Universität des Saarlandes
Prof. Dr. Dietrich Klakow
Lehrstuhl für Signalverarbeitung
NNIA Winter Term 2018/2019

Exercise Sheet 3

Gradient Descent

Deadline: 19.11.2018, 23:59

Exercise 3.1 - Vector Derivatives (1 + 1 + 1 + 1 = 4 points)

In this lecture we will often encounter functions of several variables, i.e. $f : \mathbb{R}^n \to \mathbb{R}$. Knowing how to compute the derivatives of such functions will prove helpful for understanding formulas throughout this lecture. Now let $f : \mathbb{R}^n \to \mathbb{R}$, $w \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{m\times n}$. Prove that the following rules hold:

a) $f(x) = \langle w, x\rangle$, then $\nabla_x f(x) = w$

b) $f(x) = \langle x, Ax\rangle = x^T Ax$, then $\nabla_x f(x) = Ax + A^T x$

c) $f(x) = \|Bx\|_2^2$, then $\nabla_x f(x) = 2B^T Bx$

d) $f(x) = \|Bx - c\|_2^2$, then $\nabla_x f(x) = 2B^T (Bx - c)$

Hints & Remarks:

- Previous rules can be used in later proofs. For example, one might need the result of (c) for solving (d).
- If this exercise is too abstract for you, you can start with fixed dimensions, for example, prove above rules in the two dimensional case (i.e. $n = 2$). A correct proof only in the two dimensional case will receive half of the points. The proof in the general case (i.e. in n dimensions) will receive full points.

Exercise 3.2 - Computational Issues with Softmax (0.5 + 0.5 + 0.5 + 2.5 = 4 points)

The softmax function plays an important role in neural networks, especially when performing classifications. It is defined as follows:

$$softmax : \mathbb{R}^n \to \mathbb{R}^n, softmax(x)_i = \frac{exp(x_i)}{\sum_{j=1}^{n} exp(x_j)}, i = 1, \cdots, n \tag{1}$$

a) Numerical issues might occur when computing softmax functions on a computer. Name these numerical issues and explain them [Hint: think about overflow and underflow].

b) Suggest a remedy to overcome these numerical issues occuring with Softmax computation and explain why it prevents such numerical issues.

c) Show that for any input $x \in \mathbb{R}^n$, $softmax(x)_i \geq 0$ and $\sum_{i=1}^n softmax(x)_i = 1$. That is, the vector $softmax(x) \in \mathbb{R}^n$ can be understood as a probability distribution.

d) Compute the first derivative of $softmax(x)$, i.e. compute the Jacobian matrix of $softmax(x)$.

Exercise 3.3 - Bad Step Size (2 points)

Construct a smooth (i.e. continuously differentiable) function $f : \mathbb{R} \to \mathbb{R}$, a starting point $x_0 \in \mathbb{R}$ with $f'(0) \neq 0$, and step sizes ϵ_k such that $f(x_k)$ will converge to a local maximum when applying gradient descent method.

Hint: if you choose the starting point x_0 and the initial step size ϵ_0 wisely, then it is possible to directly jump to a local maximum in one step. But do not forget to argue that it will converge to that local maximum, i.e. arrive and stay at a certain local maximum. Moreover, many functions can be used for solving this problem, for example, the sine function.

Submission instructions

The following instructions are mandatory. If you are not following them, tutors can decide to not correct your exercise.

- You have to submit the solutions of this assignment sheet as a team of 2-3 students.
- Hand in a **single** PDF file with your solutions.
- Therefore make sure to write the student ID and the name of each member of your team on your submission.
- Your assignment solution must be uploaded by only **one** of your team members to the course website.
- If you have any trouble with the submission, contact your tutor **before** the deadline.