Advanced Machine Learning - Sunspots
Midterm Project

November 2025

1 Instructions

- The entire project will be graded and will contribute 30% of the final grade.
- It is required to submit notebooks only, you should not write a final report. Nevertheless, the notebooks you submit should contain comments and explanations of your procedures.
- I will not grade you on the final quality of the classification tasks, but I will grade i) correctness of the Python implementation, ii) quality of the dataset visualization, iii) quality of the comments and findings, iv) scientific correctness of the performed steps.
- Please submit only one zip file containing all the scripts you wrote.
- Submit the zip file on iCorsi. The deadline is **November 16, 23:59**.
- You can talk about the task to your fellow students, but please do not share your implementation/code. Plagiarism in the source code or the report will be penalized with a grade of 0 and might be reported to the faculty.
- You are absolutely welcome to ask me anything about things that are unclear to you in this assignment. Feel free to contact me for feedback or suggestions.

2 Tasks

- If not specified, for evaluation of your results use the metric you think is more appropriate among those we saw in the lectures.
- The overall goal is to predict the number of monthly sunspots (https://en.wikipedia.org/wiki/Sunspot)

2.1 Task 1

Load the sunspot dataset from
https://raw.githubusercontent.com/jbrownlee/Datasets/master/monthly-sunspots.csv.

Each row of the dataset refers to a given month in a given year and contains the monthly mean total sunspot number. You can do plots and/or visualization of the dataset if you like, but it is not mandatory.

I suggest to scale the dataset with the `MinMaxScaler` between 0 and 1.

2.2 Task 2

Organize your dataset such that you have time series of a length you can decide. You want to predict the mean sunspots number, one month ahead.

Split the dataset into training+validation (80%) and test (20%).

2.3 Task 3

Build a simple Recurrent Neural Network and train it on the dataset you built. It should work pretty good. Make a plot that shows the real time series and the predicted one, highlighting where the predictions on the test set begin. Compare the performances with a simple Dense network.

2.4 Task 4

Perform again Task 3, but this time try to predict up to 6 steps ahead. We saw two different approaches in class, you should test both of them. Which one works better? Compare the performances with a simple Dense network.

2.5 Task 5

Now implement LSTM and/or GRU on Task 3 and 4.