Machine Learning A

2022-2023

Home Assignment 5

Christian Igel
Department of Computer Science
University of Copenhagen

The deadline for this assignment is **11 October 2022, 18:00**. You must submit your *individual* solution electronically via the Absalon home page.

A solution consists of:

- A PDF file with detailed answers to the questions, which may include graphs and tables if needed. Do *not* include your full source code in the PDF file, only selected lines if you are asked to do so.
- A .zip file with all your solution source code with comments about the major steps involved in each question (see below). Source code must be submitted in the original file format, not as PDF. The programming language of the course is Python.
- IMPORTANT: Do NOT zip the PDF file, since zipped files cannot be opened in speed grader. Zipped PDF submissions will not be graded.
- Your PDF report should be self-sufficient. I.e., it should be possible to grade it without opening the .zip file. We do not guarantee opening the .zip file when grading.
- Your code should be structured such that there is one main file (or one main file per question) that we can run to reproduce all the results presented in your report. This main file can, if you like, call other files with functions, classes, etc.
- Handwritten solutions will not be accepted, please use the provided latex template to write your report.

1 Principal Component Analysis (50 points)

Read section 9.2.1 in the online e-Chapter 9 of the textbook (Abu-Mostafa et al., 2012). The chapter can be downloaded from `http://book.caltech.edu/bookforum/showthread.php?t=4548`, the login is `bookreaders` and the password the first word on page 27 of the textbook. You can also find a scanned version of the section on Absalon.

PCA and preprocessing (0 points)

Not for submission: Solve and think about exercise 9.6 on page e-Chap:9–8.

1.1 PCA and centering (15 points)

Assume a $d \times d$ matrix $\boldsymbol{S}$. Recall centering (e.g., from page e-Chap:9–2). Proof that if we center the matrix, then the rank of the resulting matrix is at most $d - 1$.

1.2 Explained variance and Bessel's correction (5 points)

Consider the notebooks `PCA and explained variance using scikit-learn.ipynb`. It shines some light on the Scikit-Learn implementation of PCA and also contains a simple PCA implementation in NumPy.

It closes with „As expected, the explained variance does not depend on whether Bessel's correction[1] is used or not". Prove this statement.

1.3 PCA in practice

In this assignment, you are supposed to extend the notebook `PCA Assignment 5.ipynb`. Show the code you added to the notebook in the report as well as the plots and images you are asked to produce.

1.3.1 Explained variance (15 points)

Plot the explained variance as in the plot on the PCA lecture slides. That is, for each eigenvalue, sorted by magnitude, plot the eigenvalue divided by the sum of all

[1] If you do not know Bessel's correction, make a web search to learn about it (e.g., from Wikipedia).

eigenvalues (see `PCA and explained variance using scikit-learn.ipynb`). Use a log-scale for the y-axis.

Answer the following question: Are 10 principal components enough to explain 80% of the variance? You can directly see this if you plot the cumulative explained variance (e.g., by simply using `np.cumsum`) or by simply summing the explained variances of the first ten eigenvalues.

1.3.2 Eigendigits (15 points)

Plot the first 5 "eigendigits" similar to the "eigenfaces" in the lecture. You plot the first eigenvectors from the PCA as 2D images (e.g., using `.reshape(imshape)`, where `imshape` is defined in `PCA Assignment 5.ipynb`).

2 Logistic Regression in PyTorch (50 points)

PyTorch self-study (0 points)

We will use PyTorch in this course. If you have not worked with PyTorch before, please go through the introductory tutorial:

`https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html`

The content of this introduction is important for assignments.

In addition, work through the following basic programming section in the *Dive into Deep Learning* book Zhang et al. (2021), which will improve your general Python programming skills:

- 2.1 Data Manipulation (`https://d2l.ai/chapter_preliminaries/index.html`)
- 2.2 Data Preprocessing
- 2.3 Linear Algebra
- 2.4 Calculus
- 2.5 Automatic Differentiation
- 2.7 Documentation
- 11.5.1 Vectorization and Caches (`https://d2l.ai/chapter_optimization/minibatch-sgd.html#vectorization-and-caches`)

2.1 Logistic regression in PyTorch

You are supposed to fill in the missing lines in the notebook `Logistic Regression Assignment 5.ipynb`. The tutorial `https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html` should provide you with all knowledge you need.

The notebook considers an example where 2D data points are classified using logistic regression and the solution is visualized. First, an implementation in Scikit-Learn is given. Then the same is partly implemented using PyTorch. Your assignment is to fill in the blanks in the PyTorch implementation.

You should add some code in several cells (indicated by "MISSING"). Please write these lines of code in your report, see the LaTeX template.

References

Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin. *Learning from Data*. AMLbook, 2012.

A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. Dive into deep learning. *CoRR*, abs/2106.11342, 2021. URL `https://arxiv.org/abs/2106.11342`.