Reducing the Complexity of Digital Backpropagation with Machine Learning

Christian Häger

Department of Electrical Engineering, Chalmers University of Technology, Sweden

ACP Workshop
October 24, 2021

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

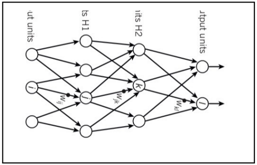

Menno van den Hout
TU/e


Sjoerd van der Heide
TU/e

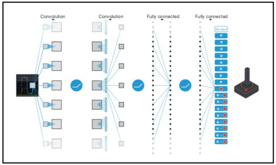

Chigo Okonkwo
TU/e

“Multi-layer” vs. “Multi-step”

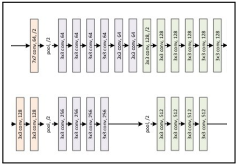
Deep Learning [LeCun et al., 2015]



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



ResNet [He et al., 2015]

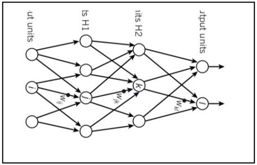


. . .

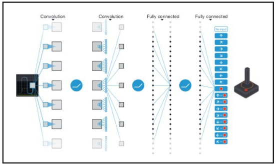
Multi-layer neural networks: impressive performance, countless applications

“Multi-layer” vs. “Multi-step”

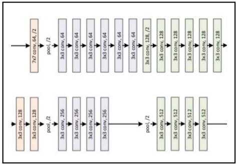
Deep Learning [LeCun et al., 2015]



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



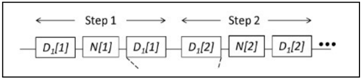
ResNet [He et al., 2015]



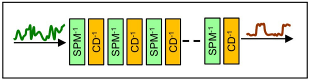
. . .

Multi-layer neural networks: impressive performance, countless applications

---



[Du and Lowery, 2010]



[Nakashima et al., 2017]

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

- “with only four steps for the entire link . . . ” [Du and Lowery, 2010]
- “up to 80% reduction in required [. . .] steps” [Rafique et al., 2011]
- “it reduces 85% back-propagation stages [. . .]” [Yan et al., 2011]
- “considerably reduces the number of spans needed ” [Napoli et al., 2014]
- “single-step digital backpropagation” [Secondini et al., 2016]

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 multi-step nonlinear equalization via learned digital backpropagation

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 multi-step nonlinear equalization via learned digital backpropagation

Complexity $\overset{?}{\approx}$ Number of 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)
3. revisit hardware-efficient multi-step nonlinear equalization via learned digital backpropagation

Complexity = Number of Steps × Complexity per Step

Outline

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

Outline

1. Machine Learning and Neural Networks

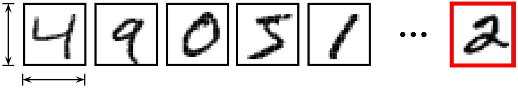
2. Physics-Based Machine Learning for Fiber-Optic Communications

3. Learned Digital Backpropagation

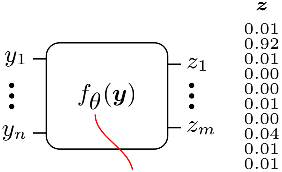
4. Conclusions

Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



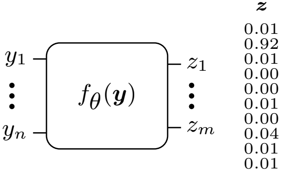
28×28 pixels $\implies n = 784$



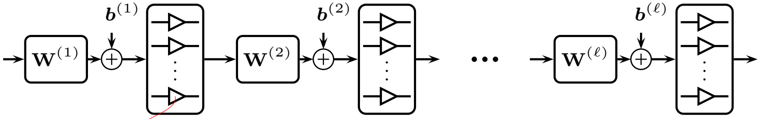
parameters
to be optimized/learned

Supervised Learning

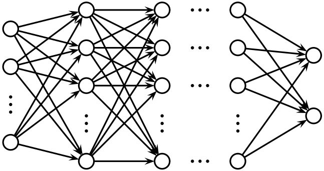
handwritten digit recognition (MNIST: 70,000 images)



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



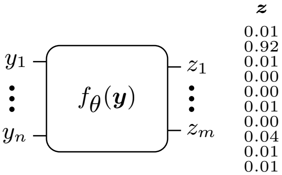
activation function



equivalent graph representation

Supervised Learning

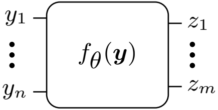
handwritten digit recognition (MNIST: 70,000 images)



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

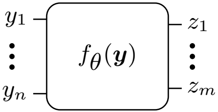
Given a data set $\mathcal{D} = \{(\boldsymbol{y}^{(i)}, \boldsymbol{x}^{(i)})\}_{i=1}^{N}$, where $\boldsymbol{y}^{(i)}$ are model inputs and $\boldsymbol{x}^{(i)}$ are labels, we iteratively minimize

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

stochastic gradient descent

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

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

stochastic gradient descent

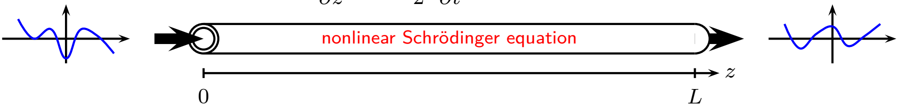
Are there other ways to design good f_θ?

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

Outline

The Split-Step Method

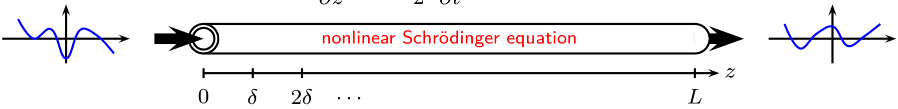
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



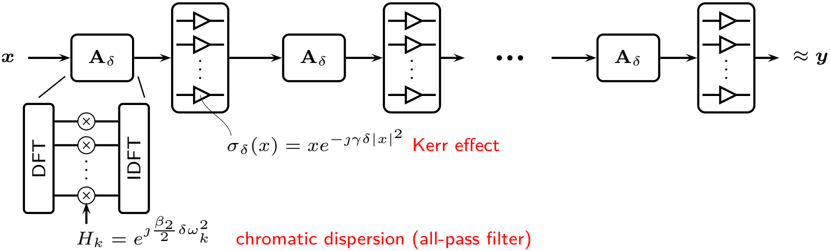
- Deterministic channel model: partial differential equation

The Split-Step Method

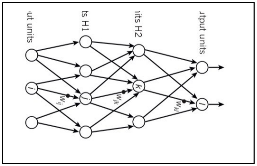
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



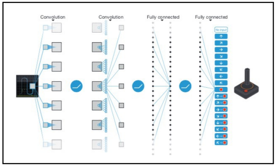
- Deterministic channel model: partial differential equation
- Split-step method with M steps ($\delta = L/M$):



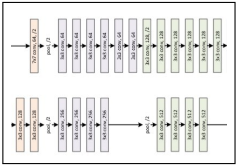
Deep Learning [LeCun et al., 2015]



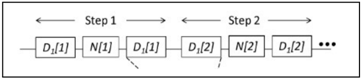
Deep Q-Learning [Mnih et al., 2015]



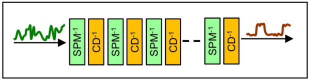
ResNet [He et al., 2015]



...

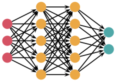


[Du and Lowery, 2010]

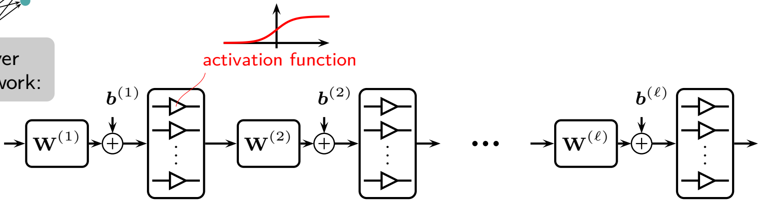


[Nakashima et al., 2017]

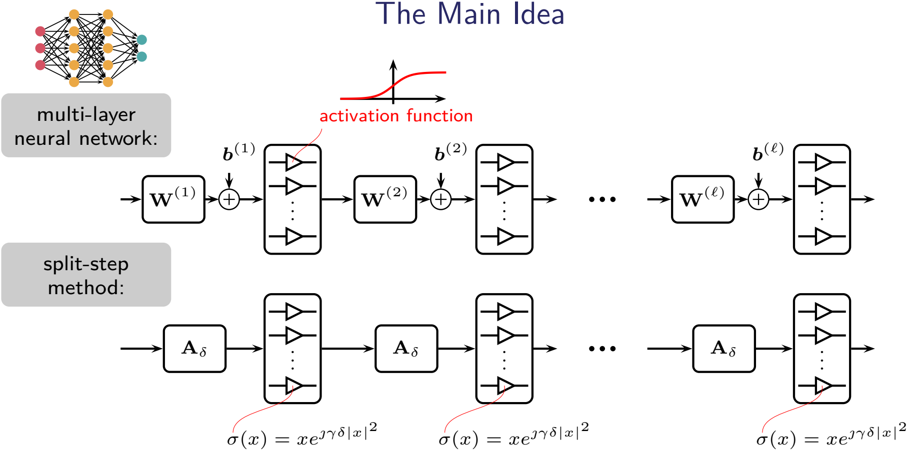
The Main Idea



multi-layer neural network:

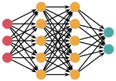


The Main Idea



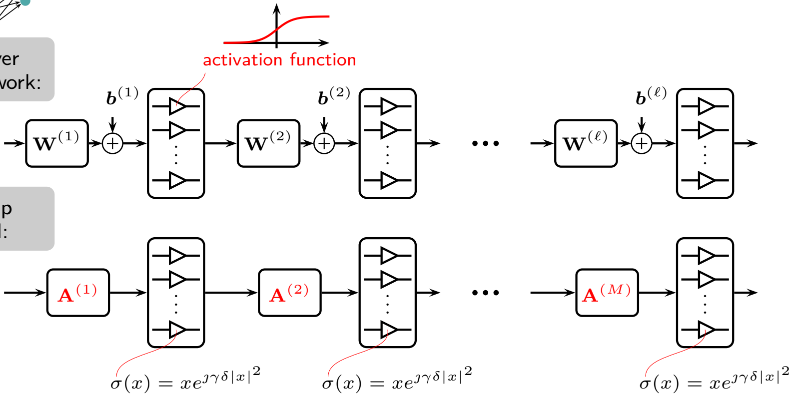
- This almost looks like a deep neural net!

The Main Idea



multi-layer neural network:

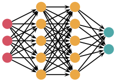
split-step method:

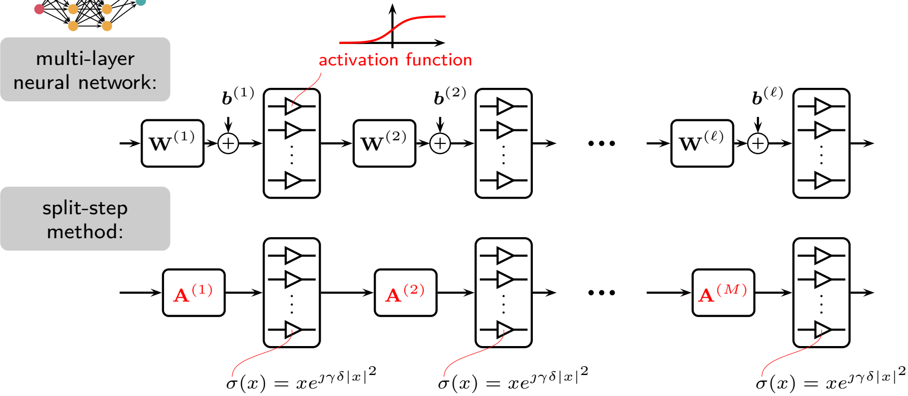


- This almost looks like a deep neural net!
- Parameterize all linear steps: f_θ with $\theta = \{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(M)}\}$

[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

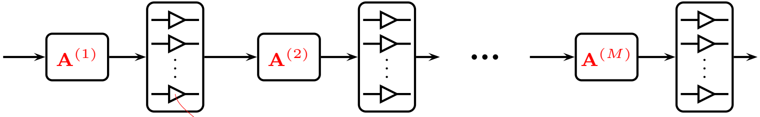




- This almost looks like a deep neural net!
- Parameterize all linear steps: f_θ with $\theta = \{\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(M)}\}$
- Special cases: step-size optimization, nonlinear operator "placement", ...

[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.*

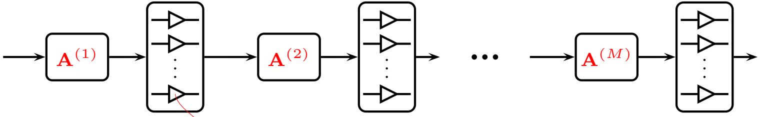
Potential Benefits



$\sigma_{\delta}(x) = x e^{\jmath \gamma \delta |x|^2}$ Kerr effect

- 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?

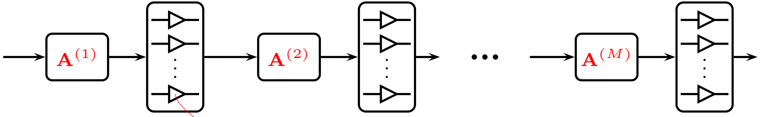
Potential Benefits



$\sigma_\delta(x) = xe^{\jmath\gamma\delta|x|^2}$ Kerr effect

- 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?

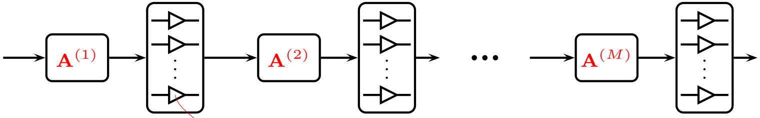
Potential Benefits



$\sigma_\delta(x) = xe^{\jmath\gamma\delta|x|^2}$ Kerr effect

- 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

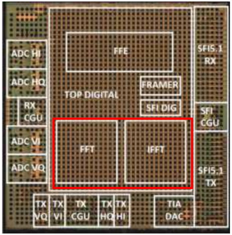


$\sigma_\delta(x) = x e^{\jmath \gamma \delta |x|^2}$ Kerr effect

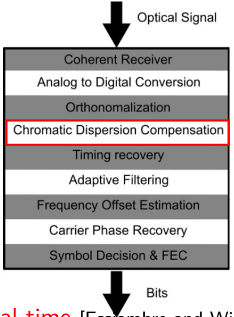
- How to choose the network architecture (#layers, activation function)? ✓
  - Activation function is fixed; number of layers = number of steps
  - Hidden feature representations ≈ signal at intermediate fiber locations
  - Parameter initialization based on conventional split-step method
- How to limit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

Real-Time Digital Backpropagation

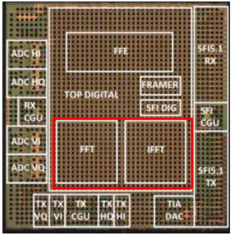


[Crivelli et al., 2014]

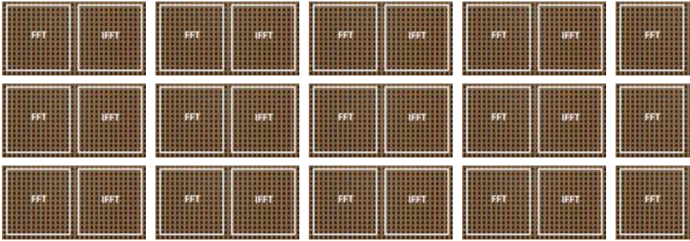


- Invert a PDE in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]: widely considered to be impractical
- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)

Real-Time Digital Backpropagation

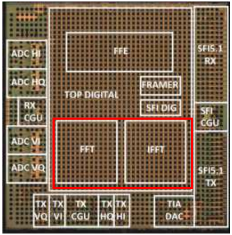


[Crivelli et al., 2014]

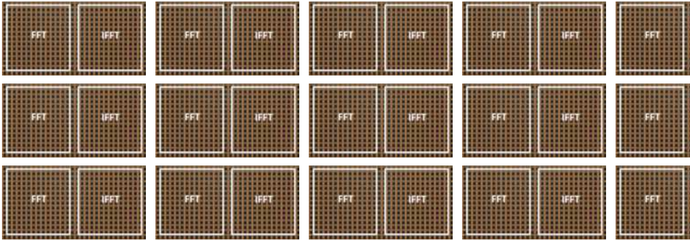


- Invert a PDE in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]: widely considered to be impractical
- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)

Real-Time Digital Backpropagation

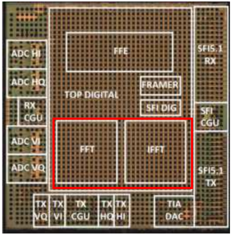


[Crivelli et al., 2014]

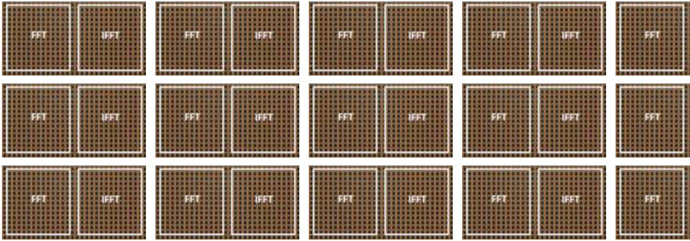


- Invert a PDE in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]: widely considered to be impractical
- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)

Real-Time Digital Backpropagation

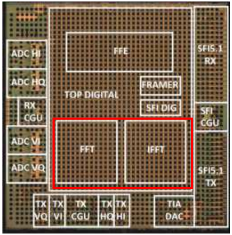


[Crivelli et al., 2014]

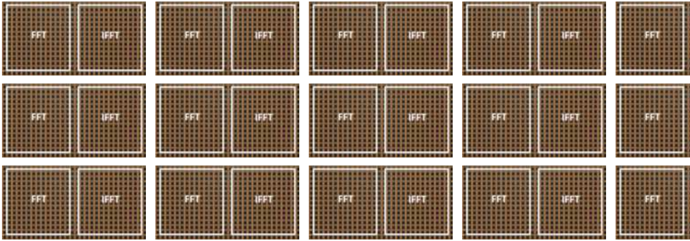


- Invert a PDE in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]: widely considered to be impractical
- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)
- Intuitive, but . . .

Real-Time Digital Backpropagation

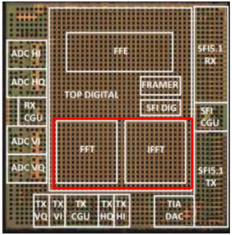


[Crivelli et al., 2014]

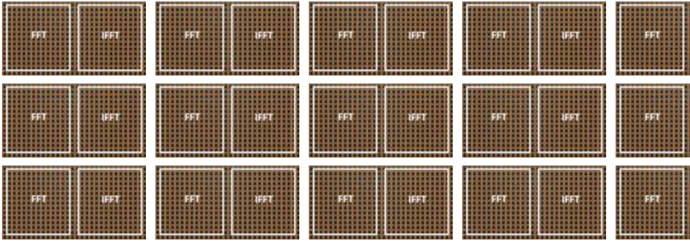


- Invert a PDE in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]: widely considered to be impractical
- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [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]



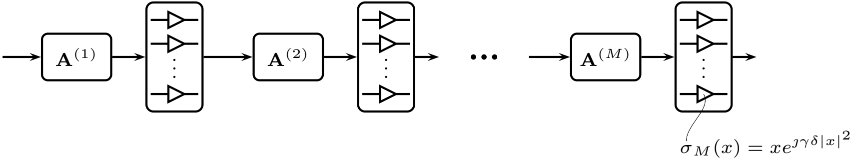
- Invert a PDE in real time [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008]: widely considered to be impractical
- Complexity increases with the number of steps M $\implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [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

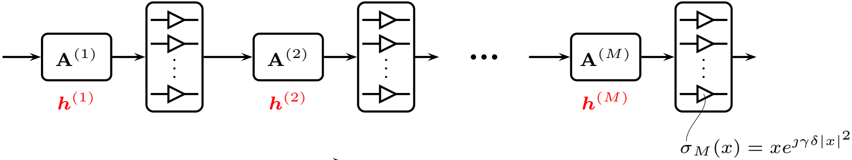
Learned Digital Backpropagation

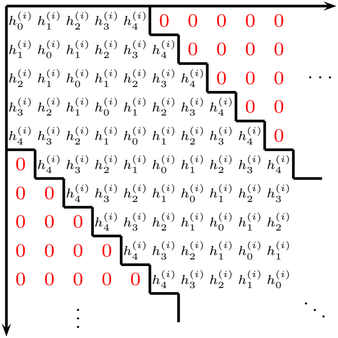
TensorFlow implementation of the computation graph $f_\theta(\boldsymbol{y})$:

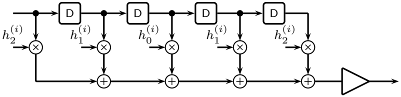


Learned Digital Backpropagation

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



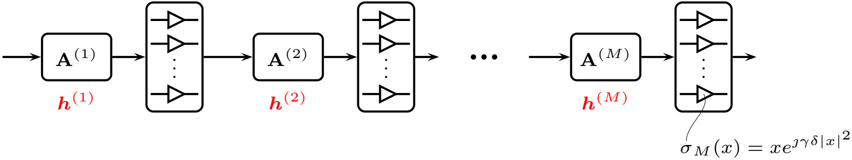




finite impulse response (FIR) filter
complex & symmetric coefficients

Learned Digital Backpropagation

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



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

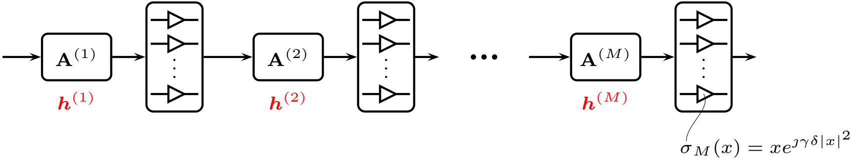
$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

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

$$\longleftarrow \text{starting length } 2K' + 1 \longrightarrow$$

$$\theta = \left\{ \begin{array}{lcccccccccccl} \boldsymbol{h}^{(1)} = (& h_{K'}^{(1)} & \cdots & h_{K}^{(1)} & \cdots & h_{1}^{(1)} & h_{0}^{(1)} & h_{1}^{(1)} & \cdots & h_{K}^{(1)} & \cdots & h_{K'}^{(1)} &) \quad \text{step } 1 \\ \boldsymbol{h}^{(2)} = (& h_{K'}^{(2)} & \cdots & h_{K}^{(2)} & \cdots & h_{1}^{(2)} & h_{0}^{(2)} & h_{1}^{(2)} & \cdots & h_{K}^{(2)} & \cdots & h_{K'}^{(2)} &) \quad \text{step } 2 \\ \vdots & \vdots & & & & & \vdots & & & & & \vdots & \\ \boldsymbol{h}^{(M)} = (& h_{K'}^{(M)} & \cdots & h_{K}^{(M)} & \cdots & h_{1}^{(M)} & h_{0}^{(M)} & h_{1}^{(M)} & \cdots & h_{K}^{(M)} & \cdots & h_{K'}^{(M)} &) \quad \text{step } M \end{array} \right.$$

Iterative Filter Tap Pruning

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

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

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

Iterative Filter Tap Pruning

$$\overleftarrow{\quad\text{starting length } 2K'+1\quad}\!\!\!\rightarrow$$

$$\overleftarrow{\quad\text{target length } 2K+1\quad}\!\!\!\rightarrow$$

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

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

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

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

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

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

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

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

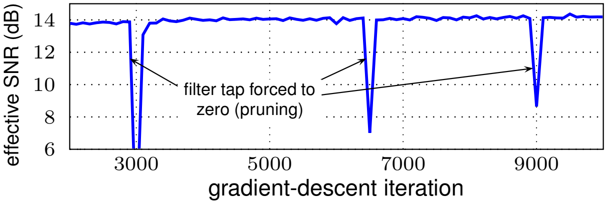
Iterative Filter Tap Pruning

starting length $2K'+1$

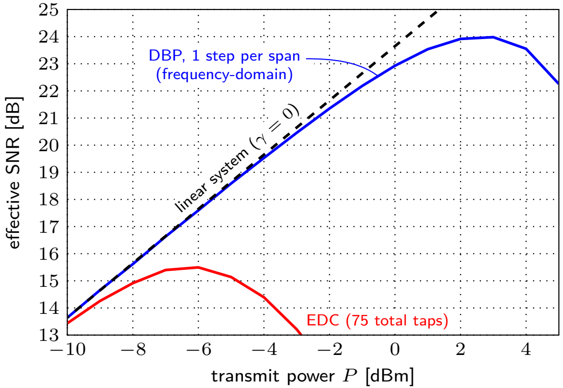
target length $2K+1$

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

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



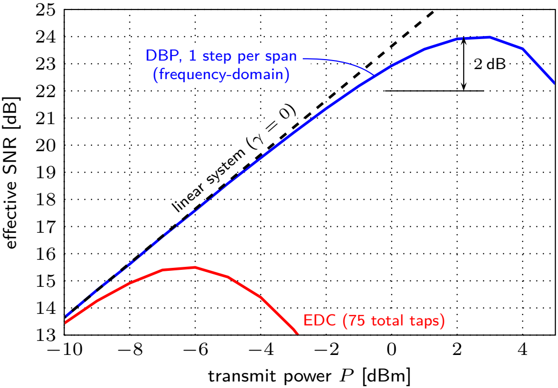
Revisiting Ip and Kahn (2008)



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)

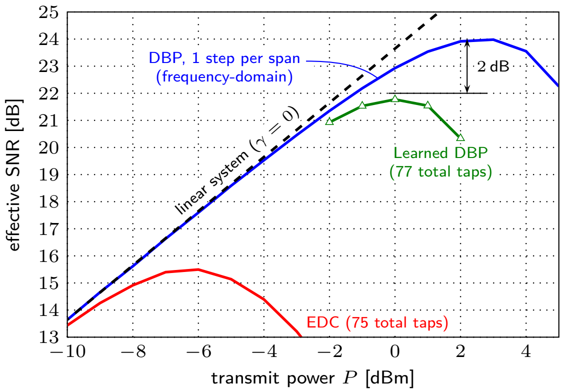


Parameters similar to [Ip and Kahn, 2008]:

- 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)

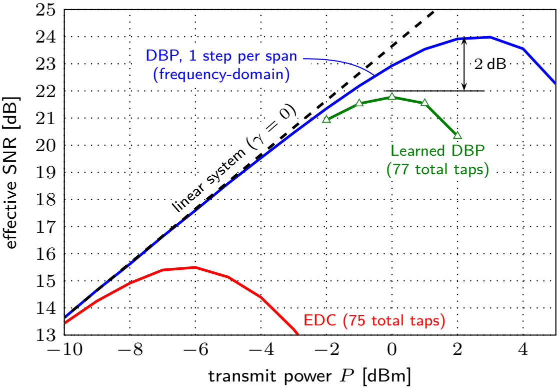


Parameters similar to [Ip and Kahn, 2008]:

- 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]

Extensions & Experimental Investigations

Wideband & WDM signals

- [Häger and Pfister, 2018], Wideband time-domain digital backpropagation via subband processing and deep learning, *(ECOC)*

ASIC implementation & finite-precision aspects

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

Polarization-dependent Effects (PMD)

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

Experimental demonstrations & implementation aspects (e.g., phase noise)

- [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

Conclusions

Conclusions

- We have proposed a physics-based machine-learning approach for fiber-optic communication systems
- We have revisited efficient multi-step digital backpropagation and shown that deep-learning tools can be used to
  - jointly optimize all linear substeps
  - prune filter taps to get very short filters
  - jointly quantize all filter coefficients
- Multi-step enables factorization into simple, elementary steps

[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

- We have proposed a physics-based machine-learning approach for fiber-optic communication systems
- We have revisited efficient multi-step digital backpropagation and shown that deep-learning tools can be used to
  - jointly optimize all linear substeps
  - prune filter taps to get very short filters
  - jointly quantize all filter coefficients
- Multi-step enables factorization into simple, elementary steps

[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

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.

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.

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

References II

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

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

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.

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.

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.

References III

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.

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.

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.