Optimization in Machine Learning | **Universität Mannheim**
HWS 2024 | Prof. Simon Weißmann, Felix Benning

Sheet 2

For the exercise class on the 03.10.2024 at 12:00.
Hand in your solutions by 10:15 in the lecture on Tuesday 01.10.2024.

Exercise 1 (Descent Directions of a Maximum). **(1 Points)**

Let $x_* \in \mathbb{R}^d$ be a strict local maximum of $f : \mathbb{R}^d \to \mathbb{R}$. Prove that every $d \in \mathbb{R}^d$ is a descent direction of f in x_*.

Exercise 2 (Convergence to Stationary Point). **(5 Points)**

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function.

(i) Let $(x_k)_{k\in\mathbb{N}}$ be defined by gradient descent

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k), \quad x_0 \in \mathbb{R}^d$$

with diminishing step size $\alpha_k > 0$ such that $\sum_{k=1}^{\infty} \alpha_k = \infty$. Suppose that $(x_k)_{k\in\mathbb{N}}$ converges to some $x_* \in \mathbb{R}^d$. Prove that x_* is a stationary point of f , i.e. $\nabla f(x_*) = 0$. (2.5 pts)

Hint. *You might want to prove for large enough i, j*

$$\langle \nabla f(x_i), \nabla f(x_j)\rangle \geq \|\nabla f(x_*)\|^2 - 2\epsilon\|\nabla f(x_*)\| - \epsilon^2 =: p(\epsilon).$$

(ii) Assume that f is also L-smooth. Prove for x_n generated by gradient descent with constant step size $\alpha \in (0, \frac{2}{L})$ we have

$$\sum_{k=n}^{m} \|\nabla f(x_k)\|^2 \leq \frac{f(x_n) - f(x_m)}{\alpha(1 - \frac{L}{2}\alpha)} \leq \frac{f(x_n) - \min_x f(x)}{\alpha(1 - \frac{L}{2}\alpha)}$$

for any $n, m \in \mathbb{N}$. Deduce for the case $\min_x f(x) > -\infty$, that we have

$$\min_{k\leq n} \|\nabla f(x_k)\|^2 \in o(1/n). \qquad (2.5 \text{ pts})$$

Hint. *Consider the minimizer from Sheet 1, Ex. 6(i).*

Exercise 3 (Optimizing Quadratic Functions). **(9 Points)**

In this exercise we consider functions of type

$$f(x) = x^T Ax + b^T x + c,$$

where $x \in \mathbb{R}^d, A \in \mathbb{R}^{d\times d}, b \in \mathbb{R}^d, c \in \mathbb{R}$.

(i) Let $H := A^T + A$ be invertible. Prove that f can be written in the forms

$$f(x) = (x - x_*)^T A(x - x_*) + \tilde{c} \tag{1}$$
$$= \tfrac{1}{2}(x - x_*)^T \underbrace{(A^T + A)}_{=:H}(x - x_*) + \tilde{c} \tag{2}$$

for some $x_* \in \mathbb{R}^d$ and $\tilde{c} \in \mathbb{R}$. Argue that H is always symmetric. Under which circumstances is x_* a minimum? (3 pts)

(ii) Argue that the Newton Method (with step size $\alpha_n = 1$) applied to f would jump to x_* in one step and then stop moving. (1 pt)

(iii) Let $V = (v_1, \dots, v_d)$ be an orthonormal basis such that

$$H = V \operatorname{diag}[\lambda_1, \dots, \lambda_d] V^T$$

with $0 < \lambda_1 \leq \cdots \leq \lambda_d$ and write

$$y^{(i)} := \langle y, v_i \rangle.$$

Express $(x_n - x_*)^{(i)}$ in terms of $(x_0 - x_*)^{(i)}$, where x_n is given by the gradient descent recursion

$$x_{n+1} = x_n - h\nabla f(x_n).$$

For which step size h do all the components $(x_n - x_*)^{(i)}$ converge to zero? Which component has the slowest convergence speed? Find the optimal learning rate h^* and deduce for this learning rate

$$\|x_n - x_*\| \leq (1 - \tfrac{2}{1+\kappa})^n \|x_0 - x_*\|.$$

with the condition number $\kappa = \frac{\lambda_d}{\lambda_1}$. (5 pts)

Exercise 4 (Programming exercise). **(9 Points)**

For the Python exercises join the GitHub classroom `https://classroom.github.com/a/8yrTMIm1`. If you are new to git, checkout `https://classroom.github.com/a/dEzm_HGt`