Addressing Function Approximation Error in Actor-Critic Methods

Scott Fujimoto [1] Herke van Hoof [2] David Meger [1]

【强化学习算法 21】TD3

张楚珩
清华大学 交叉信息院博士在读

TD3的是Twin Delayed Deep Deterministic policy gradient algorithm的简

原文传送门：

Fujimoto, Scott, Herke van Hoof, and Dave Meger. "Addressing Function Approximation Error in Actor-Critic Methods." arXiv preprint arXiv:1802.09477 (2018).

特色：

Double Q-learning是value-based类算法里面用来消除overestimation bias的方法，这篇文章研究了actor-critic类算法里面消除overestimation bias的方法。同时，还研究了target network在TD update中消除累积误差的作用。该方法也是目前比较新的state-of-the-art。相关背景不太明白的可以参看本专栏讲的DQN的改进。

过程：

1. 如何消除actor-critic类算法中的overestimation bias？

回顾double Q-learning中价值函数更新的目标定为

$$y = r + \gamma Q_{\theta_1}(s', \arg\max_a Q_{\theta_2}(s', a))$$

其中，两个网络交替更新。在QN中其中一个使用target network来代替。注意到actor-critic里面价值函数的目标不是optimal action-value function，而是当前策略的action-value function，所以没有取max的这个操作。

如果类比DQN的方法，写出来的更新目标为

$$y = r + \gamma Q_{\theta'}(s', \pi_\phi(s')). \quad (8)$$

即critic用更新较慢的target network，actor还是更新快的；但由于本身actor更新也不快，所以没啥效果。

如果类比double Q-learning，使用两个actor、两个critic写出来的更新目标为

$$y_1 = r + \gamma Q_{\theta'_2}(s', \pi_{\phi_1}(s'))$$
$$y_2 = r + \gamma Q_{\theta'_1}(s', \pi_{\phi_2}(s')). \tag{9}$$

本着“宁可低估，也不要高估”的想法（因为actor会选择高的，因此高估的会累积起来），再把目标改写成

$$y_1 = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \pi_{\phi_1}(s')). \tag{10}$$

最后发现两个actor也没啥用，就用一个actor，这个actor根据 Q_{θ_1} 来更新。两个critic的更新目标都是一样的，即 $y_2 = y_1$ 。这样的算法相比于改变之前的就等于多了一个和原来critic同步更新的辅助critic Q_{θ_2} ，在更新target的时候用来取min。

一点疑问，这样看起来， Q_{θ_2} 和原来的critic相比只有初始化不同，后面的更新都一样，这样形成的两个很类似的critic能不能有效消除function approximation error带来的overestimation bias，心存疑虑。

2. 使用target network

实验结果表明，当policy固定不变的时候，是否使用target network其价值函数都能最后收敛到正确的值；但是actor和critic同步训练的时候，不用target network可能使得训练不稳定或者发散。因此算法的中critic的更新目标都由target network计算出来

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta'.$$

并且，价值函数估计准确之后再来更新policy会更好，因此采用了delayed policy update，即以较高的频率更新价值函数，以较低的频率更新policy。

3. 使用target policy smoothing regularization

希望学到的价值函数在action的维度上更平滑，因此价值函数的更新目标每次都在action上加一个小扰动

$$y = r + \gamma Q_{\theta'}(s', \pi_{\phi'}(s') + \epsilon),$$
$$\epsilon \sim \mathrm{clip}(\mathcal{N}(0, \sigma), -c, c), \tag{14}$$

算法：

把以上三个改进综合起来，得到如下算法。文章有个typo，红线位置应该是 $,$ ，而不是 $.$ 。

Algorithm 1 TD3

Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, ϕ
Initialize target networks $\theta_1' \leftarrow \theta_1$, $\theta_2' \leftarrow \theta_2$, $\phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{B}$
for $t = 1$ **to** T **do**
  Select action with exploration noise $a \sim \pi(s) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r and new state s'
  Store transition tuple (s, a, r, s') in $\mathcal{B}$

  Sample mini-batch of N transitions (s, a, r, s') from $\mathcal{B}$
  $\tilde{a} \leftarrow \pi_{\phi'}(s) + \epsilon, \quad \epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \tilde{a})$
  Update critics $\theta_i \leftarrow \min_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  if t mod d **then**
    Update ϕ by the deterministic policy gradient:
    $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
    Update target networks:
    $\theta_i' \leftarrow \tau\theta_i + (1 - \tau)\theta_i'$
    $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$
  end if
end for

发布于 2018-10-19

算法 机器学习 强化学习 (Reinforcement Learning)