Applying Multi Layer Feed Forward Neural Networks on Large Scale Data

G. Jananii

Abstract---*Investigation on large data sets is extremely important in data mining.Large amount of data generally requires a specific learning method or of any optimization method. Particularly some standard methods are used for example Artificial Neural Network, back propagation neural network and other neural networksnecessitate very long learning time.The existing technique that does not performed well on the large data sets. So in this paper presents a new approach called multi layered feed forward neural network which can work efficiently with the neural networks on large data sets.Data is separated into several segments, and learned by anidentical network structure whereas all weights from the set of networks are integrated. The results from the experiments show that the proposed method can protect the accuracy while the training time is significantly reduced.*

Keywords---*Neural Network, Large Scale Dataset, Incremental Learning*

I. INTRODUCTION

TO think about a data set which items are described by more than three features is desired.So, the goal is to create some visual insight keen on the analyzed data set for that a multidimensional data is presented.Intended for human insight, the data should be point out in a low-dimensional space, generally of two or three dimensions.Learning in the neural network have been separated into more than a few aspects whether it is a study in structure modeling, network design, and improved performance to rapidly learn and to achieve more accurate results [1].

There are many types of neural networks are present [2], other thanthat the basic principles are alike.Each and every neuron in the network is capable to receive input signals, to process them and to send an output signal. Each neuron is linkedwithin any case one neuron, and each link is calculated by a real number, called the weight coefficient, that reflects the degree of importance of the given connection in the neural network.Numerous neural network techniques are used in a variety of fields, like data mining, image identification, weather forecasting, etc. This study will focus on the study of large data by applying multiple neural networks to find out several sub dataset.Shibata and Ikeda show that the number of neurons and the number of hidden layers in the network can have an effect on performance, because a small number of layers can practice faster than a big one. Generally, number of hidden layer can raise the precision of learningbut it mayinfluence the learning time more than a small layer. Besides, a large data set is hard to study at one time also it requires both resources and time.Hence, this paper presents atechnique which can divide the large data sets to be multiple subsets each of which is trained by means of same neural networks. Then improve the overall accuracy by means of proposed technique that restores the suitable weight from the smallest error to each node of the neuron.After that, the weights of the finest small data sets to are used to generate a new network. The accuracy is similar to the network trained by the whole data set as the training time is severely decreased in the proposed technique.

G. Jananii, Student, CIET, India.

DOI: 10.9756/BIJMMI.10051

II. MULTI-LAYER FEED-FORWARD (MLF) NEURAL NETWORKS

MLF neural networks, train with a back-propagation learning algorithm, is awell-liked neural networks [6].A MLF neural network consists of neurons that are prearranged into layers as shown in the figure 1.The first layer is a input layer, the last layer is the output layer, where n is the number of nodes of the

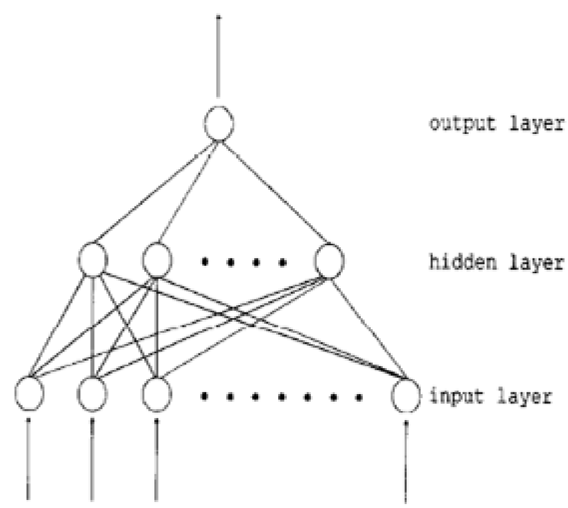

Figure 1: Feed Forward Neural Network Composed of Three Layers

output layer and the layers between are hidden layers. In support of the formal explanation of the neurons can use the mapping function Γ, that assigns for each neuron ia subset $\Gamma(i) \subseteq V$which consists of all ancestors of the given neuron.A subset $\Gamma(i) \subseteq V$ than consists of every predecessors of the given neuron *i*.Each neuron in a definite layer is linked with all neurons in the next layer. The connection between the *ith*and jth neuron is characterized by the weight coefficient$w_{ij}$and the ith neuron by the threshold coefficient ϑ_i is shown in the figure 2.The weight coefficient replicates the degree of significance of the given connection in the neural network.The output value of the *ith*neuron xi is firmed by equation 1 and 2 shown below

$$x_i = f(\xi_i) \qquad (1)$$

$$\xi_i = \vartheta_i + \sum_{j \epsilon \Gamma_i^{-1}} w_{ij} x_j \qquad (2)$$

where ξ_i is the potential of the ith neuron and function $f(\xi_i)$ is the transfer function. The threshold coefficient can be as a weight coefficient of the link with properly added neuron j, where $x_j = 1$ called as bias.

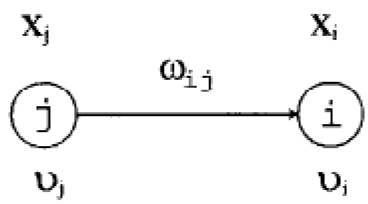

Figure 2: Connection Between Two Neurons i and j

For the transfer function it can be said that

$$f(\xi) = \frac{1}{1 + \exp(-\xi)} \qquad (3)$$

The supervised adaptation process be different the threshold coefficients f_{ii} and weight coefficients wij to minimize the sum of the squared differences between the computed and required output values. This is accomplished by minimization of the objective function E:

$$E = \sum_{O} \frac{1}{2} (x_0 - \hat{x}_o)^2 \qquad (4)$$

where$x_0$and $\hat{x}_o$, are vectors composed of the computed and necessary activities of the output neurons and summation runs over all output neurons o.

The experiments are conducted to improve the training algorithm and it is able to train a large dataset divided equally as sub datasets.The proposed method in that the smaller sub data set will consume less training time than the other.Though the fault from each sub data set is unstable and all weights are separated out. For this reason, from each node collect the weight in the lowest error network and use these weights to change the trained weight of other networks with the same structure.

Suppose the original training set (*N0*) is separated into *n* sub data sets that are$N_1$to N_n.Each separated set will contain ***X/n*** case and each sub set will be trained by the same feed forward neural network structure with a single hidden layer.When all networks are trained well, assessthe results of each network.The weight from thebest trained network is called as (Nbest) will be used as aninitial weight set in the weight integration process.***Nbest***will be useful to all other datasets to make clear that ***Nbest***has better result or the network trained fromof the set itself is better than ***Nbest***. After that the network integration model s introduced.

2.1. Network Integration Method

Change the weights in *Nbest*by means of two methods one is Weight Combination and the other one is Node Creation.If the two hidden nodes are from different networks the first method is used and is closed to each other. Thenew weights of one node in *Nbest*will be set as the average weights since both hidden nodes as in equation 5.

$$w_{Nbest} \leftarrow \frac{w_{Nbest} + w_{Ni}}{2} \quad (5)$$

where, $w_{Nbest}$and w_{Ni}denote the weight vector of two hidden nodes which are nearly closed to each other.

III. Experimental Results

To conduct the experiment two data sets from UCI repository [3] is used they are Iris and LetterRecognition. The Letter Recognition contains 20,000 records which take very long learning time using theordinary training method. A machine learning tool is used, WEKA [4], and MBP [5] in all experiments.

3.1. Network Convergence

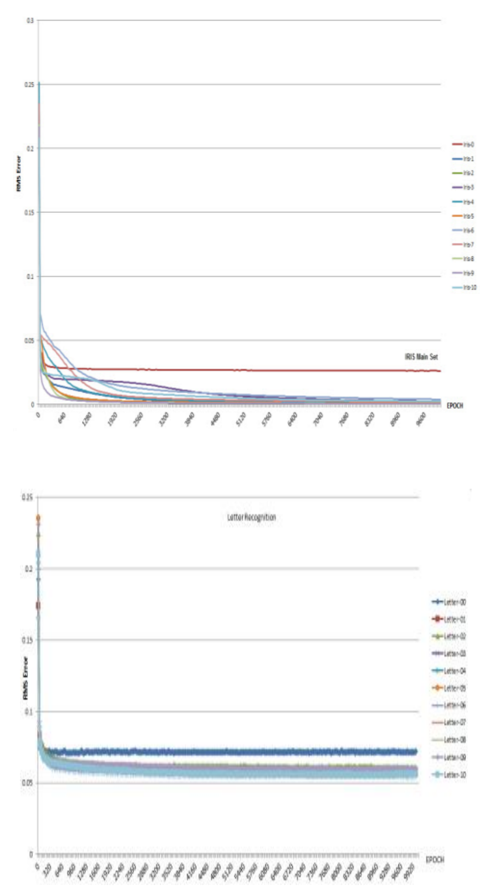

Figure 3: Error Convergence of Iris (left) and Letter Recognition (right) Dataset

Fig. 3 shows error of dataset, Iris-0 indicates the main dataset, and Iris-1 to Iris-10 denotes ten subdatasets. The right graph shows error on Letter Recognition dataset. Letter-01 to Letter-10 is error fromsub dataset 1 to 10. Letter-00 is the original training set. From both graphs that theerror of the sub datasets is much lower than the error of the original training set. The trained weightsconverge to the point that gives the lower error rate on the training set.

3.2. Experimental Results

The accuracy of sub datasets from both datasets is shown in Table 1. N9 and N6 were thebest networks from Iris and Letter Recognition. After getting the best network (*Nbest*), integrate weights from all networks and the results and the results are shown in Table 2.

Table 1: RMS Error of Each Network and Main Network on its Own Dataset

	RMS Error	
Data Set	Iris	Letter Recognition
N_{all}	0.0110874587	0.0720147958
N_1	0.0018586001	0.0571487767
N_2	0.0009121871	0.060338648
N_3	0.004561998	0.0585201252
N_4	0.0011229377	0.0576738189
N_5	0.0038304094	0.0569459839
N_6	0.0038304094	0.0547496419
N_7	0.0049999925	0.0551124568
N_8	0.0009161137	0.0581173707
N_9	0.0007008063	0.0596956224
N_{10}	0.0032190801	0.0556769854

Table 2: RMS Error of N_{best} before and after Integration

Data Set	Weight Set	RMS Error
Iris	N_{best}	0.0110874587
	N_{best} (after Integration)	0.0103989667
Letter Recognition	N_{best}	0.0720147958
	N_{best} (after integration)	0.0672054431

IV. CONCLUSION

This paper has proposed a new method called multilayered feed forward neural network for largedatasets. From the experimental results the weights of small dataset converged faster than the originaldataset. On the other hand, the weights trained from sub dataset did not attain better result when they were tested onthe original dataset.Hence, the proposed approach in which after the rule integration, the accuracy of the weight obtained from the proposed method is better than the weight set whichis the best among the sub datasets.

REFERENCES

[1] Y. Zhao, J. Gao, and X. Yang, A Survey of neural network ensembles, *IEEE Trans. Pattern Analysis and MachineIntelligence*, 2005.

[2] K. Shibata and Y. Ikeda, Effect of number of hidden neurons on learning in large-scale layered neural networks,*ICROS-SICE International Joint Conference*, 2009,P p5008-5013.

[3] UCI Data Sets, The UCI Machine Learning Repository, *Center for Machine Learning and Intelligent Systems,University of California, Irvine, United States*, 2007, http://archive.ics.uci.edu/ml/datasets.html192

[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software:An Update. *SIGKDD Explorations, Volume 11, Issue 1*, 2009.

[5] Multiple Back-Propagation, Back-Propagation Simulation Tool, *Noel de Jesus Mendonça Lopes InstitutoPolitécnico da Guarda, Portugal*, 2009, http://dit.ipg.pt/MBP

[6] Daniel Svozil, Vladimir KvasniEka, JiEPospichal, "Introduction to multi-layer feed-forward neural networks", Chemometrics and Intelligent Laboratory Systems 39 (1997) 43-62