Florian Rösel, Prof. Dr. Frauke Liers winter semester 2021/22

Mathematics of Learning – Worksheet 6

Exercise 1 [Derivative of logistic activation function]. 4 **points**

Let $\psi\colon \mathbb{R} \to \mathbb{R}$ the logistic activation function for a perceptron defined as

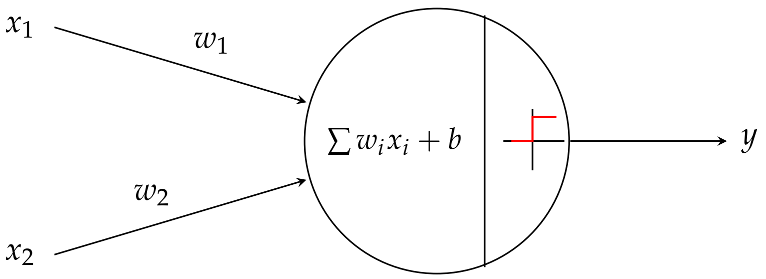
$$\psi(t) \;:=\; \frac{1}{1+e^{-t}}$$

Show that the derivative ψ' of ψ can be computed as:

$$\psi'(t) = \psi(t)(1-\psi(t)).$$

Exercise 2 [Implementation of perceptrons for binary logic functions]. 8 **points**

Let $f_\theta\colon \mathbb{R}^2 \to \mathbb{R}$ be a parametrized map realized by the following binary perceptron that maps two inputs $\vec{x} = (x_1, x_2)$ to an output y:



Here, $\theta \in \mathbb{R}^3$ is the vector of free parameters with $\theta := (w_1, w_2, b)$, where w_1, w_2 are the weights of the respective inputs and b is the bias of the perceptron. We assume that the activation function of the perceptron is the *Heavyside step function* $H\colon \mathbb{R} \to \{0,1\}$ defined as :

$$H(x) \;:=\; \begin{cases} 0, & \text{if } x < 0, \\ 1, & \text{if } x \geq 0. \end{cases}$$

Implement a Python function `perceptron(x, theta)` that return a output `y`, which is either 0 or 1. Use this function to implement a family of perceptrons, which realize the following binary logic functions:

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

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

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

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

NOR		
x_1	x_2	y
0	0	1
1	0	0
0	1	0
1	1	0

Hint: One of the binary logic functions cannot be realized by a simple perceptron. Explain the reasons for this and suggest an alternative realization.

Exercise 3 [Prediction of wine quality].

Find the data file winequality-white.csv on StudOn. It contains several columns with some measurable properties of the wine (sugar, pH value, sulfur dioxide etc.) and some quality labels which have been determined (most likely) manually by some “wine experts”. The second half is without quality labels (these will be published in a week that you can test your classifier functions).

a) Write two functions which implement labeling via linear regression (remember: you can be creative in your choice of the regression basis - taking the features alone will most likely lead to inefficient classifiers) and k-nearest-neighbor.

b) split the (labeled) data set in appropriately large subsets, train your regression/nearest-neighbor classifier, and test it on your data set. Sample the test set 1. randomly 2. take only the best wines, and compare the results.

Acknowledgement, for generating a previous version of this sheet, to Leon Bungert and Daniel Tenbrinck.