Advanced Computer Vision

Exercise Sheet 2

Winter term 2023
Prof. Dr. Niels Landwehr
Dr. Ujjwal

Available: 14.11.2023
Hand in until: 21.11.2023 11:59am
Exercise session: 24.11.2023

Task 1 – Calculate Gradients [5 points]

Calculate the gradients of the following real-valued functions of three real variables:

a) $f(x, y, z) = xz^2 e^y \cos y$,

b) $g(x, y, z) = \log(\sqrt{x^2 + y^2 + z^2})$.

Task 2 – Analytical Minimum of Rosenbrock Function [15 points]

a) Calculate the gradient of the following function:

$$f(x, y) = (a - x)^2 + b(y - x^2)^2, \quad \text{with } a, b > 0 \text{ and } x, y \in \mathbb{R}$$

b) Find a minimum of the function $f(x, y)$ analytically by setting its gradient to zero and checking that the Hessian matrix at the point you have found is positive definite. You may use that for $\mathbf{A} \in \mathbb{R}^{2\times2}$, $\mathbf{A}$ is positive definite if $trace(\mathbf{A}) > 0$ and $det(\mathbf{A}) > 0$.

The function $f(x, y)$ is the so-called Rosenbrock function, which is often used as a performance test problem for optimization algorithms.

Task 3 – Gradient Descent [15 points]

Write a Python notebook that finds the minimum of the Rosenbrock function $f(x, y)$ given above with the gradient descent method, based on the analytical gradient that you found in Task 2. As a concrete example, run the program for values $a = 3$, $b = 100$, and the starting point $(0, 0)$. Also experiment with other values for a and b. Plot the function and the development of the optimization objective $f(x, y)$ and the parameters x and y over the iterations of your gradient descent algorithm.

Task 4 – Gradient of linear model for classification [15 points]

Assume a linear model without bias term for multiclass classification with classes $\{c_1, ..., c_k\}$. The model produces k class scores, $f_{\mathbf{W}} : \mathbb{R}^d \to \mathbb{R}^k$, and is given by

$$f_{\mathbf{W}}(\mathbf{x}) = \mathbf{W}\mathbf{x}$$

where $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{W} \in \mathbb{R}^{k\times d}$. Assume a data set $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ with labels $\{y_1, ..., y_n\}$. We want to train the model using the cross-entropy loss, which is given by

$$\ell(f_{\mathbf{W}}(\mathbf{x}_i), y_i) = -\log p(y = y_i | \mathbf{x}_i, \mathbf{W})$$

where

$$p(y = c_j|\mathbf{x}_i, \mathbf{W}) = \frac{\exp(f_{\mathbf{W}}(\mathbf{x}_i)_j)}{\sum_{l=1}^{k} \exp(f_{\mathbf{W}}(\mathbf{x}_i)_l)},$$

$f_{\mathbf{W}}(\mathbf{x}_i)_j$ denotes the j-th entry in the vector $f_{\mathbf{W}}(\mathbf{x}_i)$, and log denotes the natural logarithm.

Let $L(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \ell(f_{\mathbf{W}}(\mathbf{x}_i), y_i)$ denote the loss on all training instances. Let $\mathbf{w}_m \in \mathbb{R}^{1\times d}$ denote the m-th row of matrix $\mathbf{W}$ for $1 \leq m \leq k$. Show that the gradient of the overall loss with respect to $\mathbf{w}_m$ is given by

$$\frac{\partial L(\mathbf{W})}{\partial \mathbf{w}_m} = -\frac{1}{n}\sum_{i=1}^{n}\left(\delta_{y_i c_m} - \frac{\exp(\mathbf{w}_m \mathbf{x}_i)}{\sum_{l=1}^{k} \exp(\mathbf{w}_l \mathbf{x}_i)}\right)\mathbf{x}_i,$$

where

$$\delta_{y_i c_j} = \begin{cases} 1, & \text{if} \quad y_i = c_j \\ 0, & \text{if} \quad y_i \neq c_j \end{cases}.$$

Hint: as an intermediate step, use

$$L(\mathbf{W}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} \delta_{y_i c_j} \log\left(\frac{\exp(\mathbf{w}_j \mathbf{x}_i)}{\sum_{l=1}^{k} \exp(\mathbf{w}_l \mathbf{x}_i)}\right).$$