On the Capability of Neural Networks to Approximate the Neyman-Pearson Detector. A Theoretical Study

P. Jarabo-Amores *, R. Gil-Pita, M. Rosa-Zurera, and F. López-Ferreras

Departamento de Teoría de la Señal y Comunicaciones,
Escuela Politécnica Superior, Universidad de Alcalá
Ctra. Madrid-Barcelona, km. 33.600, 28805, Alcalá de Henares - Madrid (SPAIN).

Abstract. In this paper, the application of neural networks for approximating the Neyman-Pearson detector is considered. We propose a strategy to identify the training parameters that can be controlled for reducing the effect of approximation errors over the performance of the neural network based detector. The function approximated by a neural network trained using the mean squared-error criterion is deduced, without imposing any restriction on the prior probabilities of the clases and on the desired outputs selected for training, proving that these parameters play an important role in controlling the sensibility of the neural network detector performance to approximation errors. Another important parameter is the signal-to-noise ratio selected for training. The proposed strategy allows to determine its best value, when the statistical properties of the feature vectors are known. As an example, the detection of gaussian signals in gaussian interference is considered.

1 Introduction

The objective of this paper is to study the capability of neural networks to approximate a Neyman-Pearson detector. This detector maximices the probability of detection (P_D), while maintaining the probability of false alarm (P_{FA}) lower than or equal to a specified value. The characteristics of such a detector are reflected in its ROC (Receiver Operating Characteristic) curve, that relates P_D to P_{FA} [1].

Ruck et al. [2], and Wan [3], demonstrated that a neural network can be used to approximate the optimum bayessian classifier when trained using the mean squared-error criterion.

In previous works, neural networks have been proposed for approximating the Neyman-Pearson detector in different environments [4][5]. These works highlighted the strong dependence of the neural network-based detector performance on the signal-to-noise ratio selected for training (TSNR). They also observed that the difference between the neural detector performance and the Neyman-Pearson detector one, depends on the desired P_{FA}, and so, on the corresponding detection threshold.

Recently, some attempts to reduce the dependence of the neural detector performance on TSNR have been carried out [6], based on the use of a complex pre-processing

* This work has been supported by the "Consejería de Educación de la Comunidad de Madrid" (SPAIN), under Project 07T/0036/2003 1

Jarabo-Amores P., Gil-Pita R., Rosa-Zurera M. and López-Ferreras F. (2004).
On the Capability of Neural Networks to Approximate the Neyman-Pearson Detector - A Theoretical Study.
In *Proceedings of the First International Workshop on Artificial Neural Networks: Data Preparation Techniques and Application Development*, pages 67-74
DOI: 10.5220/0001150100670074

stage that reduce this dependence at the expense of a high computational cost. Nevertheless, no effort has been made in order to explain the reasons of such a dependence, whose knowledge could help to easily select the best TSNR and training strategy for designing neural network-based detectors which approximate the Neyman-Pearson detector.

This paper deals with the theoretical explanation of the effects of the approximation errors in the performance of neural network based detector for approximating the Neyman-Pearson detector.

2 Problem Formulation

The performance of a detector that approximates the Neyman-Pearson detector must be evaluated from the difference between its ROC curve and the Neyman-Pearson detector one. For a given P_{FA}, the difference between the probabilities of detection must be as lower as possible. The decrease in P_D that is observed in the ROC curve for a given P_{FA}, is expressed in (1):

$$\triangle P_D = \frac{\partial P_D}{\partial P_{FA}} \frac{\partial P_{FA}}{\partial \eta} \triangle \eta \tag{1}$$

Practical P_{FA} values are below 10^{-6}, while practical P_D values can be higher than 0.8, so, in practical conditions, ROC curves have high positive slopes for low values of P_{FA}, and low positive slopes for high values of P_{FA}. Besides, the function that relates P_{FA} to detection threshold, η, has a negative slope. Taking into consideration this characteristic, and the fact that the ROC curve of the Neyman-Pearson detector is a characteristic that cannot be modified, the following conclusions can be extracted:

- In the low P_{FA} region, $\frac{\partial P_D}{\partial P_{FA}}$ is usually very high. For the reduction of P_D being low, the magnitude of $\frac{\partial P_{FA}}{\partial \eta} \triangle \eta$ must be very low.
- In the high P_{FA} region, $\frac{\partial P_D}{\partial P_{FA}}$ is usually very low. So a low reduction in P_D can be guaranteed although the magnitude of $\frac{\partial P_{FA}}{\partial \eta} \triangle \eta$ is big.

The Neyman-Pearson detector decision rule is the result of comparing the likelihood ratio, or any other equivalent statistic, to the detection threshold, η. For a desired P_{FA}, this threshold depends on the expression of the selected statistic. In order to study the P_D decrease due to approximation errors, expression (1) must be calculated. Taking this study as starting point, it is possible to identify the design parameters that minimize expression (1), with independence of the mean squared-error minimization strategy selected.

3 Expression of the approximated discriminant function

D. W. Ruck et al. [2] demonstrated that a multilayer perceptron (MLP) converges to a mean squared-error approximation of the Bayes optimal discriminant function, when

trained using the mean squared-error criterion. They study the two-class and multiclass problems, and extended this result to any mean squared-error minimization technique.

For binary detection, they studied a MLP with only one neuron in the output layer. The network was trained to produce 1 when the feature vector was from class H_1 and -1 when the vector is from class H_0. They proved that the neural network output approximates the Bayes optimal discriminant function $g_0(\mathbf{z})$, given in (2) where $\mathbf{z}$ is the feature vector, and $P(H_1|\mathbf{z})$ and $P(H_0|\mathbf{z})$ are the a posteriori probability of the classes.

$$g_0(\mathbf{z}) = P(H_1|\mathbf{z}) - P(H_0|\mathbf{z}) \tag{2}$$

The mean squared-error between the network output, $F(\mathbf{z}, \mathbf{W})$, for a given set of weights, $\mathbf{W}$, and the desired outputs, is given by (3). $E_s(\mathbf{W})$ is the sample mean error calculated for a set of n pre-classified feature vectors, and Z_1 and Z_0 are the sets of all possible feature vectors for class H_1 and H_0, respectively ($Z_0 \cup Z_1 = Z$, $Z_0 \cap Z_1 = \varnothing$, Z being the input space).

$$E_m(\mathbf{W}) = \lim_{n\to\infty} \frac{E_s(\mathbf{W})}{n} = \lim_{n\to\infty} \frac{1}{n}[\sum_{\mathbf{z}\in Z_1} (F(\mathbf{z}, \mathbf{W}) - 1)^2 + \sum_{\mathbf{z}\in Z_0} (F(\mathbf{z}, \mathbf{W}) + 1)^2] \tag{3}$$

Using the Strong Law of Large Numbers, expression (3) can be expressed as (4). Finally, applying the Bayes Formula and rearranging terms, (4) converts into (5).

$$E_m(\mathbf{W}) = P(H_1) \int_Z (F(\mathbf{z}, \mathbf{W}) - 1)^2 f(\mathbf{z}|H_1) d\mathbf{z} + P(H_0) \int_Z (F(\mathbf{z}, \mathbf{W}) + 1)^2 f(\mathbf{z}|H_0) d\mathbf{z} \tag{4}$$

$$E_m(\mathbf{W}) = \int_Z (F(\mathbf{z}, \mathbf{W}) - g_0(\mathbf{z}))^2 f(\mathbf{z}) d\mathbf{z} + \left[1 - \int_Z g_0^2(\mathbf{z}) f(\mathbf{z}) d\mathbf{z}\right] \tag{5}$$

If the training set represents a reasonable approximation to the input space, although the network is trained for minimizing $E_s(\mathbf{W})$, $E_m(\mathbf{W})$ will be minimized. Since the tern in braces in expression (5) is independent of $\mathbf{W}$, minimizing $E_m(\mathbf{W})$ is equivalent to minimizing (6). So the network output is an approximation of the Bayes optimal discriminant function in the mean squared-error sense.

$$E(\mathbf{W}) = \int_Z (F(\mathbf{z}, \mathbf{W}) - g_0(\mathbf{z}))^2 f(\mathbf{z}) d\mathbf{z} \tag{6}$$

In a more general problem, if the network is trained to produce t_{H_1} when the feature vector is from class H_1 and t_{H_0} when the feature vector is from class H_1, expression (4) converts into (7):

$$E_m(\mathbf{W}) = P(H_1) \int_Z (F(\mathbf{z}, \mathbf{W}) - t_{H_1})^2 f(\mathbf{z}|H_1) d\mathbf{z} + P(H_0) \int_Z (F(\mathbf{z}, \mathbf{W}) - t_{H_0})^2 f(\mathbf{z}|H_0) d\mathbf{z} \tag{7}$$

In this case, the network output is a mean squared-error approximation of the function $f_0(\mathbf{z})$ defined in (8).

$$f_0(\mathbf{z}) = \frac{P(H_1)f(\mathbf{z}|H_1)t_{H_1} + P(H_0)f(\mathbf{z}|H_0)t_{H_0}}{P(H_1)f(\mathbf{z}|H_1) + P(H_0)f(\mathbf{z}|H_0)} \tag{8}$$

If η_{net} is the detection threshold for a given P_{FA}, the decision rule approximated with the neural network is given by (9).

$$\frac{P(H_1)f(\mathbf{z}|H_1)t_{H_1} + P(H_0)f(\mathbf{z}|H_0)t_{H_0}}{P(H_1)f(\mathbf{z}|H_1) + P(H_0)f(\mathbf{z}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta_{net} \tag{9}$$

$f_0(\mathbf{z})$ is equal to $g_0(\mathbf{z})$ for $t_{H_1} = 1$ and $t_{H_0} = -1$, and for minimizing the probability of miss-classification, η_{net} must be set to 0. But what Ruck et al. [2], and Wan [3] did not notice is the fact that as $f_0(\mathbf{z})$ can be expressed as a function of the likelihood ratio, the network not only is approximating the minimum probability of error classifier, but it can approximate the Neyman-Pearson detector if the detection threshold is modified attending to probability of false alarm requirements.

The rule (9) shows that for implementing the Neyman-Pearson detector, the detection threshold for a given P_{FA} is not only a function of the likelihood functions, but also depends on the a priori probabilities and the desired outputs. These are parameters that can be selected by the designer when generating the training set, and when determining the activation function of the output neuron.

4 Effect of approximation errors on P_{FA}

The neural network will converge to an approximation of (8), so the decision rule implemented can be expressed as in (10), where $\Delta f_0(\mathbf{z}, \mathbf{W})$ is the approximation error.

$$f_0(\mathbf{z}) + \Delta f_0(\mathbf{z}, \mathbf{W}) \underset{H_0}{\overset{H_1}{\gtrless}} \eta_{net} \tag{10}$$

This decision rule can also be expressed with (11) revealing that the effect of approximation errors can be studied as the effect of erroneous detection thresholds.

$$f_0(\mathbf{z}) \underset{H_0}{\overset{H_1}{\gtrless}} \eta_{net} - \Delta f_0(\mathbf{z}, \mathbf{W}) \tag{11}$$

In order to evaluate the decrease in P_D due to threshold errors, the partial derivative of P_{FA} with respect to the detection threshold must be calculated. This calculus requires the knowledge of the likelihood functions, and can be very tedious due to the complexity of rule (9).

In practice, when designing a Neyman-Pearson detector, the first step consists in determining the likelihood ratio for the problem to be solved, as indicates expression (12). Before determining the detection threshold for a given P_{FA}, a simpler sufficient statistic is calculated applying a set of simplifications and using monotonic functions.

$$\wedge(\mathbf{z}) = \frac{f(\mathbf{z}|H_1)}{f(\mathbf{z}|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta_{cv} \tag{12}$$

Following a similar strategy, in a first step, the rule (9) can be expressed as a function of the likelihood ratio, as indicates expression (13).

$$\frac{P(H_1) \wedge (\mathbf{z}) t_{H_1} + P(H_0) t_{H_0}}{P(H_1) \wedge (\mathbf{z}) + P(H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \eta_{net} \tag{13}$$

The relation between η_{cv} and η_{net} is given in (14).

$$\eta_{cv} = \frac{P(H_0)(\eta_{net} - t_{H_0})}{P(H_1)(t_{H_1} - \eta_{net})} \tag{14}$$

The partial derivative of P_{FA} with respect to η_{net} can be calculated using the chain rule (15).

$$\frac{\partial P_{FA}}{\partial \eta_{net}} = \frac{\partial P_{FA}}{\partial \eta_{cv}} \frac{\partial \eta_{cv}}{\partial \eta_{net}} \tag{15}$$

The second factor of the right part of (15) can be calculated from (14) to obtain (16). It depends on the a priori probabilities of the classes and the desired outputs selected for training, factors that can be controlled by the designer.

$$\frac{\partial \eta_{cv}}{\partial \eta_{net}} = -\frac{P(H_0)(t_{H_0} - t_{H_1})}{P(H_1)(t_{H_1} - \eta_{net})^2} \tag{16}$$

From the analysis of expression (16) the following conclusions can be extracted:

- The function approximated by the neural network has been expressed as a function of the likelihood ratio. When this ratio is greater than or equal to η_{cv}, we decide that hypothesis H_1 is true, and if it is lower than η_{cv}, we decide in favor of hypothesis H_0. So, t_{H_1} must be greater than t_{H_0}.
- η_{net} takes values between t_{H_1} and t_{H_0}, so $\eta_{net} \leq t_{H_1}$.
- From the previous two points, we can conclude that the partial derivative of η_{cv} with respect to η_{net} is always positive.
- For η_{net} values closed to t_{H_1}, that is, for very low P_{FA} values, $\frac{\partial \eta_{CV}}{\partial \eta_{net}}$ is very high (for $\eta_{net} = t_{H_1}$ tends to infinity). We can try to compensate it in some degree by increasing the difference between the desired outputs during training, or constructing training sets where features vectors from hypothesis H_1 are more likely than those from hypothesis H_0.

The first factor of the second part of (15) depends on the likelihood functions of the problem to be solved. Its value is calculated in (17).

$$\frac{\partial P_{FA}}{\partial \eta_{cv}} = \frac{\partial}{\partial \eta_{cv}} \Big[1 - \int_{-\infty}^{\eta_{cv}} f(\wedge(\mathbf{z})|H_0) d(\wedge(\mathbf{z})) \Big] = -f(\wedge(\mathbf{z})|H_0)|_{\wedge(\mathbf{z}|H_0)=\eta_{cv}} \tag{17}$$

To gain an insight into the influence of $\frac{\partial P_{FA}}{\partial \eta_{cv}}$, we follow the strategy of looking for a simpler test statistic. If we denote this new statistic as $F(\mathbf{z})$ and the corresponding detection threshold as η_s, the decision rule can be expressed as in (18).

$$F(\mathbf{z}) \underset{H_0}{\overset{H_1}{\gtrless}} \eta_s \tag{18}$$

The relation between η_s y η_{cv} is determined by the relation that exist between the likelihood ratio and the selected statistic, so it is known. Expression (15) can be re-written as a function of η_s.

$$\frac{\partial P_{FA}}{\partial \eta_{net}} = \frac{\partial P_{FA}}{\partial \eta_s} \frac{\partial \eta_s}{\partial \eta_{cv}} \frac{\partial \eta_{cv}}{\partial \eta_{net}} \tag{19}$$

Expression (19) shows that the partial derivative of P_{FA} with respecto to η_{net} can be expressed as the product of three factors:

- The first factor, $\frac{\partial P_{FA}}{\partial \eta_s}$, is a characteristic of the problem to be solved.
- The second factor, $\frac{\partial \eta_s}{\partial \eta_{cv}}$, also is characteristic of the problem to be solved.
- The third factor, $\frac{\partial \eta_{cv}}{\partial \eta_{net}}$, not only depends on the a priori probabilities of the classes and the desired outputs selected from training, because the value of η_{net} required for a given P_{FA} depends on the problem to be solved.

The usefulness of adding a new factor in (15) only can be proved if a particular case is considered. In the next section, the problem of detecting gaussian signals in gaussian interference in considered.

5 A case study: Detection of gaussian signals in gaussian interference

The problem of detecting gaussian signals in gaussian interference is considered. The feature vector is composed by n independent gaussian samples of zero mean and unity variance under hypothesis H_0, and zero mean and a variance $\sigma_s^2 + 1$ under hypothesis H_1. The signal-to-noise ratio is defined in (20) and the value selected for constructing the training set is denoted as $tsnr$.

$$SNR = 10log(snr) = 10log(\sigma_s^2) \tag{20}$$

For a given $tsnr$, the likelihood functions are expressed in (21) and (22); the likelihood ratio and the corresponding detection rule are given by (23).

$$f(\mathbf{z}/H_0) = \frac{1}{\sqrt{(2\pi)^n}} \exp\Big(-\frac{1}{2}\mathbf{z}^T\mathbf{z}\Big) \tag{21}$$

$$f(\mathbf{z}/H_1) = \frac{1}{\sqrt{(2\pi)^n (tsnr+1)^n}} \exp\Big[-\frac{1}{2(tsnr+1)}\mathbf{z}^T\mathbf{z}\Big] \tag{22}$$

$$\wedge(\mathbf{z}) = \frac{1}{(tsnr+1)^{\frac{n}{2}}} \exp\Big[\frac{tsnr}{2(tsnr+1)}\mathbf{z}^T\mathbf{z}\Big] \underset{H_0}{\overset{H_1}{\gtrless}} \eta_{cv} \tag{23}$$

A simpler sufficient statistic can be obtained applying logarithms and re-arranging terms:

$$F(\mathbf{z}) = \mathbf{z}^T\mathbf{z} \underset{H_0}{\overset{H_1}{\gtrless}} 2\frac{tsnr+1}{tsnr} ln[\eta_{cv}(tsnr+1)^{\frac{n}{2}}] = \eta_s \tag{24}$$

As the likelihood function under hypothesis H_0 does not depend on $tsnr$, the probability density function of $F(\mathbf{z})$ does not depend on it, and for a given P_{FA}, η_s is independent on $tsnr$. Because of that, the performance of the Neyman-Pearson detector is independent on the $tsnr$ value.

Although the ROC curves and $\frac{\partial P_D}{\partial P_{FA}}$ do not depend on $tsnr$, $\frac{\partial P_{FA}}{\partial \eta_{net}}$ and the sensibility of the neural detector to approximation errors depends on it.

The partial derivative of η_s with respect to η_{cv} is given in (25).

$$\frac{\partial \eta_s}{\partial \eta_{cv}} = \frac{2(1+tsnr)}{\eta_{cv} tsnr} \tag{25}$$

$F(\mathbf{z}|H_0)$ is a chi-square random variable with n degrees of freedom. The partial derivative of P_{FA} with respect to η_s is calculated in (26).

$$\frac{\partial P_{FA}}{\partial \eta_s} = -\frac{1}{2^{\frac{n}{2}}(\frac{n}{2}-1)!}\eta_s^{(}\frac{n}{2}-1)\exp(\frac{-\eta_s}{2}) \tag{26}$$

Combining expressions (16), (25) and (26) the partial derivative of P_{FA} with respect to η_{net} has been calculated in (27).

$$\frac{\partial P_{FA}}{\partial \eta_{net}} = -\frac{1}{2^{\frac{n}{2}}(\frac{n}{2}-1)!}\eta_s^{(\frac{n}{2}-1)}\exp(\frac{-\eta_s}{2})\frac{2(1+tsnr)}{\eta_{cv} tsnr}\frac{P(H_0)(t_{H_1}-t_{H_0})}{P(H_1)(t_{H_1}-\eta_{net})^2} \tag{27}$$

6 Conclusions

In this paper, the application of neural networks for approximating the Neyman-Pearson detector is considered. We propose the calculus of the partial derivative of the probability of false alarm with respect to the detection threshold, as a tool to identify the training parameters that can be controlled for reducing the effect of approximation errors over the performance of the neural network based detector.

As a first step, the function approximated for a neural network trained using the mean squared-error criterion is deduced. Without imposing any restriction on the prior probabilities of the classes and on the desired outputs selected for training, we obtain a general expression that reveals that these parameters play an important role in controlling the sensibility of the neural network detector performance to approximation errors.

In previous works, the signal-to-noise ratio selected for training appeared as a critical design parameter, but no effort has been done so as to explain the dependence of the neural network based detector on this parameter. In this paper, we explain this dependence and provide an strategy to determine the best $tsnr$ value when the statistical properties of the feature vectors are known.

References

1. Van Trees, H.L.: Detection, estimation, and modulation theory, Vol. **1**. Wiley, (1968)
2. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., Suter, B.W.: The multilayer perceptron as an aproximation to a Bayes optimal discriminant function. IEEE Transactions on Neural Networks, vol. 1, no. 4, 296–298, December 1990.
3. Wan, E.A.: Neural Network Classification: A Bayesian Interpretation. IEEE Transactions on Neural Networks, vol.1, no. 4, pp. 303-305, December 1990.
4. Gandhi, P.P., Ramamurti, V.: Neural networks for signal detection in non-gaussian noise. IEEE Transactions on signal processing, vol. 45, no. 11, pp. 2846-2851, November 1997.
5. Andina, D., Sanz-Gonz=E1lez, J.L.: Comparison of a neural network detector vs. Neyman-Pearson optimal detector. Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 6, pp. 3573-3576, 1995.
6. Jarabo-Amores, P., Rosa-Zurera, M., López Ferreras, F.: Design of a Pre-processing Stage for Avoiding the Dependence on TSNR of a Neural Radar Detector. Lecture Notes in Computer Sciences, Vol. 2085. Springer-Verlag, Berlin Heidelberg New York (2001) 652–659,