ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Performance Comparison of Back propagation Neural Network and Extreme Learning machine for Multinomial Classification Task

Nilamadhab Dash[1], Rojalina Priyadarshini[2], Samita Rout[3]

[1]Assistant Professor, Information Technology,C V Raman College of Engg/BPUT,Unit-9,Bhubaneswar, Odisha,India, nilamadhab04@gmal.com
[2]Assistant Professor, Information Technology,C V Raman College of Engg/BPUT, Bhaktamadhu Nagar,Bhubaneswar, Odisha,India,priyadarshini.rojalina@gmal.com
[3]Assistant Professor, Computer Science & Engg,C V Raman College of Engg/BPUT, Jagannath Bihar,Bhubaneswar, Odisha,India,samita.rout83@gmal.com

Abstract: Classification and prediction tasks continue to play a vital role in the area of computer science and data processing. Clustering and classification in Data Mining are used in various domains to give meaning to the available data. Data Mining has especially become popular in the fields of forensic science, fraud analysis and healthcare, as it reduces costs in time and money. In classification modeling the data is classified to make predictions about new data. Using old data to predict new data has the danger of being too fitted on the old data. But that problem can be solved by using soft computing tools which generalizes the same type of data into one class and rest to the other which are known as binary classifiers. This paper describes and compares the application of two popular machine learning methods: Back propagation neural network and Extreme learning machine which are used as multiclass classifiers. These two approaches are applied on same type of multi class classification datasets and the work tries to generate some comparative inferences from training and testing results. The datasets are taken from UCI learning repository.

Keywords: Multinomial classification, Extreme learning machine, back propagation neural network, Normalization, Multilayer feed forward

I. INTRODUCTION

Classifiers in machine learning can be of two types: Binary classifiers and multinomial classifiers. In machine learning, multiclass or multinomial classification is the problem of classifying the given instances into more than two classes. While some classification algorithms naturally permit the use of more than two classes, others are by nature binary algorithms; these can, however, be turned into multinomial classifiers by a variety of strategies.[4] Multiclass classification should not be confused with multi-label classification, where multiple classes are to be predicted for each problem instance. In this paper we have used the recently developed extreme learning machine to classify some of the multinomial datasets. In this work we have taken a couple of classification datasets from the UCI learning repository. They are: Iris data and the seeds dataset which have four and seven predictive attributes respectively and three class attributes.

Extreme learning machine (ELM) represents one of the recent successful approaches in machine learning, particularly for performing pattern classification. It is proposed by Huang at el [5] uses Single Layer Feed forward Neural Network (SLFN) Architecture [2]. It randomly chooses the input weights randomly and thereby determines the output weights of Single Layer Feed Network. It has a better generalization performance with a faster learning speed. It requires less human interference and can run thousands times faster than those conventional methods. [3] It automatically determines all the network parameters analytically, which avoids trivial human intervention and makes it efficient in online and real- time applications.

The generalization performance of the ELM algorithm for data classification problem depends basically on three free parameters [5].)They are (i) the number of hidden neurons, (ii)the input weights and (iii) the bias values which need to be optimally chosen. Neural Networks have been extensively used in many fields due to their ability to approximate complex nonlinear mappings directly from the input sample; and to provide models for a large class of natural and artificial phenomena that are difficult to handle using classical parametric techniques. One of the disadvantages of the Neural Network is the learning time. But ELM overcomes the problems caused by gradient descent based algorithms such as Back propagation applied in ANNs. ELM can significantly reduce the amount of time needed to train a Neural Network. This paper presents the efficiency of Extreme Learning Machine (ELM) over Back propagation neural network. The paper is organized as follows, Section 2 describes about the dataset preparation and training, Section 3 presents the learning algorithm of ELM and BPN. Section 4 describes the simulated outputs results and section 5 contains the conclusion and future works.

II. Dataset preparation:

Normalization of input data is used for ranging the values to fall within an acceptable scope, and range[1]. The features obtained from many specimens are actually considered as raw data. Some amount of pre-processing is always carried out on the input and output data in order to make it suitable for the network. These procedures help in obtaining faster and efficient training. If the neurons have nonlinear transfer functions (whose output range is from -1 to 1 or 0 to 1), the data is normalized for efficiency. As our outputs are falling within these ranges, each feature in each dataset is normalized using column normalization. The normalized data are used as the inputs to the machines. The three classes of the target output data are converted to 00, 01 and 10 respectively.

III. Learning

Learning is the ability to approximate the behaviour adaptively from the training data while generalization is the ability to predict the training data.[7] Generalization is a more desirable and critical feature because the most common use of a classifier is to make good prediction on new or unknown objects. To test our approach, we have taken different data class sets from repository of data sets. We have divide the data sets into two parts. i,e training set and testing set which is not used in the training process, and is used to test and then we have simulated our results with these datasets. Almost $2/3^{rd}$ of the total dataset has taken as training set and $1/3^{rd}$ of the rest has taken as test set. This is done through the analysis of the accuracy achieved through testing against these set. Then we have simulated our network with the said data.

A. ELM learning Algorithm

ELM algorithm has three steps as follows

Given a training set $N = \{(x_i,t_i)|x_i \in R^n, t_i \in R^m, i = 1,\cdots,N\}$, activation function g, and the number of hidden nodes L,

1 Assign randomly input weight vectors or centers a_i and hidden node bias or impact factor b_i, $i = 1,\cdots,L$.

2 Calculate the hidden layer output matrix H.

3 Calculate the output weight β: $\beta = H^{\dagger}T$. where $H^{\dagger}$ is the Moore-Penrose generalized inverse of hidden layer output matrix H.

B. Training Phase For simple Artificial Neural Network

The Neural Network is designed and implemented using the MATLAB 7 with Neural Network Toolbox 4.0. There are four steps in the training process: (i) assemble the training data (ii) Create the network. (iii) Train the network. (iv) Test and validate network response to new inputs.

In this study, we have used the network with 7 nodes in the input layer,5 nodes in the hidden and 2 neurons in the output layer. We have used Logsig function as the activation function.

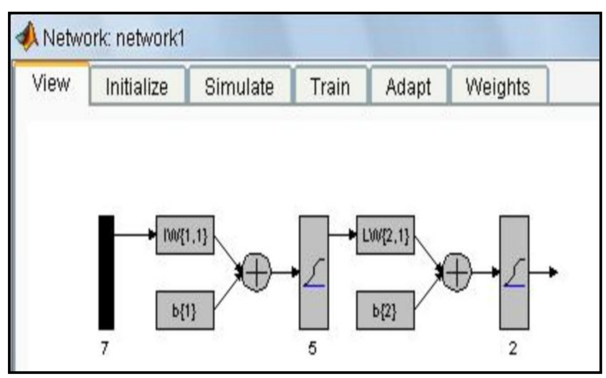


Fig I-Network Architecture (7 neurons in the input layer, 5 neurons in the hidden layer and 2 neurons in the output layer

C. Training for extreme learning machine using sigmoidal activation function

In this study we have used the said extreme learning machine for training the network. 2/3rd of the sample datasets are used as training data and remaining one third data are taken as testing data set. Here we have used the differentiable sigmoidal function as the activation function. The network has seven inputs and two output neurons. The network is trained and simulated using MATLAB 7.0.This application is run in normal configuration of the PC.

IV. Results with Simulated outputs

The below table shows the result obtained from the extreme learning machine. We have taken one third unknown data samples for testing purpose and do the simulation on the same samples. The circled ones given in the Fig3,4 and 4 are showing the misclassifications among the samples.

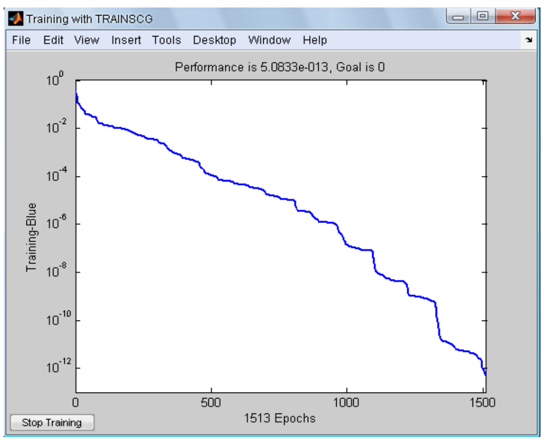
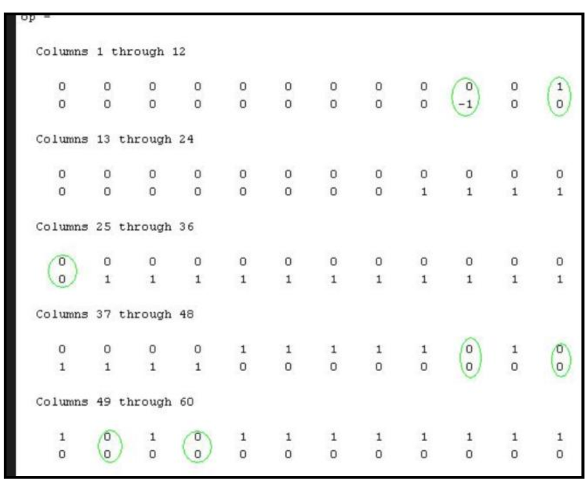
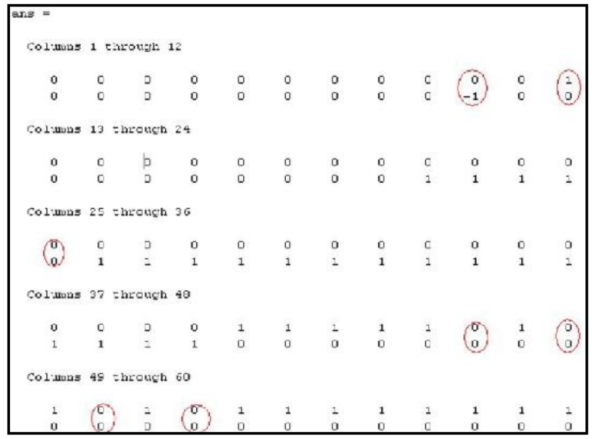


Figure-II Convergence Graph of Neural network



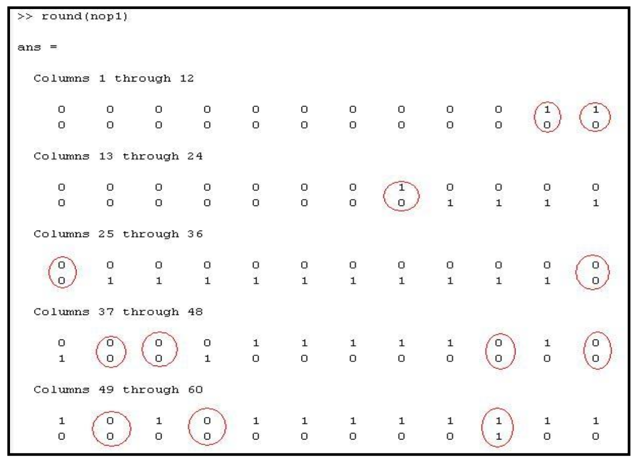
```
op =

  Columns 1 through 12

     0     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0    -1     0     0

  Columns 13 through 24

     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     1     1     1     1

  Columns 25 through 36

     0     0     0     0     0     0     0     0     0     0     0     0
     0     1     1     1     1     1     1     1     1     1     1     1

  Columns 37 through 48

     0     0     0     0     1     1     1     1     1     0     1     0
     1     1     1     1     0     0     0     0     0     0     0     0

  Columns 49 through 60

     1     0     1     0     1     1     1     1     1     1     1     1
     0     0     0     0     0     0     0     0     0     0     0     0
```

Fig-III Figure-3 (Seven misclassifications in case of Extreme Learning Machine)



```
ans =

  Columns 1 through 12

     0     0     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     0     0     0     0     0    -1     0     0

  Columns 13 through 24

     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     1     1     1     1

  Columns 25 through 36

     0     0     0     0     0     0     0     0     0     0     0     0
     0     1     1     1     1     1     1     1     1     1     1     1

  Columns 37 through 48

     0     0     0     0     1     1     1     1     1     0     1     0
     1     1     1     1     0     0     0     0     0     0     0     0

  Columns 49 through 60

     1     0     1     0     1     1     1     1     1     1     1     1
     0     0     0     0     0     0     0     0     0     0     0     0
```

Figure-IV (Seven misclassifications shown in the results obtained from Backpropagation ANN: Training-1)



```
>> round(nop1)

ans =

  Columns 1 through 12

     0     0     0     0     0     0     0     0     0     0     1     1
     0     0     0     0     0     0     0     0     0     0     0     0

  Columns 13 through 24

     0     0     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     0     1     1     1     1

  Columns 25 through 36

     0     0     0     0     0     0     0     0     0     0     0     0
     0     1     1     1     1     1     1     1     1     1     1     0

  Columns 37 through 48

     0     0     0     0     1     1     1     1     1     0     1     0
     1     0     0     1     0     0     0     0     0     0     0     0

  Columns 49 through 60

     1     0     1     0     1     1     1     1     1     1     1     1
     0     0     0     0     0     0     0     0     0     1     0     0
```

Figure-5 (Ttwelve misclassifications shown in the results obtained from Backpropagation ANN: Training-2)

V. Discussions and Conclusion:

The inferences which can be possibly drawn from our observations are as follows:

1) The learning speed of ELM is actually justifies its name as it runs extremely fast. In our simulations, the learning phase of ELM is completed in less than seconds for many datasets. Whereas conventional learning algorithm takes very long time to train network algorithms such as the feed-forward back propagation network used in our work. Whereas this technique has the advantage of skipping the training time.

(2) The proposed ELM has better generalization performance[3] than the gradient-based learning such as

back- propagation in most cases. In our results we have observed that generalization is not consistent in case of back propagation as the ratio of classified and unclassified data are not always constant. Whereas ELM produces a same result always and thus consistent and has got better generalization..

(3) From the studies it has been found out that the traditional classical gradient-based learning algorithms may face several issues like local minima, improper learning rate and overfitting etc. [6] In order to avoid these issues, some methods such as weight decay and early stopping methods may need to be used often in these classical learning algorithms. The ELM tends to reach the solutions straightforward without such trivial issues.

(4)The ELM learning algorithm looks much simpler than most learning algorithms for feed forward neural networks. That is why the code complexity of extreme learning machine is comparatively lower than simple feed forward back propagation network.

5) As random weights are used in the back propagation algorithm we are getting different outputs after each training but extreme learning machine always provides same output for a particular input sample.

From the observations it may be inferred that the performance of extreme learning machine is substantially promising in terms of response time as compared to the feed forward back propagation neural network. The training time which is still an issue in feed forward back propagation network is simply bypassed in extreme learning machine. Still then its behavior is to be tested against other transfer functions and the network parameters like number of nodes in the hidden layer, effect of learning rate parameters and bias. The future scope of the proposed work is still open to analyze the effect of number of feature attributes and the predicting attributes and how it behaves towards the sequential data sets.

Training Method	**No of Training Samples**	**No of Test samples**	**No of Misclassifications**	**Training Time**	**Execution Time/ Simulation Time**
ANN(BPN)	120	30	1	6.703 secs	.0470secs
ELM(Averaged over 10 trainings)	120	30	0	Nil	.0202secs
ANN(BPN) (Training-1)	150	60	07	7.8021 secs	.0605secs
ANN(BPN) (Training-2)			12	8.1020 secs	.0630secs
ELM(Averaged over 10 trainings)	150	60	07	Nil	.0564 secs

Table-1 shows the comparisons between Multilayer feed forward Backpropagation network and Extreme Learning Machine

VI. References

[1] Pawalai Kraipeerapun and Somkid Amornsamankul, “Applying Multiple Complementary Neural Networks to Solve Multiclass Classification Problem”, International Journal Of Applied Mathematics And Informatics, Issue 3, Volume 6, 2012,pp 134-141.

[2] J. Siva Prakash and R. Rajeshalayam,” Random Iterative Extreme Learning Machine for Classification of Electronic Nose Data”, International Journal Of Wisdom Based Computing, Vol. 1(3), December 2011,pp 24-27.

[3] R. Rajesh, J. Siva Prakash,” Extreme Learning Machines - A Review and State-of-the-art”, International Journal Of Wisdom Based Computing, Vol. 1(1), 2011, pp 35-49.

[4] Dr. Chandra.E, Rajeswari, ”A Survey on Data Classification using Machine Learning Techniques”, International Journal of Engineering Science and Technology (IJEST), Vol. 3 No.10 October 2011, pp 7397-7401

[5] Guang-Bin Huang, Qin-Yu-Zhu, Chee-Kheeng Sied,”Extreme Learning Machine-Theory and applications”, Science Direct Neurocomputing 2006, pp 489-501.

[6] M.C. Sezgin, Z. Dokur, T. Ölmez, M. Korürek,”Classification of Respiratory Sound by Using An Neural Network”, *Proceedings – 23rd Annual Conference – IEEE/EMBS*,(2001)

[7]Guoqiang Peter Zhang, “Neural Networks for Classification: A Survey ”, *IEEE Transaction on systems management and Cybernatics- Applications and Reviews Vol-30, No-4*,pp 451-462(2000).