1709.10089 - Overcoming Exploration in Reinforcement Learning with Demonstrations

Overcoming Exploration in Reinforcement Learning with Demonstrations

Ashvin Nair[1,2], Bob McGrew[1], Marcin Andrychowicz[1], Wojciech Zaremba[1], Pieter Abbeel[1,2]

- **Yunqiu Xu**
- Imitation learning + RL:
    - Goal: similar to DDPGfD, try to handle sparse reward
    - Use demonstration to overcome exploration problem
    - Combine DDPG with Hindsight
    - Test on robot arm manipulating, and some tasks not solvable by RL and behavioral cloning alone (创新点存疑, 这里不就是DQfD和DDPGfD的工作么)

---

1. Introduction

- Challenge : sparse reward
    - Hand designed reward function is prone to sub-optimal
    - Large exploration space
- Our work:
    - Replace random exploration by learning from demonstration
    - Combine RL (DDPG) with imitation learning, make learned policy better than demonstrations
    - HER: speeds up training on sparse reward

2. Related Work

- Imitation learning:
    - BC: can not exceed demonstration

- DAGGER: need expert during all of training
- **IRL: omitted, for that we have assumed the knowledge of reward function**
- RL and robot learning: omitted
- Combine RL with imitation learning $\rightarrow$ **closest to DQfD and DDPGfD**
  - DDPGfD 解决了相对简单的任务(injection), 重点在加速已经可解决的任务
  - **本工作试图探索更难解决或未被解决的任务**
    - Multi-step behaviors
    - Generalization to varying goal states

3. Background

- DDPG:
  - Model-free, off-policy, continuous action-space $\rightarrow$ suitable for demonstration learning
  - Actor: policy to maximize action value with respect to parameters, update by policy gradient
  - Critic: action-value function to evaluate Q value, update by Bellman function
- Muiti-goal RL
  - Train agents with parametrized goals
  - Sample the goal at the beginning every episode as additional input
  - **HER: more general policies**
  - UVAF: make learning with sparse reward easier
- Hindsight Experience Replay (HER)
  - Assumption: 对于每个state, 我们都可以找到对应的goal, 然后根据能否从state到达goal决定能否得到reward(binary)
  - 可以设置一个mapping function: $r(s_t, g_t) \rightarrow r_t$
  - 注意这里和sparse reward不冲突, 对于未达成目标的episode, 我们可以假定rollout中一个state为goal并将目标设置为这个state
  - 一个比较简单的例子是bitflipping游戏
  - Store an episode $(s_1, s_2, \ldots, s_T)$ in replay buffer twice:
    - One is with original goal
    - Another it with "final goal" in this episode: if the agent still fails at s_T, then

set s_T as goal for this episode

4. Method

- Second replay buffer R_D:
  - 和DDPGfD一样, 另外构建一个用于存储demonstration的replay buffer
  - Demonstration的格式同replay buffer中的transition相同
- Behavior cloning loss
  - Goal: train the actor
  - **Computed only on demonstration examples**

$$L_{BC} = \sum_{i=1}^{N_D} (\pi(s_i|\theta_\pi) - a_i)^2$$

  - Then compute its gradient to improve actor's parameters

$$\lambda_1 \nabla_{\theta_\pi} J - \lambda_2 \nabla_{\theta_\pi} L_{BC}$$

  - **Maximize J, minimize L_{BC}**
  - Why use L_{BC} : avoid improving too significantly beyond demonstration $\rightarrow$ 学到的策略可以有一定提升, 但不能与demonstration差别太大, 防止步子太大扯着蛋 (出发点类似DPPO)
- Q-filter
  - Used in L_{BC}

$$L_{BC} = \sum_{i=1}^{N_D} (\pi(s_i|\theta_\pi) - a_i)^2 \mathbb{1}_{Q(s_i,a_i) < Q(s_i,\pi(s_i))}$$

  - Why use $\mathbb{1}_{Q(s_i,a_i) < Q(s_i,\pi(s_i))}$: 仅仅使用demonstration会陷入sub-optimal, 添加filter后, 仅当demonstrator action的Q值大于actor action时才使用L_{BC}, 换言之, 我们用于学习的demonstration不能太差
- Resets to demonstration states
  - DQfD中如何使用demonstration: 先用demonstration预训练, 然后抽取demonstration 以及 self-generated transition (总共minibatch个, 二者比重可调节)
  - 本工作: 在某些training episodes, 使用demonstration episodes中的states和goals $\rightarrow$ restarts from within demonstrations
  - **assumption : we can start episodes from any given state**
  - How to reset:

- Sample a demonstration $D = (s_0, a_0, s_1, a_1, \ldots, s_N, a_N)$
- Sample a state s_i from D
- Set final state s_N as the final state of $D \rightarrow$ **same as HER**
- Then our goal for this episode is try to reach s_N from s_i

◦ **Reset demonstration will not be used in testing time**

5. Experiment

- Tasks : MuJoCo 7-DOF manipulating
- How to collect demonstration : VR environment

5.1 Comparison to Previous Work

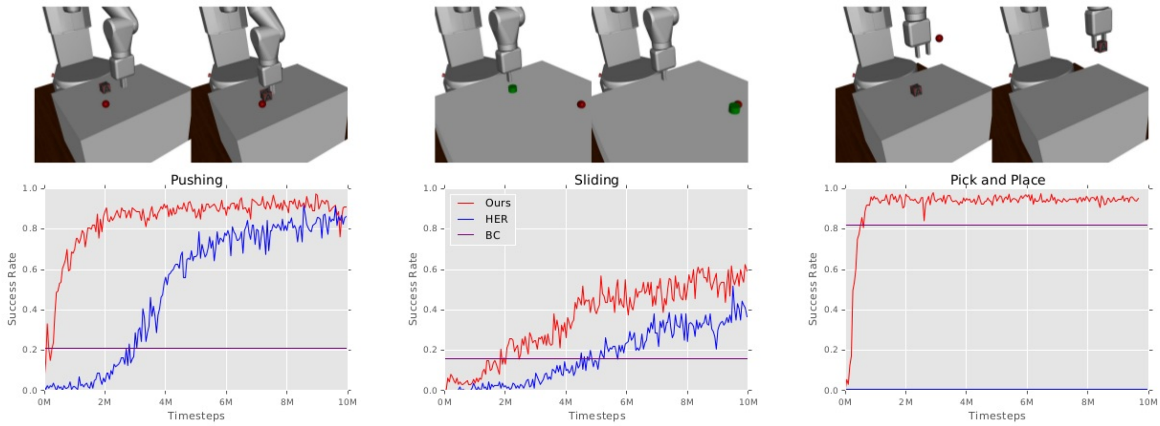

Fig. 2: Baseline comparisons on tasks from [1]. Frames from the learned policy are shown above each task. Our method significantly outperforms the baselines. On the right plot, the HER baseline always fails.

5.2 Block Stacking : Difficult Multi-step Task

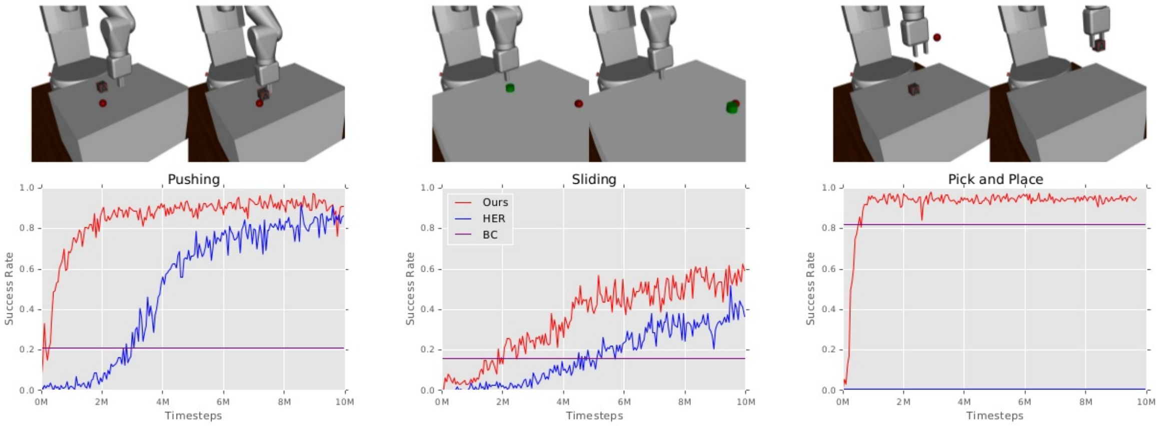

Fig. 2: Baseline comparisons on tasks from [1]. Frames from the learned policy are shown above each task. Our method significantly outperforms the baselines. On the right plot, the HER baseline always fails.

5.3 Ablation Analysis

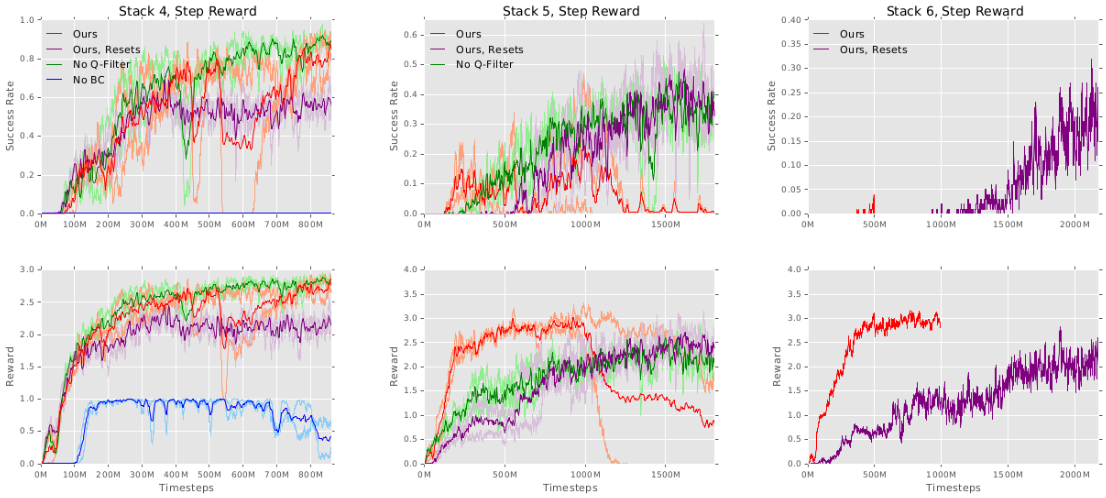

Fig. 5: Ablation results on longer horizon tasks with a step reward. The upper row shows the success rate while the lower row shows the average reward at the final step of each episode obtained by different algorithms. For stacking 4 and 5 blocks, we use 2 random seeds per method. The median of the runs is shown in bold and each training run is plotted in a lighter color. Note that for stacking 4 blocks, the “No BC” method is always at 0% success rate. As the number of blocks increases, resets from demonstrations becomes more important to learn the task.

6. Summary

- Similar to DQfD and DDPGfD, try to leverage demonstration to speed up learning
    - DDPGfD是DQfD的连续版, 使用DDPG, 实验解决injection问题
    - 本工作也使用DDPG, 算法更新过程有一定不同
- 本工作:
    - 在计算 L_{BC} 时添加Q-filter, 只学习还不错的demonstration, 防止sub-optimal

    - 使用HER解决sparse reward问题
    - 实验除了解决一些经典问题, 还尝试解决block-stacking
- Limitation:
    - 本工作和DDPGfD类似, 还需在性能上做下对比
    - 模仿学习中一些固有问题还难以解决, 比如在real world中难以获得大量demonstration
    - Reset demonstration这部分需要假设**我们可以从任意状态开始**
- 一些解决办法:
    - MIL (1709.04905 - One-Shot Visual Imitation Learning via Meta-Learning) : 将imitation learning 和 meta learning 结合, 训练用任务还是那么多demonstration, 但力图让测试用任务达到one-shot
    - Reverse curriculum learning (1707.05300 - Reverse Curriculum Generation for Reinforcement Learning) : 尝试到达接近终点的位置, 一点点倒推