Exercise 6

Amr Alkhashab - 905833
ELEC-E8125 - Reinforcement Learning

October 31, 2020

1 Task 1

Actor critic with episodic updates, important parts for task 1 are stated below:

```
#discount_reward from utilis are used to commute Returns (with
    or without discount factor)
discounted = discount_rewards(rewards, 1) #episodic
#based on done final states are set to zero
value_next = torch.where(done.byte(), torch.tensor(0.0),
    value_next)
#MSE
E = value - discounted
SE = torch.pow(E, 2)
MSE = torch.mean(SE)
#Advantages
Adv = rewards + self.gamma * value_next - value
#actor loss
weighted_probs = -action_probs * Adv.detach()
loss = torch.mean(weighted_probs)
```

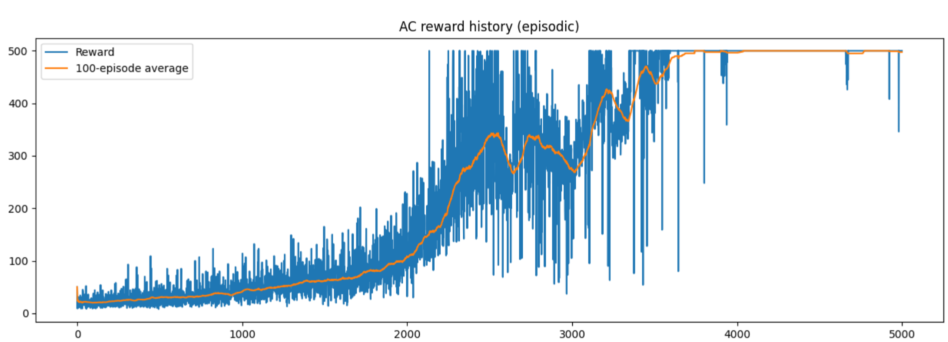

Figure 1: Task1 episodic Actor critic

2 Question 1

Both methods learn a policy and a state-value function if we choose the baseline as $\hat{v}(s, w)$ The main different between both is that Reinforce-with-baseline is not used for bootstrapping like TD(0). In otherword, REINFORCE with baseline is like Monte Carlo methods, while Actor critic is like TD(0).
For the Reinforce we had: $\nabla_\theta \log \pi_\theta(a_t|s_t)(R - b)$
Since: $Q_\pi(s_t, a_t)$ is the true expected rewards, then we can replace R with the Q.
We get: $\nabla_\theta \log \pi_\theta(a_t|s_t)(Q_\pi(s_t, a_t) - b)$
If we select the baseline as value function we get the:$(Q_\pi(s_t, a_t) - V(s_t))$
We can go further by setting Q as: $r(s_t, a_t) + \gamma V(s_{t+1})$ like we do with bellman equation
The final results is: $\nabla_\theta \log \pi_\theta(a_t|s_t)(r(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t))$
From here we can deduce that Actor critic can be derived from Reinforce by using value function as a baseline and using bootsrapped method in term of value function, rather than the return. In otherword, the relation between them is like the relation between TD(0) and Monte Carlo method, where one is design to be updated after each episode and the other for each timestep. Reinforce with baseline cannot thus be implemented online or for continuing problems, while actor-critic can.

3 Question 2

How much advantage over avearage do we get if we choose action a_t in a particular state. Meaning, the advantage is positive when that action is better than the avearage action and negative if that action is worst than avearage action. Then, We can say if advantage is positive the likelihood of performing that action will increase, if the advantages is negative then the likelihood of performing this action decrease, by moving in the positive or negative gradient direction.

4 Task 2

The end of episode already handled using torch.where

```
#discounted cirtic loss
y = rewards + self.gamma * value_next
E = value - y.detach() #non_epiosdic
SE = torch.pow(E, 2)
MSE = torch.mean(SE)
#based on done final states are set to zero
value_next = torch.where(done.byte(), torch.tensor(0.0),
   value_next)
# every 50 timestep update is done using
 count += 1
if count == 50:
     agent.update_policy(episode_number)
      count = 0
```

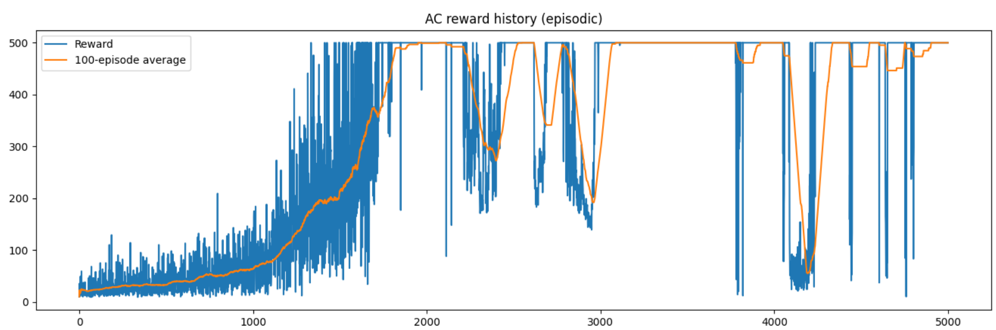

Figure 2: Task2 non-episodic Actor critic

5 Task 3

The default setting of 64 parallel envs running in 8 processes and 8 env each is used.

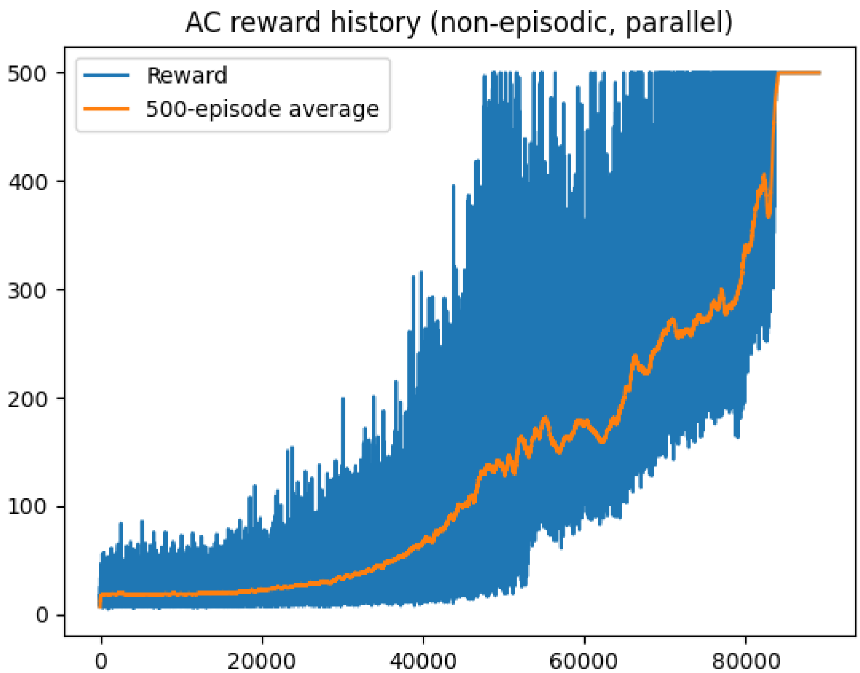

Figure 3: Task3 parrallel data collection

6 Question 3

$multiple - cartpoles.py$ shows the mean and standard deviation throughout independent training procedures. This means, that training procedures is distributed amoung workers($cpu-count()$) and each is trained independently. In otherword, the model is trained from scratch several time over several cpu. Parallel data collection, however, collect data from different environment, mix these data together and then train the agent (it is like collecting different data from different parrallel episodes), then uses all these data to train a single model. It cannot replace, even though it could be a represention of the mean value, since data is collected from different environment, however it doesn't show the variation and std which is very important in comparing algorithm.

7 Question 4

It can be said that the Reinforce is used without a parameterized baseline in this implementation. In such case there will be more parameters we are trying to optimize in case for the actor-critic. In that sense, it will take more time for actor-critic to optimze those parameter, by commuting the loss for each, that is why it needed more data at the start before it learns something useful, that can be concluded by the flat line at the start. Now, since value function and policy also share same network, and both optimize the first layer, this cause some delay since the first layer is affected not only by policy, but also the value function, thus it takes sometime to learn proper policy, as it cannot learn a proper policy without a proper value function. For the policy gradient, however, each optimization step is done to have a better policy parameter directly. That case, is likely to differ if REINFORCE also uses a critic.

8 Question 5.1

In general, since bootstrapping introduce bias, but due the state representation, actor critic have lower variance and accrelerated learning. Reinforce is unbiased, since $\hat{q}(s_t, a_t) = R$, which is accumulative return with a certain discount factor thus has no bias and will converge to local mimimum asymptotically, however, like monte carlo it is slower with high variance, because the return is the sum if many random variables, and those not only depend on reward but also the the actions the policy takes, and the state that will result from such actions. For actor critic the $\hat{q}(s_t, a_t) = r(s_t, a_t) + \gamma V(s_{t+1})$, which has only 3 variables sampled usually from a one step, in that sense we expect it to have less variance, and since the advantage in term of value function is an approximation and not the perfect estimate due to how it is represented, there is a bias which is random and depend on the neural network .

9 Question 5.2

The bias-variance tradeoff can be controlled using Actor-critic with Eligibility Trace, by combining return of different horizons. Eligibility traces applied to actor critic unifies and generalize Reinforce and actor critic method. As λ approaches 1 then it gain the characteristic of reinforce which is unbiased with high variance, and as it approach 0 we get the actor-critic which is low variance and some bias.

10 Question 6

1. Decrease exploration autonomously over time, since paramterized policy approach deterministic policy over time without any iterference.

2. Avoid failure in case of deterministic policies that has limited approximation function. Ex: like robot stuck in a corner due to deterministic policy. (stochastic policies)

3. Less complicated sometimes. For example in the mountain car problem, if the velocity is negative, the agent to to move left and if positive move right. In that case, the agent learn quickly to escape valley, while using value function will complicate things.

4. Deal better with containous action space.