Neural Filters: MLP VIS-A-VIS RBF Network

V. R. MANKAR,
Head of Electronics Deptt.,
Govt. Polytechnic, Amravati-444 604
INDIA

DR. A. A. GHATOL,
Vice Chancellor,
Dr. B. A. Technological University,
Lonere, Dist. Raigarh. (MS)
INDIA

Abstract:- Filtering of signals is of primary importance in signal processing. The design of filters to perform signal estimation is a problem that freeze up in the design of communication systems, control systems, in geophysics & in many other applications & disciplines. Optimum filters are proposed for filtering. In this paper, neural networks have been trained to filter satisfactorily with specified MSE criterion. It is found that neural networks such as multiplayer perceptron and RBF network comprising of three hidden layers with a linear transfer function elegantly filters various signals under consideration.

Key-Words:- Feed-forward Neural Network, RLS, MSE, MLP, RBF.

1 Introduction

Filtering of signals is a predominantly important topic in Digital Signal Processing with applications in a range of areas such as speech signal processing, image processing and noise suppression in communication systems. The determination of optimum linear filters requires the solution of a set of linear equations that have some special symmetry. To solve these linear equations, Levinson-Durbin Algorithm and Schur algorithm are used which provide the solution to the equation through computationally efficient measures that exploit the symmetry properties. An important class of optimum filter is Weiner filter that is widely used in many applications concerning the estimation of signals corrupted with additive noise [1], [2].

Linear estimation can also be thought of as applying the projection theorem and projecting next value of a random variable X_{n+1} onto the linear manifold generated by the observations $X_{1,..,}X_{n.}$ Clearly, in the vanguard, the only statistical information requisite is the second moment characteristics of the random process. The purpose of the filter is merely to implement the projection operation. The purpose of zero memory non-linearity is to modify the observations in such a way that the resulting linear manifold contain a large component of X_{n+1} [3].

Nonadaptive methods for signal estimation use preset basis functions and their parameters. There are two major classes of nonadaptive methods i.e. global parametric methods such as linear and polynomial regression and local parametric methods such as Kernel smoothers, piecewise linear regression and splines. Local parametric methods are applicable only to low dimensional problem due to inherent sparseness of finite sample in high dimensions. Hence, adaptive methods are only realistic alternative for high dimensional problems [4],[5].

The basic idea behind the filtering is to recognize that the MA process is also finite order auto -regressive process [6]. Linear filters are easy to implement and analyze. Linear filters minimizing the

MSE criterion can usually be found in closed form. They are optimal among the class of all filtering operations when the noise is additive and guassian [7],[8].

2 Linear Filtering

Linear filtering possibly viewed as equivalent to linear prediction, where the prediction is embedded in the linear filter, which is called the error filter. In numerous realistic applications, given an input signal x(n) consisting of the sum of desired signal s(n) and undesired noise or interference w(n), it is attempted to design a filter that suppresses undesired interference components. In such cases, the objective is to design a system that filers out the additive interference while preserving the characteristics of desired signal s(n). The problem of signal estimation is treated in the incidence of an additive noise disturbance. The estimator is constrained to be a linear filter with impulse response h(n) designed so that its output approximates some specified desired signal sequence d(n). The input sequence to the filter is x(n)=s(n)+ w(n) and output sequence is y(n). The difference between the desired signal and the filtered output is the error sequence e(n) = d(n) – y(n).

Three different cases are possible in this manner:

i) If d(n) = s(n), the linear estimation problem is referred to as 'filtering'.

ii) If d (n) = s (n+d), where d>0, the linear estimation problem is referred to as 'signal prediction'.

iii) If d(n) = s(n-d), where d>0, the linear estimation problem is referred to as 'signal smoothing'.

The criterion selected for optimizing the filter impulse response h(n) is the minimization of MSE. This criterion has the advantage of simplicity and mathematical tractability.

3 Neural Network Approach

Neural Networks can be used to attain reasonably good filters in a number of cases, though perfect prediction is hardly ever possible. At a high level, the filtering problem is a special case of function approximation problems in which the function values are represented using time series [10] . A time series is a sequence of values measured over time in the discrete or continuous time units. Multiplayer Perceptron, and RBFs are used for effective filtering [11].

Multilayer perceptrons (MLPs) are layered feedforward networks, typically trained with static backpropagation. These networks have found their way into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than network weights). It also requires specifying number of hidden layers and a number of processing elements (PEs) in a layer.

Radial basis function (RBF) networks are nonlinear hybrid networks[9], typically containing a single hidden layer of processing elements (PEs). This layer uses gaussian transfer functions, rather than the standard sigmoidal functions employed by MLPs. The centers and widths of the gaussians are set by unsupervised learning rules, and supervised learning is applied to the output layer. These networks tend to learn much faster than MLPs.

If a generalized regression (GRNN) / probabilistic (PNN) net is chosen, all the weights of the network can be calculated analytically. In this case, the number of cluster centers is by definition equal to the number of exemplars, and they are all set to the same variance. It is recommended to use the type of RBF neural network, only when the number of exemplars is so small (<100) or so dispersed that clustering is ill-defined.

For standard RBF's, the supervised segment of the network only needs to produce a linear combination of the output at the unsupervised layer, with 0 hidden layers. Hidden Layers can be added to make use of supervised learning, instead of a simple linear perceptron.

Signal filtering from present observations is a basic signal processing operation by use of filters. Conventional parametric approaches to this problem involve mathematical modeling of the signal characteristics, which is then used to accomplish the filtering. In a general case, this is relatively a complex task containing many steps for instance model hypothesis, identification and estimation of model parameters and their verification. However, using a MLP Neural Network, the modeling phase can be bypassed and nonlinear and nonparametric signal filtering can be performed. Normally, three layer Neural Network is selected. As the thresholds of all neurons are set to zeros, unknown variables for one step ahead filtering are only the connection weights between the output neurons and the j^{th} neuron in the second layer, which can be trained by available sample set. The above neural networks are used to realize the linear as well as non-linear mapping filter.

4 Simulations

NeuroSolutions simulations are vector based for efficiency. This implies that each layer contains a vector of PEs and that the parameters selected apply to the entire vector. The parameters are dependent on the neural model, but all require a nonlinearity function to specify the behavior of the PEs. In addition, each layer has an associated learning rule and learning parameters. The number of PEs, learning rule, nonlinearity and learning parameters are to be chosen accordingly.

Learning from the data is the essence of neurocomputing. Every PE that has an adaptive parameter must change it according to some prespecified procedure. Back-propagation is by far the most common form of learning. It is sufficient to say that the weights are changed based on their previous value and a correction term. The learning rule is the means by which the correction term is specified. Once the particular rule is selected, the user must still specify how much correction should be applied to the weights, referred to as the learning rate. If the learning rate is too small, then learning takes a long time. On the other hand, if it is set too high, then the adaptation diverges and the weights are unusable. The good indicator of the level of generalization that the network has achieved is the option of MSE termination, to base the stop criteria on the cross validation set (from the Cross Validation panel) instead of the training set.

Different input signals with different mathematical expressions are filtered out precisely on the basis of 640 values of the signal samples. At any instant of time, the neural network is presented the above values of the signal and it is expected to produce the desired signal. In the case studies considered, the signals with noise limited to 10% and 20% level are inputted to MLP & RBF neural networks and output signal is obtained with mean square error limited to 1% as per the expectations.

The results are tested on Neuro Solutions platform and accordingly, simulations are carried out on noisy input and desired output samples.

5 Results

The neural network containing three hidden layers with 4,5,3 neurons per layers are found to successfully filter out the inputted signals. This is obvious from the outputs of the trained neural networks, which depict how accurately these neural networks filter the given signals, as shown in tables below.

The noisy signals were inputted to Multiplayer Perceptron, and RBF Neural Networks with three hidden layers, and

4,5,3 neurons per layer with input, hidden and output layer with changeable parameters similar to processing elements, transfer function, learning rule, step size and momentum were tested in supervised learning mode with maximum epoch value, 1000. Also, different sets of data obtained by swapping were inputted for testing purpose.

After training the network on a noisy input (10 to 20% of random noise addition) and desired output data values with 640 samples, the expected results were obtained with minimum MSE values around the estimated values as tabulated below.

Table 1: Simulations Results (10% Noise Addition), for swapping pattern of 50% Training, 25% Cross-validation & 25% Testing samples.

Sr. No.	Type of ANN	Minimum MSE Criterion		
		Training	Cross validation	Testing
01	MLP	0.009930101	0.009854664	0.024038153
02	MLP	0.013028802	0.014036612	0.007454009
03	MLP	0.009939991	0.010004657	0.024494328
04	RBF	0.009905203	0.00968616	0.024301478
05	RBF	0.012244184	0.013706509	0.00728931
06	RBF	0.009928582	0.009764279	0.02458356

Table 2: Simulations Results (10% Noise Addition), for swapping pattern of 25% Testing, 50% Training & 25% Cross-validation samples.

Sr. No.	Type of ANN	Minimum MSE Criterion		
		Training	Cross validation	Testing
01	MLP	0.00994094	0.009334081	0.023195417
02	MLP	0.012946235	0.011509923	0.00734905
03	MLP	0.009944581	0.009491608	0.025281703
04	RBF	0.009843346	0.009266302	0.022793361
05	RBF	0.01341804	0.012867227	0.007951809
06	RBF	0.009902088	0.009701799	0.025283572

Table 3: Simulations Results (10% Noise Addition), for swapping pattern of 25% Cross -validation, 25% Testing & 50% Training samples.

Sr. No.	Type of ANN	Minimum MSE Criterion		
		Training	Cross validation	Testing
01	MLP	0.009941116	0.009561446	0.025597691
02	MLP	0.013109853	0.012614921	0.008224264
03	MLP	0.009934663	0.010213525	0.023523104
04	RBF	0.009860078	0.009489396	0.025309394
05	RBF	0.013166751	0.012459507	0.00810629
06	RBF	0.009879913	0.010159144	0.023345567

Table 4: Simulations Results (20% Noise Addition), for swapping pattern of 50% Training, 25% Cross-validation & 25% Testing samples.

Sr. No.	Type of ANN	Minimum MSE Criterion		
		Training	Cross validation	Testing
01	MLP	0.012533969	0.012397650	0.030594911
02	MLP	0.039805160	0.048098271	0.028999019
03	MLP	0.014756491	0.015676318	0.035723699
04	RBF	0.012262594	0.012055276	0.029935802
05	RBF	0.039627605	0.048524020	0.028789946
06	RBF	0.012992455	0.013973765	0.031466171

Table 5: Simulations Results (20% Noise Addition), for swapping pattern of 25% Testing, 50% Training & 25% Cross-validation samples.

Sr. No.	Type of ANN	Minimum MSE Criterion		
		Training	Cross validation	Testing
01	MLP	0.013407133	0.012338001	0.027701788
02	MLP	0.009848792	0.097286640	0.006009123
03	MLP	0.020247144	0.049125920	0.045460251
04	RBF	0.012915127	0.011977816	0.027006061
05	RBF	0.042938473	0.047019371	0.025330869
06	RBF	0.013504270	0.012683790	0.033100865

Table 6: Simulations Results (20% Noise Addition), for swapping pattern of 25% Cross -validation, 25% Testing & 50% Training samples.

Sr. No.	Type of ANN	Minimum MSE Criterion		
		Training	Cross validation	Testing
01	MLP	0.011965466	0.010905661	0.034343768
02	MLP	0.046702725	0.041943637	0.024766963
03	MLP	0.013103811	0.013082653	0.030802857
04	RBF	0.011777550	0.011081549	0.034173635
05	RBF	0.046432112	0.042181771	0.025187737
06	RBF	0.013284690	0.013442181	0.03300092

6 Conclusion

In this paper, it is revealed that a Multiplayer Perceptron and RBF, both the Neural Networks are proficient to filter a noisy signal fairly accurately. The difference between the actual signal and the signal predicted by the neural network is computed as a performance measure (mean square error) and is found to be in expected range.

Also, it is obvious that the minimum MSE criterion is uniformly observed in training, cross validation stages and trained neural network is successfully filtering the signal in testing phase.

References:-

[1] G.H. Hostetter, "Recursive Estimation", Electrical Engineering Deptt. University of California, Irvine, 92717.

[2] Victor Solo Xuan Kong, "Adaptive Signal Processing Algorithms Stability &

Performance", Prentice Hall Englewood Cliffs, NJ 07632, 1995

[3] Vladimir Cherkassky, Don Gehring, and Filip Mulier, "Comparison of Adaptive Methods for Function Estimation from Samples" , IEEE Trans. on Neural Networks, Vol.7, No.4, July 1996, pp. 961-984.

[4] N. Kalouptsidis, S. Theodoridis, "Adaptive System Identification and Signal Processing Algorithms", Prentice Hall Englewood Cliffs, NJ 07632, 1993

[5] Simen Haykin, "Adaptive Filter Theory", Prentice Hall Englewood Cliffs, NJ 07632, 1986

[6] Karim Abed-Merium, Eric Moulines, Philippe Loubaton, " Prediction Error Method for Second Order Blind Identification", IEEE Trans. on Signal Processing, Vol. 45, No. 3, March 1997, pp 694-705.

[7] Charles S. Williams, "Designing Digital Filters", Prentice Hall Englewood Cliffs, NJ 07632, 1986

[8] R. Merched, A. H. Sayed, " Order Recursive RLS Laguerre Adaptive Filtering", IEEE Transactions on Signal processing, Vol 48, No. 11, Nov 2000, PP 3000-3010.

[9] Lin Yin, Jaakko Astola, Yrjo Neuvo, " A New Class of Nonlinear Filters- Neural Filters" , IEEE Trans. on Signal Processing, Vol. 41, No.3, March 1993, pp. 1201 –1222.

[10] Wai-Kuen Lai, P.C. Ching, " A Novel Blind Estimation Algorithm," IEEE Trans. on Signal Processing, Vol. 45, No.7, July 1997, pp. 1763-1769.

[11] Bart Kasko, "Neural Networks For Signal Processing", Prentice Hall, Englewood Cliffs, NJ 07632, 1992.