Machine Learning  
Summer 2017  
Solutions to Exercise Sheet 1

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:

$$\frac{\partial g(x)}{\partial x} = \begin{bmatrix} \frac{\partial g(x)}{\partial x_1} & \frac{\partial g(x)}{\partial x_2} & \cdots & \frac{\partial g(x)}{\partial x_n} \end{bmatrix}.$$  
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 *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 *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

Let $A$ and $B$ be two classes, both comprising two patterns:

$$A = \{ p_1 = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, p_2 = \begin{pmatrix} 1 \\ 0.5 \end{pmatrix} \}, \quad B = \{ p_3 = \begin{pmatrix} 0.5 \\ 1.5 \end{pmatrix}, p_4 = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} \}$$

Classes $A$ and $B$ are labelled with 1 and $-1$, respectively.

Solve the following exercises either using pen and paper or a programming language of your choice. Also, visualize the partial results.

a) How many iterations are required by the pattern-based perceptron learning rule in order to separate classes $A$ and $B$ correctly if the weight vector $w$ is initialized as $(0, 1, -1)$ and step size $\eta$ is set to 0.1?

b) How many iterations are required if $\eta = 0.25$? Is the order of the considered patterns relevant? If so, give an example, otherwise, prove it.

c) After how many iterations does the gradient-based learning rule terminate for both $\eta$? In this case: Is the order of the considered patterns relevant?

*Hint:* If you need more than 10 iterations, you miscalculated.