DeepRL-Assignment 1

Javad Aminian Dehkordi

1 Analysis

1.1

The inequality can be converted into an expectation over p_{π^*}:

$$E_{p_{\pi^*}(s)}[\pi_\theta(a \neq \pi^*(s)|s)] = \sum_s p_{\pi^*}(s)\pi_\theta(a \neq \pi^*(s)|s).$$

Using the given condition:

$$\frac{1}{T}\sum_{t=1}^{T} E_{p_{\pi^*}(s_t)}[\pi_\theta(a_t \neq \pi^*(s_t)|s_t)] \leq \epsilon$$

$$\Rightarrow \sum_{t=1}^{T}\sum_{s_t} p_{\pi^*}(s_t)\pi_\theta(a_t \neq \pi^*(s_t)|s_t) \leq T\epsilon$$

Now, $E_{t,s_t} = \pi_\theta(a_t \neq \pi^*(s_t)|s_t)$.

$$Pr(\bigcup_{t=1}^{T}\bigcup_{s_t} E_{t,s_t}) \leq \sum_{t=1}^{T}\sum_{s_t} Pr(E_{t,s_t}).$$

Using the given condition,

$$\sum_{t=1}^{T}\sum_{s_t} p_{\pi^*}(s_t)\pi_\theta(a_t \neq \pi^*(s_t)|s_t) \leq T\epsilon.$$

Now, for each t, we can write the absolute difference in state visitation probabilities as:

$$|p_{\pi_\theta}(s_t) - p_{\pi^*}(s_t)| = |E[\pi_\theta(s_t) - \pi^*(s_t)]| \leq E[|\pi_\theta(s_t) - \pi^*(s_t)|].$$

Now, $|\pi_\theta(s_t) - \pi^*(s_t)|$ is 0 whenever $\pi_\theta(s_t) = \pi^*(s_t)$ and is at most 1 otherwise, so we can further upper bound this by:

$$E[|\pi_\theta(s_t) - \pi^*(s_t)|] \leq E[\pi_\theta(s_t \neq \pi^*(s_t)|s_t)].$$

Now, summing over all s_t, :

$$\sum_{s_t} E[|\pi_\theta(s_t) - \pi^*(s_t)|] \leq \sum_{s_t} E[\pi_\theta(s_t \neq \pi^*(s_t)|s_t)].$$

Using the provided condition, we can now upper bound the right side by $T\epsilon$, which gives:

$$\sum_{s_t} E[|\pi_\theta(s_t) - \pi^*(s_t)|] \leq T\epsilon.$$

Multiplying by 2 (as there are at most $2T$ time steps across the trajectories) gives the desired result:

$$\sum_{s_t} |p_{\pi_\theta}(s_t) - p_{\pi^*}(s_t)| \leq 2T\epsilon.$$

1.2

Part (a)

Given that $r(s_t) = 0$ for all $t < T$:

$$J(\pi) = E_{p_\pi(s_T)}[r(s_T)]$$

So:

$$J(\pi^*) - J(\pi_\theta) = E_{p_{\pi^*}(s_T)}[r(s_T)] - E_{p_{\pi_\theta}(s_T)}[r(s_T)] = r(s_T) \cdot (p_{\pi^*}(s_T) - p_{\pi_\theta}(s_T))$$

We know:

$$\frac{1}{T}\sum_{t=1}^{T} E_{p_{\pi^*}(s_t)}[\pi_\theta(a_t \neq \pi^*(s_t)|s_t)] \leq \epsilon$$

This implies that the imitation policy π_θ is, on average, very close to the expert policy π^*, sso the distributions of states that they induce, $p_{\pi_\theta}(s_t)$ and $p_{\pi^*}(s_t)$, are also very close for each t.

$$\begin{aligned} |J(\pi^*) - J(\pi_\theta)| = |r(s_T) \cdot (p_{\pi^*}(s_T) - p_{\pi_\theta}(s_T))| &\leq R_{\max} \cdot \epsilon \\ &\leq R_{\max} \cdot T\epsilon = O(T\epsilon) \end{aligned}$$

$R_{\max}$ is the maximum possible reward, and ϵ is the maximum expected likelihood that the imitation policy disagrees with the expert policy. This shows that the difference in expected return is on the order of ϵ, and since ϵ is multiplied by a constant $R_{\max}$, it is also $O(T\epsilon)$, since $T\epsilon \geq \epsilon$.

Part (b)

$$J(\pi) = \sum_{t=1}^{T} \sum_{s_t} p_\pi(s_t) r(s_t)$$

$$\Rightarrow J(\pi^*) - J(\pi_\theta) = \sum_{t=1}^{T} \sum_{s_t} (p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)) r(s_t)$$

Now, note that the absolute difference in expected returns can be bounded as:

$$|J(\pi^*) - J(\pi_\theta)| = \left| \sum_{t=1}^{T} \sum_{s_t} (p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)) r(s_t) \right|$$

$$\leq \sum_{t=1}^{T} \sum_{s_t} |p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)||r(s_t)|$$

$$\leq R_{\max} \sum_{t=1}^{T} \sum_{s_t} |p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)|$$

From part 1, we know that $\sum_{s_t} |p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)| \leq 2T\epsilon$, so we can say

$$\sum_{t=1}^{T} \sum_{s_t} |p_{\pi^*}(s_t) - p_{\pi_\theta}(s_t)| \leq \sum_{t=1}^{T} 2T\epsilon = 2T^2\epsilon$$

For the difference in expected returns:

$$|J(\pi^*) - J(\pi_\theta)| \leq R_{\max} \cdot 2T^2\epsilon = O(T^2\epsilon)$$

2 Editing code

Done.

3 Behavioral Cloning

3.1

As per the result obtained after applying BC on different task, I went for Half-Cheetah and Hopper as they represented the most and the least performances, respectively. The hyper-parameters mentioned in the caption of the table were maintained unchanged for generating the results.

3.2

See figure 1 in the next page.

4 DAgger

4.1

Done.

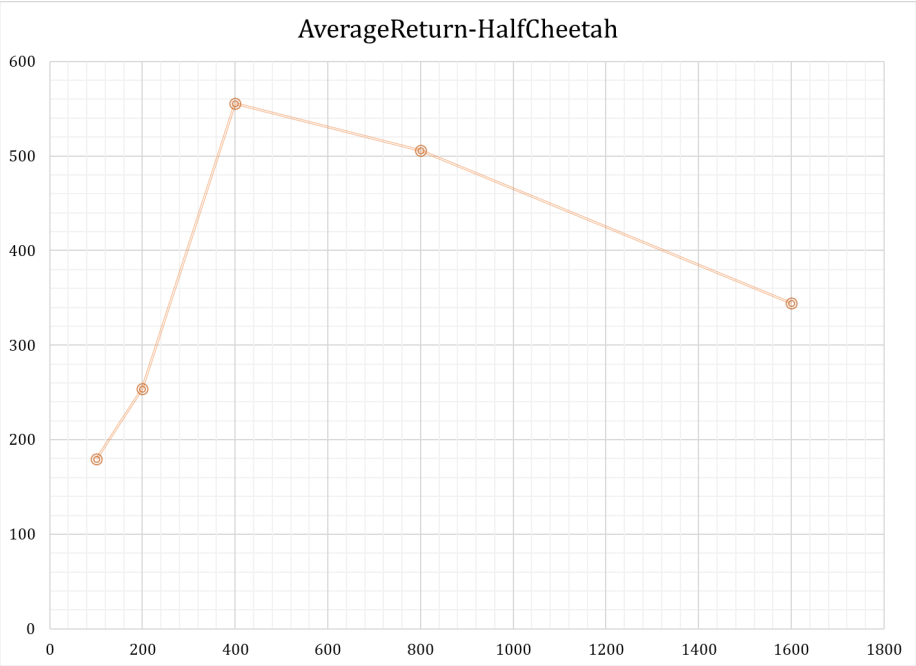

Figure 1: Figure 1- Changes of average return by increasing number of gradient steps for training policy. At each step, the number of gradient steps were doubled, while maintaining other hyper parameters unchanged. This task had the best efficiency among others and showed a very decent behavior with changes in the number of gradient steps for training policy. So this task was selected for this part of the assignment.

Table 1: Table 1- Comparison of Hopper and Half-Cheetah properties while hyper parameters including (–num-agent-train-steps-per-iter 1000 –batch-size 500 –eval-batch-size 1000 –train-batch-size 100 –n-layers 2 –learning-rate 10e-3 –ep-len 200) are kept fixed.

.

Property	**HalfCheetah**	**Hopper**
Eval_AverageReturn	**681.7300415**	**503.153656**
Eval_StdReturn	49.83815002	49.82898712
Eval_MaxReturn	731.371521	577.62323
Eval_MinReturn	598.6039429	431.9997559
Eval_AverageEpLen	200	189.8333333
Train_AverageReturn	4034.799983	3717.512994
Train_StdReturn	32.86776313	0.353036178
Train_MaxReturn	4067.667747	3717.86603
Train_MinReturn	4001.93222	3717.159957
Train_AverageEpLen	1000	1000
Training Loss	-1.382502913	-0.648096263
Train_EnvstepsSoFar	0	0
TimeSinceStart	1.291215181	1.291854382
Initial_DataCollection_AverageReturn	**4034.799983**	**3717.512994**

4.2

See figure 2 in the next page.

5 Discussion

5.1

I almost spent more than a week for this assignment.

5.2

Sometimes I felt like more information about some parts of codings would be really helpful and save an enormous amount of time, given this was the first assignment and there were some new things that I faced for the first time.

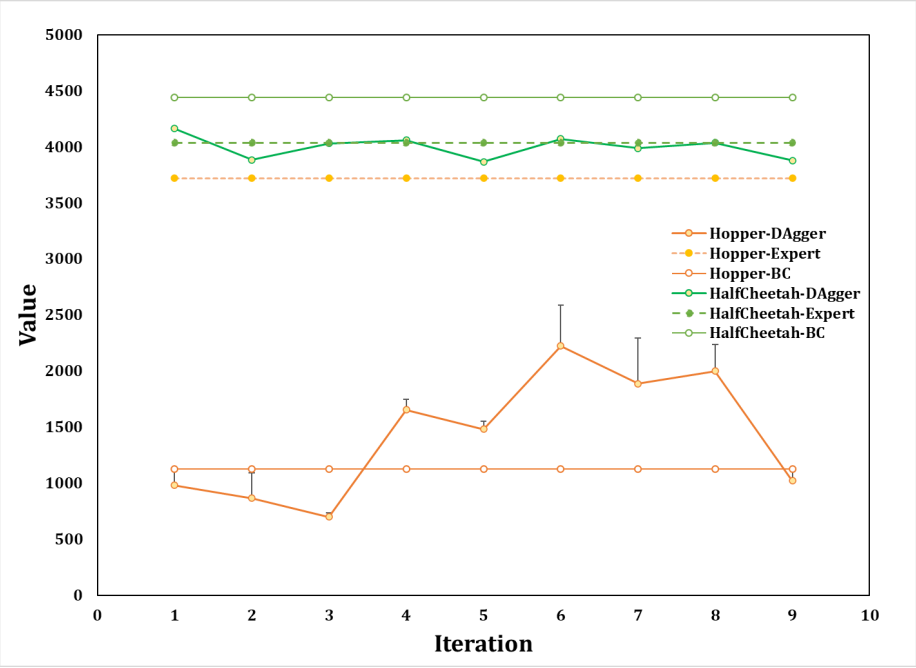

Figure 2: Figure 2-for this section, Hopper and HalfCheetah were used. There were no changes in the hyper parameters compared to the corresponding BC runs.