TEK5040 Assignment 3 d

Markus Sverdvik Heiervang - markuhei

Task 1

Write down the line numbers of the Python code in run gail.py corresponding to each statement of psuedo code in Listing 1.

In line 338, policy network, value network and discriminator are initialized

Line 350 The outer loop of N iterations starts

353 to 358, the empty lists are initialized

Line 360 inner loop of k steps

Line 362 we generate the action for current observation
Line 363 action is performed, and we get next observation Line 364 pair and get reward
Line 366 value is generated
Line 367 - 372 values are appended to lists Line 374 Make next observation current observation

Line 397 compute generalized advantages using the rewards and values list

Line 401 if iteration is a multiple of 3 Line 403 for `ppo_epochs` epochs, do Line 404, 405 update policy and value nets

Line 411 update discriminator

Task 2

Complete the statement in line 297, by specifying target values. Describe your choice of target values with respect to the compile statement in lines 287/288 and original GAIL algorithm. HINT: Try to use quantities defined in lines 264 and 265.

Because the discriminator is updated on the following form:

```
discriminator.update(np.concatenate([expert_ob_ac, policy_ob_ac], axis=0))
```

we only need to label expert trajectories as 1 and policy trajectories as 0. So it becomes just as simple as

```
self.model_prob.train_on_batch(all_ob_ac, [self.ones, self.zeros])
```

Task 3

If you are asked to modify the provided script and implement the Guided Cost Learning (GCL) algorithm, describe briefly how you would do that (You are NOT required to implement the GCL algorithm).

GCL is very similar to GAIL with the difference being that GCL approximates a reward function from the expert trajectories instead of using a discriminator in a model-to-model arms race. In the gails case, the discriminator guesses whether the trajectories are part of the expert trajectories or generated by the generator. In GCL, reward is continous. So to adapt the script, we replace the discriminator class with an appropriate reward function approximator, and base the ppo off of that.

Task 4

For this problem, it is possible to define a reward function

$$r = -(\theta^2 + 0.1\dot{\theta}^2 + 0.001a^2)$$

where a is the action(torque). The reward is miximized when the pendulum is in the upright position with a minimum (zero) effort (i.e. $\theta = 0, \dot{\theta} = 0, a = 0$). Why is the GAIL algorithm more useful for complex practical control problems in, for example, navigation and manipulation?

Usually, it is very difficult to define a good reward function for a much more complex problem than this one, and expert trajectories might be the best data we have to work with. When driving a car, how can we mathematically design a reward function for driving it? It would be much easier to just prepare a dataset by recording actions of an experienced, or expert driver.