Spring 2025 COS 434 / ECE 433 – Introduction to Reinforcement Learning

Instructors: Benjamin Eysenbach
Lectures: Tuesday and Thursday, 1:30pm – 2:50pm (room tbd)
Precepts: Fri 11:00 – 11:50, 12:30 – 1:20
Office Hours:

- **Ben Eysenbach:** tbd
- **TAs:** tbd

Prerequisites:

1. Intro to ML: COS 324, ECE 435 or equivalent
2. Probability: ORF 309, or equivalent
3. Linear Algebra

Schedule and Readings: COS435 Calendar
Website: https://ben-eysenbach.github.io/intro-rl/
Questions? Post on Ed! You will get a response faster this way than by emailing the staff. You can make questions private as needed.

What is this course about? Reinforcement learning (RL) is a core technology at the heart of modern intelligent systems that learn to make good decisions in complex environments. It encompasses technologies such as continuous variable optimization, Q learning, neural networks, policy search, and bandit exploration. In this course, we aim to give an introductory overview of reinforcement learning, its core challenges, and approaches, including exploration and generalization. In parallel, we will present a collection of case studies from intelligent systems, games and healthcare. Through a combination of lectures, written assignments and coding assignments, students will become well-versed in key ideas and techniques for RL.

Learning objectives

- Be able to identify reinforcement learning problems and understand common assumptions (e.g., differences between planning, bandits, dynamic programming, open/closed-loop control)
- Be able to identify what makes the RL problem challenging.
- Be able to implement and explain the differences between common RL algorithms, including temporal-difference methods and policy-gradient methods
- Understand how RL problems are similar to and different from other machine learning problems.
- Gain familiarity with the mathematical tools of RL, including value functions and REINFORCE gradients.
- Explore new applications and open problems in RL (final project).

High-level structure of the course. The course will be broken up into 6 *chapters*:

1. **What is RL? (1 week)** We will introduce the RL problem both intuitively and formally and discuss practical examples. We will also discuss high-level challenges associated with RL:
    a. Long-horizon reasoning (Chpt. 4)

    b. Continuous actions (Chpt. 5)
    c. Task specification (Chpt. 6)
2. **Do you really need RL? (2 weeks)** Before introducing our first RL algorithms, we will talk about some easier approaches that can be used when extra information about your problem is available. These easier approaches will begin to introduce the technical foundation for the RL algorithms that we will subsequently study.
3. **3 simple RL methods (1 week)** We will introduce our first practical RL methods, which all focus on directly learning a policy. These methods will be very simple, but will only work in limited settings. To lift these limitations, we will need to introduce the value function (next chapter).
4. **Value functions (4 weeks)** We will introduce value functions, discuss how they are learned and why they can lift limitations of the simple RL methods introduced above. DQN will be our canonical example. The main limitation of these methods is that they require a small number of discrete actions, a limitation that the next chapter will address. *This chapter will conclude with a midterm held right before Spring Break. Chapter 5 will start after Spring Break*
5. **Actor critic methods (3 weeks)** To address practical problems where actions are continuous, we will study methods that combine a learned value function with a learned policy. PPO, TD3, and SAC will be the canonical algorithms studied in this section.
6. **Where do rewards come from (3 weeks)** We will discuss techniques for specifying tasks and learning from limited feedback.

For a more detailed schedule, see the course calendar.

Assignments

- Homework (50%): Most weeks will have a homework assignment, which will include written/mathematical problems and coding problems. Each will be due on a Monday at 11:59pm and assigned the week before. There will be a total of nine homework assignments, most of which carry the same grade:
  HW_avg = (0.0 x HW0 + HW1 + HW2 + HW3 + HW4 + HW5 + 0.5 x (HW6 + HW7 + HW8)) / 6.5
  The last three homeworks are shorter, as students will simultaneously be working on the course project.
    - Any homework submitted late will receive a 0. We will not permit any late days for homeworks because they will be graded and returned immediately after the original due date.
    - To pass the class, you must submit all homeworks within 7 days of their assigned deadline.
    - We will drop two homeworks grades among HW1 – HW8 and impute them with the weighted average of the remaining homework grades.[1]
- Midterm exam (20%): Held the week before Spring break (Thur March 6).
- Final project (25%): In the final project, students will work in teams of 3 to reproduce a recent reinforcement learning paper. This will entail re-implementing the method (using standard

[1] Technically, we will drop whichever two assignments will boost your weighted average grade the most.

deep learning frameworks) and running the experiments. Due May 11 at 5pm, with student presentations in weeks prior.
- Participation (5%): Attendance in lecture and precept, answering questions on Ed, etc. Note the University's policy on attendance:

  > *Excessive absences from classes negatively affect the totality of a student's educational experience and may lead to poor academic performance. Instructors may set their own attendance policy for their courses. But more than two weeks of cumulative absences, regardless of the reason a student misses a class, may represent grounds for a failing grade in a course.*

Precepts: There will be multiple 50-min sections led by the teaching assistants. You are welcome to attend whichever precept fits your schedule, assuming there is space. The aim of the precepts is to reinforce the material from lecture (not to introduce new material). They will include discussion of homework assignments and examples of exam-like questions.

Textbooks: While all required material will be covered in lecture, students may find the following textbooks useful for providing different perspectives on the concepts covered in lecture:

- Grokking Deep Reinforcement Learning – Good intuition for several of the key concepts we'll cover, along with some code examples.
- An Introduction to Deep Reinforcement Learning – Chapters 3 – 6 provide a good summary of some of the key concepts.
- Reinforcement Learning: An Introduction – "Sutton and Barto" is the standard reference in the field. It is very thorough and detailed.
- Reinforcement Learning: Theory and Algorithms – This is a great reference for theoretically-inclined students. We won't be covering much of this in the course.

Academic Integrity:
- **Collaboration policy**: for homework assignments, you are encouraged to work in groups of size ≤3, but expected to write all solutions and code on your own. It is not permissible to divide problems among group members. Please note your collaborators on each assignment. As a rule of thumb, think: "would I feel comfortable leading a precept discussion on this question?"
- **Midterm**: You will be permitted to bring one "cheat sheet" (8.5 x 11in, double sided), but are not permitted to use any other written or electronic resources.
- **Generative AI and LLMs**: For the most part, students are not permitted to use LLMs or Generative AI tools for any portion of the course.
  - The one exception is that it is permissible to use tools to automatically find typos or grammatical errors in written writing and to use type-checking and similar coding tools (e.g., identifying missing dependencies). However, you will be held accountable for using a GenAI tool even if you did not know that this tool had a certain feature.

For example, if you installed GitHub CoPilot thinking it just would fix dependencies, but then it ended up rewriting your algorithm, we will treat this as a violation of course policy, on par with copying from a classmate. As another example, if you upload your written report to ChatGPT asking it to find some typos but it also reports some mathematical errors ([GenAI tools often do this]), we will treat this as a violation of the course policy, on par with emailing an expert to see if they can find any mistakes in your report.

    - When in doubt, ask the instructor before using a tool. As a rule of thumb, think: "would I be comfortable submitting my search history along with my assignment?"

- **Search engines** (e.g., Google, Bing) may be used to find general resources (e.g., documentation for PyTorch, papers mentioned in office hours), but not for fine-grained questions relating to the homework (e.g., "how do you implement REINFORCE?"). As a rule of thumb, think: "would I be comfortable submitting my search history along with my assignment?"
- For the **final project** you are required to work in groups. Unlike the assignments above, you will be permitted to build upon existing codebases or theoretical results, making sure to acknowledge the tools used and how your work builds on top of the prior work. Err on the side of citing too much.

Student wellbeing: Princeton University values the wellbeing of all community members. We understand that various experiences can be distressing such as relationship struggles, identity concerns, depression, anxiety, academic struggles, difficulty adjusting to college, relying on substances and alcohol. If you or someone you know needs support or is looking to access specific services, consider reaching out to these university and student-led resources:

- Your residential college advising team is always a good first resource for advice and counsel. The assistant deans of student life (DSLs), whose offices are located in each residential college, serve as case managers in crisis situations. They are also available to talk with you about well-being concerns and can refer you to appropriate campus resources.
- The Office of Disability Services facilitates reasonable accommodations to support students with disabilities. Contact them at 609-258-8840 or by e-mail at ods@princeton.edu to discuss access issues and learn more about possible accommodations.
- If you are feeling distressed and need help, please contact Counseling & Psychological Services (CPS) at 609-258-3141 for immediate support or to schedule an appointment with a counselor. CPS is a confidential resource.
- The Sexual Harassment/Assault Advising, Resources and Education (SHARE) office is a survivor-centered, trauma-informed, confidential resource on campus. SHARE provides crisis response, support, counseling, advocacy, education, and referral services to students experiencing unhealthy relationships and abuse, including harassment, sexual assault, dating/domestic violence, and stalking.
- The Princeton Peer Nightline is a student-run anonymous peer listening service. It is not affiliated with CPS or the University administration. They offer anonymous chat/call peer support.

Disability Services and Academic Accommodations: If there is anything the course can do to be more accommodating of students, please let the course staff know! This could range from suggestions for making discussions more inclusive to ways to make presentations more friendly for visually impaired students.

Electronic Devices: Please do not use laptops and cellphones in class. Studies now conclusively show that they detract from learning – for you and those around you. If you have an academic accommodation that requires the use of an electronic device, please contact the course staff.