Zaknich, A. and Attikiouzel, Y. (1997) A modified probabilistic neural network signal processor for nonlinear signals. In: 13th International Conference on Digital Signal Processing Proceedings, DSP 97, Santorini, Greece, pp. 291-294.

A Modified Probabilistic Neural Network Signal Processor for Nonlinear Signals

Anthony Zaknich, Member IEEE, Associate Member AES
Yianni Attikiouzel, Fellow IEEE, Fellow IEE, Fellow IEAust
Email: tonko@ee.uwa.edu.au
Centre for Intelligent Information Processing Systems (CIIPS),
Department of Electrical and Electronic Engineering,
The University of Western Australia, Nedlands 6907, Western Australia.

Abstract - *This paper introduces a practical and very effective network for nonlinear signal processing called the Modified Probabilistic Neural network. It is a regression technique which uses a single radial basis function kernel whose bandwidth is related to the noise statistics. It has special advantages in application to time and spatial series signal processing problems because it is constructed directly and simply from the training signal waveform features. A sonar signal processing problem is used to illustrate its operation and to compare it with some other filters and neural networks.*

I. INTRODUCTION

The Modified Probabilistic Neural Network (MPNN) is introduced to offer the practicing engineer a relatively easy solution to traditionally difficult nonlinear signal processing problems. It is a general regression technique having a sound statistical foundation in Bayesian estimation theory. It uses a radial basis function kernel with a single bandwidth related to the local noise variance. Its design parameters are simply derived from the training signal waveform characteristics and noise statistics. The motivation for its invention has been to develop a unique and practical ANN for application to a broad class of nonlinear signal processing problems which can be realised with present or foreseeable technology.

II. THE MODIFIED PROBABILISTIC NEURAL NETWORK

The modified probabilistic neural network was initially introduced by Zaknich et al in 1991 [1]. It is closely related to Specht's general regression neural network [2] and both are related to Specht's Probabilistic Neural Network (PNN) [3] classifier. The method of the basic MPNN/GRNN has similarities with the method of Moody and Darken [4]; the method of Radial Basis Functions (RBFs) [5]; the Cerebellar Model Arithmetic Computer (CMAC) [6]; and a number of other nonparametric kernel based regression techniques stemming from the work of Nadaraya [7] and Watson [8]. A standard version of the GRNN equation which is similar to the Nadaraya and Watson equations is equation (1).

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^{NS} y_i \exp\frac{-(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)}{2\sigma^2}}{\sum_{i=1}^{NS} \exp\frac{-(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)}{2\sigma^2}} \qquad (1)$$

If the y_i are individual real valued scalars, equation (1) is exactly Specht's GRNN which incorporates each and every training vector pair $\{\mathbf{x}_i \rightarrow y_i\}$ into its architecture ($\mathbf{x}_i$ is a single training vector in the input space and y_i is the associated desired scalar output). If it can be assumed that for each local region in the input space represented by a centre vector $\mathbf{c}_i$ there is a corresponding scalar output y_i that it maps into then a convenient general model to use for all forms of the MPNN and even the GRNN is:

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^{M} Z_i y_i \exp\frac{-(\mathbf{x}-\mathbf{c}_i)^T(\mathbf{x}-\mathbf{c}_i)}{2\sigma^2}}{\sum_{i=1}^{M} Z_i \exp\frac{-(\mathbf{x}-\mathbf{c}_i)^T(\mathbf{x}-\mathbf{c}_i)}{2\sigma^2}} \qquad (2)$$

$\mathbf{c}_i$ centre vector for class i in the input space.
σ learning parameter chosen during training.
y_i output related to $\mathbf{c}_i$ (real valued or quantised).
M number of unique centres $\mathbf{c}_i$.
Z_i number of vectors $\mathbf{x}_j$ associated with centre $\mathbf{c}_i$.
NS total number of training vectors (Sum of Z_i).

Equations (1) and (2) are both expressed using a Gaussian radial basis function $f_i(x)$ as defined in equation (3). There are many other suitable radial basis functions which can be chosen in place of the Gaussian function as is also the case for the PNN [3].

$$f_i(\mathbf{x}) = \exp\frac{-(\mathbf{x}-\mathbf{c}_i)^T(\mathbf{x}-\mathbf{c}_i)}{2\sigma^2} \qquad (3)$$

The MPNN equation (2) is derived from the GRNN equation (1) using the approximation of equation (4).

$$Z_i \exp\frac{-(\mathbf{x}-\mathbf{c}_i)^T(\mathbf{x}-\mathbf{c}_i)}{2\sigma^2} \approx \sum_{j=1}^{Z_i} \exp\frac{-(\mathbf{x}-\mathbf{x}_j)^T(\mathbf{x}-\mathbf{x}_j)}{2\sigma^2} \qquad (4)$$

This is a reasonable approximation if the $\mathbf{x}_j$ are close enough in the input vector space. They are close enough if they can be adequately represented by a single centre vector $\mathbf{c}_i$ computed as the mean of all the $\mathbf{x}_j$ in that local space. 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 in each class i. One solution for simple sinusoidal signals was proposed by Zaknich et al [1]. It involved uniformly quantising the noiseless desired y_i, separately grouping the y_i having positive and negative slopes in the waveform and associating 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. This was called the MPNN Method A [9]. It involved taking the desired output waveform y(t) and uniformly sampling it in time to digital sample points y(n) quantised into one of N discrete and uniform levels. With this scheme it was possible to define a phase state vector $(y(n),y(n-1),...y(n-m-1))^T$ composed of each quantised y(n) and the m-1 quantised samples immediately preceding it. As a consequence of the quantisation the phase state represented by each of these vectors defined a specific hyper-cube region of the m-dimensional output space. The greater the m and the larger the N the more uniquely a quantised output scalar value y(n) or y_i could be identified in the waveform by its past context. All the input vectors $\mathbf{x}_j$ mapping into the same output phase state vector $(y(n),y(n-1),...y(n-m-1))^T$ were averaged together to compute $\mathbf{c}_i$ which was paired with the desired output value $y(n) = y_i$. In most applications it was sufficient to use m = 1 with the y(n) samples quantised to one of N uniform levels and the y(n-1) samples to N_s levels (usually $N = N_s$ but not necessarily). Method A produced efficient networks but it could not be used where there was more than one local region in the input vector space which mapped into the same output phase state. In that case the mean $\mathbf{c}_i$ of the input vectors $\mathbf{x}_j$, did not adequately represent a local region of those vectors.

To solve this problem a further MPNN Method B was developed. It involved uniquely identifying only those vectors in local hyper-cube regions of the input vector space that mapped to given quantised outputs $y(n) = y_i$. To achieve this the noisy input vector elements were uniformly sampled and quantised to create a phase state vector $(x(n),x(n-1),....x(n-p-1),y(n))^T$ which included both input and output information. In this case the input vector quantisation was used for grouping purposes only. The means of the vectors in each phase state associated with a given quantised value $y(n) = y_i$ were computed using the original data precision. This latter method reduced to an efficient realisation of a quantised version of the GRNN. It automatically clustered close vectors in the input space that mapped to the quantised outputs y_i by simply choosing suitable quantisation parameters. The input quantisation N_x could typically be made more coarse than the output quantisation N. The more coarse the output quantisation was for both Methods A and B the smaller the network became ie. small M. However, the quantisation increment could not be allowed to increase to proportions which were significantly greater than the expected residual noise in the network output under normal operating conditions. If it was, then the output would become dominated by quantisation error rather than residual random error.

There are three types of output error variances represented by the mean squared error (mse) between then desired and actual network output; interpolation error (function mapping error), error due to y_i quantisation and residual random noise error at the output. The total error at the output is the sum of these errors. A useful performance measure for the general network is the signal to noise ratio gain of the output with respect to the input of the network defined as (S/N GAIN) = (S/N at output) / (S/N at input). For the general case dominated by residual random noise it can be shown that $1 \leq$ (S/N GAIN) $\leq$ p, where p is the input vector dimension.

Given a training data set and an independent testing data set both equations (1) and (2) are trained or optimised by the selection of a single smoothing parameter σ which represents the common radial basis function kernel bandwidth. For many problems the σ is directly related to the error or noise distribution in the input signal. In most applications there is a unique σ that produces the minimum mse between the network output and the desired output for the testing set and can be found quite easily by trial and error. Since the relation between σ and mse is usually smooth with a broad minimum mse section σ can be found very quickly by a convergent optimisation algorithm based on recurrent parabolic curve fitting [10].

The method of Moody and Darken [4] is quite similar to the method of RBFs, CMAC, GRNN and the nonparametric kernel based regression techniques of Nadaraya and Watson. The MPNN has a similar formulation as the Moody and Darken method which also uses radial basis function kernels. It is most similar to the GRNN in that a single common kernel bandwidth is tuned to achieve optimal learning. The main difference and key contribution of the MPNN over the GRNN is the clustering together of close training vectors into a finite number of centre vectors according to a very simple procedure related to signal quantisation. A number of equally sized radial basis functions are placed at each and every centre vector location which is then associated with a desired output value. The number of radial basis functions at each centre equals the number of vectors that are represented by that centre. In this way, the number of training vector pairs are reduced without changing the basic form of the network from that of the GRNN. The learning is done in exactly the same way as for the GRNN by selecting the optimal single radial basis function bandwidth which produces the lowest mse. This MPNN vector reduction approach improves computational efficiency and performance. It allows a fixed hardware design since the maximum number of training centre and desired output pairs is finite and definable according to the quantisation chosen.

III. APPLICATIONS OF THE MPNN

A number of applications of the MPNN to signal processing problems have already been published. These include the equalisation of a nonlinear frequency modulated communication channel [1], the enhancement of noisy short wave radio time and morse code signals [11], the nonlinear correlator detector [12] and the removal of impulse plus wide band random noise from a speech signal [13]. Another application described below is the detection of Doppler shifted chirp signals.

A matched filter detector is often used to detect sonar signals, however, this can be very unsatisfactory if the signals are monitoring fast moving objects like torpedos. Fast moving sonar targets cause a significant Doppler effect which frequency shifts the signals. A correlator detector matched to an unshifted signal will produce both amplitude and delay errors when detecting shifted signals. This problem can be addressed in a number of ways. It is possible to have multiple parallel correlators matched to a number of templates spread uniformly over the expected range of the shift. The correlator producing the highest peak output above the detection threshold is taken to give the best detection estimate. It is also possible to train an MPNN/GRNN or some other nonlinear filter or ANN by supervised training to learn the relationship between shifted signal inputs and desired correlator outputs.

To demonstrate this application some digitally sampled signal waveforms were simulated to produce training, testing and validation data sets; Train 1, Train 2, Test 1, Val 1 and Val 2. A sampling frequency of 48 KHz was used. Each input waveform was composed of a sequence of 1 ms chirps linearly swept from 4 KHz to 8 KHz and subjected to Doppler shifts ranging from $\delta v/c$ = -0.2 to +0.2 (δv is the change in signal velocity due to the moving target and c is the speed of sound). The output waveforms were ideal matched filter outputs as would be produced by sliding a non Doppler shifted correlation template past the chirps, assuming they were not Doppler shifted. The template chirp was 48 sample points in length, the same as the input vector **x** dimension ie. the detection aperture window. From these signals sets of input/output pairs were extracted as summarised in Table I.

TABLE I

Data Set	x/y-dims.	No. pairs	x-S/N	No. Chirps
Train 1 (MPNN)	48/1	1974	∞	21
Train 2 (Others)	48/1	3854	4.3dB	41
Test 1	48/1	1880	4.3dB	20
Val 1	48/1	1128	∞	12
Val 2	48/1	1128	7.8dB	12

Figure 1 and Table II show the comparative results of various methods used to solve this problem; the multiple correlator, the quadratic filter, Multi-layer Perceptron (MLP), RBF and MPNN (method B, N_X=2 and N=32000) filters. The filters were all trained via supervised training using a training set and then tested with the remaining sets as shown. The times quoted in Table II are for software implementations of the various networks all written in Borland C 4.5, compiled for the Windows 32 bit environment and running on a Pentium 90 PC.

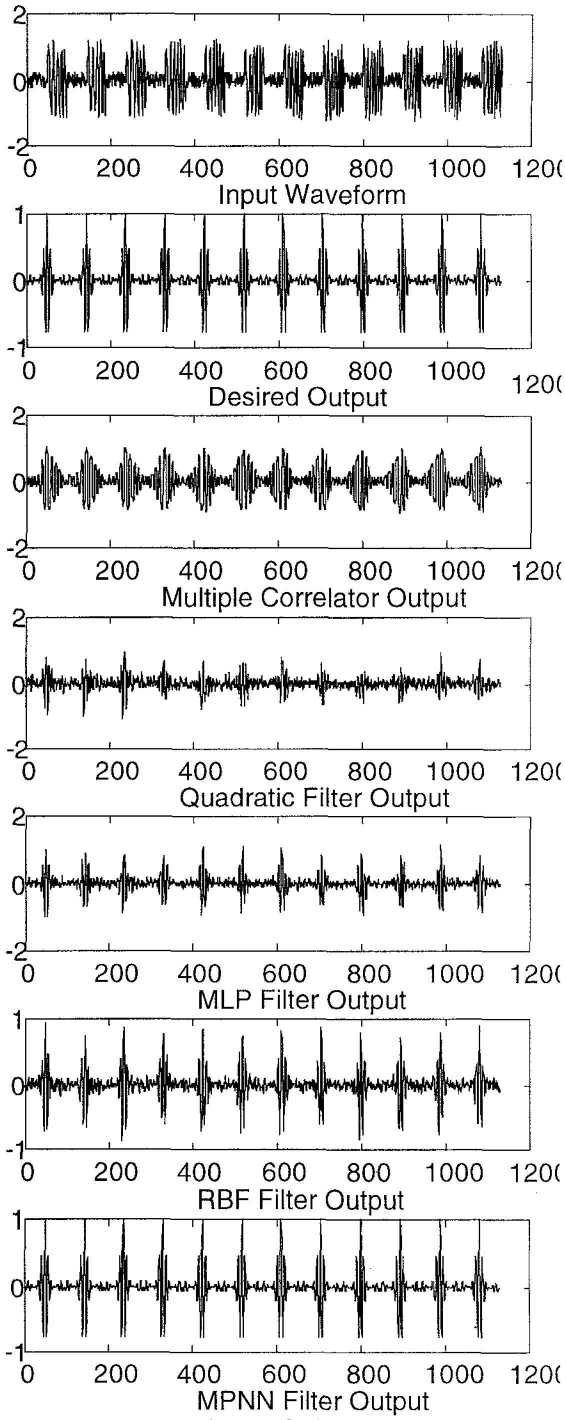

Fig. 1 Val 2 Data

TABLE II

Net. Type	Data Set	Train iter./ passes	mse/2 $x10^{-6}$	σ	Train time in sec	Vect. eval. in sec
Multiple Correlator 21 templ.	Train 1 Train 2 Test1 Val 1 Val 2	0	- 97528 100793 94147 94695	- - - --	- 0	- 0.00067
Quad $48/48^2/1$	Train 2Test1 Val 1 Val 2	12 $x10^6$	7312 7893 6189 22263	- - - -	35x10 5	0.00102
MLP 48-48-1	Train 2Test1 Val 1 Val 2	$36x10^6$	111 5 184 480	- - - --	$16x10^5$	0.00125
RBF 48-750-1 Gauss RBF	Train 2Test1 Val 1 Val 2	1	342 470 171 302	- - - --	77x10- 2	0.07410
MPNN Meth. B 48-1656-1 Gauss RBF	Train 1 Test1 Val 1 Val 2	1	- 866 39 63	- 0.32 0.32 0.32	- 131	- 0.06945

The MPNN exhibits superior performance in all respects except vector execution times. However, for parallel hardware realisations the vector execution times are not an issue. The MLP performs quite well for relatively low noise signals but it requires a considerable training time and it does not generalise well when the signal noise increases. One of the main advantages of the MPNN in this and other applications is that it can be trained with noiseless training data and still perform very well with noisy signals. As the noise increases in the input signal simple adjustments of σ are necessary to maintain an optimal performance. This is so because the optimum σ^2 is related to the input noise variance since each radial basis function models the local input noise pdf. The mse vs σ is relatively flat over a wide range of σ so the same σ can be used over a wide range of noise variance with minimal error. To perform adequately, the MLP and the other ANNs must be trained over the full range of expected noise levels because their parameters are not directly related to the noise statistics. This is a feature of the MPNN makes it extremely useful for most practical design problems where noise levels can vary considerably during operation. In respect to peak output and detection delay errors all the neural networks perform fairly well and better the quadratic filter or the multiple correlator with some degradation with noise. It is interesting to note that the RBF performs better than the MLP in respect to both peak and delay errors as noise is added and the MPNN performs almost perfectly under all noise levels tested.

VI. CONCLUSIONS

This paper has introduced the MPNN and described its relation to Specht's GRNN and other nonparametric kernel based regression techniques. The MPNN/GRNN has many practical advantages over other neural networks in respect to both design and application to nonlinear signal processing. The main features of the MPNN are that it:

1) is based on well established Bayesian theory,
2) is easy to understand and use,
3) has a quick and certain training mechanism,
4) is relatively insensitive to random noise,
5) is easy to build in hardware.

REFERENCES

[1] 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.

[2] Specht, D. F., "A general regression neural network", IEEE Transactions on Neural Networks, Vol. 2, No. 6, November 1991, pp. 568-576.

[3] Specht, D. F., "Probabilistic neural networks", International Neural Network Society, Neural Networks, Vol 3, 1990, pp. 109-118.

[4] Moody, J. and Darken, C., "Learning localiszed receptive fields", Proceedings of the 1988 Connectionist Models Summer School, Editors Touretzky, Hinton and Sejnowski, Publishers Morgan-Kaufmann, 1988, pp. 133-143.

[5] Broomhead, D. S., Lowe, D., "Radial basis-functions, multi-variable functional interpolation and adaptive networks", Royal signals and radar establishment Memorandum 4148, March 28 1988.

[6] Miller, W. T., Glanz, F. H. and Kraft, L. G., "CMAC: An association neural network alternative to backpropagation", Proceedings of IEEE, Vol 78, No. 10, October 1990, pp. 1561-1567.

[7] Nadaraya, E. A., "On estimating regression", Theory Probab. Appl., 9, 1964, pp. 141-142.

[8] Watson, G. S., "Smooth regression analysis", Sankhya Ser. A, 26, 1964, pp. 359-372.

[9] Zaknich, Anthony and Attikiouzel, Yianni, "Time series characterisation schemes for the modified probabilistic neural network", Australian Journal of Intelligent Information Processing Systems (AJIIPS), Vol. 2, No. 2, Winter 1995, pp. 1-11.

[10] 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.

[11] Zaknich, Anthony and Attikiouzel, Yianni, "Application of the modified probabilistic neural network to the enhancement of noisy short wave radio time and morse code signals", Australian Journal of Intelligent Information Processing Systems (AJIIPS), Vol. 2, No. 3, Spring 1995, pp. 9-14

[12] Zaknich, Anthony and Attikiouzel, Yianni, "The modified probabilistic neural network as a nonlinear correlator detector", IEEE Proceedings of the International Conference on Neural Networks (ICNN), Perth, Western Australia, 1-3rd December, 1995, pp. 309-313.

[13] Zaknich, Anthony and Attikiouzel, Yianni, "Application of artificial neural networks to nonlinear signal processing", IEEE Press, Chapter in "Computational intelligence: a dynamic systems perspective", 1995, ISBN 0-7803-1182-5, pp. 292-308.