Probabilistic Methods for Machine Learning 2023/24

Final Project

General instructions

The exam consists in a project to be completed in groups of **3 students** or, for well-motivated reasons, less than 3. The project is the same throughout the entire academic year.

For each exam date (*appello*), there is a deadline (around 10-15 days earlier) to complete the project, submit the deliverables using a web form (once per group), and register on AlmaEsami (each group member). If the deliverables are deemed acceptable, an oral discussion about the project takes place on the exam date.

All students from the group must be present together at the oral discussion. Different students might obtain different final scores, based on the oral discussion and on the stated contributions. Each student must be able to discuss the work of the entire group, not just their own contributions.

Deliverables

- Submit a detailed PDF **report** of **up to 4 pages**, formatted using LaTeX. The report may be written in English or Italian, with a preference for English.

  The report must include a statement describing the responsibilities and contributions of each group member. Different amounts of contributions are acceptable, provided that every member has contributed in a significant way.

- Provide the **source code** developed to solve the project in the form of a shared Colab notebook. The notebook must be well-organized and commented, preferably in English. It **must be fully runnable** using a T4 GPU instance on Colab in less than 10 minutes. If your models required longer to train, upload the model parameters to Google Drive and include code to load them within the notebook.

Restrictions

- The data provided to you is exclusively for exam purposes and must not be disclosed publicly without permission.
- Collaboration between groups is prohibited until after all involved groups have passed the exam.
- The source code must be written by the group members, with the following exceptions: you can copy code provided by us as part of this course; you can copy code from the documentation of the Python libraries you use, as long as it does not constitute a substantial part of your submission.
- Your report and source code are your intellectual property. However, please do not make them public before the last exam session of this academic year (estimated February 2025). In particular, **do not** share the link to your Colab notebook outside of your group and the professors.

Project description

This project is about a variant of the game Connect Four (*Forza 4*). The rules of the standard game can be found in the first paragraph of the Wikipedia article. In our variant, the second player has two pieces already placed in the bottom-left and bottom-right corners of the board at the beginning of the game, as shown in Figure 1.

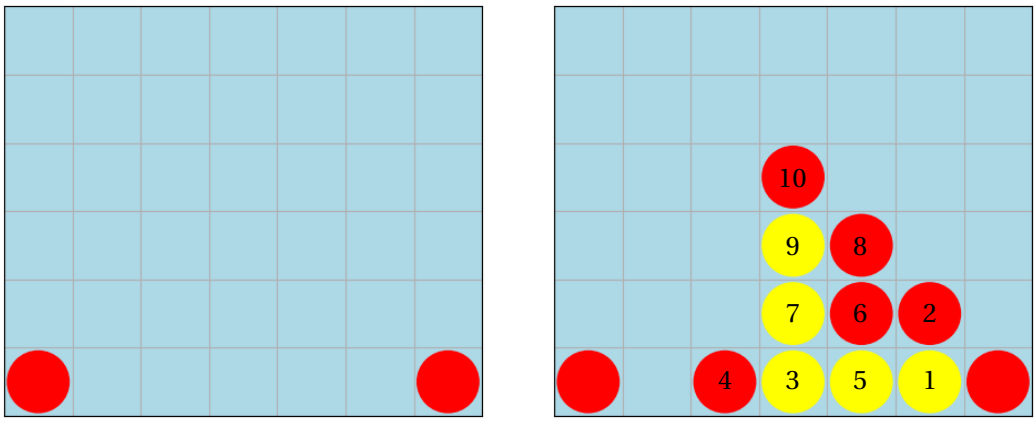

Figure 1: Our variant of Connect Four. **Left:** the initial position of the game, with two pieces belonging to the second player (red). **Right:** the final position of a valid game won by the second player (red) thanks to a diagonal line of four connected red pieces. The numbers indicate a valid sequence of moves leading to this final position.

This project has two tracks, and your group must choose **exactly one track**. Track 1 is easier than Track 2, but you can achieve the maximum grade (*30 e lode*) by selecting either track.

Track 1: Supervised Learning

For this track, we provide a supervised dataset of game positions and moves selected by an AI agent created by us (which plays the game decently, but not perfectly). Your task is to develop and train **a machine learning model that imitates the moves** performed by our agent; your task is **not** to develop a model that plays well!

You must develop, evaluate and compare **at least two substantially different approaches**.

Our dataset is split into a train set and a test set. You should use the test set as little as possible during development. At the very least, you must **report the accuracy on the test set**; additional evaluation/analysis of your model is encouraged. After you submit, we may evaluate your model(s) on a secret test set sampled from the same distribution as the dataset you have access to.

Track 2: Reinforcement Learning

In this track, your task is to **develop and train an AI agent that plays the game as well as you can**. We provide sample code to test your agent against a random agent, but you should come up with additional ways to evaluate your progress.

If your AI agent makes use of hard-coded rules, you should demonstrate that the machine learning component provides a significant improvement over the hard-coded rules alone.

In this track, you can optionally use the supervised dataset of Track 1 as a starting point; if you do so, clearly describe how you use the dataset and evaluate how much it contributes to your overall solution.

Using your agent, you should also attempt to answer at least one of the following questions: (1) Which player (if any) do you think has a winning strategy?[1] (2) What are the best opening moves?

Additional instructions for either track

This project can be solved in many different ways. You are free to choose your approach, as long as you motivate your choices with theoretical considerations as well as empirical results. Obviously, you must use machine learning in your solution, and you must develop and evaluate your solution in a rigorous way, as explained during the course.

It is ok to use machine learning approaches or architectures that were not covered during the course; if you do so, provide references to them. However, you should only use the data science / machine learning libraries encountered during the course (for instance, you must use PyTorch for Deep Learning models).

You must not search for additional data to solve this project, beyond the supervised dataset we provide. It is allowed, however, to collect data within your group and/or manually evaluate your models. If you do so, describe the followed procedure in the report.

The report should include: details on how you selected your model(s) and hyperparameters; plots and figures; details on evaluation; any findings.

Evaluation criteria

- Common to both tracks:
  - Effectiveness of the chosen approaches and models. The choices should be theoretically and/or empirically motivated.
  - Efficiency and soundness of the training procedure.
  - Neatness of the code and ease of running the notebook.
  - Clarity, thoroughness, and conciseness of the final report.
  - Original thinking will be appreciated, though developing an original approach is not necessary to achieve the maximum grade.
- Track 1:
  - Accuracy of the best model. This includes the soundness and overall coherence of the testing procedure.
  - Coherence between your reported test accuracy and the accuracy on the secret test set (computed by us). If your reported test accuracy is substantially higher than the accuracy on the secret test set, it means that you overfitted the test set.
  - To access the oral exam, your best model must have a **test accuracy of at least 35%**.
- Track 2:
  - Soundness of the testing procedure and evaluated strength of the AI agent.

[1] Since this is a finite game with perfect information (like Chess and Go), one of the two player has a winning strategy, or both players can force a draw.

  - Scientific rigor in answering the question(s) on winning player and opening moves.
  - To access the oral exam, your agent must at least be able to **consistently defeat the random agent** provided by us.

What we provide

- The supervised dataset for Track 1 (links: train set; test set).
- A sample Colab notebook (link) including: an implementation of the game rules; code to load the supervised dataset; an AI agent that plays almost randomly; a function to test your AI agent against the random agent.

Optional tournament (not part of the exam)

We tentatively plan to organize a tournament for all groups that choose Track 2. In this tournament, the developed AI agents will compete against each other and the winning group will be awarded a symbolic prize. The tournament will take place around February 2025 at the Department of Mathematics.