IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 19, 2021, accepted September 6, 2021, date of publication September 9, 2021,
date of current version September 16, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3111092*

# Length of Pseudorandom Binary Sequence Required to Train Artificial Neural Network Without Overfitting

**JONGWAN KIM AND HOON KIM, (Senior Member, IEEE)**
School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea

Corresponding author: Hoon Kim (hoonkim@kaist.ac.kr)

This work was supported by the Institute for Information and Communications Technology Promotion (IITP) Grant through the Government of Korea (MSIT) (Development of Tbps Optical Communication Technology) under Grant 2021-0-00809.

**ABSTRACT** The artificial neural network (ANN) has been applied to the various fields due to its capability to process complicated nonlinear functions involving a large amount of data. A pseudorandom binary sequence (PRBS) is commonly used to train the ANN since the PRBS is easily generated by using a linear feedback shift register and has a correlation function which is peaked at zero delay but is almost zero at other delays. However, when the PRBS length is not sufficiently long (compared to the input size of the ANN), the ANN trained by the PRBS could suffer from the overfitting where the ANN describes the behavior of the training sequence very well, but does poorly on new data inputs. In this paper, we provide a minimum length of the PRBS required by the ANN to avoid the overfitting for a given input size of the ANN. For this purpose, we analyze the minimum length of the input sequence required to estimate the PRBS pattern through theoretical study. These analyses are confirmed by numerical simulation. The findings of this paper would be used to select the PRBS length for training the ANN.

**INDEX TERMS** Pseudorandom binary sequence, artificial neural network, equalizer, overfitting.

## I. INTRODUCTION

Fueled by enormous computing power of digital processors, an artificial neural network (ANN) has gained a great deal of attention as a nonlinear computing system handling a large amount of data. The ANN is capable of processing complicated nonlinear functions even if it is difficult to describe them analytically [1]. Also, once the ANN is trained by a certain input sequence, it represents its own feature by itself, and thus can predict the output values based on the representation even when arbitrary inputs are fed to the ANN [2]. Thus, the ANNs are used widely in various fields, such as image classification [3], speech recognition [4], language translation [5], [6], face recognition [7], and fault diagnosis [8].

The ANNs are also applied to diverse areas of optical communications, including the identification of modulation formats [9]–[11], optical performance monitoring [12], [13], anomaly detection in optical networks [14], and the compensation of waveform distortions [15]–[18]. In particular, the ANN-based nonlinear equalizer (ANN-NLE) at the receiver has demonstrated its capability of correcting the waveform distortions induced by nonlinear interaction between numerous symbols [15]–[18]. For example, the ANN-NLE was shown to be effective in compensating for the waveform distortions arising from fiber nonlinearities in an 80-Gb/s coherent optical orthogonal frequency-division multiplexing system, outperforming the Volterra nonlinear equalizer (VNLE) by a couple of decibels in terms of Q-factor after 1000-km transmission [15], [16]. Performance improvement brought by the neural network was also reported for intensity-modulation/direct-detection systems [17], [18]. For example, neural network-based nonlinear equalizers exhibit the sensitivity improvement in comparison with the VNLE for 4 × 50-Gb/s and 56-Gb/s 4-ary pulse amplitude modulation signals when they are transported over a dispersion-compensated link [17] and the transceivers are band-limited [18], respectively.

For proper operation of the ANN-NLE, it is utmost important to train the equalizer using a set of sequences which mimics the real data to be equalized. A popular training sequence

The associate editor coordinating the review of this manuscript and approving it for publication was Chen Chen.


This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

is a pseudorandom binary sequence (PRBS). This sequence is easy to generate using a linear feedback shift register (LFSR) and an exclusive-OR (XOR) operation [19], and exhibits similar statistical features to a truly random sequence. However, one problem associated with the use of the PRBS as a training sequence for the ANN-NLE is the overfitting, where the ANN describes the behavior of the training sequence very well by estimating the entire PRBS from *a part* of the PRBS, but does poorly on new data inputs. This happens when the length of the PRBS (alternatively expressed as the order of the PRBS) is not sufficiently long since the XOR operation can be readily modeled by the simplest 3-layer ANN (composed of input, one hidden, and output layers) [20]. Several previous studies clearly showed the impact of the overfitting on the performance of optical transmission systems [21]–[24]. For example, an ANN-NLE trained by a specific length of PRBS exhibits low bit-error ratio (BER) performance when it is tested with the same length of the PRBS. However, there is a considerable performance degradation when PRBSs longer than the trained PRBS are fed to the ANN-NLE [21]–[24].

To avoid the overfitting, random data sequences could be used to train the neural networks [21], [22]. However, truly random sequences are not suitable for training the ANN for optical transmission systems. The receiver is away from the transmitter by a few tens to several thousand kilometers. Thus, it is highly desirable to utilize a *pre-determined, deterministic* sequence for training so that the ANN-NLE located at the receiver is able to recover the pre-determined sequence from the input sequence whose waveforms are distorted severely by the channel. For the synchronization of the training sequence, it is also desirable for the training sequence to have a unique autocorrelation characteristics.

A Mersenne Twister random sequence (MTRS) [23] generated by using the Mersenne Twister algorithm [25] satisfies these requirements of the training sequence. The most commonly used MTRS has a length of $2^{19937}$-1. However, this MTRS is too long to be used for training the ANN-NLE for optical communication systems. Not only does this MTRS incur a latency for achieving the sequence synchronization, but it also needs a large amount of shift registers and memory to generate and store the training sequence at the transmitter and receiver, respectively.

Recently, a pseudo-random binary generation method based on the combination of three different PRBSs has been proposed to train the ANN without the overfitting [24]. Here, one of the PRBSs is employed as a selection index from the other two PRBSs. However, the autocorrelation function of the pseudo-random sequence generated by this method is quite different from that of truly random data. Also, this sequence is very long since its length is determined by the least common multiple of the three PRBSs. Thus, it is expected that it takes quite long to achieve the sequence synchronization.

A PRBS generated by using an LFSR satisfies the requirements of the training sequence without posing the overfitting problem when the PRBS length is sufficiently long. Hence, in this paper, we provide a design guideline on the PRBS length required for training an ANN. We provide the minimum length of the PRBS to avoid the overfitting when the input size of the ANN is given. For this purpose, we analyze the pivotal tap indexes which govern the PRBS pattern having a length of $2^N$-1, where $N$ is the order of the PRBS. From those indexes, we derive an expression about the minimum input size required by the ANN to estimate the PRBS pattern. We confirm our theoretical analyses through computer simulation. For example, we find out the minimum input size required by an ANN to suffer from the overfitting for a given length of PRBS. The importance distribution of input data is also investigated to confirm the overwhelming importance of pivotal tap indexes when the overfitting occurs.

The rest of this paper is organized as follows. We analyze the PRBS generation implemented by using an LFSR and an XOR operation, and then provide a minimum length of an ANN input size required to estimate the PRBS pattern in Section II. Section III describes the simulation setup to verify our theoretical analysis. The comparison between the simulation results and the theoretical analyses together with the discussion about the simulation results is provided in Section IV. Finally, the conclusions of the paper are given in Section V.

## II. THEORETICAL ANALYSIS ON THE MINIMUM SEQUENCE LENGTH REQUIRED TO ESTIMATE THE PRBS PATTERN

In this section, we analyze the minimum length of sequence required to estimate the PRBS pattern. For this purpose, we express the PRBS generation rule as a couple different forms of connection representation.

### A. CONNECTION REPRESENTATION OF PRBS

A PRBS having a length of $2^N$ - 1 (hereinafter referred to as PRBS-$N$) can be generated by using and an LFSR and an XOR operation, as shown in Fig. 1. Commonly, a two-input or four-input XOR operation is used to provide a feedback loop. Fig. 1(a) shows the PRBS generator implemented by using the two-input XOR operation. In this LFSR-2 scheme,

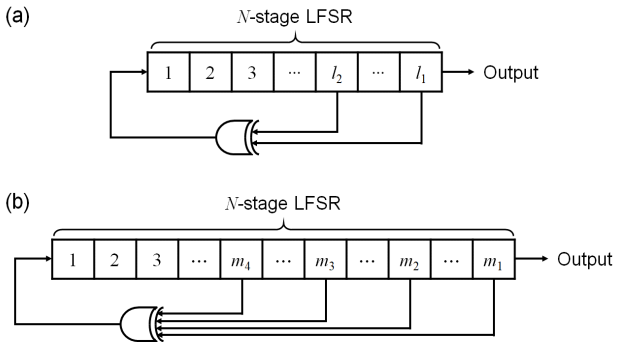


**FIGURE 1.** The PRBS-*N* generator implemented by using (a) LFSR-2 and (b) LFSR-4 schemes.

two binary bits are tapped and send to the first input bit of the LFSR after the XOR operation. Fig. 1(b) shows the PRBS generator where four bits are tapped for constructing the feedback loop. This scheme is referred to as LFSR-4. It should be noted that the tap positions are uniquely determined by the order of the PRBS, $N$, as tabulated in Table 1 [26]. Empty spaces in Table 1 imply that there is no solution for the corresponding PRBS-$N$.

**TABLE 1. The tap positions of the LFSR-2 and LFSR-4 for PRBS-$N$ [26].**

| $N$ | LFSR-2 | LFSR-4 | $N$ | LFSR-2 | LFSR-4 |
|---|---|---|---|---|---|
| 2 | 2, 1 | | 22 | 22, 21 | 22, 19, 18, 17 |
| 3 | 3, 2 | | 23 | 23, 18 | 23, 22, 20, 18 |
| 4 | 4, 3 | | 24 | | 24, 23, 21, 20 |
| 5 | 5, 3 | 5, 4, 3, 2 | 25 | 25, 22 | 25, 24, 23, 22 |
| 6 | 6, 5 | 6, 5, 3, 2 | 26 | | 26, 25, 24, 20 |
| 7 | 7, 6 | 7, 6, 5, 4 | 27 | | 27, 26, 25, 22 |
| 8 | | 8, 6, 5, 4 | 28 | 28, 25 | 28, 27, 24, 22 |
| 9 | 9, 5 | 9, 8, 6, 5 | 29 | 29, 27 | 29, 28, 27, 25 |
| 10 | 10, 7 | 10, 9, 7, 6 | 30 | | 30, 29, 26, 24 |
| 11 | 11, 9 | 11, 10, 9, 7 | 31 | 31, 28 | 31, 30, 29, 28 |
| 12 | | 12, 11, 8, 6 | 32 | | 32, 30, 26, 25 |
| 13 | | 13, 12, 10, 9 | 33 | 33, 20 | 33, 32, 29, 27 |
| 14 | | 14, 13, 11, 9 | 34 | | 34, 31, 30, 26 |
| 15 | 15, 14 | 15, 14, 13, 11 | 35 | 35, 33 | 35, 34, 28, 27 |
| 16 | | 16, 14, 13, 11 | 36 | 36, 25 | 36, 35, 29, 28 |
| 17 | 17, 14 | 17, 16, 15, 14 | 37 | | 37, 36, 33, 31 |
| 18 | 18, 11 | 18, 17, 16, 13 | 38 | | 38, 37, 33, 32 |
| 19 | | 19, 18, 17, 14 | 39 | 39, 35 | 39, 38, 35, 32 |
| 20 | 20, 17 | 20, 19, 16, 14 | 40 | | 40, 37, 36, 35 |
| 21 | 21, 19 | 21, 20, 19, 16 | | | |

The LFSR-2 and LFSR-4 schemes can be expressed by using the connection representation as

$$x(n) = x\,(n - l_1) \oplus x\,(n - l_2)\,, \tag{1}$$

and

$$x(n) = x\,(n - m_1) \oplus x\,(n - m_2) \oplus x\,(n - m_3) \oplus x\,(n - m_4)\,, \tag{2}$$

respectively.

Here, $x(n)$ refers to $n^{th}$ bit of the PRBS-$N$, $\oplus$ is the XOR operation, $l_1$ and $l_2$ are the tap positions of the LFSR-2 scheme, and $m_1$, $m_2$, $m_3$, and $m_4$ are those of the LFSR-4 scheme. The tap positions are all positive integers. We assume that $l_1$ is greater than $l_2$. We also assume that $m_1 > m_2 > m_3 > m_4$.

### B. MINIMUM LENGTH OF INPUT SEQUENCE REQUIRED BY ANN TO ESTIMATE PRBS-N GENERATED BY THE LFSR-2 SCHEME

Equation (1) shows the connection representation of the PRBS-$N$ implemented by using the LFSR-2 scheme. This equation shows that there are two pivotal tap indexes which determine the LFSR-2-based PRBS-$N$, $-l_1$ and $-l_2$. They are pivotal numbers to be used to construct the PRBS pattern. The negative sign implies that those tap indexes are preceding bits.

Due to the circular shift register architecture of the PRBS generator, equation (1) can also be expressed by using different tap indexes. Firstly, we can replace $n$ with $n + l_1$ in (1) and we have

$$x\,(n + l_1) = x(n) \oplus x\,(n + l_1 - l_2)\,. \tag{3}$$

By applying the self-inverse property (i.e., $\mathrm{A} \oplus \mathrm{A} = 0$) and the identity property (i.e., $\mathrm{A} \oplus 0 = \mathrm{A}$) of the XOR operation, we obtain

$$x(n) = x\,(n + l_1) \oplus x\,(n + l_1 - l_2)\,. \tag{4}$$

Similarly, $n$ can be replaced with $n+l_2$ in (1). Then, we also rewrite (1) as

$$x(n) = x\,(n + l_2) \oplus x\,(n + l_2 - l_1)\,. \tag{5}$$

We have three equations for the PRBS-$N$ generated by using the LFSR-2 scheme: (1), (4), and (5). We identify the pivotal tap indexes for each equation. They are plotted in Fig. 2. In the LFSR-2 scheme, we can estimate $x(n)$ when two input bits at the pivotal tap indexes are given. This implies that the ANN would suffer from the overfitting when the input sequence includes the two bits at the pivotal tap indexes. It should be noted that the three equations have different pivotal tap indexes, but they generate the same PRBS-$N$ pattern. Thus, if the input sequence includes the two pivotal taps for one of the three equations, the ANN is able to estimate the PRBS pattern.

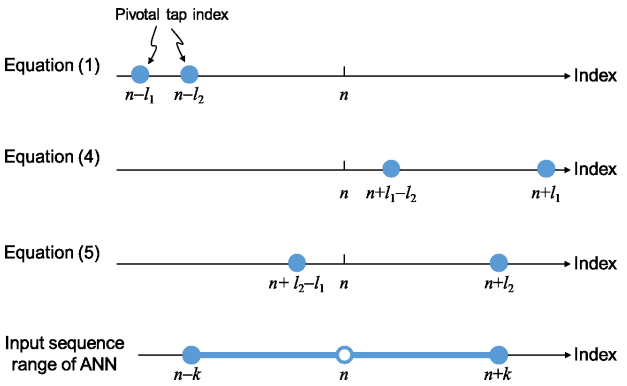


**FIGURE 2. The pivotal tap indexes for equations, (1), (4), and (5). Also, shown at the bottom is the range of the input sequence of the ANN. The hollow circle implies that it is not included in the input sequence.**

In optical communications where the fiber chromatic dispersion makes an optical pulse spread symmetrically at around its center, the ANN-NLE is typically fed by a symmetric input sequence [27]–[29]. This means that the input layer of the ANN receives a sequence ranging from $x(n - k)$ to $x(n + k)$, where $k$ is the size of the input layer. Here, $x(n)$ should be excluded from the input sequence since it is the bit to be estimated by the equalizer. In this case, the minimum length of the input sequence required by the ANN to

estimate the PRBS pattern would be determined by (5). This is because the maximum absolute values between the two pivotal tap indexes for (5) are smaller than those of the other two equations. This is illustrated in Fig. 2. For the two pivotal tap indexes of (5) to be included in the range of the input sequence of the ANN, the input size of the ANN should be

$$k \geq L_{LFSR2} = \max\left(l_2, l_1 - l_2\right). \tag{6}$$

Here, the minimum input size required by the ANN to estimate the PRBS pattern is denoted by $L_{LFSR2}$. For the tap positions of the LFSR-2 scheme listed in Table 1, $l_2$ is always greater than $l_1 - l_2$. Thus, $L_{LFSR2}$ is equal to $l_2$.

It should be noted that if the ANN is designed to receive either the preceding [i.e., $x(n-k), \ldots, x(n-1)$] or following bits [i.e., $x(n+1), \ldots, x(n+k)$] with respect to the current time index $n$, the minimum length of the input sequence to estimate the PRBS is determined by (1) or (4). In this case, therefore, the minimum input size of the ANN to suffer from the overfitting should be $l_1$.

### C. MINIMUM LENGTH OF INPUT SEQUENCE REQUIRED BY ANN TO ESTIMATE PRBS-N GENERATED BY THE LFSR-4 SCHEME

In this subsection, we derive an analytical expression about the minimum input size required by the ANN to estimate the PRBS generated by the LFSR-4 scheme, $L_{LFSR4}$. The connection representation of the PRBS-$N$ implemented by using the LFSR-4 scheme is expressed as (2). Similar to the previous subsection, we can rewrite (2) in terms of different pivotal tap indexes by the properties of the XOR operation.

By substituting $n$ with $n + m_1$, we have

$$\begin{aligned} x\left(n+m_1\right) = x(n) &\oplus x\left(n+m_1-m_2\right) \\ &\oplus x\left(n+m_1-m_3\right) \oplus x\left(n+m_1-m_4\right). \end{aligned} \tag{7}$$

Then, this equation can be rewritten by using the self-inverse property and the identity property of the XOR operation as

$$\begin{aligned} x(n) = x\left(n+m_1\right) &\oplus x\left(n+m_1-m_2\right) \\ &\oplus x\left(n+m_1-m_3\right) \oplus x\left(n+m_1-m_4\right). \end{aligned} \tag{8}$$

Additional three equations are also obtained by substituting $n$ with $n + m_2$, $n + m_3$, or $n + m_4$. They are

$$\begin{aligned} x(n) = x\left(n+m_2\right) &\oplus x\left(n+m_2-m_1\right) \\ &\oplus x\left(n+m_2-m_3\right) \oplus x\left(n+m_2-m_4\right), \end{aligned} \tag{9}$$

$$\begin{aligned} x(n) = x\left(n+m_3\right) &\oplus x\left(n+m_3-m_1\right) \\ &\oplus x\left(n+m_3-m_2\right) \oplus x\left(n+m_3-m_4\right), \end{aligned} \tag{10}$$

and

$$\begin{aligned} x(n) = x\left(n+m_4\right) &\oplus x\left(n+m_4-m_1\right) \\ &\oplus x\left(n+m_4-m_2\right) \oplus x\left(n+m_4-m_3\right). \end{aligned} \tag{11}$$

The LFSR-4-based PRBS pattern can be expressed by five different forms of the connection representation: (2), (8), (9), (10) and (11). Fig. 3 shows the four pivotal tap indexes for each equation. Similar to the discussion in the previous subsection, the ANN will be able to estimate the PRBS pattern generated by the LFSR-4 scheme when the four bits at the pivotal tap indexes are all included in the input sequence.

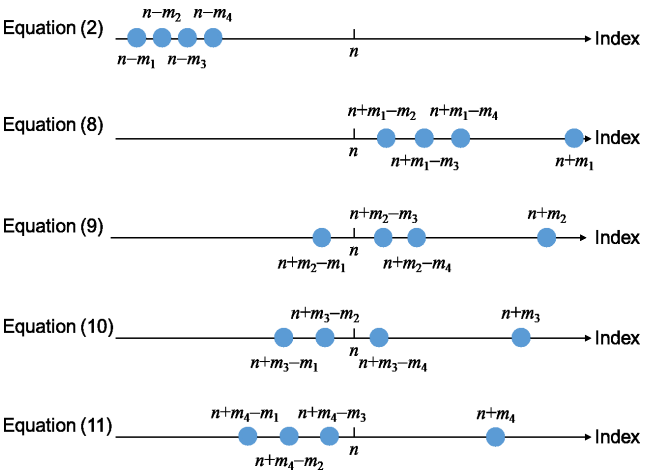


**FIGURE 3.** The pivotal tap indexes for equations (2), (8), (9), (10), and (11).

For the ANN-NLE having a symmetrical input sequence [i.e., from $x(n-k)$ to $x(n+k)$, excluding $x(n)$], the minimum length of the input sequence required by the ANN to estimate the PRBS pattern would be determined by (9), (10), and (11).

For the four pivotal tap indexes of these three equations to be included in the range of the input sequence of the ANN, the minimum input size of the ANN should be

$$k \geq L_{LFSR4} = \min \left\{ \begin{array}{l} \max\left(m_2, m_1 - m_2\right) \\ \max\left(m_3, m_1 - m_3\right) \\ \max\left(m_4, m_1 - m_4\right) \end{array} \right\}. \tag{12}$$

For the tap positions of the LFSR-4 scheme listed in Table 1, $m_2$, $m_3$, and $m_4$ are always greater than $m_1 - m_2$, $m_1 - m_3$, and $m_1 - m_4$, respectively, except for PRBS-5 and PRBS-6. Thus, $L_{LFSR4}$ becomes $m_4$ when $N$ is larger than or equal to 7. For PRBS-5 and PRBS-6, $L_{LFSR4}$ become $m_3$.

If the ANN receives an asymmetrical input sequence [i.e., either the preceding bit sequence from $x(n-k)$ to $x(n-1)$ or the following bit sequence from $x(n+1)$ to $x(n+k)$], the minimum length of the input sequence required by the ANN to estimate the PRBS pattern would be determined by (2) or (8). Therefore, the minimum input size of the ANN to experience the overfitting should be $m_1$ in this case.

## III. SIMULATION SETUP

We carry out simulation study to verify our finding presented in the previous section. Fig. 4 shows the simulation setup. We first generate a PRBS-$N$ pattern by using a shift register and XOR operation (as shown in Fig. 1) and send it to an ANN. The ANN is composed of three layers: input layer, hidden layer, and output layer. The input layer receives a symmetrical sequence composed of $k$ preceding and $k$ following bits with respect to the time index $n$. The current bit $x(n)$ is not included in the input vector of the ANN because it is an output

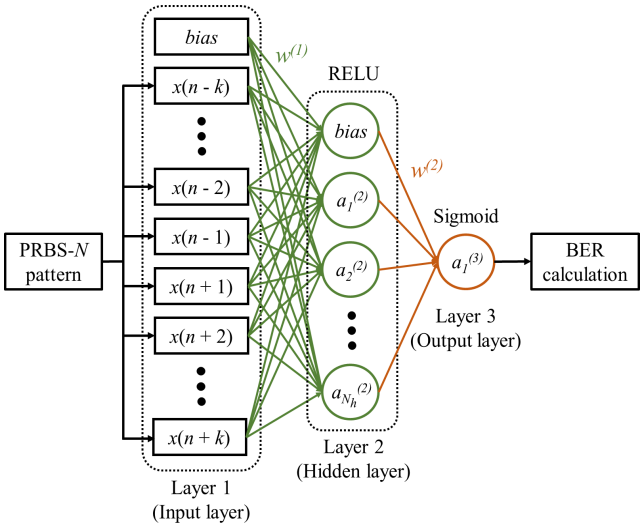


**FIGURE 4.** Simulation setup.

to be estimated by the ANN. The input vector including a bias and $2k$ bits is regarded as a set of the input data. Then, the input data are used for the calculation of outputs of the hidden neurons. We use a rectified linear unit (ReLU) function as an activation function of the hidden layer for fast convergence. The number of the hidden neurons, $N_h$, is set to be 10. The output of the $j^{th}$ hidden neuron, $a_j^{(2)}$, is calculated by the ReLU function from a weighted sum of the input data with a connection weight $w^{(1)}$ between the input layer and the hidden layer. The outputs of the hidden neurons are fed to the output layer with a bias and then the estimated result of the ANN is obtained at the output layer. In the case of the PRBS-$N$, its value is either '0' or '1'. Thus, the number of output neurons is set to be 1 and we use a sigmoid function as an activation function of the output layer. The output is estimated by the sigmoid function from a weighted sum of the bias and the outputs of the hidden neurons with a connection weight $w^{(2)}$ between the hidden layer and the output layer.

In the training process, the estimated output is compared with the desired PRBS-$N$ pattern to obtain an error between those two outputs. In this simulation, we use the cross entropy loss function [30] due to the binary nature of the output. The connection weights, $w^{(1)}$ and $w^{(2)}$, are updated from the error using the backpropagation algorithm [31] based on the gradient descent method [32]. We train the ANN repeatedly by the training data sets with a batch size of 200 and 1,000 epochs.

After training, we estimate the BER by using the direct error counting. In our simulation, 10,000 input training data sets and 1,000,000 input test data sets are used to train the ANN and estimate the BER, respectively, for each PRBS-$N$. The PRBSs are repeated when the length of data sets are longer than the PRBS length. For example, PRBS-$N$ is repeated to compose 10,000 training data sets when $N$ is smaller than 14 since $2^{13}-1 < 10{,}000$. The test data sets are also composed of repeated PRBSs when the order of PRBS is less than 20. Thus, our data sets consist of both unrepeated and repeated PRBSs, depending upon the PRBS length. It is also worth mentioning that we employ the first 10,000 bits of PRBS for training and the next 1,000,000 bits for estimating the BER. Thus, there is no overlap between the training and test data sets when $N$ is larger than or equal to 20.

## IV. SIMULATION RESULTS

In this section, we discuss the simulation results and compare them with the theoretical analyses formulated by (6) and (12) for the LFSR-2 and LFSR-4 schemes, respectively.

### A. MINIMUM INPUT SIZES OF ANN TO HAVE THE OVERFITTING FOR LFSR-2-BASED PRBS

We first obtain the BER performance for all the LFSR-2-based PRBSs listed in Table 1 as we vary the input size of the ANN, $k$. We carry out simulation for all the PRBSs listed in Table 1 to investigate the minimum input sizes of the ANN having the overfitting. Fig. 5 shows some of these results. The results exhibit a binary pattern similar to the comparator output. The BER value is either ∼0.5 or ∼0. For example, when $k$ is smaller than 6 for PRBS-7, we obtain BERs close to 0.5. The BER values drop close to 0 when $k$ is larger than or equal to 6. This is because when the ANN is capable of estimating the entire PRBS from a part of the PRBS (i.e., overfitting), the BER is measured to be zero. On the other hand, the BER should be close to 0.5 when the overfitting problem does not occur. The results show that when the input size of the ANN is sufficiently small, the ANN cannot estimate the entire PRBS from a part of the PRBS. Thus, we do not suffer from the overfitting in this case. Fig. 5 clearly shows that the largest input sizes to avoid the overfitting for PRBS-7, PRBS-15, PRBS-25, and PRBS-35 are 5, 13, 21, and 32, respectively. The overfitting occurs when the input sizes of the ANN are larger than these values.

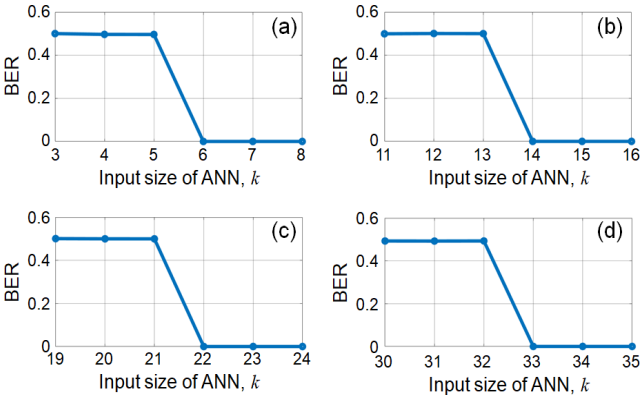


**FIGURE 5.** The measured BER values as a function of the input size of the ANN, $k$, when the order of the PRBS is (a) 7, (b) 15, (c) 25, and (d) 35. The PRBSs are generated by the LFSR-2 scheme.

Fig. 6 shows the minimum input size of the ANN to have the overfitting as a function of the order of the PRBS, $N$. For comparison, we also plot our theoretical results of (6) in this figure. The simulation results match perfectly with the

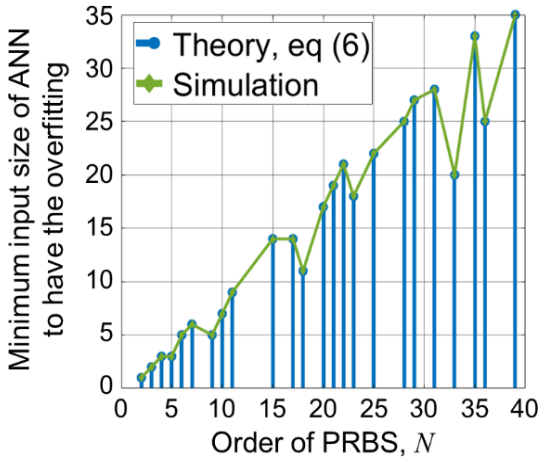


**FIGURE 6.** The minimum input size of the ANN to have the overfitting versus the order of the PRBS when the PRBS is generated by using the LFSR-2 scheme.

theoretical values. This result shows that the input size of the ANN should be smaller than $L_{LFSR2}$ to avoid the overfitting.

The occurrence of the overfitting can also be confirmed by observing the importance distribution of the input data. For this purpose, we define $W_i$ as a sum of the absolute weights between each input neuron and the hidden neurons, which is expressed as

$$W_i = \sum_{j=1}^{N_h} \left| w_{ij}^{(1)} \right|, \tag{13}$$

where $i$ is the tap index ranging from $-k$ to $k$, but $i \neq 0$. Also, $j$ is the $j^{th}$ neuron in the hidden layer and $w_{ij}^{(1)}$ is the connection weight between the input layer and the hidden layer. The importance of input data can be quantified by $W_i$. Fig. 7 shows the distribution of normalized $W_i$ as a function of time index $i$ for some PRBSs. The input size of the ANN is set to be $L_{LFSR2}$, the minimum input size of the ANN to have the overfitting as predicted in Section II. The results show that specific input bits have significantly larger $W_i$ than other input bits. For example, we have normalized $W_i$ values close to 1 for PRBS-25 when $i$ is $-3$ and 22 in Fig. 7(c). These two $i$ values correspond to the pivotal tap indexes introduced in Section II. This implies that the ANN retrieves the PRBS generation rule during the training process, and as a result, we have the overfitting. We confirm that the pivotal tap indexes have an overwhelming importance for all the PRBSs listed in Table 1 when the overfitting occurs.

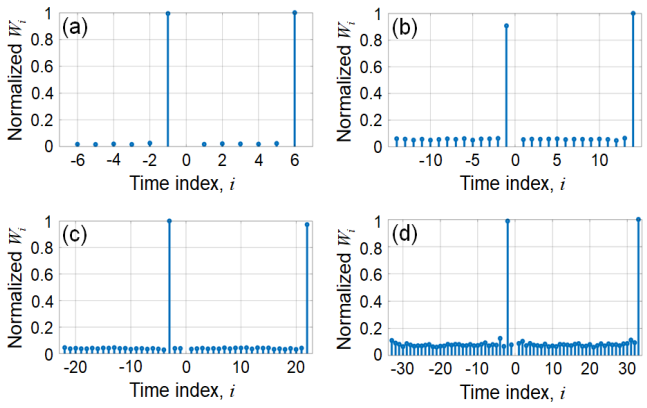


**FIGURE 7.** The importance distributions of the input data for (a) PRBS-7, (b) PRBS-15, (c) PRBS-25, and (d) PRBS-35. The PRBSs are generated by the LFSR-2 scheme.

### B. MINIMUM INPUT SIZES OF ANN TO HAVE THE OVERFITTING FOR LFSR-4-BASED PRBS

Next, we analyze the BER performance of the PRBS-$N$ implemented by using the LFSR-4 scheme. Fig. 8 shows the BER values for PRBS-8, PRBS-19, PRBS-27, and PRBS-37 as a function of the input size of the ANN, $k$. In the same manner as we observed in the previous subsection, the BER values are converged to zero when the input size is greater than a specific value, indicating the overfitting. The results show that we have a BER close to zero when the input sizes of the ANN are larger than 3, 13, 21, and 30 for PRBS-8, PRBS-19, PRBS-27, and PRBS-37, respectively. Obviously, the overfitting problem arises in these cases.

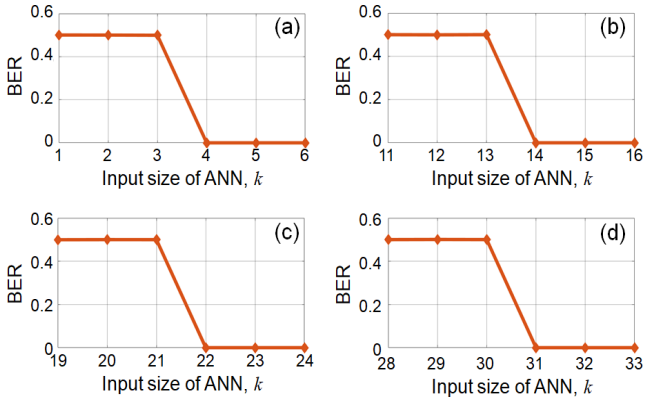


**FIGURE 8.** The measured BER values as a function of the input size of the ANN, $k$, when the order of the PRBS is (a) 8, (b) 19, (c) 27, and (d) 37. The PRBSs are generated by the LFSR-4 scheme.

Fig. 9 shows the minimum input size of the ANN to have the overfitting when the ANN is trained by using the PRBS-N implemented by using the LFSR-4 scheme. The result of our theoretical analysis formulated by (12) is also shown in this figure. The results show that our analysis agrees perfectly with the simulation results. It confirms that the input size of the ANN should be smaller than $L_{LFSR4}$ for the LFSR-4-based PRBS-N to avoid the overfitting. In other words, the length of PRBS required to train the ANN without the overfitting is $L_{LFSR4} - 1$.

Finally, we plot the importance distribution of the input data as a function of the time index for some PRBSs implemented by using the LFSR-4 scheme. The results are shown in Fig. 10. The input size of the ANN is set to be $L_{LFSR4}$. In this case, the ANN is able to estimate the entire PRBS pattern from a part of the PRBS since the input size of the

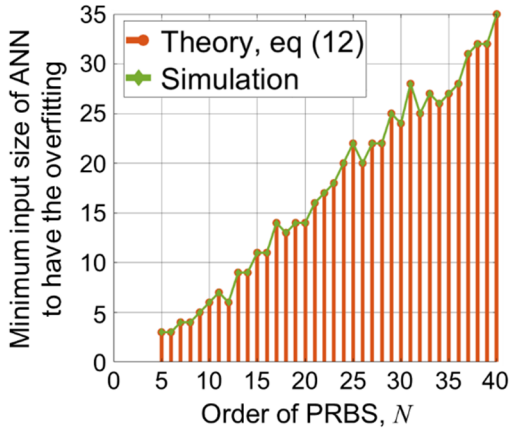


**FIGURE 9.** The minimum input size of the ANN to have the overfitting versus the order of the PRBS when the PRBS is generated by using the LFSR-4 scheme.

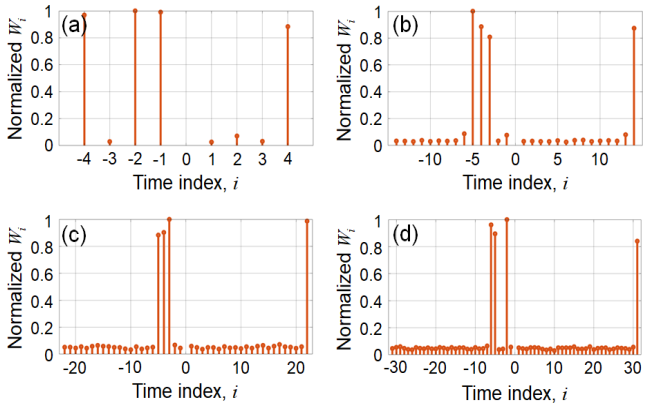


**FIGURE 10.** The importance distributions of the input data for (a) PRBS-8, (b) PRBS-19, (c) PRBS-27, and (d) PRBS-37. The PRBSs are generated by the LFSR-4 scheme.

ANN is equal to $L_{LFSR4}$, as predicted in Section II. Clearly, the results show that the normalized $W_i$ for the pivotal tap indexes have overwhelming importance in the distribution. That is why the ANN is able to estimate the PRBS pattern, and thus the BER values converge to zero for the PRBS-$N$.

## V. CONCLUSION

We have provided a design guideline on the PRBS length required for training the ANN without posing the overfitting problem. For a given input size of the ANN, the length of the PRBS generated by using the LFSR should be larger than (6) and (12) when the feedback is composed of the 2- and 4-input XOR operations, respectively. Our theoretical analyses are confirmed through computer simulation. Both the BER performance and the importance distribution clearly show that the overfitting occurs when the input size of the ANN is larger than the proposed design guideline on the PRBS length. Since the training and test data sets consist of both unrepeated and repeated PRBSs, depending upon the length of PRBS, our design guideline is applicable regardless of whether the PRBS is repeated or not. We believe that the findings of this work could be used to select the PRBS length for training the ANN.

## REFERENCES

[1] S. Ferrari and R. F. Stengel, "Smooth function approximation using neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 24–38, Jan. 2005.
[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105.
[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Oct. 2012.
[5] S. Sagiroglu, U. Yavanoglu, and E. N. Guven, "Web based machine learning for language identification and translation," in *Proc. 6th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Cincinnati, OH, USA, Dec. 2007, pp. 280–285.
[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 3104–3112.
[7] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1701–1708.
[8] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.
[9] F. N. Khan, Y. Zhou, A. P. T. Lau, and C. Lu, "Modulation format identification in heterogeneous fiber-optic networks using artificial neural networks," *Opt. Exp.*, vol. 20, no. 11, pp. 12422–12431, May 2012.
[10] F. N. Khan, K. Zhong, X. Zhou, W. H. Al-Arashi, C. Yu, C. Lu, and A. P. T. Lau, "Joint monitoring and modulation format identification in digital coherent receivers using deep neural networks," *Opt. Exp.*, vol. 25, no. 16, pp. 17767–17776, Jul. 2017.
[11] D. Wang, M. Zhang, Z. Li, J. Li, M. Fu, Y. Cui, and X. Chen, "Modulation format recognition and OSNR estimation using CNN-based deep learning," *IEEE Photon. Technol. Lett.*, vol. 29, no. 19, pp. 1667–1670, Oct. 1, 2017.
[12] Z. Dong, F. N. Khan, Q. Sui, K. Zhong, C. Lu, and A. P. T. Lau, "Optical performance monitoring: A review of current and future technologies," *J. Lightw. Technol.*, vol. 34, no. 2, pp. 525–543, Jan. 15, 2016.
[13] T. Tanimura, T. Hoshida, T. Kato, S. Watanabe, and H. Morikawa, "Convolutional neural network-based optical performance monitoring for optical transport networks," *J. Opt. Commun. Netw.*, vol. 11, no. 1, pp. A52–A59, Jan. 2019.
[14] X. Chen, B. Li, R. Proietti, Z. Zhu, and S. J. B. Yoo, "Self-taught anomaly detection with hybrid unsupervised/supervised machine learning in optical networks," *J. Lightw. Technol.*, vol. 37, no. 7, pp. 1742–1749, Apr. 1, 2019.
[15] M. A. Jarajreh, E. Giacoumidis, I. Aldaya, S. T. Le, A. Tsokanos, Z. Ghassemlo, and N. J. Doran, "Artificial neural network nonlinear equalizer for coherent optical OFDM," *IEEE Photon. Technol. Lett.*, vol. 27, no. 4, pp. 387–390, Feb. 15, 2015.
[16] E. Giacoumidis, S. T. Le, M. Ghanbarisabagh, M. McCarthy, I. Aldaya, S. Mhatli, M. A. Jarajreh, P. A. Haigh, N. J. Doran, A. D. Ellis, and B. J. Eggleton, "Fiber nonlinearity-induced penalty reduction in CO-OFDM by ANN-based nonlinear equalization," *Opt. Lett.*, vol. 40, no. 21, pp. 5113–5116, Nov. 2015.
[17] M. Luo, F. Gao, X. Li, Z. He, and S. Fu, "Transmission of 4×50-gb/s PAM-4 signal over 80-km single mode fiber using neural network," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, 2018, p. M2F-2.
[18] P. Li, L. Yi, L. Xue, and W. Hu, "56 Gbps IM/DD PON based on 10G-class optical devices with 29 dB loss budget enabled by machine learning," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, 2018, p. M2B.2.
[19] *Digital Test Patterns for Performance Measurements on Digital Transmission Equipment*, document O.150, ITU-T, 1992.
[20] Z. Yanling, D. Bimin, and W. Zhanrong, "Analysis and study of perceptron to solve XOR problem," in *Proc. Int. Workshop Auto. Decentralized Syst.*, Beijing, China, 2002, pp. 168–173.

[21] T. A. Eriksson, H. Bülow, and A. Leven, "Applying neural networks in optical communication systems: Possible pitfalls," *IEEE Photon. Technol. Lett.*, vol. 29, no. 23, pp. 2091–2094, Dec. 1, 2017.
[22] L. Shu, J. Li, Z. Wan, W. Zhang, S. Fu, and K. Xu, "Overestimation trap of artificial neural network: Learning the rule of PRBS," in *Proc. Eur. Opt. Commun.*, Roma, Italy, 2018, pp. 1–3, Paper Tu4f.1.
[23] C.-Y. Chuang, L.-C. Liu, C.-C. Wei, J.-J. Liu, L. Henrickson, C.-L. Wang, Y.-K. Chen, and J. Chen, "Study of training patterns for employing deep neural networks in optical communications systems," in *Proc. Eur. Conf. Opt. Commun.*, Roma, Italy, 2018, pp. 1–3, Paper Tu4f.2.
[24] T. Liao, L. Xue, L. Huang, W. Hu, and L. Yi, "Training data generation and validation for a neural network-based equalizer," *Opt. Lett.*, vol. 45, no. 18, pp. 5113–5116, Sep. 2020.
[25] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, pp. 3–30, Jan. 1998.
[26] R. Ward and T. Molteno, "Table of linear feedback shift registers," Dept. Phys., Univ. Otago, Dunedin, New Zealand, Electron. Tech. Rep. 2012-1, 2012.
[27] A. Argyris, J. Bueno, and I. Fischer, "Photonic machine learning implementation for signal recovery in optical communications," *Sci. Rep.*, vol. 8, no. 1, Dec. 2018, Art. no. 8487.
[28] L. Yi, T. Liao, and W. Hu, "Neural network-based equalization in high-speed PONs," in *Proc. Opt. Fiber Commun. Conf.*, San Diego, CA, USA, 2020, pp. 1–3, Paper T4D.3.
[29] W.-H. Huang, H.-M. Nguyen, C.-W. Wang, M.-C. Chan, C.-C. Wei, J. Chen, H. Taga, and T. Tsuritani, "Nonlinear equalization based on artificial neural network in DML-based OFDM transmission systems," *J. Lightw. Technol.*, vol. 39, no. 1, pp. 73–82, Jan. 1, 2021.
[30] K. P. Murphy, "Probability," in *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012, pp. 27–64.
[31] S. Haykin, "Multilayer perceptrons," in *Neural Networking Learning Machine*. Upper Saddle River, NJ, USA: Pearson, 2009, pp. 122–229.
[32] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*. [Online]. Available: http://arxiv.org/abs/1609.04747

**JONGWAN KIM** received the B.S. degree in electronics and electrical engineering from Sungkyunkwan University, Suwon, South Korea, in 2015, and the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include artificial intelligence, neural networks, optical communications, and nonlinear equalizer.

**HOON KIM** (Senior Member, IEEE) is currently an Associate Professor with the School of Electrical Engineering, KAIST. Prior to joining KAIST, in 2014, he was with Bell Laboratories, Lucent Technologies, from 2000 to 2001, Samsung Electronics, South Korea, from 2001 to 2007, and the National University of Singapore, from 2007 to 2014. His research interests include high-capacity fiber-optic communication systems, free-space optical communications, broadband optical access networks, and photonic systems. He serves as a Senior Editor for IEEE PHOTONICS TECHNOLOGY LETTERS and an Editor for *Optics Communications*.