Ludwig-Maximilians-Universitaet Muenchen
Institute for Informatics
Prof. Dr. Volker Tresp
Rajat Koner
Sebastian Schmoll

28.04.2020

Machine Learning

Summer 2021

Exercise Sheet 01

Exercise 01-1 Recap: Vector Calculus

Compute $\nabla g(\mathbf{x}) = \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}$ for the functions below with $\mathbf{x} = (x_1, x_2, \dots x_n)^T$. *Hint:* Recall that for a function $g(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ with $\mathbf{x} \in \mathbb{R}^n$ holds:

$$\nabla g(\mathbf{x}) = \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial g(\mathbf{x})}{\partial x_1} \\ \frac{\partial g(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial g(\mathbf{x})}{\partial x_n} \end{pmatrix}$$

(a) $g(\mathbf{x}) = 2x_1 + 3x_2^2 + x_3$ for $n = 3$,

(b) $g(\mathbf{x}) = \sum_{i=1}^{n} x_i$,

(c) $g(\mathbf{x}) = \langle \mathbf{x}, \mathbf{x} \rangle = \mathbf{x}^T \mathbf{x}$, the standard scalar product of $\mathbf{x}$ with itself,

(d) $g(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^2$ for $\boldsymbol{\mu} \in \mathbb{R}^n$.

Possible Solution

(a) $\nabla g(\mathbf{x}) = \begin{pmatrix} 2 \\ 6x_2 \\ 1 \end{pmatrix}$

(b) $g(\mathbf{x}) = x_1 + x_2 + \ldots + x_n \quad \Rightarrow \quad \nabla g(\mathbf{x}) = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$

(c)

$$\frac{\partial g(\mathbf{x})}{\partial x_j} = \frac{\partial \mathbf{x}^T \mathbf{x}}{\partial x_j} = \frac{\partial (x_1^2 + x_2^2 + \ldots + x_n^2)}{\partial x_j} = (n-1) \cdot 0 + \frac{\partial x_j^2}{\partial x_j} = 2x_j$$

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial x_1^2}{\partial x_1} & \frac{\partial x_2^2}{\partial x_2} & \dots & \frac{\partial x_n^2}{\partial x_n} \end{bmatrix}^T = 2 \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}^T = 2\mathbf{x}$$

(d)

$$\frac{\partial g(\mathbf{x})}{\partial x_j} = \frac{\partial \left(\mathbf{x}-\boldsymbol{\mu}\right)^2}{\partial x_j} = \frac{\partial \left(\mathbf{x}-\boldsymbol{\mu}\right)^T \left(\mathbf{x}-\boldsymbol{\mu}\right)}{\partial x_j} = \frac{\partial \sum_{i=1}^{n} \left(x_i-\mu_i\right)^2}{\partial x_j} = 2\left(x_j-\mu_j\right)$$

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = 2\left[\begin{array}{cccc} x_1-\mu_1 & x_2-\mu_2 & \ldots & x_n-\mu_2 \end{array}\right]^T = 2(\mathbf{x}-\boldsymbol{\mu})$$

Alternatively, you can use the result of (b) and the chain rule.

Exercise 01-2 Boolean Function as Perceptron

Consider the boolean function *or* ($\vee$) for two binary inputs.

(a) Illustrate the different inputs as well as possible separating hyperplanes graphically.

Possible Solution

Vier Punkte an den Stellen $p_0 = (0,0), p_1 = (0,1), p_2 = (1,0), p_3 = (1,1)$, jede mögliche trennende Hyperebene trennt $(0,0)$ von den anderen drei.

(b) Given the illustration from (a), guess weights for a perceptron (with outputs 0 and 1) such that the perceptron is a classifier for the $\vee$ function. Instead of using the $sign$ function for getting the classification output, as in the lecture, use the Heaviside function f for classification:

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

Possible Solution

Intercept and gradient can easily be guessed from the illustration.
For example, $1/2$ und -1 qualify a solution, respectively.
The equation of the hyperplane is:

$$h(x) = \mathbf{x}^T\mathbf{w} = \sum_{j=0}^{2} w_j x_j = w_0 + w_1 x_1 + w_2 x_2$$

For $h = 0$, we obtain $0 = w_0 + w_1 x_1 + w_2 x_2$.
Rearrange the equation to bring it into the form of the equation of a straight line ($y = m \cdot x + t$):

$$x_2 = \underbrace{-\frac{w_0}{w_2}}_{=\frac{1}{2}} \underbrace{-\frac{w_1}{w_2}}_{=-1} \cdot x_1.$$

If we set $w_2 = 1$, then $w_0 = -1/2$ and $w_1 = 1$ parametrize the above line.
Thereby it holds $w = (-1/2, 1, 1)$.
Check if classified correctly:
$\hat{y_0} = f(h(p_0)) = f(-1/2 + 1 * 0 + 1 * 0) = f(-1/2) = 0$ ✓

(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 $p_2 = (1, 0)$.

Possible Solution

sample	y_i	$\hat{y}_i = f(h(p_i))$	weight update
$p_3 = (1, 1)$	1	$f(p_3^T w) = f(0) = 1$	–, since $\hat{y_3} = y_3$
$p_0 = (0, 0)$	0	$f(p_0^T w) = 1$	$w_0 \leftarrow 0 + 0.2 \cdot (0 - 1) \cdot 1 = -0.2$ $w_1 \leftarrow 0 + 0.2 \cdot (0 - 1) \cdot 0 = 0$ $w_2 \leftarrow 0 + 0.2 \cdot (0 - 1) \cdot 0 = 0$
$p_1 = (0, 1)$	1	$f(p_1^T w) = 0$	$w_0 \leftarrow -0.2 + 0.2 \cdot (1 - 0) \cdot 1 = 0$ $w_1 \leftarrow 0 + 0.2 \cdot (1 - 0) \cdot 0 = 0$ $w_2 \leftarrow 0 + 0.2 \cdot (1 - 0) \cdot 1 = 0.2$
$p_2 = (1, 0)$	1	$f(p_2^T w) = 1$	–, since $\hat{y_2} = y_2$
p_3	1	$f(p_3^T w) = 1$	–
p_0	0	$f(p_0^T w) = 1$	$w_0 \leftarrow 0 - 0.2 \cdot 1 = -0.2, w_1 = 0, w_2 = 0.2$
p_1	1	$f(p_1^T w) = 1$	–
p_2	1	$f(p_2^T w) = 0$	$w_0 \leftarrow -0.2 + 0.2 \cdot 1 = 0, w_1 = 0.2, w_2 = 0.2$
p_3	1	$f(p_3^T w) = 1$	–
p_0	0	$f(p_0^T w) = 1$	$w_0 \leftarrow 0 - 0.2 \cdot 1 = -0.2, w_1 = 0.2, w_2 = 0.2$
p_1	1	$f(p_1^T w) = 1$	–
p_2	1	$f(p_2^T w) = 1$	–
p_3	1	$f(p_3^T w) = 1$	–
p_0	0	$f(p_0^T w) = 0$	–

Exercise 01-3 Applying the perceptron learning rule

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

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

Classes A and B are labeled 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 seperate classes A and B correctly if the weight vector w is initialized as $(0, 1, -1)$ and step size η 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 η? In this case: Is the order of the considered patterns relevant?

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

Possible Solution

Reminder: Straight line: $h = w_0 + w_1 x_1 + w_2 x_2$.

The more common form $y = ax+t$ in our case corresponds to $x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$.

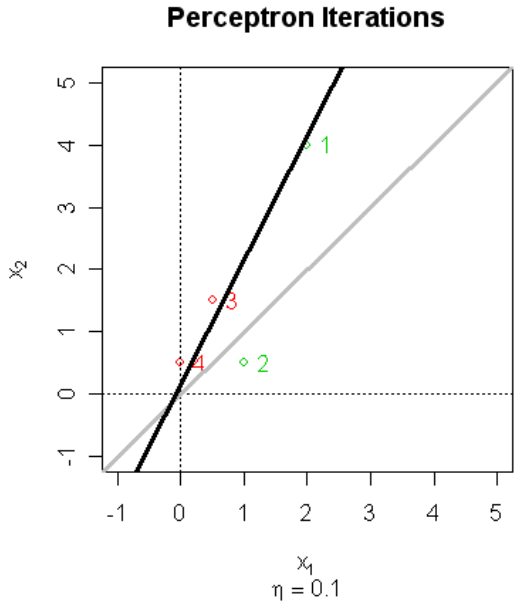
a)

Find misclassified pattern: plug all p_i in classification formula

$$\hat{y} = \text{sign}\left(\sum_{j=0}^{M-1} w_j x_{i,j} \right).$$

$\Rightarrow$ One pattern (p_1) is misclassified: $\hat{y} = \text{sign}\,(w_0 \cdot 1 + w_1 \cdot p_{1,1} + w_2 \cdot p_{1,2}) =$
$= \text{sign}\,(0 + 2 - 4) = -1 \neq 1 = y_1$. $\Rightarrow$ Plug p_1 in perceptron: $w_0 = w_0 + \eta \cdot y_1 \cdot 1 = 0 + 0.1 = 0.1$

$w_1 = w_1 + \eta \cdot y_1 \cdot p_{1,1} = 1 + 0.1 \cdot 2 = 1.2$
$w_2 = w_2 + \eta \cdot y_1 \cdot p_{1,2} = -1 + 0.1 \cdot 4 = -0.6$
Find misclassified patterns: none. $\Rightarrow$ terminates after 1 iteration.



b.w.

Possible Solution

b) Analogously to a): (p_1) is misclassified.

$w_0 = 0 + 0.25 \cdot 1 \cdot 1 = 0.25$, $w_1 = 1 + 0.25 \cdot 1 \cdot 2 = 1.5$, $w_2 = -1 + 0.25 \cdot 1 \cdot 4 = 0$
$\Rightarrow w = (0.25, 1.5, 0)$

$\Rightarrow$ 2 samples misclassified: p_3, p_4. Leaves us two options to proceed:

2 Insert p_3 into Perceptron:

$\Rightarrow w = (0, 1.375, -0.375)$, p_3 still misclassified.

3 Again, p_3 in perceptron:

$\Rightarrow w = (-0.25, 1.25, -0.75)$, p_3 correct; now p_1 is missclassified.

4 p_1 in perceptron:

$\Rightarrow w = (0, 1.75, 0.25)$, p_1 correct; now p_3 and p_4 are missclassified.

5 Choose p_3 for perceptron:

$\Rightarrow w = (-0.25, 1.625, -0.125)$, p_4 correct; p_3 still misclassified.

6 p_3 in perceptron:

$\Rightarrow w = (-0.5, 1.5, -0.5)$, all correctly classified. Done after 6 iterations.

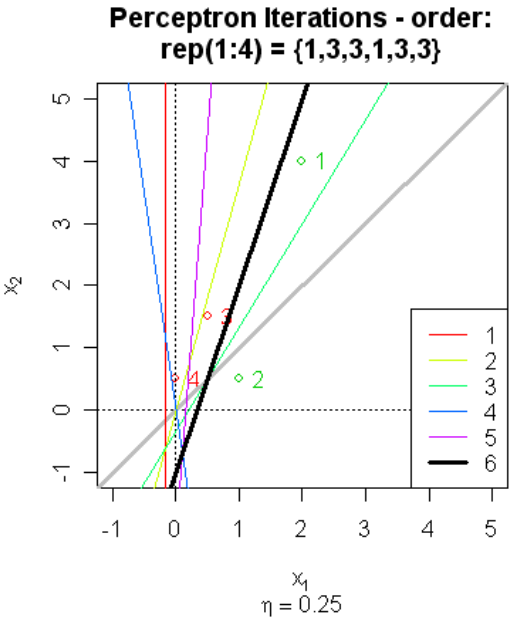
Alternative Order:

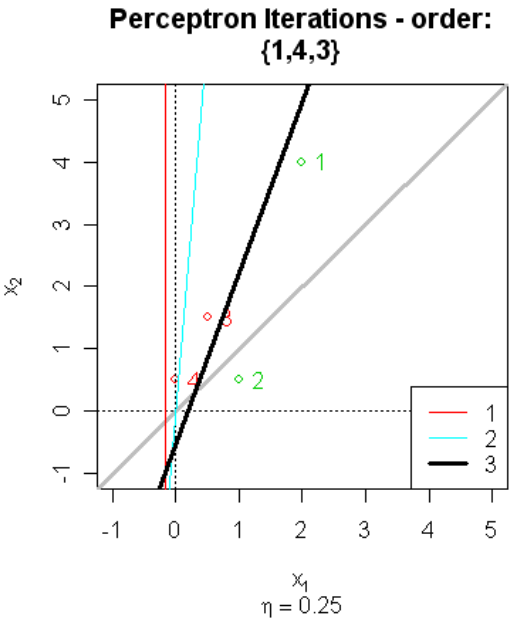
2 p_4 in perceptron:

$\Rightarrow w = (0, 1.5, -0.125)$, p_3 is missclassified.

3 p_3 in perceptron:

$\Rightarrow w = (-0.25, 1.375, -0.5)$, all correctly classified. Done after 3 iterations





b.w.

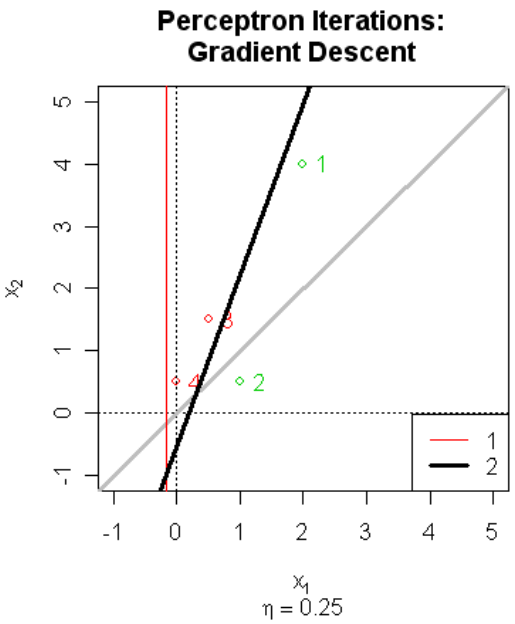
Possible Solution

c)

Using gradient descent we get the same result as with sample-based gradient descent after the first iteration. Reason: Only one pattern (p_1) is missclassified. That means, for $\eta = 0.1$ we are done after 1 iteration. ($w = (0.1, 1.2, -0.6)$)

For $\eta = 0.25$ the first iteration results in $w = (0.25, 1.5, 0)$. Therefore, p_3 and p_4 are missclassified. Next iteration yields:

$w_0 \leftarrow w_0 + \eta \sum_{i \in \mathcal{M}=\{p_3,p_4\}} y_0 = 0.25 + 0.25 \cdot (-1 - 1) = -0.25$
$w_1 \leftarrow w_1 + \eta \sum_{i \in \{p_3,p_4\}} x_{i,1} y_1 = 1.5 + 0.25 \cdot (-0.5 - 0) = 1.375$
$w_2 \leftarrow w_2 + \eta \sum_{i \in \{p_3,p_4\}} x_{i,2} y_2 = 0 + 0.25 \cdot (-1.5 - 0.5) = -0.5$



Hence, we are done after 2 iterations.

There is no order of insertion as with the sample-based learning rule, since in every iteration all wrongly classified patterns are considered.