Exercise 1 – Language identification with sklearn and skorch

From Linear to Deep

Deadlines

Deadline for Exercise 1: **Sunday, 13.10.2024, 23:59 (Zurich Time)**.

Deadline for the peer review: **Monday, 21.10.2024, 23:59 (Zurich Time).**

- Note, the peer reviews will be assigned one day after the exercise deadline. You will find instructions for the peer review process at the end of this document.

Learning goals

This exercise consists of two parts: the first part aims at deepening your understanding of linear models. The second part will already target a simple kind of multi-layered network; the multilayer perceptron (MLP). Don't worry if you don't know anything about MLPs when reading. We will cover all you need to know to solve the second part of this exercise next week in class and in the tutorial session. The learning goals of this exercise are…

- to understand linear models and use them for multiclass classification tasks.
- to be able to implement different machine learning models, including MLPs, in scikit-learn and skorch.
- to understand the role of hyper-parameters and regularization.
- to perform an error analysis of machine learning models.
- to observe the most important features that lead to a prediction of a specific class and gain some insights into how the model makes its predictions
- to conduct an ablation study.

Please keep in mind that you can always consult us and use the exercise forum if you get stuck (note that we have a separate forum for the exercises).

Deliverables

Your solutions for each part should be submitted as self-contained iPython notebooks (ipynb files). The notebooks should contain your well-documented, EXECUTED, and EXECUTABLE code. That way, your reviewers can view and execute your code without potential dependency issues and/or installing new packages or versions of packages. We encourage you to solve the exercise on Google Colab and download the .ipynb file when everything is completed.

You will also have to write a short lab report at the end of each script.

The lab report should be written in a markdown cell and contain a detailed description of the approaches you used to solve this exercise. You can discuss what worked and what didn't work, but in both cases, you please provide reasoning as to why something did or didn't work. Please also

include the results. Inside each notebook, we there are specific questions denoted with “📝❓” that should be addressed in the report.
Please submit a zip folder named **ex01.zip** containing the following working ipynb notebook scripts, named as follows:

- **ex01_lr.ipynb**
- **ex01_nn.ipynb**

Please note:

- Organise your code such that it is executable.
- The cells must have already been run and the output must be visible to anyone checking your notebook without having to run the code again.
- Your assessors must be able to run your code. If it doesn't work, you can't get the maximum score.
- Please make sure that your answers to the specific questions outlined in the notebook and denoted with “📝❓” are clearly marked. If the assessors cannot easily find the answers they need, you can’t get the maximum score. We recommend including these clearly in your lab report.
- Also, we highly recommend testing that your notebook runs as expected before submitting it. To do this, hit “Runtime” > “Restart runtime and run all”, and make sure that there are no errors!

Data

For both parts of this exercise, you will work with the same data. The goal is to classify languages based on text snippets. This is a challenging extension of the problem described in Goldberg: chapter 2. However, we will work with more languages than just six and the text segments we need to classify are a bit shorter. In the notebooks provided as starting points, there are download links directly to the data, which will allow you to fetch the raw datasets.

Part 1 - Language identification with linear classification

To facilitate the start, you can use this iPython notebook in Google Colab which loads the files using the public links. We recommend making a copy of the notebook to build on. Simply hit the “Copy to Drive” button to do this.

The aim of part 1 is to become familiar with linear regression-based classification, as well as with the basic functionality of sklearn, a useful Python library for machine learning tasks.

Your task is to work through the notebook provided and implement the necessary code to:

1. Explore the data and create training/test splits for your experiments

2. Build a LogisticRegression classifier and design some relevant features to apply it to your data

3. Conduct hyperparameter tuning to find the optimal hyperparameters for your model

4. Explore your model’s predictions and conduct an error analysis to see where the model fails

5. Conduct an ablation study using a subset of languages

6. Conduct an analysis of most important features used by the model to make its predictions

Part 2 - Your first Neural Network

Now that you've got some experience under your belt, let's see if you can beat the best linear model you've trained with sklearn with a simple **neural network** using skorch.
Again, we provide you with this iPython notebook to get you started.

Your task is to work through the notebook provided and implement the necessary code to:

1. Build a simple neural network classifier using Pytorch

2. Train your neural network classifier with Skorch on a GPU using Google Colab.

3. Try to improve the model's accuracy above 87%.

Submission & Peer Review Guidelines:

Peer Reviews will be carried out on OLAT.
As soon as the deadline for handing in the exercise expires, you will have time to review the submissions of your peers. You need to do **2 reviews** per team to get the maximum points for this exercise.
Here are some additional rules:

- **All file submissions are anonymous (for peer review purposes): Do not write your name into the iPython notebooks, or the file names. Your reports at the end of each script should also be anonymous.**
- **ONLY ONE** member of each team submits on OLAT.
- Please submit a zip folder containing all the deliverables named according to the conventions described above.

Groups & Peer Reviews:

- You must create a group of **two** people to solve the exercise together. Each member should contribute equally!
- If you did not already work together for the previous exercise (or already submitted a post with the same team members), write a small post in the "Assignment Team Submission Thread" in the exercise forum on OLAT to notify the instructors about the group.
- Only **one team member submits the solutions.**
- Only the submitting team member will have access to the peer review, however, you should distribute the workload evenly!
- If you do not submit 2 reviews, we will deduct 0.25 points from your grade for this exercise.
- Please use full sentences when giving feedback.
- Be critical, helpful, and fair!
- Please answer all the review questions of the peer review.