CHAPTER-18

Classification by Back propagation

18.1 Introduction

18.2 A Multilayer Feed-Forward Neural Network

18.3 Defining Network Topology

18.4 Propagate the inputs forward:

18.5Back propagate the error:

18.6 Terminating condition

18.7 Classification Based on Concepts from Association Rule Mining

18.8 Review Questions

18.9 References

18.Classification by Back propagation

18.1 Introduction

Back propagation is a nueral network learning algorithm. Psychologists originally kindled the field of neural networks and neurobiologists who sought to devlop and test computational analogues of neurons. Roughly speaking, a neural network is a set of connected input/output units where each connection has a weight associated with it. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input samples. Neural network learning is also referred to as connectionist learning due to the connections between units.

Neural networks involve long training times and are therefore more suitable for applications where this is feasible. They require a number of parameters that are typically best determined empirically, such as the network topology or "structure."

Neural networks have been criticized for their poor interpretability, since it is difficult for humans to interpret the symbolic meaning behind the learned weights. These features initially made neural networks less desirable for data mining.

Advantages of neural networks, however, include their high tolerance to noisy data as well as their ability to classify patterns on which they have not been trained. In

Addition, several algorithms have recently have been developed for the extraction of rules from trained neural networks. These factors contribute towards the usefulness of neural networks for classification in data mining.

The most popular neural network algorithm is the back propagation algorithm, Proposed in the 1980's . Later you will learn about multilayer feed-forward networks, the type of neural network on which the back propagation algorithm performs.

18.2 A Multilayer Feed-Forward Neural Network

The back propagation algorithm performs learning on a multilayer fee-forward neural network. The inputs correspond to the attributes measured for each raining sample. The inputs are fed simultaneously into layer of units making up the input layer. The weighted outputs of these units are, in turn, fed simultaneously to a second layer of neuron like units, known as a hidden layer. The hidden layer s weighted outputs can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only one is used. The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given samples.

The units in the hidden layers and output layer are sometimes referred to as neurodes, due to their symbolic biological basis, or as output units. Multilayer feed-forward networks of linear threshold functions, given enough hidden units, can closely approximate any function.

18.3 Defining Network Topology

Before training can begin, the user must decide on the network topology by specifying the number of units in the input layer, the number of hidden layers(if more than one), the number of units in each hidden layer, and the number of units in the output layer.

Normalizing the input values for each attribute measured in the training samples will help speed up the learning phase. Typically, input values are normalized so as to fall between 0.0 and 1.0. Discrete-valued attributes may be encoded such that there is one input unit per domain value. For example, if the domain of an attribute A is (ao,a1,a2) then we may assign three input units to represent A. That is , we may have, say, as input units. Each unit is initialized to 0. If A =a0, then it is set to 1. If A==a1 it is set to 1, and so on.One output uit may be used to represent two classes (where the value I represents one class, and the value 0 represents the other). If there are more than two classes, then one output unit per class is used

There are no clear rules as to the “best” number of hidden layer units. Network design is a trial-and –error process and may effect the accuracy of the resulting trained network. The initial values of the weights may also affect the resulting accuracy. Once a repeat the training process with a different network topology or a different set of initial weights.

Backpropagation

Back propagation learns by iteratively processing a set of training samples, comparing the network’s predicition for each sample with the actual known class label. For each training sample, the weights are modified so as to minimize the mean squared error between the network’s prediction and the actual class.

These modifications are made in the “backwards” direction, that is , form the output layer through each hidden layer down to the first hidden layer (hence the name backpropagation). Although it is not guaranteed in general the weights will eventually converge, and the learning process stops. The algorithm is summarized below. Initialize the weights. The weights in the network are initialized to small random number(e.g., ranging from -1.0 to1.0,or -0.5 to 0.5). Each unit has a bias associated with it, as explained below. The biases are similarly initialized to small random numbers.

Each training sample: X, is processed by the following steps.

18.4 Propagate the inputs forward:

In this step,the net input and output of each unit in the hidden and output layers are computed. First, the training sample is fed to the input layer of the network. Note that for unit in the input layer , its output is equal to its input layers is conputed as a linear combination to it in the previous layer. To compute the net input to the unit, each input is connected to the unit is multiplied by its corresponding weight, and this is summed. Given a unit in a hidden or output layer, the net input to unit is

$$Ij=\sum W_{ij}\, Oi+\theta j$$

Where Wij,is the weight of the connection from unit; in the previous layer to unit;0i is the output of unit j from the previous layer; and 0j is the bias of the unit. The bias acts as a threshold in that it serves to vary the activity of the unit.

Each unit in the hidden and output layers takes its net input and then applies an activation function to it . The function symbolizes the activation of the neuron represented by the unit. The logistic, or sigmoid, function is used. This function is also referred to as a squashing function., since it maps a large input domain onto the smaller

range of 0 to 1. The logistic function is non-linear and differentiable, allowing ther back propagation algorithm to model classification problems that are linearly inseparable.

18.5 Back propagate the error:

The error is propagated backwards by updating the weights and biases to reflect the error of the network's prediction. For a unit j in the output layer, the error Err_j is computed by

$$\mathbf{Err}_j = \mathbf{0}_j (1-0_j)(T_j-0_j)$$

To compute the error of a hidden layer unit j, the weighted sum of the errors of the units connected to unit j in the next layer is considered. The error of a hidden layer unit j is

$$\mathbf{Err}_i = 0_j(1-0_j)\sum \mathbf{Err}_k \mathbf{W}_{jk}$$

Where Wjk is the weight of the connection from unit; to a unit k in the next higher layer and Err_k is the error of unit k. The weights and biases are updated to reflect the i propagated errors. Weights are updated by the following equations, where DW_{ij} is the change in weight W_{ij}

$$\Delta W_{ij}=(L)\ Err_j\ 0i$$

$$W_{ij}=W_{ij}+DW_{ij}$$

The variable L is the learning rate, a constant typically having a value between 0.0 to 1.0 . Back propagation learns using a method of gradient descent to search for a set of weights that can model the given classification problem so as to minimize the mean squard distance between the networks class prediction and the actual class label of the samples. The learning rate helps to avoid getting stuck at a local minimum in decision space(i.e., where the weights appear to converge, but are not the optimum solution) and encourages finding the global minimum. If the learning rate is too small, then learning will occur at a very slow pace. If the learning rate is too large, then oscillation between Inadequate solutions may occur. A rule of thumb is to set the learning rate to 1/f, where T is the number of iterations through the training set so far.

Biases are updated by the following equations below, where $\Delta \theta j$, is the change in Bias θj;

$$\Delta \theta j =(L)\ Err_j$$

$$\theta j= \theta j+ \Delta \theta j$$

Note that here we are updating the weights and biases after the presentation of each sample. This is referred to as **case updating.** Alternatively, the weight and bias Increments could be accumulated in variables, so that the weights and biases are Updated after all of the samples in the training set have been presented . This latter strategy is called **epoch updating**, where one iteration through the training set is an **epoch**. In theory, the mathematical derivation of back propagation employs epoch updating, yet in practice, case updating is more common since it tends to yield more accurate results.

18.6 Terminating condition

Training stops when

- All ΔWij in the previous epoch were so small as to be below some specified threshold, or

- The percentage of samples misclassified in the previous epoch is below some threshold, or

- A prespecified number of epochs has expired.

In practice, several hundreds of thousands of epochs may be required before the weights will converge.

18.7Classification Based on Concepts from Association Rule Mining

Association rule mining is an important and highly active area of data mining research. Recently, data mining techniques have been developed that apply concepts used in association rule mining to the problem of classification. In this section, we study three methods in historical order. The first two, ARCS and associative classification, use association rules for classification. The third method, CAEP, mines"emerging patterns" that consider the concept of support used in mining associations.

The first method mines association rules based on clustering and then employs the rules for classification. The **ARCS** or Association Rule Clustering System, mines association rules of the form Aquan1Λ Aquan2 => Acat where Aquan1 and Aquan2 are tests on quantitative attributive ranges(where the ranges are dynamically determined), and Acat assigns a class label for a categorical attribute from the given training data. association rules are plotted on a 2-D grid. The algorithm scans the grid, searching for rectangular clusters of rules. In this way, adjacent ranges of the quantitative attributes occurring within a rule cluster may be combined. The clustered association rules generated by ARCS was empirically found to be slightly more accurate than C4.5 when there are outliers in the data. The accuracy of ARCS is related to the degree of discretization used. In terms of scalability, ARCS requires " a constant amount of memory", regardless of the database size.C4.5 has exponentially higher execution times than ARCS, requiring the entire database, multiplied by some factor, to fit entirely in main memory.

The second method is referred to as associative classification. It mines rules of the form condset=>y, where condset is a set of items (or attribute-value pairs) and y is a class label. Rules that satisfy a prespecified minimum support are frequent, where a rule has support s if . s% of the samples in the given data set contain **condset** and belong to class y. A rule satisfying minimum confidence is called **accurate**, where a rule has confidence c if c% of the samples in the given data set that contain condset belong to class y. If a set of rules has the same condset, then the rule with the highest confidence is selected as the possible rule(PR) to represent the set.

The association classification method consists of two steps. The first step finds the set of all PRs that are both frequent and accurate. It uses an iterative approach, where prior knowledge is used to prune the rule search. The second step uses a heuristic method to construct the classifier, where the discovered rules are organized according to decreasing precedence based on their confidence and support. The algorithm may require several passes over the data set, depending on the length of the longest rule found. When classifying a new sample, the first rule satisfying the sample is used to classify it. The classifier also contains a default rule, having lowest precedence, which specifies a default class for any new sample that is not satisfied by any other rule in the classifier. In general, the associative classification method was empirically found to be more accurate than C4.5 on several data sets. Each of the above two steps was shown to have linear scale-up.

The third method ,CAEP (classification by aggregating emerging patterns), uses the notion of itemset support to mine emerging patterns (EPs), which are used to construct a classifier. Roughly

speaking, an EP is an itemset (or set of items) whose support increases significantly from one class of data to another. The ratio of the two supports is called the growth rate of the EP. For example, suppose that we have a data set of customers with the classes buys^computer = “yes”, or C1, and buys_computer = “no”, or C2, the itemset {age = “<_30”,student = “no”} is a typical EP, whose support increases from o.2% in C1 to 57.6% in C2 at a growth rate of EP = 288. Note that an item is either a simple equality test; on a categorical attribute is in an interval. Each EP is a multiattribute test and can be very strong at differentiating instances of one class from another. For instance, if a new sample X contains the above EP, then with odds of 99l6% we can claim that X belongs to C2. In general, the differentiating power of an EP is roughly proportional to its growth rate and its support in the target class.

For each class C, CAEP find EPs satisfying given support and growth rate thresholds, where growth rate I computed with respect to the set of all non-C samples versus the target set of all C samples, “Border-based” algorithms can be used for this purpose. Where classifying a new sample, X, for each class C, the differentiating power of the EPs of class C that occur in X are aggregated to derive a score for C that is then normalized. The class with the largest normalized score determines the class label of X.

CAEP has been found to be more accurate than C4.5 and association0-based classification on several data sets. It also performs well on data sets where the mail class of interest is in the minority. It scales up on data volume and dimensionality An alternative classifier, called the JEP-classifier, was proposed based on jumping emerging patterns (JEPs). A JEP is a special type of EP, defined as an itemset whose support increases abruptly from zero in one data set to nonzero in another data set. The two classifiers are considered complementary.

18.8 Review Questions

1 Explain about A Multilayer Feed-Forward Neural Network

2 Defining Network Topology

3 Expalin about Back propagate the error:

4 What is Terminating condition

5 Expalian about Classification Based on Concepts from Association Rule Mining

18.9 References

[1]. Data Mining Techniques, Arun k pujari 1st Edition

[2] .Data warehousung,Data Mining and OLAP, Alex Berson ,smith.j. Stephen

[3].Data Mining Concepts and Techniques ,Jiawei Han and MichelineKamber

[4]Data Mining Introductory and Advanced topics, Margaret H Dunham PEA

[5] The Data Warehouse lifecycle toolkit , Ralph Kimball Wiley student Edition