Deep Learning of the Nonlinear Schrödinger Equation in Fiber-Optic Communications

Christian Häger[1,2] Henry D. Pfister[2]

[1] Department of Electrical Engineering, Chalmers University of Technology, Gothenburg

[2] Department of Electrical and Computer Engineering, Duke University, Durham

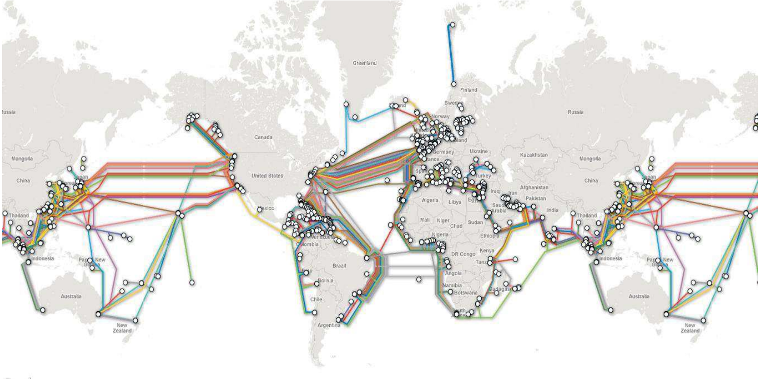
ISIT 2018, Vail, June 21, 2018

FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

DUKE UNIVERSITY

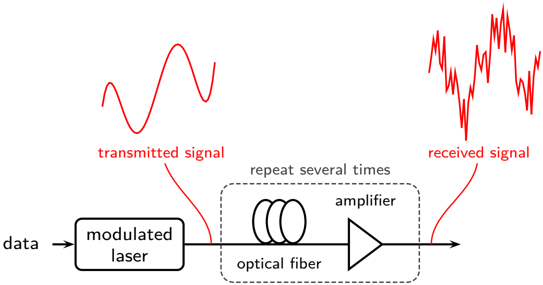
CHALMERS

Fiber-Optic Communications

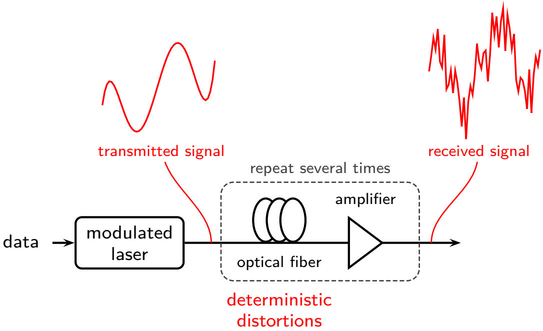


Fiber-optic communication systems enable high-speed data traffic (100 Gbit/s per channel or higher) over very long distances.

Fiber-Optic Communications

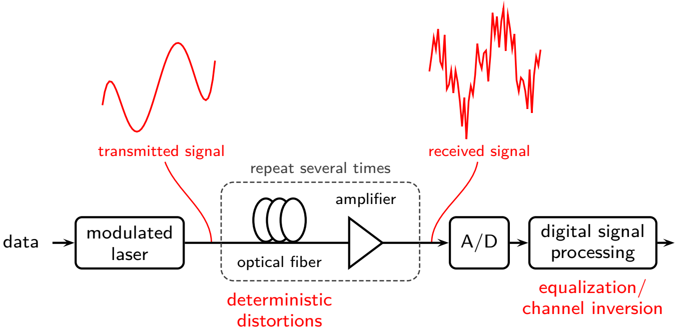


Fiber-Optic Communications



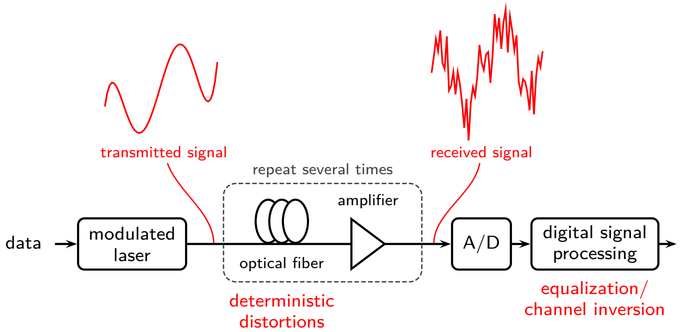
- Dispersion: different wavelengths travel at different speeds (linear)
- Kerr effect: refractive index changes with signal intensity (nonlinear)

Fiber-Optic Communications



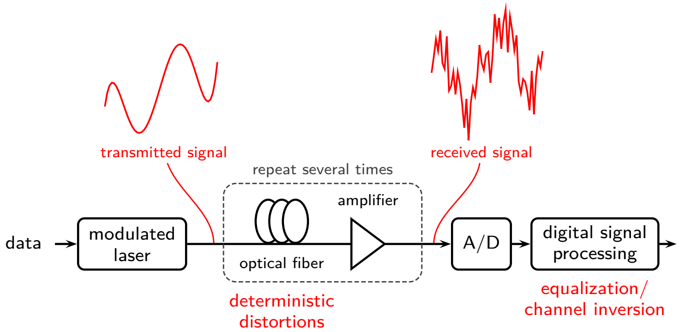
- Dispersion: different wavelengths travel at different speeds (linear)
- Kerr effect: refractive index changes with signal intensity (nonlinear)

Fiber-Optic Communications



- Dispersion: different wavelengths travel at different speeds (linear)
- Kerr effect: refractive index changes with signal intensity (nonlinear)
- Key challenge: complexity constraints due to very high data rates

Fiber-Optic Communications



- Dispersion: different wavelengths travel at different speeds (linear)
- Kerr effect: refractive index changes with signal intensity (nonlinear)
- Key challenge: complexity constraints due to very high data rates

This talk

Machine learning for low-complexity real-time channel inversion

Outline

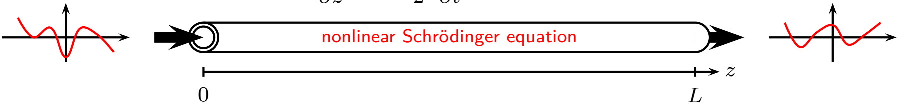
1. Channel Modeling and the Nonlinear Schrödinger Equation
2. Connection to Deep Learning
3. Joint Filter Optimization and Pruning
4. Results
5. Conclusions

Outline

1. Channel Modeling and the Nonlinear Schrödinger Equation
2. Connection to Deep Learning
3. Joint Filter Optimization and Pruning
4. Results
5. Conclusions

Channel Modeling and Digital Backpropagation

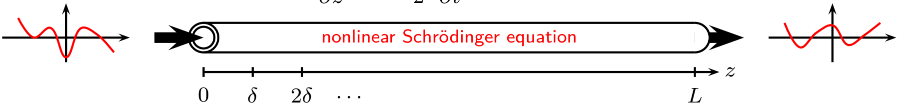
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



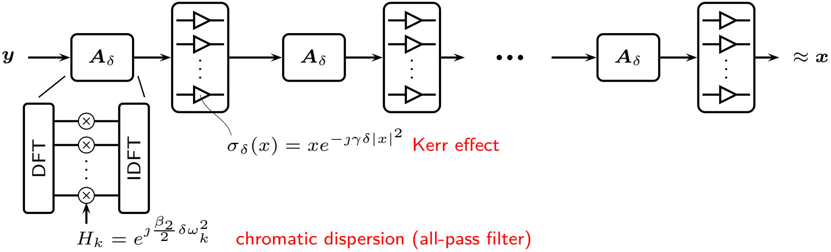
- Invert a partial differential equation in real time ([Paré et al., 1996], [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008])

Channel Modeling and Digital Backpropagation

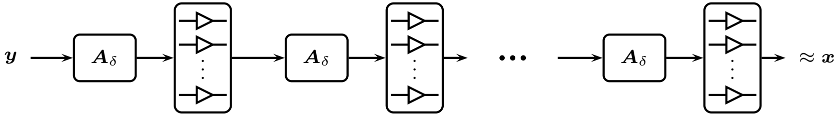
$$\frac{\partial u}{\partial z} = -\jmath \frac{\beta_2}{2} \frac{\partial^2 u}{\partial t^2} + \jmath \gamma u |u|^2$$



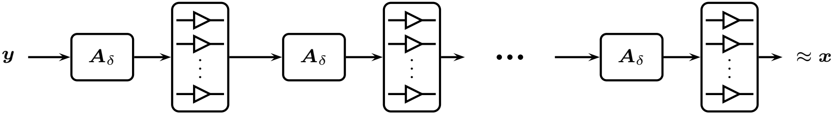
- Invert a partial differential equation in real time ([Paré et al., 1996], [Essiambre and Winzer, 2005], [Roberts et al., 2006], [Li et al., 2008], [Ip and Kahn, 2008])
- Split-step Fourier method with M steps ($\delta = L/M$):



Real-Time Digital Backpropagation

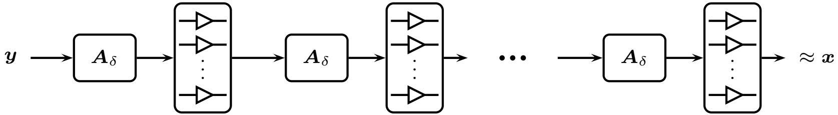


Real-Time Digital Backpropagation



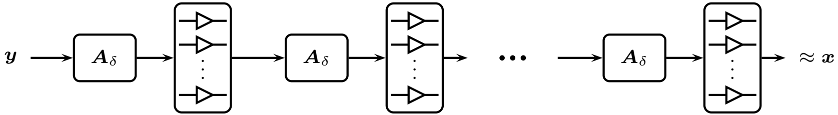
- Widely considered to be impractical (too complex): linear equalization is already one of the most power hungry DSP blocks in coherent receivers

Real-Time Digital Backpropagation



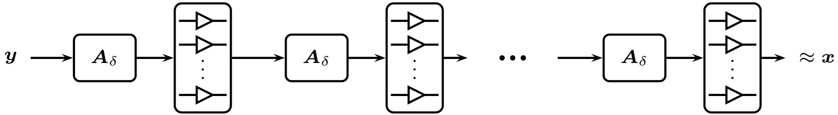
- Widely considered to be impractical (too complex): linear equalization is already one of the most power hungry DSP blocks in coherent receivers
- Complexity increases with the number of steps $M \implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], . . .)

Real-Time Digital Backpropagation



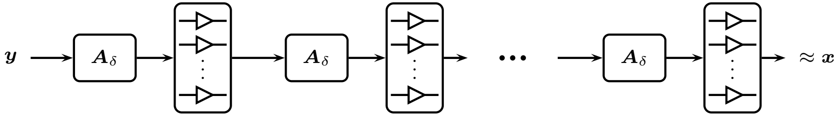
- Widely considered to be impractical (too complex): linear equalization is already one of the most power hungry DSP blocks in coherent receivers
- Complexity increases with the number of steps $M \implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et al., 2011], [Yan et al., 2011], [Napoli et al., 2014], [Secondini et al., 2016], ...)
- Intuitive, but ...

Real-Time Digital Backpropagation



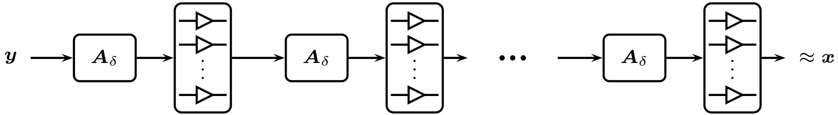
- Widely considered to be impractical (too complex): linear equalization is already one of the most power hungry DSP blocks in coherent receivers
- Complexity increases with the number of steps $M \implies$ reduce M as much as possible (see, e.g., [Du and Lowery, 2010], [Rafique et al., 2011], [Li et 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



- Widely considered to be impractical (too complex): linear equalization is already one of the most power hungry DSP blocks in coherent receivers
- 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
- Machine learning: deep computation graphs tend to work better and can be more parameter efficient than shallow ones

Real-Time Digital Backpropagation



- Widely considered to be impractical (too complex): linear equalization is already one of the most power hungry DSP blocks in coherent receivers
- 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
- Machine learning: deep computation graphs tend to work better and can be more parameter efficient than shallow ones

Main contribution

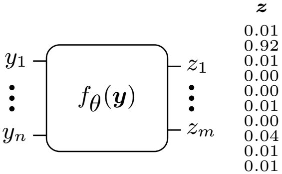
- Joint optimization of all linear steps leads to efficient channel inversion
- Power consumption comparable to published results for linear equalization

Outline

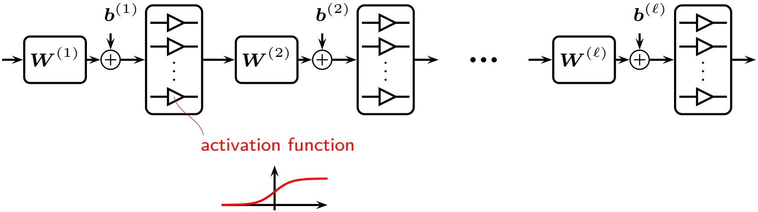
Supervised Learning

Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)

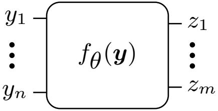


How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks



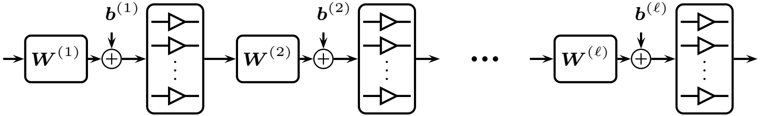
Supervised Learning

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

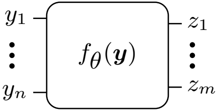


How to optimize $\theta = \{\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(\ell)}, \boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(\ell)}\}$? Deep learning

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

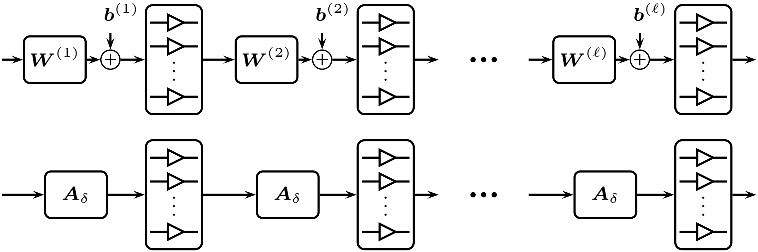
Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)



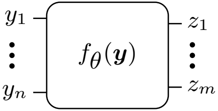
$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

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



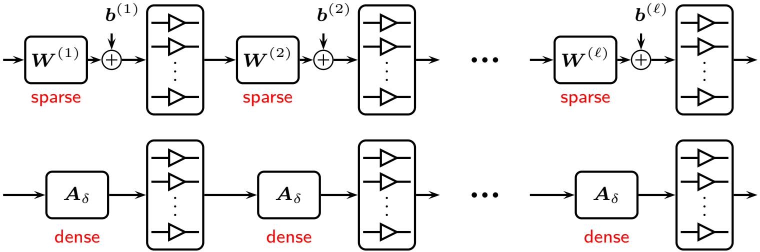
Supervised Learning

handwritten digit recognition (MNIST: 70,000 images)

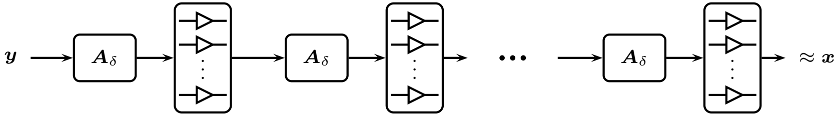


$\boldsymbol{z}$	$\boldsymbol{x}$
0.01	0
0.92	1
0.01	0
0.00	0
0.00	0
0.01	0
0.00	0
0.04	0
0.01	0
0.01	0

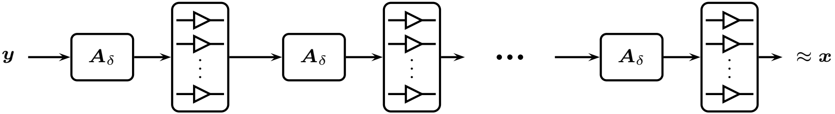
How to choose $f_\theta(\boldsymbol{y})$? Deep feed-forward neural networks

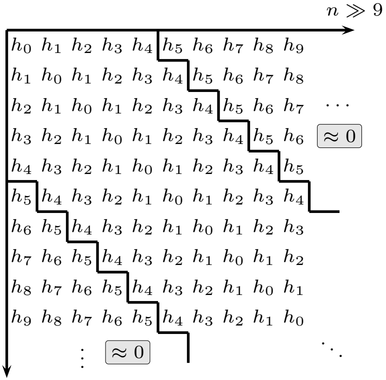


Time-Domain Implementation and Truncation

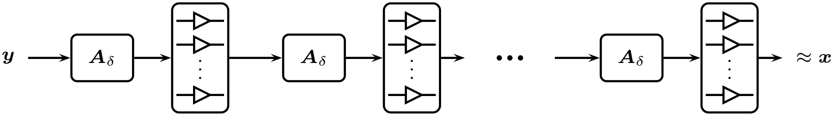


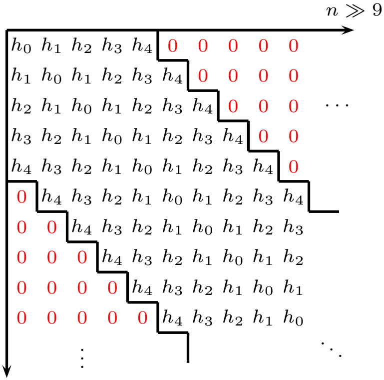
Time-Domain Implementation and Truncation

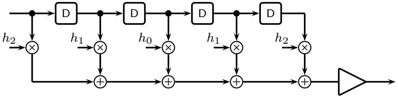




Time-Domain Implementation and Truncation

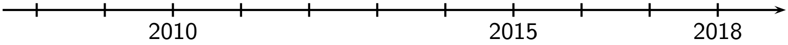






finite impulse response (FIR) filter

Time-Domain Implementation and Truncation

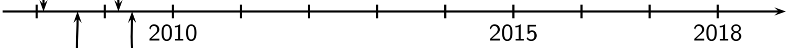


Duke University

Time-Domain Implementation and Truncation

[Li et al., 2008]: "To facilitate real-time implementation, the dispersion . . . operator can be realized using a finite impulse response (FIR) filter"

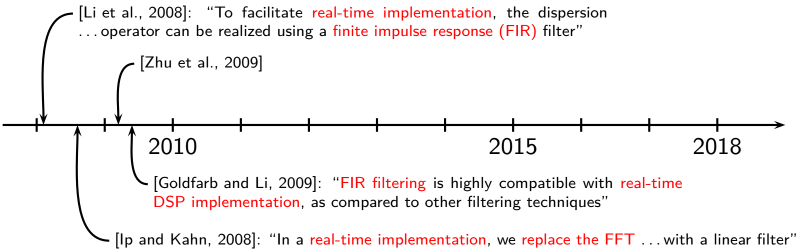
[Zhu et al., 2009]



[Goldfarb and Li, 2009]: "FIR filtering is highly compatible with real-time DSP implementation, as compared to other filtering techniques"

[Ip and Kahn, 2008]: "In a real-time implementation, we replace the FFT . . . with a linear filter"

Time-Domain Implementation and Truncation

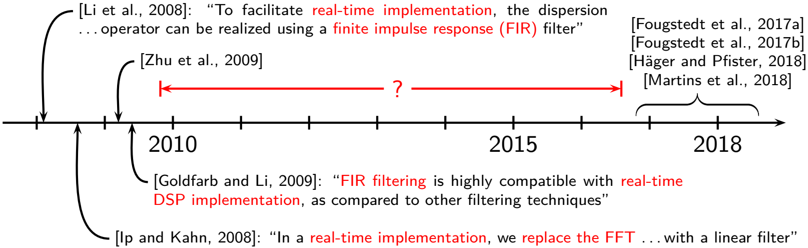


Nontrivial to achieve a good performance–complexity tradeoff!

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000$ km [Ip and Kahn, 2008]

- $\gg 1000$ total filter taps required for good performance

Time-Domain Implementation and Truncation

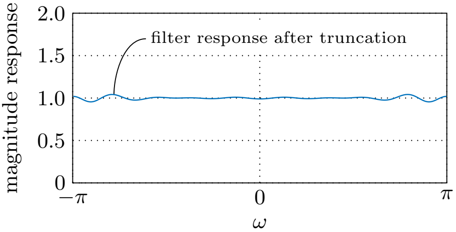


Nontrivial to achieve a good performance–complexity tradeoff!

Example for $R_{\text{symb}} = 10.7$ Gbaud, $L = 2000$ km [Ip and Kahn, 2008]

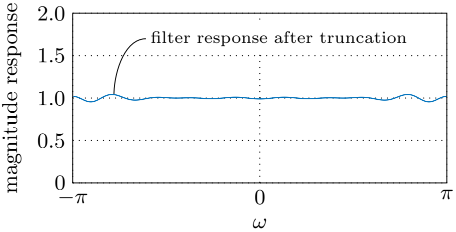
- $\gg 1000$ total filter taps required for good performance

Problem: Truncation Errors

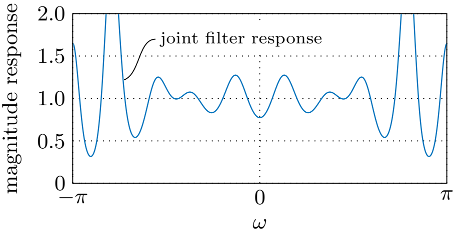


$$\boldsymbol{h}^{(1)} = \boldsymbol{h}^{(2)} = \cdots = \boldsymbol{h}^{(M)}$$

Problem: Truncation Errors

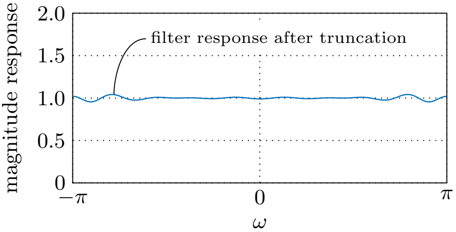


$$\boldsymbol{h}^{(1)} = \boldsymbol{h}^{(2)} = \cdots = \boldsymbol{h}^{(M)}$$

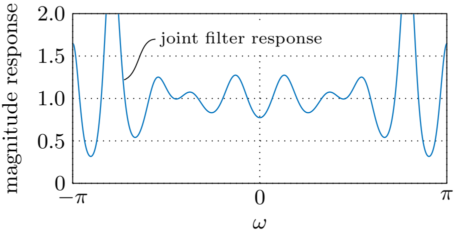


$$\boldsymbol{h}^{(1)} * \boldsymbol{h}^{(2)} * \cdots * \boldsymbol{h}^{(M)}$$

Problem: Truncation Errors



$$\boldsymbol{h}^{(1)} = \boldsymbol{h}^{(2)} = \cdots = \boldsymbol{h}^{(M)}$$



$$\boldsymbol{h}^{(1)} * \boldsymbol{h}^{(2)} * \cdots * \boldsymbol{h}^{(M)}$$

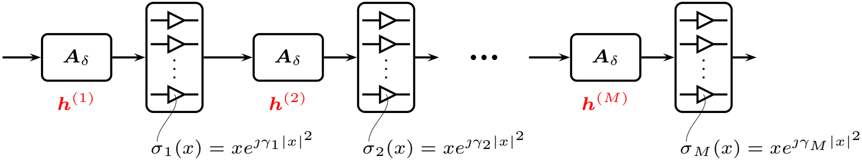
Our approach:
Optimize all M filters jointly

Outline

Joint Filter Optimization

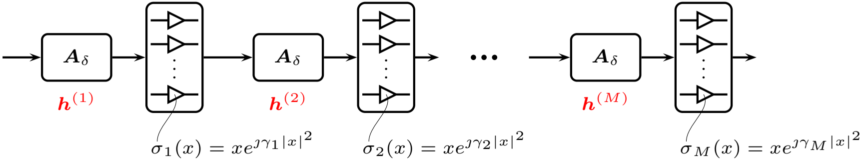
Joint Filter Optimization

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



Joint Filter Optimization

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



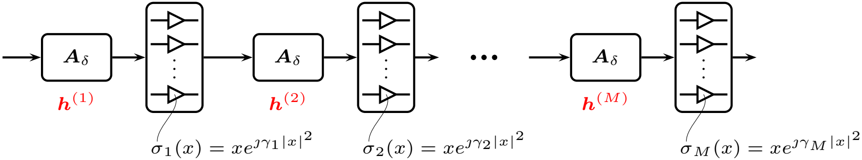
Deep learning of parameters $\theta = \{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(M)}\}$:

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

mean squared error — Adam optimizer, fixed learning rate

Joint Filter Optimization

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



Deep learning of parameters $\theta = \{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(M)}\}$:

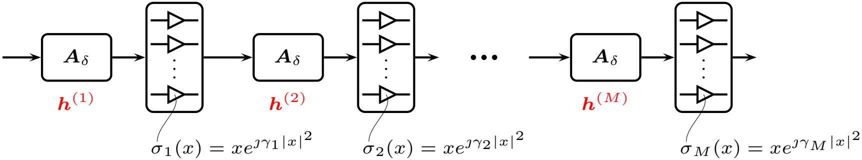
$$\min_\theta \sum_{i=1}^{N} \text{Loss}(f_\theta(\boldsymbol{y}^{(i)}), \boldsymbol{x}^{(i)}) \triangleq g(\theta) \qquad \text{using} \quad \theta_{k+1} = \theta_k - \lambda \nabla_\theta g(\theta_k)$$

mean squared error — Adam optimizer, fixed learning rate

- How to choose the starting point θ_0 and get short filters?

Joint Filter Optimization

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



Deep learning of parameters $\theta = \{\boldsymbol{h}^{(1)}, \dots, \boldsymbol{h}^{(M)}\}$:

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

mean squared error — Adam optimizer, fixed learning rate

- How to choose the starting point θ_0 and get short filters?
- Pre-optimization possible via multi-objective optimization problem

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 $2K' + 1$

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

Iterative Filter Tap Pruning

starting 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

$$\overbrace{\hspace{10em}}^{\text{starting length } 2K'+1} \quad \overbrace{\hspace{6em}}^{\text{target length } 2K+1}$$

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

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

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

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

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

Iterative Filter Tap Pruning

starting length $2K' + 1$

target length $2K + 1$

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

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

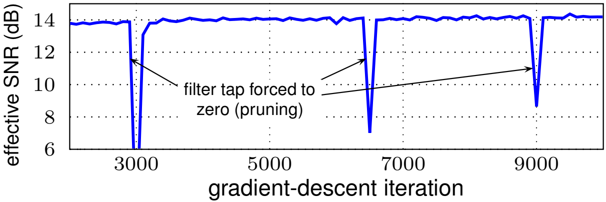
Iterative Filter Tap Pruning

starting length $2K'+1$

target length $2K+1$

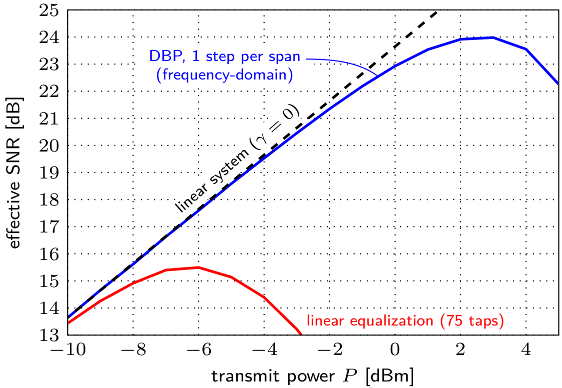
$$\theta = \begin{cases} \boldsymbol{h}^{(1)} = (\cancel{h^{(1)}_{K'}} \; \cdots \; h^{(1)}_{K} \; \cdots \; h^{(1)}_{1} \; h^{(1)}_{0} \; h^{(1)}_{1} \; \cdots \; h^{(1)}_{K} \; \cdots \; \cancel{h^{(1)}_{K'}}) & \text{step } 1 \\ \boldsymbol{h}^{(2)} = (\cancel{h^{(2)}_{K'}} \; \cdots \; h^{(2)}_{K}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

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



Outline

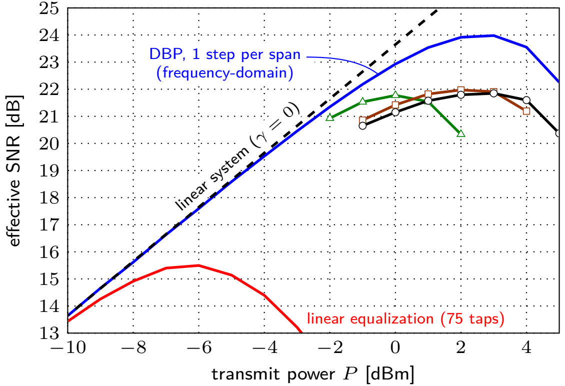
Optimization Results



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.

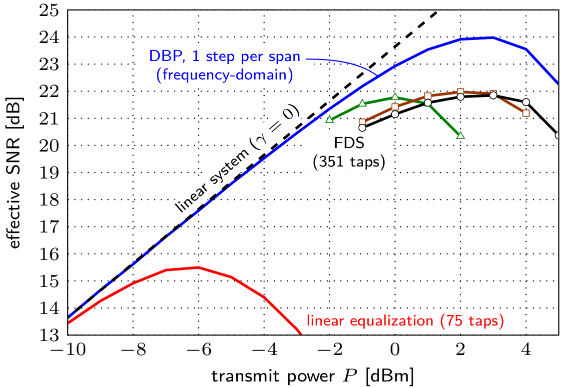
Optimization Results



Parameters similar to [Ip and Kahn, 2008]:

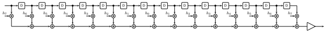
- 25×80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

Optimization Results

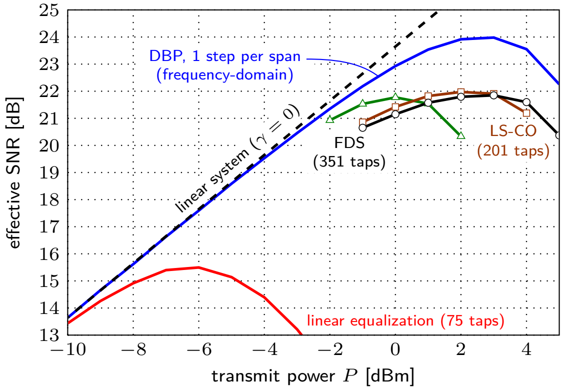


Parameters similar to [Ip and Kahn, 2008]:

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

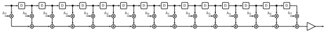
FDS  · · · frequency-domain sampling (15 taps per step)

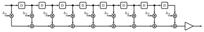
Optimization Results



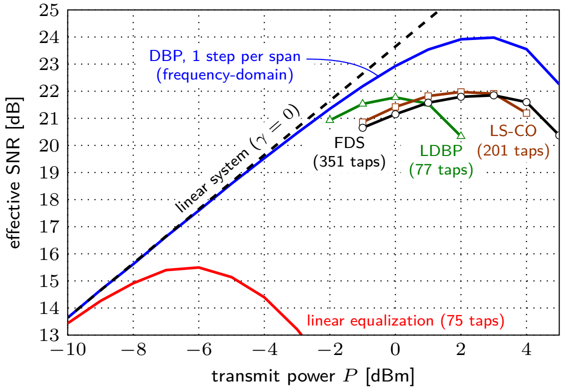
Parameters similar to [Ip and Kahn, 2008]:

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

FDS  · · · frequency-domain sampling (15 taps per step)

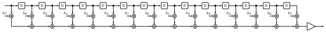
LS-CO  · · · constrained least-squares [Sheikh et al., 2016] (9 taps per step)

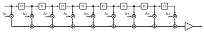
Optimization Results

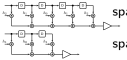


Parameters similar to [Ip and Kahn, 2008]:

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

FDS  · · · frequency-domain sampling (15 taps per step)

LS-CO  · · · constrained least-squares [Sheikh et al., 2016] (9 taps per step)

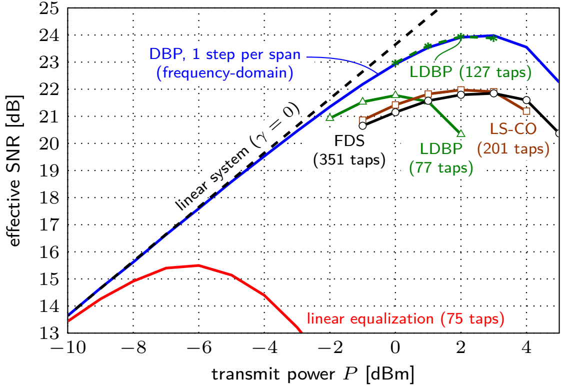
LDBP 

span 1, 3, . . .

span 2, 4, . . .

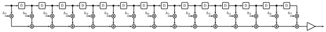
deep learning and pruning (alternate 5 and 3 taps and use different coefficients in all steps)

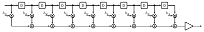
Optimization Results

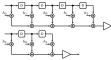


Parameters similar to [Ip and Kahn, 2008]:

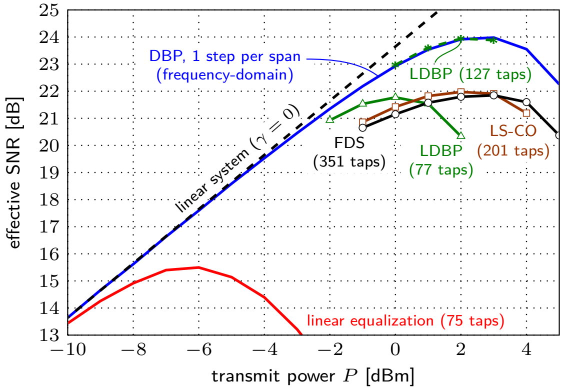
- 25 × 80 km SSFM
- Gaussian modulation
- RRC pulses (0.1 roll-off)
- 10.7 Gbaud
- 2 samples/symbol processing
- single channel, single pol.

FDS  . . . frequency-domain sampling (15 taps per step)

LS-CO  . . . constrained least-squares [Sheikh et al., 2016] (9 taps per step)

LDBP 
span 1, 3, . . .
span 2, 4, . . .
deep learning and pruning (alternate 5 and 3 taps and use different coefficients in all steps)

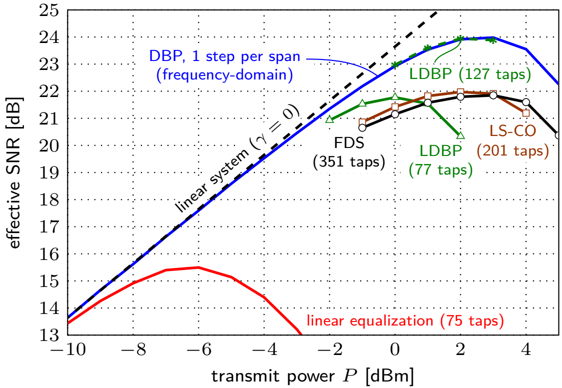
Optimization Results



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.

Optimization Results

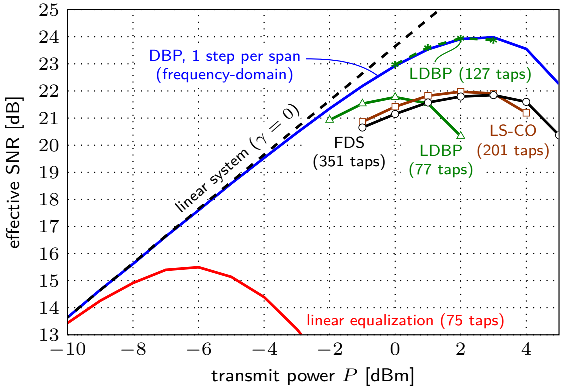


Parameters similar to [Ip and Kahn, 2008]:

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

- 28-nm CMOS synthesis results show $\approx$ 2-fold power and area reduction compared to baseline filters (slightly different system parameters)

Optimization Results



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.

- 28-nm CMOS synthesis results show $\approx$ 2-fold power and area reduction compared to baseline filters (slightly different system parameters)
- Power consumption comparable to linear equalization in [Pillai et al., 2014],[Crivelli et al., 2014]
- Nonlinearities consume $< 20\%$ of the total power

Conclusions

Conclusions

- We have addressed the problem of inverting the nonlinear Schrödinger equation for fiber-optic systems in real time
- Established numerical method (split-step Fourier method) leads to a computation graph reminiscient of a deep feed-forward neural network
- Deep learning in the resulting computation graph can be interpreted as a joint filter (i.e., linear propagator) optimization problem
- This approach requires significantly fewer parameters than previous methods $\implies$ less complexity and power consumption

Conclusions

- We have addressed the problem of inverting the nonlinear Schrödinger equation for fiber-optic systems in real time
- Established numerical method (split-step Fourier method) leads to a computation graph reminiscient of a deep feed-forward neural network
- Deep learning in the resulting computation graph can be interpreted as a joint filter (i.e., linear propagator) optimization problem
- This approach requires significantly fewer parameters than previous methods $\implies$ less complexity and power consumption

Thank you!

FORCE
FIBER-OPTIC COMMUNICATIONS RESEARCH CENTER

Duke UNIVERSITY

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.

Fougstedt, C., Mazur, M., Svensson, L., Eliasson, H., Karlsson, M., and Larsson-Edefors, P. (2017a).
Time-domain digital back propagation: Algorithm and finite-precision implementation aspects.
In *Proc. Optical Fiber Communication Conf. (OFC)*, Los Angeles, CA.

Fougstedt, C., Svensson, L., Mazur, M., Karlsson, M., and Larsson-Edefors, P. (2017b).
Finite-precision optimization of time-domain digital back propagation by inter-symbol interference minimization.
In *Proc. European Conf. Optical Communication*, Gothenburg, Sweden.

Goldfarb, G. and Li, G. (2009).
Efficient backward-propagation using wavelet- based filtering for fiber backward-propagation.
Opt. Express, 17(11):814–816.

References II

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

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

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)*, page OWW3, 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.

Martins, C. S., Bertignono, L., Nespola, A., Carena, A., Guiomar, F. P., and Pinto, A. N. (2018).
Efficient time-domain DBP using random step-size and multi-band quantization.
In *Proc. Optical Fiber Communication Conf. (OFC)*, San Diego, 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).

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

References III

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.

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.

Zhu, L., Li, X., Mateo, E., and Li, G. (2009).
Complementary FIR filter pair for distributed impairment compensation of WDM fiber transmission.
IEEE Photon. Technol. Lett., 21(5):292–294.