A Neural Network for Target Classification Using Passive Sonar

Robert H. Baran
Naval Surface Warfare Center, White Oak (code U23, rm 2-250)
Silver Spring, MD 20903-5000

James P. Coughlin
Department of Mathematics, Towson State University

Abstract

This concerns the design, training, test and evaluation of a feed-forward neural network for classifying acoustic signals emitted by ships in transit by an omnidirectional hydrophone. Relatively noisy surface ships, moving rapidly at medium to long range, emit signals which superficially resemble those of quieter submarines, moving more slowly and closer to the listening device. The neural network approach is motivated by an obvious analogy to the sonar classifier of Gorman and Sejnowski, who trained a neural network to classify active sonar returns from two undersea objects. The present problem can be solved by a similar network architecture, the outputs indicating which target type (if any) is present. The inputs represent the evolution of spectral densities for each of a number of time lags. Yet the number of target types and encounter geometries is far greater than could possibly be covered in any representative way by a training set comprised of real world data. Thus the task is to connect the network to a high fidelity, model-based digital simulator and to show that, by training on the output of the simulator, the neural network can learn to pass realistic tests.

Introduction

Noncooperative target identification is one of the fundamental problems of military computing; and pattern recognition is a classical application of artificial neural networks. Therefore it is natural to conduct research and exploratory development in the intersection of these areas and to foresee early success in the insertion of neurocomputing technology into the fusing systems of advanced ordnance.

This concerns the design, training, test and evaluation of a feed-forward neural network for classifying acoustic signals emitted by ships in transit by an omnidirectional hydrophone. Relatively noisy surface ships, moving rapidly at medium to long range, emit signals which superficially resemble those of quieter submarines, moving more slowly and closer to the listening device. This range-speed ambiguity makes discrimination between surface ships and submarines a hard practical problem which, moreover, has resisted fully satisfactory treatment by conventional methods.

This describes a neural network design-and-testing exercise based on a simplistic model that captures a few of the salient features of the problem.
submarine targets is exceeded only by the difficulty of translating such models into executable algorithms for real time discrimination. Neural networks provide an alternative to the traditional algorithmic approach. Given a comprehensive training set, consisting of typical target signatures and the encounter parameters which generated them, a suitable neural network can learn to associate encounter parameters with the inputs in real time.

The neural network approach is motivated by an obvious analogy to the sonar classifier of Gorman and Sejnowski (1988) who, in a landmark study, used a neural network for the classification of active sonar returns from two undersea objects, a metal cylinder and a similarly shaped rock. In the widely publicized DARPA Neural Network Study (1988), "metal cylinder" was transcribed as "mine". The inputs represented the spectrum levels derived from a lofargram of the echoes and the two output units registered the classification. Actual field test data was used to create the training set. The present problem can be solved by a similar network architecture, with the outputs indicating which target type (if any) is present. The inputs represent the evolution of spectral densities for each of a number of time lags. Yet the number of target types and encounter geometries is far greater than could possibly be covered in any representative way by a training set comprised of real world data. One of the most commonly cited strengths of neural network pattern recognizers is their ability to learn by example from training sets comprised of real world data. In the real world, however, the data may be costly to obtain, widely distributed across a number of carefully guarded databases, and unstandardized with regard to format. Moreover, the apparent profuseness of such data may obscure the fact that its acquisition was driven by constraints and priorities which translate into a training set that is not truly representative of the operational environment. Dividing the data into subsets, one for training and one for testing, the analyst still has no way to gauge how effectively the trained network will generalize beyond the horizons of the data.

The fundamental task therefore is to connect the network to a high fidelity, model-based digital simulator and to show that, by training on the output of the simulator, the neural network can learn to pass realistic tests. This approach was illustrated by Baran (1989) in constructing a pattern set for configuring a symmetric (Hopfield) network to classify targets using a different set of sensors and classification criteria. The present case is different by virtue of the immense barriers, including technical complexity and procedural difficulties, that stand in the way of designing the simulation to achieve the highest possible fidelity. This paper will describe a neural network design-and-testing exercise based on a simplistic model that captures a few of the salient features of the problem. It is hoped that the lessons learned in this exercise will bring the practical problem into sharper focus.

**Elements of the Model**

Begin with the notion of an acoustic point source that radiates correlated gaussian noise with the same intensity in every direction. The source is characterized by a type \( i \in \{0,1,2\} \), 1 and 2 indicating surface ships and submarines, respectively. Let \( i = 0 \) mean that no target is present. The noise intensity is measured at unit distance from the source. Let \( S(f;l; i,V) \) denote the power spectral density at frequency \( f \) at 1 unit distance when the source is of type \( i \) and moving at a constant velocity \( V \). In particular, let

\[
10 \log S(f;l; i,V) = a_i + b_iV - c_i \log(f/f_l),
\]

where target type-specific coefficients \( a \), \( b \), and \( c \) are to be defined and \( f_l \) and \( f_h \) are low- and high-frequency cutoffs. Eqn (1) constitutes the source model. It captures the two most obvious features of

![Figure 1. Radiated noise levels of surface ships and submarines over the frequency range of interest with speed as the parameter.](image-url)
ship acoustic spectra. With reference to Figure 1, the radiated noise power increases with the velocity; and there is a decreasing trend in the spectrum level (as the frequency increases). Figure 1 was produced with the source level parameters

\[
(a_1, b_1, c_1) = (160 \text{ dB}, 1.5 \text{ db/meter/sec.}, 10 \text{ dB})
\]

and

\[
(a_2, b_2, c_2) = (130 \text{ dB}, 1.0 \text{ db/meter/sec.}, 10 \text{ dB}),
\]

with \((f_p, f_H) = (100 \text{ Hz}, 10\text{KHz})\).

**Target Signature Generation**

The target signature depends on the source levels and on the straight line path followed by the source through the proximity of the listening device. The propagation model involves spreading loss and frequency-dependent attenuation (or absorption). Let \(S(f,R; i,V)\) be the power spectral density of the type \(i\) source at speed \(V\) when measured at a radius of \(R\) units, \(R > 1\). Then

\[
S(f,R; i,V) = S(f,l; i,V)e^{-\alpha(f)R/4\pi R^2}
\]

with the attenuation rate \(\alpha(f)\) being given by the equation of Schulkin and Marsh as noted by Urick (1983, p.105).

The target signature model tracks the spectral density through time \(t\) which is relative to the time of closest approach, when the separation between the source and the listening device is a minimum. Let \(H\) and \(D\) be the horizontal and vertical distances, respectively, at the origin of \(t\). The time-dependent squared separation distance is

\[
R(t)^2 = (Vt)^2 + H^2 + D^2.
\]

In particular, the range at \(t = 0\), at the point of closest approach, is

\[
R_0 = (H^2 + D^2)^{1/2}.
\]

The target signature,

\[
M(f,t; \theta) = S(f,R(t); i,V),
\]

is obtained by substituting \(R(t)\) from eqn (3) into the preceding equation. On the left side, \(\theta = (i,V,R_0)\) is the vector of encounter parameters.

**Range-Speed Ambiguity**

Relatively noisy surface ships, moving rapidly at medium to long range, emit signals which superficially resemble those of quieter submarines, moving more slowly and closer to the origin. For any single frequency which is low enough that the attenuation can be ignored, the signature (4) has the equivalent form

\[
m(t; p,q) = p/(qt^2 + 1),
\]

**Fig. 2.** The nonlinear equation (6) has one solution for each target type. This range-speed ambiguity can be resolved by obtaining simultaneous target signatures in each of several frequency bands.
in terms of new parameters
\[ p = S(f,1; \theta)/4\pi R_0^2 \]
and
\[ q = (V/R_0)^2. \]
Note that \( p/q = S(f,1; \theta)/4\pi V^2. \) Defining
\[ r = 4\pi p/q, \]
so that
\[ S(f,1; \theta) = rV^2, \]
we can equate \( 10 \log(rV^2) \) to the right side of equation (1), obtaining
\[ (6) \quad a_1 + b_1V - c_1\log(f/f_1) = 10\log(r) + 20\log(V). \]
Given a way of fitting equation (5) to the time history of the spectral density at \( f, \) suitable estimates \( \hat{p} \) and \( \hat{q} \) of the new parameters can be used to estimate
\[ \hat{r}/4\pi = \hat{p}/\hat{q}. \]
Finally this \( \hat{r} \) is substituted for \( r \) in equation (6). Taking speed \( V \) as the independent variable, equation (6) has one solution for each target type. Figure 2 shows how these solutions are obtained graphically. It assumes a type 1 source at speed \( V = 10 \) m/sec with \( R_0 = 2000 \) m. The concave curve is described by the right side of (6). The straight lines plot the left side of the equation for each target type. The concave curve intersects both straight lines at plausible values of the velocity which correspond to different target types and closest point of approach distances (c.p.a.d.'s).

This range-speed ambiguity can be resolved by repeating the same procedure for each of several nonoverlapping frequency bands, subject to having unequal rolloff rates \( c_1 \) and \( c_2. \) (Each band gives rise to its own signature and hence its own estimate \( \hat{r}. \) When the correct target type is assumed, the solution to every frequency-specific instance of equation (6) is the same. Assuming the wrong target type leads to conflicting solutions.) While there may be more powerful algorithms for resolving the range-speed ambiguity, their application to real time signal processing could become harder as the source spectra are modeled with increasing sophistication and detail. The next sections will describe the design of a neural network for implicitly solving the simultaneous nonlinear equations that extract the target type from the target signature. The neural network promises to be robust enough to learn the distinguishing features of realistic spectra. Moreover, the fully trained network is ideally suited to real time detection and classification.

Preprocessing and Data Representation

It is necessary to make some concrete assumptions about the preprocessing steps that intervene between the listening hydrophone and the neural network. For each \( t = 1, 2, ..., \) L-1 we have a block of serial data, beginning at \( t-T/2 \) and ending at \( t+T/2, \) \( T < 1 \) unit time. Each block is a segment of the time series obtained by sampling and digitizing the output of the hydrophone at a rate of \( 2B \) samples per unit time. If \( 2B \) is the Nyquist rate, \( B \approx f_h \) is the bandwidth. Each block then contains \( 2BT \) sampled data points. The blocks are Fourier transformed, each giving rise to a t-dependent spectral density \( X(f,t). \) Each of the BT values of \( X(.,t) \) is exponentially distributed with a mean of \( \sigma^2/2\pi, \) where \( \sigma \) is the r.m.s. noise (Korngold, 1964). The whole array \( X \) of spectral density values, which can be envisioned as a display of L horizontal scan lines with BT color-coded values of \( X \) on each line, is called a lofargram.

The availability of cheap, reliable devices for real time spectral density computation has made the lofargram a kind of de facto standard for data preprocessing in neural network development efforts in the realm of sonar signal processing. Yet it is not generally practical to drive a neural network directly with the lofargram data, since even the simplest problems give rise to \( X- \)arrays with hundreds of thousands of elements. Accordingly the array \( X \) is partitioned and the elements of the sub-arrays are summed (or averaged). Each such sum is assigned to one of the input nodes of the neural network.

For example, if \( B = 10 \) kHz, \( T = 1 \) sec., and \( L = 10 \) time slices, we have \( BTL = 10^3 \) elements in the array \( X \) with ten rows and 10,000 columns. Let the frequencies be lumped into bins of width \( n, \) indexed by \( k, \) as
\[ (7) \quad X_k(t) = \sum_{f = kB/n}^{(k+1)B/n} X(f,t), \quad k = 1, 2, ..., B/n. \]
With \( B/n = 3 \) we have (10 time slices)(3 bins/slice) = 30 nodes in the input layer of our neural network. Without much loss of generality it may be assumed that the acoustical background is characterized by a noise power of \( 2\pi \) irrespective of \( f. \) Then the distribution of \( X_k(t) \) is normal with mean \( \mu (k,t) \) and variance \( \lambda^2(k,t) \)
\[ (8a) \quad \mu (k,t)/n = 1 + M([2k+1]B/2n, t; \theta)/2\pi \]
and...
Network Architecture and Training

A three-layer, series-coupled perceptron was trained on the output of a model-based simulator as shown in Figure 3. Figure 4 shows the network architecture in the general case, with $N_i$ inputs, $N_h$ hidden units, and $N_o$ output units. The specifics of the present exercise are $(N_i, N_h, N_o) = (30, h, 2)$, where the number $h$ of hidden units was variable. The hidden and output layers are composed of binary (0-1) units. Following the creation of a network with random initial weights, the training procedure marches through a sequence of cycles. Each cycle follows these steps:

1. Generate an encounter parameter vector $\theta = (i, V, R_0)$. Let $R_0$ and $V$ be uniformly distributed between appropriate lower and upper bounds (which may conditioned on $i$). The target type index is zero with probability 1/2. If $i$ is nonzero, the type is 1 or 2 equiprobably. The index $i$ determines the source level parameters in equation (1).

2. Compute a target signature from the encounter parameter using the equations of section 2 above. Represent the signature as an input to the network, making use of equations (8) and generating the normal variates with the formula
   \[ X_k(t) = \mu(k,t) + 2\lambda(k,t)\Gamma(3/2)\cos(2\pi U_1)\ln U_2^{1/2}, \]
   where $(U_1, U_2)$ is a pair of computer-generated random numbers, each uniform on the unit interval. Thus the network is presented with 30 real-valued inputs.

3. Propagate the input forward through the hidden layer to the pair of output units. The output convention is "00" for no target, "01" for a type 1 target, and "11" for type 2.

4. Compare the output of the network to the desired states, which correspond to the two digits of the type index $i$ (base two).
   (4.1) If the output matches the target, return to step 1.
   (4.2) If there is an error, modify the weights of the network according to the procedure of the next section. (Then return to step 1 above.)

Backpropagation

Rosenblatt (1961, 1962) pioneered the study of learning algorithms for feed-forward neural networks with binary threshold units. The perceptron typically had a "retina" of sensory (S) units feeding forward to an association (A) layer which, in turn, drove the response (R) units. Today it is customary to refer to input units, hidden units, and output units. Mixing the new parlance with the old, most of Rosenblatt’s perceptrons had input-to-hidden layer (S-A) weights that were fixed while the hidden-to-output (A-R) weights were modified incrementally in the course of many passes through the training set. The asymptote of the individual perceptron learning curve was clearly limited by the suitability of the fixed S-A weights, which were generated by a stochastic rule.

Rosenblatt (in Neurodynamics, Chapter 13, page 292) also used a "back-propagating error correction procedure" for training the S-A weights. The essence of this backpropagation technique was a brief list of rules for assigning errors to hidden (A) units based on their interactions with output (R) units that assumed the wrong state in response to the training input. This backpropagation (BP) algorithm takes its cue from the output units, propagating corrections back towards the input end of the net if it fails to make a satisfactory connection quickly at the output end. The actual modification to the weights is formally the same whether an output unit or a hidden unit is considered. Thus if the error assigned to a unit is positive, the weights of all connections from active units are increased, eventually turning it on. If the error is negative, the weights of connections from active units are decreased. The essential feature of the method is a probabilistic procedure for assigning errors to hidden units. With reference to Figure 4, the procedure is as follows, the notation being separate from that of preceding sections:

(I) For the $r$-th output unit, let $E_r = d_r - z_r$ be the error, where $d_r$ is the desired state and $z_r \in \{0,1\}$ is the observed state. Note
   \[ z_r = \sum_{a=1}^{N_h} W_{ra} y_a \]
   in terms of the unit step function $\sum_{a=1}^{N_h} W_{ra} y_a$ and the A-R weight matrix $W$, where
   \[ y_a = \sum_{s=1}^{N_l} V_{as} x_s - u_a \]
   is the state of the $a$-th hidden unit, obtained by feed-
Fig. 3. Block diagram of the procedure for training the neural network on the output of a model-based simulation.

Fig. 4. Schematic diagram of a simple trilayer perceptron.
ing the inputs \(x\) forward through the S-A weight matrix \(V\). Thresholds \(u\) are ascribed to the hidden units. Note \(d_r \in \{-1, 0, +1\}.

(II) For the \(a\)-th hidden unit, error \(E_a\) is computed as follows for each training input: Begin with \(E_a = 0\).

(IIA) If \(y_a = 1\), and a nonzero error \(d_r\) differs from the sign of \(W_{ra}\) then \textit{decrement} \(E_a\) \textit{by} one \[\text{with probability } p_1\]

(IIB) If \(y_a = 0\), and a nonzero error \(d_r\) agrees in sign with \(W_{ra}\) then \textit{increment} \(E_a\) \textit{by} one \[\text{with probability } p_2\]

(IIC) If \(y_a = 0\), and a nonzero error \(d_r\) differs in sign from \(W_{ra}\) then \textit{increment} \(E_a\) \textit{by} one \[\text{with probability } p_3\]

For all other conditions, \(E_a\) is unchanged.

(III) Let \(\text{sgn(\cdot)}\) be the sign function. Define \(\text{sgn(0)} = 0\). The modification is

\[
\Delta W_{ra} = g y_a \text{sgn}(d_r) \text{ for A-R weights or} \\
\Delta V_{as} = g x_s \text{sgn}(E_a) \text{ for S-A weights,}
\]

using a suitable learning rate parameter \(g > 0\).

While Rosenblatt mainly restricted his attention to binary 0-1 inputs, the formula for the modifications to the upstream weights \(V\) can be used on real-valued inputs without any formal change. Thus the inputs defined by eqn (7) in section 3 above are impressed directly at the input layer except for a constant offset. To be specific, let the single index \(s\) attached to the network input \(x_s\) in Figure 4 be determined by the frequency bin \(k\) and the time slot index \(t\) through the rules

\[s(k, t) = 1 + t + 10(k - 1), k \in \{1, 2, 3\}, t \in \{0, 1, \ldots, 9\}\]

Then the network input is

\[x_s(k, t) = X_k(t)/n - \text{offset;}
\]

and it is numerically generated by substituting the Monte Carlo formula noted in step (2) of the cycle described in section 4:

\[x_s(k, t) = \{\mu(k, t) + 2\lambda(k, t)\Gamma(3/2)\cos(2\pi U_1)(-\ln U_2)^{1/2}\}/n - \text{offset.}
\]

The results described below were obtained with offset \(= 1.0\). (Initial tests with zero offset were quite discouraging.)

The results sketched below were obtained with \(g = 0.01\) and \((p_1, p_2, p_3) = (.9, .7, .1)\). These choices were made based on study of the few results exhibited by Rosenblatt and on rather extensive numerical experiments in which small perceptrons were trained to classify random patterns of binary digits (Baran, 1991).

**PERFORMANCE ANALYSIS**

The neural network can commit errors of nine different kinds, since three possible target types are classified by two bits at the output of the net. Table I assigns a nomenclature to these errors. The output is supposed to discriminate between surface ships and subs only when the right bit indicates rejection of the null hypothesis \((i = 0)\). The Table shows two kinds of false alarms (FA) and four kinds of false rest (FR). Error types CE correspond to the off-diagonal elements of the familiar "confusion matrix". All kinds of errors are treated equally by the "back-propagating error correction procedure" used in the training process. In the evaluation process, on the other hand, the output "10" in response to an input of type 0 is not an error, because the the right (least significant) output bit, which indicates the presence of a target, enables the decision-making process which begins by reading the other output bit. The false alarm probability is \(FAP = Pr(FA)\) and the false rest probability (FRP) is similarly defined. The confusion probability is \(CEP = Pr(CE)\). Finally, the total probability of error is

\[
P_E = FAP + FRP + CEP.
\]

**Table I. Classification of Errors.**

<table>
<thead>
<tr>
<th>type</th>
<th>&quot;00&quot;</th>
<th>&quot;01&quot;</th>
<th>&quot;11&quot;</th>
<th>&quot;10&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>FA</td>
<td>FA</td>
<td>--</td>
</tr>
<tr>
<td>1</td>
<td>FR</td>
<td>--</td>
<td>CE</td>
<td>FR</td>
</tr>
<tr>
<td>2</td>
<td>FR</td>
<td>CE</td>
<td>--</td>
<td>FR</td>
</tr>
</tbody>
</table>

Detection Experiments

Detection experiments were designed to show that the neural network can learn to distinguish between targets (type 1 or 2) and background \((i = 0)\) with error probabilities approaching some point on the detector operating characteristic of the optimum detection receiver (Van Trees, 1983). The detector operating
Fig. 5. Detection error probability versus signal-to-noise ratio for the optimum detection receiver.

Fig. 6. Detection error rate vs. number of cycles for a (30,6,2) BP net with nominal SNR1 = SNR2 = 1.5 dB. The rate is computed every 10 cycles as a 500-point moving average.
characteristic in general is a concave curve on $[0, 1]^2$ which shows $1 - \text{FRP}$ as a function of $\text{FAP}$, usually parameterized by a detectability measure such as signal-to-noise ratio (SNR). Since targets and background are presented equiprobably in the training process it is not surprising to find that the operating point of the trained net lies in the vicinity of $\text{FRP} \approx \text{FAP}$. Thus we use the detection error probability,

$$\text{DEP} = \text{FRP} + \text{FAP},$$

as the performance figure of merit.

By taking target velocity $V$ sufficiently small, and setting $c_1 = c_2 = 0$ in the source model, one can artificially cause all the inputs given by equation (10) to be independent and identically distributed under both the null hypothesis and the composite alternative ($i \neq 0$). This gives rise to the statistical problem of detecting a shift in the mean of a normal population subject to a proportionate increase in the variance. Figure 5 shows the theoretical DEP as a function of a nominal SNR which, with reference to eqn (1), is

$$10 \log(\text{SNR}_i) = a_i - 20 \log R_0 - a_0, \quad i \in \{1, 2\},$$

with $a_1 = a_2$, where $a_0 = 10 \log \left( \frac{\sigma^2}{2\pi} \right) = 0$ without loss of generality. Given the same c.p.a. distance for each encounter, there is no way to distinguish between surface and submarine targets in this degenerate case. Therefore the output of the target type indicator unit in the neural network was ignored in both training and evaluation.

The DEP of the neural network approaches the classical objective as the training procedure is iterated many times. Figure 6 shows the detection error rate for a net with six hidden units operating on a nominal SNR = 1.42, or $10 \log 1.42 = 1.5$ dB. The error rate is a 500-point moving average computed every 10 cycles of the training procedure. After about 1000 cycles the error rate falls below $1/500$. The time required to reach $\text{DEP} = 1/500$ the first time ranged from about 60 (in the Figure) to over 150 cycles in similar trials which differed only with respect to the seed of the random number generator. (The seed determines both the training input and the initial weights.) Higher values of the SNR gave faster convergence to the $1/500$ error level but make estimation of the long run (or limiting) DEP very time consuming.

Detection-and-Classification Experiments

The ease of the discrimination task increases with the divergence of the two source spectra, described in Section 2 above in terms of source level parameter vectors $(a_i, b_i, c_i), \quad i \in \{1, 2\}$. The source level parameters were the same as given in Section 2 except that the rolloff parameter $c_1$ for the surface ship was set at 15 dB per decade. In order to see whether the neural network could resolve the range-speed ambiguity, the

Fig. 7. Total error rate vs. number of cycles for a (30,16,2) network trained to classify submarine targets. The rate is computed every 10 cycles as a 10-point moving average. Ten trials are averaged in each plotted point.
c.p.a. distance and velocity were greater for type 1 targets. Since the surface ship is 30dB louder than the submarine at the low end of the spectrum, the typical c.p.a. distances, \( R_{01} \) and \( R_{02} \) for types 1 and 2, respectively, will satisfy

\[(13a) \quad 30 \text{ dB} = 20 \log R_{01} - 20 \log R_{02};\]

or

\[(13b) \quad \frac{R_{01}}{R_{02}} = 10^{1.5} \approx 30\]

in the worst case. To prevent the SNR of the quietest targets from falling below 6 dB, let \( R_{02} \) not exceed \( R_{2,\text{max}} \):

\[(14) \quad a_2 - 20 \log R_{2,\text{max}} = a_0 + 6 \text{ dB}.\]

With reference to Figure 1, this gives \( R_{2,\text{max}} = 500 \) meters (approximately). To keep from biasing the tests in favor of high SNRs, place a lower bound \( R_{2,\text{min}} = 100 \) meters on the c.p.a. distances of the submarine targets. Recalling (13b) then, the c.p.a. distance of the surface ship lies between \( 30R_{2,\text{min}} = 3000 \) meters and \( 30R_{2,\text{max}} = 15000 \) meters. Depending on the randomly generated target type index then, the parameter \( R_0 \) is uniformly distributed on one of these two intervals.

The target speed \( V \) is likewise restricted to a type-specific range in a manner that heightens the ambiguity. In connection with equation (5), Section 2.2 defined a parameter \( q = (V/R_0)^2 \) which controls the shape of the target signature. An obvious way to "confuse" the network is to let this speed-to-c.p.a.d. ratio approach unity. In order to keep the speeds of both target types within reasonable bounds, however, the signature generator produced speeds uniformly distributed between 20 and 40 m/sec when \( i = 1 \), and between 5 and 20 m/sec for type 2.

The speeds and distances established by these modeling assumptions give rise to signature widths up to a few hundred seconds. The preprocessing assumed in generating the normal distributions in equations (8) were made to reflect the signature duration. This was accomplished by retaining the \( T = 1 \) second integration time for each time slice and by spacing the slices 20 seconds apart.

After "reconnecting" the target type indicator unit to the neural network, minimization of the total error probability (11) was observed over the course of many training cycles each involving the presentation of simulated target signatures (or noise) at the input. Figure 7 summarizes the results of detection-and-classification experiments in which the network was given 2500 training cycles to reduce the total error rate. Each plotted point is the average of 10 independent trials differing with respect to the random numbers generated (and the initial weights). Each trial yields a 10-point moving average error rate over 2500 training cycles. The pre-training \( P_E \) is about 3/4 under the assumption that each output unit is equiprobably right or wrong. The error rate quickly drops below the 50% level. After 500 training cycles the learning curve begins to approach its asymptotic level, \( P_E = 0.04 \).

These results were obtained with 16 hidden units. Reducing the number of hidden units to 8 or fewer caused the learning curve to flatten in the vicinity of \( P_E = 0.15 \). Broadening the hidden layer, which causes the execution time to scale up as \( (N_i + 2)h \), did not appreciably improve the performance of the net after 2500 cycles.

Similar experiments were performed with larger and smaller values of the minimum SNR, which was 6dB in equation (14). The limiting \( P_E \) was observed to increase gradually as the minimum SNR was reduced, rising from 0.02 (at 10 dB) to 0.35 (at 3 dB).

Conclusions and Recommendations

The ease with which the network learned to detect targets stands in contrast to the difficulty of discriminating between surface and submarine targets. This was not surprising in so far as the target signature generation model was adjusted to highlight the range-speed ambiguity. Superb performance could probably have been obtained by presenting the neural net with an easier problem. The fact that the network attained better than 90% classification accuracy provides encouragement for further work along these lines. A detailed examination of neural net's performance needs to be preceded by a thorough analysis of classical solutions appropriate to the present case in order to set goals for the total probability of error for parameter combinations of interest. Once again, the neural network approach is regarded as a potential shortcut to near optimal performance as opposed to a means of exceeding the classically derived optimas.

It may be that academic curiosity concerning Rosenblatt's backpropagation technique diverted these efforts from the mainstream of contemporary neural network design practice. Use of sigmoidal hidden units and the Rumelhart/PDP gradient descent form of backpropagation might lead readily to better classification accuracy. Indeed, the gradual improvement in total error rate which accompanied the broadening of the hidden layer might suggest that perceptrons with binary threshold units suffer from a reduced capacity relative to today's standard BP nets. On the other hand, if an effective neural network solution can be obtained with binary units, it will be particularly easy to realize high speed detection and classification in the finished product which, in this case, could take the form of
analog integrated circuitry in which the weights are fixed conductances and the neuron-like units are high-gain amplifiers.

References


