Deep Learning – Project A.Y. 2025/2026

Gianluigi Greco, Carlo Adornetto

UNIVERSITÀ DELLA CALABRIA

General Rules

- Groups of **2 or 3 students**
  - **The amount of work and results required to pass the exam will be proportional to the number of group components**
  - If you want to work alone, **first let us know**
  - I you want to work on your own idea project, **first let us know**
- Different models to implement for **6CFU** and **9CFU** students
- Each group **must submit**:
  - Report with a detailed description of the solution (.pdf) (latex or Word) (Max 10 pages)(See the structure at the last page of this document)
  - Presentation of the solution (.pptx or .pdf)
  - Your Code (.py or .ipynb) [python, notebook or colab files]
- **At the exam** you must present your solution in **5 minutes**:
  - Each member of the group has to contribute
  - At the end of the presentation you will run some inference on the test set
    (SAVE YOUR MODELS - Just prepare a notebook cell which load your model and some test data)

Submission

Submit via email to
carlo.adornetto@unical.it
(10 days before the exam)

- **OBJECT**: DEEP LEARNING - Project
- Attach to the email **ONE ONLY zipped file** containing all the required documents/code
- the zip must be named as follows:

  [surname1]_[surnamer2].zip

Roadmap

1. **Step 0:** Understand the context. What is the real objective and why are we doing it?
   What are the challanges of the use case?
   Experts of the application domain are available for quick chats!

2. **Step 1:** Import the data and check each steps for errors and inconsistences!
   Code provided by the teacher is just an example. Correctness is your responsability!

3. **Step 2:** Implement
   - A CNN(1D) and an RNN (for 6-CFU students)
   - An RNN and a Transformer (for 9-CFU students)
   - Train with different training approaches in the same train loop (e.g. 10 epochs (e) of teacher forcing + 10e masked modeling + 5e scheduled sampling)

   Note: Make it parametric as much as possible! Students with best results will have the opportunity to run huge models and long trainings on GPUs after the exam if they are interested in a tehsis or scientific publication!

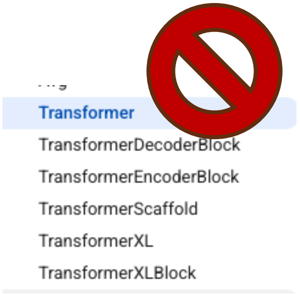
4. **Step 3:** For each model try at least 3 different configurations of Hyperparameters and at least 2 different time windows. 3 are for mandatory, if you want to do more do the best with the available resources you have.
   Show every result! (See example Table 1 iat the end) (cross-validation will be appriciated!)

5. **Step 4:** Evaluate and visualize your results as much as possible!
   - Evaluate metrics an variables separately to discover bieses and unbalancing!
   - Make nice plots (save them as jpg with at list 300dpi)

Rules

1. **No pretrained!**
   You can use them if you want but just for additional comparison with RNN, CNN or Transformers.

2. **Implement models** with only classes we have seen during labs.
   (e.g. do not use tf transformers from the nlp library)



3. **Reproducibility:** set all the random seeds. We cannot evaluate projects that doesn’t reproduce the same results when we run it.

4. **Plots** should be in .jpeg format and at least 300dpi.

5. **Evaluation of projects** will not concern the accuracy your models obtain!
   Indeed, we evaluate the methodology you use, the reasoning behind the decisions you make, consistency of the results and code, how do you present and show the results.

Suggestions

- **Start SMALL!** Start with small subset of data, sequences and small models!
  It is a step by step process. You don’t need to do all at once!
  Too long sequences? reduce them and use smaller time-windows!
  Too much variables? Apply dimensionality reduction (e.g. train an autoencoder and use smaller latent representations of the data points)

- **Explain Everything** you do and why!
  If you do experiments and they perform bad, report them and then explain why!
  Being able to show your work is as much as importat as the quality of what you do!

- **Do not go out of time** during presentation!
  Presentation which exceed the 5 min time will be penalized!
  During presentation focus on methodology (what Hp tuning, why, important decision you make and why), and results (show tables and results)

Projects

- DL-Quantum – Predict Dynamics for better chips! (quantum)
- Protein Dynamics for Drug Design. (protein)
- Ancient Swords (swords) (Data uploaded on request)
- A 3D medical model for STROKE Prediction (stroke) (to be defined)

[Link to the projects material]

Report structure

1. **Introduction**
   (explain the importance and why you are doing what you are doing)
2. **Dataset & Preprocessing**
3. **Models**
   i. *Imprementation*
      (anything important and non-trivial about the implementation)
      Code snippets are not useful. Use schemas and blocks!
   ii. *HP tuning and Hp configurations*
4. **Results**
   (structure it as you think is best)

- Max 10 pages
- Introduction and Dataset should be very brief
- Give more importance to preprocessing and all the rest
- As a suggetion: 1 page for introduction and dataset. 3 pages for preprocessing and models. All the rest for results
- Always comment figures and tables

Table 1.

This is an example on how you can show your different hyperparameter (Hp) configurations.
Show best results in bold! If you report a mean, alwys report std or (preferably) 95% confidence interval along side it.

	RNN			Tranformer		
	Hp config 1	Hp config 2	Hp config 3	Hp config 1	Hp config 2	Hp config 3
metric 1						
metric 2						
metric 3						
……..						
metric n						

Where for **RNN**:

- Hp config 1: learning rate=0.01, num_neurons=32, epochs=20, etc….
- Hp config 2: learning rate=0.001, num_neurons=64, epochs=20, etc….
- ………

Where for **Transformers**:

- Hp config 1: learning rate=0.01, num_heads=2, epochs=20, etc….
- Hp config 2: learning rate=0.001, num_heads=4, epochs=20, etc….
- ………

This is juat an example. Try your own Hp!