Exercise 1-1  Recap: Vector Calculus

Compute $\frac{\partial g(x)}{\partial x}$ for the functions below. **Hint:** For a function $g(x) : \mathbb{R}^n \to \mathbb{R}$ with $x \in \mathbb{R}^n$ holds:

$$\frac{\partial g(x)}{\partial x} = \left[ \frac{\partial g(x)}{\partial x_1} \frac{\partial g(x)}{\partial x_2} \ldots \frac{\partial g(x)}{\partial x_n} \right].$$

a) $g(x) = \sum_{i=1}^{n} x_i$,  
b) $g(x) = \langle x, x \rangle$, the standard scalar product of $x$ with itself,  
c) $g(x) = (x - \mu)^2$ für $\mu \in \mathbb{R}^n$.

Exercise 1-2  Boolean Function as Perceptron

Consider the boolean function $\text{or}$ ($\lor$) for two binary inputs.

- a) Illustrate the different inputs as well as possible separating hyperplanes graphically.
- b) Given the above picture, guess weights for a perceptron (with outputs 0 and 1) such that the perceptron is a classifier for the $\lor$ function. Instead of using the $\text{sign}$ function for getting the classification output, as in the lecture, use the Heaviside function $f$ for classification:

$$f(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

- c) Initialize the weight vector as $w = (0, 0, 0)$ and learn the right weights employing the algorithm of the lecture and a learning rate $\eta = 0.2$. Use the following learning rule:

$$w_j \leftarrow w_j + \eta \cdot (y_i - \hat{y}_i)x_{i,j}$$

Start training vector $p_3 = (1, 1)$ and proceed with increasing index (in contrast to the principle of random sampling). Use $p_0 = (0, 0), p_1 = (0, 1)$ and $p_2 = (1, 0)$.

Exercise 1-3  Applying the Perceptron learning rule in Python

The provided Jupyter Notebook will guide you through the steps of implementing and training a Perceptron in Python. Complete the specified tasks by filling in the missing code. You can further use your implementation to play around with the Perceptron and apply it to different datasets or explore its behavior by changing the learning rate or the order in which the instances are selected.