VINCENT G. SIGILLITO, SIMON P. WING, LARRIE V. HUTTON, and KILE B. BAKER

CLASSIFICATION OF RADAR RETURNS FROM THE IONOSPHERE USING NEURAL NETWORKS

In ionospheric research, we must classify radar returns from the ionosphere as either suitable for further analysis or not. This time-consuming task has typically required human intervention. We tested several different feedforward neural networks to investigate the effects of network type (single-layer versus multilayer) and number of hidden nodes upon performance. As expected, the multilayer feedforward networks (MLFN's) outperformed the single-layer networks, achieving 100% accuracy on the training set and up to 98% accuracy on the testing set. Comparable figures for the single-layer networks were 94.5% and 92%, respectively. When measures of sensitivity, specificity, and proportion of variance accounted for by the model are considered, the superiority of the MLFN's over the single-layer networks is even more striking.

INTRODUCTION

Neural networks have many potential applications in signal processing. For example, Lippmann,[1] in an introduction to computing with neural networks, describes several applications to signal processing. Gorman and Sejnowski[2] showed that neural networks can be used to discriminate with high precision between sonar signals from a mine and from a similarly shaped rock. Lapedes and Farber[3] demonstrated the use of neural networks to predict points in highly chaotic time series. Boone, Sigillito, and Shaber[4] showed that neural networks can perform as well as trained human experts in detecting certain nodules in radiological data. Other examples of using neural networks to perform signal-processing tasks can be found in the special section on neural networks in the July 1988 issue of the *IEEE Transactions on Acoustics, Speech, and Signal Processing.*[5]

We describe here the application of neural networks to a radar classification problem that normally would require human intervention. Networks were trained to discriminate "good" from "bad" radar returns from the ionosphere. Individual networks achieved an accuracy of 98% when presented with data not used in the training set. We continue with a brief characterization of the radar system and how the signals were processed, describe the networks used for the discrimination tasks, and discuss implementation issues. We then present the results of our experiments with several different networks and an analysis of the classification strategy developed by the networks.

THE RADAR SYSTEM

The radar data used in our study were collected by the Space Physics Group of The Johns Hopkins University Applied Physics Laboratory. The radar system, located in Goose Bay, Labrador, consists of a phased array of 16 high-frequency antennas, with a total transmitted power on the order of 6.4 kW and an antenna gain of about 30 dBm at frequency ranges of 8 to 20 MHz. The radar returns are used to study the physics of the ionosphere at the E- and F-layers (100- to 500-km altitude). The targets, free electrons in the ionosphere, have small cross sections on the order of 10^{-30} m^2. A typical number density of electrons would be on the order of $10^8/\text{m}^3$, and the total volume could be as large as 10^6 m^3. Additionally, the backscattering process is coherent, and the backscattered signal is therefore proportional to the square of the number density. The usual signal-to-noise ratio is in the 10- to 20-dB range, but it can be as large as 50 dB. A detailed analysis of the backscattering process can be found in Ref. 6.

We now give a simplified version of the operation of the radar system. Our discussion does not precisely describe the radar operation at Goose Bay, but captures the essential features needed to understand the input to the neural network. (See Ref. 7 for a detailed description of the system and Ref. 8 for a description of the data analysis procedures.)

The radar operates by transmitting a multipulse pattern to the ionosphere. The receiver is turned on between pulses, and the target velocity is determined by measuring the phase shift of the returns.

If we denote the received signal from the pulse at time t by

$$C(t) = A(t) + iB(t) \ ,$$

then the autocorrelation function (ACF), R, is given by

$$R(t,k) = \sum_{i=0}^{16} C(t + iT)C^*[t + (i + k)T] \ ,$$

where T is the pulse repetition period, k indicates the pulse number, and the * indicates complex conjugation.

For the Goose Bay radar, *k* lies between 0 and 16. From the ACF a number of physical parameters can be derived, most importantly, the Doppler velocity of the target. For a target moving with constant velocity, the phase of the ACF will show a shift proportional to the lag, *k*.

Figure 1 shows typical ACF's received by the radar. The two parts of the curve, real (black) and imaginary (blue), correspond to the complex ACF that results from the complex electromagnetic signal. The radar at its current operating mode produces 25 of these 17 pairs of numbers every 5 s all year round. Thus, much work is required (now performed in part manually) to weed out bad ACF's during radar data analysis. It is this task to which we applied neural networks. The 17 pairs of numbers, representing 17 discrete values of the real part and the corresponding 17 values of the complex part of an ACF, are the input to the neural networks.

METHODS

The networks we used are known as feedforward networks, which comprise an input layer of identical processing units (neurons), an intermediate or hidden layer, and an output layer. All units in any given layer are connected to all units in the layer above. There are no other connections. Input units do not perform computations, but serve only to distribute the input to the computing units in the hidden layer above. Units in the hidden layer have no direct connections to the outside world, but after processing their inputs, pass their results to the units of the output layer. In a process called training, the network is given selected input examples and the corresponding desired output response (or target). The connection weights are changed, using a learning algorithm called back propagation, until the output error is minimized in a least-squares sense. The procedure is discussed more fully in the article by Sigillito elsewhere in this issue.

After training has been completed, the network's performance is evaluated on a different set of ACF's. This set, which has no data in common with the training set, is called the testing set. For both sets, the "gold standard" classification of each ACF was made by Simon P. Wing, who is expert in this classification task.

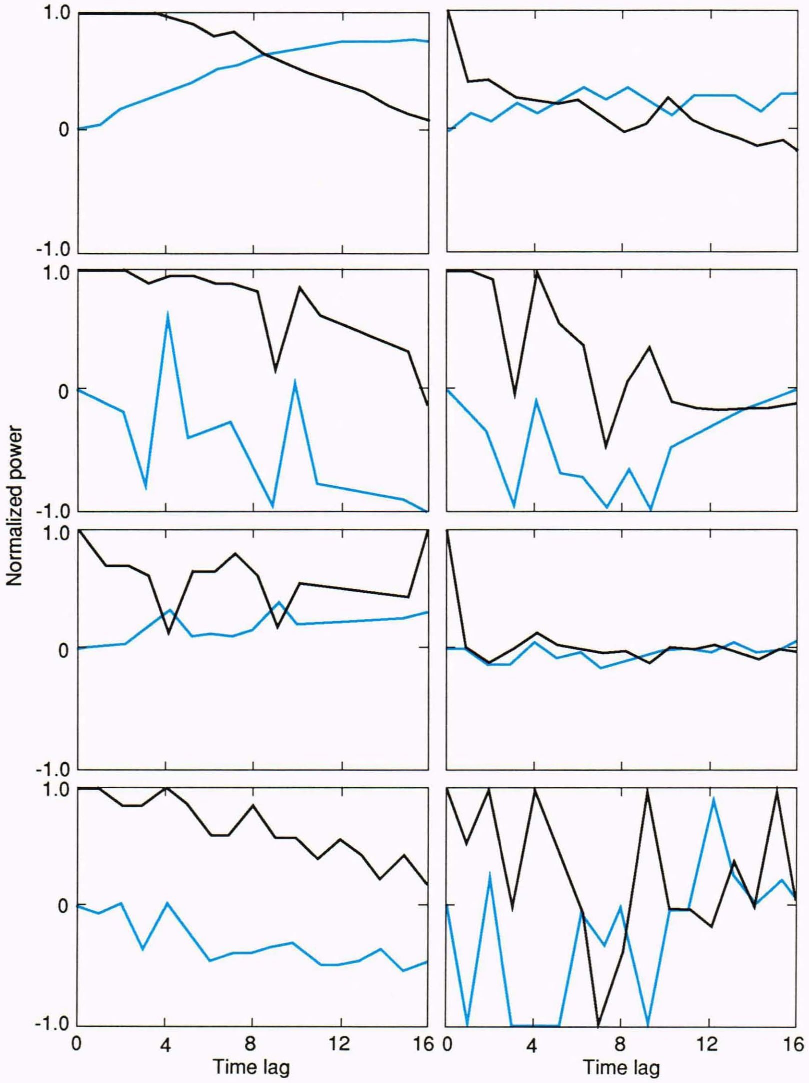


Figure 1. Typical ACF's. The four pairs of curves in the first column are good returns, the four pairs in the second column are bad returns. The black curves represent the real part, the blue curves the imaginary part.

IMPLEMENTATION

The ACF of a radar return is described by the 17 discrete returns, as discussed previously. Since each discrete return is composed of a real and an imaginary part, 34 values per ACF result. These 34 values serve as input to the network. Each input was normalized to the range [−1, 1]. The number of hidden nodes was varied from 0 (no hidden layer) to 15. Since our networks are currently used to classify the inputs into only two classes (good and bad), only one output node was needed. This node outputs a 1 for a good return and a 0 for a bad return. In general, good returns are indicated by well-defined signals, which are evidence of the presence of some type of structure in the ionosphere. Bad returns can be caused by the absence of identifiable structure (the signal passes through the ionosphere); by incoherent scattering (signals are reflected from too many structures, resulting in phase cancellation); by the absorption of radar pulses; and by interference from other transmitters. Bad returns are more diverse than good ones. That difference is reflected in the network's behavior, which we now discuss.

RESULTS AND DISCUSSION

We trained the networks with a training set of 200 returns (101 good, 99 bad). Networks having 0, 3, 5, 8, 10, and 15 hidden nodes were used. We refer to networks with no hidden units as perceptrons and those with hidden units as multilayer feedforward networks (MLFN's). It is well known that MLFN's can learn more complex mappings than can perceptrons.[9] We used the perceptrons to give us a basis for quantifying the additional power obtained by having hidden nodes in this problem. Note that if the output node of the perceptron simply outputs its input, then the output error to be minimized in the training process is

$$\frac{1}{2}\sum_{p=1}^{n_p}\left(T^p - \sum_{j=1}^{n_i+1} w_{1j}O_j{}^p\right)^2 , \qquad (1)$$

where T^p is the target (here, the correct classification, i.e., good or bad) associated with the pth input vector; $O_j{}^p$ is the output of the jth input unit when the pth input is clamped to the input layer; w_{1j} is the strength of the connection between the jth input unit and the output unit; n_i is the number of input nodes; and n_p is the number of training input/target pairs. (For more details of this process, see the article by Sigillito elsewhere in this issue.)

Equation 1 is similar to that which is minimized when a linear regression is applied to the training set, except that the regression coefficients are now found by an iterative steepest-descent method (i.e., back propagation) rather than by inverting a correlation matrix.

Figure 2 shows learning curves on the training set for two perceptrons and for a typical MLFN with five hidden nodes. The perceptrons differed only in that one used a linear transformation for its output function (identity function) and the other used the sigmoid transformation $(1/1+e^{-x})$ for its output function. All MLFN's used this sigmoid transformation. The learning curves began at values of 38% to 58% correct, and all moved to above 80% correct after 25 presentations of the training set. All nearly reached their final values by 100 presentations. The black curve in Figure 2 represents the linear perceptron, which eventually converged to 87.5% correct. The blue line represents the nonlinear perceptron, which eventually converged to 94.5% correct. The red curve represents an MLFN with five hidden nodes; it, and the other MLFN's used in this study, eventually converged to 99.5% to 100% correct. It is clear that the MLFN's are superior to the perceptrons in learning the classification task.

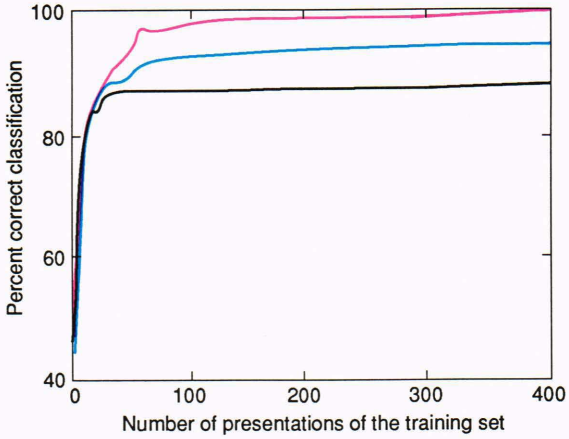


Figure 2. Network learning curves for the linear perceptron (black), the nonlinear perceptron (blue), and a typical MLFN with five hidden nodes (red).

The superiority of the MLFN's over the two perceptrons becomes more apparent when they are each tested against data not in the training set. This new set, the testing set, was composed of 150 returns, of which 123 were good and 27 were bad. (Bad returns were much less common in the data than were good returns.) The linear perceptron was able to correctly classify 90.67% from the testing set; the nonlinear perceptron, 92%. The MLFN's averaged greater than 96% correct, with a range from 94% to 98%. (Figure 3 shows the worst case; the best case; the average over 10 different starting networks for 3, 5, 8, 10, and 15 hidden-node MLFN's; and a 1-standard-deviation band around the average.)

Further analysis showed clear differences in sensitivity and specificity of the various network types. Sensitivity is a measure of accurate detection of a good return when a good return was in fact present. The sensitivity of the linear perceptron was 95.9% (it correctly classified 118 of 123 good returns); that for the nonlinear perceptron was 98.4% (121 of 123); and that for the best MLFN's was 100%. Specificity is a measure of how well the networks correctly classify bad returns. The specificity of the linear perceptron was only 66.7% (it correctly classified 18 of 27 bad returns); that for the nonlinear perceptron was 63% (17 of 27); and that for the best MLFN's was 88.9% (24 of 27). The worst MLFN's

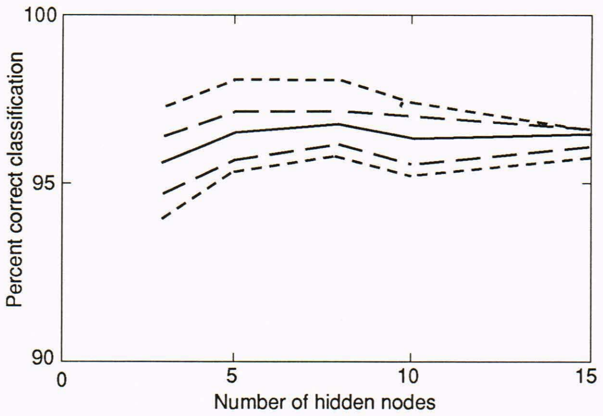


Figure 3. Percent correct classification of MLFN's on the training set as a function of the number of hidden nodes. The middle curve is an average of results for 10 MLFN's with different initial weights. The dashed lines on either side of the average are 1-standard-deviation bands. The dotted curves indicate the best and worst performance of the ten networks for 3, 5, 8, 10, and 15 hidden nodes.

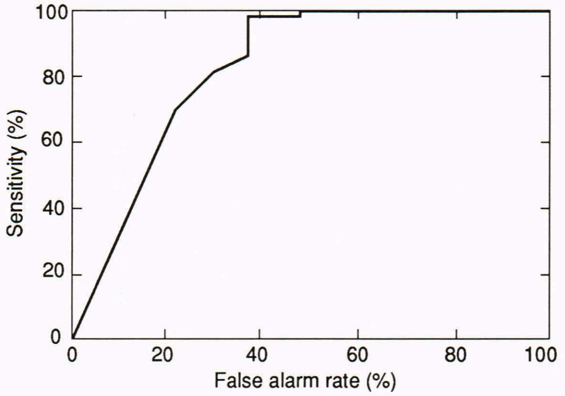


Figure 4. Sensitivity versus false alarm rate for a linear perceptron. False alarm rate is the probability of predicting a good return when a good return is not present. Sensitivity is the probability of predicting a good return when a good return is present.

had a sensitivity of 100% and a specificity of 66.7%. Thus, the worst MLFN did as well as the best perceptron.

The results are captured in the receiver operating characteristics (ROC) curves of Figures 4 and 5. Each curve shows the hit rate (sensitivity) as a function of the false alarm rate (1 minus the specificity). The ROC curve of the MLFN is far closer to that of a perfect discriminator than is that of the perceptron. This conclusion is elaborated in Figures 6 and 7, in which sensitivity, specificity, proportion of variance accounted for, and percent correct are shown as functions of the good/bad threshold* for both the linear perceptron and the best MLFN. For a threshold of 0.5, the MLFN accounted for 83.8% of the output variance; the perceptron accounted for only 49.1%.

*Inputs that produce outputs less than the threshold are classified as bad; those that produce outputs that are equal to or exceed the threshold are classified as good.

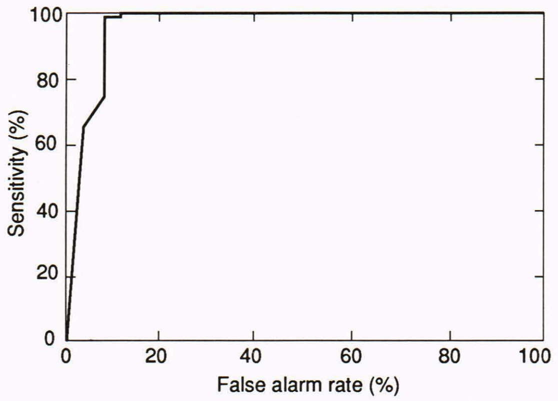


Figure 5. Sensitivity versus false alarm rate for the best MLFN. False alarm rate is the probability of predicting a good return when a good return is not present. Sensitivity is the probability of predicting a good return when a good return is present.

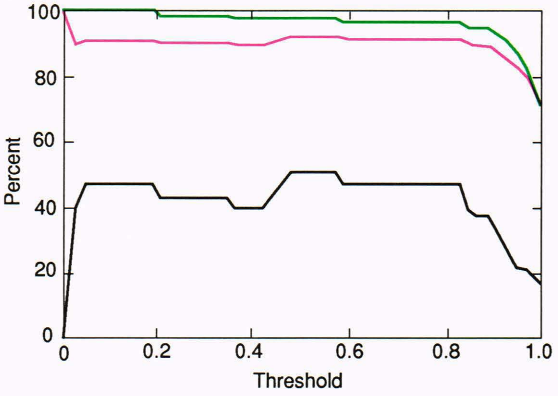


Figure 6. Sensitivity (green), specificity (blue), proportion of all cases correct (red), and proportion of variance in output accounted for by the network (black) as functions of the good/bad threshold. These results are for the linear perceptron.

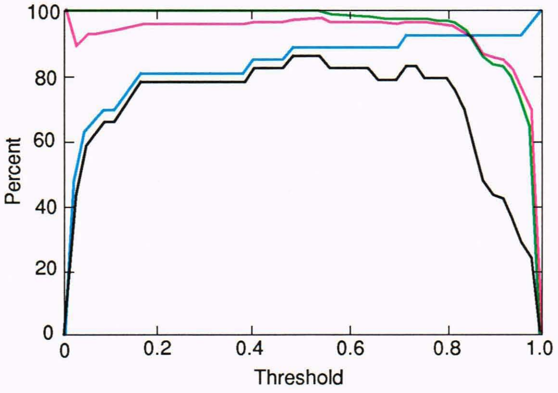


Figure 7. Sensitivity (green), specificity (blue), proportion of all cases correct (red), and proportion of variance in output accounted for by the network (black) as functions of the good/bad threshold. These results are for the best MLFN.

CONCLUSIONS

We have demonstrated that classification of radar returns is a task for which neural networks are very well suited. Further, neural networks with hidden nodes substantially outperformed those without hidden nodes. The improvement in performance extended to both sensitivity and specificity measures; MLFN's outperformed perceptrons, and perceptrons performed as well as a multiple linear regression analysis in their ability to discriminate between good and bad returns.

The original goal of our research was to demonstrate that neural networks would operate at a level of performance high enough to be a real aid in the automation of the classification task. That goal was clearly met. It would, however, be useful to ascertain whether a neural network could also be used to determine the cause of a bad return (e.g., absence of identifiable structure, incoherent scattering, absorption of radar pulses, interference from the transmitters). On the basis of our experience we believe that MLFN's should be useful in such an error analysis.

REFERENCES

[1] Lippmann, R. D., "An Introduction to Computing with Neural Nets," *IEEE ASSP Mag.*, 4–22 (Apr 1987).

[2] Gorman, R. P., and Sejnowski, T. J., "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets," *Neural Net.* **1**, 75–89 (1988).

[3] Lapedes, A., and Farber, R., *Nonlinear Signal Processing Using Neural Networks: Prediction and System Modeling*, LA-UR-87-2662, Los Alamos National Laboratory (1987).

[4] Boone, J. M., Sigillito, V. G., and Shaber, G. S., "Signal Detection Capabilities of Neural Networks: Radiology Applications," *Med. Phys.* (in press).

[5] "Special Section on Neural Networks," *IEEE Trans. Acoust. Speech Signal Process.* **36**, 1107–1190 (1988).

[6] Walker, A. D. M., Greenwald, R. A., and Baker, K. B., "Determination of the Fluctuation Level of Ionospheric Irregularities from Radar Backscatter Measurements," *Radio Sci.* **22**, 689–705 (1987).

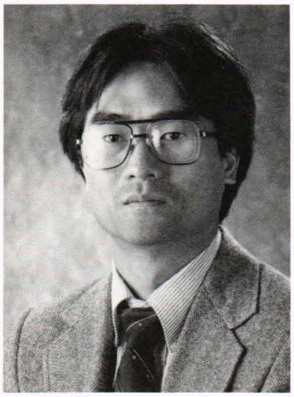
[7] Greenwald, R. A., Baker, K. B., Hutchins, R. A., and Hanuise, C., "An HF Phased Array Radar for Studying Small-Scale Structure in the High Latitude Ionosphere," *Radio Sci.* **20**, 63–79 (1985).

[8] Baker, K. B., Greenwald, R. A., Villain, J. P., and Wing, S., *Spectral Characteristics of High Frequency Backscatter from High Latitude Ionospheric Irregularities: A Statistical Survey*, Technical Report to Rome Air Development Center, RADC-TR-87-284 (Mar 1987).

[9] Rumelhart, D. E., and McClelland, J. L., eds., *Parallel Distributed Processing, Vol. 1*, MIT Press, Cambridge, Mass. (1986).

ACKNOWLEDGMENT—We would like to thank Ray A. Greenwald for providing the database that was used to train and test the neural networks.

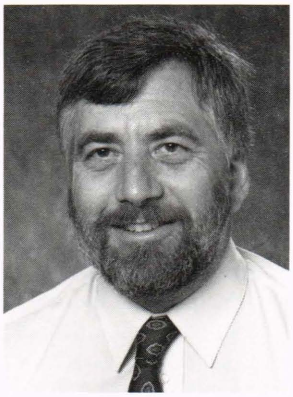
THE AUTHORS



SIMON P. WING was born in 1963. He received a B.S. degree in physics from the University of Arizona in 1984 and an M.S. degree in computer science from The Johns Hopkins University in 1988, where he is now a doctoral student in computer science. From 1984 to 1985, Mr. Wing was with GTE Telenet, where he worked on the PABX project. In 1985 he joined APL's Space Physics Group as an employee of Sachs Freeman Associates, Inc., contractor to APL. Mr. Wing has done extensive software development for the APL high-frequency radar. Recently, he has been applying neural network techniques to DMSP satellite data.

KILE B. BAKER was born in 1950 in Bozeman, Mont. He received a B.S. degree in physics from Montana State University in 1971 and a Ph.D. in astrophysics from Stanford University in 1978. From 1979 to 1981 he was with the Astronomy Department of Boston University, where he did research on the solar corona and taught astronomy and space physics. His research has included work on the magnetospheres of pulsars, the Sun, and the Earth. Since 1981 Dr. Baker has been a member of APL's Space Physics Group, where he has worked on problems relating to the Earth's ionosphere and its interaction with the magnetosphere and the solar wind.

VINCENT G. SIGILLITO's biography can be found on p. 261.



LARRIE V. HUTTON was born in 1946 in Potsdam, N.Y. He received his B.S. degree from Michigan State University in 1969, and his Ph.D. degree in experimental psychology from Ohio University in 1979. From 1970 to 1972, he served in the Army as a psychology specialist in the Department of Neuropsychiatry at Ft. Knox, Ky. From 1980 to 1987, Dr. Hutton was an assistant professor of computer science and psychology at Marietta College in Ohio. In 1988, he came to APL, where he is on the senior technical staff of the Mathematics and Information Science Group of the Milton S. Eisenhower Research Center. He teaches graduate and undergraduate courses on artificial neural networks, both at APL and at the Homewood campus of The Johns Hopkins University. Dr. Hutton's major research interests are in the development of cognitive control systems and in connectionist models of choice behavior. He belongs to IEEE, ACM, APA, Sigma Xi, the Psychonomic Society, and the International Neural Network Society.