[CSCI-GA 3033-090]
Special Topics: Deep Reinforcement Learning
Homework - 1 DAgger

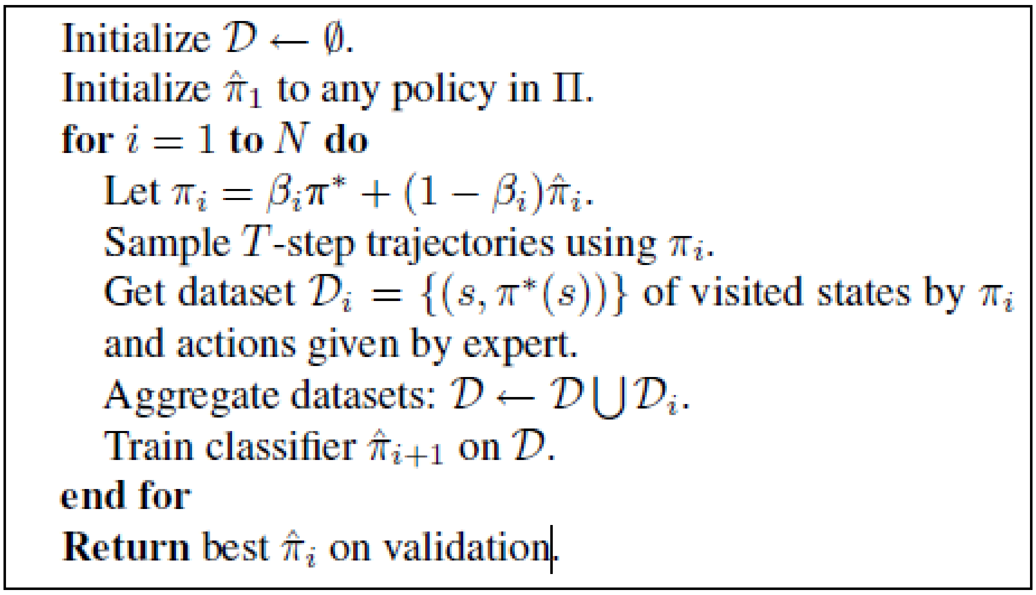
Xu Cao
xc2057@nyu.edu

25 september 2021

Question 1

All finished. Pleased check my completed code folder.

As the pseudo-code in our lecture note and the original DAgger paper is different, I build the model on the basis of the original DAgger paper's pseudo-code (Figure 1).

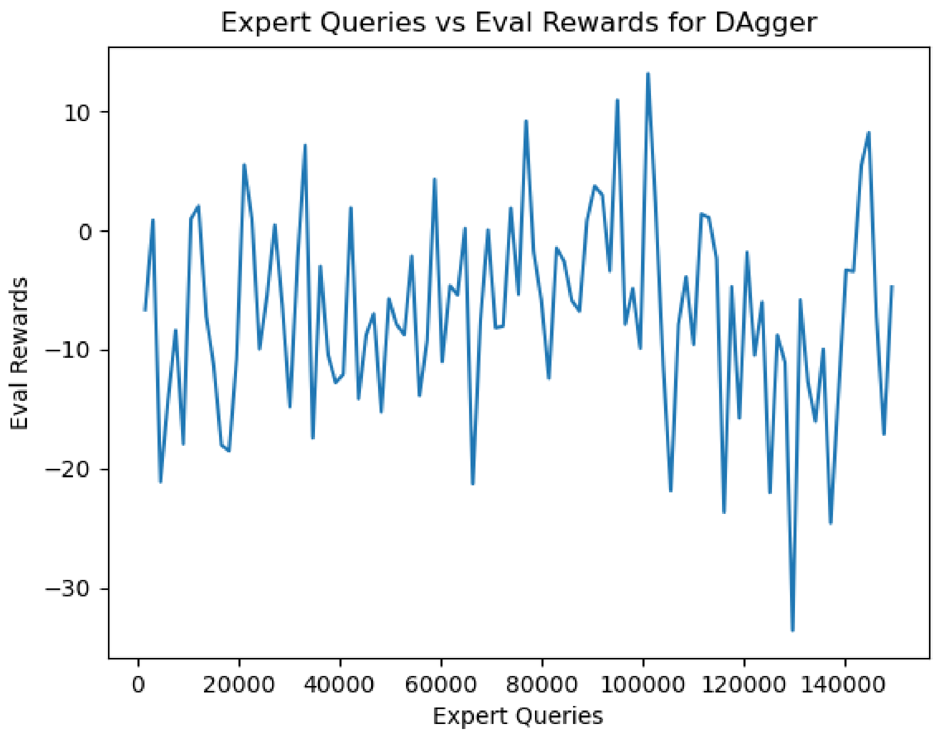


```
Initialize $\mathcal{D} \leftarrow \emptyset$.
Initialize $\hat{\pi}_1$ to any policy in $\Pi$.
for $i = 1$ to $N$ do
   Let $\pi_i = \beta_i \pi^* + (1 - \beta_i)\hat{\pi}_i$.
   Sample $T$-step trajectories using $\pi_i$.
   Get dataset $\mathcal{D}_i = \{(s, \pi^*(s))\}$ of visited states by $\pi_i$
   and actions given by expert.
   Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \bigcup \mathcal{D}_i$.
   Train classifier $\hat{\pi}_{i+1}$ on $\mathcal{D}$.
end for
Return best $\hat{\pi}_i$ on validation.
```

Figur 1: DAgger pseudo-code [3]

To implement the DAgger, I used a VGG-like deep convolution network with 6 convolution layers and 2 fully-connected layers, and using PRelu[1] as activation functions. During the training process, I have tried three different losses, L2 loss (MSE), L1 loss (MAE), and Huber loss. The huber loss performs the best among all these losses. As we knew, huber loss is less sensitive to outliers in data than MSE, that might be the reason that it performs the best. I also test different optimizers, including SGD, SGD with L2 regularization, Adam, and AdamW[2]. My obervation is: for DAgger, SGD with L2 regularization performs better than SGD; AdamW performs better than Adam. A possible explanation for this phenomenon is we need to use regularization for training because it is a $[-1, 1]$ regression problem. Besides, I replaced the weights initialization method from orthogonal initialization to He's initialization[1]

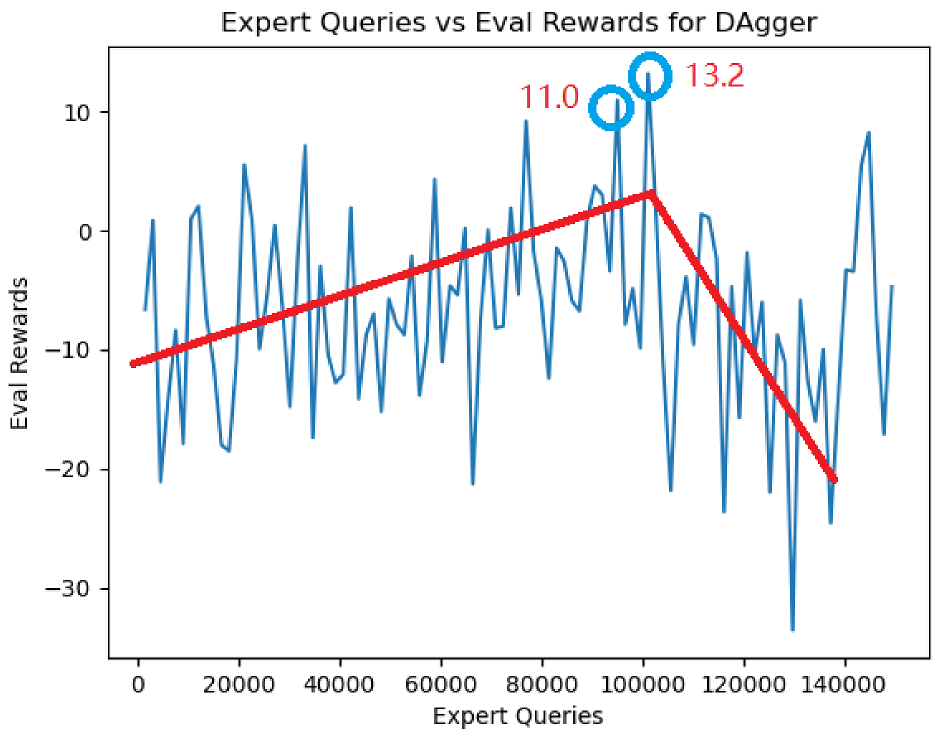
Question 2



Figur 2: number of expert queries vs performance (evaluation reward)

Figure 2 and Figure 3 are the plot and trend analysis image for question 2. For each evaluation sample point, the number of episodes is 3, which means one point on the plot means a average on 3 evaluations. At first, the average evaluation rewards increase with expert queries increase. The maximum average reward achieved 13.2 after 100000 expert queries. Then, the average reward start to decrease by continually expert queries increasing.

In order to explain this strange trend, I re-check our code and find a possible reason. Since we set batch size to 256 and number of training steps to 10000 and never change it, after data augmentation many times, the model will need more training steps to converge. Thus, poor training steps cause the decreasing reward. To solve this problem, we need to design a policy that increase the number of training steps after data augmentations.



Figur 3: The changing trend

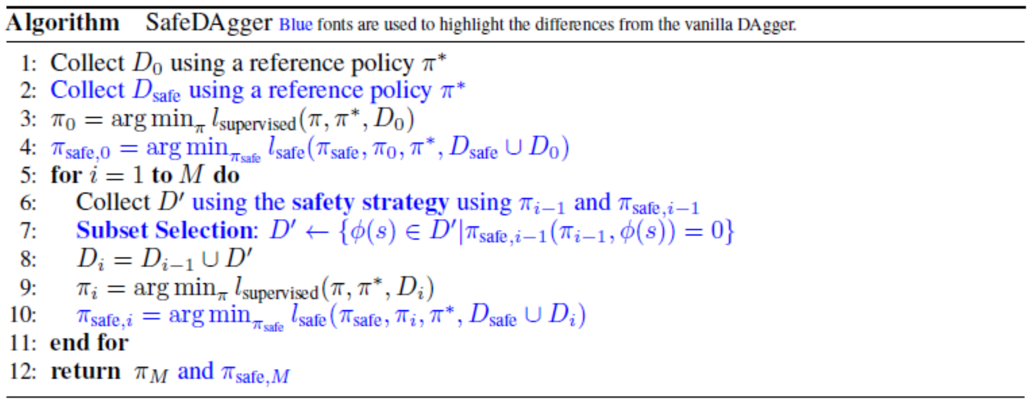
Question 3

A possible method to improve the number of queries to the expert made by the DAgger algorithm is SafeDAgger, which is firstly introduced to solve self-driving problem[4]. To reduce number of queries effectively, SafeDAgger predicts the discrepancy among the learner and supervisor. Those states with high discrepancy are those which get queried (i.e., labeled) and used in training. Prof. Kyunghyun Cho at NYU is one of the authors in this paper.

The pseudo-code for SafeDAgger is shown in Figure 4. Lines 1-6 use the expert policy D^* to collect the initial training data along with the safe data, and also to train an initial policy π_0 as well as an initial safety model $\pi_{safe,0}$. The safety strategy at Line 8 is the process of giving control to the expert policy π^* when the control model π^i cannot drive safely, and Line 9 reduces the number of queries to the expert policy. Lines 10-12 aggregate data and update policies π^i and $\pi_{safe,i}$ by retraining them on the aggregated data.

Why SafeDAgger can improve on the number of queries to the expert? My understanding is: different from original DAgger, SafeDAgger use a new policy called safety policy to take in the observation of the state $\phi(s)$ and must determine whether the primary policy π is likely to deviate from a reference policy π^* at $\phi(s)$. In brief, $\pi_{safe,i}$ determines if the reference must be queried.

Though the author said SafeDAgger performs better than DAgger with less expert queries, it may still need more time to train the model. That is because it use two networks (Original model for DAgger and Safe model) and need to update both two models in one iteration.



```
Algorithm  SafeDAgger Blue fonts are used to highlight the differences from the vanilla DAgger.
1: Collect $D_0$ using a reference policy $\pi^*$
2: Collect $D_{\text{safe}}$ using a reference policy $\pi^*$
3: $\pi_0 = \arg\min_\pi l_{\text{supervised}}(\pi, \pi^*, D_0)$
4: $\pi_{\text{safe},0} = \arg\min_{\pi_{\text{safe}}} l_{\text{safe}}(\pi_{\text{safe}}, \pi_0, \pi^*, D_{\text{safe}} \cup D_0)$
5: for $i = 1$ to $M$ do
6:     Collect $D'$ using the safety strategy using $\pi_{i-1}$ and $\pi_{\text{safe},i-1}$
7:     Subset Selection: $D' \leftarrow \{\phi(s) \in D' | \pi_{\text{safe},i-1}(\pi_{i-1}, \phi(s)) = 0\}$
8:     $D_i = D_{i-1} \cup D'$
9:     $\pi_i = \arg\min_\pi l_{\text{supervised}}(\pi, \pi^*, D_i)$
10:    $\pi_{\text{safe},i} = \arg\min_{\pi_{\text{safe}}} l_{\text{safe}}(\pi_{\text{safe}}, \pi_i, \pi^*, D_{\text{safe}} \cup D_i)$
11: end for
12: return $\pi_M$ and $\pi_{\text{safe},M}$
```

Figur 4: SafeDAgger pseudo-code[4]

Bonus points

To implement SafeDAgger on my project folder, I change the original DAgger template code to SafeDAgger by adding a new safe buffer and a safe policy model. The Safe network is a feedforward network with two fully-connected hidden layers of rectified linear units. This safety policy network takes as input the activations of last convolution layer of the primary policy network.

The maximum average reward will achieve earlier than original DAgger. However, the trend of SafeDAgger seems not increasing at all. To solve this problem, we may need to change the hyper-parameter. Due to SafeDAgger contain two networks for training, I do not have enough time to re-train my model.

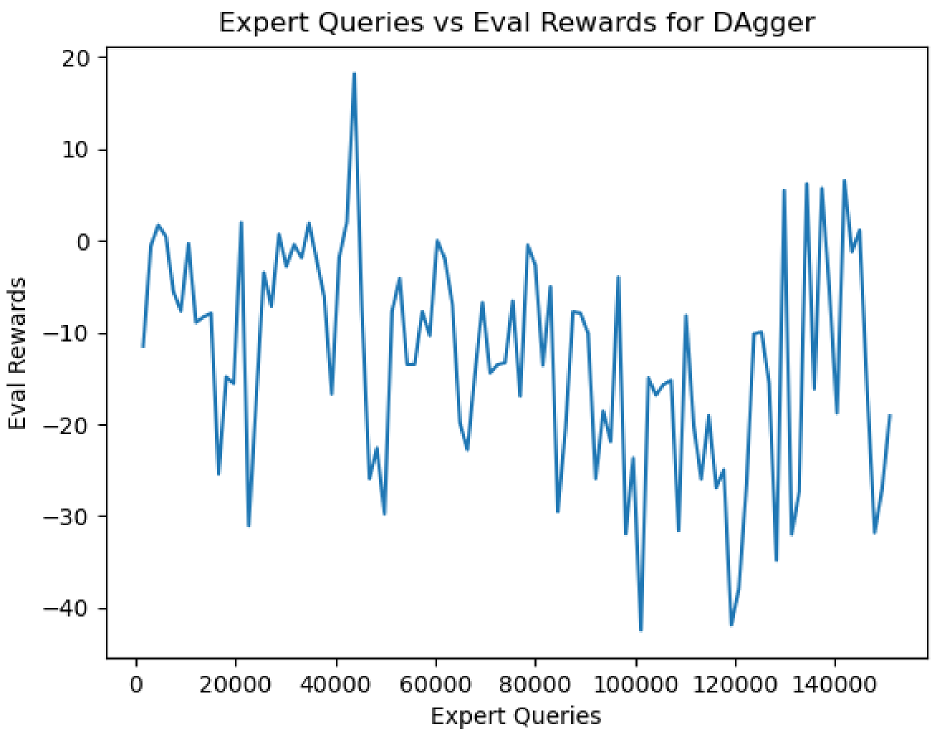
Please check the code file safedagger_template.py and utils.py for more details.

Referenser

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[2] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[3] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth*



Figur 5: number of expert queries vs performance (evaluation reward) for SafeDAgger[4]

international conference on artificial intelligence and statistics, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[4] Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.