WDS
DataScience@Web

第三部分

近端策略优化

汇报人：王瑛

2021.07.22

2021 WDS暑期讨论班

目录

- 概述
- PPO算法
- 对比分析
- 总结

目录

- 概述
- PPO算法
- 对比分析
- 总结

概述-Proximal Policy Optimization（PPO）

动机：

- DQN：无法处理连续性问题
- policy gradient methods ：步长敏感+数据利用效率低
- ACER（ Sample Efficient Actor-Critic with Experience Replay）：太复杂+额外消耗
- TRPO（ Trust Region Policy Optimization ）：太复杂+不兼容一些包含噪声或者参数共享的架构。

PPO是Policy Gradient的变体，可以在如下两个步骤之间来回迭代进行学习：

- 通过与环境进行交互，进行采样；
- 利用梯度上升的方法进行**代替的目标函数（surrogate objective function）**的优化。

●Policy Gradient

$$J^{PG}(\theta) = \hat{\mathbb{E}}_t\left[\log \pi_\theta(a_t \mid s_t)\hat{A}_t\right]$$ Advantage Function

$$\hat{g} = \hat{\mathbb{E}}[\nabla_\theta log\ \pi_\theta(a_t|s_t)\hat{A}_t]$$

$$\theta \leftarrow \theta + \eta \nabla \bar{R}_\theta$$

$$\nabla \bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\nabla \log \pi_\theta(a_t^n \mid s_t^n)$$

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} (R(\tau^n) - b)\nabla \log \pi_\theta(a_t^n \mid s_t^n)$$

$$b \approx E[R(\tau)]$$

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} (R(\tau^n) - b)\nabla log \pi_\theta(a_t^n \mid s_t^n)$$

$$\sum_{t'=t}^{T_n} r_{t'}^n \rightarrow \sum_{t'=t}^{T_n} \gamma^{t'-t} r_{t'}^n$$

Add discount factor $\gamma < 1$

●On-policy--Off-policy

Importance Sampling

$$E_{x\sim p}[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x^i) = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx = E_{x\sim q}\left[f(x)\frac{p(x)}{q(x)}\right]$$

问题：

$$\mathrm{Var}_{x\sim p}[f(x)] = E_{x\sim p}\left[f(x)^2\right] - (E_{x\sim p}[f(x)])^2$$

$$\mathrm{Var}_{x\sim q}\left[f(x)\frac{p(x)}{q(x)}\right] = E_{x\sim q}\left[\left(f(x)\frac{p(x)}{q(x)}\right)^2\right] - (E_{x\sim q}[f(x)\frac{p(x)}{q(x)}])^2$$

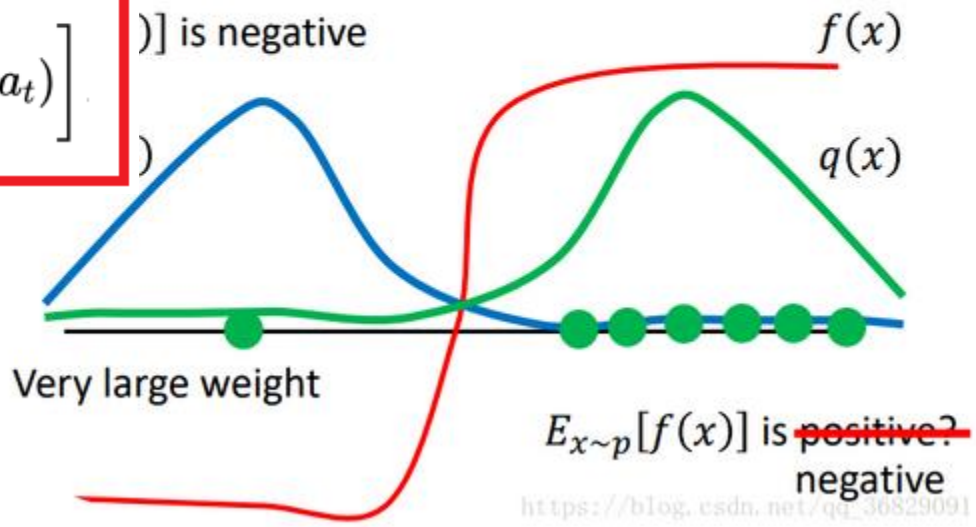
$$= E_{x\sim p}\left[f(x)^2\frac{p(x)}{q(x)}\right] - (E_{x\sim p}[f(x)])^2$$

$$E_{(s_t,a_t)\sim\pi_\theta}[A^\theta(s_t,a_t)\nabla log\,\pi_\theta(a_t \mid s_t)]$$

$$\downarrow$$

$$E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta'}(a_t \mid s_t)}A^{\theta'}(s_t,a_t)\right]$$

$$E_{x\sim p}[f(x)] = E_{x\sim q}[f(x)\frac{p(x)}{q(x)}]$$



目录

- 概述
- PPO算法
- 对比分析
- 总结

PPO1：适应性KL惩罚系数

TRPO $J_{TRPO}^{\theta'}(\theta) = E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta'}(a_t \mid s_t)} A^{\theta'}(s_t,a_t)\right] KL(\theta,\theta') < \delta$

PPO $J_{PPO}^{\theta'}(\theta) = J^{\theta'}(\theta) - \beta KL(\theta,\theta')$

1、初始化参数 θ^0

2、在每一次迭代中，使用 θ^k 来和环境互动，收集状态和行动并计算对应的 $A^{\theta^k}(s_t,a_t)$

3、不断更新参数，找到目标函数最优值对应的参数 θ

If $KL(\theta,\theta^k) > KL_{max}$, increase β

If $KL(\theta,\theta^k) < KL_{min}$, decrease β

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

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

- If $d < d_{\text{targ}}/1.5, \beta \leftarrow \beta/2$
- If $d > d_{\text{targ}} \times 1.5, \beta \leftarrow \beta \times 2$

PPO2：裁剪替代目标函数Clip

$$J^{\theta'}(\theta) = E_{(s_t,a_t)\sim\pi_{\theta'}}\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta'}(a_t \mid s_t)} A^{\theta'}(s_t,a_t)\right]$$

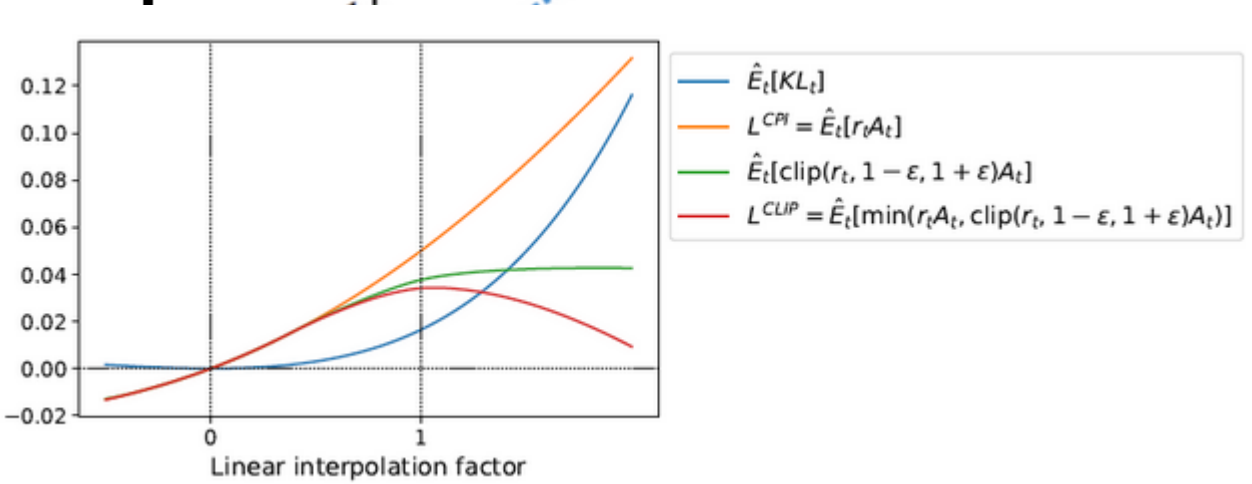
$$J^{\theta'}_{PPO}(\theta) = J^{\theta'}(\theta) - \beta KL(\theta,\theta')$$

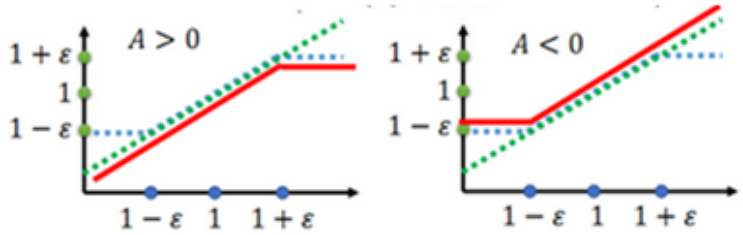
$$J^{\theta^k}_{PPO2}(\theta) \approx \sum_{(s_t,a_t)} \min\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta^k}(a_t \mid s_t)} A^{\theta^k}(s_t,a_t)\right.,$$

$$\left.\text{clip}\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta^k}(a_t \mid s_t)},\ 1-\varepsilon, 1+\varepsilon\right) A^{\theta^k}(s_t$$

$$J^{\theta^k}_{PPO2}(\theta) \approx \sum_{(s_t,a_t)}$$

$$\text{clip}\left(\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta^k}(a_t \mid s_t)},\ 1-\varepsilon, 1+\varepsilon\right) A^{\theta^k}(s_t,a_t)$$





目录

- 概述
- PPO算法
- 对比分析
- 总结

结果

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

No clipping or penalty: $L_t(\theta) = r_t(\theta)\hat{A}_t$

Clipping: $L_t(\theta) = \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta)), 1-\epsilon, 1+\epsilon)\hat{A}_t$

KL penalty (fixed or adaptive) $L_t(\theta) = r_t(\theta)\hat{A}_t - \beta\,\text{KL}[\pi_{\theta_{\text{old}}}, \pi_\theta]$

algorithm	avg. normalized score
No clipping or penalty	-0.39
Clipping, $\epsilon = 0.1$	0.76
Clipping, $\epsilon = 0.2$	**0.82**
Clipping, $\epsilon = 0.3$	0.70
Adaptive KL $d_{\text{targ}} = 0.003$	0.68
Adaptive KL $d_{\text{targ}} = 0.01$	0.74
Adaptive KL $d_{\text{targ}} = 0.03$	0.71
Fixed KL, $\beta = 0.3$	0.62
Fixed KL, $\beta = 1.$	0.71
Fixed KL, $\beta = 3.$	0.72
Fixed KL, $\beta = 10.$	0.69

Table 1: Results from continuous control benchmark. Average normalized scores (over 21 runs of the algorithm, on 7 environments) for each algorithm / hyperparameter setting . β was initialized at 1.

结果

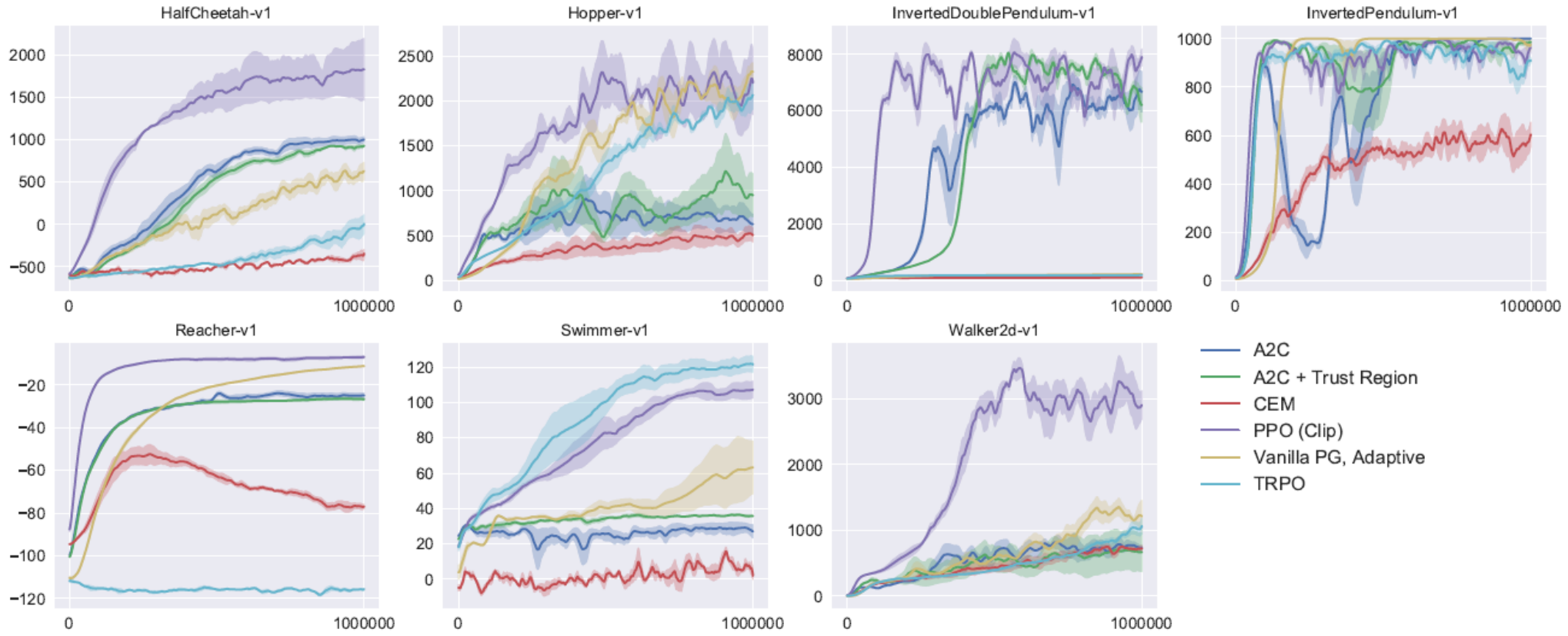


Figure 3: Comparison of several algorithms on several MuJoCo environments, training for one million timesteps.

目录

- 概述
- PPO算法
- 对比分析
- 总结

总结

- PPO是用于RL的策略梯度方法
- 该方法在通过环境交互进行数据采样与使用随机梯度上升优化“替代”目标函数之间交替进行。尽管标准策略梯度方法对每个数据样本执行一个梯度更新，但PPO使用了一种新颖的目标函数，该函数可实现多个批次的小批量更新。
- PPO具有TRPO的优点，但它实现起来更简单，更通用，并且具有更好的样本复杂性。

参考资料

- PPO: Schulman, John, et al. “Proximal Policy Optimization Algorithms.” *ArXiv Preprint ArXiv:1707.06347*, 2017.
- TRPO: Schulman, John, et al. “Trust Region Policy Optimization.” *ArXiv Preprint ArXiv:1502.05477*, 2015.
- [李宏毅深度强化学习2018]P2 Proximal Policy Optimization (PPO)
- [莫烦python]强化学习-策略梯度
- https://openai.com/blog/openai-baselines-ppo/

Thanks!