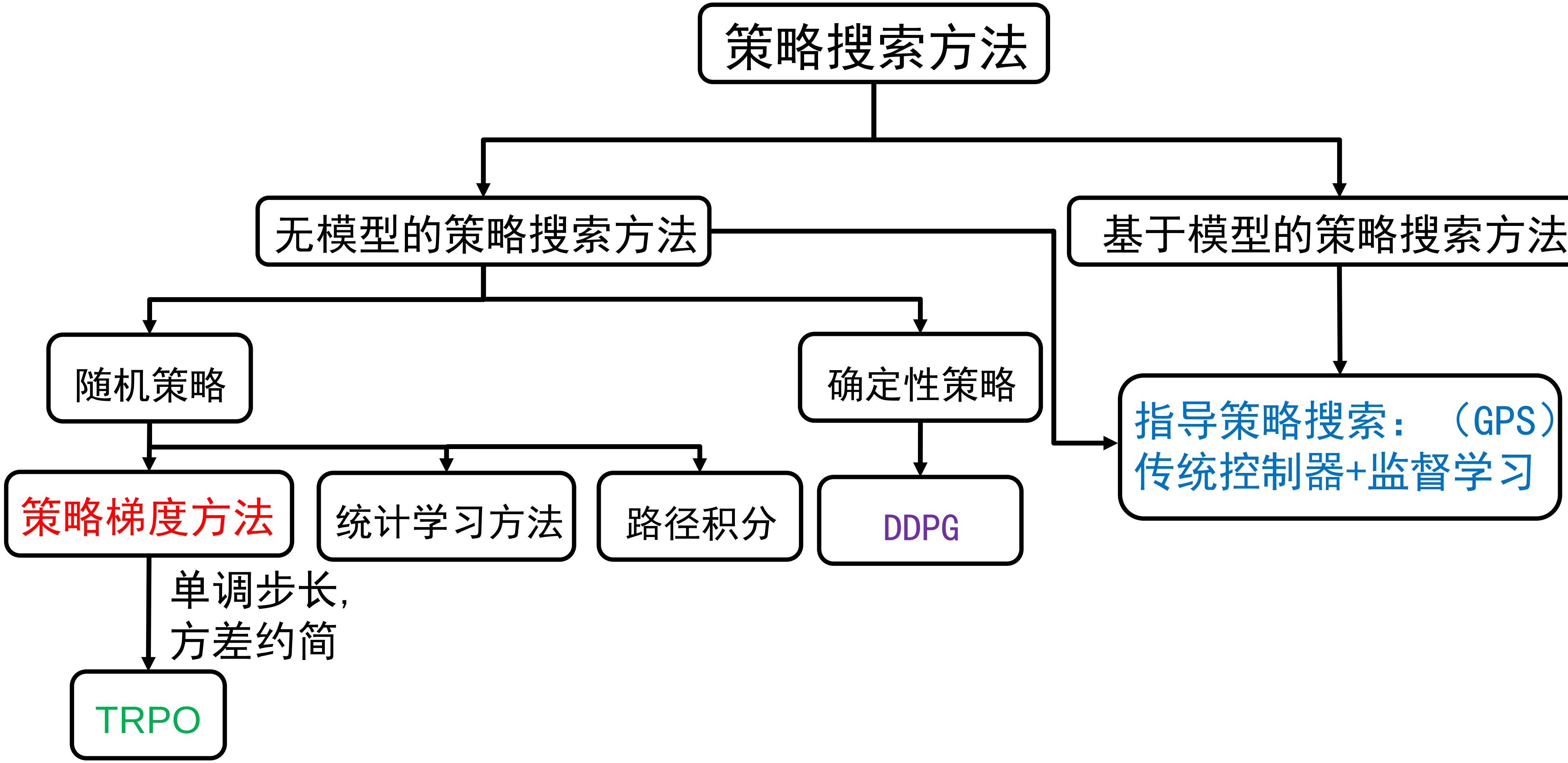
强化学习原理：
——深度确定性策略梯度

杨博渊

yby@nankai.edu.cn

2023.05.12

策略搜索方法分类


策略搜索方法
无模型的策略搜索方法
基于模型的策略搜索方法
随机策略
确定性策略
指导策略搜索：（GPS）
传统控制器+监督学习
策略梯度方法
统计学习方法
路径积分
DDPG
单调步长，
方差约简
TRPO

强化学习的目标函数

强化学习的目标函数：

$$J(\pi)=E[\sum_{k=t}^{\infty}\gamma^{k-t}r\left(s_k,a_k\right)]$$

如果表示从状态s经时间步 t 转移到状态s’ 的概率为：$p(s\rightarrow s',t,\pi)$

我们也可以表示折扣状态分布为：$\rho^{\pi}\left(s'\right):=\int_S\sum_{t=1}^{\infty}\gamma^{t-1}p_1(s)p(s\rightarrow s',t,\pi)ds$

强化学习目标可以写为：$J(\pi_\theta)=\int_S\rho^{\pi}\left(s\right)\int_A\pi_\theta\left(s,a\right)r(s,a)dads$

随机策略梯度理论

强化学习目标可以写为： $J(\pi_\theta)=\int_S \rho^\pi(s)\int_A \pi_\theta(s,a)r(s,a)dads$

策略梯度理论：

$$\nabla_\theta J(\pi_\theta)=\int_S \rho^\pi(s)\int_A \nabla_\theta \pi_\theta(a|s)Q^\pi(s,a)dads$$
$$=\mathbb{E}_{s\sim\rho^\pi,a\sim\pi_\theta}[\nabla_\theta \log\pi_\theta(a|s)Q^\pi(s,a)]$$

策略梯度很简单，尽管分布依赖于参数，但是策略梯度并不依赖于状态的分布

还没有解决的问题：如何估计行为值函数？

随机策略梯度理论

如何估计行为值函数？

$$\nabla_\theta J(\pi_\theta) = \int_S \rho^\pi(s) \int_A \nabla_\theta \pi_\theta(a|s) Q^\pi(s,a)\,dads$$
$$= \mathbb{E}_{s\sim\rho^\pi, a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s,a)\right]$$

1. 如果不用估计值，REINFORCE算法

2. 如果用函数逼近的方法来估计值函数，则称为Actor-Critic框架。

Actor调整策略网络的参数 π_θ

Critic调整行为值函数网络的参数 $Q^w(s,a)$

一般情况下，带入一个函数逼近器来代替行为值函数会引入偏差。与策略相容的无偏差的函数逼近器称为相容函数。

随机策略梯度的相容函数

策略梯度：

$$\nabla_\theta J(\pi_\theta) = \int_S \rho^\pi(s)\int_A \nabla_\theta \pi_\theta(a|s)Q^\pi(s,a)dads$$
$$= \mathbb{E}_{s\sim\rho^\pi,a\sim\pi_\theta}\left[\nabla_\theta \log\pi_\theta(a|s)Q^\pi(s,a)\right]$$

将真正的行为值函数 $Q^\pi(s,a)$ 用函数来逼近，用来逼近的函数应该满足以下相容性条件：

$$Q^w(s,a) = \nabla_\theta \log\pi_\theta(a|s)^T w$$

W使得如下目标函数最小：　$\varepsilon^2(w) = E_{s\sim\rho^\pi,a\sim\pi_\theta}\left[(Q^w(s,a) - Q^\pi(s,a))^2\right]$

随机 Off-policy Actor-Critic

采样策略为 β　　要评估的策略为 π

目标函数为：

$$J_{\beta}(\pi_{\theta}) = \int_{S} \rho^{\beta}(s) V^{\pi}(s) ds$$

$$= \int_{S} \int_{A} \rho^{\beta}(s) \pi_{\theta}(a|s) Q^{\pi}(s,a) dads$$

Off-policy 策略梯度为：

$$\nabla_{\theta} J_{\beta}(\pi_{\theta}) = E_{s\sim\rho^{\beta}, a\sim\beta}\left[\frac{\pi_{\theta}(a|s)}{\beta_{\theta}(a|s)} \nabla_{\theta} \log \pi_{\theta}(a|s) Q^{\pi}(s,a)\right]$$

随机策略 VS 确定性策略

随机策略：

$$\pi_\theta(a|s) = P[a|s;\theta]$$

在状态s，动作符合参数为θ 的概率分布。训练时按照该概率分布采样。

确定性策略：

$$a = \mu_\theta(s)$$

在状态s，动作是参数和状态的确定性函数

随机策略梯度对状态空间和动作空间积分。

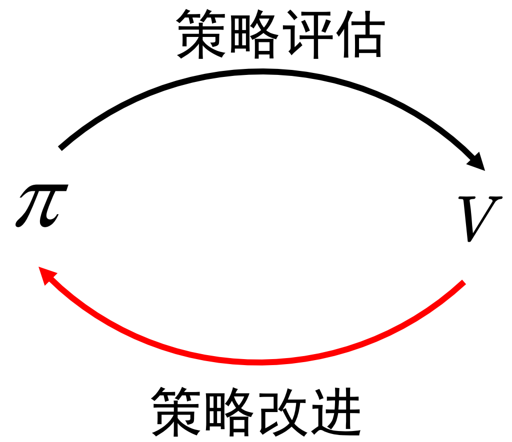
$$\nabla_\theta J(\pi_\theta) = E_{s\sim\rho^\pi, a\sim\pi_\theta}[\nabla_\theta \log\pi_\theta(a|s) Q^\pi(s,a)]$$

确定性策略梯度只对状态空间积分。

因此，计算随机策略梯度需要更多的样本，特别是当动作空间有很多维数时。

确定性策略梯度强化学习算法

强化学习的框架：广义策略迭代



对于参数化的策略，在进行策略改进时用的是梯度更新

确定性值函数

行为-值函数的定义：

$$Q^{\pi}(s_t,a_t)=\mathbb{E}_{r_{i\geq t},s_{i>t}\sim E,a_{i>t}\sim\pi}[R_t\mid s_t,a_t]$$

利用迭代方式可以得到：

$$Q^{\pi}(s_t,a_t)=\mathbb{E}_{r_{i\geq t},s_{t+1}\sim E,}[r(s_t,a_t)+\gamma\mathbb{E}_{a_{t+1}\sim\pi}[Q^{\pi}(s_{t+1},a_{t+1})]]$$

如果目标策略为确定性策略μ，则：

$$Q^{\mu}(s_t,a_t)=\mathbb{E}_{r_t,s_{t+1}\sim E,}[r(s_t,a_t)+\gamma Q^{\mu}(s_{t+1},\mu(s_{t+1}))]$$

期望仅仅依赖于环境

确定性策略梯度

策略改善：在策略梯度的方向上更新参数

$$\theta^{k+1} = \theta^{k} + \alpha E_{s \sim \rho^{\mu^{k}}}\left[\nabla_{\theta} Q^{\mu^{k}}(s, \mu_{\theta}(s))\right]$$

根据链式规则，得到确定性策略梯度理论：

$$\nabla_{\theta} J(\mu_{\theta}) = E_{s \sim \rho^{\mu}}\left[\nabla_{\theta} \mu_{\theta}(s) \nabla_{a} Q^{\mu}(s, a)\big|_{a=\mu_{\theta}(s)}\right]$$

确定性 Actor-Critic 算法

异策略AC的方法：

行动策略为随机策略 $\beta(s,a)$

要评估的策略为确定性策略：μ_θ

异策略确定性策略梯度：$\nabla_\theta J_\beta(\mu_\theta) = E_{s\sim\rho^\beta}\left[\nabla_\theta \mu_\theta(s)\nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}\right]$

更新过程：

Qlearning(off-policy)

$$\boxed{\delta_t = r_t + \gamma Q^w(s_{t+1},\mu_\theta(s_{t+1})) - Q^w(s_t,a_t)}$$

$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t,a_t)$$

$$\boxed{\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s_t)\nabla_a Q^w(s_t,a_t)|_{a=\mu_\theta(s)}}$$

确定性策略无需重采样

确定性策略梯度算法中如何评估行为值函数？

相容函数逼近

定理3：

函数逼近器 $Q^w(s,a)$ 是与确定性策略 $\mu_\theta(s)$ 及确定性策略梯度 $E[\nabla_\theta \mu_\theta(s) \nabla_a Q^w(s,a)|_{a=\mu_\theta(s)}]$ 相容条件：

1. $\nabla_a Q^w(s,a)|_{a=\mu_\theta(s)} = \nabla_\theta \mu_\theta(s)^{\mathrm{T}} w$

2. W最小化如下均方误差：

$$MSE(\theta,w) = E[\varepsilon(s;\theta,w)^T \varepsilon(s;\theta,w)]$$

其中： $\varepsilon(s;\theta,w) = \nabla_a Q^w(s,a)|_{a=\mu_\theta(s)} - \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}$

相容函数的讨论

1. $\nabla_a Q^w(s,a)|_{a=\mu_\theta(s)} = \nabla_\theta \mu_\theta(s)^{\mathrm{T}} w$

满足该条件的相容函数为：

$$Q^w(s,a) = \left(a - \mu_\theta(s)\right)^T \nabla_\theta \mu_\theta(s)^T w + V^\upsilon(s)$$

线性逼近函数对于预测全局的行为值函数并无用处，但是并不防碍作为局部 critic

相容函数的讨论

2. W最小化如下均方误差：

$$MSE(\theta, w) = E[\varepsilon(s;\theta,w)^T \varepsilon(s;\theta,w)]$$

其中： $\varepsilon(s;\theta,w) = \nabla_a Q^w(s,a)|_{a=\mu_\theta(s)} - \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}$

直接优化该目标函数不可行：因为真实的梯度 $\nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}$

在实际算法中：利用线性函数逼近器：$Q^w(s,a) = \phi(s,a)^{\mathrm{T}} w$ 和时间差分的方法。

不能严格满足第二个条件

DDPG

$$\delta_t = r_t + \gamma Q^w(s_{t+1}, \mu_\theta(s_{t+1})) - Q^w(s_t, a_t)$$

$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t, a_t)$$

$$\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s_t) \nabla_a Q^w(s_t, a_t)|_{a=\mu_\theta(s)}$$

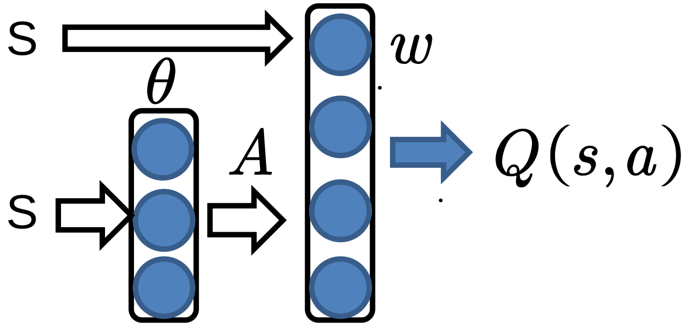
$Q^w(s,a)$， $\mu_\theta(s)$

利用神经网络进行逼近时，遇到的问题：

1. 神经网络的训练假设数据之间是独立同分布的，而强化学习顺序采集的数据明显存在相关性。
2. 利用mini批方法进行优化数据可以得到充分利用。

DQN的成功：

1. 经验回放
2. 独立的目标网络



$$\delta_t = r_t + \boxed{\gamma Q^w(s_{t+1}, \mu_\theta(s_{t+1}))} - Q^w(s_t, a_t)$$

$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t, a_t)$$

$$\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s_t) \nabla_a Q^w(s_t, a_t)|_{a=\mu_\theta(s)}$$

DDPG

DDPG算法伪代码

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer R
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  for t = 1, T **do**
    Select action $\boxed{a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t}$ according to the current policy and exploration noise
    行动策略为随机策略
    Execute action a_t and observe reward r_t and observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
    经验回放
    Set $\boxed{y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})}$
    目标网络
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q)^2)$
    Update the actor policy using the sampled gradient:

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

    Update the target networks:

$$\boxed{\begin{aligned} \theta^{Q'} &\leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \\ \theta^{\mu'} &\leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \end{aligned}}$$

    目标网络参数更新

  end for
end for

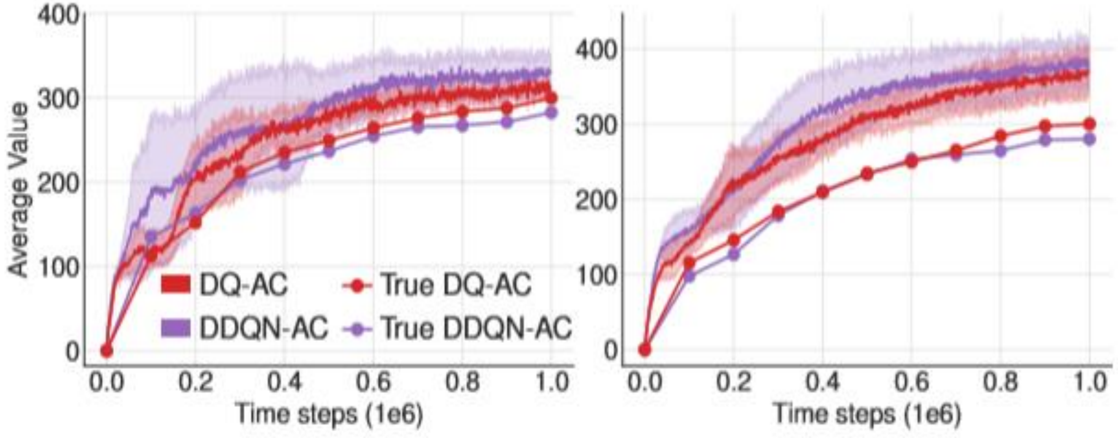
DDPG存在的问题

1. 值函数的过优估计问题

$$\delta_t = r_t + \gamma Q^w(s_{t+1}, \mu_\theta(s_{t+1})) - Q^w(s_t, a_t)$$

$$w_{t+1} = w_t + \alpha_w \delta_t \nabla_w Q^w(s_t, a_t)$$

$$\theta_{t+1} = \theta_t + \alpha_\theta \nabla_\theta \mu_\theta(s_t) \nabla_a Q^w(s_t, a_t)|_{a=\mu_\theta(s)}$$



(a) Hopper-v1 (b) Walker2d-v1

解决问题的方法:

$$y_1 = r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \pi_{\phi_1}(s'))$$

DDPG扩展TD3

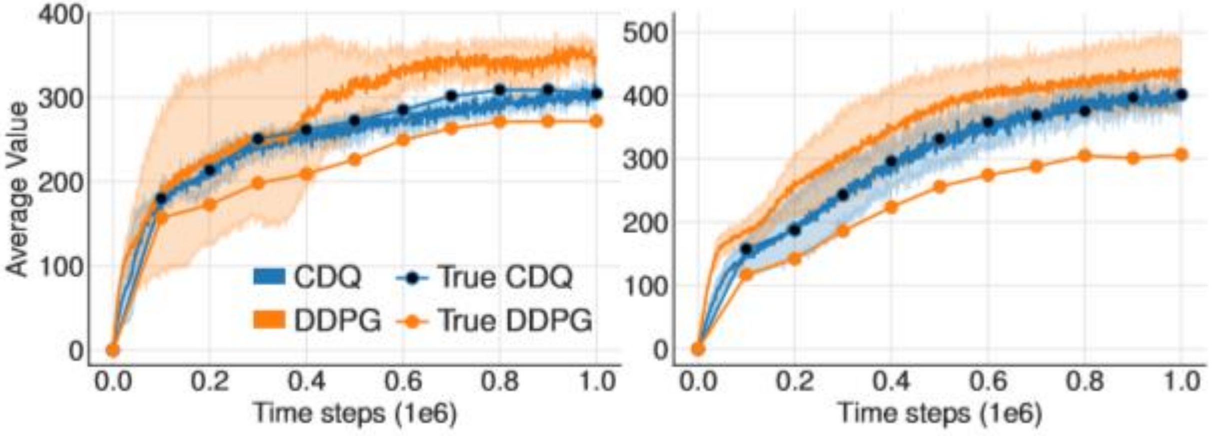
2. 值函数高的方差

解决方法：独立的目标网络

目标策略光滑：

$$y = r + Q_{\theta'}(s', \pi_{\phi'}(s') + \epsilon),$$

$$\epsilon \sim clip(\mathcal{N}(0, \sigma), -c, c)$$



(a) Hopper-v1 (b) Walker2d-v1

3. 策略网络与值函数网络之间存在耦合关系

Twin Delay 双延迟：目标网络更新，策略更新延迟

DDPG扩展TD3

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 $a \sim \pi_\phi(s) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ 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 $\theta_i \leftarrow \text{argmin}_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$
  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

Environment	**TD3**	**DDPG**	**Our DDPG**	**PPO**	**TRPO**	**ACKTR**	**SAC**
HalfCheetah	**9636.95 ± 859.065**	3305.60	8577.29	1795.43	-15.57	1450.46	2347.19
Hopper	**3564.07 ± 114.74**	2020.46	1860.02	2164.70	2471.30	2428.39	2996.66
Walker2d	**4682.82 ± 539.64**	1843.85	3098.11	3317.69	2321.47	1216.70	1283.67
Ant	**4372.44 ± 1000.33**	1005.30	888.77	1083.20	-75.85	1821.94	655.35
Reacher	**-3.60 ± 0.56**	-6.51	**-4.01**	-6.18	-111.43	-4.26	-4.44
InvPendulum	**1000.00 ± 0.00**	**1000.00**	**1000.00**	**1000.00**	985.40	**1000.00**	**1000.00**
InvDoublePendulum	**9337.47 ± 14.96**	**9355.52**	8369.95	8977.94	205.85	9081.92	8487.15

D4PG：Distributed Distributional Deep Deterministic Policy Gradient

Actor-critic算法中策略梯度依赖于学到的critic, 这就意味着任何对critic学习过程进行改善的方法将直接改善actor更新质量。

值分布建模了由于函数逼近等带来的随机性，因此利用值分布进行更新critic将直接改善学习算法的表现。

用到的技巧：

1. 值分布式强化学习更细critic
2. 分布式并行Actor
3. N-step回报
4. 优先经验回访采数据

D4PG：Distributed Distributional Deep Deterministic Policy Gradient

Algorithm 1 D4PG

Input: batch size M, trajectory length N, number of actors K, replay size R, exploration constant ϵ, initial learning rates α_0 and β_0

Initialize network weights (θ, w) at random
Initialize target weights $(\theta', w') \leftarrow (\theta, w)$
Launch K actors and replicate network weights (θ, w) to each actor
for $t = 1, \dots, T$ **do**
    Sample M transitions $(\mathbf{x}_{i:i+N}, \mathbf{a}_{i:i+N-1}, r_{i:i+N-1})$ of length N from replay with priority p_i
    Construct the target distributions $Y_i = \left(\sum_{n=0}^{N-1} \gamma^n r_{i+n}\right) + \gamma^N Z_{w'}(\mathbf{x}_{i+N}, \pi_{\theta'}(\mathbf{x}_{i+N}))$
    Note, although not denoted the target Y_i may be projected (e.g. for Categorical value distributions).
    Compute the actor and critic updates

$$\delta_w = \frac{1}{M} \sum_i \nabla_w (R p_i)^{-1} d(Y_i, Z_w(\mathbf{x}_i, \mathbf{a}_i))$$

$$\delta_\theta = \frac{1}{M} \sum_i \nabla_\theta \pi_\theta(\mathbf{x}_i)\, \mathbb{E}[\nabla_{\mathbf{a}} Z_w(\mathbf{x}_i, \mathbf{a})]\Big|_{\mathbf{a}=\pi_\theta(\mathbf{x}_i)}$$

    Update network parameters $\theta \leftarrow \theta + \alpha_t\, \delta_\theta$, $w \leftarrow w + \beta_t\, \delta_w$
    If $t = 0 \mod t_{\text{target}}$, update the target networks $(\theta', w') \leftarrow (\theta, w)$
    If $t = 0 \mod t_{\text{actors}}$, replicate network weights to the actors
end for
return policy parameters θ

D4PG：Distributed Distributional Deep Deterministic Policy Gradient

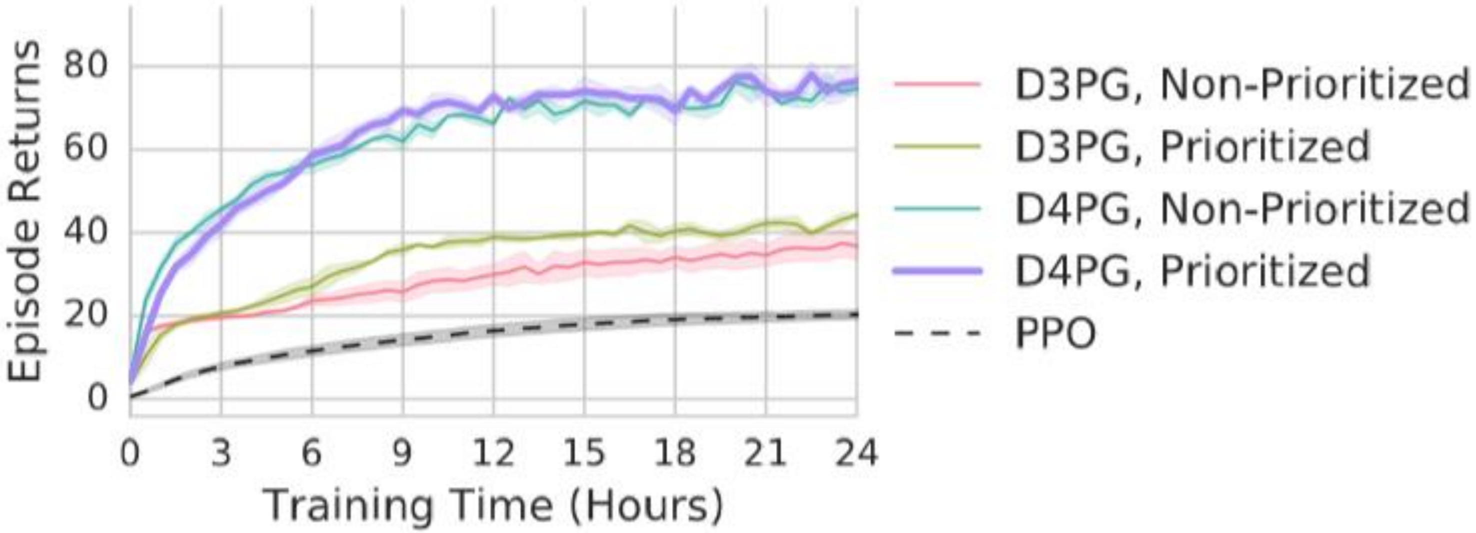


Figure 6: Experimental results for the three-dimensional (humanoid) parkour domain.

作业

两个相容近似函数证明