Reinforcement Learning Project

Rohith Reddy Kolla | rkolla

1. The algorithm implemented in this assignment is the Advantage Actor Critic (A2C). The algorithm implements a critic to estimate the action-value function. It maintains two sets of parameters – *Critic* updates the action-value function parameters and the Actor updates policy parameters in direction suggested by critic.

   Actor-critic algorithms like A2C follow an approximate policy gradient

$$\nabla_\theta J(\theta) \approx E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) Q_w(s,a)]$$
$$\Delta\theta = \alpha \nabla_\theta \log \pi_\theta(s,a) Q_w(s,a)$$

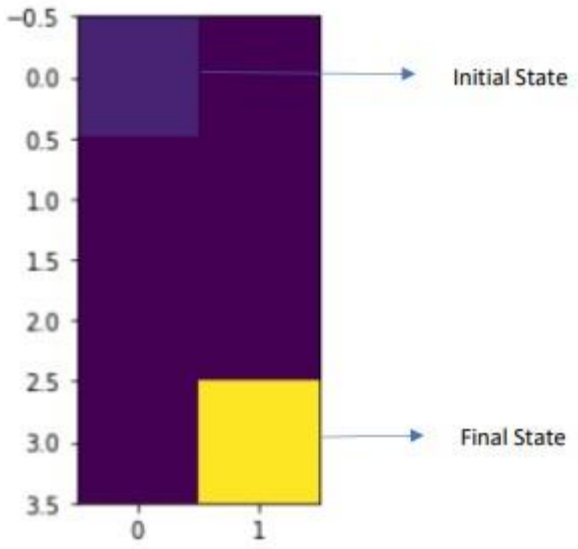
2. They key difference between the actor-critic and value based approximation algorithms is that value based algorithms try to estimate the optimal value function or Q-function and extract the policy from it whereas actor-critic based algorithms estimate the value function of the current policy and use it to improve the policy.

3. GridWorld:

   The grid world environment used is the same environment from the previous assignment which is defined with 8 states in a 4 x 2 grid where the initial state is at [1,1] and the goal is to reach the final state at [4,2].

   4 actions (down, up, left, right) are possible with any attempts to go out of bounds resulting in staying in the same state.

   The rewards are based on whether the transition brings the agent closer to the final state. +1 is it comes closer, -1 if it goes farther, 0 if it stays in the same place.



   CartPole-v1:

   The states are of the type Box (4) containing the cart position, cart velocity, pole angle and pole angular velocity.

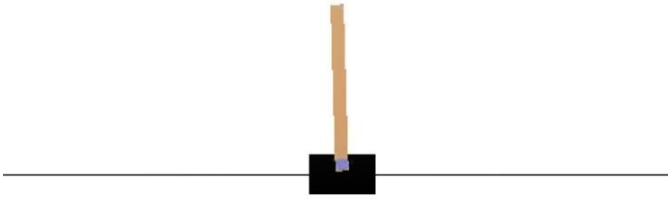
Type: Box(4)

Num	Observation	Min	Max
0	Cart Position	-4.8	4.8
1	Cart Velocity	-Inf	Inf
2	Pole Angle	-0.418 rad (-24 deg)	0.418 rad (24 deg)
3	Pole Angular Velocity	-Inf	Inf

In the starting state, all observations are assigned a uniform random value in [-0.05, 0.05].

The agent has two possible actions – push cart to the left and push cart to right. A reward of +1 is obtained for every timestep that the pole remains upright.

The episode ends after 200 timesteps or when either the pole is more that 12 degrees from the vertical or the cart moves more than 2.4 units from the center. The environment is considered solved when the return for an episode is greater than or equal to 195.

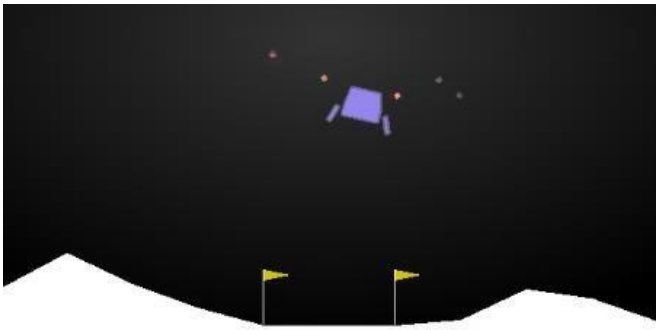


LunarLander-v2:

The Lunar Lander environment is relatively more complex with States of the type Box(8) where the first two numbers are the coordinates of the lander. The landing pad is located at the coordinates (0,0) and the goal is to land safely on it.

The agent has 4 possible actions which are firing main engine, firing left engine, firing right engine and not firing any engine.

The agent loses reward if the lander moves away from the landing pad. The episode finishes upon crashing or coming to rest receiving a reward of -100 and +100 respectively. Each leg in contact with the ground provides a reward of +10. Firing the main engine is -0.3 points and firing either of the side engines is -0.03 points. The environment is considered to be solved upon receiving a reward of +200.



Visualization of the LunarLander-v2 environment

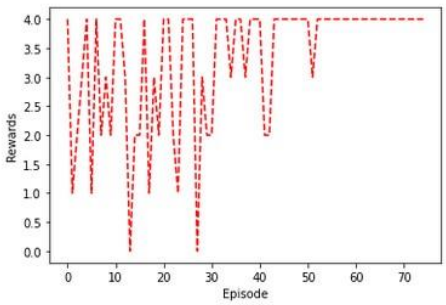
4 In order to consider the influence of hyperparameters on the overall performance, the A2C agent is used on the grid world environment with 3 different hyperparameters.

In all 3 cases the critic networks have 2 hidden layers with ReLu activation and the actor networks have 2 hidden layers with ReLu activation and a softmax activation on the output layer. The number of nodes in each of the hidden layers are different for each case. Adam optimizer is used.

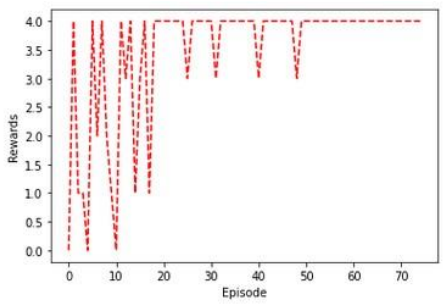
The first case has 48, 24 nodes in hidden layers of both the actor and critic network.

The second case has 32, 24 nodes in the hidden layers of the critic network and 24, 16 nodes in the hidden layers of the actor network.

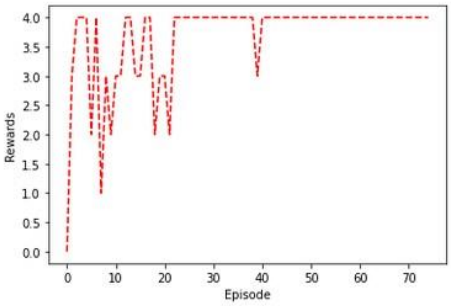
The third case has 32,64 nodes in the hidden layers of both the actor and the critic network.



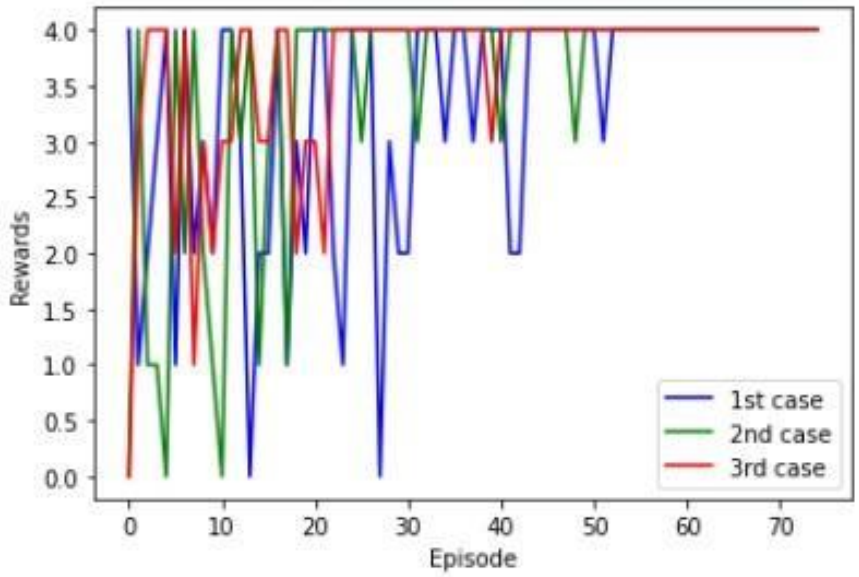
C-(32,24), A-(24,16)
Avg rewards per episode = 3.33



C-(48, 24), A-(48,24)
Avg rewards per episode = 3.50



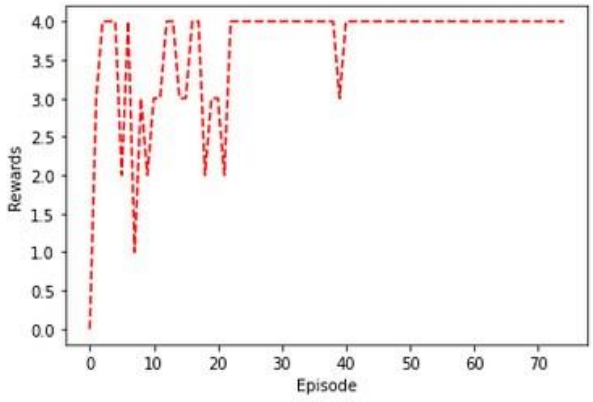
C-(32,64), A-(32,64)
Avg rewards per episode = 3.68



From the results shown above, it is evident that the best performance is in the 3rd case with the greatest number of hidden nodes in actor and critic networks followed by the 2nd case with the 2nd the greatest number of hidden nodes. Unsurprisingly, the 1st case with least number of nodes is performing the worst. It is important to note that the results have an inherent randomness and might be quite different for different random seeds but the general idea that a bigger neural network will usually perform better at function approximation is reflected in the results above. Not only are the bigger networks providing better rewards per episode but are also more stable with consistent higher rewards.

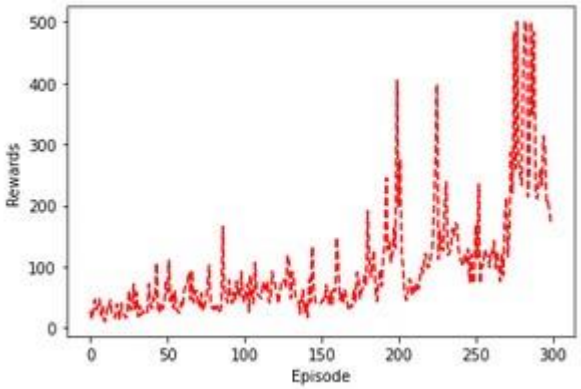
5 GridWorld:

Considering the agent in the 3rd case for the grid world environment, the environment was being solved consistently after around only 40 episodes. The agent learns to transition between states in order to reach the end state quite easily. The graphs indicating the total reward for each episode is shown below.



CartPole-v0:

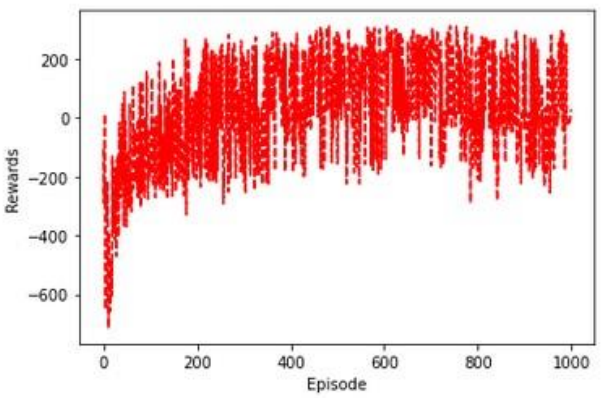
The agent used for the cartpole environment is identical to the agent used in the third case for the GridWorld environment with 36 and 64 nodes in the hidden layers of both the Actor and Critic networks. The agent uses the Adam optimizer and trains for 300 episodes. The graph indicating the total rewards per episode is shown below.

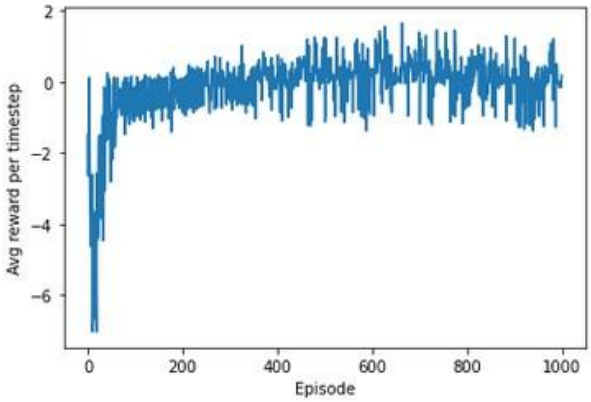


Since the environment is considered to be solved after getting a reward greater than 195, the agent is able to solve the environment for the first time at around the 180th episode. More impressively, the agent is able to accumulate a reward of around 500 in the later episodes which is more than twice required to solve the environment. It is evident that the agent is quite successful with this environment. It is able to learn to move appropriately without tipping over the pole despite the fact that the velocity with which the cart moves upon making an action is not uniform. Since the environment provides 1 reward for each timestep, the average reward per one timestep will always be 1 regardless of the algorithm so the graph has not been included.

LunarLander-v2:

The agent used for the lunar lander environment is exactly identical to the agent used in the cartpole environment including both the actor and critic networks. The agent is trained for 1000 episodes.





Since the environment is considered to be solved after receiving a reward of 200, the agent is able to solve the environment for the first time around the 180th episode. This performance is a very significant improvement over the DQN and double DQN agents deployed in the previous assignment. Not only is the A2C agent able to solve the environment for the first time sooner but it is also able to solve it more consistently. The average reward per timestep graph indicates that the agent improves rapidly in the first 100 episodes with gradual improvements over the following episodes. The Lunar Lander environment is considerably more complicated than the other environments used in this assignment with a bigger observation space and action space but the agent is able to solve it successfully using the same parameters. The agent is able to learn to use the thrusters in a manner in which the least negative rewards are provided while simultaneously lowering the speed of the lander enough for a successful landing.