W3WI DS304.1 Applied Machine Learning Fundamentals

Exercise Sheet #5 - Neural Networks / Deep Learning

Question 1 ⊠ 2021 **(Perceptron)**

Under what circumstances does the *Perceptron* learning algorithm converge?

Question 2 ⊠ 2020 **(Number of network parameters)**

You want to train a neural network on the *MNIST* dataset to recognize hand-written digits. The images of 10 possible digits (the classes) have a resolution of 28×28 pixels.

The MLP (*multi-layer perceptron*) used for the task has two hidden layers with 64 and 32 units, respectively. Each layer has a constant bias input and the classes are one-hot encoded.

How many adjustable network parameters does the model have?

Question 3 ⊠ 2020 **(Neural networks for regression)**

Your colleague suggests to use neural networks to solve a regression task. Which activation function would you have to use in the output layer of your network to achieve the desired result?

Question 4 ⊠ 2021 **(Activation functions)**

Which statements regarding the activation functions of neural networks are correct?

☐ Activation functions should be non-linear.

☐ The *softmax* activation function is usually used in the output layer of a neural network.

☐ One problem of the *ReLU* function is the vanishing gradient.

☐ The *ReLU* activation is computed according to $\min(0, x)$.

Question 5 ⊠ 2021, modified **(Network architectures)**

What types of neural network are usually applied to classify (a) images, and (b) sequences? Do some research and explain these types of networks.

Question 6 ⚡ 2023, modified (Network training with gradient descent)

Your task is to train the neural network depicted in figure 1 using the *stochastic gradient descent* algorithm. The current training example is given by the vector $\boldsymbol{x} := \begin{pmatrix} 0 & 1 \end{pmatrix}^{\mathsf{T}}$. The label is one-hot encoded and given by $\boldsymbol{y} := \begin{pmatrix} 1 & 0 \end{pmatrix}^{\mathsf{T}}$.

Network architecture: The network consists of one hidden layer with *ReLU* activation as well as an output layer with *sigmoid* activation. For simplicity you decide to use the *least squares error* as the loss function (as shown in the lecture).

Compute the weight updates for all network parameters when applying the learning rate $\alpha := 0.5$.
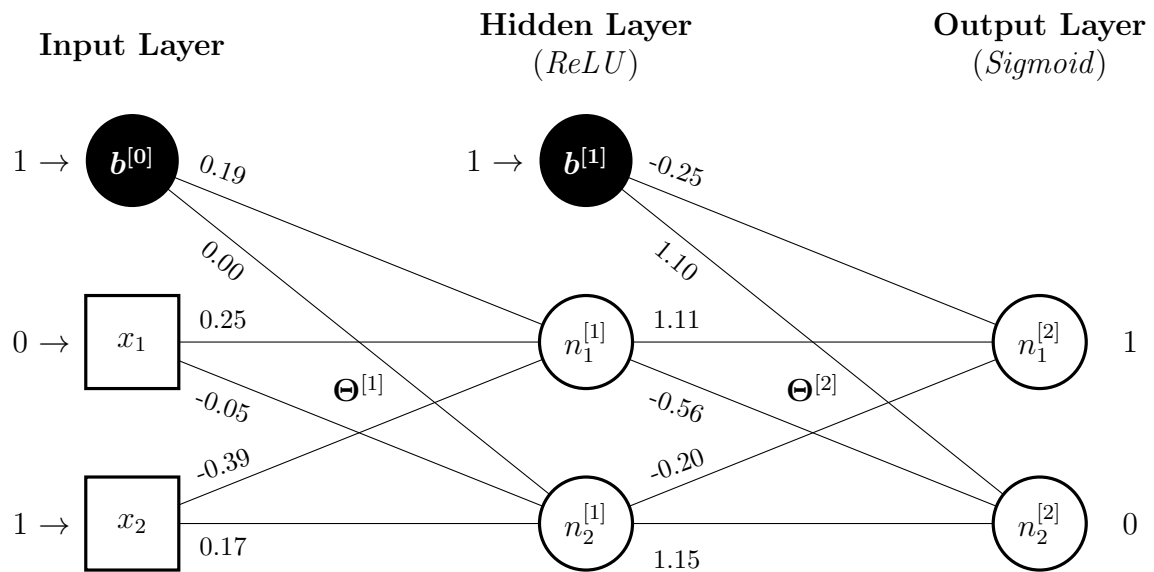

Figure 1: Visualization of the neural network used for the task above.