

Machine learning aided DSP in optical coherent receivers

L.H.P. Driessen

l.h.p.driessen@student.tue.nl, ID 0962549

Abstract—A recurrent neural network has been used as a substitution to parts of the DSP chain in optical coherent receivers. Results with simulated data show that this technique not yet comparable to conventional filtering techniques. An accuracy of 98.5952% (BER of $1.408 \cdot 10^{-2}$) has been achieved in simulated QPSK data with an SNR of 15 and a transmission differential group delay of 40ps. In real, frequency-offset compensated, 8-QAM data, an accuracy of 83.2145% (BER of $1,67855 \cdot 10^{-1}$) has been achieved.

I. INTRODUCTION

FIBER-OPTIC communication has replaced lots of the electrical communication in the last decades due to its lower interference and higher bandwidth. Conventionally, direct-detection with on-off keying was used in optical communication. However, it is possible to achieve a greater bandwidth by using a coherent communication system [1], [2]. Latest developments in coherent optical transmission have shown data communication rates in the Tb/s range [1]–[3].

Coherent optical fiber communication was studied in the 1980s. The main advantage over direct-detection was the ability to transmit data over longer distances [1]. In the 1990s and the beginning of the 2000s, research on coherent detection decreased due to the advent of wavelength division multiplexing (WDM). Research on coherent systems continued in 2005, which showed promising results [4]. The main advantage of coherent systems is that it benefits not only from light intensity, but also utilizes information such as light polarization and phase. This makes it suitable to deploy higher order modulations like QPSK and QAM, which results in higher data rates. These modulations come at the cost of higher complexity. It requires extra hardware to convert the signal to the electrical domain, as detection of polarization and phase of light is not feasible to do in the optical domain [1]. Digital signal processing (DSP) is used to recover the polarization, intensity and phase of the signal. Also, signal impairments due to channel imperfections can be compensated by DSP [1], [2], [5]. Some of these impairments, like polarization mode dispersion (PMD), are stochastic [6] and have so far been compensated by linear filters.

A different approach to recover received optical signals might be to aid the DSP with machine learning methods. This is because machine learning techniques have proven to be useful in solving complex statistical problems. Different parts in the receiver could be substituted by these techniques to enhance the DSP performance. In this paper, the proof of concept of machine learning aided DSP in optical coherent receivers is provided and its feasibility is analyzed.

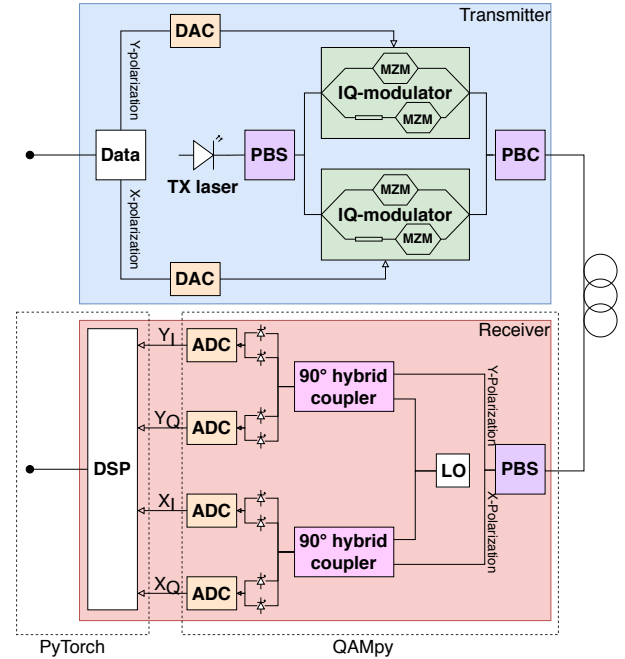


Fig. 1. Schematic representation of an optical coherent detection system

II. FIBER-OPTIC COHERENT COMMUNICATION

The overall system structure in optical coherent communication is depicted schematically in Fig. 1. The transmitter subsystem is colored blue and the receiver part red. An optical fiber is placed between the two subsystems. The use of multiple wavelengths (WDM) is omitted as this does not affect the general structure of both transmitter and receiver. The transmitter, medium and receiver are discussed next. After that, the DSP part of the receiver is addressed separately.

A. Transmitter

The transmitter in an optical coherent system consists of a continuous wave (CW) laser with a small spectral linewidth. This laser is sent through a polarizing beam splitter (PBS), which splits the incoming light into two orthogonal linear polarized beams.

The two beams are fed into in-phase and quadrature (IQ) modulators, which main function is to encode data onto the linear polarized beam. This data first goes through a digital to analog converter (DAC) and is sent into the IQ modulators. These IQ modulator are realized by substrates of $LiNbO_3$ or InP , which change their refractive indices linearly depending

on the applied voltage. This effect is called the Pockels effect. This IQ modulator consists of two Mach-Zehnder Modulators (MZM) in a Push-Pull configuration. This configuration exhibits an amplitude modulating behaviour. Having two MZMs which are phase-shifted by 90 degrees results in an in-phase and quadrature amplitude modulator.

The two IQ modulated orthogonal polarized beams are combined in the polarizing beam combiner (PBC) into an elliptical polarized beam. All required data information is incorporated in this beam. Then, the beam is sent through the optical fiber.

B. Medium

When travelling through the glass fiber, the signal is distorted. Due to group velocity dispersion (GVD) the duration of optical pulses changes. This is a result of the varying propagation velocities of wavelets within a group of wavelengths that is sent through the fiber. Also, Gaussian white noise is added to the signal as a result of quantum-mechanical vacuum fluctuations [1]. These effects are polarization independent.

Besides polarization independent impairments, there also exist impairments that depend on the polarization. PMD is an effect that occurs because of random imperfections in the optical fiber. Asymmetrical properties and other imperfections cause the two polarizations to travel at a slightly different speed. This results in a temporal difference between the two polarizations when the signal is received [7]. This is proportional to the square root of the distance of the medium. Related is the polarization dependent loss (PDL), which is energy losses that are different for the two polarizations.

Furthermore, the two polarizations are randomly mixed with each other due to the birefringent properties of the medium [1], [7].

The impairments discussed so far are linear effects. The Kerr effect is an example of a non-linear effect that happens on the transmitted signal. In this effect, the transmitted energy squared is proportional to the refractive index of the fiber.

C. Receiver

The distorted signal is received and sent through a PBS. Both polarizations are mixed with a local oscillator (LO) in a 90 degree hybrid coupler. The output ports are detected by four balanced photosensitive diodes pairs. The electrical output signals of the diode pairs represent both the in-phase and quadrature energy levels for both polarizations. Retrieving 90 degree phase shifted energy levels makes sure that no phase information is lost (phase-diversity). This is done for both polarizations (polarization-diversity), which makes sure that no polarization information is lost.

Finally, the in-phase and quadrature signals for both polarizations are converted to the digital domain by analog to digital converters (ADC) and are taken care of in the electrical domain by the DSP.

D. DSP

Until now, no information has been lost in the process of converting the optical signal into the digital electrical domain.

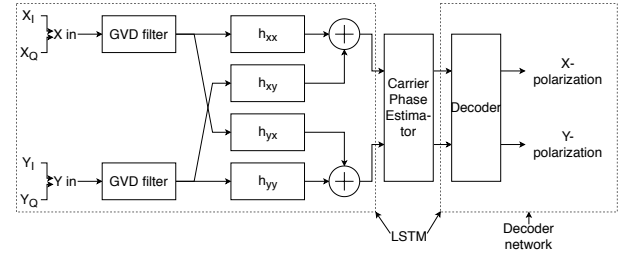


Fig. 2. DSP chain in optical coherent receiver. The \oplus operator is vectorial addition

The function of the DSP in the receiver is to recover the original data by equalizing out the impairments and finding carrier and clock phase information. How the DSP works normally is depicted in Fig. 2.

First, GVD is compensated for with a fixed filter (GVD filter). Then, all other linear impairments are compensated in a structure called butterfly structured finite impulse response (FIR) filters (\vec{h}_{xx} , \vec{h}_{xy} , \vec{h}_{yx} , \vec{h}_{yy}). These filters are adaptive since compensated impairments like PMD vary quickly over time and are stochastic. The filter tap weight algorithm used can exploit the property that incoming signals have a constant energy level. However, this Constant Modulus Algorithm (CMA) is only possible in modulation schemes with constant modulus, like M-ary PSK. The decision-directed least mean squared (DD-LMS) algorithm can be deployed on modulation schemes with varying complex moduli. This algorithm updates the filter tap weights based on the error difference between the latest estimation and the actual received bit. This algorithm works by a training sequence of symbols and providing the correct symbols to the receiver side such that it can learn how the weights of the filter taps should be set. The equations of the algorithm look as follows:

$$\vec{h}_{xx}(n+1) = \vec{h}_{xx}(n) + \mu \vec{e}_X(n) \vec{E}_x(n), \quad (1)$$

$$\vec{h}_{xy}(n+1) = \vec{h}_{xy}(n) + \mu \vec{e}_X(n) \vec{E}_y(n), \quad (2)$$

$$\vec{h}_{yx}(n+1) = \vec{h}_{yx}(n) + \mu \vec{e}_Y(n) \vec{E}_x(n), \quad (3)$$

$$\vec{h}_{yy}(n+1) = \vec{h}_{yy}(n) + \mu \vec{e}_Y(n) \vec{E}_y(n), \quad (4)$$

where n is the symbol iteration number, μ is the step-size, \vec{e}_X and \vec{e}_Y are the error signals, and \vec{E}_x and \vec{E}_y are the energies of all the included past symbols. \vec{e}_X and \vec{e}_Y are defined to be the difference between the energy of respectively the in-phase and quadrature signal components and the expected in-phase and quadrature components, which are provided in training mode.

Afterwards, the carrier phase is recovered (Carrier phase estimator). This can be done with the M-th viterbi-viterbi algorithm [1], [8], which retrieves the unwanted residue carrier phase offset. Alternatively, a single stage [9] or two-stage [10] blind phase search (BPS) algorithm can be used. After carrier phase compensation, the signal can finally be decoded and all information can be extracted (Decoder).

III. MACHINE LEARNING TECHNIQUES

Within machine learning, artificial neural networks are commonly used to develop algorithmic solutions. An example

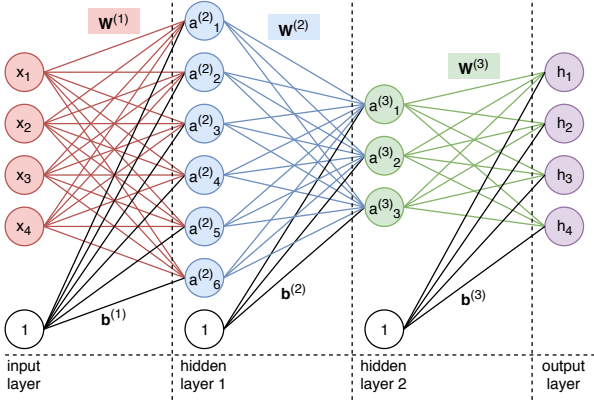


Fig. 3. Example of artificial neural network

of an artificial neural network is given in Fig. 3. In the example network, the input values, \vec{x}_i for $i \in \{1..4\}$ are multiplied with edge weights that are incorporated in a weight matrix $\mathbf{W}^{(1)}$. A bias vector $\vec{b}^{(1)}$ is added to the result. To keep the elements in this vector within certain bounds and to add non-linearity, the vector is first sent through an activation function. A commonly used activation function is the Sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. The elements of the resulting vector are the nodes of the first hidden layer in the neural network, $\vec{a}_j^{(2)}$ for $j \in \{1..6\}$. Mathematically, the next layer $\vec{a}^{(n+1)}$ is calculated by

$$\vec{a}^{(n+1)} = \sigma \left(\vec{a}^{(n)} \mathbf{W}^{(n)} + \vec{b}^{(n)} \right), \quad (5)$$

where n is a layer number. To account for the first and last layer, $\vec{a}^{(1)} = \vec{x}$ is the input layer, and $\vec{a}^{(m)} = \vec{h}$ is the output layer, where m is the amount of hidden layers in the network + 2.

The output values are compared with the correct labels with a loss-function. Based on the chosen loss-function, the network weights and biases are adapted to more suitable values through a operation called backpropagation [11], which closely resembles the DD-LMS algorithm from equations 1-4.

A. Recurrent neural networks

A special class of neural networks are recurrent neural networks (RNN). These networks differ from normal neural networks because the previous output is included in the new input. Thus, old or intermediate predictions are combined in the input layer of the next iteration. RNNs are most suitable when the input data has sequential behaviour.

However, RNNs have learning problems when the network has to remember data from the past that is too far away due to the vanishing/exploding gradient problem [12]. This problem was solved in 1997 by a network structure called the Long Short Term Memory (LSTM) [12] neural network, which is a special version of an RNN. A schematic of this type of network is shown in Fig. 4.

The network has an internal state \vec{c}_t which holds remembered information from the past. In every iteration, part of \vec{c}_t is forgotten, added, and outputted, depending on the new input \vec{x}_t and previous output h_{t-1} . First, information is forgotten by

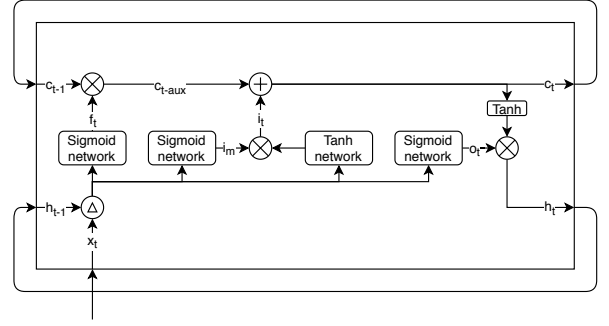


Fig. 4. Schematic depiction of an LSTM. The \triangle operator is the concatenation of the new input with the old output. The \oplus operator is vectorial addition. The \otimes operator is elementwise multiplication between vectors.

element-wise multiplying \vec{c}_t with a forget vector \vec{f}_t . This is shown in equation 6. This vector \vec{f}_t consists of values between 0 and 1: a 0 indicating to forget information, and a 1 indicating to remember information. This vector \vec{f}_t is generated by a network with a Sigmoid activation function. This is shown in equation 7. Then, new information is added to \vec{c}_t . To do this, the input is fed through a network with a $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ activation function. This information is modulated by a vector \vec{i}_m from another Sigmoid driven network as shown in equation 8. The modulated vector \vec{i}_t is stated in equation 9. In equation 10 it is shown how the internal state is updated. The output \vec{h}_t is modulated by the vector \vec{o}_t , which again is the result of a Sigmoid driven network and is shown in equation 11. The final output \vec{h}_t is the result of the tanh of the internal state, $\tanh(\vec{c}_t)$, modulated by \vec{o}_t . This is shown in equation 12.

$$c_{t-aux} = \vec{f}_t \circ c_{t-1} \quad (6)$$

$$\vec{f}_t = \sigma \left(\mathbf{W}_f \vec{x}_t + \mathbf{U}_f h_{t-1} + \vec{b}_f \right) \quad (7)$$

$$\vec{i}_m = \sigma \left(\mathbf{W}_m \vec{x}_t + \mathbf{U}_m h_{t-1} + \vec{b}_m \right) \quad (8)$$

$$\vec{i}_t = \vec{i}_m \circ \tanh \left(\mathbf{W}_i \vec{x}_t + \mathbf{U}_i h_{t-1} + \vec{b}_i \right) \quad (9)$$

$$\vec{c}_t = c_{t-aux} + \vec{i}_t \quad (10)$$

$$\vec{o}_t = \sigma \left(\mathbf{W}_o \vec{x}_t + \mathbf{U}_o h_{t-1} + \vec{b}_o \right) \quad (11)$$

$$\vec{h}_t = \vec{o}_t \circ \tanh(\vec{c}_t) \quad (12)$$

In equations 6-12, the t subscript denotes the LSTM iteration. \mathbf{W} and \mathbf{U} respectively denote the weight matrix for the new input \vec{x}_t and the old input h_{t-1} . \vec{b} is the network bias. Subscripts f , i , m and o respectively mean "forget", "input", "modulation" and "output". The operator \circ is the elementwise multiplication between vectors.

Training an LSTM happens by processing a discrete sequence of length L of input data. Entries in this sequence iterate through the LSTM until it reaches the prediction of the last entry. This last prediction is compared with the correct value in a loss function and the weights and biases are updated through backpropagation.

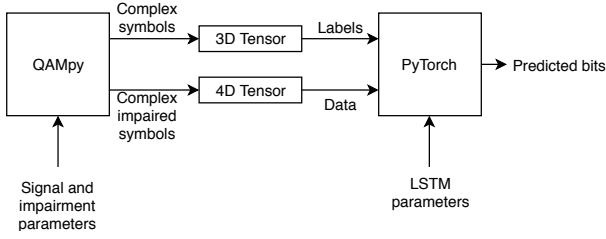


Fig. 5. Environment of the conducted experiment

TABLE I
PARAMETERS OF DATA AND NEURAL NETWORK IN DECODER EXPERIMENT

Parameter	Value
Modulation scheme	64-QAM
Generated symbols per polarization	2^{20}
SNR	15
Hidden nodes	32 (1 layer)
Activation function	ReLU
Batch size	64
Learning rate	0.001
Loss function	Cross entropy
Training batches/Testing batches ratio	75%/25%

IV. ENVIRONMENT OF EXPERIMENT

To experiment with the unique combination of optical fiber communication and LSTMs, an environment was created as depicted in Fig. 5. The two used frameworks are QAMpy and PyTorch: QAMpy to generate pseudo-random complex symbols and to artificially impair them, and PyTorch to implement the recovery network.

In the LSTM experiment, the output of the QAMpy are the original symbols and the impaired symbols, which are converted in respectively an 3D and 4D tensor. This first tensor contains per batch a 2D tensor consisting of the final correct bit predictions. The second tensor contains per batch a 3D tensor: a sequence of impaired signal energy levels.

The virtual environment can be regarded as shown in Fig. 1, in which is indicated which parts are implemented by which framework.

V. EXPERIMENT

In this section, the simulated experiment is described. First, only the decoder was substituted in simulation by a regular neural network (Fig. 2, Decoder network). Secondly, the entire DSP chain except the carrier phase estimator was substituted in simulation by an LSTM (Fig. 2, LSTM). Finally, the LSTM experiment is conducted on real data.

A. Neural network decoder

To first test if machine learning can be used in the optical receiver, the decoder has been substituted with a regular neural network. The rest of the DSP chain remains the same. A schematic depiction of the DSP chain substitution is shown in Fig. 2 (Decoder network). For this experiment, only Gaussian noise has been added. The parameters of the simulation are shown in Table I.

The results are shown in Fig. 6. The different colors show how the decision boundaries of the network look like in the

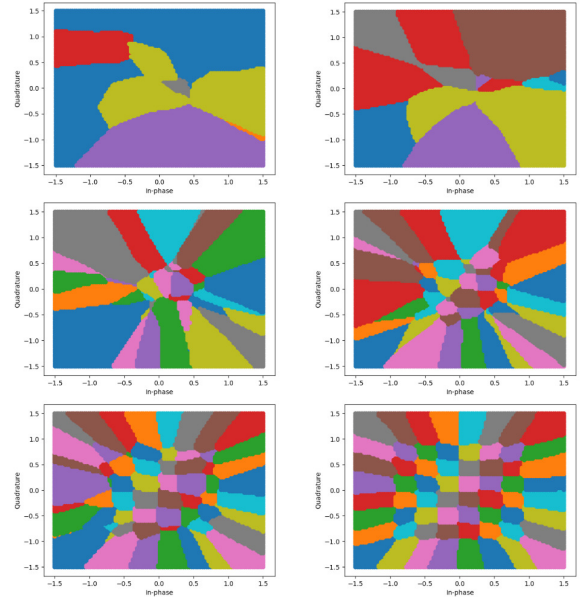
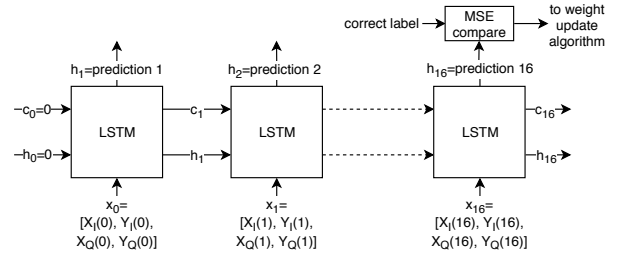


Fig. 6. Development of 64-QAM decoder neural network. Upper left after 1 training steps, upper right after 10 training steps, middle left after 300 training steps, middle right after 500 training steps, bottom left after 1000 training steps and bottom right after the last, 24576th training step

Fig. 7. Training of an LSTM, where the sequence length L is 16

constellation plot. After training, the final achieved accuracy using the test data was 99.9886%.

B. Neural network decoder and equalizer

Decoding signals which are only distorted by Gaussian noise is not a practical example of machine learning in an optical receiver. More of the receiver DSP has to be included in the machine learning algorithm. Because the conventional way of filtering includes a FIR that uses data from the near past, the machine learning network should also include the past data sequence. Therefore, a suitable machine learning network for replacing the entire DSP chain would be to use an RNN-like structure, such as an LSTM.

In this experiment, the four input nodes of the LSTM are the in-phase and quadrature energy levels of both polarizations. The N output nodes $h_k, k \in \{1..L\}$ are the predicted bit of both polarization, where N is $N = 2 \log_2(M)$, in which M is the alphabet size of the modulation. How the weights of the LSTM are trained is depicted schematically in Fig. 7.

After training, an indefinite discrete sequence of impaired symbol energy levels can be sent through the network. The

TABLE II
USED PARAMETERS OF GENERATED DATA

Parameter	Value
Modulation	QPSK
Amount of symbols per polarization	2^{20}
Symbol rate	25GBd
β of RRCos	0.1
SNR	15
θ of PMD	$\frac{\pi}{5}$
τ_{DGD} of PMD	40ps
Linewidth of lasers	1MHz
Frequency offset between lasers	1MHz

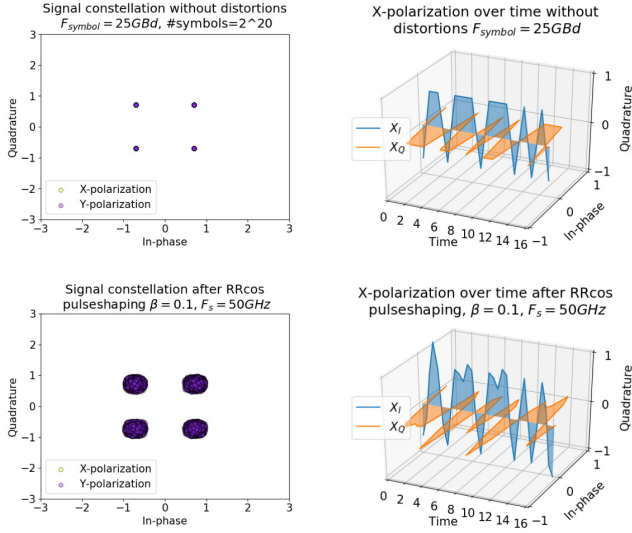


Fig. 8. Left the signal constellations of the transmitted signal and right the corresponding time plot (in symbol time = $1/25\text{GBd}$) for the X-polarization. Upper plots are original symbols, bottom plots are pulse-shaped

network gives a prediction for every given input and is no longer dependent on the predefined sequence length L .

The parameters of the data and signal impairments are shown in Table II. The resulted signal constellation of the original signal is given in the upper left plot in Fig. 8. For the X-polarization of this original signal, the first 16 symbols are shown in a 3D time-plot in the upper right plot in Fig. 8.

To limit the required bandwidth of the signal, the signal is pulse-shaped with a root raised cosine filter with a roll-off factor β of 0.1. After pulse shaping, the signal constellation and time plot are slightly distorted. The resulting signal constellation and X-polarization time plots are shown in respectively the bottom left and right subplots in Fig. 8. This signal is sent by the transmitter.

While transmitting the bandlimited signal, the signal is inevitably impaired by the effects as discussed in section II-B. These impairments were artificially added to the transmitted signal. The resulting signal constellation plots for the incrementally added impairments can be seen in Fig. 9. First, additive Gaussian white noise was added (upper left). Then, PMD impairments were included (upper right). Finally, phase noise (bottom left) and frequency offset (bottom right) were added. This final plot is what the receiver picks up after the ADCs and is input for the recovery neural network.

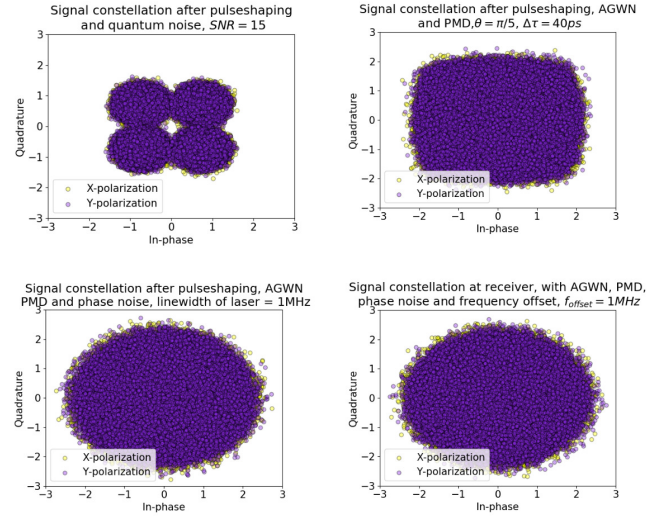


Fig. 9. Signal constellation of received signal, impaired by AGWN (upper left) and PMD (upper right) and phase noise (bottom left) and frequency offset (bottom right)

To train the network, the training batches were shuffled to increase the balance of the training data. In this way, the network does not overfit on associated noise of the previous sequence, since this is one of the pitfalls of neural network training.

Unfortunately, the neural network did not show results significantly higher than 50% (random guessing) after fully training and seemed to have a non-decreasing loss function. The reason why this is the case is because no carrier phase estimator is input in the network. Therefore, the network could not filter out impairments like phase noise and frequency offset.

To see if the network would have a better performance with extra carrier phase estimation information, the M-th power viterbi-viterbi was added in the network, but without success. Also, single-stage and two-stage blind phase search was added, but no improvement was seen. This is due to the 90 degree symmetric nature of QAM in combination with the shuffling of training data batches. The network can find out what the residue carrier phase offset is (viterbi-viterbi or blind phase search algorithm) within a 90 degree quadrant. However, the network cannot know in which quadrant it is located, because no initial conditions in a randomly selected batch sequence are known. These initial conditions are crucial to predict the integer multiple of 90 degree carrier phase offset of the signal. Thus, prediction acts as a random guess and accuracies cannot by definition have a better performance than 50%.

To see if the network would perform without the carrier phase estimator, the phase noise and frequency offset impairments were omitted in the simulation. The substitution of the DSP chain is depicted in Fig. 2 (LSTM). The parameters of the training data and LSTM are given in Table III.

The graphs of the loss functions for the LSTM for various SNRs is given in the upper plot of Fig. 10. The accuracy graphs of the prediction correctness is given in the lower plot

TABLE III
PARAMETERS OF DATA AND LSTM OF SIMULATED EXPERIMENT

Parameter	Value
Training sequence length	16
Learning rate	0.01
Hidden nodes	128 (1 layer)
Batch size	256
Training batches/Testing batches ratio	90%/10%
Loss function	Mean squared error

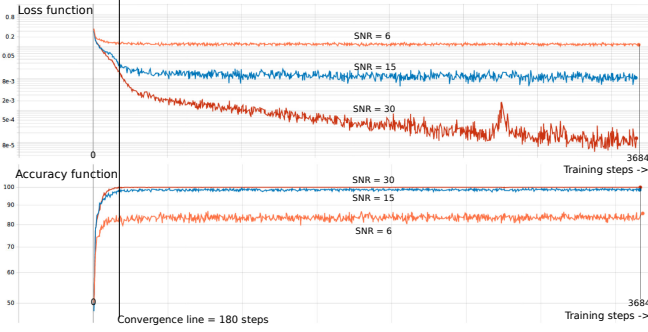


Fig. 10. Loss function and accuracy function of LSTM, red curve: SNR 30, blue curve: SNR 15, orange curve: SNR 6. X-axis depicts the amount of training steps. Logarithmic Y-axis.

of Fig. 10. The final testing data accuracies for the SNR of 30, 15 and 6 were respectively 100%, 98,5952% and 83,5871%. Convergence happens after around 50.000 symbols.

C. Real data experiment

To test the LSTM performance with real data, test data from a optical fiber lab was used. The modulation of this data is 8-QAM. 65536 pseudo-random symbols were sent which are pulse-shaped by a root raised cosine filter with roll-off factor $\beta = 0.01$. The data was sent once through a standard single mode optical fiber (SSMF) of 75km and was amplified once by an erbium-doped fiber amplifier (EDFA). The frequency offset was separately compensated for, because the network cannot filter this on its own. In Fig. 11, the input constellation can be seen on the left and the output constellation on the right. The parameters of the training data and the LSTM are given in Table IV. The final obtained accuracy of the prediction network is 83.2145%. However, this data could be detected with no errors with conventional techniques.

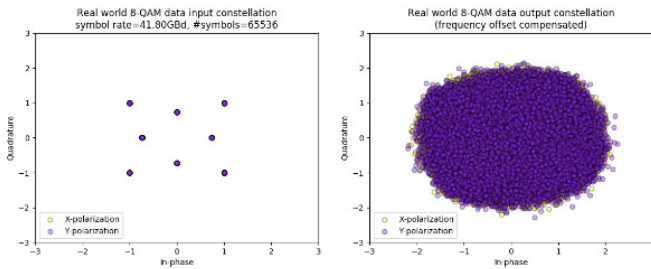


Fig. 11. Input constellation plot of real world 8-QAM data (left) and output constellation of 8-QAM signal after 75km of SSMF with one EDFA amplification (right)

TABLE IV
PARAMETERS OF DATA AND LSTM OF REAL WORLD EXPERIMENT

Parameter	Value
Training sequence length	32
Learning rate	0.02
Hidden nodes	128 (1 layer)
Batch size	128
Training batches/Testing batches ratio	90%/10%
Loss function	Mean squared error

VI. DISCUSSION

The results of the simulated and real data do not show improvement in comparison with the currently deployed DSP chain hardware. The current DSP chain has a BER of several orders of magnitude lower. However, it is difficult to find out what exactly is causing this as intermediate calculations in neural networks do not necessarily have a clear meaning. Furthermore, carrier phase estimation is not included in the LSTM and therefore it cannot properly filter out phase noise and frequency offset.

There are some advantages of a machine learning approach to the optical coherent receiver signal recovery problem over the conventional way. One of these advantages is that it is able to filter both linear and non-linear impairments due to the non-linear Sigmoid activation function embedded in the network. Furthermore, no knowledge is needed about the transmission line and signal impairments do not have to be modeled correctly, as the network will learn how to deal with any kind of impairment, given a training data sequence of a long enough length. Besides that, all the hardware blocks that are in the conventional receiver DSP can be combined into just a single neural network. Adaptive filters can be replaced by a single fixed and static neural network.

New research has to be done to increase the prediction accuracy of the network by monitoring exactly what is happening inside the LSTM and try different specializations of LSTMs to make the network more suitable to the problem. Besides that, it has to be researched how the carrier phase estimator block can be implemented inside the neural network by either a separate network or by increasing the input dimensions of the network with extra information such that the integer multiple of 90 degrees carrier phase can be recovered.

VII. CONCLUSION

The conventional optical coherent transmission system was discussed. Machine learning techniques that are suitable to aid the optical coherent receiver were studied. Afterwards, the simulation and real world experiments have shown that the DSP chain can partially be aided with machine learning techniques with regular and recurrent neural networks.

However, for now, these machine learning techniques as a substitute for DSP in optical coherent receivers is not yet feasible. Before this becomes an interesting option, there needs to be more understanding on how to improve the BER of a neural network solution and how the carrier phase estimation can be included in the network.

REFERENCES

- [1] K. Kikuchi, "Fundamentals of coherent optical fiber communications," *Journal of Lightwave Technology*, vol. 34, no. 1, pp. 157–179, Jan 2016.
- [2] Y. Kisaka, M. Tomizawa, and Y. Miyamoto, "Digital signal processor (dsp) for beyond 100g optical transport," *NTT Technical Review*, vol. 14, 09 2016.
- [3] A. Argyris, J. Bueno, and I. Fischer, "Photonic machine learning implementation for signal recovery in optical communications," *Scientific Reports*, vol. 8, p. 8487, 05 2018.
- [4] S. Tsukamoto, D. . Ly-Gagnon, K. Katoh, and K. Kikuchi, "Coherent demodulation of 40-gbit/s polarization-multiplexed qpsk signals with 16-ghz spacing after 200-km transmission," in *OFC/NFOEC Technical Digest. Optical Fiber Communication Conference, 2005.*, vol. 6, March 2005, pp. 3 pp. Vol. 5–.
- [5] S. J. Savory, "Digital filters for coherent optical receivers," *Opt. Express*, vol. 16, no. 2, pp. 804–817, Jan 2008. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-16-2-804>
- [6] C. Xie, "Impact of nonlinear and polarization effects on coherent systems," in *2011 37th European Conference and Exhibition on Optical Communication*, Sep. 2011, pp. 1–3.
- [7] Reimer, Michael, "Modeling and simulation of polarization mode dispersion and polarization dependent loss," 2007. [Online]. Available: <http://hdl.handle.net/10012/2687>
- [8] T. F. Portela, D. V. Souto, V. Rozental, H. B. Ferreira, D. Mello, and H. Griesser, "Analysis of signal processing techniques for optical 112 gb/s dp-qpsk receivers with experimental data," *Journal of Microwaves, Optoelectronics and Electromagnetic Applications*, vol. 10, pp. 155–164, 06 2011.
- [9] T. Pfau, S. Hoffmann, and R. Noé, "Hardware-efficient coherent digital receiver concept with feedforward carrier recovery for M -qam constellations," *J. Lightwave Technol.*, vol. 27, no. 8, pp. 989–999, Apr 2009. [Online]. Available: <http://jlt.osa.org/abstract.cfm?URI=jlt-27-8-989>
- [10] Q. Zhuge, C. Chen, and D. V. Plant, "Low computation complexity two-stage feedforward carrier recovery algorithm for m-qam," in *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011.* Optical Society of America, 2011, p. OMJ5. [Online]. Available: <http://www.osapublishing.org/abstract.cfm?URI=OFC-2011-OMJ5>
- [11] Hecht-Nielsen, "Theory of the backpropagation neural network," in *International 1989 Joint Conference on Neural Networks*, 1989, pp. 593–605 vol.1.
- [12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>