CIS 419/519: Homework 6

{Yupeng Li}

04.15.2020

Although the solutions are entirely my own, I consulted with the following people and sources while working on this homework: **https://www.youtube.com/watch?v=kNPGXgzxoHw**

PART I: PROBELM SET

1 Reinforcement Learning I

The reward does not communicate the goal to the robot well. The robot can only get a positive reward at the end of the maze while all the other points have 0 reward. That is to say, the robot does not know what to do until it first reaches the exit and it keeps wandering around before that. A better reward would be giving all the failed runs a negative reward so that the robot would know that it should try to navigate to the exit as quick as possible.

2 Reinforcement Learning II

(a) The signs do not matter in continuing tasks. For episodic tasks, however, the sign of these rewards do matter as in problem I.

(b) Based on the fact that:

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

Adding a constant C to all the rewards will simply yield a new reward $\tilde{R}_t$ such that:

$$\tilde{R}_t = \sum_{k=0}^{\infty} \gamma^k (r_{t+k+1} + C)$$

which is also equivalent to:

$$\sum_{k=0}^{\infty} \gamma^k (r_{t+k+1}) + \sum_{k=0}^{\infty} C\gamma^k$$

That is to say:

$$\tilde{R}_t = R_t + \sum_{k=0}^{\infty} C\gamma^k$$

Putting this in to the Value function $V^{\pi}(s) = \mathbb{E}_{\pi}[R_t \mid s_t = s]$

We can easily obtain the fact that

$$\tilde{V}_{\pi}(s) = \mathbb{E}_{\pi}[G_t + \sum_{k=0}^{\infty} \gamma^k C \mid S_t = s] = V_{\pi}(s) + \sum_{k=0}^{\infty} \gamma^k C$$

Since the discounted factor γ is always smaller than 1, the second term in the equation above can be simplified as $\frac{C}{1-\gamma}$ based on the geometric series sum.

(c) Based on the facts above, that is to say, the constant K added to the values of all states is simply

$$\frac{C}{1-\gamma}$$

PART II: PROGRAMMING EXERCISES

3 Random policy for the MountainCar gym environment

(i) The action spaces consist of three discrete actions each regarding to accelerate to left, accelerate to right and stay still. The observation space is a two dimensional space each of which represents x or y coordinate of the mountain car.

(ii) The mean reward obtained over 10 episodes remain to be -200 since the car never got to the goal based on a random policy within 200 steps.

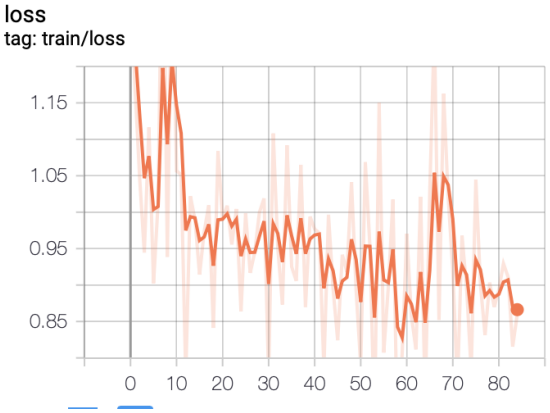
4 Train a Q-learner and generate expert trajectories

The function **discretize()** turns the infinite set of combinations of the observation space into discretized finite states and round the infinite state to its closest. Increasing the discretization argument will increase the model size since there would be more possible observation states.

5 Train an imitation policy

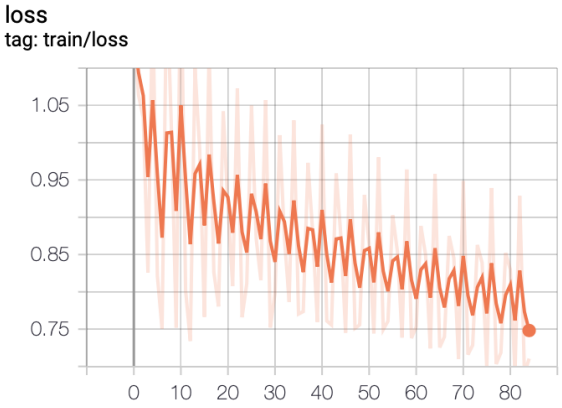
The loss and success rate of 20 episodes with 2 epochs are in the following figure

Figure 1: 20 Episodes and 2 Epochs



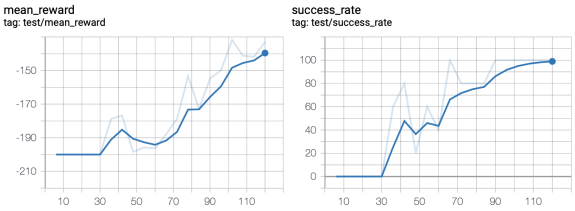
Since there are only 2 epochs, the data is too little to plot out for success rate reward The loss and success rate of 2 episodes with 20 epochs are in the following figure

Figure 2: 2 Episodes and 20 Epochs loss



The average reward and average success follows:

Figure 3: 2 Episodes and 20 Epochs Reward and Success



6 Implement DAgger

After implementing DAgger, the performance of the model further improves as in the following picture:

Figure 4: DAgger loss, reward and success