

## Adaptive Channel Equalization Using Multiplicative Neural Network for Rayleigh Faded Channel

<sup>1,2</sup>P. Sivakumar, <sup>2</sup>N. Chithra, <sup>3</sup>S.N. Sivanandam and <sup>4</sup>M. Rajaram

<sup>1</sup>ANNA University Chennai India

<sup>2</sup>EEE, SKP Engineering College Tiruvannamalai, India

<sup>3</sup>CSE, PSGTECH, Coimbatore India

<sup>4</sup>ANNA University of Technology, Tirunelveli, India

---

**Abstract: Problem statement:** Digital transmission over band-limited communication channel largely suffers from ISI and various noise sources. The presence of ISI and noise causes bit errors in the received signal. Equalization is necessary at the receiver to overcome these channel impairment to recover the original transmitted sequence. Traditionally equalization is considered as equivalent to inverse filtering and implemented using linear-perform under severe distortion conditions when Signal to Noise Ratio (SNR) is poor. Equalization can be considered as a non-linear classification problem and optimum solution is given by Bayesian solution. Non-linear techniques like Artificial Neural Networks (ANN) are very good choice for non-linear classification problems. Several non-linear equalizers have been implemented using ANN which outperformed LTE and solved the problem of equalization to the varying degree of sources. **Approach:** Forward neural network architecture with optimum number of nodes was used to achieve adaptive channel equalization. Summation at each node was replaced by multiplications which result in powerful mapping. The equalizer was tested on Rayleigh fading channel with a BPSK signal. **Results:** Results showed that proposed equalizer provides simplified architecture and improvement in the bit error rate at various levels of signal to noise ratio for Rayleigh faded channel. **Conclusion:** A high order feed forward network equalizer with multiplicative neuron is proposed in this study. Use of Multiplication allows direct computing of polynomial inputs and approximation with fewer nodes. Performance comparison in terms of network architecture and BER performance suggest the better classification capability of the proposed MNN equalizer over RRBF.

**Key words:** Channel equalization, Multiplicative Neural Network (MNN), rayleigh fading, Pi Sigma-Networks (PSN), polynomial inputs, digital communication, receiving filters, training sequence, Back Propagation (BP)

---

### INTRODUCTION

The growth in communication services during the past decades has been phenomenal. With the unimaginable development of internal technologies, efficient high speed data transmission techniques over communication channels introduce distortion in data. In digital communication, the transmission signals are generally in the form of multilevel rectangular pulses. The absolute bandwidth of multilevel rectangular pulses is infinity. If these pulses are passed through a band limited communication channel, they will spread in time. Pulse for each symbol may be smeared into adjacent time slot and interfere with adjacent symbol. This is defined as ISI (Prokias, 1995). Signal processing technique is used at the receiver to

overcome these interfaces so as to restore the transmitted symbols and recover their information. This technique is referred to as channel equalization or simply equalization (Prokias, 1995; Taba *et al.*, 2006).

In principle, if the characteristics of channel are precisely known, then it is always possible to design a pair of transmitting and receiving filters that can minimize the effect of ISI and additive noise (Haykin, 2002). The use of a fixed pair of transmitting and receiving filters designed on the basis of average channel characteristics may not adequately reduce ISI. Adaptive equalization which provides precise control over the time response of the channel is widely used to overcome this problem. Adaptive equalizers have therefore been playing a crucial role in the design of high speed communication system.

---

**Corresponding Author:** P. Sivakumar, ANNA University Chennai India, India

When significant noise is added to transmitted signal, linear boundaries are not optimal. The received signal at each instant may be considered as a nonlinear function of the past values of transmitted symbols. Since the non linear distortion varies with time from place to place effectively, the overall channel response becomes a nonlinear dynamic mapping. The problem is approached by using classification technique.

As shown in the wide range of engineering applications, neural network has been successfully used for modeling complex nonlinear systems (Bang *et al.*, 1996). A wide range of neural architectures are available for modeling the nonlinear phenomenon of channel equalization. Feed forward networks, Like Multilayer Perception (MLP), contain an input layer, an output layer and one or more hidden layers that possess nonlinear processing capabilities (Gibson *et al.*, 1989). The back propagation which is a supervised learning algorithm is used as a training algorithm (Zhang *et al.*, 1990). These neuron models process the neural inputs using the summation process.

Higher order networks recently have drawn great attention from researches due to their performance in nonlinear input output mapping, function approximation and memory storage capacity. Some examples of product unit NN are Sigma- Pi Network (SPN) and Pi Sigma-Networks (PSN). They allow neural networks to learn multiplicative interactions. Multiplication plays an important role in neural modeling of ANN (Kim *et al.*, 2002; Lyoda *et al.*, 2002).

The multiplicative neuron contains units which multiply their inputs instead of summing them and thus allow inputs to interface nonlinearly (Schmitt, 2002). In the present work, using multiplication neural networks, channel equalization has been modeled and performance is evaluated.

**Adaptive channel equalization:** The block diagram of adaptive equalization in Fig. 1 described as follows. The external time dependent input consists of the sum of desired signal  $g(k)$ , the channel nonlinearity NL and the interfacing noise  $V(k)$ . The adaptive filter has a Finite Impulse Response (FIR) structure. The impulse response is characterized by filter co-efficient. The co-efficient for a filter of order P are defined as: (Eq. 1)

$$W_k = [W_k(0), W_k(1) \dots W_k(p)]^T \quad (1)$$

A predefined delayed version of original signal forms the training sequence points for the adaptation. The parameter considered for the optimization is a cost function on the error signal which is difference between the desired signal and estimated signal given by: (Eq. 2)

$$e(k) = g(k) - y(k) \quad (2)$$

The desired signal is estimated by convolving the input signal with impulse response and it is expressed as: (Eq. 3)

$$g(k) = w_k^T \cdot x(k) \quad (3)$$

where,  $x(k) = [x(k), x(k-1), x(k-2) \dots \dots \dots x(k-p)]^T$  is the input signal vector. The filter co-efficient are updated at every instant as: (Eq. 4)

$$W_{k+1} = W_k + \Delta W_k \quad (4)$$

The optimization algorithm can be linear or nonlinear. The adaptive neural network equalizer is implemented by using a feed forward multiplicative neural network, whose architecture is shown in Fig. 2.

The transmitter sends a known training sequence to the receiver. The discrete-time BPSK signal sampled at a rate of  $f_s$  is generated by the following equation: (Eq. 5)

$$r_k(kT_s) = A \exp\{j\pi/2[1-m(kT_s)]\} \quad (5)$$

( $k = 0, \pm 1, \pm 2, \dots$ )

In order to obtain integral number of samples in each bit interval, the sampling frequency  $f_s$  is equal to  $m_s/t_b$  where  $m_s$  an integer denoting number of samples per bit duration. If  $m_k$  is defined as discrete time sampled version of the binary sequence  $m(t)$ , (5) becomes: (Eq. 6 and 7)

$$r_k = A \exp [j\pi/2(1-m_k)] \quad (6)$$

$$= \begin{cases} A & m_k = 1 \\ -A & \text{for } m_k = -1 \end{cases} \quad (7)$$

A sequence of 6000 equiprobable, BPSK complex valued symbol set, in which the input signal takes one of the values  $\{-1, 1\}$  is generated. In the absence of the noise the output signal occupies well-defined states of the BPSK signal constellation shown in Fig. 3.

When the signal is passed through the nonlinear channel, it becomes a stochastic random process. Decision boundaries can be formed in the observed pattern space to classify the observed vectors.

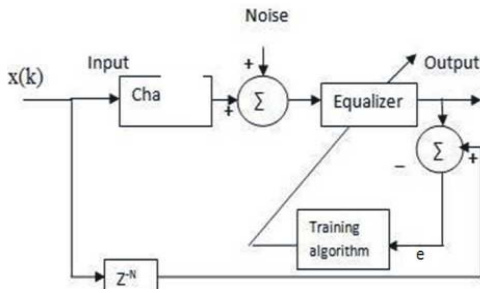


Fig.1: Schematic of adaptive channel equalization

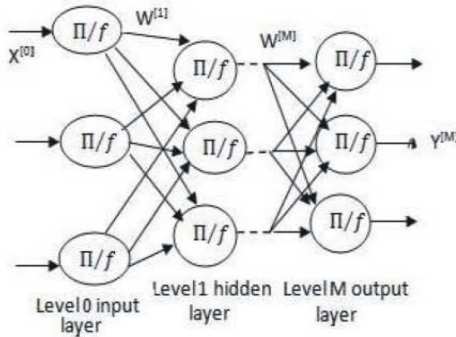


Fig. 2: Architecture of multiplicative neural network

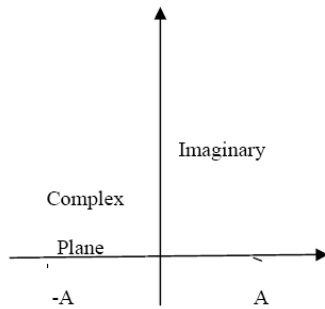


Fig. 3: BPSK signal in complex plane

For equalization, the adaptive filter is used in series with the unknown system on the test signal  $d(k)$  by minimizing the squared difference between adaptive equalizer output and the delayed test signal. The task of the equalizer is to set its coefficients in such a way that the output  $y(k)$  is a close estimate of the desired output vector, where the equalizer tries to estimate an output, which is closed to one of the transmitted values.

The neural equalizer separately processes the real and imaginary part using the multiplicative, split complex, neural network model (Kantsila *et al.*, 2004; Burse *et al.*, 2008). The block diagram of the channel equalizer using MNN is shown in Fig. 4. (Eq. 8 and 9)

$$F(x(t)) = f(x_{1R}(t), x_{2R}(t)) + if(x_{1I}(t), x_{2I}(t)) \quad (8)$$

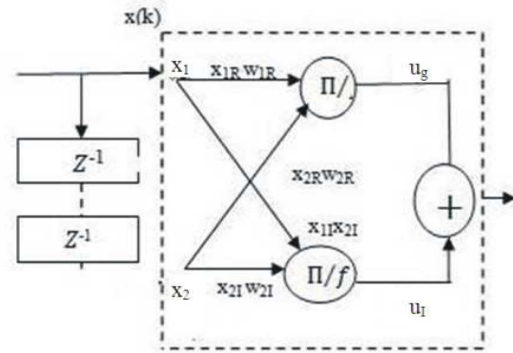


Fig. 4: Multiplicative neural network based channel equalizer

where, the input  $x_1(t) = x_{1R}(t) + ix_{1I}(t)$

And:

$$x_2(t) = x_{2R}(t) + ix_{2I}(t) \quad (9)$$

**Learning rule for multiplicative neuron:** A bipolar sigmoidal activation function is used at each node. This kind of neuron itself looks complex in the first instance but when it is used to solve a complicated problem it needs less number of parameters as compared to the existing conventional models. An error Back Propagation (BP) based learning using a norm-squared error function is described in (Yadav *et al.*, 2007; Burse *et al.*, 2009). The symbols used are:

- $X_0$  = The number of inputs in the input layer
- $x$  = The number of hidden layers in the FF network
- $X_x$  = The number  $x^{\text{th}}$  of neurons in the  $X^{\text{th}}$  hidden layer
- $Y_{jx}$  = The  $j^{\text{th}}$  neuron of the  $x^{\text{th}}$  hidden layer
- $j_x$  = The output of the  $j^{\text{th}}$  neuron of the  $x^{\text{th}}$  hidden layer
- $Y_{dk}$  = The desired output of the  $k^{\text{th}}$  neuron in the output layer
- $Y_k$  = The actual output of the  $k^{\text{th}}$  neuron in the output layer
- $W_{j_{xj_{x-1}}}$  = The weight of the connection between  $j^{\text{th}}$  neuron of the  $(x-1)^{\text{th}}$  layer and the  $j^{\text{th}}$  neuron of the  $x^{\text{th}}$  layer
- $b_{j_{xj_{x-1}}}$  = The bias of the connection between  $j^{\text{th}}$  neuron of the  $(x-1)^{\text{th}}$  layer and the  $j^{\text{th}}$  neuron of the  $x^{\text{th}}$  layer

The output of the  $j^{\text{th}}$  neuron in the first hidden layer is given as: (Eq. 10)

$$Y_{j1}^1 = f \left[ \prod_{j_{x-1}=1}^{N_{x-1}} (W_{j1j_0} x_{j_0} + x_{j1j_0}) \right] \quad (10)$$

For  $j_1 = 1, 2, \dots, N_1$  and  $x_{j_0}$  represents  $j^{\text{th}}$  input in the input layer and  $f(\cdot)$  is the activation function defined by: (Eq. 11)

$$f(y) = \frac{1 - e^{-y}}{1 + e^{-y}} \quad (11)$$

The output of the  $j^{\text{th}}$  neuron in the  $x^{\text{th}}$  hidden layer is given as: (Eq. 12)

$$Y_{j2}^2 = f \left[ \prod_{j_1=1}^{N_1} (W_{j_2j_1} Y_{j_1}^1 + b_{j_2j_1}) \right] \quad (12)$$

For  $j_2 = 1, 2, \dots, N_2$

The output of the  $j^{\text{th}}$  neurons in the  $x^{\text{th}}$  hidden layer is given as: (Eq. 13)

$$Y_{jx}^n = f \left[ \prod_{j_{x-1}=1}^{N_{x-1}} (W_{jxj_{x-1}} Y_{j_{x-1}}^{x-1} + b_{jxj_{x-1}}) \right] \quad (13)$$

For  $j_x = 1, 2, \dots, N_x$

The output of the  $k^{\text{th}}$  neuron in the output layer is given as: (Eq. 14)

$$Y_k = f \left[ \prod_{j_x=1}^{N_x} (W_{kj_x} Y_{j_x}^x + b_{kj_x}) \right]; \quad (14)$$

For  $k = 1, 2, \dots, k$

A simple gradient decent rule, using a mean square error function is used for computation of weight update: (Eq. 15)

$$E_{\text{MSE}} = \frac{1}{2PK} \sum_{k=1}^k \sum_{p=1}^p (y_{dk}^p - y_k^p)^2 \quad (15)$$

where,  $y_k^p$  and  $y_{dk}^p$  are the actual and desired values, respectively, of the output of the  $k^{\text{th}}$  neuron for the  $p^{\text{th}}$  pattern in the output layer.  $P$  is the number of training patterns in the input space. The weights are updated as below. Weights between output layer and the  $n^{\text{th}}$  hidden layer are given by: (Eq. 16-18)

$$\begin{aligned} \Delta w_{k_jx} &= \eta \frac{\partial E_{\text{MSE}}}{\partial w_{k_jx}} \\ &= \eta \delta_k \frac{\left[ \prod_{j_x=1}^{N_x} (w_{k_jx} y_{j_x}^x + b_{k_jx}) \right]}{(w_{k_jx} y_{j_x}^x + b_{k_jx})} \cdot y_{j_x}^x \end{aligned} \quad (16)$$

$$\delta_k = \frac{1}{PK} \left[ \sum_{k=1}^k \sum_{p=1}^p (y_{dk}^p - y_k^p) \cdot [(1/2)(1 + y_k^p)(1 - y_k^p)] \right] \quad (17)$$

$$\begin{aligned} \Delta b_{k_jx} &= \eta \delta_k \frac{\left[ \prod_{j_x=1}^{N_x} (w_{k_jx} y_{j_x}^x + b_{k_jx}) \right]}{(w_{k_jx} y_{j_x}^x + b_{k_jx})} \\ &= \frac{\Delta w_{k_jx}}{y_{j_x}^x} \end{aligned} \quad (18)$$

Weight between  $x^{\text{th}}$  and  $(x-1)^{\text{th}}$  hidden layer is: (Eq. 19 and 20)

$$\begin{aligned} \Delta w_{j_{x-1}} &= -\eta \frac{\partial E_{\text{MSE}}}{\partial w_{j_{x-1}}} \\ &= \frac{\eta}{PK} \left[ \sum_{k=1}^k \sum_{p=1}^p (y_{dk}^p - y_k^p) \cdot \frac{\partial y_k^p}{\partial w_{j_{x-1}}^n} \right] \cdot \frac{\partial y_{j_x}^x}{\partial w_{j_{x-1}}} \\ &= \eta \delta_k \frac{\left[ \prod_{j_x=1}^{N_x} (w_{k_jx} y_{j_x}^x + b_{k_jx}) \right]}{(w_{k_jx} y_{j_x}^x + b_{k_jx})} \cdot \Delta w_{k_jx} \cdot \frac{\partial y_{j_x}^x}{\partial w_{j_{x-1}}} \end{aligned} \quad (19)$$

$$\Delta b_{k_jx} = \frac{\Delta w_{k_jx}}{y_{j_x}^x} \quad (20)$$

Similarly, we can write equations for weight change between the hidden layer 1 and the input layer. The weights and biases are updated as: (Eq. 21 and 22)

$$w_i^{\text{new}} = w_i^{\text{old}} + \Delta w_i \quad (21)$$

$$b_i^{\text{new}} = b_i^{\text{old}} + \Delta b_i \quad (22)$$

## RESULTS AND DISCUSSION

The equalizer structure was trained with 3000 iterations and tested over 10000 samples to study the BER performance. A fading channel is a communication channel that experiences fading due to multipath propagation. In wireless communications, the presence of reflection in the environment surrounding the transmitter and receiver create multiple paths that the transmitted signal can traverse. At the receiver there is a superposition of these multipath signals which

experience different attenuation, delay and phase shift. This can result in either constructive or destructive interference, amplifying or attenuating the signal power seen at the receiver. Strong destructive interference is known as deep fade. The fading process is characterized by a Rayleigh distribution for a nonline-of-sight path. The coherence time of the channel is related to the quality known as Doppler spread of the channel. When the user or the reflectors in the environment are mobile, the users velocity causes a shift in the frequency of the signal transmitted along each signal path. The difference in Doppler shifts between different signal components contributing to a single fading channel tap is known as Doppler spread. The coherence time is inversely proportional to the Doppler spread and is given by: (Eq. 23)

$$T_c = \frac{K}{D_s} \quad (23)$$

Where:

- TX = The coherent time
- DZ = The Doppler spread and
- K = Constant taking on values between 0.25-0.5

In flat fading, the coherence bandwidth of the channel is larger than the bandwidth of the signal. Therefore, all frequency components of the signal will experience the same magnitude of fading. In our experiments we have simulated frequency-flat (“single path”) Rayleigh fading channel as a linear FIR filter, with tap weights given by: (Eq. 24)

$$g_x = \sin c\left(\frac{\tau_k}{T-x}\right)h_k \text{ for } -N_1 \leq x \leq N_2 \quad (24)$$

The summation has one term for each major path.  $\tau_k$  is the set of path delays and T is the input sample period.  $N_1$  and  $N_2$  are chosen so that  $|g(n)|$  is small.  $N_1$  determines the channel filter delay.  $h_k$  is the set of complex path gains which are not correlated to each other.

The received signal in Rayleigh fading channel is of the form: (Eq. 25)

$$y = hx + v \quad (25)$$

where, h is the complex scaling factor corresponding to Rayleigh multipath channel, x is the BPSK transmitted symbol and v is the AWGN noise. The simulation results are shown in the Fig. 5-8 where the SNR is 25db.

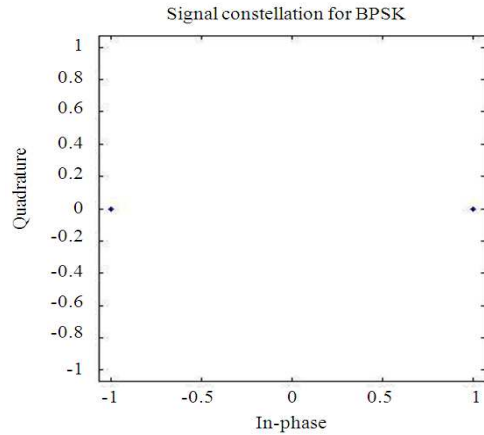


Fig. 5: Signal constellation for BPSK

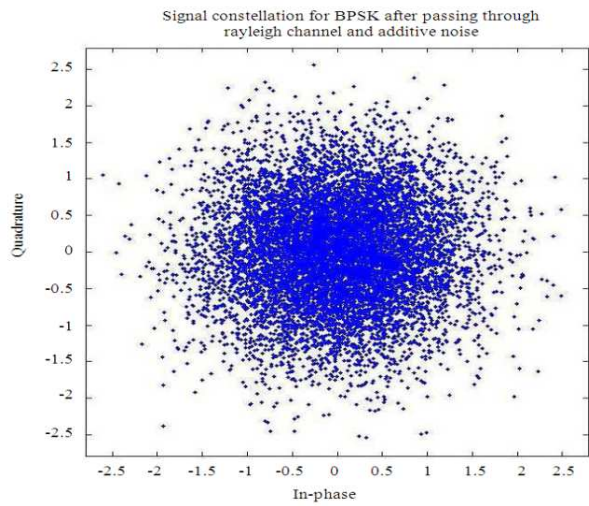


Fig. 6: Received noisy constellation at 25 db SNR

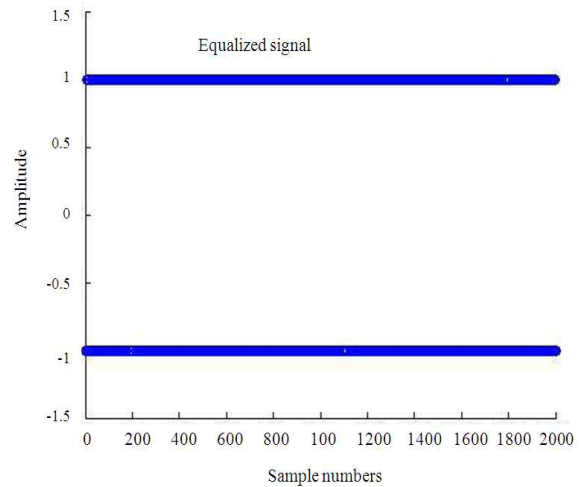


Fig. 7: Equalized signal samples at 25 db SNR

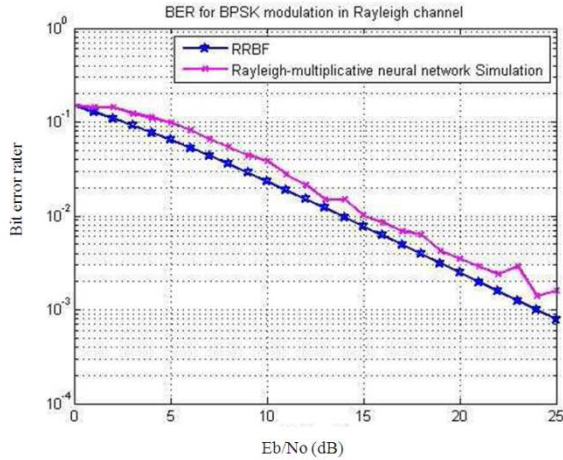


Fig. 8: BER Vs SNR for Rayleigh channel

### CONCLUSION

A high order feed forward network equalizer with multiplicative neuron is proposed in this study. Use of multiplication allows direct computing of polynomial inputs and approximation with fewer nodes. Performance comparison in terms of network architecture and BER performance suggest the better classification capability of the proposed MNN equalizer over RRF.

### REFERENCES

Bang, S., S.H. Sheu and J. Binh, 1996. Neural network for detection of signals in communication. *IEEE Trans. Circuits Syst.*, 43: 644-655. DOI: 10.1109/81.526680

Burse, K., R.N. Yadav and S.C. Shrivastava, 2008. Complex channel equalization using polynomial neuron model. *Proceeding of the International Symposium on Information Technology*, Aug. 26-28, IEEE Xplore Press, Kuala Lumpur, Malaysia, pp: 1-5. DOI: 10.1109/ITSIM.2008.4631647

Burse, K., R.N. Yadav, S.C. Shrivastava and V.P.S. Kirar, 2009. . *Proceeding of the WCSET. World Cong. Sci. Eng. Technol.*, 50: 231-234.

Gibson, G.J., S. Siu and C.F.N. Cown, 1989. Multilayer perceptron structures applied to adaptive equalizers for data communications. *Proceeding of the International Conference on Acoustics, Speech, and Signal Processing*, May 23-26, IEEE Xplore Press, Glasgow, UK, pp: 1183-1186.

Haykin, S., 2002. *Adaptive Filter Theory*. 1st Edn., Pearson Education India, Delhi Kindersley, ISBN8131708691, pp: 936.

Kantsila, A., M. Lehtokangas and J. Saarinen, 2004. Complex RPROP algorithm for neural network equalization of GSM data bursts. *Neurocomputing*, 61: 339-360 DOI: 10.1016/j.neucom.2003.11.007

Kim, T. and T. Adalai, 2002. Fully complex multi-layer perceptron network for nonlinear signal processing. *J. VLSI Sign. Proc.*, 32: 29-43. DOI: 10.1023/A:1016359216961

Lyoda, E.M., K. Hirota and F.J. Von Zuben, 2002. Sigma-Pi cascade extended hybrid neural networks. *J. Adv. Computa. Intel.*, 6: 126-134.

Prokias, J.G., 1995. *Digital Communication*. 3rd Ed., McGraw Hill, New York, ISBN: 0071138145, pp: 928.

Schmitt, M., 2002. On the complexity of computing and learning with multiplicative neurons. *Neural Comput.*, 14: 241-301 DOI: 10.1162/08997660252741121

Taba, M.T, S. Femmam and M. Bedda, 2006. Fast iterative frequential equalization applications to mobile communications. *Am. J. Applied Sci.*, 3: 2103-2107. DOI: 10.3844/ajassp.2006.2103.2107

Yadav, R.N., P.K. Kalra and J. John, 2007. Time series prediction with single multiplicative neuron model. *Applied Soft Comput.*, 7: 1157-1163. DOI: 10.1016/J.ASOC.2006.01.003

Zhang, Q., 1990. Adaptive equalization using the back propagation algorithm. *IEEE Trans. Circ. Syst.*, 37: 848-849. DOI: 10.1109/31.55048