Introduction to Neural Networks

Simulation Project
End-to-end Deep Learning of a Communication System

MOCKS 913

Mansoor I. Yousefi

Communications and Electronics Department
Télécom Paris, LTCI, Palaiseau, France

Paris, France

December 9, 2020

INSTITUT POLYTECHNIQUE DE PARIS

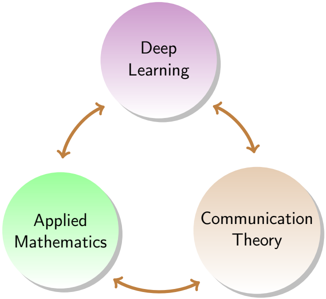
TELECOM Paris

The Theme of This Lecture

End-to-end **deep learning** of a complex communication system (TX and RX)

Linear channel	Nonlinear channel
$Y = X + N$	$Y = f(X, N)$
Easy?!	**Hard!**

Inter-disciplinary research project



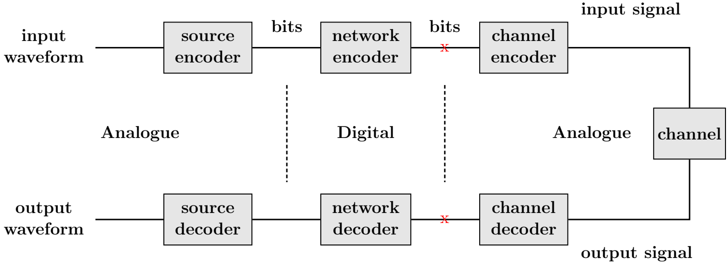
Outline

We describe the **generative neural network** (true model), and the steps to simulate that nnet to generate data sets:

1. Simulating a simple point-to-point communication system
2. Channel model
3. Generative neural network

Table of Contents

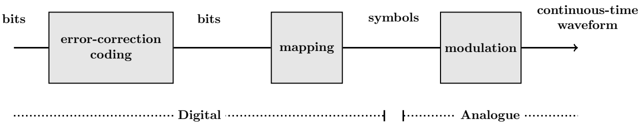
Block diagram of a communication system



Digital interfaces and layering: the *separation theorem* for point-to-point data communication implies that the source and channel can be designed separately.

We thus ignore the source, and only *learn the channel part* end to end.

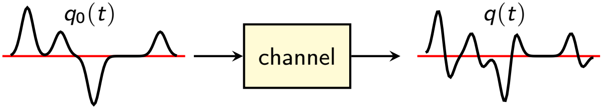
Channel encoder



- channel coding: protect against channel noise
- mapping: bits to symbols
- channel encoder: symbols to signal (modulator)

We also ignore error-correction coding (could be learned).

Channel model



- *A simple linear channel*

$$Y(t) = h(t) * X(t) + N(t)$$

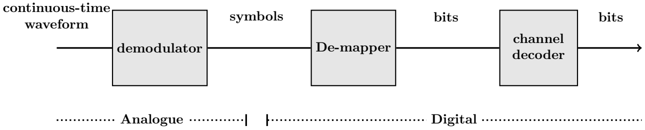
- *A complex channel described by a nonlinear PDE*. Here the signal evolves according to an evolution equation in 1+1 dimensions (time t, distance z)

$$\frac{\partial q}{\partial z} = K(q(t,z)) + n(t;z)$$

Examples: ($q_t := \partial_t q$)

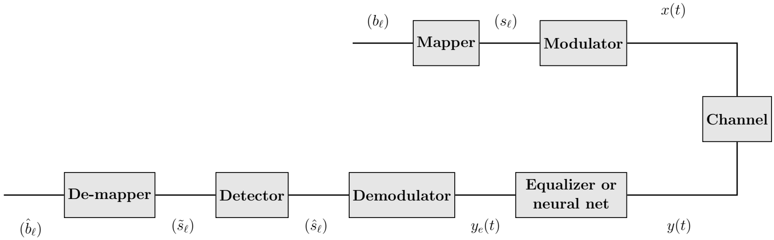
- $K(q) = |q|^2$ (memoryless nonlinearity)
- $K(q) = -j(q_{tt} + 2|q|^2 q)$ (nonlinearity with memory)

Channel decoder



- demodulator: signal to symbols mapping
- mapping: bits to symbols
- channel decoder: decoding for the channel encoder

Eliminating blocks that we ignored, in this project we consider the following block diagram.



In what follows, we describe each building block.

Linear Modulation

Let $\mathcal{H}$ be a *Hilbert space* of signals at the input of the channel with an inner product denoted by $\langle ., . \rangle$.
Let $\{\phi_\ell(t)\}_{-\infty}^{\infty}$ be an *orthonormal basis* for $\mathcal{H}$, *i.e.*, satisfying:

$$\langle \phi_i(t), \phi_j(t) \rangle = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Linear modulation consists of expanding a signal $x(t) \in \mathcal{H}$ onto a basis:

$$\boxed{x(t) = \sum_{\ell=-\infty}^{\infty} s_\ell \phi_\ell(t),}$$

where $\{s_\ell\}$ are symbols chosen from a constellation S.

Example. Consider the space of absolutely integrable signals $\mathcal{H} = L^1(\mathbb{R})$. *Fourier transform* provides a choice of basis for $\mathcal{H}$:

$$\left\{e^{j\omega t}\right\}_{\omega\in\mathbb{R}}.$$

If the signals are T-periodic, a basis is given by the Fourier series:

$$\left\{e^{jl\omega_0 t}\right\}_{l=-\infty}^{\infty}, \quad \omega_0 = \frac{2\pi}{T}.$$

□

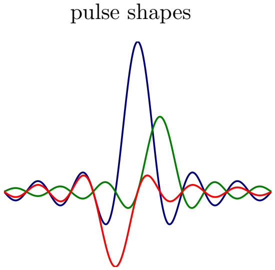
Example. Consider the space of finite-energy signals bandlimited to B Hz, denoted by $\mathcal{H} = L^2_B(\mathbb{R})$. The *Nyquist-Shannon sampling theorem* provides a choice of the basis:

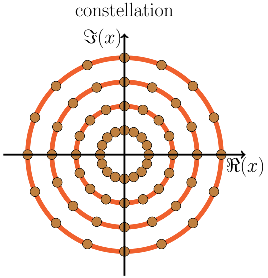
$$\left\{\sqrt{B}\,\mathrm{sinc}(Bt-\ell)\right\}_{\ell=-\infty}^{\infty}.$$

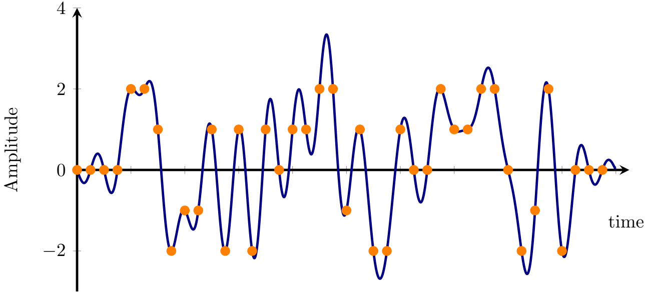
where $\mathrm{sinc}(x) = \sin(\pi x)/(\pi x)$.

□

An example of a linearly-modulated signal using a sinc pulse shape and a multi-ring constellation is shown here.







Considering the sinc basis, the transmitted signal is:

$$\boxed{x(t) = \sqrt{B} \sum_{\ell=-\infty}^{\infty} s_\ell \, \mathrm{sinc}(Bt - \ell)}\,,$$

where $s_\ell \in \mathbb{C}$ are symbols.

Channel model

Consider first a simple *additive white Gaussian noise (AWGN)* channel,

$$\boxed{y(t) = h(t) * x(t) + n(t).}$$

Here, $x(t)$ is input, $y(t)$ is output, $h(t)$ is channel impulse response, $n(t)$ is white circular symmetric Gaussian noise, *i.e.*, satisfying

$$\mathrm{E}\Big\{n(t)n^*(t')\Big\} = \sigma_0^2\delta(t-t'),$$

where σ_0^2 is noise power spectral density (PSD). Finally, $h(t) * x(t)$ denotes convolution. All signals are considered as functions from $\mathbb{R}$ to $\mathbb{C}$.

The channel filter $h(t)$ introduces a **distortion** that must be cancelled via **equalization**.

Note. Convolution of two functions $f(t)$ and $g(t)$ is defined as

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(t-\tau)g(\tau)\mathrm{d}\tau.$$

Equalization

In the frequency domain, the AWGN channel is:

$$\boxed{Y(\omega) = H(\omega) \times X(\omega) + N(\omega).}$$

Equalization is performed in the frequency domain to invert the channel

$$\begin{aligned} Y_e(\omega) &= H^{-1}(\omega) Y(\omega) \\ &= X(\omega) + Z(\omega), \end{aligned} \tag{1}$$

where $Z(\omega) = H^{-1}(\omega) N(\omega)$.
From $Y_e(\omega)$, we obtain the signal in time domain after equalization

$$y_e(t) = \mathcal{F}^{-1}(Y_e(\omega)),$$

where $\mathcal{F}$ denotes Fourier transform operator.

De-modulation

The received symbols at the output are obtained by projection onto the basis

$$\begin{aligned}\hat{s}_\ell &= \langle y_e(t), \sqrt{B}\,\mathrm{sinc}(Bt-\ell)\rangle \\ &= \sqrt{B}\int_{-\infty}^{\infty} y_e(t)\,\mathrm{sinc}(Bt-\ell)\mathrm{d}t.\end{aligned}$$

The project step is called match filtering.

Note. The inner product in $L^2(\mathbb{R})$ is

$$\langle f(t), g(t)\rangle = \int_{-\infty}^{\infty} f(t)g^*(t)\mathrm{d}t.$$

Detection

The demodulated symbols $\hat{s}_\ell$ may not be in the constellation $\mathcal{S}$. Given $\hat{s}_\ell$, we shall map $\hat{s}_\ell$ to a symbol $\tilde{s}_\ell \in \mathcal{S}$.

The optimal receiver consists of the *maximum likelihood detector* applied to the conditional probability distribution $p(\hat{s}_\ell | s_\ell)$, *i.e.*,

$$\tilde{s}_\ell = \mathrm{argmax}_{s_\ell \in S} \quad p(\hat{s}_\ell | s_\ell).$$

For an AWGN channel, the continuous-time model (1) is discretized to the discrete-time model

$$\hat{s}_\ell = s_\ell + z_\ell, \quad z_\ell \sim N(0, \sigma^2), \quad \ell = 1, 2, \cdots,$$

where $\sigma^2 = \sigma_0^2 B$ is the noise power.

As a result

$$p(\hat{s}_\ell | s_\ell) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{|\hat{s}_\ell - s_\ell|^2}{2\sigma^2}}$$

Maximizing $p(\hat{s}_\ell | s_\ell)$ amounts to minimizing the exponent. The maximum likelihood detector is simplified to the **minimum distance decoder**.

Given $\hat{s}_\ell$:

$$\tilde{s}_\ell = \mathrm{argmin}_{s_\ell \in S} \left| s_\ell - \hat{s}_\ell \right|^2.$$

A more complex channel

The design of the transmitter TX and receiver RX discussed above relied on the **linearity property** of the channel. For cases where the model is known and is tractable, the use of deep learning is debatable. In fact, good receivers are known for the AWGN channel.

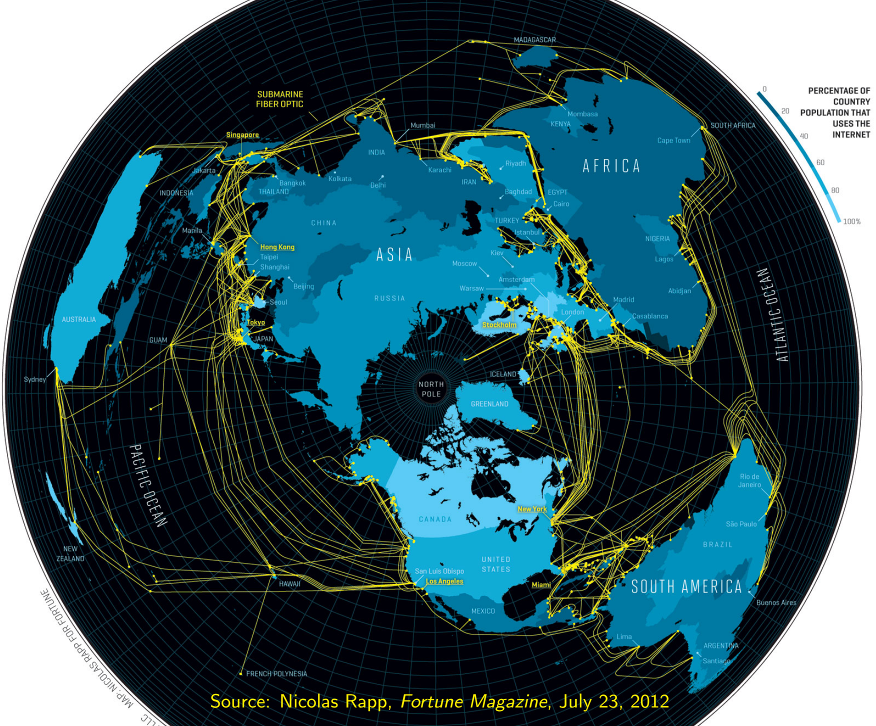
We next consider a complex nonlinear channel where the optimal receiver is unknown. This channel is optical fiber, and takes the form:

$$\begin{aligned} Y &= f(X, N) \\ &\neq X + N. \end{aligned}$$

Project: End-to-end deep learning of the TX and RX for the nonlinear optical fiber using an input output data set.

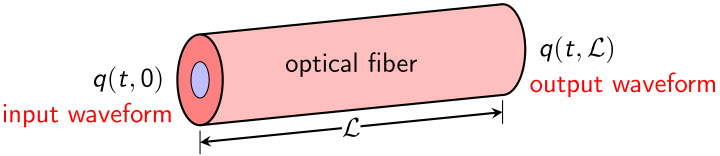
End-to-end means from bits to bits; see the diagram.

Table of Contents



Source: Nicolas Rapp, *Fortune Magazine*, July 23, 2012

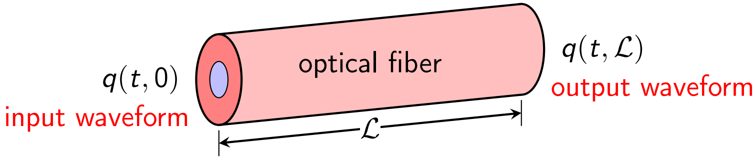
Optical Fiber



Desirable properties of **optical fiber**:

- Low loss $\sim$ 0.2 dB/km
- Large bandwidth $\sim$10's THz
- Low noise and interference

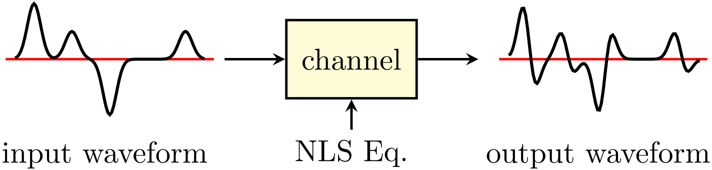
Stochastic Nonlinear Schrödinger Equation



Pulse propagation in optical fiber is modeled by the *stochastic nonlinear Schrödinger (NLS) equation*:

$$\frac{\partial q(t,z)}{\partial z} = -\underbrace{\frac{j\beta_2}{2}\frac{\partial^2 q(t,z)}{\partial t^2}}_{\textbf{dispersion}} + \underbrace{j\gamma|q(t,z)|^2 q(t,z)}_{\textbf{nonlinearity}} + \underbrace{n(t,z)}_{\textbf{noise}}$$

- *z is distance* along the fiber and *t is time*
- $q(t,z)$ is the complex envelope of the propagating signal
- β_2 is the second-order *chromatic dispersion coefficient*
- γ is the *nonlinearity parameter*
- $n(t,z)$ is bandlimited *white circular symmetric Gaussian noise*



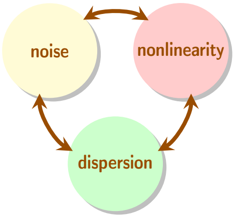
The input output of the channel are

$$X(t) = q(t, 0), \quad Y(t) = q(t, \mathcal{L}),$$

where $\mathcal{L}$ is fiber length.

The model may be generalized to include additional effects, such as residual loss, higher-order dispersion, multiple input multiple output transmission, etc.

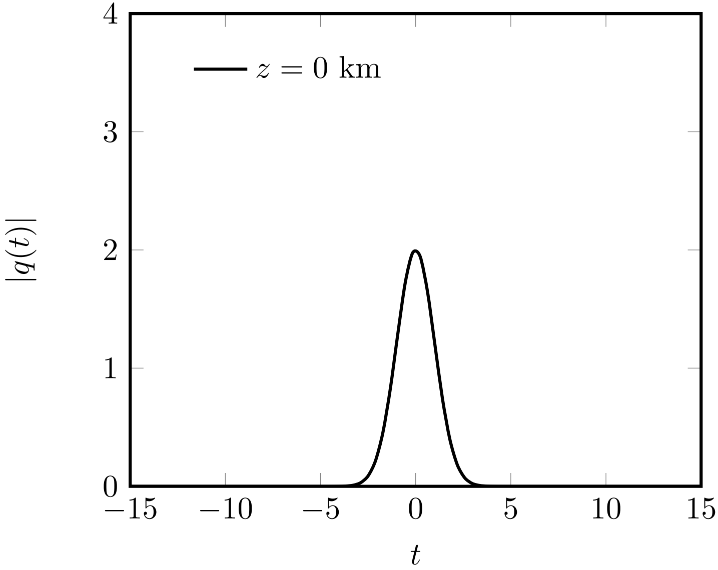
Physical Effects in the Channel



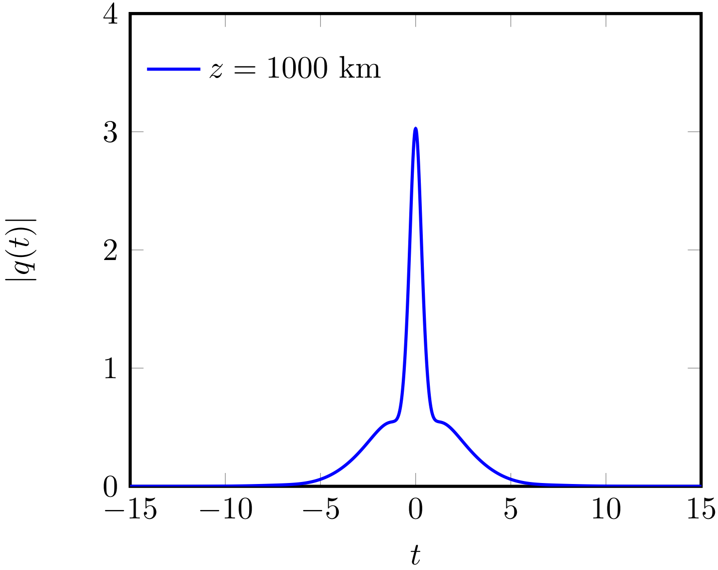
The model is governed by a *nonlinear dispersive partial differential equation (PDE)*, modeling the physical effects of **dispersion**, **nonlinearity** and **noise**.

This PDE cannot be solved analytically.

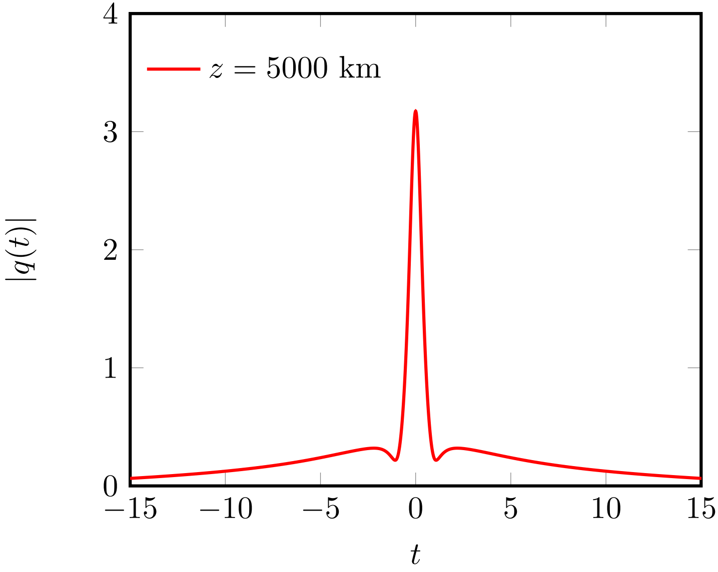
Example of Signal Propagation – a Gaussian input



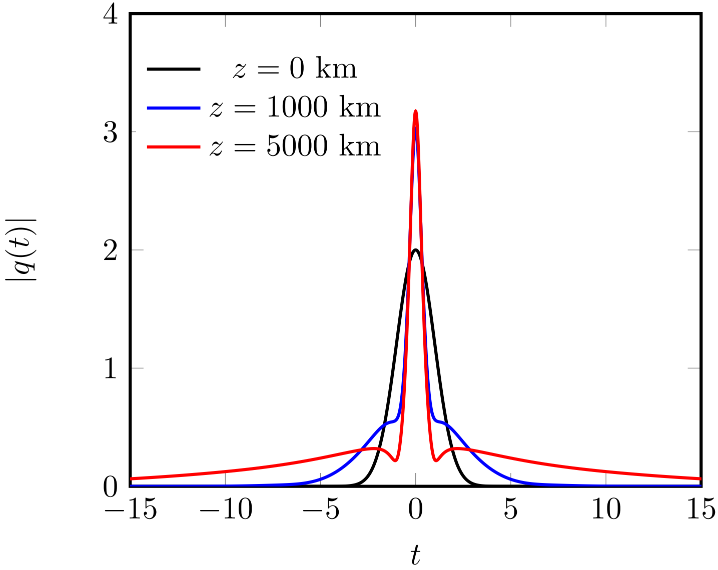
Example of Signal Propagation – a Gaussian input



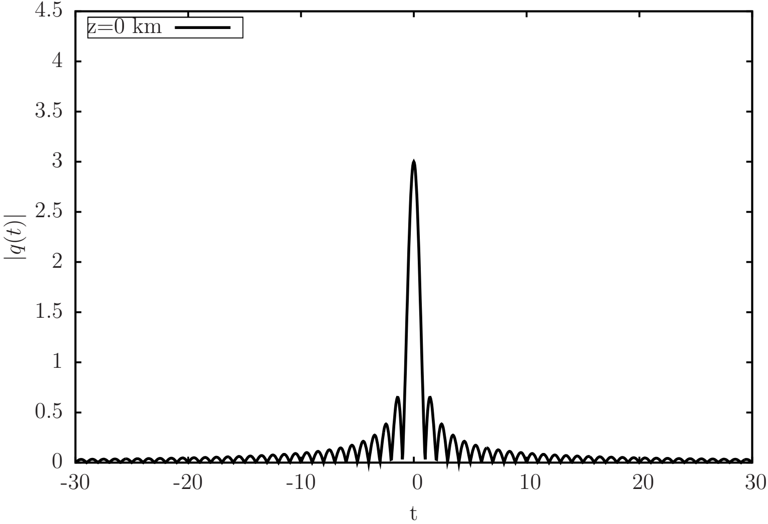
Example of Signal Propagation – a Gaussian input



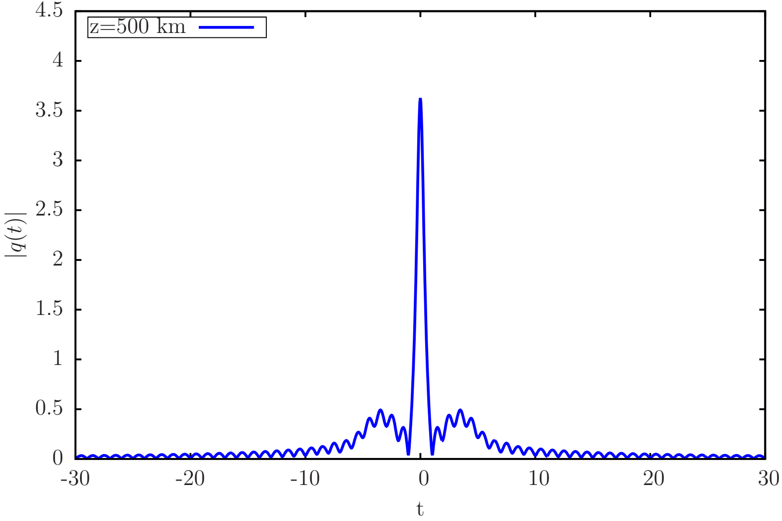
Example of Signal Propagation – a Gaussian input



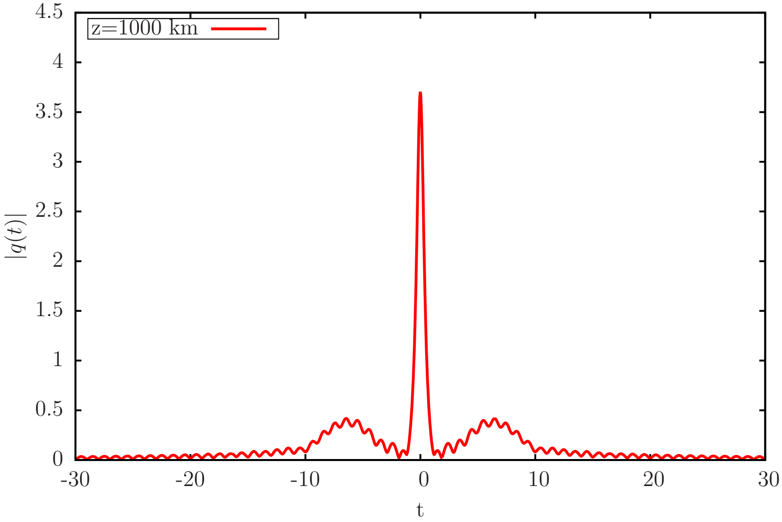
Example of Signal Propagation – a sinc input



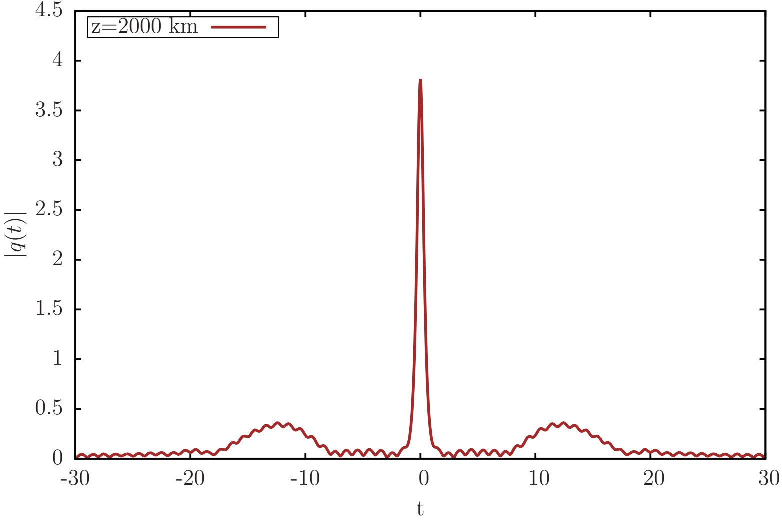
Example of Signal Propagation – a sinc input



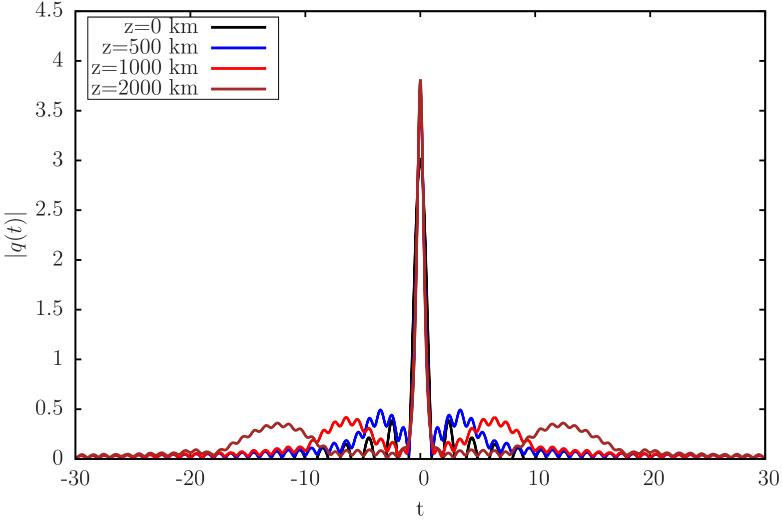
Example of Signal Propagation – a sinc input



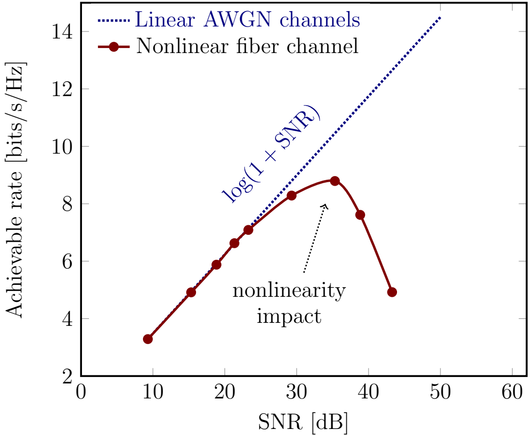
Example of Signal Propagation – a sinc input



Example of Signal Propagation – a sinc input



Achievable Rates of Existing Approaches



Capacity problem in fiber

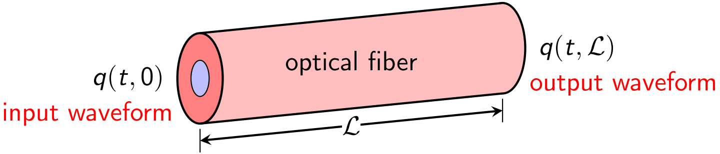
Fiber nonlinearity places an upper limit on capacity.

Can neural nets learn the optimal TX and RX, help solve this problem?

We now describe the **generative neural network**. This model is called *split-step Fourier method* in communications.

We begin by discretizing the nonlinear PDE.

Table of Contents



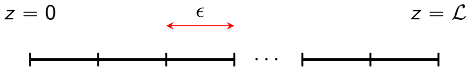
Consider a general model:

$$\frac{\partial q(t,z)}{\partial z} = \underbrace{L_L(q)}_{\textbf{linear}} + \underbrace{L_N(q)}_{\textbf{nonlinear}} + \underbrace{n(t,z)}_{\textbf{noise}}$$

where L_L and L_N are linear and nonlinear operators.

Example. In the stochastic nonlinear Schrödinger equation:

$$L_L(q) = -\frac{j\beta_2}{2}\frac{\partial^2 q}{\partial t^2},$$
$$L_N(q) = j\gamma|q|^2 q.$$



Discretize the distance into a large number $M \to \infty$ of segments of length $\epsilon = \mathcal{L}/M$.
Discretize the time interval $[-T/2, T/2]$ into a large number $n \to \infty$ of intervals of step size $\mu = T/n$.

Break down the PDE into 3 parts. Perform successive linear, nonlinear and noise transformations, each of which in one spatial segment:

- **Linear step**:

$$\partial_z q = L_L(q) \quad \Rightarrow \quad \mathrm{V} = W\mathrm{X}$$

  where $\mathrm{X} \in \mathbb{C}^n$, $\mathrm{V} \in \mathbb{C}^n$ and $W \in \mathbb{C}^{n\times n}$ is a weight matrix.

- **Nonlinear step**:

$$\partial_z q = L_N(q) \quad \Rightarrow \quad \mathrm{U} = \sigma(\mathrm{V})$$

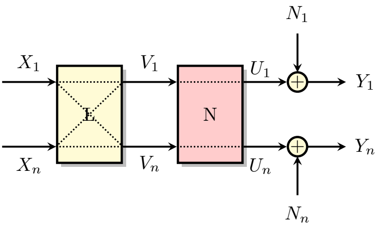
  where $\mathrm{U} \in \mathbb{C}^n$, and $\sigma(x) : \mathbb{C} \mapsto \mathbb{C}$ is the activation function acting per-component.

- **Noise addition:**

$$\mathrm{Y} = \mathrm{U} + \mathrm{N}, \qquad \mathrm{N} \sim N(0, \sigma^2 I_n)$$

Yellow and blue colors refer to continuous- and discrete-time models.

One layer of neural network



The diagram depicts one layer of the neural network, consisting of a linear, a nonlinear and a noise transformation. The nnet will consists of a cascade of 10–100 fully-connected layers.

Note. In standard neural networks, one performs successive linear and nonlinear transformations. There is no noise step. Here, we include a noise step, because this is a generative nnet. For the predictive nnet there is no noise step.

Linear Step

$$\frac{\partial q(t,z)}{\partial z} = -\underbrace{\frac{j\beta_2}{2}\frac{\partial^2 q(t,z)}{\partial t^2}}_{\textbf{dispersion}}$$

Define the Fourier transform with respect to t with convention:

$$\hat{q}(\omega,z) = \mathcal{F}(q)(\omega) = \int_{-\infty}^{\infty} q(t,z)e^{-j\omega t}\mathrm{d}t.$$

Taking Fourier transform of both sides:

$$\frac{\partial \hat{q}(\omega,z)}{\partial z} = \frac{j\omega^2\beta_2}{2}\hat{q}(\omega,z) \quad \Rightarrow \begin{cases} \hat{q}(\omega,z) = e^{j\frac{\beta_2 z}{2}\omega^2}\hat{q}(\omega,0) & (*) \\ q(t,z) = \frac{1}{\sqrt{j2\pi\beta_2 z}}e^{j\frac{1}{2\beta_2 z}t^2} * q(t,0) \end{cases}$$

where we used $e^{-\frac{t^2}{2\lambda}} \leftrightarrow \sqrt{2\pi\lambda}e^{-\frac{\lambda\omega^2}{2}}$.

Linear Step

Dispersion is a simple phase change (all-pass filter) in the frequency domain

Dispersion is a convolution in the time domain, giving rise to memory and pulse broadening

If the input output of the linear step in one layer are $\mathsf{X} \in \mathbb{C}^n$ and $\mathsf{V} \in \mathbb{C}^n$, discretizing (*)

$$\mathsf{V} = W\mathsf{X}$$

where $W = D^H \Gamma D$, where D is the discrete Fourier transform DFT matrix. Further, $\Gamma = \mathrm{diag}(e^{j\frac{\beta_2 \epsilon}{2}\omega_i^2})$ where (ω_i) is discretization of ω.

Linear step is multiplication by a (convolutional) weight matrix W.

Note. In generative model, W is not learned. □

Nonlinear Step

$$\frac{\partial q(t,z)}{\partial z} = \underbrace{j\gamma|q(t,z)|^2 q(t,z)}_{\textbf{nonlinearity}}$$

Set $q(t,z) = r(t,z)e^{j\phi(t,z)}$. We have

$$(\dot{r} + jr\dot{\phi})e^{j\phi(t,z)} = j\gamma r^3 e^{j\phi(t,z)}$$

Then:

$$\dot{r} + jr\dot{\phi} = j\gamma r^3 \qquad \Rightarrow \qquad \begin{cases} \dot{r} = 0 & \Rightarrow \quad r(t,z) = r(t,0) \\ \dot{\phi} = \gamma r^2 & \Rightarrow \quad \phi(t,z) = \phi(t,0) + \gamma z r^2(t,0) \end{cases}$$

Thus:

$$q(t,z) = r(t,z)e^{j\phi(t,z)} = r(t,0)e^{j\phi(t,0)+j\gamma z r^2(t,0)} = q(t,0)e^{j\gamma|q(t,0)|^2 z}$$

$$\begin{cases} q(t,z) = q(t,0)e^{j\gamma|q(t,0)|^2 z}, & (**) \\ \hat{q}(\omega,z) = \hat{q}(\omega,0) * \mathcal{F}\left(e^{j\gamma|q(t,0)|^2 z}\right)(\omega) \end{cases}$$

Nonlinearity is a simple signal-dependent phase change in the time domain

Nonlinearity is a convolution in the frequency domain, giving rise to spectral broadening

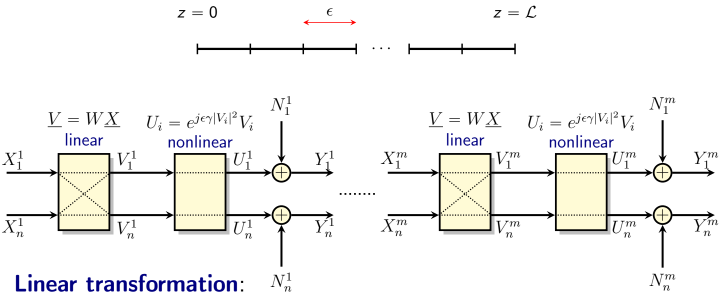
Note that $(**)$ is memoryless. Thus, if the input output of the nonlinear step in one layer are $\mathrm{V} \in \mathbb{C}^n$ and $\mathrm{U} \in \mathbb{C}^n$, discretizing (**)

$$U_i = \sigma(V_i),$$

where $\sigma(x)$ is the activation function

$$\sigma(x) = xe^{j\gamma\epsilon|x|^2}.$$

Generative Neural Network



- **Linear transformation**:

$$\mathrm{V} = \mathrm{WX}$$

where W is the weight matrix.

- Memoryless **nonlinear transformation**:

$$U_i = \sigma(V_i),$$

where $\sigma(x) = xe^{j\epsilon\gamma|x|^2}$ is the activation function.

- **Noise addition:**

$$\mathrm{Y} = \mathrm{U} + \mathrm{N}, \quad \mathrm{N} \sim N(0, \sigma^2 I_n)$$