# Proximal Policy Optimization Algorithms

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov OpenAI

{joschu, filip, prafulla, alec, oleg}@openai.com

## 【强化学习算法 4】PPO



张楚珩 🔮

清华大学 交叉信息院博士在读

4 人赞同了该文章

#### 原文链接:

Schulman, John, et al. "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347 (2017).

**特色:** TRPO很成功,但是计算的过程太复杂了,每步更新运算量大、耗时长。在此基础上进行改进避免复杂的对于KL divergence矩阵的求Hessian。

分类: Model-free、Policy-based、On-policy、Continuous State Space、Continuous Action Space、Support High-dim Input

**理论根据:** 和TRPO一样,需要限制每一步在**策略空间**上的步长,即新旧策略的KL divergence不能太大。

#### 过程:

还是考虑和TRPO(single path)一样的优化目标  $\mathbf{L}(\theta) = \mathbf{R} \mathbf{i} \frac{\pi_{\theta}(\mathbf{u} \mid \mathbf{a})}{\pi_{\theta_{u}}(\mathbf{u} \mid \mathbf{a})} \hat{\mathbf{A}} \mathbf{i}$  ,考虑策略空间上变化不太大,就是说希望  $\frac{\pi_{\theta}(\mathbf{u} \mid \mathbf{a})}{\pi_{\theta_{u}}(\mathbf{u} \mid \mathbf{a})}$  尽可能接近1,这里对于这个目标做一个截断,即如果更新使得这一项远离1了,对于目标优化就产生不了任何效果了。

每步优化:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[ \min(r_t(\theta) \hat{A}_t, \operatorname{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t) \right]$$

## 算法:

#### Algorithm 1 PPO, Actor-Critic Style

 $\begin{array}{l} \textbf{for iteration}{=}1,2,\dots\,\textbf{do} \\ \textbf{for actor}{=}1,2,\dots,N\,\,\textbf{do} \\ \text{Run policy } \pi_{\theta_{\text{old}}} \text{ in environment for } T \text{ timesteps} \\ \text{Compute advantage estimates } \hat{A}_1,\dots,\hat{A}_T \\ \textbf{end for} \\ \text{Optimize surrogate } L \text{ wrt } \theta, \text{ with } K \text{ epochs and minibatch size } M \leq NT \\ \theta_{\text{old}} \leftarrow \theta \end{array}$ 

end for

知于 四张楚圻

## 另外:

文中还提到了另外一种方法就是考虑一个KI-penalized objective,但是约束的系数可以动态调整。 不过效果没有前一种方法好。

• Using several epochs of minibatch SGD, optimize the KL-penalized objective

$$L^{KLPEN}(\theta) = \hat{\mathbb{E}}_t \left[ \frac{\pi_{\theta}(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t - \beta \operatorname{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_{\theta}(\cdot \mid s_t)] \right]$$

- Compute  $d = \hat{\mathbb{E}}_t[\mathrm{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_{\theta}(\cdot \mid s_t)]]$

 $\begin{array}{ll} - \text{ If } d < d_{\rm targ}/1.5, \ \beta \leftarrow \beta/2 \\ - \text{ If } d > d_{\rm targ} \times 1.5, \ \beta \leftarrow \beta \times 2 \end{array} \\ \begin{array}{ll} \text{如果上次更新的比较少就减小约束} \\ \text{如果上次更新的比较大就增大约束} \end{array}$ 

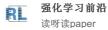
知乎 @张楚珩

编辑于 2018-09-19

算法(书籍) 强化学习 (Reinforcement Learning) 算法

● 3条评论 ▼分享 ● 喜欢 ★ 收藏 …

## 文章被以下专栏收录



进入专栏