Deep Learning Fundamentals

1st Semester - 2025/2026

Mini-Project Assignment – Exploring Hyperparameters in Neural Networks

Version 1.0 (2025-09-27)

- **Due date:** November 2nd at 23:59 (no late submissions)
- **Group work:** this assignment must be performed in pairs (two students per group).
- **Language:** your project may be written either in Portuguese or in English (preferred).

1. Introduction

In this mini-project, you will design, train, and evaluate an Artificial Neural Network (ANN) while systematically exploring the impact of different hyperparameters on training dynamics, generalization, and performance. The goal is not only to obtain good accuracy, but mainly to understand how hyperparameter choices influence the learning process.

2. Objectives

By completing this project, you should be able to:

- Identify the main hyperparameters of a neural network.
- Experiment with different hyperparameter settings and compare their effects.
- Interpret training/validation curves to detect underfitting, overfitting, and convergence issues.
- Produce a short-written analysis of your results.

3. Dataset and Model

You are free to choose a dataset among common benchmarks such as MNIST, Fashion-MNIST, CIFAR-10, or a small tabular dataset (e.g., from scikit-learn).

Your model must be an Artificial Neural Network (ANN). The architecture choice (depth, number of units, etc.) is left to you. You may experiment with simple or more complex designs.

4. Tasks

You should explore the following hyperparameters:

- Number of layers and units
- Activation functions (ReLU, Sigmoid, Tanh, others)
- Optimizers (SGD, Adam, RMSprop)

- Learning rate
- Batch size
- Number of epochs
- Weight initialization strategies (e.g., Xavier, He)
- Dropout rate
- Batch Normalization (with vs without)

For each experiment, keep track of the training and validation accuracy/loss curves. Compare the results and highlight the main differences observed.

5. Deliverables

Your submission should include:

- A Jupyter Notebook with your code and experiments.
- Plots of training and validation accuracy/loss for experiments.
- A short report in doc format discussing:
  - Which hyperparameters had the strongest effect on performance.
  - Which combinations led to overfitting or underfitting.
  - How regularization (dropout, batch norm, weight) affected generalization.
  - Your final choice of hyperparameters and justification.

Note: Writing this report does not replace the need for commenting on your code. You should add inline explanations, observations and reflections directly in the notebook, so that your reasoning process is clear during experiments.
If you prefer, you can also deliver your analysis (short report) in the notebook. But it should be presented in text, eventually with support images, in a text format.

6. Evaluation

Your work will be evaluated based on:

- Completeness of the experiments performed.
- Clarity of plots and results.
- Depth of analysis in the written report.
- Correctness and organization of the code.

7. Requirements and Submission Guidelines

Please note the following requirements for your submission:

- The code must be implemented using PyTorch Lightning. Yes, in class we developed using Keras, but the goal here is to gain versatility in multiple frameworks.
- The project must be carried out in groups of 2 students.
- Submission deadline: 2-Nov at 23:59.
- The project report may be written either in Portuguese or in English (English is preferred).
- Your notebook and report must be well organized and follow the good practices introduced in Module 1, including:
  - Use of clear sections and structure.

  - Integration of code with explanations and commentary.
  - Compliance with coding conventions and readability standards.
  - Ensuring reproducibility of experiments.
  - Proper handling of imports and installation of dependencies (if required).
- Ensuring reproducibility is always a good idea: fixed seeds, library requirements, clear instructions.
- Submit through Moodle: notebook, report and any other required files

8. Remarks

- You can copy code from the web, but
  - always provide the reference of the code that you copy
  - Make sure that you explain the used code so that another student of the course understands it and can use it
- During the development stage of your work, you should perform as many tests as you wish. However, you must submit a clean version of your notebook, containing only the relevant steps. In order to do so, before submitting you should copy your development notebook into a new file and clean it by deleting spurious tests and code.

- It is not always necessary to have a full-blown analysis, but you are expected to think! For each step, ask yourself why this step makes sense and then explain this in your notebook. Inline explanations in the notebook are required — your reasoning must be visible.

9. Submission Instructions

Your notebook should be submitted through the moodle platform. You can make several submissions, taking into account that a submission replaces the previous one.
If the data can be easily retrieved for the provided address, and was not subject of change during the assignment, just provide the link to the data in your assignment notebook. Otherwise, submit the data together with your work.

10. Policies

Students may share and/or exchange ideas with each other about the assignments and/or solving them. However, the work turned in must correspond to the individual effort of each group. The following situations are considered fraud:

- Partially copied work
- Facilitating copying by sharing files
- Using other people's material without mentioning the source.

In case of detection of any type of fraud, the work in question will not be evaluated, and will be sent to the Pedagogical Commission or the Pedagogical Council, depending on the gravity of the situation, which will decide the sanction to be applied to the students involved.
Plagiarism tools will be used for automatic copy detection.