Robot Learning Homework 3

Carena Simone, s309521
s309521@studenti.polito.it

1 Introduction

The purpose of this assignment is to develop a reinforcement learning algorithm to control the *cartpole* system. In particular, a *Q-Learning* algorithm is implemented, using an ϵ-greedy policy for the behavior policy. Different configurations and settings are tested throughout this experiment. First a comparison concerning the choice of the ϵ value is presented, analyzing the results given by a constant ϵ in contrast to those given by a GLIE approach. Then an observation underlying the effects of the initial value of the Q-function (having $\epsilon = 0$) is performed. The system states are discretized into a finite grid, with each state having dimension $|\mathcal{X}| = 16$ and the action space has dimension 2. The resulting Q grid matrix will then be of shape `(16,16,16,16,2)`.

2 Effect of Different Choices of ϵ

In the implementation of the Q-Learning algorithm, a ϵ-greedy behavior policy is adopted. This approach is used to enforce exploration. In particular, the action to take is chosen greedily, i.e. the one yielding the best value of the Q-function, with probability $1-\epsilon$ and randomly with probability ϵ

$$\pi(A_t = a|S_t = s) = \begin{cases} \frac{\epsilon}{|\mathcal{A}|} + (1-\epsilon) & \text{if } a^* = \arg\max_{a\in\mathcal{A}} Q(s,a) \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases}$$

where $\mathcal{A}$ is the set of all possible actions. The Q-Learning algorithm is an off-policy RL method, used to approximate the optimal action-value function q_*, to which the Q-function converges. The Q-learning update is defined as follows:

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_{a\in\mathcal{A}} Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

The action A_t is chosen using an ϵ-greedy policy, which allows the agent to explore, while the successive action is chosen to maximize the action-value function in the next step.

Since the behavior policy determines which action is to be taken in the current state S_t, and since it's defined by the parameter ϵ, a variation of ϵ determines how much the agent tends to explore in contrast to exploiting the current knowledge it possesses: the higher the value of ϵ, the more likely the agent is to take a random action from state S_t, the lower is ϵ, the more likely it is to take the action it know to be the best up to that point.

2.1 Constant ϵ

A first observation can be made by choosing ϵ to be constant. In particular a value of $\epsilon = 0.2$ has been taken.

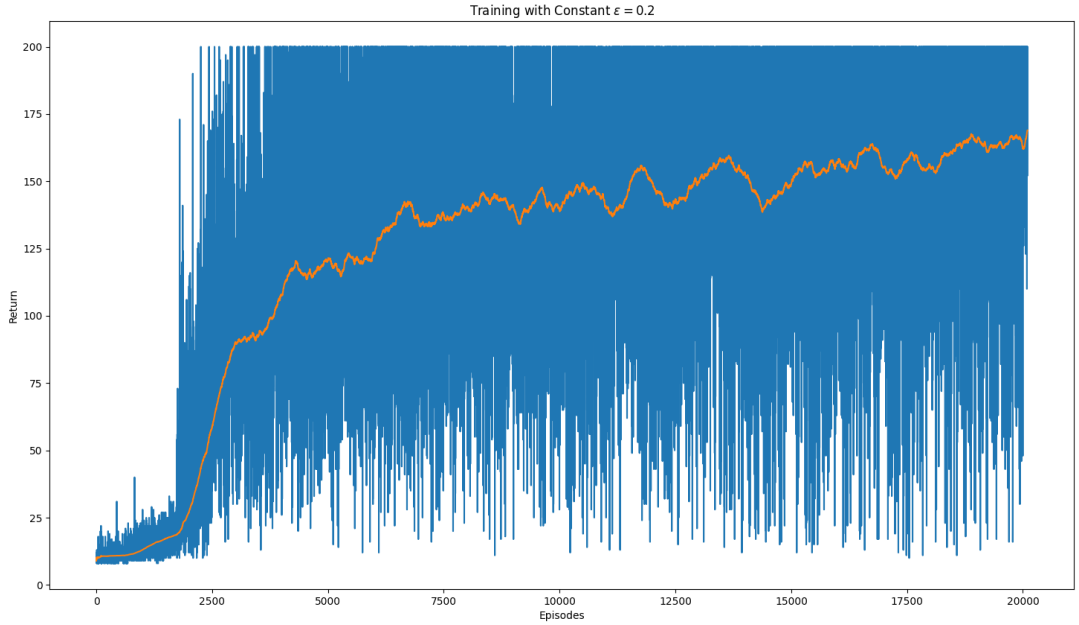


Figure 1: Training Return with Constant ϵ

As it possible to see from the plot in figure (1), the return averaged over the last 500 steps of each episode (the orange line), is increasing, meaning that the agent is improving its policy. This is due to the fact that since the agent is able to explore the it can keep improving its policy and not getting stuck in sub-optimal policies, as the number of episodes increases.

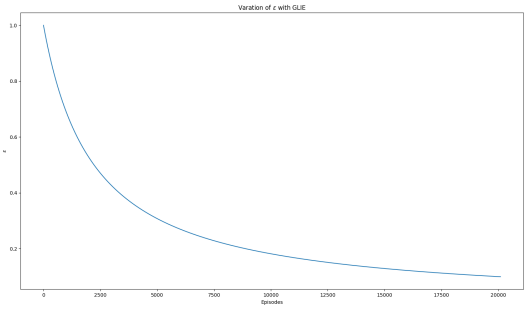
2.2 GLIE ϵ

An alternative to choosing a constant epsilon is to reduce it over time with GLIE (*Greedy in the Limit with Infinite Exploration*). In this case the value of ϵ decreases every episode as described by

$$\epsilon_k = \frac{b}{b+k}$$

Where b is a parameter that in this case is chosen equal to 2222 so that $\{\epsilon_k\}$ converges to 0.1 after 20000 episodes.

Training Return with GLIE ϵ



Evolution of ϵ_k Over the Episodes

The evolution of the average training return is similar to the one of (1), with the main difference that the GLIE ϵ yields an higher return at the beginning. This is due to the fact that, as it possible to see from the evolution of ϵ_k, at the beginning the probability of taking a random action, and

thus exploring, is higher. In particular in the first 2500 episodes the GLIE ϵ is greater than 0.5, meaning that the agent will take a random action instead of the best one with probability 1/2. After approximately 2500 episodes the two methods behave very similarly.

3 Value Function Evaluation

Once the optimal q_* has been computed, the optimal state-value function ν_* can be derived as

$$\nu_*(s) = \max_{a \in \mathcal{A}} q_*(s, a)$$

It is then possible to plot the heatmap of the computed state-value function in terms of x and θ, averaged over $\dot{x}$ and $\dot{\theta}$.

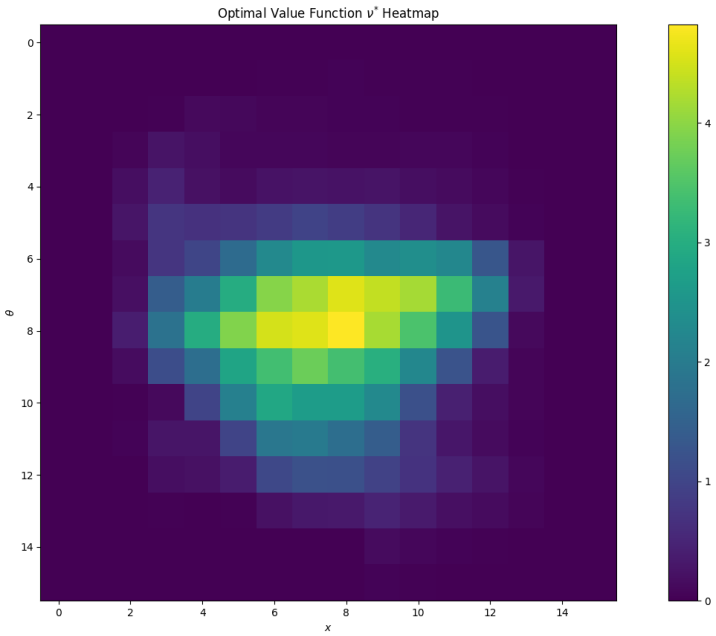


Figure 2: ν_* Heatmap with respect to x and θ

The value function describes which state sit is best to be in. In particular, regarding figure (2), it is possible to see that the states that yield the most value are the ones closer to $x = 0$ and $\theta = 0$. This happens because states around $(0, 0)$ are closer to the desired behavior of having the cartpole system stabilized.

The heatmap presented in (2) is the one attained at the end of the 20000 episodes. If we consider the heatmap before the training, it would be monochromatic, since at the beginning Q is initialized as $\mathbf{0}^{16,16,16,16,2}$ and thus every action would be equally attain the maximum of $Q(s, a)$ over a. Intermediate steps would show a map converging to the final one of figure (2).

4 Effect of the Initial Value of Q

All of the considerations above were made considering the fact that the initial value of the Q grid matrix was $\mathbf{0}^{16,16,16,16,2}$. Other initial values of the Q functions can be used. Choosing a starting grid with values greater than 0 can encourage the agent to explore, since all around itself it will see action-values pairs yielding great value and thus it will visit them all several times before converging, decreasing the values each episode. To better appreciate this behavior an $\epsilon = 0$ is set in the Q-Learning algorithm. This has the effect of having the agent always choosing the action yielding the best values and not trying randomly exploring them from time to time.

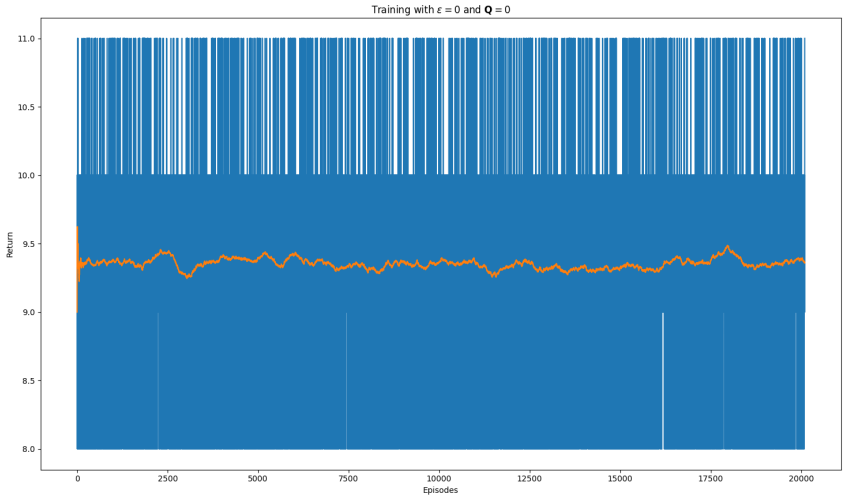


Figure 3: Training Return with $\epsilon = 0$ and $Q = 0$

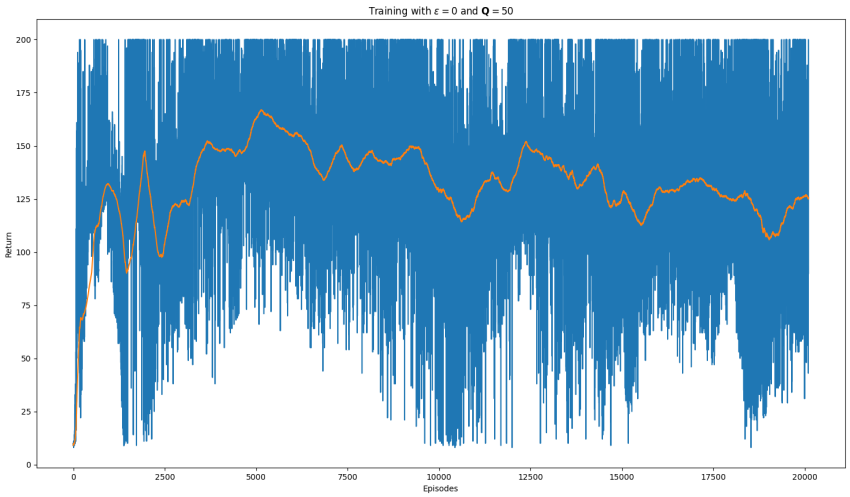


Figure 4: Training Return with $\epsilon = 0$ and $Q = 50$

Figure (3) and figure (4) show the behavior of the agent when having $\epsilon = 0$ and, respectively, $Q = 0$ and $Q = 50$. As it is possible to see, when $Q = 0$, and $\epsilon = 0$ the agents performs poorly, since it has no incentive to explore and all actions are equally low-value. When we consider $Q = 50$, the performance is better. Though the trend is not strictly increasing as for $\epsilon = 0.2$ and GLIE ϵ, having a Q-function comprised of states yielding high values enforces the agent to explore, and thus improve its policy.

5 Q-Learning in Continuous State Spaces and Action Spaces

5.1 Continuous State Spaces

With continuous state space it is possible to use the Q-Learning method, given that a function approximation method for the action-value function is used. Functions approximations approaches comprise methods like linear function approximation and neural network function approximation.

5.2 Continuous Action Spaces

The problem of applying Q-Learning with a continuous action space comes from the policy improvement step of the algorithm

$$Q(S_t, A_t) \longleftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(S_{t+1}, a) - Q(S_t, A_t) \right]$$

The $\max_{a \in \mathcal{A}}$ operation becomes unfeasible as the size of the action space increases. The best course of action is then to discretize the action space.