Machine Learning Equalization Techniques for High Speed PAM4 Fiber Optic Communication Systems

I. Lyubomirsky

CS229 Final Project Report, email: lyuboptics@gmail.com

Abstract *In this work, we apply machine learning regression and classification algorithms to the problem of equalization and detection in high-speed fiber optic communication systems.*

Introduction

Conventional linear feed forward equalizers (FFEs) are widely used in fiber optic communication systems to mitigate inter-symbol interference resulting from a limited bandwidth of electro-optic components, as well as fiber chromatic dispersion [1]. However, linear equalization techniques (analogous to linear regression in machine learning) provide only a limited benefit for the low-cost direct detection receivers used in Ethernet applications [2,3]. Chromatic dispersion is a major transmission impairment in these systems. Dispersion acts on the electric field while the square law direct detection receivers detect intensity; all optical phase information is lost during the direct detection process. Moreover, optical amplifiers generate additive white Gaussian noise (AWGN) in the optical field, but after going through square law detection, the noise is no longer AWGN. When significant chromatic dispersion is combined with optical noise, the electrical waveform after photo-detection can be so distorted that a linear FFE is not sufficient to provide an acceptable symbol error rate. In this work, we investigate and compare several machine learning algorithms for applications in digital equalization and detection, with the goal of discovering new techniques which are more robust to non-Gaussian noise statistics, and the nonlinear distortions due to interaction of chromatic dispersion and square law detection. As with a linear FFE, the input features are taken from N consecutive samples of the received waveform (sampling rate is 2x the baud rate), with N ~ 5 to 40. At the high speeds of interest to fiber optic communication systems, the equalizer complexity and memory requirements put a major constraint on the system design. Hence, we focus on regression and classification algorithms with an eye toward practical implementation in DSP ASICs.

System Model

Figure 1 shows the system model and simulation setup. We simulate a 4-ary pulse amplitude modulation (PAM4) fiber optic communication system. The symbol rate is 56 Gbaud for a bitrate of 112 Gb/s; this bit rate includes native 103 GbE and overhead for FEC. The PAM4 optical modulator is driven by 2 binary electrical data streams, with each pair of bits corresponding to a PAM4 symbol taken from the alphabet {-3, -1, 1, 3}. We employ pseudo-random bit patterns (PRBS) with length of 2^{16}-1 symbols to drive the modulator. Training and test data are generated using different

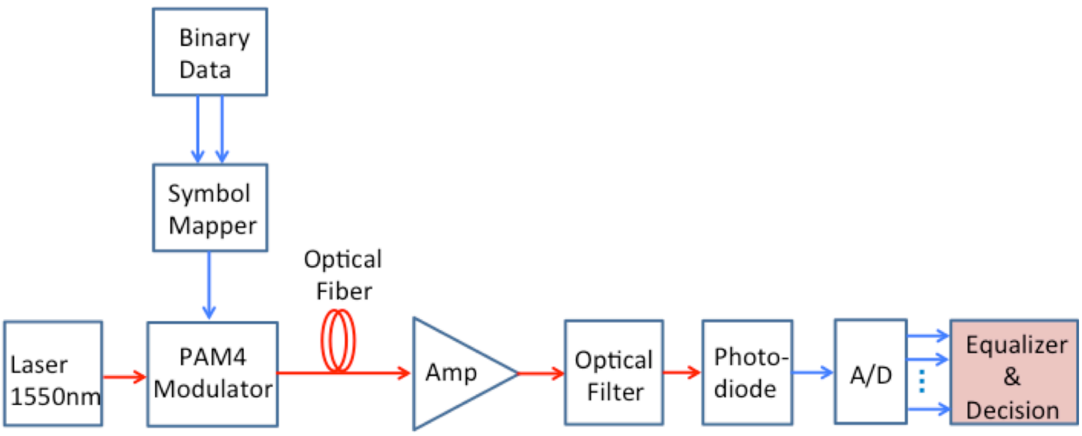


Fig. 1: System simulation block diagram. The highlighted block for equalizer and decision device is the focus of this paper.

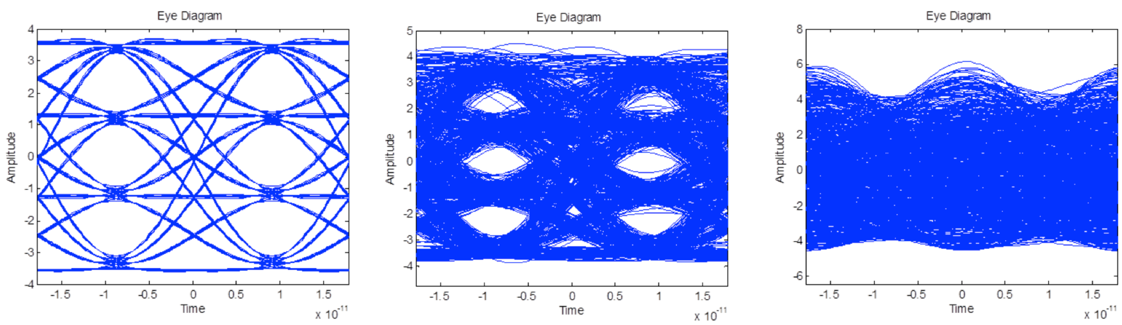
PRBS polynomials, and different random seeds for the noise generators. The output of the modulator propagates through a fiber optic link; the main optical channel impairments are due to fiber loss and chromatic dispersion. At the receiver, an optical amplifier boosts the signal to compensate the loss, while also adding white Gaussian noise to the optical field. An optical filter is used to reduce the optical noise entering the receiver. A photo-diode converts the optical signal to an electrical current through square law detection. The electrical signal is then sampled by an analog-to-digital converter (assumed ideal) at 2 samples per symbol. Finally, the samples are processed by a digital equalizer and a decision circuit. The equalizer and decision circuit block is the main focus of this work. The optical system simulations are performed using the commercial software package OptSim, with simulation in Matlab for the receiver digital signal processing.

Figure 2 shows some simulated eye diagrams at the receiver. An eye diagram displays all the waveform symbols simultaneously by overlaying in the same plot. An ideal PAM4 eye diagram would show wide open "eyes," as in Fig. 2 a). for the case of no noise and no dispersion. The PAM4 eyes close up after adding noise from the optical amplifier as shown in Fig. 2 b). We also observe more noise on the higher PAM4 levels; this is due to the action of square law detection on the optical field as discussed above. When both noise and significant chromatic dispersion are included in the simulation, the eyes completely close, as shown in Fig. 2 c). where we included chromatic dispersion corresponding to ~ 4km of standard single mode fiber. Although it's hard to discern a pattern in the noise-like waveform of Fig. 2 c)., a powerful machine learning algorithms is able to detect the pattern of transmitted symbols, as shown below.

Features

Figure 3 a). shows the block diagram for a conventional linear equalizer based on an FIR filter structure. We can also think of the linear equalizer as a linear predictor (or linear regression), where the features are N input samples, and the linear predictive model is defined by N+1 parameters learned during training by minimizing the MSE. The output of FIR filter goes to a decision device, which makes hard decisions based on minimum Euclidean distance to the nearest PAM4 symbol. We generalize the linear equalizer to the structure shown in Figure 3 b)., where the FIR filter processing is replaced with a more general machine learning algorithm for predicting the output symbol. The input features of the machine learning algorithm are the same N samples but the algorithm may be based on either regression (followed by hard decision) or classification giving the predicted PAM4 symbols directly.

Figure 4 shows a feature correlation heat map of 9 samples for a). no dispersion and b). with dispersion. The correlation heat map shows that neighbouring samples are strongly correlated due to the pulse spreading in time caused by dispersion. Hence, algorithms which rely on a statistical independence of the features, such as Naive Bayes, may not be suitable for this problem. Figure 4 c). and d). show pairwise scatter diagrams for 5 samples for the case of no dispersion, and with dispersion, respectively. The points are labelled by different colors corresponding to the received PAM4 symbols. Note the output PAM4 symbols are time aligned with the center sample. As revealed in Fig. 4 c). and d)., without dispersion, it should be easy to separate the classes;



a). b). c).

Fig. 2: Simulated eye diagrams for a). no noise, no dispersion; b). including optical noise but without chromatic dispersion; c). including both optical noise and chromatic dispersion.

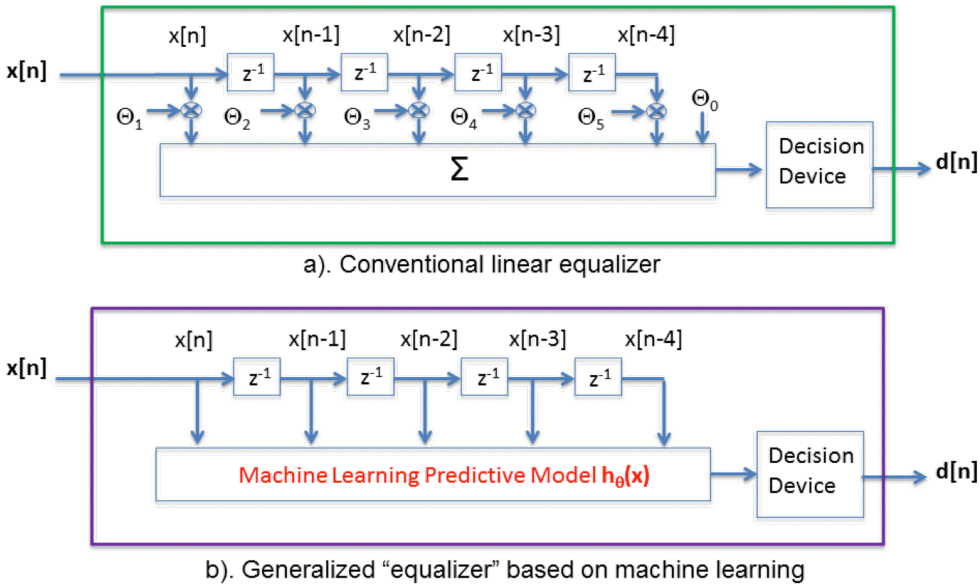


Fig. 3: Block diagram of a). conventional linear equalizer, and b). generalized equalizer based on machine learning.

however, fiber dispersion tends to mix the different colored clouds together, making the regression and/or classification problem very challenging. A nonlinear regression or classification algorithm is required to separate the classes in this case.

Machine Learning Algorithms

All algorithms are implemented in Matlab using the Statistics and Machine Learning Toolbox, and Neural Network Toolbox. Linear regression is the baseline algorithm for comparison, as this is the simplest and most popular approach in digital equalization. The strait forward generalization of linear regression to nonlinear systems is polynomial regression, also called Volterra equalization in the communications literature [4]. For simplicity, we consider only second order Volterra equalization, where the output of the equalizer is given by

$$h_{\theta,\beta}(x) = \theta_0 + \sum_{k=0}^{N-1} \theta_i x[n-k] + \sum_{k=0,l=0}^{N-1} \beta_{kl} x[n-k] x[n-l]$$

The parameters {θ,β} are learned during training by minimizing MSE. We use the Matlab **fitlm** function for linear and polynomial regression. Perhaps one of the most powerful algorithms for nonlinear regression is based on Neural Networks. We used the Matlab **feedforwardnet** function to model a Neural Network. A network design based on 2 hidden layers, with N neurons per layer, was found to work well by experimentation on a separate validation data set. The parameters of the Neural Network are learned in training using the Levenberg-Marquardt version of the back-propagation algorithm [5].

Classification algorithms offer a different approach to this problem by combining the functions of "equalization" and "detection" together to output the predicted received symbols directly. For classification algorithms, we tried softmax regression using the Matlab function **mnrfit,** with logit link function. In softmax regression [6], the output is a probability for each PAM4 symbol, and decision is made in favor of the highest probability symbol. Finally, we tried a K nearest neighbors (KNN) classification scheme [7] using the Matlab function **fitcknn**. For a given input vector X, the KNN classification algorithm estimates the probability P(m|X), where m is the PAM4 class label, from the K nearest neighbors to X in the training set based on the plurality of the K observations. K = 5 was found to be an optimum choice on a separate validation data set.

Results and Discussion

Figure 5 shows the simulation results on the test data set symbol error rate (SER) versus number of features (or samples) for the case where both noise and dispersion are included. Increasing the number of sample inputs generally improves algorithm SER performance as the span of the samples includes more of the pulse spreading due to dispersion. Once all the pulse spreading is included in the span of the N samples, increasing the number of samples further does not provide any additional benefit and SER saturates. KNN is an exception; it achieves optimum performance at N ~ 9 samples and then SER performance degrades rapidly due to

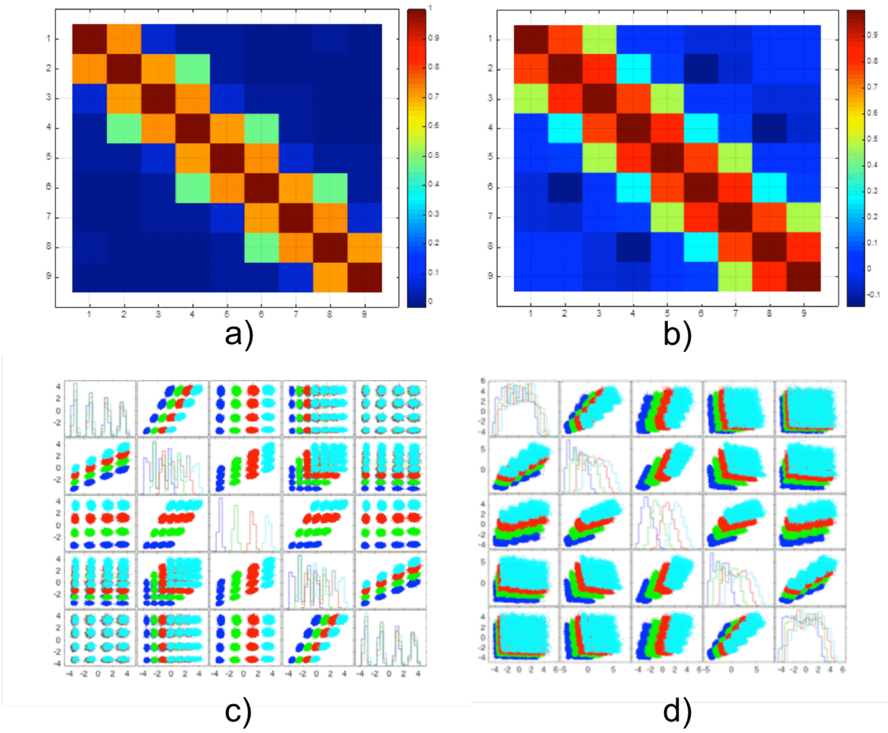


Fig. 4: Correlation heat maps for a). no dispersion, b). with dispersion, and pairwise scatter plots for c). no dispersion, d). with dispersion. The received PAM4 symbols (4 classes) are labeled by color in c). and d).

the well known problem called the “curse of dimensionality,” i.e. the K neighboring points are spread further apart in higher dimensions [7].

As expected, the simple linear equalizer gives the worst SER performance. Softmax regression yields only a small improvement over linear regression. For softmax regression, the logodds for each class is linear in the features, so one may expect similar results as in linear regression. The small improvement of softmax regression may come from improved robustness to non-Guassian noise statistics. The scatter diagrams in Figure 4 d). clearly show the need for nonlinear regression techniques. Indeed, the nonlinear Volterra equalizer significantly improves the SER over linear regression and softmax regression. We found by the process of backward selection [7] that good performance with Volterra equalization can be achieved with only the X_j*X_i nonlinear interaction terms added to the linear terms; this simplifies the hardware implementation considerably. The Neural Network equalizer also achieves good performance but at the expense of higher complexity. Interestingly, the simple KNN classification algorithm with K=5 achieves the best performance at small N (but not best overall). KNN may provide an interesting option if it can be implemented efficiently in DSP hardware using a lookup table.

Conclusion and Future work

Machine learning regression and classification algorithms offer a wealth of new ideas to explore for digital equalization and detection. In this paper, we’ve considered linear and polynomial regression, softmax regression, neural networks, and KNN classification. Extensions of this work may involve a more detailed study of optimum Neural Network architectures, including classification schemes. The time series nature of the equalization problem also offers the possibility of using feedback; for example, so-called decision feedback equalization provides the previously decided symbol (which may be in error) as an additional feature to the input samples. Finally, one may explore other powerful regression and classification techniques, such as based on support vector machines [6] or boosted regression trees [7]. Any machine learning algorithm developed for high-speed digital equalization must be simple enough to implement in a real-time DSP ASIC; this is a unique aspect of the equalization problem that makes it particularly challenging but also exciting.

References

[1] J. G. Proakis and M. Salehi, “Digital Communications,” McGraw-Hill, 5th Edition, New York, 2008.

[2] C. Cole, I. Lyubomirsky, A. Ghiasi, and V. Telang “Higher-order modulation for client optics,” IEEE Comm. Mag., vol. 51, no. 3, pp. 50-57, March, 2013.

[3] V. Curri, R. Gaudino, A. Napoli, and P. Poggiolini “Electronic equalization for advanced modulation formats in dispersion-limited systems,” IEEE Phot. Tech. Lett., vol. 16, issue 11, pp. 2556-2558, Nov., 2004.

[4] S. Benedetto and E. Biglieri, “Nonlinear equalization of digital satellite channels,” IEEE J. Sel. Areas Commun., vol. 1, no. 1, pp. 57–62, Jan.,1983.

[5] M. H. Beale, M. T. Hagan, H. B. Demuth, “Neural Network Toolbox User’s Guide,” The MathWorks, 2015.

[6] Andrew Ng, CS229 class notes, Stanford University, 2015.

[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, “An introduction to statistical learning,” Springer, New York, 2014.

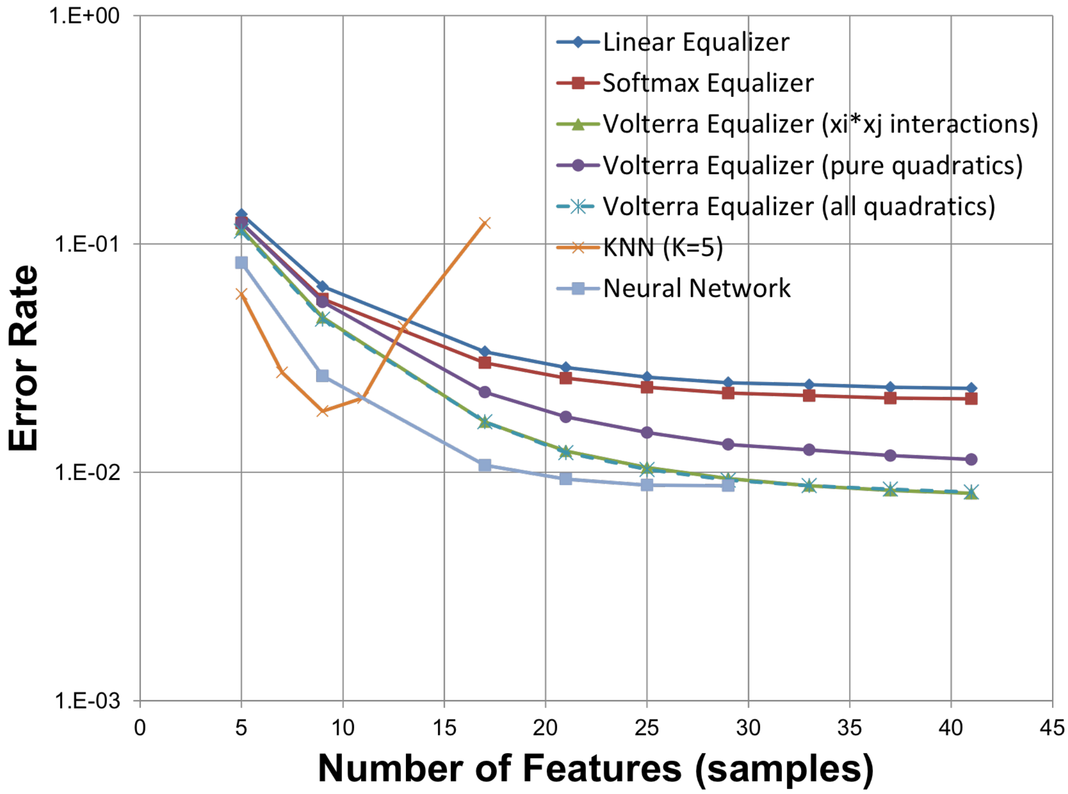


Fig. 5: Simulated symbol error rate (SER) versus number of features (or samples).