Imperial College London

COURSEWORK 2

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Reinforcement Learning

Author:
Arnold Cheung (CID: 01184493)

Date: November 21, 2019

Part 1

Question 1 (i)

Comparing Figure 1 and Figure 2, it can be clearly seen that the variance of MSE loss is much lower when using experience replay buffer comparing to online learning. There is also a clear decreasing trend when using experience replay buffer which cannot be seen in online learning. Using experience replay buffer will therefore result in a more stable training. This is due to the fact that in online learning, each transition is discarded after a single training, the agent is not able to train on that transition again without being in the exact same state. Secondly, the training is highly correlated between subsequent transitions, which causes the distribution of training data to change over the training session. Using experience replay buffer, allows the agent to train on a randomly sampled mini batch from a stored set of previous transitions that solve both problems.

Question 1 (ii)

Using the experience replay buffer is more efficient in terms of improving the Q-network's predictive accuracy in a given amount of wall clock time. Although the agent trains on about the same number of transitions in a given time in both cases, training using the experience replay buffer gives the agent more meaningful and representative data to train on and is therefore more efficient in improving the Q-network.

Question 2 (i)

It is more likely that the bottom left corner has a more accurate Q-value predictions than the upper right region. This is due the agent starts exploring randomly from the bottom left corner, it will therefore have trained more from the transitions around the bottom left corner than upper right corner, especially in a short episode length before the agent is reset.

Question 2 (ii)

From inspection of the greedy policy, the agent will not be able to reach the goal state. It can be seen that the agent repeats the action towards the upper boundary until the end of the episode, as the action with the highest value is going up at that state which makes it stay at the current state.

Question 3 (i)

It can be seen from Figure 5 that the loss increases up until around 100 steps before decreasing, this due to the full Bellman equation uses Q-values from future states to update the Q-value of the current state, meaning if the Q-values of future states

are inaccurate, neither will the Q-value of the current state. The environment is essentially under-explored for the first 100 steps and the agent uses inaccurate future Q-values. As the agent explores more (after step 100), the Q-values begin to converge, resulting in decreasing loss.

Question 3 (ii)

Figure 6 shows spikes in loss every 20 steps taken, this aligns with the fact that the target network updates from the Q-network every 20 steps taken. Between the target network update intervals, the agent is trained using the pre-updated target network, and a decrease in loss is expected. Whenever the target network updates, there will be an abrupt change in the TD target, the TD error will therefore increase abruptly demonstrated by the spikes in the loss.

Question 4 (i)

The optimal value of δ has been found to be 0.05. It is superior to $\delta = 0$ as $\delta = 0$ means that ϵ will remain 1.0 and the agent will always randomly choose its action except for the optimal action. Even if the agent finds the optimal action, the agent will never train with transitions close to the goal.

Question 4 (ii)

It is also superior to $\delta = 1.0$ as $\delta = 1.0$ means that ϵ will instantly decrease to 0 after one step, and chooses its action greedily with only the first step for exploration. The agent is essentially not trained and will only reinforce the initialised path as optimal.

Question 5 (i)

The custom reward function compares the agent's current distance from goal with the agent's next state distance from the goal, and rewards based on whether the distance has increased or decrease. If the distance had decreased, reward is +15; if the distance had increased, reward is -10; if the next state is the goal, reward is +100. This is a superior reward function as it creates a larger difference in reward a good and a bad action, also increases the importance of reaching the goal. It can be seen from Figure 8 that using the custom reward function reaches a lower final distance and at a smaller step count.

Question 5 (ii)

The sparse reward function will give worse result than the above described custom reward function. Only one state (goal) rewards the agent, meaning that unless the agent randomly arrive at the goal, no rewards will be given to train the Q-network and therefore essentially no training will be done.

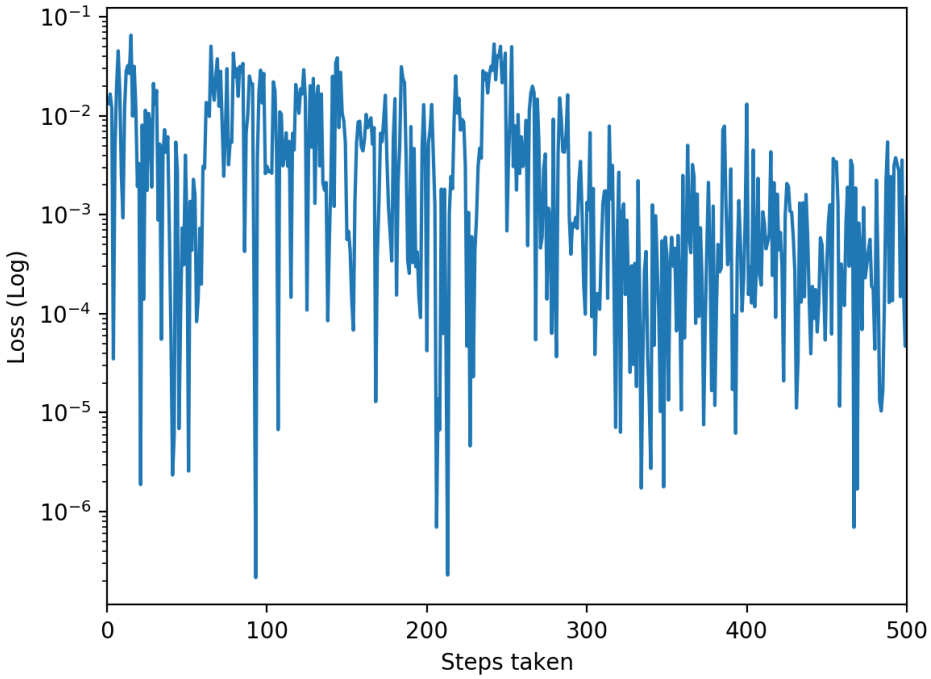

Figure 1: Log MSE loss against steps taken with online learning

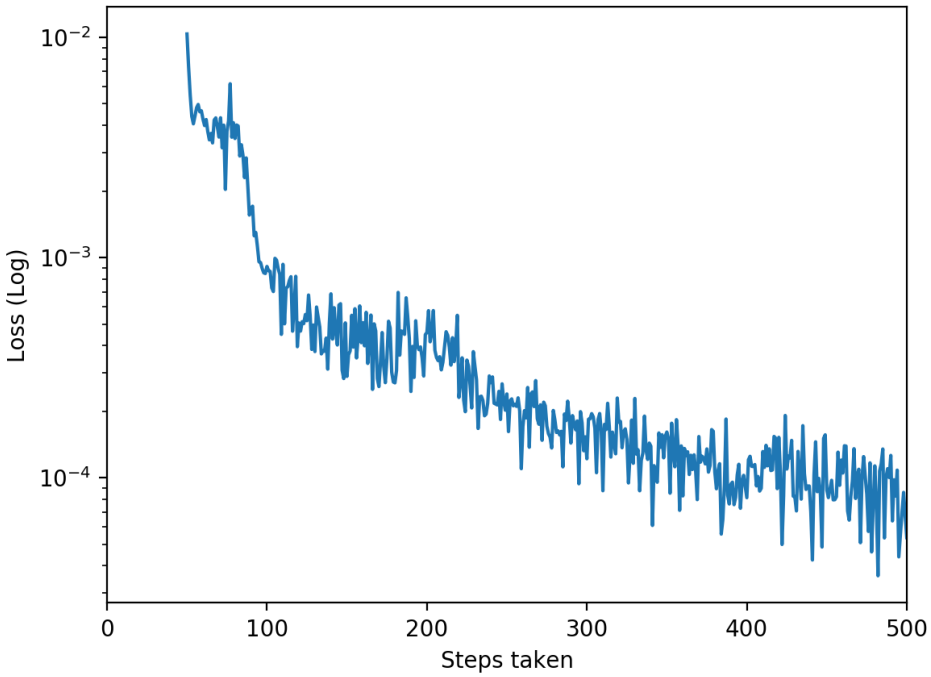

Figure 2: Log MSE loss against steps taken with experience replay learning

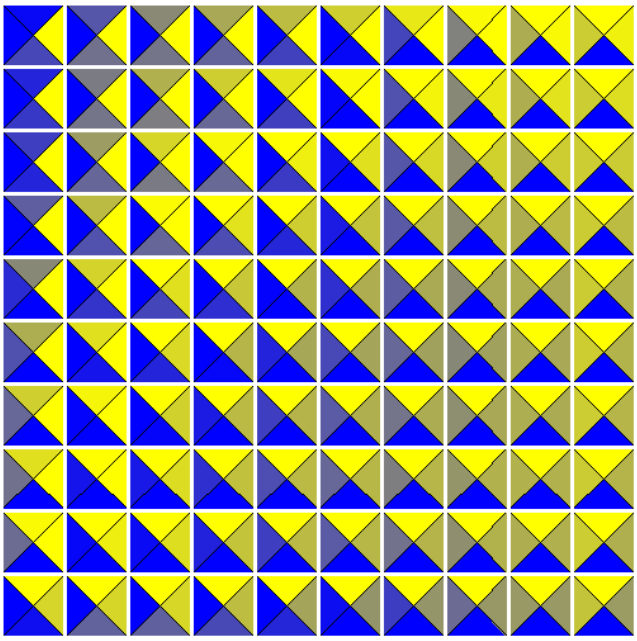

Figure 3: Visualisation of the Q-values, the action with highest value is coloured in yellow and lowest value in blue, the other two actions are coloured using linear interpolation between the highest and lowest value.

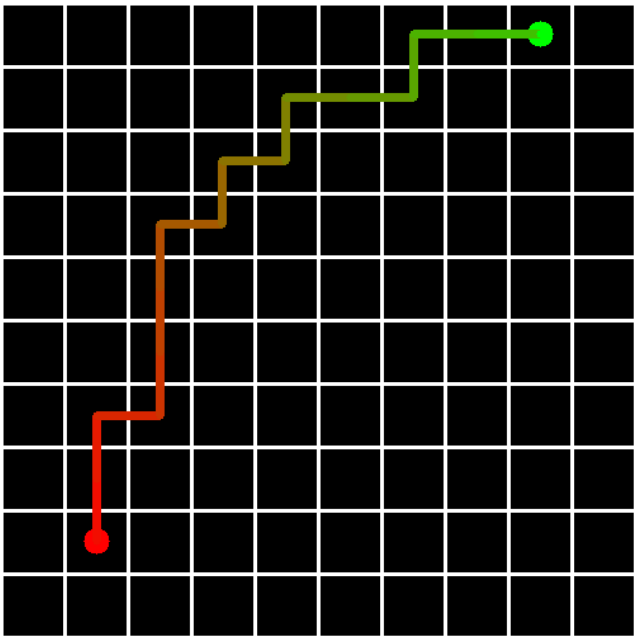

Figure 4: Visualisation of the greedy policy, the agent starts from the position coloured in red and ends in the position coloured in green.

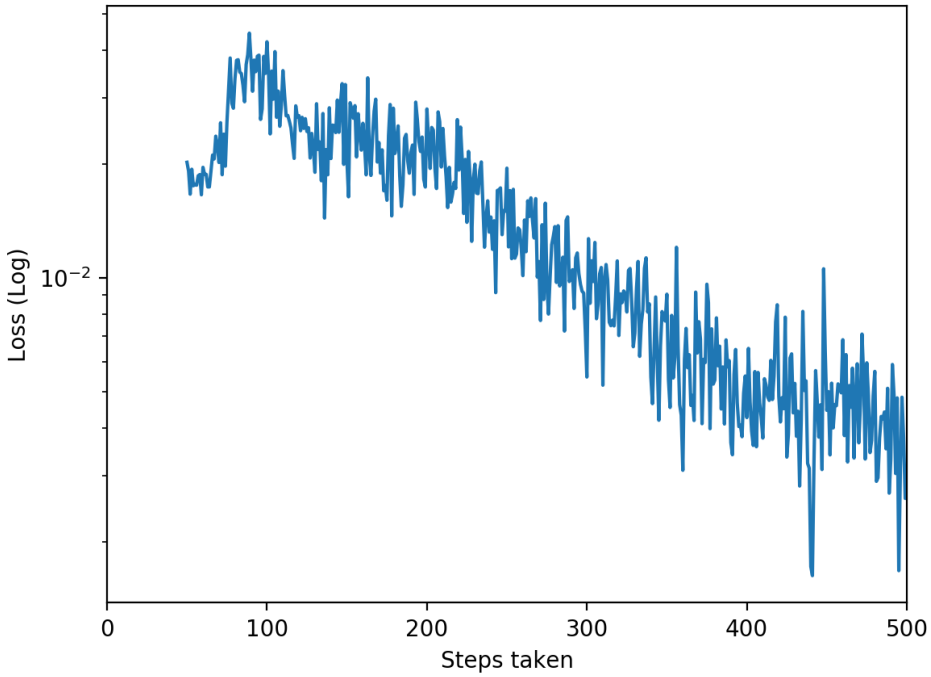

Figure 5: Log MSE loss against steps taken with the full Bellman equation.

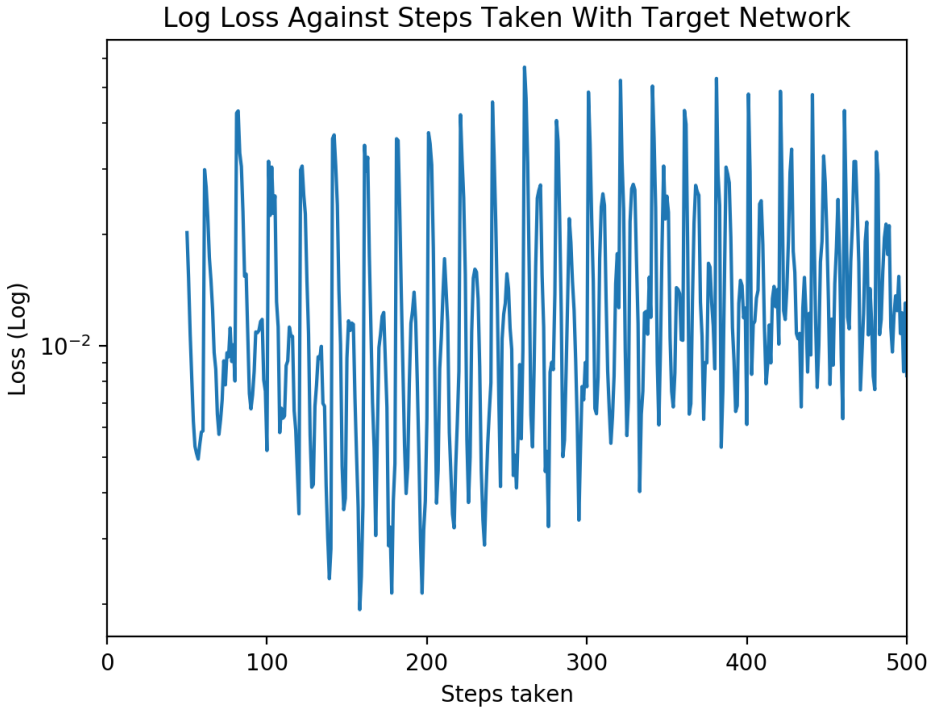

Figure 6: Log MSE loss against steps taken with the full Bellman equation and target network.

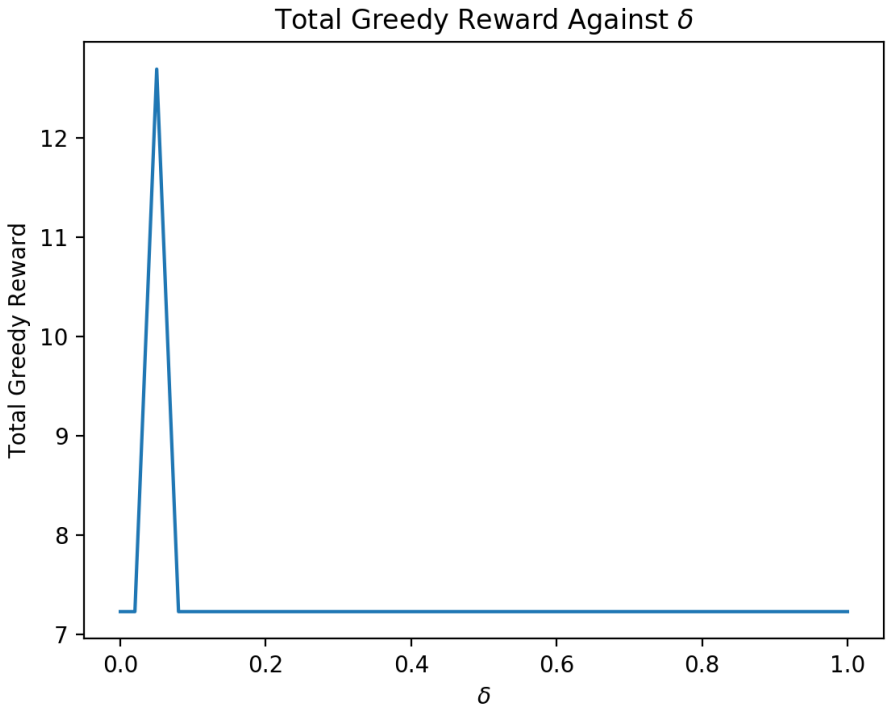

Figure 7: Log MSE loss against steps taken with the full Bellman equation.

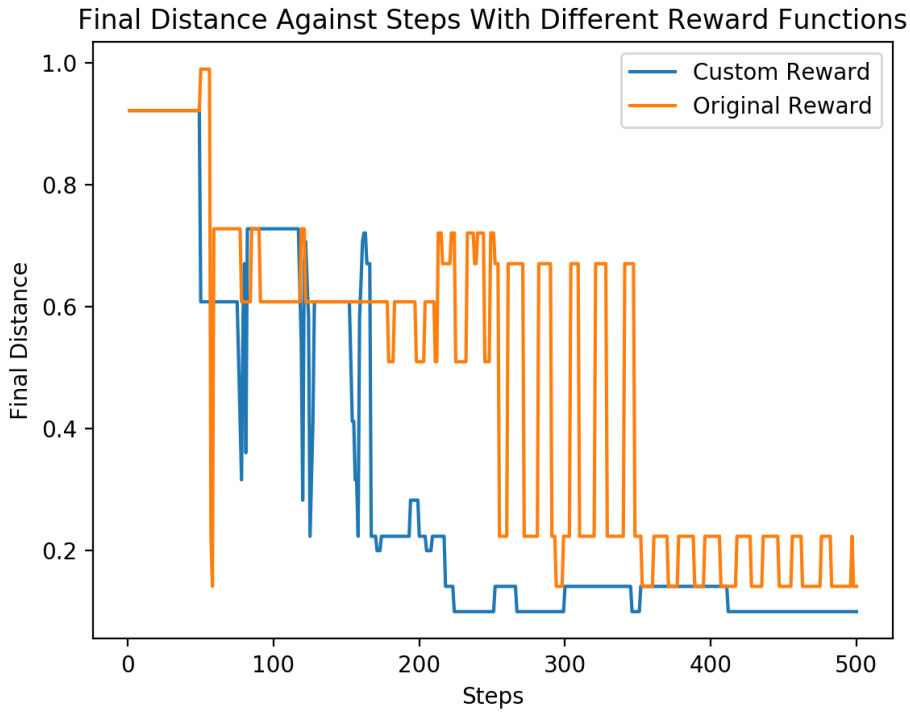

Figure 8: Log MSE loss against steps taken with the full Bellman equation and target network.

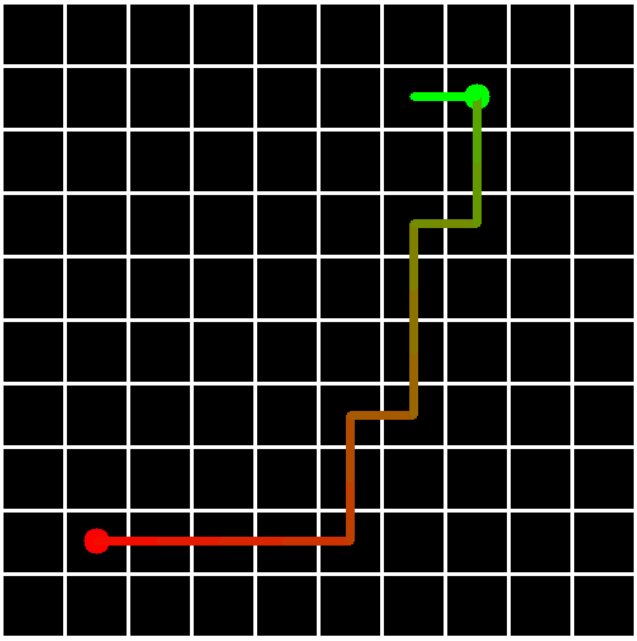

Figure 9: Log MSE loss against steps taken with the full Bellman equation and target network.