Reinforcement Learning Assessed Coursework 2

Name: Huazhen Xu

CID: 01918712

Department: Department of Computing

Course: MSc Artificial Intelligence

Question 1: Tuning the DQN– total 20 pts

Question 1.1: Hyperparameters– 10 pts

Label	Hyperparameter	Value
A	hidden_size	64
B	num_hidden	2
C	learning_rate	0.0005
D	replay_buffer_size	10000
E	num_episodes	600
F	epsilon • epsilon_start • epsilon_end • epsilon_decay_rate	Epsilon decay implemented instead, the original constant epsilon was not used • 1.0 • 0.1 • 0.995 Epsilon decay formula: epsilon = max(0.1, epsilon_start * (epsilon_decay_rate ** i_episode))
G	reward_scaling_factor	1
H	batch_size	256
I	target_update_frequency_steps	100

Note: For this implementation, I used gymnasium package instead of gym to resolve the issues caused by numpy versions.

For parameter F: epsilon, I implemented epsilon decay instead of using a constant epsilon value. The reason for this implementation is that with a constant epsilon, the agent continues to explore the environment even after lots of episodes. From those models, the average return either oscillate up and down the threshold of 100, or continue to increase without converging and resulting in network not learning meaningful policies for states with high cart velocities in second question.

Exploration schedule under my implemented epsilon decay:

epsilon_start = 1.0

epsilon_end = 0.1

epsilon_decay_rate = 0.995

epsilon = max(0.1, epsilon_start * (epsilon_decay_rate ** i_episode))

Initially, when the agent first starts to push the cart, epsilon is set to 1. Meaning that agent is always exploring the environment by taking random actions of pushing left or right.

Then, after each episode, reduce epsilon value with the decay rate of 0.995 and lower limit of 0.1, using this formula: epsilon = max(0.1, epsilon_start * (epsilon_decay_rate ** i_episode)). This allows the agent to gradually exploit greedy action instead of randomly exploring.

Question 1.2: Learning curve– 10 pts

Please see Figure 1 below for the learning curve of the trained DQN agent.

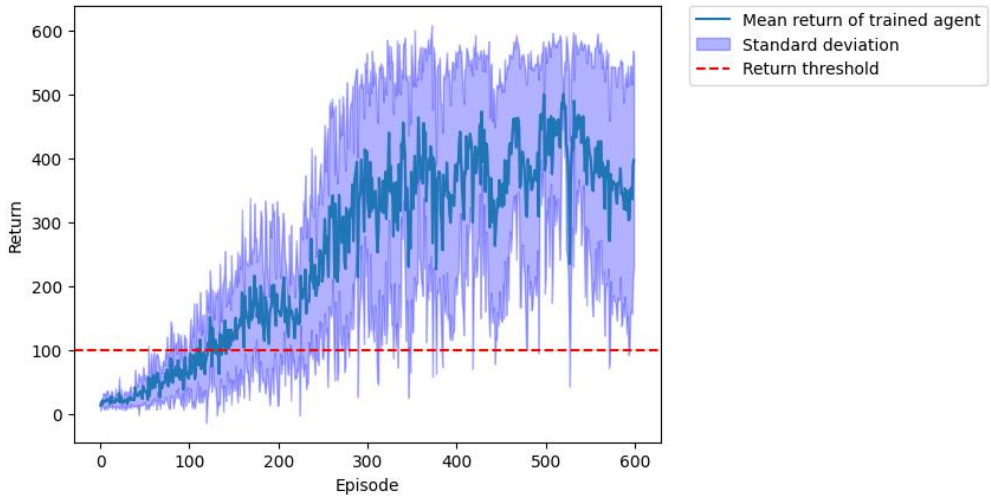


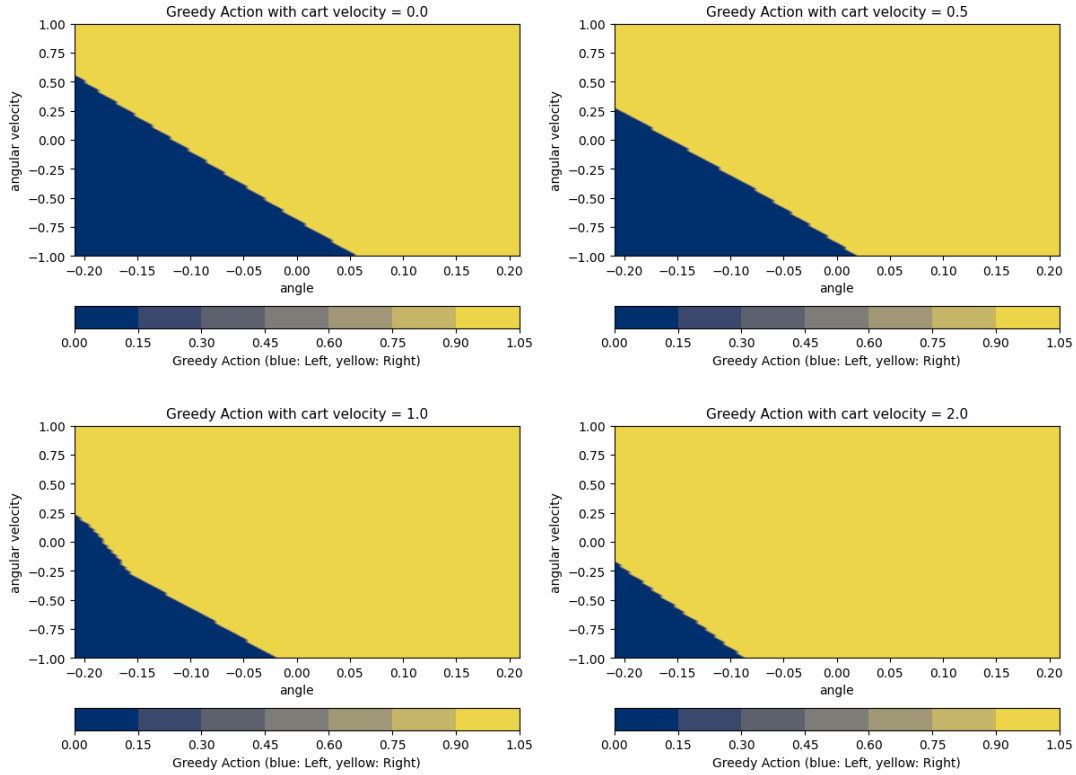
Figure 1: Learning curve of DQN agent plotted as return per episode over time

The chart shows the average return over 10 runs, where each run has 600 episodes. Initially the average return is quite low, which is expected as policy is randomly initialised. The trained DQN agent achieves average reward of 100 over 10 runs of training at around episode 140, and consistently stayed above the threshold for the rest of the episodes. The learning is successful. The average return also converged to around 400. This means that the number of steps that the cart can be pushed before the pole falls is around 400. The standard deviation increases with learning, this could be due to sometime agent fails to balance to pole.

Question 2: Visualise your DQN policy– total 20 pts

Question 2.1: Slices of the greedy policy action– 10 pts

(next page)



Please see the charts above for the greedy policy according to the trained DQN. **Blue region indicates that the cart is pushed to the left and yellow region to the right**. The graph plots assume cart position is fixed to zero (centre of the track) and illustrates situations under 4 different cart velocity: 0, 0.5, 1, 2. From the result, we can see that, under higher cart velocity, pushing the cart to the right is the greedy action for most time.

Comparing to the expectation of an optimal agent for the following four aspects:

• The regions of the plot where the agent chooses to push left or right:

- Expectation: for region where angle is negative (pole tilted to the left) and angular velocity is negative (the pole is quickly falling to left), an optimal agent will push the cart to the left to counter the action. This region should be painted blue. Where there is a positive angle and positive angular velocity, the cart should be pushed to the right, the region should be yellow.
- Performance of my agent meets the expectations. In particular, towards bottom left corner, the region is blue. Towards top right corner, the region is yellow.

• The general shape of the action decision boundary:

- Expectation: a perfect agent will have action decision boundary look like a diagonal line with negative slope through the (0,0) point.
- Performance of my agent roughly meets the expectation, slightly shifted to the left.

• The symmetries of the action decision boundary when the velocity is 0

- Expectation: when velocity is 0. The action is entirely dependent on the angle and

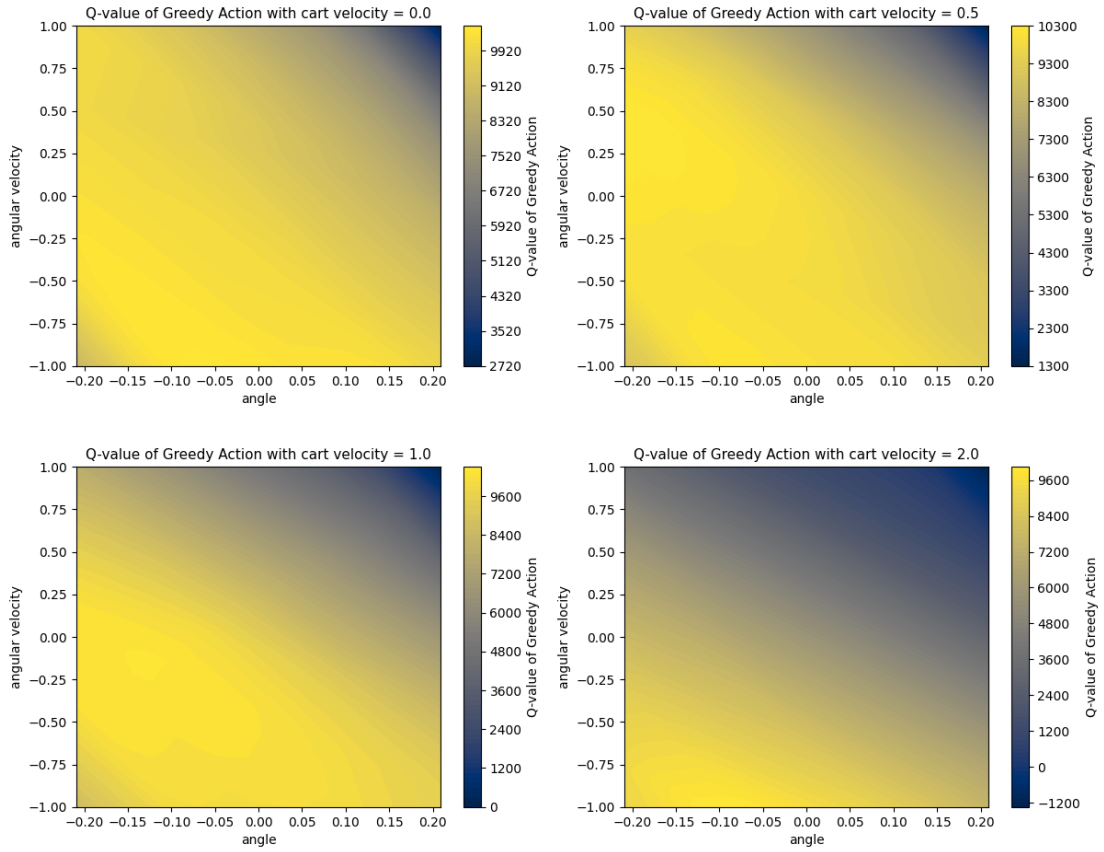
angular velocity. The chart will be symmetric with respect to the diagonal line through (0,0) and (-0.2094,-1).

- Performance of my agent roughly meets the expectations.

• How the action decision boundary shifts as velocity increases

- Expectation: as the cart velocity increases, the agent needs to push the cart earlier than before because the quicker the cart is moving, the larger the momentum and harder for agent to change the direction of the cart. Therefore, the boundary will shift down the shown graph. For example, if a pole currently has positive angular velocity and is tilted to right, then agent need to push the cart to the right to balance the pole back. If the cart velocity is high, then we need to start pushing cart towards right even when it is still largely tilted towards left. Reflecting on the graph, a point towards the lower left corner might be blue under low velocity, but will tend to be yellow under high velocity. i.e. the boundary shifted down.
- Performance of my agent meets the expectation.

Question 2.2: Slices of the Q function– 10 pts



Please see the charts above for the Q-values of greedy actions according to the trained DQN. Blue region indicates low Q-values and yellow indicates high Q-values. The graph plots assume cart position is fixed to zero (centre of the track) and illustrates situations under 4 different cart velocity: 0, 0.5, 1, 2.

Comparing to the expectation of an optimal agent for the following four aspects:

• The regions of the plot where values are relatively higher or lower

- Expectation: the Q-values are generally high when angle and angular velocity are close to zero (i.e. towards the centre of the graph). This is because under this state, the pole is unlikely to tilt to one side or fall at that moment. Q-values will be low on bottom left corner and top right corners. As the pole is most likely to fall under these conditions.
- Performance of my agent meets the expectation.

• The range of values your agent has learned, both close and far from the edge of the episode termination region

- Expectation: the max Q-values of an optimal agent are around 10000 when the agent is at the centre (i.e. far from edge of the episode termination region), as it is unlikely to fall. As it gets closer to the edge of termination, the Q-values drops to about 0 as the pole is most likely to fall to right or left.
- Performance of my agent meets the expectation. From the graphs, we can see that the max values between 9600~10300 occur around the centre, and the min values between -1200~2720 occurs at the corner, which are regions where the pole is most likely to fall.

• The symmetries of the learned values when the velocity is 0

- Expectation: the Q-values of a perfect agent under cart velocity = 0 will be perfectly symmetric with respect to the diagonal line through (0,0) with negative slope. This is because there is no external momentum from the cart movement. The stability under positive angle-angular velocity pair and negative angle-angular velocity pair should be the same
- Performance of my agent roughly meets the expectation. There are blue regions on top right and bottom left corners of the graph. The top right has slightly darker blue region, indicating more unstable states and lower Q-values.

• How the values change as velocity increases

- Expectation: as the cart velocity increases, the cart momentum will increase, making it harder for the agent to balance the pole. The boundary between yellow and blue will likely to shift down, meaning more states will be unstable and hard to balance. Therefore, there are more regions of low Q-values.
- The performance of my agent meets the expectation as blue region increased in size.