University of South Carolina
College of Engineering and Computing

CSCE 775: Deep Reinforcement Learning
Spring 2025

Forest Agostinelli, Ph.D. Assistant Professor Computer Science and Engineering foresta@cse.sc.edu	Blackboard Class Time: MW 2:20pm-3:35pm Class Location: 300 Main Street B101

Course Description

Reinforcement learning studies how artificially intelligent agents can learn to make decisions from their own experience. It has achieved superhuman performance on Atari games, Go, Chess, and the Rubik's cube and is now being used in chemistry, physics, robotics, quantum computing, and large language models. Facilitating this success is deep learning and search.

Deep learning the study of the intersection of machine learning algorithms and artificial neural network architectures. Using deep learning, we can train computers to recognize images, recognize speech, translate languages, solve problems in the natural sciences, and generate data. Most of this can be done using raw data, such as pixels. Reinforcement learning leverages deep learning for prediction, control, and constructing a model of the world. The combination of deep learning and reinforcement learning is known as deep reinforcement learning.

Search allows one to "think" before making decisions by anticipating the outcomes of actions. Even for problems where the number of possible states is greater than what one could explore in a lifetime, we can use search to prioritize more promising states over less promising ones to quickly find a solution. In order to do this well, deep reinforcement learning has been used to learn how to prioritize these states. Search has an essential role in many tasks, such as playing Go, solving the Rubik's cube, designing chemical structures, robotics, and large language models.

Prerequisites

You should be a proficient programmer. Programming assignments will be in Python. If you do not know Python, it will be expected that you learn it independently and rapidly.

You should also have a basic knowledge of:

- Probability and statistics: Baye's rule, distributions, mean, standard deviation
- Linear algebra: Matrix multiplication
- Calculus: Derivatives and integrals

Learning Outcomes

Students who take this class will learn to:

- Characterize a wide variety of problems, including those in the natural sciences, as Markov decision processes (MDPs).
- Solve problems posed as MDPs using tabular reinforcement learning methods
- Construct and train deep neural networks
- Integrate reinforcement learning with deep learning
- Solve problems posed as MDPs with deep reinforcement learning methods
- Improve learning and decision making using search algorithms
- Understand and critique current deep reinforcement learning and search papers
- Propose new research ideas in the field of deep reinforcement learning and search

Required Textbooks

Much of the course material will draw from the second edition Sutton and Barto's book on reinforcement learning which is freely available online: http://www.incompleteideas.net/book/RLbook2020.pdf

Other useful books and papers include:

- Pearl, Judea. *Heuristics: Intelligent search strategies for computer problem solving.* Addision Wesley (1984).
- Puterman, Martin L. *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons, 2014.
- Hart, Peter E., Nils J. Nilsson, and Bertram Raphael. "A formal basis for the heuristic determination of minimum cost paths." *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968): 100-107.
- Browne, Cameron B., et al. "A survey of monte carlo tree search methods." *IEEE Transactions on Computational Intelligence and AI in games* 4.1 (2012): 1-43.
- Bertsekas, Dimitri P., et al. *Dynamic programming and optimal control.* Belmont, MA: Athena scientific, 1995 (Volumes 1 and 2).
- Bertsekas, Dimitri P., and John N. Tsitsiklis. *Neuro-dynamic programming.* Athena Scientific, 1996.

All reading materials comply with copyright/fair use policies.

Course Overview

The class will consist of face-to-face lectures. All required material for the class will be posted online. Class sessions will sometimes require reading and/or coding, beforehand. The instructor, along with the students, will be able to help each other with conceptual questions through Piazza.

The expected turnaround time for discussion board postings and communication via email is one business day. The expected turnaround time for homework assignments is one week. Students are encouraged to attend the weekly office hours to ask questions and get feedback on ideas.

Technology

Students must have access to a unix based system such as linux, macOS, or a virtual linux environment on Windows. The students will be given an Anaconda environment that contains all the Python packages needed to complete the homework assignments. Students should be familiar with unix and be experienced programmers.

Homework

The homework will consist of programming exercises to assess your understanding of the practical aspects of reinforcement learning, deep learning, and search. Homework will consist of 3-4 coding assignments. The main purpose of the homework is to build a solid base for your class project and for the future work you do in this area throughout your career. Therefore, to receive credit for coding assignments, the code turned in must run. The lowest homework assignment grade will be dropped.

Quizzes

In class quizzes will be given to assess your conceptual knowledge. There will be 4-5 quizzes throughout the semester. Your lowest quiz will be dropped. To receive credit, all answers must be legible. Your lowest quiz grade will be dropped.

Exams

There will be a midterm exam to assess your conceptual knowledge. To receive credit, all answers must be legible.

Project

The class project gives you the opportunity to implement deep reinforcement learning and search algorithms to solve a problem of your choosing. The project will be done in teams of two to three. You are encouraged to incorporate aspects of your own research into your project. While novel research ideas and applications are encouraged, they are not necessary to receive full credit. I will set aside time to meet with teams one-on-one to discuss ideas.

Each team will write a project proposal that outlines the goals for their project and their plan to implement it by the end of the semester. Afterwards, each team will write a progress report that details what progress has been made and what is left to be done. The purpose of the proposal and the report is to ensure each team is on-track to complete a quality project.

During the last two weeks of the semester, each team will give a presentation to the class detailing the background information, your approach, your results, and lessons learned. Students can ask each other questions and give each other feedback. The project report should be about 6-8 pages and should contain the following sections: abstract, introduction, related work, background, methods, results, conclusion, and references. As expected, especially in a research setting, not every idea will work. However, if that is the case, you should give an in-depth analysis as to why, what we can learn from this, and other possibilities for future research.

Paper Presentation

Students will give a paper presentation on a research paper involving deep reinforcement learning and search. The presentation should be 10 minutes long and should be well-rehearsed. There will be 1-2 minutes of questions and answers after each presentation. The goal of the presentations is for students to get familiar with understanding and critiquing research papers.

Piazza

We will be conducting all class-related discussion on Piazza. We encourage you to ask questions when you're struggling to understand a concept—you can even do so anonymously. To ensure everyone turns in their own work, please do not post code or solutions (or partially completed solutions) to homework assignments on Piazza.

Late Work

Any homework turned in after the assigned deadline will be marked late regardless of how close to the deadline it may be. Homework can be turned in a maximum of 1 day late with a penalty of 20 percentage points. Students should aim to submit their homework early in order to avoid any last minute issues.

Grading

Homework	20%
Quizzes	20%
Midterm	25%
Project Proposal	5%
Project Progress Report	5%
Project Presentation	10%
Project Report and Code	10%
Paper Presentation	5%

Grades will be determined on the following scale:

A	[90 – 100]
B+	[86 – 90)
B	[75 – 86)
C+	[70 – 75)
C	[60 – 70)
D+	[55 – 60)
D	[40 – 55)
F	[0 – 40)

Incomplete

A grade of Incomplete ("I") is only given in extreme cases when a student is unable to complete some portion of the assigned course work because of a significant incident. These may include an unanticipated illness, accident, work-related responsibility, family hardship, or verified learning disability. An incomplete will only account for 20% of the overall course grade, and it only applies to work after the reported incident. In addition, a student must be in good grade standing, a "C" or greater, at the time of the incident to qualify.

Course Schedule

Week 1: Introduction and Markov Decision Processes

Week 2: Dynamic Programming and Model-Free RL I

Week 3: Model-Free RL II

Week 4: Deep neural networks

Week 5: Approximate dynamic programming

Week 6: Policy gradients

Week 7: Model-Based RL

Week 8: A* Search and Monte-Carlo Tree Search

Week 9: Sparse rewards

Week 10: Partially observability and exploration vs exploitation

Week 11: Explainable RL

Week 12: Meta Learning

Week 13: Paper Presentations

Week 14: Project Presentations

Week 15: Project Presentations

Attendance Policy

Attendance is an essential part of this class as asking questions, discussing lecture material, and proposing new ideas will be greatly enhance your learning experience.

Request for Accommodations

If you are a student with a disability and require accommodation to participate, notify me immediately and contact the Student Disability Resource Center (http://www.sa.sc.edu/sds, 1705 College Street, Close-Hipp, Suite 102 Columbia, SC 29208, 803-777-6142, sadrc@mailbox.sc.edu) for verification of eligibility and determination of specific accommodations. In addition, please provide me the required accommodation letter from the Student Disability Resource Center.Disability Resource Center.

Academic Integrity

All work turned in must be your own. Plagiarism of any kind, including from online sources, is strictly prohibited. All potential Honor Code violations will be reported to the Office of Academic Integrity. Honor Code violations of any kind (including plagiarism) on the homework assignments will result in a zero on that assignment. Furthermore, students who have plagiarized a homework assignment or quiz will not be able to drop their lowest grade. Honor Code violations of any kind (including plagiarism) on the midterm will result in failure of the course. You can familiarize yourself with the Honor Code here: http://www.sc.edu/policies/ppm/staf625.pdf.

Student Interaction

- Student-to-Instructor: There will be weekly face-to-face sessions held for synchronous instruction. There will be weekly office hours.
- Student-to-Student: Students will be able to interact with each other on Piazza and during synchronous sessions.
- Student-to-Content: Students will have access to the slides.

Health and Safety

Students are expected to comply with all university health and safety guidelines including those about COVID-19. For current COVID-19 guidelines, visit
`https://sc.edu/safety/coronavirus/safety_guidelines`.

Syllabus Change Policy

This syllabus is a guide and every attempt is made to provide an accurate overview of the course. However, circumstances and events may make it necessary for the instructor to modify the syllabus during the semester and may depend, in part, on the progress, needs, and experiences of the students. Changes to the syllabus will be made with advance notice.