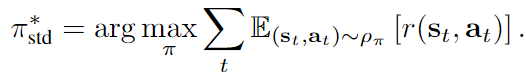
# 最大熵强化学习

《强化学习：原理与Python实现》读书笔记

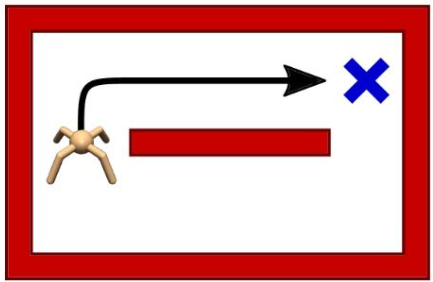
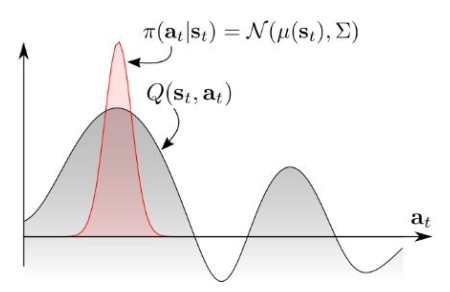
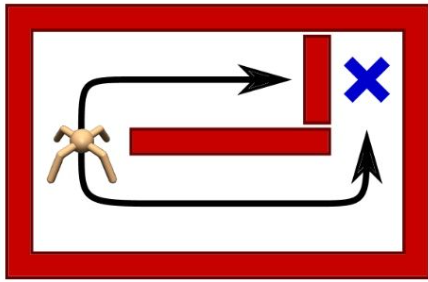
近年来，强化学习发展火热，最大熵强化学习类算法是强化学习的主流算法之一。本文的写作受启发于《强化学习：原理与Python实现》一书，这是一本不可多得的介绍强化学习的好书。特此记录一番。

## 问题引入

传统的强化学习的目标是最大化来自环境的奖励：

（1）

典型的代表就是Q-learning算法，它定义了一个Q(s,a)函数，表示在当前状态s下采取动作a所获得的累计奖励的期望值。而这样的目标设置存在一定的局限性。请看下图：

****

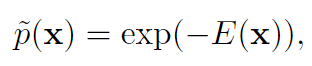
A1 B1 A2

A1图中，智能体可以通过上下两条路径到达目标点，向上的路径比向下的路径短，对应的奖励也稍高，这时候的Q函数可以用A2图的灰色曲线表示。传统的基于Q-learning的强化学习的策略比如DDPG等会选择Q值最大的action，再加上exploration noise构成一个高斯分布，如图A2的红色曲线部分。这样的策略是single modal，而原来的Q函数的分布是multimodal。传统Q-learning对应的策略无法表达这种multimodal，进而在测试中如果遇到B1图所示的障碍时候，智能体会一直选择向上的那条路径，无法到达目标点。

那如何解决上诉问题呢？一种策略是让智能体在选择动作时，不仅要考虑最优动作，还要考虑其他动作（0），这样智能体的探索能力和抗噪声能力都会得到提升。一种满足我们要求的方法是基于Energy-Based Model（EBM）制定策略。

## Energy-Based Model(EBM)

对于这样的假设，EBM表示为：

（2）

**E(x)**被称作是能量函数（energy function）。对所有的z，exp(z)都是正的，这保证了没有一个能量函数会使得某一个状态x的概率为0。基于EBM的策略正是利用这种特性来定义新的策略：

我们可以将能量函数定义为Q函数的负数：

（3）

其中为用于控制探索程度的参数，策略表示为：

（4）

具体的：

（5）

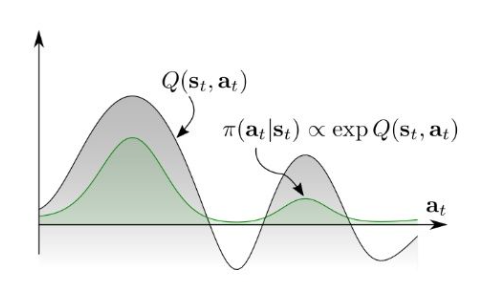
其中：

（6）

我们称作配分函数，它是仅与状态s有关的常数。

所以：

（7）

B2

图B2绿色部分展示了Energy-Based策略基于Q函数所形成的分布。可以直观地看出Energy-Based策略能够很好地反应出Q值的分布形态，表达出Q值的multimodal特性。这正是我们想要的结果。

还有个疑问，为什么在这里叫做soft呢？回想一下一般的RL每次都是选择Q值最大的动作作为策略，而此处的策略是

上式分母其实是LogSumExp的积分形式。如果非常大，同时由于exp()的存在，分母的值将主要取决于非常大时相对应的action，这时策略将以接近1的概率选择这个动作。而当各个动作的都相似或者没有特别突出大的值的时候，策略就将趋向于对各个动作赋予近似均匀的概率，每个动作都有被选择的机会。这就非常soft，即对当前局面的动作明确时，果断选择，而当动作不明确时候，随机探索。对比一般Q-Learning的Hard max策略，这里便是soft max。

接下来的过程中，对于控制探索程度的参数，若无特别声明，默认为1，即：

我们策略的优化方向从B2就能看出——策略尽可能逼近Q函数的分布,我们可以用KL散度来表示：

（8）

那这样的策略优化的目标是什么呢？接着这个式子推下去:

注意:Q函数是对未来累计奖励的期望,即：

所以：

（9）

推到这一步，我们发现，Energy-Based策略的基于KL散度的优化方向实际上就是在优化最大熵RL的目标——使得每个状态的奖励和该状态下动作的熵最大化。那什么是最大熵RL？最大熵RL起什么作用？

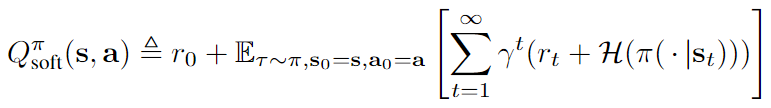
## 最大熵RL

我们定义最大熵RL的策略目标如下。

（10）

其中的是平衡探索程度的参数。传统的策略是最大化环境给予的奖励，最大熵RL的策略是在最大化环境给予奖励的基础上，还要最大化动作的熵。这样的策略实际上是一种stochastic policy, 即让策略随机化，输出的每一个action的概率尽可能分散，而不是集中在一个action上。可以看到最大熵RL的策略优化目标和Energy-Based策略是等价的。

根据最大熵RL的思想，将一个状态动作的熵作为奖励的一部分，我们可以定义最大熵Q函数：

（11）

有了策略的定义，策略的表示，策略的优化目标以及Q函数的定义，接下来我们给出策略改进定理并证明：

对于两个策略和，其中是依据优化得来，

如果，则，.

证：

对于我们每次都至少可以选择与相同的策略, 所以，

因此：

然后：

注意:基于最大熵RL的奖励包含有entropy项，所以：

=

得证