PPO
从零到深入(1)

东川路第一可爱猫猫虫

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

主要内容

感谢粉丝大佬
W0NDE3RFULHE4VEN
的高档包月充电!

- TRPO的做法
- PPO的改进
- PPO的另一种变体
- Clipped Surrogate Objective function
- PPO里梯度的传播
- 直观看每一段的梯度

TRPO的做法

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t\right]$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t\left[\mathbf{KL}\left[\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)\right]\right] \leq \delta$$

- TRPO里

  拉格朗日对偶 KKT条件

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

- TRPO不采用惩罚项形式

  TRPO用的是硬约束

  复杂、速度慢、开销高

雪碧孙尚香 对我的视频发表了评论

有没有简单点的打法[星星眼][星星眼]

今天 00:58 回复 点赞

- 有的兄弟，有的！
- 既然TRPO的硬约束不便于实现

  我们可不可以用软约束？

  不等式约束一定要严格遵守吗？偶尔违反几次会不会影响不大？
- 既然β不好选择

  使用动态的β

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

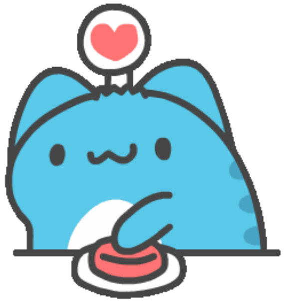
penalty

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

- 先确定一个目标KL散度target
- 通过新旧策略的差异（KL散度）来动态调整β
- 若KL散度过小，减小β
- 若KL散度过大，增大β

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

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



- PPO with Adaptive KL Penalty

Algorithm 4 PPO with Adaptive KL Penalty

Input: initial policy parameters θ_0, initial KL penalty β_0, target KL-divergence δ

for $k = 0, 1, 2, ...$ **do**

Collect set of partial trajectories $\mathcal{D}_k$ on policy $\pi_k = \pi(\theta_k)$

Estimate advantages $\hat{A}_t^{\pi_k}$ using any advantage estimation algorithm

Compute policy update

$$\theta_{k+1} = \arg\max_{\theta} \mathcal{L}_{\theta_k}(\theta) - \beta_k \bar{D}_{KL}(\theta || \theta_k)$$

by taking K steps of minibatch SGD (via Adam)

if $\bar{D}_{KL}(\theta_{k+1} || \theta_k) \geq 1.5\delta$ **then**

$\beta_{k+1} = 2\beta_k$

else if $\bar{D}_{KL}(\theta_{k+1} || \theta_k) \leq \delta/1.5$ **then**

$\beta_{k+1} = \beta_k / 2$

end if

end for

PPO-clip

- PPO-penalty随着时间的推移改变β
- PPO-clip

  直接把策略的改动限制在一个范围里
- PPO-penalty用KL散度衡量新旧策略的差异
- PPO-clip用重要性权重衡量新旧策略的差异
- CLIP函数

$$clip(x, l, r) = max(min(x, r), l)$$

$$clip(p_t(\theta),\ 1-\epsilon,\ 1+\epsilon) = \begin{cases} 1-\epsilon & \text{if } p_t(\theta) < 1-\epsilon \\ 1+\epsilon & \text{if } p_t(\theta) > 1+\epsilon \\ p_t(\theta) & \text{else} \end{cases} \qquad p_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$$

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

- min函数的梯度

$$\frac{\partial min(x,y)}{\partial x} = \begin{cases} 1 & \text{if } x \leqslant y \\ 0 & \text{else} \end{cases}$$

$$\frac{\partial min(x,y)}{\partial y} = \begin{cases} 1 & \text{if } y < x \\ 0 & \text{else} \end{cases}$$

- CLIP函数的梯度

$$\frac{\partial clip(x,a,b)}{\partial x} = \begin{cases} 1 & \text{if } a \leqslant x \leqslant b \\ 0 & \text{else} \end{cases}$$

- 注意这些导数在数学上可能并不严谨

  只是深度学习包所采用的

$$L_t^{CLIP}(\theta) = min\left(p_t(\theta)A_t, clip(p_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)$$

PPO里梯度的传播

$$\frac{\partial min(x,y)}{\partial x} = \begin{cases} 1 & \text{if } x \leqslant y \\ 0 & \text{else} \end{cases} \quad \frac{\partial min(x,y)}{\partial y} = \begin{cases} 1 & \text{if } y < x \\ 0 & \text{else} \end{cases}$$

$$\frac{\partial clip(x,a,b)}{\partial x} = \begin{cases} 1 & \text{if } a \leqslant x \leqslant b \\ 0 & \text{else} \end{cases}$$

- 我们的目标是最大化$L_t^{CLIP}(\theta)$
- 主流深度学习优化器:最小化优化函数
- 取负

对$-L_t^{CLIP}(\theta)$求梯度

$$\frac{\partial -L_t^{CLIP}}{\partial \pi_\theta(a_t|s_t)} = \frac{\partial -L_t^{CLIP}}{\partial L_t^{CLIP}} \left(\frac{\partial L_t^{CLIP}}{\partial p_t(\theta)A_t} \frac{\partial p_t(\theta)A_t}{\partial p_t(\theta)} + \frac{\partial L_t^{CLIP}}{\partial clip(p_t(\theta))A_t} \frac{\partial clip(p_t(\theta))A_t}{\partial clip(p_t(\theta))} \frac{\partial clip(p_t(\theta))}{\partial p_t(\theta)} \right) \frac{\partial p_t(\theta)}{\partial \pi_\theta(a_t|s_t)}$$

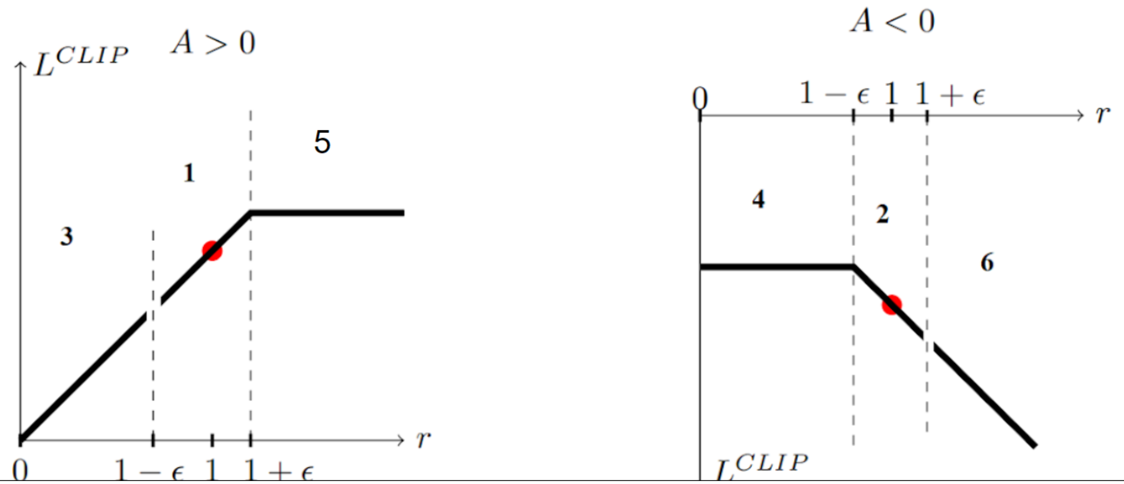
$$\begin{aligned} \frac{\partial -L_t^{CLIP}}{\partial \pi_\theta(a_t|s_t)} = -1 * \Bigg(& \begin{cases} 1 & \text{if } p_t(\theta)A_t \leq clip(p_t(\theta), 1-\epsilon, 1+\epsilon)A_t \\ 0 & \text{else} \end{cases} * A_t + \\ & \begin{cases} 1 & \text{if } clip(p_t(\theta), 1-\epsilon, 1+\epsilon)A_t < p_t(\theta)A_t \\ 0 & \text{else} \end{cases} * A_t * \\ & \begin{cases} 1 & \text{if } 1-\epsilon \leqslant p_t(\theta) \leqslant 1+\epsilon \\ 0 & \text{else} \end{cases} \Bigg) * \frac{1}{\pi_{\theta_{old}}(a_t|s_t)} \end{aligned}$$

直观看每段的梯度

university of groningen
faculty of science and engineering
Towards Delivering a Coherent Self-Contained Explanation of Proximal Policy Optimization
Daniel Bick daniel.bick@live.de

	$p_t(\theta)\quad > 0$	A_t	**Return Value of** min	**Objective is Clipped**	**Sign of Objective**	**Gradient**
1	$p_t(\theta) \in [1-\epsilon, 1+\epsilon]$	$+$	$p_t(\theta)A_t$	no	$+$	✓
2	$p_t(\theta) \in [1-\epsilon, 1+\epsilon]$	$-$	$p_t(\theta)A_t$	no	$-$	✓
3	$p_t(\theta) < 1-\epsilon$	$+$	$p_t(\theta)A_t$	no	$+$	✓
4	$p_t(\theta) < 1-\epsilon$	$-$	$(1-\epsilon)A_t$	yes	$-$	**0**
5	$p_t(\theta) > 1+\epsilon$	$+$	$(1+\epsilon)A_t$	yes	$+$	**0**
6	$p_t(\theta) > 1+\epsilon$	$-$	$p_t(\theta)A_t$	no	$-$	✓

$$p_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$$



- 策略只在两种情况下更新

  $p_t(\theta)$落在邻域里

  $p_t(\theta)$未落在邻域里但优势函数引领$p_t(\theta)$更靠近邻域