Deadline: Th. Dec. 13, 10:00 am Drop your printed or legible handwritten submissions into the boxes at Samelsonplatz, or upload them as `.pdf` or `.ipynb` files onto the LearnWeb.

Exercise 1 (`XOR` revisited - 10 points)**.** In tutorial 5.1 we have seen that the logistic regression classifier (with $\sigma(x) = \frac{1}{1+e^{-x}}$)

$$\hat{y}(x) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_N x_N) = \sigma(\begin{bmatrix}1, x^\intercal\end{bmatrix} \cdot \beta) \tag{1}$$

can correctly model the binary `AND` function, but not `XOR`. A single such classifier is also known as a Perceptron. In this exercise we will see that although a single Perceptron cannot solve `XOR`, a network consisting of Perceptron units – a so called Multi-Layer-Perceptron (MLP) – can.

1. (5) Show that the binary `XOR` function can realized by combining the binary `AND` and `OR` functions plus negations.
2. (5) Design a MLP consisting of 3 Perceptrons which correctly models the `XOR`-function.

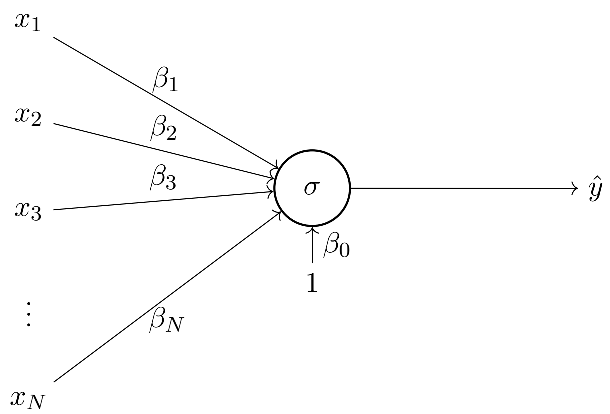

Figure 1: Perceptron model

Exercise 2 (Back-propagation - 10 points)**. Note:** This is not a programming exercise! Consider the following neural network (MLP without bias terms & no regularization):

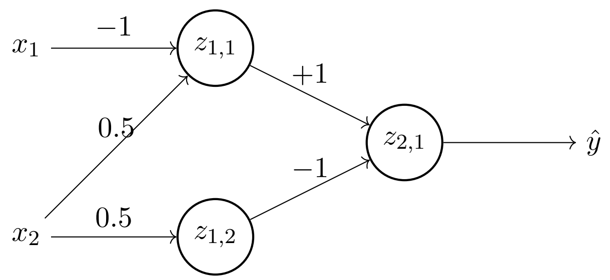

Figure 2: MLP model

1. (3) Perform the forward pass for the single data-point $x^* = (1, 2)$ (compute u_l and z_l)
2. (5) Given that $y(x^*) = 1$ update all the weights once via back-propagation with learn-rate $\eta = 1$, using the log-likelihood loss function $\ell = y \log(\hat{y}) + (1-y) \log(1 - \hat{y})$. (compute g_l and the updated weights)
3. (2) Perform another forward pass, using the updated weights. Comment on the result.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

(a) "OR"

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

(b) "AND"

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

(c) "NAND"

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

(d) "XOR"

Table 1: Some classical binary functions