CS 4320 Final Project
Initial submission: Sat, November 15 at 11:59 PM
Final submission: Mon, Dec 8 at 11:59 PM

Groups: You are expected to work in a group of two or three for this assignment. Please sign up for a group as soon as possible.

Goal: To extend our study of reinforcement learning to more complex environments, and incorporate more advanced learning techniques.

As the project component of the course, this is designed to be more flexible than a typical assignment and to require you to do some additional learning and investigation on your own. Within some parameters, you will be able to make some choices about environments, specific algorithms/methods to investigate, and tools to use. You are encouraged to make use of existing libraries and toolkits to support your efforts as long as you do something on your own with them and acknowledge what you use; learning about these and connecting them to the concepts of the course is one of the potential learning goals.

Project Outline

The standard project is designed to build on our discussion and homework on reinforcement learning. In particular, you will apply RL methods to new, more complex environments, and investigate some techniques for scaling RL to these domains and speeding up learning for more realistic settings.

The first part of this assignment is to identify a new environment that you will try to learn an effective policy for. I will give some suggested options, but ultimately this is up to you. You can choose to implement your own simulation or use environments like the OpenAI Gymnasium (https://gymnasium.farama.org/index.html) that have many existing environments. For this assignment please consider:

1. The environment should be of low to moderate complexity; poses some challenge but not TOO hard to learn
2. The effort to implement the environment should not be too high; we want to focus more on the learning aspect
3. Ideally, you can start with a simpler version and vary the complexity

The second part is to investigate more advanced RL learning strategies, building on your understanding of basic TD/Q-learning methods from PA2. There are two general strategies you MUST try in some form (there are many specific variations of each of these):

1. **Function Approximation:** You must try some form of function approximation (i.e., non-tabular) method for representing the value function.

This can be a neural network or any other method you prefer; you are also free to use libraries to implement this.

2. **Experience Replay Buffer:** You must implement some version of an experience replay buffer, where you store previous experiences and use them to perform more efficient updates.

In addition, you should consider at least a couple of additional variations, which might include things like:

- Different learning updates (e.g., Proximal Policy Optimization (PPO))
- Reward shaping methods (add heuristics, subgoals, etc.)
- More sophisticated exploration strategies (e.g., curiosity-driven)
- Curriculum training strategies (start simple, increase complexity)

Deliverables

The overall focus is on exploration and learning, and documenting what you have tried. The specific results are less important than showing that you have put in significant effort to try some interesting approaches, documented what you have discovered (whether it worked or not!), and tried to understand and learn from the results.

Initial report (1-2 pages): The initial focus should be on background research and identifying environments, techniques, and tools you will use. There is a short initial report due a week or so in that should include:

- Team members and responsibilities
- What environment/problem you plan to address
- A brief plan for what techniques you plan to try
- Any issues or problems you anticipate

Final report (5-10 pages and source code): Your final report will document what you have done and the results you have achieved. You will also submit any source code and supplementary files, including additional results, etc. to fully document your project. This should include at least:

- A short overview/motivation for the project
- Description of the contributions of the team members
- A description of your environment
- A description of the specific methods you implemented
- Results that show how effective your methods are; you should evaluate each method individually
- Discussion and analysis of your results, including what you learned and any challenges you encountered