

Mathematical Foundations of Reinforcement Learning:
EE619 Project #1

EE619 PROJECT #1

1 Introduction

In this project, you will implement Q-learning & Double Q-learning in *Lecture Note 5* and use the GridWorld environment to train an agent via these algorithms. We provide the GridWorld environment you will use.

Next, there are information for this project.

- Due Date: April 30(Sunday), 2023
- Submission Form: a **single pdf** file containing the contents mentioned in Section "What to Submit".
- Cautions: this project is a personal project. Do not discuss it with anyone

2 Specifications

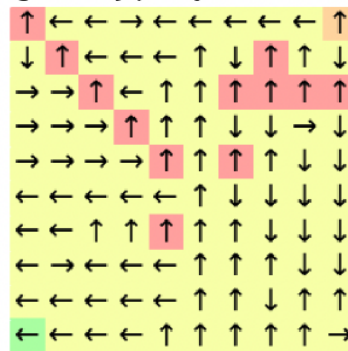
- Language to Use: Python 3.6 or higher
- Files: We provide three python files.
 1. main.py
This file contains codes that you have to fill in and other codes helping your experiment.
 2. algorithm.py
This file contains classes for Q-learning and Double Q-learning that you have to implement.
 3. GridWorld.py
for the GridWorld environment. **Do not modify** this file.

3 What to Submit

As mentioned before, you have to submit a single pdf file, and we limit your submission up to five pages. Most importantly, your submission file should contain the following contents.

1. (30 points) Summarize Q-learning Double Q-learning with your own words.
2. (20 points) Implement these above algorithms, and put your codes in your submission. You can see blank space to fill in your codes in **algorithm.py**.
3. (20 points) In the main.py, Q-learning, Double Q-learning is trained, and the learned policy is saved every 'save_policy_interval'. Implement **plot_policy() at main.py** which **plots this saved policy** as shown in the figure below. The plotted result does not have to be identical to the one below. As long as your implementation expresses the policy with arrows on the grid clearly, you can design it freely. If done correctly, you will get two files: 'Q_learning_policy.pdf' and 'Double_Q_learning_policy.pdf'. Please include these files in your report.

Q-learning Greedy policy at 10000 th episode



4. (20+10 points) If you complete the implementation, then you can generate **two result files**: *learning_curve.pdf* and *max_Q_for_each_state.pdf*.
 (20 points) Put the two result files in your submission and analyze these results.
 (10 points) Explain whether your Q-tables overestimate or not. You can add an additional experimental result for your explanation if you need.