在介绍PPO算法之前，我们需要先引出两个概念介绍

(1）on-policy:要学习的agent跟与环境互动的agent是同一个agent时的策略；

(2)  off-policy：要学习的agent跟与环境互动的agent不是同一个agent时的策略。

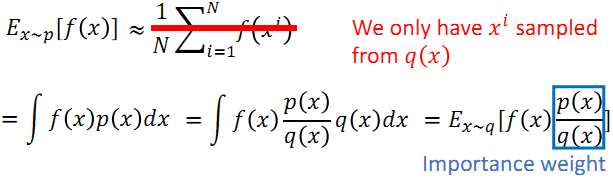
根据策略梯度方法，可以通过多次采样的方式进行强化学习。但是每学习一次，策略就发生了变化，根据梯度上升的公式：

如果我们使用Πθ来收集数据，那么参数θ被更新后，我们需要重新对训练数据进行采样，这样会造成巨大的时间消耗。

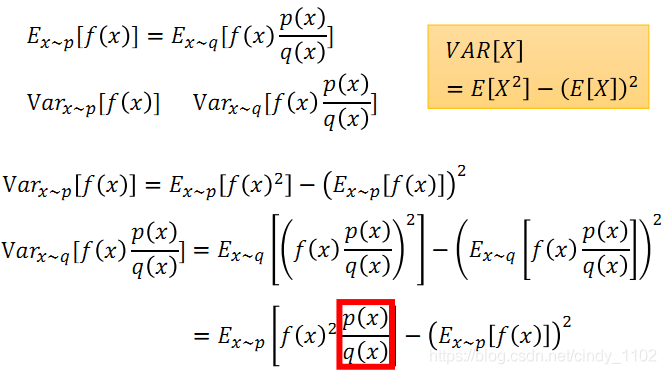
目标：利用Πθ′来进行采样，将采集的样本拿来训练θ，θ′是固定的，采集的样本可以被重复使用。

由此引出了重要性采样：

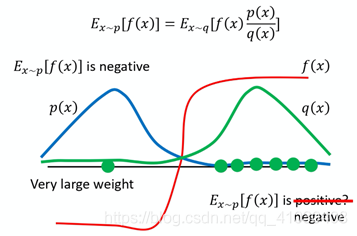
也就是说，x服从p分布，那么f(x)的期望等于f(x)p(x)/q(x)的期望，

p(x)/q(x)称为重要性权重

根据重要性采样，也就是说，可以利用其他分布采样得来的数据，来求另一种分布的期望。当然，这两种分布相差不能太远，越相似越好，因为除了考虑期望，两个分布的方差也要考虑：如果两个分布的方差差别太大，那么在采样的情况下由于数据不均匀不全面，计算得到的期望可能会有很大差距。



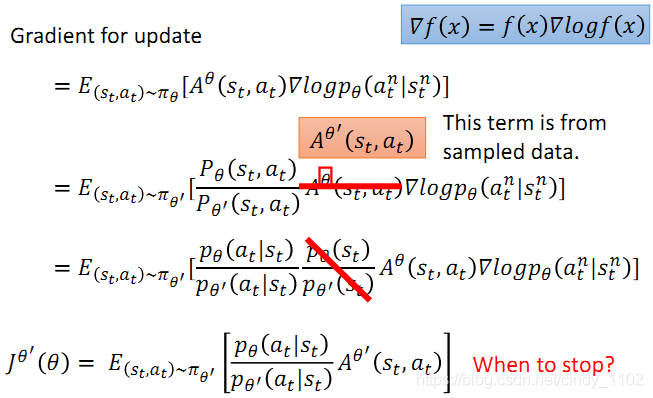
比如：



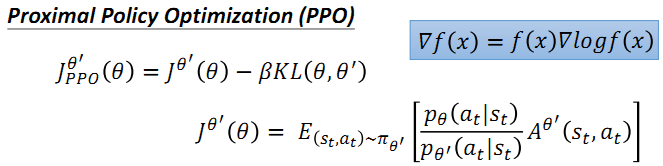
Advantage function (收益reward减去基准baseline) 也应该随着新的采样参数而变化

2.在不同的参数情况下，某一个状态state出现的概率几乎没有差别，因此可以将这一项近似地消掉

3.stop criteria取决于两个分布之间的差别大小



PPO在原目标函数的基础上添加了KL divergence 部分，用来表示两个分布之前的差别，差别越大则该值越大。那么施加在目标函数上的惩罚也就越大，因此要尽量使得两个分布之间的差距小，才能保证较大的目标函数。



PPO 算法

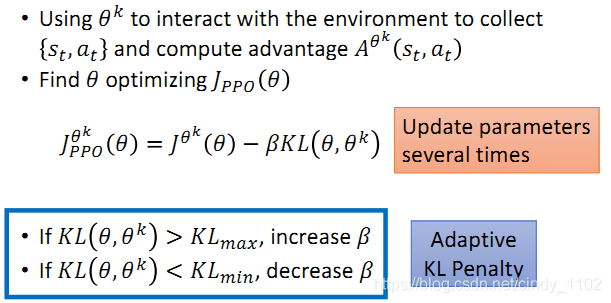
1.初始化policy的参数θ0，

2.在每一次迭代中,使用θ k来和环境互动，收集状态和行动并计算对应的advantage function，3.不断更新参数，找到目标函数最优值对应的参数 θ

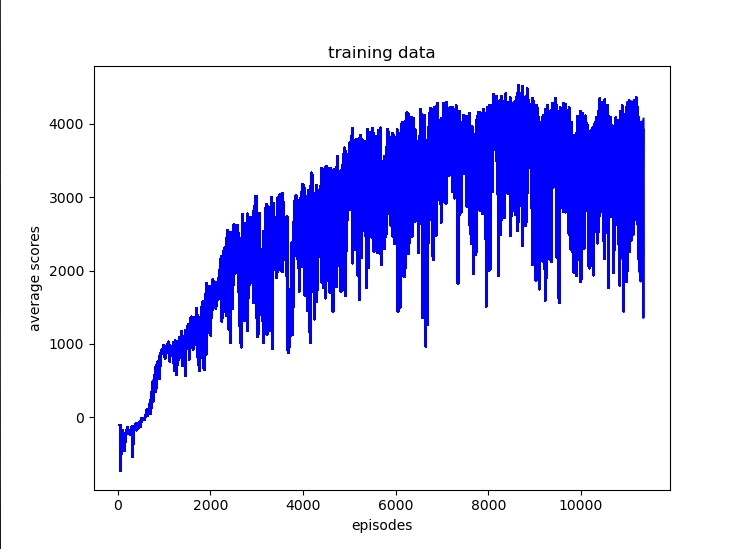
在训练的过程中采用适应性的KL惩罚因子：

当KL过大时，增大β值来加大惩罚力度

当KL过小时，减小β值来降低惩罚力度



结果展示



视频\*2