CSE 4/546: Reinforcement Learning

G. Laxmi Devi (50400730)

1. Advantage Actor-Critic (A2C)

Actor-Critic Algorithms use both Policy-based and Value-based algorithms together. Policy-based agents learn a policy mapping states to actions using the probability distribution of actions. Value-based algorithms are used to learn to select actions based on the predicted value. This algorithm has two networks, actor and critic. The actor chooses an action, and the critic evaluates the value of a state. As the critic learns which states are bad, the actor avoids the bad states.

In Advantage Actor-Critic we replace the value function with the advantage function to avoid high variability.

$$A(s_t, a_t) = Q(s_t, a_t) - V(a_t)$$

This function calculates the extra reward for the agent if chosen an action by subtracting the Q value for an action in a state and the average value of that state.

2. Difference between the actor-critic and value-based approximation algorithms

- **Value-based**: estimate value function or Q-function of the current policy. (No explicit policy)
- **Actor-critic**: estimate the value function or Q-function of the current policy and use it to improve the policy.

3. Briefly describe THREE environments

1. Grid Environment

The environment depicts “A rabbit reaching a bunch of carrots avoiding/passing through hurdles and traps and collecting carrots in the grid environment”. The rabbit on its way if it falls into a trap /hurdle gets a negative reward and similarly, if it collects a carrot on its way gets a positive reward.

The main objective is that the rabbit should reach the bunch of carrots for its family avoiding the traps and hurdles and gaining a high cumulative reward.

1.1 Actions

A ϵ {up, down, right, left}

1.2 States

The environment is a **4*4** grid that has **16** states. The rabbit starts from the position [0,0] and the goal (a bunch of carrots) is at [3,3].

1.3 Rewards

- Goal (a bunch of carrots) at [3,3] = +6
- Carrots at [1,3] and [2,0] = +2 each
- Hurdle at [3,2] = -2
- Trap at [1,1] = -2
- Other positions = 0

2. CartPole-v1

The environment depicts “A cartpole agent where the cart is trying to balance the pole vertically, with a little shift of the angle. **The goal is to prevent it from falling over** (to keep the pole upright for as long as possible).”

2.1 Actions

A ϵ {0,1}
0 - left
1- right

2.2 States

There are four values representing the state: cart position, cart-velocity, pole angle, and pole velocity respectively.

	Min	Max
• Cart Position	-4.8	4.8
• Cart Velocity	-Inf	Inf
• Pole Angle	-0.418 rad (-24°)	0.418 rad (24°)
• Pole Angular Velocity	-Inf	Inf

The episode terminates if the cart position leaves the (-2.4, 2.4) range. The episode terminates if the pole angle is not in the range (-.2095, .2095) (or **±12°**).

2.3 Rewards

There is **only one** unique reward in the environment.

R ϵ {+1}

The reward for every step taken is +1, including the termination step. The threshold for rewards is 475 for v1.

3. LunarLander-v2

This environment is a classic rocket trajectory optimization problem. According to Pontryagin’s maximum principle, it is optimal to fire the engine at full throttle or turn it off.

3.1 Actions

A ϵ {0,1,2,3}

There are four discrete actions available: do nothing, fire the left orientation engine, fire the main engine, fire the right orientation engine.

3.2 States

There are 8 states: the coordinates of the lander in x & y, its linear velocities in x & y, its angle, its angular velocity, and two booleans that represent whether each leg is in contact with the ground or not.

The episode finishes if:

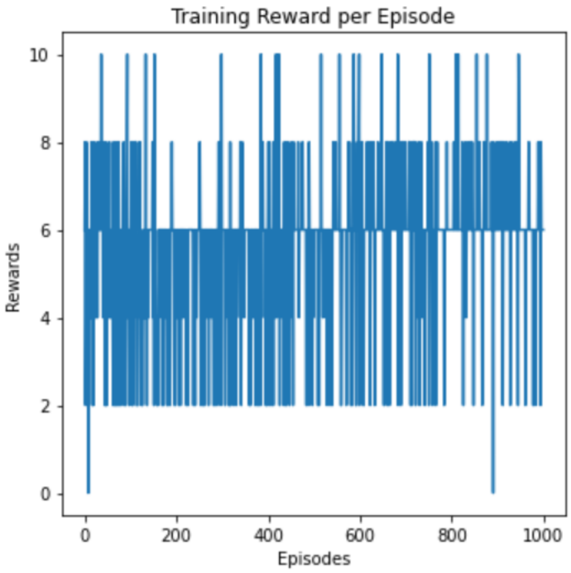
- The lander crashes (the lander body gets in contact with the moon)
- The lander gets outside of the viewport (x coordinate is greater than 1)
- The lander is not awake.

3.3 Rewards

- Reward for moving from the top of the screen to the landing pad and coming to rest is about 100-140 points.
- If the lander moves away from the landing pad, it loses reward.
- If the lander crashes, it receives an additional -100 points.
- If it comes to rest, it receives an additional +100 points.
- Each leg with ground contact is +10 points.
- Firing the main engine is -0.3 points each frame.
- Firing the side engine is -0.03 points each frame.
- Solved is 200 points.

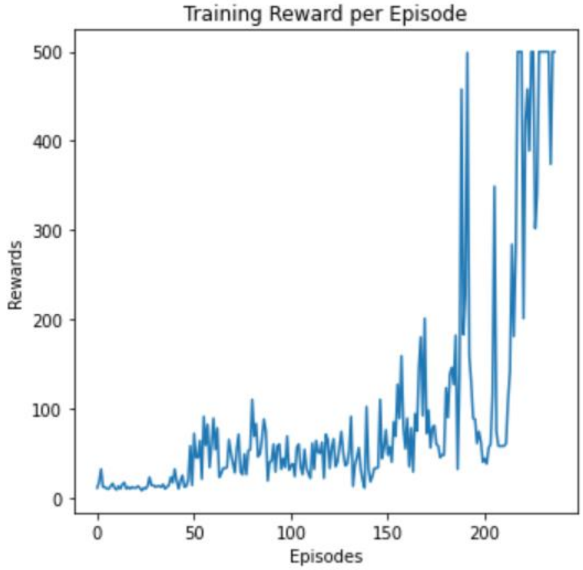
4. Training Results

a. Grid Environment



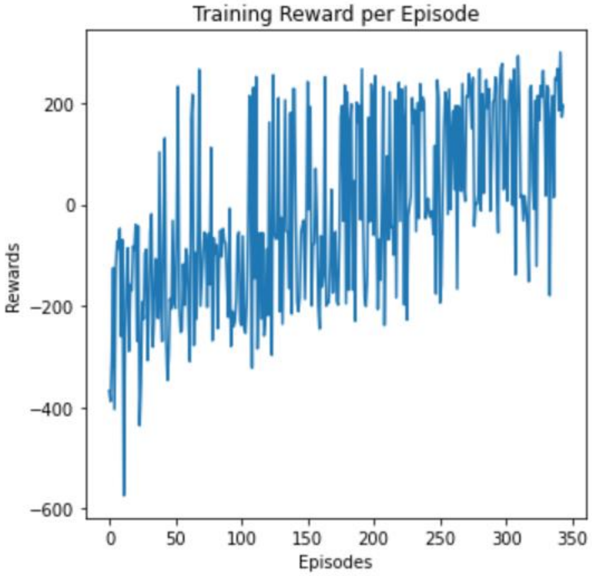
From the above plot, we can see that the agent in the first few episodes the cumulative reward is 2 or 4, as the episodes increase the cumulative rewards increase and reaches the goal in most of the episodes than the prior episodes learning the optimal policy.

b. Cartpole



From the above plot, we can see that as the episodes increase the cartpole balances for more steps, in the later episodes it reaches the average reward of 500.

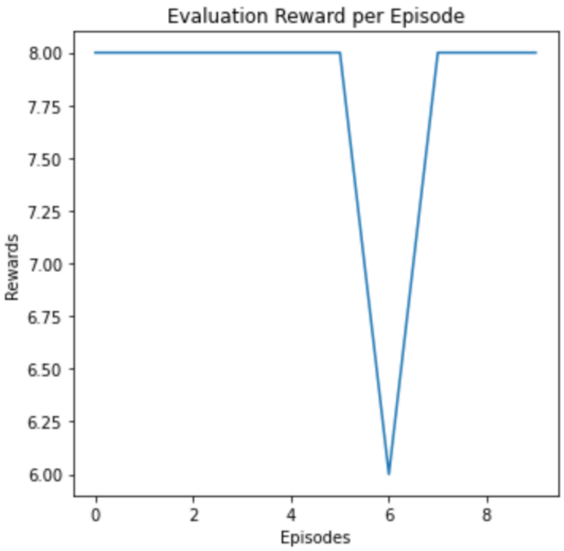
c. LunarLander



Though the cumulative reward is negative and random in the first few episodes , as the episodes increase the agent reaches the goal with a cumlative reward greater than 200.

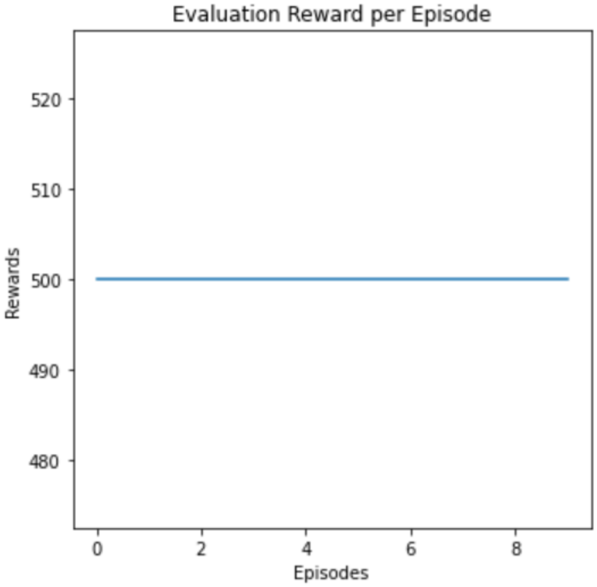
5. Evaluation Results

a. Grid Environment



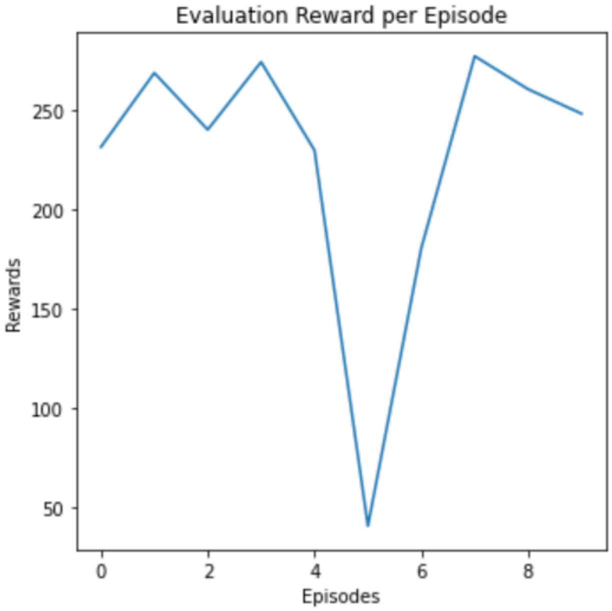
In the above ten episodes, the agent reaches the goal selecting good actions and reaching the goal. In one of the episodes the agent reaching the goal without collecting any small path and chooses an optimal path.

b. Cartpole



The agent selects good actions reaching the goal with a cummulative reward of 500 in each episode.

c. LunarLander



The agent selects good actions reaching the goal with a cummulative reward greater than 200 in each episode part from one episode.