Abstract: This paper deals with the application of Radial-Basis Function Networks (RBFN) to radar detection. Attention has been paid on the influence of the Training Signal to Noise Ratio (TSNR) on detection performance. The influence of the network size has been studied too. Results show that the dependence on TSNR value is very high, but more important for complex networks. On the other hand, the results for different network sizes trained with their best TSNR values are very similar. Very good performance can be obtained using small networks trained with the appropriate TSNR value.

Key-Words: RBFN, radar, detection, TSNR.

1 Introduction

The radar detection problem can be formulated as a binary hypothesis test: given a set of N observations, the detection system has to decide whether they are originated from noise (the null hypothesis $H_0$) or from both noise and target (the alternative hypothesis $H_1$). The objective is to minimize a risk function that is defined as the average cost [1]:

$$C = \sum_{i=0}^{1} \sum_{j=0}^{1} C_{ij} P(D_i \mid H_j)P(H_j)$$

where:

- $P(D_i \mid H_j)$ is the probability of deciding $H_i$ when $H_j$ is the true hypothesis.
- $P(H_j)$ is the prior probability of hypothesis $H_j$.
- $C_{ij}$ is the cost associated with deciding $H_i$ when the true hypothesis is $H_j$.

To specify detector performance, the probability of detection ($P_D$) and the probability of false alarm ($P_{FA}$) are used. $P_D$ is the probability of deciding $H_1$ when it is the true hypothesis. $P_{FA}$ is the probability of deciding $H_1$ when $H_0$ is true.

Neural networks have been successfully applied to radar target detection. In [2],[3] the in-phase and quadrature components of the received signal measured at intermediate frequency were applied directly to a Multi-Layer Perceptron (MLP). For simulating radar echoes, Swerling models [4] were used. In this approach, the selected Signal-to-Noise-Ratio for training the MLP (TSNR) appeared as a critical parameter, especially for low probability of false alarm values, the most important case in radar detection.

In order to improve the robustness and the performance of the neural network based detector, a pre-processing stage was proposed in [5],[6]. Results demonstrated that this new scheme is nearly independent on the training signal-to-noise-ratio.

In this paper, Radial-Basis Function Networks (RBFN) are analyzed in order to test their capabilities for radar detection.

2 Received signal

We assume that the received signal has passed through a synchronous detector and its quadrature components are applied to the detection scheme. The complex envelope of the received signal can be represented with expression (2).

$$r(t) = y_{\text{target}}(t) + y_{\text{noise}}(t)$$

$y_{\text{noise}}(t)$ are samples of Additive-White-Gaussian-Noise (AWGN) of zero mean and unity variance, and $y_{\text{target}}(t)$ are samples of target echoes which are characterized by the instantaneous reflected energy, $E_R(t)$ and the instantaneous phase, $\theta(t)$:

$$y_{\text{target}}(t) = \sqrt{2E_R(t)} \exp[j\theta(t)]$$

$E_R(t)$ fluctuates principally as a result of changes in the vector summation of the returns of the many scattering elements that contribute to the target's effective cross section. If scanning radar that collects $n$ target echoes in a scan is assumed, in a Swerling-1 target model, the received $n$ complex samples have
all the same magnitude, and this magnitude varies from scan to scan according to a Rayleigh probability density function (pdf). Therefore, the instantaneous received energy, $E_R(t)$, varies according to an exponential pdf. It is also supposed that the $n$ samples have the same phase, $\theta(t)$, whose value varies uniformly from scan to scan.

$r$ is the $2n$-length vector composed of the quadrature components of the $n$ complex samples, and its probability density function under hypothesis $H_0$ is expressed in equation (4).

$$f(r \mid H_0) = \frac{1}{(2\pi)^n} \exp\left(-\frac{1}{2} \|r\|^2\right) \quad (4)$$

If $E_R$ denotes the received target energy during a radar scan, and $\theta$ is the target echo phase, the corresponding quadrature components are:

$$y_{\text{target}} = y_p + jy_q = \sqrt{2E_R} \cos(\theta) + j\sqrt{2E_R} \sin(\theta) \quad (5)$$

and the probability density function under hypothesis $H_1$ is given by (6).

$$f(r, E_R, \theta \mid H_1) = \exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} [r_i - y_p]^2 - \frac{1}{2} \sum_{i=n+1}^{2n} [r_i - y_q]^2 \right\} \left(\frac{1}{2\pi}\right)^n \quad (6)$$

$y_p$ and $y_q$ are Gaussian random variables of zero mean and variance $E_R$, the expected value of $E_R$. Both are constant for each pattern, $r$, and vary from pattern to pattern (from scan to scan). If we define the average value of the signal-to-noise ratio as:

$$SNR(dB) = 10 \log_{10} \left(\frac{E_R}{\sigma^2_{\text{noise}}}\right) \quad (7)$$

for a AWGN of zero mean and unity variance, $SNR(dB) = 10 \log_{10} (E_R)$.

Figures 1 and 2 show the joint scatter plots of 100 points for $H_0$ (stars) and $H_1$ (circles), in a two dimension problem ($n=1$) for two different signal-to-noise ratios (SNR=5 and 0 dB). These figures show how the distributions overlap significantly, making the detection problem very difficult.

3 Radial-Basis Function Networks

In previous works [3][5][6], Multi-Layer Perceptron (MLP) based detectors were constructed and analyzed. A MLP is based on units that compute a non-linear function of the scalar product of the input vector and a weight vector. Unlike the MLP, in Radial-Basis Function Networks (RBFNs), the functions associated to the hidden units (radial-basis functions) apply to the distance between the input vector and a prototype vector.

In its most basic form, a RBFN is composed of three layers:

- An input layer of $d_0$ nodes that connects the network to its environment.
• A hidden layer composed of \(d_1\) nodes that compute an arbitrary function (generally nonlinear) of the distance between the input vector and a prototype vector.

• An output linear layer, of \(d_2\) nodes, that provides the outputs of the network to the input pattern.

Figure 3 shows the network architecture.

The most common radial basis function is the general multivariate gaussian. For the \(i\)-th hidden unit it can be expressed using (8),

\[
g_i(\mathbf{x}) = \frac{1}{(2\pi)^{d_0} \left| \Sigma_i \right|^{\frac{1}{2}}} \exp \left\{ - \frac{1}{2} \left( \mathbf{x} - \mathbf{t}_i \right)^T \Sigma_i^{-1} \left( \mathbf{x} - \mathbf{t}_i \right) \right\}
\]

(8)

where:

• \(\mathbf{x} = (x_1, x_2, \ldots, x_{d_0})^T\) is the input vector.

• \(\mathbf{t}_i = (t_{i1}, t_{i2}, \ldots, t_{id_0})^T\) is the centre of the \(i\)-th gaussian function.

• \(\Sigma\) is the covariance matrix, which controls the smoothness properties of the function (is a \(d_0 \times d_0\) real matrix).

If we consider the weighted norm [7], whose squared form is defined as:

\[
\| \mathbf{x} \|^2_c = (\mathbf{C} \mathbf{x})^T \mathbf{C} \mathbf{x} = \mathbf{x}^T \mathbf{C}^T \mathbf{C} \mathbf{x}
\]

(9)

where

\[
\frac{1}{2} \Sigma^{-1} = \mathbf{C}^T \mathbf{C}
\]

(10)

equation (8) can be written as

\[
g_i(\mathbf{x}) = \frac{1}{(2\pi)^{d_0} \left| \Sigma_i \right|^{\frac{1}{2}}} \exp \left\{ - \frac{1}{2} \left\| \mathbf{x} - \mathbf{t}_i \right\|_c^2 \right\}
\]

(11)

Taking into consideration that the hidden unit output is multiplied by a weight that will be adjusted during training, the RBF function \(G(\cdot)\) in figure 3 is

\[
G(\mathbf{y}) = \exp \left\{ - \mathbf{y}^2 \right\}
\]

(12)

There are different learning strategies for designing a RBF network [7][8]. The radial basis functions centres (also called prototype vectors) are obtained from the available data set using different techniques: selection at random from the training data set, self-organized selection, or supervised selection.

The matrices \(\Sigma\) can be set to a scalar multiple of the unit matrix, to a diagonal matrix with different diagonal elements or to a non-diagonal matrix. In all cases, a unique matrix can be used for all the hidden units or different matrices can be adjusted for each hidden unit.

4 Application of RBFN to radar detection

The problem of radar detection can be formulated as a binary classification one where target-plus-noise signals correspond to class or hypothesis \(H_1\), and noise signals correspond to class \(H_0\). In our experiments, the input patterns are composed of the quadrature components of eight received pulses. So, the input space dimension is 16.

When the input pattern is applied directly to a MLP, the learning process must decide which of its features should be represented by the hidden neurons [2][3]. The system must be capable of extracting all the information from the observation vector.

In an attempt to improve the MLP based detector performance, we designed and analysed a preprocessing stage composed of a time-frequency analyser followed by a feature extractor [5][6]. Using time-frequency analysis, the input pattern information is distributed in the time-frequency domain, where signal and noise may be separable easily, at the expense of a significant increase in the problem dimension. In order to simplify the design of the MLP, Principal Component Analysers (PCAs) were proposed.

Aimed by the results obtained using the pre-processing stage, we have analyzed the performance of radial-basis function network based detectors.
The hidden units of these networks process the input data in order to obtain a new representation in a new domain where they may be linearly separated. The question to answer is: Can a RBF network perform an input space transformation so that the resulting RBFN detector is almost independent on TSNR? That is, can this transformation perform on a similar or better way than the two-steps pre-processing stage combined with a MLP?

Results have been obtained using three network architectures, with 8, 16 and 32 hidden units, respectively. Only one output has been considered with a desired value \( s_1 = 1 \) if the input pattern belongs to class \( H_1 \), and \( s_1 = 0 \) otherwise.

### 4.1 Training strategy

We have applied a three learning phases strategy ([7][8][9]):

1. The centres of the radial basis functions are determined by fitting a gaussian mixture model with circular covariances using the EM algorithm. The mixture model is initialized using a small number of iterations of the k-means algorithm.

2. The basis function widths are set to the maximum inter-centre squared distance.

3. The hidden to output weights that give rise to the least squares solution can be determined using the pseudo-inverse.

### 4.2 Experimental results

The training set is composed of 2000 patterns of noise and signal-plus-noise, randomly distributed. Different values of TSNR have been considered in order to study the dependence of the network performance on this parameter. TSNRs ranging from -3dB to 15 dB have been considered.

Once the RBFNs have been trained, the \( P_D \) has been evaluated for different Signal to Noise Ratios, and different values of \( P_{FA} \), in order to assess the generalization capability. This experiment has been repeated using different number of radial basis functions in the network, in order to examine the dependence of performance on the network size.

Figures 4 to 9 show the results of these experiments when 8 and 24 radial basis functions are considered.
Figures 4 to 6 show that the best TSNR among those considered for training a RBFN with 8 radial basis functions is -1dB. RBFN performance presents a high dependence on TSNR value. $P_D$ should increase monotonically with SNR, but this is not the general behaviour. There are TSNR values for which the $P_D$ versus SNR curve is convex.

On the other hand, for the highest values of TSNR, the worst results are obtained, showing that the optimum TSNR for this architecture will be low. A similar behaviour has been observed when analyzing the performance of a RBFN with 16 radial basis functions. In this case, the best TSNR among those considered is -1dB.

Figures 7 to 9 show that the dependence on TSNR value increases with the network complexity, especially for high SNR values. Furthermore, some kind of over-fitting is observed, because it is more likely to obtain convex $P_D$ versus SNR curves.

For low values of SNR, the best TSNR among those considered is now -1dB, but similar results are obtained with some other values of TSNR, among them, TSNR=2dB. For higher values of SNR, the best results are obtained with TSNR=2dB, existing a great difference with the results for other TSNR values. So, TSNR=2dB can be considered the best one for training, in general.

Another important issue that has been analyzed is the dependence on the size of the hidden layer once the best TSNR values have been determined. Tables 1 to 3 present values of $P_D$ versus SNR(dB), for different values of $P_{FA}$, when the number of hidden units is 8, 16 and 24. The results are very similar for all the considered sizes, showing that network size is not a critical parameter once the best TSNR value has been found.

Table 1. $P_D$ vs. SNR(dB) for $P_{FA}=1.2\cdot10^{-4}$ and three RBFN structures.

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>16/8/1</th>
<th>16/16/1</th>
<th>16/24/1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.331650</td>
<td>0.340388</td>
<td>0.325175</td>
</tr>
<tr>
<td>3</td>
<td>0.556575</td>
<td>0.564175</td>
<td>0.549725</td>
</tr>
<tr>
<td>5</td>
<td>0.683175</td>
<td>0.689363</td>
<td>0.678100</td>
</tr>
<tr>
<td>7</td>
<td>0.783075</td>
<td>0.787375</td>
<td>0.778888</td>
</tr>
<tr>
<td>9</td>
<td>0.855212</td>
<td>0.858275</td>
<td>0.851975</td>
</tr>
<tr>
<td>10</td>
<td>0.886550</td>
<td>0.893636</td>
<td>0.884288</td>
</tr>
<tr>
<td>11</td>
<td>0.907488</td>
<td>0.909625</td>
<td>0.905263</td>
</tr>
<tr>
<td>12</td>
<td>0.926588</td>
<td>0.928325</td>
<td>0.925125</td>
</tr>
<tr>
<td>13</td>
<td>0.938938</td>
<td>0.940775</td>
<td>0.936913</td>
</tr>
<tr>
<td>15</td>
<td>0.961925</td>
<td>0.962775</td>
<td>0.946463</td>
</tr>
</tbody>
</table>
Table 2. $P_D$ vs. SNR(dB) for $P_{FA} = 5 \times 10^{-4}$ and three RBFN structures.

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>16/8/1</th>
<th>16/16/1</th>
<th>16/24/1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.395038</td>
<td>0.396075</td>
<td>0.402175</td>
</tr>
<tr>
<td>3</td>
<td>0.610475</td>
<td>0.611288</td>
<td>0.615763</td>
</tr>
<tr>
<td>5</td>
<td>0.724463</td>
<td>0.725638</td>
<td>0.729975</td>
</tr>
<tr>
<td>7</td>
<td>0.813131</td>
<td>0.814100</td>
<td>0.817313</td>
</tr>
<tr>
<td>9</td>
<td>0.903038</td>
<td>0.903775</td>
<td>0.905638</td>
</tr>
<tr>
<td>10</td>
<td>0.921575</td>
<td>0.921825</td>
<td>0.922738</td>
</tr>
<tr>
<td>11</td>
<td>0.937513</td>
<td>0.938313</td>
<td>0.938700</td>
</tr>
<tr>
<td>12</td>
<td>0.948200</td>
<td>0.948888</td>
<td>0.948200</td>
</tr>
<tr>
<td>13</td>
<td>0.967738</td>
<td>0.967850</td>
<td>0.954413</td>
</tr>
</tbody>
</table>

Table 3. $P_D$ vs. SNR(dB) for $P_{FA} = 10^{-3}$ and three RBFN structures.

<table>
<thead>
<tr>
<th>SNR(dB)</th>
<th>16/8/1</th>
<th>16/16/1</th>
<th>16/24/1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.428963</td>
<td>0.437275</td>
<td>0.434600</td>
</tr>
<tr>
<td>3</td>
<td>0.637588</td>
<td>0.643563</td>
<td>0.642663</td>
</tr>
<tr>
<td>5</td>
<td>0.745125</td>
<td>0.750400</td>
<td>0.749313</td>
</tr>
<tr>
<td>7</td>
<td>0.828013</td>
<td>0.832638</td>
<td>0.832338</td>
</tr>
<tr>
<td>9</td>
<td>0.886550</td>
<td>0.888725</td>
<td>0.888538</td>
</tr>
<tr>
<td>10</td>
<td>0.911125</td>
<td>0.913175</td>
<td>0.913238</td>
</tr>
<tr>
<td>11</td>
<td>0.927988</td>
<td>0.930088</td>
<td>0.928763</td>
</tr>
<tr>
<td>12</td>
<td>0.943313</td>
<td>0.944650</td>
<td>0.943888</td>
</tr>
<tr>
<td>13</td>
<td>0.952700</td>
<td>0.953788</td>
<td>0.952413</td>
</tr>
<tr>
<td>15</td>
<td>0.970550</td>
<td>0.970963</td>
<td>0.957138</td>
</tr>
</tbody>
</table>

5 Conclusion

In this paper, RBFNs have been applied to radar detection. Swerling-1 targets have been used for simulating radar echoes. The objective of this work is to study the influence of the Training-Signal to Noise Ratio and the number of hidden units on detection performance.

Results show that the dependence on TSNR value is very high, and more important for complex networks. For a given structure, the best TSNR value has been found among a wide range of values. Afterward, the performances of RBFNs trained with their best TSNR values have been compared, showing that the size of the network is not a critical parameter, once its best TSNR is determined. So, the most important problem to be solved is the determination of the best TSNR.

The general conclusion of this work is that very good performances can be obtained using small networks trained with the appropriate TSNR value.

References: