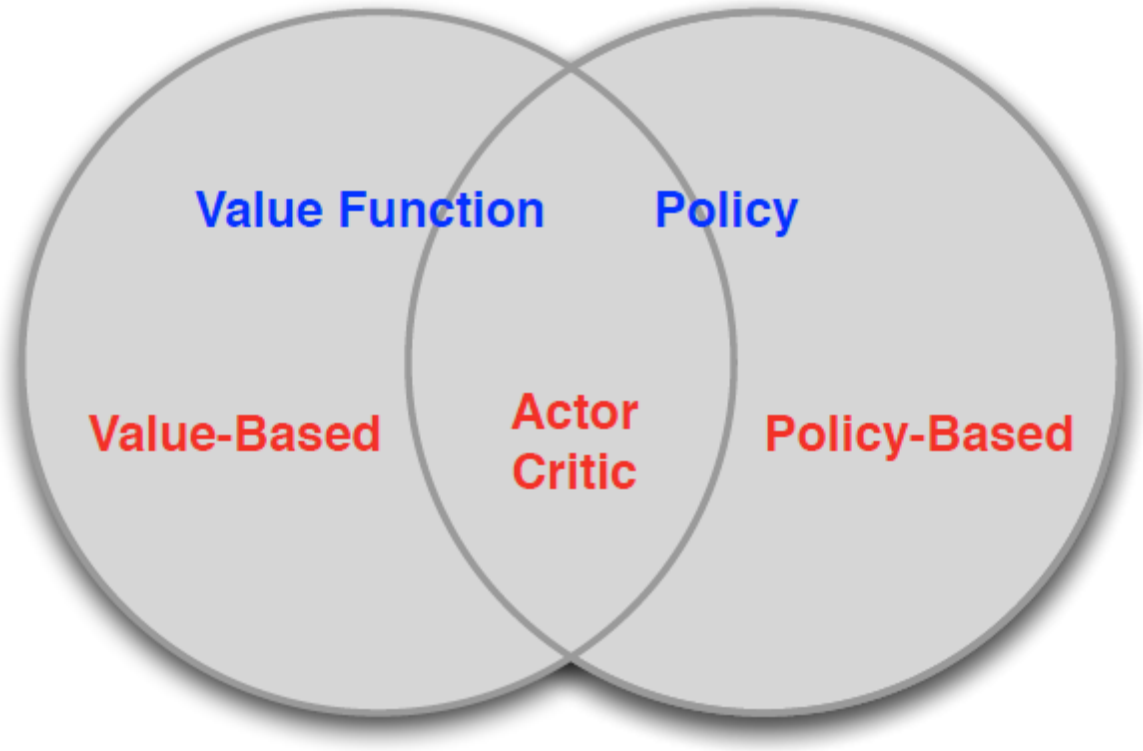
强化学习基础篇（二十九）策略梯度(一)

之前我们一直都是对价值函数或者动作值函数进行参数化近似：

$$V_\theta(s) \approx V^\pi(s)$$
$$Q_\theta(s,a) \approx Q^\pi(s,a)$$

其中策略是间接得通过值函数进行贪婪策略产生，但本文将介绍如何在model-free场景中对策略进行参数化：$\pi_\theta(s,a) = \mathbb{P}[a \mid s, \theta]$

1、model-free的强化学习算法



Model-free的强化学习算法可以分为这三类：

- Value Based强化学习算法

  这类方法是直接学习价值函数，然后隐似得通过贪婪方法得到策略
- Policy Based强化学习算法

  这类方法是不学习价值函数，而是直接对策略进行学习。
- Actor-Critic强化学习算法

  这类方法同时学习价值函数与策略。

2、Policy-based强化学习算法的优势

一般来说，Policy-based强化学习算法具有更好的收敛性，更加适用于高维的动作空间或者连续的动作空间的环境，并且可以学到随机策略。但是其劣势是通常只会收敛到局部最优解，而不是全局最优解。对策略的评估也比较低效并且有着较大的方差。

例如玩石头剪刀布的游戏：

如果是一个确定性策略，则很容易输掉游戏（例如你一直出剪刀，那对方一直出石头！）。一个均匀分布的随机策略（石头剪刀布随机）才是最优的（满足纳什均衡）。

随机策略和带随机性的策略是两个不一样的概念！带随机性的策略——例如$\epsilon - greedy$策略，是为了更好的探索！随机策略和探索无关，而是最后求出的策略就是一个随机的策略！

3、策略目标函数

Policy-based强化学习算法的目标就是在有θ参数化的策略$\pi_\theta(s,a)$下，找到最优的参数θ。这个最优的θ可以有几种方式进行评价其好坏：

- 在按幕的环境，可以直接评价初始状态的价值：

$$J_1(\theta) = V^{\pi_\theta}(s_1) = \mathbb{E}_{\pi_\theta}[v_1]$$

- 在无结束点的连续环境中，可以使用平均价值进行评价：

$$J_{avV}(\theta) = \sum_s d^{\pi_\theta}(s) V^{\pi_\theta}(s)$$

  连续环境状态（无开始状态）下，个体一直与环境进行交互，此时应该针对每个可能的状态计算从该时刻开始一直持续与环境交互下去能够得到的奖励，按该时刻各状态的概率分布求和。

- 或者使用每个时间步的平均奖励进行评价：

$$J_{avR}(\theta) = \sum_s d^{\pi_\theta}(s) \sum_a \pi_\theta(s,a) \mathcal{R}_s^a$$

  其中$d^{\pi_\theta}(s)$是基于当前策略下马尔科夫链关于状态的静态分布。在一个确定的时间步长里，求出每种状态下采取所有行为能够得到的即时奖励，然后按各状态的概率分布求和。

4、策略最优化

基于策略的强化学习算法是一个最优化的问题，我们需要找到一个θ，在改参数下对目标函数$J(\theta)$极大化，要达到这个目的，我们可以使用两种类型的算法：

一是不基于梯度的算法，比如，爬山算法 (Hill climbing)，单纯形法（Simplex），amoeba算法，Nelder-Mead方法，或者遗传算法 (Genetic algorithm)。

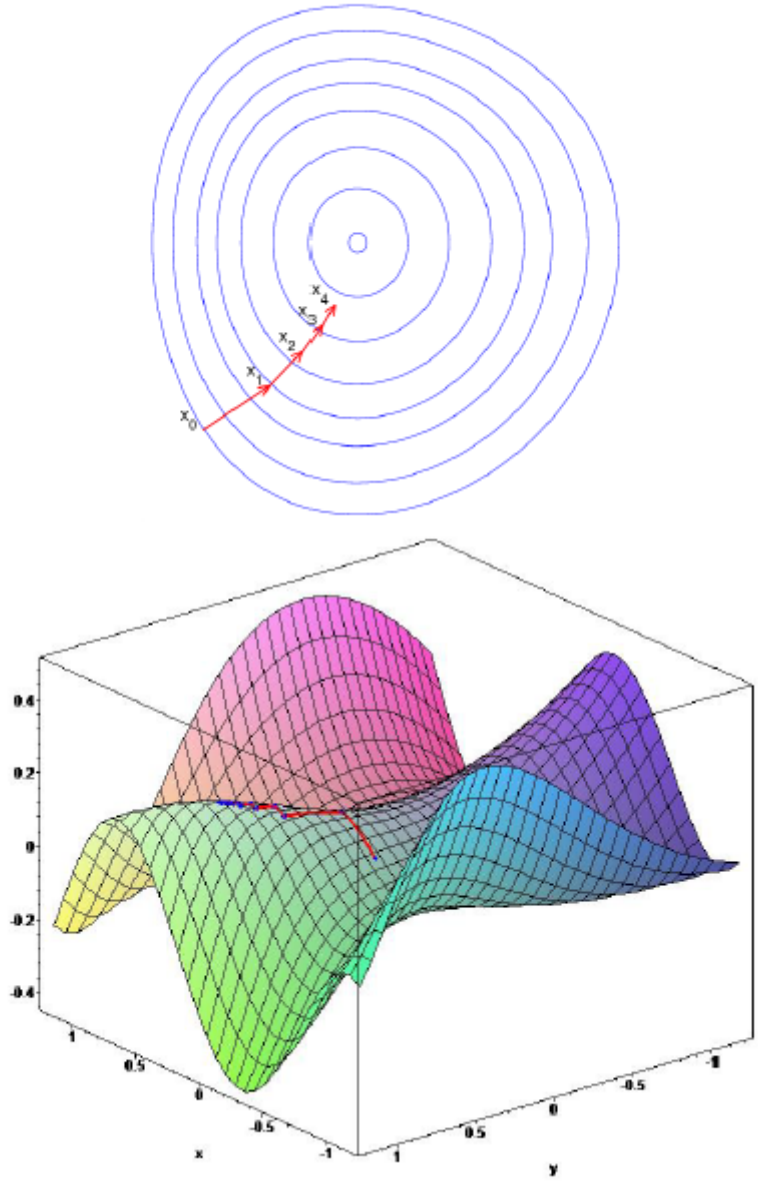
二是基于梯度的算法，比如，梯度下降算法(Gradient descent），共轭梯度法（Conjugate gradient method）以及拟牛顿法 (Quasi-Newton method)。

本节主要集中介绍梯度下降法。

策略梯度

当我们将策略目标函数$J(\theta)$作为优化目标，策略梯度算法是需要在参数空间θ中，通过梯度上升算法找到目标函数$J(\theta)$的局部极大值。

$$\Delta\theta = \alpha\nabla_\theta J(\theta)$$



α是更新步长的超参数，$\nabla_\theta J(\theta)$就是策略梯度：

$$\nabla_\theta J(\theta) = \begin{pmatrix} \frac{\partial J(\theta)}{\partial \theta_1} \\ \vdots \\ \frac{\partial J(\theta)}{\partial \theta_n} \end{pmatrix}$$

若用差分（Finite Difference）的方式进行$\pi_\theta(s,a)$的策略梯度的数值梯度法求解，则可以在每个参数的维度$k \in [1,n]$上对参数进行一个很小的扰动ϵ求偏导数，

$$\frac{\partial J(\theta)}{\partial \theta_k} \approx \frac{J\left(\theta + \epsilon u_k\right) - J(\theta)}{\epsilon}$$

其中u_k是单位向量，表示在第k个维度上值是1，在其他维度上值都是0。这种方法求梯度比较简单，噪声很大，很低效，但是也扛不住有时候效果还不错，特别是在策略没法微分的时候很有用。