Deadline: Th. December 12th , 10:00 Drop your printed or legible handwritten submissions into the boxes at Samelsonplatz. Alternatively upload a `.pdf` file via LearnWeb. (e.g. exported Jupyter notebook)

1. Neural Networks – Practice (10 points)

Consider the Neural Network depicted in Figure 1. Bias terms are omitted in this exercise.

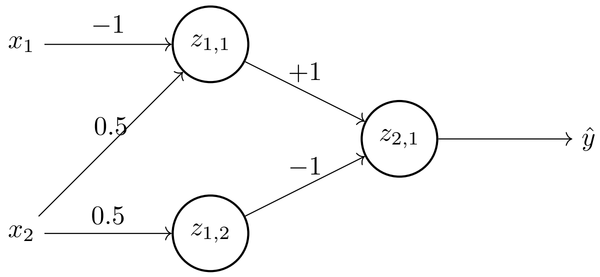


Figure 1: Neural Network model

A. [3p] Perform the forward pass for the single input datapoint $x = (1, 2)$.

B. [5p] Given the single training instance $x = (1, 2)$, $y = 1$ update all the weights once via back-propagation, using the log-likelihood objective function function $\ell = y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})$, sigmoid activation function, learn rate $\eta = 1$ and without any regularization.

C. [2p] Perform another forward pass, using the updated weights. Comment on the result.

2. Neural Networks – Theory (10 points)

In tutorial 3 we have seen that a logistic regression model, i.e. a single artificial neuron with sigmoid activation function cannot solve the XOR dataset.

(a) OR

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

(b) AND

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

(c) NOR

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

(d) NAND

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

(e) XOR

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

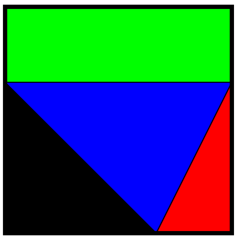
Table 1: Some binary functions

A. [3p] Show that the binary XOR function can be realized as combination of the binary AND and OR functions plus negations.

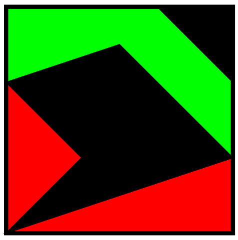
B. [7p] Design a Neural Network consisting of 3 neurons which realizes the binary XOR function. Provide a full explicit description of the network! (activation function, weights, etc.)

3. Neural Networks – Bonus (10 points)

In the lecture you learned about the "Universal Approximation" property of Neural Networks. In this problem we want to see how neural networks might learn certain function "exactly" and how adding more layers allows us solve more complicated problems.



(a) Partition into convex polytopes



(b) Partition into general polytopes

A. [5p] Show that any classification problem where each class occupies a distinct region in space, given by a single convex polytope (cf. Figure 2a) can be solved exactly by a neural network with 2 hidden layers.

B. [5p] Show that any classification problem where each class occupies a distinct region in space, given by a finite set of (possibly non-convex) polytopes (cf. Figure 2b) can be solved exactly by a neural network with 3 hidden layers.

Hint: The same ideas as in Problem 2 might be useful here.