AI1072: Machine learning, exercise sheet 4

1 Preparations (on paper)

Compute the following derivatives!

a) $\frac{\partial}{\partial x_3} \sum_{i=1}^{5} x_i$

b) $\frac{\partial}{\partial x_6} \sum_{i=1}^{5} x_i$

c) $\frac{\partial}{\partial x_3} \sum_{i=1}^{5} (2x_i - 5)^2$

d) $\frac{\partial}{\partial x_3} \sum_{i=1}^{5} (x_i^2 - 5x_i)^2$

e) $\frac{\partial}{\partial x_3} \cos(x_3)x_3$

f) $\frac{\partial}{\partial x_3} \cos(x_3)x_4$

g) $\frac{\partial}{\partial x_3} \cos(\exp(x_3))$

2 Vector-vector chain rule

Given the function $\vec{g}(\vec{x})$ with $g_i(\vec{x}) = 2x_i$, and $f(\vec{x}) = ||\vec{x}||^2$: compute the partial derivative $\frac{\partial f(g(\vec{x}))}{\partial x_1}$!

3 Vector-vector chain rule

Given the function $\vec{y}(\vec{x}) = \vec{S}(\vec{x})$ and $\mathcal{L}(\vec{y}) = -\sum_i t_i \ln(y_i)$: compute the partial derivative $\frac{\partial \mathcal{L}(\vec{y}(\vec{x}))}{\partial x_1}$! $\vec{S}$ is the softmax function, what does this remind you of??

4 Gradient descent (on paper)

Consider the function $f(\vec{x}) = x_1^2 + 2x_2^2$. Assuming a starting point of $\vec{x}_0 = [1, 3]$ and a step size of $\epsilon = 0.1$: perform 3 steps of gradient descent, that is, compute the values of x_1, x_2, x_3, x_4, and x_5 by iteration.

5 Gradient descent (on the computer)

Write a program that performs N steps of gradient descent for f in Python, with starting point and steps sizes that are constants and which can be changed easily. Implement the function $f(x)$ and its gradient $gradf(x)$ as functions that take a single 1D numpy array x as argument and return the function value as a scalar, or the gradient as an 1D array. After each iteration, display the current value of $\vec{x}$!