Reinforcement Learning (Spring 2025)

Description: This course describes theory and methods for Reinforcement Learning (RL), which revolves around decision making under uncertainty. The course covers classic algorithms in RL as well as recent algorithms under the lens of contemporary optimization. The group project enables the students to familiarize with the state-of-the-art RL modeling techniques and algorithms. ⋆ you can let us know what you expect from this course by filling this form: `https://go.epfl.ch/rl-expectations-2025`.

Learning outcomes: By the end of the course, the students are expected to understand the core challenges (like the exploration-exploitation tradeoff, sample complexity etc.) in RL. In particular, students must be able to:

1. Define the key features of RL that distinguishes it from standard machine learning.
2. Understand strengths, limitations and theoretical properties of RL algorithms.
3. Recognize the common, connecting boundary of optimization and RL.
4. Formulate and solve sequential decision-making problems by applying relevant RL tools.

Prerequisites: Previous coursework in optimization, machine learning, probability theory, and linear algebra is required. Familiarity with deep learning and programming in python is useful.

Language: English

Class Times: Thursdays 13:15-17:00 in AAC231 and CM 1 4 (Overflow room)
Except:

- 13th March 2025 (week 11)
- 10th April 2025 (week 15)
- 17th April 2025 (week 16)

when the overflow room (CM 1 4) is not available.

Lab & office hours: By appointment.

Instructor: Prof. Volkan Cevher, ELE 233, volkan.cevher@epfl.ch

Head TAs: Luca Viano, ELD 243, luca.viano@epfl.ch
Elias Abad Rocamora, ELD 243, elias.abadrocamora@epfl.ch

Credits: 6

Course Moodle: `https://go.epfl.ch/rl-moodle`

Resources: We will provide corresponding reading resources during lectures.

Assessment Methods: The students are required to present a lecture for a class and do a group project. The guidelines on the project are provided separately.

Course Outline[1]

Lecture 1: **Introduction to Reinforcement Learning + Dynamic Programming I**
Definition of Markov Decision Processes, policy and performance criteria.
Dynamic programming with known transition dynamics: Value Iteration, Policy Iteration.

Lecture 2: **Dynamic Programming II**
Dynamic programming with unknown transition dynamics: Q-Learning.

Lecture 3: **Linear Programming**
Algorithms based on Primal and Dual Linear Programming formulation of RL: constraint sampling, REPS and DICE methods.

Lecture 4: **Policy Gradient I**
Policy Parameterization, REINFORCE and techniques to compute unbiased estimator of the policy gradient.

Lecture 5: **Policy Gradient II**
Non concavity of the policy gradient objective, global convergence of projected gradient descent, Global convergence of natural policy gradient, TRPO and PPO.

Lecture 6: **Deep and Robust Reinforcement Learning**
Importance of robustness in RL, Robust RL as a Zero Sum Markov Game.

Lecture 7: **Imitation Learning**
Motivations, Setting, maximum causal entropy IRL, GAIL and LP approaches.

Lecture 8: **Alignment and Reasoning with Reinforcement Learning**
Small intro to Language Models, Alignment, RLHF, Reasoning, Reasoning in modern models (GPT-o1, DeepSeek-R1).

Final Lecture: **Project Presentations**

[1] Each lecture is 2/3h. After Lecture 8, the lecture time onwards is reserved for performing the course projects.

Class Project Guidelines

Project ideas: We will provide a list of EPFL labs that you are allowed to contact regarding course projects, and you will then discuss the project proposal and supervision with them individually. We will also release a list of project ideas from our lab. We believe that those ideas may lead to publication in top conferences (e.g., NeurIPS whose deadline will be around mid-May).

Group: You will work in groups of three people. We also ask for a statement explaining the role of each group member along with the final report. Only one person should submit the project documents. Group members will typically (but not necessarily) get the same grade.

Options We provide two possible project options:

- **Practice**: Either implementing existing algorithms in new environments or try to improve existing algorithms on common environments.
- **Theory**: Read 3 theory papers in an active RL research area (we will provide pointers). Summarize them, understand which problems are still open (maybe solve them).

Timeline: Final report and poster deadlines are strict:

15th May 2024 11:59 PM Poster is due

30th May 2024 11:59 PM Final report is due

22th May 2024 Class Period Poster presentations

Final Report: We expect a 6-8 pages report using the NeurIPS template. Your report should follow the general format of a scholarly paper in this area. The following is a suggested structure:

1. The title, and Author(s)
2. Abstract
3. Introduction
4. Background/Related Work
5. Approach
6. Results
7. Conclusion
8. References

For RL experiments and presentation of results:

1. Submit your code (with a detailed README file) as a single project.zip file, or include a GitHub link in your report. You may use any existing code, libraries, etc. However, you must cite your sources in your report and clearly indicate your contributions.
2. For theoretical results, provide detailed proofs.

Presentation: Projects will be presented during a poster session at the class time during the last week of the semester.

Grading: Grade allocation is as follows:

1. Attendance: 1 point
2. Jupyter Notebook 1: 1 point
3. Jupyter Notebook 2: 1 point
4. Jupyter Notebook 3: 1 point
5. Project or Scribe: 2 points

Scribe: Only PhD students may replace the project by writing lecture notes for a lecture we assign them to (no group work for this option). The Scribe must be written in a specific template we will provide.
Project: 0.5 points for the Poster presentation and 1.5 for the Report.
Notebooks: 0.75 points for questions where the TAs can help. 0.25 points for $\star$ questions where the TAs cannot help.