Theory and Methods for Reinforcement Learning (Spring 2020)

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 (Monte-Carlo methods, TD-Learning etc.) as well as recent algorithms (TRPO, DDPG, SAC etc.) based on the exploration-exploitation trade-offs. The group project enables the students to familiarize with the implementation of some of the state-of-the-art RL algorithms.
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. Given a relevant application problem, formulate it as an RL problem, and identify best-suited algorithms to solve it. 3. Implement recently published articles on RL to solve standard control tasks (e.g., MuJoCo environment). 4. Understand the techniques to address the core challenges in RL.
Prerequisites:	Previous coursework in optimization, probability theory, and linear algebra is required. Familiarity with deep learning and programming in python is useful.
Language:	English
Class Times:	Thursdays 10:15-12:00 in CM1113.
Lab & office hours:	Thursdays 9:15-10:00 in CM1113.
Instructor:	Prof. Volkan Cevher, ELE 233, volkan.cevher@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 and do a group project. The guidelines on the project are provided separately.

Course Outline

Lecture 1: Introduction to Reinforcement Learning.
Reading: Chapter 3 of [38]

Lecture 2: Dynamic Programming. (Student Lecture)
Reading: Chapter 4 of [38]

Lecture 3: Monte Carlo Methods. (Student Lecture)
Reading: Chapter 5 of [38]

Lecture 4: Temporal-Difference Learning. (Student Lecture)
Reading: Chapter 6 of [38]

Lecture 5: n-step Bootstrapping. (Student Lecture)
Reading: Chapter 7 of [38]

Lecture 6: Value-based Methods for Deep RL. (Student Lecture)
Reading: [25, 45, 43, 33, 14]

Lecture 7: Policy Gradient Methods for Deep RL I. (Student Lecture)
Reading: Chapter 13 of [38], and papers [39, 19, 34, 35, 36]

Lecture 8: Policy Gradient Methods for Deep RL II. (Student Lecture)
Reading: Chapter 13 of [38], and papers [39, 19, 34, 35, 36]

Lecture 9: Actor-Critic Methods for Deep RL. (Student Lecture)
Reading: [22, 9, 11, 12]

Lecture 10: Model-based RL. (Student Lecture)
Reading: Chapter 8 of [38]

Lecture 11: Deep Model-based RL. (Student Lecture)
Reading: [24, 10, 2]

Lecture 12: Inverse Reinforcement Learning. (Student Lecture)
Reading: [28, 27, 1, 31, 47, 46, 40, 17, 16]

Lecture 13: Robust Reinforcement Learning.
Reading: [29, 41, 23]

For each student lecture, we assign a presenter and two questioners from the enrolled students pool. We will provide the lecture materials (including source files) to the assigned students. Students could improve the materials as well.

Class Project Guidelines

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: Note that the following deadlines are strict:

13 March 11:59 PM Project Proposal

29 May 11:59 PM Final Report

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. Theoretical results (if relevant)
7. Experiment results (if relevant)
8. Conclusion
9. References

For RL experiments and presentation of results, we expect you to follow the recommended best practices [13]. 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.

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: 1 point
2. Participation as questioner: 1 point
3. Participation as student lecture presenter: 2 points
4. Class project: 2 points

Project ideas: Students are encouraged to come up with their own idea. Below we provide some sample projects:

1. Attempt to make progress on a fundamental problem in RL:
   (a) Safety constraints in RL [3, 7, 18].
   (b) Improving the sample-complexity of RL via expert demonstrations [15, 37, 21].
2. Applications of RL – review how RL algorithms have been applied to a specific domain of interest, and extend it further:
   (a) Wireless Communication [6, 8].
   (b) Neural Architecture Search [48, 5, 30].
   (c) Combinatorial Optimization [4, 20, 26].
3. RL for games – familiarize with the game platform and implement deep RL algorithms in the game and test the performance:
   (a) Starcraft [44, 32].
   (b) AlphaZero [42].

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[2] David Abel, Dilip Arumugam, Kavosh Asadi, Yuu Jinnai, Michael L Littman, and Lawson LS Wong. State abstraction as compression in apprenticeship learning. In *Proceedings of the AAAI Conference on Artificial Intelligence. AAAI Press*, 2019.

[3] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.

[4] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

[5] Irwan Bello, Barret Zoph, Vijay Vasudevan, and Quoc V. Le. Neural optimizer search with reinforcement learning. In *International Conference on Machine Learning*, pages 459–468, 2017.

[6] Sandeep Chinchali, Pan Hu, Tianshu Chu, Manu Sharma, Manu Bansal, Rakesh Misra, Marco Pavone, and Sachin Katti. Cellular network traffic scheduling with deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[7] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8103–8112, 2018.

[8] Colin de Vrieze, Shane Barratt, Daniel Tsai, and Anant Sahai. Cooperative multi-agent reinforcement learning for low-level wireless communication. *arXiv preprint arXiv:1801.04541*, 2018.

[9] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1582–1591, 2018.

[10] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. *arXiv preprint arXiv:1906.02736*, 2019.

[11] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865, 2018.

[12] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[13] 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.

[14] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[17] Jonathan Ho, Jayesh Gupta, and Stefano Ermon. Model-free imitation learning with policy optimization. In *International Conference on Machine Learning*, pages 2760–2769, 2016.

[18] Jessie Huang, Fa Wu, Doina Precup, and Yang Cai. Learning safe policies with expert guidance. In *Advances in Neural Information Processing Systems*, pages 9123–9132, 2018.

[19] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274, 2002.

[20] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, pages 6348–6358, 2017.

[21] Hoang Le, Nan Jiang, Alekh Agarwal, Miroslav Dudik, Yisong Yue, and Hal Daumé. Hierarchical imitation and reinforcement learning. In *International Conference on Machine Learning*, pages 2923–2932, 2018.

[22] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[23] Shiau Hong Lim, Huan Xu, and Shie Mannor. Reinforcement learning in robust markov decision processes. In *Advances in Neural Information Processing Systems*, pages 701–709, 2013.

[24] Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*, 2018.

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[26] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems*, pages 9861–9871, 2018.

[27] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[28] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[29] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2817–2826. JMLR. org, 2017.

[30] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[31] Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736. ACM, 2006.

[32] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

[33] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[34] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[36] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

[37] Wen Sun, J. Andrew Bagnell, and Byron Boots. Truncated horizon policy search: Deep combination of reinforcement and imitation. In *International Conference on Learning Representations*, 2018.

[38] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 2. MIT press Cambridge, 2018.

[39] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[40] Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, pages 1449–1456, 2008.

[41] Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. *arXiv preprint arXiv:1901.09184*, 2019.

[42] Yuandong Tian, Jerry Ma, Qucheng Gong, Shubho Sengupta, Zhuoyuan Chen, James Pinkerton, and C Lawrence Zitnick. Elf opengo: An analysis and open reimplementation of alphazero. *arXiv preprint arXiv:1902.04522*, 2019.

[43] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[44] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[45] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1995–2003, 2016.

[46] Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

[47] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[48] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.