Power System Security Assessment Using Neural Network

Mohd Fathi b. Zakaria
Faculty of Electrical Engineering
Universiti Teknologi MARA Malaysia
40450 Shah Alam, Selangor, Malaysia
e-mail: fat86espy@yahoo.com

Abstract — **One of the most significant considerations in applying neural networks to power system security assessment is the proper selection of training features. Modern interconnected power systems often consist of thousands of pieces of equipment each of which may have an effect on the security of the system. Neural networks have shown great promise for their ability to quickly and accurately predict the system security when trained with data collected from a load flow using Newton Raphson technique. A case study is performed on the IEEE 6-bus system to illustrate the effectiveness of the proposed techniques. This paper presented an application of Artificial Neural Network (ANN) in steady state stability classifications. A multi layer feed forward ANN with Back Propagation Network algorithm is proposed in determining the steady state stability classifications. The classification is divided into two, which is stable and unstable state. Extensive testing and training of the proposed ANN based approach indicates its viability for power system steady state stability classification assessment.**

Index terms - Steady state stability, Artificial Neural Network, Back Propagation Algorithm, Power system.

I. INTRODUCTION

Power System Security and Contingency evaluation is one of the most important tasks encountered by planning and operation engineers of bulk power systems. In power system planning, contingency analysis is used to examine the performance of a power system and the need for new transmission expansion due to load growth or generation expansion. In operation, contingency analysis assists engineers to operate the power system at a secure operating point where equipment are loaded within their safe limits and power is delivered to customers with acceptable quality standards. In this type of analysis the objective is to identify the instability in power system if load power change. Identification of these contingencies and the determination of corrective actions often involve exhaustive load flow calculations [1].Contingency analysis is an important aspect of power system security assessment. As various probable outages compose a contingency set, some cases in the contingency set may lead to transmission line overloads or bus voltage limit violations during power system operations. Such critical contingencies should be quickly identified for further detailed evaluation or where possible, corrective measures taken. A set of security analysis functions is usually developed to help the operator monitor and control the security of the power system. These functions involve assessing the security level of the variables obtained from outage studies and control to raise the security level of the system. Static security is defined as the ability of the system to reach a state within the specified secure domain following a contingency impact on the system operation. The main issues in security assessment are the fast identification of the set of critical contingencies and their evaluation related to the severity level and to predict future vulnerabilities [1].

Artificial neural network (ANN) methods are efficient computing models with the ability to solve nonlinear pattern matching problems. They can capture the inherent non-linearity in the input patterns and use them for classification. Therefore, ANN based-method for contingency screening is a good alternative. The paper presents the design of an ANN for fast line-flow and power contingency screening. A multi-layer perceptron network with back propagation learning technique is used for line flow and power contingency screening. The proposed method was tested on the standard IEEE 6-bus system.

II. NEURAL NETWORK DESIGN

The neural network used in this work is the feed forward structure. It consists of three layers, one for the input, one hidden and one for the output. The size of the input layer was determined by the size of the input pattern. The information to the neurons is passed to the hidden layer in a weighted form. The neurons in the hidden layer process this information with a nonlinear function. The hidden neurons pass the information to the output neurons, which are linear current nodes. The data of each study was normalized before training the neural network. After the data normalization, it was divided into two groups, one used for training and the other for testing. For proper neural network performance, test data and training data must be different although belonging to the same source. During the training, the neural network was closely monitored to prevent network over-training, or saturation. The presence of any of these problems would have resulted in unreliable results. When a neural network memorizes the training data, it reproduces acceptable results for patterns that have used during the training, but unacceptable results with high errors when tested on unseen patterns [2]. There are different techniques to ensure that a neural network has learned and not memorized. They are all based on the fact that a properly trained neural network should respond with equal error measures to both training and testing patterns [1].

A. MULTI-LAYERED PERCEPTRON

The multi-layered perceptron (MLP) [3] has been chosen because of its excellent training convergence characteristics, particularly on such a huge dimensionality. Another feature that makes the MLP more ideal for this type of problems is its ability to recognize unknown patterns without retraining the neural network. MLPs are trained by the popular supervised learning algorithm known as the error back-propagation algorithm [4].

B. BACK PROPAGATION NETWORK ARCHITECTURE

Backpropagation is an algorithm used to teach feed forward. It works by providing a set of input data and ideal output data to the network, calculating the actual outputs and backpropagating the calculated error (the difference between the ideal and actual outputs) using gradient descent. This is useful for learning specific patterns of input and output data in order to be able to reproduce them afterwards even from slightly different input data [5]. Neural networks are able to learn any function that applies to input data by doing a generalization of the patterns they are trained with. The network consists of three layers:

i. Input
ii. Hidden
iii. Output

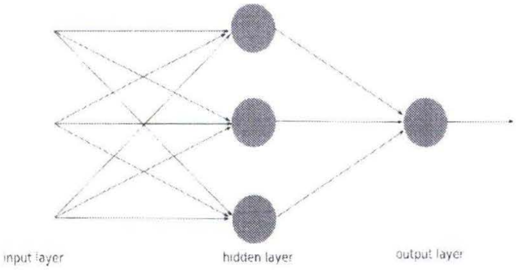

Figure 1: BPN Basic Architecture

During the training, several sets of input are taken and the corresponding output vector s is considered. Each layer has a weight matrix **W**, a bias vector **b**, and an output vector **a**. Then training phase is used to obtain the accurate weight and bias between the input, hidden and output layer before apply to the new input.

The simplest implementation of backpropagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly – the negative of the gradient [6]. One iteration of this algorithm can be written:

$$X_{k+1} = X_k - \alpha_k g_k \tag{6}$$

X_k = Vector of current weights and biases
g_k = Current gradient
α_k = Learning rate

The training process modifies the weights associated with the various interconnections and finds the optimal weights so that the training error E converges to a predetermined value which insures the actual error within desired accuracy. The network is presented with the training patterns and the network is trained when the error E is minimized [7, 8]. The error E is defined as:

$$E = \frac{1}{2}\sum[tn - On]^2 \tag{8}$$

E = Error
tn = Desired output from neuron
On = Output from neuron

The steps of the training algorithm for multilayer:

1. Initialize the neural network parameters, weight and biases using initializing routines and at the same time define the structure of the network.

2. Define the parameters associated with the algorithm like error goal, maximum number of iterations (epochs) etc.

3. Train until obtain the correctly output.

III. METHODOLOGY

Figure 2 shows the step of the neural network from the generating data until to check the method of Newton Raphson and Neural Network have similar result or not. In this step the hard part is training and testing which need perform properly and need more patients to obtain the accurate result desired.

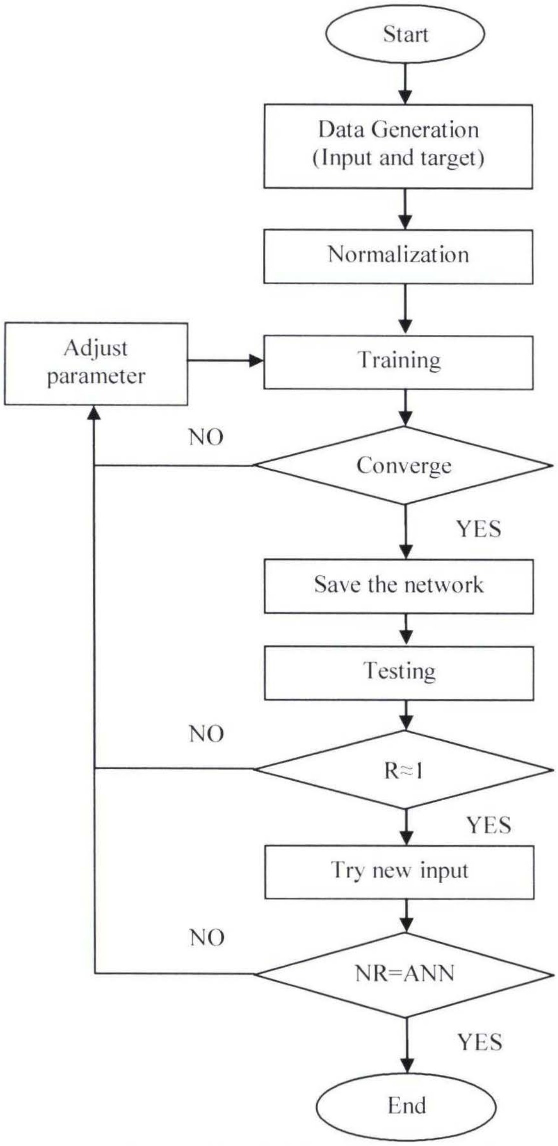

Figure 2: Flow chart for neural network operation.

A. GENERATION OF TRAINING DATA

Once the selection of the input and output parameters is done, the next step is to prepare the training data set. Many system variables change during operation of a power system and these variables include:

- Total generation.
- Capacitor/reactor at different locations.
- Transformer taps.
- Phase shifter angles.
- Topology of the power system
- Total load.

Ideally, variations of all of the above parameters should be considered for generating the training data. However, for the sake of simplicity and to keep the training data set to a minimum length, only the total load of the system was altered during the generation of the training data. For MW and MVAR contingency test, six hundreds input data scenarios were generated. To generate large data and to cover entire power system operating conditions in practical load has been varied with randomly [1].

The input vector for the BPN is the real and reactive powerloads at numerous of power systems. Thus

$P(n)= [P_3\ P_4 \dots P_n\ Q_3\ Q_4 \dots Qn]$

N = is the number of buses

Several sets of loads were created by the following:

a) Varying only the real power loads at a single load bus of the power system.

b) Varying only the reactive power loads at a single load bus of the power system.

c) Varying both the real and reactive power loads simultaneously at a single load bus of the power system.

Then all the input data are inserted to the feedforward backpropagation to be trained [9].

B. NORMALIZATION

Normalization of the data is an important aspect for training of the neural network. Without normalization, there is a risk of the simulated neurons reaching saturated conditions. If the neurons get saturated, then the changes in the input value will produce a very small change or no change in the output value. This affects the learning a great extent. One popular method of normalization is to normalize the mean and standard deviation of the training set and principal component analysis to reduce the dimension of the input vectors. In this proposed method, all training parameters are normalized by this method [1, 6].

C. TRAINING

In the training phase, the network will find the best weight and bias value before implement to the new input. During training, the weights and biases of the network are iteratively adjusted to minimize the network performance function. The default performance function for feedforward networks is mean square error (MSE) which is the average squared error between the network outputs (***a***) and the target outputs (***t***).During the training, output result should be equal with the target to produce the accurate network. If the network achieves the performance target, it can be used to test the new input. This is how the network's performance is assessed. Testing phase can assess network performance based solely on the success of the network in learning an isolated training set. Tests must be done to confirm that the network is also capable of classifying samples outside of the training set [10].

The steps for train the backpropagation depicted at Figure 3.

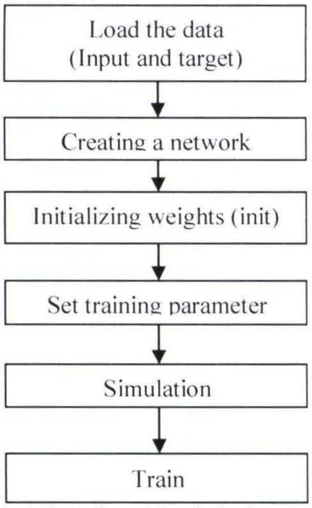

Figure 3: Steps for training the backpropagation

D. TESTING

During the testing phase, the focus is to monitor the value of regression and error network. The regression should be equal to one and low the error if the network properly training. Testing technique is applied to improve ANN generalization by preventing the training from overfitting the problem. In the context of neural networks, overfitting is also known as overtraining where further training will not result in better generalization. The error of the validation set is periodically monitored during the training process [11]. The training error usually decreases as the number of iterations grows, and so does the testing error initially. When the overtraining starts to occur, the testing error typically tends to increase. Therefore, it would be useful and time saving to stop the training after the validation error has increased for some specified number of iterations [11, 12].

The regression test was conducted in order to determine the accuracy of the predicted power stability condition after the testing process. Regression test is a process of measuring how well the trend in the predicted values following the profile of the past actual value. Therefore, it is a measure of how well the predicted values from a forecasted model "fit" with the real-life data [13, 14]. The regression test will produce a value called regression value or also known as the correlation coefficient, R. The correlation coefficient is a number between 0 and 1. If there is no relationship between the predicted values and the actual values, the correlation coefficient is 0 or very low. On the other hand, the correlation coefficient value approaches 1.0 as the strength of the relationship between the predicted values and actual values increases. A perfect fit would give a correlation coefficient of 1.0. Thus, higher the correlation coefficient indicates better prediction [14].

IV. RESULT AND DISCUSSION

The output vector refers to whether the power system would be stable or unstable for the given load. In order to find out whether the power system would be stable or unstable, the power flow equation are solved considering the loads at various buses as specified by the input vector $X(k)$. If the solution of the power flow equations exists implying that the power system is stable or converged, the output vector will be defined as:

$$O(k) = [0]$$

Then if the power flows equations do not have a solution (diverged) considering the loads denoted by the input vector $X(k)$, the unstable output will be defined as:

$$O(k) = [1]$$

Several input and output pairs are generated by the method explained and are stored. Power flow equations are solved for considering each of the generated loads configurations.

In order to test the ability and effectiveness of the proposed ANN in predicting voltage instability in a power system, the standard IEEE 6 bus system is used. The available simulation data were separated into 2 categories, training data and testing data. 400 data were used for the training process, while 200 data were utilized for the testing process.

The training process was carried out many times until it meets a stopping criterion. It was followed by the testing process using the solutions obtained from training programmed.

In this classification method, training process is considered successful when the MSE reaches the value 1e-05. On the other hand, the training process fails to converge when it reaches the maximum training time before reaching the desired MSE. The training time of the algorithm is defined as the number of epochs required to meet the stopping criterion.

TABLE 1: COMPARISON OF BACKPROPAGATION ALGORITHM

Method	Error	MSE	Epoch	R
LM	0.00674	9.41×10^{-6}	26	0.9983
BFG	0.00674	6.41×10^{-6}	72	0.9771
BR	0.0028	2.77×10^{-6}	166	0.9347

From the Table 1, the results obtained for error and mean square error with using different backpropagation training algorithm have only bit different. In the case of 6-IEEE bus, the architecture of backpropagation has the same number of hidden layer, number of neurons in the hidden layer has tested with different algorithm of backpropagation. It is shown that the Levenberg Marquadt (LM), BFGS Quasi-Newton algorithm (BFG) and Bayesian Regularization (BR) are between the methods being able to assess the state of security power system. The Levernberg Marquadt algorithm has shown the better result compare the other algorithms because it has a lowest epoch for training and quite lower for the error and mean square error. Meanwhile the Bayesian rule has an excellent result in error and mean square error but it lack of the epoch performance because their epoch too high compare with leverberg marquadt. Conversely, error and mean square error for the BFGS Quasi-Newton algorithm present the result almost similar with Levenberg Marquadt algorithm (LM) except for their epoch has higher before meeting the stopping criterion. In the dimension of regreesion, it clearly shows the Lavenberg Marquadt also has a highest value of regression. This means Levenberq Marquadt is better to avoid the overfitting phenomenon.

TABLE 2: RESULT OF OUTPUT WITH VARYING REAL POWER (P)

P(Watt)	*T(NR)*	O_{LM}	O_{BFG}	O_{BR}
$P_3 = 170$	1	1	1	1
$P_4 = 80$	0	4.54e-17	7.05e-8	7.81e-7
$P_5 = 60$	1	1	1	1
$P_6 = 130$	1	1	1	1

TABLE 3: RESULT OF OUTPUT WITH VARYING REACTIVE POWER (Q)

Q(VAR)	T(NR)	O_{LM}	O_{BFG}	O_{BR}
$Q_3 = 65$	0	4.57e-17	7.05e-8	4.52e-10
$Q_4 = 88$	1	1	1	1
$Q_5 = 30$	1	1	1	1
$Q_6 = 44$	0	4.57e-17	7.05e-8	5.69e-8

TABLE 4: RESULT OF OUTPUT WITH VARIED REAL AND REACTIVE POWER (P & Q)

P(Watt)/Q(VAR)	T(NR)	O_{LM}	O_{BFG}	O_{BR}
P3 = 120 $Q_3 = 43$	1	1	1	1
P4 = 120 $Q_4 = 30$	1	1	1	1
P5 = 40 $Q_5 = 20$	1	1	1	1
P6 = 80 $Q_6 = 10$	0	4e-17	7e-8	1e-8

Table 2 and 3 shows the result of classification of power system stability when real power and reactive power at single load bus was varied randomly. The neural network using multi layer perceptron will indicate 0 for converge and 1 as diverge. Then, the results were compared with analysis from Newton Raphson method using load flow which target was obtained from this method. In this result, when new inputs apply to the neural network it produces accurately one when it diverges and approximately equal to zero when it converges. The entire training algorithm gives the right output and the best performance algorithm is Levenberg Marquadt. The Levenberg Marquat algorithm yields the output result very close with target when the power system is converge compared to the other algorithms. Meanwhile, from the Table 4.it is observed when the both of real power and reactive power change simultaneously, the entire algorithm also produce the output accurately with the target.The Levenberg Marquadt again present the result closely with target obtain from Newton Raphson method.

V. CONCLUSION

This paper has proposed a neural network technique for predicting the power stability condition of a power system. The developed system was tested using the IEEE 6 bus reliability test system and the results shows there is a good concurrence between the desired output and the predicted output. In this study, the proposed technique has able to predict the power stability condition of a power system and therefore could be a valuable tool for fast real-time steady state stability assessment. ANN can respond very fast compared to the traditional analytical methods, which means that it is more suitable for online application. Though the chosen neural network architecture seems to work well in this paper, this not means it is optimal. Indeed, the process of choosing the number of hidden layers, the number of neurons for each layer, how to deal with overfitting during the training procedure, remains too subjective to determine.

VI. FUTURE DEVELOPMENT

For the future, this study could be extended with using a large power system and with more varies input data such as total generation, phase shift angles and transformer taps.

ACKNOWLEDGMENT

The author would like to thank Dr. Zuhaina and my friends for their valuable comments and editing to improve the quality of this paper.

REFERENCES

[1] K. Shanti Swarup, G. Sudhakar - Neural network approach to contingency screening and ranking in power system.

[2] S. *Kajan* - Gui for classification using multilayer perceptron network.

[3] L. Srivastava, S.N. Singh, J. Sharma - A hybrid neural network model for fast voltage contingency screening and ranking, Electric. Power Energy (2000) 35–42.

[4] X. Luo, A. D. Patton, C. Singh - Real Power Transfer Capability Calculations Using Multi-Layer Feed Forward Neural Networks.

[5] R. Rojas: Neural Networks, Springer-Verlag, Berlin, 1996 The Backpropagation Algorithm.pp 151-152.

[6] Howard Demuth, Mark Beale - Neural Network Toolbox for use with MATLAB.Backpropagation.pp. 5-9.

[7] 1. J.H. Park, Y.S. Kim, I.K. Eom and K.Y.a Lee, "Economic load dispatch for piecewise quadratic cost function using hopfield neural network". IEEETrans. on Power Systems, Vol. 8, No. 3, Aug. 1992, 1030-1038.

[8] J. Nanda, A. Sachan, L. Pradhan , M.L. Kothari, Koteswara Rao , L.L. Lai , Mata Prasad – Application of Artificial Neural Network to Economic Load Dispatch.

[9] Mohd Imran Bin Shamsudin - Artificial Neural Network Approach in Determining Steady State Stability Classification.

[10] Soo-See Chai1, Bert Veenendaal1, Geoff West2, Jeffrey PhilipWalker3-Backpropagation neural network for soil moisture retrieval using nafe'05 data: a comparison of different training algorithms.

[11] Abaza, Mahmoud M. , student, IEEE and Starrett, Shelli K. - Real Time Voltage Collapse Prediction Using Artificial Neural Network.

[12] The Mathworks Inc., 2007 Neural Network Toolbox™User's Guide for using in MALAB.

[13] Financial Forecast Center, "What Is The Correlation Coefficient", http://www.forecasts.org/cc.htm.

[14] S. I. Suliman, T. K. Abdul Rahman, I. Musirin – Artificial Immune-Based For Voltage Stability Prediction in Power System.