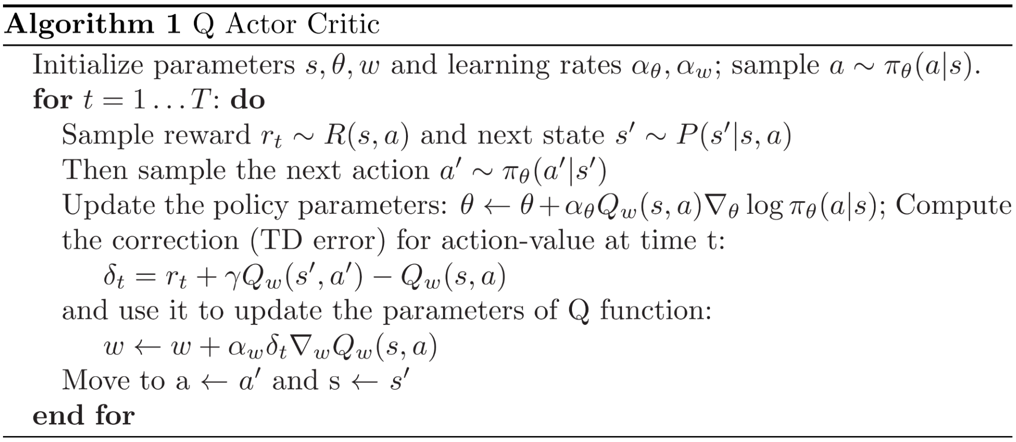
# ASSIGNMENT 3
# TEAM 30

1. The algorithm we have used to solve the environment is the Q Actor-Critic Algorithm. Here, the Q value can be learned by parameterizing the Q function using a neural network. The Critic updates the Q value while the Actor updates the policy distribution suggested by the Critic. At every update, both Critic network and Value network is updated.



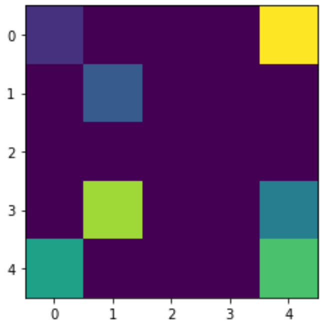
```
Algorithm 1 Q Actor Critic
  Initialize parameters $s, \theta, w$ and learning rates $\alpha_\theta, \alpha_w$; sample $a \sim \pi_\theta(a|s)$.
  for $t = 1 \ldots T$: do
    Sample reward $r_t \sim R(s, a)$ and next state $s' \sim P(s'|s, a)$
    Then sample the next action $a' \sim \pi_\theta(a'|s')$
    Update the policy parameters: $\theta \leftarrow \theta + \alpha_\theta Q_w(s, a) \nabla_\theta \log \pi_\theta(a|s)$; Compute
    the correction (TD error) for action-value at time t:
      $\delta_t = r_t + \gamma Q_w(s', a') - Q_w(s, a)$
    and use it to update the parameters of Q function:
      $w \leftarrow w + \alpha_w \delta_t \nabla_w Q_w(s, a)$
    Move to a $\leftarrow a'$ and s $\leftarrow s'$
  end for
```

2. Value based approximation algorithms are unbiased offline algorithms, that uses Monte Carlo estimates of the gradient after completing an episode. Variants of Actor-Critic algorithms are based on online algorithm that updates every step using predictions of future return.

3. In Grid-World,

   Actions : {0, 1, 2, 3}

   States : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24}
   Rewards : {0, 10, 8, -5, -3}
   Goal : To reach the goal state while collecting maximum rewards.



In ‘CartPole-v1’,

- Action : {-1, +1}
- Possible states : Anything within 15 degrees from the vertical and within 2.4 units from the center

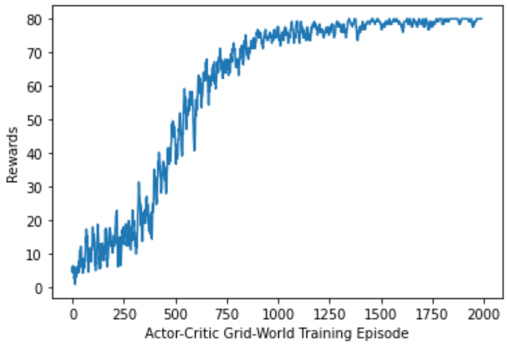
- Reward : +1 for every timestep when the pole is upright
- Goal : To prevent the pendulum from falling over

In ‘Lunar Lander’,

Landing pad is always at coordinates (0,0). Coordinates are the first two numbers in state vector. Reward for moving from the top of the screen to landing pad and zero speed is about 100..140 points. If lander moves away from landing pad it loses reward back. Episode finishes if the lander crashes or comes to rest, receiving additional -100 or +100 points. Each leg ground contact is +10. Firing main engine is -0.3 points each frame. Solved is 200 points. Landing outside landing pad is possible. Fuel is infinite, so an agent can learn to fly and then land on its first attempt. Four discrete actions available: do nothing, fire left orientation engine, fire main engine, fire right orientation engine.

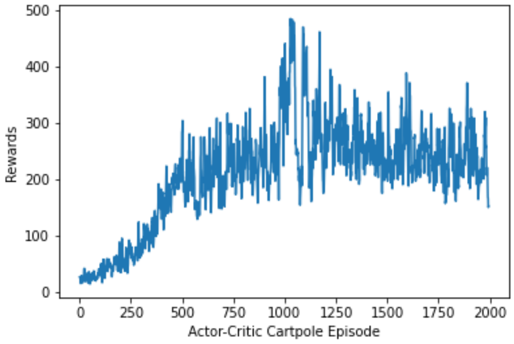
The main objective is to land on the launch pad in between the flags.

4. Grid-World :



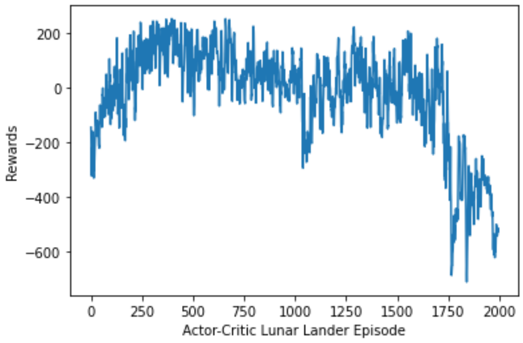
It can be seen that the rewards are increasing as number of episodes are increasing. So, we can say that the Actor Critic algorithm is working well on our grid-world environment.

CartPole-v1 :



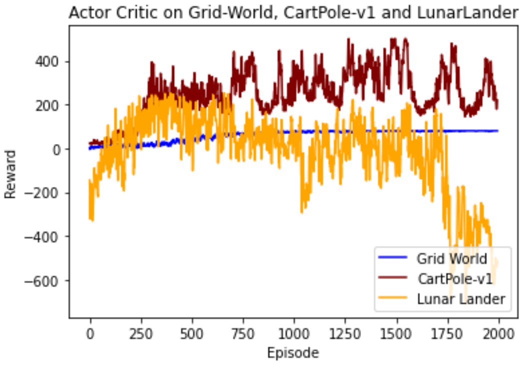
The graph is converging and attains rewards of 475+. Thus, we can say that our Q Actor Critic algorithm has solved the CartPole-v1 environment.

LunarLander-v2 :



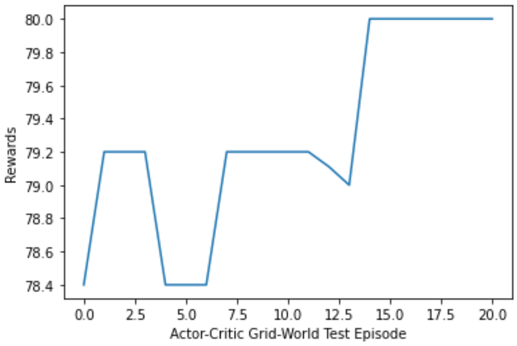
The algorithm is getting rewards over 200 in 1000 episodes, which thus shows that the Q Actor Critic Algorithm is solving the Lunar Lander environment.

Comparison of the three environments :

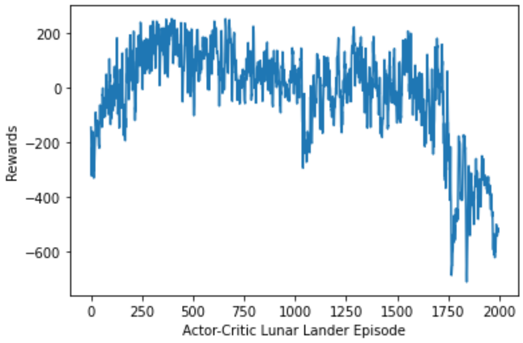


It is seen that all the three environments are having a good reward score and our Q-Actor Critic Algorithm is solving the environments as expected.

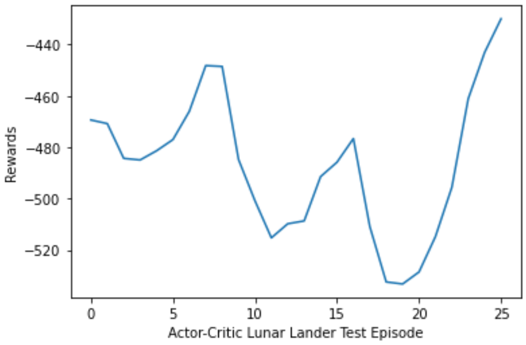
5. Grid-World :



CartPole-v1 :



LunarLander-v2 :


Rewards
Actor-Critic Lunar Lander Test Episode