TD3: Addressing Function Approximation Error in Actor-Critic Methods 论文剖析

Scott Fujimoto, Herke van Hood, David Meger

一、文章信息

Addressing Function Approximation Error in Actor-Critic Methods

https://arxiv.org/abs/1802.09477

https://github.com/sfujim/TD3

二、写作动机

在Q-learning中，噪声源自函数逼近中无法避免的不精确估计。这种不精确估计在每次更新时被不断累加，并通过估计的最大化被不断估计为高值，导致高估问题（OverStimation）。作为DQN的拓展，DDPG在解决连续控制问题时也会存在高估问题。

本文借鉴Double DQN解决DQN中高估问题时的经验，解决连续动作控制问题中出现的高估问题。

三、相关工作

1. DDPG

DDPG解决了DQN不能用于连续控制的问题，采用基于DPG的Actor-Critic结构。其Actor部分采用估计网络 $\mu_\theta(s)$ 负责输出确定性动作，其目标网络与估计网络结构相同但参数不同，主要负责更新价值网络Critic；Critic部分负责价值函数 $Q_\omega(s,a)$ 的拟合，其估计网络以Actor估计网络的输出动作为输入，参与Actor和Critic的优化。目标网络Critic参与Critic优化时 Q_{target} 的计算。

价值网络Critic的更新主要基于TD-error的梯度下降，具体更新公式如下：

$$L_{critic} = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

策略网络Actor的更新则是基于梯度上升，从而最大化价值网络输出的Q值，其更新公式如下：

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

2. Double DQN

针对DQN中出现的高估问题，Double DQN使用两个独立的估计量来进行无偏估计。其将传统DQN中最优动作选取与最优动作价值函数计算进行解耦，采用目标网络 θ^- 与策略网络 θ 分别进行计算，具体计算形式如下：

$$a^* = \underset{a \in \mathbb{A}}{argmax} Q(s, a|\theta_t^-)$$
$$Y_t^Q = r + \gamma Q(S_t = s, A_t = a^*; \theta_t)$$

通过减少traget的过估计，Double DQN减少了对动作价值函数的估计。

3. Averaged-DQN

Averaged-DQN提出了一种不同的高估问题解决方法，其聚焦于方差最小化，通过平均Q值来降低target近似方差（TAE），并且不同于Ensemble DQN，Averaged-DQN中采用最近K个版本的Q Function计算平均目标，通过采用过去的target network来避免K-Fold计算，获得更好的方差化简结果，对Variance的计算形式如下：

$$Var[Q_i^A(s_0, a)] = \sum_{m=0}^{M-1} D_{K,m} \gamma^{2m} \sigma_{s_m}^2$$

Averaged-DQN的计算公式如下：

$$Q_N^A(s, a) = \frac{1}{K} \sum_{k=0}^{K-1} Q(s, a; \theta_{N-k})$$

四、TD3

在Double DQN中，高估问题通过解耦最优动作选取与最优动作价值函数计算得到了缓解，这种思路在TD3中得到了延续。作为Actor-Critic框架下的确定性强化学习算法，TD3结合了深度确定性策略梯度算法和双重网络，在缓解DDPG算法的高估问题时取得了优秀表现。

1. 双重网络

不同于DDPG中Actor-Target Actor和Critic-Target Critic的四网络结构，TD3算法在DDPG的基础上新增了一套Critic_2网络。两个Target Critic网络计算出各自的Q-target后，选取 $\min(targetQ_1, targetQ_2)$ 作为原DDPG中 $Q(a')$ 的替代去计算更新目标Q-target，公式如下：

$$Y_{target} = r + \gamma \min(targetQ_1, targetQ_2)$$

两个Critic网络根据Actor的输出分别计算出各自的值 Q_1, Q_2 后，通过 MSE 计算Q值与Q-target之间的Loss，进行参数更新，公式如下：

$$\theta_i = argmin_{\theta_i} \frac{1}{N} \sum (y - Q_{\theta_i}(s, a))^2$$

当Critic网络更新完成后，Actor网络采用延时更新的方法，通过梯度上升自我更新。target网络则是采用软更新的方式进行更新，二者更新公式如下：

$$Actor: \ \nabla_\phi J(\phi) = \frac{1}{N} \sum \nabla_a Q_{\theta_1}(s, a)|a = \pi_\phi(s) \nabla_\phi \pi_\phi(s)$$
$$target: \ \theta = \tau\theta' + (1 - \tau)\theta$$

2. 目标策略平滑正则化

DDPG中采用的确定性策略会在更新Critic网络时极易被函数逼近误差所影响，致使目标估计的方差值增大，估值异常。TD3中引入了平滑化思想（Smoothing），通过向目标动作中添加正态分布噪声并求平均值来近似动作期望，本质是一种正则化方法，采用噪声来平滑掉Q峰值，从而防止其过拟合。公式表示为：

$$a' = \pi_{\phi'}(s') + \epsilon,$$
$$\epsilon \sim clip(N(0, \tilde{\sigma}), -c, c)$$

3. 延迟更新

在训练Actor和Critic网络时，文章发现Actor与Critic之间的相互作用会导致Actor一直在被动的跟随Critic网络进行更新，这种不稳定的状态会使得策略函数会根据不准确的估值朝着错误方向进行更新，并在多次更新中累积这些差异，最终陷入劣化循环。因此Actor网络需要一个稳定的Critic网络来协助其进行更新。

文章调高了Critic网络的更新频率，使其高于Actor。保证先尽可能的降低估计误差，确保TD-error足够小。待获得更加稳定的结果后再去协助Actor网络进行更新。

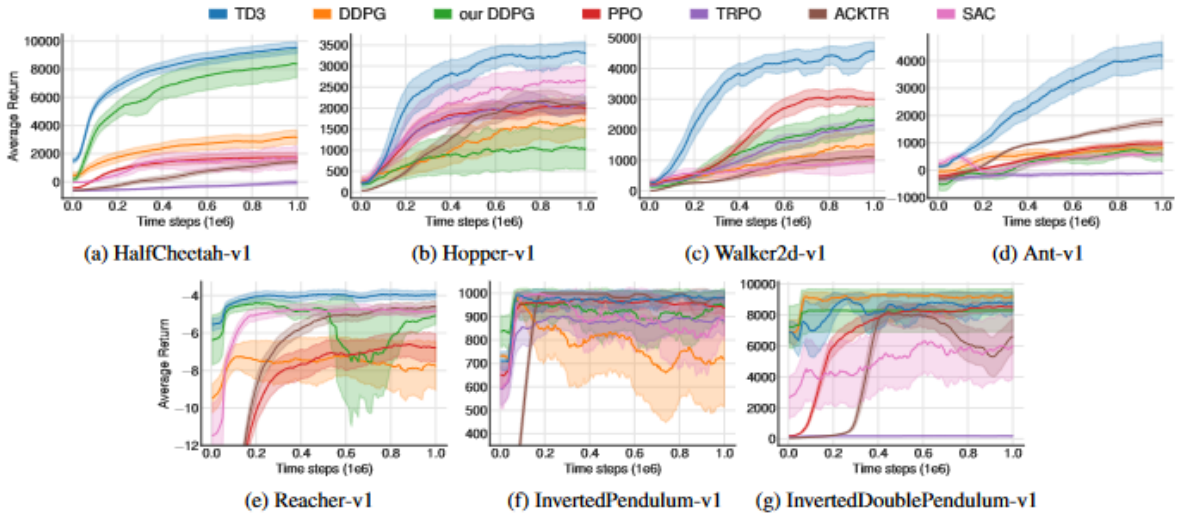
五、实验环节

1. 评估

本文采用MuJoCo连续控制任务对TD3与PPO、DDPG及其他算法进行评估，其中DDPG模型通过一个两层的前馈神经网络实现，并在每层之间对Actor和Critic均采用了ReLU和Tanh单元，同时作者还针对原始DDPG算法进行了修改，将状态和动作共同接受并输入第一层中。两个网络参数都采用Adam进行更新，设置学习率为10^{-3}。

目标策略平滑通过添加$\epsilon \sim N(0, 0.2)$实现，将其剪辑到区间$(-0.5, 0.5)$。延迟更新策略设置每2次迭代更新Actor与Target Critic。在进行软更新时，设置$\tau = 0.005$。

在HalfCheetah-v1、Hopper-v1等环境中学习曲线越靠上，算法效果越好。具体结果如下图所示：



可见TD3算法的性能相较于其他算法有较明显提升。

2. 消融实验

本节作者针对延迟更新策略DP、目标策略平滑TPS及双Q值学习梯度截取CDQ三个策略进行了消融实验；此外作者还对Double Q-learning和Double DQN中Actor-Critic结构的有效性进行了讨论，具体结果如下：

Method	HCheetah	Hopper	Walker2d	Ant
TD3	9532.99	**3304.75**	**4565.24**	**4185.06**
DDPG	3162.50	1731.94	1520.90	816.35
AHE	8401.02	1061.77	2362.13	564.07
AHE + DP	7588.64	1465.11	2459.53	896.13
AHE + TPS	9023.40	907.56	2961.36	872.17
AHE + CDQ	6470.20	1134.14	3979.21	3818.71
TD3 - DP	9590.65	2407.42	**4695.50**	3754.26
TD3 - TPS	8987.69	2392.59	4033.67	**4155.24**
TD3 - CDQ	9792.80	1837.32	2579.39	849.75
DQ-AC	9433.87	1773.71	3100.45	2445.97
DDQN-AC	**10306.90**	2155.75	3116.81	1092.18

可以看到DP和TPS策略的消融实验对结果影响略小，CDQ结构去掉后结果存在明显变化，证明抑制无偏估计量的高估对于提升算法性能具有显著提高作用。

六、文章贡献

在本文中，贡献主要包括如下几点：

1. 设置类似Double DQN的网络，通过两套Critic网络进行Q值估算，解决了DDPG中存在的Q值高估问题，并通过消融实验证明其重要性。
2. 针对Actor与Critic网络相互作用导致Actor不稳定的问题，通过设置Actor延迟更新进行解决。
3. 通过添加平滑噪声的方式优化方差计算时的峰值，解决由于估值函数对真实值拟合不精确带来的方差问题。

========

作者：王振凯

研究方向：深度学习、强化学习

河北地质大学研究生在读