ECE598 Learning-based Robotics

Machine Problem 3

Yutong Xie (yutongx6)

Oct. 25, 2020

Problem1 Behavior Cloning from States

1. Implementation

I implemented the training loop to solve the 3 tasks. For discrete tasks, I choose cross-entropy loss, and for continuous tasks, I use the mean-squared error loss. For optimizer, I used Adam optimizer with a learning rate of 1e-4. The test results are shown below.

CartPole-v2:

```
Epoch: 250 Loss: tensor(43.6520, grad_fn=<AddBackward0>)
Epoch: 500 Loss: tensor(42.8889, grad_fn=<AddBackward0>)
Epoch: 750 Loss: tensor(46.7658, grad_fn=<AddBackward0>)
Epoch: 1000 Loss: tensor(46.6182, grad_fn=<AddBackward0>)
100%|| 200/200 [00:00<00:00, 516.32it/s]
E1025 19:59:27.418987 4607008192 evaluation.py:54] Episode Reward:   199.930, upright_frames:   1.000.
```

DoubleIntegrator-v1

```
Epoch: 250 Loss: tensor(0.4258, grad_fn=<AddBackward0>)
Epoch: 500 Loss: tensor(0.1393, grad_fn=<AddBackward0>)
Epoch: 750 Loss: tensor(0.0242, grad_fn=<AddBackward0>)
Epoch: 1000 Loss: tensor(0.0105, grad_fn=<AddBackward0>)
100%|| 200/200 [00:01<00:00, 163.37it/s]
E1025 20:06:56.113190 4447751616 evaluation.py:54] Episode Reward:  -418.803, success:   1.000.
```

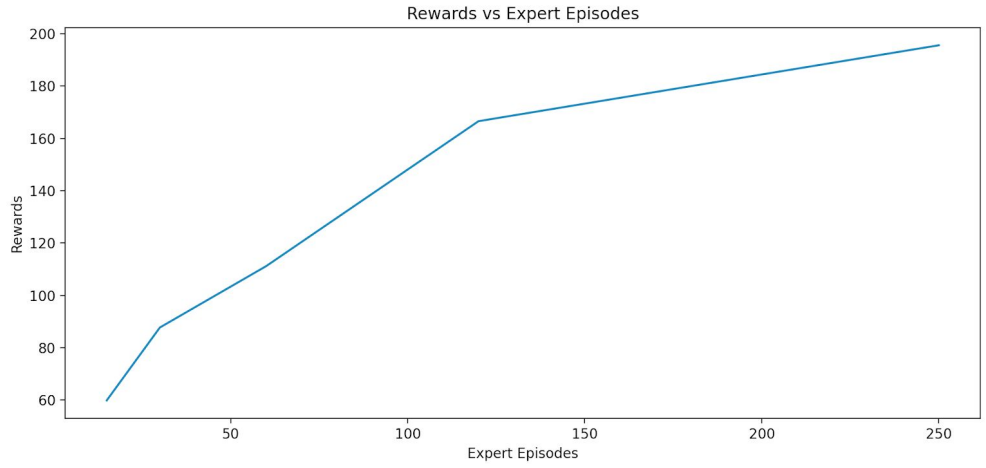
PenduluBalance-v1

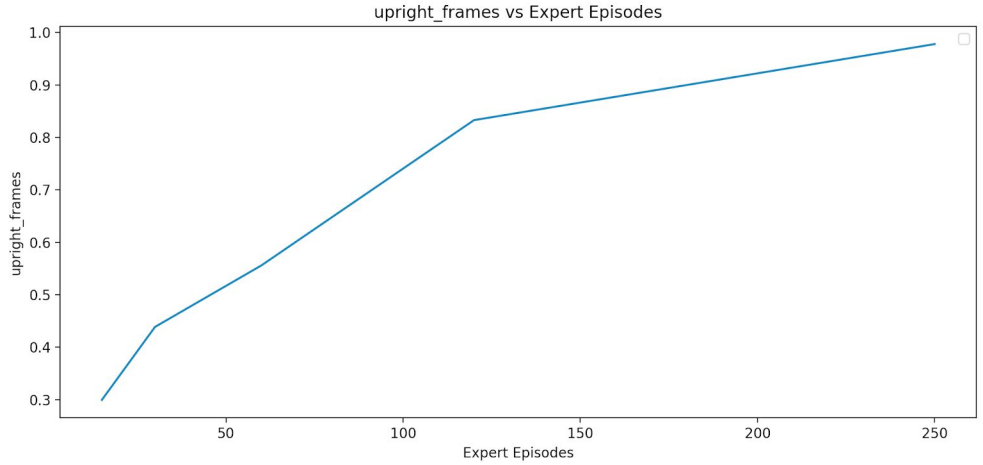
```
Epoch: 250 Loss: tensor(0.0068, grad_fn=<AddBackward0>)
Epoch: 500 Loss: tensor(0.0020, grad_fn=<AddBackward0>)
Epoch: 750 Loss: tensor(0.0012, grad_fn=<AddBackward0>)
Epoch: 1000 Loss: tensor(0.0011, grad_fn=<AddBackward0>)
100%|| 200/200 [00:02<00:00, 88.74it/s]
E1025 20:10:28.261155 4801117632 evaluation.py:54] Episode Reward:    -5.406, fraction_upright:   0.973.
```

2. Data efficiency of learning

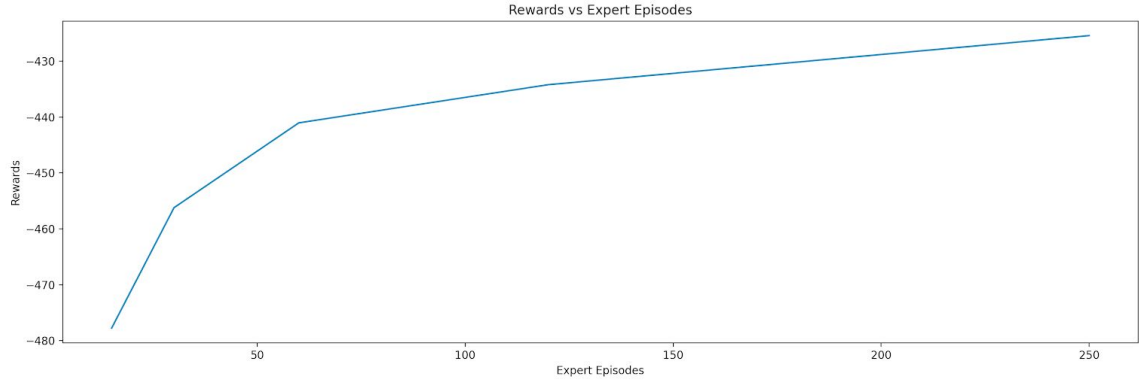
I tried to study the performance of the policy as a function of the number of expert demonstrations. I plot the metric vs the number of expert episodes and rewards vs the number of expert episodes for three environments. The results are shown below.

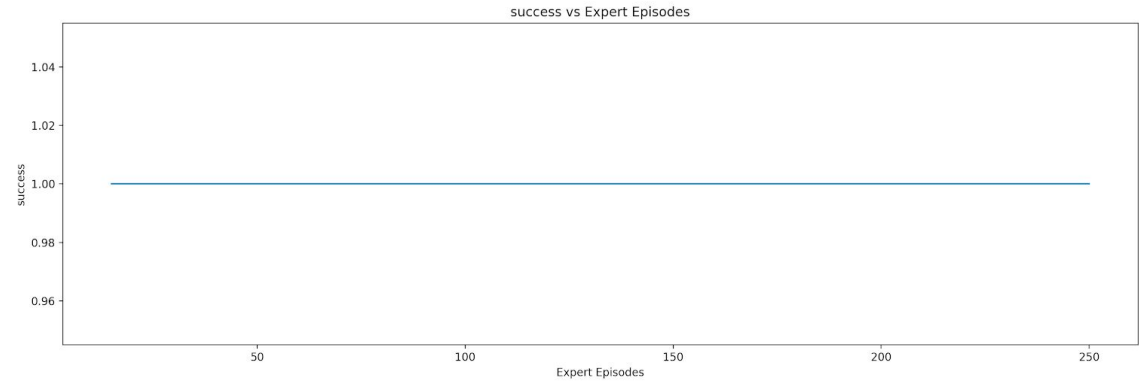
CartPole-v2





DoubleIntegrator-v1





PeduluBalance-v1 w