Final Projects for the RL course

...

Some guidelines

When will the exams be?

QLS diploma students:

- July 1st, individual projects

  All exams will be in presence. **Define your project by June 17 at the latest**

  QLS students must attend all presentations

UNITS Master students:

- Whenever you are ready. First opportunity, July 1st.

  Group projects are possible

  Write us in advance (min. 10 days). Exams can be in presence or online

Two main flavours of projects (A degree of hands-on is always recommended).

I - Hands-on focus

What: Construct the solution to a specific problem/game using RL.

Why: Understand the different approaches to a problem, discuss problems and solutions.

II - Theory focus (harder!)

What: Delve deeper into one particular subject

Why: Explore a theme outside the class material (but with some connection to it)

- All projects -

The exam can cover also theory questions both on the project *and* on the rest of the program

How to choose the subject of the project

Choose something that interests you

It can be an *extension of the theory* seen in class.

It can be an algorithm used on a *different problem*.

Once you have an idea (and especially if you *do not* have any ideas):

- **write to us, so that we can give the project some "boundaries"**

What should you produce at the end of the project?

1) Presentation of 20/30 minutes

2) GitHub repositories / Jupyter notebook with code

What is the aim of the hands-on side of a project?

Show that you fully understand the process of applying RL methods to a problem.

- How to "translate" a problem in a formal setting (generic problem to MDP...)
- How to tackle the problem (methods/algorithm...)
- How to overcome issues (simplify the problem / use learning tricks...)

What is *not* its aim?

Spending weeks on the coding part of the problem at the expense of understanding

- Avoid choosing too complex problems
- Avoid solutions which which require huge amounts of data/GPUs

Results: How to present

i) **Describe and motivate** the problem.

ii) How did you **translate** the problem **into a RL framework**?

iii) **How** did you (try to) solve it? **Why** did you use one algorithm or another?

iv) Is there **something else** could you have done?

v) Did it work? (**Can you interpret** the optimal policy?)

vi) Are the results consistent with your expectations? Can you explain them?

What is the aim of the theory part of a project?

Show that you can navigate the literature in order to build on your knowledge and learn a new method, and get a deeper understanding of something treated in class.

Examples:
- a chapter of the Sutton and Barto's book that was not covered
- a deep dive on Actor-Critic methods
- applications of RL in a specific sector (advertisement, medical science...)

Success stories:

https://github.com/lorenzobasile/RLProject

https://github.com/mariagraziaberni/Reinforcement_Learning

Can you use AI?

Yes, but...

- you **must** declare it explicitly.
  (this part of the code was done by, the text / graphs / analysis, ... , the idea ...)

- ChatGPT is ***not*** a good scientist:
  . use with caution
  . double-check everything
  . question everything

- ChatGPT ***may be*** good at:
  . translating code from one language to another
  . code faster **(assuming you already know what it is)**

Probability ChatGPT (or related) makes your project better: **low**

Best/Worst practice

- **DO** share the code with clear annotations
- **DO** show your reasoning in the presentation
- **DO** question your results
- **DO** show final policies vs only learning curves
- **DON'T** talk only about code
- **DON'T** present learning curves without "target score"
- **DON'T** say/write things you don't understand

Can we work in groups?

Yes (for Master students)

Note that:

- We expect that the “project size” scales with “group size”
- Presentation is collective
- Make sure that individual contributions are clearly defined