SECOND SYMPOSIUM ON EXPERT SYSTEMS APPLICATION
TO POWER SYSTEMS, JULY 17-20, 1989, SEATTLE, USA

Dynamic Security Assessment of Power Systems Using Back Error Propagation Artificial Neural Networks

M. A. El-Sharkawi, R. J. Marks, M. E. Aggoune, D. C. Park, M. J. Damborg, L. E. Atlas

Department of Electrical Engineering
University of Washington
Seattle, WA 98195

ABSTRACT

In this paper we are proposing the use of Artificial Neural Networks (ANN's) as an operator aid in the dynamic security assessment of power systems. The basic role of ANN's in this study is to provide assessment of the system's security based on training examples from off-line analysis.

The ANN's in this study are trained either by 1) randomly distributed data which facilitate the utilization of both real measurements and off-line simulations; or by 2) data obtained by the interval halving method for more accurate results around the boundaries. The training algorithm is based on the Back-Error Propagation method which provides a high degree of accuracy.

Keywords: Power System Security, Artificial Neural Networks.

INTRODUCTION

The security assessment problem results from the continually changing topologies of power systems [1]. In this paper, artificial neural networks (ANN's) are proposed for on-line security assessment as an operator aid. Specifically, here we are concerned with security relative to dynamic stability. The basic concept is to use off-line data to explore the region of system security in a space of critical operating variables. These variables then serve as inputs to an ANN which is trained with this off-line data to yield the proper response; "secure" or "insecure". The trained ANN could then be used on-line, i.e. it could be fed with the on-line values of the input variables and yield a warning to the system operator if the system is in the insecure region. An important feature of ANN's that is fundamental to this approach is that they can interpolate among the training cases to give an appropriate response for cases described by neighboring inputs [2,3].

In dynamic stability analysis, it is appropriate to examine the eigenvalues of a linearized version of the system model about the assumed operating point. Hence, to examine many operating points, the nonlinear model must be linearized and analyzed for each point. A model representing the complete system that concerns a typical operator is much too large for an on-line linearization and eigenvalue analysis, even when dynamic equivalents are used [4-6]. Hence, on-line aid must probably be built using off-line analysis.

In addition to the above mentioned difficulties, the on-line dynamic security assessment is hard to achieve for a number of reasons such as: (1) To be complete, the number of cases which must be examined is very large; (2) The system is never actually operating at the states that were examined so the operator must interpolate among cases; (3) The operator must have a way of cataloging and retrieving the appropriate cases for the current system state, and this must be done quickly!

ANN's, when adequately trained, can alleviate the above mentioned difficulties. They can provide on-line assessment to the system security faster than any known technique. Previous work in steady state security [3] and dynamic security [2] show the potentials of ANN's as an operator aid.

In this paper, the ANN approach for dynamic security problem is further investigated. More specifically, the following areas are presented:

1. Back-Error Propagation algorithm is used in ANN training. This algorithm provides a high degree of accuracy during testing.

2. The ANN's are trained either by a) randomly distributed data to facilitate the mix of real system measurements with off-line simulations; or by b) data generated by an interval halving method to enhance the accuracy of the ANN at the boundaries of the security regions.

ARTIFICIAL NEURAL NETWORK CLASSIFIERS

Artificial Neural Networks (ANN's) loosely resemble the architecture and algorithmic performance of their biological counterparts. Generally, an ANN can be defined as a highly connected array of elementary processors called neurons. A popular model for classification ANN's is the layered one shown in Figure 1 [7-13]. The top layer receives the input vector that stimulates the network. Each element of this vector is weighted by the input to hidden interconnects to form at the middle (or hidden) layer a weighted sum.

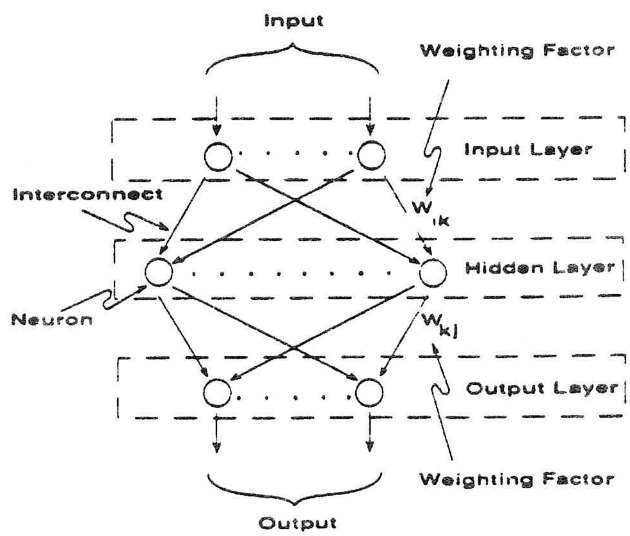

Figure 1. Structure of a Three-Layered ANN

This sum is altered by a nonlinearity (e.g. sigmoid) to establish the state of each hidden neuron. The states of hidden neurons propagate to the output layer to decide the states of the output neurons just as

M.A. El-Sharkawi, R.J. Marks II, M.E. Aggoune, D.C. Park, M.J. Damborg and L.E. Atlas,
"Dynamic security assessment of power systems using back error propagation artificial neural networks",
Proceedings of the 2nd Annual Symposium on Expert Systems Applications to Power Systems, pp.366-370, 17-20 July 89, Seattle.

the input neurons do to the hidden neurons. Layered ANN's can be trained by iteratively inputting training data [7,8,14,15]. The performance of the referenced iterative techniques is dictated by the structure of the classification partition boundaries: the more complicated the boundaries, the more hidden neurons are required. In some instances, a second hidden layer is needed. A highly regarded tutorial on other aspects of classification ANN's is given in [15]

BACK ERRPR PROPAGATION ALGORITHM

Back error propagation learning requires a set of input and output (target) pairs. Basically, an output vector can be produced by presenting an input pattern to the network. According to the difference between the produced output and the target vector, the network's weights are adjusted to reduce the output error.

The ANN used in this paper consists of input, hidden, and output layers. Four neurons, including a bias neuron, are assigned to the input layer. Ten neurons are used in the hidden layer, and one neuron in the output layer. For convenience, we number the bias neuron with 0, the input neurons with 1 to 3, hidden neurons with 4 to 13, and the output neuron with 14.

Define,

w_{ij}	:	the weight between neuron i and j
o_i	:	the output state of neuron i
t	:	the target state of output neuron

Unless the neuron k is one of the input neurons, the state of the neuron k is:

$$o_k = f(\Sigma_i w_{ik} o_i)$$

where $f(x) = 1 / (1 + e^{-x})$, and Σ is over all the neurons in the adjacent layer.

The error at the output neuron can be defined as [8]

$$E = (1/2)(t - o_k)^2$$
$$= (1/2)(t - f(\Sigma_{i=4,13} w_{ij} o_i + w_{0k}))^2$$

here k = 14 since the output neuron is number 14. w_{0k} is the bias of the k th output neuron.

The gradient descent algorithm adapts the weights according to the gradient of error, i.e,

$$\Delta w_{ij} \quad \propto \quad -(\partial E/\partial w_{ij}) = -(\partial E/\partial o_j)(\partial o_j/\partial w_{ij})$$

Specifically, we define the error signal as

$$\delta_j = -\partial E/\partial o_j$$

With some manipulation, we can get the following back error propagation adaptation rule:

$$\Delta w_{ij} = \epsilon \delta_j o_i$$

where ϵ is an adaptation gain,

$$\delta_j = (t - o_j) o_j (1 - o_j), \qquad \text{for } j = 14$$
$$= o_j (1 - o_j) \Sigma_k \delta_k w_{jk}, \qquad \text{for } j \neq 14$$

In order to increase the speed of convergence, we utilize a momentum term α which augments the learning rule to

$$\Delta w_{ij}(n+1) = \epsilon \delta_j o_i + \alpha \Delta w_{ij}(n)$$

where n denotes the iteration index. The momentum gain affects the past weight changes on the current direction of movement in the weight space.

DATA SELECTION BY INTERVAL HALVING

The ANN's in this study are trained either by randomly chosen data or by data generated by interval halving. The internal halving method results in training data closer to the boundary of the security region.

Assume that a set of 1000 randomly chose data is available for training the ANN. With the interval halving method, we can generate another set of 1000 data points that are closer to the boundary of the security region.

The interval halving procedure can be explained as follows: Assume we have a secure point A and an insecure point B in the classification space. Let C be the mid point between A and B. The system equations are queried to determined the security status at point C. If C is insecure, then we can delete point B from our data base. This is because points B and C have the same security status, and point C is assumed to be closer to the boundary than point B. (This is always true if the security region is convex). A second iteration of the interval halving procedure will find a mid point D between A and C. If the system equations indicate that point D is insecure, then A can be deleted, and so on. In our study, we used only two such interval halving iterations.

It is worth mentioning that the interval halving method requires the system model. Whenever uncertain arises during the interval halving procedure, the system equations are queried.

TEST SYSTEM AND ALGORITHM

The test system used in this study is shown in Fig.2. It is composed of 9 buses, 11 transmission lines, and 3 generators. The complete data is given in [2]. The multimachine state space model for stability analysis used in this study is derived in [5,6].

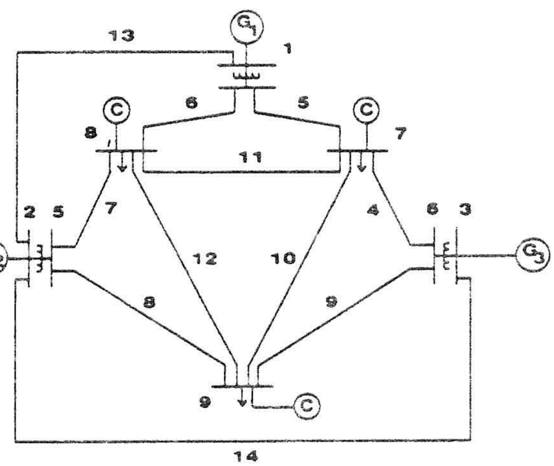

Figure 2. Test System

The ANN technique for power system dynamic security assessment developed in this paper involves the following general steps [2]:

Step 1: Identify a List of Contingencies: Specify a set of contingencies which have impact on the power system stability; such as the loss of a transmission line, a generating unit, a load, or a combination of these.

Step 2: Generate off-line Training Data: For each of the contingencies listed in step 1, the stability of the power system is checked and the training data is generated. In this study the training data set is one of two types a) randomly selected; or b) interval halving data.

Step 3: Train the ANN: The network is trained with one of the data sets generated in step 2.

Step 4: Test the ANN: Once the training is complete, the ANN's classification ability is checked using operating points which are not necessarily part of the training set.

TEST RESULTS

Several contingencies were evaluated during the course of this study. However, for brevity, only one case is reported here. The security problem reported here is related to the real and reactive power generated by machine number 3 (P3 and Q3), in addition to the apparent power of machine number 2 (S2). The security region in this case is three dimensional with the above mentioned variables as the axes.

The ANN is composed of three input neurons (and for each of P3, Q3 and S2) and one output neuron. It also contains 10 hidden neurons.

The ANN is trained either by using the randomly distributed data or by the data generated by interval halving. The back error propagation method is used in both cases to train the network. Samples of the test results are given in Figure 3. Figure 3a represents the test results when the ANN is trained by the randomly chosen data, and Figure 3b is for the case of interval halving. In both cases, the security contour is generated for S2 = 0.8 pu. In Figure 3a, the dashed line shows the security contour generated by the ANN during the test. The solid line, which is added for comparison, represents the "actual" security contour. This "actual" contour is obtained by simulating the power system model.

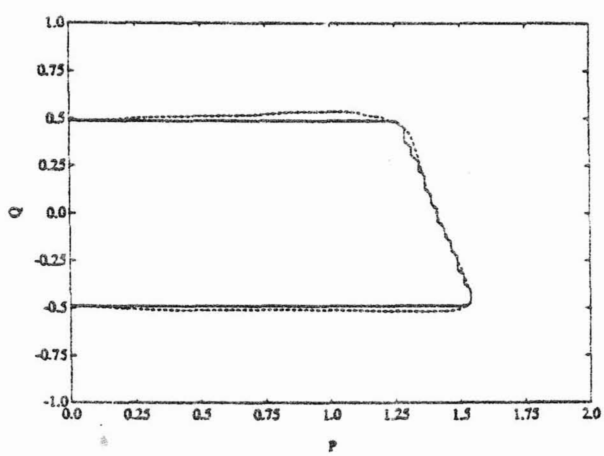

Figure 3a. Security Contour Obtained Using Random Data for S2=0.8 p.u.

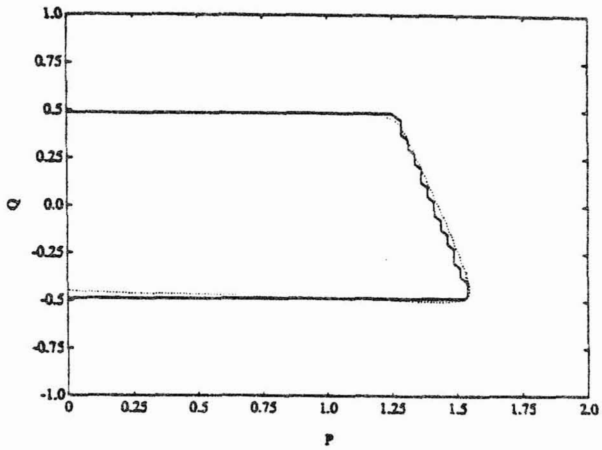

Figure 3b. Security Contour Obtained Using Interval Halving Data for S2=0.8 p.u.

Figure 3b shows the case when the ANN is trained by the interval halving method. The dotted line is for the security contour obtained by the ANN, and the solid line is for the "actual" contour.

From the above study, it is shown that both methods provide good classification of the security regime. Moreover, Figure 3b shows fewer misclassifications as compared to Figure 3a.

Another test case is shown in Figure 4. In this case the reactive power Q3 is kept constant at 0.3 pu. The three curves show how the ANN interpolates among the training data points. The dashed line shows the case when the ANN is trained by the random data, the dotted line shows the case of the interval halving, and the solid line shows the "actual" boundary. The figure, again, shows that the ANN is capable of providing adequate interpolation in both cases. The case of interval halving shows a better interpolation as compared to the case of random data.

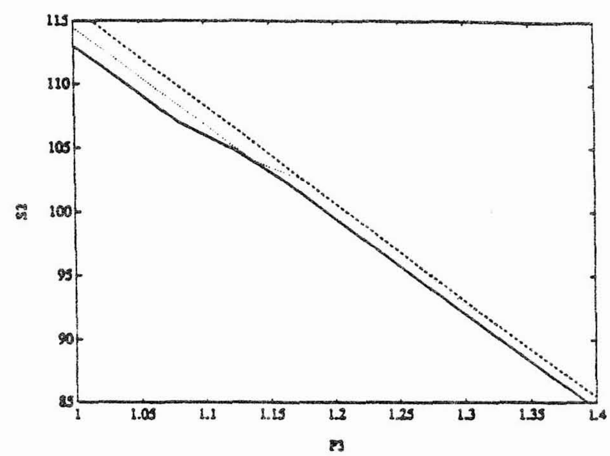

Figure 4. Network's Generalization

Figure 5 shows the "learning error" versus the number of iterations. The learning error is defined here as the cumulative error after each iteration of the back error propagation method. It is seen in this curve that the learning error initially decreases rapidly as the number of iterations increases. After about 600 iterations it starts to level off.

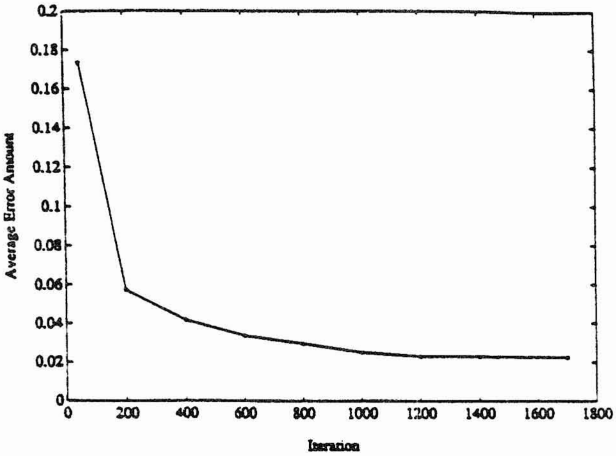

Figure 5. Network's Learning Curve

Figure 6 shows the "testing error" versus the number of iteration in the learning stage (the number of iteration used to train the ANN). The testing error in percent corresponds to the number of misclassifications with respect to the total number of tested data points. The figure shows two curves: the solid line is for the ANN trained by using the interval halving data and the dashed line is for the ANN trained by the random data. The figure shows a very small amount of misclassification in both cases. Again, the interval halving method provides a lower error rate that is leveled off for any ANN trained by more than about 400 iterations.

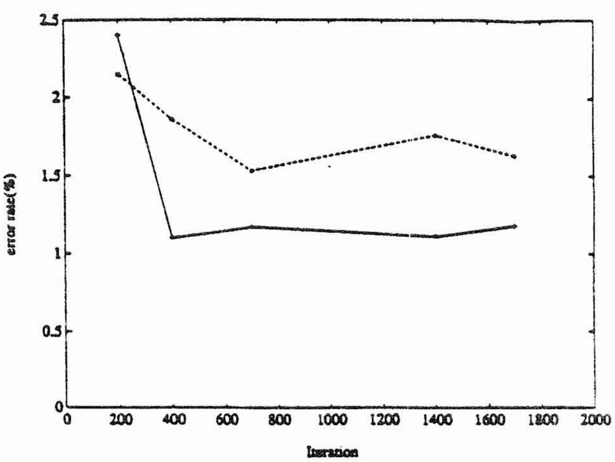

Figure 6. Network's Performance Curves

CONCLUSIONS

We have demonstrated that the Artificial Neural Network is able to properly interpolate among training data sets to recognize security contours. The importance of this result is that, once trained, the network represents the complex mathematical relationships of the power system which otherwise must be explicitly simulated. The ANN is trained by either randomly selected data or by data generated by the interval halving method. The training algorithm is based on the back error propagation technique.

All the results show a very low misclassification rate.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation and the Washington Technology Center at the University of Washington.

REFERENCES

[1] A. A. Fouad and Working group, "Dynamic security assessment practices in North America," IEEE trans. on PAS., Vol.3, pp.1310-1321, Aug.1988.

[2] M. E. Aggoune, M. A. El-Sharkawi, D. C. Park, M. J. Damborg, and R .J .Marks II, "Preliminary Results on Using Artificial Neural Networks for Security Assessment," 16th Power Industry Computer Application Conference, Seattle, WA, PICA'89, May 1989.

[3] M. E. Aggoune, L. E. Atlas, D. A. Cohn, M. J. Damborg, M. A. El-Sharkawi, and R. J. Marks II, "Artificial Neural Networks for Power System Static Security Assessment," International Symposium on Circuits and Systems. Portland, Or, 1989.

[4] R. Podmore, "Identification of Coherent Generators for Dynamic Equivalents," IEEE Trans. on Power App. and Sys., pp. 1344-1354, Aug. 1978.

[5] M. A. El-Sharkawi, "Choice of Model and Topology for External Equivalent Systems," IEEE Trans. on Power App. and Sys., pp. 3761-3768, Dec. 1983.

[6] M. A. El-Sharkawi and B. J. Brewer, "Excitation Control Design of Local Machine in a Large-Scale Power System by Using Dynamic Equivalencing Technique," Electrical Power and Energy Systems, pp. 165-174, July 1985.

[7] B. Widrow, R. G. Winter & R. A. Baxter, "Layered neural nets for pattern recognition," IEEE Trans. on Acoust., Speech, and Signal Proc., Vol.(36), pp.1109-1118 (1988).

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, Vol.(323), pp.533-536 (1986).

[9] R. J. Marks II, S. Oh and L. E. Atlas, "Alternating projection neural networks," IEEE Trans. CAS in print.

[10] R. J. Marks II, S. Oh, L. E. Atlas and J. A. Ritcey, "Homogeneous and layered alternating projection neural networks," Proceedings of the International Symposium on Optical Engineering and Industrial Sensing for Advanced Manufacturing Technologies, 26-30 June 1988, Dearborn Hyatt, Michigan.

[11] R. J. Marks II, L. E. Atlas, S. Oh and J. A. Ritcey, "The performance of convex set projection based neural networks," Neural Information Processing Systems, Dana Z. Anderson, editor, (American Institute of Physics, New York, 1988), pp. 534-543.

[12] R. J. Marks II, L. E. Atlas and S. Oh, "Generalization in layered classification neural networks," Proceedings of the IEEE International Symposium on Circuits and Systems, Helsinki, June 7-9, 1988.

[13] R. J. Marks II, L. E. Atlas, D. C. Park and S. Oh, "The effect of stochastic interconnects in artificial neural network classification," Proceedings of the IEEE International Conference on Neural Networks, San Diego, July 24-27, 1988, vol.II, pp.437-442.

[14] J. L. McClelland, D. E. Rumelhart and the PDP research group. Parallel distributed processing, Vols. I, II and III, Bradford Books, Cambridge, MA. (1986)

[15] R. P. Lippmann, "An introduction to computing with neural nets," IEEE ASSP Magazine, pp.4-22. 1987.