Available online at www.isas.org.in/jisas
JOURNAL OF THE INDIAN SOCIETY OF AGRICULTURAL STATISTICS 69(1) 2015 101-106

Web Based Software for Back Propagation Neural Network with Weight Decay Algorithm

Rakesh Kumar Ranjan, Anu Sharma, A.K. Jha, S.B. Lal and Alka Arora
ICAR-Indian Agricultural Statistics Research Institute, New Delhi

Received 16 August 2013; Revised 25 August 2014; Accepted 02 March 2015

SUMMARY

Artificial Neural Networks (ANNs) are non-linear structures used for prediction and classification problems. ANNs identify and learn correlated patterns between input data sets and corresponding target values. Trained ANNs are used to predict the outcomes of independent variables. Over fitting and under fitting are two major problems that may arise in ANNs. When two or more predictor variables in a model are highly correlated, called as multi-collinearity, they provide redundant information about the response and leads to overtraining. This problem is handled by using ANN with weight decay algorithm. Many software are available for analyzing the data using ANN but either they are very expensive or difficult to use. This study describes a web based software for back propagation neural networks with weight decay algorithm. This software is useful for statisticians and researchers implementing ANNs for various data mining task and facing non-convergence problem.

Keywords: Artificial neural networks, Software, Web, Weight decay.

1. INTRODUCTION

Artificial Neural Networks (ANNs) are non-linear mapping structures based on the function of human brain. The neural network model is mostly data driven and free from any stringent model assumptions prevalent with other statistical methods. ANNs have powerful pattern classification and recognition capabilities (Rajasekaran and 2003). Traditionally regression model and other related statistical techniques have been employed for classification and pattern recognition (Werbos 1974).The most widely used learning algorithm in an ANN is the Back Propagation Neural Networks (BPNN) Algorithm (Jobson 1991). Real life data sets often have noisy inputs and/ or outputs leading to over fitting or under fitting of the ANN model. One of the reason for over fitting of the model is the presence of multi-collinearity among the input variables *i.e.* some of the variables in the data set are linearly correlated or collinear (Carpio *et al.* 2002 and Rognvaldsson 1998). Multicollinearity in neural network model is dealt with two kinds of learning functions used in training *i.e*. Scaled Conjugate Gradient and Backpropagation with Weight Decay (Carpio *et al.* 2002, Kärkkäinen 2002, Svozil 1997, Krogh 1992). In this article we have selected BPNN models with weight decay algorithm to deal with multicollinearity.

Statistical software packages have been used to perform the artificial neural network analysis for prediction. An extensive review for the available software for ANN were done and it was found that a large number of software are available on open domain. Amygdala, Annie, Artificial Intelligence Recurrent Asymmetric Networks (NARIA), Lightweight Neural Network++, Neuroph, Simbrain, Torch, University of Hertfordshire Neural Network Software, Neural

Corresponding author: Anu Sharma
E-mail address: anu@iasri.res.in

Network Leaves Recognition and Neuropilot Project are among stand-alone open source software simulating neural networks. Commonly used artificial neural network simulators include the Stuttgart Neural Network Simulator (SNNS), Emergent, JavaNNS, Neural Lab and NetMaker, Neuron, GENESIS, NEST and Brian. Other simulators are XNBC and the NN Toolbox for MATLAB. Some of the freeware/ shareware and stand-alone software available are Cortex, Genesis and NetMaker. Among commercial software SPSS, STATISTICA, SAS and MATLAB, AlyudaNeuroIntelligence, NeuroShell Predictor, NeuroSolutions, Tradecision and EasyNN are available (Neural network software 2014).

Some of these commercial software provide extensive features including ANN functionalities but they also need expertise for source code development. It is also to be noted that it is required to be paid or purchased for using these software. The advancements on the internet technology front have expanded the potential for statistical packages. Availability of an online software for this purpose would provide the users to run this software on a web browser with internet connectivity. This kind of software development environment would allow datasets and analyses to be shared among researchers to communicate with each other quickly and conveniently. This paper describes the functionality and features of a web based software named "Web Based software for Back Propagation Neural Network with Weight Decay algorithm (WBPNN-WD)". It is online software accessible through any internet browser.

2. SOFTWARE ARCHITECTURE

WBPNN-WD is based on three-tier client-server architecture. It has three layers namely, User Interface or Client Side Interface Layer (CSIL), Business Logic Layer or Server Side Application Layer (SSAL) and Data Access Layer or Database Layer (DBL). Fig.1 shows the architecture of WBPNN-WD.

1. *Client Side Interface Layer (CSIL):* CSIL has been implemented using Hyper Text Markup Language (HTML) and JavaScript (Willard 2009, Powell and Schneider 2004). The CSIL consist of forms for accepting information from the user and validating those forms using JavaScript. Web page designing was done using Cascading Styling Sheet (CSS).
2. *Server Side Application Layer (SSAL):* Application layer has been implemented using (ASP.NET). The ASP.NET provides the web developers with a framework to create dynamic content on the server, which is secure, fast and independent of server platform (Evjen *et al.* 2008). C#.NET language is used to develop code behind pages for various web forms (Balagurusamy 2010)
3. *Database Layer (DBL):* Data base layer is implemented using MS-Access database for storing user's information (*i.e*. login name, login password).

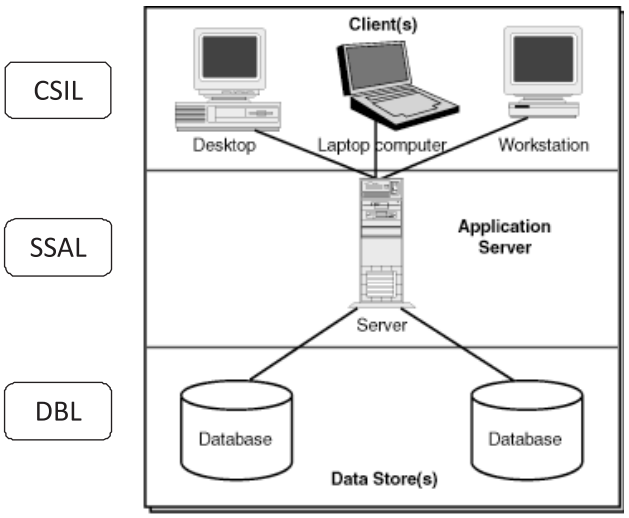


Fig.1: Three-Tier architecture of WBPNN-WD

WBPNN-WD is developed using Visual Studio-2010; an Integrative Development Environment for developing ASP.NET based web applications (Griffiths *et al.* 2003). Programming has been done using object oriented programming language named C#. Database connectivity has been done with ADO.NET which provides improved support for the disconnected programming model.

3. SOFTWARE PROCESS MODEL AND DESIGN

Software process model is the framework that describes the activities to be performed at each stage of a software development project.Waterfall model was used during development process of software for computation of WBPNN-WD. In this approach, the whole process of software development was divided into five separate process phases namely requirement

analysis and definition, software design, implementation and testing, integration and system testing and operation and maintenance (Goswami *et al*. 2010). Sequence diagrams model is the flow of logic within system in a visual manner. It enables documentation and validation of logic as shown in Fig. 2 (Sommerville 2009).

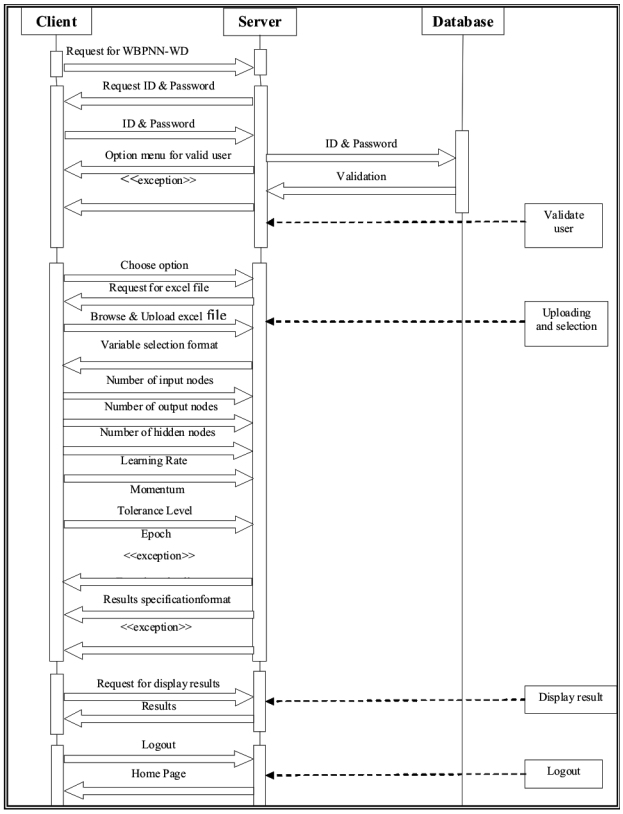


Fig. 2. Sequence Diagram for WBPNN-WD Computation

Table 1. Modules in Software

Module Name	Description
Login	Provide facility of login to users
WBPNN-WD computation	The main module, which provide WBPNN-WD (including dependent variables and independent variables)
Help	Provide online help about software
Contact Us	Contact details of developer team
Sample Data Download	Download sample data to understand format of input data
Signup	Provide facility of sign up to new user
Changed Password	An option for change of password
Password Recovery	Provide facility to recover password

The system for WBPNN-WD computation was broken down into manageable parts called modules. Modules identified for WBPNN-WD application are presented in Table 1.

Database for the system is maintained using MS-Access at server level. Database contains independent table namely login table. The schema design for table is presented in Table 2.

Table 2. Login

ID	Field name	Type	Description
1	ID	Auto Number	Primary Key
2	User ID	Text	Login id of user
3	User name	Text	First name of user
4	User surname	Text	Surname of user
5	User password	Text	Password of user
6	Security question	Text	Question to user
7	Security answer	Text	Answer of question provided by user

4. FEATURES OF SOFTWARE

WBPNN-WD is a web based software that is freely accessible to users on internet. User authentication is needed to ensure security. The Home page (Fig. 3) of the software presents the user with a brief welcome note on the software.

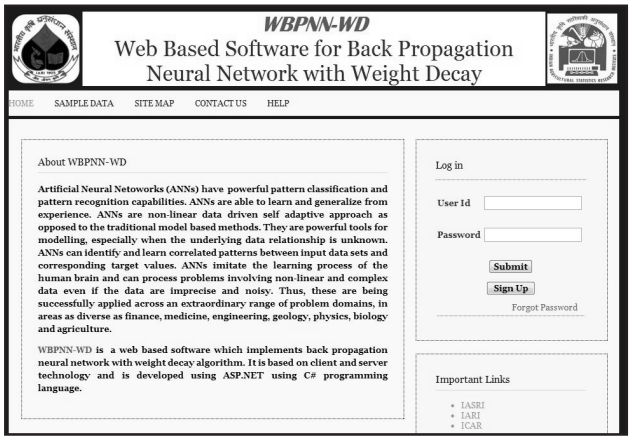


Fig. 3. Home page of WBPNN-WD

The home page has links to “Sample Data”, “Site Map”, “Contact Us” and “Help”. “Sample Data” provides the soil dataset for download by users for using the software. A detailed description of data attributes, their coding schemes and actual data is

provided to make users understand the data format. "Help" gives the user knowhow of using the software and the theoretical background of the back propagation neural network with weight decay algorithm.

4.1 User Management

This module provides the facility for creating a new user and to change and retrieve the password. A new user can register by clicking on the "Sign Up" button and filling in all the required details for registration process. Options are also provided to change the existing user password and to retrieve the password in the situations when user forgot the password.

4.2 Input Data Handling

This module has been designed and developed for uploading the data as an Excel file. The Excel data sheet

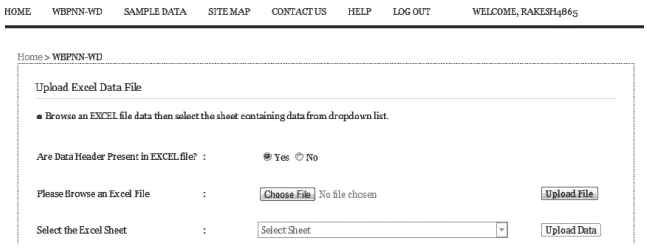


Fig 4. Uploading excel sheet

should contain data for training and testing without any missing or blank cell.

The software also provides the facility for normalization of data. Input data file is automatically scanned for columns in the data set which are not normalized. Such columns are shown in different colour to enable user to normalize the data using the given option (Fig. 5). The method used for data normalization is:

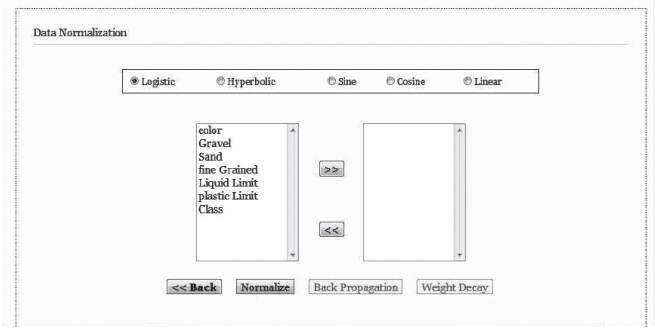


Fig. 5. Data Normalization page

$$Z = \frac{X - \mu}{\sigma}$$

where

X = Value of random variable,
μ = mean,
σ = standard deviation

4.3 Prediction/ Classification using WBPNN-WD

This section describes the various steps involved in performing predictions and classification using software.

Step 1: Selecting Training and Test Data

This step involves the selection of training and test data from the given dataset. The software provides the facility for selecting data for training and test data with the limit that user can select at the most 60% data for training and 40% data for testing. Records from the beginning are taken as training data and records from the end of the data file are taken as data for testing.

Step 2: Selecting the Activation Function

This step involves the selection of activation function. It determines the relationship between inputs and outputs of the network. The software provides the options for sine, cosine, hyperbolic, sigmoid and linear activation functions. User can select any of these activation functions.

Step 3: Variable Selection

Proper selection of variables is important to get appropriate results. WBPNN-WD provides this facility through "Variable Selection Wizard". It is mainly associated with choosing the dependent and independent variables. Software has two lists, one for the selection of dependent and other for the selection of independent variables. User can select desired variables and by clicking on ">>" button moves the selected variables in the list containing selected variables.

Step 4: Learning Scheme Specification

Learning scheme for using back propagation neural network with weight decay algorithm training requires the specification of the various parameters like number of input nodes, number of output nodes, learning rate, the momentum rate, number of epochs,

number of hidden layers, number of hidden nodes in the hidden layer and weight decay. WBPNN-WD allows the user to enter these values. However, for each of these parameters default values are provided by the software. The description of these values is given below:

- The default value for number of input nodes is equal to number of independent variables used in prediction.
- The default value for number of output nodes is equal to number of dependent variables used in prediction.
- WBPNN-WD assumes only one hidden layer for the purpose of calculations.
- Number of hidden nodes are calculated using the formula:

$$\text{Number of hidden node} = \sqrt{\text{Number of inputs} \times \text{Number of output}}$$

- Momentum can take any value between 0.5 and 0.9.

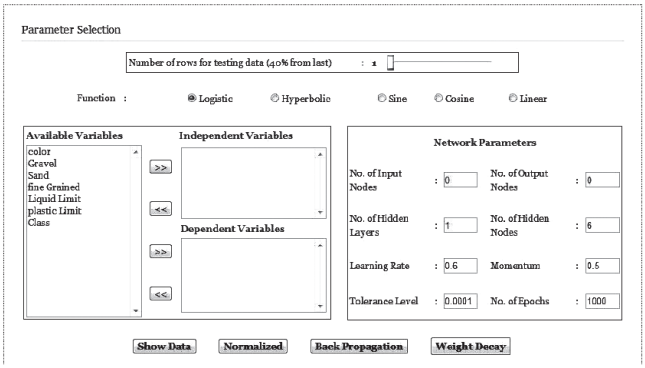


Fig. 6. Parameter Selection Page

Fig. 6 shows the web form for entering the parameters for doing prediction or classification.

4.4 Output Handling

There are two tables of outputs. The first table contains the weights of synapses connecting input neurons to hidden neurons. Each column shows the input neuron (starting form 0) and each row shows the hidden neuron (number of rows shows the total number of hidden neurons). The second table contains the weights of synapses connecting hidden neurons to

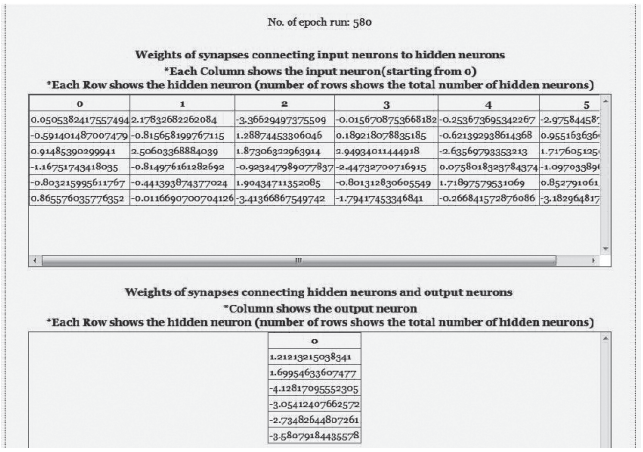
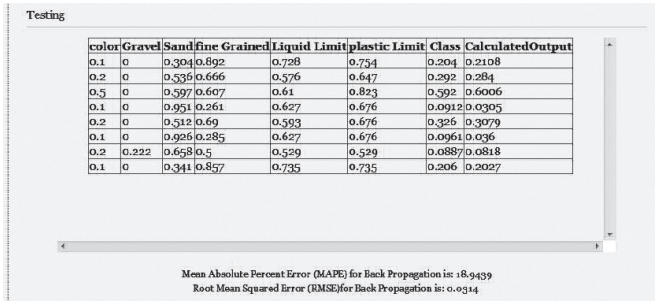


Fig. 7. Final weights calculation page of WBPNN-WD

output neurons and each row shows the hidden neuron (number of rows shows the total number of hidden neurons). Fig. 7 shows the outputs containing various weights as generated by WBPNN-WD.

In testing, there are two types of outputs namely tabular and graphical outputs. In tabular output, there are number of columns each corresponding to one input and an additional column for actual output (*i.e.* class) and in last column, there is calculated output or predicted output. In graphical output, there is a chart between actual and predicted values by software. Fig. 8 and 9 shows the tabular and graphical output generated by the software. The software also calculates



color	Gravel	Sand	fine Grained	Liquid Limit	plastic Limit	Class	CalculatedOutput
0.1	0	0.304	0.892	0.728	0.754	0.204	0.2108
0.2	0	0.536	0.666	0.576	0.647	0.292	0.284
0.5	0	0.597	0.607	0.61	0.823	0.592	0.6006
0.1	0	0.951	0.261	0.627	0.676	0.0912	0.0305
0.2	0	0.512	0.69	0.593	0.676	0.326	0.3079
0.1	0	0.926	0.285	0.627	0.676	0.0961	0.036
0.2	0.222	0.658	0.5	0.529	0.529	0.0887	0.0818
0.1	0	0.341	0.857	0.735	0.735	0.206	0.2027

Fig. 8. Tabular report generated for actual and predicted values in WBPNN-WD

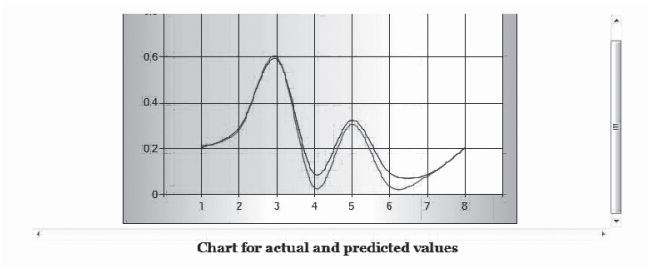


Fig. 9. Chart for actual and predicted values page of WBPNN-WD

the root mean squared error (RMSE) and mean absolute percent error (MAPE) values for comparing the performance of model using actual and predicted values. Use can save the output to Excel sheet.

4.5 Help

An online help facility is provided which includes the details of how to use the software and also the basic theoretical background of the back propagation neural network with weight decay algorithm.

5. CONCLUSION

WBPNN-WD provides online facility for prediction using ANN with weight decay. The software is user friendly and does not demand expertise of computer programming. User can register, login, compute WBPNN-WD, see results and save result in Excel file for further processing using client interface online. Administrator interface of the software helps in development and maintenance of user database.

REFERENCES

Balagurusamy, E. (2010). *Programming in C# - A Primer*. Tata McGraw Hill Education Private Limited, New Delhi.

Evjen, B., Hanselman, S. and Rader, D. (2008). *Professional ASP.NET 3.5.* Wiley Publishing, Indiana.

Goswami, A., Arora, N. and Sharma, A. (2010). *Fundamentals of Software Engineering*. Lakhanoal Publishers, Amritsar, India.

Griffiths, L., Flanders, J. and Sells, C. (2003). *Mastering Visual Studio.NET*. O'Reilly and Associates, Inc, USA.

Powell, T.A. and Schneider, F. (2004). *JavaScript: The Complete Reference*. Tata McGraw Hill Private Limited, New Delhi, India.

Rajasekaran, S. and VijayalakshmiPai, G.A. (2003). *Neural Networks, Fuzzy Logic, and Genetic Algorithms Synthesis and Applications.* Prentice Hall, New Delhi.

Sommerville, I. (2009). *Software Engineering A Practitioner's Approach.* Tata McGraw Hill International Edition Publishers, New Delhi, India.

Werbos, P.J. (1974). Beyond regression: New tools for prediction and analysis. *Behavioral Sci.* **Vol. 1,** 321-342.

Willard, W. (2009). *HTML: A Beginner's Guide*. Tata McGraw Hill Education Private Limited, India.

Rognvaldsson, T.S. (1998). A simple trick for estimating the weight decay parameter. *Neural Networks: Tricks of the Trade.* Lecture Notes in Computer Science Vol. 1524, pp 71-92. doi. 10.1007/3-540-49430-8_4.

Carpio, K.J.E. and Hermosilla, A.Y. (2002). On multicollinearity and artificial neural networks. *Complexity International*, Vol. 10, Paper-ID: hermos01. URL: http://www.complexity.org.au/ci/vol10/hermos01/ .

Kärkkäinen, T. (2002). MLP in layer-wise form with applications to weight decay. In: *Neural Computation,* Ausgabe/Number: **6(14)** Erscheinungsjahr/Year: 2002. Seiten/Pages: 1451-1480.

Svozil, D., Kvasnickab,V. and Pospichalb, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics Intell. Lab. Sys.* **39(1)** 43-62.

Krogh, A. and Hertz, J. (1992). A simple weight decay can improve generalization. *Adv. Neural Inform. process. Sys.,* 4.

Neural network software. (2014). In Wikipedia, The Free Encyclopedia. Retrieved 06:16, April 29, 2014, from http://en.wikipedia.org/w/index.php? title=Neural_network_software&oldid=594377161.