FPGA Implementation of Deep Neural Network Based Equalizers for High-Speed PON

Noriaki Kaneda, Ziyi Zhu*, Chun-Yen Chuang, Amitkumar Mahadevan, Bob Farah, Keren Bergman*, Doutje Van Veen, and Vincent Houtsma,
Nokia Bell Labs, 600 Mountain Ave, Murray Hill NJ
**Columbia University, 116th and Broadway, New York, NY*
Noriaki.kaneda@nokia-bell-labs.com

Abstract: A fixed-point deep neural network-based equalizer is implemented in FPGA and is shown to outperform MLSE in receiver sensitivity for 50 Gb/s PON downstream link. Embedded parallelization is proposed and verified to reduce hardware resources. © 2020 The Author(s)

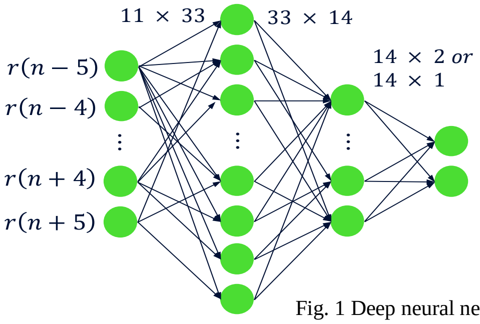
OCIS codes: (060.2330) Fiber optics communications; (200.4260) Neural networks

1. Introduction

As the data traffic in optical access networks experiences growth, the needs for higher data rates passive optical networks (PONs) beyond 10 Gb/s are imminent. Single wavelength 25 Gb/s PON is in the process of being standardized in IEEE, while the single wavelength 50 Gb/s PON has been proposed and is being discussed in ITUT as well. For 50 Gb/s TDM-PON the downstream wavelength of 1342+/-2 nm is under consideration. While using O-band wavelengths has advantage to be near dispersion zero for the standard single mode fiber, the dispersion at 1342 nm is no longer negligible for the high baud rate at 50 Gbaud. Combined with intensity modulation and direct detection (IMDD) proposed as the modulation and detection format for 50 Gb/s PON, the dispersion effect produces nonlinear distortion in the received signal that is the best handled by advanced equalizers to meet the required dispersion penalty. One such proposal includes usage of receiver side equalization by MLSE (maximum likelihood sequence estimator) to keep the dispersion penalty less than 1 dB at the Soft-FEC limit [1]. While MLSE is generally considered as one of the best performing equalizers for optical network systems, recent studies show the promising results with new approaches for signal equalization employing machine learning techniques, particularly deep neural networks (DNN) [2-4]. While the effectiveness of the DNN-based equalizers has been often reported against FFE (Feed Forward Equalizer) for example, there are limited reports with clear comparison to a better equalizer such as MLSE. In addition, the complexity analysis and report with respect to the real-world implementation of such DNN-equalizer has been scant except for the recent report of the 8 hidden layer DNN-based equalizer to recover 4.07 Gbaud 8-PAM signal in 5-bit quantized FPGA implementation [4]. In this paper, we show that the the 2 hidden layer DNN-based equalizers can outperform MLSE in sensitivity for 50 Gb/s PON link even when DNN are quantized. We then present the implementation of the DNNs with and without embedded parallelization in FPGA at lower clock rates to understand the required complexity of DNN equalizers applicable for 50 Gb/s PON link.

2. Deep Neural Network with Multiple Symbol Outputs

We chose a standard DNN architecture shown in Fig. 1 as the receiver side equalizer except that the output stage may have 1 or 2 symbol outputs. The DNN has 2 hidden layers and the number of inputs, the first hidden layer neurons, and the second hidden layer neurons are 11,33,14. The number of outputs can be either one or two with each representing a soft output of one or two OOK symbols, respectively. In this way, parallelization is embedded within the DNN design. The neurons between layers are fully connected with Eq. 1 representing the input-output signals of each layer where $\overrightarrow{a_k}$ is the output signal vector, $\overrightarrow{b_k}$ is the bias vector and $\boldsymbol{W_k}$ is the weight matrix at kth layer. The rectified linear unit (ReLU) which is shown in Eq. 2 is used as the activation function σ for the 2 hidden layers, while the linear function shown in Eq. 3 is used at the output neurons. While bipolar activation functions [3,4] have been reported previously, ReLU is unsigned with 1 bit more efficient usage of resolution and the simplest to realize in hardware. The required number of layers and number of neurons for each layer are determined by the parameter search. One of the important aspects in implementing the DNN in hardware is that the number of output symbols from a DNN is desired to be more than just one for an efficient parallelization. When the first hidden layer is sufficiently larger than the input width, we observe the number of output symbols can be more than 1 without significantly sacrificing the resulting BER. This observation is consistent with the sufficiency study of 2 hidden layer DNN reported in [5]. The number of input width on the other hand depends on the number of output width plus the expected impulse response of the inverse channel to deal with the dispersive channel. The weights and biases are obtained by the



$$\vec{a}_k = \sigma(\boldsymbol{W}_{\boldsymbol{k}}\vec{a}_{k-1} + \vec{b}_k) \qquad (1)$$

$$\sigma(d) = \begin{cases} d, when\ \ d_{max} \geq d \geq 0, \\ \quad 0, when\ d < 0 \end{cases} \qquad (2)$$

$$\sigma(d) = d, when\ \ d_{max} \geq d \geq\ d_{min}, \qquad (3)$$

Fig. 1 Deep neural network with 2 hidden layers and 2 symbol outputs.

backward propagation of DNN using Python TensorFlow with quantization embedded into the outputs, weights and biases of all DNN layers during the training. For the DNN-based equalization relevant to 50Gb/s PON channels, 8-bit resolution is found to be the minimum resolution before the BER performance is negatively affected.

2. Experimental Setup

The experimental setup of 50 Gb/s PON is shown in Fig.2. The Lithium niobate (LN) Mach-Zehnder modulator (MZM) is used for ease of downstream wavelength selection. We set the tunable laser wavelength to 1342 nm to emulate the elevated amount of dispersion for PON downstream. We used up to 30 km single-mode fiber to test the capability of the equalizer. Zero dispersion of the fiber used is believed to be ~1310 nm, translating into 83 ps/nm total fiber dispersion for 30 km. The signal is modulated with 88 GS/s CMOS 8-bit DAC (digital to analog converter) with pulse shaping to compensate for the bandwidth profile of the DAC and its eval board. The exact data rate used is 50.2857 Gb/s to adopt 1.75 sample per symbol. The 25 Gb/s class APD (avalanche photo diode) integrated with TIA (transimpedance amplifier) is used as the photoreceiver and the output of the receiver is captured in the real-time sampling scope. The received eye in Fig.2 shows large distortion from bandwidth limitation and 30km fiber dispersion. The captured data is then preprocessed offline down to 1 sample per symbol and used in the offline DNN training and testing as well as in the online testing of the DNN in FPGA. FPGA used in the test is the Xilinx XCZU9EG-FFVC900 device which is a ZynqMPSoC device with 4 onchip ARM application processors. The custom FPGA carrier board is developed to interface Trenz Electronic SoM (system on module) child board that has 4 GByte DDR4 memory directly connected to the PS (processor system) side of the FPGA. The measured data as well as the weights and bias values are stored first in SD card memory, then read into the DDR4 memory and finally are loaded into FPGA fabric for the online DNN process. Petalinux, the embedded Linux on Xilinx FPGA, is used as operating system (OS) to handle the data flow between storage, memories and FPGA fabric as well as to interface the user commands. We use a PRBS15 pattern based on the polynomial $X^{15}+X^{14}+1 = 0$ as transmit pattern. Since we center the 1 or 2 output symbols in the middle of the DNN input signal vector as in Fig.1, the DNN is not capable of learning the PRBS15 pattern until DNN input width exceeds 28 [6], therefore it is ensured that the presented results are not overfitted.

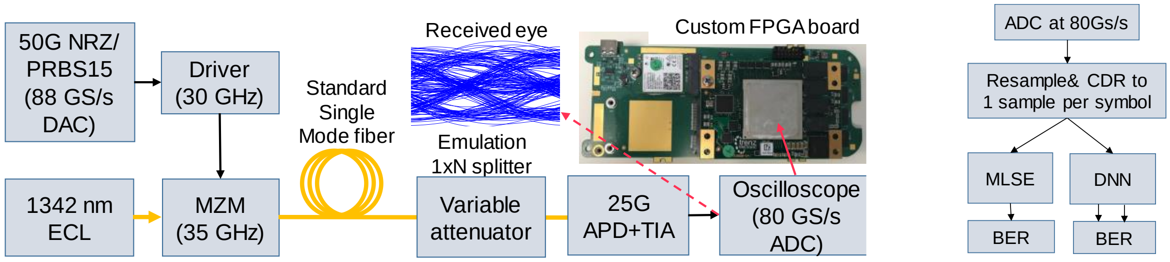


Fig.2 Left: Experimental setup of 50Gb/s downstream PON at 1342nm. Right: DSP block diagram for the MLSE and DNN.

2. DNN Equalizer Results and FPGA Resource Utilization

The measured data are processed offline both with MLSE and fixed point DNN and are plotted in Fig.3. The DSP (digital signal processing) diagrams are shown in Fig. 2 right. The number of taps used for MLSE is 6 and no gains are observed by increasing MLSE taps beyond 6 for both BTB (back-to-back) and 30km data. When they are compared at BER=1e-2, the DNN outperforms MLSE by 0.7 dB in sensitivity with 30 km link and ~0.2 dB with back-back measurement. The DNN results for 1 and 2 symbol outputs are almost overlapping near 1e-2 while the 1 output DNN outperforms at BER=1e-3 by 0.6 dB. This indicates that we can find the DNN topology with embedded parallelization with small penalty. When the number of output symbols of a single DNN are increased beyond 2, the presented DNN shows larger BER degradation.

The high-level synthesis (HLS) which is essentially the automatic translation of C codes to hardware description, has been gaining both capability and popularity over the years. We prepare our RTL (register transfer level) description using Vivado HLS for this work. One of the key aspects in writing hardware in C language is the throughput of the circuits we implement, as HLS can often produce the RTL with multiple clock intervals per circuit which indicate that the designed circuit is not capable of handling stream signals. We carefully pipeline all steps of the DNN computation represented in Eq. 1-3 including matrix element multiplications and summation at each layer and make sure that the throughput is 100% of the data rate to be able to process the stream signal at every clock cycle. Table 1 shows the summary of the resource utilization for 2 different DNN designs implemented in FPGA. One has 1 symbol output while the other has 2 symbol outputs per DNN. For both DNN designs, 4 copies of DNNs are implemented to increase the total throughput. The resource utilization shows small increase for the 2 outputs DNN while the throughput is doubled. All designs met the timing at 325 MHz FPGA clock with 100% throughput, and the total data rate up to 2.6 Gb/s is presented in Table 1. In order to meet with the 50Gb/s data throughput, the same DNN needs to be parallelized by 20 times. This can suggest the higher resource requirement of DNN compared to the 32-state MLSE (equivalent to 6-tap) implemented in a similar-size FPGA for 10 Gb/s [7] albeit the exact comparison is not available. Numbers of approaches including reducing the total number of neurons, pruning neuron cross-connects, having more parallel outputs per DNN need to be explored to further reduce the hardware complexity.

3. Conclusion

We have shown that the fixed point DNN based equalization can outperforms MLSE in sensitivity with expense of increase in hardware resource. The 8-bit fixed point DNNs are successfully implemented in FPGA for the analysis of the DNN complexity to equalize experimental data of 50 Gb/s PON link. The embedded parallelization of DNN is presented and is successfully demonstrated that such architecture reduces the required hardware resource with small sensitivity penalty.

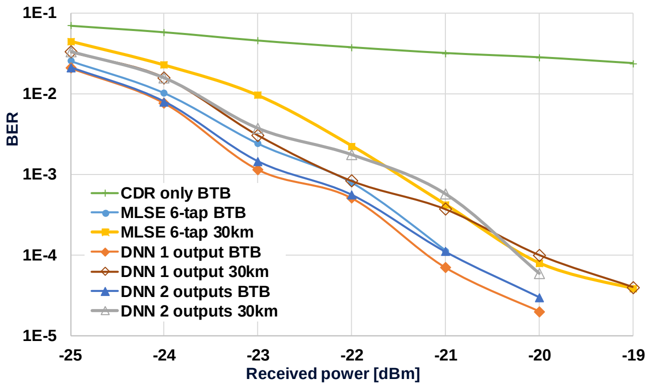


Fig. 3 Measured sensitivity of MLSE and DNN receiver equalizers for 1342nm 50Gb/s PON link.

Table. 1 Resource utilization of 2 different DNN equalizers implemented in FPGA. LUT: look-up table, RAM: random access memory, FF: Flip-flop, DSP48: 48-bit DSP unit.

DNN equalizers	11,33,14,2	11,33,14,1
Outputs per DNN	2	1
Parallel DNNs	4	4
LUT, LUTRAM	74%, 20%	74%,19%
FF, DSP48	31%,100%	29%,100%
Data throughput	2.6Gb/s	1.3Gb/s

3. References

[1] M. Tao, et al, “Improved dispersion tolerance for 50G-PON downstream transmission via receiver-side equalization,” in Proc. OFC, M2B.3 (OSA, San Diego, 2019), pp. 1-3.

[2] J. Estaran et al., “Artificial neural networks for linear and non-linear impairment mitigation in high-baudrate IM/DD systems,” in Proc. Eur. Conf. Opt. Commun. M.2.B.2. (ECOC, Düsseldorf, Germany, 2016), pp. 1-3.

[3] V. Houtsma, E. Chou, and D. van Veen, “92 and 50 Gbps TDM-PON using neural network enabled receiver equalization specialized for PON,” in Proc. OFC, M2B.6 (OSA, San Diego, 2019), pp. 1-3.

[4] M. Chagnon, J. Siirtola, T. Rissa, A. Verma, “Quantized deep neural network empowering an IM-DD link running in real-time on a field programmable gate array,” in Proc. Eur. Conf. Opt. Commun. (ECOC, Dublin, Ireland, 2019), pp. 1-3.

[5] S. Jalali, C. Nuzman, and I. Saniee, “Efficient deep learning of GMMs” in arXiv:1902.05707v1. (arXiv, 2019) pp. 1-25.

[6] L. Shu, J. Li, Z. Wan, W. Zhang, S. Fu and K. Xu, “Overestimation trap of artificial neural network: learning the rule of PRBS,” in Proc. Eur. Conf. Opt. Commun. (ECOC, Rome, Italy, 2018), pp. 1-3.

[7] I. Stamoulias, K. Georgoulakis, S. Blionas and G.O. Glentis, “FPGA Implementation of an MLSE Equalizer in 10Gb/s Optical Links,” in Proc. ICDSP (IEEE, Signapore, Singapore, 2015), pp. 794-798.