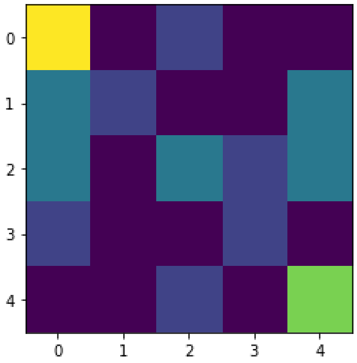
CSE 4/546: Reinforcement Learning

Fall 2021

Assignment 3 - Actor- Critic

Akhil Kanike
University at Buffalo
akhilkan@buffalo.edu

1. Grid Environment

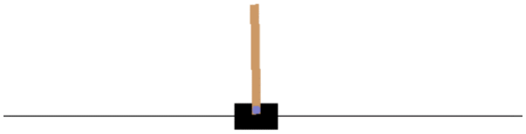


Above Grid is the environment to which we have applied the actor critic. The yellow square is our Agent, and the green square is the Goal to reach. Agent receives +1 reward when falls on cyan square and receives a penalty of –1 when it falls on Blue square. The Purple squares do not affect the Agent, it can move freely.

2. CarPole-v1

2.1 Objective

The objective of Cartpole is to maximize the time an agent can keep a pole attached to a cart on a frictionless track by moving the cart to either left or right, and the agent is receiving a reward of +1 for the time it is keeping the pole up in an entire episode and the episode terminated if the pole angle exceeds, i.e., pole fell down or the cart leaves the screen, or the environment is converged.



2.2 Attributes

1. Set of actions: {0, 1} , Where, 0 = Push the cart to the left ; 1 = Push the cart to right
2. Rewards: {1} : Reward is 1 for every step taken, including the termination step
3. State space = [cart position, cart velocity, pole angle, pole angular velocity]

2.3 Observation

After taking the step 's' the agent observes the four details about the environment which are:

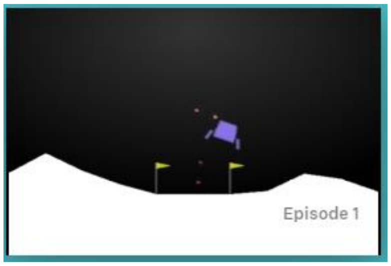
1. Cart Position: Value lie between –4.8 to 4.8
2. Cart Velocity: Value ranges from –∞ to + ∞
3. Pole Angle: ranges from -24° to +24°
4. Pole Angular Velocity: Value ranges from –∞ to + ∞

2.4 Episode Termination:

1. Pole-angle exceed 12°.
2. Cart reaches the end/edge of the display.
3. Accumulated rewards exceed 500 for a single episode.
4. The environment is converged: the agent must accumulate a reward greater than 475 for 100 back-to-back episodes.

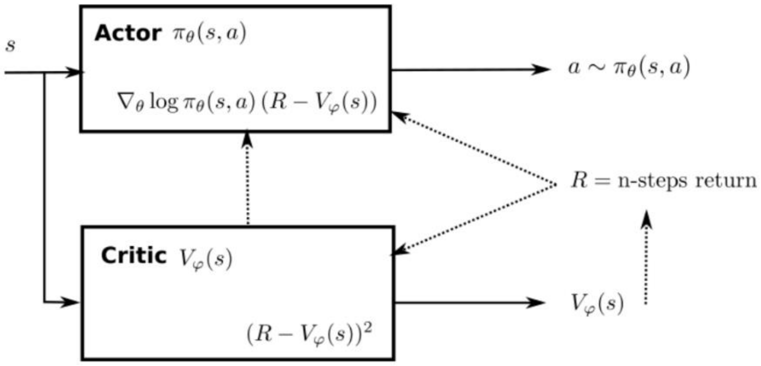
3. LunarLanding-v2 (Discrete)

As part of this Lunar Landing, we need to land the spacecraft in the designated goal position I.e., Landing Pad. 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 the 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.



We would like to apply Continuous control with deep reinforcement learning to this environment and analyze the rewards per episodes.

4. Implementing Advantage Actor Critic & Solving above 3 environments.



A2C has an actor-critic architecture as shown above:

> The actor outputs the policy πθ for a state s, i.e. a vector of probabilities for each action.
>
> The critic outputs the value Vφ(s) of a state s.

Having a computable formula for the policy gradient, the algorithm is rather simple:

1. Acquire a batch of transitions (s,a,r,s′) using the current policy πθ (either a finite episode or a truncated one).
2. For each state encountered, compute the discounted sum of the next n rewards and use the critic to estimate the value of the state encountered n steps later
3. Update the actor below equation

$$\nabla_\theta J(\theta) = \sum_t \nabla_\theta \log \pi_\theta(s_t, a_t)\,(R_t - V_\varphi(s_t))$$

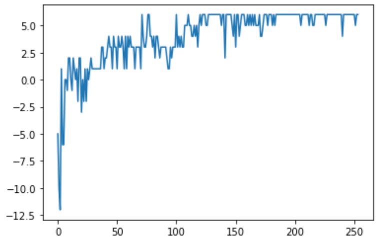
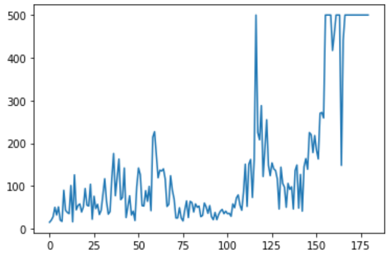
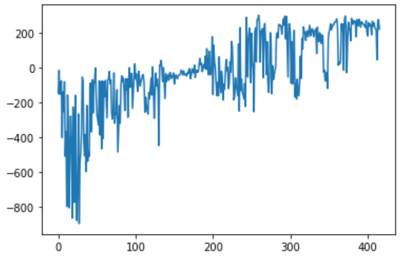
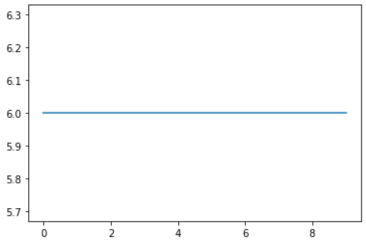
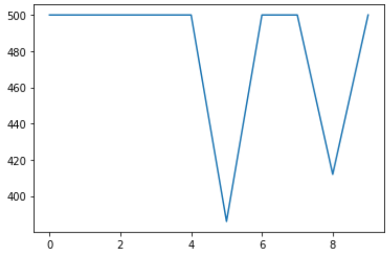
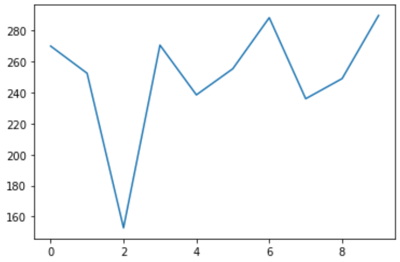
4. Update the critic to minimize the TD error between the estimated value of a state and its true value.

$$\mathcal{L}(\varphi) = \sum_t (R_t - V_\varphi(s_t))^2$$

5. Repeat

The main difference between the Actor Critic and Value based approximation is that value based approximations use value functions whereas actor critic use both value function and policy evaluation.

5. Results:

Metric	Grid World	Cart Pole	Lunar Lander
Reward per episode.			
Test: Running for 10 episodes		- 	

Based on the results, A2C is performing decently well in all the environments. It has learnt quickly compared to the previously applied algorithms in all 3 environments i.e., Convergence is quicker than other algorithms applied till now.

After training, Agents in all Test episodes in each environment have performed excellently acquiring great number of rewards.

6. References:

Deep Reinforcement Learning (julien-vitay.net) and class notes.