Theory and Methods for Reinforcement Learning (Spring 2023)

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_form2023`.
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 09:15-12:00 in INM11.
Lab & office hours:	By appointment.
Instructor:	Prof. Volkan Cevher, ELE 233, volkan.cevher@epfl.ch
Teaching assistants:	Luca Viano, ELD 243, luca.viano@epfl.ch Angeliki Kamoutsi, ELD 243, angeliki.kamoutsi@epfl.ch
Credits:	3
Course Website:	`https://moodle.epfl.ch/course/view.php?id=15887`
Resources:	We will provide corresponding reading resources during lectures.
Honor Code:	The EPFL honor code applies to the course: `http://wiki.epfl.ch/delegues/code.honneur`.
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.**
Definition of Markov Decision Processes, policy and performance criteria.

Lecture 2: **Dynamic Programming**
Dynamic programming with known and unknown transition dynamics: Value Iteration, Policy Iteration, 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 1**
Policy Parameterization, REINFORCE and techniques to compute unbiased estimator of the policy gradient.

Lecture 5: **Policy Gradient 2**
Non concavity of the policy gradient objective, global convergence of projected gradient descent, Global convergence of natural policy gradient, TRPO and PPO.

Lecture 6: **Imitation Learning**
Motivations, Setting, maximum causal entropy IRL, GAIL and LP approaches.

Lecture 7: **Markov Games**
Motivations, Setting, different notions of equilibria, Policy Gradient algorithms for Zero Sum Games.

Lecture 8: **Deep and Robust Reinforcement Learning**
Importance of robustness in RL, Robust RL as a Zero Sum Markov Game.

Final Lecture: **Project Presentations**

[1]Each lecture is 3h in our final edition of this PhD level course. Next year, it will be offered at the MSc level. After Lecture 8, the lecture time onwards is reserved for performing the course projects.

Class Project Guidelines

Project ideas: We welcome ideas related to students current research topic. However, we will also release a list of project ideas that match the lab expertises. We believe that those ideas may lead to publication in top conferences (e.g., NeurIPS whose deadline will be around mid-May).

Group: You may work in groups of up to three people. The expectations for the project scope will scale with the group size. 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.

Timeline: Students that chooses a project proposed by the lab are welcome to start working on it from week 1. Otherwise, please submit your project proposal before the deadline below. Final report deadline is strict and common to both options:

30 March 09:15 AM Project Proposal

1 June 11:59 PM Final Report

1 June Class Period Presentations

Project Proposal: A brief description of the project (1-2 page) which includes the following:

1. the names of the project team members
2. summary of the project and its importance
3. a reading list and directions to be explored
4. special computational resource requirements or licensing requirements (e.g., MuJoCo)

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, we expect you to follow the recommended best practices [1]. Also include the following supplementary materials:

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, you need to provide detailed proofs.

Presentation: Projects will be presented in class – about 15 minutes per project.

Failure Event: When the project does not work as expected, you need to carefully justify the failure. Ensure that you get periodic feedback from us.

Grading: Grade allocation is as follows:

1. Attendance (minimum of 4 lectures): 1 point
2. Lecture Presentation: 2 point
3. Class project: 3 points

References

[1] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.