International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.2 (2014), pp.75-84
http://dx.doi.org/10.14257/ijsip.2014.7.2.08

Design Low Pass FIR Filter Using Generalized Regression Neural Network

Navneet Gupta and Ravindra Pratap Narwaria

Electronics Engineering Department
Madhav Institute of Technology & Science, Gwalior
navneetgupta19902000@gmail.com, narwaria10@yahoo.co.in

Abstract

In this paper a low pass finite impulse response (FIR) filter has been designed using artificial neural network. The optimization of the network has been done using generalized regression algorithm. The proposed approach has been compared with rectangular window method. The accuracy of this algorithm is about 98%, that is much higher than multi layer perceptron (MLP) back propagation algorithm.

Keywords: *Neural Network, Algorithm, Accuracy*

1. Introduction

Digital signal processing (DSP) is the most powerful technologies that will shape science and engineering in the 21st century. One of the important DSP applications is digital filter. The performance of digital filter in most cases better than the equivalent analog filter. Various methods of filter designing and analysis have been developed over past three decades. Digital filters are play a very important role in DSP. Filter designing is the process of transformation of input sequence to obtain desired output sequence. Filter designing process can be described as an optimization problem where each requirement terms to an error function which should be minimized. Digital filter are of two types, Finite Impulse Response (FIR) filter and Infinite Impulse response (IIR) filter. In this paper the concept of Artificial Neural Network, which is also a wide area of research. The main goal of the design is to find the recursive coefficients that define the filter transfer function. In this research work, DSP an neural networks were combined to produce an excellent algorithm for digital filter design.

In the previous work, the fir filter was designed by least square method and frequency sampling method. MLP and RBF algorithm were used to train the neural network model. The efficiency of these above method is about 93%, but when generalized regression algorithm is used the accuracy is about 98%. That is the advantage of this method.

2. Design FIR Filter using Window Method

The windowing method requires minimum amount of computational effort; so window method is simple to implement. For the given window, the maximum amplitude of ripple in the filter response is fixed. Thus the stop band attenuation is fixed in the given window, but there are some drawback also of this method [2]. The design of fir filter is not flexible. The frequency response of fir filter shows the convolution of spectrum of window function & desired frequency response because of this; the pass band & stop band edge frequency cannot be precisely specified [4]. In this work we use rectangular window method.

ISSN: 2005-4254 IJSIP

3. Artificial Neural Network

Artificial neural network (ANN) is an information processing network. In this network, the element neurons, process the information [1]. The signals are transmitted by the connection links. These links have an associated weight, which is multiplied with the incoming signal for any neural network. The output signal is obtained by applying activation function to the net processed input. The artificial neuron is sorted by architecture, training and activation function [5]. Here we describe two important types of training algorithm.

3.1. Radial Basis Network

Radial basis networks may require more neurons than the feed-forward back propagation network [19]. RBFN consists of three layers: an input layer, a hidden (kernel) layer, and an output layer as shown in Figure 1. Each layer are connected to the previous layer. The input variables are assigned to the nodes and they pass directly to the hidden layer without weights. The transfer functions of hidden nodes are radial basis function. An radial basis function is symmetrical about a given mean or a center point in a multidimensional space [20]. In the RBFN, the hidden nodes with RBF activation functions are designed a feed forward parallel architecture.Radial basis network can be either design by *newrbe* or *newrb* in MATLAB.

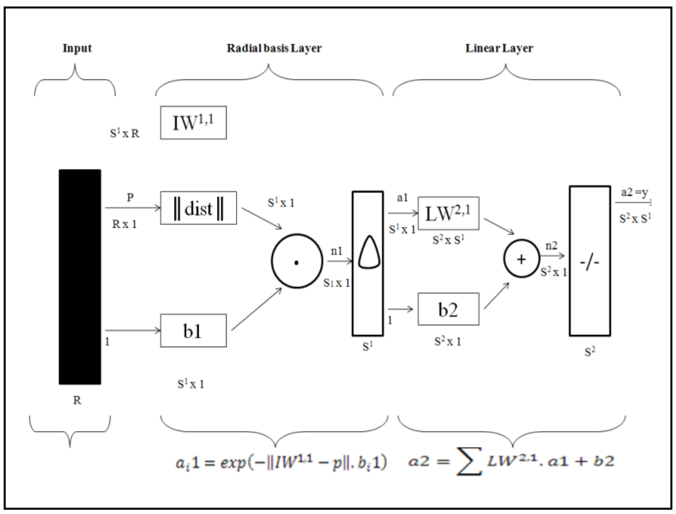

Figure 1. Network Architecture of RBF Network

$$radbase(n) = e^{-n^2} \quad \text{.....................} \quad (1)$$

3.2. Generalized Regression Neural Network

Generalized regression neural network (GRNN) is used for the function approximation. A GRNN is a quite similar to the radial basis neural networks, which is based on kernel regression networks. A GRNN does not require any training procedure as in the back propagation networks. It approximates any function between input and output dataset, drawing the function estimate directly from the training data [17]. A GRNN consists of four layers: input layer, pattern layer, summation layer and output layer as shown in Figure 2. The number of units in input layer depends on the number of the observation parameters. The first input layer connects to the pattern layer and each neuron shows a training pattern and its output. The pattern layer is connected to the summation layer, it has two different types of summation process, which are a single division unit and summation units. The summation and output layer together perform a normalization of output dataset. In training of network, radial basis function and linear activation functions are used in hidden and output layers [23].

$$Y'_i = \frac{\sum_{i=1}^{n} y_i.exp - D(x,x_i)}{\sum_{i=1}^{n} exp - D(x,x_i)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$D(x, x_i) = \sum_{k=1}^{m} \left(\frac{x_i - x_{ik}}{\sigma}\right)^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

In the above equation, Y'_i is the weight connection between the i^{th} neuron in the pattern layer and the S-summation neuron, n is the number of the training patterns, D is the Gaussian function, m is the number of elements of an input vector, x_k and x_{ik} are the j^{th} element of x and x_i, respectively, r is the spread parameter, whose optimal value is determined experimentally. The architecture for the GRNN is shown below in figure 3. It is same as the radial basis network, but it has a slightly different second layer. This type of network can be design by *newgrnn*.

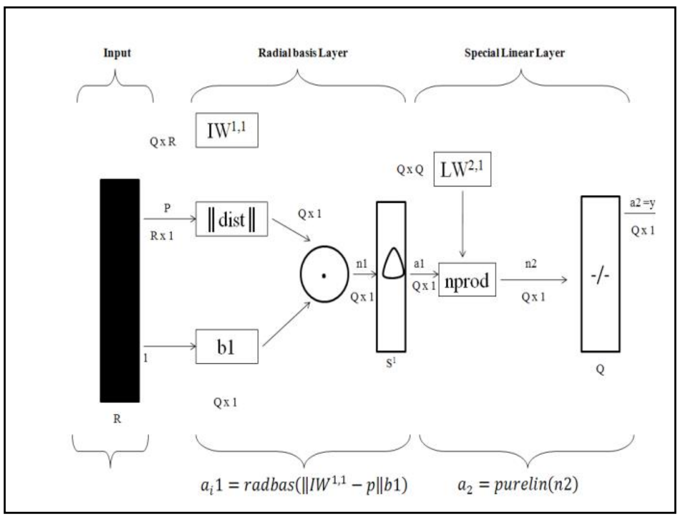

Figure 2. Network Architecture of GRNN

4. Formulation of Problem

The aim of this paper is to estimate the cut off frequency of any given coefficients of fir filter and increase the accuracy of result from previous work. Accuracy is determined by comparing the result from *fdatool* and *nntool*. For training purpose we use input output dataset derived from *fdatool*, and for testing purpose we use a test input and simulate it using *nntool*.

5. Methodology

In this methodology we use rectangular window technique for filter designing and generalized regression neural network for training purpose.

5.1. Step 1

Low pass fir filter designed by *fdatool*. The order of filter is 10. We use cut of frequency ranges from 0 to 1. Then set the value of f_c=0.05 and design the filter. Repeat the same process for value of f_c from 0.1 to 1. So this give the 19 dataset of input and output data. Out of these 19 dataset we use 17 for training and 2 for testing. Here input is h(n) and output is f_c. The screenshot of first step is given below in Figure 3.

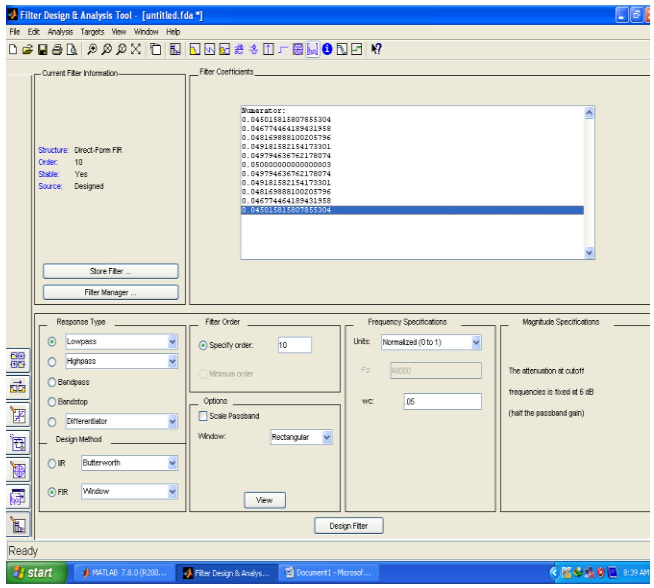

Figure 3. Filter Designing by *fdatool*

5.2. Step 2

Export the coefficients on the matlab workspace as shown in Figure 4.

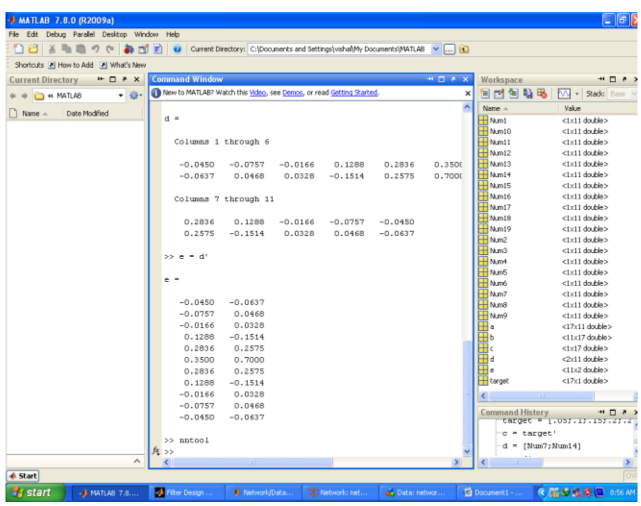

Figure 4. MATLAB Workspace

5.3. Step 3

In this step we design the neural network model by *nntool* shown in Figure 5. Training is done by GRNN and spread constant is set to 0.1. After training, simulate the network by testing input shown in Figure 6, and the simulation result shown in Figure 7.

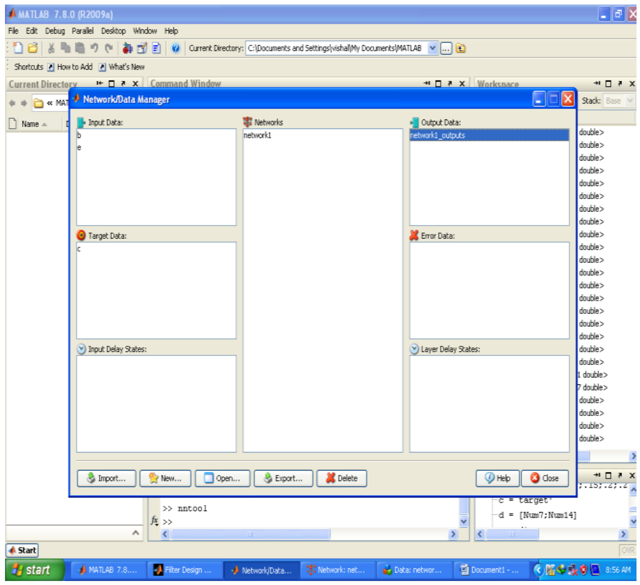

Figure 5. Create the Netwotk by *nntool*

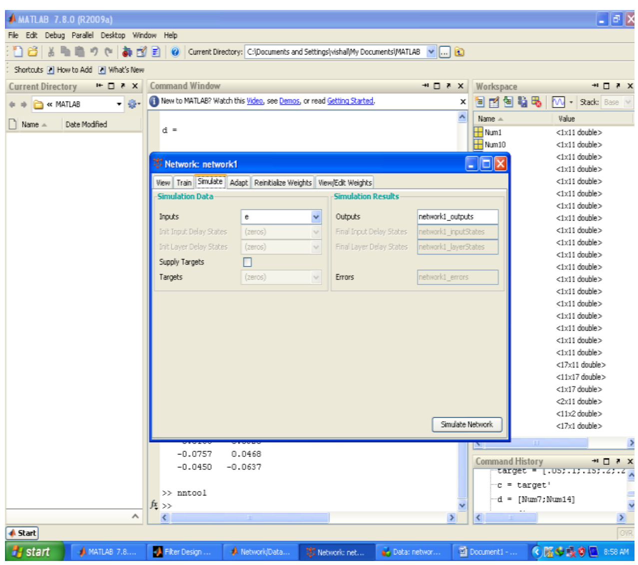

Figure 6. Simulate the Network with Test Input

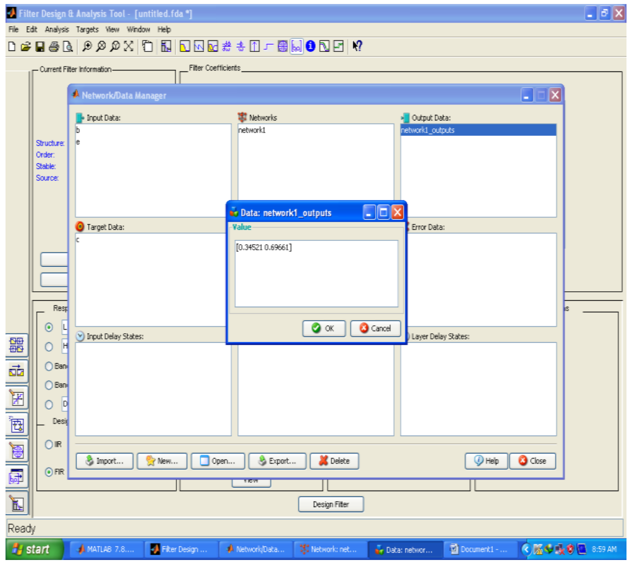

Figure 7. Result of Simulation Process

6. Results and Discussion

In the above experiment, According the Table 2: (Result from *fdatool*) For the test input h(n), the output fc is 0.35 & 0.70. this result is deduce from filter designing tool. According to the Table 3: (Simulation result from *nntool*) the output for the same test input h(n) is 0.34521 & 0.69661. This result is taken from neural network tool. So the results from *nntool* and *fdatool* are nearly same, and by this process we can easily estimate the cut off frequency of any fir filter.

7. Conclusion

From this experiment we estimate the cut off frequency and other parameter from filter coefficient by the help of GRNN, and it is quite simple method than complex calculative window method. The above figures show that results come from rectangular window method and artificial neural network is almost same. For the filter designing purpose This GRNN training algorithm is much better than other training algorithm like MLP, RBF *etc*. In the previous work, when MLP is used as a training algorithm, result is about 93% accurate. By the using of generalized regression algorithm, the accuracy of result is almost 98%. So there is increment of 5% in accuracy, which is very effective.

Table 1. Result from *fdatool* (for Train Input)

h(n) (Train Input)											Cut off frequency (Target)
h(0)	**h(1)**	**h(2)**	**h(3)**	**h(4)**	**h(5)**	**h(6)**	**h(7)**	**h(8)**	**h(9)**	**h(10)**	**Fc**
0.045	0.046	0.048	0.049	0.049	0.05	0.049	0.049	0.048	0.046	0.045	0.05
0.063	0.075	0.085	0.093	0.098	0.1	0.098	0.093	0.085	0.075	0.063	0.10
0.045	0.075	0.104	0.128	0.144	0.15	0.144	0.128	0.104	0.075	0.045	0.15
0.0	0.046	0.100	0.151	0.187	0.2	0.187	0.151	0.100	0.046	0.0	0.20
-0.045	0.0	0.075	0.159	0.225	0.25	0.225	0.159	0.075	0.0	-0.045	0.25
-0.063	-0.046	0.032	0.151	0.257	0.3	0.257	0.151	0.032	-0.046	-0.063	0.30
0.0	-0.075	-0.062	0.093	0.302	0.4	0.302	0.093	-0.062	-0.075	0.0	0.40
0.045	-0.046	-0.094	0.049	0.314	0.45	0.314	0.049	-0.094	-0.046	0.045	0.45
0.063	0.0	-0.106	0.0	0.318	0.5	0.318	0.0	-0.106	0.0	0.063	0.50
0.045	0.046	-0.094	-0.049	0.314	0.55	0.314	-0.049	-0.094	0.046	0.045	0.55
0.0	0.075	-0.062	-0.093	0.302	0.6	0.302	-0.093	-0.062	0.075	0.0	0.60
-0.045	0.075	-0.016	-0.128	0.283	0.65	0.283	-0.128	-0.016	0.075	-0.045	0.65
-0.045	0.0	0.075	-0.159	0.225	0.75	0.225	-0.159	0.075	0.0	-0.045	0.75
0.0	-0.046	0.100	-0.151	0.187	0.8	0.187	-0.151	0.100	-0.046	0.0	0.80
0.045	-0.075	0.104	-0.128	0.144	0.85	0.144	-0.128	0.104	-0.075	0.045	0.85
0.063	-0.075	0.085	-0.093	0.098	0.9	0.098	-0.093	0.085	-0.075	0.063	0.90
0.045	-0.046	0.048	-0.049	0.049	0.95	0.049	-0.049	0.048	-0.046	0.045	0.95

Table 2. Result from *fdatool* (for Test Input)

h(n) (Test Input)											Cut off frequency (Output)
h(0)	**h(1)**	**h(2)**	**h(3)**	**h(4)**	**h(5)**	**h(6)**	**h(7)**	**h(8)**	**h(9)**	**h(10)**	**Fc**
-0.045	-0.075	-0.016	0.128	0.283	0.35	0.283	0.128	-0.016	-0.075	-0.045	0.35
-0.063	0.046	0.032	-0.151	0.257	0.7	0.257	-0.151	0.032	0.046	-0.063	0.70

Table 3. Simulation Result from *nntool* (for Test Input)

h(n) (Test Input)											Cut off frequency (Output)
h(0)	**h(1)**	**h(2)**	**h(3)**	**h(4)**	**h(5)**	**h(6)**	**h(7)**	**h(8)**	**h(9)**	**h(10)**	**fc**
-0.045	-0.075	-0.016	0.128	0.283	0.35	0.283	0.128	-0.016	-0.075	-0.045	0.34521
-0.063	0.046	0.032	-0.151	0.257	0.7	0.257	-0.151	0.032	0.046	-0.063	0.69661

References

[1] S. Haykins, "Neural Networks – A comprehensive foundation", Prentice – Hall of India Private Limited, New Delhi, **(2003)**.

[2] D. Bhattacharya and A. Antoniou, "Real Time Design of FIR Filters of Feedback Neural Network", vol. 3, **(1996)**, pp. 1070.

[3] K. B. Englihart, B. S. Hudgins, M. Stevenson and P. A. Parker, "Myoelectric Signal assification using a Finite Impulse Response Neural Network", Technical Report, The university of New runswork, Canada, **(1994)** June, pp. 803-820.

[4] G. Rama Marthy, "Finite Impulse Response FIR filter Model of Synapses: Associated Neural Network", Proceeding of the Fourth Annual IEEE International Confidences on Natural Computation, **(2008)**, pp. 3304-3309.

[5] K. J. Hintz and J. J. Spofford, "Evoling Neural Network", Proceedings of the IEEE Transactions on Communication and Intelligence, **(1990)** May, pp. 333-338.

[6] A. Varun., J. O. Wesley and M. O. Una, "Filter approximation using Explicit Time and Frequency Remain specification", Proceeding, of the Annual Symposium on Artificial Intelligence, 2006, Seattle, Washington, **(2006)**, pp. 174-165.

[7] I. Ioan., R. Corina and I. Arpad, "The optimization of feedforward", Proceedings of the international and information –ICTAMI 2004, Thessalo-niki, Greece, **(2004)**.

[8] C. I. Feachor Emmonual and W. Jervis Barriel, "Digital Signal Processing: A Practical Approch", Person Education (Singapore) Ltd., 2001, Second Edition, **(2001)**.

[9] J. G. Proakis and D. G. Manolakis, "Digital Signal Processing – Principle, Algorithms and Applications", Prentice –Hall of India, New Delhi, **(2000)**.

[10] J. Larsen, "Design of Neural Network Filters", Ph.D. Thesis, Electronics Institute, Technical University of Denmark, **(1993)**.

[11] M. O. Ahmad and J. D. Wang, "An analytical least square solution to the design problem of two-dimensional FIR filters with quadrantally symmetric or antisymmetric frequency response", IEEE Trans. Circuits Syst., vol. CAS-36, **(1989)**, pp. 968-979.

[12] S. Burrus, J. A. Barreto and I. W. Selesnick, "Iterative reweighted least-squares design of FIR filters", IEEE Trans. Signal Processing, vol. 42, **(1994)**, pp. 2926-2936.

[13] W. Tank and J. Hopfield, "Simple neural optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit", IEEE Trans. Circuits Syst., vol. CAS-33, **(1986)**, pp. 533-541.

[14] L. R. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing", Englewood Cliffs, NJ: Prentice-Hall, **(1989)**.

[15] R. Algazi and M. Suk, "On the frequency weighted least squares design of finite duration filters", IEEE Trans. Circuits Syst., vol. CAS-34, **(1987)**, pp. 80-95.

[16] T. Saramaki, "Finite impulse response filter design", Handbook for Digital Signal Processing, S. K. Mitra and J. F. Kaiser, Eds.,Wiley, New York, NY, USA, **(1993)**.

[17] D. F. Specht and P. D. Shapiro, "Training speed comparison of probabilistic neural networks with back-propagation networks", Proc. Int. Neural Network Conf. (Paris, France), vol. 1, **(1990)** July, pp. 440- 443.

[18] D. F. Specht, "Series estimation of a probability density function", Technometrics, vol. 13, no. 2, **(1971)** May.

[19] P. Burrascano, "Learning vector quantization for the probabilistic neural network", IEEE Trans. Neural Nerw'orks, vol. 2, **(1991)** July, pp. 458- 461.

[20] H. B. Celikoglu, "Application of radial basis function and eneralized regression neural networks in non-linear utility function specification for travel mode choice modelling", Math Comput Model, vol. 44, **(2006)**, pp. 640-58.

[21] E. J. Tkacz and P. Kostka, "An application of wavelet neural network for classification patients with coronary artery disease based on HRV analysis", Proceedings of the Annual International Conference on IEEE Engineering in Medicine and Biology, **(2000)**, pp. 1391-1393.

[22] H. Demuth, and M. Beale, "Neural Network Toolbox for Use with MATLAB", User's Guide, Version 3.0.

[23] D. F. Specht, "A practical technique for estimating general regres- sion surfaces", Lockheed Missiles & Space Company, Inc., Palo Alto, CA, LMSC-6-79-68-6, also available as Defense Technical Information Center AD-672505 or NASA N68-29513, **(1968)** June.

Authors

Navneet Gupta, he received the B.Tech. degree in Electronics & Communication from Uttar Pradesh Tecnical University, Lucknow in 2011 and Masters in Microwave Engineering (Persuing) from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal. He is currently student in the Dpartment of Electronics Engineering at Madhav Institute of Technology & Science, Gwalior.

Ravindra Pratap Narwaria, he received the B.E. degree in Electronics & Communication from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal in 2003 and Masters in Measurement & control from Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal in 2005. He has 08 years of teaching experience. He is currently working as Assistant Professor with the Department of Electronics Engineering at Madhav Institute of Technology & Science, Gwalior.

