



Murdoch
UNIVERSITY

MURDOCH RESEARCH REPOSITORY

<http://dx.doi.org/10.1109/ICNN.1995.488115>

Zaknich, A. and Attikiouzel, Y. (1995) The modified probabilistic neural network as a nonlinear correlator detector. In: Proceedings., IEEE International Conference on Neural Networks, 27 November - 1 December, Perth, Western Australia, pp. 309 - 313.

<http://researchrepository.murdoch.edu.au/18387/>

Copyright © 1995 IEEE

Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

The Modified Probabilistic Neural Network as a Nonlinear Correlator Detector

Anthony Zaknich, Member IEEE, Associate Member of AES
Yianni Attikiouzel, Senior Member IEEE, Fellow IEE, Fellow IEAust

Centre for Intelligent Information Processing Systems (CIIPS),
Department of Electrical and Electronic Engineering,
The University of Western Australia, Nedlands 6907, Western Australia

Abstract

A Nonlinear correlator detector for the detection of a signal class with some intra class variance is developed using the Modified Probabilistic Neural Network and the General Regression Neural Network. An application, involving the detection of regular tone bursts transmitted over a poor and noisy radio channel subjected to fading, random noise and impulse noise effects, is used to show the effectiveness of the method as compared to a linear correlator.

The statistical dependency between two vector variables \mathbf{x}_1 and \mathbf{x}_2 can be of two types: linear dependency or nonlinear dependency [1]. Linear dependency is what is normally regarded as correlation. If the variables are uncorrelated, they are no longer linearly dependent, but they may still be statistically dependent. Nonlinear dependency is called "residual" dependency because it is the dependency that remains after the linear dependency is removed. Two zero mean variables \mathbf{x}_1 and \mathbf{x}_2 are uncorrelated if the expected value of their product is zero.

$$E[\mathbf{x}_1 \mathbf{x}_2] = 0, \text{ (uncorrelated)}$$

But \mathbf{x}_1 and \mathbf{x}_2 are independent if and only if their joint probability density function (pdf) is equal to the product of the individual densities.

$$p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_1) p(\mathbf{x}_2), \quad \forall \mathbf{x}_1 \text{ and } \mathbf{x}_2, \text{ (independent)}$$

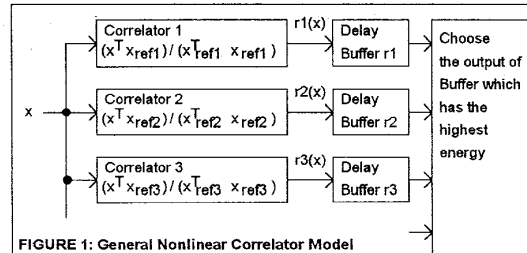
Therefore, it is possible for \mathbf{x}_1 and \mathbf{x}_2 to be uncorrelated but still dependent. It is very useful to be able to assign the degree of dependency of a random variable to some specific process. This can apply to signal detection where the same process can generate a range of different signal waveforms rather than one specific waveform. This is typical in classical linear correlation, or matched filter detector systems. A complex example of this application is the detection and identification in time of a sheep's single chew from its non deterministic jaw sound (This is a research problem that the authors are currently investigating in relation to sheep feeding studies). A simpler example is the accurate detection in time of deterministic signal codes transmitted over a radio channel which have been affected by various linear and nonlinear stochastic channel effects. These problems can be solved via a number of nonlinear signal detection techniques including statistical and neural network pattern recognition. However, these methods usually require sufficient training vectors available from the desired process together with the likely range of noise or non process signals to estimate a posteriori pdfs of each. Also, they are usually designed to test the hypothesis of signal present or absent rather than to give an accurate estimate of the signal location in time. It may be more convenient to have a nonlinear matching filter that works in a similar way to the correlator or optimal linear matching filter. In this case the signal presence would be determined by the output of the correlator exceeding a preset threshold and its location in time established by the position of the highest peak output as it is commonly done in the linear case.

A neural network system can be developed and trained to measure the degree of dependency between a random vector and a process for which there are ample example vectors. To do this a suitable signal and system model must be specified as follows. Assume a specific continuous time process which is generating a class of aperiodic waveform sections of a specified duration with intervals of lower level noise sections in between. The noise may be a mixture of zero mean Gaussian plus other nonlinear noise such as impulse noise. If the processes signal is digitally sampled at a uniform sampling rate of F_s amplitude sample points per second the waveform length is

defined to be p points long. An arbitrary vector \mathbf{x} composed of p successive amplitude sample points from the process can be defined as:

$\mathbf{x} = [x(n), x(n-1), x(n-2), \dots, x(n-p+1)]$, n is the discrete time index.

The class waveform reference sample set can be specified as the vector set $\{ \mathbf{x}_{ref\ i} \mid i = 1, \dots, \text{NUM} \}$ and a general model for a nonlinear correlator can be developed as shown in Figure 1.



There are NUM parallel correlators with each of the $\mathbf{x}_{ref\ i}$ being a correlator matching template. The correlation of vector \mathbf{x} with each reference template is then represented by $(\mathbf{x}^T \mathbf{x}_{ref\ i})$. It is then normalised to one by dividing by $(\mathbf{x}_{ref\ i}^T \mathbf{x}_{ref\ i})$. The normalised output of each correlator is fed into a delay line $2p$ sample points long. The system output is taken from the output of the delay line which has the highest energy if it exceeds a minimum level, otherwise the output is made zero. This introduces a system delay of $2p$, so instead of taking the output of the highest energy delay line it would be acceptable to take the current input to that delay line instead. A neural network can be developed to embody this model and provide a continuous nonlinear correlator output $r(\mathbf{x})$ as the vector \mathbf{x} is taken from the process signal by sliding a sampling window p sample points long forward in time a point at a time.

The Modified Probabilistic Neural Network (MPNN) [2,3] and the General Regression Neural Network (GRNN) [4] have an ideal architecture for this type of problem although any feed forward neural network can be used. Both these networks are related to the Probabilistic Neural Network (PNN) [5] which is normally considered to be a classifier but can also be used for non parametric pdf estimation. They can both be described as general regression or mapping functions given by:

$$y^{\wedge}(\mathbf{x}) = \frac{\sum_{i=1}^{NS} [y_i \exp(-\frac{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}{2\sigma^2})]}{\sum_{i=1}^{NS} [\exp(-\frac{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}{2\sigma^2})]} \quad (1)$$

If the y_i are allowed to have real scalar values equation (1) becomes exactly Specht's GRNN which incorporates each and every training vector pair $\{\mathbf{x}_i \rightarrow y_i\}$ into its architecture as does the PNN (\mathbf{x}_i is a single training vector in the input space and y_i is the associated desired scalar output). If it can be assume that there is only one centre in the input space per output y_i then a convenient general model to use for all forms of the MPNN and the GRNN is:

$$y^{\wedge}(\mathbf{x}) = \frac{\sum_{i=1}^M [Z_i y_i \exp(-\frac{(\mathbf{x} - \text{cent}\mathbf{x}_i)^T (\mathbf{x} - \text{cent}\mathbf{x}_i)}{2\sigma^2})]}{\sum_{i=1}^M [Z_i \exp(-\frac{(\mathbf{x} - \text{cent}\mathbf{x}_i)^T (\mathbf{x} - \text{cent}\mathbf{x}_i)}{2\sigma^2})]} \quad (2)$$

$\text{cent}\mathbf{x}_i$ is the centre or mean vector for class i in the input space (real valued or quantised).

σ is the single learning or smoothing parameter chosen during network training.

y_i is the output related to $\text{cent}\mathbf{x}_i$ (real valued or quantised).

M is the number of unique centres i in the MPNN structure.

Z_i is the number of input training vectors \mathbf{x}_j associated with $\text{cent}\mathbf{x}_i$.

NS = $\sum_{i=1}^M Z_i$, is the total number of training vectors.

Equation (2) can be derived from the GRNN equation (1) through the following approximation:

$$Z_i \exp(-(\mathbf{x}-\mathbf{centx}_i)^T(\mathbf{x}-\mathbf{centx}_i)/2\sigma^2) \approx \sum_{j=1}^{Z_i} \exp(-(\mathbf{x}-\mathbf{x}_j)^T(\mathbf{x}-\mathbf{x}_j)/2\sigma^2) \quad (3)$$

This is a reasonable approximation if the \mathbf{x}_j are close together in a relatively small local space and can be adequately represented by a single centre vector \mathbf{centx}_i . The key to the practical application of the general MPNN equation (2) is related to the method of selection of the y_i and the grouping of the associated input vectors into the centre vector \mathbf{centx}_i for each class i . One solution to this selection and grouping for simple sinusoidal signals proposed by Zaknich et al [2] was to uniformly quantise the noiseless desired y_i , separately group the y_i having positive and negative slopes in the waveform and associate them with the mean of the input vectors mapping to each group. This simple case led to a more general approach, for both simple and more complex signals, of uniquely identifying the quantised y_i having a similar local waveform pattern which was called the MPNN Method A [6]. Method A involved taking the desired waveform $y(t)$ and uniformly sampling it in time to $y(n)$ digital sample points which were then uniformly quantised into one of N quantisation levels to be able to define a desired output phase state vector composed of $y(n)$ and the $m-1$ quantised samples immediately preceding it in time, ie. $(y(n), y(n-1), \dots, y(n-m-1))$. The greater the m the more uniquely a quantised output value $y(n)$ or y_i could be identified in the waveform for the purpose of associating all the input vectors \mathbf{x}_i mapping into the same phase state. In most applications it was sufficient to use $m = 1$ with the $y(n)$ sample quantised to one of N uniform levels and the $y(n-1)$ to N_S levels (usually $N = N_S$ but not necessarily).

The training of the GRNN and the MPNN Method A can be achieved by taking a long section of the process and identifying the exact p length window positions where the NUM class reference waveform vectors are located. A sequence of training vectors \mathbf{x}_j can be collected by starting at the beginning and sliding one point at time through the whole training signal. The training set of {input -> output} vector pairs are created as follows:

{ $\mathbf{x}_j \rightarrow (\mathbf{x}_j^T \mathbf{x}_{ref i}) / (\mathbf{x}_{ref i}^T \mathbf{x}_{ref i})$ | $j = 1, \dots, NS$ } if \mathbf{x}_j overlaps $\mathbf{x}_{ref j}$, else { $\mathbf{x}_j \rightarrow 0$ | $j = 1, \dots, NS$ }, assuming only one $\mathbf{x}_{ref i}$ overlaps with \mathbf{x}_j at a time.

After the GRNN or MPNN has been trained and optimised it produces the normalised linear correlation output with respect to nearest matching template to the input vector \mathbf{x} . Therefore, although the templates may not be linearly dependent they are non linearly dependent and this dependence becomes quantified by the nonlinear correlator output values $r(\mathbf{x})$. After training this nonlinear correlator can be used for the detection of class vectors if the output exceeds a preset detection threshold θ and their location in time by their peak output.

An application based on the detection of regular tone bursts transmitted over a poor and noisy radio channel was contrived to test the MPNN/GRNN Nonlinear Correlator described above. It was assumed that the tone frequency, amplitude fading and phase effects were unknown and were considered to be normal intra-class variations. Only the burst length was fixed and known. This application tested the Nonlinear Correlator for signal detection and location in time and it was compared with the best linear correlator solution since there was significant linear correlation between the tone bursts. The variable tone burst signal in this case can be thought of as a deterministic signal with some intra-class variance due to the fading effects plus random zero mean white noise and impulse noise.

Two independent data sets, a training and testing set, each having 710 known vector pairs were constructed by sampling synthesised signals at a sampling frequency of 1.667 KHz (0.426 seconds). Each signal had 8 tone bursts at a frequency of 400 Hz subjected to amplitude fading effects. The tones had a starting phase between 0 to 2π and were 30 points long (18.0 ms) with noisy gaps of 61 points (36.6 ms) in between. The noise was added to the whole signal and consisted of zero mean Gaussian noise plus impulse noise with random amplitude and random locations. The input vector dimension was $p=30$ which represented the length of each tone. Each discrete nonlinear correlator output sample point $y(n)$ mapped from the input vector \mathbf{x} made up of 30 discrete time series sample

points of the input signal which included the current sample point $x(n)$ and the 29 sample points immediately prior to it, ie.

$$\mathbf{x} = [x(n), x(n-1), x(n-2), \dots, x(n-p-1)]^T, \text{ vector.}$$

The data sets used in this application were as follows:

training data, NS = 710, p = 30, K = 1, delay = 0.

testing data, NS = 710, p = 30, K = 1, delay = 0.

Figure 2 shows the testing input or source signal and the desired or normalised nonlinear correlator output signal. The training signal pair was another independent signal pair which looked very similar to the testing signal pair. All the peak nonlinear correlator outputs were 1.0.

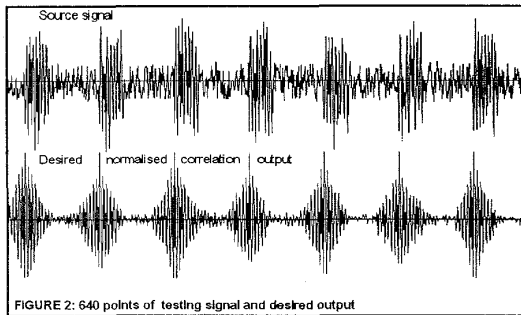


FIGURE 2: 640 points of testing signal and desired output

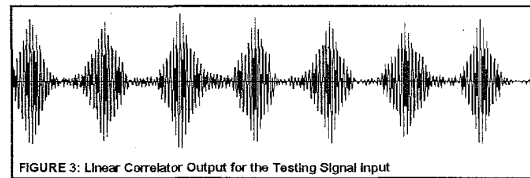


FIGURE 3: Linear Correlator Output for the Testing Signal Input

The peak outputs of the desired normalised correlation output all had the same amplitude according to the nonlinear correlator model defined above. When the source signal was passed through a linear correlator with a matching template made from a 18 ms long sine burst, having a frequency of 400 Hz and constant amplitude the waveform shown in Figure 3 resulted. The peak output was 0.8380 which was less than the maximum possible value of 1.0. The individual peak amplitudes varied according to the varying amplitudes of the tones in the source signal which were subject to the fading effects. A similar output was achieved when the source signal was filtered by a 30 tap finite impulse response (FIR) filter trained with the same training data. This FIR filter actually became a linear correlator so it was no surprise that its output and performance was very similar.

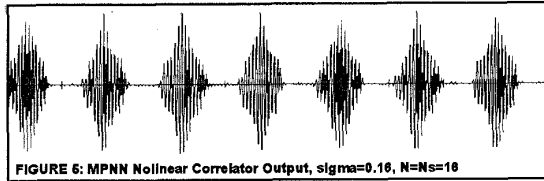
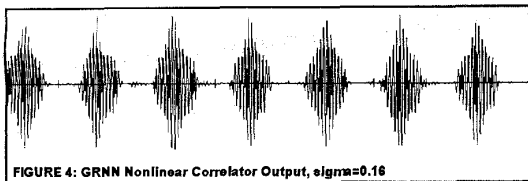
FIR FILTER (1 st Layer = 30 nodes, Last = 1 node)				
Data set	mse	iterations	training time	execution time/vector
Training	0.000843	50000	81.95 seconds	
Testing	0.156837			0.0007042 seconds

The times in the tables are obtained using Borland C 3.0 implementations on an 80486 PC running at 33 MHz, and they are quoted for reference only. A GRNN was trained to implement the nonlinear correlator and the following results were achieved.

GRNN Equation (1)			
mse	σ	training time	execution time/vector
0.167257	0.08	231.18 seconds	0.3256 seconds
0.166078	0.16	244.59 seconds	0.3449 seconds
0.153585	0.32	244.63 seconds	0.3445 seconds
0.078727	0.64	244.64 seconds	0.3446 seconds
0.053380	1.28	241.62 seconds	0.3403 seconds

Training for both the GRNN and MPNN usually involves the selection of a single optimal parameter σ which gives the minimum mean square error (mse) between the desired correlator outputs and the actual network outputs as the whole testing set of data is passed through the network. The training data set is incorporated into the network architecture, either directly as for the GRNN or indirectly as

for the MPNN. In this case the mse approached a minimum as $\sigma \rightarrow \infty$ but it resulted in a degenerate solution. As $\sigma \rightarrow \infty$ the output approached a constant output irrespective of the input x . The mse measure therefore, was not an appropriate one for this problem. It was better to seek a minimum absolute error between the output and the desired response. It was more important to ensure that the peak outputs remained close to 1.0 rather than let them degenerate for the sake of an overall low mse. Considering the nature of the mapping, it would be best to seek the response of the nearest training vector to x . This implies that a lower σ which produced more of a nearest neighbour solution was most appropriate for the GRNN (and the MPNN). A $\sigma = 0.16$ was chosen even though the mse was relatively high because it produced the best output waveform with the peaks at the right height and at the right time. The mse was relatively higher than expected mainly due to the fact that there were relatively few training vectors compared to the possible variety that they could achieve from that process.



The peak output achieved by passing the testing signal through the GRNN was 0.9998 which showed it was working correctly. Another benefit of the GRNN result was that the gap noise was quite low, lower in fact than that present in the training data, as can be seen in Figure 4. This indicated that the GRNN was providing some extra beneficial noise smoothing.

The MPNN Method A implementation of the nonlinear correlator gave similar results as for the GRNN, as can be seen in Figure 5, but with a smaller or more efficient network size. It provided more noise smoothing as expected. The peak output from the testing signal was 0.9997. The results for a few different quantisation selections were as follows.

Data MPNN Method A, Equation (2), m=2						
M	N	N _S	mse	σ	training time	execution time per vector
403	32	32	0.166718	0.16	139.67 secs	0.19672 seconds
249	16	16	0.167216	0.16	86.29 secs	0.12153 seconds
117	8	8	0.166750	0.16	40.59 secs	0.05717 seconds

Conclusions

Both the GRNN and MPNN Method A produced better results than the linear method and thus demonstrate their suitability for the intended class of signal detection applications. The method would be more effective if more training vectors were used but there was sufficient evidence to indicate that the method was valid.

- [1] Makhoul, J., Roucos, S. and Gish, H., "Vector quantisation in speech coding", Proceedings of the IEEE, Vol. 73, No. 11, November 1985, pp. 1551-1588.
- [2] Zaknich, Anthony, deSilva, Christopher and Attikiouzel, Yianni, "The probabilistic neural network for nonlinear time series analysis", IEEE International Joint Conference on Neural Networks (IJCNN), Singapore, 17-21st November 1991, pp. 1530-1535.
- [3] Zaknich, Anthony and Attikiouzel, Yianni, "Automatic optimization of the modified probabilistic neural network for pattern recognition and time series analysis", Proceedings of the First Australian and New Zealand Conference on Intelligent Information Systems, Perth, Western Australia, 1-3rd December, 1993, pp. 152-156.
- [4] Specht, D. F., "A general regression neural network", IEEE Transactions on Neural Networks, Vol. 2, No. 6, November 1991, pp. 568-576.
- [5] Specht, D. F., "Probabilistic neural networks", Neural Networks, Vol 3, 1990, pp. 109-118.
- [6] Zaknich, Anthony and Attikiouzel, Yianni, "Time Series Characterisation Schemes for the Modified Probabilistic Neural Network", Australian Journal of Intelligent Information Processing Systems, Vol. 2, No. 2, Winter 1995, submitted for publication.