Practical Work

Deep Learning

Informatics Engineering Master – Data Engineering
1st year 2nd semester
2023/2024

1. Objectives

General Objective:

- Develop, implement, and test Deep Learning models in practical problems from different application domains.

Specific Objectives:

- Define the work methodology.
- Select and apply techniques, methods, and concepts of Deep Learning.
- Implement and test Deep Learning models using Python.
- Evaluate and compare the performance of Deep Learning models.
- Writing a Technical Paper with Data Analysis.

2. Schedule

Release of work proposals: until April 2, 2024

Work submission: until **June 2, 2024** (23:55)

Defense and discussion: to be scheduled by the PL professor.

3. Rules

- The group (**maximum 3 members**) must be the same in both iterations of the Practical Work.
- Python should be used as the tool for data processing support.
- The deadline for **SUBMISSION** the Practical Work is **June 2, 2024**, in Moodle. Regardless of this deadline, groups should be able to report the development status of the Practical Work when requested by the teacher's.
- The Practical Work consists of a Scientific Paper.
- You should submit all documents in a compressed file. The zip file should contain:
    - the scientific paper in PDF format
    - datasets (pre-processed versions)
    - a complete (and commented) script of the code created in Jupyter Notebook to solve the proposed problem(s).
- The filename should follow the following notation: **APPROF_YYY_XXX_StudentNo1_StudentNo2_StudentNo3.zip**, where **YYY** represents the PL teacher's acronym, and **XXX** represents the PL class.

  Example: APPROF_DMB_M1A_7777777_8888888_9999999.zip.
- Works whose designation does not respect the indicated notation **will be penalized by 10%.**
- **The practical work must be submitted on Moodle by the defined deadline. Late submissions will not be accepted.**
- The defense and discussion of the practical work will take place on a day and time to be scheduled by each Laboratory classes teacher. On the day of the presentation, **ALL** teamwork members must be present. Absent members will not receive a grade. The defense and discussion will be conducted as a team with questions directed to each individual member.
- Each teamwork is responsible for managing its development process. Difficulties and problems should be communicated to the laboratory class teacher in a timely manner.
- Code of conduct: (cf. IPP Student Disciplinary Regulations)
    - No student or teamwork can claim ownership of work done by others or developed in collaboration with third-parties.
    - The use of materials, artifacts, or code from others without proper and explicit indication of origin is expressly prohibited.
    - Code from other sources must be clearly identified within the code itself, indicating the source.
    - Cases of unauthorized appropriation of materials, artifacts, and/or code subject to assessment will be reported to the ISEP Dean.
    - The use of AI coding/drawing assistance tools (e.g. ChatGPT) must be mentioned.

- The use of the Bitbucket version control tool is mandatory.

3.1. Scientific Paper

The scientific paper should document all phases of the followed work methodology, data preparation and exploration, analysis and discussion of results, and conclusions (maximum of 8 pages using the IEEE template provided on Moodle). Consider the following aspects:

- The paper should have an initial section with a summary of all used techniques and a final section with a summary of the main conclusions drawn from the resolution of the different questions identified in the statement.
- Each solved problem should have a brief explanation of the technique(s) used, the process of hyperparameters optimization, model validation, and a conclusion based on the interpretation and analysis of the obtained results. A synthesis of these conclusions should be included in the final conclusions section.

3.2. Evaluation

In the evaluation of the Practical Work assessment, the following aspects and weights specified in Table 1 will be considered:

a) Contextualization and objectives (Summary and Introduction)

b) Quality of Python code and its documentation

c) The quality of the model developed, its performance in solving the problem at hand, the techniques used and the different experiments made, including data analysis support

d) Organization, quality of writing, presentation, and clarity of the scientific paper

e) Defense and discussion

f) Individual participation of each group member

Table 1 – Practical work Evaluation Grid

Sumário	15%
Questão 1	35%
Questão 2	35%
Conclusão e referências	15%

Note: The grade of each group member will be determined based on their participation (in %). The practical work evaluation team will validate, at the time of the work defense (which may be through video conference), the participation of each group member in achieving the work and group objectives. **Absent members will not receive a grade.**

4. Work Description

With the execution of the Practical Work, students are expected to develop skills in the development of approaches based on Deep Learning techniques [1-3], culminating in the writing of a scientific paper with the definition of the problem, Deep Learning techniques used, test plan, hyperparameters, models and obtained results, analysis and discussion of results, and conclusions.

The primary objective is to foster skills in the development of Deep Learning models, using TensorFlow [4], capable of effectively addressing the requirements delineated within a predetermined problem dataset. Students are faced with two problems and are expected to solve, one using Convolutional Neural Networks (CNN) and the other employing Recurrent Neural Networks (RNN).

During the development of the Practical Work, the following set of considerations should be kept in mind and made visible:

- **Methodology**: The methodology for each problem must include different model configurations, including ***model.summary()* of the best architecture**. It is strongly recommended to include a diagram of the network architecture. Additionally, examples of the best-case scenarios (at least 3 examples) and worst-case scenarios (at least 3 examples) should be included in the paper.
- **Paper Content**: The paper must contain proper sources cited.
- **Pretrained Models**: If utilizing pretrained models, students must specify why the model was chosen. The final model cannot solely rely on the pretrained model unless a valid reason is provided.
- **Data Preprocessing**: Data preprocessing steps must be specified along with the reasons for each step. If any steps are omitted, the reason must also be explained.
- **Results**: Results obtained from experiments must be shown, including the number of experiments conducted, the corresponding results and model curves.
- **Model Configuration**: The rationale behind the choice of loss function, optimizer, and evaluation metrics must be specified.

4.1. CNN Model

For this problem set, students are tasked with developing a deep learning model aimed at classifying various species of birds based on images of the birds. The dataset contains images of different bird species [5].

Link: https://www.kaggle.com/datasets/veeralakrishna/200-bird-species-with-11788-images?select=segmentations.tgz

The objective of this problem is to achieve the highest possible model performance. It is expected that the accuracy for each species, on average, exceeds 80%. Furthermore, the work conducted for this problem set should address the following questions, ideally within the paper:

- What is the impact of different filters, strides, padding, and pooling methods on the model's performance?
- Illustrate the convolution result of a specific layer using example input images. What insights can be derived from it?
- Apply data augmentation and make conclusions regarding the model performance with and without those techniques.
- Analyze overfitting. If detected, specify, and apply a solution.
- Train with a pre-trained network (e.g. Resnet50) and compare the results with your model.

Evaluate the model results with learned techniques during the classes and make conclusions.

4.2. RNN Model

For this problem set, students are tasked with developing a deep learning model using Recurrent Neural Networks (RNNs) to classify various motion activities based on sensory data from motion sensors. The dataset [6], contains data on activities such as sitting, jogging, walking, standing, etc., captured as time series.

Link: https://www.kaggle.com/datasets/malekzadeh/motionsense-dataset?select=A_DeviceMotion_data

The objective of this problem is to explore the impact of different layers (e.g., LSTM and RNNs) and different time windows on the classification performance. Due to the significant number of output classes, only 3 should be chosen to develop models. It is expected that the accuracy for each class, on average, exceeds 80%. Furthermore, the work conducted for this problem set should address the following questions, ideally within the paper:

- How does the choice of RNN layers, such as LSTM and traditional RNNs, affect the classification performance?
- What is the impact of varying time windows on the model's ability to classify different motion activities?
- How do different combinations of RNN layers and time windows influence the model's overall accuracy and robustness?

Evaluate the model results with learned techniques during the classes and make conclusions.

4.3. Analysis and Discussion of Results

Provide a synthesis of the most important results and conclusions obtained in this work, justifying whenever necessary (conclusion).

5. Bibliographic References

[1]. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning, MIT Press, 2016 (https://www.deeplearningbook.org/).

[2]. Eugene Charniak, Introduction to Deep Learning, MIT Press, 2019.

[3]. F Chollet, Deep Learning with Python, Manning Publications, 2021.

[4]. Online resources and documentation from Tensorflow (https://www.tensorflow.org/overview)

[5]. Dataset 1, https://www.kaggle.com/datasets/veeralakrishna/200-bird-species-with-11788-images?select=segmentations.tgz

[6]. Dataset 2, https://www.kaggle.com/datasets/malekzadeh/motionsense-dataset?select=A_DeviceMotion_data