PRELIMINARY RESULTS ON USING ARTIFICIAL NEURAL NETWORKS FOR SECURITY ASSESSMENT

M. Aggoune, M.A. El-Sharkawi, D.C. Park, M.J. Damborg and R.J. Marks II

Department of Electrical Engineering
University of Washington
Seattle WA 98195

Abstract

In this paper, Artificial Neural Networks techniques (ANN's) are explored as a tool to assess the dynamic security of power systems. The basic role of ANN's is to provide assessment of the system's stability based on training examples from off-line analysis. Such an assessment would be useful as an operations aid. In essence, ANN's interpolate among the planning analysis data.

This paper contains the results of a study to assess the capability of ANN's to "learn" from off-line stability analysis results and give accurate stability assessments when queried with data representing the current system status. The important feature of the result is that correct stability assessments are provided by the ANN not only when it is queried with an element of the training set of data but also at other operating conditions.

Keywords: Power System Security, Artificial Neural Networks

Introduction

This paper explores the suitability of using artificial neural networks (ANN's) 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 "operations planning" 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.

The security assessment problem results from the constantly changing behavior of a power system. Many of these changes can be anticipated, e.g. changing load patterns. However, others, such as a line outage, happen without warning. We can think of the system's operation as being represented by a large set of key system variables. The system operating "state" is then a point in this high dimensional variable space. Changes in system operation correspond to moving this state in the "operating space."

Some regions of this space represent satisfactory system behavior where all constraints are met (e.g. thermal limits of lines), system generation is distributed for good economy and the system is stable and would remain stable in the face of all probable disturbances. Such regions are said to represent the "normal" operating states [1]. Other regions, the "alert" states, represent situations where realistic disturbances, if they occur, will cause constraint violations and could lead to instabilities. A system in a normal state is called "secure" while one in an alert state is "insecure."

Operators monitor the system and attempt to assure that the system remains secure. The attentive operator can identify some pending problems and take corrective action. But identifying these problems is difficult, even impossible in many cases given the present system monitoring ability. Ideally, the operator would like to be able to "view" the point representing a system in a normal operating state and observe when it approaches a security boundary, i.e. when it approaches alert states. Unfortunately, such a tool does not exist. However, we suggest that it may be possible to develop on-line aids which will give the operator some of this assistance.

In this paper we are concerned with operator aids for identifying regions of dynamic, or steady state, instability. For steady state stability analysis it is appropriate to examine the eigenvalues of a linearized version of the system model, linearized 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 reducing the model using dynamic equivalents yields a large model and one that may be unreliable for large classes of disturbances [2,3]. Hence, on-line aids must probably be built using off-line analysis. That is, "operations planners" are able to invest the time necessary to analyze a number of "critical" cases. If these cases suitably explore the secure and insecure regions (the normal and alert states), then the operator can use them as a guide.

While this approach is currently used, at least in principle, it is not really satisfactory for a number of reasons such as: (1) To be

This paper was sponsored by the IEEE Power Engineering Society for presentation at the IEEE Power Industry Computer Application Conference, Seattle, Washington, May 1 - 5, 1989. Manuscript was published in the 1989 PICA Conference Record.

0885-8950/91/0500-0890$01.00©1991 IEEE

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!

This description of a desirable operator aid suggests that pattern recognition might be suitable [4]. Indeed, some such attempts have been made for security assessment in the case of transient stability. Pang, et.al. [5] developed a pattern recognition system to identify secure and insecure states. Results on a modestly sized system showed approximately a 90% correct classification rate for a single fault location. Hakimmashhadi and Heydt [6] showed they could improve the rate of correct classification when they included a function of the transient energy in the patterns. Yamashiro [7] used two carefully chosen features which represented transient energy and margins. Good classification results were obtained with a very simple discriminating function. All these studies were able to find classifiers which gave good results for small systems and single fault locations. There have been no attempts to scale or generalize them.

An ANN is rather similar, in principal, to a pattern recognition system. However, there is a fundamental difference. The basic steps in building a pattern recognition system are to determine the patterns and features required, specify the class of functions used for recognition and then identify the free parameters [4]. The complexity of the relationships that can be modeled depends on the features used and on the prespecified class of discriminating functions. In other words, there must be considerable prior knowledge about the functional relationships that are to result.

On the other hand, building an ANN requires selecting appropriate input and output variable sets, an appropriate architecture (neurons and interconnection structure) and an appropriate training algorithm. Here, complexity is determined by the number of neurons in the network and the functional relationships used for interconnections. The ANN's effectiveness results because the number of neurons may be large and the interconnections may be nonlinear. Further, the functional dependencies between input and output need not be prespecified but, rather, they evolve during the training process and can be highly complex.

The approach of this paper to use an ANN as an operator aid for monitoring security relative to dynamic stability is motivated by the fact that ANN's have demonstrated the ability to store very complex relationships when applied to signal processing or classification problems [8-11]. They have also modeled complex relationships in power systems as demonstrated by Sobajic and Pao who used an ANN to identify the critical clearing time for transient stability for a single fault in a small power system [12].

To obtain satisfactory performance from an ANN, it is necessary to have an appropriate structure (inputs, outputs, neurons and interconnections), an appropriate training algorithm and a sufficient set of training cases. We illustrate this approach with an example which trains an ANN to recognize the region of steady state stability for a 9 bus system involving 3 machines. For this small system, we find very promising behavior from an ANN consisting of 50 to 80 neurons depending on the number of input variables. Though this example is small, we find it encouraging because the system model is mathematically complex. We present this example after a discussion of the basic principles of ANN's.

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 [13-19]. The top layer receives the input vector, i, that stimulates the network. Each element of this vector is weighted by the input to hidden interconnects, t_{ik}, to form at the middle, or hidden, layer a weighted sum. This sum is altered by a nonlinearity (e.g. sigmoid) to establish the state of each hidden neuron. A linear combination of the hidden neural states is used to generate the output states, denoted by the vector, o. The interconnects between the hidden and output neurons is denoted in the figure by c_{kj}. Layered ANN's can be trained by iteratively inputting training data [8,11,13,14], or can be trained by observing the training data only once. 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. When training is achieved by viewing the data only once, the number of hidden neurons must exceed the cardinality of the set of training vectors. In this paper, we use such an ANN. A highly regarded tutorial on other aspects of classification ANN's is given by Lippmann [11]

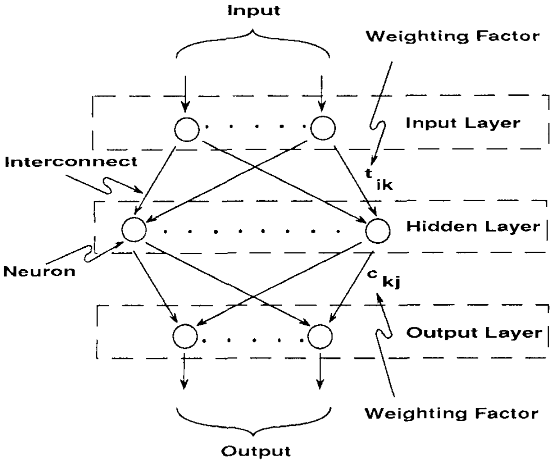

Figure 1. Structure of Three-Layered ANN

The ability of an ANN classifier to respond to multifarious training data increases with the number of available neurons. Assuming that each neural state is in some sense uncorrelated with those remaining, each neuron represents a computational degree of freedom available to the network. The number of degrees of freedom can be artificially increased through the use of neurons in a hidden layer, the states of which can be almost any nonlinear combination of the stimulus neural states [15-19]. One approach is to generate nonlinearities with stochastically chosen interconnects between the input and hidden neural layers with a sigmoidal nonlinearity at each hidden neuron [19]. The hidden to output interconnects are chosen to be a (trainable) projection matrix [18] whose values are a function of the stochastically chosen interconnects and the training data. Preliminary simulations of such networks show an approach to fixed generalization classification partition boundaries as the number of hidden neurons becomes larger [19].

Although the use of hidden neurons with arbitrarily determined nonlinear states is potentially applicable to a large number of artificial neural networks, we limit our investigation here to the projection artificial neural network.

A Projection Based Artificial Neural Network

A set of stimuli vectors $\{s_n \mid 1 \le n \le N\}$ is to be made to correspond to a set of response vectors $\{r_n \mid 1 \le n \le N\}$. That is, we wish to design a classifier that will output, say, r_3 when the input is s_3. We define the stimulus and response matrices respectively as

$$S = [\, s_1 \mid s_2 \mid \dots \mid s_N \,] \tag{1}$$

and

$$R = [\, r_1 \mid r_2 \mid \dots \mid r_N \,] \tag{2}$$

The hidden states will be denoted by $\{ h_n \mid 1 \le n \le N \}$, where

$$h_n = \Phi\, s_n \,; \quad 1 \le n \le N \tag{3}$$

and Φ is some nonlinear operation. The hidden layer matrix follows as

$$H = [\, h_1 \mid h_2 \mid \dots \mid h_N \,] \tag{4}$$

In an artificial neural network architecture, the number of input neurons is equal to the length of a stimulus vector. Each input neuron is connected to each hidden neuron in order to achieve a nonlinear mapping. The interconnects between the hidden and output neurons are given by elements of the projection matrix

$$C = R\,[\, H^T H \,]^{-1} H^T \tag{5}$$

In practice, the hidden to output interconnects are trained using an updating rule that requires examination of the training data only once [18].

Once trained, the network output, o, corresponding to an input vector, i, is given by

$$o = C\Phi\, i \tag{6}$$

A nonlinearity Φ that is useful in artificial neural network architectures is

$$\Phi\, s_n = \eta\, T\, s_n \tag{7}$$

where T is the matrix of input-to-hidden interconnects and η is a nonlinear pointwise vector operator (e.g. sigmoid). A hidden neuron adds the contribution from all the input neurons, and adopts a state equal to a nonlinear function of this sum. The state of the output neurons is then a weighted sum of the hidden neural states. Equation (6) then becomes

$$o = C\eta\, T\, i \tag{8}$$

Almost any nonlinearity will allow the trained artificial neural network to respond correctly to training data. The manner in which the network responds to data outside of the training set (i.e. how the network generalizes) is dependent on the choice of the nonlinearities. In this paper, the elements of the interconnect matrix, T, are chosen stochastically from a zero mean uniform probability distribution. Such a procedure has been shown to result in good classification diversity [19].

Test System and Algorithm

Figure 2 shows the working example used in this study. It is a simple system composed of 3 machines, 9 buses, 11 transmission lines, 3 loads and 3 capacitive compensators. The data of the system is given in tables 1 to 3.

Table 1 Generator Data

Generator #	1	2	3
Type	nuclear	steam	steam
Rated kVA	245.000	192.000	128.000
kV	14.400	18.000	13.800
P,F	0.850	0.850	0.850
X_d'	0.320	0.315	0.232
X_d	1.710	1.670	1.680
ξ	0.100	0.100	0.100
M	9.254	6.214	4.766
T_{d0}'	7.100	5.000	5.890
T_A	0.200	1.670	1.680
K_A	50.000	50.000	50.000

Table 2 Transmission Line Data

line	Bsh	Rser	Xser
1	0.0000	0.0000	0.0151
2	0.0000	0.0000	0.0140
3	0.0000	0.0000	0.0213
4	0.0010	0.0017	0.0301
5	0.0027	0.0018	0.0412
6	0.0018	0.0017	0.0530
7	0.0017	0.0016	0.0223
8	0.0025	0.0024	0.0305
9	0.0015	0.0016	0.0116
10	0.0058	0.0039	0.0520
11	0.0062	0.0045	0.0610
12	0.0061	0.0049	0.0730
13	0.0058	0.0045	0.0810
14	0.0062	0.0049	0.0930

Table 3 Load and Capacitors Data

bus	P	Q	B_c
7	1.210	0.350	0.031
8	1.110	0.250	0.011
9	0.814	0.200	0.020

Machine number 3 is arbitrarily selected as the study machine. The interactions among the machines is represented by the multi-machine power system model [3,20].

The proposed ANN technique for power system security assessment, related to system's stability, is described by the following procedure:

Step 1: Identify the Contingency Parameters: Specify the parameter(s), variable(s) and/or topologies which have impact on power system security; such as transmission line status, load status, machines excitations and generation level. These contingency parameters are the input stimulus for the ANN.

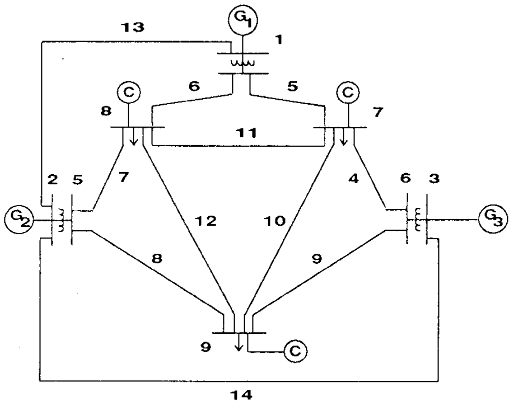

Figure 2. Test System

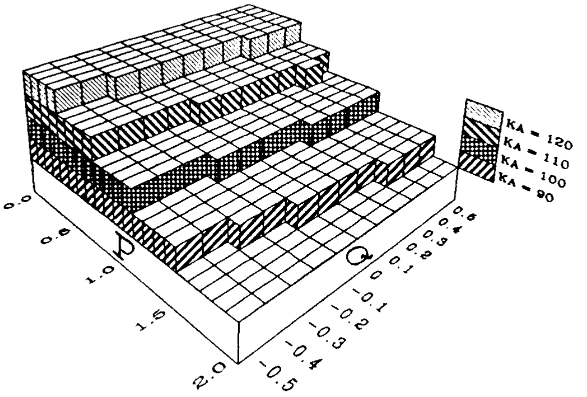

Figure 3. Three-Dimension Security Contours

Step 2: Establish off-line Security Contours: Obtain the security contours of the system for some values of each contingency parameter. These security contours, which enclose the steady state stability area, can be obtained by assessing the eigenvalues of the entire power system.

An example of the security contours is shown in the three dimension plot of Figure 3. The eigenvalues of the entire system inside the security contours have negative real components. The contingency parameters in this case are the real and reactive powers of machine 3, and its excitation gain. The figure shows 4 layers of security contours, each one corresponding to one value of the excitation gain. The upper and lower limits on the reactive power represent operation restraints.

Step 3: Train the ANN: In the training process, the ANN comprehends the security contours established by step 2. Instead of using the entire contour for training, only selected points that represent stable and unstable operations can be utilized.

The input neurons of the ANN receive an input pattern that contains the stimuli vectors contained in matrix (S). The input pattern is a set of contingency parameters. The magnitudes of these parameters form the stimulus matrix (S). The output neurons receive the status of the system stability (response matrix R) that correspond to the input stimuli vectors. This training process is depicted in Figure 1. Training the network means identifying the topology of the ANN and the weights of its interconnects (t_{ik}, c_{kj}).

Step 4: Test the ANN: After the ANN is trained by a few values of the contingency parameters, it is tested at other values not necessarily part of the training set. In this case, which is represented by equation (8), the input neurons are given a set of input vector (i) which contains a set of contingency parameters, and the output neurons produce the status of system stability (o).

Test Results

The ANN presented in this paper is designed to classify the system stability for various input patterns (contingency parameters). The input patterns used to train the ANN, and to test the network, are summarized by the following cases:

Case 1: The input pattern is composed of (1) real power of the study machine; (2) reactive power of the study machine and (3) excitation gain of the study machine.

Case 2: The input pattern is composed of (1) and (2) of case 1 in addition to the real power generated by machine 2.

Case 3: The input pattern is composed of (1) and (2) of case 1 in addition to the load demand at bus 8.

Case 4: The input pattern is composed of (1), (2) and (3) of case 1 in addition to the status of lines 9 and 10. In this case the ANN has 4 inputs.

Case 5: The input pattern is similar to those in case 1 in addition to the load demand at bus 8. In this case the ANN has 4 inputs.

Extensive results were obtained during this study. However, for brevity, only typical ones are presented in this paper.

In all the following figures, P and Q are the real and reactive power of the study machine (machine 3), K_A is the excitation gain of machine 3, D_l is the load demand at bus 8, and P2 is the real power of generator 2.

The following figures are divided into two categories: one shows the training data and the second shows the test results. When the ANN is tested, the values of the input patterns were different from those used in training.

Figure 4 shows the training data points that form the stimuli vectors (S) for case 1. Three different values of excitation gain (K_A) of machine 3 were used: 90%, 110% and 120%. The "+'s", "*'s" and "x's" are the training data corresponding to unstable operating points at the above mentioned values of K_A. The dots are the training data corresponding to stable operating points.

The ANN was tested at various operating conditions other than those used in training. For example, Figure 5 shows the test result at excitation gain K_A = 100%. This value of K_A was not used in training the ANN. The dots in the figure represent test points for stable operation as indicated by the ANN. For verification purpose, a security contour at K_A = 100% is also shown in the figure with "x's". This security contour was not used in training the ANN.

Figure 6 shows the training data for case 2. The ANN is trained at three values of real power generated by machine 2 (P2): 50%, 100% and 125%. The dots are the training data corresponding to stable operating points. The "+'s", "*'s" and "x's" are the training data corresponding to unstable operating points at the three different values of P2.

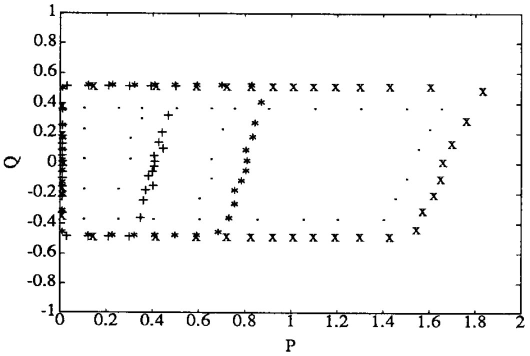

Figure 4. Training Data for Case 1
(x) K_A = 90%; (*) K_A = 110%; (+) K_A = 120%

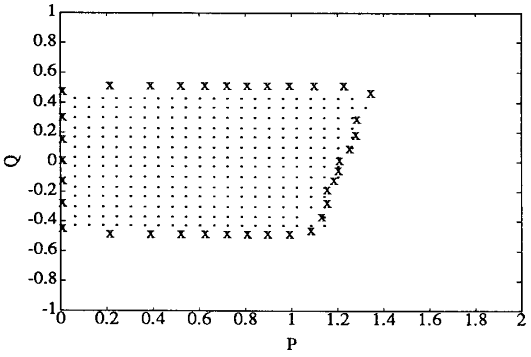

Figure 5. Test Results at K_A = 100%

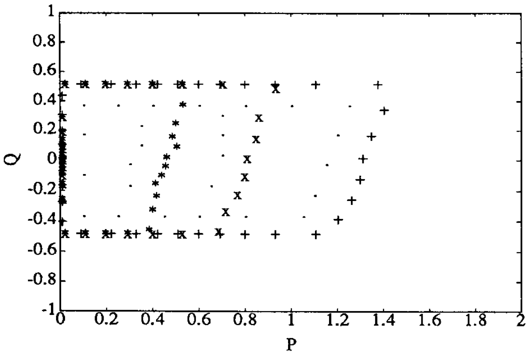

Figure 6. Training Data for Case 2
(x) P2 = 100%; (*) P2 = 125%; (+) P2 = 50%

Figure 7 shows the result of testing the ANN when P2 = 75%. The dots in the figure represent test points classified by the ANN as stable operating points. For verification purpose, a security contour for P2 = 75% is also shown in the figure with "x's". This security contour was not used in the training process.

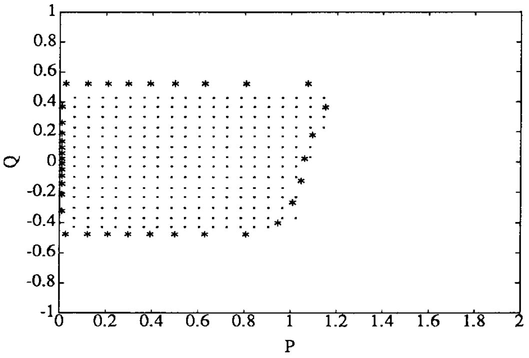

Figure 7. Results at P2 = 75%

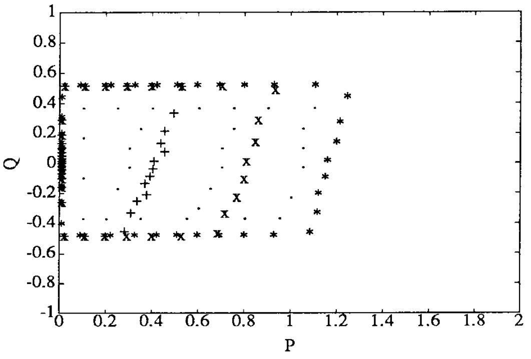

Figure 8. Training Data for Case 3
(x) D_l = 100%; (*) D_l = 200%; (+) D_l = 0%

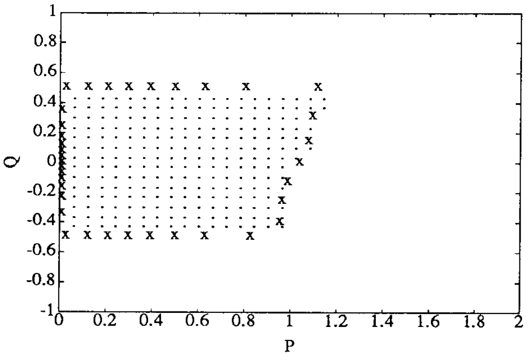

Figure 9. Test results at D_l = 150%

Figure 8 shows the training data for case 3. The ANN is trained at three different values of load demand at bus 8 (D_l): 0%, 100% and 200%. Figure 9 shows the result of testing the ANN at a load demand of 150%. The security contour at D_l = 150% is also shown in the figure for verification purpose only. It was not used in training the ANN.

Figure 10 shows the training data for case 4. The line statuses for training the ANN were arbitrary selected as follows: line 9 is out, or both lines 9 and 10 are out. In each case two different values of the excitation are used: 90% and 110%. In this case the ANN is trained by an input pattern composed of four variables. The ANN is tested at an excitation of 100% with the same line statuses used in training. A sample of the test results is shown in Figure 11. The security contour shown with "x's" in the figure is used for comparison purposes only. It was not used during the training process.

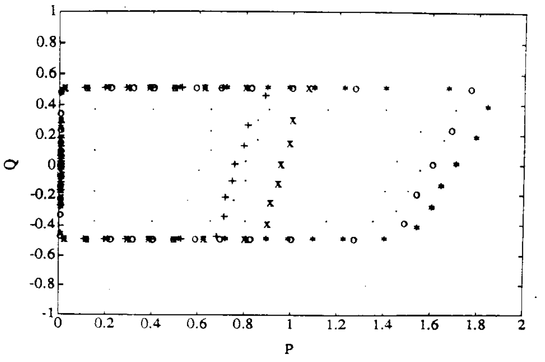

Figure 10. Training Data for Case 4
(x) K_A = 110% and L9 & L10 out; (*) K_A = 90% and L9 & L10 out; (+) K_A = 110% and L10 out; (o) K_A = 90% and L10 out

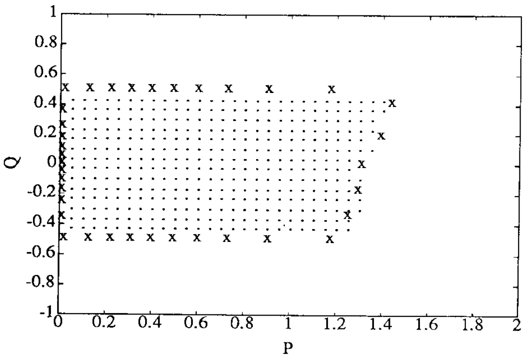

Figure 11. Test results at K_A = 100 and L9 & L10 out

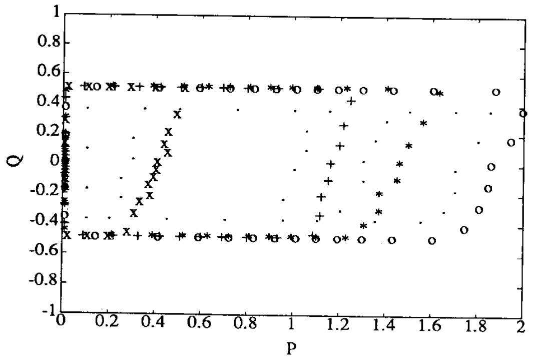

Figure 12. Training Data for Case 5
(x) K_A = 110% and D_I = 0%; (*) K_A = 90% and D_I = 0%; (+) K_A = 110% and D_I = 200%; (o) K_A = 90% and D_I = 200%

Results of case 5 are depicted in Figures 12 and 13. Figure 12 shows the training data. In this case the input pattern is composed of four variables. Figure 13 shows a sample of the test results when the load is at 100% and the excitation is at 90%. This test data was not used to train the ANN. Only for comparison purpose, the security contour for this test case is shown in the figure.

As seen from all the above test cases, the ANN was very successful in classifying the status of power system stability. Only very few minor misclassifications are observed near the boundary of the security contour.

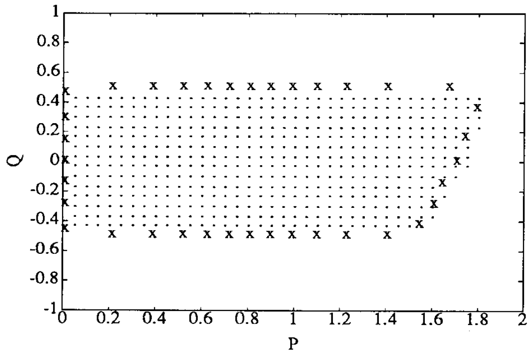

Figure 13 Test Results at K_A = 90% and D_I = 100%

Conclusions

We view these results as being rather encouraging for the future of ANN's in power system applications. We have demonstrated a number of cases where the network was able to properly interpolate among training data sets to recognize stability contours. As emphasized in the introduction, a considerable part of the importance of this result is due to the complexity of the mathematical relationships being represented. However, one suspects that the smoothness and regularity of the contours contributes to this successful demonstration.

Naturally, the primary concern about whether this concept can become useful as an on-line aid resides in the question of scaling. As with the cited pattern recognition, the working example in this paper is a very small system. We see at least two possibilities for developing realistic applications. First, it is possible that the present technology for training large ANN's can be extended to full scale power network application. This may require methods for partitioning the ANN and the training process. It may also require special ANN computers which are currently being developed at a rather fast acceleration. The second possibility is that a special purpose ANN can be built to monitor a specific operational situation. For example, there may be a particular but important relationship between stability and a few system variables. It may be quite feasible now to build a modest size ANN to monitor such a special situation. Then, of course, it is reasonable to contemplate a few ANN's to monitor a few such special situations. We may be able to use this type of ANN application in the rather near future.

Acknowledgments

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

References

[1] L.H. Fink and K Carlsen, "Operating Under Stress and Strain," IEEE Spectrum, pp. 48-53, March 1978.

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

[3] 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.

[4] K.S. Fu, Sequential Methods in Pattern Recognition, Academic Press, 1968.

[5] C.K. Pang, F.S. Prabhakara, A.H. El-Abiad and A.J. Kiovo, "Security Evaluation in Power Systems Using Pattern Recognition," IEEE Trans. on Power App. and Sys., pp. 969-976, May/June 1974.

[6] H. Hakimmashhadi and G.T. Heydt, "Fast Transient Security Assessment," IEEE Trans. on Power App. and Sys., pp. 3816-3824, Dec. 1983

[7] S. Yamashiro, "On Line Secure-Economy Preventive Controller of Power System by Pattern Recognition," IEEE Trans. on Power App. and Sys., pp. 214-219, Aug. 1986.

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

[9] Grossberg, S. (1988) Neural Networks and Natural Intelligence, MIT Press (1988).

[10] Kohonen, T. Self-organization and associative memory), 2nd edition, Springer-Verlag, Berlin.(1987)

[11] Lippmann, Richard P. "An introduction to computing with neural nets" IEEE ASSP Magazine, pp.4-22 (1987).

[12] D.J. Sobajic and Y. Pao, "Artificial Neural-Net Based Dynamic Security Assessment for Electric Power Systems," Presented at the IEEE Power Engineering Society Winter Meeting, Feb. 1988.

[13] Widrow, B., 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).

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

[15] R.J. Marks II, S. Oh, L.E. Atlas and J.A. Ritcey "Alternatin projection neural networks", IEEE Trans. CAS (in review).

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

M. Aggoune was born in Tunisia in 1955. He received his B.S. degree in 1981 from Algeria and his M.S.E.E from the University of Washington in 1984. He is currently finishing his Ph.D. degree at UW. His major areas of research include power system stability and artificial neural networks applications.

M. A. El-Sharkawi (S'76-M'80-SM'83) was born in Cairo, Egypt, on April 28, 1948. He received his B.SC. in Electrical Engineering from Cairo High Institute of Technology, Egypt in 1971. His M.A.SC and Ph.D. in Electrical Engineering were received from the University of British Columbia in 1977 and 1980 respectively. In 1980 he joined the University of Washington as a faculty member where he is presently an associate professor.

His major areas of research include power system dynamics and control, application of artificial neural networks to transient stability, electric drives, and power electronic. Most of his research in all these areas are funded by the US government, and by public and private industrial organizations.

M.J. Damborg received his B.S. Degree in Electrical Engineering in 1962 from Iowa State University, and the M.S. and Ph.D. degrees from the University of Michigan in 1963 and 1969, respectively. He spent the academic year 1966-67 as a Fulbright Fellow at the University of Delft, Netherland.

Since 1969, Dr. Damborg has been at the University of Washington where he is now Professor of Electrical Engineering and Associate Chairman for Graduate Study. His research interests concern analysis and control of dynamic systems with emphasis on power systems. Associated interests include database applications to computer aided engineering systems.

R.J. Marks II received his B.S. and M.S. Degrees from Rose-Hulman Institute of Technology in 1972 and 73 respectively. His Ph.D. Degree was received in 1977 from Texas Tech University. Dr. Marks is the Chairman of the Technical Society on Neural Systems and Applications in the Circuit and Systems Society. He has published a numerous number of papers in the field of Artificial Neural Networks and Optics. Since 1978 he joined the department of Electrical Engineering at the University of Washington where he is currently a Professor.