Skoltech
Skolkovo Institute of Science and Technology

Course Syllabus

Course Title (in English)	Reinforcement Learning
Course Title (in Russian)	Обучение с подкреплением
Lead Instructor(s)	Osinenko, Pavel

Contact Person	Pavel Osinenko
Contact Person's E-mail	p.osinenko@skoltech.ru

Course Description

Reinforcement learning (RL) is a vanguard method of machine learning aimed at dynamical applications, ranging from video games to autonomous cars, robots, drones etc. Composed of an agent and an environment, it is meant to resemble the behavior of living beings somewhat. RL is truly an interdisciplinary subject that can be studied from different kinds of perspectives - machine learning, control theory, dynamical system theory, pure math (e.g. approximation theory) etc.

In this course, we dive into RL with the goal of understanding and trying out the key principles thereof. We will study how agents interact with the environment and optimize their actions to improve rewards. Speaking of examples, imagine a video game speed run. An agent, the protagonist, interacts with the game environment and wants to beat the game as fast as possible, by dynamically adjusting his or her controls while learning on-the-fly.

Topics covered include policy gradient methods, actor-critic, deep RL, predictive RL and more. Convergence, safety and stability of RL are studied.

Course Prerequisites / Recommendations	Although necessary recalls will be made, it is expected that the students be knowledgeable in basics of differential equations, probability theory, optimization. Awareness of dynamical systems is beneficial.

Аннотация

В данном курсе изучается один из авангардных методов машинного обучения -- обучение с подкреплением. Этот метод основан на концепции агента взаимодействующего со средой и оптимизирующего свои действия с целью достижения максимальной "выгоды" (в абстрактном смысле, который конкретизируется в приложениях).

Следующие темы адресованы в курсе:
- динамические среды;
- динамическое программирование;
- градиентные методы оптимизации политик;
- актор-критик;
- сходимость, безопасность и устойчивость обучения с подкреплением;
- глубокое и предиктивное обучение с подкреплением.

Course Academic Level	Master-level course suitable for PhD students
Number of ECTS credits	6

Topic	Summary of Topic	Lectures (# of hours)	Seminars (# of hours)	Labs (# of hours)
Environments	Environments as dynamical systems; discrete-time systems (deterministic, Markov decision processes); continuous-time systems; stochastic differential equations; control; optimal control; cost-to-go; RL as optimal control	3	3	
Dynamic programming	Recursion on cost-to-go; derivation of the dynamic programming principle; Counterexample; Fixed-point equations; Hamilton-Jacobi-Bellman equations; Value, policy iteration; Q-learning; Curse of dimensionality; Origin of RL	3	3	6
Policy gradient	REINFORCE; vanilla policy gradient; natural policy gradient	3	3	6
Actor-critic	Actor and critic; on-policy and off-policy methods; TD(N) and TD(lambda); advantage function: neural dynamic programming; model-free vs. model-based; online vs. offline; A2C and A3C	3	3	6
Convergence of DP-like RL	Convergence of discounted DP; contraction; admissibility; linear case: linear quadratic regulator, linear Gaussian regulator	3	3	0

Convergence of actor-critic RL	Asymptotic convergence vs. uniform ultimate boundedness; abstract energy; convergence of actor-critic neural network weights; issues of convergence; persistence of excitation; exploration vs. exploitation;	3	3	0
Deep RL	Deep actor-critic; advanced NN architectures in RL; non-convexity issues; convergence of nonlinearly parametrized models;	3	3	6
Predictive RL	Use of models; model-predictive control vs. RL; convergence of MPC; fusion of MPC and RL, rollout RL; RL dreamer	3	3	0
Safety and stability of RL	Human-expert; Formal verification, shields; MPC+RL for safety and stability; stability under MPC; Lyapunov-based RL	3	3	0

Assignment Type	Assignment Summary
Homework Assignments	Implement and test a DP-like RL agent
Homework Assignments	Implement and test a policy gradient RL agent
Homework Assignments	Implement and test an A2C RL agent
Homework Assignments	Implement and test a deep RL agent
Final Project	Pick a research work; study it; select a case study; implement algorithms; test; organize project on Github; report: a presentation, no detailed report needed; evaluation: based on presentation and Github repo; depending on the student number, the final project may be assigned in groups

Type of Assessment Graded

Grade Structure

Activity Type	Activity weight, %
Homework Assignments	70
Final Project	30

A: 86

B: 76

C:	66
D:	56
E:	46
F:	0
Attendance Requirements	Mandatory with Exceptions

Maximum Number of Students

	Maximum Number of Students
Overall:	150
Per Group (for seminars and labs):	5

Course Stream	Science, Technology and Engineering (STE)
Course Term (in context of Academic Year)	Term 1
Course Delivery Frequency	Every year

Students of Which Programs do You Recommend to Consider this Course as an Elective?

Masters Programs	PhD Programs
Advanced Computational Science Advanced Manufacturing Technologies Data Science Energy Systems Information Science and Technology Internet of Things and Wireless Technologies Life Sciences Materials Science Mathematical and Theoretical Physics Petroleum Engineering Photonics and Quantum Materials Space and Engineering Systems All Master Programs	Computational and Data Science and Engineering Engineering Systems All PhD Programs

Course Tags	Math Physics Programming Engineering

Required Textbooks	ISBN-13 (or ISBN-10)
R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. MIT press, 2018	9780262039246

Recommended Textbooks	ISBN-13 (or ISBN-10)
Bertsekas, Dimitri P. Reinforcement learning and optimal control. Belmont, MA: Athena Scientific, 2019	978-1886529397
M. L. Puterman. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons, 2014	978-0-471-72782-8

Papers	DOI or URL
Bertsekas, D. and Tsitsiklis, J. "Neuro-dynamic programming: an overview." Proceedings of 1995 34th IEEE Conference on Decision and Control. Vol. 1. IEEE, 1995	10.1109/CDC.1995.478953
Werbos, P., Miller, W., and Sutton, R. "A menu of designs for reinforcement learning over time." Neural networks for control (1990): 67-95	
Sokolov, Y. et al. "Complete stability analysis of a heuristic approximate dynamic programming control design." Automatica 59 (2015): 9-18	10.1016/j.automatica.2015.06.001
Lewis, F. and Vrabie, D. "Reinforcement learning and adaptive dynamic programming for feedback control." IEEE circuits and systems magazine 9.3 (2009): 32-50	10.1109/MCAS.2009.933854
Fairbank, M., and Alonso, E.. "The divergence of reinforcement learning algorithms with value-iteration and function approximation." The 2012 International Joint Conference on Neural Networks (IJCNN). IEEE, 2012	10.1109/IJCNN.2012.6252792

Web-resources (links)	Description
https://pathmind.com/wiki/deep-reinforcement-learning	Nice and short description of RL with graphical material

Equipment
No specific equipment is involved per se, but may be introduced depending on the situation, e.g., for final projects.

Software
Python

Labs for Education	Computer Lab

Knowledge
Reinforcement learning essentials
Derivation of reinforcement learning methods; a menu of methods
Advanced topics in reinforcement learning (convergence, stability etc.)

Skill
Design of reinforcement learning agents (DP-like, policy gradient, A2C, deep, predictive etc.)
Convergence analysis of reinforcement learning
Stability and safety analysis of reinforcement learning

Experience
Implementation of reinforcement learning methods in Python, incl. environments and agents

Select Assignment 1 Type	Homework Assignments
Input Example(s) of Assignment 1 (preferable)	Implement and test a DP-like RL agent
Assessment Criteria for Assignment 1	Answers are autograded based on metrics specified in the homework assignment based on mathematical functions related to respective RL methods
Select Assignment 2 Type	Final Project
Input Example(s) of Assignment 2 (preferable)	Pick a research work; study it; select a case study; implement algorithms; test; organize project on Github; report: a presentation, no detailed report needed; evaluation: based on presentation and Github repo; depending on the student number, the final project may be assigned in groups
Assessment Criteria for Assignment 2	The report highlights the necessary components and the description is correct. Presentation is concise and clear, incl. the part about the Github repo