Robot Learning Homework 3

Matteo Zulian s310384
Automation and Intelligent Cyber-Physical Systems
Politecnico di Torino

This exercise asks to implement Q-learning algorithm to a Cartpole System provided by gym library.

1 Introduction to Tabular Q-learning

Q-learning is a model-free reinforcement learning algorithm that is used to learn the optimal action-selection policy for a given finite Markov decision process (MDP). In this algorithm, the learned action-value function, Q, directly approximates q^*, the optimal action-value function, independent of the policy being followed. This dramatically simplifies the analysis of the algorithm and enables early convergence proofs.

The learned Q-values can be used to extract an optimal policy. The action with the highest Q-value in each state is chosen as the optimal action At each time step, the agent observes the current state s, selects an action a based on its exploration-exploitation strategy (commonly using epsilon-greedy), and takes the chosen action. Then Q is updated for the current state-action pair using the Q-learning update rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \cdot \Big(R + \gamma \cdot \max_a Q(s',a) - Q(s,a)\Big) \tag{1}$$

2 Exploration and Exploitation

The exploration-exploitation tradeoff is a fundamental concept in reinforcement learning and decision-making in general. It refers to the dilemma faced by an agent when deciding whether to explore new possibilities (exploration) or exploit known information to maximize immediate rewards (exploitation). Striking the right balance between exploration and exploitation is crucial for effective learning and decision-making in uncertain environments. Several strategies are commonly used in reinforcement learning. This exercise asked to evaluate an **Epsilon-Greedy Strategy** with different behaviours of ϵ during time.

2.1 Epsilon-Greedy with constant Epsilon ϵ

The epsilon-greedy strategy is a common approach used in reinforcement learning to balance exploration and exploitation. The agent chooses a random action (exploration), with probability ϵ, while with probability 1-ϵ, it chooses the action with the highest estimated value (exploitation)

With a constant epsilon, the exploration rate remains the same throughout the learning process. This means that a fixed percentage of the agent's actions will be exploratory, and the rest will be exploitative. It may be crucial in the early stages of learning when the agent has limited knowledge of the environment. However, as the agent gains more experience, a fixed exploration rate may become less necessary.

2.2 GLIE ϵ_k

The GLIE (Greedy in the Limit with Infinite Exploration) approach is a concept used in reinforcement learning, particularly in the context of ensuring that an agent explores sufficiently while also converging to an optimal policy. The key idea behind GLIE is to gradually reduce the exploration rate over time as the agent accumulates more experience, ensuring that the agent explores the environment thoroughly in the early stages of learning. This is crucial for discovering the dynamics of the environment and avoiding premature convergence to suboptimal policies.

2.3 Results

The epsilons evaluated in this exercise are :

- fixed rate $\epsilon = 0.2$
- GLIE $\epsilon_k = \frac{b}{b+k}$, with parameter $b = 2222$ (tuned in order to obtain $\epsilon_k = 0.1$ when it reaches $k = 20000$ episodes)

Plotting the average return value obtained at each episode shows little differences in the overall behaviour of the two approaches. However it can be seen that, in the beginning, GLIE has better results and a smoother line, while the fixed ϵ has a steep curve around episodes 2500-4000. This is due to a higher ϵ for GLIE in the first thousands steps of the algorithm (as shown in figure 1 $\epsilon_{0-5000} > 0.2$) that allows it to explore more and faster. When the constant one has explored enough, in it's own way, then it is able to use what it has learned to fill the gap between the two curves around episode 5000, because at that point it can use the Exploitation phase with higher probability than GLIE. Also it shows that fixed ϵ has a higher variance than GLIE especially in the last episodes where the algorithm is expected to be optimal, this is because sometimes it happens that the cart is in a state that hasn't been explored yet so it doesn't know what is the best action to take.

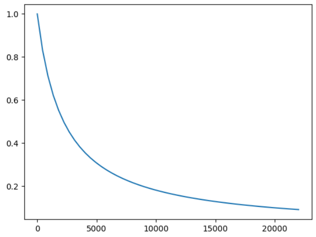


Figure 1: ϵ_k over time

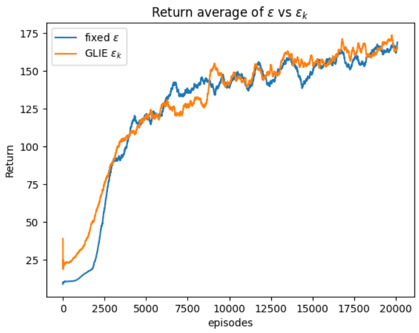
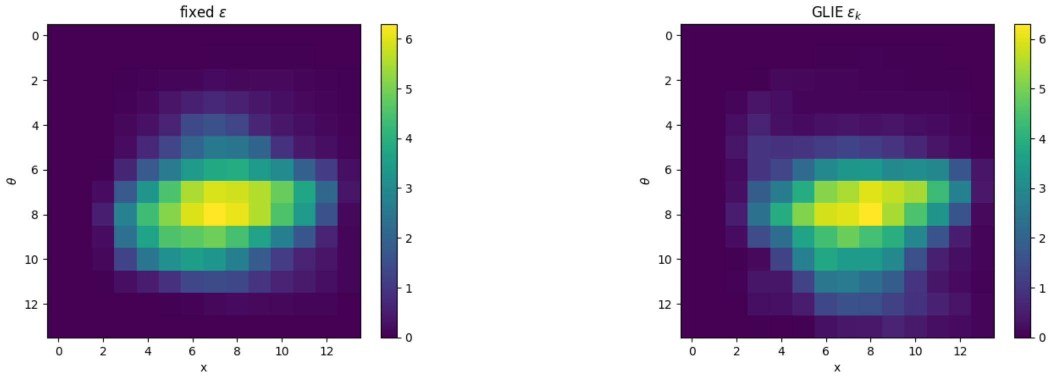


Figure 2: Performance of the two strategies proposed

Additionally it can be shown the heatmap plot of the value function $Q(s,a)$ of cart's position x and angle of the pole θ, while $\dot{x}$ and $\dot{\theta}$ are averaged and the value of the action maximized. These plots show as expected a concentration of value around the state $x = 0, \theta = 0$ which is the optimal state for the Cartpole problem. With a closer look it can be noticed that GLIE solution has explored more states while the fixed one shows a more restricted circle, this is due to the increased initial randomness of GLIE in the beginning.



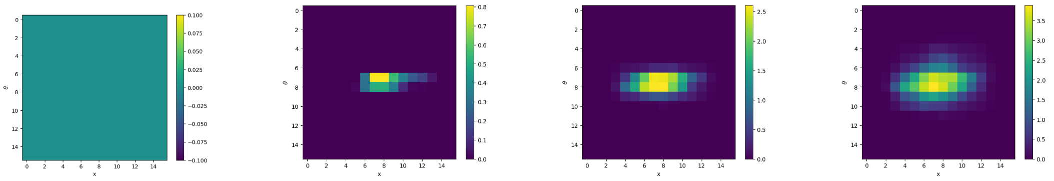
(a) fixed ϵ

(b) GLIE ϵ_k

Figure 3: Heatmaps of the two strategies

2.4 Heatmap evolution

the Q-value starts from with all zeros, then while the agent explores some of the possible state configurations, it is gradually updated. As seen in figure (4) the evolution spreads around the centre of the heatmap where the states are $x = 0$ and $\theta = 0$ as the cart drifts away from the centre of the screen.



(a) Q initial value

(b) ep 2000

(c) ep 5000

(d) ep 10000

Figure 4: Heatmaps of the two strategies

3 Optimistic Initial Values

The idea behind optimistic initial values is to initialize the estimated values (Q-values) with values that are higher than the true expected values. This encourages the agent to explore more and helps in overcoming local optima during exploration-exploitation trade-off because since the beginning every state is considered equally optimal. What happens during the simulation is the opposite as normal, in fact at each step the Q-value $Q(s, a)$ is decreased, accordingly to the formula (1). Over time the agent explores and the value function is modified until the effects of the initial values are compensated.

3.1 Results

In this exercise were analysed two cases:

- Q initialized at zero
- Q initialized at fifty

Both of those cases were solved with $\epsilon = 0$ forcing an Exploitation-only approach.

However if Q is all zeros, all actions are considered wrong, this combined with a lack of exploration cause problems in the learning process.

The plot of the average return (figure 5) shows that both strategies don't converge to an optimal solution and the overall performances are worst than what showed before (figure 2). In particular the solution with Q initialized at zero is highly penalized by the lack of exploration and doesn't seem to have learned at all.

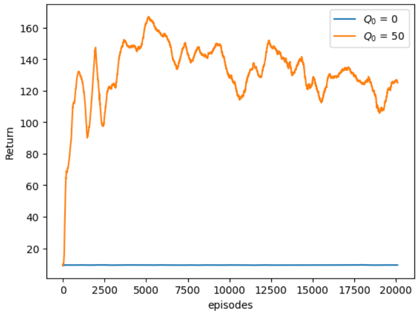
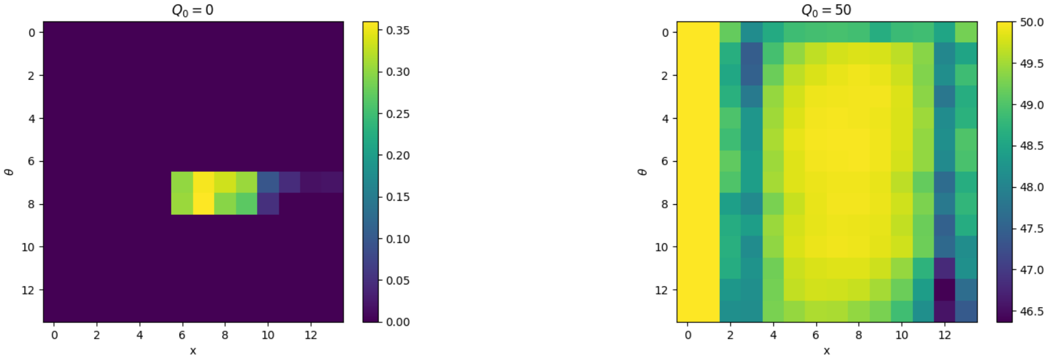


Figure 5: Performance of the two strategies proposed

The heatmap on the left shows that when the agent is not allowed to explore, the Q-value function doesn't improve and remains at zero for the most part. While on the right it can be see the effects of the decreasing values during exploration phase which is sort of carving out the heatmap around the centre. (the yellow band in the left part just indicate that those states weren't explored)

In conclusion the choice of initial values is very important and has a significant impact on the performances of the algorithm. The ideal is to choose the values close to the optimal values in order to minimize the time that the algorithm will need to decrease the values to reach the optimal.



(a) $Q_0 = 0$

(b) $Q_0 = 50$

Figure 6: Heatmaps of the two strategies

4 Q-learning in continuous spaces

4.1 State discretization

In Q-learning, discretizing states refers to the process of converting continuous state spaces into discrete states allowing updating and maintaining of Q-values for each state-action pair. In continuous state spaces, there could be an infinite number of possible states, making it computationally infeasible to represent and update Q-values for all of them. Discretization reduces the state space to a finite set, making it more manageable and efficient.

Cartpole system has a continuous state-space, so it was discretized in a 4-D Tensor with 16 possible configuration for each state. The possible actions of the Cartpole are 'left' and 'right' so in order to map the relation between each possible state configuration and the value of the two possible actions, Q is a 5-D tensor in the form :

$$Q(s,a) : (s^{\{16\times16\times16\times16\}}, a^{\{2\}}) \longrightarrow Value \tag{2}$$

4.2 Continuous state-space

In Q-learning, the traditional tabular approach assumes a discrete state space, which can be limiting in scenarios where the state space is continuous. However, several techniques and extensions have been developed to apply Q-learning to continuous state spaces. One popular approach is to use **function approximation**.

function approximation involves using a parameterized function to estimate the Q-values, typically implemented using a neural network or other function approximators. This allows the agent to generalize its knowledge across similar states.

4.3 Continuous action-space

When dealing with a continuous action space in Q-learning, similar challenges arise as in the case of continuous state spaces. Traditional Q-learning is designed for discrete action spaces, where the agent can choose from a finite set of actions in each state, while dealing with infinite possibilities increases the difficulties of action selection. As before one solution might be to discretize the action space to avoid the problem, however, with large state spaces it is more common to use some form of function approximation for the action value.

These solutions are: Deep Q-Networks (DQN), Actor-Critic Methods and Deterministic Policy Gradients (DPG).