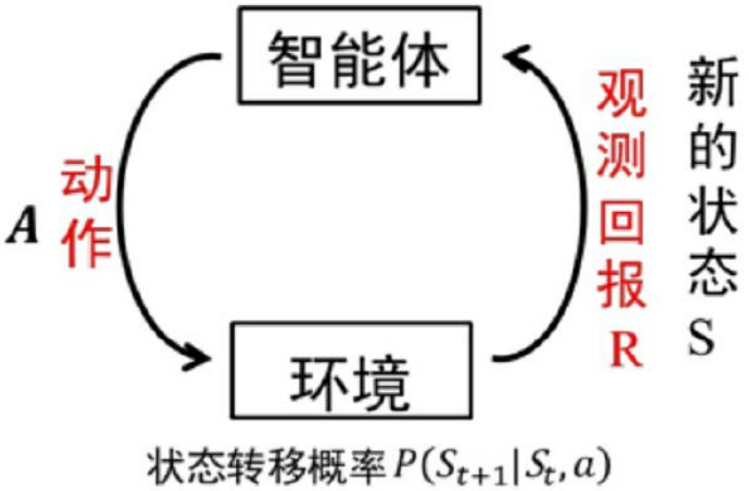
强化学习基础

教材：《深入浅出强化学习》

强化学习如何解决问题

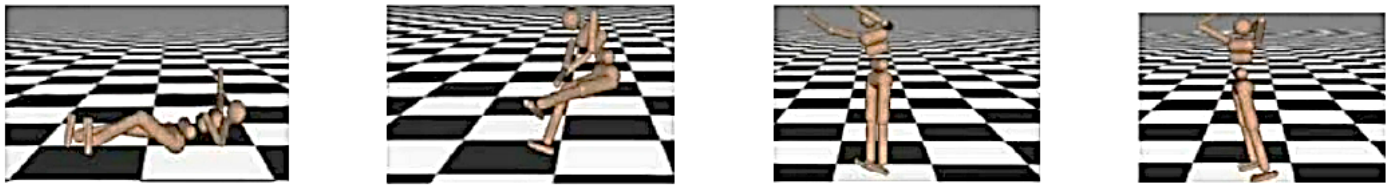


智能体在完成某项任务时，首先通过动作A与周围环境进行交互，在动作A和环境的作用下，智能体会产生新的状态，同时环境会给出一个立即回报。如此循环下去，智能体与环境不断地交互从而产生很多数据。强化学习算法利用产生的数据修改自身的动作策略，再与环境交互，产生新的数据，并利用新的数据进一步改善自身的行为，经过数次迭代学习后，智能体能最终学到完成相应任务的最优动作（最优策略）。

强化学习解决什么问题

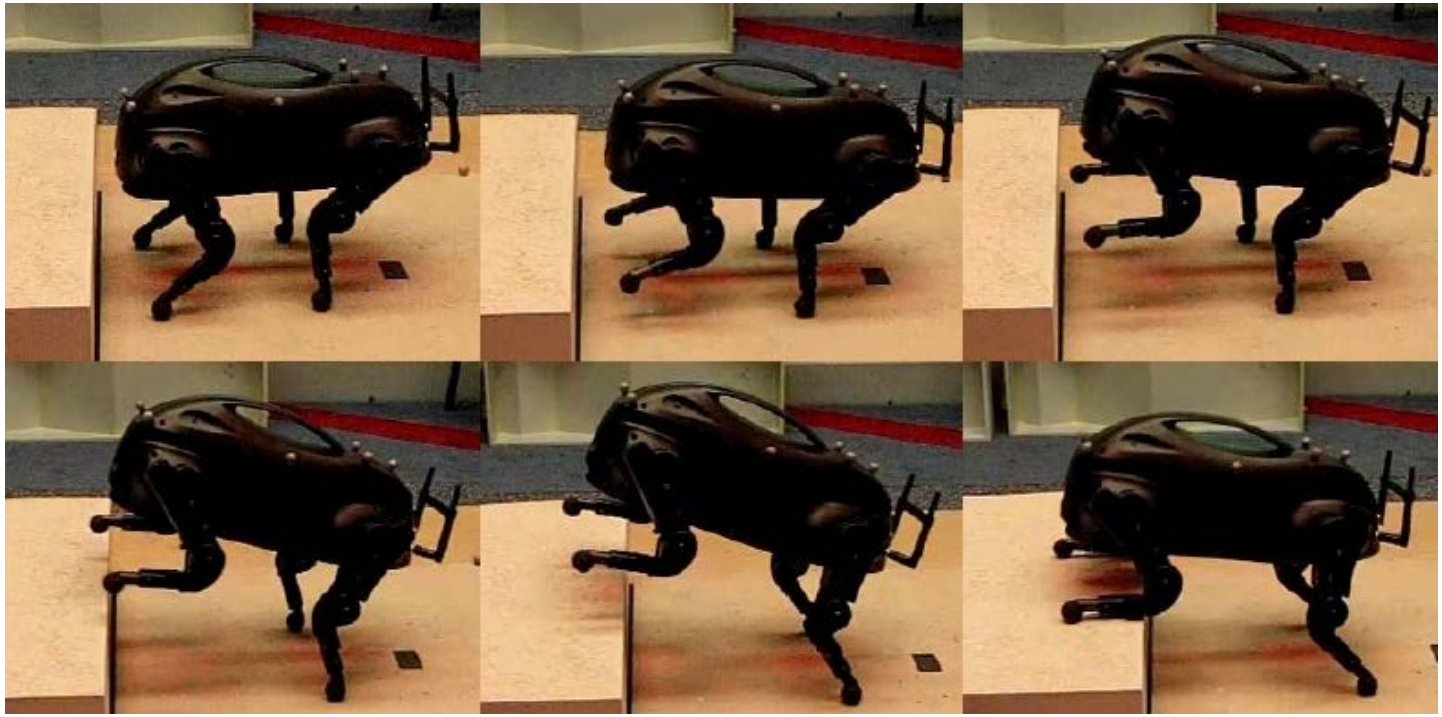
图A 非线性系统二级倒立摆

图B AlphaGo与柯洁第二盘棋



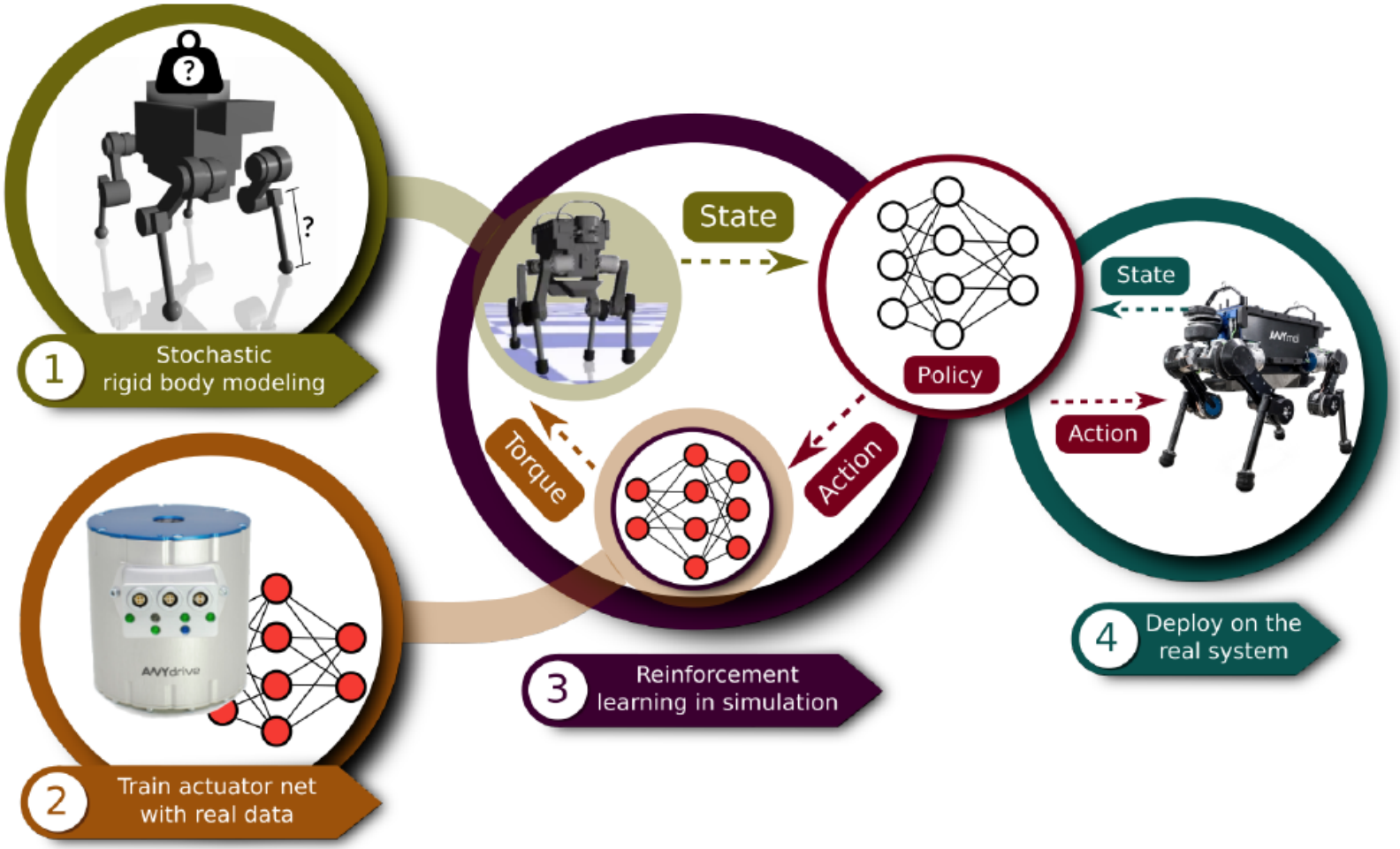
图C 机器人学习站立

强化学习在机器人领域中的应用



强化学习为机器人技术提供了一个框架和一组工具，用于设计复杂和难以工程实现的行为。

强化学习在机器人控制中的应用



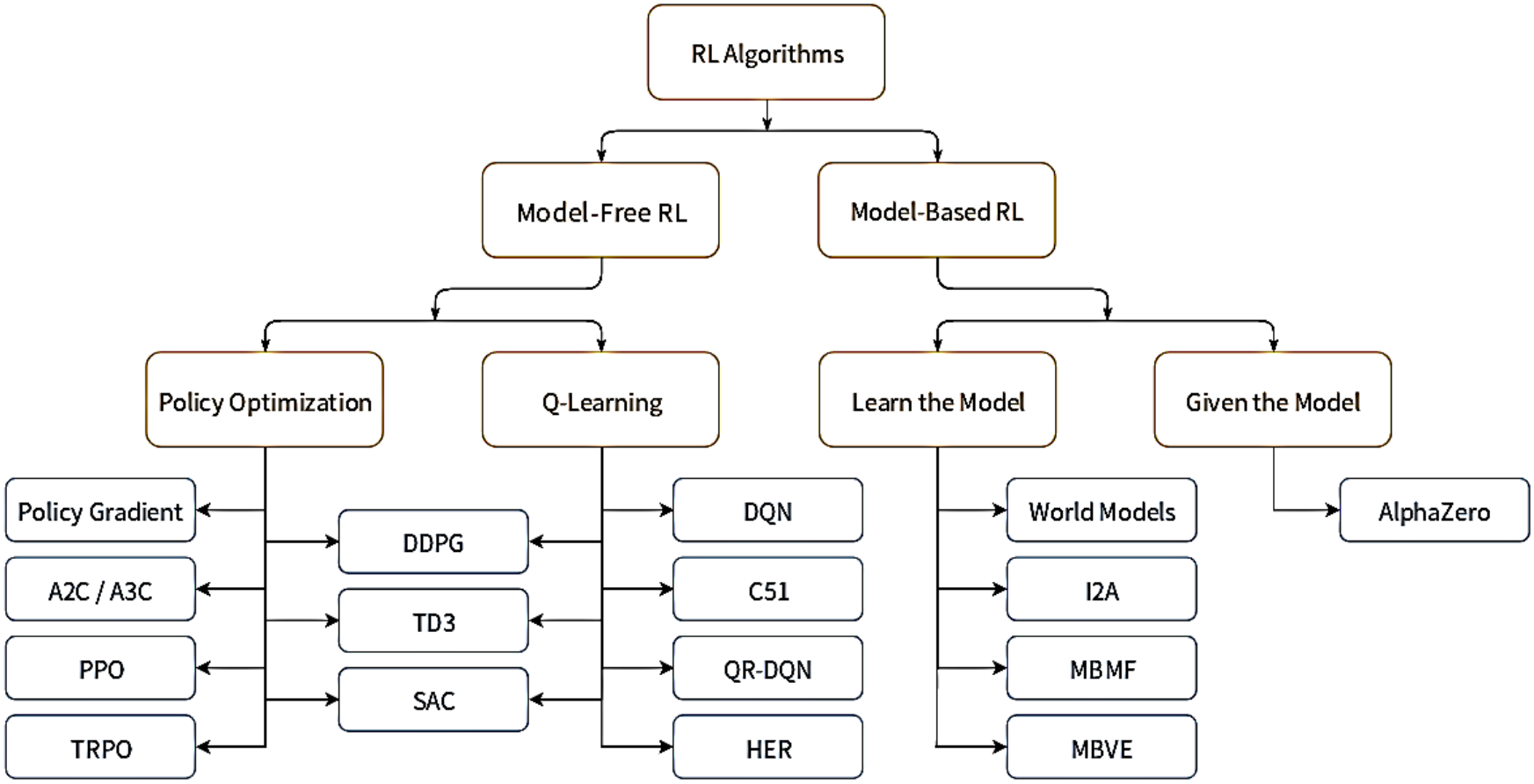
强化学习的分类

- **根据强化学习算法是否依赖模型可以分为基于模型的强化学习算法和无模型的强化学习算法。**这两类算法的共同点是通过与环境交互获得数据，不同点是利用数据的方式不同。基于模型的强化学习算法利用与环境交互得到的数据学习系统或者环境模型，再基于模型进行序贯决策。无模型的强化学习算法则是直接利用与环境交互获得的数据改善自身的行为。两类方法各有优缺点，一般来讲基于模型的强化学习算法效率要比无模型的强化学习算法效率更高，因为智能体在探索环境时可以利用模型信息。但是，有些根本无法建立模型的任务只能利用无模型的强化学习算法。由于无模型的强化学习算法不需要建模，所以和基于模型的强化学习算法相比，更具有通用性。

强化学习的分类

- **根据策略的更新和学习方法，强化学习算法可分为基于值函数的强化学习算法、基于直接策略搜索的强化学习算法以及AC的方法。**所谓基于值函数的强化学习方法是指学习值函数，最终的策略根据值函数贪婪得到。也就是说，任意状态下，值函数最大的动作为当前最优策略。基于直接策略搜索的强化学习算法，一般是将策略参数化，学习实现目标的最优参数。基于AC的方法则是联合使用值函数和直接策略搜索。

强化学习的分类


RL Algorithms
Model-Free RL
Model-Based RL
Policy Optimization
Q-Learning
Learn the Model
Given the Model
Policy Gradient
A2C / A3C
PPO
TRPO
DDPG
TD3
SAC
DQN
C51
QR-DQN
HER
World Models
I2A
MBMF
MBVE
AlphaZero

马尔科夫决策过程

- **马尔科夫性** 所谓马尔科夫性是指系统的下一个状态st+1仅与当前状态st有关，而与以前的状态无关。

马尔科夫性描述的是每个状态的性质，但真正有用的是如何描述一个状态序列。数学中用来描述随机变量序列的学科叫随机过程。所谓随机过程就是指随机变量序列。若随机变量序列中的每个状态都是马尔科夫的，则称此随机过程为马尔科夫随机过程。

马尔科夫决策过程

- **马尔科夫过程** 马尔科夫过程是一个二元组（S，P），且满足：S是有限状态集合，P是状态转移概率。状态转移概率矩阵为：

$$P=\begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & \vdots & \vdots \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}$$

马尔科夫决策过程

- 强化学习的目标是给定一个马尔科夫决策过程，寻找最优策略。所谓策略是指状态到动作的映射，策略常用符号 π 表示，它是指给定状态s时，动作集上的一个分布，即

$$\pi(a \mid s) = p\left[A_t = a \mid S_t = s\right]$$

公式的含义是：策略 π 在每个状态s执行一个动作概率。如果给出的策略 π 是确定性的，那么策略 π 在每个状态s执行一个确定的动作。

状态值函数

- 当智能体采用策略π时，累积回报服从一个分布，累积回报在状态s处的期望值定义为状态值函数：

$$\upsilon_{\pi}(s)=\mathrm{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k}R_{t+k+1}\mid S_{t}=s\right]$$

状态-行为值函数

- 当智能体采用策略π时，累积回报服从一个分布，累积回报在状态s处并且执行动作a后的期望值定义为状态-行为值函数：

$$q_\pi(s,a) = E_\pi\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \mid S_t = s, A_t = a\right]$$

贝尔曼方程

- 状态值函数的贝尔曼方程：

$$\upsilon(s) = E\left[R_{t+1} + \gamma\upsilon(S_{t+1}) \mid S_t = s\right]$$

- 状态-行为值函数的贝尔曼方程：

$$q_\pi(s,a) = E_\pi\left[R_{t+1} + \gamma q(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a\right]$$

- 在环境模型已知的情况下，值函数的计算用到了bootstrapping的方法。所谓bootstrpping本意是指自举，此处是指当前值函数的计算用到了后继状态的值函数。

强化学习算法的形式化描述

我们定义一个离散时间有限范围的折扣马尔科夫决策过程M=（S，A，P，r，ρ_0，γ，T），其中S为状态集，A为动作集，P：S×A×S→R是转移概率，r：S×A→[-R_{max} ，R_{max}]为立即回报函数，ρ_0：S→R是初始状态分布，γ∈[0，1]为折扣因子，T为水平范围（其实就是步数）。τ为一个轨迹序列，即τ=（s_0，a_0，s_1，a_1，…），累积回报为$R=\sum_{t=0}^{T}\gamma^t r_t$，强化学习的目标是找到最优策略π，使得该策略下的累积回报期望最大，即 $\max_{\pi}\sum R(\tau)p_{\pi}(\tau)d\tau$。

基于值函数的强化学习方法

- 在环境模型已知的情况下，值函数的计算用到了bootstrapping的方法。再利用值函数改进当前策略。
- 当环境模型未知时，要评估智能体的当前策略，我们可以利用策略产生很多次试验，每次试验都是从任意的初始状态开始直到终止，先进行策略评估，也就是计算当前策略所对应的值函数，再利用值函数改进当前策略。

基于值函数的强化学习方法

- 若行动策略和评估及改善的策略是同一个策略，我们称为on-policy，可翻译为同策略。
- 若行动策略和评估及改善的策略是不同的策略，我们称为off-policy，可翻译为异策略。异策略可以保证充分的探索性。

基于直接策略搜索的强化学习方法

策略搜索是将策略参数化，即 $\pi_\theta(s)$：利用参数化的线性函数或非线性函数（如神经网络）表示策略，寻找最优的参数，使强化学习的目标——累积回报的期望 $E\left[\sum_{t=0}^{H} R(s_t)|\pi_\theta\right]$ 最大。

基于直接策略搜索的强化学习方法

- 在值函数的方法中，我们迭代计算的是值函数，再根据值函数改善策略；
- 而在策略搜索方法中，我们直接对策略进行迭代计算，也就是迭代更新策略的参数值，直到累积回报的期望最大，此时的参数所对应的策略为最优策略。
- 直接策略搜索的方法主要应用在机器人和游戏等领域。

基于直接策略搜索的强化学习方法

用τ表示一组状态-行为序列$s_0, u_0, \cdots, s_H, u_H$。

符号$R(\tau) = \sum_{t=0}^{H} R(s_t, u_t)$表示轨迹$\tau$的回报，$P(\tau;\theta)$表示轨迹$\tau$出现的概率；强化学习的目标函数可表示为

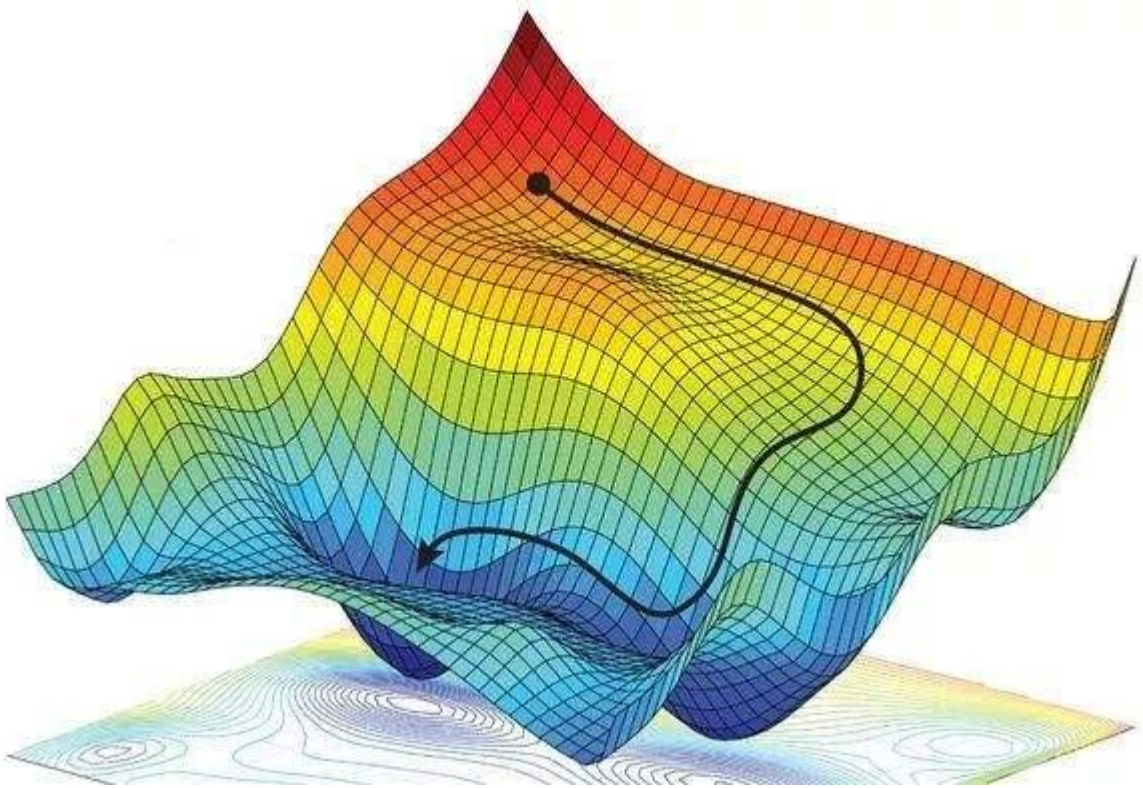
$$U(\theta) = E\left(\sum_{t=0}^{H} R(s_t, u_t); \pi_\theta\right) = \sum_{\tau} P(\tau;\theta) R(\tau)$$

强化学习的目标是找到最优参数，使得

$$\max_{\theta} U(\theta) = \max_{\theta} \sum P(\tau;\theta) R(\tau)。$$

基于直接策略搜索的强化学习方法

- 这时，策略搜索方法实际上变成了一个优化问题。解决优化问题有很多方法，比如最速下降法、牛顿法、内点法等。



基于直接策略搜索的强化学习方法

其中最简单、也最常用的是最速下降法，此处称为策略梯度的方法，即 $\theta_{\text{new}} = \theta_{\text{old}} + \alpha \nabla_\theta U(\theta)$，问题的关键是如何计算策略梯度 $\nabla_\theta U(\theta)$。

我们对目标函数求导：

$$\nabla_\theta U(\theta) = \nabla_\theta \sum_\tau P(\tau;\theta) R(\tau)$$

$$= \sum_\tau \nabla_\theta P(\tau;\theta) R(\tau)$$

$$= \sum_\tau \frac{P(\tau;\theta)}{P(\tau;\theta)} \nabla_\theta P(\tau;\theta) R(\tau)$$

$$= \sum_\tau P(\tau;\theta) \frac{\nabla_\theta P(\tau;\theta) R(\tau)}{P(\tau;\theta)}$$

$$= \sum_\tau P(\tau;\theta) \nabla_\theta \log P(\tau;\theta) R(\tau)$$

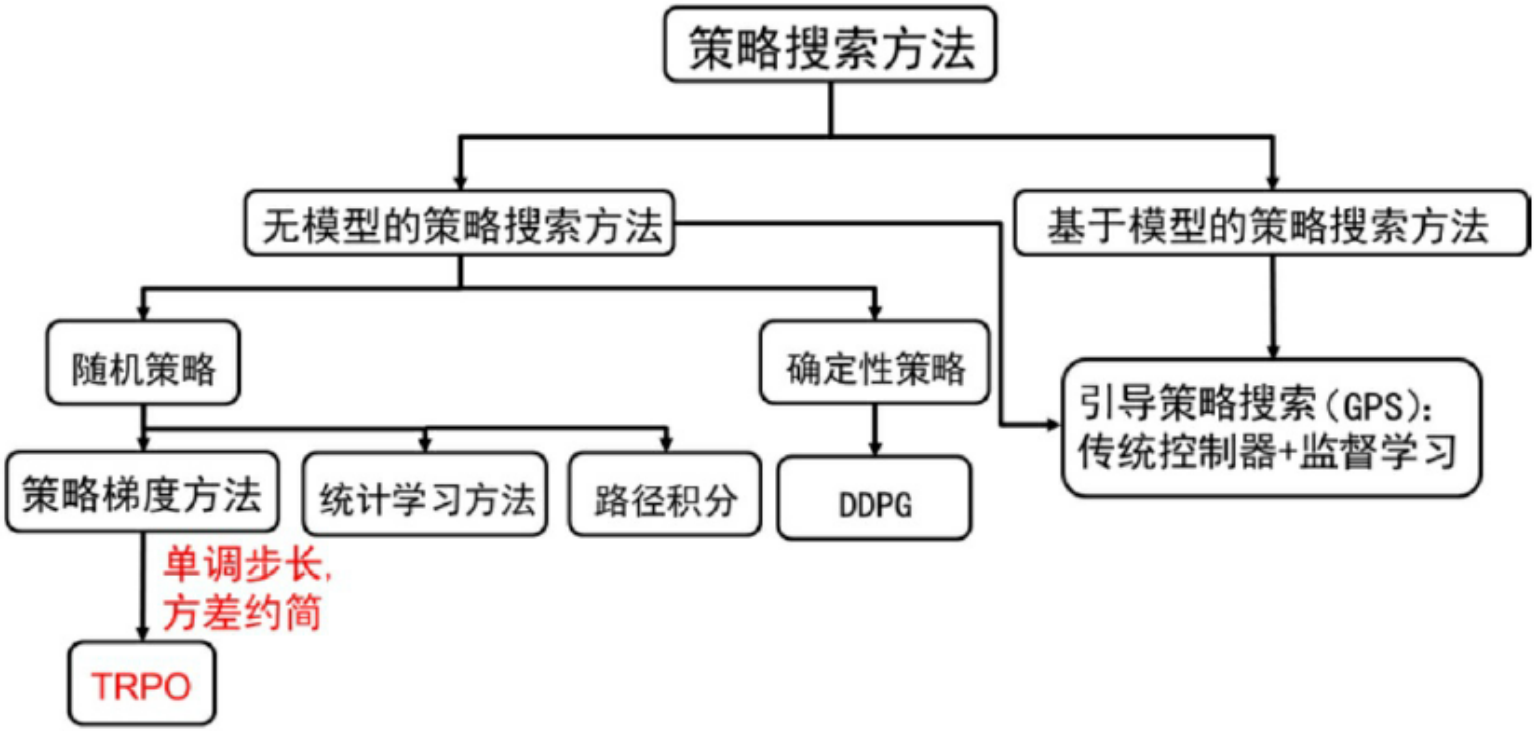
基于直接策略搜索的强化学习方法

最终策略梯度变成求$\nabla_\theta \log P(\tau;\theta) R(\tau)$的期望，这可以利用经验平均估算。因此，当利用当前策略π_θ采样m条轨迹后，可以利用m条轨迹的经验平均逼近策略梯度：

$$\nabla_\theta U(\theta) \approx \hat{g} = \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \log P(\tau;\theta) R(\tau)$$

基于置信域策略优化的强化学习方法

- 置信域策略优化（TRPO，Trust Region Policy Optimization）在强化学习体系中的位置：

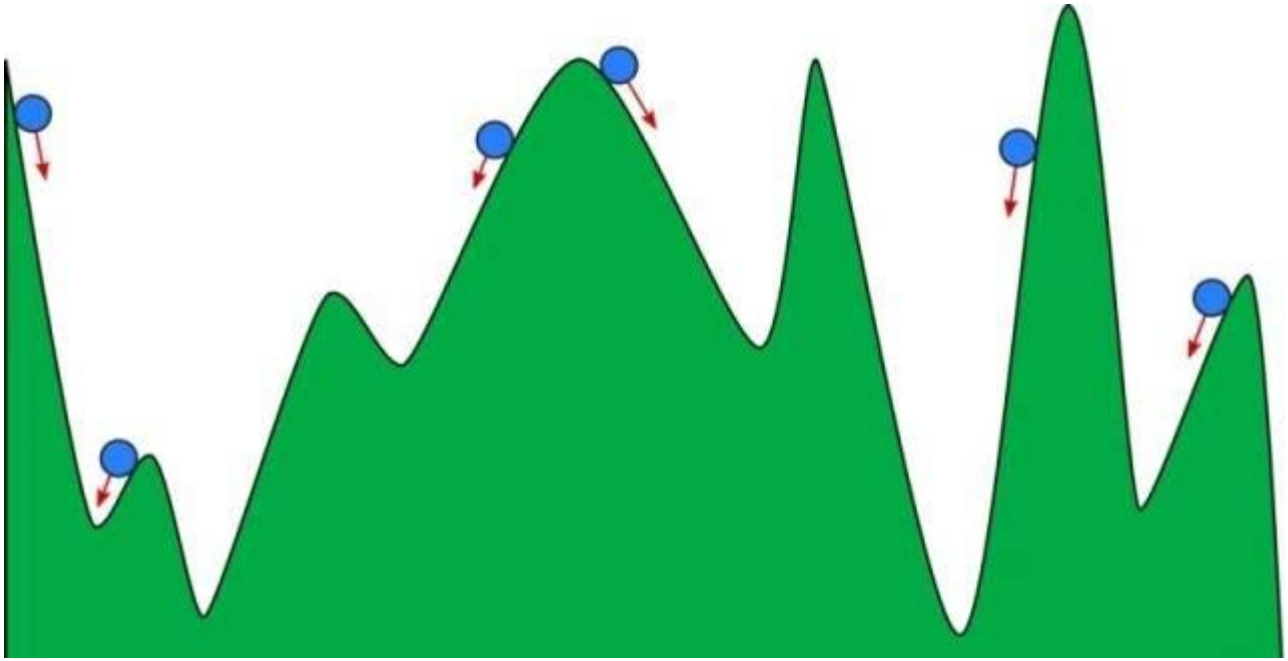


基于置信域策略优化的强化学习方法

- **什么才是合适的步长？**

策略梯度算法的硬伤就在更新步长，当步长不合适时，更新的参数所对应的策略是一个更不好的策略，当利用这个更不好的策略采样学习时，再次更新的参数会更差，因此很容易导致越学越差，最后崩溃。所以，合适的步长对于强化学习非常关键。

合适的步长是指当策略更新后，回报函数的值不能更差。



基于置信域策略优化的强化学习方法

- **什么才是合适的步长？**

一个自然的想法是能否将新的策略所对应的回报函数分解成旧的策略所对应的累积回报函数加其他项。这样，只要新的策略所对应的其他项大于等于零，那么新的策略就能保证回报函数单调不减。

基于置信域策略优化的强化学习方法

- 强化学习的累积回报函数为：

$$\eta(\tilde{\pi}) = E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^{t}(r(s_t))\right]$$

基于置信域策略优化的强化学习方法

- TRPO的起点是这样一个等式，我们只需要使这个目标函数其最大，就可以让两个策略的累积回报差距最大，以至于非负（为负我们就不接受新策略，认为当前策略已经无法改进）：

$$\eta(\tilde{\pi}) = \eta(\pi) + E_{s_0, a_0, \cdots \sim \tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t)\right]$$

这里我们用π表示旧的策略，用$\tilde{\pi}$表示新的策略。其中

$$A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s) = E_{s' \sim P(s'|s,a)}[r(s) + \gamma V^\pi(s') - V^\pi(s)]$$

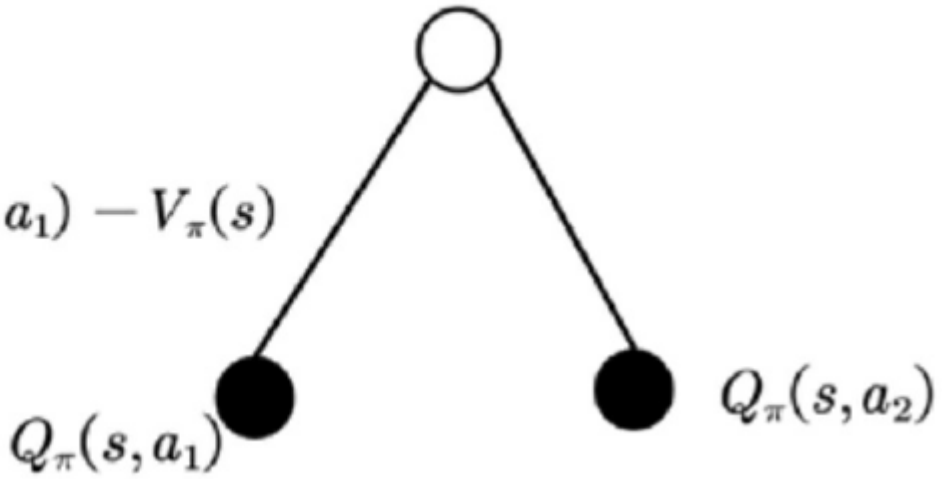
是优势函数。

基于置信域策略优化的强化学习方法

- 如何理解优势函数？

$$V_\pi(s) = \pi(a_1|s)Q_\pi(s,a_1) + \pi(a_2|s)Q_\pi(s,a_2)$$

$$A(s,a_1) = Q_\pi(s,a_1) - V_\pi(s)$$



值函数是该状态下所有动作值函数关于动作概率的平均值。而动作值函数是单个动作所对应的值函数，优势函数能评价当前动作值函数相对于平均值的大小。注意状态值函数和状态动作值函数都是可以相互计算的，参考教材。

基于置信域策略优化的强化学习方法

证明

$$E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t A_{\pi}(s_t,a_t)\right]$$

$$=E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s)+\gamma V^{\pi}(s_{t+1})-V^{\pi}(s_t)\right)\right]$$

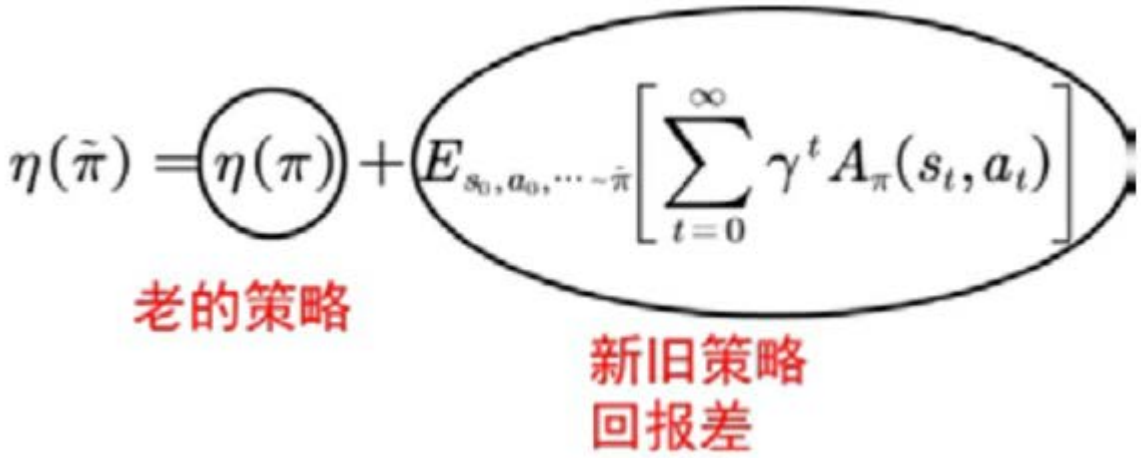
$$=E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t)\right)+\sum_{t=0}^{\infty}\gamma^t\left(\gamma V^{\pi}(s_{t+1})-V^{\pi}(s_t)\right)\right]$$

$$=E_{\tau|\tilde{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t\left(r(s_t)\right)\right]+E_{s_0}\left[-V^{\pi}(s_0)\right]$$

$$=\eta(\tilde{\pi})-\eta(\pi)$$

基于置信域策略优化的强化学习方法

- 如何理解？



基于置信域策略优化的强化学习方法

优势函数的期望可以写成下面这样：

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\tilde{\pi}) \sum_{a} \tilde{\pi}(a|s) \gamma^t A_{\pi}(s,a) \qquad (8.3)$$

其中$P(s_t = s|\tilde{\pi})\tilde{\pi}(a|s)$为$(s,a)$的联合概率，

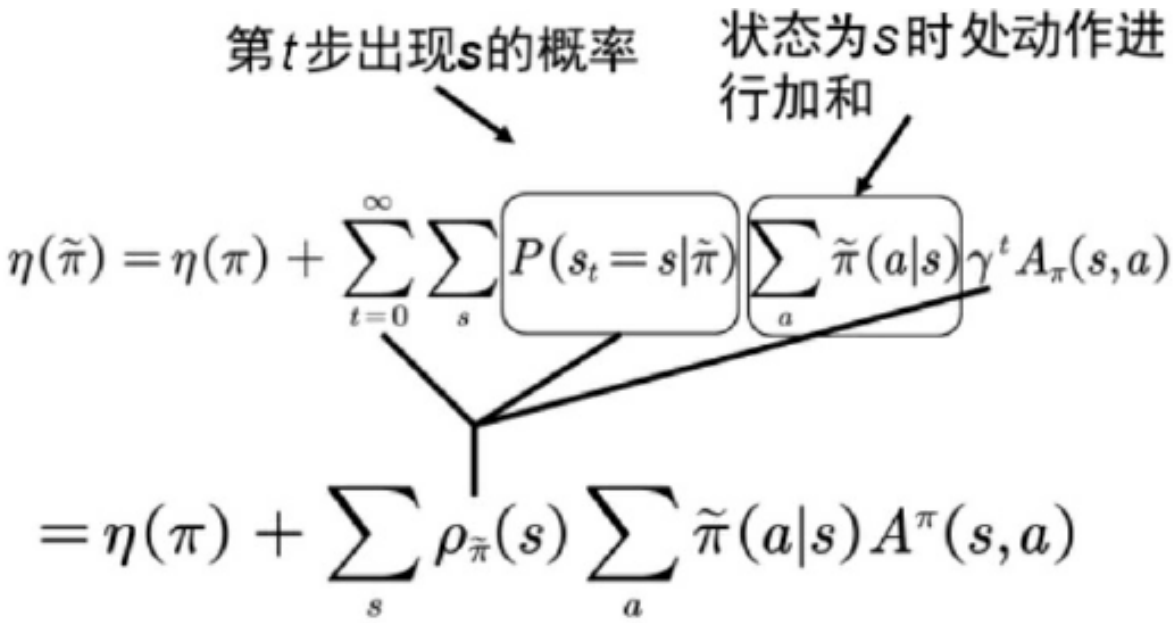
$\sum_{a} \tilde{\pi}(a|s) \gamma^t A_{\pi}(s,a)$为求对动作$a$ 的边际分布，也就是说在状态s下对整个动作空间求和；$\sum_{s} P(s_t = s|\tilde{\pi})$为求对状态$s$ 的边际分布，

即对整个状态空间求和；$\sum_{t=0}^{\infty} \sum_{s} P(s_t = s|\tilde{\pi})$求整个时间序列的和。

基于置信域策略优化的强化学习方法

我们定义$\rho_\pi(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \cdots$

则



基于置信域策略优化的强化学习方法

- 此时代价目标函数转换为：

$$L_{\pi}(\tilde{\pi}) = \eta(\pi) + \sum_{s} \rho_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) A^{\pi}(s,a)$$

基于置信域策略优化的强化学习方法

- TRPO算法在推导过程中运用了四个技巧。第一，采用旧策略所对应的状态分布。这是对原代价函数的第一次近似。其实，当新旧参数很接近时，我们用旧的状态分布代替新的状态分布也是合理的：

$$L_{\pi}(\tilde{\pi}) = \eta(\pi) + \sum_{s} \rho_{\pi}(s) \sum_{a} \tilde{\pi}(a|s) A^{\pi}(s,a)$$

此时第二项中只有后半部分是新策略，是具有需要优化的参数的，优化的时候仅仅对后面的因子进行求导即可，而前面的因子已经是一个已知项。由于改进后的代价函数与原代价函数是一阶近似，不影响梯度下降，所以这样做是合理的。

基于置信域策略优化的强化学习方法

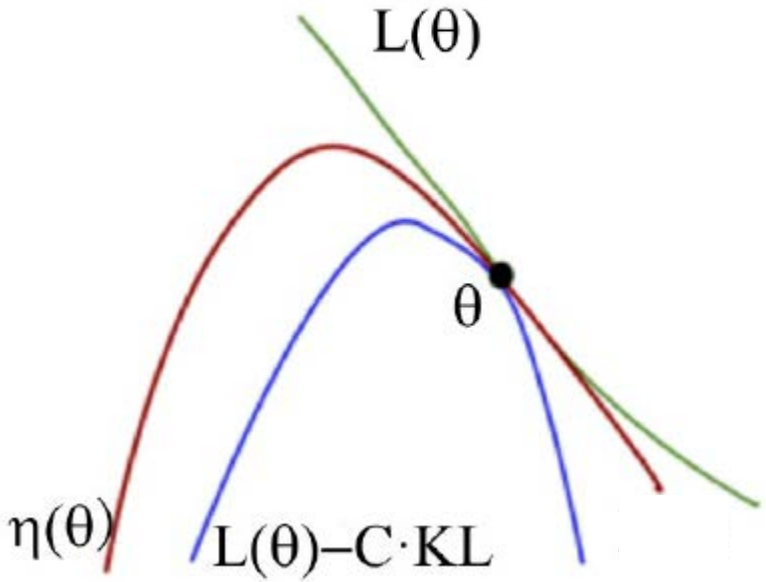
- TRPO的第二个技巧是利用重要性采样处理动作分布：

$$L_{\pi}(\tilde{\pi}) = \eta(\pi) + E_{s\sim\rho_{\theta_{old}}, a\sim\pi_{\theta_{old}}}\left[\frac{\tilde{\pi}_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s,a)\right]$$

这样进一步优化了原代价函数的采样合理性，避免出现采样方差过大的情况，采样更加能够反映真实情况。

基于置信域策略优化的强化学习方法

- 如何理解？



一阶近似的L函数实质上从优化的角度来看，梯度方向是正确的，接下来就是保证这种近似能够代替原目标函数来进行优化即可。

基于置信域策略优化的强化学习方法

- 经过证明，下面的不等式成立：

$$\eta(\tilde{\pi}) \geqslant L_{\pi}(\tilde{\pi}) - CD_{\mathrm{KL}}^{\max}(\pi, \tilde{\pi})$$

$$\text{where } C = \frac{2\varepsilon\gamma}{(1-\gamma)^2}$$

而C是非负的，那么说明不等号右边实际上是原目标函数的一个下界函数。我们如果能够使得这个下界函数都是最大的，那么原目标函数就会是最大的。我们自然就想用这个下界函数作为目标函数。

这里涉及一个KL散度的概念，KL散度是衡量两个分布差别的量。

基于置信域策略优化的强化学习方法

- **进一步简化：**

$$\begin{aligned}&\text{maximize}_{\pi} L_{\pi_{\text{old}}}(\pi)\\&\text{s.t.}\quad D_{\text{KL}}^{\max}(\pi_{\text{old}},\pi)\leqslant\delta\end{aligned}$$

我们将一个无约束问题转换成了一个有约束（信任域δ）问题。实质上由于：

$$L_{\pi}(\tilde{\pi})=\eta(\pi)+E_{s\sim\rho_{\theta_{old}},a\sim\pi_{\theta_{old}}}\left[\frac{\tilde{\pi}_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\theta_{old}}(s,a)\right]$$

前一项在优化中是常量，因此目标函数进一步简化为：

$$\begin{aligned}&\underset{\theta}{\text{maximize}}\,E_{s\sim\rho_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_{\theta}(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}A_{\theta_{\text{old}}}(s,a)\right]\\&\text{subject to}\ D_{\text{KL}}^{\max}(\theta_{\text{old}},\theta)\leqslant\delta\end{aligned}$$

基于置信域策略优化的强化学习方法

- 遗憾的是，上述问题是一个不可解问题，这就引出TRPO算法的第三个技巧，利用平均KL散度代替最大KL散度，问题就转化为：

$$\underset{\theta}{\text{maximize}} E_{s\sim\rho_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}A_{\theta_{\text{old}}}(s,a)\right]$$

$$\text{subject to } \bar{D}_{\text{KL}}^{\rho_{\theta_{\text{old}}}}(\theta_{\text{old}},\theta)\leqslant\delta$$

- TRPO算法的第四个技巧，涉及KL散度的计算：

$$s\sim\rho_{\theta_{\text{old}}}\rightarrow s\sim\pi_{\theta_{\text{old}}}$$

- 最终的优化问题为：

$$\underset{\theta}{\text{maximize}}\ E_{s\sim\pi_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}A_{\theta_{\text{old}}}(s,a)\right]$$

$$\text{subject to } E_{s\sim\pi_{\theta_{\text{old}}}}\left[D_{KL}(\pi_{\theta_{\text{old}}}(\cdot|s)||\pi_\theta(\cdot|s))\right]\leqslant\delta$$

基于置信域策略优化的强化学习方法

- 为了实际编程，再次对TRPO的目标函数进行一阶逼近，约束条件进行二阶逼近（类似于泰勒展开），得到可实际操作的优化问题：

$$\underset{\theta}{\text{minimize}} \quad -\left[\nabla_\theta L_{\theta_{\text{old}}}(\theta)\big|_{\theta=\theta_{\text{old}}} \cdot (\theta-\theta_{\text{old}})\right]$$

$$\text{subject to} \quad \frac{1}{2}(\theta_{\text{old}}-\theta)^T A(\theta_{\text{old}})(\theta_{\text{old}}-\theta) \leqslant \delta$$

$$\text{其中} L_{\theta_{\text{old}}}(\theta)\big|_{\theta=\theta_{\text{old}}} = E_{s\sim\pi_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} A_{\theta_{\text{old}}}(s,a)\right]$$

- 最后，实际中是使用最优化方法中的共轭梯度法来进行计算的，主要是为了避免求逆矩阵的计算量。然后利用拉格朗日乘子将约束条件引入目标函数中。最后就形成了一般的优化方法，通过目标函数梯度方向，步长来优化策略模型 $\pi(a|s)$ 的参数，从而达到最小化目标函数的目的。注意，深度强化学习中，策略模型 $\pi(a|s)$ 就是神经网络。

基于置信域策略优化的强化学习方法

- **现在回过头看，为什么叫做信任域策略优化？**

再来看看优化问题：

$$\underset{\theta}{\text{maximize}}\ E_{s\sim\pi_{\theta_{\text{old}}},a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}A_{\theta_{\text{old}}}(s,a)\right]$$

$$\text{subject to}\ E_{s\sim\pi_{\theta_{\text{old}}}}\left[D_{KL}(\pi_{\theta_{\text{old}}}(\cdot|s)||\pi_\theta(\cdot|s))\right]\leqslant\delta$$

实际上，约束条件就是一个区域，在这个区域的限定范围内的优化步长，我们认为是可以信任的，不会出现普通策略梯度算法中，由于步长不合理而造成的越训练越偏离最优值的情况，而这是由我们做两种策略累积回报的差并使其为正来保证的，从而这种方法得名信任域策略优化。**总结：**TRPO避免由于样本采样不合理而导致的优化步长不合理，从而无法收敛的问题。通过约束不等式划定一个可信任域，在这个范围内的步长都是合理的，从而达到弥补采样不合理的目的，让优化更加容易收敛。这种TRPO算法基于两种分布的KL散度范围的限定。

作业

- **初阶作业（最高70分）：**

在参考资料中找到OpenAI的开源链接，搭建gym环境，运行一个仿真环境（如：CartPole-v1），加入随机动作，使其运行起来。

- **中阶作业（最高90分）：**

在参考资料中找到OpenAI的开源链接，搭建gym环境，再参考资料中mujoco的链接，运行一个基于mujoco的仿真环境（如：Ant-v2），加入随机动作，使其运行起来。参考资料中OpenAI baseline的链接，尝试如何将OpenAI baseline中的强化学习算法移植到仿真环境中并运行，运行成功可以达到90分。（注意mujoco不是免费的，申请试用30天）

- **高阶作业（最高100分）：**

在中阶作业的基础上，参考课程资料中《Learning agile and dynamic motor skills for legged robots》论文的链接，将ANYmal机器人的模型导入仿真环境，并移植OpenAI baseline的TRPO算法，使用默认神经网络使机器人强化学习的整个流程运行起来，运行成功得满分。