World Applied Sciences Journal 29 (8): 1087-1093, 2014
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2014.29.08.1434

Comparison of Training, Testing and Validation Sets in the Application of Analog Signals

B.P. Bhuvana

Department of ECE, Bharath University, Chennai - 73, India

Abstract: In the modern era all electronic devices use error free signals. Here int his project we comare the various results of the training, testing and validation datas for reducing the errors. Analog devices are compact and use less power and area when compared with the digital circuits. A novel approach to minimize error in Analog VLSI Signals. The accuracy of analog Signals is reduced by the limitation of analog hardware. The accuracy and efficiency of analog signals can be increased by training the signals till we achieve our desired efficiency. Hence, simple techniques are used to increase the accuracy and performance of analog signals by the neural network.

Key words: Analog VLSI • Neural network

INTRODUCTION

As of today we are in the advanced hi-tech world that requires system without any fault in it. The output of the system is expected to be very accurate without any error in it. Modern electronics uses artificial neural networks and fuzzy logics to optimize the performance of electronic devices. Performance of analog circuits can be optimized by the neural networks. These analog circuits consume very less power when compared with the digital circuits. Implementation of advanced algorithms related to very large scale integrated(VLSI) circuits solves the area and power constraints. These are specially used in advanced portable systems [1].

In my paper I used Levenberg–Marquardt Backpropagation algorithm to train the analog signals. Nonlinear Auto Regresssive with external input(NARX) is chosen for training. Voice samples have been collected for training purpose. I have trained a particular analog signal and the error is found to be 2.4117xe-5. When a different analog signal is applied to the network the mean square error is high to a rate of 8.2310xe-5. Various analog signals is given to the network and all are found to have mean square error greater than 7.2311xe-5. By using Levenberg–Marquardt Backpropagation algorithm the network is trained. As the number of epoch increases the error starts to decrease. The number of hidden layers is chosen as 10 hidden layers and the delay as 60. As we are increasing the delay the time taken for training increase but the error keeps on reducing as the number of the iteration increases [2].

Neural network time series prediction method is used for training the network. Levenberg–Marquardt algorithm (LMA) is also known as the damped least-squares (DLS) method. It provides a numerical solution to the problem of minimizing a function, generally nonlinear, over a space of parameters of the function. These minimization problems arise especially in least squares curve fitting and nonlinear programming. The LMA interpolates between the Gauss–Newton algorithm (GNA) and the method of gradient descent. The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. For well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as Gauss–Newton using a trust region approach [3].

The LMA is a very popular curve-fitting algorithm used in many software applications for solving generic curve-fitting problems. However, the LMA finds only a

Corresponding Author: B.P. Bhuvana, Department of ECE, Bharath University, Chennai - 73, India.

local minimum, not a global minimum. It is often the fastest backpropagation algorithm in the toolbox and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms [4].

The input data is classified into three types before training. They are Training data, validation data and testing data. Training data is repeatedly used to estimate theweights of design.Validation data is used repeatedly used to estimate the non-trainingperformance error. Also it is used to stop training once thenon-trainingvalidation error estimate stops decreasing. Test data is used once and only once on the best design to obtain an unbiased estimate for the predicted error of unseen non-trainingdata [5].

The standard NARX network is a two-layer feed forward network, with a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer. This network also uses tapped delay lines to store previous values of the *input* and output sequences. The output of the NARX network is fed back to the input of the network. This has two advantages. The first The second is that the resulting network has a purely feed forward architecture and therefore a more efficient algorithm can be used for training.

Image and sound are analog signals that can be processed by the biological system. Artificial Neural Networks is used to depict the character of biological neurons. Cyril Prasanna Raj P proposed a neural network architecture with new technology for weight storage and with backpropogation algorithm in analog domain for signal processing applications [6]. Mauro Tucci and Marco Raugi presented a new approach for self organising maps. Here a neuron act as finite impulse response (FIR) and during sequential learning process the co-efficient of filters are updated [7].

Analog device have device mismatch, charge leakage and nonlinear transfer function. GonzalaCarrajal presented an analysis of these effects on the residual error by LMS algorithm [8]. Miguel Figueroa described adaptive signal processing in mixed signal VLSI based on anti hebbian learning [9].

Neural network is used for face recognization based applications. This system usually uses a dimensionality-reduction network. The co-efficient can be learned or programmed to perform principal component analysis (PCA) and linear discriminant analysis (LDA) for dimensionality reduction [10]. Yuzo Hirai and Kuninori Nishizawa fabricated principal component analysis (PCA) learning network by asynchronous PDM in a FPGA circuits. Differential equations and the circuits are used to express the generalizedhebbian algorithm [11].

Chris Diorio, Paul Hasler developed a complimentary pair of four terminal silicon synapses for analog learning applications. A non-volatile memory is used. Electron tunneling and hot electron injection allow bidirectional memory updates [12]. Chun-Hsien Chen developed neural network architecture for syntax analysis. Artificial Neural Networks can be used for symbol processing applications. A new approach is proposed for lexical analysis, stack and parse tree construction [13].

Digital perturbative learning in analog VLSI neural network was proposed by Vincent F Koosh. He used two feed forward neural networks. First uses analog synapses and second uses a computer for controlling global operations [14]. Momchil Milew proposed an analog implementation of artificial neural network with quadratic non-linearity of synapses [15].

We have selected a specific application to apply neural networks in analog circuits. A pattern was assumed by Widrow and Hoff. It is an adaptive algorithm. Filter co-efficient are adjusted by LMS for minimizing the cost function. Recursive least square algorithm uses matrix operations, hence operations. But LMS doesn't use such operations; hence it uses less time and resources. LMS algorithm is simple and less complicated. It also uses gradient based steepest decent method. It does not require correlation function calculation and matrix inversions.For understanding the concepts of LMS algorithm. Let us consider a linear Perceptron. Here the weights are updated so as to reduce the error. This error is the difference between the output and the external reference [16].

MATERIALS AND METHODS

Levenberg–Marquardt Backpropagation Algorithm: Backpropagation is also called as the generalized delta rule. This is a gradient descent is that the input to the feed forward network is more accurate.

Approach to minimize the total squared error of the output which is computed by the network. Process of training in backpropagation involves three steps. They are:

- Feed forwardof the input vector.
- Calculation and backpropagation of the error.
- Adjustment of the weight.

Many changes have been made to the algorithm to increase the speed. One hidden layer is enough but in case if problem is very critical, then more than one hidden layer is needed [17].

As discussed earlier, backpropagation training involves three major steps. They are Feed forward of the input vector, Calculation and backpropagation of the error, adjustment of the weight [18]. During the feed forward operation, the input vectors receive the input signal and broadcasts it to the hidden layers B1....Bp. This is followed by the computation of hidden layer activation function and it broadcasts it to the output layer [10].

Each output layer then computes the activation function which will be the output of the network for the given input pattern. Now the output vector compares c_k with the target value t_k for finding the error for the corresponding input pattern. The factor δ_k should be calculated so that the error can be distributed at the output vector C_k back to all vectors in the hidden layer. This is used to update the weight between the output and the hidden layer [19].

Similarly, the factor δ_j should be calculated so that the error can be distributed at the hidden vector B_k back to all vectors in the input layer. This is used to update the weight between the hidden and the input layer. Once all δ values are determined then the weight for all values are adjusted. The weight w_{jk} is adjusted and the adjustment is based on the factor δ_k and activation function b_j of the hidden unit B_j. Similarly, The weight $v_{i,j}$ is adjusted and the adjustment is based on the factor δ_j and activation function a_j of the input unit A_j.

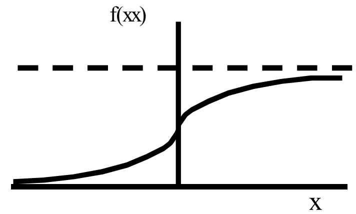
Activation Function: Some of the features of the activation function which is used in the back propagation are as follows. It should be

- Continuous
- Differentiable
- Monotonically non-decreasing

For easy computation, the derivative used should be easy to compute. The value of these derivatives can be expressed in terms of the value of the function [20]. One of the typical activation function which suits all requirements is the binary sigmoid function which has a range of (0,1) and it can be defined as:

$$f_1(a) = \frac{1}{1+\exp(-a)}$$

With



Another activation function is bipolar sigmoid function, which has a range of (-1,1) and is defined as:

$$f_2(a) = \frac{2}{1+\exp(-a)} - 1$$

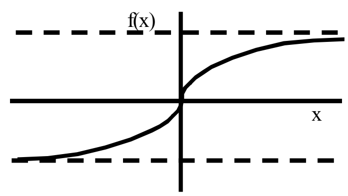
with,

$$f_2'(a) = \frac{1}{2}[1 + f_2(a)][1 - f_2(a)]$$

Bipolar function is closely related to the function:

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$$

This function is illustrated in figure:



Random Initialization: The choice of initial weights shows whether the neural net reaches a minimum error and if it reaches minimum error then how fast it converges. Weight update between any two units depend on both the derivative of the upper unit activation function and the lower unit activation function [21].

Hence, it should be noted that the initial weights which makes activations or the derivative of activation function zero should be avoided. Initial weight values should not be too large also. A common method to initialize the weight is to assign random values between -0.5 and 0.5. These values can be either positive or negative.

Training of Neural Net: Aim of the backpropagation net is to achieve a balance between the correct response to training pattern and the good response to the new input pattern and also it is not mandatory to continue the process of training till the mean square error reaches the minimum value.

Hecht-Nielsen suggested two sets of data during the training process. They are as follows, a set of training patterns and a set of training –testing patterns. These are disjoint sets. Based on the training patterns weights are adjusted. But the error is calculated by the training–testing data. Training continues as long as the error in training-testing decreases. When the error is increased, net memorizes the training patterns and training is terminated at this point [22].

Number of Training Pairs: Consider P as the number of training patterns available, W as the number of weights to be trained and e is the expected accuracy of classification. Some of the questions which are asked regarding the number of training pairs include “how can a net classify the percentage of testing and training patterns correctly?” if there is enough testing pattern available, then the net will generalise properly. Training patterns are determined by:

$$\frac{W}{p} = e$$

Data Representation: In some cases, input and output vectors same range values in some components. This is because of the fact that a single factor in the weight correction expression is the activation of the lower unit whose activation function is zero. From this it is clear that learning may be improved if the inputs can be represented in the bipolar form and bipolar sigmoid function is used [10].

In the applications of the neural network, data may be represented in the form of continuous valued variable or a set or ranges. Consider an example the temperature of a food can be represented by the actual temperature or by the ranges of temperature like: hot, room temperature, chilled and frozen. This can be represented in neural network with the former case stating a single neuron could be used in general and in later case four neurons each with bipolar values could be appropriate.

Neural Network Implementation: Artificial Neural Network can be built using analog, digital or hybrid electronic hardware. Neurocomputing algorithms can be implemented in dedicated or general purpose. One of the commercial neurocomputer involves simulation of error back-propagation. In conventional programmable host computer, neurons are trained. The digital circuitry performs subsequent transfer and storage of weights which result from the training.

However, through analog computation information stored in the network is recalled within the electronic circuitry. Fast recall is performed by the network based on some features. Those features are the analog computation mode and parallel form of information processing. Numerical simulation is done in off chip in the neurocomputer. Digital transfer and data storage circuitry and a dedicated analog neural network perform a recall.

The most commonly encountered operations are scalar and outer product vector multiplication and matrix vector multiplication. These operations are performed are as a series of multiply and add operations. Generation of the non-linear activation function involves additions and subtractions of matrices and vectors and ordinary multiplications and they, are less frequent, but are also indispensible for most learning and recall task.. Efficiency of training can be improved by the following reasons [11].

- Programmable computers have dedicated hardware architectures.
- Parallel architectures of densely connected neural nodes contains multiply and multiply-add procesors.
- By using generalization property of networks, training data volume data is reduced.
- Design of Neurocomputing IC is done by analog, digital or digital/analog arrays.

Sequence of steps are illustrated in fig 4 showing the problem-algorithm-model flowchart. Alternatively, the users problem can be transferred directly to the artificial neural system model for execution but sometimes artificial neural system model is done through modelling of the algorithm after neural processing of algorithm. Solution is obtained by processing the artificial neural system model [12].

RESULTS

Matlab 7.12.0 have been used to compare the training, testing and validation sets of the analog signal.

Validation value kept constant at 5%, training and testing values are changed and its corresponding MSE.

MSE VS Testing

Fig. 1:

Table 1: Testing value kept constant at 5%, training and validation values are changed and its corresponding MSE with various delays

		MSE with Delay=20(x e-5)		
Training	Testing	Training	Validation	Testing
90%	5%	2.63770	2.77059	2.57664
85%	10%	2.64463	2.73923	2.69581
80%	15%	2.65826	2.87535	2.89389
75%	20%	2.75014	2.67705	2.77703
70%	25%	2.57919	2.74868	2.70607
65%	30%	2.69086	2.69086	2.71660
60%	35%	2.72459	2.88534	2.85427

Table 2: Validation value kept constant at 5%, training and testing values are changed and its corresponding MSE with various delays

		MSE with Delay=20(x e-5)			MSE with Delay=60(x e-5)		
Training	Validation	Training	Validation	Testing	Training	Validation	Testing
90%	5%	8.20310	8.27360	8.52412	2.63770	2.77059	2.57664
85%	10%	8.22669	8.20032	8.16889	2.75331	2.91235	2.85945
80%	15%	8.20939	8.12184	8.31035	2.64982	2.68984	2.69834
75%	20%	8.17215	8.54679	8.80296	2.62034	2.66437	2.72208
70%	25%	8.13892	8.38633	8.00138	2.58701	2.65923	2.70743
65%	30%	8.37165	8.24611	8.50219	2.92653	2.94331	2.86027
60%	35%	8.19546	8.27563	8.30747	2.66546	2.81286	2.09614

Table 3: Training value kept constant at 5%, validation and testing values are changed and its corresponding MSE with various delays

		MSE with Delay=20(x e-5)			MSE with Delay=60(x e-5)		
Testing	Training	Training	Validation	Testing	Training	Validation	Testing
90%	5%	8.20310	8.27360	8.52412	2.63770	2.77059	2.57664
85%	10%	8.23112	8.39814	7.96272	2.64463	2.73923	2.69581
80%	15%	8.14961	8.43205	8.36912	2.65826	2.87535	2.89389
75%	20%	8.25182	7.75878	8.11216	2.75014	2.67705	2.77703
70%	25%	8.20976	7.82808	8.28608	2.57919	3.44868	2.70607
65%	30%	8.18087	8.28360	8.27261	2.69086	2.69086	2.71660
60%	35%	8.20296	8.34668	8.23698	2.72459	2.88534	2.85427

Table 4:

		MSE with Delay=20(x e-5)			MSE with Delay=60(x e-5)		
Validation	Testing	Training	Validation	Testing	Training	Validation	Testing
35%	5%	8.26711	8.40508	9.00512	2.75696	2.86725	2.91087
30%	10%	8.28975	8.08951	8.18075	2.63059	2.74032	2.89370
25%	15%	8.21038	8.20386	8.16358	2.61100	2.76149	2.68280
20%	20%	8.29345	8.11220	8.21901	2.66405	2.76583	2.96431
15%	25%	8.17933	8.51365	8.08110	2.73923	2.75175	2.92726
10%	30%	8.16659	8.52880	8.26261	2.87139	3.20191	3.00833
5%	35%	8.38356	8.12407	8.30877	2.83364	2.90113	2.94659

MSE Vs Validation

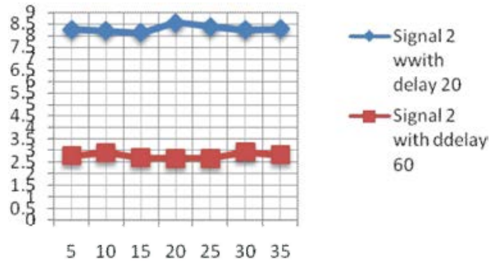


Fig. 2:

MSE Vs Training

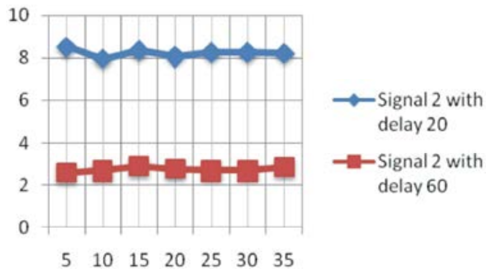


Fig. 3:

MSE VS Testing

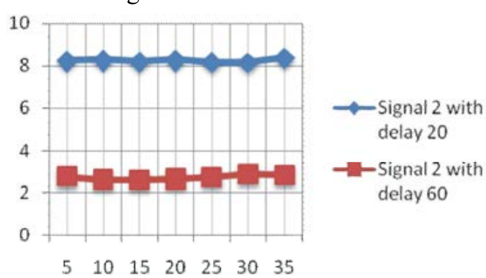


Fig. 4

DISCUSSION

The work done in this paper evaluates the impact of errors in analog signals and it shows the method of error rectification by neural networks. In particular, we reduce the error by using Levenberg–Marquardt Backpropagation algorithm [23-16].

CONCLUSION

Error reduction is necessary in all electronic devices as error causes the efficiency of the circuits to diminish. Here, for an analog signal the mean square error is reduced to 2.41171xe-5 from 8.2310xe-6 by training the signals by using Levenberg–Marquardt Back propagation algorithm and increasing delays in neural network time series tool.

REFERENCES

1. Chen, H., S. Saighi, L. Buhry and S. Renaud, 2010. Real-time simulation ofbiologically realistic stochastic neurons in VLSI, IEEE Trans. Neural Netw., 21(9): 1511-1517.
2. Montalvo, A. J., R. S. Gyurksic and J. J. Paulos, 1997. Toward a generalpurposeanalog VLSI neural network with on-chip learning, IEEE Trans. Neural Netw., 8(2): 413-423.
3. Gatet, L., H. Tap-Beteille and F. Bony, 2009. Comparison between analog anddigital neural network implementations for range-finding applications, IEEE Trans. Neural Netw., 20(3): 460-470.
4. Dlugosz, R., T. Talaska, W. Pedrycz and R. Wojtyna, 2010. Realization ofthe conscience mechanism in CMOS implementation of winner-takesallself-organizing neural networks, IEEE Trans. Neural Netw., 21(6): 961-971.
5. Vittoz, E., 1985. Micropower techniques, in VSI circuits for telecommunications, Y.P Tsividis and P.Antognetti, Eds. Englewood Cliffs, NJ: Prentice - Hall.
6. Prasanna, C. and S. Pinjare, 2009. Design and analog VLSI implementationof neural network architecture for signal processing, Eur. J. Sci. Res., 27(2): 199-216.
7. Tucci, M. and M. Raugi, 2010. Adaptive FIR neural model for centroidlearning in self-organizing maps, IEEE Trans. Neural Netw., 21(6): 948-960.
8. Carvajal, G., M. Figueroa and S. Bridges, 2006. Effects of analog-VLSIhardware on the performance of the LMS algorithm, in Proc. Int. Conf.Artif. Neural Netw., pp: 963-973.
9. Figueroa, M., E. Matamala, G. Carvajal and S. Bridges, 2006. Adaptive signalprocessing in mixed-signal VLSI with anti-Hebbian learning, in Proc.IEEE Comput. Soc. Annu. Symp. VLSI, Karlsruhe, Germany, Mar., pp: 133-140.
10. Carvajal, G., W. Valenzuela and M. Figueroa, 2008. Subspace-based facerecognition in analog VLSI, in Advances in Neural InformationProcessing Systems 20. Cambridge, MA: MIT Press, pp: 225-232.
11. Hirai, Y. and K. Nishizawa, 2000. Hardware implementation of a PCAlearning network by an asynchronous PDM digital circuit, in Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw., 2: 65-70.
12. Diorio, C., P. Hasler, B.A. Minch and C. Mead, 1997. A complementarypair of four-terminal silicon synapses, Analog Integr. Circuits Signal Process., 13: 1-2, 153-166.

13. Chun-Hsien Chen and Vasant Honavar, 1999. A Neural-Network Architecture for Syntax Analysis, IEEE Transaction on Neural Networks, 10(1).
14. Vincent F. Koosh and Rodney M. Goodman, 2002. Analog VLSI Neural Network with Digital Perturbative Learning, IEEE Transactions On Circuits And Systems-Ii: Analog And Digital Signal Processing, 49(5).
15. Momchil Milevand Marin Hristov, 2003. Analog Implementation of ANN with InherentQuadratic Nonlinearity of the Synapses, IEEE Transactions On Neural Networks, 14(5): 1187.
16. Gonzalo Carvajal, Miguel Figueroa, Daniel Sbarbaro and Waldo Valenzuela, 2011. Analysis and Compensation of the Effects ofAnalog VLSI Arithmetic on the LMS Algorithm, IEEE Transactions On Neural Networks, 22(7).
17. www.ips.gov.au.
18. Jamil Ahnad M. and G.N. Tiwari, 2009. Optimization of tilt angle for Solar Collector to receive Maximum Radiation, The Open Renewable Energy Journal, 2: 19-24.
19. Sundarraj, M., 2013. Study Of Compact Ventilator, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 16(12): 1741-1743.
20. Sundar Raj, M., T. Saravanan and R. Udayakumar, 2013. Energy Conservation Protocol for Real time traffic in wireless sensor networks, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1822-1829.
21. Sundar Raj, M. and T. R. Vasuki, 2013. Automated Anesthesia Controlling System, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1719-1723.
22. Sundar Raj, M., T. Saravanan and R. Udayakumar, 2013. Data Acquisition System based on CAN bus and ARM, Middle-East Journal of Scientific Research, ISSN: 1990-9233, 15(12): 1857-1860.
23. Mueen Uddin, Asadullah Shah, Raed Alsaqour and Jamshed Memon, 2013. Measuring Efficiency of Tier Level Data Centers to Implement Green Energy Efficient Data Centers, Middle-East Journal of Scientific Research, 15(2): 200-207.
24. Hossein Berenjeian Tabrizi, Ali Abbasi and Hajar Jahadian Sarvestani, 2013. Comparing the Static and Dynamic Balances and Their Relationship with the Anthropometrical Characteristics in the Athletes of Selected Sports, Middle-East Journal of Scientific Research, 15(2): 216-221.
25. Anatoliy Viktorovich Molodchik, 2013. Leadership Development. A Case of a Russian Business School, Middle-East Journal of Scientific Research, 15(2): 222-228.
26. Meruert Kylyshbaevna Bissenova and Ermek Talantuly Nurmaganbet. The Notion of Guilt and Problems of Legislative Regulations of its Forms. The Notion of Guilt in the Criminal Law of Kazakstan, Middle-East Journal of Scientific Research, 15(2): 229-236.