DDA 4230: Reinforcement learning **CUHKSZ**

Assignment 3

TA: Sheng Xu

Due Date: Nov. 20th, 11:59 pm
Total points available: 100 pts.

Problem 1: Q-learning Computation 1 [20 pts.]

Consider the MDP with two states $S1$ and $S2$ represented in the figure below. In $S1$, the agent can perform either action a or action b. In $S2$, the agent chooses between actions c and action d. $S1$ and $S2$ are associated with rewards $R1 = -10$ and $R2 = 10$, respectively.

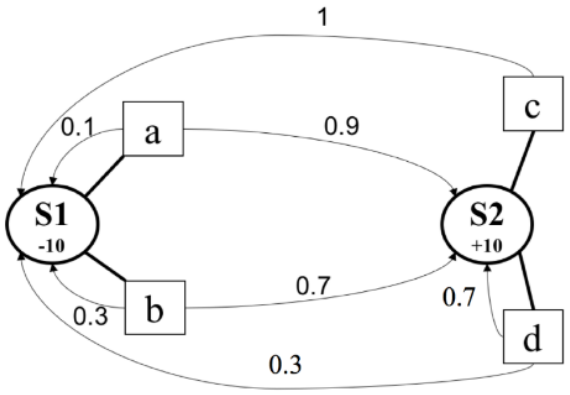

Figure 1: A simple MDP example for Problem 1.

Assume that the transition probabilities are unknown, and that the agent's first 3 transitions are given by the tuples $(S1, a, S2)$, $(S2, d, S2)$ and $(S2, c, S1)$. Apply the Q-learning algorithm for three iterations with the discount factor $\gamma = 0.9$ and learning rate $\alpha = 0.1$. Report a table with the Q-function computed at every iteration. What is the best policy obtained after these 3 iterations?

Problem 2: Q-learning Computation 2 [20 pts.]

Here is a simple MDP, where

- 4 states: $0, 1, 2, 3$
- 5 actions: $0, 1, 2, 3, 4$. Action $0 \leq i \leq 3$ goes to state i, while action 4 makes the agent stay in the same state.
- Rewards: Going to state i from states 0, 1, and 3 gives a reward $R(i)$, where $R(0) = 0.1, R(1) = -0.3, R(2) = 0.0, R(3) = -0.2$. If we start in state 2, then the rewards defind above are multiplied by -10. See the following table for the full transition and reward structure.
- One episode lasts 5 time steps (for a total of 5 actions) and always starts in state 0 (no rewards at the initial state).

State (s)	Action (a)	Next State (s')	Reward (R)
0	0	0	0.1
0	1	1	-0.3
0	2	2	0.0
0	3	3	-0.2
0	4	0	0.1
1	0	0	0.1
1	1	1	-0.3
1	2	2	0.0
1	3	3	-0.2
1	4	1	-0.3
2	0	0	-1.0
2	1	1	3.0
2	2	2	0.0
2	3	3	2.0
2	4	2	0.0
3	0	0	0.1
3	1	1	-0.3
3	2	2	0.0
3	3	3	-0.2
3	4	3	-0.2

What is the maximum sum of rewards that can be achieved in a single trajectory in the test environment, assuming $\gamma = 1$? Show first that this value is attainable in a single trajectory, and then briefly argue why no other trajectory can achieve greater cumulative reward.

Problem 3: Deep Q Network Implementation [60 pts.]

In this problem, we try to implement Deep Q Network in a simple environment CartPole-v1. Code skeleton is given at here. Please fill in the code where a "pass" function exists, and answer the following questions:

1. Explain your logic of the code in line 367. Why is epsilon essential?

2. When implementing the QNetwork agent, you can try different structures of neural networks, also, you can use different learning rates, batch sizes, or memory size parameters. In particular, the loss might oscillate and you have to find a good criterium. Choose your parameters with explanations (should include both intuitive and experimental reasons after some attempts), and fix them in the next two problems.

3. Produce a training graph where the x-axis indicates the steps (more than 300k) and the y-axis indicates the episodic return (already recorded in the info), with mean and std results over random seeds 1, 123, and 321.

4. Produce an evaluating graph where the x-axis indicates the steps (more than 300k) and the y-axis indicates the episodic return (already recorded in the info), with random seed 666.

If you feel hard to complete the task, you can go to this repo cleanrl-dqn for reference.