# Successive Quantization of the Neural Network Equalizers in Optical Fiber Communication

Jamal Darweesh, Nelson Costa, Yves Jaouën, Antonio Napoli, Jaoa Pedro, Bernhard Spinnler, Mansoor Yousefi

# Successive Quantization of the Neural Network Equalizers in Optical Fiber Communication


Jamal Darweesh
*Telecom Paris*
Palaiseau, France
jamal.darweesh@telecom-paris.fr

Nelson Costa
*Infinera*
Carnaxide, Portugal
NCosta@infinera.com

Yves Jaouën
*Telecom Paris*
Palaiseau, France
yves.jaouen@telecom-paris.fr

Antonio Napoli
*Infinera*
Munich, Germany
ANapoli@infinera.com

Joao Pedro
*Infinera*
Carnaxide, Portugal
JPedro@infinera.com

Bernhard Spinnler
*Infinera*
Munich, Germany
BSpinnler@infinera.com

Mansoor Yousefi
*Telecom Paris*
Palaiseau, France
yousefi@telecom-paris.fr



***Abstract*—A pragmatic successive quantization approach is applied to a neural network equalizer in a 16-QAM dual-polarization fiber transmission experiment over a 9x50km TWC fiber link. Quantization at 5 bits reduces the complexity by 85%, with a negligible Q-factor penalty.**



***Index Terms*—Optical fiber communication, nonlinearity mitigation, neural network equalization, quantization.**


## I. Introduction

The capacity of the optical fiber transmission systems is limited by the interaction between the chromatic dispersion (CD), Kerr nonlinearity and the amplified spontaneous emission noise. The advent of the coherent receiver paved the away for the compensation of the transmission effects in the electrical domain using the digital signal processing (DSP). In particular, linear transmission effects such as the CD and polarization mode dispersion (PMD) can be accurately compensated with DSP [1]. However, compensation of the nonlinear distortions is more challenging. A number of algorithms have been proposed for the nonlinearity mitigation, such as the digital back-propagation (DBP) [2]. These algorithms are usually computationally complex due to, *e.g.,* excessive application of the fast Fourier transforms.

Neural networks (NNs) have recently been considered for equalization in optical fiber communication [3], [4]. Compared to DBP, NNs do not require the fiber link parameters, and may mitigate the impairments with lower complexity [3], [5]. There are two categories of NN equalizers in optical fiber communication. In model-driven NNs, a discretization of the channel model is parameterized and learned. For example, in learned digital back-propagation [4], the parameters of the split-step Fourier method for the discretization of the nonlinear Schrödinger equation are optimized using a variant of the stochastic gradient descent (SGD). In model-agnostic NNs, a vanilla NN is used independent of the channel model [6].

To implement NNs in practice, it is desirable to reduce their complexity as much as possible. The computational complexity and memory requirements of the NNs can often be drastically reduced using the quantization and pruning with little impact on the prediction accuracy [7]. Various NN quantization schemes have been explored in the literature. In post-training quantization (PTQ), the weights and activations of the NN are quantized after training in full precision [8]. In contrast, in training-aware quantization (TAQ), quantization is integrated in the training algorithm [9]. The best known example of TAQ is the straight-through estimator [10], described in Sec. III-B. The reader is referred to [11], [12] for some of the recent developments in quantization of NNs. With the exception of a few papers [13]–[15], quantization of the NNs for equalization in optical fiber transmission has largely not been explored.

This paper applies a successive PTQ (SPTQ) approach to NNs used for the fiber nonlinearity mitigation. The main idea is to compensate for the "quantization noise" in the training. In this approach, the parameters (weights and activations) of the NN are partitioned into several sets and sequentially quantized based on a PTQ scheme (see Sec. III-B for details) [16]. In stage $i$, the parameters in the sets $k \leq i$ are quantized based on a PTQ scheme and fixed, while those in the sets $k > i$ are trained in the full precision in order to compensate for the quantization noise resulting from the previous stages. This approach is simple and tends to perform well in practice, with a good PTQ scheme and hyper-parameter optimization [16].

The paper studies the efficacy of SPTQ compared to PTQ and TAQ with the straight-through estimator (TAQ-STE), w/o mixed-precision. Quantization is applied to a NN in a dual-polarization 34.4 GBaud 16-QAM fiber transmission experiment over a 9x50km link. The NN is placed after the linear DSP to compensate for the dual-pol nonlinearities. The proposed model consists of two parallel convolutional filters for the compensation of the CD, a small hidden dense layer for the cross-pol nonlinearities and their interaction with CD, followed by an output layer with two neurons for linear regression. The performance of the several quantization algorithms is evaluated using the Q-factor as a function of the launch power and quantization rate.

The findings demonstrate that SPTQ at the average rate of

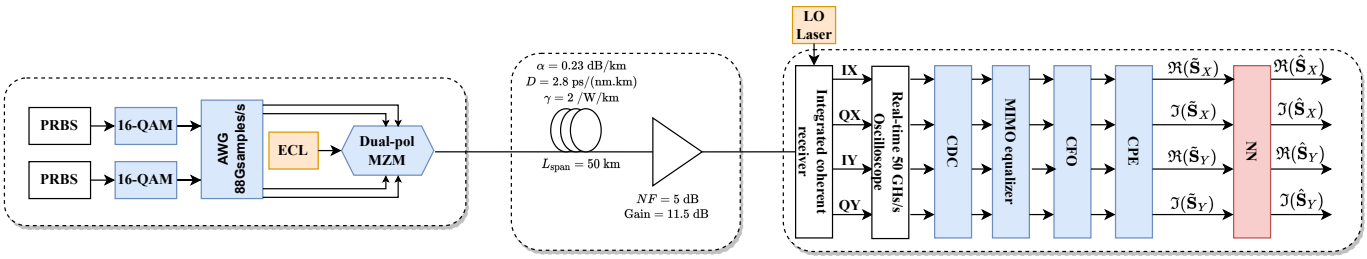


Fig. 1. The block-diagram of the 9x50 km DP-16QAM experimental transmission setup.

5 bits/weight incurs a minimal drop in the Q-factor, around 0.2 dB, while reducing the computational complexity and the required memory of the NN by 85%. Further, 5-bit SPTQ outperforms 6-bit PTQ by 1.4 dB and 6-bit TAQ-STE by 0.7 dB at 2 dBm. Compared to the previously published result [17], SPTQ improves the quantization rate by 2 bits, while simultaneously being simpler to implement than the non-uniform TAQ in [17]. The improvement is due to the incremental compensation of the quantization noise.

## II. TWC Transmission Experiment

The fiber-optic transmission experiment considered in this paper is illustrated in Fig. 1. At the transmitter (TX), two pseudo-random bit streams (PRBS) for the $x$ and $y$ polarizations are generated. They are mapped to two sequences of symbols taking values in a 16-QAM constellation. The complex-valued symbols are separated into the real and imaginary parts and fed to an arbitrary wave generator (AWG). The AWG modulates the sequence of symbols using a root raised-cosine (RRC) filter with the roll-off factor of 0.1 at 34.4 GBaud, and outputs four continuous-time electrical signals corresponding to the I and Q components of each polarization. The digital-to-analog converters (DACs) in the AWG operate at 88 Gsamples/s.

The four electrical signals are converted to optical signals and polarization multiplexed with a dual-polarization Mach-Zehnder modulator (MZM), driven by an external cavity laser (ECL) at wavelength $1.55$ $\mu m$ with the line-width 100 KHz. The optical signal is transmitted over a Truewave Classic Fiber (TWC) link, consisting of 9 spans of length 50 km in a straight line in the lab. At the end of each span, an Erbium-doped fiber amplifier (EDFA) with 5 dB noise figure is placed. The fiber has $0.23$ dB/km loss, 2.8 ps/(nm-km) CD, and 2 $(\text{Watt} \cdot \text{km})^{-1}$ nonlinearity parameter. The TWC fiber is deployed in some commercial systems. This fiber has a low dispersion and high nonlinearity coefficient, and thus operates in the nonlinear regime at high powers.

At the receiver, the optical signal undergoes polarization demultiplexing, and is transformed to four electrical signals through an integrated coherent receiver. The electrical signals are converted to the discrete-time signals by a 50-Gsamples/s oscilloscope, and up-sampled at 2 samples/symbol. The oscilloscope includes analogue-to-digital converters (ADCs) that quantize the signals at the effective number of bits of 5. The equalization is performed by the conventional dual-polarization linear DSP [1], followed by a NN. The linear DSP consists of a cascade of the frequency-domain CD compensation, multiple-input multiple-output (MIMO) equalization via the radius directed equalizer (RDE) to compensate for PMD [18], polarization separation, and the carrier-phase estimation (CPE) using the two-stage carrier phase estimation algorithm of Pfau et al. to compensate for the phase offset [19]. Lastly, the nonlinear equalization is performed by a NN, which takes the linearly-equalized symbols and mitigates dual-polarization nonlinearities, as well as the distortions introduced by the components at TX and RX.

## III. Quantization of the Neural Networks for Nonlinearity Mitigation

### A. Neural network equalizers

The architecture of the proposed NN is shown in Fig. 2. The four real-valued symbols of the $x$ and $y$ polarizations after the CPE over $T$ time steps are denoted by the vectors $\Re(\tilde{\mathbf{s}}_x)$, $\Im(\tilde{\mathbf{s}}_x)$, $\Re(\tilde{\mathbf{s}}_y)$ and $\Im(\tilde{\mathbf{s}}_y)$. The resulting array of shape $(T, 4)$ is fed to the NN. The corresponding symbols at the output of the NN are $\Re(\hat{\mathbf{s}}_x)$, $\Im(\hat{\mathbf{s}}_x)$, $\Re(\hat{\mathbf{s}}_y)$ and $\Im(\hat{\mathbf{s}}_y)$, respectively. The NN operates in a sliding-window fashion: as the vector at the input of the NN is shifted forward two steps in time, two complex symbols are produced. Thus, $T$ is arbitrary.

Due to the constrains of the practical systems, a low-complexity architecture is considered. The model consists of a cascade of three small layers. The first layer includes two parallel real-valued one-dimensional convolutional filters $\left(h_R^{(i)}\right)_{i=1}^{K}$ and $\left(h_I^{(i)}\right)_{i=1}^{K}$ of length $K = 41$ with no activation, for the compensation of CD in the symbols of the $x$ and $y$ polarizations. Each filter is convolved with each of its input vectors separately, with stride 1 and the same padding. There are total $2K = 82$ real-valued filter taps, far less than in generic convolutional layers used in the literature with large feature maps. The outputs of the convolutional filters are suitably added and subtracted in order to implement 2 complex-valued convolutions from 8 real-valued ones, resulting in four vectors.

The four outputs of the convolutional filters are concatenated in a vector and passed to a fully-connected (FC) layer with $N_D = 100$ hidden neurons, and tangent hyperbolic (tanh) activation. The FC layer processes the two polarizations jointly in order to compensate the cross-pol nonlinear interactions

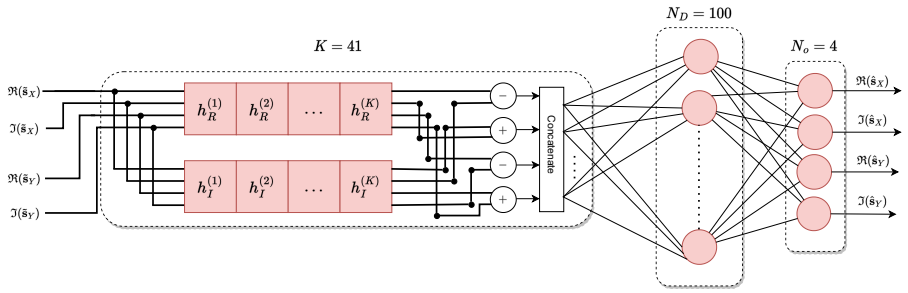


Fig. 2. The architecture of the NN. The input is the linearly-equalized symbols $\tilde{s}_x$ and $\tilde{s}_y$, and the output is the fully-equalized symbols $\hat{s}_x$ and $\hat{s}_y$. The convolutional filter taps are indicated by $h_R^{(i)}$ and $h_I^{(i)}$. The activation is $\tanh$ is the dense layer, and does not exist in the convolutional and output layer.

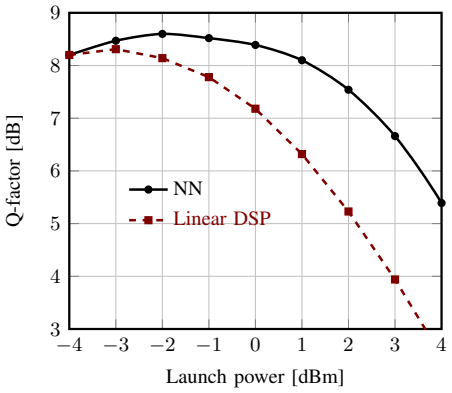


Fig. 3. Performance of the proposed NN equalizer relative to the linear DSP.

during the propagation. Finally, there is an output layer with $N_o = 4$ neurons, 2 per each polarization symbol, followed by the nearest-neighbor symbol detection.

The NN performs nonlinear regression by minimizing the mean-squared error (MSE) between its output and the expected output (*i.e.,* the transmitted symbols) in a training data set. The computational complexity of the NN, measured by the number of the floating-point (FP) real multiplications per complex symbol per polarization, is

$$\mathcal{C} = 4K + 2N_D + \left\lceil \frac{N_D N_o}{2K} \right\rceil. \tag{1}$$

Fig. 3 compares the Q-factors of the proposed (unquantized) NN and linear DSP with respect to the average power of the transmitted signal. The improvement results from the mitigation of the cross-pol nonlinearities, as well as equipment's distortions. The Q-factors of the NN is comparable to that of a DBP with 3 steps/span at 2 dBm on the same experimental data set [5]. The raw data before the linear DSP was not available to add the DBP curve to Fig. 3. The quantization approach proposed in this paper can, however, be applied to any NN equalizer.

### *B. Quantization of the neural networks*

The real numbers are usually represented in FP32 format with 32 bits, or in FP64 with 64 bits. To implement the NN in memory or computationally-constrained environments, it is desirable to represent the weights, biases, activations and the input data with fewer bits [14]. To do this, a full-precision real number $w \in \mathbb{R}$ is mapped by a quantizer $Q(.)$ to a quantized value $\hat{w} = Q(w) \in \mathcal{W}$, where $\mathcal{W}$ is the quantization codebook

$$\mathcal{W} = \{0, w^{(1)}, \cdots, w^{(N-1)}\},$$

in which $w^{(i)}$ are the quantization symbols or levels. The quantization rate or precision of $\mathcal{W}$ is $b = \log_2 N$ bits.

In uniform quantization, the quantization symbols $w^{(i)}$ are uniformly placed between a minimum and maximum value. Let $w$ be a full precision parameter anywhere in the NN, and $(a, c)$ the clipping range, *e.g.,* the smallest interval containing the unquantized parameters. The clipping range is often tuned in a process called calibration, which may use some unlabeled training data depending on the algorithm. The uniformly quantized weight is [9]:

$$\hat{w} = \left\lfloor \frac{c(w, a, c) - a}{s(a, c, N)} \right\rceil s(a, c, N) + a,$$

where $c(w, a, c) = \min(\max(w, a), c)$ is the function that clips $w$ in the interval $(a, b)$, $s(a, c, N) = (c - a)/(N - 1)$, and $\lfloor . \rceil$ denotes the nearest integer. The clipping range and bit width $b$ are hyper-parameters that are optimized. The non-uniform quantization can be defined via similar relations [11].

PTQ begins with training the model in FP32, and quantizes the resulting weights, activations and the input tensor [20]. This reduces the computational complexity and storage for the inference phase [20]. This technique has low overhead, and is advantageous in applications where the training data for calibration is unavailable. However, PTQ below 8 bits can lead to a substantial reduction in the accuracy [21].

In TAQ, the quantization and training algorithms are simultaneously developed. This technique usually enhances the prediction accuracy of the model by accounting for the quantization noise during the training. However, learning via the backpropagation of errors in SGD is not possible directly, since the quantizer is a piece-wise flat function with zero derivative almost everywhere. The straight-through estimator is an empirical method that addresses the problem of the zero gradient by modifying the chain rule for differentiation in SGD to ensure a non-zero approximate gradient [10]. The most

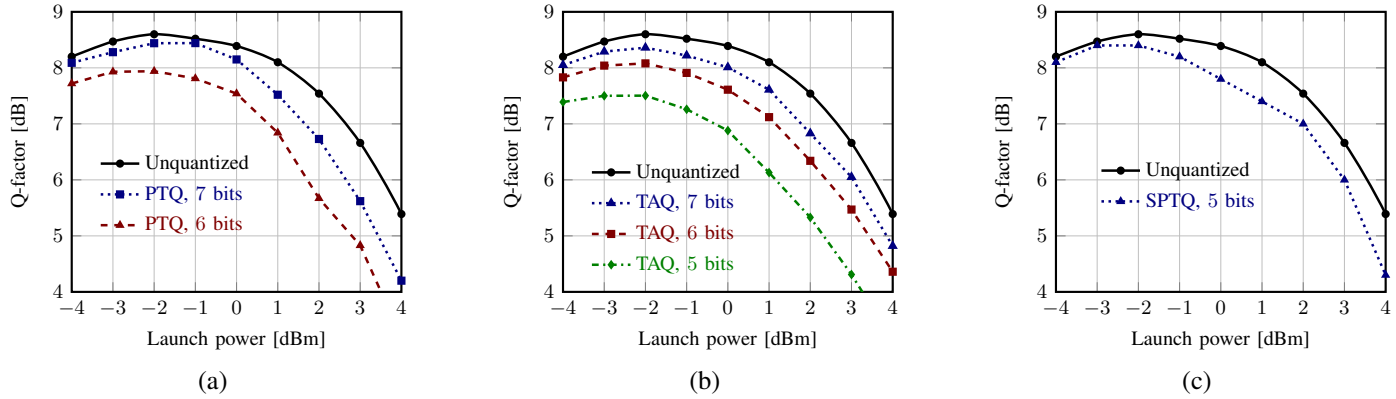


Fig. 4. The Q-factor versus launch power at several quantization rates, for (a) PTQ, (b) TAQ, and (c) SPTQ at 5 bits.

widely used surrogate for the gradient is the identity function, in which $d\hat{w}/dw \triangleq 1$ [22]. Even though one is not a good approximation of zero, STE works surprisingly well in some models. TAQ typically provides higher prediction accuracy than PTQ when quantizing at low number of bits, at the cost of increased computational and implementation complexity. On the other hand, if the approximation technique is not carefully chosen, TAQ may perform even worse than PTQ [23].

SPTQ is a combination of PTQ and TAQ, without the complexity of TAQ, or having to address the zero gradient problem [16]. At stage $i$, the set of weights in the layer $\ell$ distinguished by an index set $\mathcal{P}_i^{(\ell)}$ is partitioned into two subsets $\mathcal{P}_{i,1}^{(\ell)}$ and $\mathcal{P}_{i,2}^{(\ell)}$ corresponding to the quantized and unquantized weights respectively, *i.e.,*

$$\mathcal{P}_i^{(\ell)} = \left\{\mathcal{P}_{i,1}^{(\ell)}, \mathcal{P}_{i,2}^{(\ell)}\right\}, \quad \mathcal{P}_{i,1}^{(\ell)} \cap \mathcal{P}_{i,2}^{(\ell)} = \emptyset.$$

The corresponding weights are denoted by $W_i^{(\ell)} \in \mathcal{P}_i^{(\ell)}$, $W_{i,1}^{(\ell)} \in \mathcal{P}_{i,1}^{(\ell)}$ and $W_{i,2}^{(\ell)} \in \mathcal{P}_{i,2}^{(\ell)}$. The model is first trained over $W_i^{(\ell)}$ in FP32. Then, the resulting weights $W_{i,1}^{(\ell)}$ are quantized under a suitable PTQ scheme. Next, $W_{i,1}^{(\ell)}$ is fixed, and the model is retrained by minimizing the loss function with respect to $W_{i,2}^{(\ell)}$, starting from the previously trained values. The second group is retrained in order to compensate the quantization noise in the first group, and make up for the loss in accuracy. In stage $i+1$, the above steps are repeated upon substitution $\mathcal{P}_{i+1}^{(\ell)} \triangleq \mathcal{P}_{i,2}^{(\ell)}$. The weight partitioning, group-wise quantization, and retraining is repeated until the network is fully quantized.

In another version of this algorithm, the partitioning for all stages is set initially. That is to say, the weights are partitioned into a number of groups and successively quantized, such that at each stage the weights of the previous groups are quantized and fixed, and those of the remaining groups are retrained.

The hyper-parameters of the SPTQ are the choice of the quantizer function in PTQ and the partitioning scheme. There are several choices for the partitioning scheme, such as random grouping, neuron grouping and local grouping. Research has demonstrated that models trained with SPTQ provide classification accuracies comparable to their baseline counterparts trained and deployed in 32-bit, with fewer bits [16].

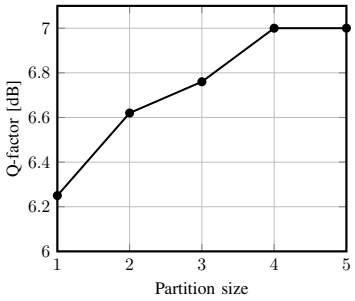


Fig. 5. The Q-factor versus the partition size in SPTQ.

In the mixed-precision quantization, different layers, feature maps, channels, weight groups or activations are quantized generally at different rates, depending on the sensitivity of the loss function. In the NN in Sec. III-A, the convolutional and dense layers are quantized at $b_1$ and $b_2 \leq b_1$ bits, respectively.

## IV. Gains of Quantization

The dataset required for training the NN is obtained from the TWC transmission experiment described in Sec. II. The training set contains 600,000 symbols from a 16-QAM constellation. A test set of 100,000 symbols is used to assess the performance of the NN. Each dataset is measured at a given power, during which the BER may fluctuate in time due to the environmental changes. The symbols on the boundary of the data frame are eliminated to remove the effects of anomalies. The NN at each power is trained and evaluated with independent datasets of randomly chosen symbols at the same power.

The NN model described in Sec. III-A is considered for nonlinearity mitigation. The hyper-parameters of this model are the size of the convolutional filters $K$ and the number of hidden neurons $N_D$. The filters' length is determined by the channel memory measured in the number of symbols due to the residual dispersion left after the CD compensation. This is estimated to be 40 symbols, through the correlation

function of the received symbols after CPE, or performance evaluation. The minimum number of hidden units is 100, below which the performance rapidly drops. The NN is built, trained and evaluated in the Python's TensorFlow library. The loss function is the mean-squared error, and the learning algorithm is the Adam-Optimizer with the learning rate of 0.001. The quantization is implemented in the open-source library Larq in Python.

Three quantization algorithms are applied and compared. First, PTQ is performed on the unquantized FP-trained model, where all layers are quantized at 6 or 7 bits. Then, QAT is implemented, where the weights of all layers are randomly initialized and subsequently quantized with STE at 6 or 7 bits. QAT can also be performed starting with the weights quantized by PTQ. The resulting PTQ-QAT does not improve much upon TAQ, and is more complex. Finally, the SPTQ described in Sec. III-B is applied, by assigning a bit-width of 5 for both weights and activations of the dense layer uniformly. The convolutional layer is given 8 bits, but in our model this layer has few weights, and little impact on the complexity; see (1).

Fig. 4(a) shows the Q-factor of PTQ with $b_1 = b_2$. PTQ implemented uniformly with 6 bits leads to a Q-factor drop of 0.7 dB at -2 dBm, and 1.9 dB at 2 dBm. The results demonstrate that the performance degradation caused by the quantization grows weakly as the transmission power increases. As depicted in Fig. 4(b), TAQ-STE improves upon PTQ by reducing the Q-factor penalty to 0.5 dB at -2 dBm, and to 1.2 dB at 2 dBm, at 6 bits. SPTQ attains the best results, with a Q-factor drop of 0.2 dB at -2 dBm, and 0.5 dB at 2 dBm when quantizing at as few as 5 bits. It can be seen in Fig. 4(c) that SPTQ is generally subject to a smaller Q-factor penalty across the whole range of power, at even a lower bit-width, than PTQ and TAQ-STE. The performance rapidly drops below the threshold value of $b_1 = b_2 = 5$ bits. Compared to our previously published result [17], uniform SPTQ outperforms a more complex non-uniform TAQ by 2 bits at the same average signal power, in this experiment.

The impact of the partition size in SPTQ is depicted in Fig. 5. By increasing the number of partitions in the dense layer, the Q-factor is enhanced. This is because a larger partition size reduces the number of the quantized weights at any given stage. A plateau in performance is observed after a certain partition size. We have observed that, as the transmission power increases, the nonlinear effects grow, making the task more challenging for the NN, and hence, requiring more partitions to maintain a good performance.

## V. Conclusions

The paper compares post-training, a proposed successive post-training, and training-aware quantization with the straight-through estimator, in application to a NN used for the nonlinearity mitigation in a 16-QAM dual-polarization TWC fiber transmission experiment. The Q-factor of these quantization algorithms are compared at several launch powers and quantization rates. Successive post-training quantization at 5 bits lowers the complexity by 85% with negligible Q-factor penalty, outperforming the more complex alternatives by 2 bits at the same launch power.

## VI. Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant Agreement No. 813144, as well as from the European Research Council (ERC) under the Grant Agreement No. 805195.

## References

[1] S. J. Savory, "Digital coherent optical receivers: Algorithms and subsystems," *IEEE J. Sel. Top. Quantum Electron.*, vol. 16, no. 5, pp. 1164–1179, May 2010.

[2] E. Ip and J. M. Kahn, "Compensation of dispersion and nonlinear impairments using digital backpropagation," *IEEE J. Lightw. Technol.*, vol. 26, no. 20, pp. 3416–3425, Oct 2008.

[3] S. Zhang, F. Yaman, E. Mateo, and Y. Inada, "Neuron-network-based nonlinearity compensation algorithm," in *Proc. Eur. Conf. Opt. Commun.*, Roma, Italy, 2018, pp. 1–3.

[4] R. M. Butler, C. Hager, H. D. Pfister, G. Liga, and A. Alvarado, "Model-based machine learning for joint digital backpropagation and pmd compensation," *IEEE J. Lightw. Technol.*, vol. 39, no. 4, pp. 949–959, Feb. 2021.

[5] P. J. Freire, Y. Osadchuk, B. Spinnler, A. Napoli, W. Schairer, N. Costa, J. E. Prilepsky, and S. K. Turitsyn, "Performance versus complexity study of neural network equalizers in coherent optical systems," *IEEE J. Lightw. Technol.*, vol. 39, no. 19, pp. 6085–6096, Jul. 2021.

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

[7] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *J. Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, Jan. 2017.

[8] R. Banner, Y. Nahshan, E. Hoffer, and D. Soudry, "Post-training 4-bit quantization of convolution networks for rapid-deployment," in *Conf. Neural Info. Proc. Sys.*, Dec. 2019, pp. 1–9.

[9] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *IEEE Conf. Comp. Vision Pattern Recognition*, 2018, pp. 2704–2713.

[10] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin, "Understanding straight-through estimator in training activation quantized neural nets," in *The Int. Conf. Learning Rep.*, May 2019, pp. 1–30.

[11] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. Van Baalen, and T. Blankevoort, "A white paper on neural network quantization," *arXiv:2106.08295*, Jun. 2021.

[12] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Low-Power Computer Vision*, 1st ed., G. K. Thiruvathukal, Y.-H. Lu, J. Kim, Y. Chen, and B. Chen, Eds. Boca Raton, Florida, USA: CRC Press, 2022, ch. 13, pp. 291–325.

[13] T. Koike-Akino, Y. Wang, K. Kojima, K. Parsons, and T. Yoshida, "Zero-multiplier sparse dnn equalization for fiber-optic qam systems with probabilistic amplitude shaping," in *Proc. Eur. Conf. Opt. Commun.*, 2021, pp. 1–4.

[14] P. He, F. Wu, M. Yang, A. Yang, P. Guo, Y. Qiao, and X. Xin, "A fiber nonlinearity compensation scheme with complex-valued dimension-reduced neural network," *IEEE Photon. J.*, vol. 13, no. 6, pp. 1–7, 2021.

[15] D. A. Ron, P. Freire, J. Prilepsky, M. Kamalian-Kopae, A. Napoli, and S. Turitsyn, "Experimental implementation of a neural network optical channel equalizer in restricted hardware using pruning and quantization," *Sci. Reports*, vol. 12, no. 1, May 2022.

[16] A. Zhou, A. Yao, Y. Guo, L. Xu, and Y. Chen, "Incremental network quantization: Towards lossless CNNs with low-precision weights," in *The Int. Conf. Learning Rep.*, Apr. 2017, pp. 1–24.

[17] J. Darweesh, N. Costa, A. Napoli, B. Spinnler, Y. Jaouën, and M. Yousefi, "Few-bit quantization of neural networks for nonlinearity mitigation in a fiber transmission experiment," in *Proc. Eur. Conf. Opt. Commun.*, 2022, p. We4C.4.

[18] I. Fatadin, D. Ives, and S. J. Savory, "Blind equalization and carrier phase recovery in a 16-QAM optical coherent system," *IEEE J. Lightw. Technol.*, vol. 27, no. 15, pp. 3042–3049, May 2009.

[19] T. Pfau and R. Noé, "Phase-noise-tolerant two-stage carrier recovery concept for higher order qam formats," *IEEE J. Selec. Topics Quantum Electron.*, vol. 16, no. 5, pp. 1210–1216, 2009.

[20] Y. Choukroun, E. Kravchik, F. Yang, and P. Kisilev, "Low-bit quantization of neural networks for efficient inference," in *IEEE/CVF Int. Conf. Comp. Vision Workshop*, Oct. 2019, pp. 3009–3018.

[21] I. Hubara, Y. Nahshan, Y. Hanani, R. Banner, and D. Soudry, "Improving post training neural quantization: Layer-wise calibration and integer programming," in *The Int. Conf. Learning Rep.*, May 2021, pp. 1–15.

[22] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv:1308.3432*, pp. 1–12, Aug. 2013.

[23] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K.-T. Cheng, "Bi-real net: Enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm," in *Euro Conf. Comp. Vision*, Sep. 2018, pp. 722–737.