ARTIFICIAL NEURAL NETWORKS FOR POWER SYSTEM STATIC SECURITY ASSESSMENT

M.E.Aggoune, L.E.Atlas, D.A.Cohn, M.J.Damborg, M.A.El-Sharkawi, and R.J.Marks II

Department of Electrical Engineering, University of Washington, Seattle WA 98195

Abstract

An Artificial Neural Network (ANN) is used to assess the static security of a test system. The technique is contrasted with that of using a nearest neighbor search. The ANN is shown to perform significantly better in terms of classification, recall time, and data storage requirements.

Introduction

Static security of a power system addresses whether, after a disturbance, the system reaches a steady state operating condition that does not violate given system operating constraints. These constraints ensure the power in the network is properly balanced, the magnitude of all bus voltages are within acceptable limits, and the thermal limit of each transmission line is not exceeded [1]. If any one constraint is violated, the system may experience disruptions that could result in a "brown out" or even a "black out". Hence, the power system is insecure.

Security assessment typically involves two steps: off-line analysis and on-line assessment. With off-line analysis, the status of the power system is evaluated for various probable disturbances, such as the loss of a transmission line or a generating unit. In steady state, the static security is evaluated using the load flow equations. The load flow is solved for various types of disturbances and the results are compared with the system constraints. Violations, if any, are then identified and the operating condition is labeled secure or insecure [2]. For convenience, the results of this off-line analysis are stored in lookup tables.

In step two (on-line security assessment), an operator may check the security of the system using the lookup tables. If the operating condition has previously been simulated, the security status is determined. However, when the system operating condition has not been simulated, a precise assessment of the system security cannot be made. A method such as a nearest neighbor search must then be used. For a lookup table to include all possible operating conditions, a huge number of simulations must be made. This is a prohibitive process that demands excessive computational time.

In this paper, a different approach is proposed for static security assessment. This approach is based on Artificial Neural Networks (ANN's). By using ANN's, (1) security regions can be defined by training the ANN through examples; (2) the training data can be randomly obtained from real system operation or simulated by off-line techniques; (3) the amount of data used in training is much less than that in lookup tables and yields comparable (if not better) assessment accuracy; and (4) on-line security assessment can be done much faster.

ANN's can be loosely defined as highly connected arrays of elementary processors. When used for classification, the processors, or neurons, are typically partitioned into layers. Such ANN's have the ability to learn from training data rather than from a set of rules. Other attributes include fault tolerance, regularized architectural structure and computational parallelism. Design of neural networks and analysis of their performance to date has relied primarily on steepest descent and energy reduction algorithms [3]. For learning, steepest descent algorithms can perform painfully slowly but with quite impressive final results. Indeed, increasing the speed of network learning is presently one of the most important needs in neural networks [4]. ANN's, when used for static security assessment, operate in two modes: training and recall. In the training mode, the ANN learns from data such as real measurements or off-line simulation. In the recall mode, the ANN can provide an assessment of system security even when the operating conditions are not contained in the training data.

This paper presents preliminary results of using ANN for static security assessment. The AEP 8-bus is chosen as a test system and used to demonstrate the capability of the proposed technique. The results in this paper show that the ANN in the recall mode performs much faster and more accurately than using nearest neighbor classification.

Static Security Assessment

The normal steady state operation of a power system requires that the generator power satisfy

$$\Sigma_i P_{Gi} = P_D + P_L \qquad (1a)$$

$$\Sigma_i Q_{Gi} = Q_D + Q_L \qquad (1b)$$

Where P_{Gi} and Q_{Gi} are the real and reactive powers of generator at bus (i); P_D and Q_D are the total real and reactive load demands; P_L and Q_L are the real and reactive losses in the transmission network.

Inequality constraints must always be imposed on the system to ensure secure operation. All bus voltages must be bounded, all line currents must not exceed the respective thermal limits, and all generator power outputs must be limited. These constraints can be expressed as follows:

$$\begin{aligned} V_{min} &< V_j < V_{max} &&; j=1,n_B \\ S_l &< S_{l,max} &&; l=1,n_L \\ P_{i,min} &< P_i < P_{i,max} &&; i=1,n_G \\ Q_{i,min} &< Q_i < Q_{i,max} &&; i=1,n_G \end{aligned} \qquad (2)$$

where V_j is the voltage at bus (j), and S_l the apparent power of line (l). n_B, n_L, and n_G are the number of buses, lines, and generators respectively.

(217) 333-4789 Further information can be obtained from Dr. W. Kenneth Jenkins.

We assume that the steady state operating condition of the power system is governed by an optimal dispatch strategy [5,6]. In this case, the total power cost function is given by:

$$C = \Sigma_i C_i \qquad ; i = 1, n_G \tag{3}$$

where C_i, the cost function of individual generators, is given by:

$$C_i = C_{i2} P_{Gi}^2 + C_{i1} P_{Gi} + C_{i0} \tag{4}$$

where C_{i0}, C_{i1} and C_{i2} are constant coefficients for a given generator.

The optimum dispatch problem is solved by minimizing the cost function C while satifying equation (1). Mathematically, the problem reduces to minimizing the following augmented cost index J,

$$J = C - \lambda \, (\Sigma_i P_{Gi} - P_D - P_L) \tag{5}$$

where λ is a Lagrange multiplier.

Classification Algorithms

The problem addressed in this paper is concerned with the static security following a line outage and under various levels of apparent power , S, of a given load. The security of the system is assessed for various values of real power, P, and reactive power, Q, of a second load.

Back Propagation Algorithm

The neural network topology used for the results presented in this section consists of a 3-input, 2-output and 10 hidden unit network. This network (with a constant input bias of 1.0 for all thresholds) is shown in Figure 1. Ideally, the network will predict secure system operation by having the output unit produce a value of 1 and will predict insecure operation by having an output value of 0.

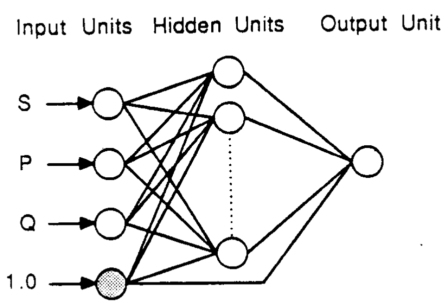

Figure 1. Neural network architecture

As in other trainable pattern classifiers, a set of input-output examples is used to determine the classifier parameters which, in the case of ANN's, are the interconnect values between adjacent neuron layers. The network we use has 1 hidden layer to allow for arbitrary convex decision regions. The learning rule is a variant of steepest descent which is back-propagated to estimate the weights of the hidden units [7].

As seen in Figure 1, the three network inputs are S, P, and Q. In the forward path the output signal from the n-th neural unit in the input layer is multiplied by a weight and accumulated over the inputs as

$$x_n^{i+1} = \sum_{m=1}^{4} (w^i_{mn} x^i_m) \tag{6}$$

After passing through a weighted bias θ_n^{i+1} and a function f, the output value of the n-th neural unit in the hidden layer is obtained by

$$x_n^h = f(x_n^{i+1}, \theta_n^{i+1}) \tag{7}$$

For the output unit,

$$x^{i+2} = \sum_{m=1}^{10} (w_m^{i+1} x_m^h) \tag{8}$$

and the calculated output value is

$$x^o = f(x^{i+2}, \theta^{i+2}) \tag{9}$$

An important value is the output error

$$e^o = 0.5\,(d^o - x^o)^2 \tag{10}$$

where d^o is the desired output (zero or one).

The back-propagation training technique (which attempts to minimize e^o) consists of an error calculation for each unit and the adaptation of weights. At the output, the error is calculated by

$$\delta^o = (d^o - x^o)\, f'(x^{i+2}, \theta^{i+2}) \tag{11}$$

where f ' is the derivative of f. At the hidden layer, the errors are estimated as

$$\delta_m^h = f'(x_m^{i+1}, \theta_m^{i+1})\, \delta^o w_m^{i+1} \tag{12}$$

The trained weights are determined for both layers by the iterations

$$w^i_{nm}(t+1) = w^i_{nm}(t) + \eta\, \delta_n^h x_m^i \tag{13a}$$

$$w_n^{i+1}(t+1) = w_n^{i+1}(t) + \eta\, \delta^o x_n^h \tag{13b}$$

The training of this network requires many iterations until the values of the interconnect weights converge. The use of a trained network, however, requires only one forward pass.

Nearest Neighbor Algorithm

The nearest neighbor algorithm is also trainable by examples. It is intended as a baseline classifier which is used for performance comparisons to the neural network classifier. The nearest neighbor scheme memorizes each training point as

$$t_i = (P_i, Q_i, S_i, O_i) \tag{14}$$

In order to use the classifier, a set of inputs, t, is compared exhaustively with the first three dimensions of all the training points and the value of O_i (zero or one) for the closest match using the distance

$$d(t, t_i) = [(P-Pi)^2 + (Q-Qi)^2 + (S-Si)^2]^{0.5} \tag{15}$$

where S, P, and Q are values for which system security is unknown. The security is found via

$$d(t, t_{min}) = \min_i \{d(t, t_i)\} \tag{16}$$

The value O_{min} then is chosen to indicate the predicted security.

This nearest neighbor algorithm is, for large amounts of training data, asymptotically very close to the optimal Bayer classifier. However, since an exhaustive search through the training data is very inefficient, the nearest neighbor algorithm becomes impractical for realistically-sized problems.

Test System

The test system is shown in Figure 2. This system is composed of 8 buses, 14 transmission lines, and 4 generators.

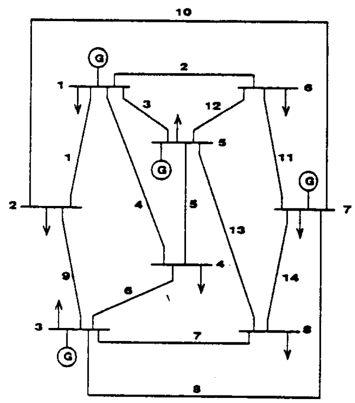

Figure 2. Test system

Test Results

Comparative Performance for a Fixed Level of Apparent Power

Training data was generated under the scenario that line 4 is disconnected and apparent power of the load at bus 8 is fixed at S=100% of its normal value. A total of 2000 training points were randomly chosen from a total set of 6561 points generated by varying the real power, P (in MW), and the reactive power, Q (in Mvar), of the load at bus 6. The back-propagation and nearest neighbor techniques were used to predict security. This fixed apparent power case allowed 2 input parameters (P and Q) to be varied, thus making it possible to interpret classification performance in a 2-dimensional picture.

The back-propagation result is shown in Figure 3. The interior of both curves represents secure regions. The dashed curve represents the boundary between the secure and insecure results. This boundary was obtained by a nearly exhaustive search. The solid curve represents the predictions of the back-propagation algorithm. It should be noted that while 2000 points were used for training, the boundary curve represents the result of testing the network with the complete set of 6561 points.

The back-propagation result shows that the hidden units allowed for a non-linear and smooth partition boundary. While the algorithm did not give a perfect fit, the maximum deviation from the true boundary was quite small.

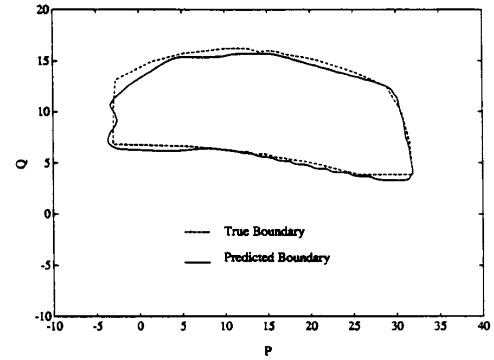

Figure 3. Back-Propagation classifier (S=100%)

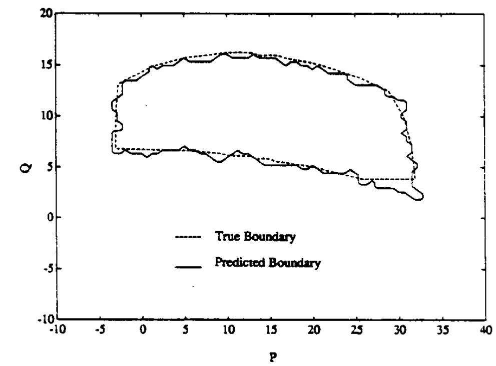

Figure 4. Nearest-Neighbor classifier (S=100%)

The nearest neighbor result is shown in Figure 4. Note that the predicted boundary is less smooth than for the back-propagation case and that, for small reactive power, the maximum deviation from the true boundary is quite large.

The amount of data required to train a classifier is an important performance measure. Figures 5 and 6 represent this tendency to asymptotic performance for both classifiers.

It is encouraging that the number of back-propagation failures (in terms of false secure states) is, in general, no worse than the nearest-neighbor result. As the training set size increased by a factor of 4, the number of falsely predicted secure states decreased by about a factor of 3.

The lower-cost error of false insecure states occurrs at a rate similar to that of false secure states. Since the classification techniques do not assign any explicit cost to the type of error, this result is expected. Either classifier can be easily augmented to allow for an asymmetric cost function.

(217) 333-4789 Further information can be obtained from Dr. W. Kenneth Jenkins.

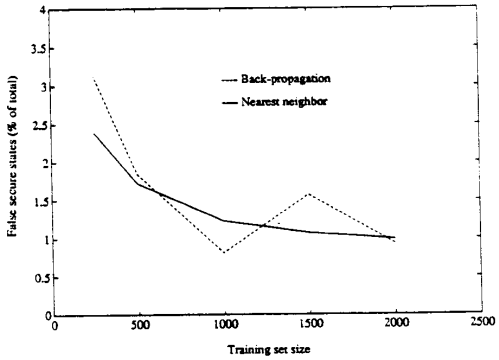

Figure 5. Training set size vs. false secure states

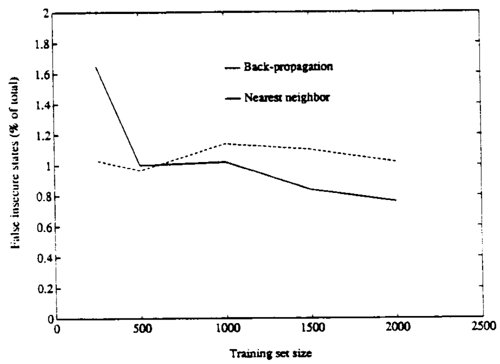

Figure 6. Training set size vs. false insecure states

Effect of Variable Apparent Power

In the previous case, the apparent power of the load at bus 8 was fixed. In this section, we explore the effect of also varying this apparent power.

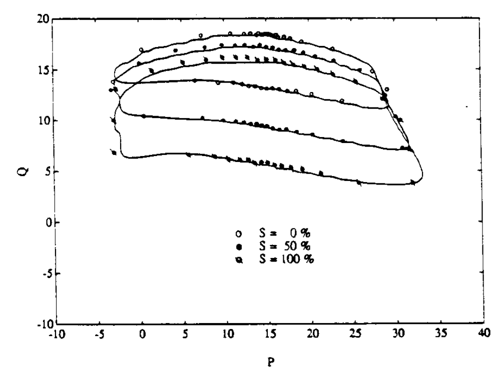

Figure 7. Back-Propagation trained on 1469 points

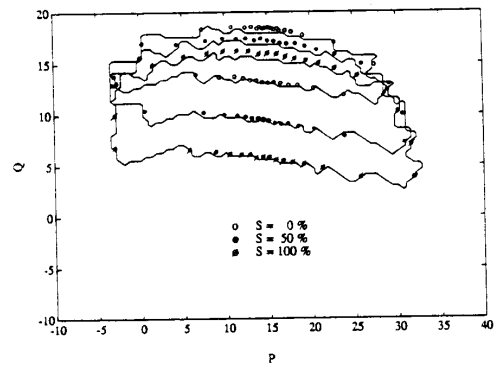

Figure 8. Nearest-Neighbor trained on 1469 points

Figures 7 and 8 show the contours learned by the back-propagation and the nearest neighbor models. From these figures it is clear that the back propagation model gives smoother contours making a better fit to the points lying on the true boundaries.

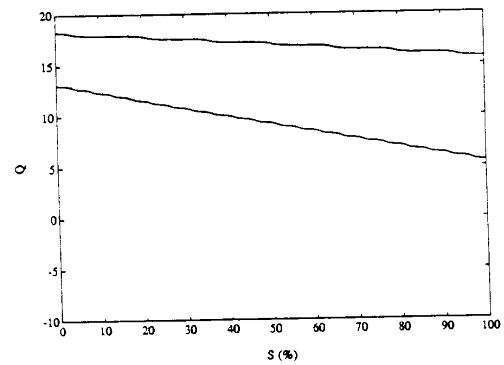

Figure 9. Back-Propagation trained on 1469 points

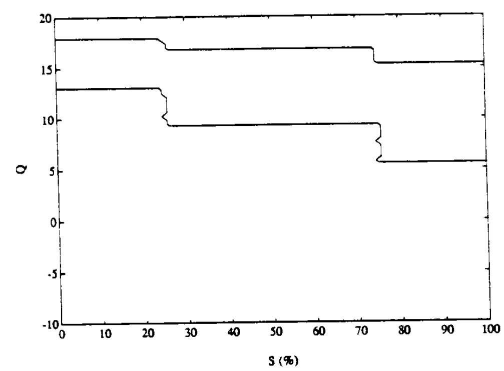

Figure 10. Nearest-Nneighbor trained on 1469 points

Figures 9 and 10 show the classification boundaries for Q as a function of S when P is is fixed at 15 MW. The region between the curves is the predicted secure area. While the training took place only at discrete values of S, the back-propagation technique interpolated much more smoothly than the nearest neighbor classifier.

Summary of Time to Train and Classify

The simulations for both classifiers were done on a Balance Sequent Parallel Computer. The results plotted below are estimates of neural network performance in terms of speed. The metric of CPU seconds refers to time on one of the Sequent CPUs and is loosely indicative of time expected for a dedicated classifier,

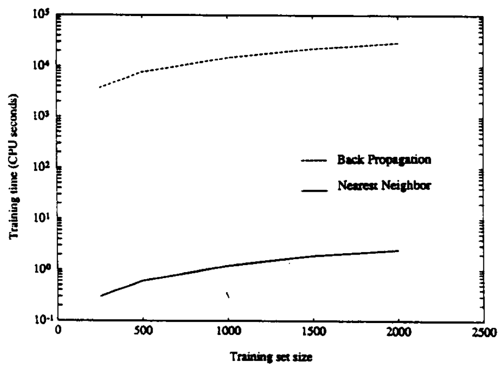

Figure 11. Training set size vs. training time

In Figure 11, the dashed line represents results for the back-propagation network. The training time is about 4 orders of magnitude higher than the nearest-neighbor classifiers. We expect that the long training time may be the most significant challenge to applying the neural network classifiers to large-scale power systems.

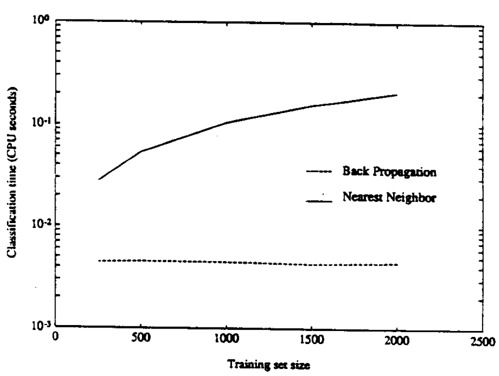

Figure 12. Training set size vs. classification time

The most important performance measure may be the recall time of the classifier. As shown in Figure 12, this time grows linearly for the nearest neighbor classifier, yet is constant and mush shorter for the neural network.

Conclusions

We have demonstrated that an artificial neural network (ANN) can potentially be a significant useful tool for static security assessment of power systems. We have shown that ANN's perform significantly better than a nearest neighbor search in terms of classification, recall time, and data storage requirements.

The ANN, however, requires a great deal of time for off-line training. This problem will be compounded as the system size increases. Learning complexity theory may be applied to better understand this scaling problem. Alterations which may lead to better performance include accelerated learning algorithms and the use of oracle based learning.

Acknowledgement

This work was supported by the National Science Foundation and the Washington Technology Center. The authors wish to express their thanks and gratitude for the valuable help provided by J.J.Choi and D.C.Park.

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] R.Fishl, T.F.Halpin, and A.Guvenis, 'The application of decision theory to contengency selection,' **IEEE Trans. on Circuits and Systems**, Vol.29, pp.712-723, Nov.1982.

[3] K.F.Cheung, L.E.Atlas, and R.J.Marks II, 'Synchronous versus asynchronous behavior of Hopfield's CAM neural net,' **Applied Optics**, Vol.26, pp.4809-4813, 1987.

[4] R.A.Jacobs, 'Increased rates of convergence through learning rate adaptation,' **Neural networks**, Vol.1, pp.295-308, 1988.

[5] H.H.Happ, 'Optimal power dispatch-a comprehensive survey,' **IEEE trans. on PAS.**, Vol.96, pp.841-854, May/June 1977.

[6] H.W.Dommel and W.F.Tinney, 'Optimal power flow solutions,' **IEEE trans. on PAS.**, Vol.87, pp.1866-1876, Oct.1968.

[7] J.L.McClelland, D.E.Rumelhart and the PDP Research Group, **'Parallel Distributed Processing: Exploitations of the Microstructure of Cognition,'** Vol.II, Bradford Books, Cambridge, MA, 1986.

(217) 333-4789

Further information can be obtained from Dr. W. Kenneth Jenkins.