7

A NEURAL NETWORK METHOD FOR HIGH RANGE RESOLUTION TARGET CLASSIFICATION

R. G. Atkins, R. T. Shin, and J. A. Kong

7.1 Introduction

Classification or identification of radar targets from measurements of their radar signatures continues to be an area of considerable interest and active research. In the past, a variety of classification algorithms have been proposed [1-16], and these techniques have yielded varied levels of effectiveness in differing applications. For practical discrimination, some type of measurement diversity demonstrating variation across classes must be exploited.

Polarization diverse measurements are utilized by Kong et. al. [1] who employ a quadratic distance measure with a feature vector consisting of horizontal, vertical, and cross-polarized measurements, to identify terrain cover. This classifier is shown to be optimal in minimizing the Bayes risk under the assumption of Gaussian statistics. Swartz [2] introduces a polarimetric matched filter to provide optimal contrast for discriminating between two ground terrain classes, and ap-

plies this method to the processing of polarimetric synthetic aperture radar (SAR) images. Lim et. al. [3] employ both supervised and unsupervised classification techniques with polarimetric SAR data, and Holm [4] uses polarimetric measurements to distinguish between the response of stationary targets and ground clutter.

Additional measurement diversity is exploited by other classification schemes which utilize multi-frequency measurements or transient target signatures. Mains and Moffatt [5] describe a method of identifying targets using their resonant characteristics, a technique further developed by Kennaugh [6] who describes a K-pulse method, and by Chen et. al. [7,8] and Rothwell et. al. [9,10] who present similar E-pulse and S-pulse techniques. These methods are shown to obtain discrimination by correlating the late-time response with an angularly independent waveform unique to each target, which results in either a zero or single mode response for the correct class. Other algorithms utilize the entire pulse transient or high range resolution target profile to provide a range image of the scattering centers of the target. Chamberlain [11] combines frequency and polarization diversity to obtain a time-domain polarization signature or transient polarization response, and uses this data with nearest neighbor or syntactic methods to classify aircraft targets.

Most of the applications presented in the past utilize one of a collection of conventional classifiers, each of which has been shown to possess its own limitations. Profile matching, perhaps one of the most well known and widely used methods for classifying targets from high range resolution profiles, relies upon correlating the unknown signature with a library of measured signatures for each target. This matched filter approach is in theory optimal for classifying known profiles in statistically independent Gaussian noise [12]. In practice, however, even in the absence of noise sources, the unknown profile is unlikely to exactly match any recorded in the library, since profiles can only be stored for a finite number of aspects. In addition, multi-path and target-clutter interactions may lead to noise sources which are non-Gaussian and correlated with the free-space target signature. For these reasons, profile matching can often generate sub-optimal results. With a large number of target classes, profile matching has the added practical constraint of requiring prohibitive processing and storage. Sometimes this computational burden may be lessened somewhat when approximate aspect information is available, as is the case with many airborne targets, and

some reduction in storage can also be achieved by averaging profiles for several adjacent angles, and storing only the averaged profiles. K-pulse [6] and other similar methods [7–10] eliminate both storage and computational burdens by using an aspect independent response, however, since these methods utilize the late-time, small amplitude signature, they are generally more susceptible to noise.

Parametric or statistical methods, which extract relevant features of the profiles and store only the estimated first or second order statistical information for these features [13], escape the need to store the entire set of profiles and to process these for each unknown. In the simplest case, the feature vector extracted is taken to be the sampled range profile itself, and its mean and covariance matrix are determined by averaging over all aspects. In other cases, more elaborate features such as target length or positions of peaks in the profile have been suggested as alternative features. The Bayes classifier [1,3,13,14] may be used to optimally reduce classification error once assumptions have been made regarding the underlying distributions of the feature vectors. Alternately, Novak [14,15] shows that the linear Fisher classifier is only slightly sub-optimal to the Bayes classifier, and may perform better in practice where the estimated statistics are not known exactly. In addition, distance methods such as the Euclidean distance or Mahalanobis distance classifiers [13] can be employed without making any explicit assumptions toward the distribution. Implicit in these techniques, however, is the assumption that the distribution of the feature vector is adequately described by its first and second order statistics. For Gaussian-like distributions this assumption is not severe, however, in the general case where the distributions are multi-modal, an attempt to model with only first and second order statistics may discard useful information, and lead to poor accuracy. More complex methods such as Gaussian mixtures or radial basis function classifiers [16] provide some improvement by allowing formation of more arbitrary decision regions, but these methods do not escape the inherent limitation of most conventional techniques, which each assume a specific fixed distance metric in the space of the input feature vector.

One novel group of classification techniques which overcome this limitation, and which has recently received considerable attention, is a set of methods based on the use of neural networks [17–26]. As classifiers, neural networks have shown greater flexibility than conventional algorithms, adapting themselves more easily to distributions possess-

ing a high degree of complexity, and have found application in diverse problems such as speech recognition [18–21,28] and multi-sensor fusion [17]. Presented a set of sample feature vectors during the training process, the neural network tries to selectively and automatically choose those characteristics of the feature vector which lead to the greatest separability between classes.

For a number of applications, neural networks have been compared to conventional classifiers, and have shown an accuracy equivalent to or exceeding the previously used methods. Huang and Lippmann [18] compare the performance of neural network classifiers to a quadratic Gaussian classifier, and show that even on problems with Gaussian distributions, the neural network approach performs similarly. When outliers are present, the Bayes classifier fails, while the robustness of the neural network allows it to discount the outliers and perform well. Lippmann [19], Huang [20] and Waibel et. al. [21] both apply neural networks in speech recognition problems, and show that the network classifiers can sometimes outperform conventional methods. Gorman and Sejnowski [22] apply multi-layer perceptron networks to the problem of classifying sonar targets from wide-band pulse signatures, and obtain performance with a three layer network which exceeds the accuracy of both nearest neighbor classifiers, and that of trained human listeners. Similar networks are used by Sigillito et. al. [23] to classify radar returns from the ionosphere. Decatur [24,25] favorably compares neural networks to Bayesian classifiers in the problem of identifying ground terrain from polarimetric SAR data, and Farhat [26] uses hetero-associative neural networks to identify aircraft from sinogram images.

The neural networks used in the above applications can be viewed as filters which are both non-linear and adaptive. As shown in the flow graph of Fig. 7.1, a set of inputs $\{x_1, x_2, \cdots, x_n\}$ are applied to the network either continuously or at discrete times, and from these are derived a set of outputs $\{y_1, y_2, \cdots, y_m\}$. The processing is much like that of a simple linear flow graph, except that the subset of possible functional blocks is extended to include non-linear elements, and the coefficient weights on the paths between nodes are allowed to evolve with time. Unlike a linear network, where each node is simply a summing junction which adds a number of weighted inputs, the node appearing in the neural network differs in the addition of a non-linear element at the output of the sum, as show in Fig. 7.2. Although the

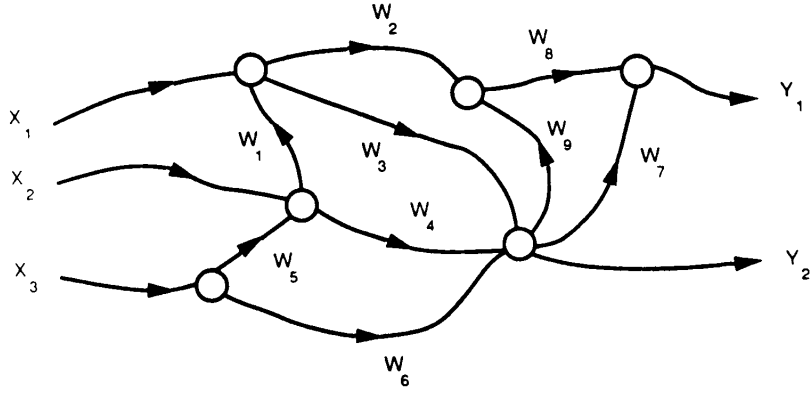

Figure 7.1 Flow graph representation of an arbitrary neural network structure, with inputs X_1, X_2, and X_3, and outputs Y_1 and Y_2.

type of non-linearity which is employed varies with the implementation of the network and its intended purpose, two types are in common use. The threshold or step non-linearity assumes a value of one for inputs above a specific level, and is zero for smaller inputs. In contrast, the sigmoid non-linearity changes continuously from zero to one, providing a continuously differentiable function needed for training algorithms such as Back Propagation [29], and a localization effect helpful in obtaining better performance with noisy inputs.

A large number of different types of neural networks exist, and these are often grouped according to the geometry of the network, the function to be performed, and the training algorithm used to adapt the network weights. One specific type which has been shown useful in classification problems is the Multi-Layer Perceptron [27], an example of which is shown in Fig. 7.3. This network consists of several layers of nodes, connected in a strictly flow-forward manner to permit propagation from left to right without feedback. Any connection geometry which preserves this one directional flow is permitted, and the number of layers and number of nodes in each layer can be chosen arbitrarily. Although four or more layers could be used, three has been shown to provide the full generality in allowing completely arbitrary decision regions for classification [28].

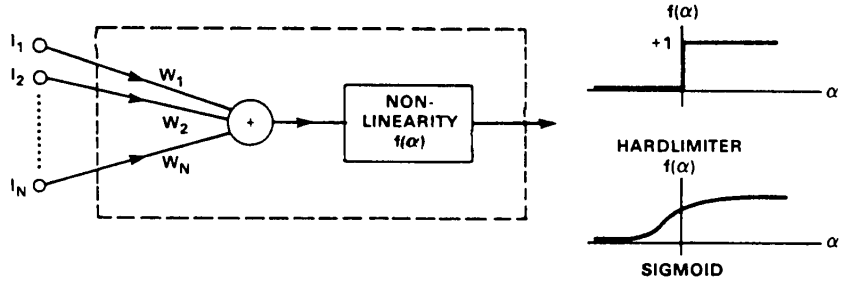

Figure 7.2 Individual node structure within the neural network. The sum of weighted inputs is passed through a non-linearity of either the hard-limiting or sigmoid types.

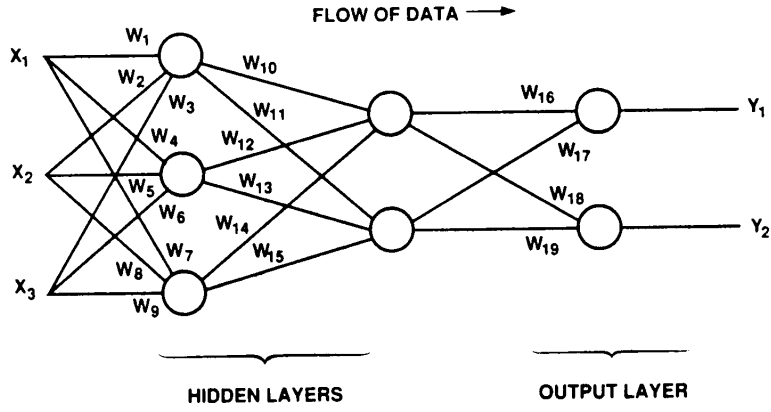

Figure 7.3 Structure of a Multi-Layer Perceptron neural network consisting of two hidden layers and one output layer, with three inputs and two outputs.

The method in which this type of neural network is applied consists of two stages. In the first, the network is trained by adapting its

weights to perform the function for which it is desired. This training is performed by minimizing the difference between the desired output for a set of inputs, and the actual output, obtained as a function of the connection weights. Back Propagation [29] adapts the weights by successively applying a set of training vectors as input to the system, calculating the actual output with the present set of weights, and forming an error term at the output. This error is propagated back through the network, adapting the weight values using a gradient descent technique to reduce the output error. This process is repeated for each input vector in the training set, and then repeated for the entire training set until convergence is attained. After this first training stage, the network is used to perform the desired function. The connection weights remain fixed, and inputs are applied and processed through the network from the input layer through successive hidden layers to the output.

The method by which this procedure can be applied in a simple classification problem is shown in Fig. 7.4. The problem presented is a two class example where the observed vector from which classification will be performed consists of two features, X_1 and X_2. The observed distribution of these features for the two classes are shown with '+' for class one, and 'o' for class two. The network used to classify feature vectors for this simple problem is the single node structure of Fig. 7.5. Assuming a threshold non-linearity is employed, the decision boundary in the space of the features will be linear, since the output of the node will be one for features such that,

$$A\,X_1 + B\,X_2 + \gamma > 0 \tag{1}$$

By choosing appropriate values of the constants A and B, and for the threshold, γ, this decision boundary can be established to separate the two classes, and provide a mechanism for identifying a given feature vector. In this manner, the neural network can be used to realize the structure of a simple Gaussian Maximum Likelihood (ML) classifier for the case of identical covariances across classes [19].

The advantage of the neural network approach becomes clearer for the more complicated example shown in Fig. 7.6, where the two classes can no longer be separated by a single linear decision boundary. For this situation, a more complex network, such as that of Fig. 7.7, must be employed. Each of the nodes in the bottom layer of this network functions in an analogous manner to that of the simpler problem above,

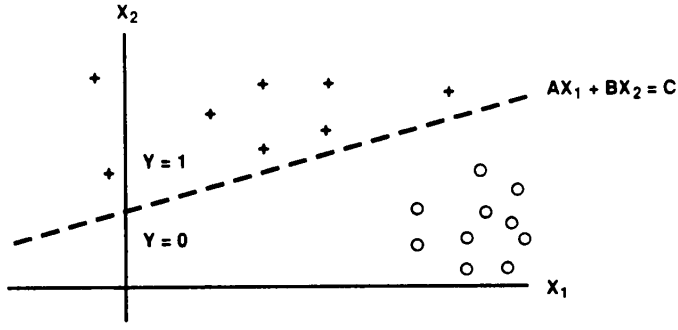

Figure 7.4 Simple two-class, two-feature classification problem where class 1 is represented by '+' and class 2 by 'o', and where a linear decision boundary can be produced between the two using a single node Perceptron.

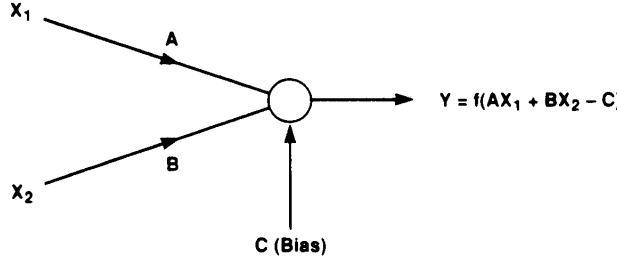

Figure 7.5 Single node Perceptron with two inputs, X_1 and X_2, corresponding to the two features. This classifier creates a linear decision boundary in the space of the inputs.

with each forming one of the six linear boundaries shown in Fig. 7.6. From these boundaries, the two disjoint regions, R_1 and R_2, can be formed by the second layer of the network, which with the appropriate weights, performs a logical 'AND' operation on the results output from

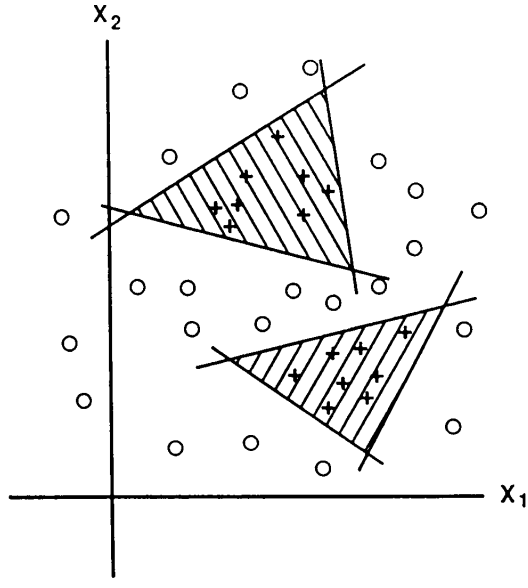

Figure 7.6 Complex two-class, two-feature classification problem where the first class is clustered in two disjoint regions.

the first layer. Hence, the output of either of the second layer nodes will be one only if the input vector lies in the corresponding region. Finally, the last layer performs a logical 'OR' operation such that the final output of the network is one for feature vectors in either of the two shaded regions, and is zero for all other pairs of inputs.

With this method and with the appropriate number of nodes in each layer, it is possible to synthesize arbitrarily complex decision boundaries which may form multiple disjoint regions. This technique is therefore better adapted to handle multi-modal distributions which are not easily modeled as Gaussian, and for which Gaussian ML or distance classifiers do not generally perform well. In addition, the Back Propagation training algorithm provides a simple method for choosing the connection weights of the network, and automatically generating decision boundaries.

This chapter considers the application of neural networks to the problem of target classification from high range resolution profiles. The effectiveness of the neural network classifier is demonstrated us-

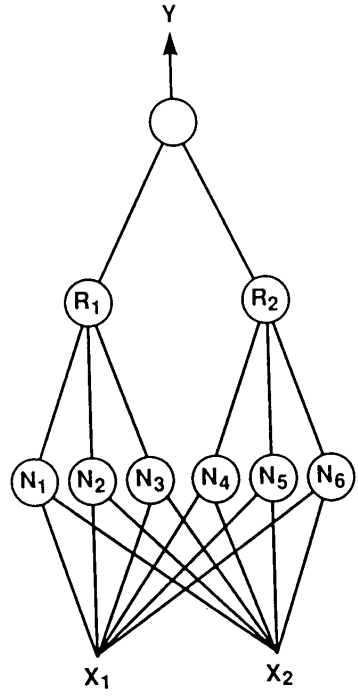

Figure 7.7 Multi-Layer Perceptron classifier for the two-feature, two-class problem with two disjoint regions. The first layer produces linear boundaries, the second forms the two regions, and the last merges the two regions.

ing synthetically generated range profiles of two groups of geometries, produced using RCS prediction techniques. For both groups, the neural network approach is compared with the conventional techniques of profile matching, and Euclidean and Mahalanobis distance classifiers. In addition, the performance of both conventional and neural network classifiers in the presence of additive noise and alignment uncertainty is explored. Finally, a comparison of the computational and storage requirements of each approach is presented.

7.2 Neural Network and Conventional Classifier Results

The two sets of targets which are used to test the neural network and other classifiers are pictured in Figs. 7.8 and 7.9. The three targets shown in Fig. 7.8 (Targets 1–3) are all canonical geometries for which radar cross section predictions can easily be performed, and all are assumed to be composed of perfect electrical conductors. The first target is a circular cylinder, 1 meter long and 1 meter in diameter. The second is a longer, more slender cylinder, 3 meters long and 0.5 meters in diameter, and the third is a rectangular box, 2 meters long, and 1 meter in both height and width, constructed from six flat plates. Although this first group consists of simple, unrealistic structures, the targets provide geometries with which the classification errors may be more easily interpreted.

The second set (Targets 4–6) provides, in contrast, a group of targets more realistically modeling vehicles of potential interest in practical classification applications. The first target is an aircraft fighter model, the second is a delta-wing aircraft, and the third is a cruise missile replica. The first two targets are approximately the same size with lengths of 15–16 meters and wing-spans of 8–10 meters, while the final target is considerably smaller with a length of only 5.6 meters and a wing-span of 2.5 meters. As with the above targets, this second group is modeled using simple canonical shapes such as plates, cylinders, and cones.

The radar cross section prediction method which was used to generate the synthetic range profiles is the Physical Theory of Diffraction (PTD) [30]. The PTD predictions include both the Physical Optics term accounting for surface reflections, and the diffraction term accounting for edge scattering. Coherent fields were predicted for each geometrical component composing the targets, and these were summed to determine the overall response. Since PTD treats scattering as a local phenomenon, all effects such as multiple scattering and creeping waves are neglected, and consequently, the time delays of scattered returns in the transient response correspond directly to the positions of the scatterers in range. Hence, the maximum time extent of the pulse response will be equal to the two-way propagation delay along the longest target dimension, and the late-time response from multiple interactions will not be predicted.

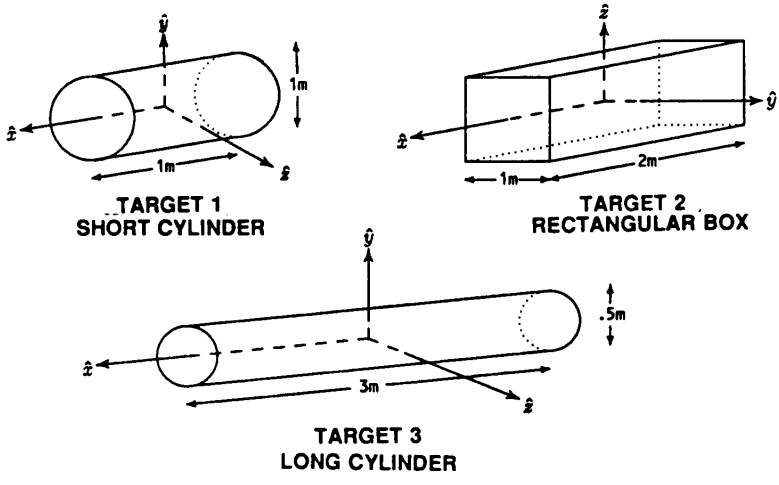

Figure 7.8 Simple canonical targets composing the first set of test objects. Target 1 is a cylinder 1 m long and 1 m in diameter, Target 2 is a cylinder 3 m long and .5 m in diameter, and Target 3 is a rectangular box 1 m×1 m×2 m.

In order to develop high range resolution profiles [31] for these targets, radar cross section predictions were made with stepped, single frequency illumination for 360 aspects spaced 1° apart around the targets. For the simple geometries, predictions were done in the xy-plane, using a horizontal polarization in which the electric field lies in the measurement plane. To obtain the range information, the frequency used in the predictions was stepped over 32 discrete values spaced 40 MHz apart in a band from 1.6 to 2.84 GHz. The samples in frequency for a given aspect were then windowed with a Hamming window, and this 32 point frequency response was transformed via an FFT algorithm. The squared magnitude of the resulting complex envelope was taken to provide a 32 point range profile of the target corresponding to samples of the returned power every .117 meters over a total range interval of 3.75 meters.

For the realistic targets a similar approach was used, but the aspect cuts were taken with a depression angle of 7° below the wing plane of

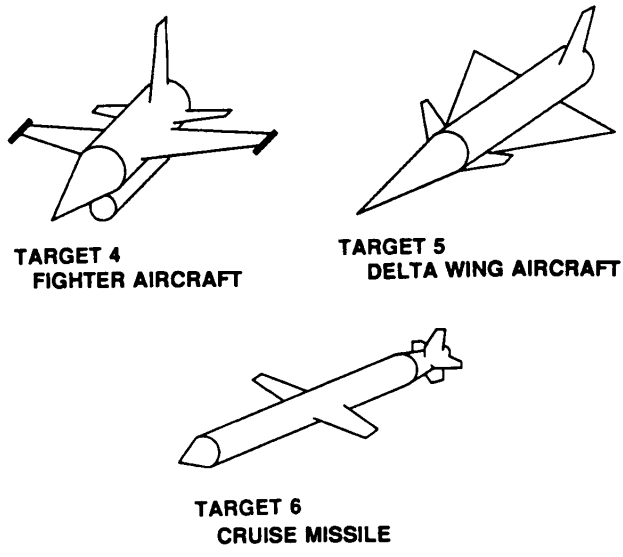

Figure 7.9 Realistic targets composing the second set of test objects. Target 1 and Target 2 have lengths of 15–16 m and wing spans of 8–10 m. Target 3 has a length of 5.6 m and a wingspan of 2.5 m.

the targets. In addition, at each aspect, the predictions are repeated for 64 frequencies, 8 MHz apart, in a band from 1.744 to 2.248 GHz, and the results are windowed and transformed to obtain a 64 point range profile.

Profiles generated for several of the targets from the simple and realistic groups are shown in Figs. 7.10 and 7.11 respectively. The first profile (top) of Fig. 7.10 is that of the shorter cylinder (Target 1) when illuminated normal to the front circular endcap ($\phi = 0°$). The two peaks arising in this signature correspond to a large specular return from the front surface, and a diffraction return from the rear edge. The second profile (bottom) is done for the rectangular box (Target 3) at an incidence angle 45° from the lengthwise axis through the center of the box. This signature shows three peaks in the return which can be seen to arise from the three forward edges of the box. The final

corner is shadowed at this incidence direction, and the PTD algorithm predicts no return from this hidden corner. The resolution of both profiles is poor, however, this deficiency is compensated for by the simplicity of the targets and their signatures, and has little effect on the recognition of the targets for most aspects, since there are few closely spaced features to resolve.

Figure 7.11 shows a sample profile of the fighter target (Target 4) for the nose on aspect. From the dashed outline of this target, the scattering centers responsible for each peak in the return can be determined. It is clear that scattering arises from the front and rear ends of the engine, from the tip of the wing mounted missiles, and from the corners of the wing and tail surfaces.

In order to provide for both training and testing of the neural network and other classifiers, the 360 profiles generated for each target were split into two groups. The odd angle profiles were used for training, and the interspersed even angle profiles for testing. In addition, to avoid the practical problems which might arise with calibration of a realistic measurement system, the profiles are all normalized to have a correlation coefficient of one. All of the networks employed were initialized with random weights, and trained with the Back Propagation algorithm. To overcome the effects of the random initial weights, the training of each network was repeated several times with different initializations, and the results given below are those averaged over these individual trials.

For the first group containing the simplistic targets (Targets 1-3) several network sizes were tested, and the results of two such sizes are presented here. The simplest network utilized is a single-layer perceptron, consisting of three nodes corresponding to the three targets, where each node is fully connected to all 32 inputs. The second network is a larger topology with three layers of nodes, 12 in the first hidden layer, 6 in the second, and three in the output as before. All inputs or nodes are fully connected to all nodes in the next most immediate layer, but no connections are made between non-adjacent layers.

For comparison purposes, three types of conventional classifiers were also utilized. In the profile matching scheme, each of the signatures in the library (training data) is correlated with the unknown profile, and the hypothesized class is taken as that for which the largest correlation coefficient is found. Since the profiles have been normalized, and since the effects of time misalignment in the received profile

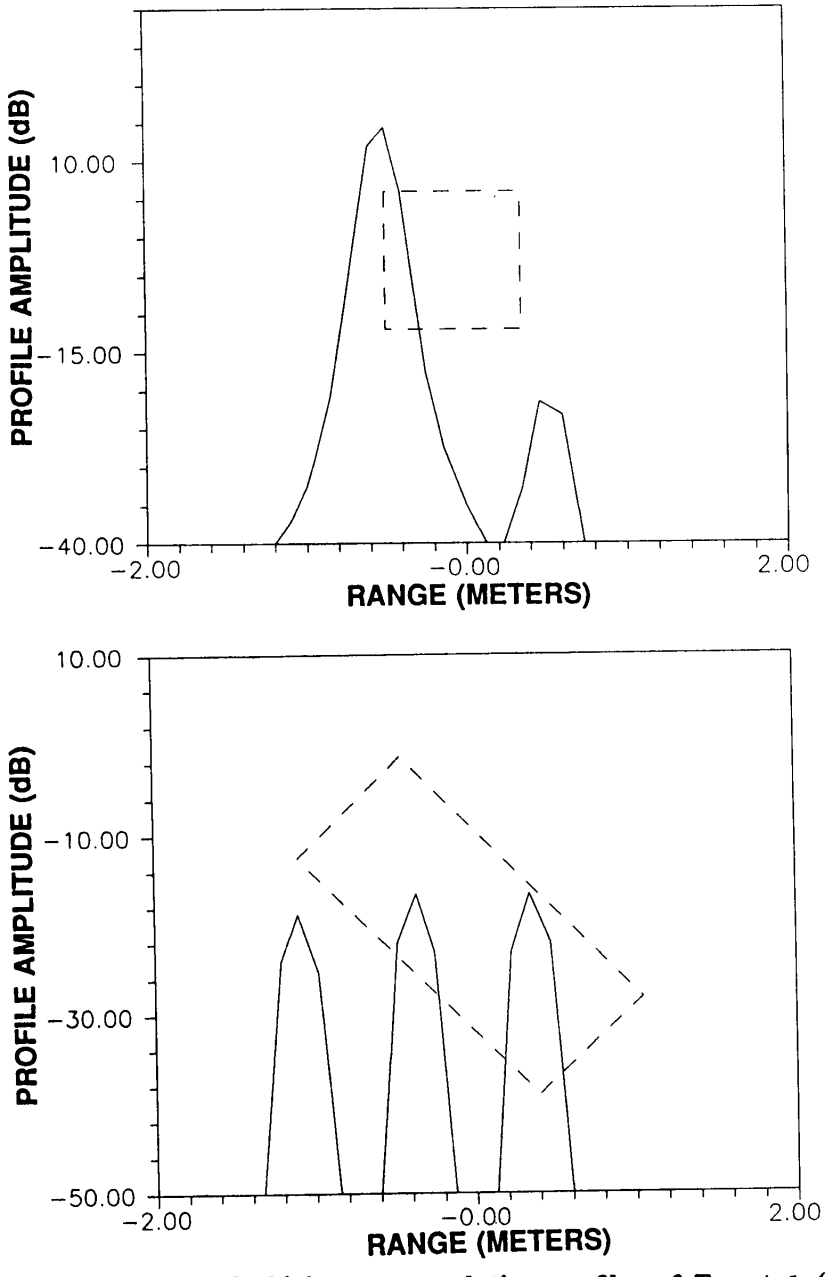

Figure 7.10 Sample high range resolution profiles of Target 1 (top) illuminated at end-on incidence, and of Target 3 (bottom) illuminated at an angle of 45° from its lengthwise axis.

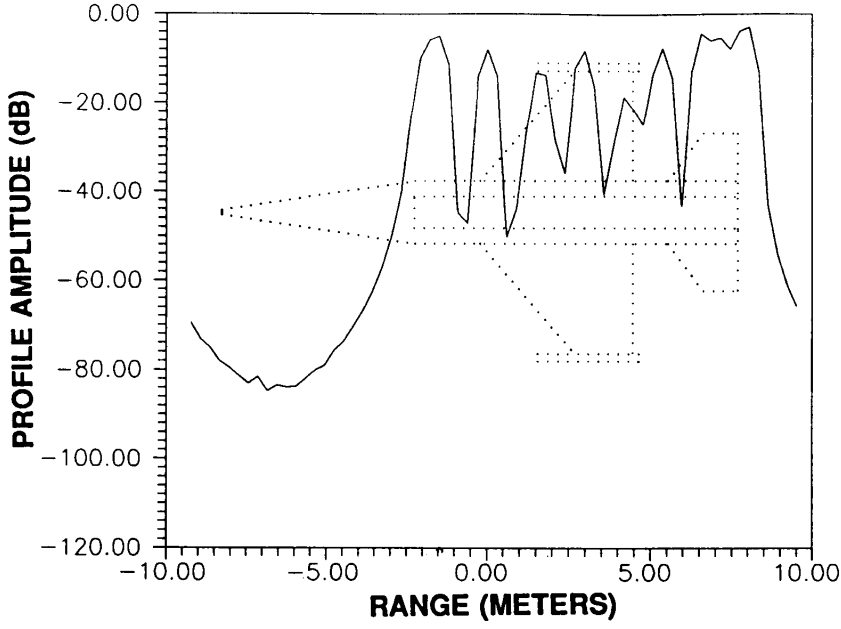

Figure 7.11 High range resolution profile of Target 4 for the case of nose on incidence.

are not currently being considered, the correlation coefficient is simply computed as the inner product of the two profile vectors as given by (2).

$$\rho = X_{library}^{T} Y_{unknown} \tag{2}$$

Two distance classifiers were employed using the Euclidean and Mahalanobis distance metrics given by (3) and (4) respectively.

$$d_{Euclidean}^{2} = (Y - X_i)^T (Y - X_i) \tag{3}$$

$$d_{Mahalanobis}^{2} = (Y - X_i)^T \Lambda_{X_i}^{-1} (Y - X_i) \tag{4}$$

The first requires only the mean vector, X_i, for each class, which was computed by averaging over all training profiles for each target. The second metric calculates a statistical distance weighted by the covariance matrix, which is again estimated from the training data.

The results of the two neural network classifier approaches are given in Tables 7.1 and 7.2. The overall accuracy with the smaller, three node network was found to be 93.6%, and the larger network showed some improvement, increasing the accuracy to 97%. Further increasing the size of the network, however, was found to display no additional increase in classification accuracy. For the smaller network, the confusion matrix of Table 7.1 shows that Target 1, the short cylinder, was classified correctly at all times, while the error was caused by sometimes mistaking Targets 2 and 3 for the first target. In contrast, for the larger network, it is Target 2, the longer cylinder, which is always classified correctly, and Targets 1 and 3 are sometimes confused for one another. This later result is the more expected distribution of error, since Target 2 is longer and more slender than either of the other targets, and, hence, more distinguishable at all aspects. In contrast, it is likely that Targets 1 and 3 will be confused part of the time, since they both have the same widths, and their broadside profiles are very similar. Thus, the larger network experiences error where it should logically occur, indicating that the error is due to the likeness of the profiles, and not to the incapability of the classifier to effectively separate them. The smaller network, however, not only unexpectedly classifies Target 1 correctly for all test profiles, but mistakes Targets 2 and 3 for Target 1 without ever mistaking Target 1 for either of the other two. This pattern of error shows a favoring of Target 1 by the classifier, an asymmetry which should not exist if the classification error is truly due to an indistinguishability between profiles. Instead, the error is forced by the simplicity of the network which does not allow formation of sufficiently complex decision boundaries to classify the profiles in as optimal a fashion as possible.

Similar trends are shown in the results of the conventional classifiers. Profile matching was found to have an overall accuracy of 96.3%, very close to that of the larger neural network. In addition, the confusion matrix given in Table 7.3 shows that the error distribution is the same as that of the larger network, with Target 2 classified correctly, and Targets 1 and 3 sometimes confused. In contrast, the Euclidean distance classifier behaves much more like the simple network, although its overall accuracy of 71.8% is significantly lower. The confusion matrix of Table 7.4 shows that the Euclidean distance method classifies Target 1 with 100% accuracy, confusing Targets 2 and 3 for Target 1,

Target	Classification		
	Target 1	Target 2	Target 3
Target 1 (short cylinder)	1.000	.000	.000
Target 2 (long cylinder)	.104	.896	.000
Target 3 (rectangular box)	.089	.000	.911

Table 7.1 Confusion Matrix for Simple Targets Classified by the 3-node Perceptron.

Target	Classification		
	Target 1	Target 2	Target 3
Target 1 (short cylinder)	.978	.000	.022
Target 2 (long cylinder)	.000	1.000	.000
Target 3 (rectangular box)	.067	.000	.933

Table 7.2 Confusion Matrix for Simple Targets Classified by the 21-node Multi-Layer Perceptron.

and Target 3 for 2. The reason for the successful classification of the first target stems from the fact that of all the targets it is the most symmetric in aspect, and, hence, its profiles with varying aspect differ less from the mean profile than do the other two classes. The test profiles for Target 1 all lay clustered close to the mean, and all are, thus, closer to their own mean than to any for another class. Again, as with the simple neural network classifier, the accuracy of the classification is not limited by the the data, but by limitations of the classifier itself.

The Mahalanobis distance classifier provides a considerable improvement in overall accuracy over the Euclidean metric, obtaining a classification rate of 95.8%. The confusion matrix of Table 7.5 reveals that the distribution of error is a compromise between that of

Target	Classification		
	Target 1	Target 2	Target 3
Target 1 (short cylinder)	.956	.000	.044
Target 2 (long cylinder)	.000	1.000	.000
Target 3 (rectangular box)	.067	.000	.933

Table 7.3 Confusion Matrix for Simple Targets Classified by Profile Matching.

Target	Classification		
	Target 1	Target 2	Target 3
Target 1 (short cylinder)	1.000	.000	.000
Target 2 (long cylinder)	.222	.511	.267
Target 3 (rectangular box)	.356	.000	.644

Table 7.4 Confusion Matrix for Simple Targets Classified with the Euclidean Distance Metric.

the simpler, Euclidean metric and perceptron network classifiers, and that of the more complex profile matching and multi-layer network classifiers. Although Target 1 is no longer classified 100% correctly, Target 2 is identified with significantly greater accuracy than with the Euclidean metric, and there is less confusion of Target 3 for Target 1. By employing covariance information, the Mahalanobis metric is able to overcome the problem of the Euclidean classifier in which Target 1 is heavily favored. The overall high accuracy of the Mahalanobis metric shows that the assumption of singly-peaked distributions is not too limiting for this data set, although some improvement is still possible by relaxing this assumption, as seen in the neural network results.

Target	Classification		
	Target 1	Target 2	Target 3
Target 1 (short cylinder)	.978	.000	.022
Target 2 (long cylinder)	.022	.956	.022
Target 3 (rectangular box)	.061	.000	.939

Table 7.5 Confusion Matrix for Simple Targets Classified with the Mahalanobis Distance Metric.

It is apparent from these results that it is necessary to match the complexity of the classifier to the complexity of the distribution of feature vectors. Both the Euclidean distance classifier and the smaller perceptron network are not sufficiently complex to take full advantage of the information contained in the training profiles, and their structure is too limiting. Despite this limitation, however, and the similarity between the structure of the smaller network and that of a simple Gaussian Maximum Likelihood classifier, the perceptron classifier shows a surprisingly high accuracy. This result demonstrates the advantage of the neural network approach which is not limited to a Gaussian assumption, but free to place its decision boundaries more arbitrarily. The neural network training allows placement of the decision boundaries in positions which they might only occur in ML schemes if non-gaussian distributions were assumed.

A number of different size networks were also tested with the group of more realistic targets, and the overall accuracies achieved are shown in Fig. 7.12. A single layer perceptron, as well as several two and three layer models are compared. Classification accuracy is seen to initially increase as the network is made larger, but this effect saturates at a size of approximately 27 nodes, and the accuracy fluctuates after this point. A decrease in accuracy for larger networks is possible because too many nodes may lead to overly specific definitions of decision regions, which do not generalize properly to classify the testing data.

Table 7.6 shows the confusion matrix for the three-layer, 27 node network containing 16 nodes in the first hidden layer, 8 in the second, and 3 in the output layer, and which achieves an accuracy of 86.8%.

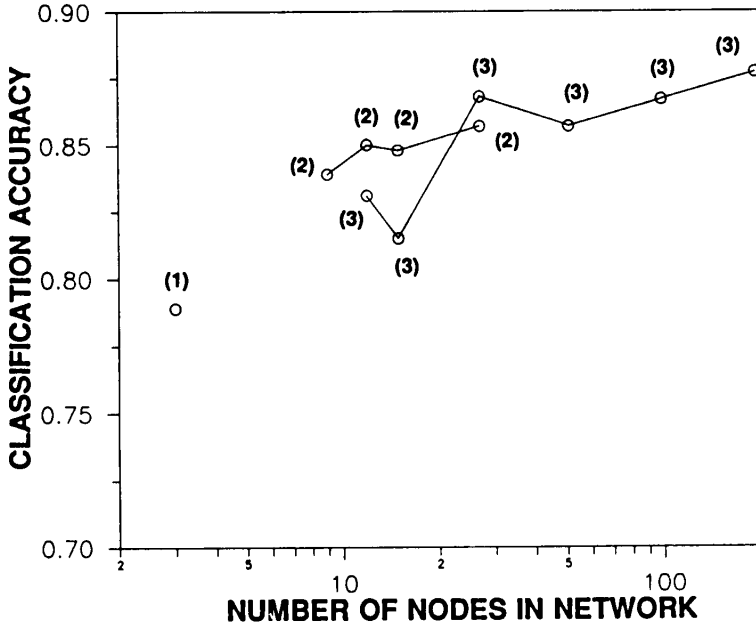

Figure 7.12 Accuracy of the neural network classification for the realistic targets as a function of network size. The number of layers to each network is shown in parentheses.

Most of the error arises in misclassification of Targets 4 and 5, which are both classified with only 82% accuracy. In contrast, Target 6, the cruise missile model is much shorter than the other two, and this characteristic allows it to be discriminated more easily, with 97% accuracy.

Profile matching when applied to the set of realistic targets yields a 78.4% accuracy overall, which is slightly lower than that exhibited with the neural network approach. This comparison indicates that the neural network classifier is better able to generalize from the training data, where the profile matching approach experiences inaccuracies due simply to the differences between the profiles used for training and the interspersed profiles used for testing. Hence, the neural network approach is likely to suffer less than profile matching when the training

Target	Classification		
	Target 4	Target 5	Target 6
Target 4 (Fighter Aircraft)	.816	.153	.031
Target 5 (Delta-wing Aircraft)	.080	.818	.102
Target 6 (Cruise Missile)	.004	.024	.971

Table 7.6 Confusion Matrix for Realistic Targets Classified by the 27-node Multi-Layer Perceptron.

Target	Classification		
	Target 4	Target 5	Target 6
Target 4 (Fighter Aircraft)	.789	.144	.067
Target 5 (Delta-wing Aircraft)	.144	.656	.200
Target 6 (Cruise Missile)	.011	.081	.908

Table 7.7 Confusion Matrix for Realistic Targets Classified by Profile Matching.

profile azimuths are moved further apart. The distribution of error for profile matching, as seen in Table 7.7, is, however, very similar to that of the neural network, with Target 6 classified with relatively high accuracy, and a greater error arising between Targets 4 and 5.

The accuracies of both Euclidean and Mahalanobis distance classifiers were found to be lower than profile matching, with rates of 65.2% and 63.8% respectively. Though the overall accuracies of these two metrics are similar, Tables 7.8 and 7.9 reveal that the individual target accuracies are very different. Both classify Target 4 with approximately the same accuracy, but the Euclidean metric makes large errors with Target 5 and classifies Target 6 with 100% accuracy, while the Maha-

Target	Classification		
	Target 4	Target 5	Target 6
Target 4 (Fighter Aircraft)	.656	.089	.256
Target 5 (Delta-wing Aircraft)	.378	.300	.322
Target 6 (Cruise Missile)	.000	.000	1.000

Table 7.8 Confusion Matrix for Realistic Targets Classified with the Euclidean Distance Metric.

Target	Classification		
	Target 4	Target 5	Target 6
Target 4 (Fighter Aircraft)	.700	.300	.000
Target 5 (Delta-wing Aircraft)	.141	.836	.023
Target 6 (Cruise Missile)	.096	.526	.379

Table 7.9 Confusion Matrix for Realistic Targets Classified with the Mahalanobis Distance Metric.

lanobis metric makes large errors on Target 6. The high accuracy of the Euclidean metric with the cruise missile target is expected, since like the short cylinder of the first data set, this target is the most symmetric for all aspects, and, hence, the most tightly clustered about its mean. The loss of accuracy for Target 6 when changing to the Mahalanobis metric suggests that the distribution for Target 5 has a much greater spread than that of Target 6, and that much or all of the cluster for Target 6 is contained in a region statistically close to the mean profile for Target 5.

7.3 Noise Performance

One practical issue which is not addressed by the above analysis concerns the performance of each of the algorithms when corrupting noise is present in the profile to be classified. The above results were obtained with clean training and testing profiles, and represent the limiting case of infinite signal-to-noise (S/N) ratio. For many applications, such as the discrimination of ground targets obscured by sources presenting large clutter returns, it is necessary that accuracy be maintained despite profile distortion from additive noise.

To provide a comparison of the performance of each algorithm with noisy measurements, the methods used with the realistic targets above were reapplied after artificially adding noise to the test profiles. The noise added was of Gaussian amplitude and uniform phase, and was uncorrelated between range cells. The S/N ratio was defined as the ratio of the average power in each uncorrupted range cell to the average added noise power. Testing of the algorithms was performed at varying S/N ratios, and the results are shown in Fig. 7.13.

At high S/N levels the neural network classifier outperforms the conventional methods, as shown previously. Profile matching is next most accurate, outperforming each of the two distance classifiers, which both have approximately the same accuracy. As more noise is added to the test profiles, however, the accuracy of the Mahalanobis distance classifier begins to drop rapidly. This decline arises from the fact that the classifier is still trained with clean data, and the effect of the noise on the covariance matrix of the observed feature vector is not accounted for. This case corresponds to a situation where the noise level is unknown and where an estimate of the noise is either unavailable or un-utilized. As a result, a small amount of added noise may produce a large statistical deviation from the mean profile, as measured using the statistics of the clean training set. This occurrence is particularly noticeable for the cruise missile target, which with its small length is expected to have little return in the range cells at either end of the profile. As shown in Table 7.10, a small amount of added noise (S/N = 30 dB) causes a large statistical distance for class 6, and results in this target being mistaken for the other two.

A similar decrease in accuracy with increasing noise is experienced by the neural network classifier. For S/N ratios above 20 dB, the neural network performs well, but a sharp decrease begins at this noise level,

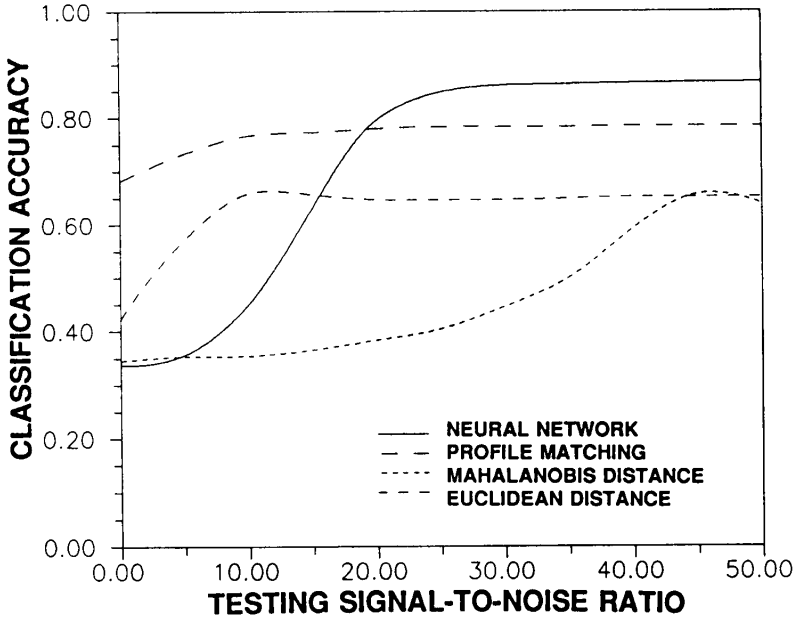

Figure 7.13 Accuracy of classification methods vs. signal-to-noise ratio when applied to the realistic target set, and trained with noise-free data.

and for very low S/N levels (0 dB), the classifier does little better than random guessing. The reason for this failure is again the use of clean profiles during the training of the network. The neural network is trained to recognize uncorrupted feature vectors, and then utilized for the alternate task of recognizing profiles buried in noise. Since the network has experienced only uncorrupted profiles, however, it has not been trained to reject the noise. The errors which result in the neural network classifier are shown in Table 7.11 for a S/N ratio of 15 dB. As shown above for the Mahalanobis distance method, the worst error occurs with the cruise missile target, suggesting that the network classifier also relies highly on the short length and resulting invariance of the end portions of the profile with aspect, as important factors in its classification.

The overall excellent performance of the profile matching scheme

Target	Classification		
	Target 4	Target 5	Target 6
Target 4 (Fighter Aircraft)	.533	.467	.000
Target 5 (Delta-wing Aircraft)	.215	.782	.003
Target 6 (Cruise Missile)	.220	.761	.019

Table 7.10 Confusion Matrix for Realistic Targets Tested at S/N = 30 dB with Mahalanobis Distance Metric.

Target	Classification		
	Target 4	Target 5	Target 6
Target 4 (Fighter Aircraft)	.943	.056	.001
Target 5 (Delta-wing Aircraft)	.424	.560	.016
Target 6 (Cruise Missile)	.322	.288	.390

Table 7.11 Confusion Matrix for Realistic Targets Tested at S/N = 15 dB with the 27-node Multi-Layer Perceptron.

for all noise levels is not unexpected. Profile matching differs only from a matched filter in that the library profiles are not perfectly matched to the test profiles because of the aspect difference between them. At low S/N levels, however, the difference between training and testing profiles becomes insignificant in comparison to the added noise, and the accuracy of any classifier will be primarily limited by the noise and not the differences between training and testing sets. Hence, under these conditions, profile matching delivers near optimal accuracy. Finally, the insensitivity of the Euclidean metric to added noise is a result of the simplicity of the algorithm, as well as the fact it escapes incorrect assumptions which plague the Mahalanobis distance and neural network classifiers as discussed above.

The Mahalanobis distance technique can clearly be improved if an estimate of the noise covariance is included in the covariance matrix with which the statistical distances are calculated, preventing the classifier from becoming overly sensitive to noise. The result of this improved procedure is shown in Fig. 7.14. The Mahalanobis classifier was trained with several varying levels of noise added to the training data, and tested for S/N levels from 0 to 50 dB, as before. The accuracy of the classifier is seen to peak at varying noise levels depending on the amount of noise used in the training. Note, however, the accuracy does not always peak at the S/N level at which the classifier was trained but often at higher S/N levels. In addition, for testing S/N levels below 20 dB, it is always possible to obtain better performance by using a classifier trained with more noise than is actually present. This result is unexpected if the distributions underlying the clean target profiles are in fact Gaussian. Hence, this result suggests again that the distributions are more complex, and the mismatched noise level serves to de-emphasize the use of a statistical distance which is not suited to the actual distributions present. By assuming more noise, the resulting covariance matrix becomes more that of the noise alone, and in the limit, becomes a weighted identity matrix characteristic of the noise alone, and identical for each class. In this limit the Mahalanobis distance reduces to the Euclidean distance classifier, shown previously to perform better than the Mahalanobis metric. Hence, even by including the effects of the noise in the classifier, the overly constraining assumptions of the Mahalanobis scheme prevent it from performing as well as the simpler Euclidean metric.

This same type of improvement in performance is possible with the neural network classifier if the training of the network is performed with profiles which have been corrupted by additive noise. The same network used previously was re-trained several times, varying the amount of noise added to the training profiles for each trial. The results shown in Fig. 7.15 demonstrate a trend similar to that of the Mahalanobis distance classifier, in which the peak accuracy is obtained at a S/N level which varies according to the level with which the network was trained. Unlike the Mahalanobis classifier, however, the peak testing S/N ratio is always that at which the network was trained, where the network classifier is matched to the noise in the testing profiles. The neural network is, thus, capable of determining the level of noise in the training set, and adapting its classification procedure to expect

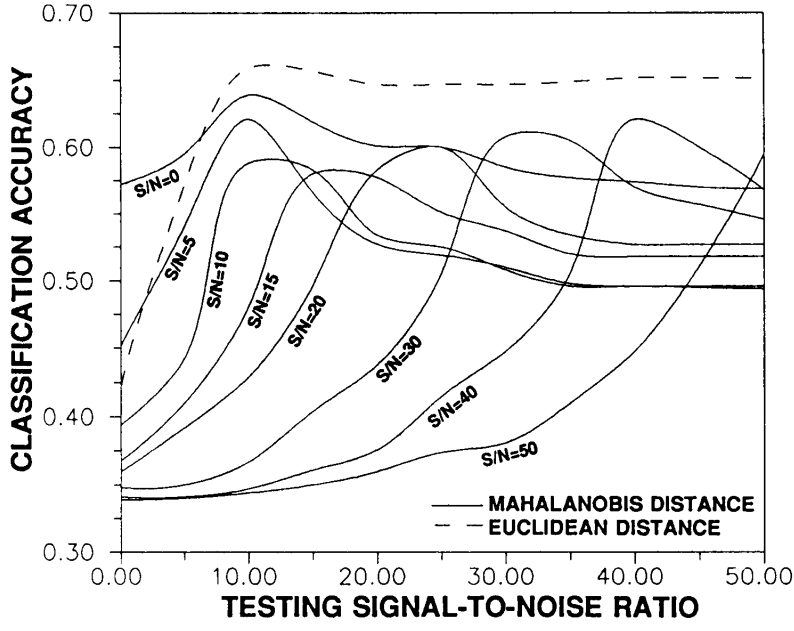

Figure 7.14 Accuracy of the Mahalanobis distance classifier (solid) vs. signal-to-noise ratio of the testing data when trained with noisy data. The signal-to-noise ratio of the training data is shown beside each curve. Also shown for comparison is the Euclidean distance result (dashed).

that level of noise. The neural network can use this information in an efficient fashion, and since no assumptions are made regarding the distributions of the target profiles, the network does not benefit from assuming more noise is present than what actually has been added. The resulting accuracies obtained at low S/N levels far exceed those of the network trained without additive noise, and are also better than those of the profile matching algorithm. This demonstrates the capability of the neural network in rejecting noise at least as well as the correlation method, while still maintaining the ability to generalize and more efficiently overcome the differences between training and testing profile sets.

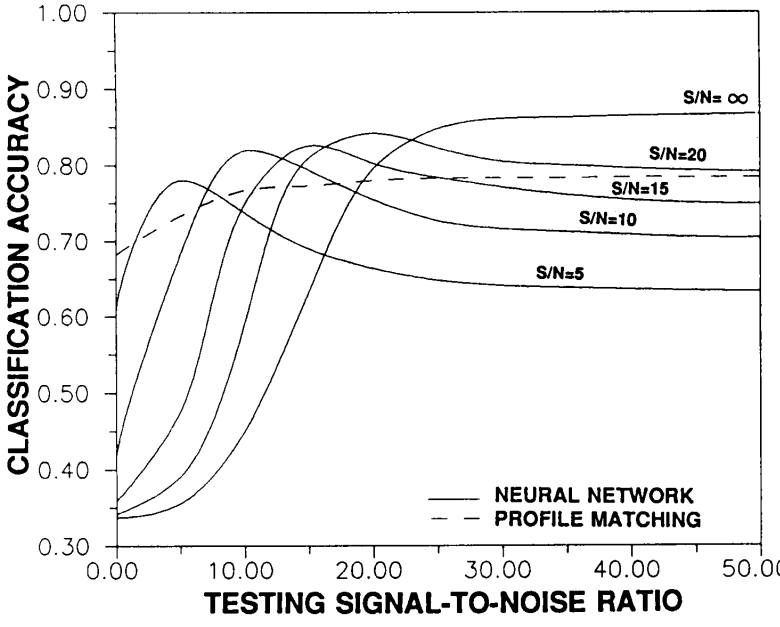

Figure 7.15 Accuracy of the neural network classifier (solid) vs. signal-to-noise ratio of the testing data when trained with noisy data. The signal-to-noise ratio of the training data is shown beside each curve. Also shown for comparison is the profile matching result (dashed).

7.4 Alignment Uncertainty

A second practical issue which complicates the process of identifying profiles in a realistic application is the problem of aligning a measured profile with those in the library. Given a measured transient response, it is necessary to extract from this the 64 cell range profile, determining the start and end of the profile within the overall response. With noise present in the measurements, it is unlikely that this can be accomplished without the introduction of some shifting of the profile. Although the correlation nature of the profile matching scheme allows

	Misalignment						
	-3	-2	-1	0	1	2	3
Accuracy	.469	.501	.641	.868	.652	.563	.511

Table 7.12 Accuracy of 27-node Multi-Layer Perceptron vs. Misalignment for Training without Shifting.

this error to be overcome by simply sliding the profile against the library exemplars, the neural network and distance classifier approaches expect the profile to be correctly aligned.

The accuracy of the neural network approach for misaligned profiles is given by Table 7.12. Each of the testing profiles for the realistic targets was presented to the 27 node network used previously in Section 7.2, but here the profiles have been shifted by -3 to +3 increments. The results obtained when it is falsely assumed that the profiles are correctly aligned show a significant decrease in accuracy even for shifts of only ± 1, and this accuracy is further reduced for shifts of ± 2 or ± 3. The overall accuracy assuming that any of the seven shift positions shown are equally likely, is reduced from 86.8% to merely 60.1%.

In contrast, the results of the profile matching scheme show far less decrease in accuracy when alignment uncertainty is introduced. To overcome this error, the profile to be classified is slid over a range of -3 to +3 with respect to the position in which it is assumed to be correctly aligned, and the maximum correlation coefficient for all library profiles and all shift positions is taken to indicate the correct class. This scheme is utilized with all test profiles, where an arbitrary shift between -3 and +3 is introduced before the sliding is performed. Thus, in all cases the correct alignment position is one of the 7 positions considered, but the other 6 positions will vary depending on the initially introduced alignment error. The results shown in Table 7.13 indicate that profile matching is relatively insensitive to the initial alignment error, and the overall accuracy of 74.7%, assuming equally probable likelihoods of each shift, is not significantly lower than the 78.5% accuracy obtained when the profiles are always aligned correctly.

It is possible that a sliding scheme similar to that used with the

	Misalignment						
	-3	-2	-1	0	1	2	3
Accuracy	.722	.741	.748	.752	.767	.756	.744

Table 7.13 Accuracy of Profile Matching with Sliding Correlation vs. Misalignment.

	Misalignment						
	-3	-2	-1	0	1	2	3
Accuracy	.770	.805	.821	.842	.856	.849	.785

Table 7.14 Accuracy of 27-node Multi-Layer Perceptron vs. Misalignment for Training with Shifting.

profile matching algorithm could provide some improvement to the neural network classifier. This approach, however, ignores the real problem effecting the accuracy of the network approach which parallels the problem encountered in the previous section when noise was introduced in the testing profiles. Because the network is trained with profiles of only correct alignment, the classification algorithm developed internally learns to expect only this case. The solution is, therefore, to train the network with profiles showing the entire range of misalignment which is expected to occur in later testing.

The above type of training was implemented using the same 27 node network considered previously, but now shifting each of the training profiles over the range of -3 to +3 cells. The results shown in Table 7.14 are significantly better than those shown for the previous network above. Some taper in accuracy is still seen for the larger shifts of ± 3, but the overall accuracy, again assuming all 7 positions are equally likely, is now 81.8%. Although this is a reduction from the 86.8% experienced when the profiles are correctly aligned, it is still

better than the profile matching result given above. Where before the network was trained to classify on the basis of values for fixed range cells, this result demonstrates the ability of the network to instead learn and utilize the relative positions of profile features.

7.5 Computational and Storage Requirements

The results presented above demonstrate that the accuracy of the neural network classifier is comparable to and often better than that of more conventional methods. In addition, however, the neural network technique has the added advantage of requiring less storage and computation than conventional techniques such as profile matching or the Mahalanobis distance method. Table 7.15 compares these costs for the implementations of the algorithms used previously to classify the three realistic target models.

Profile matching requires storage of 270 training profiles (taking advantage of the targets' symmetry), each of length 64, for a total requirement of 17,280 values. The Mahalanobis distance method requires storage of the 64×64 covariance matrix and mean profile for each class, resulting in an overall requirement of 6,432. The 27 node neural network used in the preceeding analysis requires storage of the interconnection weights and threshold values for each node. This network leads to a requirement of 1203 values, which represents a reduction by more than 5 times over the distance method, and by more than 14 times over profile matching. This comparison demonstrates the relative efficiency of the neural network in allocation of storage to obtain equivalent classification accuracy. While profile matching stores the entire set of profiles, and the distance classifiers store a prescribed set of statistics, the training process for the neural network allows an adaptive selection of the significant features for each target class, and is not limited by preconceived prejudices regarding what these features should be.

The comparison of the computational requirements for the three algorithms is similar to that for storage. Profile matching requires correlation of the unknown profile with each from the library, and results in 17,280 multiplies and 17,010 additions. The Mahalanobis distance measure requires vector addition, vector-matrix multiplication, and a vector inner product. These operations result in 12,480 multiplies and

Algorithm	Storage	Computation		
		Multiples	Adds	Non-linearities
Profile Matching	17,280	17,280	17,010	0
Mahalanobis Distance	6,432	12,480	12,477	0
Neural Network	1,203	1,176	1,176	27

Table 7.15 Comparison of Storage and Computational Requirements for Classifying the Realistic Targets.

12,477 additions for each unknown to be classified. Finally, the neural network requires multiplication by the connection weights, and the summing of terms at each node, resulting in 1176 multiplies and 1176 additions. Again this is a reduction of over one order of magnitude when compared to either profile matching or Mahalanobis distance methods. The neural network, however, has the added requirement of performing non-linear operations in each of the 27 nodes. In the case of threshold non-linearities, this operation is trivial, but for the sigmoid non-linearity used in the above examples, this calculation is more involved, requiring, as a minimum, a table look-up to compute the non-linear function.

7.6 Summary

This chapter has presented a novel technique for the classification of targets from high range resolution profiles using Multi-Layer Perceptron neural networks. The Physical Theory of Diffraction was employed to artificially generate high range resolution profiles for two sets of targets, and neural networks were developed to classify targets within each set. For comparison purposes, the more conventional profile matching, Mahalanobis distance, and Euclidean distance classifiers were also applied.

Training and testing with noise-free data, it was shown that the neural network achieved an accuracy exceeding that of conventional methods. In addition, for networks of sufficient complexity, the error is experienced where it is expected, between targets which are the most

visually similar. In contrast, the Euclidean metric tended to classify the most aspect-independent target well, but its simplicity led to large errors even for relatively distinguishable targets. The Mahalanobis metric worked well for the simpler target set, but offered no improvement over the Euclidean distance for the more realistic targets, where the profile statistics seem to deviate further from the implicitly assumed gaussian nature. Profile matching was the most accurate of the conventional techniques, but lacked the facility of the neural network in generalizing from the training data to overcome aspect-imposed differences between training and testing profiles.

When the testing profiles were corrupted with noise it was seen that both neural network and Mahalanobis distance classifiers suffered a significant error for low S/N ratios. This error was attributed to the misleading training, which was performed with noise-free training profiles, causing an over-sensitivity to noise in both algorithms. Profile matching performed well despite the noise, because of its matched filtering approach, and the simplicity of the Euclidean metric granted it a relatively high indifference to noise as well. The accuracies of the neural network and Mahalanobis distance methods were seen to improve significantly, however, when training was performed with noise corrupted profiles. The accuracy of the neural network approach was seen to peak when the noise level in the training process was matched to that expected in testing, and with this match, the accuracy achieved exceeded that of profile matching, even for low S/N levels.

When the testing profiles were shifted one or more range cells before applying the neural network classifier, the accuracy obtained dropped significantly. In contrast, the correlation nature of the profile matching algorithm permitted insensitivity to alignment uncertainty by sliding the profile when performing the correlation. Again, however, it was seen that the shortfall of the neural network was due to misleading training, and after retraining with shifted data, the accuracy of the neural network once again exceeded profile matching.

Finally, the storage and computational requirements for profile matching and Mahalanobis distance classifiers were compared to that of the neural network. The network was seen to achieve a reduction of over 10 times when compared to profile matching, and 5–10 times over the distance method. This efficiency, combined with the above accuracy, which equals or exceeds that of conventional algorithms, suggests that neural network classifiers show promise in overcoming many of

the practical problems which currently limit the effectiveness and applicability of target classification systems.

Acknowledgments

This material is based upon work supported under a National Defense Science and Engineering Graduate Fellowship for RGA and supported by contracts from the NASA Contract NAGW-1617, the ARMY Corp of Engineers Contract DACA39-87-K-0022, and U.S. Army Cold Regions Research and Engineering Laboratory DACA89-90-K-0016.

References

[1] Kong, J. A., A. A. Swartz, H. A. Yueh, L. M. Novak, and R. T. Shin, "Identification of terrain cover using the optimum polarimetric classifier," *J. of Elec. Waves and Appl.*, **2**, No. 2, 171–194, 1987.

[2] Swartz, A. A., *The Optimal Polarimetric Matched Filter for Achiev -ing Maximum Contrast in Radar Imagery*, Master's Thesis, MIT, Cambridge, MA, June 1988.

[3] Lim, H. H., H. A. Yueh, J. A. Kong, R. T. Shin, and J. J. van Zyl, "Contrast and classification studies of polarimetric SAR images for remote sensing of earth terrain," *Progress in Electromagnetic Research Symposium*, Boston, MA, July 25–26, 1989.

[4] Holm, W. A., "On the use of polarization in radars for discriminating stationary targets from ground clutter," *Progress in Electromagnetic Research Symposium*, Boston, MA, July 25–26, 1989.

[5] Mains, R. H., and D. L. Moffatt, "Complex natural resonances of an object in detection and discrimination," TR-3424-1, Ohio State University, Electroscience Laboratory, Columbus, OH, June 1974.

[6] Kennaugh, E. M., "The K-pulse concept," *IEEE Trans. Antennas Propagat.*, **AP-29**, No. 2, 329–331, Mar. 1981.

[7] Chen, K. M., D. P. Nyquist, E. J. Rothwell, W. M. Sun, and

P. Ilavarasan, "Radar target discrimination using E-pulses and S-pulses," *Progress in Electromagnetic Research Symposium*, Boston, MA, July 25–26, 1989.

[8] Chen, K. M., D. P. Nyquist, E. J. Rothwell, L. L. Webb, and B. Drachman, "Radar target discrimination by convolution of radar returns with extinction-pulses and single-mode extraction signals," *IEEE Trans. Antennas Propagat.*, **AP-34**, No. 7, 896–904, July 1986.

[9] Rothwell, E. J., D. P. Nyquist, K. M. Chen, and B. Drachman, "Radar target discrimination using extinction-pulse technique," *IEEE Trans. Antennas Propagat.*, **AP-33**, No. 9, 929–936, Sept. 1985.

[10] Rothwell, E. J., K. M. Chen, D. P. Nyquist, and W. M. Sun, "Frequency domain E-pulse synthesis and target discrimination," *IEEE Trans. Antennas Propagat.*, **AP-35**, No. 4, 426–434, Apr. 1987.

[11] Chamberlain, N. F., E. K. Walton, and F. D. Garber, "Radar target identification of aircraft using time domain polarimetric signatures," *Progress in Electromagnetic Research Symposium*, Boston, MA, July 25–26, 1989.

[12] Poor, H. V., *An Introduction to Signal Detection and Estimation*, Springer-Verlay, New York, 1988.

[13] Thomas, I. L., V. M. Benning, and N. P. Ching, *Classification of Remotely Sensed Images*, Adam Hilger, Bristol, 1987.

[14] Novak, L. M., "On the sensitivity of bayes and fisher classifiers in radar target detection," *IEEE Proceedings of the Eighteenth Asilomar Conference on Circuits, Systems, and Computers*, Pacific Grove, CA, Nov. 5–7, 1984.

[15] Novak, L. M., and M. B. Sechtin, "On the performance of linear and quadratic classifiers in radar target detection," *IEEE Proceedings of the Twentieth Asilomar Conference on Signals, Systems, and Computers*, Monterey, CA, Nov. 10–12, 1986.

[16] Lee, Y., *Classifiers : Adaptive Modules in Pattern Recognition Systems*, Master's Thesis, MIT, Cambridge, MA, May 1989.

[17] Benediktsson, J. A., P. H. Swain, and O. K. Ersoy, "Neural network approaches versus statistical methods in classification of mul-

tisource remote sensing data," *IEEE Trans. Geo. and Rem. Sens.*, **GE–28**, No. 4, 540–552, July 1990.

[18] Huang, W. Y., and R. P. Lippmann, "Comparisons between neural net and conventional classifiers," *IEEE First International Conference on Neural Networks*, San Diego, CA, June 21–24, 1987.

[19] Lippmann, R. P., and B. Gold, "Neural-net classifiers useful for speech recognition," *IEEE First International Conference on Neural Networks*, San Diego, CA, June 21–24, 1987.

[20] Huang, W. Y., R. P. Lippmann, and B. Gold, "A neural net approach to speech recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, NY, April 11–14, 1988.

[21] Waibel, A., T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition : neural networks vs. hidden Markov models," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, NY, April 11–14, 1988.

[22] Gorman, R. P., and T. J. Sejnowski, "Learned classification of sonar targets using a massively parallel network," *IEEE Trans. Acoust., Speech, and Sig. Proc.*, **ASSP-36**, No. 7, 1135–1140, July 1988.

[23] Sigillito, V. G., S. P. Wing, L. V. Hutton, and K. B. Baker, "Classification of radar returns from the ionosphere using neural networks," *John Hopkins APL Technical Digest*, **10**, No. 3, 1989.

[24] Decatur, S. E., "Application of neural networks to terrain classification," *International Joint Conference on Neural Networks*, 1989.

[25] Decatur, S. E., *Applications of Neural Networks to Terrain Classification*, Master's Thesis, MIT, Cambridge, MA, June 1989.

[26] Farhat, N. H., "Microwave diversity imaging and automated target identification based on models of neural networks," *Proceedings of the IEEE*, **77**, No. 5, 670–681, May 1989.

[27] Lippmann, R. P., "An introduction to computing with neural nets," *IEEE ASSP Magazine*, **4**, 4–22, 1987.

[28] Lippmann, R. P., "Neural network classifiers for speech recognition," *The Lincoln Laboratory Jour.*, **1**, No. 1, 107–124, 1988.

[29] Rumelhart, D. E., G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 1986.

[30] Knott, E. F., J. F. Shaeffer, and M. T. Tuley, *Radar Cross Section : Its Prediction, Measurement, and Reduction*, Artech House, Dedham, MA, 1985.

[31] Wehner, D. R., *High Resolution Radar*, Artech House, Norwood, MA, 1987.