BER Analysis of Neural Equalizer in OFDM systems

Jaswanth V, S. Vasanthadev Suryakala*, S. Kolangiammal, Nikil*, Dinesh Reddy G, Trived Sai G

Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, India.

*Email: suryakas@srmist.edu.in, kolangis@srmist.edu.in

Abstract: Digital communication gives larger data capacity and best communication speed. The signal transmitted through the wireless channels in digital communication affected by nonlinearities like noise. Inter Symbol Interference (ISI) dominantly causes huge data loss. It is caused due to multipath propagation, band-limited channels noise. Equalization, a process of inducing inverse channel response to compensate the effects of ISI is used to combat these effects. Orthogonal Frequency Division Multiplexing (OFDM) system supports high data rates and offers resistance towards the channel effect. Adding equalizer to OFDM systems make it more robust. In real time situations the response of channel is not known in prior. So, adaptive equalizers are implemented to compensate channel effects with the help of pilot symbols which are used to estimate the channel response and equalize the signal accordingly. Minimum Mean Square Error (MMSE), Least Square (LS) are some of the existing algorithms used to implement the equalizers. These conventional algorithms require some statistical properties of channel which is not possible in real time scenarios. Also, Linear Transverse equalizer failed to provide satisfactory results in the case of Non Minimum Phase channel. Neural Networks is capable to perform this task and can be used as an alternative to these algorithms. Back Propagation Neural Networks (BPNN) and Deep Neural Networks (DNN) can achieve better results when compared to conventional algorithms. Here, Deep Neural Networks with Adam optimization is proposed to enhance the performance of equalization. DNN can be used in both linear and non-linear channels. Also, Adam optimization used with Back Propagation uses variable and adaptive learning rate which reduces slow convergence and increases the efficiency of neural network unlike stochastic gradient descent which uses fixed learning rate.

In this work the performance of proposed equalizer in OFDM systems is analyzed over flat and frequency selective fading of Rayleigh channels. The performance is measured in terms of Bit Error rate over SNR range -30dB to 25 dB. The results are compared with that of MMSE and it is observed that over the mentioned range of SNR the proposed equalizer achieved acceptable results and has low Bit error rate than that of MMSE in both flat and frequency selective fading channels.

1. Introduction

COMMUNICATION is the most important aspect of our day to day lives. To ensure proper communication between the sender and receiver the design of communication systems should be reliable. Most of the communication today is achieved through wireless and so no physical medium is present between sender and receiver. Advancement in technology and invention of mobile phones increased the reliability of wireless systems. Also demand for higher data rates and broadband communication is increased. Multicarrier schemes of Modulation are more concerned to meet these requirements. Among them OFDM is given high priority for its resistance towards multipath fading. Due to absence of physical medium, the transmitted signal may undergo distortion generally caused due to presence of barriers

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Published under licence by IOP Publishing Ltd
such as hilly terrain, tall buildings, uneven surfaces between the end devices and these barriers causes the signal to reflect, refract and scatter, so signal is received as different samples and at different times due to path difference. This effect is known as multipath fading. This can be of type Rayleigh Fading or Rician Fading. All these effects cause distortion and so equalization is used to compensate these effects.

There are several adaptive equalization algorithms like RLS, LMS, Decision Feedback Equalizer explained in [1] to implement the equalizers. These algorithms can be used for both linear and nonlinear scenarios. All these algorithms utilize LTE structure for implementation. But it has some limitations in some cases. The channel model in the real time can be of minimum phase or of non-minimum phase. Using LTE structure desired results cannot be obtained in the case of non-minimum phase channel [2]. Also, LTE is not used in the presence of non-linearity. Decision feedback Equalizers are used in that case. Even though there is a problem of error propagation with high modulation orders. So, an alternate approach is required.

In this work, deep learning is used to implement the equalizers. Deep learning is a network that has capability of extracting feature that relates the stimuli and neural responses of brain. Many literatures proved that ANN can achieve desired results as an equalizer [2-6]. Also it can be used for linear and nonlinear frameworks. Here, Deep Neural Network with variable learning parameter is used. DNN can achieve better results than ANN as explained in [7]. DNN is different from ANN in terms of number of hidden layers are used. DNN uses more than 2 hidden layers. The variable learning parameter is selected by Adam Optimizer. This helps in faster convergence and enhances the performance of equalizer [8, 9]. OFDM has resistance towards ISI and has high data rates and so OFDM is chosen here. The equalizer is applied to the system in Rayleigh fading conditions and BER is calculated for both flat and frequency selective fading.

2. Methodology

System architecture, Deep Neural Networks and Adam optimization are explained in this section.

2.1 System Design:

![Figure 1: Schematic representation of System model](image)

The proposed system model is similar to conventional OFDM communication systems. Estimation or Equalization is done using Deep Neural Network instead of conventional algorithms. OFDM involves in different stages of processing the input signal. The functionality of OFDM lies in FFT and IFFT conversions. These stages are driven by parallel data streams. Initially the data to be transmitted is generated and required modulation is applied. The data is then driven into parallel stream and provided to subcarriers. Number of subcarriers define the length of IFFT and FFT blocks. The parallel data is then shifted to time domain using IFFT. The parallel stream is now converted back to series and sent through channel. Now in receiver the signal is applied to FFT to shift back to frequency domain. Now estimation and equalization is applied and demodulated. The equalization is obtained with the help of pilot symbols present in the input stream of data. Pilots can be arranged in different ways. They can be provided at regular intervals or in a random manner. These are the data symbols known to both sender and receiver. So, based on the output obtained for these data symbols the equalizer is trained and error is reduced.
2.2 Deep Neural Networks (DNN)

The structure of DNN is similar to Multi-Layer Perceptron [10] where neurons are segregated in the form of layers. Here DNN is designed such that it has an input layer, output layer and three hidden layers. Each layer has an initiated weight and followed by an activation function. The steps involved in training DNN is explained in the following sections.

2.2.1 Batch Normalization

The training data to the network is provided in the form of batches. It is performed before the batch data is provided at the input layer. And this step is repeated at every layer. Batch Normalization is a technique used in neural network to standardize the input data. This process is also known as whitening. The distribution of data is standardized to mean 0 and standard distribution 1 at every layer. The data available to a layer in the network is the weighted sum output of the previous layer and so the data may be deviated from mentioned mean and variance. Hence, normalization is done at every layer. This ensures the distribution of data is same throughout the training. The weights are updated using back propagation with necessary algorithms. So, the error estimate is propagating back in every step. When the weights are update in this manner the data distribution parameters are changed and every time a perception of new training data arises. This is known as covariate shift. This leads to many factors such as low learning rate, increase in number of epochs etc which retards and prolongs the training process. To avoid these effects batch normalization is performed.

2.2.2 Back Propagation

The input provided initially is fed forward through all the layers of network and the loss is calculated based on predicted value and target value at the output using formula:

\[ L = \frac{1}{N} \sum (\hat{x}(k) - x(k))^2 \]  

(1)

![Figure 2: Representation of Deep Neural Network](image)

The output of a layer is passed to activation function before moving to next layer and hyperbolic tangent is used here. In this step the loss is propagated in reverse direction ie from output layer to the input layer. Now, the weights are to be updated in order to reduce this loss. For this calculated loss is propagated back in the network. The loss obtained in the final layer is the total loss of network. Every node contributes to certain
loss in the whole. So, in order to reduce the loss function the weights should be updated based on the extent of loss to which the node contributes. This is the reason for backward error propagation. And the contribution of loss at a node is determined using the partial derivative of the loss function. Based on initial weight at the corresponding node and partial derivative obtained the weights are updated using the following formula.

\[ W = W_i - \eta \cdot \frac{dL}{dW} \]  

(2)

Here, the learning rate parameter \( \eta \) is determined using Adam optimization unlike conventional Back Propagation Neural Network based on stochastic gradient descent which uses fixed value in the whole process. The process of Adam Optimization and how learning rate is adapted is explained in the next section. This step is repeated for number of times so, that loss is minimized. A single iteration performed is known as epoch and the number of epochs is determined based on the convergence factors. In this model, the network is preset for 1000 epochs.

![Figure 3: Representation of DNN algorithm.](image_url)
ADAM optimization

A good choice of optimizer for the deep learning algorithm can mean a difference in days, hours, minutes. One such effective optimizer that we use in our deep learning model is the Adam optimizer. There are many factors which affect the training of the data sets for obtaining target value. Saddle points, local minima, learning parameter are some of them. Here the optimization is considered with respect to the learning parameter. Stochastic gradient descent has been used for many years in machine learning, computer vision etc. In that method the learning rate of the network is kept constant throughout the process. So, in order not to miss the minimum local gradient low rate should be followed. This prolongs the training time. That algorithm follows first order moments. So, it can give information only about the speed with which the loss is declining and is completely unknown about the properties of curvature like shape and direction of movement. This problem can be addressed using higher order moments. Adam optimization is one such optimization technique. This adds the advantage of adapting learning rate of the network. This optimization includes information of gradients varying itself. The past gradients are used along with the present to specify the direction of optima. The RMS Prop and Momentum heuristics are used so that minima direction is specified and oscillations are prevented. Here gradient squares and exponential average is calculated. The learning step to be varied is obtained by multiplying learning rate with gradient average and dividing with rms of exponential average. If \( W \) is a parameter then for each parameter the concerned equations are given as follows.

\[
V_t = \beta_1 V_{t-1} - (1 - \beta_1) g_t \\
S_t = \beta_2 S_{t-1} - (1 - \beta_2) g_t^2 \\
\Delta W_t = -\eta \frac{V_t}{\sqrt{S_{t-\epsilon}}} \\
W(t + 1) = W_t + \Delta W_t
\]

Where \( \eta \) is the learning rate that is specified initially, \( \beta_1 \) is the gradient exponential average, \( \beta_2 \) is the gradient squares exponential average and \( \epsilon \) is the small constant to avoid divided by zero error.

3. Design Parameters

The design parameters used in this study is discussed in this section.

3.1 OFDM

The data is generated in random initially. Then data is mapped and modulated using 16 QAM modulation. 8 pilots per OFDM block are added to data streams. FFT and IFFT size is set to be 64 and length of cyclic prefix added is set to be 16. The design parameters used for OFDM is tabulated in Table 1.

Table1: Simulation parameters of OFDM

| Parameter                  | Value          |
|----------------------------|----------------|
| Modulation                 | 16QAM          |
| Cyclic Prefix length       | 16             |
| FFT size                   | 64             |
| Number of pilots per OFDM block | 8              |
| Pilot pattern              | Scattered type |
| noise type                 | AWGN           |

3.2 DNN
MLP structure is used to implement the network. The proposed DNN structure has three hidden layers one input layer and one output layer. Non-linear function tanh is used for activation in the hidden layers. The output layer is provided linear. Number of neurons in each layer and other parameters are mentioned in the Table 2.

### Table 2: Simulation parameters of DNN

| Parameter                   | Value                        |
|-----------------------------|------------------------------|
| Number of hidden layers     | 3                            |
| Number of neurons in hidden layers | 500,250,120                  |
| Activation function used    | tanh                         |
| Algorithm                   | DNN with Back Propagation    |
| Optimization                | Adam                         |

The entire data is divided into training and testing data sets. Pilot symbols are used for training and the process is carried in the form of batches. Each batch is formed with 256 pilot symbols. This set of 256 includes both real and complex components of symbol. The target data and training data here is in the complex form. The simulation software does not accept data in complex form as input to the neural net. So, the real and complex parts are given to the input as two different variables. After the training they are added again. Hence both are provided as real variables and trained individually.

### 4. Results and Discussion:

Tensor Flow and Matlab simulation software are used for simulation of results. The proposed OFDM system with DNN equalizer is implemented using Tensor Flow and Bit Error Rate is calculated. These calculated values are considered and finally compared with MMSE through plots using Matlab. The simulation is performed for two different fading types (i) Flat fading and (ii) Frequency selective fading. Table 3 summarizes the obtained BER values for various SNR for both DNN equalizer and MMSE equalizer over flat fading scenario.

### Table 3: SNR Vs BER of DNN and MMSE equalizer for flat fading channel

| SNR (dB) | DNN Equalizer | MMSE Equalizer |
|----------|---------------|----------------|
| -10      | 0.46          | 0.98           |
| -5       | 0.40          | 0.92           |
| 0        | 0.29          | 0.90           |
| 5        | 0.15          | 0.85           |
| 10       | 0.064         | 0.76           |
| 15       | 0.025         | 0.58           |
| 20       | 0.007         | 0.30           |
| 25       | 0.003         | 0.06           |

The performance of DNN equalizer over conventional MMSE equalizer is shown in Figure 4 for flat fading channel.
Form the graph it is clear that DNN performance is much better than that of MMSE in the SNR range -10 to 25.
The BER of DNN equalizer and MMSE equalizer for different SNR values is listed in Table 4.

Table 4: SNR Vs BER of DNN and MMSE equalizer for frequency selective fading channel

| SNR (dB) | BER |  |
|----------|-----|---|
|          | DNN Equalizer | MMSE Equalizer |
| -10      | 0.44 | 0.98 |
| -5       | 0.37 | 0.93 |
| 0        | 0.29 | 0.91 |
| 5        | 0.23 | 0.84 |
| 10       | 0.20 | 0.77 |
| 15       | 0.19 | 0.59 |
| 20       | 0.18 | 0.30 |
| 25       | 0.09 | 0.13 |

From Table 4 it is evident that the BER is greater for MMSE equalizer when compared to DNN equalizer. The BER plot of DNN equalizer over conventional MMSE equalizer is plotted in Figure 5 for frequency selective fading channel.

In this plot BER is compared for both the equalizers over frequency selective fading. It is evident that in this case also DNN equalizer performs better in the SNR range -10 to 20 dB. But for the higher SNR values MMSE outperforms the DNN equalizer.

5. Conclusion
This proposed study is about the performance of DNN based equalizer over conventional equalizer. The OFDM system is implemented with the proposed DNN structure at the receiver end. The whole system is implemented in Tensor Flow. The equalization is done with the help of pilots. They are
used to provide Channel Response Information at the receiver. Less number of pilots are used and so bandwidth can be used effectively. Finally BER is calculated in flat fading and frequency selective fading channels. The graphs show that desired results are obtained with the use of DNN as an equalizer. The results are compared to equalization using MMSE and the proposed architecture outperforms the conventional one.

References

[1] Malik, G. and Sappal, A. S , “Adaptive equalization algorithms: An overview”, International Journal of Advanced Computer Science and Applications, 2(3), 2011.
[2] Parida, A. K., Panda, S., and Singh, R. P, “A strategy on channel equalization for digital communication based on neural network system.” 2019 5th International Conference on Advanced Computing Communication Systems (ICACCS). 152–156, 2019
[3] Makka, J., Monga, H., and Baghla, S, “Reduction of inter-symbol interference using artifical neural network system in multicarrier OFDM system.” International Journal of Wireless and Microwave Technologies, 8, 10–18, 2018.
[4] Rahman, Q. M., Ibnkahla, M., and Bayoumi, M, “Neural network based channel estimation and performance evaluation of time varying multipath satellite channel.” 3rd Annual Communication Networks and Services Research Conference (CNSR’05). 74–79, 2005.
[5] Zerdoumi, Z., Chikouche, D., and Benatia, D, “Neural networks based equalizer for signal restoration in digital communication channels.” International Letters of Chemistry, Physics and Astronomy, 55, 187–200, 2015.
[6] Raghavendra, K. and Tripathy, A, “An efficient channel equalizer using artificial neural networks.”Neural Network World, 16, 2006.
[7] Juergen Schmidhuber, “Deep Learning in Neural Networks: An Overview”. Neural Networks, Vol 61, pp 85–117, 2015.
[8] Zaheer, R. and Shaziya, H. (2019). “A study of the optimization algorithms in deep learning.”2019 Third International Conference on Inventive Systems and Control (ICISC). 536–539.
[9] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,”arXiv preprint arXiv: 1412.6980, 2014.
[10] Zahid, M. and Meng, Z, “Recent advances in neural network techniques for channel equalization: A comprehensive survey.” 2018 International Conference on Computing, Electronics Communications Engineering (iCCECE). 178–182, 2018.