Assignment 3 – Data-driven modelling of optical amplifiers

Darko Zibar

January 9, 2025

Exercise 1 (30%)
In this exercise, we would like to demonstrate the capabilities of a neural network to model various functions. The objective of this exercise is that you implement a single hidden layer neural network with one input and one output. The number of hidden nodes should be variable that can be adjusted. Then, you should demonstrate that the implemented neural network can be used to model the following functions:

1. $f(x) = x^2$
2. $f(x) = x^3$
3. $f(x) = \sin(x)$
4. $f(x) = |x|$

In the same plot, please show the training data and the prediction (modeling) capabilities of the implemented neural network on the test data. Figures similar to Fig. 5.3, page 231 in Bishop book or lecture slides should be produced. To implement the training of the neural network, use the approach based on random projection method described in the slides. The activation function should be $\tanh(\cdot)$ and you need to choose the appropriate variance for the neural network's first layer random weights initialization as well as for the number of hidden nodes. You will typically need to try different values and experiment a bit.

The training data should be generated by sampling $N = 200$ points from a uniform distribution between -1 and 1. i.e. $x \sim \text{Uniform}(-1, 1)$. Investigate the performance by varying the number of hidden nodes and the number of training data N. What are the conclusions?

Next, we will investigate the performance of the neural network in learning a bit more complicated function and its inverse. Investigate if your implemented neural network can learn the following function: $y = x + 0.3\sin(2\pi x) + n$, where n is white Gaussian noise with variance $\sigma^2 = 10^{-3}$. The training data should be generated by sampling $N = 200$ points from a uniform distribution between -1 and 1. i.e. $x \sim \text{Uniform}(-1, 1)$. Plot the training data and evaluation of the neural network on the test data in the same figure. To make it work, you will need to experiment with different number of hidden nodes. Comment on the results.

Investigate if the implement neural network can learn the inverse function. For a forward function x is input and the generated $y = 0.3\sin(2\pi x) + n$ is the output. For the inverse function modelling, the generated y is the input while x is the output. Plot the training data for the inverse function and the learned inverse function that the neural network learns. Comment on the results. Explain What the difference is in learning the forward and the inverse function? Investigate the performance as you sweep the number of hidden layers, variance for the initialization of the neural network and training data size. For the inspiration have a look at Fig. 5.19

in Bishop book.

Exercise 2 (70%)
The objective of this exercise is to demonstrate the capabilities of a neural network to learn a forward and inverse model of a Raman amplifier from the training data. The gain of the Raman amplifier, $= G([p_1, ..., p_p, \lambda_1, ..., \lambda_p])$, is controlled by pump powers, $[p_1, ..., p_p]$ and pump wavelengths, $[\lambda_1, ..., \lambda_p]$, where p is the number of pumps. For a Raman amplifier the relationship between the input, X, and the output signal, $y(t)$, is governed by: $y(t) = G(p_1, ..., p_p, \lambda_1, ..., \lambda_p, X$. We consider only three pumps ($p = 3$) in this exercise.

The training data set is generated by uniformly sampling different combinations of pump powers and wavelengths and recording the corresponding gain per channel.i.e.
$\mathcal{D} = \{p_1[k], ..., p_3[k], \lambda_1[k], ..., \lambda_3[k] | G_1[k], ..., G_M[k]\}_{k=1}^N$. N is the number of training points and M is the the number of channels for which the gain is measured. The training data is stored in the file `TrainingDataRamanAmp.mat`

1. Load the data set `TrainingDataRamanAmp.mat` and determine M and N. Plot pump powers, pumps wavelength and 50 examples of the gain. What do you observe? Does the data look as expected?

2. Split the data into training and test set set. Implement a multi-layer neural network and show that it can be used to learn the forward model, i.e. mapping between the pump powers and wavelengths and gains. You will need to experiment with the number of hidden units, weight initialization and the number of layers. For the training of multi-layer neural network, use Moore-Penrose Pseudo Inverse (random projection method). All weights, expect or the last layer should be randomly initialized. Implement the model averaging to improve the prediction performance. Plot the ten examples of the predicted and the true gains. Compute the prediction error on the test set and plot the histogram of the prediction error. Comment on the figures.

3. Investigate if the multi-layer neural network can learn the inverse model i.e. mapping between the gains and pump powers and wavelengths. Plot the predicted pump powers and wavelengths versus the true ones. Comment on the figures. Does the model averaging improve the performance

4. Determine pump powers and wavelengths to necessary to obtain flat gain profiles ranging from 3 dB to 20 dB in steps of 1. Test if the predicted pump powers and wavelength produce flat gain profiles.