Exercises
Continuous Optimization

Prof. Dr. Peter Ochs

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

— Summer Term 2024 —

— Assignment 8 —

Exercise 1. [7 points]

Compute the normal and tangent cone to the following set

$$C := \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 : x_1,\, x_2 \geq 0,\, x_1 x_2 = 0 \right\},$$

at the point $(0,0)^\top \in C$, and also visualize the normal and tangent cone.

Exercise 2. [10 points]

Compute the normal and tangent cone to the following set

$$C := \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 : |x_1| + |x_2| \leq 1 \right\},$$

for any point $\bar{x} \in C$, and also visualize the normal and tangent cone.

Exercise 3. [8 points]

Denote the following

$$D := \left\{ \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \in \mathbb{R}^2 : z_1^2 + z_2^2 \leq 1 \right\},$$

$$C = \left\{ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2 : \begin{pmatrix} x_1^2 + 3x_2^2 + 2x_2 \\ 3x_1^4 + 2x_2^3 + x_1 + x_2 \end{pmatrix} \in D \right\}.$$

For any $\bar{x} := \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} \in \mathbb{R}^2$ satisfying $4\bar{x}_1(6\bar{x}_2^2 + 1) > (6\bar{x}_2 + 2)(12\bar{x}_1^3 + 1)$, compute the normal cone $N_C(\bar{x})$, for which full expressions must be written explicitly.

Exercise 4. [15 points]

We consider the problem of binary classification. Here, we are provided with input (training) data $(x_i, y_i)_{i=1,\ldots,m}$, where $x_i \in \mathbb{R}^{30}$ and $y_i \in \{0,1\}$ for all $i \in \{1,\ldots,m\}$. The goal is to construct a function (classifier) that maps an input $x \in \mathbb{R}^{30}$ to a decision $y \in \{0,1\}$. Ideally, the classifier function should predict the correct output y_i for input x_i, for all (x_i, y_i) in the training data, $i \in \{1,\ldots,m\}$, while being able to generalize to other new data, i.e., making correct predictions for points that are not contained in the training data.

Firstly, we require some definitions. The sigmoid function $\sigma : \mathbb{R} \to \mathbb{R}$ for $z \in \mathbb{R}$ is given by

$$\sigma(z) := \frac{1}{1 + \exp(-z)}.$$

We define a family of prediction functions $x \mapsto p(x; W_1, b_1, W_2, b_2)$ depending on parameters $W_1 \in \mathbb{R}^{10\times 30}, b_1 \in \mathbb{R}^{10}, W_2 \in \mathbb{R}^{1\times 10}, b_2 \in \mathbb{R}$, for the input $x \in \mathbb{R}^{30}$, given by

$$p(x; W_1, b_1, W_2, b_2) = \sigma(W_2\sigma(W_1 x + b_1) + b_2)\,,$$

where σ is applied element-wise, by which we mean that the function σ is applied to each element of a matrix or a vector individually. For input x, the prediction function p provides the probability for the output 1, and $1-p$ is the probability for the output 0. The final classification is obtained via the so-called classifier function $C : \mathbb{R}^{30} \to \{0, 1\}$, which thresholds the prediction function p, defined by fixed parameters $\bar{W}_1 \in \mathbb{R}^{10\times 30}$, $\bar{b}_1 \in \mathbb{R}^{10}$, $\bar{W}_2 \in \mathbb{R}^{1\times 10}$, $\bar{b}_2 \in \mathbb{R}$, as provided below

$$C(x) = \begin{cases} 0, & \text{if } p(x; \bar{W}_1, \bar{b}_1, \bar{W}_2, \bar{b}_2) \le 0.5\,, \\ 1, & \text{otherwise}\,. \end{cases}$$

In order to evaluate the correctness of the prediction for certain parameters (W_1, b_2, W_2, b_2) for the training data $(x_i, y_i)_{i=1,\dots,m}$, we define the following function $f : \mathbb{R}^{10\times 30} \times \mathbb{R}^{10} \times \mathbb{R}^{1\times 10} \times \mathbb{R} \to \mathbb{R}$, called loss function, by

$$f(W_1, b_1, W_2, b_2) = -\frac{1}{m}\sum_{i=1}^{m} \left(y_i \log(p(x_i; W_1, b_1, W_2, b_2)) + (1-y_i)\log(1 - p(x_i; W_1, b_1, W_2, b_2))\right)\,,$$

where $W_1 \in \mathbb{R}^{10\times 30}, b_1 \in \mathbb{R}^{10}, W_2 \in \mathbb{R}^{1\times 10}, b_2 \in \mathbb{R}, x \in \mathbb{R}^{30}$. The function f is known as cross-entropy loss, and it penalizes the deviation of the prediction $p(x_i; W_1, b_1, W_2, b_2)$ from the true output y_i for x_i, for all $i \in \{1, \dots, m\}$ simultaneously. The cross-entropy loss is used when the prediction is in $[0, 1]$ rather than $\{0, 1\}$. The quality of the constructed classifier for the training data is evaluated using the so-called classification error, defined as a function $E : \mathbb{R}^{30} \to [0, 1]$, which is given by

$$E(x) = \frac{1}{m}\sum_{i=1}^{m} I_i(x_i)\,, \quad \text{where } I_i(x_i) = \begin{cases} 1, & \text{if } C(x_i) \ne y_i, \\ 0, & \text{otherwise.} \end{cases}$$

In order to find the best prediction function p, in terms of optimal parameters W_1, b_1, W_2, b_2, your task is to solve the following optimization problem

$$\min_{W_1\in\mathbb{R}^{10\times 30},\, b_1\in\mathbb{R}^{10},\, W_2\in\mathbb{R}^{1\times 10},\, b_2\in\mathbb{R}} f(W_1, b_1, W_2, b_2)\,. \tag{1}$$

As an optimization algorithm, we use Gradient Descent algorithm with Backtracking Line Search method, by filling in the TODOs given in `ex08_neural_network.py`. The above given optimization problem in (1) is related to neural networks in the machine learning field, which we leave it to the reader for further exploration. The crucial task in the code is to obtain the gradient ∇f using the backward mode of automatic differentiation.

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 `ex08_neural_network_Solution(Name).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 18.06.2024, 2:00 p.m. (always Tuesday 2 p.m.) via Moodle.