

MILITARY TECHNICAL COLLEGE
CAIRO-EGYPT



FIRST INTERNATIONAL CONF. ON
ELECTRICAL ENGINEERING

TRANSIENT STABILITY EVALUATION USING A CLASSIFIER NEURAL NETWORK

A.Y. Abdelaziz¹

ABSTRACT

The transient stability evaluation studies were mainly related to the following factors: 1) the construction parameters of the network and operating conditions of the power system. 2) the fault type and location. 3) the fault clearing time.

Application of artificial neural networks to power systems has resulted in an overall improvement of solutions in many areas. This paper presents a multi-layer feedforward perceptron type neural network (NN) for transient stability evaluation. Numerical integration and Lyapunov direct method are used to obtain the training set of the NN. The neural network architecture adopted as well as the selection of input features for training the neural networks is described. The approach is tested on a 3-generator power system using a stochastic backpropagation algorithm.

The capabilities of the developed technique is tested through computer simulation for a typical case study. By comparison with the conventional methods, the proposed method can quickly estimate the transient stability of the power system at different fault locations, clearing times and load levels of the system. The results demonstrate that this approach has an excellent classification performance and simple for on-line application.

KEYWORDS: Neural Networks, Power Systems, Transient Stability, Stochastic Backpropagation.

¹ Assistant Professor, Department of Electrical Power & Machines, Faculty of Engineering, Ain Shams University, Cairo, Egypt

NOMENCLATURE

- E error of a neuron in a neural network
 E.D total energy deviation w.r.t post fault equilibrium point, at the instant of fault clearing
 $f(I)$ activation function
 I_i net weighted input received by neuron i from a total of n neurons in the network
 K.E generator kinetic energy deviation at the instant of fault clearing
 P_m generator mechanical input power
 T threshold value
 w_{ij} weight on the connection directed from neuron j to neuron i .
 x_j incoming signal from the j th neuron
 y_i output of a neural network
 β learning constant of a neural network
 Δw_{ij} change in the weight of a neural network
 γ momentum constant of a neural network

1. INTRODUCTION

Transient stability evaluation of a power system pursues a two fold objective : first to appraise the system's capability to withstand major contingencies, and second to suggest remedial actions whenever needed. The first objective is the concern of analysis, the second is a matter of control.

To assess transient stability, multitude of techniques [1-9] and methods are available encompassing traditional time domain state numerical integration, Lyapunov based techniques, probabilistic methods, pattern recognition and recently neural networks.

The approach presented in this paper is based on backpropagation trained neural networks. The major benefit of neural networks is that, once trained, it can quickly classify a new pattern as belonging to a known predefined class of patterns. Using neural networks, the transient stability evaluation is a classification problem with two classes. Compared with recent approaches, the presented method is very simple for on-line assessment and has an excellent classification performance.

2. ARTIFICIAL NEURAL NETWORKS

A neural network consists of a number of very simple and highly interconnected processors called neurons. The neurons are connected by a large number of weighted links, over which signals can pass.

A neuron in a neural network receives input stimuli along its input connections and translates those stimuli into an output response, which is transmitted along the neuron's output connection. Firstly, the neuron computes the net weighted input (I_i) it is receiving along its input connections as:

$$I_i = \sum_{j=1}^n w_{ij} x_j \quad (1)$$

where, I_i is the net weighted input received by neuron i from a total of n neurons in the network, x_j is the incoming signal from the j th neuron and w_{ij} is the weight on the connection directed from neuron j to neuron i .

The second step is converting the net input to an activation level for the neuron. The activation is expressed by a sigmoid function as :

$$f(I) = \frac{1}{1 + e^{-I}} \quad (2)$$

This function has the useful property that its derivative is exceptionally easy to compute:

$$df(I)/dI = f(I)(1-f(I)) \quad (3)$$

The final step is to convert the neuron's activation level to an output signal as:

$$y_i = \{ f(I), \text{ if } f(I) > T \}$$

$$y_i = \{ 0, \text{ otherwise } \} \quad (4)$$

where T is the threshold value.

A well known supervised training algorithm is the backpropagation algorithm, which is used in this paper. The backpropagation algorithm is basically a generalization of the Least Mean Square (LMS) rule or the generalized delta rule algorithm described in reference [10].

A backpropagation network operates in a two-step sequence during training. First, an input pattern is presented to the network's input layer. The resulting activity flows through the network from layer to layer until the network's response is generated at the output layer. In the second step, the network's output is compared to the desired output for that particular input pattern. If it is not correct, an error is generated, which is passed or propagated backward through the network - from the output layer back to the input layer, with the weights on the intra-layer connections being modified as the error backpropagates. The generalized delta rule specifies the change in a given connection as:

$$\Delta w_{ij} = \beta \cdot E \cdot f'(I) \quad (5)$$

E is the error for this neuron, β is the learning constant, a parameter between 0.0 and 1.0, and $f'(I)$ is the input to the neuron. For reasons of mathematical stability, the net weighted error in each middle-layer neuron is multiplied by the derivative of the activation function of the middle-layer neuron. The final form of the error computation is :

$$E_j^{output} = y_j^{desired} - y_j^{actual} \quad (6)$$

$$E_i^{middle} = \frac{df(I_i^{middle})}{dI} \sum_{j=1}^n w_{ij} \cdot E_j^{output} \quad (7)$$

In these expressions, the superscripts "output" and "middle" designate the layer of the neuron that is modifying the weights on its incoming connections. Neuron j is an arbitrary neuron in the output layer; neuron i is an arbitrary neuron in the middle layer.

Adding a momentum term to the generalized delta rule is an attempt to avoid the lengthy training time of the neural network. The change is redefined as:

$$\Delta w_{ij} = \beta \cdot E \cdot x_i + \gamma \cdot \Delta w_{ij}^{previous} \tag{8}$$

The parameter γ is the momentum constant and has a value between 0.0 and 1.0.

3. APPLICATION OF NEURAL NETWORKS TO TRANSIENT STABILITY

In this section, the method and the procedures for application of neural networks to transient stability evaluation are described. The example system under study is a nine bus power system that has three generators and three loads. A one line diagram for the system is shown in Fig. 1, and the system characteristics are given in [7].

To develop a neural network for a classification problem, it is necessary to:

- (i) choose a network architecture;
- (ii) choose a training algorithm;
- (iii) apply the algorithm to a set of training data until parameter convergence is achieved;
- (iv) supply the trained network with on-line data to perform the classification required.

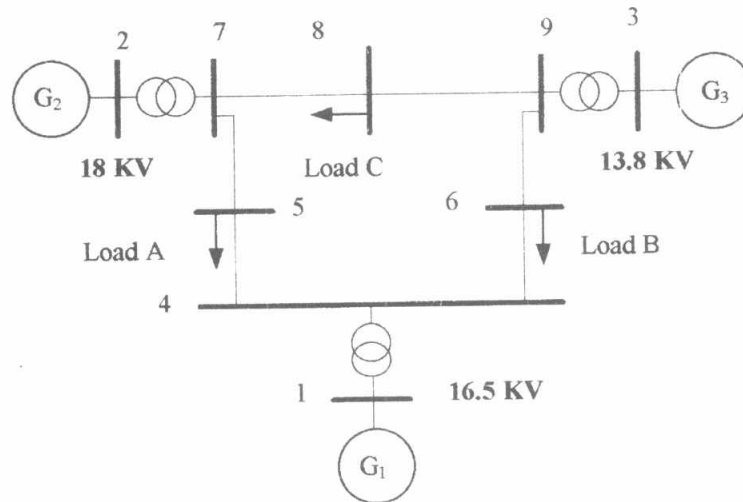


Fig. 1. Power system under study

3.1 Generation of Samples

It is assumed that the loads are randomly distributed and that they have a normal distribution shape with the following means:

$$[P_A, P_B, P_C] = [1.25, 0.9, 1.0] \text{ p.u.}$$

For load flow analysis, Bus 1 is taken as the swing bus and Buses 2 and 3 are voltage controlled buses with voltage magnitude of 1.025 p.u. For each load sample, the loading of the generators is determined by economical dispatch of the total load among generators, followed

by a load flow analysis. A three phase short circuit is assumed to occur at one line very close to one of the buses of the system and the fault is removed by tripping out the faulted line. The Runge-Kutta numerical integration approach is applied to find the class of each sample. A sample is classified as unstable if the rotor angle of any of the generators reaches 180 degrees within one second [7], and it is given a stability index of 0, otherwise the sample is classified as stable, and it is given a stability index of 1.

Generation of samples is performed by changing the fault location, loading conditions of the system prior to the occurrence of the fault and the clearing time. A group of samples is generated at six different fault locations with two different load levels (1.5 , 0.5) p.u. for each of the three loads of the power network under study and three different clearing times (100, 150, 200) ms. resulting in 144 samples.

In order to improve the classification performance, a normalization process is performed to all the variables of the training set (and the test set).

3.2 Selection of Features

There are three important quantities which have significant indication of stability and as a result their pattern would dominate the pattern of stable and unstable classes,

- 1) The loading of generators (P_m) , the higher the loading of the generators, the higher the risk of instability.
- 2) The kinetic energy deviation of each generator ($K.E = 0.5 M \omega^2$) at the time that the fault is cleared. These variables carry some information about the energy distance of generators and present an indication of the path of rotor angles.
- 3) The total energy deviation (E.D) with respect to the post fault equilibrium point, which is directly related to stability.

Taking into consideration that the loads are economically dispatched among the generators, then P_{m1} , P_{m2} , ... , P_{mn} would have the same information as $\sum P_m$ because all P_{mi} are linearly dependent, therefore the chosen features are:

$$\sum_{i=1}^n P_{mi}, 0.5 M \omega_1^2, 0.5 M \omega_2^2, 0.5 M \omega_3^2 \text{ and E.D}$$

" P_m " is a direct outcome of the load flow results. The other features are determined from a transient stability program using the second-order model of the machine. Full details of the derivation of the energy deviation using Lyapunov direct method is mentioned in [5].

3.3 Description of the Neural Networks Used

Two neural networks are used in this paper. The first NN contains three layers (1 input layer, 1 hidden layer and 1 output layer) and the second NN contains four layers (1 input layer, 2 hidden layers and 1 output layer). A schematic diagram for the two neural networks is shown in Fig.2.

1. The input layer contains five neurons loaded with the five input features.

2. The middle (hidden) layer : the number of neurons in this layer is chosen to be sufficient to represent the complex relationships required. Generally, the more complex the relationship, the greater the number of training data sets and the hidden layer neurons.
3. The output layer contains only one neuron.

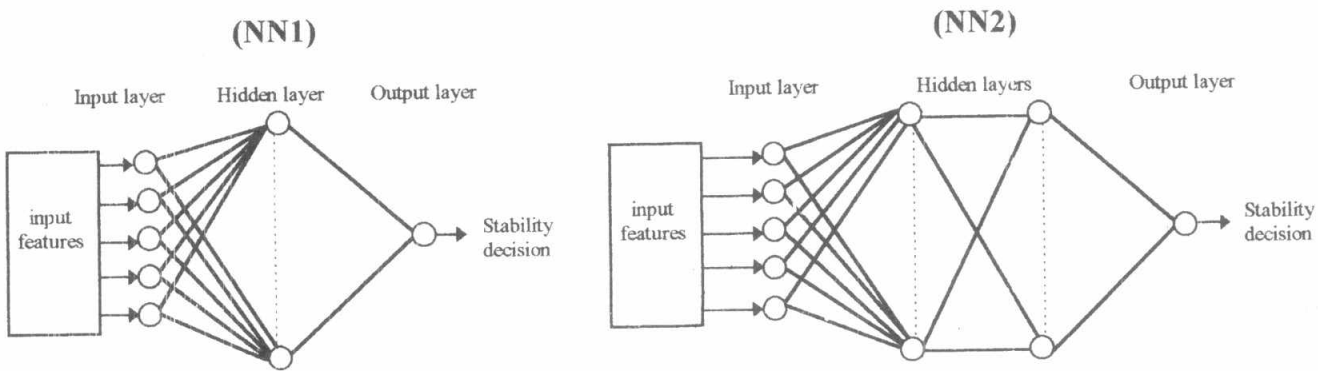


Fig. 2

It is supposed to have an output value from the neural network between 0 and 1. Normally, the sample can be classified as unstable if its output value from the neural network is less than 0.5 and stable if the value is greater than 0.5.

3.4 Validity of Classification Using the Backpropagation Algorithm

It is allowed to train the neural networks until the least square error is reduced to $1 \cdot 10^{-3}$, then the weights of the neural networks are determined, and the transient stability is then predicted using the trained neural networks.

There is an important factor which measures the degree of success of the classification process, which is, the number of misclassified samples among the total number of samples in the training set. A misclassified sample is a stable sample classified as unstable, or an unstable sample classified as stable.

Using the standard backpropagation algorithm in classification, with a learning constant $\beta = 0.2$ and momentum constant $\gamma = 0.9$, (the most commonly used values), some unacceptable results are obtained. It is noticed that the output value from the neural network for all the samples of the training set ranged from 0.9 to 1. This might occur due to the sequence of the samples. It is concluded that care must be taken with the order in which the patterns are presented. For example, when using the same sequence repeatedly the network may become focused on the first few patterns. This problem can be overcome by using a permuted training method. The program is therefore modified in order to randomize the order of the samples in the training process. This operation is called the stochastic training approach [11].

Table 1. shows the effect of varying the number of the neurons in the hidden layers on the performance of the classification process using the stochastic backpropagation algorithm. In the table, (u) is an unstable sample misclassified as stable and (s) is a stable sample misclassified as unstable.

Table 1. Effect of varying the number of neurons in the hidden layers on the classification performance of the training set using two different neural network constructions

Neural Network Chosen	Number of neurons in the hidden layer		Number of misclassified samples	
<u>N.N.1</u>	15		1(u)	2(s)
	20		----	1(s)
	25		1(u)	2(s)
	30		1(u)	1(s)
<u>N.N.2</u>	<u>Layer 1</u>	<u>Layer 2</u>		
	5	5	1(u)	2(s)
	5	10	1(u)	1(s)
	10	5	1(u)	1(s)
	10	10	----	1(s)

3.5 Performance Testing

The last step in the neural network approach is the *generalization process* by which a complete verification of the capabilities of the neural network in predicting the class of unknown samples is performed. The generalization ability is best stated in probabilistic terms as the probability of correct classification. It can serve as an index of satisfactory performance of the classification in unknown situations. This step is conducted by testing the chosen neural network using an adequate test set. The samples of the test set should cover a wide spectrum of operating conditions and contingencies that the machine under study may be subjected to. The generation of samples for the test set is performed in a similar way to the training set. The test set is generated at six fault locations with three different load levels (1.6 , 1.0 , 0.4) p.u. for each of the three loads and two different clearing times (125 , 175) ms. This produces 324 samples.

The prediction of the stability of each sample is obtained by running the chosen neural network for each sample of the test set and obtaining its output value. If the output value of the sample is less than 0.5 (as a threshold), it will be considered as unstable, otherwise, it will be considered as stable. Table 2. shows the results of the correct classification percentages for some selected neural networks used. The correct classification percentage is defined as:

$$[1 - (\text{total number of misclassified samples} / \text{total number of samples in the test set})] * 100.$$

Table 2. Results of testing selected neural networks using the test set

Neural Network Chosen	Number of neurons in the hidden layer		Number of misclassified samples	correct classification %
<u>N.N.1</u>	20		3	99.1
	30		4	98.8
<u>N.N.2</u>	<u>Layer 1</u>	<u>Layer 2</u>		
	5	10	4	98.8
	10	5	4	98.8
	10	10	3	99.1

4. CONCLUSIONS

- The successful identification of stable and unstable classes is significantly dependent on the chosen primary variables. For example, voltage magnitudes usually vary in a narrow band and near rated values and are unlikely to be sufficiently discriminatory. Also, voltage angles are not normally available in actual systems. So, the primary variables should assess the properties of each class and should fulfill discrimination, reliability and independence.
- The significant reduction in misclassification error demonstrates that the implicit assumption of other researchers regarding the sufficiency of steady state variables for presentation of stable and unstable classes is unjustified. This means that selection of only steady state variables as input features would result in overlapping of stable and unstable classes and consequently some level of error. This error cannot be reduced with any refinement in the process of feature extraction.
- The introduction of transient variables (K.E and E.D) as input features guarantees a low error which is acceptable from a practical point of view [12]. Since transient variables are closely related to the phenomenon of stability, it is expected that the misclassification error would not change significantly with the size of the system.
- The number of inputs and outputs of the neurons of the neural network are related to the training and test (target) set. The optimum number of hidden neurons depends on the type of problem and it directly affects the prediction accuracy. In general, the number should be chosen to be just sufficient to represent the complex relationship required.
- The stochastic backpropagation algorithm gives acceptable results especially with an appropriate number of neurons in the hidden layer. Although an increased number of middle layer neurons performs well as a result of the higher dimensionality of the error space, it appears that there is some upper limit on the number of hidden neurons which, when exceeded, results in the system becoming trapped in a local minima.

5. REFERENCES

- [1] El-Abiad, A.H. and Nagappan, K., "Transient Stability Regions in Multimachine Power Systems", IEEE Trans. on PAS, Vol. 85, pp. 169-179, Feb. (1966).
- [2] Fouad, A.A. and Stanton, S. E., "Transient Stability of a Multimachine Evaluation Power System", Part I, II, IEEE Trans. on PAS, Vol. 100, No. 7, pp. 3408-3424, July (1981).
- [3] Ribbens-Pavella, M., Cutsem, T.V., Dhifau, R. and Toumi, B. "Energy-Type Lyapunov-Like Direct Criteria for Rapid Transient Stability Analysis", Proc. of the Int. Symp. on power system stability, Iowa, May (1985).
- [4] Billinton, R. and Kuruganty, P.R.S. "Probabilistic Assessment of Transient Stability in Practical Multimachine Power System", IEEE Trans. on PAS, Vol. 100, (1981).
- [5] Hakimashhadi, H., "Fast Transient Security Assessment", Ph. D. thesis, Purdue University, Indiana, USA, August (1982).
- [6] Chang, C.S., "Online Transient Stability Evaluation of Interconnected Power Systems Using Pattern Recognition Strategy", IEE proceedings-C, Vol. 140, No. 2, pp. 115-122, Aug. (1993).

- [7] El-Arabaty, A.M., Talaat, H.A., Mansour, M.M. and Abdelaziz, A.Y., "Out-of-step Detection Based on Pattern Recognition", International Journal of Electrical Power & Energy Systems, Vol. 16, No. 4, pp. 269-275, Aug. (1994).
- [8] Abdelaziz, A.Y., Irving, M.R., El-Arabaty, A.M. and Mansour, M.M. "Out-of-step Prediction Based on Artificial Neural Networks", Electrical Power Systems Research, Vol. 34, No. 2, pp. 135-142, Aug. (1995).
- [9] Hobson, E. and Allen, G.N., "Effectiveness of Artificial Neural Networks for First Swing Stability Determination of Practical Systems", IEEE Trans. on Power Systems, Vol. 9, No. 2, pp. 1062-1068, May (1994).
- [10] Caudill, M. and Butler, C., "Understanding Neural Networks", Vol. I: Basic Networks, The MIT Press, London, England, (1992).
- [11] Rumelhart, D.E. and McClelland, J.L., "Parallel Distributed Processing", Explorations in the Microstructure of Cognition, Vol. 1: Foundation, The MIT Press, London, England, (1986).
- [12] Venikov, V., "Transient Processes in Electrical Power Systems", Mir Publishers, Moscow, (1977).

