Young, J.P., Zaknich, A. and Attikiouzel, Y. (2001) Center reduction algorithm for the modified probabilistic neural network equalizer. In: Proceedings of the International Joint Conference on Neural Networks, IJCNN '01, 15 - 19 July, Washington, DC, USA, pp. 1966-1970.

Center Reduction Algorithm for the Modified Probabilistic Neural Network Equalizer

James P. Young, Anthony Zaknich and Yianni Attikiouzel

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

Abstract

The applicability of the Modified Probabilistic Neural Network (MPNN) to channel equalization can be severely limited by the size of the network. The size of the network grows exponentially with the order of the channel and the dimension of the input vectors. As a result, the standard network is practical only for low order channels with small input alphabet size. An algorithm is proposed to alleviate such an undesirable constraint by finding a much smaller network representation with a similar decision surface.

1. Introduction

Many neural network based adaptive non-linear filters have been studied and have demonstrated superior performances (measured in error probability) over linear filters for equalizing high-speed digital communication channels. However, the improvement in performance comes with a price of increased complexity and added restrictions. For example, the Multi-layer Perceptron (MLP) is limited by its long training time and the possibility of convergence to undesirable local extremas [1]. The Recurrent Neural Network (RNN) with Real-time Recurrent Learning achieves a reasonable convergence speed [2], but the computational complexity increases exponentially with the network size. Like the MLP, it may converge to local extremas as well. The Radial Basis Function Network (RBFN) on the other hand has faster training and has better performance in terms of bit error rate (BER) [3]. However it can be used only for lower order channels with smaller signal alphabet size to keep the network to a manageable size.

The focus of this paper is on the Modified Probabilistic Neural Network (MPNN). The MPNN approximates an optimal Bayesian solution. It has a similar network structure to the RBFN, and therefore suffers from a dimensionality problem as well. The network size needs to be kept small. This restriction makes it unacceptable for some practical applications.

This paper introduces a method to reduce the network size of the MPNN while maintaining the advantageous properties of the MPNN.

2. The MPNN As An Equalizer

The MPNN was developed by Zaknich et al [4,5], inspired by Specht's work on Probabilistic Neural Network (PNN) [6]. The work was done in parallel with Specht's other research, later published as the General Regression Neural Network (GRNN) [7]. The MPNN and the GRNN share the same theoretical background and basic network structure. The difference is that MPNN uses a k-means like clustering method for the formation of its centers. The clustering technique effectively reduces the number of centers by grouping together centers that are close to each other. In channel equalization, clustering reduces the effect of noise.

2.1. Problem Formulation

Figure 1 depicts a model of a digital communication system. Consider that an i.i.d. M-ary symbol sequence $\{\mathbf{S_n}\}$ is being transmitted through a dispersive channel **h**. When the channel transfer function is linear, the output of the channel is the convolution of the input sequence $\mathbf{S_n}$ with **h** ($\mathbf{u = S_n * h}$). For non-linear channels, the output of the channel is $\mathbf{u = h(t, S_n)}$. The output is further corrupted by zero-mean Gaussian noise **v**. The output of the channel is $\mathbf{w = u + v}$. The last m sequences from **w** is taken as the input to the MPNN equalizer to restore the original signal.

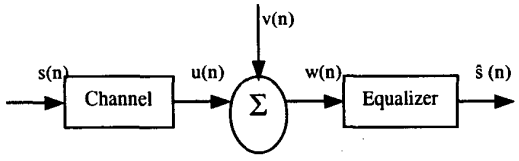

Figure 1: Discrete time model of data transmission system

0-7803-7044-9/01/$10.00 ©2001 IEEE

When the channel characteristic is of a non-linear nature, a non-linear equalizer is necessary to efficiently equalize the distorted signals.

2.2. The MPNN Structure

The training of the MPNN, unlike the MLP, the RNN or the RBFN, is not a gradient descent based training. The training is memory based and requires only a one-pass training. During training, the network stores the training input vectors as centers of the network. These centers are assigned the values of the desired output.

During classification, the "closeness" of a test input vector to each of the centers is determined via the kernel function. The output associated with the center closest to the input vector hence is the most likely output. It is said, therefore, that the network approximates an optimal Bayesian solution.

In equalization, the input, **x**, is taken as the last m sequence of the channel output.

$$\mathbf{x}_i = [w_i \quad w_{i-1} \quad \cdots \quad w_{i-m}]^T \tag{1}$$

Suppose y is the output to be estimated. Using a statistical model, vector **x** can be treated as random variable with a conditional probability of **x** given y. During training the *a priori* distribution of y is specified based on what is known about y. It is then possible to obtain the conditional distribution of y given **x**. During classification phase, current data updates are used to yield posterior information about y.

Specht [7] has given the general regression equation of the scalar output y given an input vector random variable **x** as expressed by (2).

$$y(\mathbf{x}) = E[y \mid \mathbf{x}] = \frac{\int_{-\infty}^{\infty} yp(\mathbf{x}, y)dy}{\int_{-\infty}^{\infty} p(\mathbf{x}, y)dy} \tag{2}$$

Where $p(\mathbf{x}, y)$ is the joint continuous probability density function. This is estimated *a posteriori* from the training set $(\mathbf{x}_n, y_n)$.

The equation is derived via a semiparametric technique using the Parzen-Rosenblatt density estimator [8]. The key to this formulation is the kernel function, which has the underlying properties associated with a probability density function. The final estimator equation is given in (3), which includes a clustering variable Z_i. Without center clustering, the equation is also known as the Nadaraya-Watson Regression Estimator [8]. The kernel function shown in (4), has been formulated to accommodate for complex input vectors and complex outputs.

With the m-dimensional complex vector input **x** and complex output y implementing a mapping $\{\mathbf{f} : C^N \rightarrow C\}$, the transfer function is

$$\hat{y}(\mathbf{x}) = \frac{\sum_{i=1}^{M} Z_i y_i \Phi_i(\mathbf{x}, \mathbf{c}_i, \sigma)}{\sum_{j=1}^{M} Z_j \Phi_j(\mathbf{x}, \mathbf{c}_j, \sigma)} \tag{3}$$

Where the Kernel function is

$$\Phi_i(\mathbf{x}, \mathbf{c}_i, \sigma) = \exp\left\{\frac{-[\mathbf{x}-\mathbf{c}_i]^H[\mathbf{x}-\mathbf{c}_i]}{2\sigma_i^2}\right\} \tag{4}$$

- **x** $\mathbf{x} \in C^N$ is the input vector to the network with complex members
- $\mathbf{c}_i$ $\mathbf{c}_i \in C^N$ is the center or mean vector for class i in the input space
- σ_i sigma is a real valued smoothing parameter
- y_i $y_i \in C$ is the weight / output relating to center c_i
- M is the total number of unique centers
- Z_i is the number of input training vector associated with center $\mathbf{c}_i$
- $[.]^H$ denotes Hermitian operator (or conjugate transpose)

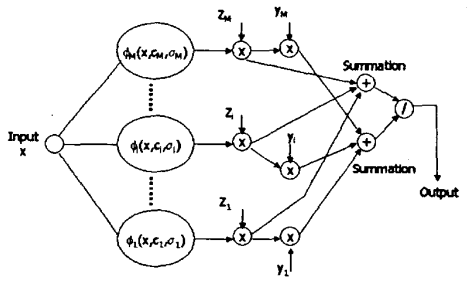

Figure 2: Basic MPNN Architecture

The output of each of the kernel functions is real. The complex MPNN equation is the same as the real MPNN equation in [4,5], except that the Hermitian operator is used inside the kernel function instead of a vector transpose operator. The network output is the sum of the product from each center normalized by the sum of the output of the basis function scaled by Z_i.

In general, for very small σ, the network functions as a nearest neighbor classifier. For a very large number σ, the network functions as a matched filter.

2.3. Unsupervised Clustering of Centers

The clustering method is similar to the k-means clustering. Close centers that are mapped to the same output are clustered together to form one new center. The location of the new center is the mean of all the centers

being clustered. The total number of centers belonging to this new center is Z_i. As a result, a smaller network size can be realized

Learning involves the clustering of centers $\mathbf{c}_i$ and the assignment of an appropriate weight y_i and smoothing parameter σ_i to each center. A slight variation to Zaknich's clustering technique [9] is used for the positioning of centers.

Define a parameter, Rx, called the radius of influence. Training starts with the first training pair $\{\mathbf{x}_i, y_i\}$ and establishing a center $\mathbf{c}_i$ at $\mathbf{x}_i$ with a corresponding weight of y_i. Given n training data pairs, the pseudo-code for the clustering method is as follows:

for i = 1: n,
  if
    $([\mathbf{x}_i - \mathbf{c}_i]^H[\mathbf{x}_i - \mathbf{c}_i] \le R_x)$ and $(y_i(k) = y(k))$ (5)
  then
    $Z_i^{new} = Z_i^{old} + 1$ (6)
    $\mathbf{c}_i^{new} = \frac{(Z_i - 1)\mathbf{c}_i^{old} + \mathbf{x}_i}{Z_i}$ (7)
  else
    $Z_i^{new} = 1$
    $c_i = x_i$
  end
end

3. Center Vector Reduction Algorithm

In the case of channel equalization, as well as many other signal processing applications, the contribution of each center to determining the decision surface varies. In many cases, the decision surface can be represented by a smaller set of selected centers. Without loss of generality, Let us consider binary separations. The two classes are class 1 and class 2.

The properties of the radius of influence, Rx, (as discussed in section 2.3) are utilized in the reduction process. As the value of Rx increases, it is observed that the number of centers becomes sparse. As well, the centers from different classes move away from each other. As the gap between the classes widens, the resolution of the decision surface also looses resolution. Figure 3 shows clustering of the same set of two-dimensional input data with varying values of radius of influence, Rx.

Therefore, by choosing only centers closest to the decision surface for varying Rx, we can capture the general form of the decision surface. At the same time the resolution of the decision surface is retained.

The reduction method is as follows:

1. Compose a list of values for Rx in ascending order, beginning with a small value.

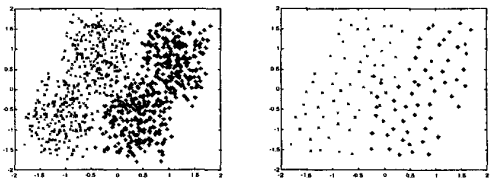

Figure 3a (Rx=0.01),**3b**(Rx=0.05)

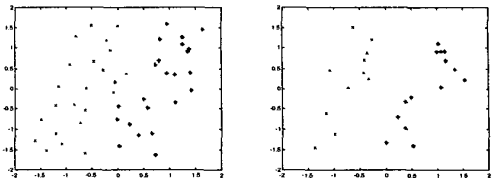

Figure 3c(Rx=0.2),**3d**(Rx=0.5)

Figure 3. Clustering of centers on the same set of data with different Rx value.

2. Cluster the training inputs with the first value of R_x.
3. Choose randomly p centers from class 1. For each $\mathbf{x}_{p1}$, find a center from class 2 closest to it. This closest center, $\mathbf{c}_{k2}$ is the candidate center for the reduced network. Again, for each $\mathbf{c}_{k2}$, find a center closest to it belonging to the class 1, $\mathbf{c}_{k1}$.
4. For remaining centers in class 1 that are not $\mathbf{c}_{k1}$, select it if the distance to the closest $\mathbf{c}_{k1}$ is greater than any center in class 2.
5. Repeat from step 2 and cluster existing centers with the next value of Rx. Stop if all Rx have been tried.

The size of p in step 3 can be arbitrarily selected ranging from 1 to the total number of centers belonging to the same class. The selection of p does not affect greatly the final result. However, choosing extreme values of p (1 or the maximum value) may increase the amount of computation.

The result is a smaller group of centers realizing a decision boundary very similar to the decision boundary of an unreduced network without the requirement of further adaptation of weights.

4. Simulation Results

The channel used in the simulation was sourced from the SPIB database [10]. The database contains FIR models of different digital microwave radio channel impulse responses. Channel 7 was used. The fractionally sampled factor was taken into consideration. The channel was down-sampled to 32 taps. A further non-linearity characteristic was introduced as follows:

$$y(t) = x(t) + 0.2x^2(t) - 0.1x^3(t) + v(t) \quad (8)$$

Figure 4 provides a visualization of the centers chosen in the reduced network for the case when the input vector is of dimension two. The centers belonging to the two classes in the full MPNN are marked by dots and crosses. The circles are the centers in the reduced MPNN.

The dimension of the input vectors in figure 5 was four. The non-linear equalizers were trained with 1000 input vectors. The number of centers in both the full network and the reduced network for different signal to noise levels are tabulated in Table 1. The algorithm achieves better than 1/25 reduction for less noisy channels. The reduction ratio for the 8dB signal to noise ratio case was 1/10. The reduction ratio was not as good for noisy cases because the network tried to generalize the noise characteristics. Noise causes regions of the two classes to overlap. More centers were required to generalize the overlapping regions. Generalizing noise is undesirable.

The selection of radius of influence, Rx, determines the center reduction ratio and the performance of the reduced network. Generally there is a vague trade-off between the reduction ratio and the performance. In this simulation, The Rx values for the each iteration were [0.01, 0.03, 0.05,1]. The more number of iterations and the smaller the gap between subsequent Rx values, the better the performance of the reduced network.

The result shown in figure 5 indicates that the reduction algorithm slightly degraded the performance from the full MPNN solution. However, the much reduced computational requirement justifies the slight loss in performance.

SNR (dB)	8	10	12	14	16
Number of MPNN Centers	941	936	930	894	863
Number of Reduced MPNN Centers	90	59	45	35	34

Table 1. Number of centers in full MPNN and reduced MPNN for different noise level

5. Conclusion

The algorithm has been tested on a realistic high order channel and it has been found that the reduction provides practical implementation for the MPNN. The reduction in network size means faster implementation on a sequential computation machine. Other desirable properties of MPNN such as its fast training and good performance were preserved. The network reduction algorithm offers the MPNN a greater applicability to data analysis problems.

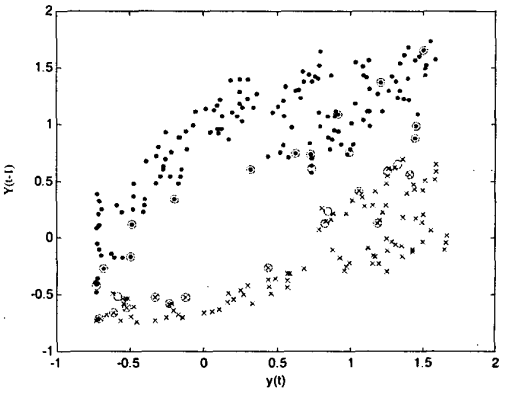

Figure 4: Center of the Full Network and of the Reduced Network

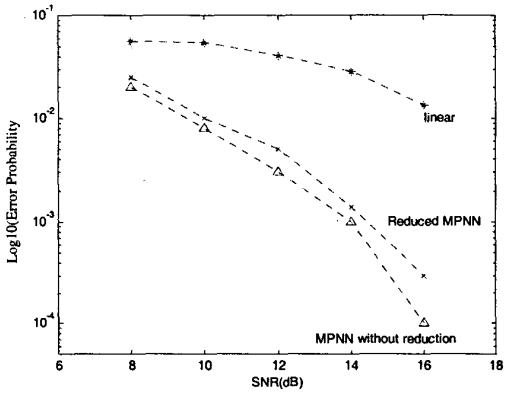

Figure 5: Performance comparison of reduced MPNN with full MPNN and linear filter

6. References

[1] G.Gibson, S.Siu and C. Cowan, "Multilayer Perceptron Structures Applied to Adaptive Equalisers for Data Communications", *in Proc. of ICASSP'89*, pp. 1183-1186, Glasgow, Scotland, May 1989.

[2] G.Kechriotis, E.Zervas and E.S.Manolakos, "Using Recurrent Neural Networks for Adaptive Communication Channel Equalization", *IEEE Trans. Neural Networks*, Vol.5, No.2, March 1994.

[3] S. Chen, B. Mulgrew and P.M. Grant, "A Clustering Technique for Digital Communications Channel Equalization Using Radial Basis Function Networks", *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 570-579, 1993.

[4] A.Zaknich, C.deSilva and Y.Attikiouzel, "The Probabilistic Neural Network for Nonlinear Times Series Analysis", *in Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN)*, Singapore, Nov. 1991.

[5] A. Zaknich, "Introduction to the modified probabilistic neural network for general signal processing applications", *IEEE Transactions on Signal Processing*, Vol. 46, No. 7, pp. 1980-1990, July 1998.

[6] D.F.Specht, "Probabilistic Neural Networks", Int. Neural Network Soc., *Neural Networks*, Vol.3, pp.109-118, 1990.

[7] D.F.Specht, "A General Regression Neural Network", *IEEE Trans. Neural Networks*, Vol.2, Nov. 1991.

[8] S.Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Ed, Prentice-Hall, 1999.

[9] A. Zaknich and Y. Attikiouzel, "An unsupervised clustering algorithm for the modified probabilistic neural network", *IEEE International Workshop on Intelligent Signal Processing and Communications Systems*, Melbourne, Australia, pp. 319-322, 4-6th November, 1998.

[10] The Signal Processing Information Base (SPIB), Rice University. {http://spib.rice.edu}