

CS 6955 HW4: Q-Learning

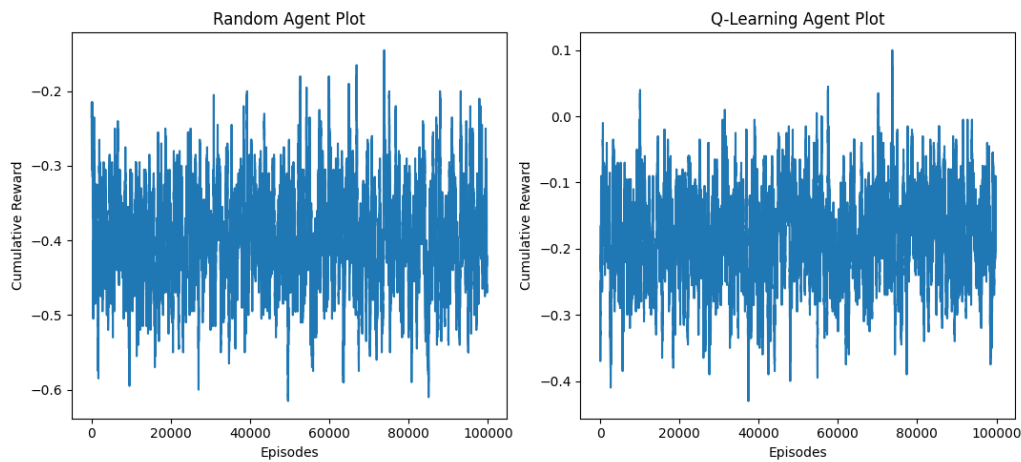
Kutay Eken
u1322888

February 2025

1 Part 1:

Below is the algorithm results:

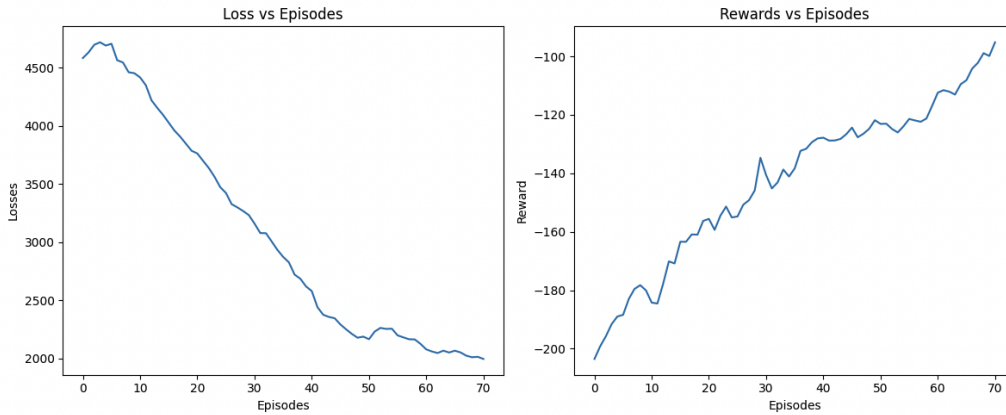
- **Random Agent Success Rate:** %39.516
- **Q-Learning Agent Success Rate:** %28.138



The given plots support the algorithm's results and show that the Q-Learning agent performed better. The agent was trained over 100,000 episodes with a discount factor (gamma) of 0.95 and a learning rate (alpha) of 0.1. Epsilon discounting was not used in this part. The cumulative reward for the random agent was centered around -0.4, whereas the Q-Learning agent's cumulative reward was centered around -0.2, indicating better performance. These plots were smoothed using a moving average with a window size of 200.

2 Part 2:

My DQN agent was trained over 120 episodes and tested over 50 episodes. Below are the plots of losses and rewards over episodes during the training process. The plots have been smoothed using a moving average with a window size of 50.



As seen in the graphs, in general, the loss values decrease as training progresses and the agent receives higher rewards at each step. The spikes were expected due to exploration. This demonstrates that my DQN agent is learning and improving over time.

Below are the average rewards for each agent over 50 runs:

- **Random Agent:** -179.08885095961
- **DQN Agent:** -39.59084012697525

When I visualized my DQN agent, I observed that it performed better than both a random agent and my own gameplay. One key observation was that my DQN agent could make minor adjustments to improve accuracy toward the target. However, it struggled when the balance of the lander changed significantly, especially when it was close to the ground, often ending up landing hard.