Towards FPGA Implementation of Neural Network-Based Nonlinearity Mitigation Equalizers in Coherent Optical Transmission Systems

Pedro J. Freire[(1)], Michael Anderson[(1)], Bernhard Spinnler[(2)], Thomas Bex[(2)], Jaroslaw E. Prilepsky[(1)], Tobias A. Eriksson [(3)], Nelson Costa[(4)], Wolfgang Schairer[(2)], Michaela Blott[(5)], Antonio Napoli[(2)], Sergei K. Turitsyn[(1)]

[(1)] Aston University, United Kingdom p.freiredecarvalhosourza@aston.ac.uk [(2)] Infinera, Munich, Germany [(3)] Infinera, Stockholm, Sweden[(4)] Infinera, Lisbon, Portugal [(5)] AMD/Xilinx Inc, Dublin, Ireland

Abstract *For the first time, recurrent and feedforward neural network-based equalizers for nonlinearity compensation are implemented in an FPGA, with a level of complexity comparable to that of a dispersion equalizer. We demonstrate that the NN-based equalizers can outperform a 1-step-per-span DBP.*

Introduction

Upcoming technologies and innovations (6G, high-speed access networks, etc.) will push optical networks to meet even more stringent traffic requirements. However, the Kerr effect limits the information rates that optical fiber communication systems can achieve[1]–[3]. Machine learning-based solutions have lately been touted as a possible approach to mitigate the impact of this fiber transmission effect[4]–[8]. In particular, neural network (NN) algorithms have already been shown to outperform traditional digital backpropagation (DBP) solutions while requiring less computational complexity[9],[10]. However, research has often been carried out with simulated datasets only, with the complexity comparisons being made in terms of number of real multiplications.

In this paper, we make an important step forward in assessing the viability of NN-based equalizers for industrial applications by benchmarking i) their performance versus the 1-step-per-span (StpS) DBP using 2.3 samples/symbol (Sa/symbol) in experiments and ii) their computational complexity by comparing an FPGA implementation against the full electronic chromatic dispersion compensation (CDC) block (used, e.g., in standard DSP chain[11]), which needs far less resources than 1 StpS DBP. In addition, to the best of our knowledge, for the first time, we present the FPGA implementation of a NN-based equalizer that employs the bidirectional recurrent layer with LSTM cells (biLSTM). By transmitting a 34 GBd single-channel, dual-polarization (SC-DP) 16QAM signal over 17×70 km of large-effective area fiber (LEAF) (both simulated and experimental), we report ≈ 1.7 dB Q-factor improvement over a standard DSP chain while requiring only $\approx 2.5\times$ more FPGA resources than the CDC block implementation to achieve 400G transmission.

Neural Network Design on FPGAs

The two investigated NN architectures are depicted in Fig. 1 (a) and (b) for the biLSTM-based equalizer and the deep CNN-based equalizer, respectively. The shape of both architectures is similar, but the mathematical operation nature in each case is different: a) consists of a recursive type whereas b) of a feedforward type. A total of 81 symbols are used as input to simultaneously retrieve 61 symbols on the output. In Fig. 1, the hidden layer in (a) consists of a biLSTM layer with $n_h = 35$ hidden units, and in (b) it is made up of a CNN layer with 70 filters ($n_{f_1} = n_{f_2} = 35$), with zero padding applied to retain the shape. The output layer in both designs is a convolutional layer with $n_f = 2$ filters, a kernel size of $n_k = 21$, and without any padding[1]. Both hidden layers used Tanh activation functions, and the output layer includes only a linear function.

Concerning the FPGA realization, as depicted in Fig. 1 (c), the design can be broken down into three steps: i) training, ii) high-level synthesis (HLS), and iii) hardware synthesis (RTL synthesis + physical design). In step i), we used the standard TensorFlow library to train the NNs and saved their weights in a fixed-point representation (`int32`) to be used as input to the NN realization. Then, for each TensorFlow layer, we wrote the respective C++ functions. Afterwards, in step ii), we translated the C++ NN architectures into a hardware description language (VHDL in our case) using the Xilinx software Vitis HLS. Finally,

---

[1]Those parameters were chosen based on grid search analysis to meet the hardware constraints, required throughput, and optical performance.
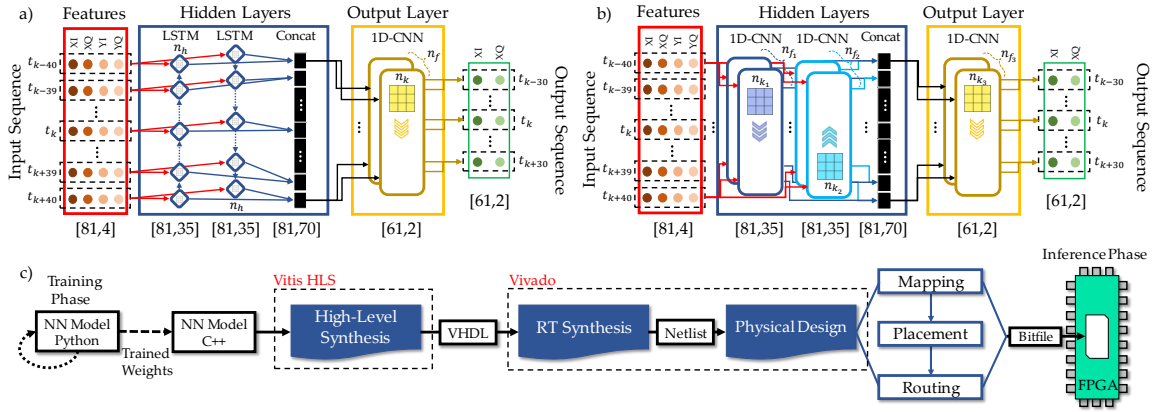
Fig. 1: Recurrent equalizer using a) the bidirectional LSTM layer; b) the feedforward equalizer using a 1D-convolutional layer. The pipeline process of the FPGA design from the training in Python to the final realization step on the FPGA is depicted in c).

in step iii), the bitstream was generated using the Xilinx software Vivado. This is when the place-and-route process is performed and utilization, chip routing, and other time constraints are verified and defined in a final implemented design. Regarding the CDC implementation, we designed a time-domain equalizer with 556 taps in C++ and followed the same design steps ii) and iii) as in the NN implementation.

Experimental and Simulation Setups

At the transmitter, a DP-16QAM 34 Gbd symbol sequence was mapped out of the data bits generated by a Mersenne Twister algorithm. Then, a digital RRC filter with 0.1 roll-off was applied. The resulting filtered digital samples were resampled and uploaded to a digital-to-analog converter (DAC) operating at 88 GSamples/s. The output of the DAC was amplified by a four-channel electrical amplifier which drove a dual-polarization IQ 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 17×70 km spans of LEAF, with the loss in each fiber span being fully compensated at its output using Erbium-doped fiber amplifiers (EDFAs). The EDFAs noise figure was in the 4.5 to 5 dB range. The parameters of the LEAF are: attenuation coefficient $\alpha = 0.225$ dB/km, chromatic dispersion coefficient $D = 4.2$ ps/(nm·km), and effective nonlinear coefficient $\gamma = 2$ (W· km)$^{-1}$. On the RX side, the optical signal was converted to the electrical domain using an integrated coherent receiver. The resulting signal was sampled at 80 Gsamples/s by a digital sampling oscilloscope and processed using the offline DSP described in[12]. 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. Concerning the simulation, we mimic the experimental trans-

mission setup. The signal propagation along the fiber was simulated by solving the Manakov equations. At the receiver, after full CDC and down-sampling to the symbol rate, the received symbols were normalized to the transmitted ones. In addition, Gaussian noise was added to the data signal, representing additional transceiver distortions observed in the experiment. As a result, the Q-factor level of the simulated data matched the experimental one.

Unlike the NN equalizer, which operates with 1 Sa/symbol, the DBP operated with 2.3 Sa/symbol (and with 1 StpS with the fiber parameters optimized for its best performance). We trained the NN with the data retrieved when setting the highest launch power, namely with 2^{18} symbols randomly picked after each epoch from a training dataset of size 2^{20}, mini-batch size of 2000 and learning rate equal to 0.0005 over 30k epochs[2]. The test dataset had 2^{18} symbols never considered before (not used in the training phase). This dataset was used to report performance in terms of the Q-factor calculated from the BER.

Results and Discussion

Figures 2 summarize our studies of computational complexity and optical performance. First, the results referring to the simulated data are given in Fig. 2 (a). We can observe that the biLSTM equalizer gives approximately the same performance as 1 StpS DBP, improving the optimal power from -1 dBm to 2 dBm and the Q-factor by 1.3 dB. The deep CNN performed worse, improving the optimal power from -1 dBm to 1dBm and the Q-factor by 0.8 dB. Moving to the experimental data[3], we observe that biLSTM and deep

---

[2]For applying the trained model to other launch powers, we used transfer learning[13], thus requiring less than 5 epochs to adjust the NN weights to the other launch powers.

[3]The Q-factor produced by the Python model and the FPGA implementation were virtually identical because we did not consider quantization at this moment.
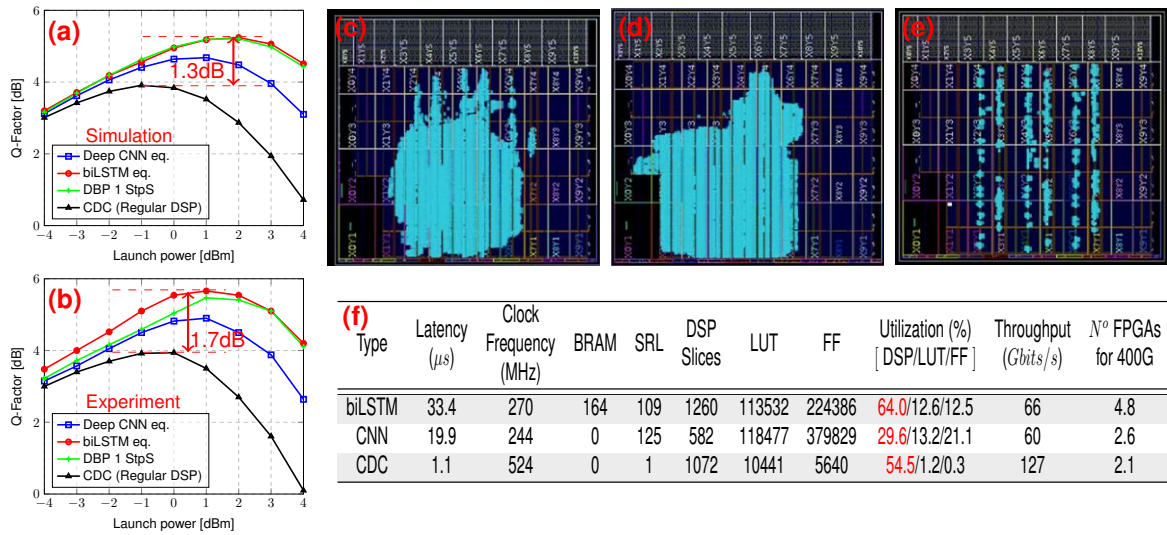
Fig. 2: Q-factor versus launch power for (a) simulation and (b) experiment corresponding to the transmission of a SC-DP 16QAM 34 GBd signal along 17×70km of LEAF. Implementation in the EK-VCK190-G-ED Xilinx FPGA of the (c) biLSTM eq., (d) Deep CNN eq., (e) CDC block. Table (f) shows the latency, hardware usage and throughput for the different designs required to recover 61 symbols simultaneously as well as the estimated amount of FPGAs required to achieve 400G transmission.

Type	Latency (μs)	Clock Frequency (MHz)	BRAM	SRL	DSP Slices	LUT	FF	Utilization (%) [DSP/LUT/FF]	Throughput ($Gbits/s$)	N^o FPGAs for 400G
biLSTM	33.4	270	164	109	1260	113532	224386	64.0/12.6/12.5	66	4.8
CNN	19.9	244	0	125	582	118477	379829	29.6/13.2/21.1	60	2.6
CDC	1.1	524	0	1	1072	10441	5640	54.5/1.2/0.3	127	2.1

CNN provided 1.7 dB and 1 dB in Q-factor improvement, respectively, and the optimal power shifts from 0 dBm to 1 dBm in both cases. For the DBP case, the Q-factor improvement in the experiment was 1.5 dB and in the simulation was 1.3 dB. Noticeably, in the experimental study, the NN led to a higher Q-factor improvement, even in the linear regime, when compared with the simulation results[4].

Figures 2 (c), (d), and (e) show the real implementation of the biLSTM, the deep CNN equalizers, and the CDC, respectively, on the state-of-the-art EK-VCK190-G-ED Xilinx FPGA chip[14]. This chip is partitioned into 40 clock regions, with the blue areas in Figs. 2 (c), (d), and (e) corresponding to the used chip resources. In Fig. 2 (f), we summarize the most important information on the implementation of such equalizers in the FPGA. We come to three important conclusions: 1) although biLSTM provided higher Q-factor improvement, its feedback loop connections caused a bottleneck in the design, resulting in higher latency (33 μs) and lower clock frequency (270 MHz). On the other hand, deep CNN and CDC could be parallelized more efficiently, reducing their latency to 19.9 μs and 1.1 μs, respectively. However, because the CDC has one filter whereas the deep CNN has 70 filters, we see this parallelization as being easier in the CDC implementation due to hardware restrictions, resulting in an operating frequency of 524 MHz for the CDC case and 244 MHz for the CNN case. Fig. 2 (e) clearly shows the CDC parallelization; 2) with re-

spect to FPGA utilization, the biLSTM equalizer was the only one that used Block RAM (BRAM) to save future/past recurrent states, while CNN and CDC did not need such blocks. Concerning the usage of DSP slices, LUT, and FF in each equalizer type, the biLSTM used 64% DSP slices and 13% of LUT and FF, the deep CNN used 30% DSP slices, 13% of LUT and 21% of FF, and the CDC used 54% DSP slices and 1% of LUT and FF; 3) with regard to throughput, the clock frequency is the maximum that each implementation can handle to comply with a zero-negative slack design. In this sense, the total throughput considering 16QAM modulation format is 66G, 60G, and 127G, respectively, for the biLSTM, the deep CNN, and the CDC block. Finally, considering that the FPGA can safely use 80% of its resources without having routing/congestion problems and using the maximum utilization reported in Fig. 2 (f), we observe that 400G transmission can be achieved using an equivalent FPGA that has the same capacity as ≈ 5 FPGAs (VCK190) in the case of biLSTM, ≈3 FPGAs in the case of deep CNN, and ≈ 2 FPGAs in the case of CDC.

Conclusions

We demonstrate the FPGA implementation of a recurrent NN equalizer with reduced complexity. We show that the biLSTM equalizer requires only ≈ 2.5 times more FPGA resources than the CDC block. Furthermore, the biLSTM equalizer can outperform DBP with 1StpS, showing its potential to mitigate nonlinear fiber transmission effects.

---

[4]This highlights the ability of the NN to mitigate the impact of not only the Kerr nonlinearity, but also of the components.

References

[1] J. C. Cartledge, F. P. Guiomar, F. R. Kschischang, G. Liga, and M. P. Yankov, "Digital signal processing for fiber nonlinearities [invited]", *Opt. Express*, vol. 25, no. 3, pp. 1916–1936, Feb. 2017. DOI: 10.1364/OE.25.001916. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-25-3-1916.

[2] A. Ghazisaeidi, I. Fernandez de Jauregui Ruiz, L. Schmalen, *et al.*, "Submarine transmission systems using digital nonlinear compensation and adaptive rate forward error correction", *Journal of Lightwave Technology*, vol. 34, no. 8, pp. 1886–1895, 2016. DOI: 10.1109/JLT.2016.2518929.

[3] Y. C. Gultekin, A. Alvarado, O. Vassilieva, *et al.*, "Mitigating nonlinear interference by limiting energy variations in sphere shaping", in *2022 Optical Fiber Communications Conference and Exhibition (OFC)*, 2022, pp. 1–3.

[4] F. Musumeci, C. Rottondi, A. Nag, *et al.*, "An overview on application of machine learning techniques in optical networks", *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1383–1408, 2018.

[5] F. N. Khan, Q. Fan, C. Lu, and A. P. T. Lau, "An optical communication's perspective on machine learning and its applications", *Journal of Lightwave Technology*, vol. 37, no. 2, pp. 493–516, 2019.

[6] V. Bajaj, F. Buchali, M. Chagnon, S. Wahls, and V. Aref, "Deep neural network-based digital pre-distortion for high baudrate optical coherent transmission", *Journal of Lightwave Technology*, vol. 40, no. 3, pp. 597–606, 2022. DOI: 10.1109/JLT.2021.3122161.

[7] S. Gaiarin, F. Da Ros, R. T. Jones, and D. Zibar, "End-to-end optimization of coherent optical communications over the split-step fourier method guided by the nonlinear fourier transform theory", *Journal of Lightwave Technology*, vol. 39, no. 2, pp. 418–428, 2021. DOI: 10.1109/JLT.2020.3033624.

[8] P. J. Freire, Y. Osadchuk, B. Spinnler, *et al.*, "Performance versus complexity study of neural network equalizers in coherent optical systems", *Journal of Lightwave Technology*, vol. 39, no. 19, pp. 6085–6096, 2021. DOI: 10.1109/JLT.2021.3096286.

[9] R. M. Butler, C. Hager, H. D. Pfister, G. Liga, and A. Alvarado, "Model-based machine learning for joint digital backpropagation and pmd compensation", *Journal of Lightwave Technology*, vol. 39, no. 4, pp. 949–959, 2021. DOI: 10.1109/JLT.2020.3034047.

[10] H. Ming, X. Chen, X. Fang, L. Zhang, C. Li, and F. Zhang, "Ultralow complexity long short-term memory network for fiber nonlinearity mitigation in coherent optical communication systems", *Journal of Lightwave Technology*, vol. 40, no. 8, pp. 2427–2434, 2022. DOI: 10.1109/JLT.2022.3141404.

[11] M. Kuschnerov, F. N. Hauske, K. Piyawanno, *et al.*, "Dsp for coherent single-carrier receivers", *Journal of Lightwave Technology*, vol. 27, no. 16, pp. 3614–3622, 2009.

[12] M. Kuschnerov, M. Chouayakh, K. Piyawanno, *et al.*, "Data-aided versus blind single-carrier coherent receivers", *IEEE Photonics Journal*, vol. 2, no. 3, pp. 387–403, 2010.

[13] P. J. Freire, D. Abode, J. E. Prilepsky, *et al.*, "Transfer learning for neural networks-based equalizers in coherent optical systems", *Journal of Lightwave Technology*, vol. 39, no. 21, pp. 6733–6745, 2021. DOI: 10.1109/JLT.2021.3108006.

[14] *Vck190 evaluation board*, UG1366, v1.1, Xilinx, 2022. [Online]. Available: https://www.xilinx.com/content/dam/xilinx/support/documents/boards_and_kits/vck190/ug1366-vck190-eval-bd.pdf.