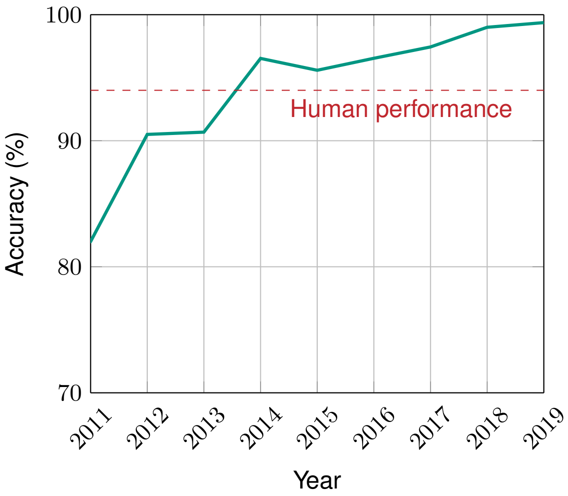
Karlsruhe Institute of Technology

End-to-End Learning in Optical Fiber Communications: Concept and Transceiver Design

Boris Karanov, Polina Bayvel, Laurent Schmalen

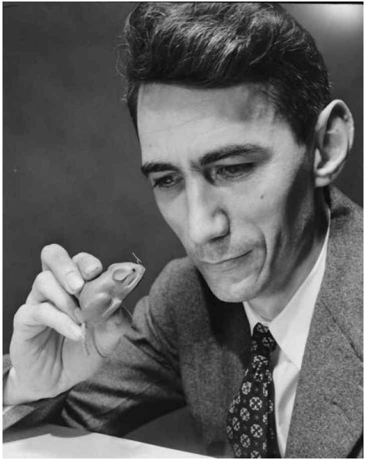
Machine Learning Success: Image Recognition



- Image recognition accuracy on CIFAR-10 dataset[1]
- In less than 10 years, advances in machine learning and in particular **neural networks** and **deep learning** yield systems that surpass human capabilities
- Almost perfect recognition accuracy

[1] Accuracy data from `https://benchmarks.ai/cifar-10`

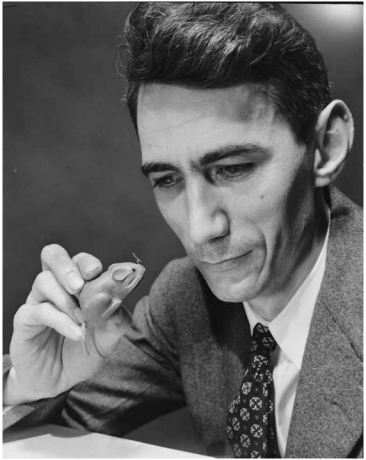
Machine Learning for Communications



- 70 years of communications engineering have brought us close to the Shannon limit[2] (Shannon capacity)
- **But:** this applies only to a sub-class of all communication problems
- We don't know neither good nor optimal transceivers for many communication systems

[2] Image source: `https://commons.wikimedia.org/wiki/File:Claude_Shannon_1776.jpg`

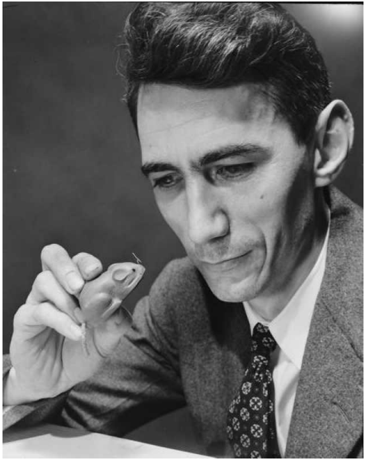
Machine Learning for Communications



- 70 years of communications engineering have brought us close to the Shannon limit[2] (Shannon capacity)
- **But:** this applies only to a sub-class of all communication problems
- We don't know neither good nor optimal transceivers for many communication systems
- **Machine learning** can help us identify novel transceivers
- **This talk:** Concept of using neural networks for learning of optical communication systems

[2] Image source: `https://commons.wikimedia.org/wiki/File:Claude_Shannon_1776.jpg`

Machine Learning for Communications



- 70 years of communications engineering have brought us close to the Shannon limit[2] (Shannon capacity)

- **But:** this applies only to a sub-class of all communication problems
- We don't know neither good nor optimal transceivers for many communication systems

- **Machine learning** can help us identify novel transceivers

- **This talk:** Concept of using neural networks for learning of optical communication systems
- **Talk TH1D-1:** Experimental Demonstration and Future Trends (Thursday, December 10th)

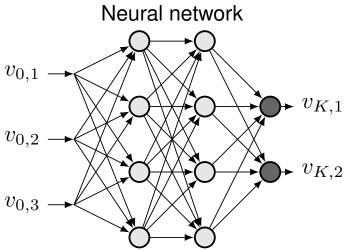
[2] Image source: `https://commons.wikimedia.org/wiki/File:Claude_Shannon_1776.jpg`

Neural Networks: A Whirlwind Tour

Feed-Forward Neural Networks (FFNN)

- Maps an input vector $\mathbf{v}_0 = \begin{pmatrix} v_{0,1} & \dots & v_{0,M} \end{pmatrix}$ to an output vector $\mathbf{v}_K = \begin{pmatrix} v_{K,1} & \dots & v_{K,n} \end{pmatrix} = f_{\text{NN}}(\mathbf{v}_0)$
- The neural network (NN) is a composed function consisting of layers, where each layer computes

$$\mathbf{v}_k = g_{\text{NL},k}(\mathbf{W}_k \mathbf{v}_{k-1} + \mathbf{b}_k), \qquad k = 1, \dots, K$$



Neural Networks: A Whirlwind Tour

Karlsruhe Institute of Technology

Feed-Forward Neural Networks (FFNN)

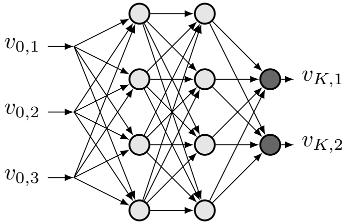
- Maps an input vector $\mathbf{v}_0 = \begin{pmatrix} v_{0,1} & \dots & v_{0,M} \end{pmatrix}$ to an output vector $\mathbf{v}_K = \begin{pmatrix} v_{K,1} & \dots & v_{K,n} \end{pmatrix} = f_{\text{NN}}(\mathbf{v}_0)$
- The neural network (NN) is a composed function consisting of layers, where each layer computes

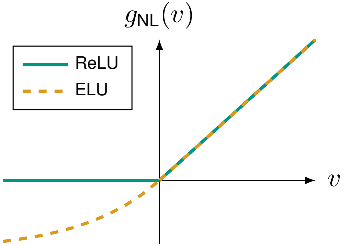
$$\mathbf{v}_k = g_{\text{NL},k}(\mathbf{W}_k \mathbf{v}_{k-1} + \mathbf{b}_k), \qquad k = 1, \dots, K$$

- Activation function g_{NL} introduces **nonlinear** relation between layers
- A popular choice for g_{NL} is the ReLU activation function (or one of its variants, e.g. the ELU function)

$$\mathbf{x} = g_{\text{NL}}(\mathbf{v}) = g_{\text{ReLU}}(\mathbf{v}) \qquad \text{with } x_i = \max(0, v_i)$$

Neural network



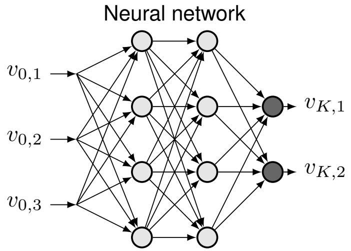


Neural Networks: A Whirlwind Tour

Feed-Forward Neural Networks (FFNN)

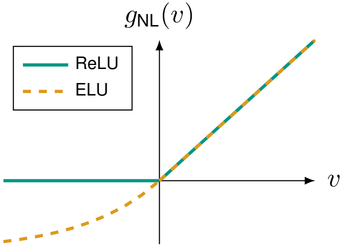
- Maps an input vector $\mathbf{v}_0 = \begin{pmatrix} v_{0,1} & \dots & v_{0,M} \end{pmatrix}$ to an output vector $\mathbf{v}_K = \begin{pmatrix} v_{K,1} & \dots & v_{K,n} \end{pmatrix} = f_{\text{NN}}(\mathbf{v}_0)$
- The neural network (NN) is a composed function consisting of layers, where each layer computes

$$\mathbf{v}_k = g_{\text{NL},k}(\mathbf{W}_k \mathbf{v}_{k-1} + \mathbf{b}_k), \qquad k = 1, \dots, K$$



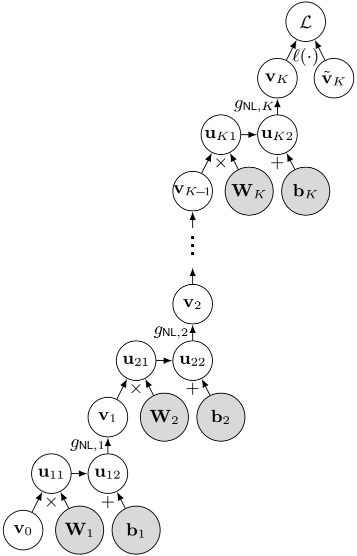
- Activation function g_{NL} introduces **nonlinear** relation between layers
- A popular choice for g_{NL} is the ReLU activation function (or one of its variants, e.g. the ELU function)

$$\mathbf{x} = g_{\text{NL}}(\mathbf{v}) = g_{\text{ReLU}}(\mathbf{v}) \qquad \text{with } x_i = \max(0, v_i)$$



- **Objective of learning:** Find parameters $\mathbf{W}_k$ and $\mathbf{b}_k$ such that $f_{\text{NN}}(\mathbf{v}_0)$ approximates an (unknown) function whose inputs and outputs can be observed and fed to a training algorithm

Neural Networks: Computational Graph



Deep Learning

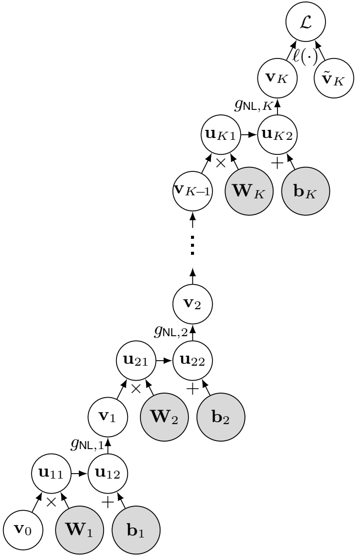
- Objective is to find NN parameters $\{\mathbf{W}_k, \mathbf{b}_k\}$ that minimize a **loss function**

$$\mathcal{L}\left(\{\mathbf{W}_k, \mathbf{b}_k\}_K\right) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{v}_{0,i}, \tilde{\mathbf{v}}_{K,i}) \in \mathcal{S}} L\left(f_{\text{NN}}\left(\mathbf{v}_{0,i}\right), \tilde{\mathbf{v}}_{K,i}\right)$$

  where $(\mathbf{v}_{0,i}, \tilde{\mathbf{v}}_{K,i})$ are **examples** of inputs $(\mathbf{v}_{0,i})$ and observed outputs $(\tilde{\mathbf{v}}_{K,i})$ of the system we like to approximate

- **Training data set** $\mathcal{S}$ contains these examples

Neural Networks: Computational Graph



Deep Learning

- Objective is to find NN parameters $\{\mathbf{W}_k, \mathbf{b}_k\}$ that minimize a **loss function**

$$\mathcal{L}\left(\{\mathbf{W}_k, \mathbf{b}_k\}_K\right) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{v}_{0,i}, \tilde{\mathbf{v}}_{K,i}) \in \mathcal{S}} L\left(f_{\mathsf{NN}}\left(\mathbf{v}_{0,i}\right), \tilde{\mathbf{v}}_{K,i}\right)$$

  where $(\mathbf{v}_{0,i}, \tilde{\mathbf{v}}_{K,i})$ are **examples** of inputs $(\mathbf{v}_{0,i})$ and observed outputs $(\tilde{\mathbf{v}}_{K,i})$ of the system we like to approximate

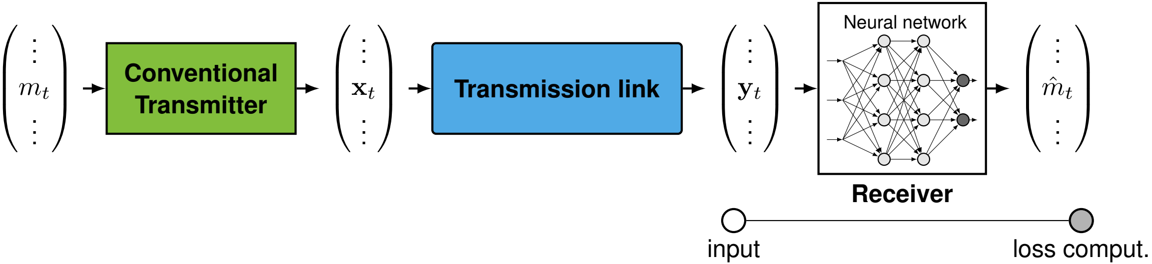
- **Training data set** $\mathcal{S}$ contains these examples

- Parameters $\{\mathbf{W}_k, \mathbf{b}_k\}$ are optimized using **gradient descent**

$$\mathbf{W}_k \leftarrow \mathbf{W}_k - \epsilon \nabla_{\mathbf{W}_k} \mathcal{L}\left(\{\mathbf{W}_k, \mathbf{b}_k\}_K\right)$$
$$\mathbf{b}_k \leftarrow \mathbf{b}_k - \epsilon \nabla_{\mathbf{b}_k} \mathcal{L}\left(\{\mathbf{W}_k, \mathbf{b}_k\}_K\right)$$

- Efficient computation of gradient using a **computational graph**

Receiver Processing Using Neural Networks

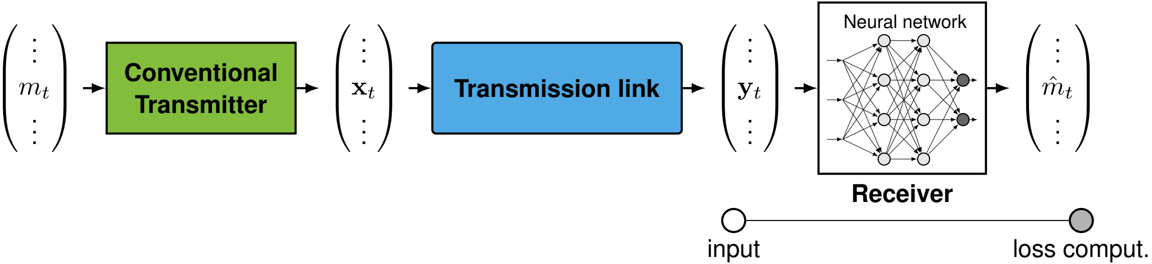


- Transmission of known sequence of messages (e.g., pilot) m_t
- Based on received data ($\mathbf{y}_t$), try to recover m_t as closely as possible
- Training dataset is database of received signals and associated symbols $\mathcal{S} = \{(\mathbf{y}_t, m_t) : t = 1, \dots N\}$
- Proposed for optical communications **[Lyu15]** and used for example as receivers in PON systems **[HCvV19]**

[Lyu15] I. Lyubomirsky, "Machine learning equalization techniques for high speed PAM4 fiber optic communication systems," *CS229 Final Project Report*, Stanford University, 2015

[HCvV19] 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. Optical Fiber Communications Conference (OFC)*, 2019

Receiver Processing Using Neural Networks

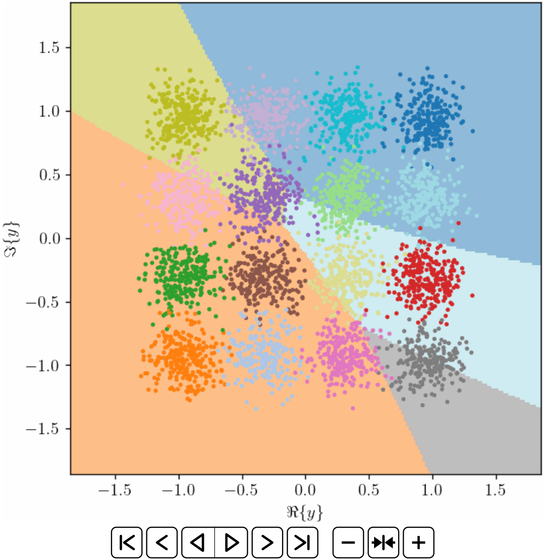


- Transmission of known sequence of messages (e.g., pilot) m_t
- Based on received data ($\mathbf{y}_t$), try to recover m_t as closely as possible
- Training dataset is database of received signals and associated symbols $\mathcal{S} = \{(\mathbf{y}_t, m_t) : t = 1, \dots N\}$
- Proposed for optical communications **[Lyu15]** and used for example as receivers in PON systems **[HCvV19]**
- **Example:** AWGN channel

[Lyu15] I. Lyubomirsky, "Machine learning equalization techniques for high speed PAM4 fiber optic communication systems," *CS229 Final Project Report*, Stanford University, 2015

[HCvV19] 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. Optical Fiber Communications Conference (OFC)*, 2019

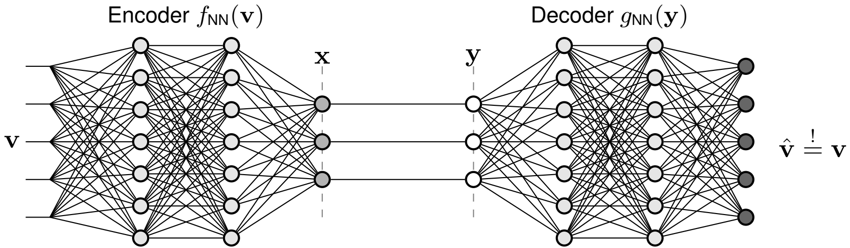
Example: Deep NN Detection in AWGN Channel



- Implementation using PyTorch[1]
- Source code available online[2]

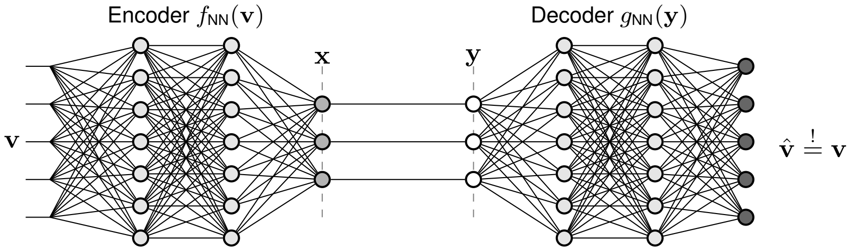
[1] http://pytorch.org
[2] https://github.com/kit-cel/ecoc_20_learning

Auto-encoders – Basic Concept



- Consists of an **encoder** $f_{\text{NN}}(\mathbf{v})$ and a **decoder** $g_{\text{NN}}(\mathbf{y})$
- **Goal:** Try to reproduce $\mathbf{v}$ by $\hat{\mathbf{v}}$ as close as possible

Auto-encoders – Basic Concept



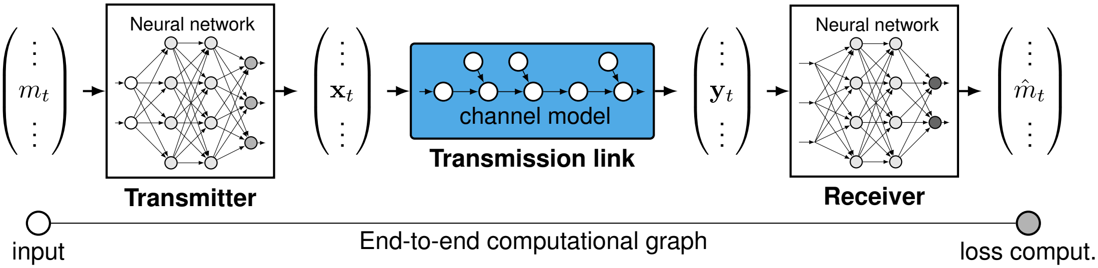
- Consists of an **encoder** $f_{\text{NN}}(\mathbf{v})$ and a **decoder** $g_{\text{NN}}(\mathbf{y})$
- **Goal:** Try to reproduce $\mathbf{v}$ by $\hat{\mathbf{v}}$ as close as possible

- Resembles a communication system (Goal: reproduce information as close as possible)
- Can we use auto-encoders to design communication systems? **[OKC16]**, **[OH17]**

[OKC16] T. O'Shea, K. Karra and T. C. Clancy, "Learning to Communicate: Channel Auto-encoders, Domain Specific Regularizers, and Attention," *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2016

[OH17] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563-575, 2017.

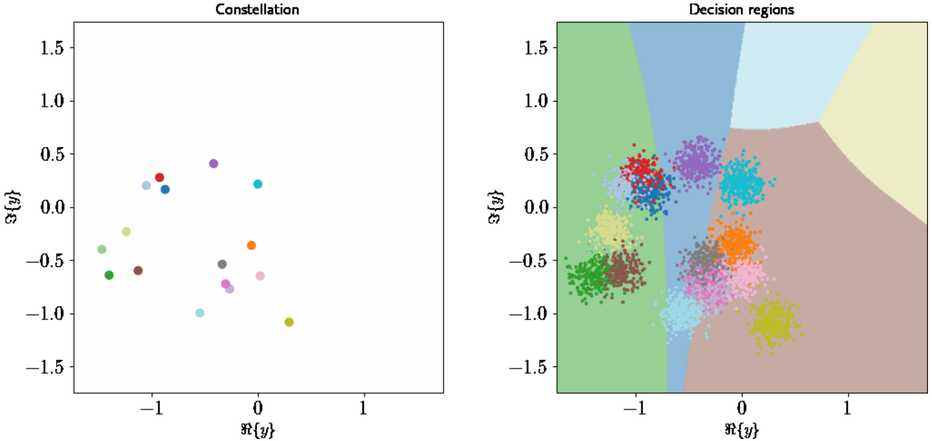
Communication System as Auto-Encoder



- Replacing (parts of) transmitter by neural network
- Transmission link needs to be modeled by an adequate, **differentiable** channel model that can be part of the computational graph

- **Example:** AWGN channel $\mathbf{y}_t = \mathbf{x}_t + \mathbf{n}_t$ with $\nabla_{\mathbf{W}} \mathbf{y}_t = \nabla_{\mathbf{W}} \mathbf{x}_t$

Example: Auto-Encoder in AWGN Channel



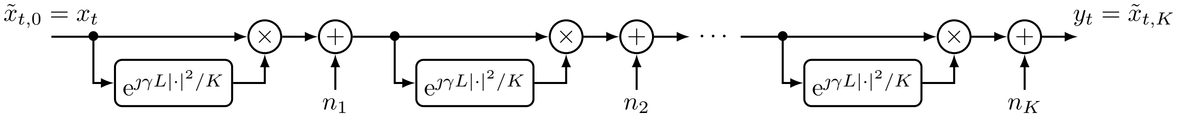
- Implementation using PyTorch
- Source code available online[3]

[3] https://github.com/kit-cel/ecoc_20_learning

Example: Zero-Dispersion Optical Fiber

Karlsruhe Institute of Technology

- A more complicated model is the zero-dispersion optical fiber model used in [LHG+18]
- Transmission over an optical fiber of length L having nonlinearity parameter γ and zero dispersion ($\beta_i = 0$)



- Let $\tilde{x}_{x,0} = x_t$ and recursively compute

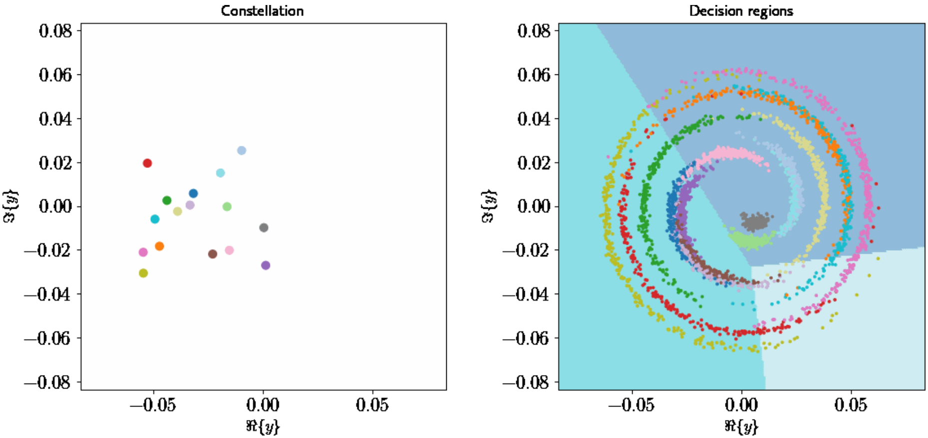
$$\tilde{x}_{t,i+1} = \tilde{x}_{t,i} \cdot e^{\jmath L\gamma|\tilde{x}_{t,i}|^2/K} + n_{i+1}, \quad 0 \leq i < K$$

  where $n_{i+1} \sim \mathcal{CN}(0, P_N/K)$ is complex, circularly symmetric Gaussian noise (assuming ideal distributed amplification) and K the number of steps (ideally, $K \to \infty$)
- Computation graph is fully **differentiable**

[LHG+18] S. Li, C. Häger, N. Garcia, and H. Wymeersch, "Achievable Information Rates for Nonlinear Fiber Communication via End-to-end Autoencoder Learning," *Proc. ECOC*, Rome, Sep. 2018

Example: Zero-Dispersion Optical Fiber

Karlsruhe Institute of Technology



- Implementation using PyTorch
- Source code available online[4]

[4] https://github.com/kit-cel/ecoc_20_learning

Auto-Encoders in Optical Communications

- End-to-end learning applied to optimize multidimensional constellations for IM/DD links **[KCT+18]**, **[CBS18]**
- End-to-end learning of transceivers for the nonlinear Fourier transform **[GJD+20]**
- Constellation optimization for the zero-dispersion channel **[LHG+18]**
- Constellation optimization for geometric shaping **[JEY+18]**, **[JYZ19]**, **[GAC+20]**
- Differentiable channel models and nonlinearity compensation for coherent systems **[HP20]**

[KCT+18] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-end deep learning of optical fiber communications," *Journal of Lightwave Technology*, vol. 36, no. 20, pp. 4843–4855, 2018

[CBS18] M. Chagnon, B. Karanov, and L. Schmalen, "Experimental demonstration of a dispersion tolerant end-to-end deep learning-based IM-DD transmission system," *Proc. European Conference on Optical Communication (ECOC)*, 2018

[GJD+20] S. Gaiarin, R. T. Jones, F. Da Ros, and D. Zibar, "End-to-end optimized nonlinear Fourier transform-based coherent communications," *Proc. CLEO: Science and Innovations*, May 2020

[LHG+18] S. Li, C. Häger, N. Garcia, and H. Wymeersch, "Achievable Information Rates for Nonlinear Fiber Communication via End-to-end Autoencoder Learning," *Proc. European Conference on Optical Communication (ECOC)*, Sep. 2018

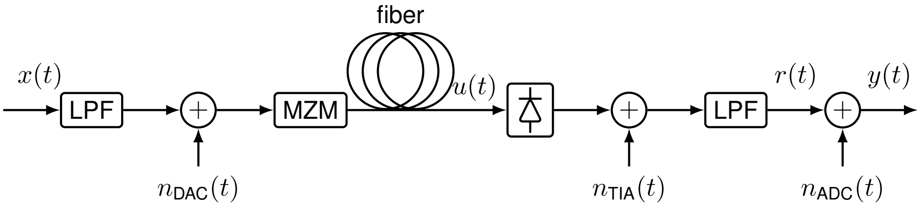
[JEY+18] R. T. Jones, T. A. Eriksson, M. P. Yankov, and D. Zibar, "Deep learning of geometric constellation shaping including fiber nonlinearities,", *Proc. European Conference on Optical Communication (ECOC)*, 2018

[JYZ19] R. T. Jones, M. Yankov, and D. Zibar, "End-to-end learning for GMI optimized geometric constellation shape," arXiv:1907.08535, 2019

[GAC+20] K. Gümüş, A. Alvarado, B. Chen, C. Häger, and E. Agrell, "End-to-End Learning of Geometrical Shaping Maximizing Generalized Mutual Information," *Proc. Optical Fiber Communications Conference (OFC)*, Mar. 2020

[HP20] C. Häger and H. D. Pfister, "Physics-Based Deep Learning for Fiber-Optic Communication Systems," *Journal of Selected Areas in Communications*, 2021

End-to-end Learning for IM/DD



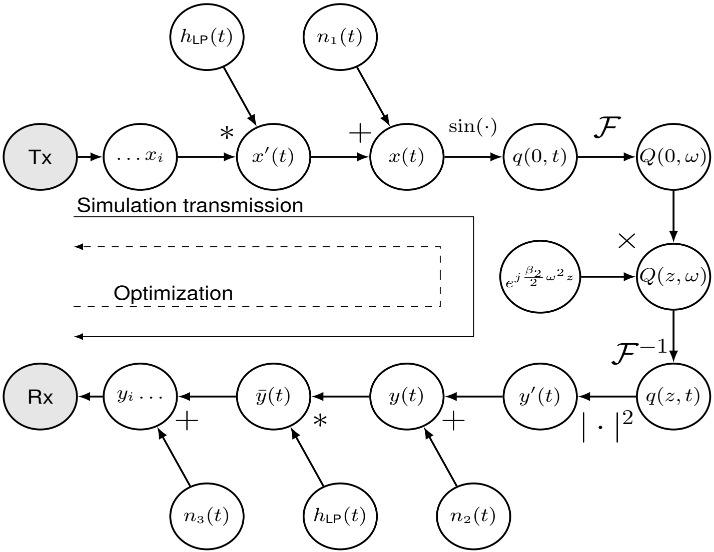
- IM/DD (intensity modul., direct detect.) ubiquitous in access networks and data center interconnects

$$y(t) = \left| \hat{h}_{\text{Fiber}} \left(\hat{h}_{\text{MZM}} \left(x(t) + n_{\text{DAC}}(t) \right) \right) \right|^2 + n_{\text{TIA}}(t) + n_{\text{ADC}}(t)$$

- Fiber adds only chromatic dispersion (no nonlinearities) with $H(\omega, z) = \exp\left(\jmath \frac{\beta_2}{2} z \omega^2 \right)$
- Nonlinear channel with memory, *however*, optimal **computationally feasible algorithms absent**
- Model is fully **differentiable**

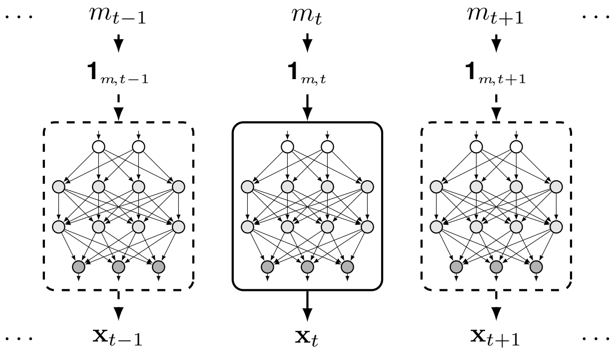
- Contrary to previous examples, channel input is **waveform**, not modulation symbols
- We need to **learn optimal waveforms**

Computational Graph of Channel Model



- Chromatic dispersion is added in frequency domain
- Fourier transform is linear and differentiable
- Channel model can be directly implemented in machine learning software

First Transmitter Attempt Using FFNN

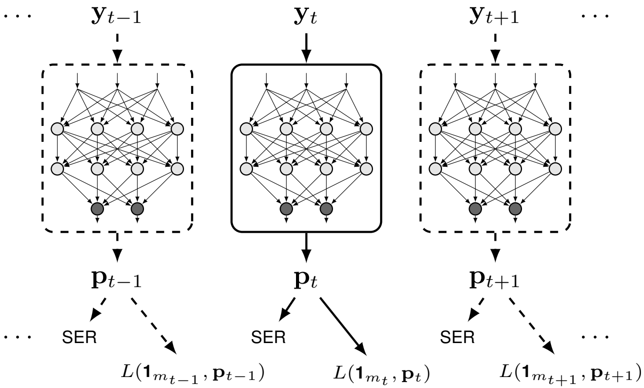


- Transform message m_t (from an alphabet containing M messages (e.g., a modulation alphabet)) into **one-hot**[5] vector $1_{m,t}$
- FNN encodes $1_{m,t}$ into n **oversampled waveform samples** $\mathbf{x}_t$
- Apply low-pass-filter (LPF, part of the channel model) to **smear** the waveforms

[KCT+18] B. Karanov, M. Chagnon, F. Thouin, T. Eriksson, H. Bülow, D. Lavery, P. Bayvel, L. S., "End-to-end Deep Learning of Optical Fiber Communications," *Journal of Lightwave Technology*, Oct. 2018

[5] A one-hot vector $\mathbf{1}_{m,t}$ of length M contains only zeros except a single "1" at position m_t

First Receiver Attempt Using FFNN

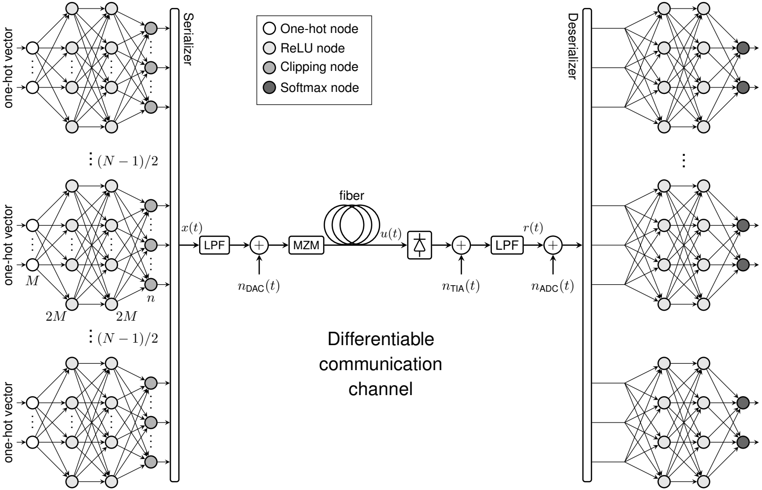


- Receiver input is chopped into blocks $\mathbf{y}_t$ of n received **samples**
- Samples are processed by an FFNN with a **softmax** output activation function ("●")
- Softmax output function generates a probability vector $\mathbf{p}_t$
- Decision according to most probable symbol
- Training using cross-entropy loss function

$$L(\mathbf{1}_{m_t}, \mathbf{p}_t) = -\log(p_{t,m_t})$$

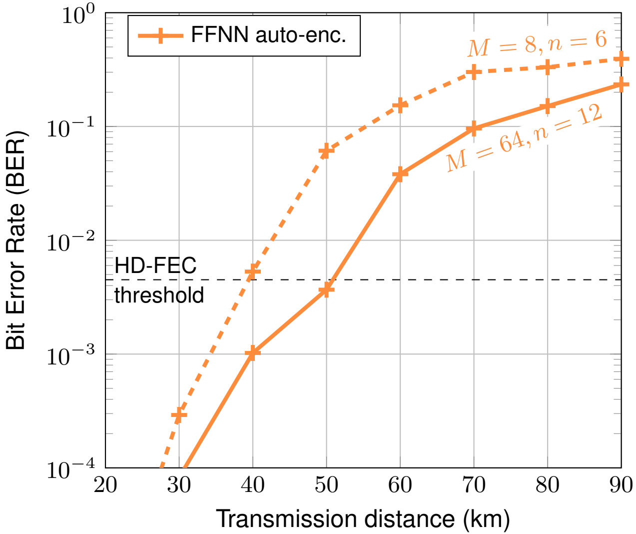
Putting Everything Together

Karlsruhe Institute of Technology



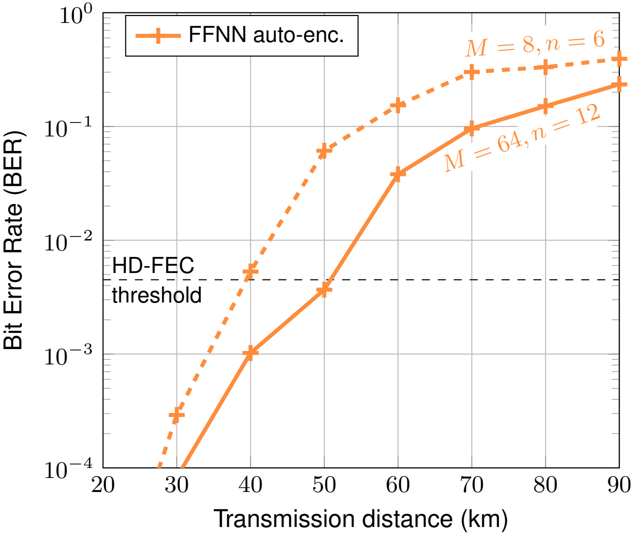
[KCT+18] B. Karanov, M. Chagnon, F. Thouin, T. Eriksson, H. Bülow, D. Lavery, P. Bayvel, L. S., "End-to-end Deep Learning of Optical Fiber Communications," *Journal of Lightwave Technology*, Oct. 2018

Simulation Results



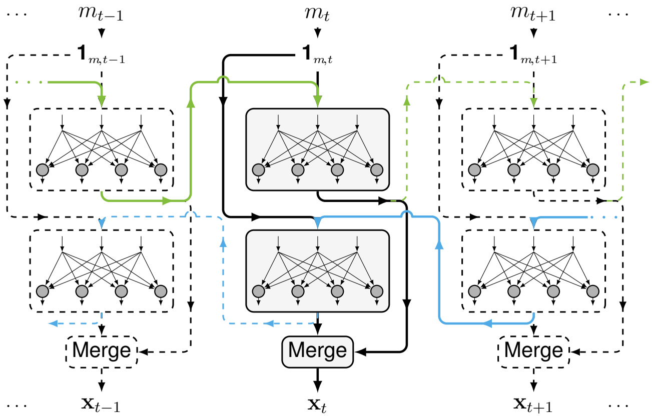
- State-of-the-art IM/DD system with 42 Gb/s net rate
- For small alphabets (M small) and neural networks, reach is limited
- Increasing the alphabet and the neural networks enables extra reach (multidimensional constellations) as more dispersion can be compensated for
- Increasing the the NN size (i.e., M and thus n) leads to unfeasibly large networks

Simulation Results



- State-of-the-art IM/DD system with 42 Gb/s net rate
- For small alphabets (M small) and neural networks, reach is limited
- Increasing the alphabet and the neural networks enables extra reach (multidimensional constellations) as more dispersion can be compensated for
- Increasing the the NN size (i.e., M and thus n) leads to unfeasibly large networks
- **New approach tailored to dispersive nature of channel needed**

Bidirectional Recurrent NN Based Transmitter

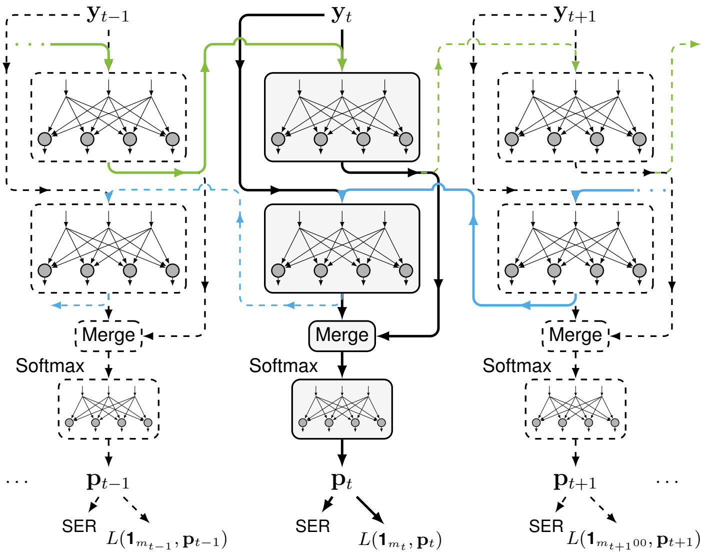


- Transmitter based on **bidirectional recurrent neural networks (BRNN)** [FG18] to account for memory due to chromatic dispersion
- Current message m_t jointly processed with previous and future messages vie **bidirectional recurrent neural network (BRNN)**
- Complexity gain using **small** networks inside RNN cells

[KLB+19] B. Karanov, D. Lavery, P. Bayvel, L. Schmalen, "End-to-end optimized transmission over dispersive intensity modulated channels using bidirectional recurrent neural networks," *Optics Express*, Jul. 2019

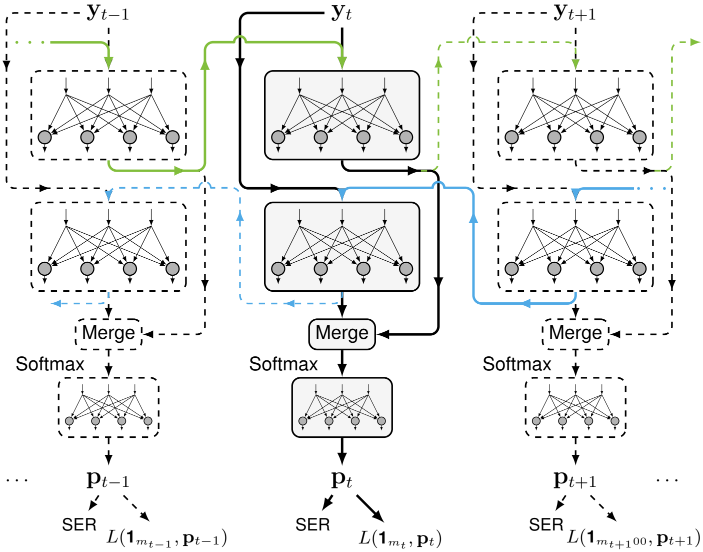
[FG18] N. Farsad, A. Goldsmith, "Neural network detection of data sequences in communication systems," *IEEE Trans. Signal Process*, 2018

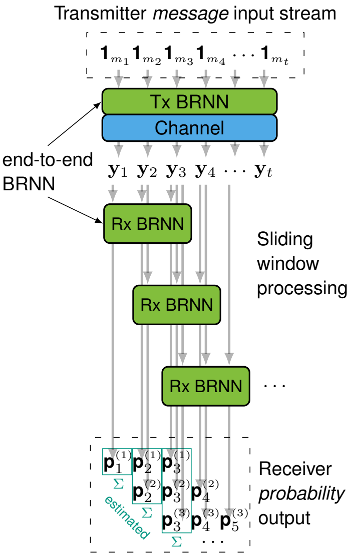
Bidrectional Recurrent NN Based Receiver



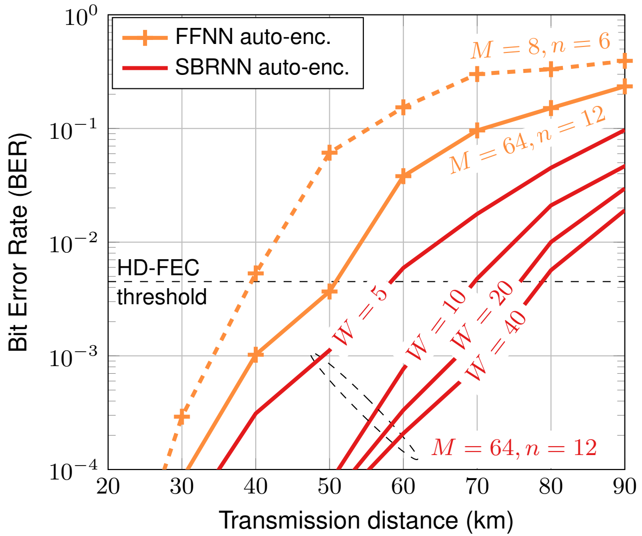
- Receiver structure similar to transmitter
- Additional **softmax** layer to generate probability vectors
- We use **sliding window** processing at the receiver to trade complexity and latency with performance
- A **window** of W stages is jointly processed at the receiver [FG18]

Bidirectional Recurrent NN Based Receiver





Simulation Results

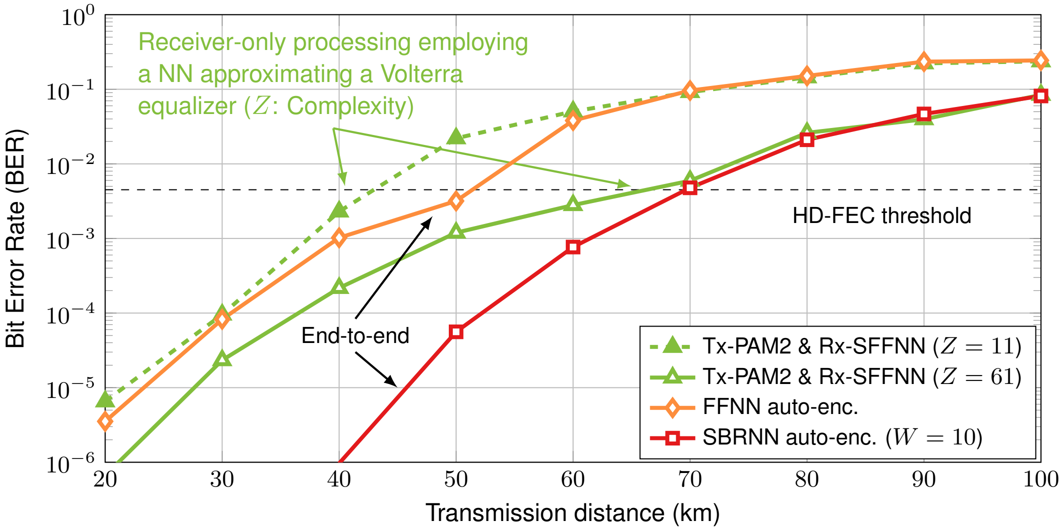


- SBRNN outperforms simple FFNN implementation at significantly **lower complexity**
- 20 km distance gain for $W = 10$
- SBRNN can **outperform** MLSD if complexity is constrained [KLA+19]
- Now: Comparison with pure receiver NN processing approximating a Volterra equalizer [Lyu15]

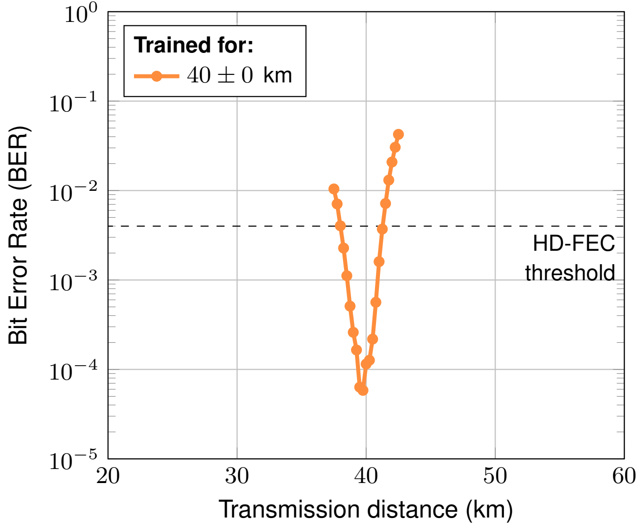
[KLA+19] B. Karanov, G. Liga, V. Aref, D. Lavery, P. Bayvel, and L. Schmalen, “Deep learning for communication over dispersive nonlinear channels: Performance and comparison with classical digital signal processing”, *Proc. Allerton Conf. on Commun., Control, and Computing*, 2019

[Lyu15] I. Lyubomirsky, “Machine learning equalization techniques for high speed PAM4 fiber optic communication systems,” *CS229 Final Project Report*, Stanford University, 2015

Simulation Results (2)

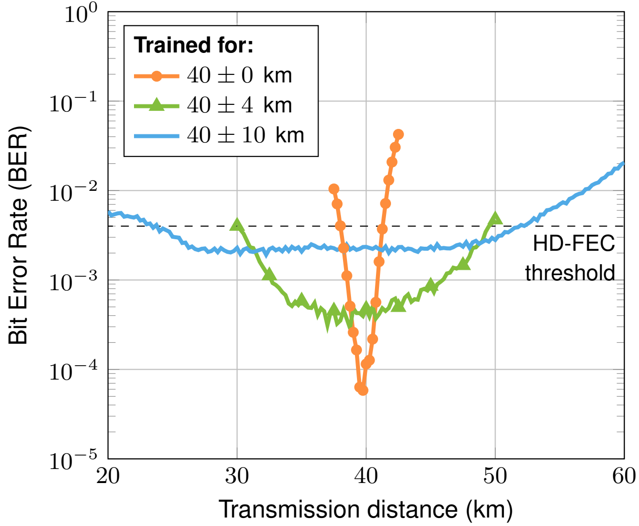


Simulation Results (3)



- **How well does the training generalize?**
- If no precautions are taken, not too well

Simulation Results (3)



- **How well does the training generalize?**
- If no precautions are taken, not too well
- **Multi-task learning** enables transceivers that operate on a range of distances **without reconfiguration** [KCT+18]
- Big step towards **distance agnostic** transceivers

[KCT+18] B. Karanov, M. Chagnon, F. Thouin, T. Eriksson, H. Bülow, D. Lavery, P. Bayvel, L. S., "End-to-end Deep Learning of Optical Fiber Communications," *Journal of Lightwave Technology*, Oct. 2018

Conclusions

- Machine learning suitable in cases where optimal algorithms are absent or computationally prohibitive
- Recurrent neural networks tailored to the dispersive properties of the channel
- Performance close to state-of-the-art benchmarks

Conclusions

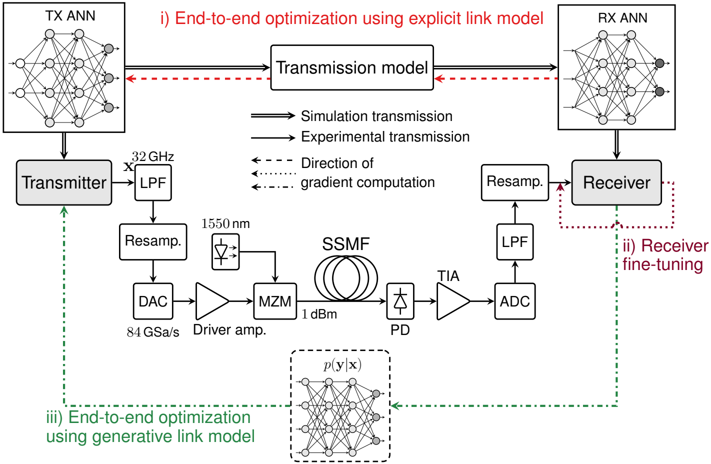
- Machine learning suitable in cases where optimal algorithms are absent or computationally prohibitive
- Recurrent neural networks tailored to the dispersive properties of the channel
- Performance close to state-of-the-art benchmarks

Take Home Messages

- We can harness vast ML toolbox being developed as we speak
- Configuration neural networks still hand-tuned using *educated guesses*
- Still requiring expert knowledge for fine-tuning NN
- End-to-end learning of complete transceiver very ambitious
- Start understanding impact of deep learning on parts of transceiver
- Still many open questions and problems to solve

Experimental Demonstration and Advances

- B. Karanov et al., "**End-to-End Learning in Optical Fiber Communications: Experimental Demonstration and Future Trends**," Talk **Th1D-1**, Thursday, December 10th



Questions?

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie-Skłodowska-Curie grant agreement No.676448