International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:3, 2007

Classifier Based Text Mining for Neural Network

M. Govindarajan, and R. M. Chandrasekaran

***Abstract*—**Text Mining is around applying knowledge discovery techniques to unstructured text is termed knowledge discovery in text (KDT), or Text data mining or Text Mining. In Neural Network that address classification problems, training set, testing set, learning rate are considered as key tasks. That is collection of input/output patterns that are used to train the network and used to assess the network performance, set the rate of adjustments. This paper describes a proposed back propagation neural net classifier that performs cross validation for original Neural Network. In order to reduce the optimization of classification accuracy, training time. The feasibility the benefits of the proposed approach are demonstrated by means of five data sets like contact-lenses, cpu, weather symbolic, Weather, labor-nega-data. It is shown that , compared to exiting neural network, the training time is reduced by more than 10 times faster when the dataset is larger than CPU or the network has many hidden units while accuracy ('percent correct') was the same for all datasets but contact-lences, which is the only one with missing attributes. For contact-lences the accuracy with Proposed Neural Network was in average around 0.3 % less than with the original Neural Network. This algorithm is independent of specify data sets so that many ideas and solutions can be transferred to other classifier paradigms.

***Keywords*—**Back propagation, classification accuracy, text mining, time complexity.

I. Introduction

IN supervised learning, we are given a set of example pairs (x,y), x ϵ X, y ϵ Y and the aim is to find a function f in the allowed class of functions that matches the examples. In other words, we wish to *infer* the mapping implied by the data; the cost function is related to the mismatch between our mapping and the data and it implicitly contains prior knowledge about the problem domain. In this article we start with the following assumptions.

1) Multi-Layer Perceptions (MLP, [11] are used for classification. The simple feedforward neural network is actually called a Multilayer perception (MLP).

Manuscript received March 10, 2007. This work was supported in part by the first author got Career Award for Young Teachers (CAYT) grant from All India Council for Technical Education, New Delhi.

M. Govindarajan is with the Annamalai University, Annamalai Nagar, Tamil Nadu, India (phone: 91-4144-221946; e-mail: govind_aucse@yahoo.com).

R. M. Chandrasekaran is with Annamalai University, Annamalai Nagar, Tamil Nadu, India (phone: 91-4144-238444; e-mail: aurmc@sify.com).

An MLP is a network of perceptions. The neurons are placed in layers with outputs always flowing toward the output layer. If only one layer exists, it is called a perception. If multiple layers exist, it is an MLP.

2) Back Propagation algorithm (BP algorithm, [11] is a learning technique that adjusts weights in neural network by propagating weight changes backward from the sink to the source nodes.

Tasks that fall within the paradigm of supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation).The supervised learning paradigm is also applicable to sequential data (e.g., for speech and gesture recognition). This can be thought of as learning with a "teacher," in the form of a function that provides continuous feedback on the quality of solutions obtained thus far.

A. Performances of Neural Network Systems

One concern in machine learning community is that a system trained on small samples may not perform well on test data. On the other hand, if training data sets are too large, our concern is how well and efficiently a system can learn. The objective of this study [1] is what neural network systems are better suited for applications that have small or large training data. For studying neural learning from small training data we chose five data sets like contact-lenses, cpu, weather symbolic, Weather, labor-nega-data. All five collections have rather balanced distribution among all classes, and the number of pattern classes is not too large. First, we utilized our developed text mining algorithms, including text mining techniques based on classification of data in several data collections. After that, we employ exiting neural network to deal with measure the training time for five data sets. Experimental results show that the accuracy ('percent correct') was the same for all datasets but Contact-lences, which is the only one with missing attributes. For Contact-lences the accuracy with Proposed Neural Network was in average around 0.3 % less than with the original Neural Network. The bigger the dataset, the larger the improvement in speed. Other informal experiments with larger datasets showed that Proposed Neural Network can be more than 10 times faster when the dataset is larger than CPU or the network has many hidden units.

B. Advantages and Disadvantages of Neural Networks

The computed output [5] is compared to the known output. If the computed output is correct, then nothing more is necessary. If the computed output is incorrect, then the weights are adjusted so as to make the computed output closer to the known output. This process is continued for a large number of cases, or time-series, until the net gives the correct output for a given input. The entire collection of cases learned is called a "training sample" (Connor, Martin, and Atlas, 1994). In most real world problems, the neural network is never 100% correct. Neural networks are programmed to learn up to a given threshold of error. After the neural network learns up to the error threshold, the weight adaptation mechanism is turned off and the net is tested on known cases it has not seen before. The application of the neural network to unseen cases gives the true error rate (Baets, 1994).

Artificial neural networks present a number of advantages over conventional methods of analysis. First, artificial neural networks make no assumptions about the nature of the distribution of the data and are not therefore, biased in their analysis. Instead of making assumptions about the underlying population, neural networks with at least one middle layer use the data to develop an internal representation of the relationship between the variables (White, 1992). Second, since time-series data are dynamic in nature, it is necessary to have non-linear tools in order to discern relationships among time-series data. Neural networks are best at discovering non-linear relationships (Wasserman, 1989; Hoptroff, 1993; Moshiri, Cameron, and Scuse, 1999; Shtub and Versano, 1999; Garcia and Gencay, 2000; and Hamm and Brorsen, 2000). Third, neural networks perform well with missing or incomplete data. Whereas traditional regression analysis is not adaptive, typically processing all older data together with new data, neural networks adapt their weights as new input data becomes available (Kuo and Reitch, 1994). Fourth, it is relatively easy to obtain a forecast in a short period of time as compared with an econometric model. However, there are some drawbacks connected with the use of artificial neural networks. No estimation or prediction errors are calculated with an artificial neural network (Caporaletti, Dorsey, Johnson, and Powell, 1994). Also, artificial neural networks are “black boxes,” for it is impossible to figure out how relations in hidden layers are estimated (Li, 1994).

In addition, a network may become a bit overzealous and try to fit a curve to some data even when there is no relationship. Another drawback is that neural networks have long training times. Reducing training time is crucial because building a neural network forecasting system is a process of trial and error. Therefore, the more experiments a researcher can run in a finite period of time, the more confident he can be of the result.

C. Applications

The utility of artificial neural network Models lies in the fact that they can be used to infer a function from observations. This is particularly useful in applications where the complexity of the data or task makes the design of such a function by hand impractical.

D. Real Life Applications

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, Including time series prediction and modeling.
- Classification, including pattern and sequence
- Recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering,
  Blind source separation and compression.

Application areas include system identification and control (Vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

E. Back Propagation Algorithm

Back propagation algorithm is in no way a new idea, but exiting approaches typically suffer from the problems of a high runtime and classification accuracy in small dataset. Often, these two problems are closely related: Due to a high runtime and classification accuracy in a smaller datasets. Hence, we take over the best from existing approach like a cross validation that reduce the runtime and classification accuracy of the BP algorithm radically. In particular, we use methods [2] for:

1) k-fold cross validation (Back propagation algorithm)
2) In stratified cross – validation
3) Leave-one-out

k-fold cross validation: The initial data are randomly partitioned into k mutually exclusive subsets or “folds”, $s_1,s_2,s_3….s_k$, each of approximately equal to size. Training and Testing is performed k times. The accuracy estimate is the overall number of correct classifications from the k-iterations, divided by the total number of samples in the initial data.

In stratified cross: validation, the folds are stratified so that the class distribution of the samples in each fold is approximately the same as that in the initial data.

Leave-one-out: k-fold cross validation with k set to s, number of initial samples. In general, stratified 10-fold cross-validation is recommended for estimating classifier accuracy (even if computation power allows using more folds) due to its relatively low bias and variance. The use techniques as well as innovative novel ideas. Its advantages are outlined by

means of five data sets like contact-lenses, cpu, weather symbolic, Weather, labor-nega-data. compared to Multilayer perception neural network, the training time is reduced by more than 10 times faster when the dataset is larger than CPU or the network has many hidden units while accuracy ('percent correct') was the same for all datasets but contact-lenses, which is the only one with missing attributes. For contact-lenses the accuracy with Proposed Neural Network was in average around 0.3 % less than with the Multilayer perception Neural Network.

The remainder of this article is structured as follows. First, the state of the art is analyzed to motivated our work (Section II) and the Back propagation algorithm was described (Section III). Then, the BP for architecture optimization is introduced – with a strong focus on the innovative aspects mentioned above (Section IV). After that, the advantages of the approach are set out by means of various datasets (Section V). Finally, the main findings are summarized and an outlook on future work is given (Section VI).

II. State of the Art

In this section, the state of the art concerning cross validation of BP algorithm is investigated. The results of this survey will motivate a new approach.

A. Related Work

This article focuses on training time and classification accuracy using cross validation of BP algorithm. Cross validation methods are described in [2]. In general, filter approach was described. The problem of training time and classification accuracy for neural networks is discussed in [3] [4].

Here, we discuss examples of the combination of MLP and BP algorithm. Altogether, we investigated five datasets where cross validation methods are applied to optimize BP algorithm. The following steps are carrying out to classify the back propagation preparation [2].

1. Initialize the weights
2. Propagate the inputs forward
3. Back propagate the error

B. Motivation for a New Approach

- The original Neural Network (MLP) uses linear output nodes when the class is numeric; here we assume the class is nominal. Therefore the code can be used only for classification and does not support regression.
- Last attribute must correspond to the class (this could be easily changed though)
- There's no GUI, so the topology cannot be modified during training time
- There is only 1 hidden layer
- Missing attributes are eliminated using a Filter. This is the reason for having new feature, not present in original implementation:

If there are no missing attributes, setting flag -F allows a slightly faster processing. If training and test sets don't have missing attributes, the results we got were exactly the same, as shown in the table below.

We claim that the runtime of MLP approach must be reduced by more than 10 times faster when the dataset is larger than CPU or the network has many hidden units while accuracy ('percent correct') was the same for all datasets but contact-lenses, which is the only one with missing attributes. For contact-lenses the accuracy with Proposed Neural Network was in average around 0.3 % less than with the original Neural Network (MLP).

If more difficulty solutions could be investigated with short time, it can also be expected that better solutions can be achieved [8].

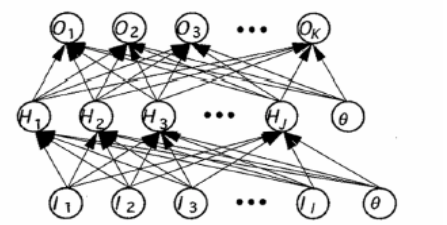

Fig. 1 The typical structure of a back propagation network

III. Classification with Back Propagation Algorithm

Back propagation [11] is a learning technique that adjusts weights in the NN by propagating weight changes backward from the sink to the source nodes. Back propagation is the most well know form of learning because it is easy to understand and generally applicable. Back propagation can be thought of as a generalized delta rule approach. During propagation, data values input at the input layer flow through the network, with final values coming out of the network at the output layer. The propagation occurs by applying the activation function at each node, which then places the output value on the arc to be sent as input to the next nodes. In moat cases, activation function produces only one output value that is propagated to the set of connected nodes. The NN can be used for classification and/or learning. During the classification process, only propagation occurs. However, when learning is used after the output of the classification occurs, a comparison to the known classification is used to determine how to change the weights in the graph. In the simplest types of learning, learning progresses from the output layer backward to the input layer. Weights are changed based on the changes that were made in weights in subsequent arcs. The backward learning process is called back propagation [11].

Algorithm

Input:

N // Starting neural network

X={x1….xh} // Input tuple from training set

D= {d1,….dm} // Output tuple desired

Output:

N // Improved neural network

Back propagation algorithm:

// Illustrate back propagation

Propagation (N,X);

$$E=1/2 \sum_{i=1}^{m} (d_i - y_i)^2 ;$$

Gradient (N, E) ;

The MSE (Mean Squared Error) is used to calculate the error. Each tuple in the training set is input to this algorithm. The last step of the algorithm uses gradient descent as the technique to modify the weights in the graph. The basic idea of gradient descent is to find the set of weights that minimizes the MSE.

IV. Optimization of BP Algorithm

In this section, a schematic overview of cross validation used BP optimization is given. Then, the standard techniques are sketched and our innovative extensions are described in detail.

A. Overview

From an algorithmic perspective, optimization is a least value for the minimization that can be used to solve a wide range of optimization tasks including the most important parameters are optimized of neural network.

B. Standard Methods of the Cross Validation

The development of the new approach was guided by the idea that well –known cross validation methods should be applied as far as possible. To keep the runtime of the cross validation, only the most important parameters are optimized. We discuss techniques [2] for estimating runtime and classifier accuracy, such as the

(i) Holdout
(ii) K-fold cross validation

Holdout: The given data are randomly partitioned into two independent sets, a training set and test set. Random sub sampling is a variation of the holdout method in which the holdout method is repeated k times. The overall accuracy estimate is taken as the average of the accuracies obtained from each iteration.

K-fold cross-validation: The initial data are randomly partitioned into k mutually exclusive subsets or "folds", $s_1,s_2,s_3….s_k$, each of approximately equal to size. Training and Testing is performed k times. The accuracy estimate is the overall number of correct classifications from the k-iterations, divided by the total number of samples in the initial data.

In stratified cross – validation, the folds are stratified so that the class distribution of the samples in each fold is approximately the same as that in the initial data.

Bootstrapping: Given training instances uniformly with replacement.

Leave-one-out: k-fold cross validation with k set to s , number of initial samples.

In general, stratified 10-fold cross- validation is recommended for estimating classifier accuracy (even if computation power allows using more folds) due to its relatively low bias and variance. The use of such techniques to estimate classifier accuracy increases the overall computation time, yet is useful for among several classifiers.

Increases classifier Accuracy:

(i) Bagging (or bootstrap aggregation)

(ii) Boosting

C. Innovative Methods of the Cross Validation

In this section, innovative extensions to our approach are described. Their development was guided by the idea that the runtime of the BP must be reduced significantly, so that it can be deployed in real-world applications.

1) The original Neural Network (MLP) uses linear output nodes when the class is numeric; here we assume the class is nominal. Therefore the code can be used only for classification and does not support regression.
2) Last attribute must correspond to the class (this could be easily changed though).
3) There's no GUI, so the topology cannot be modified during training time.
4)There is only 1 hidden layer.
5) Missing attributes are eliminated using a Filter. This is the reason for having new feature, not present in original implementation.

If there are no missing attributes, setting flag -F allows a slightly faster processing. If training and test sets don't have missing attributes, the results I got were exactly the same, as shown in the table below.

V. Experimental Results

In this section we demonstrated the properties and advantages of our approach by means of five data sets like contact-lenses, cpu, weather symbolic, Weather, labor-nega-data. The performance of classification algorithms is usually examined by evaluating the accuracy of the classification. However, since classification is often a fuzzy problem, the correct answer may depend on the user. Traditional algorithm evaluation approaches such as determining the space and time overhead can be used, but these approaches are usually secondary.

Classification accuracy [11] is usually calculated by determining the percentage of tuples placed in the correct class. This ignores the fact that there also may be a cost associated with an incorrect assignment to the wrong class.
This perhaps should also be determined. We examine the Performance of classification much as is done with information retrieval systems. With only two classes, there are four possible outcomes with the classification. The upper left and lower right quadrants are correct actions. The remaining two quadrants are incorrect actions.

TABLE I
Properties of Data Sets

Dataset Factor of	INSTANCES	Attribues
Contact-lences	24	5
cpu	209	7
weather.symbolic	14	5
weather	14	5
labor-neg-data	57	17

TABLE II
Training Time (Seconds)

Dataset Factor of	Proposed NeuralNetwork (PNN)	Original Neural Network (ONN)	Faster by
Contact-lences	0.93	1.48	1.5
cpu	4.67	4.78	1.0
weather.symbolic	0.61	1.16	1.9
weather	0.39	0.44	1.1
labor-neg-data	16.31	16.59	1.0

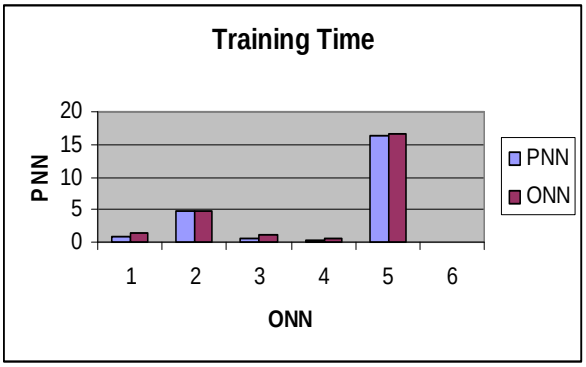

Fig. 2 Training Time

TABLE III
Classification Accuracy

Dataset Factor of	(PNN) % Correct class using 10-fold cross validation	(ONN) % Correct class	Classification Accuracy
Contact-lences	70.8333 %	100 %	0.3%
cpu	99.25%	100%	0.0075%
weather.symbolic	71.4286 %	100 %	0.2%
weather	78.5714 %	100 %	0.2%
labor-neg-data	85.9649%	100%	0.1%

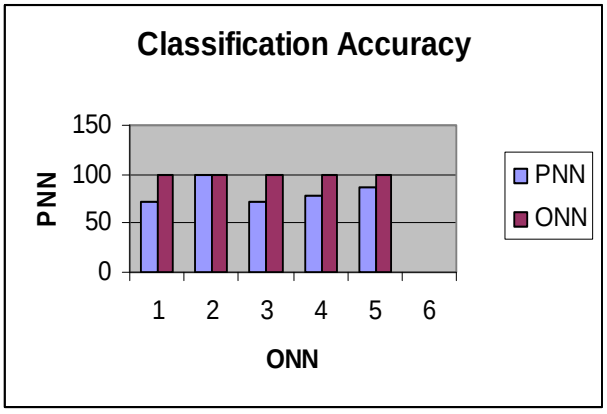

Fig. 3 Classification Accuracy

VI. Conclusion

In this work we developed one text mining classifier using Neural Network methods to measure the training time for five data sets like contact-lenses, cpu, weather symbolic, Weather, labor-nega-data. First, we utilized our developed text mining algorithms, including text mining techniques based on classification of data in several data collections. After that, we employ exiting neural network to deal with measure the training time for five data sets. Experimental results show that the accuracy ('percent correct') was the same for all datasets but Contact-lences, which is the only one with missing attributes. For Contact-lences the accuracy with Proposed Neural Network was in average around 0.3 % less than with the original Neural Network. The bigger the dataset, the larger the improvement in speed. Other informal experiments with larger datasets showed that Proposed Neural Network can be more than 10 times faster when the dataset is larger than CPU or the network has many hidden units.

Acknowledgment

Authors gratefully acknowledge the authorities of Annamalai University for the facilities offered and encouragement to carry out this work. This part of work is supported in part by the first author got Career Award for Young Teachers (CAYT) grant from All India Council for Technical Education, New Delhi. They would also like to thank the reviewer's for their valuable remarks

References

[1] Guobin Ou,Yi Lu Murphey, "Multi-class pattern classification using neural networks", Pattern Recognition 40 (2007).
[2] Jiawei Han, Micheline Kamber "Data Mining – Concepts and Techniques" Elsevier, 2003, pages 303 to 311, 322 to 325.
[3] Intrusion Detection: Support Vector Machines and Neural Networks**,** Srinivas Mukkamala, Guadalupe Janoski, Andrew Sung {srinivas, silfalco, Department of Computer Science, New Mexico Institute of Mining and Technology, Socorro, New Mexico 87801, 2002, IEEE.
[4] N. Jovanovic, V. Milutinovic, and Z. Obradovic, *Member, IEEE,* "Foundations of Predictive Data Mining" (2002).
[5] Yochanan Shachmurove, Department of Economics, The City College of the City, University of New York and The University of Pennsylvania, Dorota Witkowska, Department of Management, Technical University of Lodz "CARESS Working Paper #00-11Utilizing Artificial Neural Network Model to Predict Stock Markets" September 2000.
[6] Bharath, Ramachandran. *Neural Network Computing*. McGraw-Hill, Inc., New York, 1994. pp. 4-43.
[7] Luger, George F., and Stubblefield, William A. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving,* (2nd Edition)*.* Benjamin/Cummings Publishing Company, Inc., California, 1993, pp. 516-527.
[8] Off-line Handwriting Recognition Using Artificial Neural Networks Andrew T. Wilson.
[9] Skapura, David M., *Building Neural Networks.* ACM Press, New York. pp. 29-33.
[10] Bhavit Gyan, University of Canterbury, Kevin E. Voges, University of Canterbury Nigel K. Ll. Pope, Griffith University "Artificial Neural Networks in Marketing from 1999 to 2003: A Region of Origin and Topic Area Analysis".
[11] Margaret H.Dunham, "Data Mining- Introductory and Advanced Topics" Pearson Education, 2003, pages 106-112.

M. Govindarajan received the B.E and M.E and Pursuing Doctoral Degrees in Computer Science and Engineering from Annamalai University, Tamil Nadu, India in 2001 and 2005 and 2006 respectively. He is currently a lecturer (Senior Scale) at the Department of Computer Science and Engineering, Annamalai University, Tamil Nadu, India. He has presented and Published more than 10 papers in Conferences and Journals. His current Research Interests include Data Mining and its applications, Algorithms, Text Mining, Neural Networks, genetic Algorithms, support vector machine, Radial Basis Function, ontology based Reasoning, Case Based Reasoning.

He has conducted National Conference on Recent Trends in Data Mining and its Applications (March 11-12, 2006). He was the recipient of the Achievement Award for the field and to the Conference Bio-Engineering, Computer science, Knowledge Mining (2006), Prague, Czech Republic and All India Council for Technical Education "Career Award for Young Teachers (2006), New Delhi, India. He is Life Member of Computer Society of India, Indian Society for Technical Education and Session Member of Indian Science Congress Association.

Dr. R. M. Chandrasekaran received the B.E Degree in Electrical and Electronics Engineering from Maduari Kamaraj University in 1982 and the MBA (Systems) in 1995 from Annamalai University, M.E in Computer Science and Engineering from Anna University and PhD Degree in Computer Science and Engineering from Annamalai University, Tamil Nadu, India in 1995, 1998 and 2006 respectively.

He is currently working as a Professor at the Department of Computer Science and Engineering, Annamalai University, Annamalai Nagar, Tamil Nadu, India. From 1999 to 2001 he worked as a software consultant in Etiam, Inc, California, USA. He has conducted Workshops and Conferences in the Areas of Multimedia, Business Intelligence and Analysis of algorithms, Data Mining. He has presented and published more than 32 papers in conferences and journals and is the author of the book Numerical Methods with C++ Program (PHI, 2005). His Research interests include Data Mining, Algorithms, Networks, Software Engineering, Network Security, Text Mining. He is Life member of the Computer Society of India, Indian Society for Technical Education, Institute of Engineers, Indian Science Congress Association.