Model-Based Machine Learning for Physical-Layer Communication over Optical Fiber

Christian Häger

Department of Electrical Engineering, Chalmers University of Technology, Sweden

Summer School on "AI for Optical Networks & Neuromorphic Photonics for AI Acceleration"
September 6, 2021

FORCE
FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

CHALMERS

Thank You!

Henry D. Pfister
Duke

Christoffer Fougstedt
Chalmers (now: Ericsson)

Lars Svensson
Chalmers

Per Larsson-Edefors
Chalmers

Rick M. Bütler
TU/e (now: TU Delft)

Gabriele Liga
TU/e

Alex Alvarado
TU/e

Vinícius Oliari
TU/e

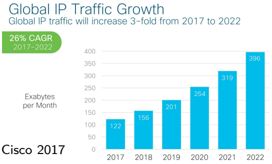
Sebastiaan Goossens
TU/e

Menno van den Hout
TU/e

Sjoerd van der Heide
TU/e

Chigo Okonkwo
TU/e

Motivation and Challenges



Source: Cisco 2017

- The COVID-19 pandemic has highlighted the importance of our global communication infrastructure
- Data traffic has been and will continue to grow exponentially
- Simply scaling current technology is not sustainable: fiber infrastructure would consume all world-wide electricity within less than 10 years

Motivation and Challenges

Source: Cisco 2017

Higher data rates?

More energy efficiency?

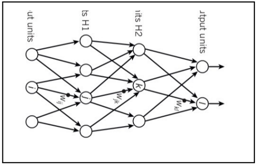
New functionalities?

- The COVID-19 pandemic has highlighted the importance of our global communication infrastructure
- Data traffic has been and will continue to grow exponentially
- Simply scaling current technology is not sustainable: fiber infrastructure would consume all world-wide electricity within less than 10 years

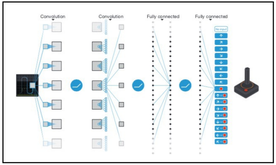
How can machine learning (ML) be used productively in communications to improve future systems?

This work started with a simple observation . . .

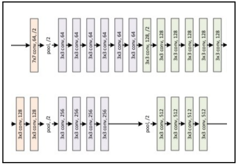
Deep Learning [LeCun et al., 2015]



Deep Q-Learning [Mnih et al., 2015]



ResNet [He et al., 2015]

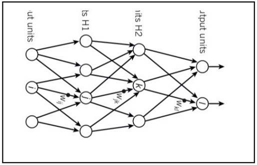


. . .

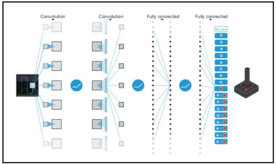
Multi-layer neural networks: impressive performance, countless applications

This work started with a simple observation . . .

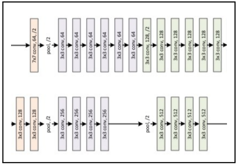
Deep Learning [LeCun et al., 2015]



Deep Q-Learning [Mnih et al., 2015]

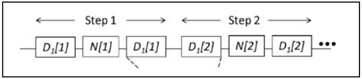


ResNet [He et al., 2015]

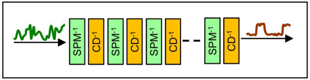


. . .

Multi-layer neural networks: impressive performance, countless applications



[Du and Lowery, 2010]



[Nakashima et al., 2017]

Split-step methods for solving the propagation equation in fiber-optics

Agenda

In this talk, we . . .

Agenda

In this talk, we . . .

1. show that multi-layer neural networks and the split-step method have the same functional form: both alternate linear and pointwise nonlinear steps

Agenda

In this talk, we . . .

1. show that multi-layer neural networks and the split-step method have the same functional form: both alternate linear and pointwise nonlinear steps
2. propose a physics-based machine-learning approach based on parameterizing the split-step method (no black-box neural networks)

Agenda

In this talk, we . . .

1. show that multi-layer neural networks and the split-step method have the same functional form: both alternate linear and pointwise nonlinear steps
2. propose a physics-based machine-learning approach based on parameterizing the split-step method (no black-box neural networks)
3. revisit hardware-efficient nonlinear equalization via learned digital backpropagation

Outline

1. Machine Learning and Neural Networks for Communications

2. Physics-Based Machine Learning for Fiber-Optic Communications

3. Learned Digital Backpropagation

4. Polarization-Dependent Effects

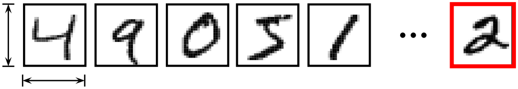
5. Conclusions

Outline

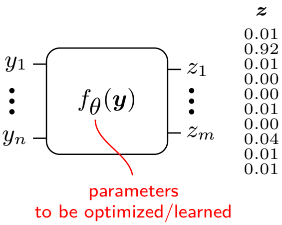
1. Machine Learning and Neural Networks for Communications
2. Physics-Based Machine Learning for Fiber-Optic Communications
3. Learned Digital Backpropagation
4. Polarization-Dependent Effects
5. Conclusions

Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)

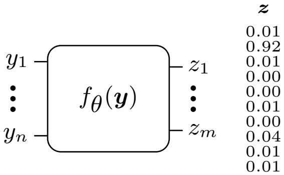


28×28 pixels $\implies n = 784$

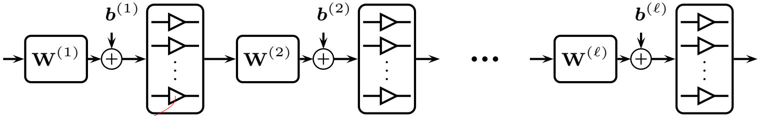


Supervised Learning

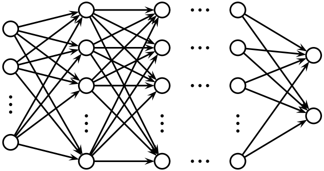
handwritten digit recognition (MNIST: 70,000 images)



How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



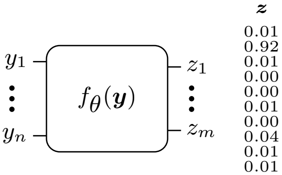
activation function



equivalent graph representation

Supervised Learning

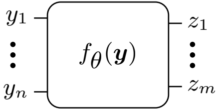
handwritten digit recognition (MNIST: 70,000 images)



How to optimize $\theta = \{\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(\ell)}, \boldsymbol{b}^{(1)}, \dots, \boldsymbol{b}^{(\ell)}\}$?

Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

How to optimize $\theta = \{\boldsymbol{W}^{(1)}, \dots, \boldsymbol{W}^{(\ell)}, \boldsymbol{b}^{(1)}, \dots, \boldsymbol{b}^{(\ell)}\}$?

Given a data set $\mathcal{D} = \{(\boldsymbol{y}^{(i)}, \boldsymbol{x}^{(i)})\}_{i=1}^{N}$, where $\boldsymbol{y}^{(i)}$ are model inputs and $\boldsymbol{x}^{(i)}$ are labels, we iteratively minimize

$$\frac{1}{|\mathcal{B}_k|} \sum_{(\boldsymbol{y},\boldsymbol{x}) \in \mathcal{B}_k} \mathcal{L}(f_\theta(\boldsymbol{y}), \boldsymbol{x}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

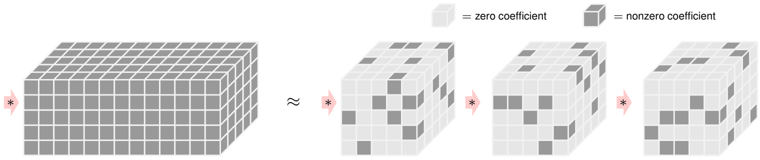
stochastic gradient descent

- $\mathcal{B}_k \subset \mathcal{D}$ and $|\mathcal{B}_k|$ is called the batch (or minibatch) size
- Typical loss function: mean squared error $\mathcal{L}(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\|^2$ (regression)
- λ is called the step size or learning rate

Why Deep Models?

Many possible answers

One advantage is complexity: deep computation graphs tend to be more parameter efficient than shallow graphs [Lin et al., 2017]

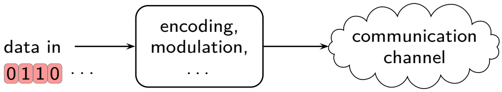


- Sparsity can emerge due to (approximate) factorization (even for linear models, e.g., FFT)
- Deep computation graphs allow for very simple elementary steps
- Deep models typically have many "good" parameter configurations that are close to each other $\Longrightarrow$ robustness to, e.g., quantization noise

Physical-Layer Design: Conventional vs. Machine Learning

- Conventional: handcrafted DSP blocks based on mathematical modeling

Physical-Layer Design: Conventional vs. Machine Learning



- Conventional: handcrafted DSP blocks based on mathematical modeling

Physical-Layer Design: Conventional vs. Machine Learning

- Conventional: handcrafted DSP blocks based on mathematical modeling

Physical-Layer Design: Conventional vs. Machine Learning

- Conventional: handcrafted DSP blocks based on mathematical modeling
  - Model deficiency: no good model might be available
  - Algorithm deficiency: infeasible algorithms may require simplifications

Physical-Layer Design: Conventional vs. Machine Learning

- Conventional: handcrafted DSP blocks based on mathematical modeling
  - Model deficiency: no good model might be available
  - Algorithm deficiency: infeasible algorithms may require simplifications
- Use function approximators and learn parameter configurations θ from data

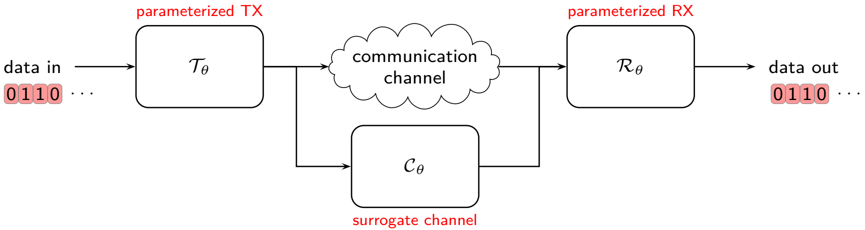
[Shen and Lau, 2011], Fiber nonlinearity compensation using extreme learning machine for DSP-based ..., (*OECC*)
[Giacoumidis et al., 2015], Fiber nonlinearity-induced penalty reduction in CO-OFDM by ANN-based ..., (*Opt. Lett.*)
...

Physical-Layer Design: Conventional vs. Machine Learning

- Conventional: handcrafted DSP blocks based on mathematical modeling
  - Model deficiency: no good model might be available
  - Algorithm deficiency: infeasible algorithms may require simplifications
- Use function approximators and learn parameter configurations θ from data
- Joint transmitter–receiver learning via autoencoder [O'Shea and Hoydis, 2017]

[Karanov et al., 2018], End-to-end deep learning of optical fiber communications (*J. Lightw. Technol.*)
[Li et al., 2018], Achievable information rates for nonlinear fiber communication via end-to-end autoencoder learning, (*ECOC*)
. . .

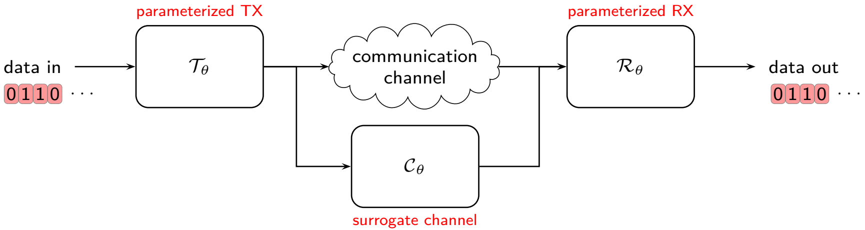
Physical-Layer Design: Conventional vs. Machine Learning



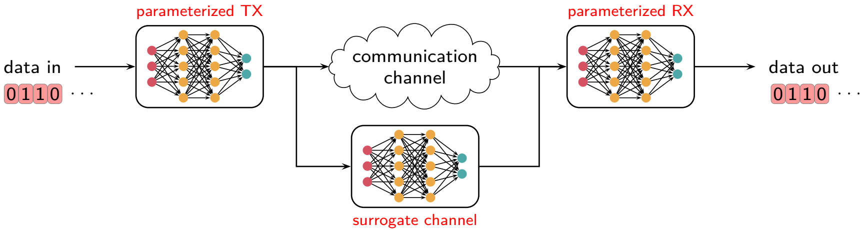
- Conventional: handcrafted DSP blocks based on mathematical modeling
  - Model deficiency: no good model might be available
  - Algorithm deficiency: infeasible algorithms may require simplifications
- Use function approximators and learn parameter configurations θ from data
- Joint transmitter–receiver learning via autoencoder [O'Shea and Hoydis, 2017]
- Surrogate channel models for gradient-based TX training

[O'Shea et al., 2018], Approximating the void: Learning stochastic channel models from observation with variational GANs, (*arXiv*)
[Ye et al., 2018], Channel agnostic end-to-end learning based communication systems with conditional GAN, (*arXiv*)
...

Physical-Layer Design: Conventional vs. Machine Learning



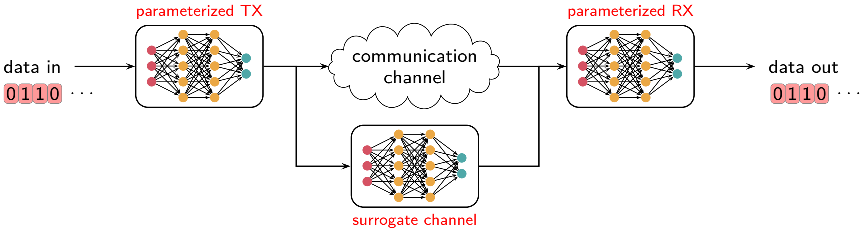
Physical-Layer Design: Conventional vs. Machine Learning



Using (deep) neural networks for $\mathcal{T}_\theta, \mathcal{R}_\theta, \mathcal{C}_\theta$? Possible, but ...

- How to choose the network architecture (#layers, activation function)?
- How to limit the number of parameters (complexity)?
- How to interpret the solutions? Any insight gained?
- ...

Physical-Layer Design: Conventional vs. Machine Learning



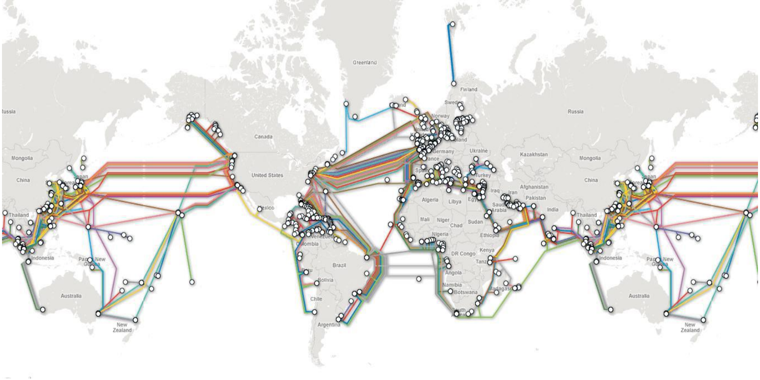
Using (deep) neural networks for $\mathcal{T}_\theta, \mathcal{R}_\theta, \mathcal{C}_\theta$? Possible, but ...

- How to choose the network architecture (#layers, activation function)? ✗
- How to limit the number of parameters (complexity)? ✗
- How to interpret the solutions? Any insight gained? ✗
- ...

Our contribution: designing "neural-network-like" machine-learning models by exploiting the underlying physics of the propagation.

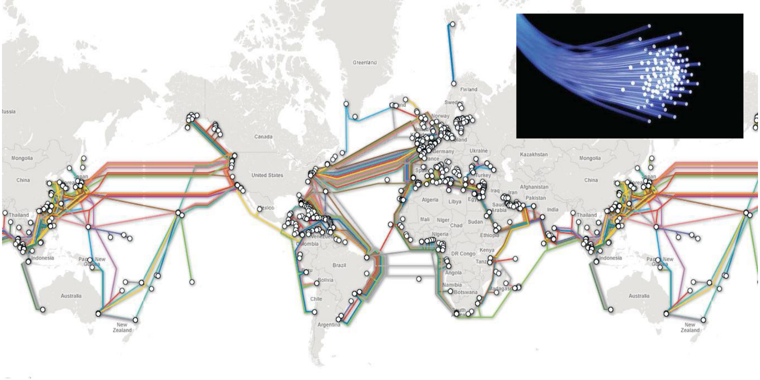
Outline

Fiber-Optic Communications



Fiber-optic systems enable data traffic over very long distances connecting cities, countries, and continents.

Fiber-Optic Communications

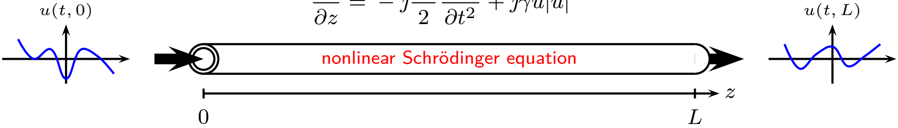


Fiber-optic systems enable data traffic over very long distances connecting cities, countries, and continents.

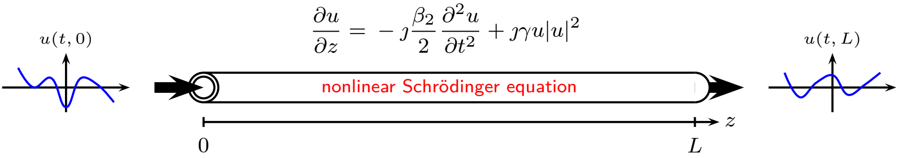
- Dispersion: different wavelengths travel at different speeds (linear)
- Kerr effect: refractive index changes with signal intensity (nonlinear)

Nonlinear Fiber Channel Modeling

$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



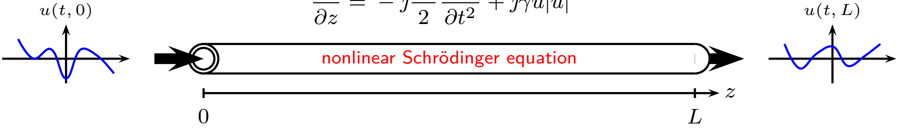
Nonlinear Fiber Channel Modeling



- Partial differential equation without general closed-form solution

Nonlinear Fiber Channel Modeling

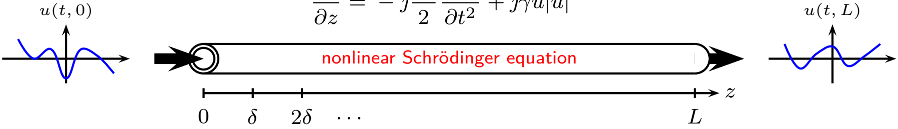
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



- Partial differential equation without general closed-form solution
- Sampling over a fixed time interval: $\boldsymbol{x} \in \mathbb{C}^n$ (input), $\boldsymbol{y} \in \mathbb{C}^n$ (output)

Nonlinear Fiber Channel Modeling

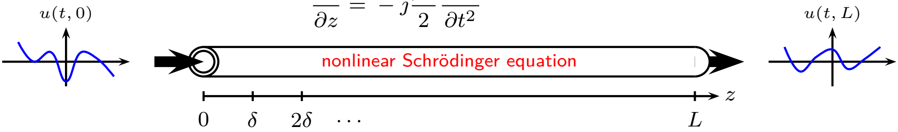
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



- Partial differential equation without general closed-form solution
- Sampling over a fixed time interval: $\boldsymbol{x} \in \mathbb{C}^n$ (input), $\boldsymbol{y} \in \mathbb{C}^n$ (output)
- Split-step method with M steps ($\delta = L/M$):

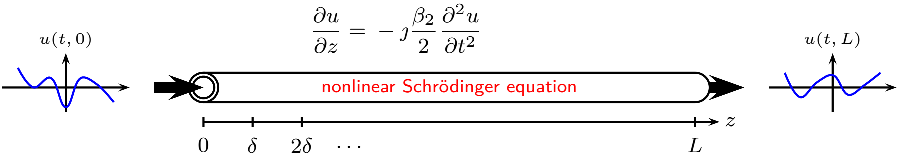
Nonlinear Fiber Channel Modeling

$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2}$$

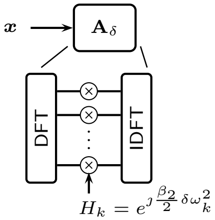


- Partial differential equation without general closed-form solution
- Sampling over a fixed time interval: $\boldsymbol{x} \in \mathbb{C}^n$ (input), $\boldsymbol{y} \in \mathbb{C}^n$ (output)
- Split-step method with M steps ($\delta = L/M$):

Nonlinear Fiber Channel Modeling



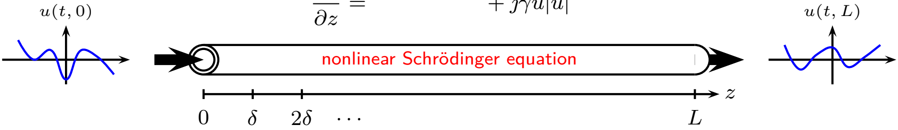
- Partial differential equation without general closed-form solution
- Sampling over a fixed time interval: $\boldsymbol{x} \in \mathbb{C}^n$ (input), $\boldsymbol{y} \in \mathbb{C}^n$ (output)
- Split-step method with M steps ($\delta = L/M$):



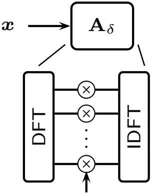
dispersion (all-pass filter)

Nonlinear Fiber Channel Modeling

$$\frac{\partial u}{\partial z} = \qquad + \jmath\gamma u|u|^2$$



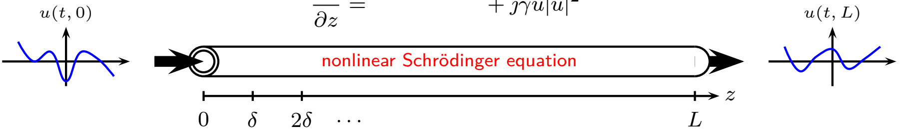
- Partial differential equation without general closed-form solution
- Sampling over a fixed time interval: $\boldsymbol{x} \in \mathbb{C}^n$ (input), $\boldsymbol{y} \in \mathbb{C}^n$ (output)
- Split-step method with M steps ($\delta = L/M$):



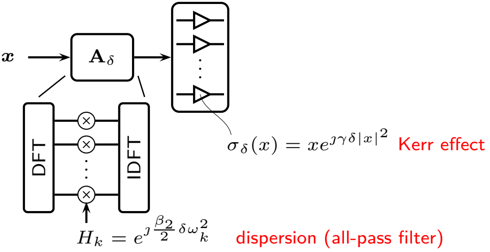
$H_k = e^{\jmath \frac{\beta_2}{2} \delta \omega_k^2}$ dispersion (all-pass filter)

Nonlinear Fiber Channel Modeling

$$\frac{\partial u}{\partial z} = \qquad + \jmath\gamma u|u|^2$$

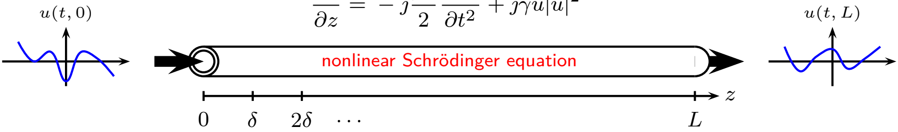


- Partial differential equation without general closed-form solution
- Sampling over a fixed time interval: $\boldsymbol{x} \in \mathbb{C}^n$ (input), $\boldsymbol{y} \in \mathbb{C}^n$ (output)
- Split-step method with M steps ($\delta = L/M$):

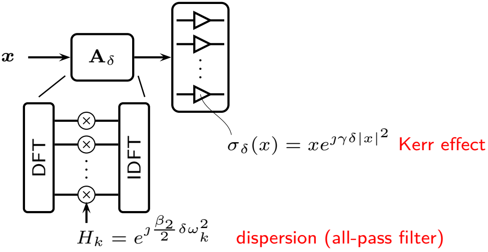


Nonlinear Fiber Channel Modeling

$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$

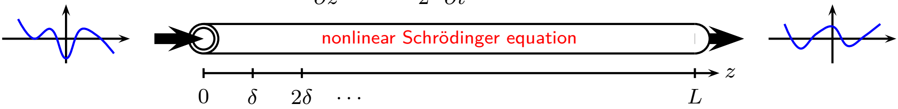


- Partial differential equation without general closed-form solution
- Sampling over a fixed time interval: $\boldsymbol{x} \in \mathbb{C}^n$ (input), $\boldsymbol{y} \in \mathbb{C}^n$ (output)
- Split-step method with M steps ($\delta = L/M$):

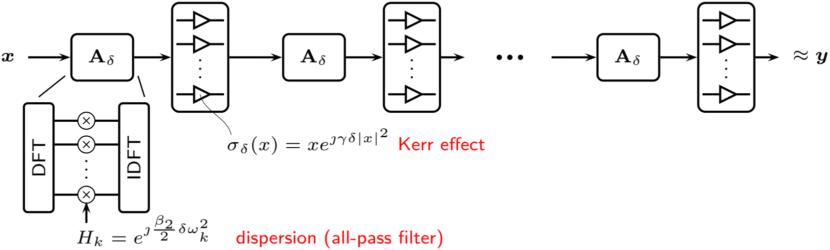


Nonlinear Fiber Channel Modeling

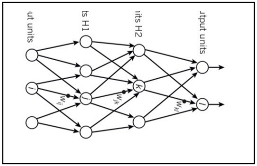
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2}\frac{\partial^2 u}{\partial t^2} + \jmath\gamma u|u|^2$$



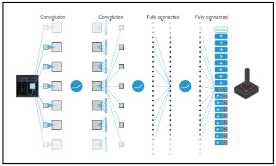
- Partial differential equation without general closed-form solution
- Sampling over a fixed time interval: $\boldsymbol{x} \in \mathbb{C}^n$ (input), $\boldsymbol{y} \in \mathbb{C}^n$ (output)
- Split-step method with M steps ($\delta = L/M$):



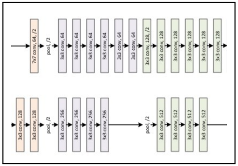
Deep Learning [LeCun et al., 2015]



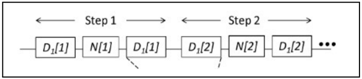
Deep Q-Learning [Mnih et al., 2015]



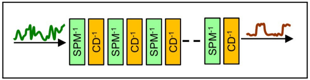
ResNet [He et al., 2015]



...

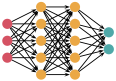


[Du and Lowery, 2010]

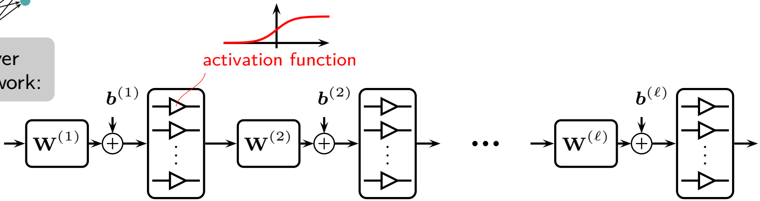


[Nakashima et al., 2017]

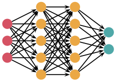
The Main Idea

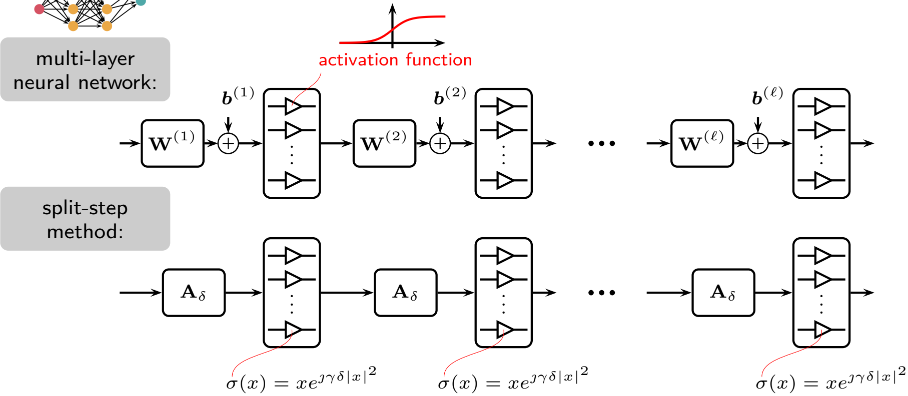


multi-layer
neural network:



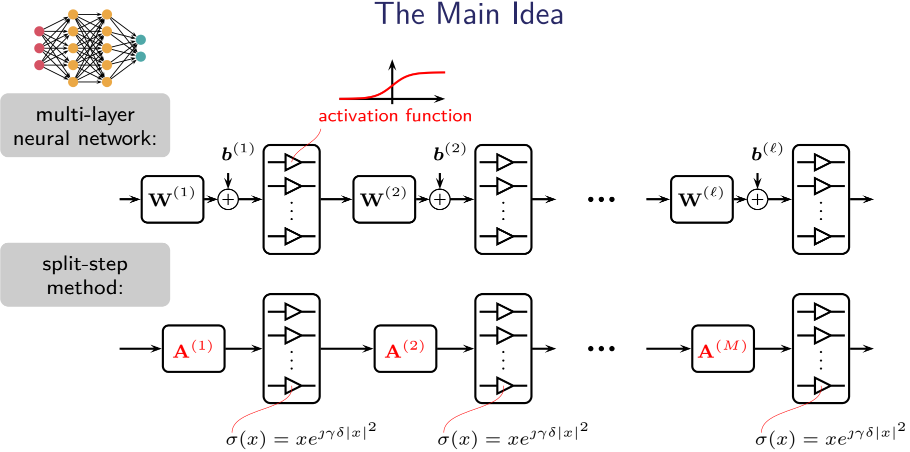
The Main Idea





- This almost looks like a deep neural net!

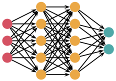
The Main Idea



- This almost looks like a deep neural net!
- Parameterize all linear steps: f_θ with $\theta = \{\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(M)}\}$

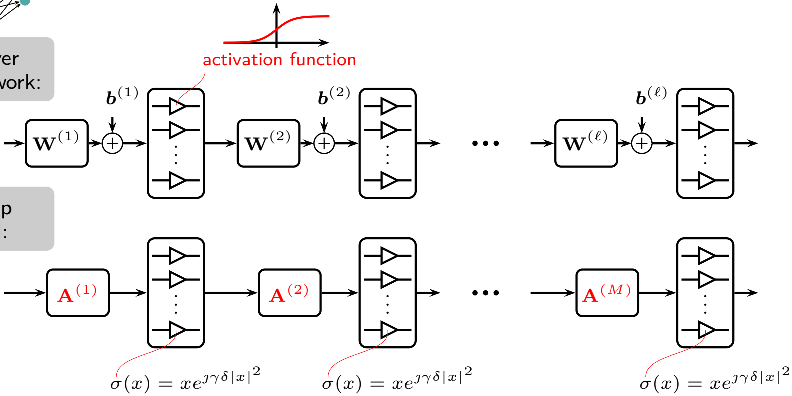
[Häger & Pfister, 2018], Nonlinear Interference Mitigation via Deep Neural Networks, *(OFC)*
[Häger & Pfister, 2021], Physics-Based Deep Learning for Fiber-Optic Communication Systems, *IEEE J. Sel. Areas Commun.*

The Main Idea



multi-layer neural network:

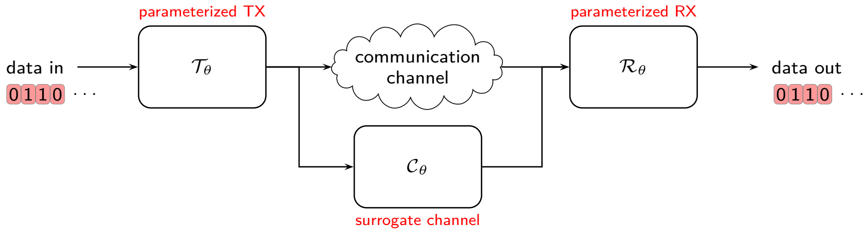
split-step method:



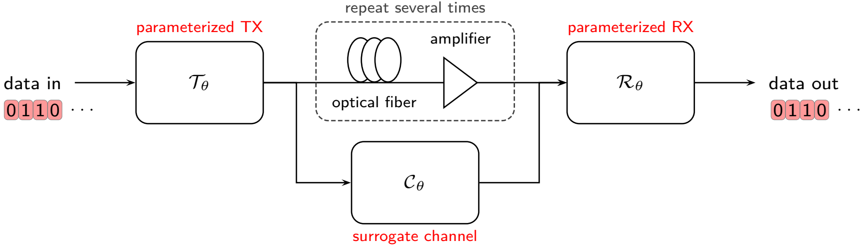
- This almost looks like a deep neural net!
- Parameterize all linear steps: f_θ with $\theta = \{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(M)}\}$
- Special cases: step-size optimization, nonlinear operator “placement”, ...

[Häger & Pfister, 2018], Nonlinear Interference Mitigation via Deep Neural Networks, *(OFC)*
[Häger & Pfister, 2021], Physics-Based Deep Learning for Fiber-Optic Communication Systems, *IEEE J. Sel. Areas Commun.*

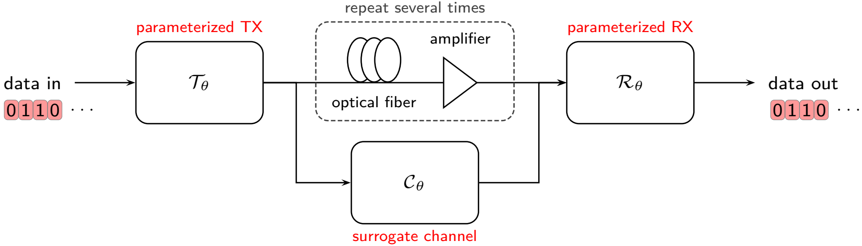
Possible Applications



Possible Applications

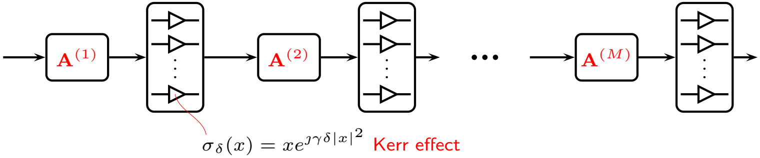


Possible Applications



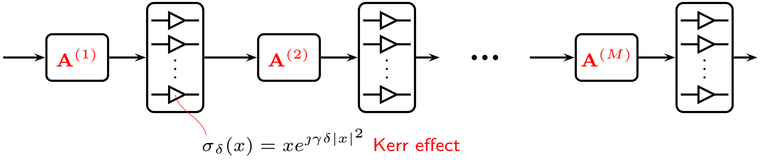
- Channel $\mathcal{C}_\theta$: fine-tune model based on experimental data, reduce simulation time [Leibrich and Rosenkranz, 2003], [Li et al., 2005]
- Receiver $\mathcal{R}_\theta$: nonlinear equalization (focus in this talk)
- Transmitter $\mathcal{T}_\theta$: digital pre-distortion [Essiambre and Winzer, 2005], [Roberts et al., 2006], "split" nonlinearity compensation [Lavery et al., 2016]

Potential Benefits



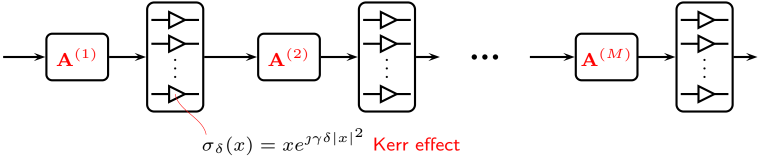
- How to choose the network architecture (#layers, activation function)?
- How to limit the number of parameters (complexity)?
- How to interpret the solutions? Any insight gained?

Potential Benefits



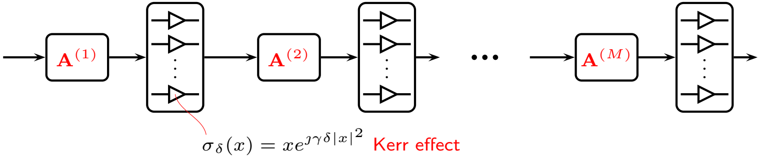
- How to choose the network architecture (#layers, activation function)? ✓
  - Activation function is fixed; number of layers = number of steps
  - Hidden feature representations ≈ signal at intermediate fiber locations
  - Parameter initialization based on conventional split-step method
- How to limit the number of parameters (complexity)?
- How to interpret the solutions? Any insight gained?

Potential Benefits



- How to choose the network architecture (#layers, activation function)? ✓
  - Activation function is fixed; number of layers = number of steps
  - Hidden feature representations ≈ signal at intermediate fiber locations
  - Parameter initialization based on conventional split-step method
- How to limit the number of parameters (complexity)? ✓
  - Propagation dynamics are "embedded" in the model through nonlinear steps
  - Filter symmetry can be enforced in the linear steps
  - Model compression (e.g., parameter pruning, quantization)
- How to interpret the solutions? Any insight gained?

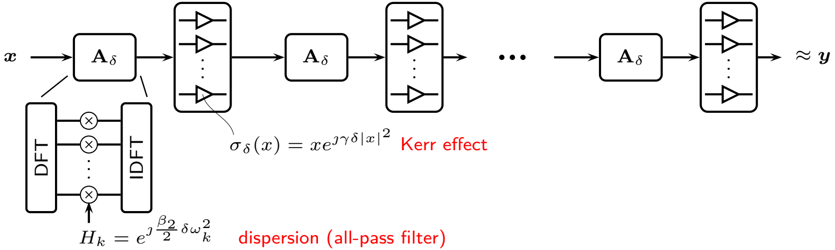
Potential Benefits



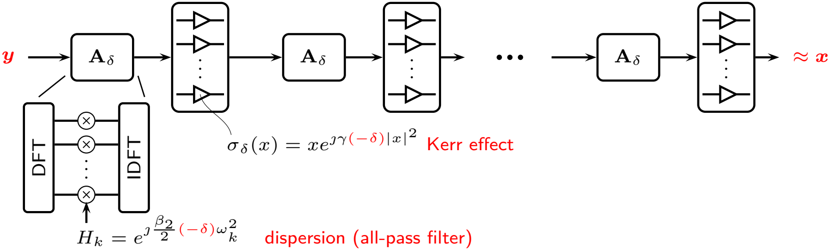
- How to choose the network architecture (#layers, activation function)? ✓
  - Activation function is fixed; number of layers = number of steps
  - Hidden feature representations ≈ signal at intermediate fiber locations
  - Parameter initialization based on conventional split-step method
- How to limit the number of parameters (complexity)? ✓
  - Propagation dynamics are "embedded" in the model through nonlinear steps
  - Filter symmetry can be enforced in the linear steps
  - Model compression (e.g., parameter pruning, quantization)
- How to interpret the solutions? Any insight gained? ✓
  - Learned parameter configurations are interpretable
  - Satisfactory explanations for benefits over previous handcrafted solutions

Outline

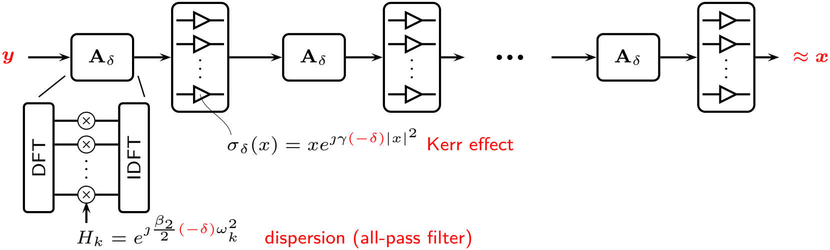
Digital Backpropagation



Digital Backpropagation

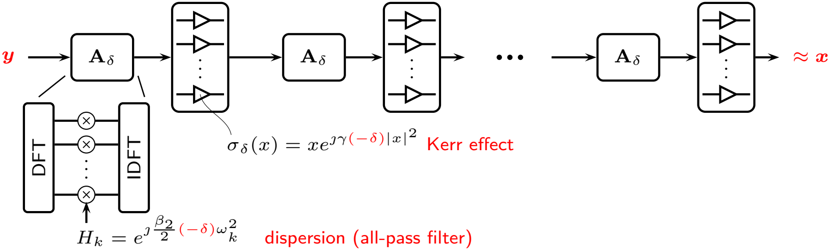


Digital Backpropagation



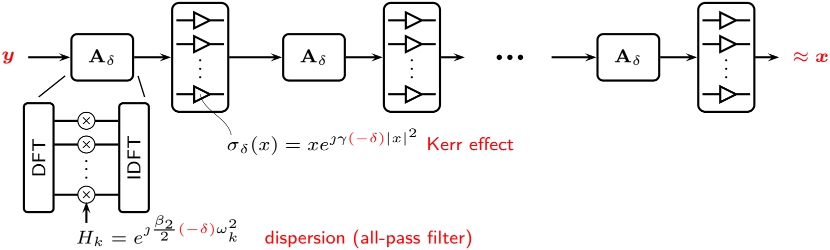
- Fiber with negated parameters ($\beta_2 \to -\beta_2$, $\gamma \to -\gamma$) would perform perfect channel inversion [Paré et al., 1996] (ignoring attenuation)

Digital Backpropagation



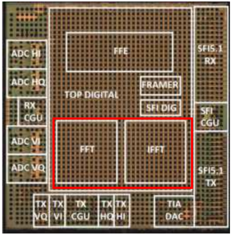
- Fiber with negated parameters ($\beta_2 \to -\beta_2$, $\gamma \to -\gamma$) would perform perfect channel inversion [Paré et al., 1996] (ignoring attenuation)
- Digital backpropagation: invert a partial differential equation in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]

Digital Backpropagation

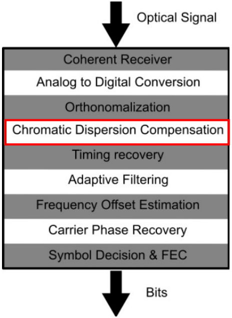


- Fiber with negated parameters ($\beta_2 \to -\beta_2$, $\gamma \to -\gamma$) would perform perfect channel inversion [Paré et al., 1996] (ignoring attenuation)
- Digital backpropagation: invert a partial differential equation in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]
- Widely considered to be impractical (too complex): linear equalization is already one of the most power-hungry DSP blocks in coherent receivers

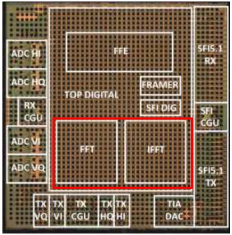
Real-Time Digital Backpropagation



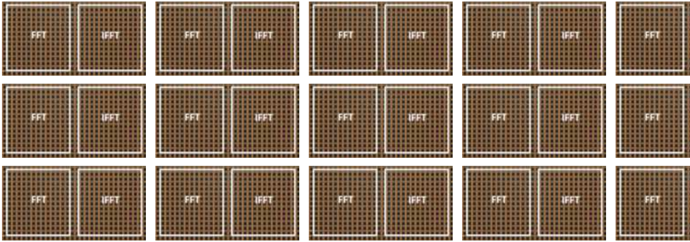
[Crivelli et al., 2014]



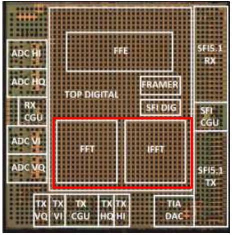
Real-Time Digital Backpropagation



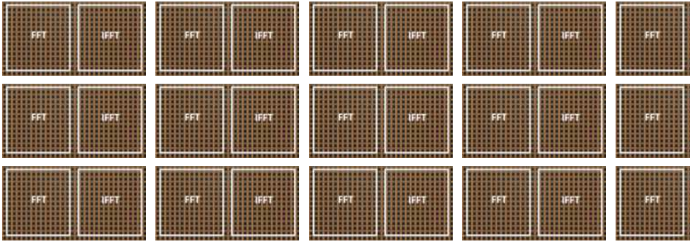
[Crivelli et al., 2014]



Real-Time Digital Backpropagation

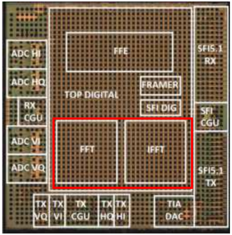


[Crivelli et al., 2014]

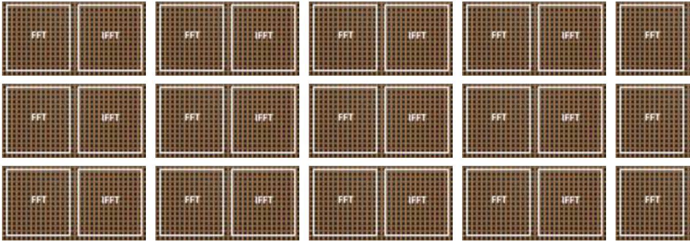


- Complexity increases with the number of steps M $\Longrightarrow$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)

Real-Time Digital Backpropagation

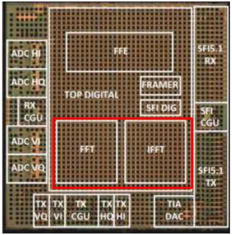


[Crivelli et al., 2014]

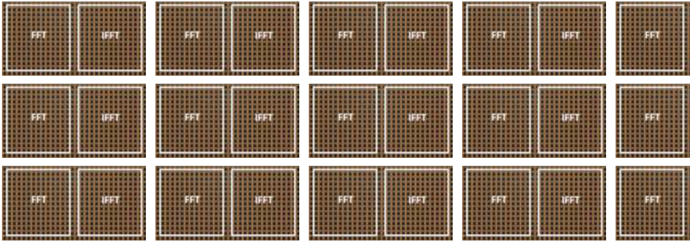


- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)
- Intuitive, but . . .

Real-Time Digital Backpropagation

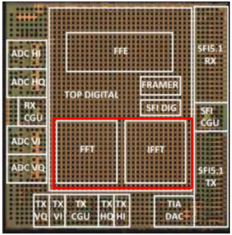


[Crivelli et al., 2014]

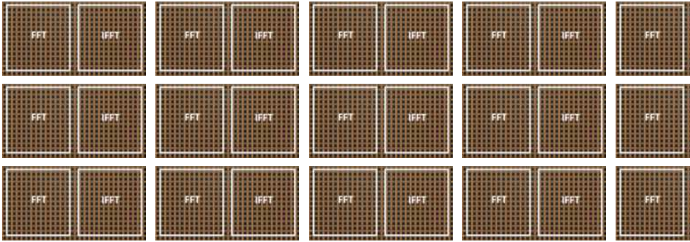


- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], ...)
- Intuitive, but ... this flattens a deep (multi-layer) computation graph

Real-Time Digital Backpropagation



[Crivelli et al., 2014]



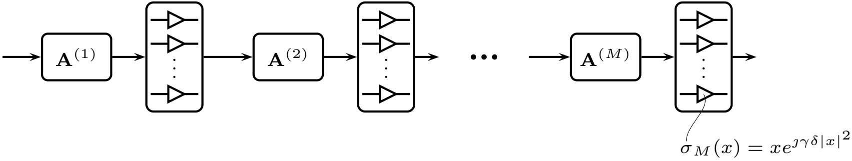
- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)
- Intuitive, but . . . this flattens a deep (multi-layer) computation graph

Our approach: many steps but model compression

Joint optimization, pruning, and quantization of all linear steps $\implies$ hardware-efficient digital backpropagation

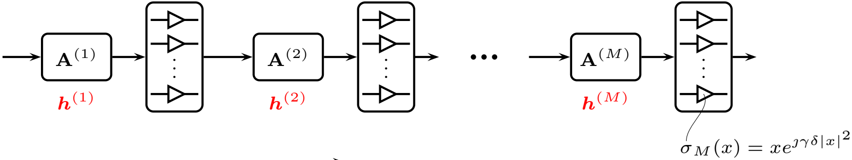
Learned Digital Backpropagation

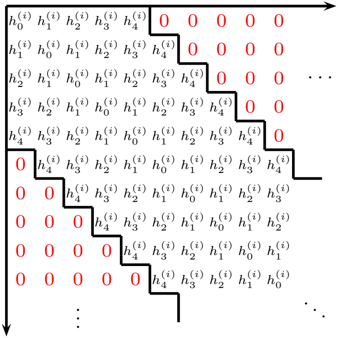
TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:

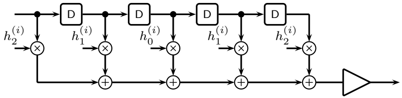


Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



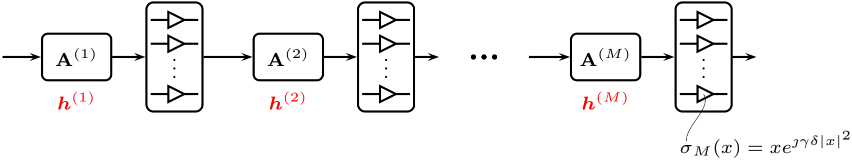




finite impulse response (FIR) filter
complex & symmetric coefficients

Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



Deep learning of all FIR filter coefficients $\theta = \{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(M)}\}$:

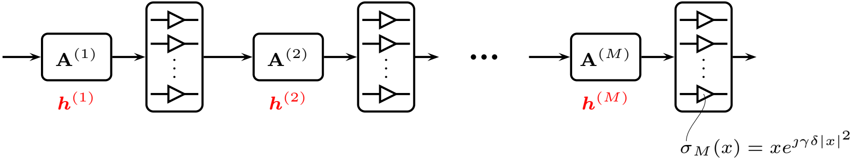
$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

mean squared error

Adam optimizer, fixed learning rate

Learned Digital Backpropagation

TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:



Deep learning of all FIR filter coefficients $\theta = \{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(M)}\}$:

$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

mean squared error — Adam optimizer, fixed learning rate

Iteratively prune (set to 0) outermost filter taps during gradient descent

Iterative Filter Tap Pruning

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} \\ \boldsymbol{h}^{(2)} \\ \vdots \\ \boldsymbol{h}^{(M)} \end{cases}$$

Iterative Filter Tap Pruning

starting length $2K' + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; h^{(1)}_{K'} \; \cdots \; h^{(1)}_{K} \; \cdots \; h^{(1)}_{1} \; h^{(1)}_{0} \; h^{(1)}_{1} \; \cdots \; h^{(1)}_{K} \; \cdots \; h^{(1)}_{K'} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; h^{(2)}_{K'} \; \cdots \; h^{(2)}_{K} \; \cdots \; h^{(2)}_{1} \; h^{(2)}_{0} \; h^{(2)}_{1} \; \cdots \; h^{(2)}_{K} \; \cdots \; h^{(2)}_{K'} \;) & \text{step } 2 \\ \vdots \\ \boldsymbol{h}^{(M)} = (\; h^{(M)}_{K'} \; \cdots \; h^{(M)}_{K} \; \cdots \; h^{(M)}_{1} \; h^{(M)}_{0} \; h^{(M)}_{1} \; \cdots \; h^{(M)}_{K} \; \cdots \; h^{(M)}_{K'} \;) & \text{step } M \end{cases}$$

Iterative Filter Tap Pruning

$\longleftarrow$ starting length $2K' + 1$ $\longrightarrow$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (h_{K'}^{(1)} \cdots h_{K}^{(1)} \cdots h_{1}^{(1)}\ h_{0}^{(1)}\ h_{1}^{(1)} \cdots h_{K}^{(1)} \cdots h_{K'}^{(1)}) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (h_{K'}^{(2)} \cdots h_{K}^{(2)} \cdots h_{1}^{(2)}\ h_{0}^{(2)}\ h_{1}^{(2)} \cdots h_{K}^{(2)} \cdots h_{K'}^{(2)}) & \text{step } 2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots & \\ \boldsymbol{h}^{(M)} = (h_{K'}^{(M)} \cdots h_{K}^{(M)} \cdots h_{1}^{(M)}\ h_{0}^{(M)}\ h_{1}^{(M)} \cdots h_{K}^{(M)} \cdots h_{K'}^{(M)}) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; h_{K'}^{(1)} \; \cdots \; h_K^{(1)} \; \cdots \; h_1^{(1)} \; h_0^{(1)} \; h_1^{(1)} \; \cdots \; h_K^{(1)} \; \cdots \; h_{K'}^{(1)} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; h_{K'}^{(2)} \; \cdots \; h_K^{(2)} \; \cdots \; h_1^{(2)} \; h_0^{(2)} \; h_1^{(2)} \; \cdots \; h_K^{(2)} \; \cdots \; h_{K'}^{(2)} \;) & \text{step } 2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots & \\ \boldsymbol{h}^{(M)} = (\; h_{K'}^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_1^{(M)} \; h_0^{(M)} \; h_1^{(M)} \; \cdots \; h_K^{(M)} \; \cdots \; h_{K'}^{(M)} \;) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; h^{(1)}_{K'} \; \cdots \; h^{(1)}_{K} \; \cdots \; h^{(1)}_{1} \; h^{(1)}_{0} \; h^{(1)}_{1} \; \cdots \; h^{(1)}_{K} \; \cdots \; h^{(1)}_{K'} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; h^{(2)}_{K'} \; \cdots \; h^{(2)}_{K} \; \cdots \; h^{(2)}_{1} \; h^{(2)}_{0} \; h^{(2)}_{1} \; \cdots \; h^{(2)}_{K} \; \cdots \; h^{(2)}_{K'} \;) & \text{step } 2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots & \\ \boldsymbol{h}^{(M)} = (\; h^{(M)}_{K'} \; \cdots \; h^{(M)}_{K} \; \cdots \; h^{(M)}_{1} \; h^{(M)}_{0} \; h^{(M)}_{1} \; \cdots \; h^{(M)}_{K} \; \cdots \; h^{(M)}_{K'} \;) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\; \cancel{h_{K'}^{(1)}} \; \cdots \; h_{K}^{(1)} \; \cdots \; h_{1}^{(1)} \; h_{0}^{(1)} \; h_{1}^{(1)} \; \cdots \; h_{K}^{(1)} \; \cdots \; \cancel{h_{K'}^{(1)}} \;) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\; \cancel{h_{K'}^{(2)}} \; \cdots \; h_{K}^{(2)} \; \cdots \; h_{1}^{(2)} \; h_{0}^{(2)} \; h_{1}^{(2)} \; \cdots \; h_{K}^{(2)} \; \cdots \; \cancel{h_{K'}^{(2)}} \;) & \text{step } 2 \\ \vdots \qquad \vdots \qquad \vdots \qquad \vdots & \\ \boldsymbol{h}^{(M)} = (\; h_{K'}^{(M)} \; \cdots \; h_{K}^{(M)} \; \cdots \; h_{1}^{(M)} \; h_{0}^{(M)} \; h_{1}^{(M)} \; \cdots \; h_{K}^{(M)} \; \cdots \; h_{K'}^{(M)} \;) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]

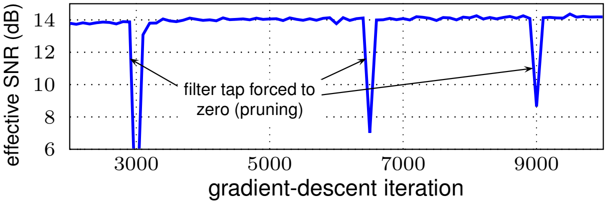
Iterative Filter Tap Pruning

starting length $2K'+1$

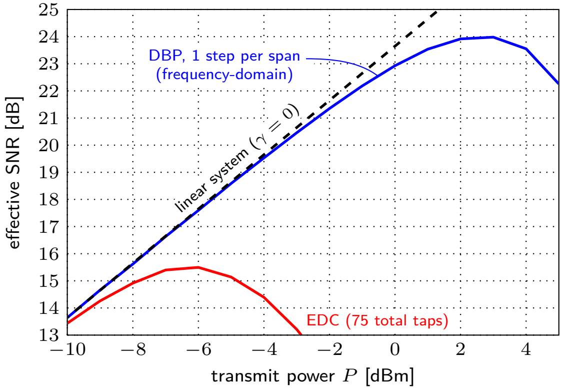
target length $2K+1$

$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\ \cancel{h_{K'}^{(1)}} \ \cdots \ h_K^{(1)} \ \cdots \ h_1^{(1)} \ h_0^{(1)} \ h_1^{(1)} \ \cdots \ h_K^{(1)} \ \cdots \ \cancel{h_{K'}^{(1)}} \ ) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\ \cancel{h_{K'}^{(2)}} \ \cdots \ h_K^{(2)} \ \cdots \ h_1^{(2)} \ h_0^{(2)} \ h_1^{(2)} \ \cdots \ h_K^{(2)} \ \cdots \ \cancel{h_{K'}^{(2)}} \ ) & \text{step } 2 \\ \vdots & \\ \boldsymbol{h}^{(M)} = (\ h_{K'}^{(M)} \ \cdots \ h_K^{(M)} \ \cdots \ h_1^{(M)} \ h_0^{(M)} \ h_1^{(M)} \ \cdots \ h_K^{(M)} \ \cdots \ h_{K'}^{(M)} \ ) & \text{step } M \end{cases}$$

- Initially: constrained least-squares coefficients (LS-CO) [Sheikh et al., 2016]
- Typical learning curve:



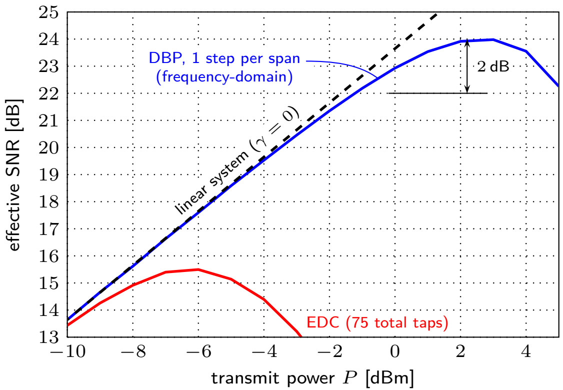
Revisiting Ip and Kahn (2008)



Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

Revisiting Ip and Kahn (2008)

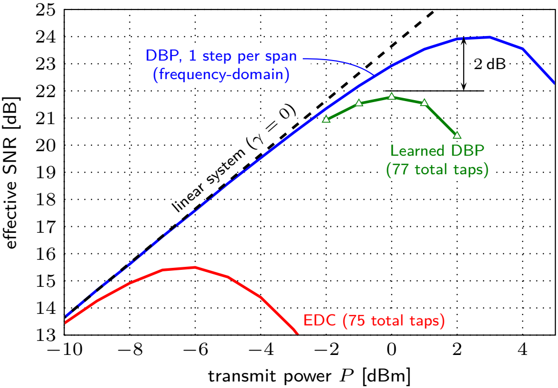


Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

- $\gg 1000$ total taps (70 taps/step) $\implies$ $> 100\times$ complexity of EDC

Revisiting Ip and Kahn (2008)

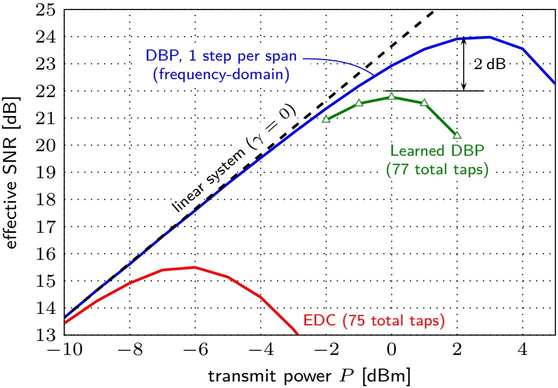


Parameters similar to [Ip and Kahn, 2008]:

- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

- $\gg 1000$ total taps (70 taps/step) $\implies$ $> 100\times$ complexity of EDC
- Learned approach uses only 77 total taps: alternate 5 and 3 taps/step and use different filter coefficients in all steps [Häger and Pfister, 2018a]

Revisiting Ip and Kahn (2008)

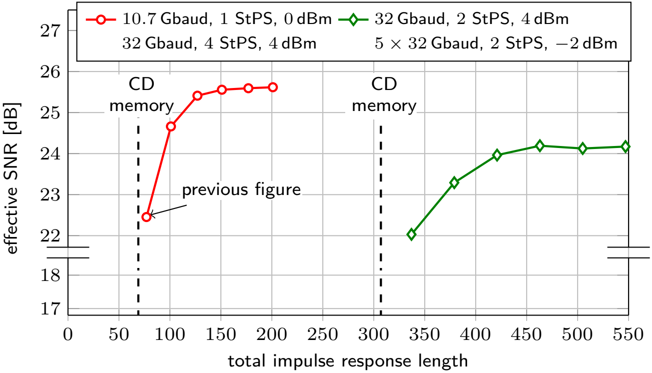


Parameters similar to [Ip and Kahn, 2008]:

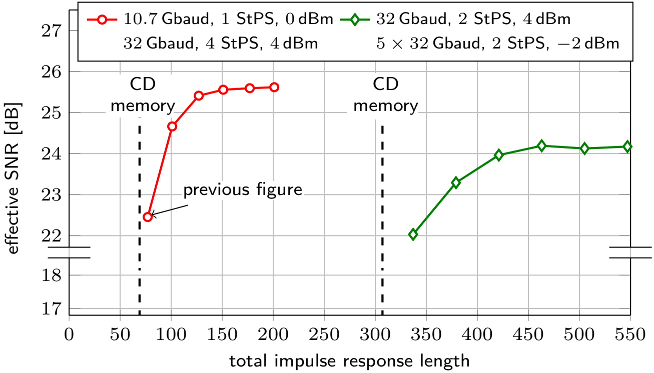
- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

- $\gg 1000$ total taps (70 taps/step) $\implies$ $> 100\times$ complexity of EDC
- Learned approach uses only 77 total taps: alternate 5 and 3 taps/step and use different filter coefficients in all steps [Häger and Pfister, 2018a]
- Can outperform "ideal DBP" in the nonlinear regime [Häger and Pfister, 2018b]

Performance–Complexity Trade-off



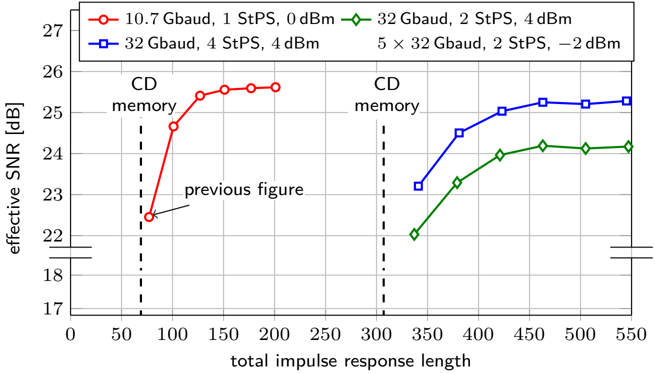
Performance–Complexity Trade-off



Conventional wisdom: Steps are inefficient $\Longrightarrow$ reduce as much as possible

Complexity $\overset{?}{\approx}$ Number of Steps

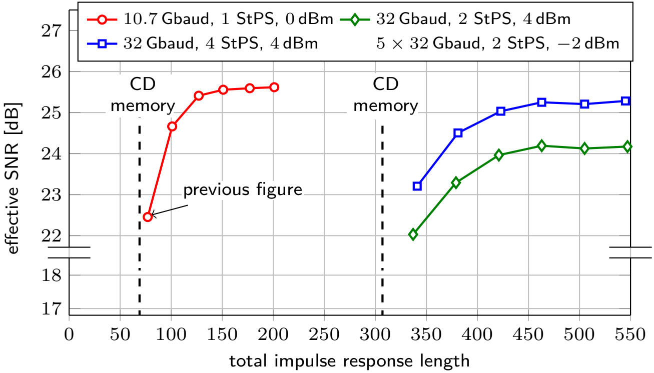
Performance–Complexity Trade-off



Conventional wisdom: Steps are inefficient $\Longrightarrow$ reduce as much as possible

Complexity $\overset{?}{\approx}$ Number of Steps

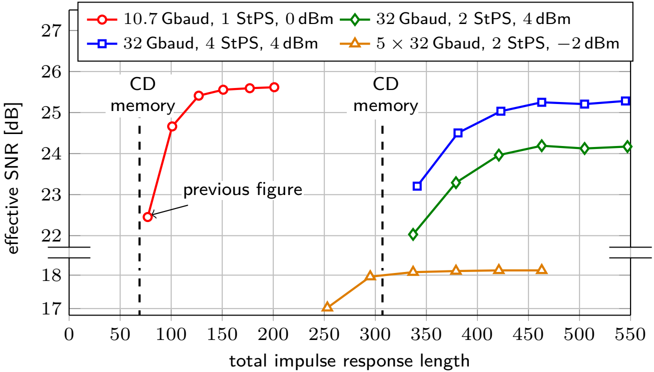
Performance–Complexity Trade-off



Conventional wisdom: Steps are inefficient $\Longrightarrow$ reduce as much as possible

Complexity = Number of Steps × Complexity per Step

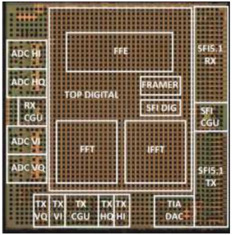
Performance–Complexity Trade-off



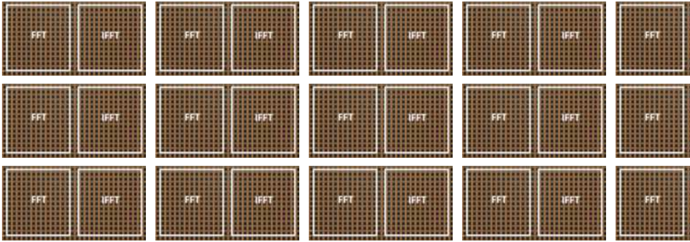
Conventional wisdom: Steps are inefficient $\Longrightarrow$ reduce as much as possible

Complexity = Number of Steps × Complexity per Step

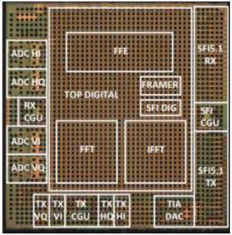
Real-Time ASIC Implementation



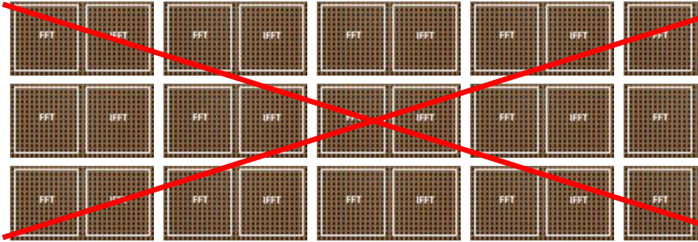
[Crivelli et al., 2014]



Real-Time ASIC Implementation

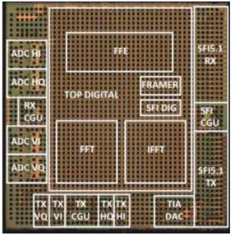


[Crivelli et al., 2014]

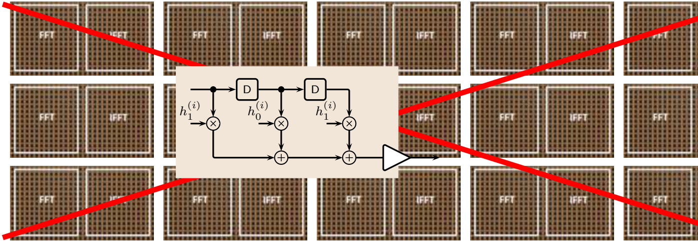


[Fougstedt et al., 2017], Time-domain digital back propagation: Algorithm and finite-precision implementation aspects, (*OFC*)
[Fougstedt et al., 2018], ASIC implementation of time-domain digital back propagation for coherent receivers, *(PTL)*
[Sherborne et al., 2018], On the impact of fixed point hardware for optical fiber nonlinearity compensation algorithms, (*JLT*)

Real-Time ASIC Implementation

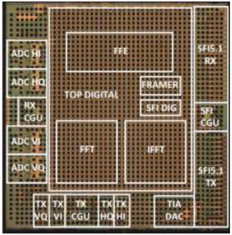


[Crivelli et al., 2014]

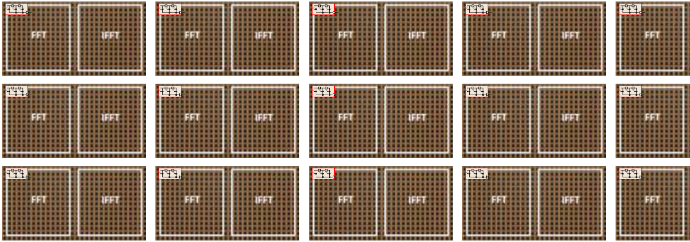


- Our linear steps are very short symmetric FIR filters (as few as 3 taps)

Real-Time ASIC Implementation



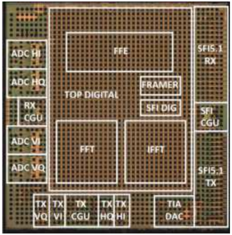
[Crivelli et al., 2014]



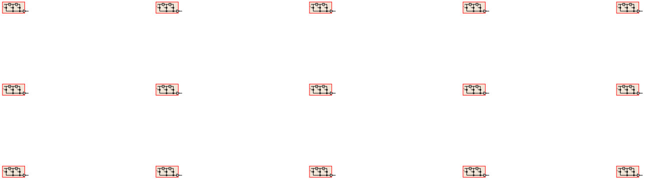
- Our linear steps are very short symmetric FIR filters (as few as 3 taps)
- 28-nm ASIC at 416.7 MHz clock speed (40 GHz signal)
  - Only 5-6 bit filter coefficients via learned quantization
  - Hardware-friendly nonlinear steps (Taylor expansion)
  - All FIR filters are fully reconfigurable

[Fougstedt et al., 2018], ASIC implementation of time-domain digital backpropagation with deep-learned chromatic dispersion filters, *(ECOC)*

Real-Time ASIC Implementation



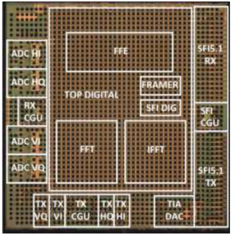
[Crivelli et al., 2014]



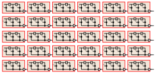
- Our linear steps are very short symmetric FIR filters (as few as 3 taps)
- 28-nm ASIC at 416.7 MHz clock speed (40 GHz signal)
  - Only 5-6 bit filter coefficients via learned quantization
  - Hardware-friendly nonlinear steps (Taylor expansion)
  - All FIR filters are fully reconfigurable

[Fougstedt et al., 2018], ASIC implementation of time-domain digital backpropagation with deep-learned chromatic dispersion filters, *(ECOC)*

Real-Time ASIC Implementation



[Crivelli et al., 2014]



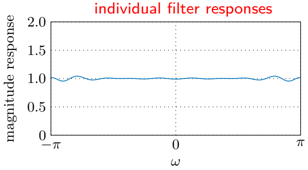
- Our linear steps are very short symmetric FIR filters (as few as 3 taps)
- 28-nm ASIC at 416.7 MHz clock speed (40 GHz signal)
  - Only 5-6 bit filter coefficients via learned quantization
  - Hardware-friendly nonlinear steps (Taylor expansion)
  - All FIR filters are fully reconfigurable
- $< 2\times$ power compared to EDC [Crivelli et al., 2014, Pillai et al., 2014]

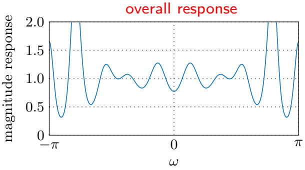
[Fougstedt et al., 2018], ASIC implementation of time-domain digital backpropagation with deep-learned chromatic dispersion filters, *(ECOC)*

Why Does The Learning Approach Work?

Previous work: design a single filter or filter pair and use it repeatedly.

$\Longrightarrow$ Good overall response only possible with very long filters.





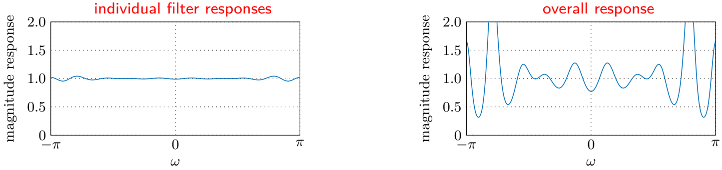
From [Ip and Kahn, 2009]:

- "We also note that [. . .] 70 taps, is much larger than expected"
- "This is due to amplitude ringing in the frequency domain"
- "Since backpropagation requires multiple iterations of the linear filter, amplitude distortion due to ringing accumulates (Goldfarb & Li, 2009)"

Why Does The Learning Approach Work?

Previous work: design a single filter or filter pair and use it repeatedly.

$\Longrightarrow$ Good overall response only possible with very long filters.



From [Ip and Kahn, 2009]:

- “We also note that [. . .] 70 taps, is much larger than expected”
- “This is due to amplitude ringing in the frequency domain”
- “Since backpropagation requires multiple iterations of the linear filter, amplitude distortion due to ringing accumulates (Goldfarb & Li, 2009)”

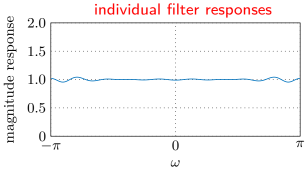
The learning approach uncovered that there is no such requirement!

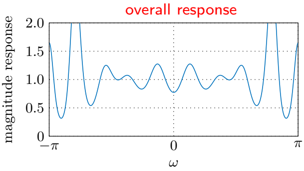
[Lian, Häger, Pfister, 2018], What can machine learning teach us about communications? (*ITW*)

Why Does The Learning Approach Work?

Previous work: design a single filter or filter pair and use it repeatedly.

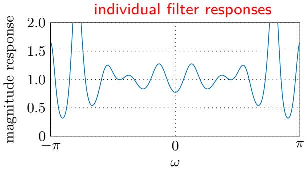
$\Longrightarrow$ Good overall response only possible with very long filters.

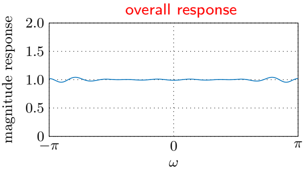




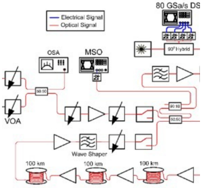
Sacrifice individual filter accuracy, but different response per step.

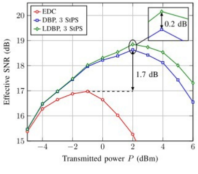
$\Longrightarrow$ Good overall response even with very short filters by joint optimization.

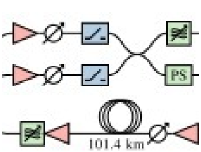


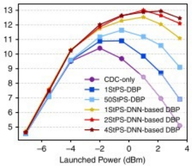


Experimental Investigations









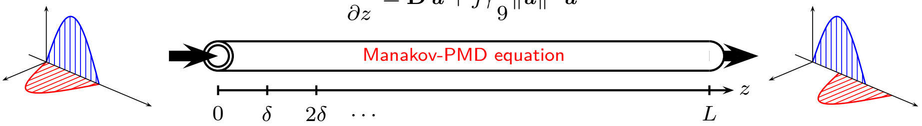
Training with real-world data sets including presence of various hardware impairments (phase noise, timing error, frequency offset, etc.)

- [Oliari et al., 2020], Revisiting Efficient Multi-step Nonlinearity Compensation with Machine Learning: An Experimental Demonstration, *(J. Lightw. Technol.)*
- [Sillekens et al., 2020], Experimental Demonstration of Learned Time-domain Digital Back-propagation, *(Proc. IEEE Workshop on Signal Processing Systems)*
- [Fan et al., 2020], Advancing Theoretical Understanding and Practical Performance of Signal Processing for Nonlinear Optical Communications through Machine Learning, *(Nat. Commun.)*
- [Bitachon et al., 2020], Deep learning based Digital Back Propagation Demonstrating SNR gain at Low Complexity in a 1200 km Transmission Link, *(Opt. Express)*

Outline

Evolution of Polarization-Multiplexed Signals

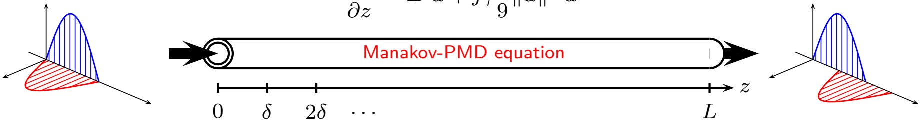
$$\frac{\partial \boldsymbol{u}}{\partial z} = \hat{\mathbf{D}}\,\boldsymbol{u} + \jmath\gamma\frac{8}{9}\|\boldsymbol{u}\|^2\,\boldsymbol{u}$$



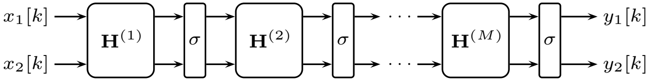
- Jones vector $\boldsymbol{u} \triangleq (u_1(t,z), u_2(t,z))^\top$ with complex baseband signals
- linear operator $\hat{\mathbf{D}}$: attentuation, chromatic & polarization mode dispersion

Evolution of Polarization-Multiplexed Signals

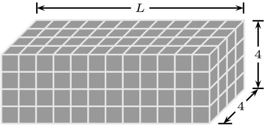
$$\frac{\partial \boldsymbol{u}}{\partial z} = \hat{\mathbf{D}}\,\boldsymbol{u} + \jmath\gamma\frac{8}{9}\|\boldsymbol{u}\|^2\,\boldsymbol{u}$$



- Jones vector $\boldsymbol{u} \triangleq (u_1(t,z), u_2(t,z))^\top$ with complex baseband signals
- linear operator $\hat{\mathbf{D}}$: attentuation, chromatic & polarization mode dispersion
- Split-step method: alternate linear and nonlinear steps $\sigma(\boldsymbol{x}) = \boldsymbol{x}e^{\jmath\gamma\frac{8}{9}\delta\|\boldsymbol{x}\|^2}$

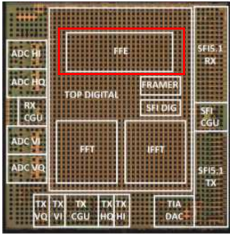


$e^{\delta\hat{\mathbf{D}}} \approx$

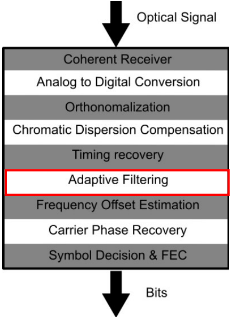


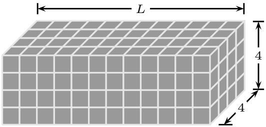
2 × 2 complex or
4 × 4 real MIMO filters $\Longrightarrow$ complexity!

Real-Time Compensation of Polarization Impairments



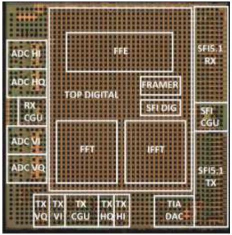
[Crivelli et al., 2014]



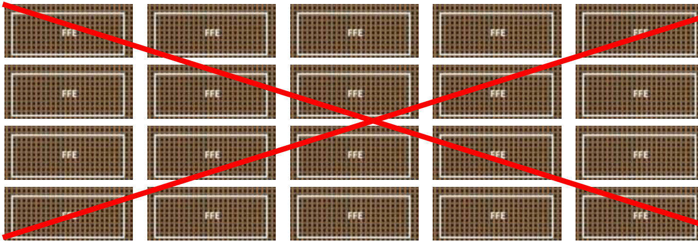


- time-varying effects (e.g., drifts) & apriori unknown realizations
- $\Longrightarrow$ adaptive filtering (via stochastic gradient descent) required

Real-Time Compensation of Polarization Impairments



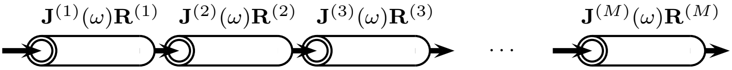
[Crivelli et al., 2014]



- time-varying effects (e.g., drifts) & apriori unknown realizations
- $\Longrightarrow$ adaptive filtering (via stochastic gradient descent) required

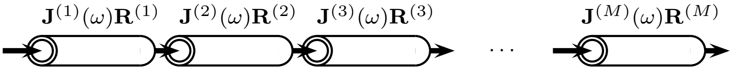
- Using (and updating) full MIMO filters in each step is not feasible.
- We propose a hardware-efficient machine-learning model based on the propagation characteristics

Modeling of Polarization Mode Dispersion (PMD)



The overall PMD is modeled via M sections, where each section introduces

Modeling of Polarization Mode Dispersion (PMD)

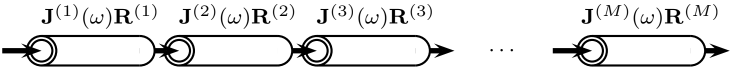


The overall PMD is modeled via M sections, where each section introduces

1. a differential group delay (DGD) $\tau^{(k)}$, described by

$$\mathbf{J}^{(k)}(\omega) = \begin{pmatrix} \exp\left(-\jmath\omega\frac{\tau^{(k)}}{2}\right) & 0 \\ 0 & \exp\left(\jmath\omega\frac{\tau^{(k)}}{2}\right) \end{pmatrix}$$

Modeling of Polarization Mode Dispersion (PMD)



The overall PMD is modeled via M sections, where each section introduces

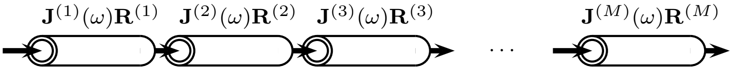
1. a differential group delay (DGD) $\tau^{(k)}$, described by

$$\mathbf{J}^{(k)}(\omega) = \begin{pmatrix} \exp\left(-\jmath\omega\frac{\tau^{(k)}}{2}\right) & 0 \\ 0 & \exp\left(\jmath\omega\frac{\tau^{(k)}}{2}\right) \end{pmatrix}$$

2. a rotation of the polarization state, described by $\mathbf{R}^{(k)} \in \mathrm{SU}(2)$, where

$$\mathrm{SU}(2) = \left\{ \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix} : a, b \in \mathbb{C}, |a|^2 + |b|^2 = 1 \right\}$$

Modeling of Polarization Mode Dispersion (PMD)



The overall PMD is modeled via M sections, where each section introduces

1. a differential group delay (DGD) $\tau^{(k)}$, described by

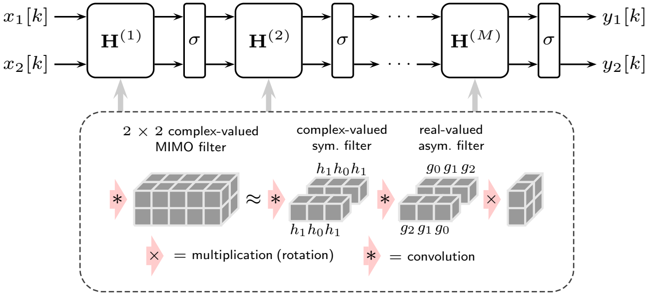
$$\mathbf{J}^{(k)}(\omega) = \begin{pmatrix} \exp\left(-\jmath\omega\frac{\tau^{(k)}}{2}\right) & 0 \\ 0 & \exp\left(\jmath\omega\frac{\tau^{(k)}}{2}\right) \end{pmatrix}$$

2. a rotation of the polarization state, described by $\mathbf{R}^{(k)} \in \mathsf{SU}(2)$, where

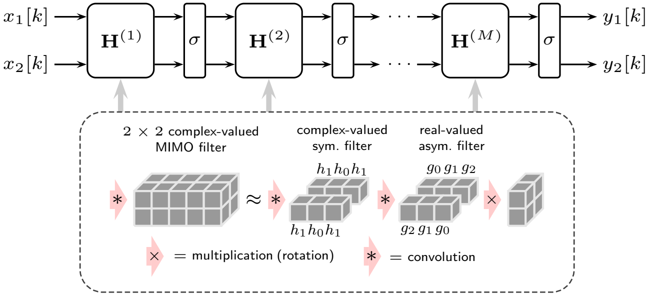
$$\mathsf{SU}(2) = \left\{ \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix} : a, b \in \mathbb{C}, |a|^2 + |b|^2 = 1 \right\}$$

(i) We integrate these operations in each step/layer
(ii) We use real-valued (asymmetric) FIR filters to approximate DGD

The Final Machine-Learning Model



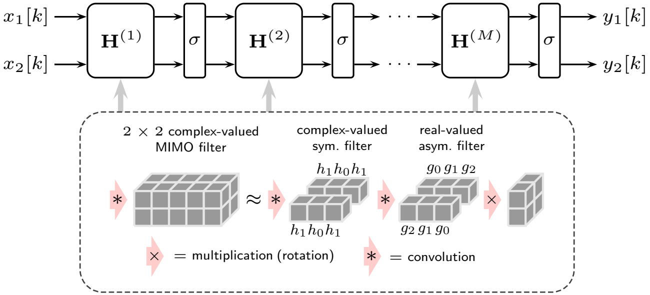
The Final Machine-Learning Model



Each linear step consists of 3 trainable components

1. complex-valued (symmetric) filters that mainly account for dispersion
2. real-valued (asymmetric) filters for DGD
3. memoryless "rotation" matrices $\begin{pmatrix} a & -b^* \\ b & a^* \end{pmatrix}$, where $a, b \in \mathbb{C}$ (4 real parameters)

The Final Machine-Learning Model

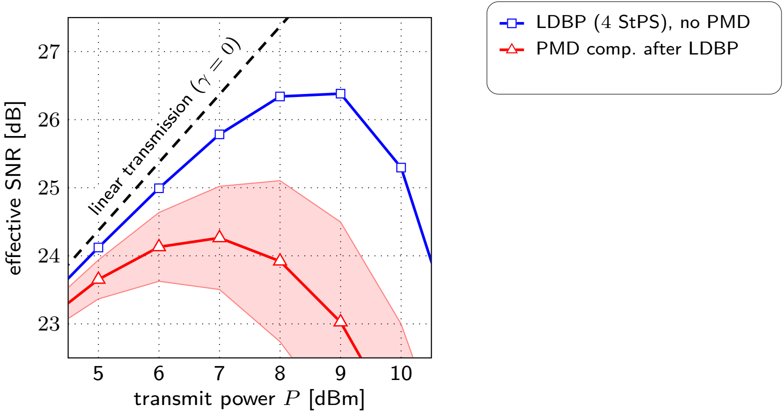


Compared to prior work, our learning-based approach ...

- assumes no knowledge about PMD realizations or accumulated PMD
- is FIR-filter based! Avoids frequency-domain (FFT-based) filtering

[Goroshko et al., 2016], Overcoming performance limitations of digital back propagation due to polarization mode dispersion, *(CTON)*
[Czegledi et al., 2017], Digital backpropagation accounting for polarization-mode dispersion, *(Opt. Express)*
[Liga et al., 2018], A PMD-adaptive DBP receiver based on SNR optimization, *(OFC)*

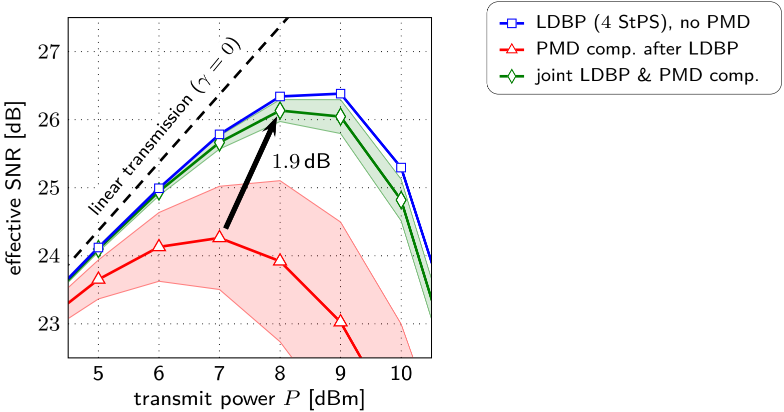
Results (32 Gbaud, 10×100 km, 0.2 ps/$\sqrt{\text{km}}$ PMD)



- Similar parameters & simulation setup compared to [Czegledi et al., 2016], results averaged over 40 PMD realizations

[Bütler et al., 2021], Model-based Machine Learning for Joint Digital Backpropagation and PMD Compensation, *(J. Lightw. Technol.)*, see arXiv:2010.12313

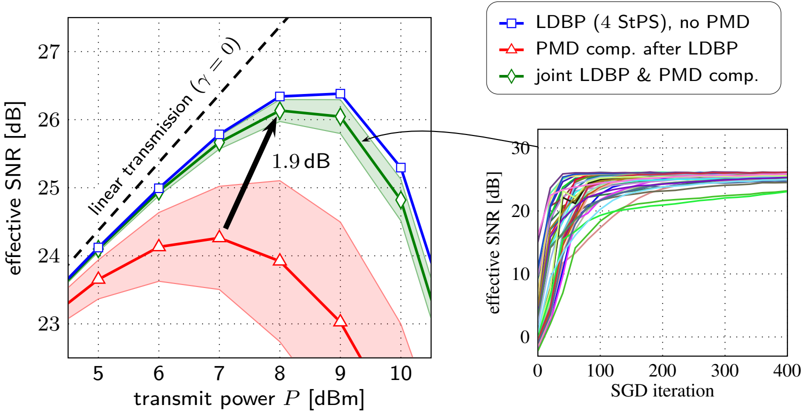
Results (32 Gbaud, 10×100 km, 0.2 ps/$\sqrt{\text{km}}$ PMD)



- Similar parameters & simulation setup compared to [Czegledi et al., 2016], results averaged over 40 PMD realizations

[Bütler et al., 2021], Model-based Machine Learning for Joint Digital Backpropagation and PMD Compensation, *(J. Lightw. Technol.)*, see arXiv:2010.12313

Results (32 Gbaud, 10×100 km, 0.2 ps/$\sqrt{\text{km}}$ PMD)



- Similar parameters & simulation setup compared to [Czegledi et al., 2016], results averaged over 40 PMD realizations
- Reliable convergence "from scratch" + only 9 real parameters per step

[Bütler et al., 2021], Model-based Machine Learning for Joint Digital Backpropagation and PMD Compensation, *(J. Lightw. Technol.)*, see arXiv:2010.12313

Outline

1. Machine Learning and Neural Networks for Communications
2. Physics-Based Machine Learning for Fiber-Optic Communications
3. Learned Digital Backpropagation
4. Polarization-Dependent Effects
5. Conclusions

The Bigger Picture

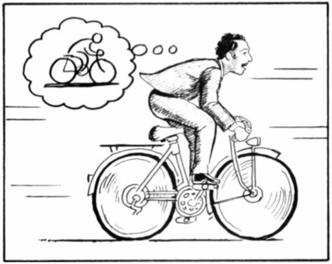
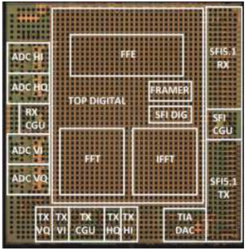


Figure 1. A World Model, from Scott McCloud's *Understanding Comics.* (McCloud, 1993; E, 2012)



[Crivelli et al., 2014]

- Optical receivers build models of their "environment"

[Ha & Schmidhuber, 2018], "World Models", arXiv:1803.10122 [cs.LG]

The Bigger Picture

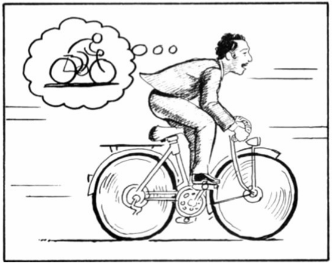
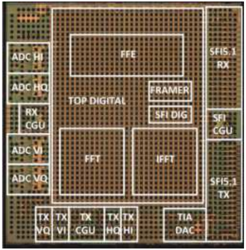


Figure 1. A World Model, from Scott McCloud's *Understanding Comics.* (McCloud, 1993; E, 2012)



[Crivelli et al., 2014]

- Optical receivers build models of their "environment"
- Currently these models are linear and/or rigid (non-adaptive)
- Interpretable physics-based "multi-layer" models for machine learning can be obtained by exploiting our existing domain knowledge

[Ha & Schmidhuber, 2018], "World Models", arXiv:1803.10122 [cs.LG]

Conclusions

Conclusions

neural-network-based ML
universal function approximators
good designs require experience and fine-tuning
black boxes, difficult to "open"

Conclusions

neural-network-based ML	**model-based ML**
universal function approximators	application-tailored
good designs require experience and fine-tuning	relies on domain knowledge (algorithms, physics, . . .)
black boxes, difficult to “open”	familiar building blocks (e.g., FIR filters) can enable interpretability

Conclusions

neural-network-based ML	**model-based ML**
universal function approximators	application-tailored
good designs require experience and fine-tuning	relies on domain knowledge (algorithms, physics, . . .)
black boxes, difficult to “open”	familiar building blocks (e.g., FIR filters) can enable interpretability

[Häger & Pfister, 2020], “Physics-Based Deep Learning for Fiber-Optic Communication Systems”, in *IEEE J. Sel. Areas Commun.* (to appear), see https://arxiv.org/abs/2010.14258

Code: https://github.com/chaeger/LDBP

Conclusions

neural-network-based ML	**model-based ML**
universal function approximators	application-tailored
good designs require experience and fine-tuning	relies on domain knowledge (algorithms, physics, . . .)
black boxes, difficult to "open"	familiar building blocks (e.g., FIR filters) can enable interpretability

[Häger & Pfister, 2020], "Physics-Based Deep Learning for Fiber-Optic Communication Systems", in *IEEE J. Sel. Areas Commun.* (to appear), see `https://arxiv.org/abs/2010.14258`

Code: `https://github.com/chaeger/LDBP`

Thank you!

FORCE Fiber-Optic Communications Research Center — Duke University — TU/e

References I

Crivelli, D. E., Hueda, M. R., Carrer, H. S., Del Barco, M., López, R. R., Gianni, P., Finochietto, J., Swenson, N., Voois, P., and Agazzi, O. E. (2014).
Architecture of a single-chip 50 Gb/s DP-QPSK/BPSK transceiver with electronic dispersion compensation for coherent optical channels.
IEEE Trans. Circuits Syst. I: Reg. Papers, 61(4):1012–1025.

Czegledi, C. B., Liga, G., Lavery, D., Karlsson, M., Agrell, E., Savory, S. J., and Bayvel, P. (2016).
Polarization-mode dispersion aware digital backpropagation.
In *Proc. European Conf. Optical Communication (ECOC)*, Düsseldorf, Germany.

Du, L. B. and Lowery, A. J. (2010).
Improved single channel backpropagation for intra-channel fiber nonlinearity compensation in long-haul optical communication systems.
Opt. Express, 18(16):17075–17088.

Essiambre, R.-J. and Winzer, P. J. (2005).
Fibre nonlinearities in electronically pre-distorted transmission.
In *Proc. European Conf. Optical Communication (ECOC)*, Glasgow, UK.

Häger, C. and Pfister, H. D. (2018a).
Deep learning of the nonlinear Schrödinger equation in fiber-optic communications.
In *Proc. IEEE Int. Symp. Information Theory (ISIT)*, Vail, CO.

Häger, C. and Pfister, H. D. (2018b).
Nonlinear interference mitigation via deep neural networks.
In *Proc. Optical Fiber Communication Conf. (OFC)*, San Diego, CA.

References II

He, K., Zhang, X., Ren, S., and Sun, J. (2015).
Deep residual learning for image recognition.

Ip, E. and Kahn, J. M. (2008).
Compensation of dispersion and nonlinear impairments using digital backpropagation.
J. Lightw. Technol., 26(20):3416–3425.

Ip, E. and Kahn, J. M. (2009).
Nonlinear impairment compensation using backpropagation.
In *Optical Fiber New Developments, Chapter 10*. IntechOpen, London, UK.

Lavery, D., Ives, D., Liga, G., Alvarado, A., Savory, S. J., and Bayvel, P. (2016).
The benefit of split nonlinearity compensation for single-channel optical fiber communications.
IEEE Photon. Technol. Lett., 28(17):1803–1806.

LeCun, Y., Bengio, Y., and Hinton, G. (2015).
Deep learning.
Nature, 521(7553):436–444.

Leibrich, J. and Rosenkranz, W. (2003).
Efficient numerical simulation of multichannel WDM transmission systems limited by XPM.
IEEE Photon. Technol. Lett., 15(3):395–397.

Li, L., Tao, Z., Dou, L., Yan, W., Oda, S., Tanimura, T., Hoshida, T., and Rasmussen, J. C. (2011).
Implementation efficient nonlinear equalizer based on correlated digital backpropagation.
In *Proc. Optical Fiber Communication Conf. (OFC)*, Los Angeles, CA.

References III

Li, X., Chen, X., Goldfarb, G., Mateo, E., Kim, I., Yaman, F., and Li, G. (2008).
Electronic post-compensation of WDM transmission impairments using coherent detection and digital signal processing.
Opt. Express, 16(2):880–888.

Li, Y., Ho, C. K., Wu, Y., and Sun, S. (2005).
Bit-to-symbol mapping in LDPC coded modulation.
In *Proc. Vehicular Technology Conf. (VTC)*, Stockholm, Sweden.

Lin, H. W., Tegmark, M., and Rolnick, D. (2017).
Why does deep and cheap learning work so well?
J. Stat. Phys., 168(6):1223–1247.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015).
Human-level control through deep reinforcement learning.
Nature, 518(7540):529–533.

Nakashima, H., Oyama, T., Ohshima, C., Akiyama, Y., Tao, Z., and Hoshida, T. (2017).
Digital nonlinear compensation technologies in coherent optical communication systems.
In *Proc. Optical Fiber Communication Conf. (OFC)*, Los Angeles, CA.

Napoli, A., Maalej, Z., Sleiffer, V. A. J. M., Kuschnerov, M., Rafique, D., Timmers, E., Spinnler, B., Rahman, T., Coelho, L. D., and Hanik, N. (2014).
Reduced complexity digital back-propagation methods for optical communication systems.
J. Lightw. Technol., 32(7):1351–1362.

References IV

O'Shea, T. and Hoydis, J. (2017).
An introduction to deep learning for the physical layer.
IEEE Trans. Cogn. Commun. Netw., 3(4):563–575.

Paré, C., Villeneuve, A., Bélanger, P.-A. A., and Doran, N. J. (1996).
Compensating for dispersion and the nonlinear Kerr effect without phase conjugation.
Opt. Lett., 21(7):459–461.

Pillai, B. S. G., Sedighi, B., Guan, K., Anthapadmanabhan, N. P., Shieh, W., Hinton, K. J., and Tucker, R. S. (2014).
End-to-end energy modeling and analysis of long-haul coherent transmission systems.
J. Lightw. Technol., 32(18):3093–3111.

Rafique, D., Zhao, J., and Ellis, A. D. (2011).
Digital back-propagation for spectrally efficient wdm 112 gbit/s pm m-ary qam transmission.
Opt. Express, 19(6):5219–5224.

Roberts, K., Li, C., Strawczynski, L., O'Sullivan, M., and Hardcastle, I. (2006).
Electronic precompensation of optical nonlinearity.
IEEE Photon. Technol. Lett., 18(2):403–405.

Secondini, M., Rommel, S., Meloni, G., Fresi, F., Forestieri, E., and Poti, L. (2016).
Single-step digital backpropagation for nonlinearity mitigation.
Photon. Netw. Commun., 31(3):493–502.

Sheikh, A., Fougstedt, C., Graell i Amat, A., Johannisson, P., Larsson-Edefors, P., and Karlsson, M. (2016).
Dispersion compensation FIR filter with improved robustness to coefficient quantization errors.
J. Lightw. Technol., 34(22):5110–5117.

References V

Yan, W., Tao, Z., Dou, L., Li, L., Oda, S., Tanimura, T., Hoshida, T., and Rasmussen, J. C. (2011).
Low complexity digital perturbation back-propagation.
In *Proc. European Conf. Optical Communication (ECOC)*, page Tu.3.A.2, Geneva, Switzerland.