This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

Performance versus Complexity Study of Neural Network Equalizers in Coherent Optical Systems

Pedro J. Freire, Yevhenii Osadchuk, Bernhard Spinnler, Antonio Napoli, Wolfgang Schairer, Nelson Costa, Jaroslaw E. Prilepsky, Sergei K. Turitsyn

***Abstract*—We present the results of the comparative performance-versus-complexity analysis for the several types of artificial neural networks (NNs) used for nonlinear channel equalization in coherent optical communication systems. The comparison is carried out using an experimental set-up with the transmission dominated by the Kerr nonlinearity and component imperfections. For the first time, we investigate the application to the channel equalization of the convolution layer (CNN) in combination with a bidirectional long short-term memory (biLSTM) layer and the design combining CNN with a multi-layer perceptron. Their performance is compared with the one delivered by the previously proposed NN-based equalizers: one biLSTM layer, three-dense-layer perceptron, and the echo state network. Importantly, all architectures have been initially optimized by a Bayesian optimizer. First, we present the general expressions for the computational complexity associated with each NN type; these are given in terms of real multiplications per symbol. We demonstrate that in the experimental system considered, the convolutional layer coupled with the biLSTM (CNN+biLSTM) provides the largest Q-factor improvement compared to the reference linear chromatic dispersion compensation (2.9 dB improvement). Then, we examine the trade-off between the computational complexity and performance of all equalizers and demonstrate that the CNN+biLSTM is the best option when the computational complexity is not constrained, while when we restrict the complexity to some lower levels, the three-layer perceptron provides the best performance.**

***Index Terms*—Neural network, nonlinear equalizer, computational complexity, Bayesian optimizer, coherent detection.**

I. Introduction

AMONGST the variety of different nonlinearity compensation methods, the machine learning (ML) based techniques are gaining momentum as a promising and flexible tool capable to efficiently unroll fiber and component-induced impairments. In the past several years, the research on artificial neural networks (NN) for optical channel equalization has already led to the development of a noticeable number of novel digital signal processing (DSP) methods that can provide the performance better than that rendered by the "conventional" DSP approaches [1]–[10]. The fast development of NN-related research and the growing ML developers community incites testing different novel NN architectures to mitigate fiber propagation impairments. In terms of the experimental verification of NN-based equalizers, several works dealt with the intensity-modulation with direct-detection (IM/DD) links. It was demonstrated that the application of the NNs with different internal structures, such as multi-layer perceptron (MLP) [11], [12] (i.e. a simple densely connected feed-forward NN architecture), convolutional NNs (CNN) [13], [14], echo state networks (ESN) [15], and long short-term memory (LSTM) NNs [16], is efficient in improving optical system-level performance. However, the test of similar NN architectures in coherent optical systems has been carried out, mainly, numerically [17]–[20], or in short-haul experiments [21]–[24]. It is worth noticing that some very recent works evaluated the functioning of NN-based equalizers in metro/long-haul trials [4], [5], [8]–[10].

The variety of existing and emerging channel equalizers makes a comparative analysis of the different solutions a timely challenge. The NN-based channel equalization refers to two important aspects: i) the improvement of performance by the reduction of bit-error rate (BER), and ii) the complexity of the algorithms, which is a fundamental issue for practical implementation. Clearly, the comparison can be carried out only for specific systems: some approaches can be more suitable for certain transmission links, while the others are favorable for different systems.

To gain a thorough understanding of how each of the aforementioned NN architectures performs, we need to pick a benchmark system for the comparison. In this work, we perform such a comparison using, as a benchmark, a single channel transmission of a dual-polarization (DP) 16-QAM signal with 34.4 GBd rate transmitted over 9×50km TrueWave Classic (TWC) fiber spans at the power of 2 dBm. Such a choice of the fiber and the power level ensures that the system is in the strongly nonlinear regime, as we intend to study how the NNs unroll the Kerr nonlinearity effects. In our work, we analyze both the synthetically simulated and the experimental data. We, first, analyze the performance of several previously studied NN models: MLP, bidirectional LSTM (biLSTM), and ESN. Next, we compare their performance with that rendered by new composite NN structures: i) the convolutional layer coupled with the MLP (CNN+MLP); ii) the combination of the convolutional layer with the biLSTM (CNN+biLSTM). These

This paper was supported by the EU Horizon 2020 program under the Marie Sklodowska-Curie grant agreement 813144 (REAL-NET). YO acknowledges the support of the SMARTNET EMJMD programme (Project number - 586686-EPP-1-2017-1-UK-EPPKA1-JMD-MOB). JEP is supported by Leverhulme Trust, Grant No. RP-2018-063. SKT acknowledges support of the EPSRC project TRANSNET

Pedro J. Freire, Yevhenii Osadchuk, Jaroslaw E. Prilepsky and Sergei K. Turitsyn are with Aston Institute of Photonic Technologies, Aston University, United Kingdom, p.freiredecarvalhosouza@aston.ac.uk.

Antonio Napoli is with Infinera London, UK, anapoli@infinera.com.

Wolfgang Schairer and Bernhard Spinnler are with Infinera R&D, Sankt-Martin-Str. 76, 81541, Munich, Germany.

Nelson Costa is with Infinera Unipessoal, Lda, Rua da Garagem nº1, 2790-078 Carnaxide, Portugal, ncosta@infinera.com.

Manuscript received xxx 19, zzz; revised January 11, yyy.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

new designs are, then, tested in the same environment, allowing us to infer the performance characteristics pertinent to each type among the 7 different NN topologies. We point out that the term "topology" in our research identifies *the particular NN structure (architecture) with a specific fixed distribution of hyper-parameters*. We emphasize that, in contrast to other similar investigations, we employ *the Bayesian optimization procedure* [9] for each NN type studied. This provides the optimal distribution of hyper-parameters pertaining to each NN type, such that we identify the best functioning regime (in terms of the performance delivered) for each architecture without complexity constraints. We show that the new CNN+biLSTM combination performs better than all other studied types. For each NN type considered hereafter, we also present the analytical expressions for the complexity, i.e. the number of multiplications attributed to each specific NN per recovered symbol. The highest complexity for the optimized NN equalizers corresponds to the new CNN+biLSTM composition that also renders the best performance.

The completely new subject in the remit of this manuscript is what happens when we restrain the complexity of different NN types: no work has previously addressed the comparison of the performance rendered by different NNs considering *the identical levels of computational complexity*. Our findings demonstrate a nontrivial behavior: while at the relatively high complexity levels the best performing model is the CNN+biLSTM, when we constraint the complexity to lower values, the simple MLP equalizer outperforms the advanced NN structures with the same complexity. Nevertheless, we notice that the goal of this paper is not to reach a broad conclusion about the trade-off between the complexity and performance for all possible transmission scenarios; rather, we aim at emphasizing the importance of accounting for this issue in the equalizer design stage, and we provide the tools for one's correctly assessing the DSP-type complexity of the most popular NN-layers.

The paper is organized as follows. In Sec. II we describe the details of the different NN equalizers analyzed in our study. Sec. III presents how to compute the computational complexity on all NN-based equalizers considered in this paper. Sec. IV describes the experimental setup and contains the results, including the comparison between the performance and computational complexity of different NN topologies; the performance is also compared with the digital back-propagation with 3 steps per span. Our findings are summarized in the conclusion.

II. A ZOO OF NEURAL NETWORK-BASED EQUALIZERS

In this section, we revisit the most popular NN architectures that have been proposed and investigated so far in coherent optical channel post-equalization. We also introduce two new composite NN equalizer structures that can be deemed as the extension of previously proposed NN configurations.

To enhance the reproducibility of our methods, we provide a thorough summary of each NN architecture. The code of the algorithms implemented in Python 3.6.9 with TensorFlow (2.2.0) GPU backend and Keras (2.3.1), is provided in Zenodo [25].

Before addressing the details of the NN-based equalizers, let us describe how the datasets used in this work are created. When dealing with the optical channel equalization, we require the NN to process not only the symbol of interest but also the neighboring ones insofar as both the chromatic dispersion and the drive amplifier add the memory to the channel. The latter means that the NN performs better if it is given information about the correlations between the symbols in the sequence. Therefore, the input of the real-value NN models used in this paper (in the regression task), is the time-domain vector delayed by k symbols (the memory vector) containing the real and imaginary parts of both polarizations for the symbol at the time-step k and its $2N$ neighboring (past and future) symbols. In the NN signal processing, due to the computational memory constraints the input layer receives just a portion of the total data, called the mini-batch, as far as the finite computational resources limit the length of the sequences with which we can operate. The NN input mini-batch shape can be defined by three dimensions: $(B, M, 4)$, where B is the mini-batch size, M is the memory size defined through the number of neighbors N as $M = 2N + 1$, and 4 is the number of features for each symbol, referring to the real and imaginary parts of two polarization components. The output target is to recover the real and imaginary parts of the k-th symbol of one of the polarization, so the shape of the NN output batch can be expressed as $(B, 2)$.

In general, for all the NNs considered in this paper, we use the mean square error (MSE) loss estimator, since this choice corresponds to the conventional loss function frequently used for the regression tasks [26]. The other types of loss functions such as the mean absolute error, the Huber Loss, and the Log-Cost loss, were also considered for our NNs, but they did not show any noticeable benefits compared to the MSE. Moreover, it is important to highlight that we decided to present just the regression task in this paper because (for our test case scenario) the results achieved by regression and classification algorithms were close, but some fewer epochs were needed in the case of regression to reach the lowest BER.

The classical Adam algorithm was chosen for the stochastic optimization step with the default learning rate equal to 0.001 [27]. All NNs were trained for at most 1000 epochs (if not stopped earlier because of negligible changes in the loss function value over 150 epochs) and, after every training epoch, we calculated the BER obtained using the independently generated testing dataset.

The dataset was composed of 2^{20} symbols for the training dataset and 2^{18} independently generated symbols for the evaluation. To eliminate any possible data periodicity and overestimation [28] in our experiment, a pseudo-random bit sequence (PRBS) of order 32 was used to generate those datasets with different random seeds for each of them. The periodicity of the data is, therefore, 2^{10} times higher than our training dataset size, since the modulation format used in our study was the 16 QAM. For the simulation, the Mersenne twister generator [29], which has periodicity equal to $2^{19937} - 1$, was used with a different random seed. Additionally, we highlight that the NN training data were shuffled using `numpy.random.shuffle` function in Python before

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

feeding the dataset into the NN: such a shuffling helps to mitigate overfitting. The experimental setups and scenarios in which the datasets were acquired are described in the following sections.

The following subsections will delve deeper into the design of the NN models used within this paper.

A. A multi-layer perceptron

The first and, perhaps, simplest and well-studied NN-based equalizer that we consider is the MLP, proposed for the short-haul coherent system equalization in [22] and the long-haul systems in [30]. The MLP is a deep feed-forward densely connected NN structure that handles the I/Q components for each polarization jointly, providing two outputs for each processed symbol: its real and imaginary parts. Due to the MLP's ability to process joint I/Q components, the equalizer can learn the nonlinear phase impairments in addition to the amplitude-related nonlinearities. When using the MLP, the channel and device-induced memory effects are taken into account by incorporating the time-delayed versions of the input signal, as was carried out in [30].

In a simulation environment, the MLP equalizer showed performance metrics similar to those delivered by the "traditional" digital back-propagation (DBP) with 2 steps-per-span and 2 samples-per-symbol at 1000 km of standard single-mode fiber [30]. In our current paper, we use the same 3-layer MLP as in [22], but in our case here *the number of neurons and the activation function optimized for each layer*. Importantly, the number of layers in MLP, which is 3, has been found as optimal for our particular transmission scenario by the Bayesian optimizer (BO). However, this MLP topology rendered the BO, can alter essentially for different transmission scenarios.

The general equation, in a matrix form, describing the output vector y given the input x passing through the 3-layer MLP, is:

$$y=\phi\left\{\phi\left[\phi(x\times W_{n_1}+b_1)\times W_{n_2}+b_2\right]\times W_{n_3}+b_3\right\}\times W_{out}, \quad (1)$$

where x is the input vector with n_i elements, y is the output vector with n_o elements, ϕ is a nonlinear activation function, $W_{n_1} \in \mathbb{R}^{n_i\times n_1}$, $W_{n_2} \in \mathbb{R}^{n_1\times n_2}$, $W_{n_3} \in \mathbb{R}^{n_2\times n_3}$ and $W_{out} \in \mathbb{R}^{n_3\times n_o}$ are the real weight matrices of the respective dimensions participating in each layer of the MLP, $b_{1,2,3}$ are the bias vectors, the indexes $n_{1,2,3}$ stand for the number of neurons in each hidden layer, and $\times$ in (1) is the matrix-vector convolution.

B. Long short-term memory NNs

Compared to static (memoryless) systems where the MLPs can be efficient, the time sequences usually ought to be approached dynamically. Thus, recurrent NNs (RNNs) are often favored over other NN models for time sequences. However, training the recurring connections can be a much more complicated task compared to the MLP training, so that the network weights are usually changed almost imperceptibly. This aspect of RNNs often leads to the well-known vanishing gradient problem [26], [31]. The LSTM networks were built to solve it and to harness the memory-related effects. The LSTM comprises a gateway architecture that includes three gate types: the input (i_t) gates, the forget (f_t) gates, and the output (o_t) gates, as shown in Fig. 1. The compact form of the forward pass LSTM cell equations for a time-step t, (i.e. when we process the input feature sequence x_t having the size n_i) is [32], [33]:

$$\begin{aligned} i_t &= \sigma(W^i x_t + U^i h_{t-1}), \\ f_t &= \sigma(W^f x_t + U^f h_{t-1}), \\ o_t &= \sigma(W^o x_t + U^o h_{t-1}), \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(W^c x_t + U^c h_{t-1}), \\ h_t &= o_t \odot \tanh(C_t), \end{aligned} \quad (2)$$

where C_t is the cell state vector, h_t is the current hidden state vector of the cell with size n_h and h_{t-1} is the previous hidden state vector. Note that n_i is equal to the number of features, and n_h is the number of hidden units that will be chosen in the design process. The trainable parameters of the LSTM network are represented by the matrices $W \in \mathbb{R}^{n_h\times n_i}$ and $U \in \mathbb{R}^{n_h\times n_h}$ with the respective upper indices i, f, o, and c, referring to the particular LSTM gates mentioned previously. More details are given in Fig. 1. In (2), $\odot$ is the element-wise product, and σ denotes the logistic sigmoid activation function. The aim of the *input* i-gate is to store the content to the cell; the *forget* f-gate defines what information is to be erased; the *output* o-gate defines what information has to be passed to the next cell.

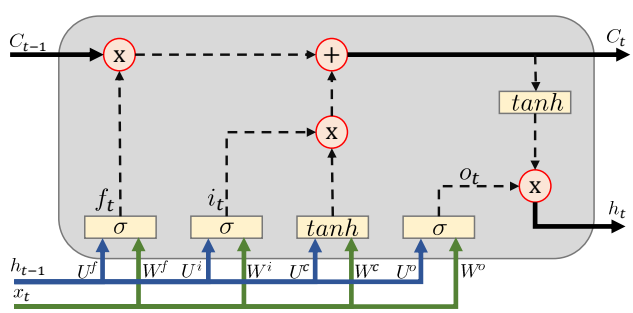


Fig. 1: The Long Short-Term Memory (LSTM) cell diagram representing graphically the operations described by (2) for one time-step. The arrows represent the "flow" of respective variables (the blue/green ones refer to the previous state and current input), the rectangles identify the nonlinear functions, while the symbols in circles identify the respective mathematical operations.

What makes the usage of the LSTM a dynamical approach is: the time sequence is processed by the array of LSTM cells ranging over the t-interval of interest, which is the memory size in our case. Besides the "dynamical" LSTM property, the *bidirectional* LSTM (biLSTM) provides a more robust solution for time series since with the bidirectional structure, we are learning the weights from the past visible values to the future hidden values, and that corresponds to our learning which features of the past values are useful for a

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

particular symbol value prediction [34]. In the optical channel equalization context, the key advantage of biLSTM is that it can efficiently handle intersymbol interference (ISI) between the preceding and the following symbols.

In the context of channel equalization, the LSTM was suggested in [35], [36] to reduce the transmission impairments in IM/DD systems with pulse amplitude modulation (PAM). The LSTM-based approach was developed further in [17], where, for the first time, the biLSTM was used in an optical coherent system to compensate for fiber nonlinearities, but only in a simulation environment. Additionally, it was shown that the biLSTM also outperformed a low-complexity DBP [17]. More recently, a bi-vanilla RNN was applied as well for the soft-demapping nonlinear ISI [10]. In our current study, we use a similar structure as in [17] , where the NN model is made up of a bidirectional LSTM layer followed by a dense layer. Finally, we note that, in contrast to the previous studies where the grid search was executed to guess the optimal number of hidden unities and memory size, this paper uses the BO to identify the best-performing biLSTM structure [9].

C. Echo state networks

The ESN is a promising type of RNNs due to its relaxed training complexity and its ability to preserve the temporal features from different signals over time [21], [37]–[40]. The ESNs are in the reservoir computing (RC) category because in the ESNs only the output weights are trainable. In Fig. 2, the grey-colored area is the reservoir "main" structure containing the randomly connected "neurons" that capture the time features of the signal, while the output weights are trained to define which states are more relevant to describe the desired output. In this paper, we use the concept of leaky-ESN [41] containing no output feedback connections. Our motivation to choose the leaky-ESN architecture is that there was an experimental observation that the leaky-ESN configuration outperforms the traditional ESN in feature extraction for noisy time series [38]. The latter is, evidently, an important property in optical transmission-related tasks. The leaky-ESN is formalized for a certain time-step t, as follows:

$$a_t = \phi\left(W_r \times s_{t-1} + W_{\text{in}} \times x_t\right), \tag{3}$$

$$s_t = (1-\mu)s_{t-1} + \mu a_t, \tag{4}$$

$$y_t = W_o \times s_t, \tag{5}$$

where $s_t \in \mathbb{R}^{N_r}$ is the system state at time-step t, N_r is the number of hidden neurons units in the dynamic layers, which represents the dimensionality in the reservoir; $x_t \in \mathbb{R}^{n_i}$ and $y_t \in \mathbb{R}^{n_o}$ are the input and the output vector of the ESN, respectively; $W_r \in \mathbb{R}^{N_r \times N_r}$ is a reservoir weight matrix that defines which neuron units are connected (including the self-connections); this matrix is also characterized by a sparsity parameter s_p defining the ratio of connected neurons to the total possible connections number. Finally, $W_{in} \in \mathbb{R}^{N_r \times n_i}$ is the input weight matrix, μ is the leaking rate parameter, and $W_o \in \mathbb{R}^{n_o \times N_r}$ is the output weight matrix which is the only one that is trainable using a regression technique. This training phase in ESN does not affect the dynamics of the system, which makes it possible to operate with the same reservoir for different tasks [37]. A schematic representation of a leaky-ESN, including the sequential input, dynamic, static, and output layers, as depicted in Fig. 2.

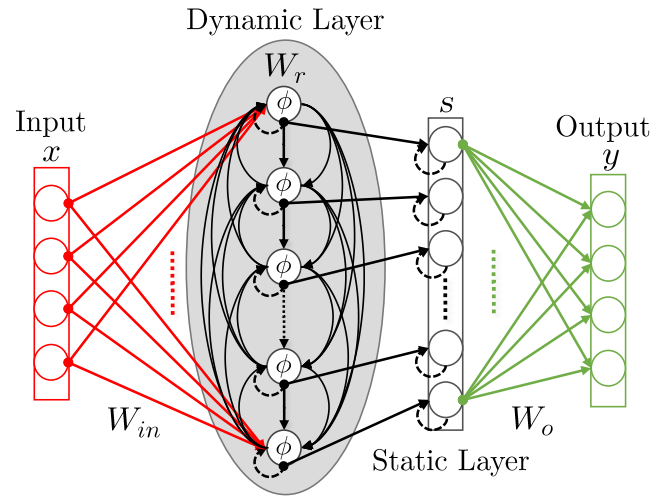


Fig. 2: Schematic of a leaky-ESN. A dynamical core called a reservoir is driven by input signal x. The states of the reservoir s are combined non-linearly to produce the output y. The reservoir consists of N interconnected nodes followed by a static (leaking) layer. The circular dashed lines in dynamic and static layers represent the past state values, while the solid lines represent the current time step value. The reservoir and input weights are fixed after initialization, while the output weights are learned using a regression technique.

The signal passing through the dynamic layer in Fig. 2 is represented by (3), and this layer is the core of the reservoir structure. Then it is followed by a static layer, represented by (4), which incorporates the leaky-ESN behavior through accumulating (integrating) its inputs, but it is also losing exponentially (leaking) accumulated excitation over time. Finally, the output layer defines which units are relevant to the description of the current task (for the equalization, in our case), and it is described by (5).

Concerning the previous ESN applications for optical channel equalization [39], the ESN was implemented in the optical domain for the distortions' mitigation: a 2 dB gain in Q^2-factor was achieved for 64-QAM 30 GBaud signals transmitted through 100 km fiber at 10 dBm input power. In addition, the same as it is in our paper, the reservoir can be applied in the digital domain. In [21], the leaky-ESN was successfully applied after the analog-to-digital converter to enable 80 km transmission to reach below KP4-FEC limit [42] for a 32 GBd on-off keying signal.

D. Convolutional neural networks

Due to their feature extraction propensity [26], the CNNs have become one of the most commonly used NN structures in such areas as 2D image classification and 3D video applications [43], [44]. Convolution layers have also been found efficient in the analysis of temporal 1D sequences with several applications to time series sensors, audio signals, and natural

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

language processing [45], [46]. For longer sequences, the CNN layer can be used as a *pre-processing step* due to its ability to reform the original sequence and extract its high-level features used for further processing cycles [24].

Here, we investigate, for the first time, two new models for the equalization of signal distortions in metro systems, combining a 1D convolutional layer performing the effective signal pre-processing with two previously proposed NN-based equalizers: the MLP described in Sec. II-A, and the biLSTM, Sec. II-B. These new structures, CNN+biLSTM and CNN+MLP, are addressed in our study because it was shown that the convolution layers are efficient in image denoising [47] and array signal processing [48], where the CNNs can reduce the background and quantization noise effects on coded signals. Therefore, we can naturally surmise that in our model the first convolutional layer can enhance the received signal by removing a part of the embedded noise before it enters the next neural layer. Also, generally, by adding the CNN layer, we end up with a NN model with less trainable parameters without losing performance, which can be yet another advantage. To that end, in the current study, we analyze how the combined NN architectures work for the optical channel equalization task. A simplified CNN+MLP combination was already successfully used in [24] at the transceiver for the high-baud-rate 80 km system.

The convolutional layer is primarily characterized by three key parameters: the number of filters, the size of its kernel, and the layer activation function. The extracting functionality is achieved by applying n_f filters, sliding over the raw input sequence, and generating the number of output maps equal to n_f, with a fixed kernel size n_k. The convolutional layer is constructed as a squash function, which means that the input is mapped to a lower-dimensional representation, in which only the main (or desirable) characteristics are retained. Since the CNNs were mainly developed in the context of image recognition and spacial feature extraction, other parameters such as padding, dilation, and stride, are also used in the design of the convolutional layers. Considering that the input shape is $(B, M, 4)$, the output shape after the CNN layer with all those parameters is defined as (B, L_{out}, n_f), where the parameter L_{out} is the function of the CNN hyper-parameters and defined as:

$$L_{out} = \left[\frac{M + 2 \cdot padding - dilation \cdot (n_k - 1) - 1}{stride} + 1 \right]. \quad (6)$$

However, in this paper, we will not focus on the investigation of those additional parameters. Consequently, we fix the default convolutional layer configuration with the padding equal to 0 (which corresponds to "valid" in Keras), the dilation equal to 1, and the stride equal to 1. Then, the input-output mapping of the convolution layer for this configuration can be described as follows:

$$y_i^f = \phi \left(\sum_{n=1}^{n_i} \sum_{j=1}^{n_k} x_{i+j-1,n}^{in} \odot k_{j,n}^f + b_{j,n}^f \right), \quad (7)$$

where y_i^f is the output feature map of the i-th input element produced by filter f in the CNN layer, x^{in} is the input raw data vector, k_j^f is the j-th trained convolution kernel of the filter f, and b_j^f is the bias of the filter f. Further, n is the feature index of the kernel and input data, ranging from 1 to n_i, corresponding to the number of features in the data; ϕ, as before, denotes the nonlinear activation function used in the convolutional layer. Note that (7) is true for all $i \in [1, ..., L_{out}]$. Moreover, since the pooling layer captures only the most important features in the data and ignores the less important ones [49], the pooling discretization process is not used in our equalizers to avoid the downsampling of feature sequences.

The output collection of feature maps, y^f, emerging from the convolutional layer, is then fed into one of the structures described above: either into two dense layers (MLP, where the number of layers is, again, dictated by the BO), forming the CNN+MLP structure or into the one biLSTM layer, resulting in CNN+biLSTM. We recall that we use the convolutional layer before the following layers to extract the middle-level locally invariant features from the input series.

Here we mention that even the CNNs alone are extremely powerful deep learning instruments that have a complicated multi-parametric structure that combines filters, kernel size, padding, stride, dilation, and pooling. However, having performed an exhaustive experimental exploration, we observed that deep CNNs have not reached the substantial performance level, like the one achieved by CNN+ MLP or CNN+biLSTM in our test case. Therefore, in this work, we utilize the convolution layers as pre-processing feature-extracting step and do not include deep CNN architectures in our current study.

III. Computational complexity of the NN-based equalizers

In this section, the computational complexity in terms of real multiplications per recovered output symbol is examined for all introduced NN architectures. We notice that the number of additions is typically neglected for such estimation in ordinary DSP techniques [50]. The major reason for this is that the typical algorithms for multiplying two integers with n digits have a computational complexity of $\mathcal{O}(n^2)$, whereas adding the same two numbers has a computational complexity of $\Theta(n)$ [51]. As a result, due to dealing with float values with 16 decimal digits, multiplication is by far the most time-consuming part of the implementation procedure.

Here we point out that the training complexity will not be considered since we evaluate the real-time computation complexity (evaluation phase), which is the most critical part, while the training of a NN equalizer is carried out offline (calibration phase). Also, the computational complexity of nonlinear activation functions is not considered in our framework, due to the fact that typically their operation is based on an approximation approach, rather than on direct multiplicative calculation. In the classical lookup tables-based (LUTs) approximation method, direct mapping can be digitally implemented with much fewer computations required to apply such activation functions [52], [53].

Early works presented the results regarding the complexity of the MLP [30], RNN [54], and LSTM [36] layers. However,

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

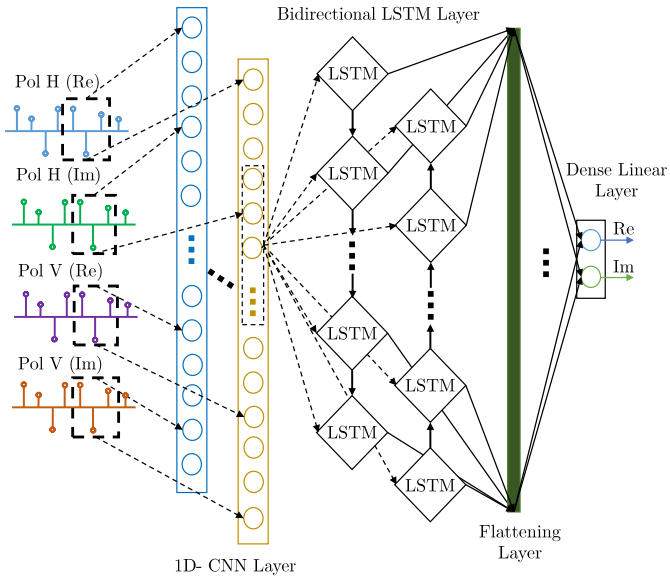


Fig. 3: Schematic of the convolutional-recurrent NN composed of an input layer corresponding to a time series with 4 features, followed by a 1D- convolutional layer (represented with two rectangles containing neurons) and a biLSTM layer, shown with the two lines (bi-directional) of lozenges, and ending with the flattering layer (a thick vertical deep-green line) and the output layer consisting two linear neurons to represent the real and imaginary part of the recovered symbol.

to enhance the understanding of this subject and clarify it, in our work we directly relate those complexities to the parameters of the most widely used machine learning platforms (Keras, TensorFlow, and PyTorch) without losing generality, and specifically addressing the composite NN types described before. Let the mini-batch size be B, n_s be the input time sequence size, with $n_s = M$, where M is the memory size (see also Sec. II), and n_i be the number of features, which in our case is equal to 4. Since we recover the real and imaginary parts of each symbol, the number of outputs per symbol, n_o, is equal to 2. For ESN, biLSTM, and CNN layers, as they require inputs in the form of tensors of rank 3, the input of the NN equalizer can be parametrized as $[B, n_s, n_i]$, the three numbers defining the dimensions of the input tensor, as mentioned above. The parametrization for the MLP equalizer is simpler, with $[B, n_s \cdot n_i]$ defining the dimensions of the 2D tensor input. We use flattening layers when it was necessary to reduce the dimensionality of the data.

In this case, considering three dense layers with n_1, n_2, and n_3 neurons, respectively, the complexity C_{MLP} of the resulting NN is given by:

$$C_{\text{MLP}} = \underbrace{n_s n_i n_1}_{a_1} + \underbrace{n_1 n_2 + n_2 n_3}_{b_1} + \underbrace{n_3 n_o}_{c_1}, \tag{8}$$

where a_1 is the contribution of the input layer, b_1 is the contribution of the hidden layer, and c_1 refers to the contribution of the output layer. The subindex "1" in a, b, and c explicitly associates these parameters with the MLP architecture

The next part presents the computational complexity for an NN-based equalizer composed of a biLSTM layer. Assuming that the biLSTM layer has n_h hidden units, the complexity of such a NN is given by:

$$C_{\text{biLSTM}} = 2\underbrace{n_s n_h (4n_i + 4n_h + 3 + n_o)}_{a_2}, \tag{9}$$

where a_2 is the contribution of the only layer, while the subindex "2" attributes the number a to the biLSTM. This expression is easier to understand if we analyze the mathematical description of the LSTM cell, see (2) and Fig. 1. We have several contributions to the cell's complexity. In the first layer we have $4n_i n_h$ multiplications associated with the input vector x_t. Then, $4n_h^2$ multiplications are due to the operations with the previous cell output h_{t-1}. Afterward, $3n_h$ and $n_o n_h$ multiplications due to the internal multiplications identified with $\odot$ and involving the current cell output (h_t) going into the output layer, respectively, are added. Lastly, we multiply the number of operations by the number of time steps in the layer, n_s. Since the topology is bidirectional, the total contribution is also multiplied by 2.

Following Sec. II, now we address the computational complexity associated with the ESN equalizer. Before presenting the respective expression, it is important to emphasize two aspects. First, the implementation of the ESN in the digital domain does not benefit from the fact that only the output layer weights are trainable, since, as mentioned previously, the training is not a key bottleneck as it is carried out during the offline calibration process. Second, the complexity of the ESN can potentially drop drastically if we implement it in the optical domain as an ESN dynamic layer, as it was noted in [39]. However, in this paper, we analyze the ESN implementation in the digital domain, similarly to [21].

Considering the leaky-ESN definition given by (3)–(5), the computational complexity of this equalizer can be expressed as:

$$C_{\text{ESN}} = n_s \left(\underbrace{n_i N_r + N_r^2 s_p}_{a_3} + \underbrace{2N_r}_{b_3} + \underbrace{N_r n_o}_{c_3} \right). \tag{10}$$

In the expression above, a_3 represents the contributions of (3), where the input layer adds $n_i N_r$ multiplications whereas the dynamic layers add $N_r^2 s_p$. b_3 refers to the contributions of (4) describing the static layer, and c_3 represents the multiplications in the output layer, (5). This overall process is repeated for all n_s time steps. Note that in the case of a potential optical implementation of the ESN, a_3 and b_3 would be equal to zero, and only the final weights would be learned in the digital domain.

Finally, let us address the complexity of the composite structures: CNN+MLP and CNN+biLSTM. The computational complexity of a 1-D convolutional layer is described as:

$$C_{\text{CNN}} = n_i n_f n_k \left[\frac{n_s + 2\,padding - dilation(n_k - 1) - 1}{stride} + 1 \right] \tag{11}$$

However, we assumed (7) that the convolutional layer is defined by the number of filters n_f, the kernel size n_k, and that the number of time steps $n_s \geq n_k$, according to (6)

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

the output size for each filter of the CNN is $(n_s - n_k + 1)$. (12) and (13) are the expressions for the complexity of a convolutional layer combined with two dense layers or one biLSTM layer, respectively:

$$C_{\text{CNN+MLP}} = \underbrace{n_i n_f n_k (n_s - n_k + 1)}_{a_4} + \underbrace{(n_s - n_k + 1) n_f}_{b_4} \underbrace{n_1 + n_1 n_2 + n_2 n_o}_{c_4}, \tag{12}$$

$$C_{\text{CNN+biLSTM}} = \underbrace{n_i n_f n_k (n_s - n_k + 1)}_{a_5} + \underbrace{(n_s - n_k + 1)}_{b_5} \underbrace{2 n_h [4 n_f + 4 n_h + 3 + n_o]}_{c_5}. \tag{13}$$

In this scenario, the two-layer MLP has n_1 and n_2 neurons in each respective layer, and the biLSTM layer has n_h hidden units. In the equations above, a_4 and a_5 are the contributions of the convolutional layer, b_4 is the correction factor for the transition between layers since the flattening layer was placed before the dense layers; b_5 is the number of time-steps for the following biLSTM layer; c_4 is the contribution of the two-layer MLP; and c_5 is the contribution of the biLSTM layer where, in this case, the number of filters, n_f, is equal to the number of features entering the LSTM cell.

Finally, we would like to express the computational complexity of the DBP-based receiver used in this paper for benchmark purposes. We considered a basic implementation of the DBP algorithm [55], where each propagation step comprises a linear part for dispersion compensation followed by a nonlinear phase cancellation stage. The linear part is achieved with a zero-forcing equalizer by transforming the signal in the frequency domain and multiplying with the inverse dispersion transfer function of the propagation section. The complexity of the DBP in terms of RMpS is [30], [50]:

$$C_{\text{DBP}} = 4 N_{span} N_{step} \left(\frac{n\, N_{\text{FFT}} [\log_2(N_{\text{FFT}}) + 1]}{(N_{\text{FFT}} - N_D + 1)} + n \right), \tag{14}$$

where N_{step} is the number of steps per span used, N_{FFT} is the FFT size, n is the oversampling ratio, and $N_D = \tau_D / T$, where τ_D corresponds to the dispersive channel impulse response and $T = 1/R_s$ is the symbol duration. We have considered that $N_{\text{FFT}} = 256$ and τ_D defined as:

$$\tau_D = \frac{1.1 R_s c |D| L_{span}}{f_c^2 N_{steps}}, \tag{15}$$

where f_c is the optical carrier reference frequency that in our case was 193.41 THz, c is the speed of light, L_{span} is the span length and D is the fiber dispersion parameter.

IV. Performance versus Computational Complexity Trade-off Analysis

In this section, we initially describe the numerical and experimental scenarios used in this paper to analyze and compare the functioning of the equalizers detailed in Sec. II. After that, the two types of analysis for our set of NN structures are carried out. First, we present the maximum performance improvement (in terms of Q-factor gain compared to the non-equalized case) that each equalizer can deliver and compare this gain to the respective computational complexity corresponding to each optimized equalizer. Then, we decrease the computational complexity of six NN topologies from Sec. II and present the gain improvement provided by each NN-equalizer when all NNs have approximately the same computational complexity. This enables us to investigate the dependence of optical performance on the computational complexity and to identify which equalizer is better for a certain complexity level.

A. Experimental and numerical setups

The setup used in our experiment is depicted in Fig. 4. At the transmitter, a DP-16QAM 34.4 Gbaud symbol sequence was mapped out of data bits generated by a $2^{32} - 1$ PRBS. Then, a digital RRC filter with roll-off 0.1 was applied to limit the channel bandwidth to 37.5 GHz. The resulting filtered digital samples were resampled and uploaded to a digital-to-analog converter (DAC) operating at 88 GSamples/s. The outputs of the DAC were amplified by a four-channel electrical amplifier which drove a dual-polarization in-phase/quadrature Mach–Zehnder modulator, modulating the continuous waveform carrier produced by an external cavity laser at $\lambda = 1.55\,\mu m$. The resulting optical signal was transmitted over 9×50 km spans of TWC optical fiber with EDFA amplification. The optical amplifier noise figure was in the 4.5 to 5 dB range. The parameters of the TWC fiber – at $\lambda = 1.55\,\mu m$ – are: attenuation coefficient $\alpha = 0.23$ dB/km, dispersion coefficient $D = 2.8$ ps/(nm·km), and effective nonlinear coefficient γ = 2.5 $(W \cdot km)^{-1}$.

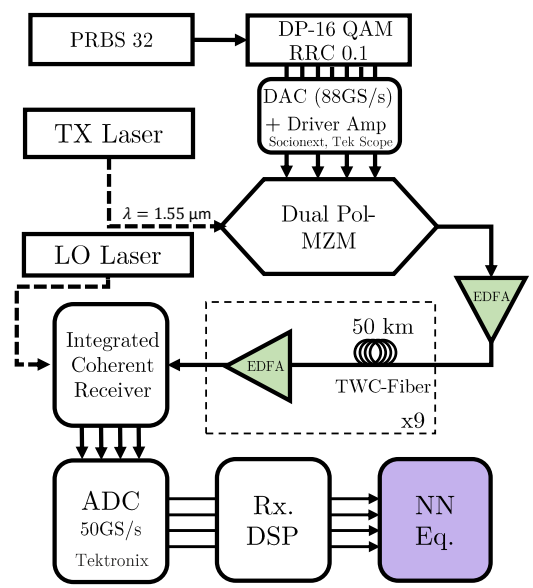


Fig. 4: Experimental setup used to analyze the performance of different NN equalizers; further details are reported in Sec. IV-A. The input of the NN (shown as the purple rectangle at the bottom right) is the soft output of the regular DSP just before the decision unit.

At the RX side, the optical signal was converted into the

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

electrical domain using an integrated coherent receiver. The resulting signal was sampled at 50 Gsamples/s by a digital sampling oscilloscope and processed by an offline DSP based on the algorithms described in [56]. Firstly, the bulk accumulated dispersion was compensated using a frequency domain equalizer, which was followed by the removal of carrier frequency offset. A constant-amplitude zero-autocorrelation-based training sequence was then located in the received frame and the equalizer transfer function was estimated from it. After the equalization, the two polarizations were demultiplexed and the signal was corrected for clock frequency and phase. Carrier phase estimation was then achieved with the help of pilot symbols. Thereafter, the resulting soft symbols were used as input for the NN equalizers. Finally, the pre-FEC BER was evaluated from the signal at the NN output.

With regard to simulation, we mimic the experimental transmission setup[1]. The optical signal propagation along an optical fiber was simulated by solving the Manakov equations via the split-step Fourier method (with a resolution of 1km per step). Every span was followed by an optical amplifier with the noise figure NF $= 4.5$ dB, which fully compensates fiber losses and adds amplified spontaneous emission noise. At the receiver, after full electronic chromatic dispersion compensation (CDC) by the frequency-domain equalizer and downsampling to the symbol rate, the received symbols were normalized to the transmitted ones. Finally, we added Gaussian noise to the signal representing an additional transceiver distortion that we may have in the experiment, such that the Q-factor level of the simulated data matched the experimental one. The system performance is evaluated in terms of the Q-factor, defined as: $Q = 20\log_{10}\left[\sqrt{2}\,\mathrm{erfc}^{-1}(2\,\mathrm{BER})\right]$.

B. Optimized NN-based architectures

In this section, we show the maximum achievable Q-factor for all equalizers without constraining the computational complexity. The Bayesian optimization (BO) tool, introduced in [9] for optical NN-based equalizers, was implemented to identify the optimum values of hyper-parameters for each NN topology, which provides the best Q-factor in the experimental test dataset. As it was recently shown, the BO renders superior performance compared to other types of search algorithms for machine learning hyperparameter tuning [57]. The same topologies (without further optimization) were tested for the numerical analysis as well. The search space used in the BO procedure was defined via the allowed hyper-parameters intervals: N =[1 to 50], n_f =[1 to 1000], n_k =[1 to 20], n_h =[1 to 1000], n_1 =[1 to 1000], n_2 =[1 to 1000], n_3 =[1 to 1000], N_r =[1 to 1000], s_p =[0 to 1], μ =[0 to 1], and spectral radius=[0 to 1] .

In Table. I, the line marked with the "**Best Topology**" label, summarizes the hyper-parameters obtained by the BO. These values are used to count the real multiplications per symbol recovery (complexity), and to assess the equalizers' performance expressed via the Q-factor gain, Fig. 5. Note that for all equalizers, the same optimal number of taps found by the BO was $N = 20$, which means that the memory in our equalizers is $M = 41$ and the mini-batch size, B, is equal to 4331. Moreover, for the ESN, the BO found the best value $\mu = 0.57$, and the optimal spectral radius equal to 0.667. The activation functions found for every hidden NN layer are summarized as following: 1D-CNN layer – '$linear$' activation function followed by $LeakyReLU$ (Leaky version of a Rectified Linear Unit) with negative slope coefficient alpha = 0.2; biLSTM layer – hyperbolic tangent ('$tanh$')

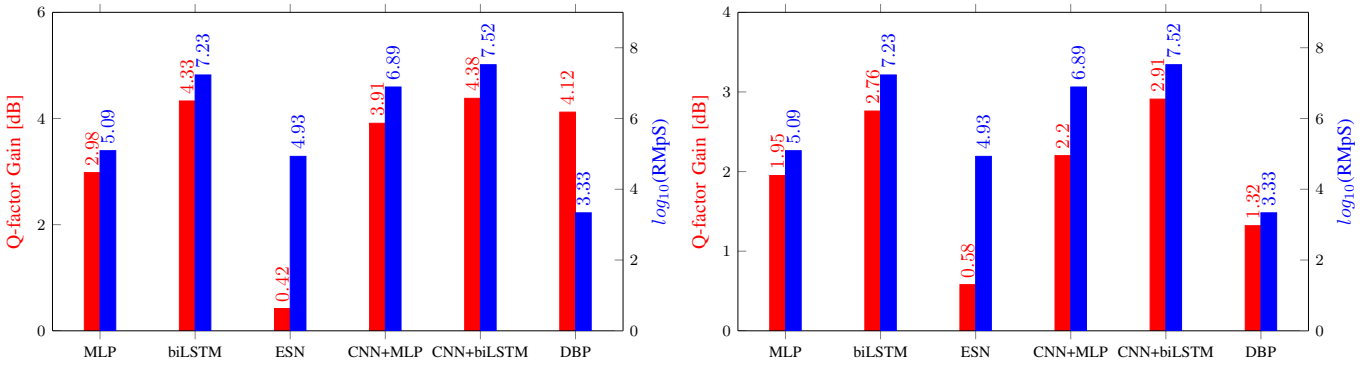
[1]We consider a DP-16QAM, single-channel signal at 34.4 Gbaud pre-shaped by an RRC filter with 0.1 roll-off transmissions with an upsampling rate of 8 samples per symbol (275.2 GSamples/s) over a system consisting of 9×50 km TWC-fiber spans.

TABLE I: Summary of the complexity attributing to each NN equalizer topology: the topology type is identified in the left-most column. The complexity corresponding to each topology and the NN type is expressed in terms of real multiplications per symbol recovered (RMpS), highlighted in red. In this table, we also depict the hyper-parameters distributions found by the BO: the cell marked as "Best topology" and other 6 topologies (Topologies from 1 to 6, referring to the increasing complexity threshold number) for the study of complexity versus performance. In addition, for all topologies, the values of n_s, n_i and n_o were: 41, 4 and 2, respectively, and these are not reported in the table.

Best Topology	**CNN+biLSTM**				**biLSTM**		**ESN**		
	n_f	n_k	n_h	RMpS	n_h	RMpS	N_r	s_p	RMpS
	244	10	226	2.7E+07	226	1.7E+07	88	0.18	8.6E+04
	CNN+MLP						**MLP**		
	n_f	n_k	n_1	n_2	RMpS	n_1	n_2	n_3	RMpS
	470	10	456	467	7.7E+06	149	132	596	1.2E+05
Topology 1	**CNN+biLSTM**				**biLSTM**		**ESN**		
	n_f	n_k	n_h	RMpS	n_h	RMpS	N_r	s_p	RMpS
	1	10	1	2.1E+03	1	2.0E+03	6	0.18	2.2E+03
	CNN+MLP						**MLP**		
	n_f	n_k	n_1	n_2	RMpS	n_1	n_2	n_3	RMpS
	2	5	10	10	2.3E+03	10	10	25	2.0E+03
Topology 2	**CNN+biLSTM**				**biLSTM**		**ESN**		
	n_f	n_k	n_h	RMpS	n_h	RMpS	N_r	s_p	RMpS
	5	10	3	1.3E+04	4	1.2E+04	22	0.18	1.1E+04
	CNN+MLP						**MLP**		
	n_f	n_k	n_1	n_2	RMpS	n_1	n_2	n_3	RMpS
	9	5	12	30	1.1E+04	40	40	80	1.1E+04
Topology 3	**CNN+biLSTM**				**biLSTM**		**ESN**		
	n_f	n_k	n_h	RMpS	n_h	RMpS	N_r	s_p	RMpS
	20	10	10	1.1E+05	16	1.1E+05	100	0.18	1.1E+05
	CNN+MLP						**MLP**		
	n_f	n_k	n_1	n_2	RMpS	n_1	n_2	n_3	RMpS
	50	9	30	100	1.1E+05	170	170	300	1.1E+05
Topology 4	**CNN+biLSTM**				**biLSTM**		**ESN**		
	n_f	n_k	n_h	RMpS	n_h	RMpS	N_r	s_p	RMpS
	50	10	41	1.0E+06	53	1.0E+06	350	0.18	1.0E+06
	CNN+MLP						**MLP**		
	n_f	n_k	n_1	n_2	RMpS	n_1	n_2	n_3	RMpS
	300	10	70	200	1.1E+06	600	600	900	1.0E+06
Topology 5	**CNN+biLSTM**				**biLSTM**		**ESN**		
	n_f	n_k	n_h	RMpS	n_h	RMpS	N_r	s_p	RMpS
	244	10	108	1.0E+07	172	1.0E+07	1150	0.18	1.0E+07
	CNN+MLP						**MLP**		
	n_f	n_k	n_1	n_2	RMpS	n_1	n_2	n_3	RMpS
	600	12	500	500	1.0E+07	2100	2100	2500	1.0E+07
Topology 6	**CNN+biLSTM**				**biLSTM**		**ESN**		
	n_f	n_k	n_h	RMpS	n_h	RMpS	N_r	s_p	RMpS
	400	10	455	1.0E+08	550	1.0E+08	3660	0.18	1.0E+08
	CNN+MLP						**MLP**		
	n_f	n_k	n_1	n_2	RMpS	n_1	n_2	n_3	RMpS
	1000	10	2900	2200	1.0E+08	7050	7050	7000	1.0E+08

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology



(a) Simulation results (b) Experimental results

Fig. 5: Comparison of the computational complexity versus performance for the different NN-based equalizer considered within this paper with their optimized architectures and the DBP with 3 StPS. The number over each bar gives the 10 logarithm of the number of multiplications per recovery symbol.

activation function; ESN layer – '$tanh'$ activation function; MLP layer – '$tanh'$ activation function.

The results obtained by using the numerical synthetic data are presented in Fig. 5a. First, the CNN+biLSTM turned out to be best-performing in terms of the Q-factor gain: it achieved a 4.38 dB Q-factor improvement when compared to the conventional DSP algorithms [56], 0.05 dB when compared to the biLSTM equalizer level, 0.47 dB when compared to the CNN+MLP equalizer level, 1.4 dB when compared to the MLP equalizer level, and 3.96 dB when compared to the ESN equalizer level. Second, when adding the convolutional layers to MLP and biLSTM, we observed the improvement in terms of the number of epochs needed to reach the highest performance: the single-layer biLSTM required 119 epochs, while the CNN+biLSTM reduced this number to 89 epochs; the MLP itself needed 214 epochs to reach the best performance level, and the CNN+MLP required just 100 epochs. Thus, we conclude that the addition of a convolutional layer indeed renders the enhancement in the NN structure's performance and assists in the training stage.

When considering how NN equalizers function with the experimental data, Fig. 5b, we can mention two major observations. First, similarly to the numerical results, the CNN+biLSTM is best-performing among all the considered NN structures in terms of the Q-factor gain. The CNN+biLSTM demonstrated a 2.91 dB improvement when compared to the conventional DSP, 0.15 dB when compared to the biLSTM equalizer, 0.61 dB when compared to the CNN+MLP equalizer, 0.96 dB when compared to the MLP equalizer, and 2.33 dB when compared to the ESN equalizer. Additionally, as was also observed in the numerical analysis, a lower number of training epochs was necessary to reach the best performance point when we add a convolutional layer: using the CNN+biLSTM we needed 169 epochs, while for the pure biLSTM this number was 232 epochs; the number of epochs required for the CNN+MLP to reach the best performance was 107, and for the pure MLP it was 753 epochs. Second, compared to the simulation, the overall gain of all NN-based equalizers is slightly reduced. This can be explained by the existing "reality gap" between the numerical model and the true experimental transmission results. In a real transmission, extra nonlinearity and the non-ideal behavior of transceivers (signal clipping by the ADC/DAC, harmonic and intermodulation distortions of the driver amplifier (DA), I/Q skew, etc.) add extra noise and complexity to the process of channel inversion. We believe that with just the split-step method, the NNs can unroll the synthetic propagation effects more easily than reverting the actual propagation in the experimental condition. We also point out that even though the gain numbers are different in the numerical and experimental data, the NN structures' performance followed the same pattern for both numerical and experimental cases: the best performance was attributed to the CNN+biLSTM, the next level performance pertains to the biLSTM, followed by the CNN+MLP, the MLP and, finally, the ESN.

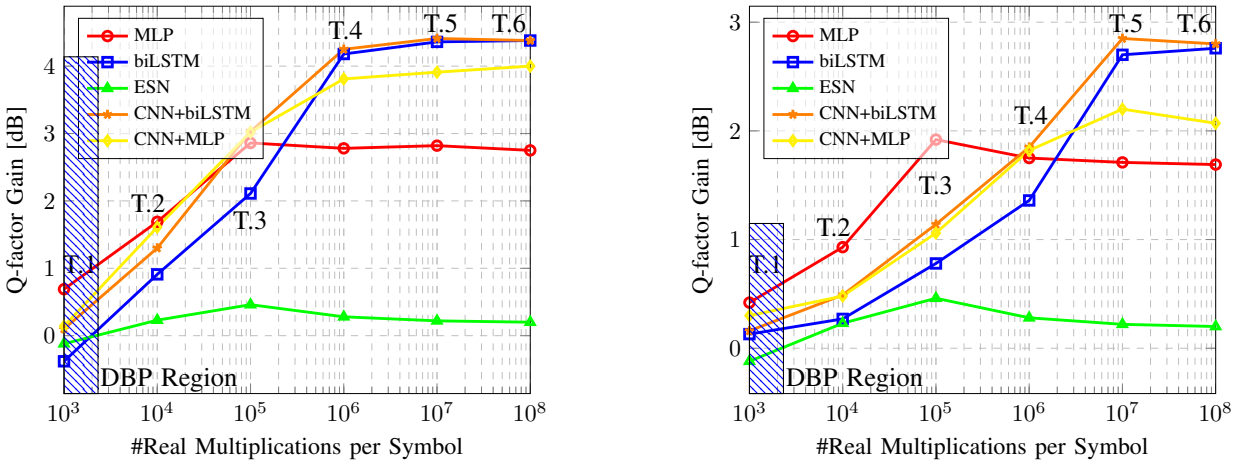
Finally, of all equalizer types investigated in this study, the DBP 3 StPS applied with two samples per symbol was still the least complex method. In all simulation and experiment test cases, however, the CNN+biLSTM outperformed the 3 StPS DBP, as shown in Fig. 5. Even by optimizing the DBP's nonlinear coefficient parameter (γ), the DBP approach was able to enhance the Q-factor only by 1.32 dB, whereas the CNN+biLSTM equalizer improved it by 2.91 dB, in the experimental case. The boost in the performance in the experiment scenario provided by the CNN+biLSTM relative to the DBP demonstrates the NN-equalizer's power in mitigating transmission impairments in a practical application.

C. Comparative analysis of different NN-based equalizers with the fixed computational complexity

The analysis given above does not address the question of which NN topology would provide the best gain if we restrict the NN structure's complexity to a certain level. To answer this question, we retested the equalizers constraining the total number of real multiplications per recovered symbol (RMpS). We considered the complexity values in the range from 10^3 to 10^8 RMpS. We note that the NN structures with large RMpS ($\sim 10^8$) can be prohibitively complex for efficient hardware implementation. However, Ref. [58] demonstrated an efficient

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology



(a) Simulation results

(b) Experimental results

Fig. 6: Q-factor gain dependence on the constrained multiplications number for the equalizers having different architectures, presented in Sec. II, in the case of DP-16 QAM single channel TWC-fiber 9×50km. The power level is 2 dBm, which guarantees the high enough nonlinearity transmission regime.

FPGA implementation of LSTM NN with 256 and 512 hidden units. This result reveals that the architectures outlined in this research are still feasible for realistic signal processing when advanced techniques for NN hardware implementation are used.

The hyper-parameters distributions for each NN architecture with the complexity constraint are summarized in Table I in the cells marked from ''Topology 1" to "Topology 6". The parameters of those topologies were also tuned by the BO: for each case, we reduced the allowed BO search range to comply with each computational complexity constraint.

As seen in Fig. 6, for different allowed computational complexity levels, the performance ranking of equalizer types changes. Several conclusions can be drawn analyzing the results emerging from the simulated (Fig. 6a) and experimental (Fig. 6b) data. First, in the experimental scenario, the best complexities corresponding to the maximum gain coincide with the complexities identified by the BO procedure, which confirms the effectiveness of the BO in finding the "right" NN architecture. Second, in simulations, the maximum performance is reached already at a lower complexity level compared to the experimental results. As it can be seen from the experimental figure, the CNN+biLSTM, CNN+MLP, and biLSTM equalizers need $\approx 10^7$ RMpS, while in the simulation $\approx 10^6$ RMpS was already enough to achieve the best performance. This observation further confirms that the NN can cope with the reversion of the simulated channel more easily than with the reversion of experimentally obtained data. Third, when we increase the complexity above the level determined by the BO, the gain remains nearly constant: this is due to overfitting and it is particularly pronounced in the MLP scenario. The key concept of the function approximation capability of the MLP belongs to its number of i) feed-forward hidden layers and ii) hidden neurons; these two parameters define the NN's capacity [59]. Changing the MLP's capacity by adjusting the complexity levels frequently leads to unpredictable changes in the NN's performance. Starting at the 10^5 complexity level for both simulation and experimental layouts, we can see that the MLPs with oversized capacity suffer from overfitting, as the network memorizes the properties of the training set in such detail that it can no longer efficiently recover the information from the inference dataset [59]. The latter blockades the equalizer from providing further Q-factor improvement. Thus, we argue that the architectures found by the BO identify the most appropriate NN equalizer's capacity (structure) matching our problem, and a further increase in complexity cannot render any noticeable performance improvement.

Next, we note that for the high level of RMpS (Topologies 4, 5, and 6), the best-performing equalizer is the CNN+biLSTM. However, once we reduce the number of real multiplications from Topology 3 and below, the best-performing equalizer turns out to be the traditional MLP. This can be explained by the fact that advanced architectures, such as CNN and biLSTM, require more filters and a higher number of hidden units, respectively, to learn the complete dynamics of the data. Also, we observe that the CNN+biLSTM performs similarly to the CNN+MLP at low complexity levels (orange and yellow curves in Fig. 6), and similarly to the biLSTM (blue line) at high complexity. Consequently, we can infer how the addition of a convolutional layer works: while for high complexity the blue and orange curves are approximately the same, at a lower allowed complexity level the CNN+biLSTM performs better.

In addition, we used the hatched blue zone in both simulation and experimental cases, attributed to the traditional DBP with 3 StPS, to highlight the performance of the NN equalizers with similar computational complexity to the DBP. Then, it is evident that reducing the number of neurons, filters, and hidden units is not the optimal technique to achieve low complexity architectures, because the performance fell below the DBP level. As a possible alternative, pruning and quantization techniques [60], [61] can be used to minimize the computational complexity of the NN equalizers without compromising their performance, making the NN equalizers appealing not only for their good performance but also for

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

their decreased complexity.

Finally, the performance shown by the ESN does not meet the expectations, where we observed the lowest achievable gain numbers. However, [62] contains the results explaining the poor ESN performance for the nonlinear wireless scenario. It was shown that in the channel with a high level of noise, the ESN-based indeed performs poorly. Furthermore, in that Ref. it was demonstrated that by increasing the ESNs' number of neurons (i.e. its complexity), and, thus, effectively increasing the hidden dimensionality of the representation, the equalization performance worsens. Moving to the nonlinear optical channel equalization, we observed both aforementioned effects: the performance was relatively poor due to the high level of noise, and the performance did not improve when we increased the complexity, as can be seen from the behavior of the green curve in Fig. 6.

V. Conclusion

In this paper, we proposed and examined the novel designs of combined NN-based post-equalizers: (a) CNN+MLP and (b) CNN+biLSTM, for the equalization of coherent optical fiber channels. We reviewed and compared several key existing NN-based methods with the proposed new algorithms using both the numerically simulated synthetic data and the experimental data from the benchmark transmission system. One of the important outcomes of our work lies in the reported analytical expressions for the complexity (the number of real multiplications) associated with each NN type considered in the paper. Although a comparative analysis has been carried out for a specific benchmark system, we believe that our findings are relatively generic and can be applied to other scenarios.

Fiber Kerr nonlinearity was the predominant source of signal deterioration in the experimental benchmark system used for comparing different channel equalizers. In order to analyze the equalizers functioning with the clear nonlinear signal distortions, we used a low dispersion TWC fiber and processed the data at 2 dBm signal launch power. We emphasize that the trade-off conclusions for each NN equalizer's performance and complexity are unique to the system under consideration in this paper. However, we believe that our research paves the way for a rigorous methodology that can be used for estimating the computational cost of various NN-based channel equalizers.

We described in detail the design of the selected most promising NN-based equalizers. To derive the best-performing NN structures, we utilized the Bayesian optimization of each NN type that provides the optimized set of hyper-parameters for each particular NN-based equalizer type. For these optimized structures, we found that the best performance of the test system was rendered by the new CNN+biLSTM architecture, though the performance of the pure biLSTM was only slightly lower. However, the optimized CNN+biLSTM design corresponded to the highest complexity among all cases studied.

The important part of the analysis was the comparison of the performance under the condition of the restricted complexity: the respective results are given in the last section. We found that at high complexity levels, the best-performing NN among studied cases was the CNN+biLSTM. However, when reducing the complexity, we observed the transition: when the allowed complexity is relatively low, the best-performing structure turned out to be the simple MLP. We can explain this behavior as follows: the advanced architectures (the CNN and biLSTM) require more complexity-hungry components (filters or hidden units) to learn the data dynamics, while the MLP is less demanding using just the summation and activation functions at the basic level. Overall, we conclude that the addition of the convolutional layer can be beneficial if we do not restrain the complexity. However, complexity can play a crucial role in the hardware implementation of the NN equalizers. Our analysis demonstrates that even the simple NN structures, like the MLP, can outperform the more advanced counterparts when the complexity is constrained to relatively low levels.

References

[1] M. A. Jarajreh, E. Giacoumidis, I. Aldaya, S. T. Le, A. Tsokanos, Z. Ghassemlooy, and N. J. Doran, "Artificial neural network nonlinear equalizer for coherent optical ofdm," *IEEE Photonics Technology Letters*, vol. 27, no. 4, pp. 387–390, 2015.

[2] D. Wang, M. Zhang, Z. Li, C. Song, M. Fu, J. Li, and X. Chen, "System impairment compensation in coherent optical communications by using a bio-inspired detector based on artificial neural network and genetic algorithm," *Optics Communications*, vol. 399, pp. 1–12, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030401817303346

[3] C. Häger and H. D. Pfister, "Nonlinear interference mitigation via deep neural networks," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*. IEEE, 2018, pp. 1–3.

[4] C. Huang, S. Fujisawa, T. F. de Lima, A. N. Tait, E. Blow, Y. Tian, S. Bilodeau, A. Jha, F. Yaman, H. G. Batshon, H. Peng, B. J. Shastri, Y. Inada, T. Wang, and P. R. Prucnal, "Demonstration of photonic neural network for fiber nonlinearity compensation in long-haul transmission systems," in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*, 2020, pp. 1–3.

[5] B. I. Bitachon, A. Ghazisaeidi, M. Eppenberger, B. Baeuerle, M. Ayata, and J. Leuthold, "Deep learning based digital backpropagation demonstrating snr gain at low complexity in a 1200km transmission link," *Opt. Express*, vol. 28, no. 20, pp. 29 318–29 334, Sep 2020. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-28-20-29318

[6] Y. Zhao, X. Chen, T. Yang, L. Wang, D. Wang, Z. Zhang, and S. Shi, "Low-complexity fiber nonlinearity impairments compensation enabled by simple recurrent neural network with time memory," *IEEE Access*, vol. 8, pp. 160 995–161 004, 2020.

[7] M. M. Melek and D. Yevick, "Nonlinearity mitigation with a perturbation based neural network receiver," *Optical and Quantum Electronics*, vol. 52, no. 10, pp. 1–10, 2020.

[8] S. Zhang, F. Yaman, K. Nakamura, T. Inoue, V. Kamalov, L. Jovanovski, V. Vusirikala, E. Mateo, Y. Inada, and T. Wang, "Field and lab experimental demonstration of nonlinear impairment compensation using neural networks," *Nature communications*, vol. 10, no. 1, pp. 1–8, 2019.

[9] P. J. Freire, V. Neskornuik, A. Napoli, B. Spinnler, N. Costa, G. Khanna, E. Riccardi, J. E. Prilepsky, and S. K. Turitsyn, "Complex-valued neural network design for mitigation of signal distortions in optical links," *Journal of Lightwave Technology*, vol. 39, no. 6, pp. 1696–1705, 2021.

[10] M. Schaedler, F. Pittala, G. Böcherer, C. Bluemm, M. Kuschnerov, and S. Pachnicke, "Recurrent neural network soft-demapping for nonlinear isi in 800gbit/s dwdm coherent optical transmissions," in *46th European Conference on Optical Communication (ECOC 2020)*, 2020.

[11] J. Estaran, R. Rios-Müller, M. Mestre, F. Jorge, H. Mardoyan, A. Konczykowska, J.-Y. Dupuy, and S. Bigo, "Artificial neural networks for linear and non-linear impairment mitigation in high-baudrate im/dd systems," in *ECOC 2016; 42nd European Conference on Optical Communication*. VDE, 2016, pp. 1–3.

[12] L. Yi, T. Liao, L. Xue, and W. Hu, "Neural network-based equalization in high-speed pons," in *2020 Optical Fiber Communications Conference and Exhibition (OFC)*. IEEE, 2020, pp. 1–3.

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

[13] 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," *Optical Fiber Communication Conference*, no. paper M2B.2., 2018.

[14] M. Liang and J. Du, "Research on signal recovery method of im/dd optical fiber transmission system based on multi-bit and multi-class classification convolutional neural network," in *Asia Communications and Photonics Conference*. Optical Society of America, 2020, pp. M4A–231.

[15] S. M. Ranzini, F. Da Ros, H. Bülow, and D. Zibar, "Optoelectronic signal processing for chromatic dispersion mitigation in direct detection systems," in *22nd International Conference on Transparent Optical Networks*. IEEE, 2020.

[16] X. Dai, X. Li, M. Luo, Q. You, and S. Yu, "Lstm networks enabled nonlinear equalization in 50-gb/s pam-4 transmission links," *Applied optics*, vol. 58, no. 22, pp. 6079–6084, 2019.

[17] S. Deligiannidis, A. Bogris, C. Mesaritakis, and Y. Kopsinis, "Compensation of fiber nonlinearities in digital coherent systems leveraging long short-term memory neural networks," *Journal of Lightwave Technology*, vol. 38, no. 21, pp. 5991–5999, 2020.

[18] O. Sidelnikov, A. Redyuk, S. Sygletos, M. Fedoruk, and S. K. Turitsyn, "Advanced convolutional neural networks for nonlinearity mitigation in long-haul wdm transmission systems," *Journal of Lightwave Technology*, pp. 1–1, 2021.

[19] O. S. Sidelnikov, A. A. Redyuk, S. Sygletos, and M. P. Fedoruk, "Methods for compensation of nonlinear effects in multichannel data transfer systems based on dynamic neural networks," *Quantum Electronics*, vol. 49, no. 12, p. 1154, 2019.

[20] C. Häger and H. D. Pfister, "Physics-based deep learning for fiber-optic communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 280–294, 2020.

[21] S. M. Ranzini, R. Dischler, F. Da Ros, H. Buelow, and D. Zibar, "Experimental investigation of optoelectronic receiver with reservoir computing in short reach optical fiber communications," *Journal of Lightwave Technology*, 2021.

[22] M. Schaedler, C. Bluemm, M. Kuschnerov, F. Pittala, S. Calabro, and S. Pachnicke, "Deep neural network equalization for optical short reach communication," *Applied Sciences*, vol. 9, no. 21, p. 4675, 2019.

[23] B. Karanov, M. Chagnon, V. Aref, F. Ferreira, D. Lavery, P. Bayvel, and L. Schmalen, "Experimental investigation of deep learning for digital signal processing in short reach optical fiber communications," in *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, 2020, pp. 1–6.

[24] V. Bajaj, F. Buchali, M. Chagnon, S. Wahls, and V. Aref, "Single-channel 1.61 tb/s optical coherent transmission enabled by neural network-based digital pre-distortion," in *2020 European Conference on Optical Communications (ECOC)*. IEEE, 2020, pp. 1–4.

[25] P. J. F. de Carvalho Souza, "Zoo of neural network based equalizers," Mar. 2021, . [Online]. Available: https://doi.org/10.5281/zenodo.4582298

[26] C. C. Aggarwal, *Neural networks and deep learning*. Springer, 2018.

[27] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.

[28] T. A. Eriksson, H. Bülow, and A. Leven, "Applying neural networks in optical communication systems: possible pitfalls," *IEEE Photonics Technology Letters*, vol. 29, no. 23, pp. 2091–2094, 2017.

[29] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, vol. 8, no. 1, pp. 3–30, 1998.

[30] O. Sidelnikov, A. Redyuk, and S. Sygletos, "Equalization performance and complexity analysis of dynamic deep neural networks in long haul transmission systems," *Optics Express*, vol. 26, no. 25, pp. 32 765–32 776, 2018.

[31] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition,," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[32] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE access*, vol. 6, pp. 1662–1669, 2017.

[33] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[34] T. Osogami, H. Kajino, and T. Sekiyama, "Bidirectional learning for time-series models with hidden units," in *International Conference on Machine Learning*, 2017, pp. 2711–2720.

[35] X. Dai, X. Li, M. Luo, Q. You, and S. Yu, "LSTM networks enabled nonlinear equalization in 50-gb/s PAM-4 transmission links," *Applied Optics*, vol. 58, no. 22, pp. 6079–6084, 2019.

[36] X. Lu, C. Lu, W. Yu, L. Qiao, S. Liang, A. P. T. Lau, and N. Chi, "Memory-controlled deep LSTM neural network post-equalizer used in high-speed PAM VLC system," *Optics Express*, vol. 27, no. 5, pp. 7822–7833, 2019.

[37] P. Verzelli, C. Alippi, and L. Livi, "Echo state networks with self-normalizing activations on the hyper-sphere." *Scientific reports*, vol. 9, no. 1, pp. 1–14, 2019.

[38] C. Sun, M. Song, S. Hong, and H. Li, "A review of designs and applications of echo state networks," *arXiv preprint arXiv:2012.02974*, 2020.

[39] M. Sorokina, "Dispersion-managed fiber echo state network analogue with high (including thz) bandwidth," *Journal of Lightwave Technology*, vol. 38, no. 12, pp. 3209–3213, 2020.

[40] H.-P. Ren, H.-P. Yin, C. Bai, and J.-L. Yao, "Performance improvement of chaotic baseband wireless communication using echo state network," *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6525–6536, 2020.

[41] Q. Wu, E. Fokoue, and D. Kudithipudi, "On the statistical challenges of echo state networks and some potential remedies," *arXiv preprint arXiv:1802.07369*, 2018.

[42] E. El-Fiky, A. Samani, D. Patel, M. Jacques, M. Sowailem, and D. V. Plant, "400 gb/s o-band silicon photonic transmitter for intra-datacenter optical interconnects," *Optics express*, vol. 27, no. 7, pp. 10 258–10 268, 2019.

[43] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Twenty-second international joint conference on artificial intelligence*, 2011.

[44] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[45] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354 – 377, 2018.

[46] Y. LeCun and Y. Bengio, *Convolutional Networks for Images, Speech, and Time Series*. Cambridge, MA, USA: MIT Press, 1998, p. 255–258.

[47] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[48] Z. Zhao, H. Liu, and T. Fingscheidt, "Convolutional neural networks to enhance coded speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 663–678, 2018.

[49] A. Hassan and A. Mahmood, "Convolutional recurrent deep learning model for sentence classification," *Ieee Access*, vol. 6, pp. 13 949–13 957, 2018.

[50] B. Spinnler, "Equalizer design and complexity for digital coherent receivers," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 16, no. 5, pp. 1180–1192, 2010.

[51] S. Jahani, "Zot-mk: a new algorithm for big integer multiplication," *MSc MSc, Department of Computer Science, Universiti Sains Malaysia, Penang*, 2009.

[52] F. Piazza, A. Uncini, and M. Zenobi, "Neural networks with digital lut activation functions," in *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, vol. 2, 1993, pp. 1401–1404 vol.2.

[53] Y. Xie, A. N. Joseph Raj, Z. Hu, S. Huang, Z. Fan, and M. Joler, "A twofold lookup table architecture for efficient approximation of activation functions," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 12, pp. 2540–2550, 2020.

[54] Q. Zhou, C. Yang, A. Liang, X. Zheng, and Z. Chen, "Low computationally complex recurrent neural network for high speed optical fiber transmission," *Optics Communications*, vol. 441, pp. 121–126, 2019.

[55] C.-Y. Lin, A. Napoli, B. Spinnler, V. Sleiffer, D. Rafique, M. Kuschnerov, M. Bohn, and B. Schmauss, "Adaptive digital back-propagation for optical communication systems," in *Optical Fiber Communication Conference*. Optical Society of America, 2014, pp. M3C–4.

[56] M. Kuschnerov, M. Chouayakh, K. Piyawanno, B. Spinnler, E. De Man, P. Kainzmaier, M. S. Alfiad, A. Napoli, and B. Lankl, "Data-aided versus blind single-carrier coherent receivers," *IEEE Photonics Journal*, vol. 2, no. 3, pp. 387–403, 2010.

[57] R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, and I. Guyon, "Bayesian optimization is superior to random search

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JLT.2021.3096286, Journal of Lightwave Technology

for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020," *arXiv preprint arXiv:2104.10201*, 2021.

[58] Z. Que, Y. Zhu, H. Fan, J. Meng, X. Niu, and W. Luk, "Mapping large lstms to fpgas with weight reuse," *Journal of Signal Processing Systems*, vol. 92, no. 9, pp. 965–979, 2020.

[59] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[60] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *arXiv preprint arXiv:1710.09282*, 2017.

[61] X. Long, Z. Ben, and Y. Liu, "A survey of related research on compression and acceleration of deep neural networks," *Journal of Physics: Conference Series*, vol. 1213, p. 052003, jun 2019. [Online]. Available: https://doi.org/10.1088/1742-6596/1213/5/052003

[62] S. S. Mosleh, L. Liu, C. Sahin, Y. R. Zheng, and Y. Yi, "Brain-inspired wireless communications: Where reservoir computing meets mimo-ofdm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4694–4708, 2018.

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