Exercises

Continuous Optimization

Prof. Dr. Peter Ochs

www.mop.uni-saarland.de/teaching/OPT24

— Summer Term 2024 —

— Assignment 7 —

Exercise 1. [20 points]

Consider the following function f with three input variables x_1, x_2, x_3:

$$f(x_1, x_2, x_3) = (\exp(x_1^2 x_2^3 + \sin(x_3^4)))^2 \,.$$

Your task is to express this function as a computational graph and to note down the operations at every node during the forward mode and backward mode of automatic differentiation for computing the derivative.

Exercise 2. [20 points]

In this exercise, we are given two sets of n points in $\mathbb{R}^2$ represented as columns of matrices $P, Q \in \mathbb{R}^{2\times n}$. We are interested in the "best" Euclidean transformation that maps the columns of P to the columns of Q, i.e., we assume that

$$Q \approx R(\theta)P + t1^\top \,,$$

where

$$R(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \quad \text{and} \quad t1^\top = \begin{pmatrix} t_1 & t_1 & \dots & t_1 \\ t_2 & t_2 & \dots & t_2 \end{pmatrix} \in \mathbb{R}^{2\times n}$$

for some $\theta, t_1, t_2 \in \mathbb{R}$. The goal of this exercise is to find parameters θ, t_1, t_2 that solve the following non-linear least squares problem:

$$\min_{t_1,t_2,\theta} \frac{1}{2}\|R(\theta)P + t1^\top - Q\|^2$$

where $\|C\|^2 := \sum_{i,j} C_{i,j}^2$.

(a) (*4 Points*) Rewrite the optimization problem in the following form

$$\min_{t_1,t_2,\theta} \frac{1}{2}\|\tilde{P}A(\theta, t_1, t_2) - \tilde{Q}\|^2 \,, \tag{1}$$

by defining the matrices $\tilde{P}$ and $\tilde{Q}$ in the right way, where the non-linear mapping A is given by

$$A\colon \mathbb{R}^3 \to \mathbb{R}^6 \,, \quad \begin{pmatrix} t_1 \\ t_2 \\ \theta \end{pmatrix} \mapsto \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ -\sin(\theta) \\ \cos(\theta) \\ t_1 \\ t_2 \end{pmatrix} \,.$$

(b) (*4 Points*) The formulation in (1) can be solved using the Gauss–Newton method, i.e. we iteratively solve linear least squares (sub-)problems. Derive the linear least squares problem that needs to be solved, given some current approximation $(\bar{t}_1, \bar{t}_2, \bar{\theta})$. Also, explicitly state the linear system of equations that arises as optimality condition of the linear least squares problem.

(c) (*6 + 6 = 12 Points*) Implement the Gauss–Newton Method and Levenberg-Marquardt algorithm in the code templates that are provided with this exercise.

- `data.py` contains a dictionary `data` with fields `P_`, `Q_` containing the $2 \times n$ matrices and the number of points `n`.
- `ex06_02_regression.py`: This file loads the data points and solves the non-linear least squares problem. The files has two sections, one for the implementation of Gauss-Newton method and other for the implementation of Levenberg-Marquardt algorithm.
- Section 1 in `ex06_02_regression.py`: Implement the Gauss-Newton method in this section. There are 3 `TODO`s:
  - The first one corresponds to part (a) of this exercise (note $\tilde{P}$ is `P` and $\tilde{Q}$ is `Q` in the file).
  - The second one to part (b): Set up the linear least squares problem in each iteration.
  - The third `TODO` is solving the linear system of equations (derived from the optimality condition of the linear least squares problem). Use `np.linalg.solve()` to solve the linear subproblems.
- Section 2 in `ex06_02_regression.py`: Implement the Levenberg-Marquardt algorithm in this section. The TODOs are similar to the Gauss-Newton method. In Levenberg-Marquardt algorithm, choose an appropriate value of α_k in order to solve the linear system of equations.
- For simplicity, set the step-size τ_k in the both the methods to 1.

Background:
Problems of this type appear, for example, in 3D reconstruction problems. The data points are 3D points that are reconstructed from different view points. In order to have all 3D points from different cameras in the same 3D coordinate system, the motion of the camera needs to be estimated. The ("extrinsic") camera parameters are given by a Euclidean transformation (Rotation + Translation) in the 3D space. Once the transformation between successive camera recordings are known, all data points can be "moved" into the same 3D coordinate system, which yields a 3D reconstruction. More advanced 3D reconstruction systems (e.g. Bundle Adjustment) try to also optimize the locations of the points; a problem that can also be solved using a Gauss–Newton method.

Submission Instructions: This assignment sheet comprises the theoretical and programming parts.

- **Theoretical Part:** Write down your solutions clearly on a paper, scan them and convert them into a file named *theory(Name).pdf* where Name denotes the name of the student submitting on behalf of the group. Take good care of the ordering of the pages in this file. You are also welcome to submit the solutions prepared with LaTeX or some digital handwriting tool. You must write the full names of all the students in your group on the top of first page.
- **Programming part:** Submit your solution for the programming exercise with the filename `ex06_02_regression_solution.py` where Name is the name of the student who submits the assignment on behalf of the group. You can only use python3.
- **Submission Folder:** Create a folder with the name *MatA_MatB_MatC* where MatA, MatB and MatC are the matriculation number (Matrikelnummer) of all the students in your group; depending on the number of people in the group. For example, if there are three students in a group with matriculation numbers 123456, 789012 and 345678 respectively, then the folder should be named: *123456_789012_345678*.

- **Submission:** Add all the relevant files to your submission folder and compress the folder into *123456_789012_345678.zip* file and upload it on the link provided on Moodle.
- **Deadline:** The submission deadline is 11.06.2024, 2:00 p.m. (always Tuesday 2 p.m.) via Moodle.