MATLAB BASED ARTIFICIAL NEURAL NETWORK ALGORITHM FOR VOLTAGE STABILITY ASSESSMENT

D.Thukaram
Dept. of Electrical Engineering
Indian Institute of Science
Bangalore,Karnataka,India
dtram@iisc.ernet.in

Sukumar Kamalasadan **Adel A. Ghandakly**
Dept.of Electrical Engineering and Comp.Science
University of Toledo
Toledo,Ohio,43606,USA
skamalas @eng.utoledo.edu **adel.ghandakly@utoledo.edu**

Abstract

This paper presents an analysis and selection of various techniques/parameters in developing a suitable feed forward neural network for performing static voltage stability analysis using MATLAB 5.2 NEURAL NETWORK TOOLBOX. Initially, an approach based on the input-output relation of real/reactive power and voltage vectors for generator as well as load buses with the voltage stability Index-L is used to train feed forward neural network. Further various toolbox functions such as different types of feed forward neural network, training functions, activation functions, learning functions, initialization functions and performance functions available are tested and the most suitable combination is selected. We summarize the analysis comparing the results with each training function and conclude the suitability of ANN training for Voltage Stability Assessment. Development of the proposed approach shows the ability of ANN in voltage stability assessment and improvement using the well known L-index. This approach can be used in Energy Management Center and shows the vulnerable buses as indicated by index-L at the same time reduce the computational complexity in mathematical calculations.

1 INTRODUCTION

Voltage stability is concerned with the ability of the power system to maintain acceptable voltages at all system buses under normal conditions as well as after being subjected to a disturbance. Thus the analysis of voltage stability deals with finding the voltage levels at all buses in the system under different loading conditions to ascertain the stability limit and margin. Several works have been conducted recently for the prediction of voltage stability and collapse based on the steady state analysis by conventional ways [1-4]. Some of these focus on static voltage stability analysis with the use of instability measuring indicators. Among them the suitable and simple way of finding the stability margin and the limit by means of the popular L-index is as described in [5]. This index gives a sufficiently accurate and more practical means of the assessment and can express the stability analysis in a simple way. In this paper an algorithm developed based on the reference [5-6] is used for training and simulating the various parameters in the MATLAB Neural Network Toolbox.

Application of Artificial Neural Network (ANN) to the above mentioned problem has attained increasing importance mainly due to the efficiency of present day computers. Moreover real-time use of conventional methods in an energy management center can be difficult due to their significant large computational times. One of the main features, which can be attributed to ANN, is its ability to learn nonlinear problem offline with selective training, which can lead to sufficiently accurate online response. Recently some works on ANN approach to voltage stability assessment and improvement has been proposed and various neural network combinations have been used for solving the problem [10-13]. The ability of ANN to understand and properly classify such a problem of highly non-linear relationship has been established in most of them and the significant consideration is that once trained effectively ANN can classify new data much faster than it would be possible with analytical model. Two of the relevant works are explained below.

L. Srivastava et al [7] proposes a parallel self-Organizing Hierarchical Neural network (PSHNN) to predict maximum loadability margin. Here the input to PSHNN was selected based on a) Entropy concept and b) real and reactive power obtained from the critical buses. The input data is then normalized and the maximum loadability is estimated using the proposed method. The method uses four-stage neural network with revised back-propagation algorithm, along with forward-backward training of stage neural networks. A static voltage stability analysis modeling the whole system totally considering voltage stability index and locally by defining appropriate voltage margins to detect the area where instability occurs was proposed in [8]. The neural network used here is the Layered Feed Forward Neural Network (LFNN) where the mentioned indices is the output for a predefined set of input variables which influence the most on voltage stability.

In this paper an investigation is carried out for the application of ANN to find the voltage stability Index-L using a developed training algorithm for all the buses in the system. The main purpose is to explore the ability of MATLAB Neural Network Toolbox functions and parameters to learn the analytically simulated database and to perform various non simulated conditions with acceptable degree of accuracy. Once when a suitable combination of network functions and parameters are found out this can be used online for practical power system networks. The setup even can be extended to be able to use in an energy management center for online establishment of voltage stability margin and to find out the limits of each bus.

The paper is organized as follows. In section 2 methodology of testing various combinations of network parameters is explained. Section 3 shows the network architecture and analyses the TOOLBOX functions to find the most suitable combination. Results and discussion is done in section 4 and section 5 concludes the work.

2 METHODOLOGY

The approach used for investigating the different parameters and functions in the MATLAB TOOLBOX, is given by the following algorithm.

Step1: A conventional voltage stability algorithm is run with the test system for simulated loading conditions. For this first the base case and the maximum loading conditions of the test system are determined using the conventional software [6]. Then the load conditions are varied from base case till full load and training samples are generated.

Step2: Create a database for the Input vector in the following form $[Pg^T, Qg^T, Vg^T, P_l^T, Q_l^T, V_l^T]^T$ where P_g, Q_g, P_l and Q_l are the real and reactive power in generator as well as load buses respectively and V_g and V_l are bus voltage at generator and load buses. Moreover, target vector is created in the form of L-indices for the corresponding Input vectors.

Step 3: Find the minimum and maximum values of the input vector, remove redundancies and normalize to suit to train the selected feed forward neural network.

Step 4: Select the set of parameters to train the network. The main parameters selected are number of epochs, learning increment and rate, performance goal with Mean Squared Error (MSE) and minimum and maximum gradient.

Step 5: Train the network based on a set of activation functions and number of neurons. The number of neurons in each layer is varied initially and optimum combination is found out depending on the training period and performance error.

Step 6: Check the performance of the network for behavioral accuracy. If not change the activation functions and test the network again. Find the most suitable combination of the activation function. Behavioral accuracy depends on the uniformity in values of L-indices at all the buses. It can happen that the network gives output which is accurate for some buses but may be unacceptable on some other buses.

Step 7: Change the training function keeping same transfer functions and optimum number of neurons in each layer.

Step 8: Find the most suitable network based on the simplicity least possible Mean Square Error and computational speed. Further use various test functions to confirm the effectiveness of the proposed neural network. At this state the functions and all the parameters are finalized for this combination.

3 NETWORK ARCHITECTURE AND ANALYSIS OF THE FUNCTIONS

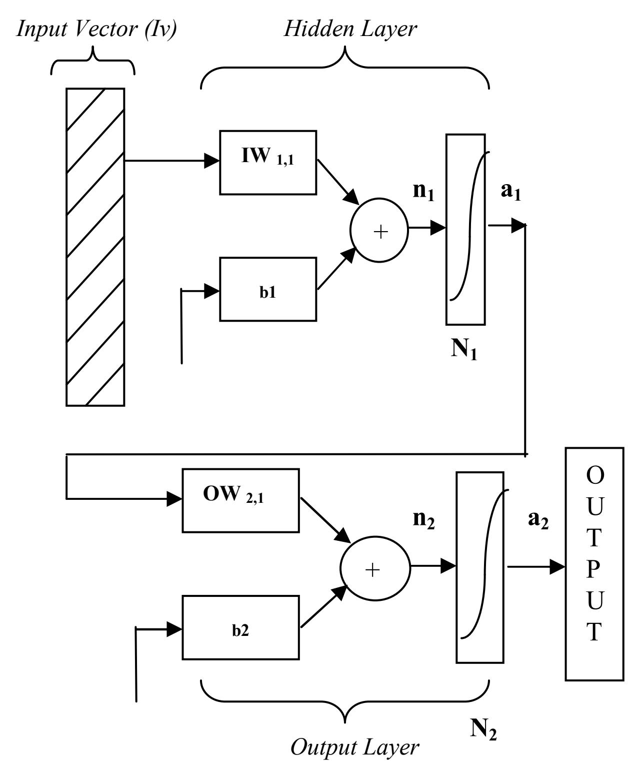

Fig. 1: Proposed Feed Forward Neural Network (FFNN) architecture

Figure 1 shows the proposed feedforward neural network. The architecture consists of an input layer, a hidden layer and an output layer. Input vector is fed to the Input layer for all the buses of the selected system. The network weights and biases are adjusted using the 'dotprod' function in the toolbox and adaptation is done using 'adaptwb' function which changes both weights and biases. First to obtain the best combination of number of neurons, training and activation function a test system is developed and the toolbox parameters are applied.

Following Toolbox functions are analyzed based on the above methodology [9]:

- Neural network architecture and types
- Training functions
- Activation functions
- Learning function
- Initialization functions
- Performance functions

It was found that Quasi Newton Algorithm *based on Lavenberg Marquardt (LM) training algoritm* performs the best considering the criteria's as Mean Square Error for the L-index values at each bus, training time and overall accuracy. Moreover, sigmoid activation functions ('transig') for the Input and hidden layer and the linear activation functions ('purelin') for the output layer was the most suitable [9].

Following functionality are incorporated in the developed ANN algorithm. Initially from the selected Input data set the minimum and maximum values are found out. Then the FFNN architecture is selected defining the number of neurons, training and activation functions. Further the Input data is normalized and filtered for redundancy. In the next step, network training parameters such as learning ratio (0.08), learning increment (2.00), number of epochs (400), Parameter Goal (1e-5), Minimum and Maximum Gradient (1e-10 and 1e-10 respectively) have been set based on the analysis done. After training the network for the set of samples, network performance is evaluated with a new set of non simulated data and compared the output with the L-index values obtained from the conventional algorithm. The trained network nomenclature is assigned and further a new set of NN is trained for contingencies.

4 RESULTS AND DISCUSSION

For the case study IEEE 30 bus system is used. Keeping the suitable activation functions, each of the training function is analyzed for two different conditions. Initially the training was conducted for same network with constant number of neurons in all the layers to identify the performance of each function and also to calculate the CPU time. It was observed that the computational time for 'TRAINGDX' and 'TRAINRP' functions are same (3 seconds) but the Mean Square Error as 0.0732 and 0.01454 respectively. Thus it can be concluded that 'TRAINRP' function is the most suitable one if both the computational time is considered as the most critical criteria. Also function 'TRAINCGB' found to be the optimal as the error function is 0.00679 and the computation time is 5 seconds for ten epochs.

Further, the neurons are varied and the performance is analyzed to improve the error function. It can be seen that 'TRAINLM' function is the optimal as although the number of neurons are more than 'TRAINCGB' it gave better performance goal of 0.000227 with a computational time of 4 seconds for ten epochs. Table 1 compiles the performance of each of these training functions for same number of neurons (cond.1) and best suitable neurons in each layer (cond.2).

Figure 2 shows the performance of TRAINLM function to calculate the L-index on the test system. The line indicates the L-indices for each bus for the 30-bus system calculated for a particular operating condition, which ranges from 0-0.12. The circles indicate the trained network response to calculate the index for each bus for this untrained set. It is observed that the L-indices produced matches closely with the actual values. To check the response of the network trained a post regression analysis has been conducted between the target and the output. It was observed that the correlation coefficient is 0.999 while training with 'TRAINLM'. Figure 3 shows the slope and Y-intercepts of the analysis.

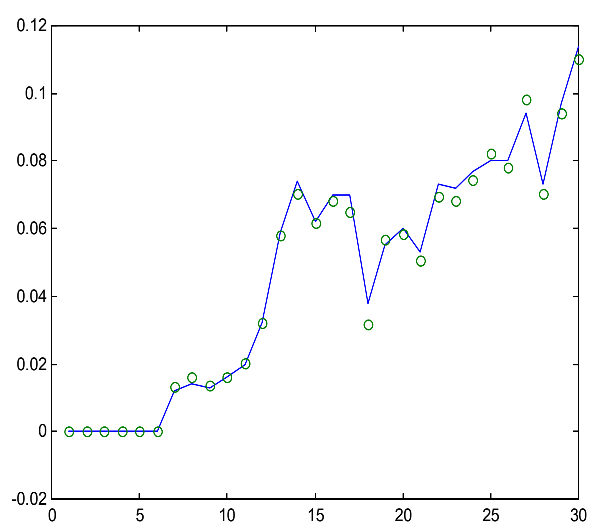

Fig.2: L-index value comparison for IEEE30 bus system

TABLE 1
PERFORMANCE COMPARISON FOR VARIOUS TRAINING FUNCTIONS

Comparison between various most suitable training Functions in BackPropagation Network			
Type of Network	Descripton of Training Pattern	Training Function	Perf. Goal (MSE)
Condition 1			
FeedForward	(12,10,1) Tansig,Tansig Purelin	Traincgb	0.0068
		Trainoss	0.0077
		Trainbfg	0.0065
		Traincgf	0.0110
		Traincgp	0.0072
		Trainscg	0.0169
		Trainrp	0.0145
		Traingdx	0.0732
Condition 2			
Feedforward	(24,20,1)Tansig,Tansig,Purelin	Traincgb	0.0018
	(32,30,1)Tansig,logsig,Purelin	Trainoss	0.0059
	(18,18,1)Tansig,Tansig,Purelin	Trainbfg	0.0046
	(24,20,1)Tansig,logsig,Purelin	Traincgf	0.0037
	(24,20,1)Tansig,Tansig,Purelin	Traincgp	0.0038
	(24,20,1)Tansig,Tansig,Purelin	Trainscg	0.0032
	(24,20,1)Tansig,Tansig,Purelin	Trainrp	0.0075
	(30,25,1)Tansig,Tansig,Purelin	Traingdx	0.035
	(30,25,1)Tansig,Tansig,Purelin	Trainlm	0.0002

Legend :- (x,x,x) Tansig,Tansig,Purelin --> (Nh,Nh,No) Transfer Function

5 CONCLUSIONS

Highly nonlinear problems like this can be solved using single hidden layer neural network and propose its ability in online assessment of voltage stability and margin. It is found that in such an analysis, the training algorithm is the most important factor in the performance and accuracy of the network, since any variation to be done in the number of neurons and the network parameters is limited. Post regression analysis and the comparison of the output at each bus shows that the output obtained are sufficiently accurate if well-organized training is conducted reducing the redundancy and normalizing the inputs as well as outputs. Moreover, the performance of each of the training algorithms depends also on the size of the power system network under consideration.

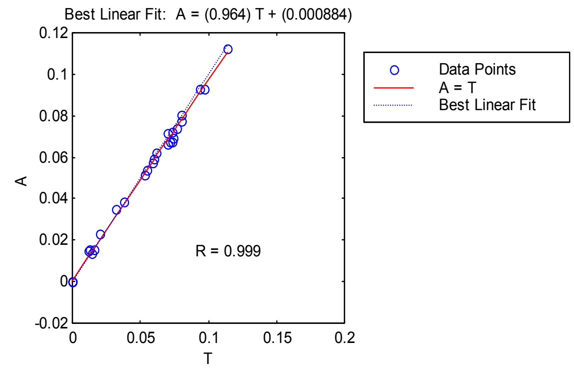

Fig.3: Post Regression Analysis on TRAINLM

5 REFERENCES

[1] N.Yorino, H.Sasaki, Y. Masuda, et al., "An investigation on voltage stability problems", IEEE Trans. on Power Systems, Vol7, No2, pp 600-607, May 1992.

[2] Y Tamura et.al., " Relationship between voltage instability and multiple load flow solutions in electric power systems" IEEE Trans. on Power Systems. Vol PAS – 102, No.5, pp.1115-1125, May 1993.

[3] O Crisan and M.Liu "Voltage Collapse prediction using an improved sensitivity approach" Elec. Power System Research, 28, pp.181-190, (1984).

[4] P-A Lof et.al., "Voltage Stability indices of the stressed power system" IEEE Trans on Power Systems, Vol 8, No.1, pp326-335, Feb 1993,.

[5] Bansilal, D.Thukaram and K.Parthasarathy "Optmial rective power dispatch algorithm for Voltage stability improvement" Electrical Power and Energy Systems, Vol 18, No.7, pp461- 468, 1996.

[6] D.Thukaram et.al., "Voltage stability Improvement : Case Studies of Indian Power networks" Electric Power Systems Research 44 pp.35-44 , (1998).

[7] L.Srivastava, S.N. Singh, J.Sharma "Estimation of loadability margin using Parallel Self-Organising Hierarchical Neural Network", Computers and Electrical Engineering 26, pp. 151-167 (2000).

[8] M. La Scala, M. Trovato and F. Torelli "A neural Network based method for voltage security monitoring", IEEE Trans on Power system Vol 11, No.3, pp1332-1341, August 1996.

[9] The Mathworks Inc. Neural Network TOOLBOX, Users guide-For using in MATLAB, Jan 1998.

[10] B. Jeyasurya," Artificial Neural Networks for power system steady–state voltage instability evaluation", Electric Power Systems Research, Vol 29, pp.85-90, 1994.

[11] A.A.El-Keib and X.Ma, "Application of artificial neural networks in voltage stability assessment", IEEE Trans on Power Systems, Vol.10, No.4, pp.1890-1896, November 1995.

[12] D.Salatino,R.Sibrizzai,et al., "Online voltage stability assessment of load centers by using neural Networks", Electric Power Systems Research, Vol 32, pp.85-90 , (1995),.

[13] Y.H.Pao, D. J. Sobajic, "A Neural Network based dynamic Security Assessment for electrical Power System", IEEE Trans On Power System, Vol. 4, No 1, pp-220-228, Feb 1989.