X. Li, Z. Wu, M. Temple, and F. Garber, "Novel overlay/underlay cognitive radio waveforms using SD-SMSE framework to enhance spectrum efficiency—Part I," IEEE Trans. Commun., vol. 57, no. 12, pp. 3794–3804, Dec. 2009.

[3] Top 10 Algorithms in Data Mining, Xindong Wu, Vinay Kumar V. Chakravarthy, Z. Wu, M. Temple, F. Garber, and X. Li, "Cognitive radio centric overlay-underlay waveform," in Proc. 3rd IEEE Symp. New Frontiers Dynamic Spectrum Access Netw., 2008, pp. 1–10.

[4] https://en.wikipedia.org/wiki/Artificial_neural_network

[5] https://en.wikipedia.org/wiki/Data_mining