

SYLLABUS–CMSC389L

Samuel Kinstlinger, Andrew Xu, & Brian Fugh

1. Class Description and Logistics

Course Name: Reinforcement Learning

Course Description: This course provides a comprehensive introduction to reinforcement learning. We begin with a review of key concepts in machine learning and planning, such as gradient descent and neural networks. We then explore core reinforcement learning algorithms—including value-based, policy-gradient, and actor-critic methods—along with essential tools like TensorFlow and Gymnasium. The course focuses on applications in robotics and autonomous systems, finance, and games.

Learning Outcomes: By the end of the course, students will be able to:

1. Formally describe reinforcement learning problems
2. Understand the theoretical foundations and practical applications of different algorithms
3. Gain familiarity with the primary tools used in reinforcement learning research and projects

Prerequisites: Minimum grade of C- in CMSC320; MATH241, 1 course with a minimum grade of C- from (MATH240, MATH341, MATH461); and permission of CMNS-Computer Science department.

Time & Place: TBD

Course Dates: TBD

Credits: 1

Instructors:

Samuel Kinstlinger - samkinst@terpmail.umd.edu

Andrew Xu - andyxu55@terpmail.umd.edu

Brian Fugh - bfugh@terpmail.umd.edu

Advisor: Dr. Teli - nayeem@umd.edu

Office: IRB 2224

Office Hours:

Andrew: 5-6 PM Mondays (Discord)

Samuel: 11-12 AM Wednesdays (Discord)

Brian: 1-2 PM Thursdays (Discord)

Textbook: Robotics: Perception & Planning by Samuel Kinstlinger, Andrew Xu, Sameer Chawla, & Brian Fugh:

<https://sam-kinstlinger.github.io/Robotics-Perception-Planning-Textbook/>

Lecture PowerPoints will also be published to Canvas in Files

2. Assignments and Grading

You are responsible for all material covered in lectures, as well as any information posted on the class webpage, including announcements, deadlines, and policies.

The primary deliverables include:

1. **Canvas Discussions:** Regular discussions on Canvas covering various reinforcement learning concepts. These will be graded based on completion and coherence according to the provided rubric
2. **Projects:** Students will complete five comprehensive projects that apply reinforcement learning principles. All projects must be submitted electronically following the instructions provided in class. Submissions via other methods (e.g., email) will not be accepted.

It is your responsibility to thoroughly test your program and ensure it functions correctly before submission. Late submissions will be accepted up to 24 hours after the deadline with a 10% penalty. To maintain academic integrity, do not share or publicly post your project solutions online. Any violation of this policy will be considered academic misconduct.

Grading Criteria:

- Completion of all required functionality
- Correct execution and implementation
- Performance of the final result
- Cleanliness of the script

Advice on Debugging Code:

- Read the entire project description before starting
- Theoretical questions about class content or project specifications take precedence over project debugging
- When you are stuck as how to start or what to do next ask yourself the following questions:
 - What is the goal?
 - What are the steps needed to reach that goal? (Alternatively, how would you complete this task manually?)
 - What step are you on?
- When staff does look at code, staff will not help debug code that is not well commented. One thing to note is that commenting should not only state what the code is doing, but also *why* you need that code.

Each assignment is due by **11:59 PM EST on the Friday following the class in which it is assigned**. Your final grade will be based on your performance in both the Canvas discussions and the projects.

Assignment	Percentage
Projects	80%
Project One - Blackjack Q-Learning	15%
Project Two - Cartpole Deep Q-Learning (DQL)	20%

Project Three – Lunar Landing Deep Deterministic Policy Gradient (DDPG)	20%
Final Project - Student's Choice	25%
Discussions	10%
Attendance	5%
Class Participation	5%

Grades: Course grades cutoffs are expected to be approximately: A+ $\geq 95\%$; A $\geq 91\%$; A- $\geq 88\%$; B+ $\geq 85\%$; B $\geq 81\%$; B- $\geq 78\%$; C+ $\geq 75\%$; C $\geq 71\%$; C- $\geq 68\%$; D $\geq 58\%$; F $< 58\%$.

To ensure fairness, grading standards will be applied consistently to all students. That said, the primary goal of this course is to help you learn and understand reinforcement learning, not to enforce rigid grading barriers. If you attend class regularly and complete all assignments in good faith, you should be well-positioned to earn an A.

3. Schedule

Week	Topics	Textbook Section	Principal Lecturer
9/5	Introduction to Course, Machine Planning Intro	1	Samuel
9/12	Machine Planning, Local Planning	3.2, 3.3	Andrew
9/19	Intro to Reinforcement Learning, Reward Functions	4.1, 4.2	Samuel
9/26	Exploration vs. Exploitation	4.3	Samuel
10/3	Q-Learning, Gymnasium, Blackjack Project Assigned	4.4	Andrew
10/10	Value Iteration, Continuous State	4.5	Andrew

	Spaces, Learned Continuous Q-Function		
10/17	Machine Learning, Gradient Descent, Introduction to UFA	2.1, 2.2, 2.3.1	Samuel
10/24	Shallow Neural Networks	2.3.3	Samuel
10/31	Deep Neural Networks, Deep Q-Learning, Cartpole DQL Project Assigned	2.3.4, 4.5	Samuel & Andrew
11/7	REINFORCE, VAC	N/A	Andrew

11/14	DDPG, Lunar Landing Project Assigned	4.4.4	Samuel
11/21	SAC, Final Assigned	N/A	Andrew
11/28 (Thanksgiving)	N/A	N/A	N/A
12/5	Final Presentations	N/A	N/A
12/12	Final Presentations	N/A	N/A

4. Other Policies

Academic Integrity: You should be familiar with the University's policies on https://faculty.umd.edu/main/activity/teaching-policies-guidelines#acad_integrity. Programming projects are to be written **individually**, therefore cooperation or use of unauthorized materials on projects is a violation of the University's Code of Academic Integrity. Both the person receiving assistance **and the person providing assistance** are in violation of the honor code. **Any evidence** of this, or of unacceptable use of computer accounts, use of unauthorized materials or cooperation on exams or quizzes, or other possible violations of the Honor Code, **will be**

submitted to the Student Honor Council, which could result in an XF for the course, suspension, or expulsion. For learning the course concepts (including the programming languages), students are welcome to study together or to receive help from anyone else. You may discuss with others the project requirements, the features of the programming languages used, what was discussed in class and in the class web forum, and general syntax errors.

Honor Code: The University of Maryland Code of Academic Integrity is administered by the Student Honor Council. The Code sets standards for academic integrity. Note that copying solutions from other sources is plagiarism.

Religious observances: If your religion dictates that you cannot hand in assigned work on a particular date, then contact me **at the beginning of the semester** to discuss alternatives. You are responsible for making these arrangements.

Course Related Policies: <http://www.ugst.umd.edu/courserelatedpolicies.html> This page provides important information on academic policies and procedures, for example in regards to academic integrity, accommodations, excused absences, and other topics.

Right to Change Information: Although every effort has been made to be complete and accurate, unforeseen circumstances arising during the semester could require the adjustment of any material given here. Consequently, given due notice to students, the instructors reserve the right to change any information on this syllabus or in other course materials.

Copyright: All course materials are copyright Samuel Kinstlinger & Andrew Xu. All rights reserved. Students are permitted to use course materials for their own personal use only. Course

materials may not be distributed publicly or provided to others (except other students in the course), in any way or format.