Reinforcement Learning Coursework 2 Report - Leili Barekatain

Question 1.1: Hyperparameters

Label	Hyperparameter	Value
A	Hidden Layer Size (number of neurons)	128
B	Number of Hidden Layers	2
C	Learning Rate	0.0002
D	Replay Buffer Size	10000
E	Number of Episodes	500
F	Epsilon Start Value	1.0
G	Reward Scale Factor	1.0
H	Batch Size (Minimum Buffer Size)	64
I	Target Network Update Frequency	100
eps_decay	Epsilon Decay Rate	1/E = 1 over number of episodes

Table 1: Hyperparameters Table

Exploration Schedule: I used epsilon decay in order to balance exploration and exploitation with a decay rate of 1/E, where E is the number of episodes as mentioned above. Specifically, in each training stage, I updated the epsilon using this:

$$F = \max(F*\text{eps\_decay}, 0.005)$$

which ensures that the agent explores more in the initial episodes, where learning is crucial, and gradually shifts towards exploiting the learned policy as training progresses. It also maintains a lower bound of 0.005, ensuring the agent continues exploring in later stages to avoid getting stuck in suboptimal policies.

Other Changes: I also used AdamW optimizer. I didn't modify the utils.py file.

Question 1.2: Learning curve

As it can be seen, early episodes exhibit low and unstable returns, reflecting the agent's initial exploration and lack of policy optimization. Beyond episode 200, the returns improve significantly, with an average above 100. This suggests the agent is learning to handle complex scenarios. Variability (high variance) after episode 200 suggests progress but not full stability yet.

Note: the return for each episode is equal to the episode length since the agent receives a reward of +1 for each step.

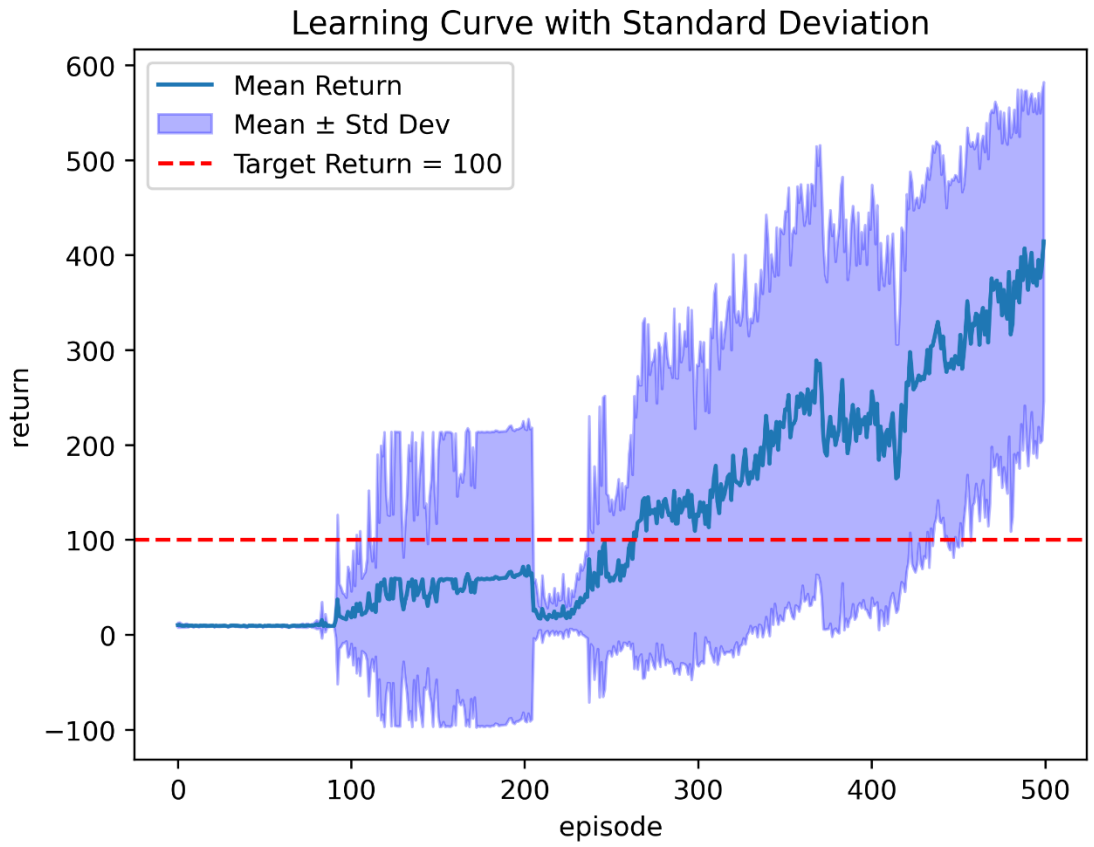

Figure 1: Learning curve of DQN agent plotted as return per episode over 10 runs showing average return, standard deviation, and our target return

Question 2.1: Slices of the greedy policy action

- The regions of the plot where the agent chooses to push left or right:
  I based my explanation for this part on the scenario where the cart velocity is zero. This is further explained in the fourth part when discussing velocity increases.
    - Optimal Agent: An optimal agent would normally decide this:
        - **Positive Pole Angle, Positive Pole Angular Velocity (top right in the plot):** In this case, the cart is pushed **right** in order to balance.
        - **Positive Pole Angle, Negative Pole Angular Velocity (bottom right in the plot):** In this case, it still pushes **right** to ensure the pole is not tilting for small angular velocity, but for large negative angular velocity, it pushes **left** to prevent overcorrection.
        - **Negative Pole Angle, Negative Pole Angular Velocity (bottom left in the plot):** In this case, the cart is pushed **left** to counteract the tilt, ensuring balance.
        - **Negative Pole Angle, Positive Pole Angular Velocity (bottom right in the plot):** In this case, it still pushes **left** to ensure the pole is not tilting for small angular velocity, but for large negative angular velocity, it pushes **right** to prevent overcorrection.
        - **Zero Pole Angle, Zero Pole Angular Velocity:** In this case, the actions are chosen with similar probability because the policy learns similar Q-values for both actions.
    - Our Agent: The behavior described above is also approximately reflected in the plot for our agent.
- General shape of the action boundary:
    - Optimal Agent: The decision boundary should be linear and diagonal, with a negative slope (as discussed above), reflecting the interaction between pole angle and angular velocity. More precisely, when both are in the same direction, the action counteracts them to restore balance. When they are in opposite directions, the action depends on the angular velocity: for small values, it aligns with the angle, and for large values, it counteracts the angle to prevent overcorrection. In other words, the agent considers both pole angle and angular velocity to reach the goal of stability.
    - Our Agent: The decision boundary is diagonal with a negative slope. Here, this relationship is piecewise linear, which is approximately linear.
- The symmetries of the action decision boundary when the cart velocity is zero:
    - Optimal agent: The decision boundary should be **perfectly symmetrical** around the origin (similar probability for action in both directions). This is due to the fact that environment is symmetrical, which means actions (e.g., push left or right) will have the same exact effect, but in the opposite direction.
    - Our Agent: It is **reasonably symmetrical** and well-aligned with expectations. It appropriately accounts for both pole angle and angular velocity, suggesting the agent has learned to act effectively in this stationary case.
- How the action decision boundary shifts as velocity increases:
    - Optimal Agent: First, pushing a cart moving at a constant velocity would have a similar impact to a cart with zero velocity. When the cart velocity increases, the decision boundary shifts to the left and down. This happens because when the cart has a

rightward movement, the goal is to prevent it from exceeding 2.4 from the center (which causes termination). However, pushing the cart immediately to the left might cause the pole to fall. Therefore, the optimal action is to first push it to the right to establish an initial angle, followed by a leftward push to balance the pole. This is why the immediate optimal action is to push right. In cases where the pole angle is highly negative and the angular velocity is also very negative, the pole might lean excessively to the left, requiring a leftward push to regain balance. At higher velocities (e.g., 2 m/s), an optimal agent would still predominantly push right but would adapt its policy to handle extreme cases (very negative pole angle and angular velocity) with leftward corrective actions when necessary.

- Our Agent: For my agent, at a cart velocity of 0.5 m/s and 1 m/s, the decision boundary also shifts to the left and down, showing that the agent has partially learned to adapt to the cart's small rightward velocity. The policy demonstrates reasonable corrective actions that are consistent with the behavior expected from an optimal agent. However, as the velocity increases further (e.g., 2 m/s), the agent always pushes right. Although this aligns with the need for an initial rightward push to maintain stability, it becomes a little suboptimal in cases where a leftward push is required to counteract extreme pole tilts or angular velocities (such as very negative pole angles and angular velocities). This limitation suggests that the agent has not fully generalized its policy for high-velocity scenarios, where a more finer balance is required.

Here, the greedy policy derived from the DQN (with hyperparameters defined in question 1) is visualized, with pole angular velocity on the y-axis and pole angle on the x-axis, while the cart position is fixed at zero and cart velocities are set to 0, 0.5, 1, and 2 across four separate plots:

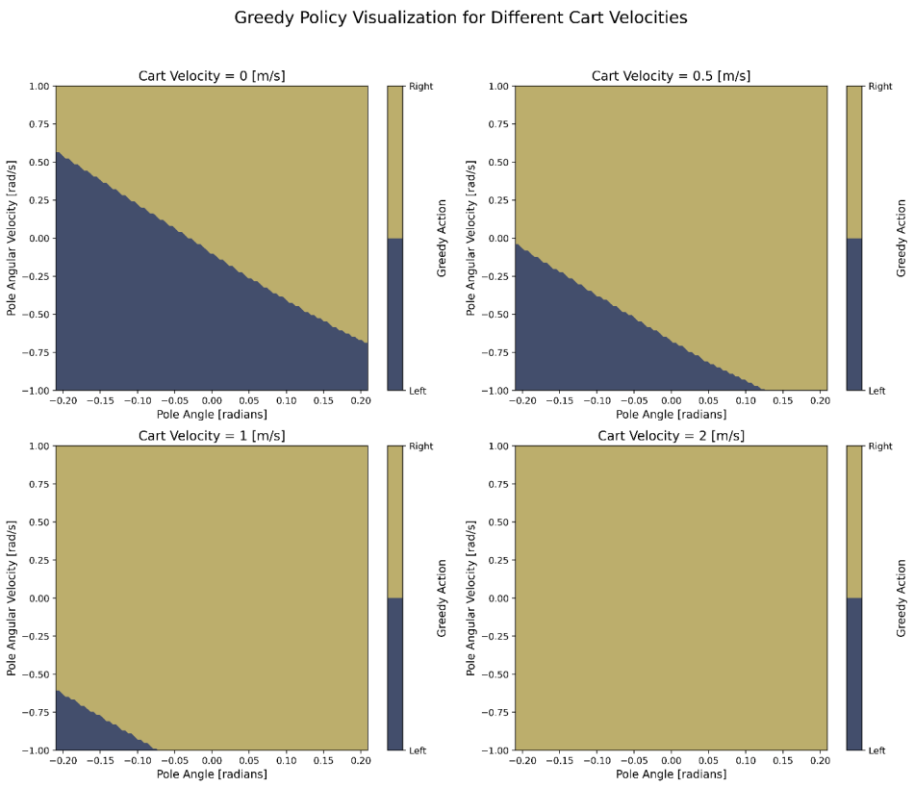

Figure 2: Greedy policy visualization of the DQN agent with respect to pole angle and pole angular velocity for four different cart velocities

Question 2.2: Slices of the Q function

Considering that Q-values represent expected cumulative rewards (not just the immediate reward),

- ### The regions of the plot where values are relatively higher or lower:

  I based my explanation for this part on the scenario where the cart velocity is zero. This is further explained in the fourth part when discussing velocity increases. An optimal agent would decide this when the cart velocity is zero:

  - Optimal Agent:
    - **Zero Pole Angle, Zero Pole Angular Velocity (the center):** In this case, the Q-values are the highest; this is because of the agent's confidence in maintaining stability, as balancing the pole is easiest when it is upright with minimal movement and Q-values are expected to be higher where the risk of termination is low.
    - **Non-zero pole angle and angular velocity (away from center):** In these cases, the Q-values are lower (specifically at the corners) because the agent expects lower rewards due to the increased difficulty of recovering balance. This is consistent with the environment dynamics, as extreme states (large pole angles and angular velocities in the same direction) are closer to the episode termination region. It is important to note that as long as the pole angular velocity and pole angle are in the opposite direction, if they even increase, the Q-value is still not low because the agent is self-correcting itself.

- Our Agent: The behavior described above is also approximately reflected in the plot for our agent.

- The range of values your agent has learned, both close and far from the edge of the episode termination region:
  - Optimal Agent: when the velocity is zero, the highest Q-values (can go up to very high values) are close to the center because of the upright and stable position of the pole and it is far from the terminal states. The lowest Q-values (can be close to zero) are close to corners because of large amount of pole angle and angular velocity in the same direction which is close to the terminal states. The Q-values are always non-negative because it represents the expected length of the episodes. The maximum Q-values should decrease as the cart's velocity increases due to the increased risk of termination and the greater challenge in recovering balance. For higher velocities the changes in range of the Q-values is explained further below in the fourth part.
  - Our Agent: when the velocity is zero, the range of the Q-values are between [284-644], with the highest values around the center and the lowest close to the corners, as we expect from an optimal agent. However, the lower bound for Q-values are overestimated for extreme cases which are close to termination. For higher velocities the changes in range of the Q-values is explained below in the fourth part.
- The symmetries of the learned values when the velocity is 0:
  - Optimal Agent: For an optimal agent, the Q-value distribution should be **perfectly symmetric around the origin**, reflecting equal dynamics in both directions when the cart has no speed. This is due to the fact that environment is symmetrical with respect to pole angle and angular velocity, which means actions (e.g., push left or right) will have the same exact effect, but in the opposite direction.
  - Our Agent: The Q-value distribution is **reasonably symmetric** around the origin. For example, the Q-values for positive and negative pole angles with equal angular velocities are similar, indicating that the agent treats these states equivalently, as expected for a stationary cart.
- How the values change as velocity increases:
  - Optimal Agent: When the cart velocity increases, **the plot shifts to the left and is no longer symmetrical**. This means that the Q-values for having a positive angle and positive angular velocity decreases, because these states are inherently riskier as they are closer to terminal conditions (the pole falling over). Even if the agent pushes right to stabilize in the short term, the likelihood of successful recovery from such states is lower. This increased risk of termination leads to lower expected cumulative rewards (Q-values) for these states. In contrast, for negative pole angles and angular velocities, the system is naturally moving toward a more stable configuration. The agent's actions which is explained in question 2.1 (e.g., pushing right first, then left) are more likely to succeed in these cases, leading to higher expected rewards and, consequently, higher Q-values. The Q-value reflects the long-term effectiveness of the agent's policy, not just the immediate action. At higher speeds, such as **2 m/s**, the Q-value distribution becomes **very skewed** and the Q-values for positive angles and angular velocities are lower because the cart’s momentum and pole instability make recovery harder, increasing the risk of termination. In this case, only very negative angles and very

negative angular velocities have high Q-values, because this only makes stabilization possible. **The maximum Q-values decreases** as the cart's velocity increases due to the increased risk of termination and the greater challenge in recovering balance.

- My Agent: The described behavior for optimal agent can also be seen for my agent. However, for higher speeds, the maximum Q-values are still overestimated for positive angles and angular velocities.

Here, the greedy Q-values learned by the DQN are visualized, with pole angular velocity on the y-axis and pole angle on the x-axis, while the cart position is fixed at zero and cart velocities are set to 0, 0.5, 1, and 2 across four separate plots:

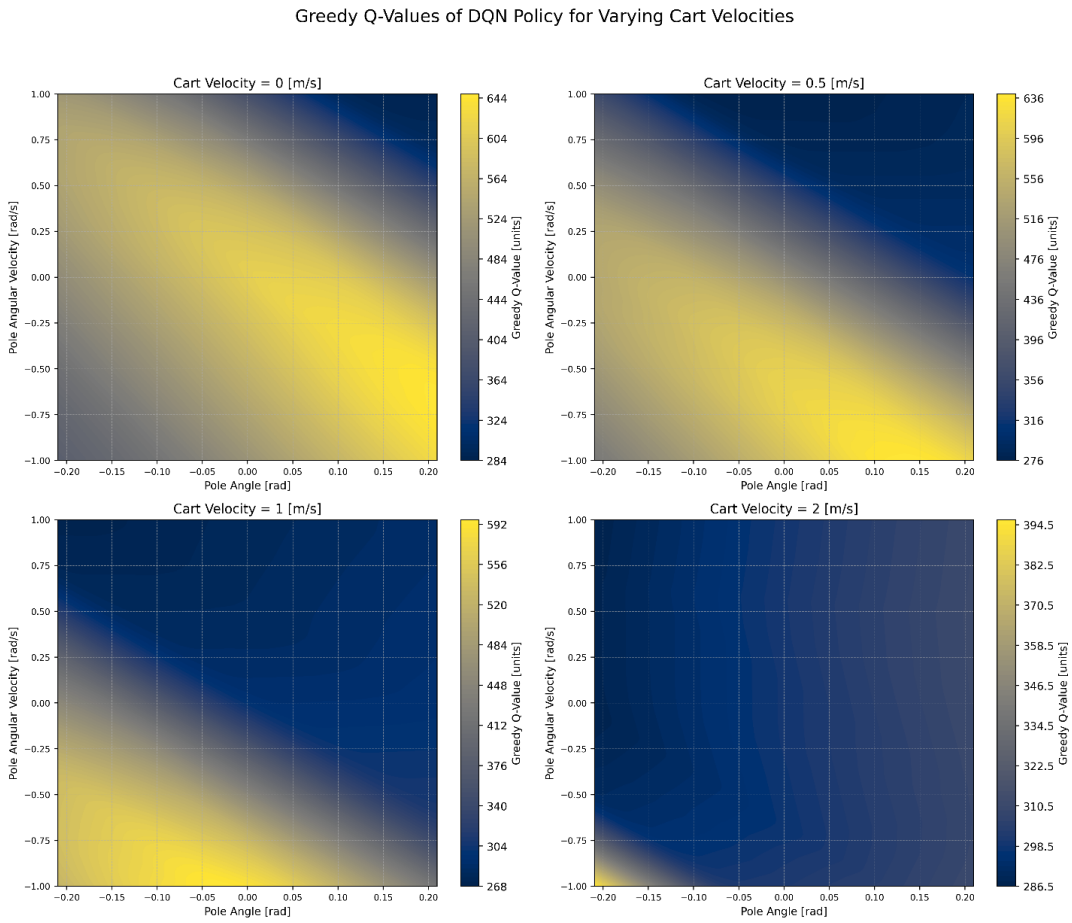

Figure 3: Greedy Q-values visualization of the DQN agent with respect to pole angle and pole angular velocity for four different cart velocities