Recovery of signal distorted by nonlinearity in optical communications using deep learning

Alexander Blagodarnyi [1] **Alexey Larionov** [1] **Ilya Kuk** [1] **Mohammed Deifallah** [1] **Razan Dibo** [1] **Sergei Gostilovich** [1] **Stanislav Krikunov** [1]

Abstract

The rapid development of the global information infrastructure and the massive involvement of digital technologies into human activities are accompanied by growing requirements for the bandwidth of telecommunication networks. The research of recent years has proven usefulness of Machine Learning and Deep Learning for improving performance and robustness of transmission over different mediums. In this work we aim to revise one of the latest papers in signal processing for nonlinear optical communications through machine learning. We experiment with already proposed ideas, as well as our own, in order to understand the limitations or advantages of such methods.

Github repo: github
Video presentation: video presentation

1. Introduction

Optical communications are the backbone of all forms of information technology infrastructure in our modern society. As global Internet traffic grows by 60% per year1, research breakthroughs in optical communication speeds are much needed to meet the connectivity demands in the future. Due to the large bandwidth of fiber, fiber-optic telecommunications remains a major technology capable of tackling these difficulties. In the late 1990s and early 2000s, a technological breakthrough in fiber-optic telecommunications led to the development of coherent optical telecommunication systems, allowing optical systems to be improved by an order of magnitude. Coherent data transmission is a key technology for high-speed data transfer at the moment. However, there are major limits related with the nonlinearity of the optical fiber on the way to its further development. The nonlinearity of the optical fiber causes a distortion in the pulse phase, which is the parameter responsible for information transmission. As the transmission length or the performance of the communication line increases, non-linear limitations arise. One of the major challenges in high-speed telecommunications is expanding the use of coherent transmission in non-secure systems. Currently, there are various approaches to its solution, in particular, digital back propagation (DBP) based on computer simulation. In fiber-optic communications, inverting the nonlinear Schrödinger equation in real time to reverse the deterministic effects of the channel is a critical problem. Interestingly, the split-step Fourier method (SSFM) produces a computation graph that looks like a deep neural network. This finding enables the use of machine learning methods to reduce complexity. In this project we suggest to use deep learning in conjunction with DBP algorithms to attempt to recover a signal that has undergone severe distortions caused by the nonlinearity of an optical fiber. The problem is of great importance in improving the performance of optical communication transmission systems, this can be useful for both increasing the bitrate as well as increasing the length of transmission lines.

2. Related Work

Coherent data transmission is currently a primary technology for high-speed data transmission [1]. When it comes to furthering its development and improving its results, however, it faces numerous obstacles. Fiber Kerr nonlinearity-induced that causes pulse phase distortion, interactions between fiber nonlinearity-induced self-phase modulation (SPM), chromatic dispersion (CD), and inline optical amplifier noise (amplified spontaneous emission (ASE) noise) are all examples of these issues [2]. These interactions are much too complex to be directly defined. There are several solutions to the problem of compensating for nonlinearity in optical fibers currently available. These methods almost used The (DBP) algorithm that calculates the transmitted signal from the received signal by solving the inverse nonlinear Schrödinger equation of a fiber optic line using the step separation Fourier method (SSFM).In other words, it

[1] Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Ilya Kuk <Ilya.Kuk@skoltech.ru>.

Final Projects of the Deep Learning 2021 Course, Skoltech, Moscow, Russian Federation, 2021.

requires transmitting the received signal through a fictitious fiber with inverse parameters. The fiber link is divided into several steps with small distance, and at each step, it is modeled as a concatenation of linear and nonlinear sections [3] [4]. In article [5], Machine learning techniques were combined with the digital back propagation (DBP) algorithm by the authors. It was proposed to use a multilayer fully-connected (dense) deep neural network (DNN). According to the authors, such a combination will optimize network parameters, resulting in improved performance and computational savings. The receiver digital signal processing does not exactly invert the linear and nonlinear steps of the fiber propagation effects with this method, according to the researchers. It should instead strike a balance between compensating for transmission impairments and suppressing the additional distortions caused by inline amplifier noise's imperfect linear and nonlinear compensation steps. According to [6], When the effect of signal spectral broadening is taken into account, the performance of multi-channel digital back-propagation (MC-DBP) for compensating fibre nonlinearities in Nyquist-spaced optical communication systems is investigated. It is discovered that, regardless of modulation formats used, accounting for the spectral broadening effect is critical for achieving the best output of DBP in single-channel and multi-channel communication systems. The loss of DBP output in multi-channel systems due to ignoring the spectral broadening effect in the compensation is more noticeable for outer channels. In article [7], The authors investigate the performance of a machine learning classification technique called the Parzen window in the sense of dispersion managed and dispersion unmanaged systems to minimize fiber nonlinearity. The technique is applied for detection at the receiver side and deals with the non-Gaussian nonlinear effects by designing improved decision boundaries. They also suggest a two-stage mitigation technique for dispersion unmanaged systems that uses digital back propagation and the Parzen window. Digital back propagation compensates for the deterministic nonlinearity in this situation, while the Parzen window deals with the stochastic nonlinear signal-noise interactions that digital back propagation misses. In [8], Machine learning-based nonlinear equalization in long-haul transmission systems is discussed. As compared to traditional digital back propagation approaches, the authors show that dynamic multi-perceptron networks can cope with the memory properties of the fiber channel and provide efficient mitigation of nonlinear impairments at a lower computational cost. One of the guiding principles for effective compensation of fiber nonlinearity appears to be: fewer steps are better and more efficient. The authors of the article [9] challenge this assumption, demonstrating that carefully planned multi-step methods can achieve better performance–complexity trade-offs than their single-step counterparts. In the article [10], as in [5], the method is also based on DL DBP, but has 6 times less computational complexity than the usual one, to achieve the same level of gain. The DL-DBP uses static hidden layers (which do not change during training) so that it can be trained at the symbol level. And after training these static layers are replaced with the usual digital signal processing layers performing chromatic dispersion compensation, clock recovery, polarization rotation compensation, and phase estimation. This way the algorithm is blind not only to polarization rotation state, frequency offset and phase offset but also to timing error. In addition, DL-DBP uses a low-pass filter in its learned buried layers to reduce out-of-band modulation due to nonlinearity compensation.

3. Method Description

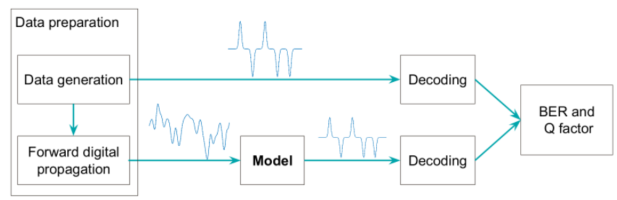

Figure 1. Workflow of the project

3.1. Data generation

To get necessary dataset we created the data generetor. It can produce bit-sequence which represents data launched at the front end of the communication system. The bit-sequence is represented by periodic train of Gaussian pulses with Differential Phase Shift Keying (DPSK). The generator formula is the following:

$$E(0,t) = \sum_{k=1}^{N} a_k \pi^{-1/4} e^{-\frac{1}{2}(t-kT^2)}, \qquad (1)$$

where $a_k = 1$ or -1 with probability $p = 0.5$.

The output of this procedure look like a signal on Fig. 14 with blue line.

3.2. Forward digital propagation

Let $E(z,t)$ at $z = 0$ be the electric field of a signal at the transmitter. In the simplest form, signal propagation in optical fibers is described by the stochastic scalar nonlinear Schrodinger equation (NLSE)

$$\frac{\partial E(z,t)}{\partial z} = \underbrace{\left(-\frac{1}{2}\alpha - j\beta_2 \frac{1}{2}\frac{\partial^2}{\partial t^2}\right)}_{D} E(z,t) +$$

$$\underbrace{+j\gamma\left|E\left(z,t\right)\right|^{2}}_{N} E\left(z,t\right) \;=\; (D+N)E(z,t), (2)$$

In this formulation, D and N are the linear and nonlinear operators, respectively, and α, β_2, and γ denote attenuation, group velocity dispersion, and fiber nonlinearity coefficient.

3.3. Data preparation

A generated electric field of a bit sequence can be represented in two ways:

- 1D time data - the electric field is taken as is (simple time series)
- 2D time data - regrouping time in 2D blocks in accordance to the original paper [5] (Fig. 2). The samples of signal are split into blocks, and sliding window over these blocks forms a 2D representation of the sequence. Since each sliding window contains blocks to the left and to the right of a subsequence, we need to compute losses only over the main (non-intersecting) part of 2D data.

For training we pregenerated 6 variants of datasets with different coefficient of nonlinearity and with/without dispersion compensation. In the end we used only two datasets with a mixture of nonlinearities. Our repository also allows to use simulation data, i.e. every training batch will be unique and it's generated as the model trains. However since it takes about 4 minutes to generate 1000 sequences (which is a typical size of an epoch), we opted to only train on the pregenerated datasets.

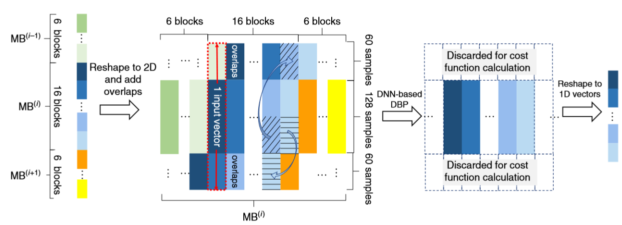

Figure 2. Data preparation, as proposed in [5]

3.4. Metric Estimation

LOSS FUNCTIONS: ERROR VECTOR MAGNITUDE (EVM)

For training, we initially chose the error vector magnitude (EVM) as a loss function. EVM is defined as the mean of

$$\frac{|\hat{E}(0,t) - E(0,t)|^2}{|E(0,t)|^2.} \qquad (3)$$

EVM is closely related to mean-squared error (MSE). It should be noted that the signal-to-noise ratio (SNR) is essentially proportional to 1/EVM or 1/MSE and can be used as loss function in principle. We use EVM/MSE as the cost function as they are common in both optical communications and machine-learning community [5]. During training our models we encountered difficulties using EVM loss, so finally we decided to use MSE as a train loss.

EVALUATION: BIT ERROR RATE (BER) AND Q-FACTOR

For model evaluation during testing we decided to use standard metrics for evaluation in single-channel single-polarization systems, such as bit error ratio (BER) or the corresponding quality (Q) factor. BER is the ratio of correct decoded bites to all number of bites. The BER formula is the following:

$$BER = \frac{N_{error}}{N_{total}} \qquad (4)$$

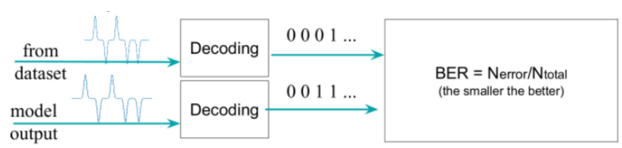

Figure 3. metric estimation

4. Models

4.1. Convnet

We can see the architecture of Convnet that we use in 4.a

4.2. Fully connected model

Fully connected network (FC) architectures are commonly used for solving different kinds of tasks. The main advantage of FC networks is the fact that FC offers learns features from all the combinations of the features of the previous layer, while convolutional networks rely on local spatial coherence with a small receptive field. However, FC networks are computationally expensive, so we are limited in the number of layers. We implemented a universal FC network architecture, in which we can easily vary the number of layers and their shapes with the config file. Also, the model can be considered as a set of elementary blocks 4.b which can also be tuned. Using the configuration file, we can tune the type of activation function, Batch Normalization, and dropout

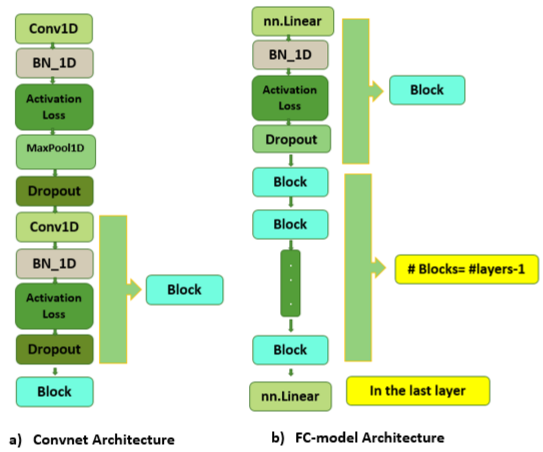

Figure 4. Models architectures

However, the input vector is complex, and since torch cannot train models with complex values, they can be processed differently. We proposed two types of FC models: with parallel real and imaginary part processing in independent layers and with the concatenation of both parts into a single input vector. The second case can take into account the fact that real and imaginary parts can be dependent (in terms of power), but it is more expensive in terms of memory consumption. That is why we also proposed to train both parts of complex vectors in independent layers.

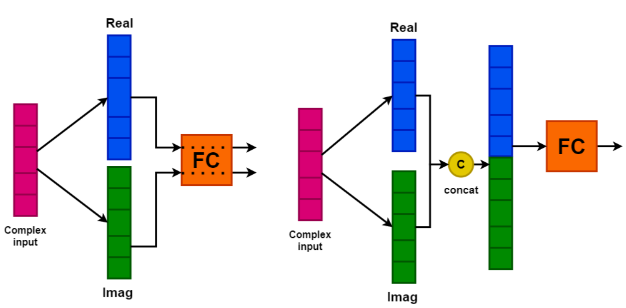

Figure 5. Two types of FC networks with different complex processing. Left - FC with independent layers. Right - FC with real and imaginary concatenation into a single vector

5. Experimental Results

5.1. FC-model with concatenation of imaginary and real parts

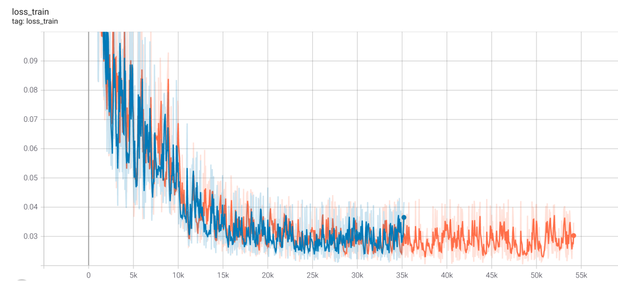

Figure 6. Train loss. Blue line with dispersion compensation, orange line - without

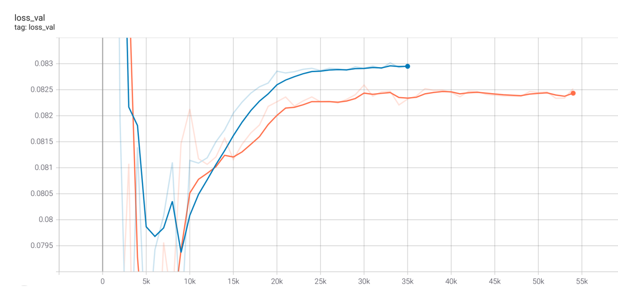

Figure 7. Validation loss. Blue line with dispersion compensation, orange line - without

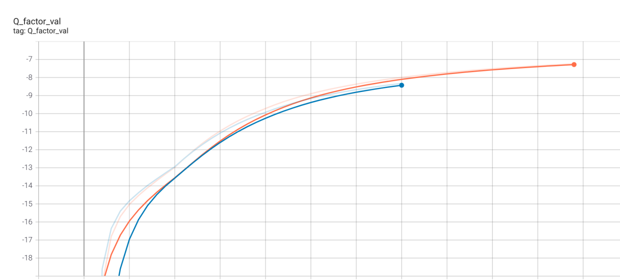

Figure 8. Validation Q factor. Blue line with dispersion compensation, orange line - without

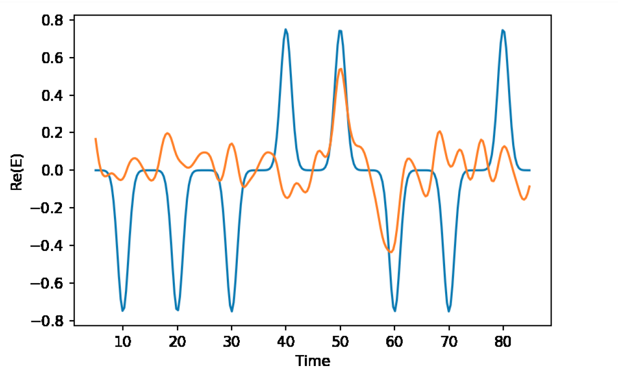

Figure 9. Signals example, experiment without dispersion compensation. Blue line - target, Orange line - predicted

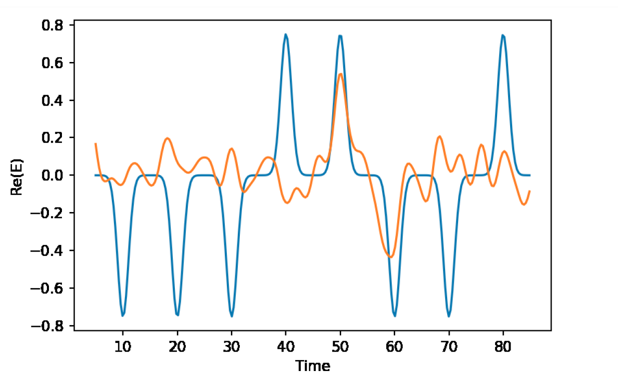

Figure 10. Signals example, experiment with dispersion compensation. Blue line - target, Orange line - predicted

5.2. FC-model with real and imaginary data parts being processed by the same fully-connected network

This experiment turned out to perform worse on average than the previous, which is explainable, as having two separate networks for real and imaginary perts seems more logical (the parts may carry different statistacs, which cannot be learned by a single model for two parts).

In search for better hyperparameters of this model we performed 3 experiments:

- 3 layers, 4096 each, dropout 0.3, LeakyReLU in between layers, Cosine annealing scheduling starting from learning rate 0.01 with 10 epochs cycle, and AdamW optimizer with small weight decay 0.0001.

  The model quickly saturates in terms of validation Q-metric, yet we can see gradual improvement, which was much worthy as Q-score of minus 13 is very poor, loss however finds occasional rapid downfall.

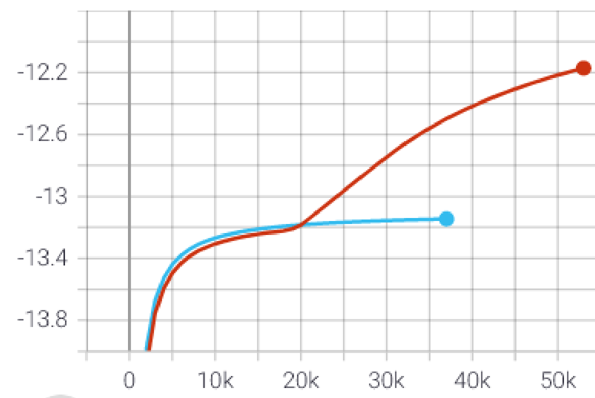

Figure 11. Validation Q factor over training steps. Blue line - dataset with dispersion compensation, Orange line - w/o

- Same as previous, but 4 layers, and hidden layers are of size 1024, instead of 4096. The training shows a bit more steady training loss decreasing and just a tiny bit bigger Q-score, yet predictions are still meaningless

- A baseline (and surprisingly the best) setup from the reference paper [5] of 3 layers $\times 4096$, dropout 0.3 and Adam optimizer with StepLR learning rate scheduling (LR decreases by 10 every 10 epochs). With this model we see a much better improvement in terms of Q-score (well above -13 and rising), loss is decreasing but it meets certain plateaus. The predictions hower are still not impressing, yet the certain peaks start to resemble truth.

  With this best experiment we also performed a training when the dataset is generated with dispersion compensation (visually it makes input signal less distorted and much more organized), however as we can see and as other research suggests, enabling this option confuses our model quite significantly, in can't learn anything, the same as the previous experiments.

6. Conclusion and future work

In this work we've reimplemented the reference paper [5] and tested this approach with some of our ideas, e.g. usage of simulated dataset (possibly infinite), rather than real-world one, and also addition of samples of different nonlinearities in a single dataset, compared to a single nonlinearity. We've found that most of them only harm the performance of the models, sometime even unexpectedly (like with dispersion compensation in the previous section).

For the future we would like to finally discover working improvements for the method from [5]. This can be for example from application of other types of Deep Learning approaches (like RNNs, transformers) or by refining our dataset generation procedure, so that it will greatly support

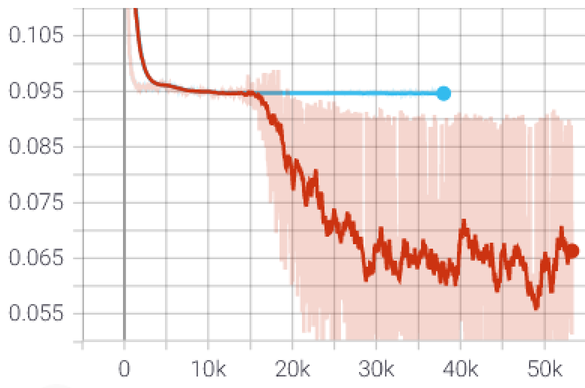

Figure 12. Training loss over training steps (smoothed). Blue line - dataset with dispersion compensation, Orange line - w/o

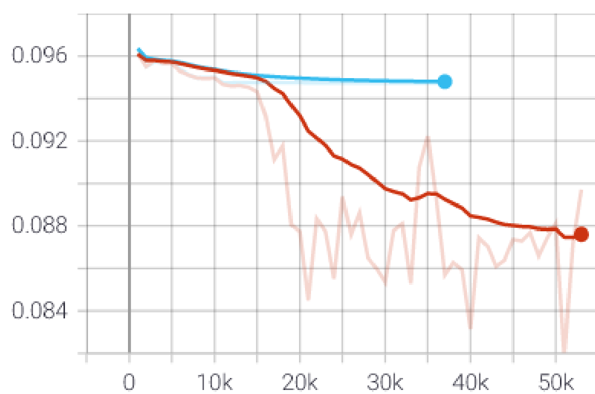

Figure 13. Validation loss over training steps. Blue line - dataset with dispersion compensation, Orange line - w/o

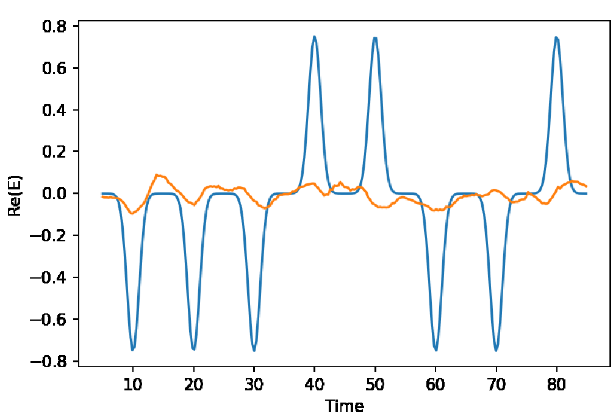

Figure 14. Example of prediction. Blue line - target, Orange line - predicted

learning progress (e.g. gradually increasing difficulty of input distoted signal).

7. References

[1] Kikuchi, K. (2008, September). Coherent transmission systems. In 2008 34th European Conference on Optical Communication (pp. 1-39). IEEE.

[2] Kikuchi, K. (2015). Fundamentals of coherent optical fiber communications. Journal of Lightwave Technology, 34(1), 157-179.

[3] Asif, R., Lin, C. Y., Schmauss, B. (2011). Digital backward propagation: a technique to compensate fiber dispersion and non-linear impairments. In Applications of Digital Signal Processing. IntechOpen.

[4] Amari, A., Dobre, O. A., Venkatesan, R., Kumar, O. S., Ciblat, P., Jaouën, Y. (2017). A survey on fiber nonlinearity compensation for 400 Gb/s and beyond optical communication systems. IEEE Communications Surveys Tutorials, 19(4), 3097-3113.

[5] Fan, Q., Zhou, G., Gui, T., Lu, C., Lau, A. P. T. (2020). Advancing theoretical understanding and practical performance of signal processing for nonlinear optical communications through machine learning. Nature Communications, 11(1), 1-11.

[6] Xu, T., Karanov, B., Shevchenko, N. A., Lavery, D., Liga, G., Killey, R. I., Bayvel, P. (2017). Digital nonlinearity compensation in high-capacity optical communication systems considering signal spectral broadening effect. Scientific reports, 7(1), 1-10.

[7] A. Amari, X. Lin, O. A. Dobre, R. Venkatesan and A. Alvarado, "A Machine Learning-Based Detection Technique for Optical Fiber Nonlinearity Mitigation," in IEEE Photonics Technology Letters, vol. 31, no. 8, pp. 627-630, 15 April15, 2019, doi: 10.1109/LPT.2019.2902973.

[8] Sygletos, S., Redyuk, A., Sidelnikov, O. (2019, July). Nonlinearity compensation techniques using machine learning. In Signal Processing in Photonic Communications (pp. SpT2E-2). Optical Society of America.

[9] C. Häger, H. D. Pfister, R. M. Bütler, G. Liga and A. Alvarado, "Revisiting multi-step nonlinearity compensation with machine learning," 45th European Conference on Optical Communication (ECOC 2019), 2019, pp. 1-4, doi: 10.1049/cp.2019.0947.

[10] Ghazisaeidi, A., Eppenberger, M., Baeurle, B., Ayata, M., Leuthold, J. (2020). Deep learning based digital backpropagation demonstrating SNR gain at low complexity in a 1200 km transmission link. Optics Express, 28(20), 29318-29334.

A. Team member's contributions

Mohammed Deifallah:

- Optimization of Data Generation and Transformation.
- Q-Factor (performance metric) implementation.
- Implementation of CNN model using 1D convolutional layers.

Alexey Larionov:

- Creating of GitHub repository, README file, sole review of all the pull requests and the fixes they required.
- Implementation of the whole project structure, training pipeline using PyTorch Lightning, including starting a training from YAML config files, saving of checkpoints, easy Jupyter Notebook for more advanced launch of training
- Pregenerating 6 variants of datasets of different nonlinearities
- Training and collecting results of FC model
- Filling in parts of this report

Ilya Kuk:

- Proposing a project idea, holding a seminar with a detailed explanation of the problem, proposing an idea for the implementation of neural network models.
- At all stages, advising team members on the implementation of the code.
- Implementation of the original code for generating data using the split-step method
- Fc-model with concatenation training
- Partial work on report

Razan Dibo:

- Prepare the template of the report.
- Responsible for Introduction, related work, Models architecture figures, references in the report.
- search for alternative models and propose CNN+biLSTMP: a CNN model using 1D convolutional layers and a biLSTMP layer.

Alexander Blagodarnyi:

- Video presentation preparation.
- Both fully connected linear models preparation and testing.
- A part of introduction and literature review preparation.

Stanislav Krikunov:

- Video presentation preparation.
- Help with fully connected linear models preparation and testing.
- A part of introduction and literature review preparation.

Sergei Gostilovich:

- Make on final presentation
- Design code for BER estimation
- Work on Section 3 of report. (Method description, mainy on parts: 3.1, 3.2, 3.4)