# Machine learning for linear and nonlinear signal equalization

Course 34242
Lecture 4

Darko Zibar
dazi@dtu.dk

# Agenda

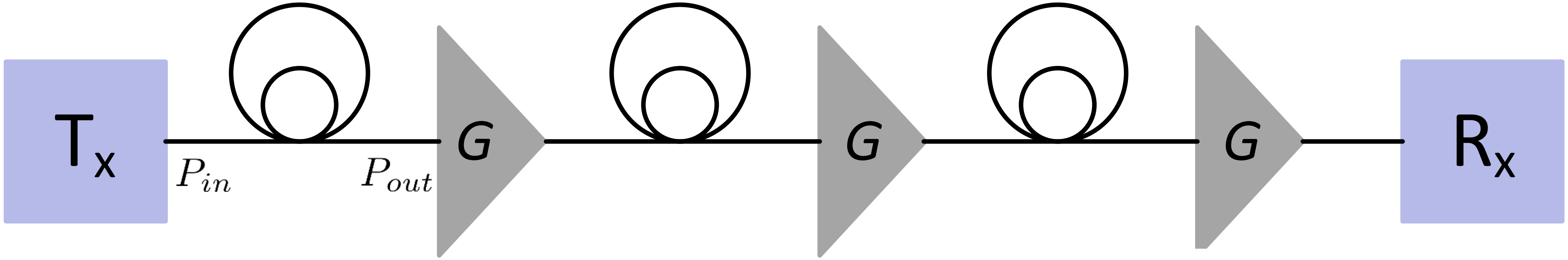
- Basics of digital communication
- Machine learning for communication system optimization
- Linear and nonlinear channels
- ISI free communication
- Linear and nonlinear filters for equalization
- Learning algorithm for linear filters
- Learning algorithm for nonlinear filters

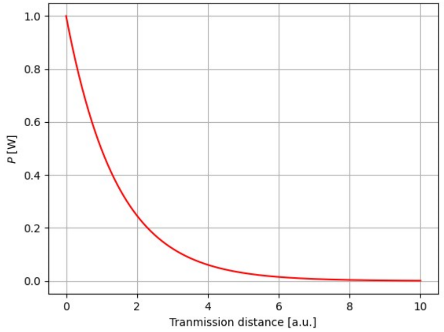
# Reading material

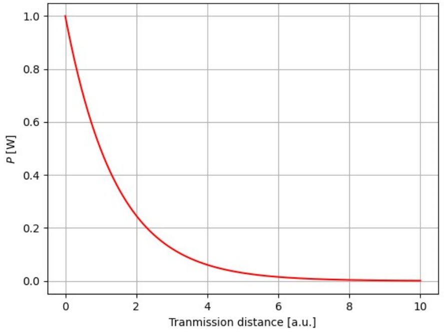
1. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer 2006
    - Chapter 3 (pp. 137 - 143)
    - Chapter 5 (pp. 232 - 246)

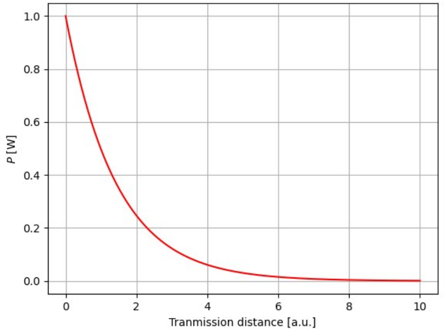
2. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, 2016
    - Chapter 8

# Optical transmission system



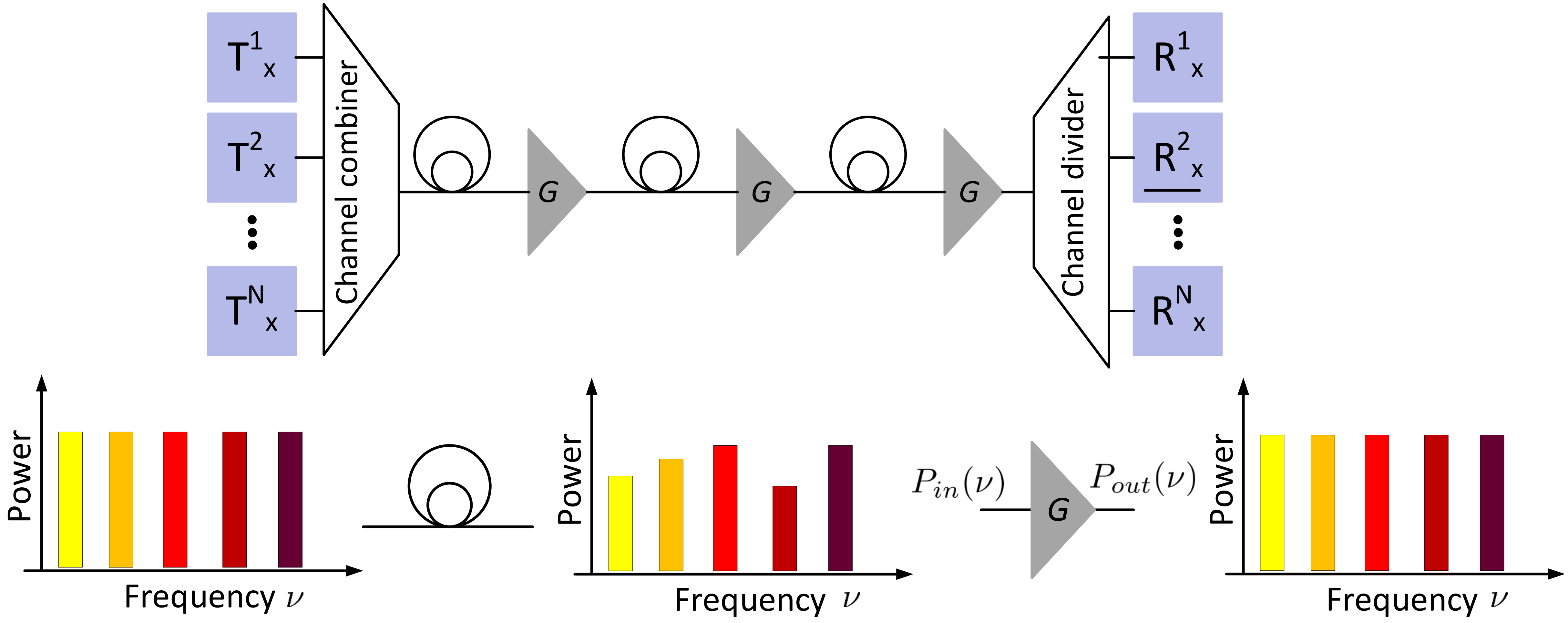

$$\boxed{P_{out}(z) = P_{in} e^{-\alpha z}}$$

$$\boxed{P_{in} = G P_{out}(L)}$$

$\alpha$ : fiber attenuation

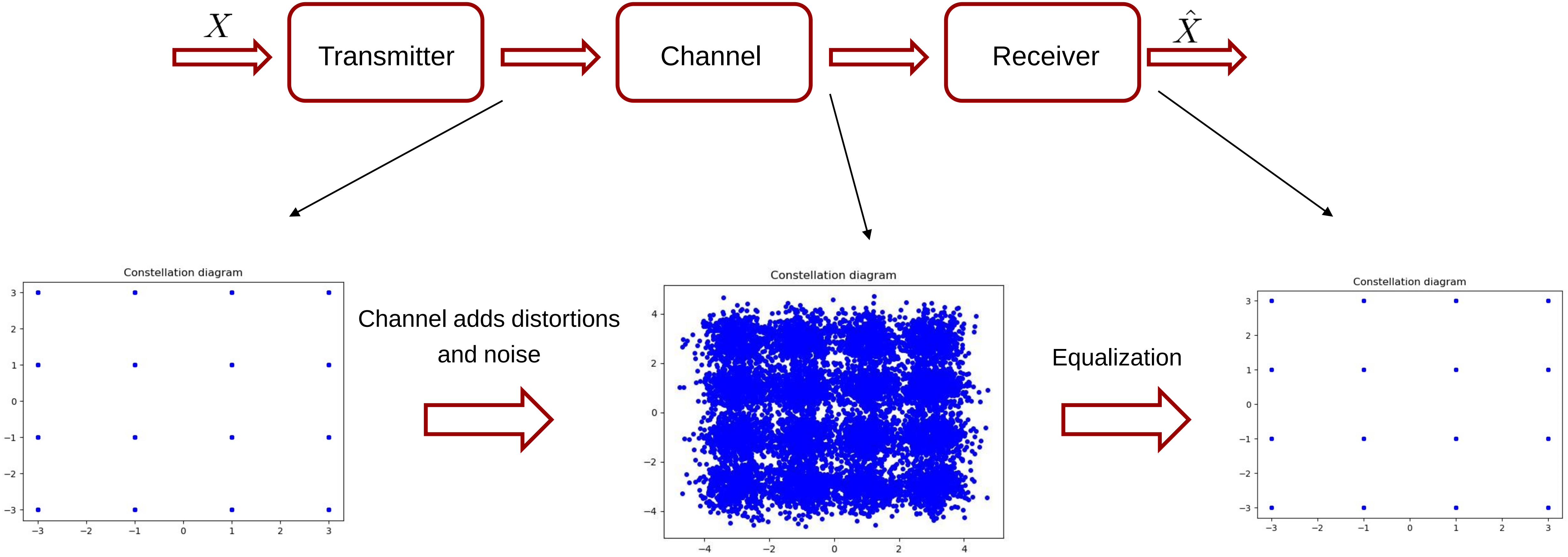
$z$ : variable (distance)

# Multi-channel transmission system

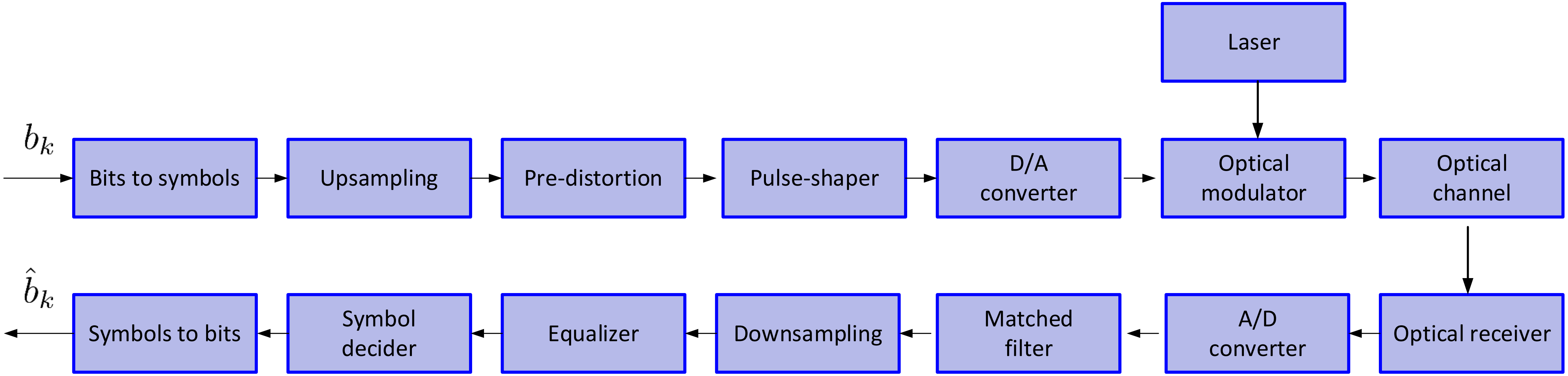



$$G(\nu) = \frac{P_{out}(\nu)}{P_{in}(\nu)}$$

# Signal distortion in data-communication



# Fiber-optic communication systems




Machine learning can be used to perform global optimization in the presence of system impairments

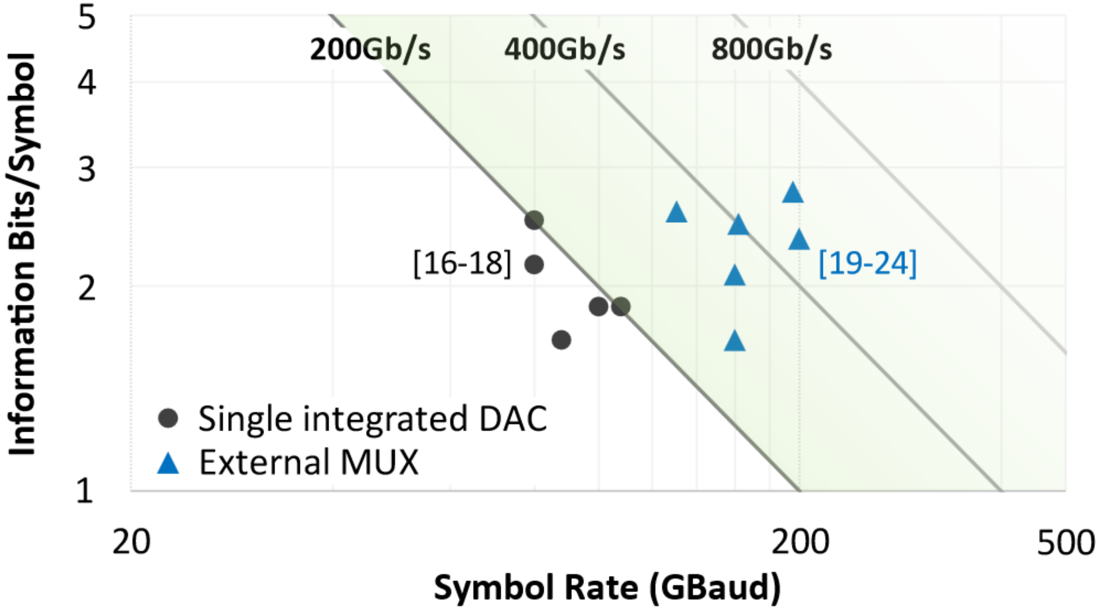
# System Impairments

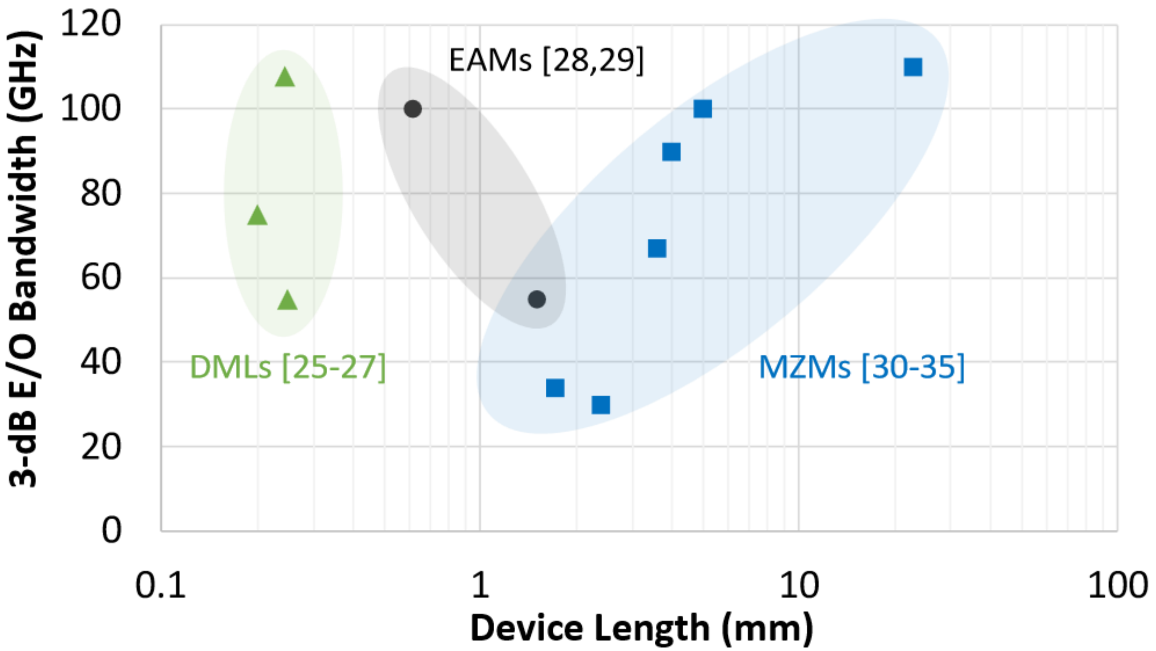
- Transmitter
  - DAC bandwidth limitations
  - DAC resolution
  - Chirp
  - Optical modulator nonlinearity (MZM, EAM, ring resonator)

- Optical fibre channel
  - Chromatic dispersion (ISI)
  - Kerr nonlinearity
  - Polarization mode dispersion
  - Optical amplifier noise

- Receiver
  - Photodiode  bandwidth limitations
  - ADC resolution
  - Front-end noise

# Next generation data-centre links



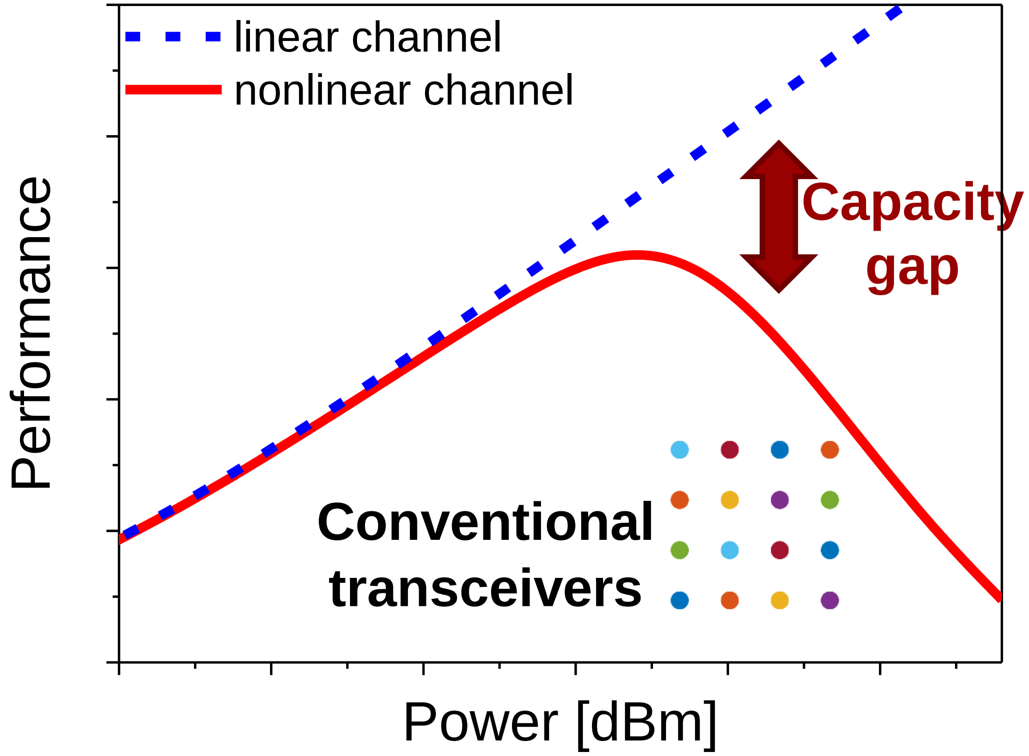




Next-generation will operate of links will operate > 800 Gb/s
Innovative DSP for compensation of strong ISI and component nonlinearity

Che et al., JLT 2023

# How to decrease the capacity gap?

- Capacity of fibre-channel **unknown**
- Optimal transceiver **unknown**
- Optimal receiver architecture **unknown**
- Optimal modulation and pulse-shapes **unknown**

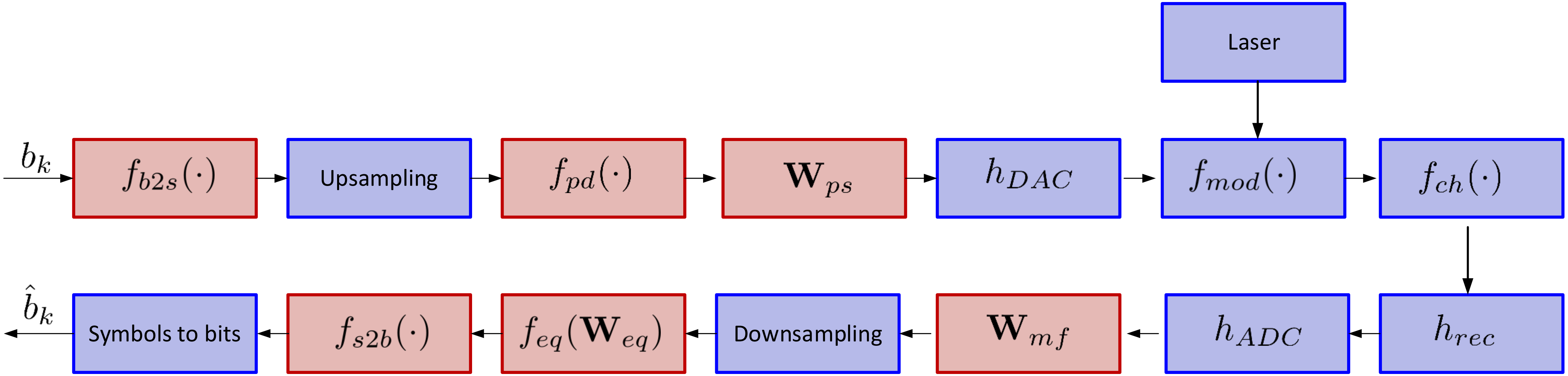



If something is complex, has no analytical solutions and it complex to optimize we turn to “the dark side”

# End-to-end Learning in Optical Communication

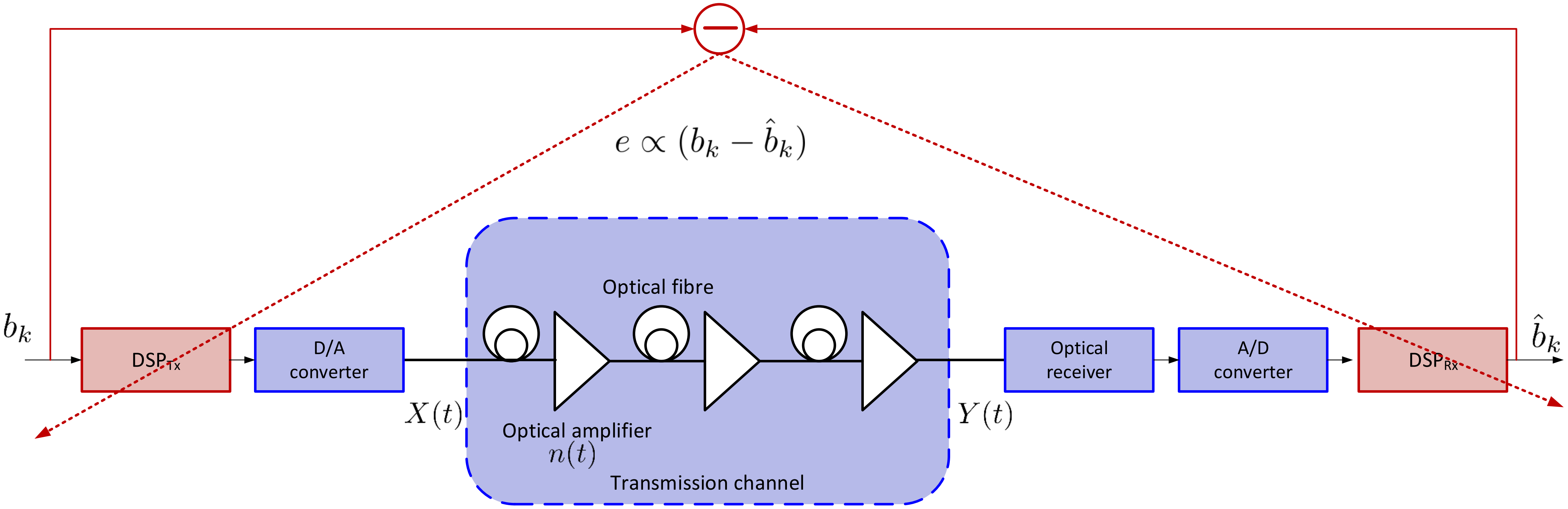
- Geometric constellation shaping[1-18]
  - [1] R. T. Jones, et. al., ECOC, 2018.
  - [2] R. T. Jones, et. al., ECOC, 2019.
  - [3] S. Li, et. al., ECOC, 2018.
  - [4] M. Schaedler, et. al., OFC, 2020.
  - [5] K. Gümüş, et. al., OFC, 2020.
  - [6] V. Talreja, et. al., ECOC, 2020.
  - [7] V. Neskorniuk, et. al., ECOC, 2021.
  - [8] O. Jovanovic, et. al., ECOC, 2021.
  - [9] V. Aref, et. al., OFC, 2022.
  - [10] A. Rode, et. al., OFC, 2022.
  - [11] O. Jovanovic, et. al., JLT, 2022.
  - [12] B. M. Oliveira, et. al., CLEO, 2022.
  - [13] X. Guan, et. al., CLEO, 2022.
  - [14] V. Neskorniuk, et. al., CLEO, 2022.
  - [15] A. Rode, et. al., ECOC, 2022.
  - [16] M. P. Yankov, et. al., ECOC, 2022.
  - [17] B. M. Oliveira, et. al., Optics Express, 2022.
  - [18] V. Neskorniuk, et. al., Optics Express, 2023.
- Waveforms for dispersive fiber[19-21]
  - [19] B. Karanov, et. al., JLT, 2018.
  - [20] B. Karanov, et. al., Optics Express, 2019.
  - [21] B. Karanov, et. al., OFC, 2021.
- Waveforms for nonlinear frequency division multiplexing[22,23]
  - [22] S. Gaiarin, et. al., CLEO, 2020.
  - [23] S. Gaiarin, et. al., JLT, 2021.
- Superchannel transmission[24,25]
  - [24] J. Song, et. al., OFC, 2021.
  - [25] J. Song, et. al., JSTQE, 2022.
- Experimental test-bed using a generative model[26]
  - [26] B. Karanov, et. al., OFC, 2020.
- Gradient-free optimization for non-differentiable channels[27,28]
  - [27] O. Jovanovic, et. al., JLT, 2021.
  - [28] J. Song, et al., ECOC, 2021.

# Degrees of freedom for the optimization



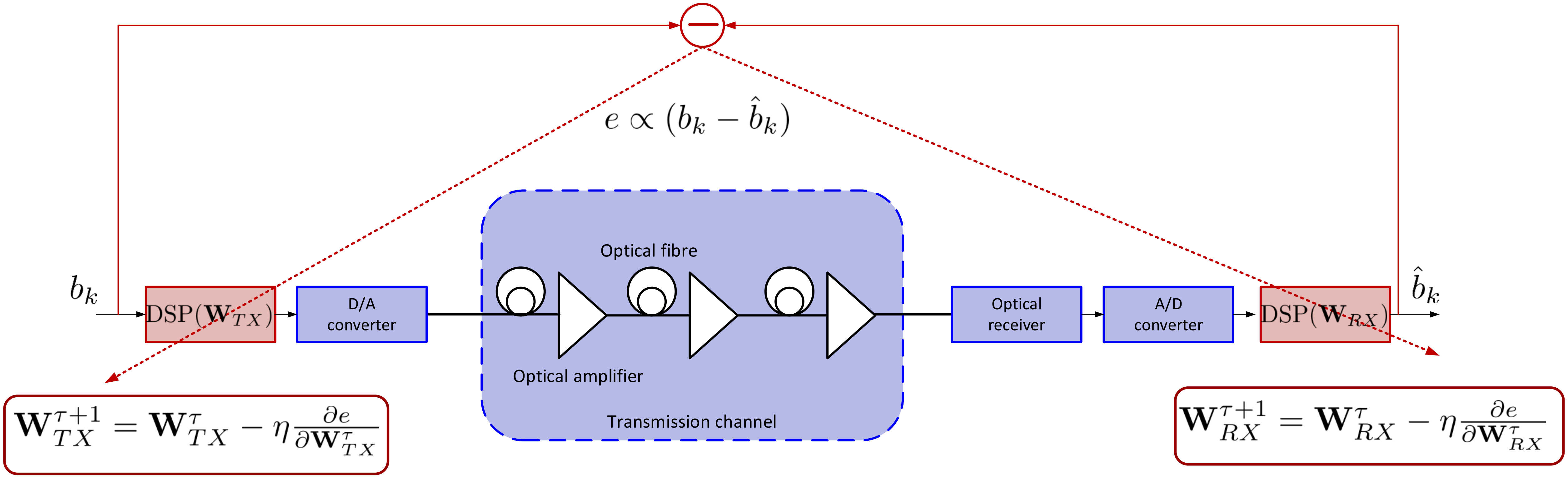

End-to-end learning allows for joint optimization of the transmitter and receiver side DSP blocks

# Long haul fiber-optic communication channel



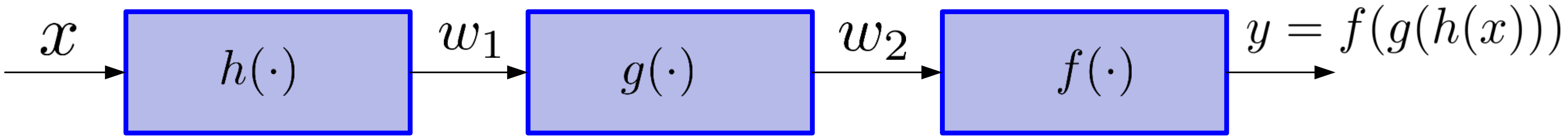

$$\frac{dX(t)}{dz} = \underbrace{-\frac{\alpha}{2}X(t)}_{\text{Fiber loss}} - i\overbrace{\frac{\beta_2}{2}\frac{d^2X(t)}{dt^2}}^{\text{Chromatic dispersion}} + i\gamma\underbrace{|X(t)|^2}_{\text{Nonlinear Kerr effect}}X(t) \Rightarrow \boxed{Y(t) = f(X(t), n(t))}$$

# Tx and Rx optimization




End-to-end learning requires computation of gradients through the system

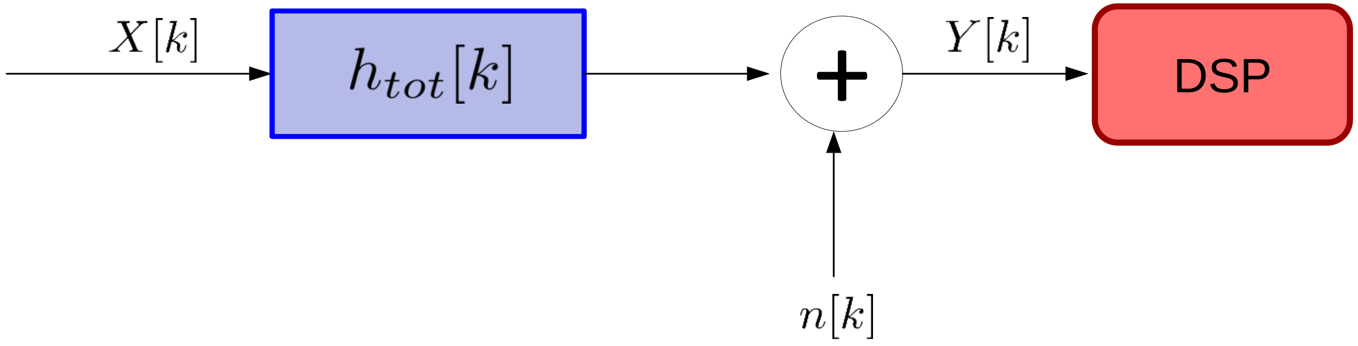
# All we need is the chain rule



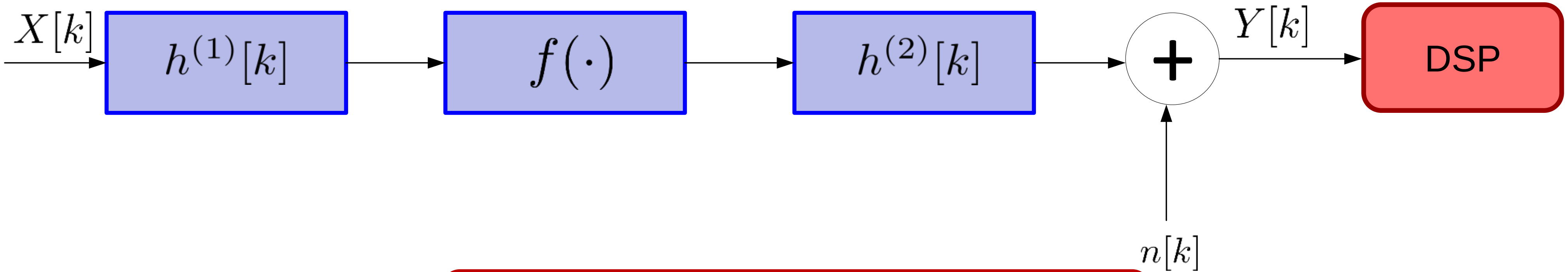

$$
\begin{aligned}
y &= f(w_2) \\
w_2 &= g(w_1) \\
w_1 &= h(x)
\end{aligned}
$$

$$\boxed{\frac{dy}{dx} = \frac{dy}{dw_2}\frac{dw_2}{dw_1}\frac{dw_1}{dx}}$$

# Linear and nonlinear channel models

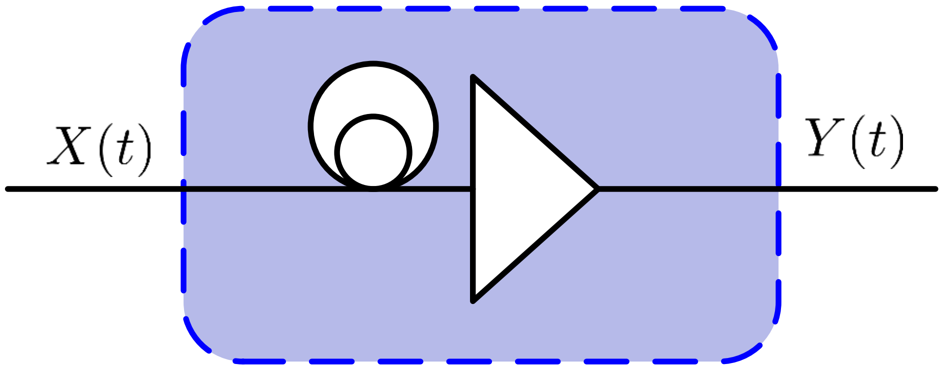



$$Y[k] = h_{tot}[k] * X[k] + n[k]$$

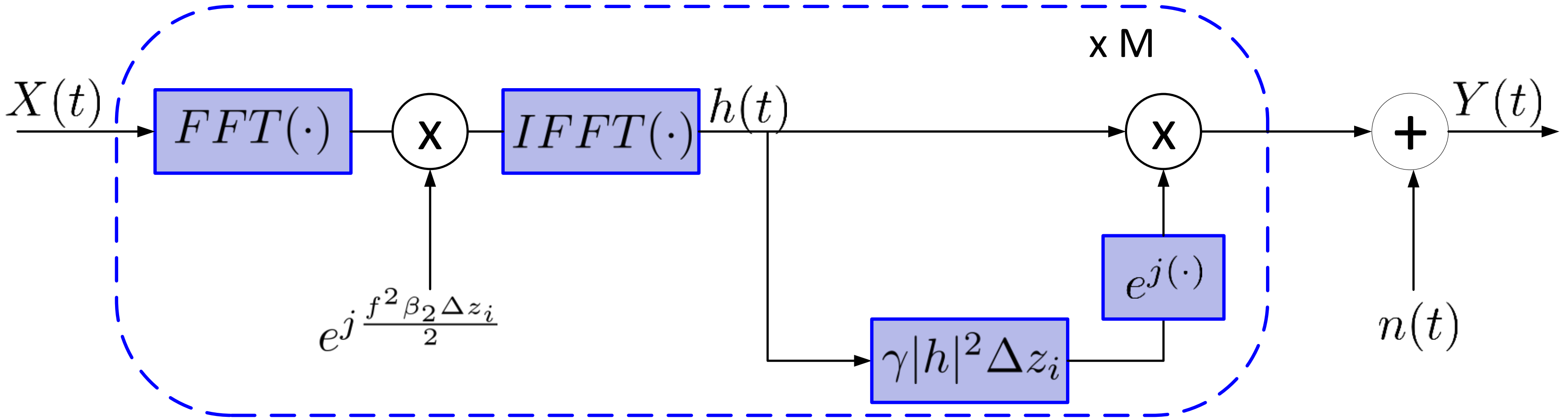



$$Y[k] = h^{(2)} * (f(h^{(1)}[k] * X[k])) + n[k]$$

# Optical fiber channel model




$$\frac{dX}{dz} = -\frac{\alpha}{2}X - i\frac{\beta_2}{2}\frac{d^2X}{dt^2} + i\gamma|X|^2X$$



# Auxiliary channel model for the optical fiber

$$\frac{dA}{dz} = \underbrace{-\frac{\alpha}{2}A}_{\text{Fiber loss}} \overbrace{- i\frac{\beta_2}{2}\frac{d^2A}{dt^2}}^{\text{Chromatic dispersion}} \underbrace{+ i\gamma|A|^2A}_{\text{Nonlinear Kerr effect}}$$

The nonlinear interference noise (NLIN) model:

$$y[k] = x[k] + n[k]$$

$$n[k] \sim N(0, \sigma^2_{\mathrm{ASE}} + \sigma^2_{\mathrm{NLIN/GN}}(P_{\mathrm{Tx}}, \mu_6, C))$$

Average power

Modulation property (peak power)

*Dual power constraint – nonobvious optimal characteristics and optimization strategies*

R. Dar et al., Opt. Exp. 21(22) (2013), pp. 25685-25699

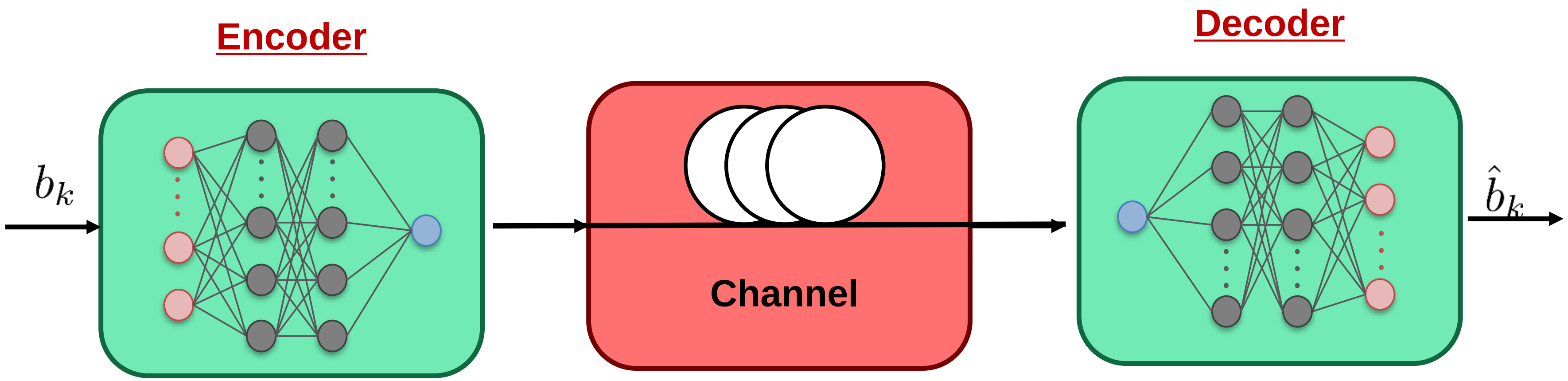
# Optical fiber channel models

$$\frac{dA}{dz} = \underbrace{-\frac{\alpha}{2}A}_{\text{Fiber loss}} \overbrace{- i\frac{\beta_2}{2}\frac{d^2A}{dt^2}}^{\text{Chromatic dispersion}} \underbrace{+ i\gamma|A|^2A}_{\text{Nonlinear Kerr effect}}$$

| | Additive white Gaussian noise (AWGN) | Nonlinear interference noise (NLIN)[1] | Split-step Fourier method (SSFM) [2] |
|---|---|---|---|
| Model output | Symbol observations | | Complete WDM waveform |
| Modelling of channel memory | No | No | Yes |
| Modelling of modulation dependence | No | Yes | Yes |
| Computational cost | Low | Low | High |

[1] R. Dar et al., Opt. Exp,2013.
[2] O. V. Sinkin et al., JLT, 2003.

# Geometric constellation shaping



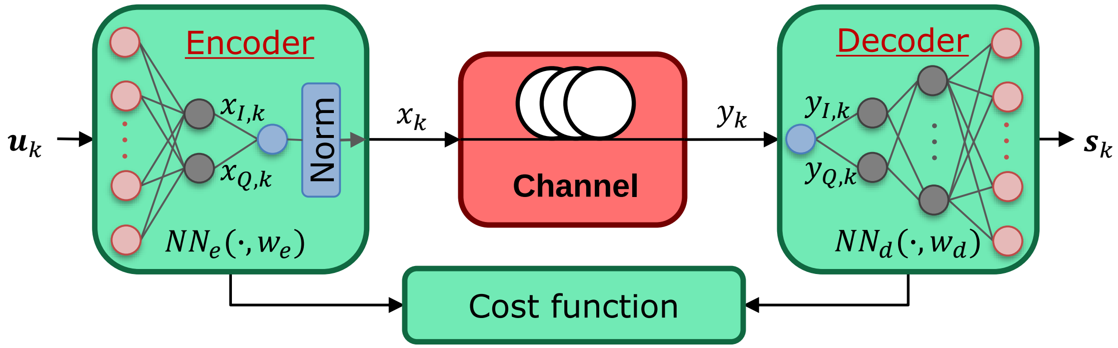

*Tx NN finds a robust symbol mapping*

*RX NN reconstruct the transmitted symbols*

Joint optimization of Tx and Rx neural networks leads to constellations that are robust to channel impairments

[1] T. O'Shea, et. al., Trans. on Cognitive Comm. and Networking, 2017

# Cost function and relation to achievable information rate




– For classification problems (e.g. symbol detection), cross-entropy (CE) is commonly used

$$J_{CE}(\boldsymbol{W}) = \mathbb{E}_k\left[-\sum u_k \log s_k\right]$$

– For binary classification problems (e.g. bit demapping), log-likelihood (LL) is commonly used

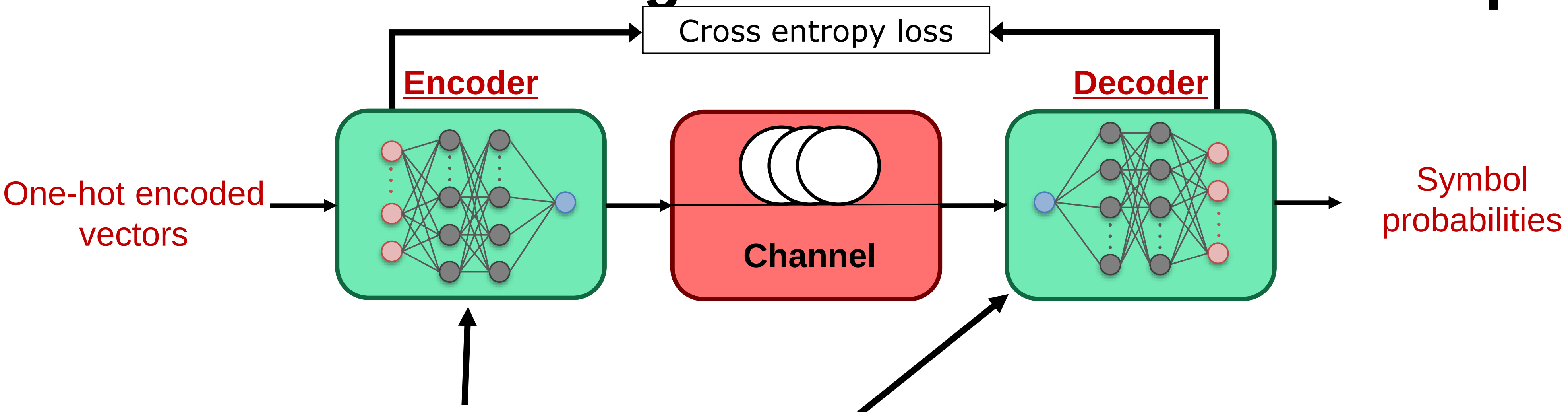
$$J_{LL}(\boldsymbol{W}) = \mathbb{E}_k\left[-\sum [u_k \log s_k + (1-u_k)\log(1-s_k)]\right]$$

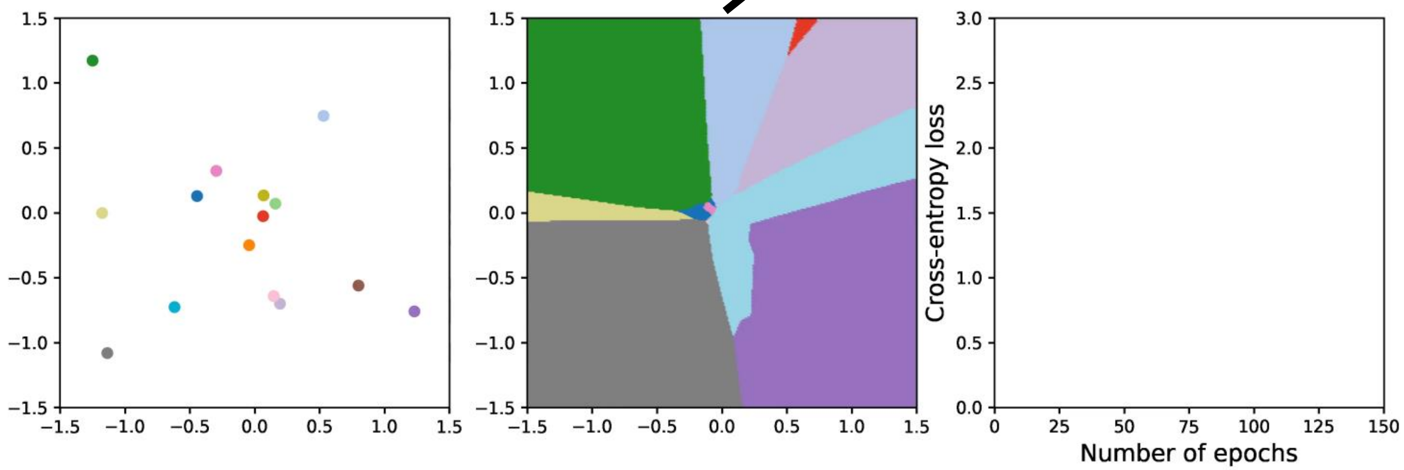
– Relation to achievable information rate (AIR):

$$I(X;Y) = H(X) - H(X|Y) \geq H(X) - \hat{H}(X|Y)$$

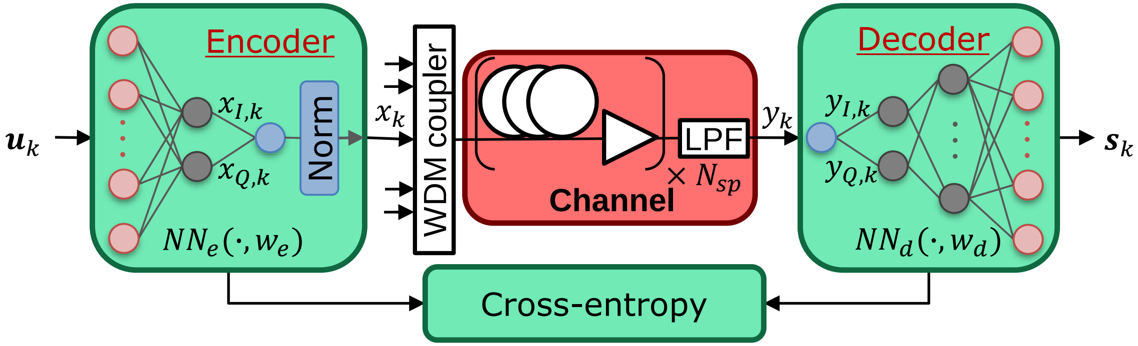
Achievable information rate when using decoder neural network

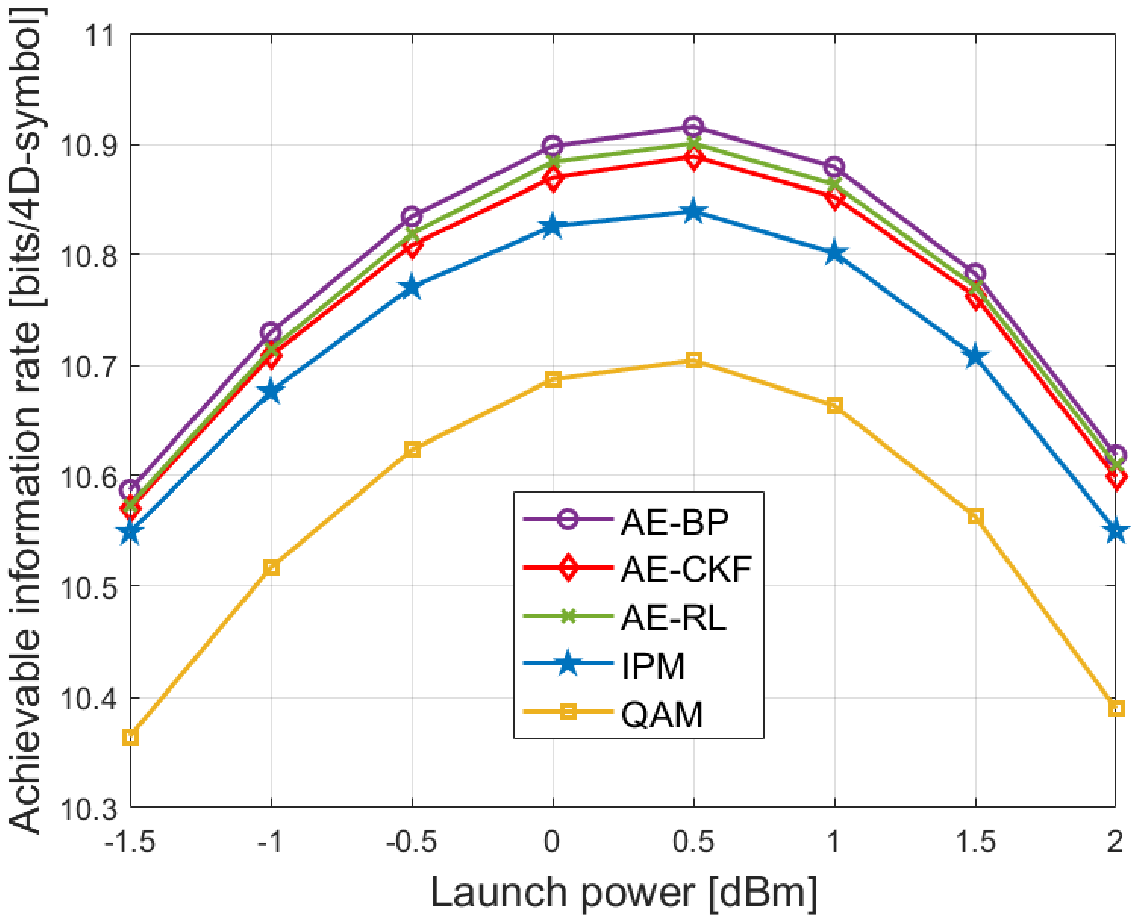
# Autoencoders for geometric constellation shaping



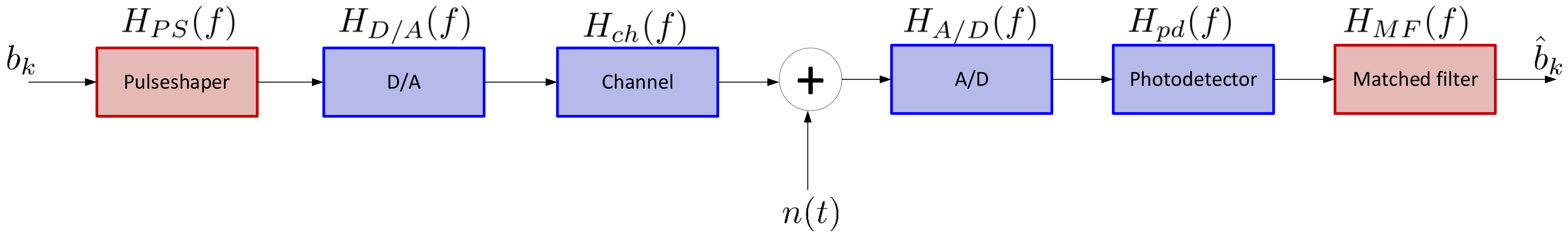



# System performance



# Communication system for zero ISI



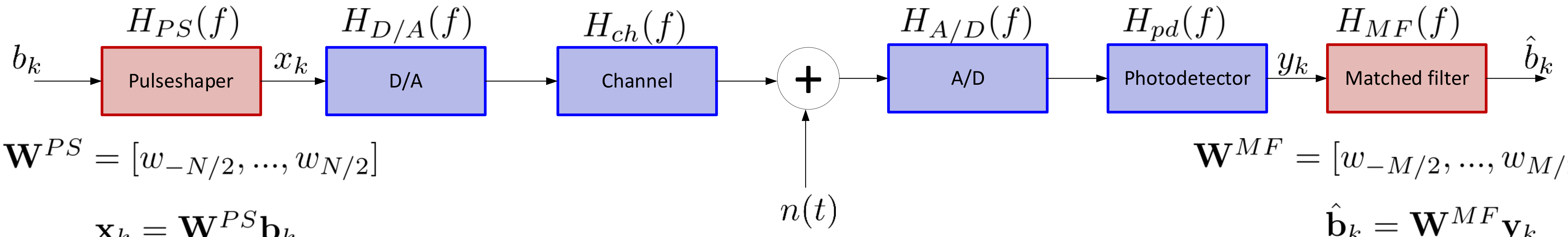

Total transfer function:

$$\boxed{X(f) = H_{PS}(f)H_{D/A}(f)H_{ch}(f)H_{A/D}(f)H_{pd}(f)H_{MF}(f)}$$

ISI-free condition:

$$\boxed{\sum_{m=\infty}^{\infty} X(f + m/T) = C}$$

# Optimization of pulse-shaper and matched filter




$$\mathbf{W}^{PS} = [w_{-N/2}, \ldots, w_{N/2}]$$

$$\mathbf{x}_k = \mathbf{W}^{PS}\mathbf{b}_k$$

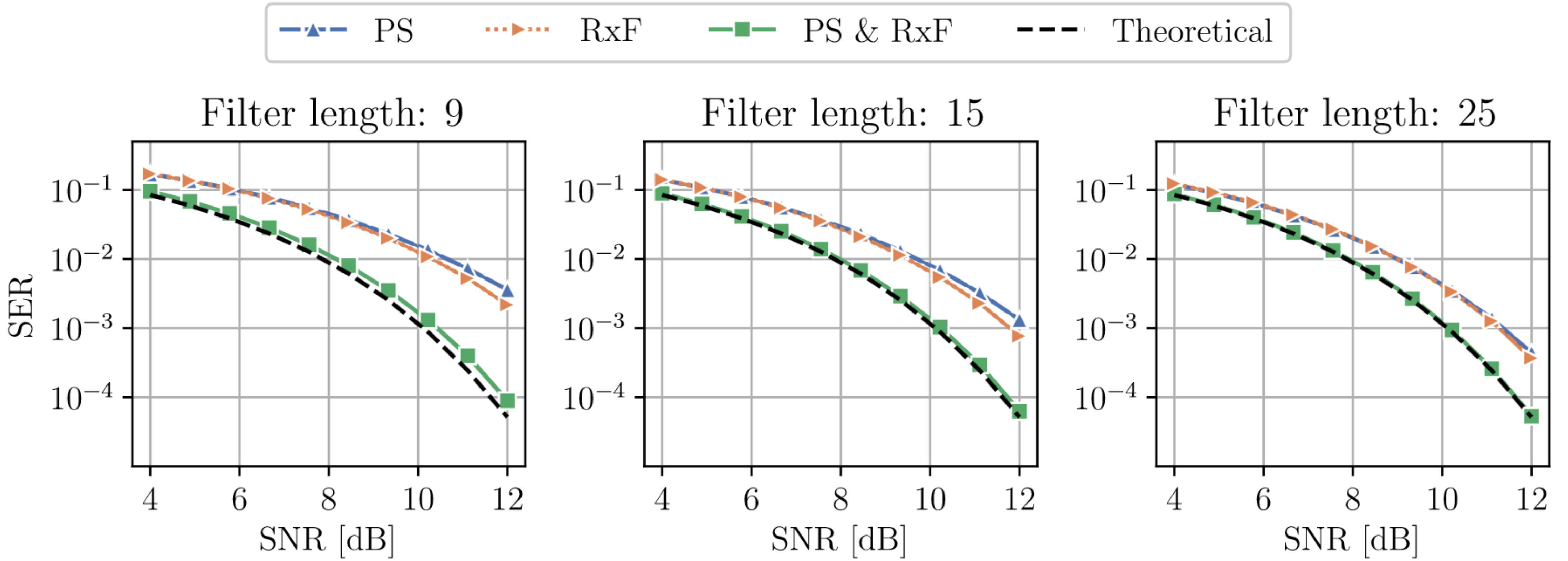
$$\mathbf{W}^{MF} = [w_{-M/2}, \ldots, w_{M/2}]$$

$$\hat{\mathbf{b}}_k = \mathbf{W}^{MF}\mathbf{y}_k$$

Define the error: $e = (b_k - \hat{b}_k)^2$

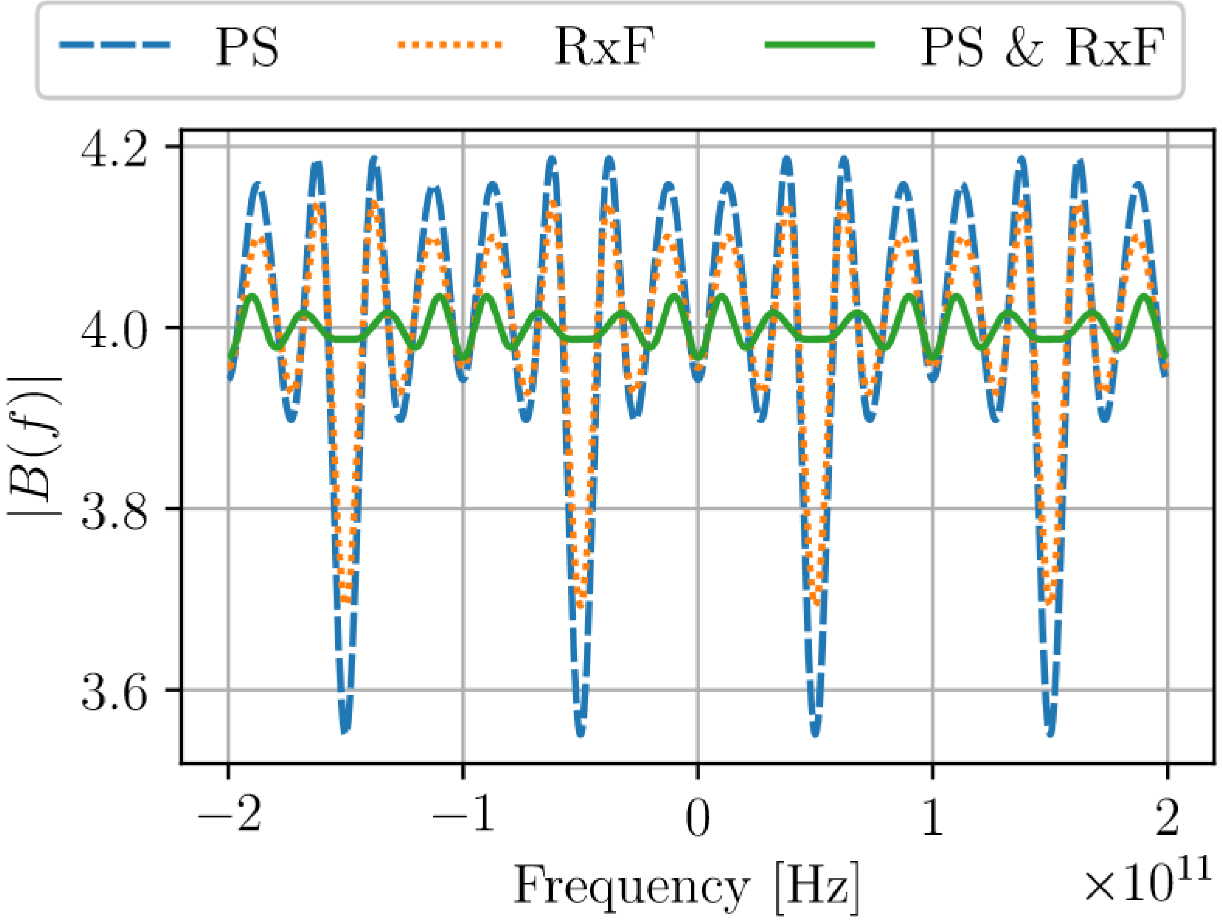
$$\mathbf{W}^{PS}_{\tau+1} = \mathbf{W}^{PS}_{\tau} - \eta\frac{\partial e}{\partial \mathbf{W}^{PS}_{\tau}}$$
$$\mathbf{W}^{MF}_{\tau+1} = \mathbf{W}^{MF}_{\tau} - \eta\frac{\partial e}{\partial \mathbf{W}^{MF}_{\tau}}$$

# Back-to-back results




S. F. Nielsen, JLT 2024

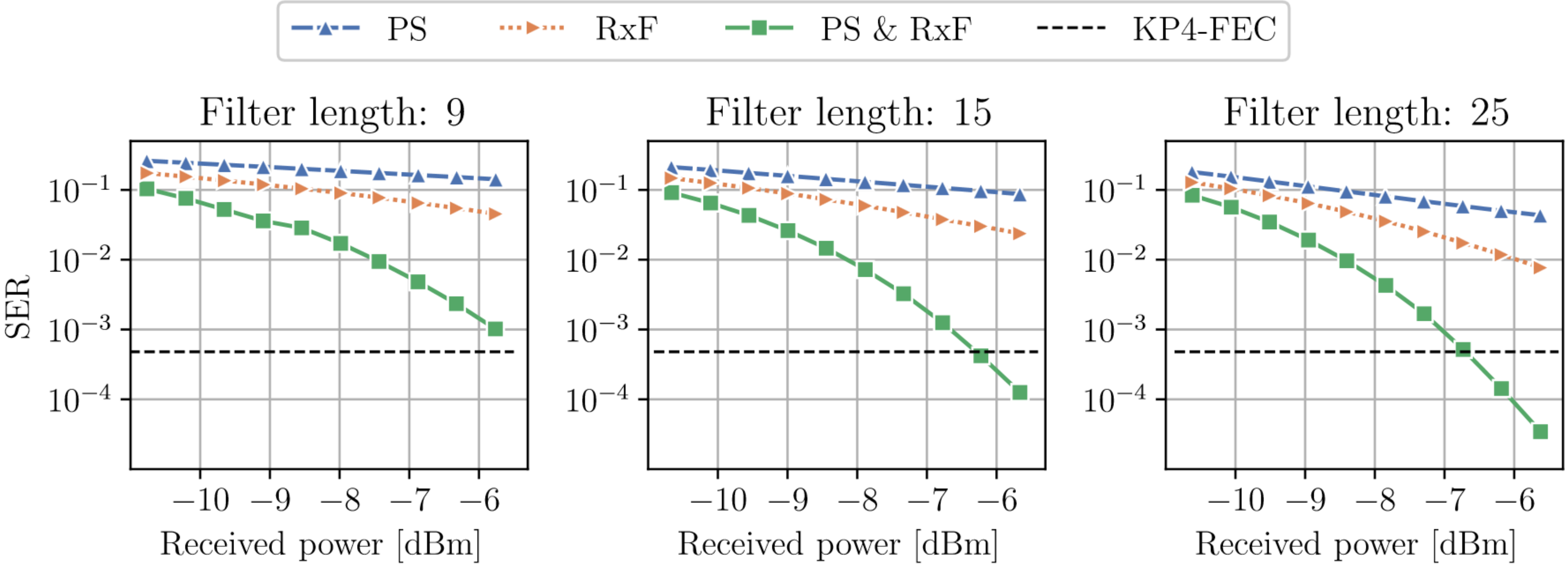
# Total transfer function




$$B(f) = \sum_{m=\infty}^{\infty} X(f + m/T) = C$$

S. F. Nielsen, JLT 2024

# After 2 km of transmission

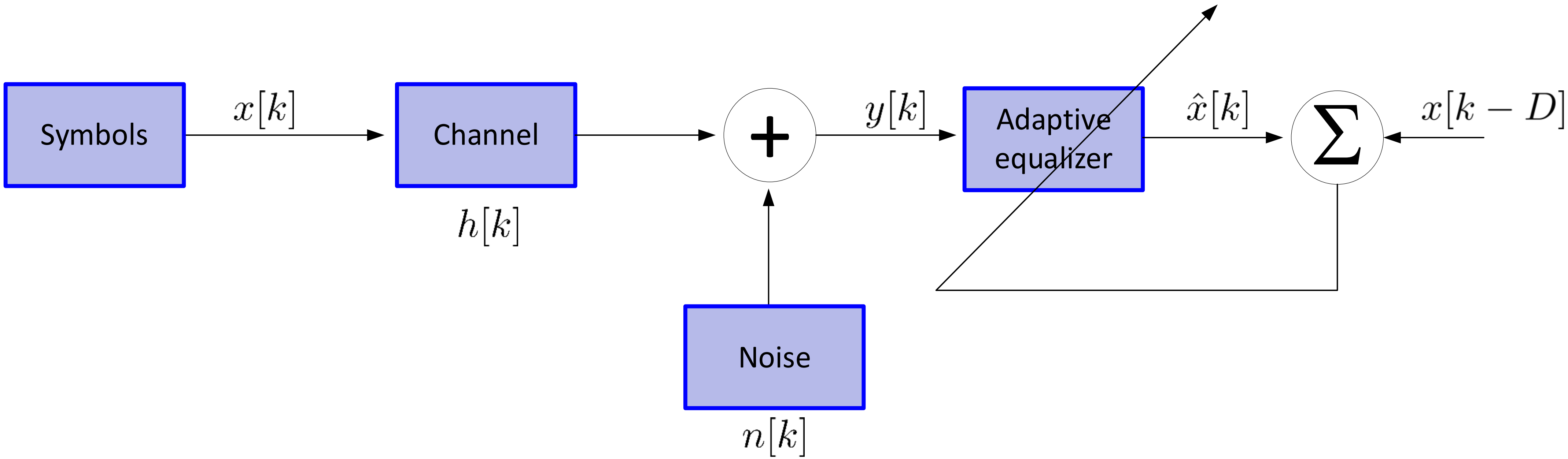



S. F. Nielsen, JLT 2024

# Summary

- Communication theory for linear channels well established
- Communication theory for non-linear channels not well established
- Many blocks within communication system are learnable
- Machine learning can help us learn models from data
- Significant advantages already demonstrated
- Efficient Gradient-free optimization needs to be developed  for experiments

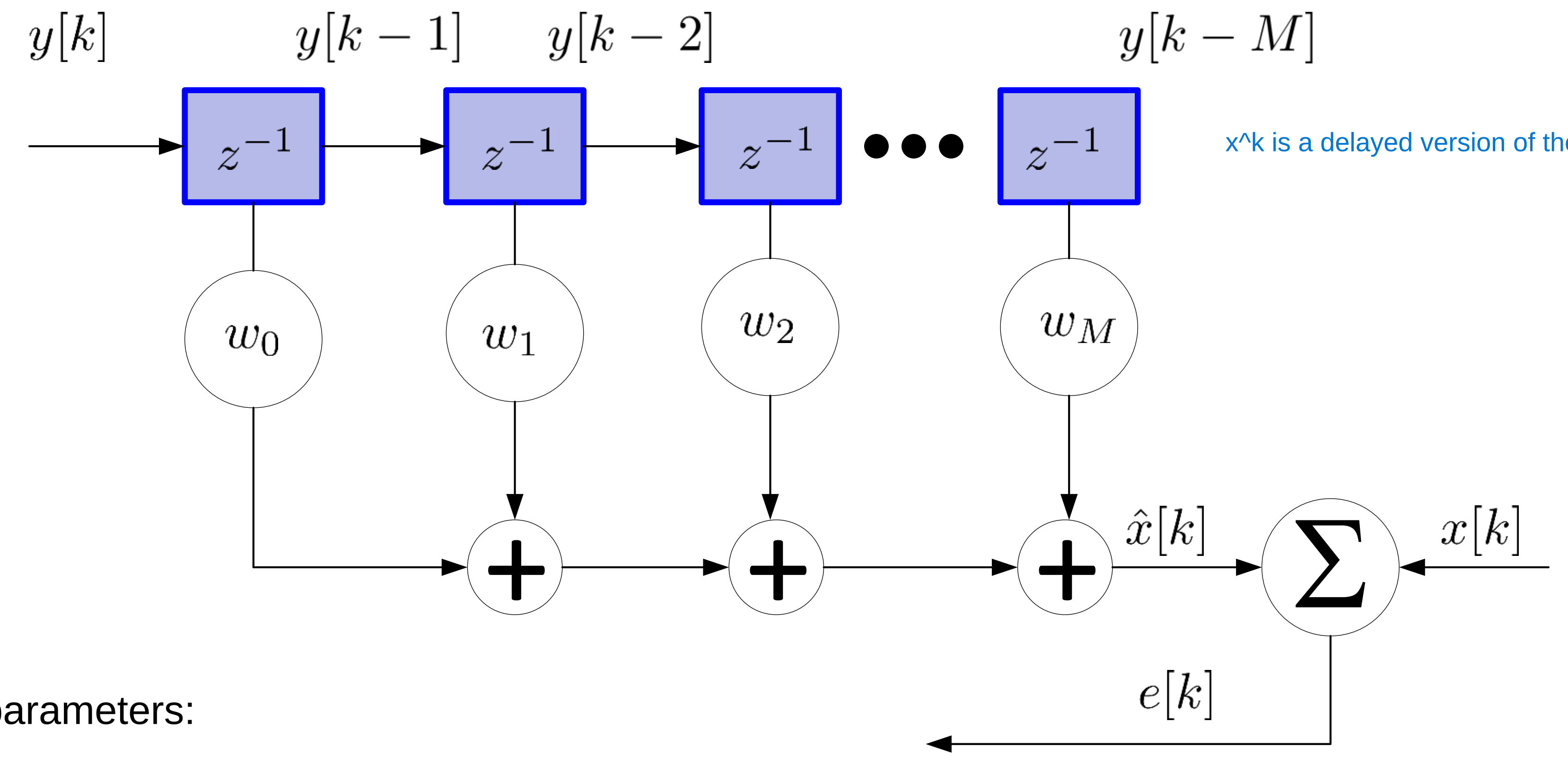
# Discrete-time linear communications system model




$$\boxed{y[k] = h[k] * x[k] + n[k] = \sum_{m=1}^{M} h[m]x[k-m] + n[k]}$$

$$h[k] = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{2\pi}{W}(k-2)\right)\right] & \text{for } k = 1, 2, 3 \\ 0 & \text{otherwise} \end{cases}$$

# Linear adaptive equalizer




x^k is a delayed version of the symbol x

Adaptable model parameters:

$$\mathbf{w} = [w_0, w_1, w_2, ...w_M]^T$$

The objective if to determine weight vector **W** by minimizing the error

# Deriving the update algorithm

Weight vector updated using gradient descent:

$$\mathbf{w}[k+1] = \mathbf{w}[k] - \mu \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}$$

The output of the equalizer:

$$u[k] = [w_0, w_1, ..., w_M][y[k], y[k-1], ..., y[M-1]]^T = \mathbf{w}^T \mathbf{y}$$

Mean square error:

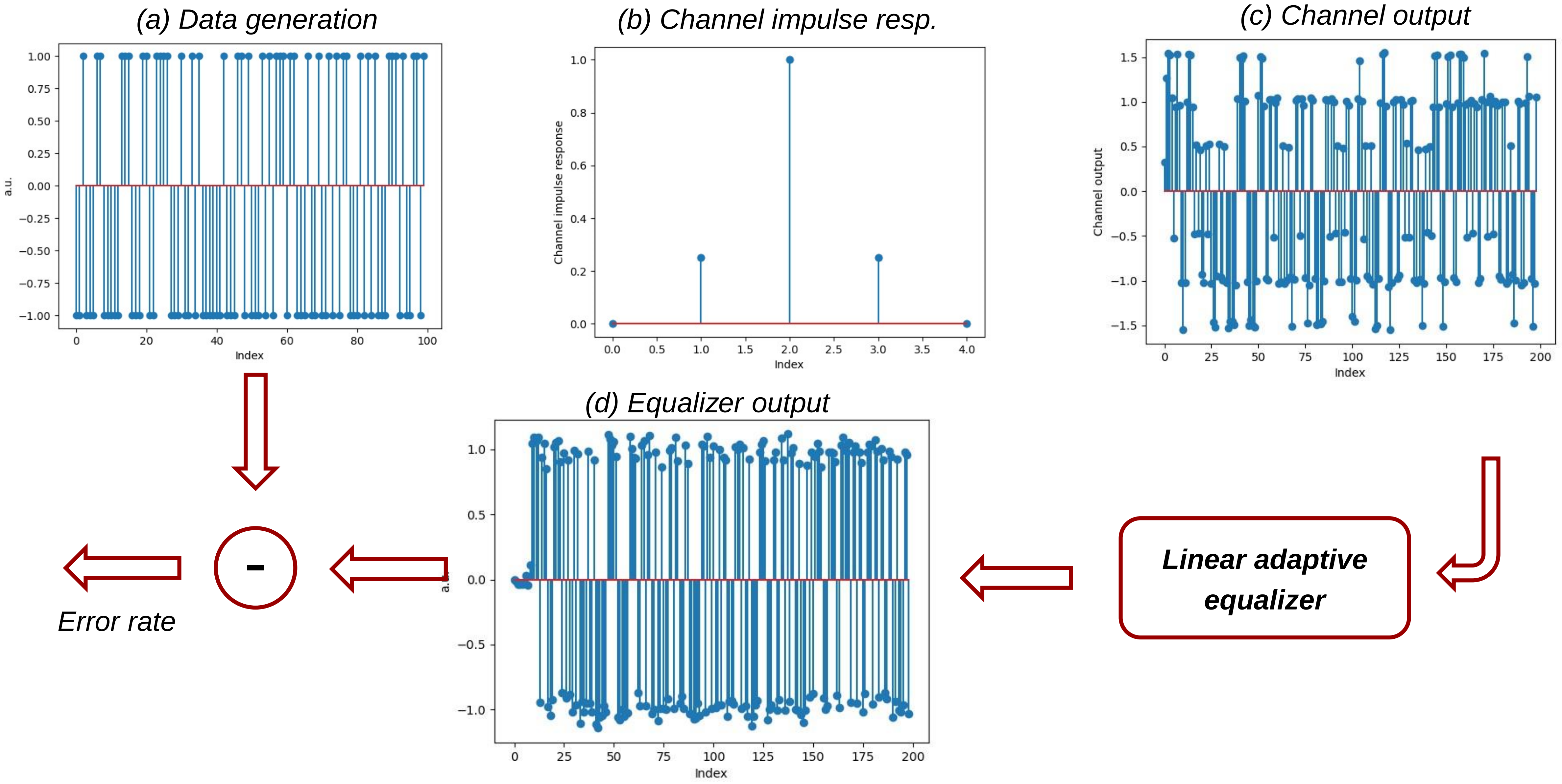
$$E[k] = -\tfrac{1}{2}\left(x[k] - \mathbf{w}^T \mathbf{y}\right)^2$$

Computing the derivative:

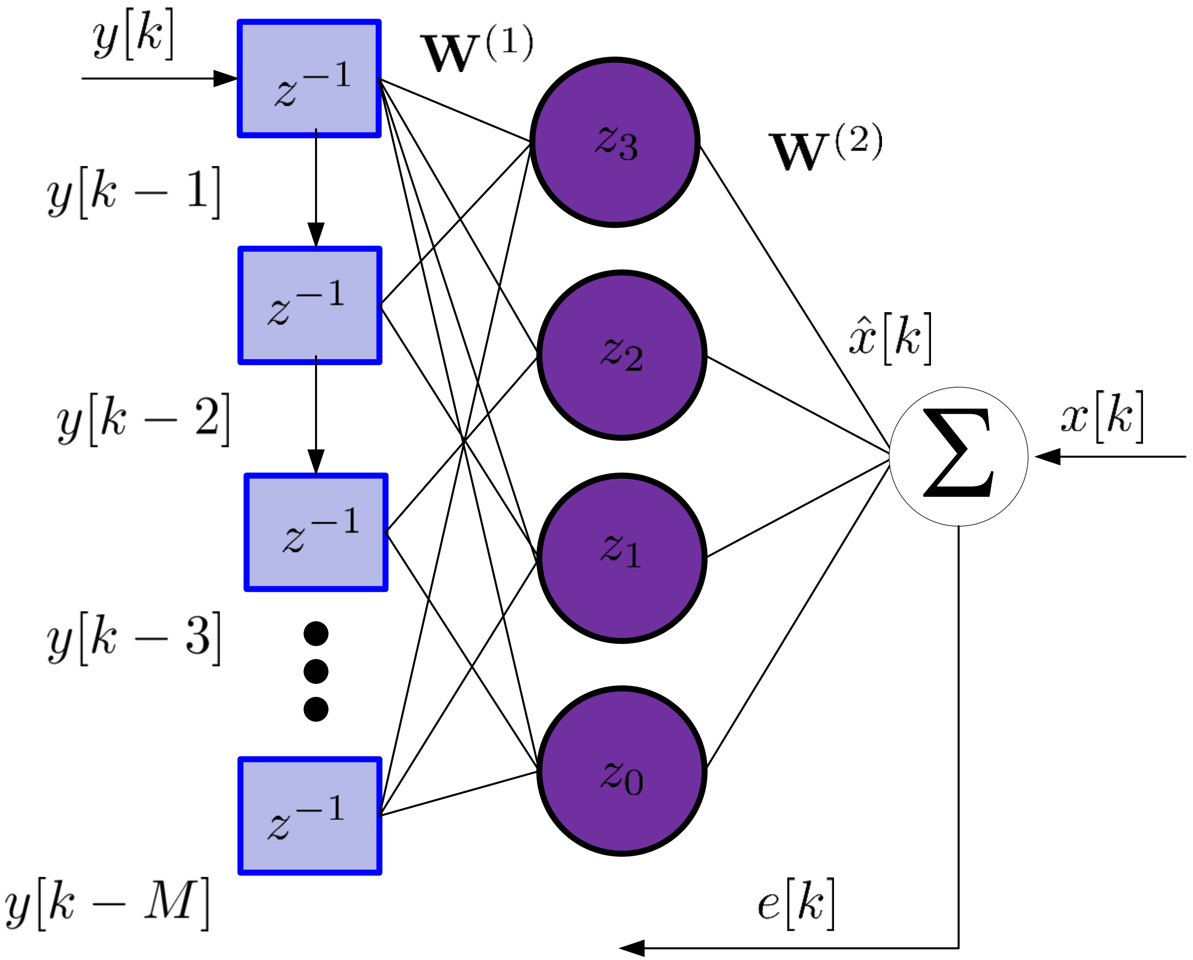
$$\frac{\partial E}{\partial \mathbf{w}} = -(x[k] - \mathbf{w}^T)\mathbf{y}$$

The update rule becomes:

$$\boxed{\mathbf{w}[k+1] = \mathbf{w}[k] + \mu \mathbf{y}(x[k] - \mathbf{w}^T \mathbf{y}) = \mathbf{w}[k] + \mu \mathbf{y} e[k]}$$

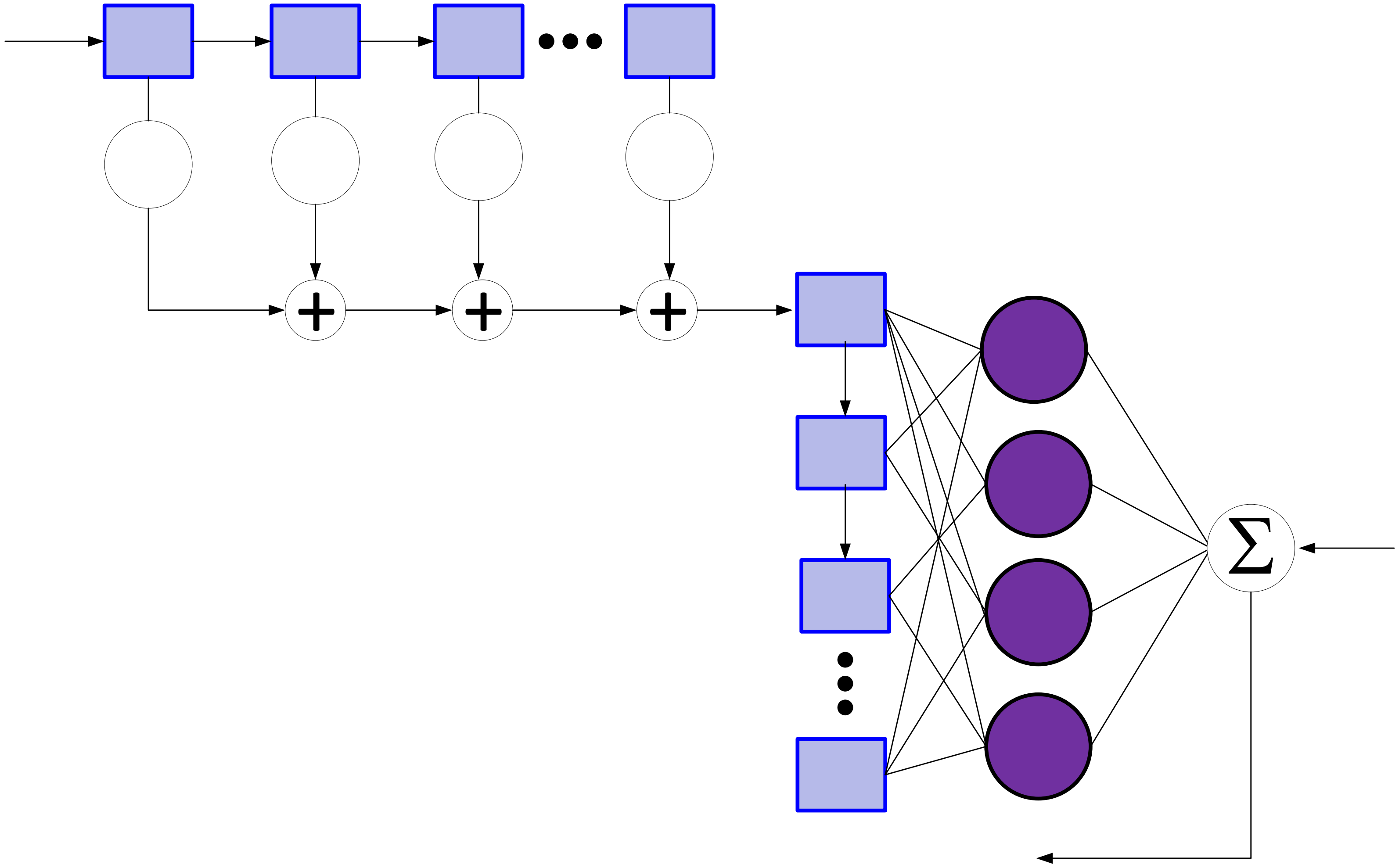

(a) Data generation
a.u.
Index
(b) Channel impulse resp.
Channel impulse response
Index
(c) Channel output
Channel output
Index
(d) Equalizer output
a.u.
Index
Linear adaptive equalizer
-
Error rate

# Nonlinear adaptive equalizer (time-delay NN)




$$\boxed{\hat{x} = \mathbf{W}^{(2)} f\left(\mathbf{W}^{(1)}\mathbf{y}\right)}$$

# Combining linear and nonlinear equalizer (1D CNN)



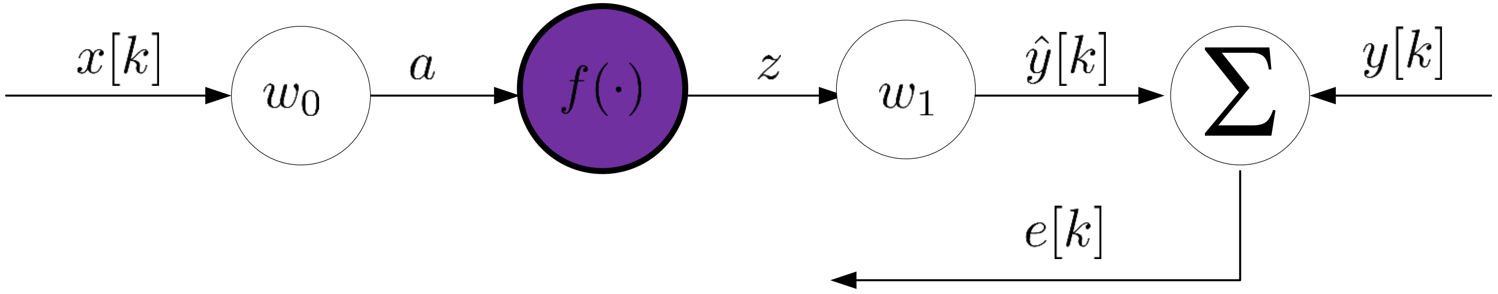
# All we need is the chain rule

$$x \longrightarrow \boxed{h(\cdot)} \xrightarrow{w_1} \boxed{g(\cdot)} \xrightarrow{w_2} \boxed{f(\cdot)} \longrightarrow y = f(g(h(x)))$$

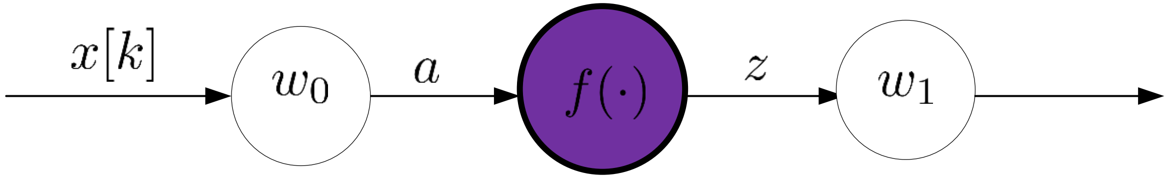
$$y = f(w_2)$$
$$w_2 = g(w_1)$$
$$w_1 = h(x)$$

$$\boxed{\frac{dy}{dx} = \frac{dy}{dw_2}\frac{dw_2}{dw_1}\frac{dw_1}{dx}}$$

# Very simple example




$$\hat{y}[k] = w_1 \overbrace{f(w_0 x[k])}^{z}$$

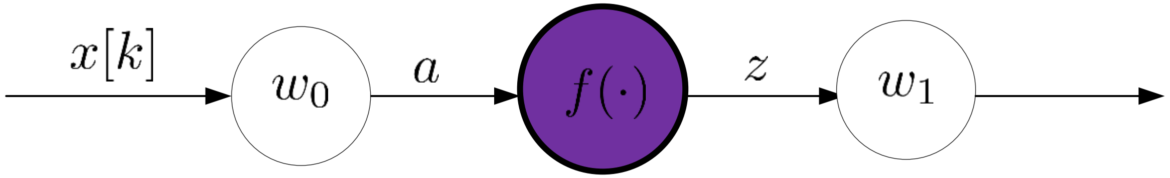



$$e[k] = \frac{1}{2}(\hat{y}[k] - y[k])^2 = \frac{1}{2}(w^1 z - y[k])^2$$

$$w_0[k+1] = w_0[k] - \mu \frac{de}{dw_0}$$

$$w_1[k+1] = w_1[k] - \mu \frac{de}{dw_1}$$

# Update rules




$$e[k] = \frac{1}{2}(\hat{y}[k] - y[k])^2 = \frac{1}{2}(w_1 z - y[k])^2$$

$$a = w_0 x[k]$$

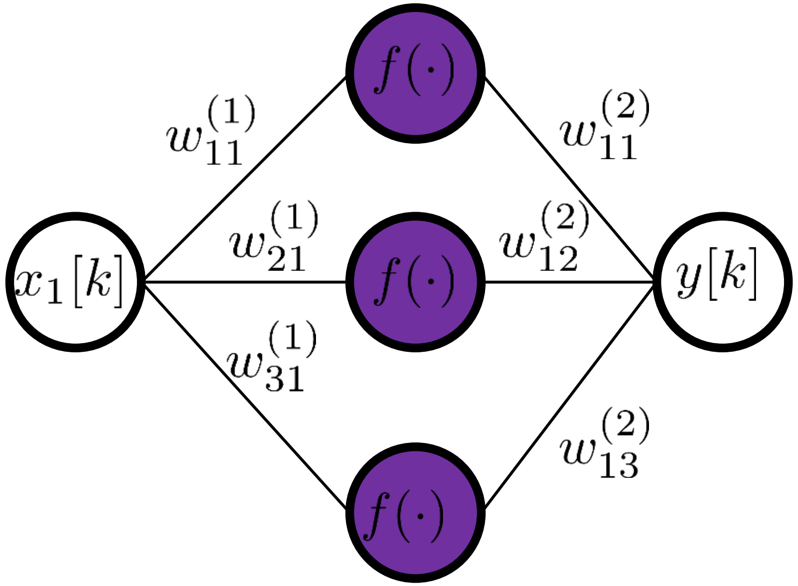
$$z = f(a)$$

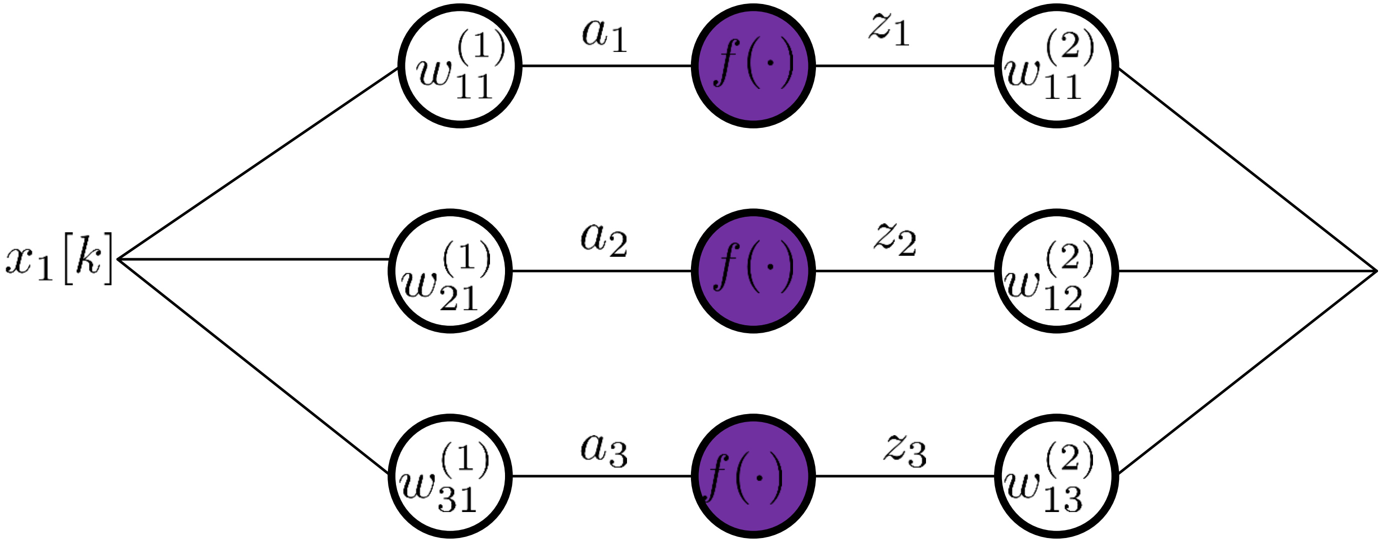
$$e[k] = \frac{1}{2}(w^1 z - y[k])^2$$

$$\frac{de}{dw_1} = (w_1 z - y[k])z$$

$$\frac{de}{dw_0} = \frac{de}{dz}\frac{dz}{da}\frac{da}{dw_0} = (w_1 z - y[k])w_1 f'(a)x[k]$$

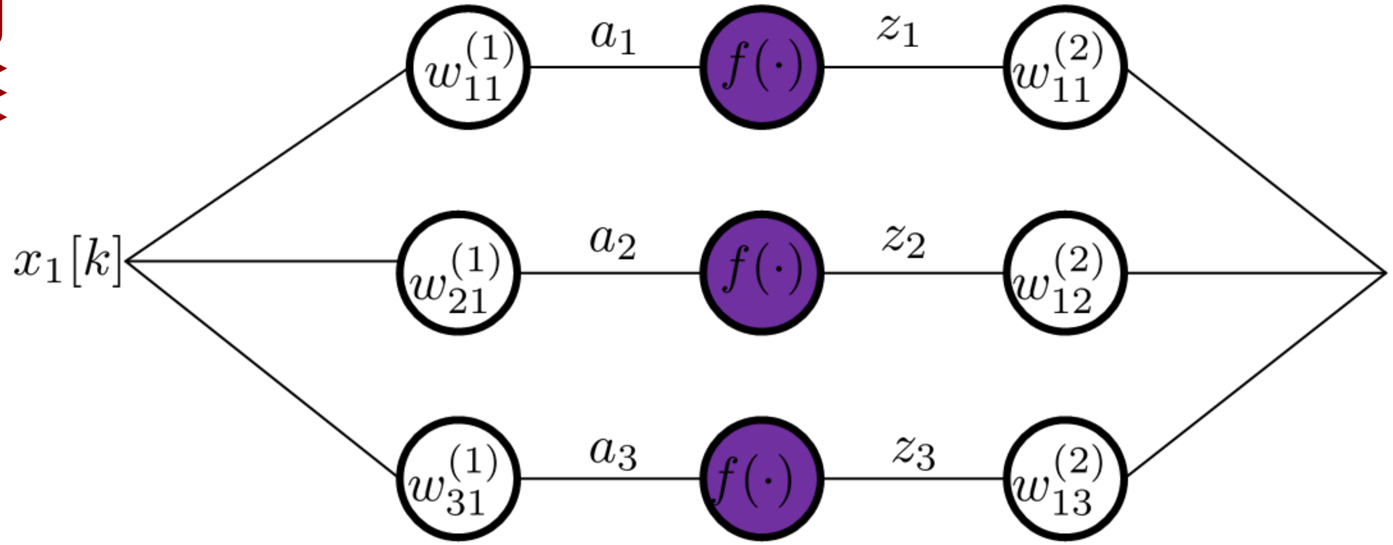
# One hidden layer neural network




$$e[k] = \frac{1}{2}(w_{11}^{2} z_1 + w_{12}^{2} z_2 + w_{13}^{2} z_3 - y[k])^2$$

$$e[k] = \frac{1}{2}\left(\sum_{j=0}^{3} w_{1j}^{(2)} z_j - y[k]\right)^2$$




$$e[k] = \frac{1}{2}(w_{11}^2 z_1 + w_{12}^2 z_2 + w_{13}^2 z_3 - y[k])^2$$

$$e[k] = \frac{1}{2}\left(\sum_{j=0}^{3} w_{1j}^{(2)} z_j - y[k]\right)^2$$

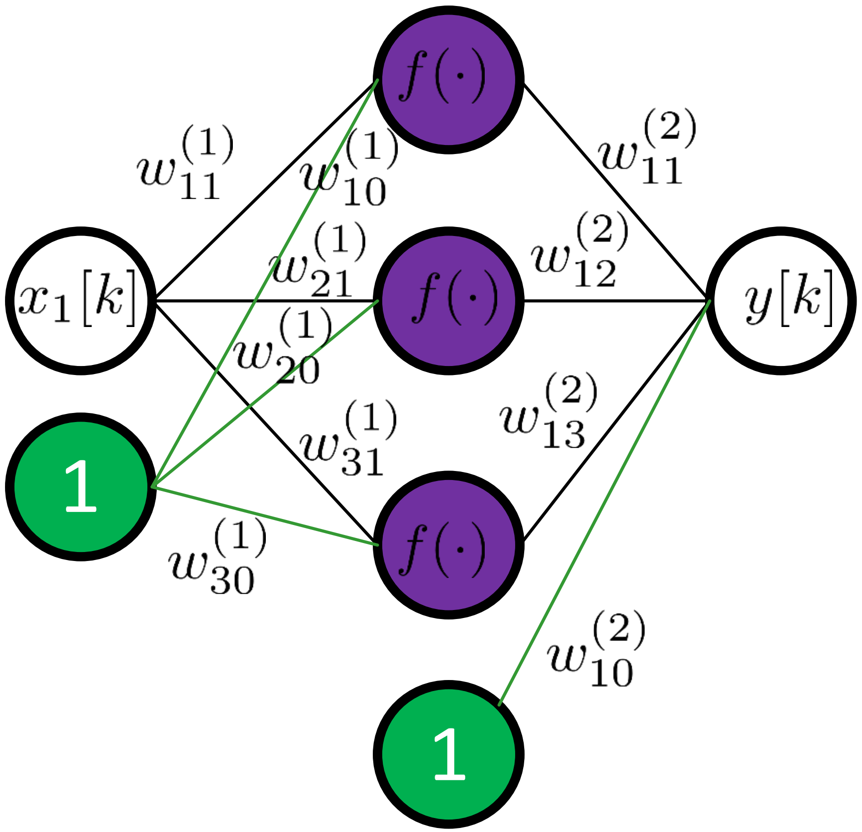
$$\boxed{w_{j1}^{(1)}[k+1] = w_{j1}^{(1)}[k] - \mu \frac{de}{dw_{j1}^{(1)}}}$$

$$\boxed{w_{1j}^{(2)}[k+1] = w_{1j}^{(2)}[k] - \mu \frac{de}{dw_{1j}^{(2)}}}$$

$$\frac{de}{dw_{1j}^{(2)}} = \left(\sum_{j=0}^{3} w_{1j}^{(2)} z_j - y[k]\right) z_j$$

$$\frac{de}{dw_{j1}^{(1)}} = \frac{de}{dz_j}\frac{dz_j}{da_j}\frac{da_j}{dw_{j1}^{(1)}} = \left(\sum_{j=0}^{3} w_{1j}^{(2)} z_j - y[k]\right) w_{1j}^{(2)} f'(a_j) x_1[k]$$

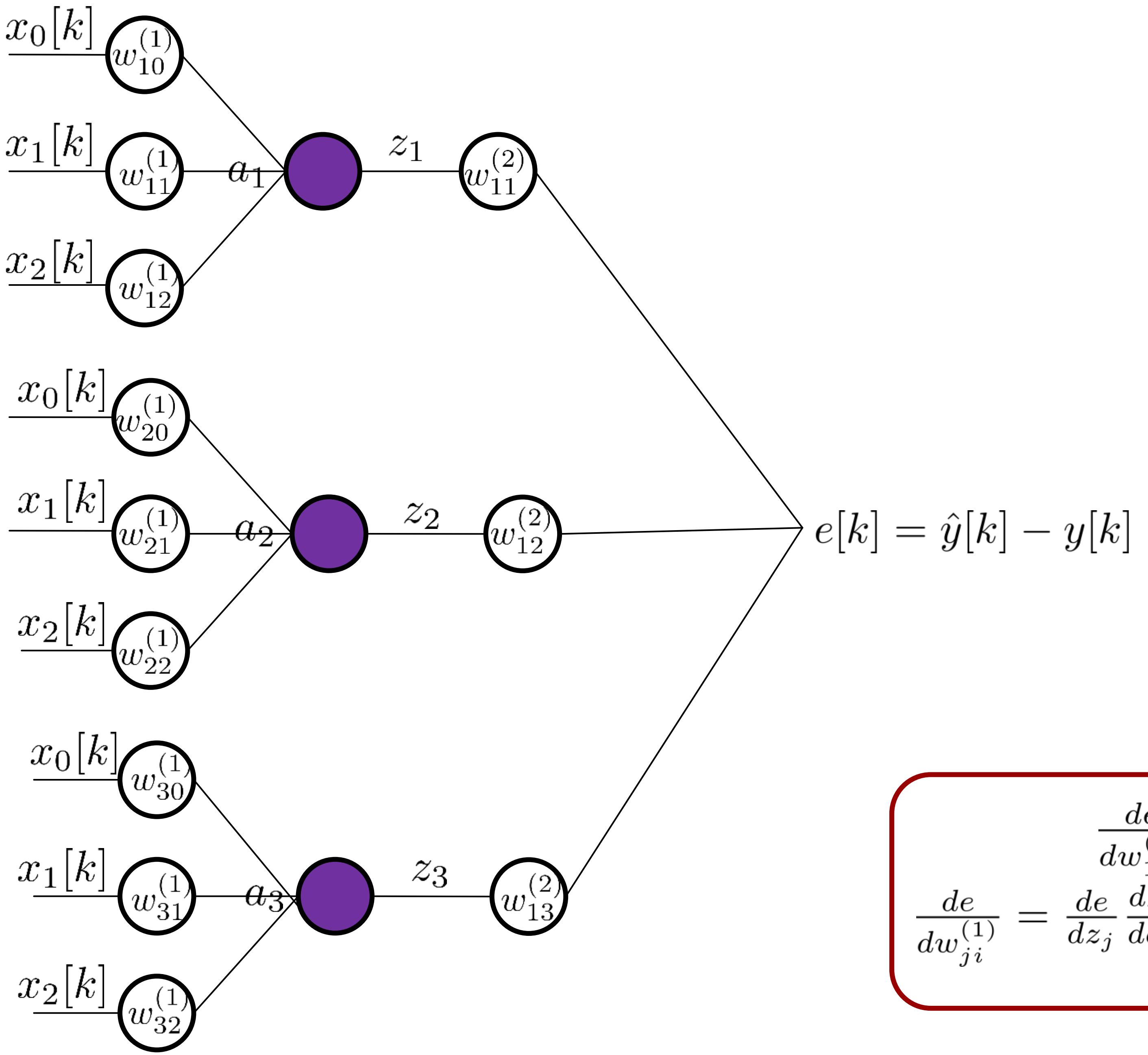
# One hidden layer neural network with biases




```
for i = 1 : L_iter
   for k = 1 : L_train
        compute e(k)
        grad = de(k)dw;
        W = W + eta*grad;
    end
    e(i) = mean(e.^2);
end
```

Can be done in one step

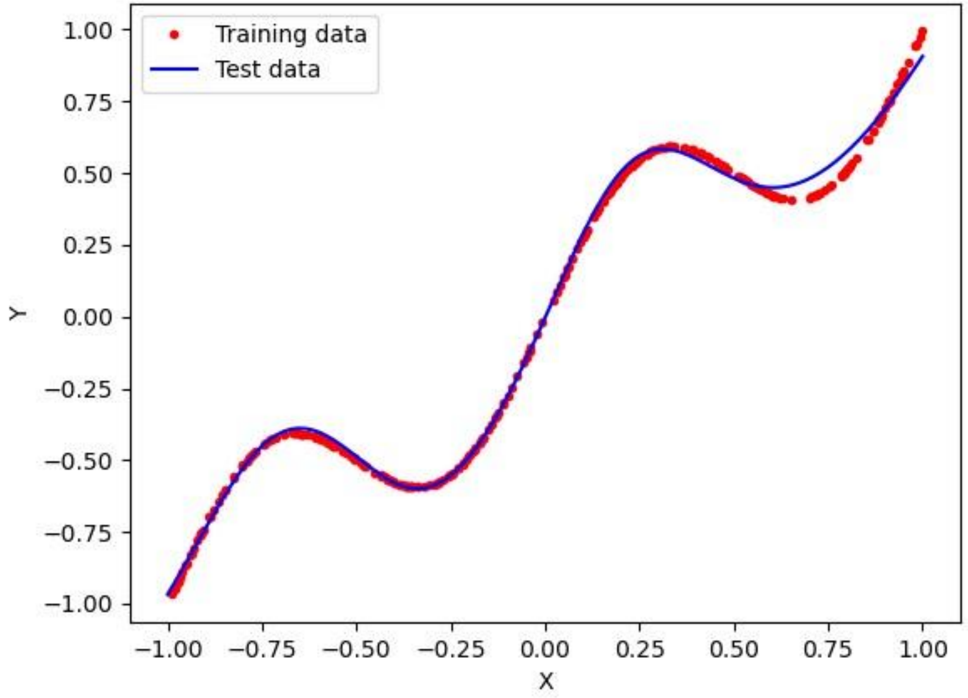
Pseudo code for training of neural networks

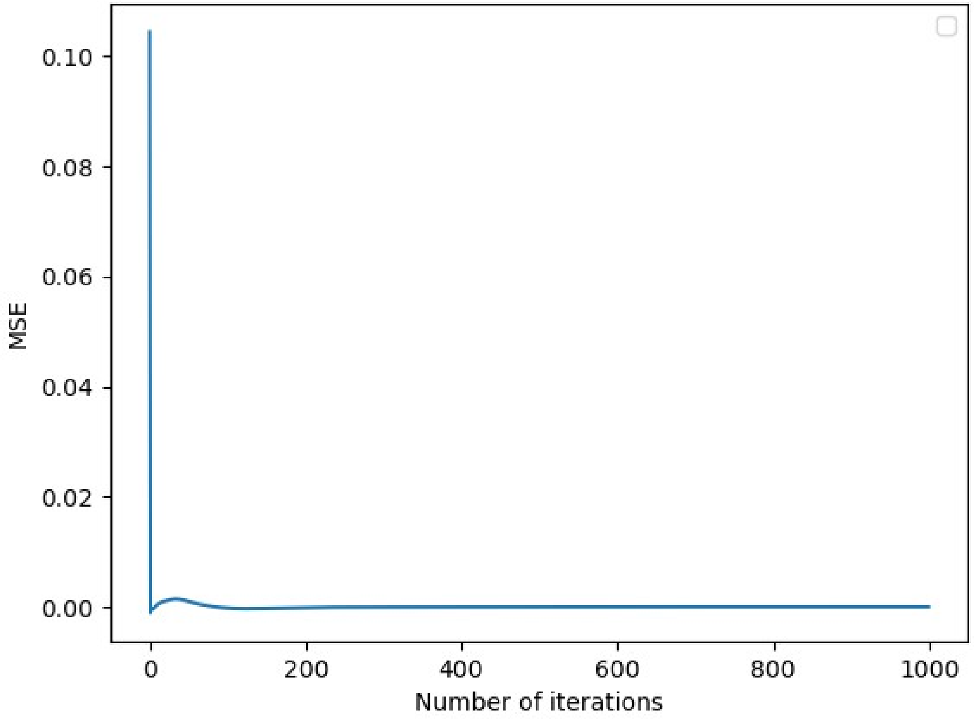



$$a_j = \sum_{i=0}^{D} w_{ji}^{(1)} x_i[k]$$
$$z_j = f(a_j)$$
$$\hat{y}[k] = \sum_{j=1}^{M} w_{1j}^{(2)} z_j$$

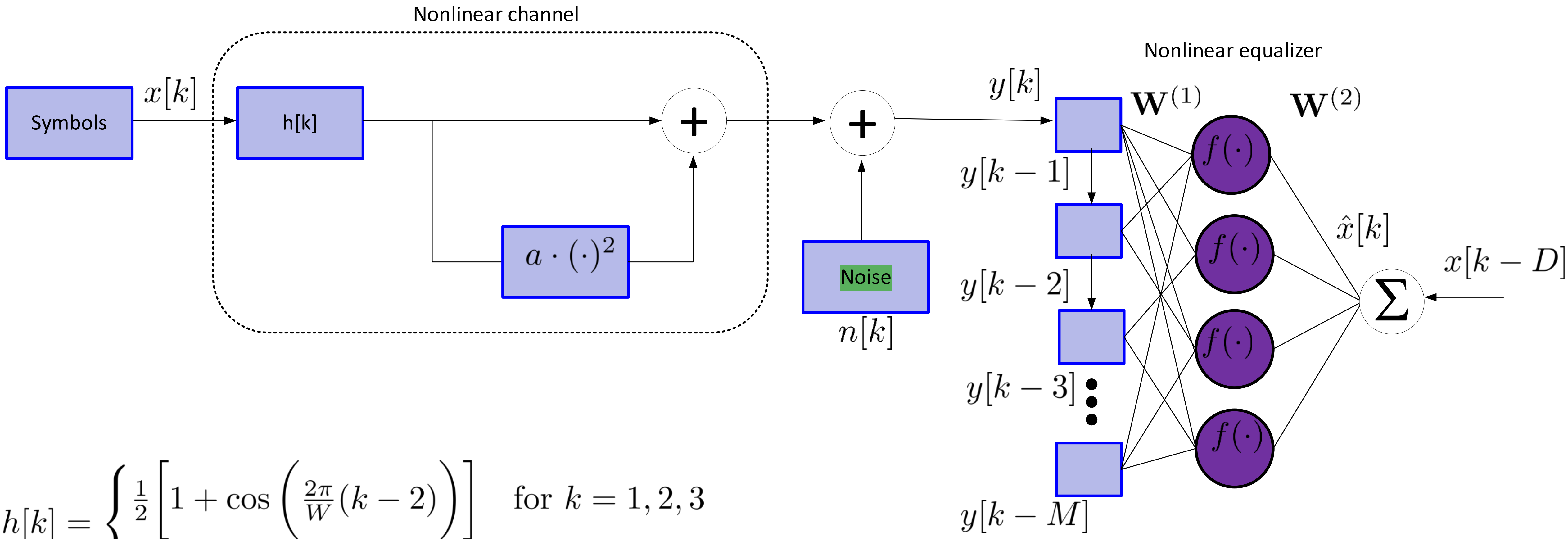
$$\frac{de}{dw_{1j}^{(2)}} = (\hat{y}[k] - y[k]) z_j$$
$$\frac{de}{dw_{ji}^{(1)}} = \frac{de}{dz_j} \frac{dz_j}{da_j} \frac{da_j}{dw_{ji}^{(1)}} = (\hat{y}[k] - y[k]) w_{1j}^{(2)} f'(a_j) x_i[k]$$

# Learning functions with 1D NN



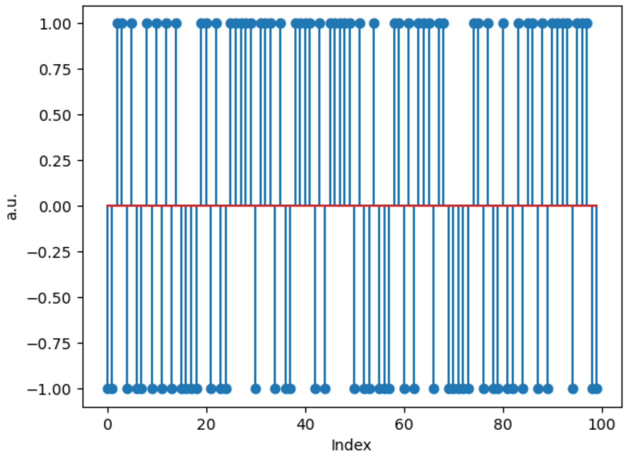
# Nonlinear discrete-time communication channel with nonlinear equalization




$$h[k] = \begin{cases} \frac{1}{2}\left[1+\cos\left(\frac{2\pi}{W}(k-2)\right)\right] & \text{for } k=1,2,3 \\ 0 & \text{otherwise} \end{cases}$$

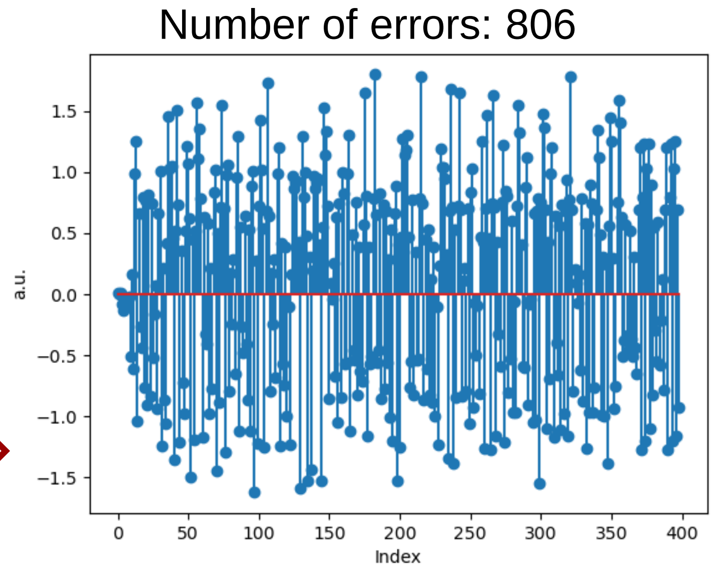
$$\boxed{y[k] = h[k] * x[k] + a \cdot (h[k] * x[k])^2 + n[k]}$$

# Performance of nonlinear vs. linear equalizer

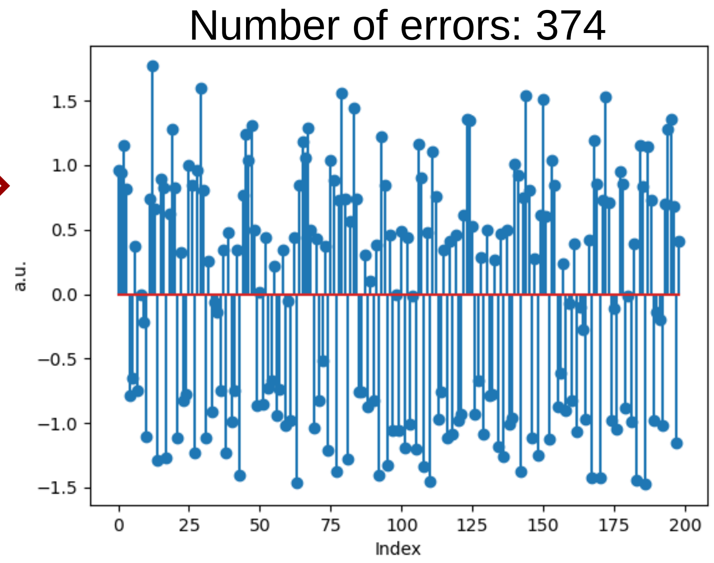



*Nonlinear channel*

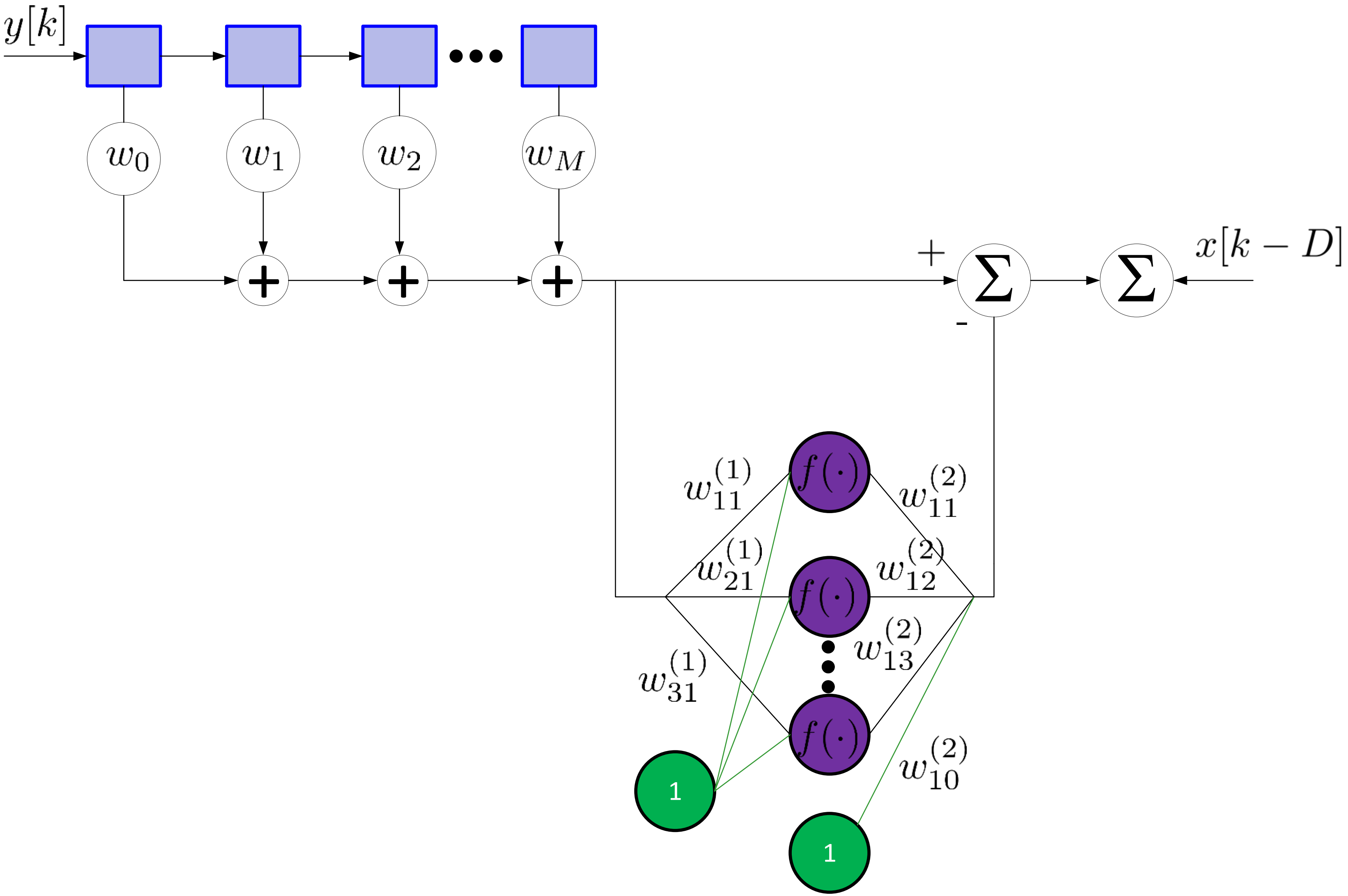
*Linear adaptive equalizer*




*Nonlinear adaptive equalizer*



# Convolutional nonlinear equalizer



# In this lecture we have learned….

- Basics of digital communication
- How machine learning can be used to optimize the performance
- How to use the chain rule to derive learning algorithms
- How to derive learning algorithms for a single layer neural network
- How to derive learning algorithms for linear and nonlinear equalization
- The impact of linear and nonlinear equalization