IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Special Issue-1 E-ISSN: 2347-2693

A Framework for Selection of Neural Network Training Functions towards the Classification of Yeast Data

Shrayasi Datta*
Department of Information Technology
Jalpaiguri Government Engineering College
Jalpaiguri,WestBengal,India
shrayasi.datta@gmail.com

Dr. J. Paulchoudhury
Department of Information Technology
Kalyani Government Engineering College
Kalyani,Nadia,WestBengal,India
jnpc193@yahoo.com

Abstract— **Yeast is among the various important components for the formulation of medicine and various chemical products, so yeast data classification is an important bioinformatics task. Yeast data classification has been approached by various machine learning techniques for last few years. In this paper, an artificial neural network system with back propagation training algorithm is presented with different training functions for the classification of yeast dataset. Here an effort has been made to decide the suitable training functions of artificial neural network system for the classification of yeast protein. The training functions that have been used are, respectively, Batch Training, Batch Gradient Descent, Gradient Descent with momentum, Resilience back propagation, One-step secant back propagation, Scaled Conjugate back propagation, Conjugate Gradient back propagation with Polak-Riebre updates (CGP) and Conjugate Gradient back propagation with Fletcher-Reeves updates (CGF), BFGS and Levenberg-Marquardt training algorithm . The yeast dataset used for this purpose has been chosen and from UCI machine learning repository. The performance of the classification network has been tested by various performance measures like correctness of classification, mean square error, and regression analysis.**

Keywords - Yeast Dataset Classification, Back Propagation Artificial Neural Network, Training Function Of Artificial Neural Network

I. INTRODUCTION

A cell usually contains approximately 1 billion protein molecules [1]. These molecules reside in various compartments of a typical cell called "protein subcellular locations ". The information regarding the protein subcellular locations helps researchers to know the function of a cell properly. With the help of machine learning techniques [2], prediction of determining the subcellular localization of a protein can be done with a great percentage of success. Among the various types of prokaryotic and eukaryotic cells, the subcellular localization of yeast protein always snatches the attention of the researcher because of the demand of yeast in medicine and food technology domains and also because of the similarities of yeast cell structure with human cell.

Kanehisa and Nakai [3,4] developed a rule base expert system "PSORT" for classifying yeast and E. Coli cell's protein subcellular location, which has been considered as the first approach for predicting the localization sites of proteins from their amino acid sequences. After that, Horton and Nakai [5] have proposed a probabilistic model for classification of yeast and E. Coli protein based on their subcellular locations. It has achieved accuracy of 81% on E. Coli dataset and 55% on yeast dataset. This accuracy has improved significantly when the authors implemented another expert system using three machine learning algorithms, namely k-nearest neighbour algorithm, binary decision tree, and naïve Bayes classifier for classifying protein subcellular locations in yeast dataset and E. Coli

dataset [6]. Performance of these three techniques with the Probabilistic method [5] has also been compared and it has been shown that the performance of k-nearest neighbour algorithm is better among these four, with an accuracy of about 60% on yeast dataset.

Chen Y. [7] has implemented two machine learning algorithms for classification of E. Coli and yeast cells: decision tree and artificial neural network, and it is concluded that these techniques have similar performance measures for these two dataset.

Bo Jin, Yuchun Tang, Yan-Qing Zhang, Chung-Dar Lu and Irene Weber [8] have proposed and designed SVM with fuzzy hybrid kernel based on TSK fuzzy model and have showed that fuzzy hybrid kernel achieved better performance in SVM classification.

Algorithm based on Fuzzy rule base technique is proposed in heart disease [9]. Some work also has been done by the authors [10] on the comparison of different machine learning techniques for the classification task of yeast data. Authors [11] have also proposed the suitable membership function for yeast dataset classification when classifying using fuzzy rule based system.

There are several research works for describing the classification task of yeast dataset [3, 8] using machine learning techniques like artificial neural network, k-nearest neighbour algorithm, and support vector machine techniques, as described above. But most of them concentrated on selecting the machine learning approach by which the classification has been done. Very little concentration has been given for different aspects of one particular approach. Hence, it comes to the purpose of the present research work. Here, classification of yeast dataset has been done with artificial neural network. The yeast dataset is obtained from UCI machine learning repository [12]. 8 attributes of a yeast protein from yeast dataset is taken as an input to the artificial neural network. After that, artificial neural network with back propagation training algorithm has been applied on the input data values. In this context, different training functions like batch training, Batch Gradient Descent with Momentum, Conjugate Gradient Algorithms, Quasi-Newton Algorithms, Levenberg-Marquardt algorithm etc. have been applied for each input and output variable. Finally, performances of these different training functions have been computed and conclusion on selection of training function has been made.

The paper is organized as follows; Section I is devoted to the importance of the research work and a brief literature review is furnished. In section II, a brief discussion on the methodologies used in this work with description of the dataset is presented with a discussion on performance analysis. Section III describes the implementation work. In Section IV, results and the performances are analyzed. Finally, Section V concludes the paper.

II. METHODOLOGY

A. Dataset Description

In this research work, the yeast data set obtained from UCI machine learning repository has been used [12]. The dataset is related with predicting the localization sites of yeast proteins from their amino acid sequences. Each input of the dataset corresponds to a protein whereas the output is the predicted subcellular location of that protein. The dataset consists of 8 attributes and 1 class-name. The attributes are mcg, gvh, alm, mit, erl, pox, vac, and nuc. Each of the attributes has been used to classify the localization site of a protein which is a score (between 0 and 1) corresponding to a certain feature of the protein sequence. The higher the score is, the more possible the protein sequence has such feature. Proteins are classified into 10 classes, these are cytosolic or cytoskeletal (CYT), nuclear (NUC), mitochondrial (MIT), membrane protein without N-terminal signal (ME3), membrane protein with uncleaved signal (ME2), membrane protein with cleaved signal (ME1), extracellular (EXC), vacuolar (VAC), peroxisomal (POX), endoplasmic reticulum lumen (ERL). There is no missing value for attributes and it contains a total of 1484 number of instances.

B. Artificial Neural Networks

Artificial neural network (ANN) follows a computational paradigm that is inspired by the structure and functionality of the brain. The ANN consists of an interconnected group of artificial neurons processing the information to compute the result.

C. Multilayered Feed Forward Neural Network

A Multilayer Feed-forward ANNs (MLFFNN) is made up of multiple layers. It possesses an input and an output layer and also has one or more intermediary layers, called hidden layers. The computational units of the hidden layer are known as the hidden neurons or hidden units.

D. Training a neural network

The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance. There are generally four steps in the training process:

1. Assemble the training data.
2. Create the network object.
3. Train the network.
4. Simulate the network response to new inputs.

1) Back propagation training algorithm

The back propagation algorithm is used to update the weights and biases of the neural networks. As the name suggests, the weights are adjusted backwards through the neural network, starting with the output layer and working through each hidden layer until the input layer is reached. Back propagation network with biases, a sigmoid transfer function layer, and a linear transfer function output layer is capable of approximating any function. Weights and biases are updated using a variety of gradient descent algorithms. The gradient is determined by propagating the computation backwards from output layer to the first hidden layer.

2) Training Functions:

There are a lot of training functions which are used to train a network. Some of them, which are implemented in this work, have been discussed below.

3) Batch Training

Batch training mode trains a network with batch updates. The weights and biases are updated at the end of an entire pass through the input data.

4) Batch Gradient Descent

This network training function updates weight and bias values according to direction of the negative gradient descent. This is the slowest training function.

5) Batch Gradient Descent with Momentum

It provides faster convergence than Batch Gradient Descent. Momentum allows the network to respond to not only the local gradient, but also to recent trends in the error surface. Momentum allows the network to ignore small features in the error surface. Without momentum a network may get stuck in a shallow local minimum.

6) Resilient backpropogation

The purpose of the resilient back propagation training algorithm is to eliminate these harmful effects caused by the magnitudes of the partial derivatives in multilayer feed forward network. Here only the sign of the derivative is used for determining the direction of the weight update. The size of the weight change is determined by a separate update value.

7) Scaled Conjugate back propagation

This is a network training function that updates weight and bias values according to the scaled conjugate gradient method. The scaled conjugate gradient algorithm is based on conjugate directions.

8) Conjugate Gradient back propagation with Polak-Riebre updates(CGP)

Another version of the conjugate gradient algorithm was proposed by Polak and Ribiére. The storage requirements for Polak-Ribiére are larger than for Fletcher-Reeves.

9) Conjugate Gradient back propagation with Fletcher-Reeves updates (CGF)

This conjugate gradient algorithm is usually much faster than other algorithms but the result depends on the problem.

10) BFGS Quasi-Newton Algorithm

There is a class of algorithms that is based on Newton's method, but that doesn't require calculation of second derivatives. These are called quasi-Newton methods. They update an approximate Hessian matrix at each iteration

of the algorithm. The update is computed as a function of the gradient. BFGS quasi-Newton method is among those quasi-Newton methods which provide faster convergence.

11) Levenberg-Marquardt

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. This provides the fastest training with memory reduction feature.

12) One-step Secant Back propagation

Since, in each iteration, the BFGS algorithm requires more storage and computation than the conjugate gradient algorithms, there is need for a secant approximation with smaller storage and computational requirements. The one-step secant (OSS) method is an attempt to minimize the gap between the conjugate gradient algorithms and the quasi-Newton (secant) algorithms. This algorithm requires less storage and computation per epoch than the BFGS algorithm. It requires slightly more storage and computation per epoch than the conjugate gradient algorithms. It can be considered a compromise between full quasi-Newton algorithms and conjugate gradient algorithms.

E. Performance Analysis

The performance measures depicted here are percentage (%) of correct classified sample, mean squared error (mse), regression (R) value, ROC plot.

- Mean squared error (mse) has been defined as the average squared error between the network outputs (t_i) and the target outputs (O_i), as in (1).

$$MSE = \frac{\sum_{i=1}^{n} (t_i - o_i)^2}{n} \text{ --(1)}$$

where n is the number of data.

- The Regression analysis function compares the actual outputs of the neural network with the corresponding desired outputs (targets). It returns the correlation coefficient (R) between them and also the slope and the intercept of the best-linear-fit equation. R can be in the range [0.0, 1.0]. The more the values of R are near to 1.0, the more correct the response of the network.

- The Receiver Operating Characteristic (ROC) is a metric used to check the quality of classifiers. For each class of a classifier, ROC applies threshold values across the interval [0, 1] to the outputs. For each threshold, two values are calculated, the True Positive Ratio (the number of outputs greater or equal to the threshold, divided by the number of one targets), and the False Positive Ratio (the number of outputs less than the threshold, divided by the number of zero targets).

III. IMPLEMENTATION

This research work has been implemented using Matlab 2013.

Step 1: As already stated, yeast dataset [12] consists of 10 numbers of attributes. At first the first attribute (sequence name) is discarded, as this attribute is not necessary for the classification task.

Step 2: The output class names are of non-numeric type for example MIT, CYT, VAC, etc. These are replaced by numeric values 1, 2, 3 etc. The class names with their replaced numeric values are listed in table 1.

Table 1: Class name and numerical value

Class name	Numerical value
MIT	1
NUC	2

CYT	3
ME1	4
EXC	5
ME2	6
ME3	7
VAC	8
POX	9
ERL	10

Here, 8 attributes of the dataset have been taken for input and the last one as class name replaced with a numerical value. Now the dataset is ready to be classified using artificial neural network.

Step 3: One multilayer feed forward back propagation neural network (8 input node, 10 hidden node and 1 output node), with back propagation training algorithm has been implemented. The learning method was supervised. The activation function used here is sigmoid activation function. Number of maximum allowable epochs was 1000.

Step 4: 80% of the dataset samples have been allocated for training and the remaining 10% for validation and 10% is for testing.

Step 5: The network has been trained and tested using back propagation algorithm with various training functions.

IV. RESULT AND DISCUSSION

After testing, the result are computed and listed in Table 2.

Table 2: Performance analysis of different training functions

	Training Algorithm	% of correct classification	Mean absolute error(mse)	Regression(R)
1.	Batch Training	35%	0.109114%	0.247589%
2.	Batch gradient decent	30%	0.141749%	0.209469%
3.	Batch gradient decent with momentum	25%	0.140658%	0.168064%
4.	Resilencebackpropogation	60%	0.057270%	0.603127%
5.	One-step secant backpropagation	58%	0.059252%	0.585095%
6.	Scaled Conjugate back propagation	61%	0.056633%	0.609138%
7.	Conjugate Gradient backpropagation with Polak-Riebre updates(CGP)	56%	0.060453%	0.573083%
8.	Conjugate Gradient backpropagation with Fletcher-Reeves updates (CGF)	59%	0.057875%	0.597687%
9.	BFGS Quasi-Newton Algorithm	56%	0.063088%	0.548289%
10.	Levenberg-Marquardt algorithm	63%	0.050986%	0.658511%

From the Table 2, it is clear that Levenberg-Marquardt training function gives the best result as compared to other functions. Therefore, it is concluded to use Levenberg-Marquardt training function with artificial neural network with back propagation when classification of yeast dataset has to be implemented. The ROC (Receiver operation characteristic) curve and performance plot for the said classification using Levenberg-Marquardt algorithm have been depicted in fig. 1 and fig. 2 respectively.

V. CONCLUSION

In this work, classification of yeast dataset has been made using multilayer feed-forward artificial neural network with back propagation algorithm with various training functions. A study has been made for the search of a proper training function for the classification task of yeast dataset, and it is concluded that Levenberg-Marquardt training function is best suitable for this classification task since it gives the best result as compared to other training functions. The same technique may be used in other classification problems as well. Further work is needed to increase the accuracy of this classification of yeast dataset.

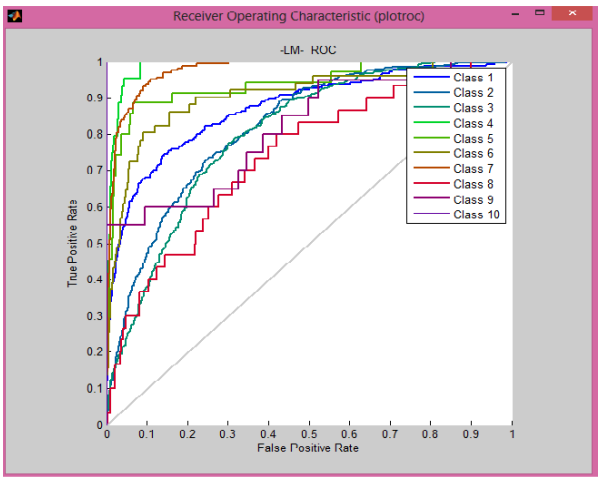

Fig1: ROC curve of classification with Levenberg-Marquardt training algorithm

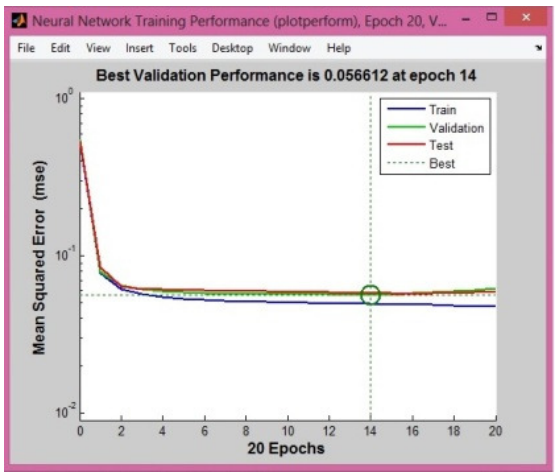

Fig2: performance plot of classification with Levenberg-Marquardt training algorithm

REFERENCES

[1] BB. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, J.D. Watson, Molecular Biology of the Cell, Garland, New York, 1994.

[2] Shavlik, J., Hunter, L. &Searls, D. (1995).Introduction. Machine Learning, 21: 5-10.

[3] Nakai, K., Kanehisa, M.: Expert system for predicting protein localization sites in gram-negative bacteria. Proteins: Structure, Function, and Genetics.11,95-110 (1991).

[4] Nakai, K., Kanehisa, M.: A knowledge base for predicting protein localization sites in eukaryotic cells. Genomics. 14, 897-911 (1992).

[5] Horton, P., Nakai, K.: A probabilistic classification system for predicting the cellular localization sites of proteins. In :Proceedings of Intelligent Systems in Molecular Biology, pp 109-115. St. Louis, USA (1996).

[6] Horton, P. ,Nakai, K.: Better prediction of protein cellular localization sites with the k Nearest Neighbors classifier, pp. 147-152. AAAI Press. Halkidiki, Greece (1997).

[7] Yetian Chen, Predicting the Cellular Localization Sites of Proteins Using Decision Tree and Neural Networks, http://www.cs.iastate.edu/~yetianc/cs572/files/CS572_Project_YETIANCHEN.pdf.unpublished.

[8] Support Vector Machine with the Fuzzy Hybrid Kernel for Protein Subcellular Localization Classification ";Bo Jin, Yuchun Tang, Yan-Qing Zhang, Chung-Dar Lu and Irene Weber; The 2005 IEEE International Conference on Fuzzy Systems;pages 420-423.

[9] M. Barman,Dr. J Palchoudhury, S. Biswas,"A Framework for the Neuro Fuzzy Rule Base System in the diagonosis of heart disease", International journal of Scientific and Engineering Research,vol-4,Issue 11,November 2013.

[10] S.Datta,Dr. J Palchoudhury,"A Comparative Study on the Performance of Fuzzy Rule Base andArtificial Neural Network towards Classification of Yeast Data",International Journal of Information Technology and Computer science,in press.
[11] S.Datta,Dr. J Palchoudhury,"A Framework for Selection of Membership function using Fuzzy Rule Base System for the Classification of Yeast Data", Proceeding of international conference on Emerging trends in Computer science and Information Technology(ETCSIT 2015),Department of Information Technology,KalyaniGovernment Engineering College,West Bengal,India.January,2015.
[12] UCI machine learning repository, :http://archive.ics.uci.edu/ml.