Letter

Vol. 45, No. 18 / 15 September 2020 / *Optics Letters* 5113

Optics Letters

Training data generation and validation for a neural network-based equalizer

Tao Liao, Lei Xue, Luyao Huang, Weisheng Hu, and Lilin Yi*

State Key Lab of Advanced Optical Communication Systems and Networks, Shanghai Institute for Advanced Communication and Data Science, Shanghai Jiao Tong University, Shanghai 200240, China

**Corresponding author: lilinyi@sjtu.edu.cn*

Received 26 March 2020; revised 18 July 2020; accepted 23 July 2020; posted 28 August 2020 (Doc. ID 393808); published 10 September 2020

The neural network (NN) has been widely used as a promising technique in fiber optical communication owing to its powerful learning capabilities. The NN-based equalizer is qualified to mitigate mixed linear and nonlinear impairments, providing better performance than conventional algorithms. Many demonstrations employ a traditional pseudo-random bit sequence (PRBS) as the training and test data. However, it has been revealed that the NN can learn the generation rules of the PRBS during training, degrading the equalization performance. In this work, to address this problem, we propose a combination strategy to construct a strong random sequence that will not be learned by the NN or other advanced algorithms. The simulation and experimental results based on data over an additive white Gaussian noise channel and a real intensity modulation and direct detection system validate the effectiveness of the proposed scheme.

https://doi.org/10.1364/OL.393808

The neural network (NN) has become popular and been shown to be effective in the fields of computer vision and natural language processing. It has been shown that an NN can fit and express any function if it has at least one hidden layer and enough hidden nodes [1]. Additionally, an NN can easily extract the features of big data and fit and express a complex nonlinear model. NN is thus widely used in applications of regression, classification, and decision making. In the case of a problem that cannot be solved with a simple mathematical model, the NN provides a more direct way to deal with the problem as a black box. Owing to its effectiveness, the NN has been widely investigated in fiber optical communications, and its value in optical network performance monitoring [2], proactive fault detection [3], modulation format recognition [4], and channel equalization [5–14] has been demonstrated.

The NN is applied as a promising equalization tool to compensate transmission impairments, including both linear and nonlinear distortion, especially in intensity modulation and direct detection (IMDD) systems. For both a passive optical network and short-reach optical links, owing to the requirement of low-cost optical devices, the bandwidth of the channel is inevitably limited, resulting in severe intersymbol interference. Moreover, this distortion can mix with other effects, such as nonlinearities of the optical devices, which further degrades the signal quality during transmission. In this case, conventional algorithms perform poorly and do not satisfy the requirements for equalization. Studies have demonstrated the capability of the NN in terms of equalization [5–13], with the NN-based equalizer seemingly outperforming traditional algorithms.

However, in-depth studies on the NN-based equalizer found that the performance of NN-based equalization may be overestimated owing to the use of a pseudo-random bit sequence (PRBS) [14]. When data of the same PRBS are used to train and test the NN, the NN will learn and use the generation rules of the PRBS, resulting in abnormally high performance [14–16]. References [14,15] reported that an NN with one hidden layer containing at least two nodes can learn PRBS rules. Even though the training data and test data have different patterns, the NN learns the generation rules instead of the data patterns, while the learned rules are not adaptive for other data with different generation rules or just real-world data. Therefore, once such different data are used to test the well-trained NN, the performance will be much worse [16]. The NN misled by the PRBS is thus not suitable for equalization. Besides, not only for NN, advanced algorithms such as the Volterra nonlinear equalizer can also learn the PRBS rule if its second-order size can cover the generation rule. Thus, effective methods of generating random data for training are essential. Some works apply more complex random data generated by the Mersenne twister algorithm to avoid this problem [16,17], but the data is still probably learned by a more advanced algorithm. Besides, real random data from the physical method is also a solution [18]. However, this way is difficult to conduct.

In this work, we establish a simple combination method of constructing random data. This method leverages three random sequences generated by any independent rules, where one sequence acts as an index to guide the combination of the other two sequences. The operation can be repeated time after time, continuously hiding the data features and eventually obtaining strong random data that cannot be modeled by an NN or other advanced equalization algorithms. A cross-test strategy is used to validate the effectiveness of the training data. Simulation and experimental results based on signals transmitted over an additive white Gaussian noise (AWGN) channel and a real IMDD

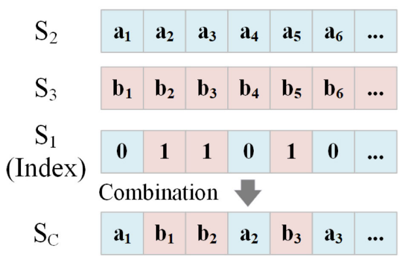

Fig. 1. Combination scheme for random sequence construction.

system confirm that the method effectively hides data features and ensure the correct training.

The proposed combination scheme is shown in Fig. 1. The method requires three independent pseudo-random sequences, such as three different PRBS sequences. One of the sequences, S_1, is selected as the index, while the other two sequences, S_2 and S_3, provide the bit values. Let S_C denote the combined sequence. The method will scan the bit in S_1. If the current bit is zero, then the bit at the head of S_2 will be shifted to the tail of S_C. Otherwise, the bit at the head of S_3 will be shifted to the tail of S_C. This operation will be conducted repeatedly until the length of S_C reaches the target. Besides, all of S_1, S_2, and S_3 are recycled during the procedure.

We adopt a PRBS to demonstrate how the proposed scheme hides the data feature and helps correctly train the NN-based equalizer. A basic PRBS is generated by linear feedback shift registers based on corresponding generating polynomials, which is essentially a simple exclusive-or (XOR) operation. For instance, the generation rule of PRBS15 can be expressed as

$$x(n) = x(n-15) \oplus x(n-14), \tag{1}$$

where $\oplus$ is the XOR operation, and x is the bit in the sequence with the index in parentheses. NN can characterize XOR relationships easily. Since these bits in a rule are close to each other, the input data of the NN-based equalizer can easily cover all of the related bits. Therefore, NN learns the PRBS rule and applies it in equalization, which brings an unreliable performance.

The combination scheme can complicate the random sequence to hide its rule. Let the source sequence S_1, S_2, and S_3 be PRBS sequences, then the combined sequence S_C is a mixed PRBS, assuming that S_C has a universal generation rule that will be learned by the NN-based equalizer, which is expressed as

$$x(n) = f(x(n-t_1), x(n-t_2), \ldots, x(n-t_k)). \tag{2}$$

It is impossible for all the bits to satisfy this rule, since the bits come from different sequences. However, if the NN-based equalizer can learn and leverage such a rule, then this rule should be general enough. Besides, only if the input data is large enough to cover all the related bits in the rule can an NN learn and model this rule.

As the bits in S_2 and S_3 are irrelevant, all the bits in the rule should originate from the same sequence, S_2 or S_3. Hence, the bits of index sequence S_1 are equal in the corresponding position, as

$$x_1(n) = x_1(n-t_1) = x_1(n-t_2) = \cdots = x_1(n-t_k) = v, \quad v = 0, 1, \tag{3}$$

where x_1 is the bit of S_1. If the rule of Eq. (2) originates from S_2, then $v = 0$, otherwise $v = 1$. Without loss of generality, let $v = 0$, then these bits form a subsequence with all zero. As we mentioned above, if the rule can be learned by NN, then the rule should be general enough.

If the span of the rule is within the length of S_1, then S_1 should contain a large number of such all-zero subsequences, while the bits in S_1 have the same distance of index. To make sure that the rule of Eq. (2) exists, the subsequences should appear periodically. However, S_1 is also a random sequence, so S_1 will not exhibit such periodic rule.

Therefore, if such a rule exists, the span must exceed the length of S_1. Since the source sequences are recycled during the combination, the index sequence S_1 can also be used repeatedly. It can form a cyclic index sequence S_1^*. Let d denote the length of S_1, then any two bits in S_1^* with a distance of d should be equal. Meanwhile, there should be a fixed amount of 0 and 1 between them. Hence, the corresponding bits in S_c should have a certain distance in its original sequence. A rule covering such bits with large distances may exist. In this case, the span of a rule must be greater than the length d.

In general, we can draw a conclusion that there is no short rule in the combined sequence S_c. If a rule exists, then the span of the bits in the rule should exceed the length of the index sequence. For a PRBS-N, its length is 2^N, which means even though the mixed PRBS has a learnable generation rule, the span of the rule will be exactly greater than 2^N. The length can easily exceed the size of the input for an NN-based equalizer. By the way, NN cannot learn the generation rule of the data and can be correctly well-trained.

In addition, the proposed scheme can work iteratively. It can employ the new combined sequence as one of the source sequences to construct a new sequence. The combined sequence has no direct correlation with the source sequences, thus they can work together in a new combination. The way can repeatedly increase the complexity of the new sequence. Any given advanced equalization algorithm as the NN should have a fixed ability. Therefore, the iteration can finally get a sequence that can effectively train the algorithm.

For instance, the cycle length of the combined sequence constructed by PRBS should be

$$l_c = \mathrm{lcm}(l_1, \mathrm{lcm}(l_2, l_3) \times 2), \tag{4}$$

where lcm is the least common multiple. l_1, l_2, l_3 are the length of S_1, S_2, S_3, respectively. Since $l_i = 2^{m_i}$, if we drop out a bit in each source sequence, then we can get a much larger l_c. Next, we can apply the new sequence S_c as the index sequence. According to the analysis above, the potential generation rule in the next new combined sequence should have a larger span that is greater than l_c. By the way, the sequence can effectively hide the data feature.

The proposed method is similar to the shuffle algorithm, but it is simpler to modify a bit sequence, due to it being able to cooperate with the original generation rule with an overhead of $O(N)$ complexity.

To validate the proposed scheme, several simulations are conducted to evaluate and compare the NN behavior on the mixed PRBS, basic PRBS, and Mersenne twister random data. The mixed PRBS is constructed with PRBS15, PRBS23, and PRBS31, where PRBS31 is the index. The Mersenne twister random data is generated by MATLAB following the Mersenne

Fig. 2. Sequences of the cross-test method.

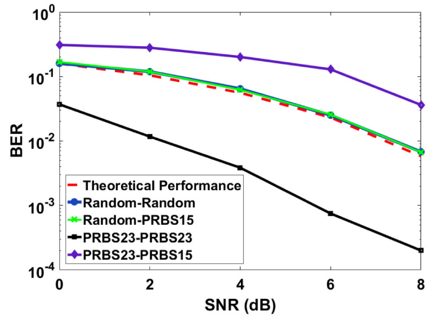

Fig. 3. BER performance of the NN on different training and test data combinations for different SNRs in basic cases. The notation A–B means training on A and testing on B.

twister rule. It has been proved that a common NN cannot learn the rule of Mersenne random sequence [17], so we use it to show a comparison.

In the simulation, a non-return-to-zero (NRZ) signal with only AWGN is generated as the experimental data to avoid the channel interference. In that way, the distortion of each bit is independent and random. We can just evaluate the influence of data rule detection. An NN with two hidden layers is used for evaluation, while the layer sizes are 201/128/128/2. The activation function of the hidden node is ReLU, while the output vector is activated by Softmax.

We apply a cross-test method to measure the performance of the NN-based equalizer. This method requires two data sets with different generation rules, as shown in Fig. 2. One is divided into the training set and test set, while another is only for test. The NN-based equalizer will be trained with the training set and tested with the two test sets. If the NN does not learn the data rules, then the bit error rate (BER) performances on the two test sets should be exactly equal.

We first verify this cross-test method with PRBS and Mersenne twister random data. We measure the performance of the NN on different training and test data combinations. Results are shown in Fig. 3, where notation of measurement data A–B means that the model is trained on A and tested on B. All of the training and test data for the same case belong to different data patterns, though some of the cases have the same generation rule as “A–A”. The “Random” denotes the signal based on Mersenne twister random data. The hard-decision curve represents the theoretically optimal performance of such an AWGN channel with a real random signal, which is leveraged as a benchmark.

The results show that, when the training data are Mersenne twister random data, all test BER performance curves are close to the theoretical curve, which confirms that such Mersenne twister random data will not be characterized by the evaluated NN model. However, when the training data are PRBS23, the test results present different performance behaviors on different test data. In the case PRBS23–PRBS23, the test data have the

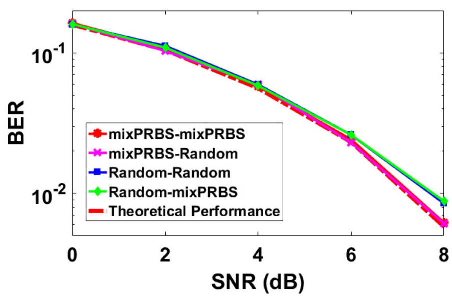

Fig. 4. BER performance of the NN on a mixed PRBS and Mersenne twister random data versus the SNR. The notation A–B means training on A and testing on B.

same generation rule as the training data, while the test BER performance is much better than the hard-decision performance. This abnormally high performance does not exist in the case of PRBS23–PRBS15, because the test data are generated by a different rule. The different performances on the same training data show that the NN learns many generation rules. In brief, internal data features of the signal are learned by the NN, which misleads the equalization.

Figure 4 shows the results of the cross test with the mixed PRBS and Mersenne twister random data. Using mixed PRBS or random data for NN training can get the same BER performance on different test data, while the performance is equal to the theoretical performance. This means that no internal rules of data are characterized by the NN.

We also validate the training effectiveness of NN with various scales under the combined sequence. The NN still has two hidden layers. We set a size coefficient of sz to quantify the scale. Each hidden layer contains $4sz$ nodes, while the input size is $2sz - 1$. Other configurations are the same as above. The data are still the combination of the mixed PRBS and Mersenne twister random data with an SNR of 4 dB. The results are shown in Fig. 5. The BER performances on different test data are still approximately equal, indicating no detection of the data rules. As the scale exponentially increases, the BER degrades slightly because of unpreventable overfitting induced by the complex structure. It is because the complex model overfits the data patterns instead of learning the data generation rules unless the test performance of different combinations should be significantly different.

The simulation results confirmed that the proposed scheme can effectively complicate and hide the data feature of the PRBS sequence. For an equalizer based on a simple NN model, the mixed PRBS can be a promising training data as the Mersenne

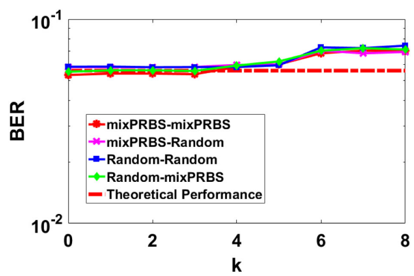

Fig. 5. BER performance of the NN on a mixed PRBS and Mersenne twister random data verses the NN scale, where $k = \log_2(sz)$.

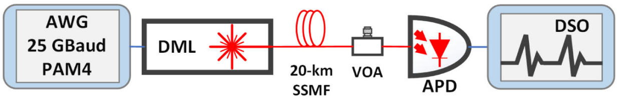

Fig. 6. Experimental setup of a 25 Gbaud PAM4 IMDD system based on 10 G-class optical devices.

twister random data. Once an advanced equalization algorithm can learn the rule of the mixed PRBS and Mersenne twister. The proposed scheme can also apply the Mersenne twister random data as the source sequence to construct a more complicated random sequence. The cross-test method can be used to validate if the generated data is learned by the advanced equalization algorithm.

Furthermore, we test the combination random data on a real IMDD system, as shown in Fig. 6. A Keysight M8195A arbitrary waveform generator (AWG) with a sampling rate of 64 GSa/s provides a 25 GBaud/s four-level pulse amplitude modulation (PAM4) random sequence generated by Matlab with the Mersenne twister algorithm. The PAM4 signal is modulated on a 10 G-class O-band directly modulated laser. After 20 km standard single-mode fiber (SSMF) transmission, a variable optical attenuator (VOA) is applied for measurement of the receiver sensitivity. The optical signal is then detected by a 20 G-class avalanche photodiode (APD). The detected signal is finally sampled by a LeCroy digital sample oscilloscope (DSO) with a bandwidth of 45 GHz and a sampling rate of 120 GSa/s. The mixed PRBS and Mersenne twister random data are modulated as the PAM4 signal and transmitted over the system, respectively.

In the experiment, the NN should equalize the PAM4 signal instead of the binary bit; hence, the output size of the NN is four, corresponding to the four levels of the PAM4 signal. Additionally, the sizes of the input layer and two hidden layers are 201, 128, and 128, respectively. The lengths of the training set and the test set are both 100,000. The results are shown in Fig. 7, where the launching power into the fiber is 8 dBm. The experimental results on real transmission data show that there are no differences in the BER performance among the data combinations, even with the bandwidth limitation and nonlinear effect. Hence, there is no detection of the generation rule in the training. This confirms that our proposed method effectively hides the data feature for NN training, which can be leveraged in research on the NN-based equalizer and other advanced algorithms.

In this work, we proposed a combination scheme of constructing random data to hide simple internal data features. The combined sequence will not expose features to mislead the NN-based equalizer during training. Although we cannot promise that the internal features of the combined random data cannot be learned by any algorithms, our proposed method can be used repeatedly time after time. After each iteration, the method will hide the data feature in a higher dimension. The capability of a certain algorithm is fixed and limited, and the exponentially increasing span of the dependent bits will thus exceed the algorithm capability with enough iterations. Therefore, once we detect internal data feature learning of the NN through the cross-test method, the combination scheme can be used to further hide the internal features. By repeating these operations, we finally obtain random data that will not be learned by the equalization algorithm. We hope the proposed

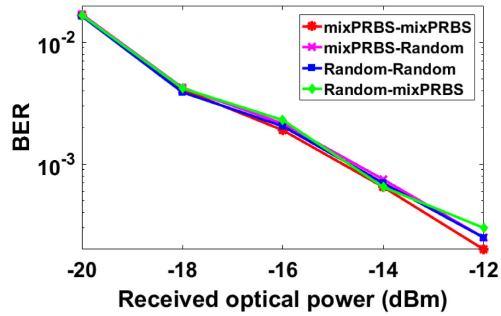

Fig. 7. BER performance of the NN on a mixed PRBS and Mersenne twister random data for a 25 Gbaud PAM4 IMDD system. The notation A–B means training on A and testing on B.

method will be suitable for the generation and validation of training data for an NN-based equalizer and other advanced equalizers with learning ability.

Funding. National Key Research and Development Program of China (2019YFB1803800).

Disclosures. The authors declare no conflicts of interest.

REFERENCES

1. M. A. Nielsen, *Neural Networks and Deep Learning* (Determination, 2015).
2. X. Wu, J. A. Jargon, R. A. Skoog, L. Paraschis, and A. E. Willner, J. Lightwave Technol. **27**, 1392 (2009).
3. D. Rafique, T. Szyrkowiec, H. Griesser, A. Autenrieth, and J.-P. Elbers, J. Lightwave Technol. **36**, 1443 (2018).
4. F. N. Khan, Y. Zhou, A. P. T. Lau, and C. Lu, Opt. Express **20**, 12422 (2012).
5. A. G. Reza and J. K. K. Rhee, IEEE Photon. Technol. Lett. **30**, 1416 (2018).
6. Z. Yang, F. Gao, S. Fu, X. Li, L. Deng, Z. He, M. Tang, and D. Liu, Opt. Lett. **43**, 3542 (2018).
7. C. Chuang, L. Liu, C. Wei, J. Liu, L. Henrickson, W. Huang, C. Wang, Y. Chen, and J. Chen, *Optical Fiber Communications Conference (OFC)* (2018), paper W2A.43.
8. C. Y. Chuang, C. C. Wei, T. C. Lin, K. L. Chi, L. C. Liu, J. W. Shi, Y. Chen, and J. Chen, *European Conference on Optical Communication (ECOC)* (2017), paper W.2.D.2.
9. M. Luo, F. Gao, X. Li, Z. He, and S. Fu, *Optical Fiber Communications Conference (OFC)* (2018), paper M2F.2.
10. Z. Wan, J. Li, L. Shu, M. Luo, X. Li, S. Fu, and K. Xu, Opt. Express **26**, 10631 (2018).
11. L. Yi, P. Li, T. Liao, and W. Hu, *European Conference on Optical Communication (ECOC)* (2018), paper Mo4B.5.
12. J. Estaran, R. Rios-Müller, M. A. Mestre, F. Jorge, H. Mardoyan, A. Konczykowska, J. Y. Dupuy, and S. Bigo, *European Conference on Optical Communication (ECOC)* (2016), paper M.2.B.2.
13. C. Hager and H. D. Pfister, *Optical Fiber Communications Conference (OFC)* (2018), paper W3A.4.
14. T. A. Eriksson, H. Bülow, and A. Leven, IEEE Photon. Technol. Lett. **29**, 2091 (2017).
15. L. Shu, J. Li, Z. Wan, W. Zhang, S. Fu, and K. Xu, *European Conference on Optical Communication (ECOC)* (2018), paper Tu4F.
16. L. Yi, T. Liao, L. Huang, and W. Hu, J. Lightwave Technol. **37**, 1621 (2018).
17. C. Chuang, L. Liu, C. Wei, J. Liu, L. Henrickson, C. Wang, Y. Chen, and J. Chen, *European Conference on Optical Communication (ECOC)* (2018), paper Tu4F.2.
18. V. Houtsma, E. Chou, and D. van Veen, *Optical Fiber Communications Conference (OFC)* (2019), paper M2B.6.