Exercise Sheet 5

Exercise 19 Function Approximation

a) Construct a multi-layer perceptron with about 10 neurons that approximates the function $y = x^2$ in the interval $[0.5, 4.5]$ by a step function! Consider that there are (at least) two possibilities for doing this: one that uses two hidden layers, and one that uses only one hidden layer.

b) How can the approximation be improved? (State at least two possibilities.)

Exercise 20 Function Approximation

a) Consider the indicator function of the rational numbers over the set of the real numbers (also known as the Dirichlet function), that is, the function

$$f : \mathbb{R} \to \{0, 1\}, \qquad x \mapsto \begin{cases} 1, & \text{if } x \in \mathbb{Q}, \\ 0, & \text{otherwise.} \end{cases}$$

Is it possible to approximate this function with a neural network (multi-layer perceptron) with arbitrary precision (for simplicity: in the interval $[-1, 1]$)?

b) Consider the simple function

$$g : \mathbb{R} \to \mathbb{R}, \qquad x \mapsto x^{-1}.$$

Is it possible to approximate this function with a neural network (multi-layer perceptron) with arbitrary precision (for simplicity: in the interval $[-1, 1]$)?

c) What can be concluded from the result of parts a) and b) about the computational capabilities of neural networks (multi-layer perceptrons)?

Exercise 21 Gradient Descent

Approximate the minimum of the function $f(x_1, x_2) = 2x_1^2 - 2x_1 + x_2^2 - x_2$ with the help of gradient descent! Start from the initial approximation $(x_1, x_2) = (0, 0)$ and use $\eta = 0.2$ as the step width parameter (learning rate)!

Exercise 22 Gradient Descent

In Exercise 15a we considered fitting a straight line $y = a + bx$ to the data points $(-2, 0), (0, 1), (1, 3), (2, 5)$. In that exercise we used the analytical solution via setting up and solving the system of normal equations. In this exercise we want to solve the same problem with the help of gradient descent.

a) What is the error function for this task?

b) What is the gradient descent update rule for this task?

c) Compute a few update steps starting from $a = b = 0$! Try different step width parameters, e.g. from the sequence 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, 3 etc.!