Value-based approach -》 学习一个critic

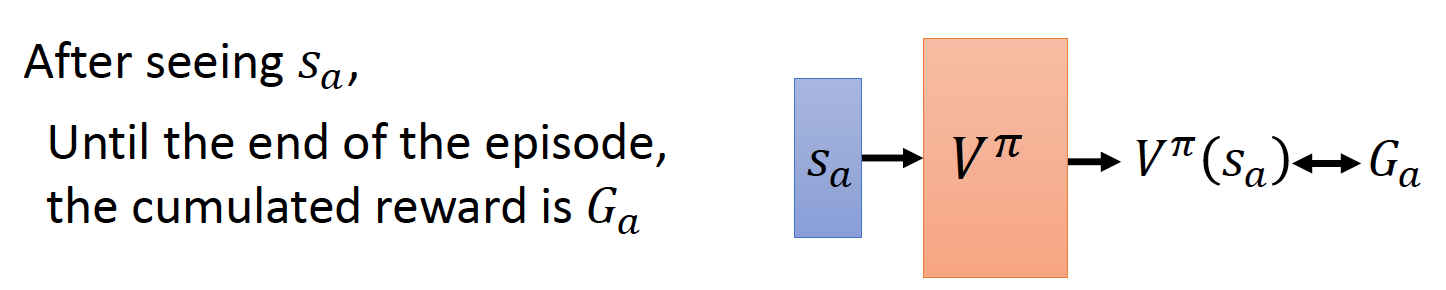
给定一个actor，评价这个actor的好坏，并不是直接决定选择哪个action。

1. State-value function：

当使用actor后，能够获得的期望奖励在访问状态s之后。

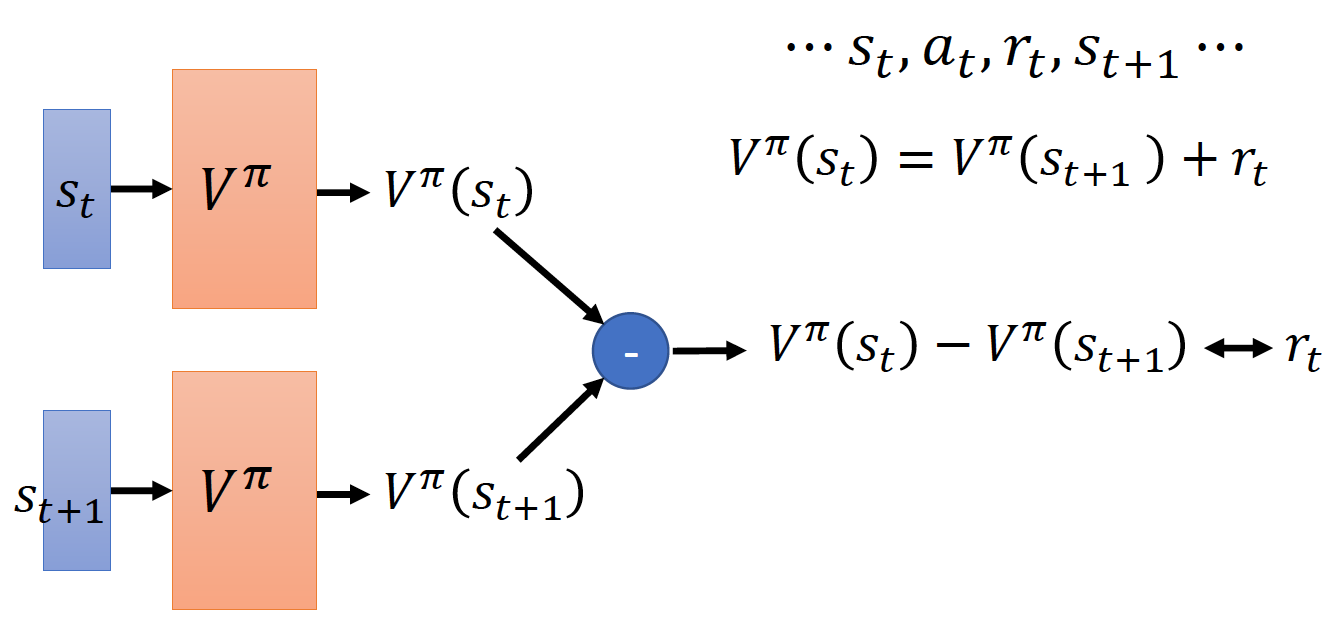
两种估计value的方法：

Monte-carlo(MC) based approach

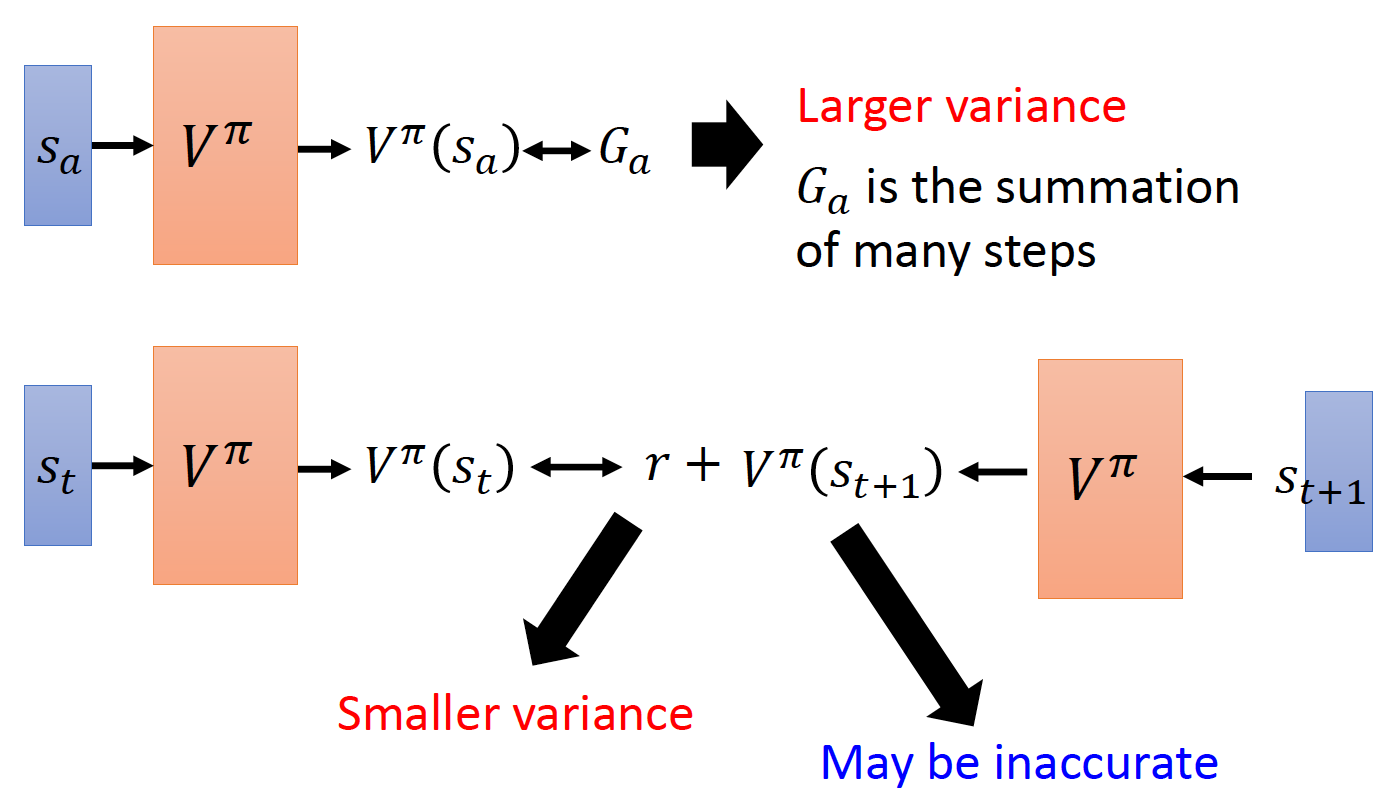


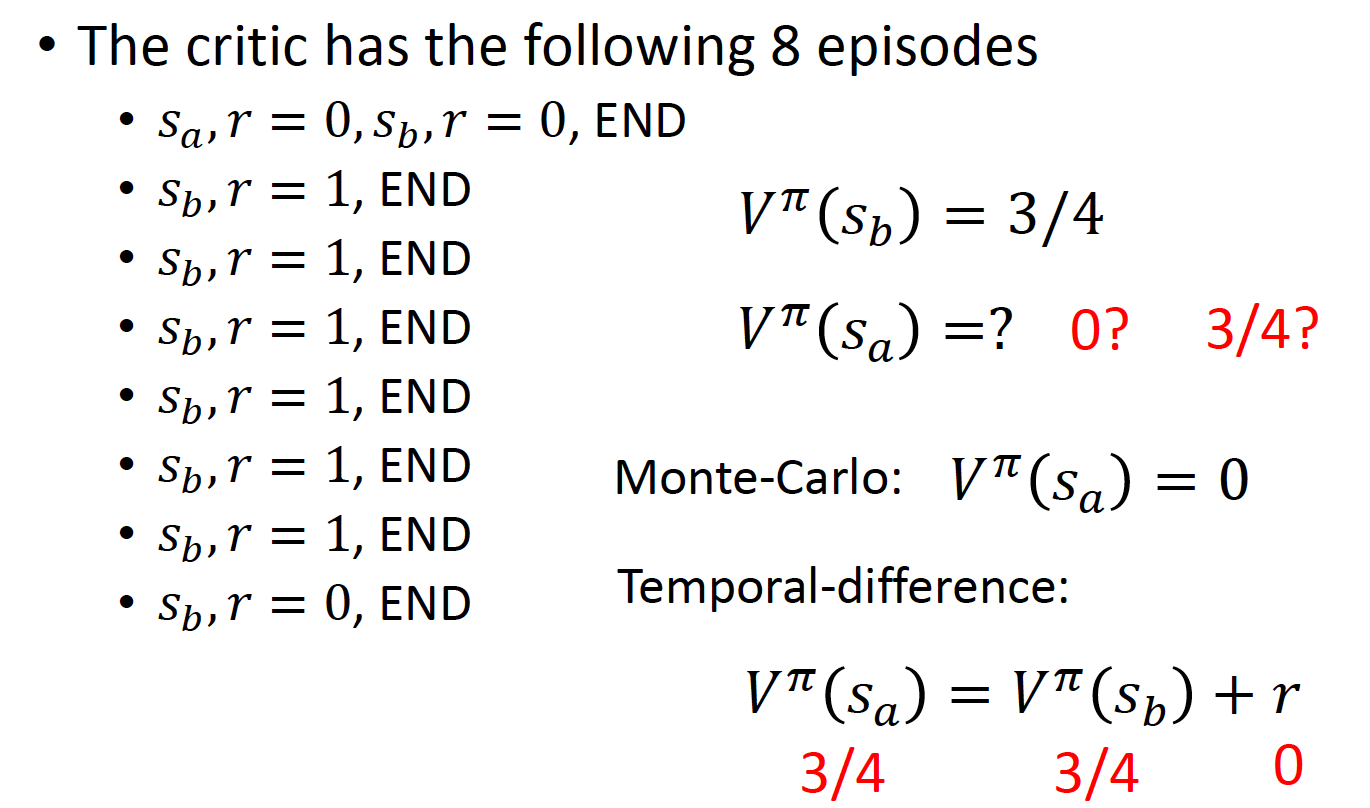
Temporal-difference(TD) approach

有些应用其完整的链路很长，延迟所有的学习直到一个dialogue完成太慢了。



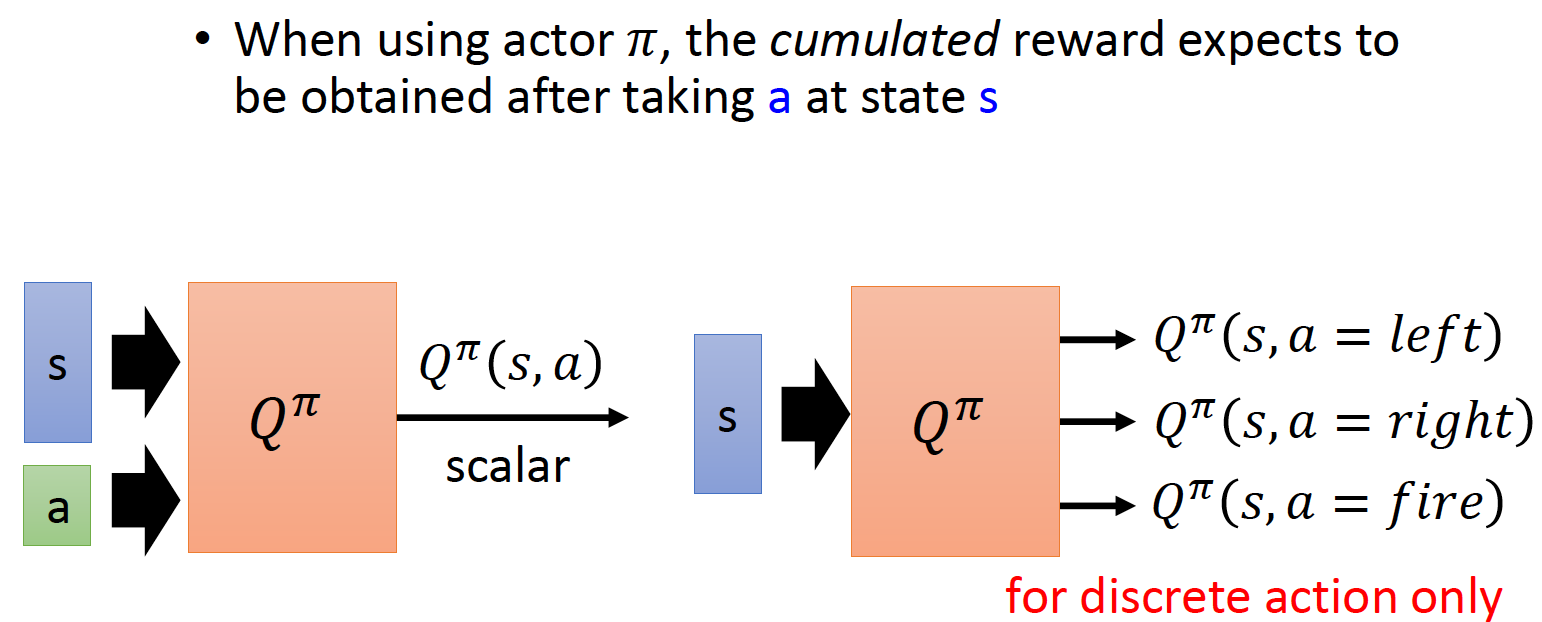
MC VS TD



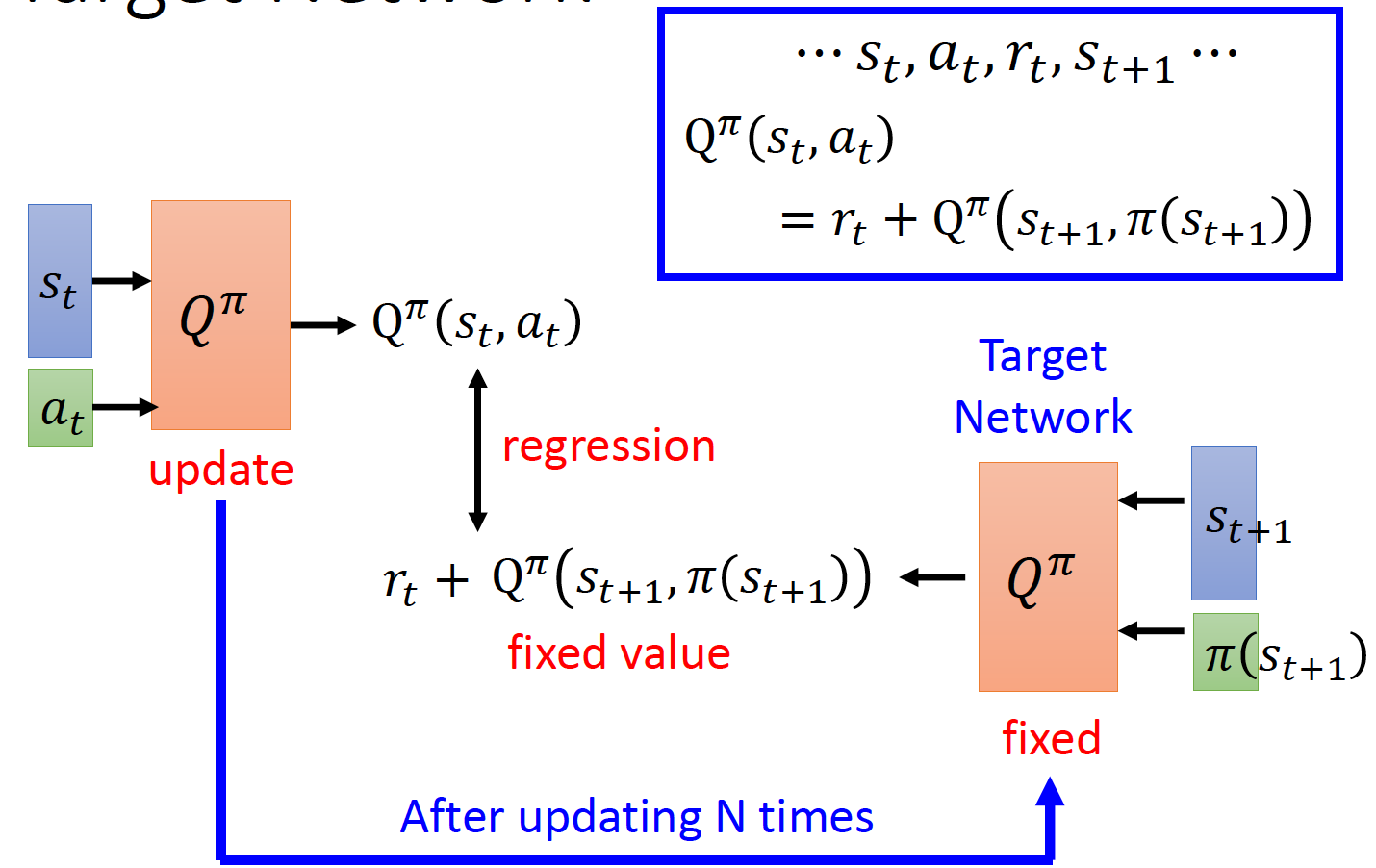


1. 另外一种形式：

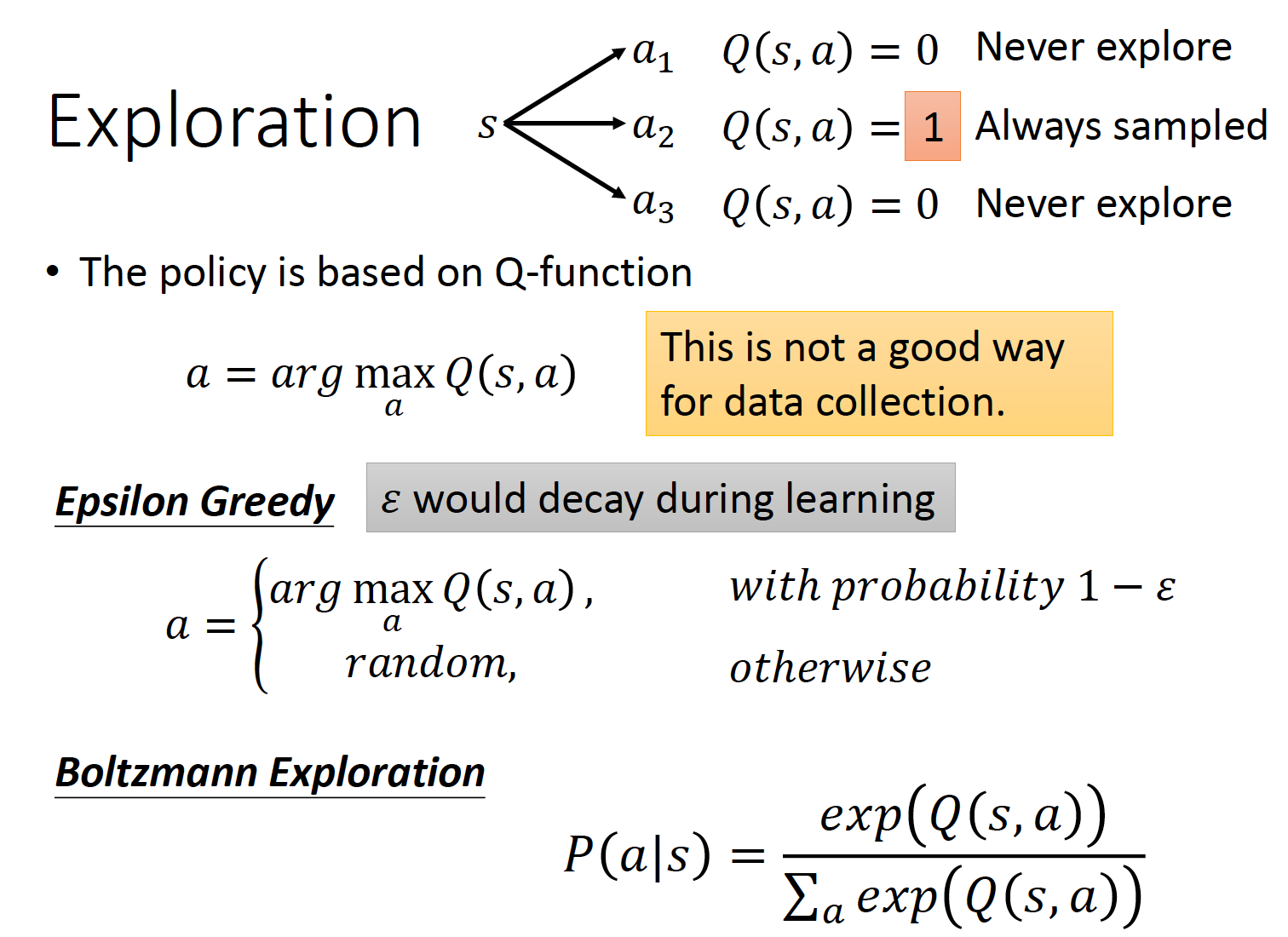
state-action value function



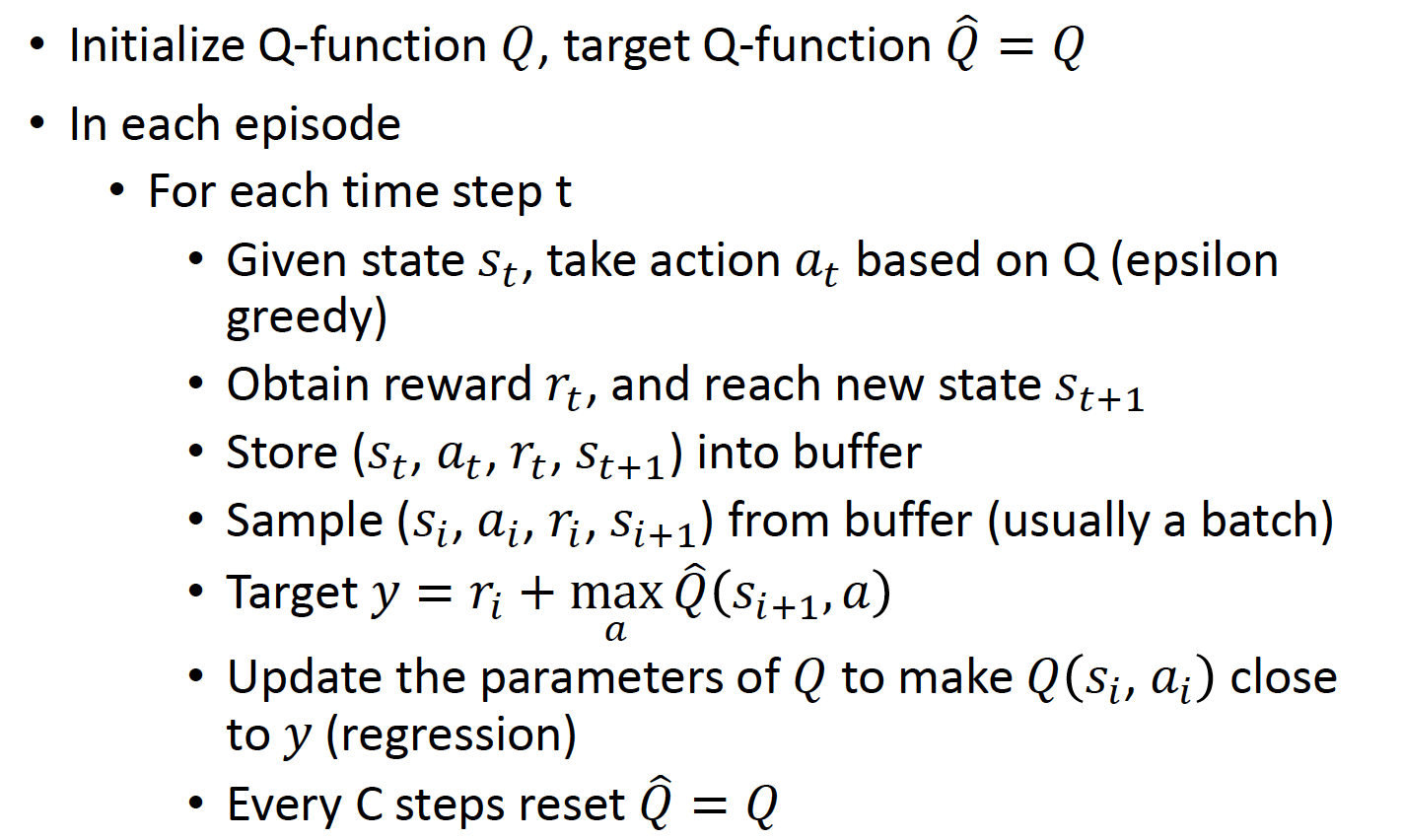
网络结构：



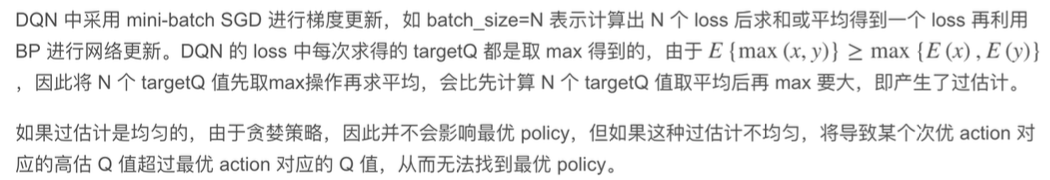
如何探索？

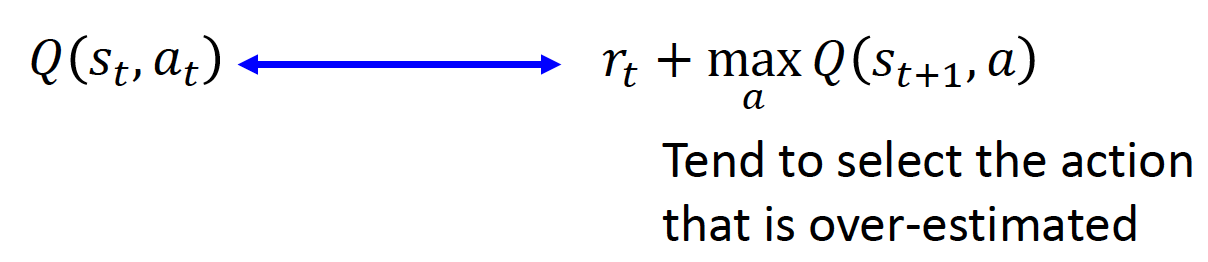


经典的Q-learning的学习算法：



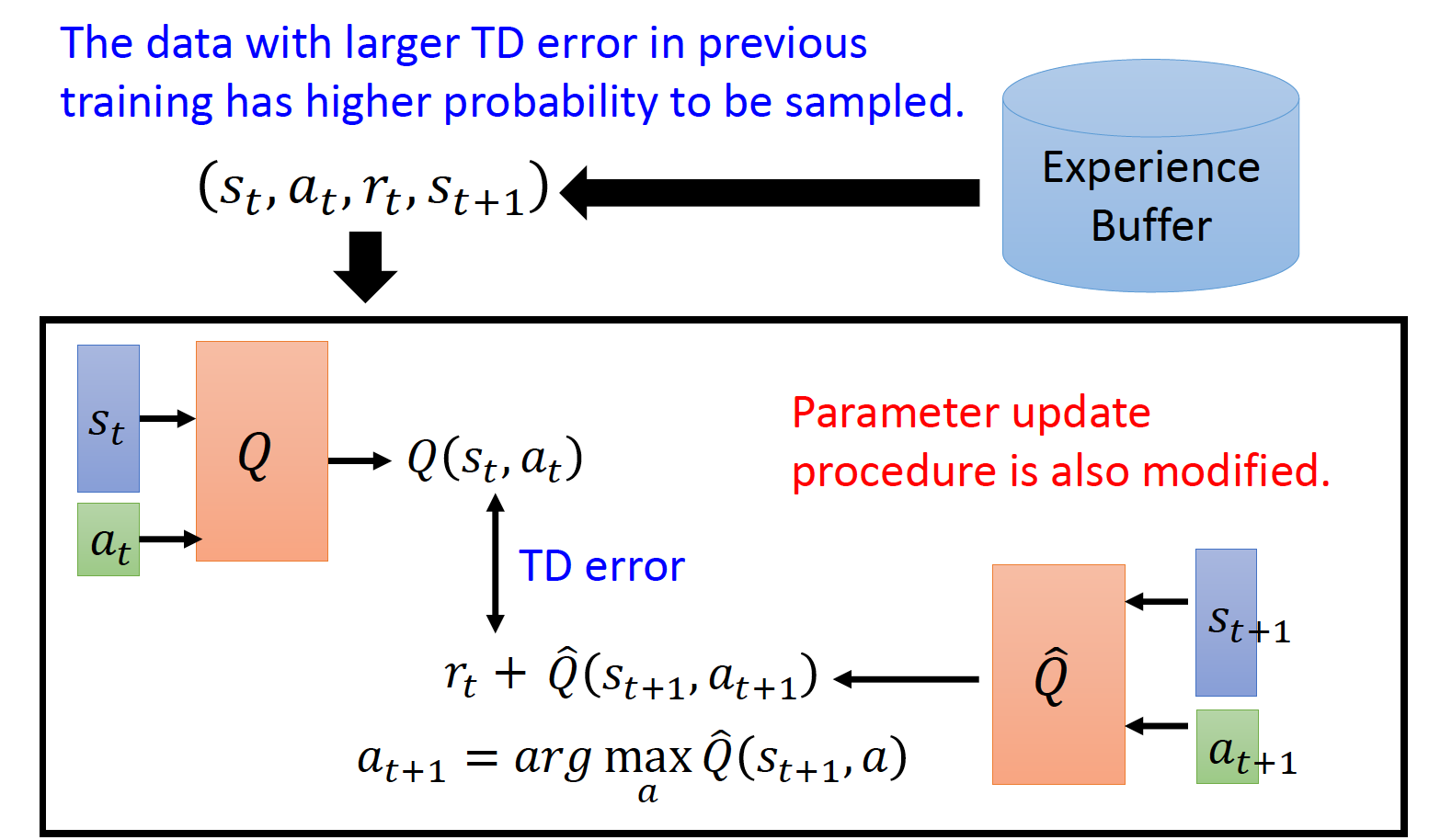
经典的Q-learning的问题：过估计Q值，即估计的值函数比真实值函数要打，且估计误差随action数增大而上升，过估计的根源在于Q-learning中target值的最大化操作。



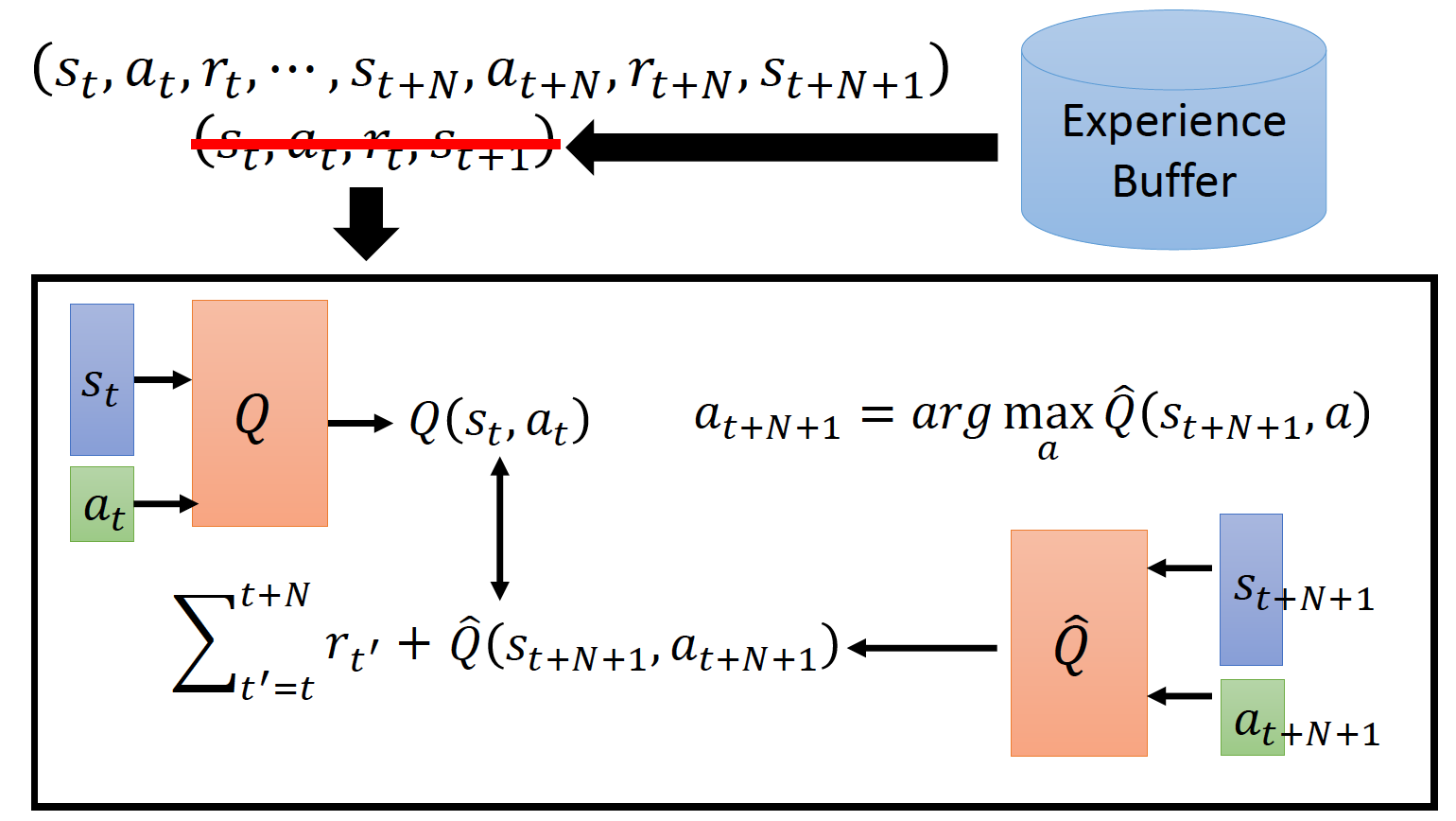


解决方法：将action的选择和评估解耦开，选择两个网络来做这两件事。

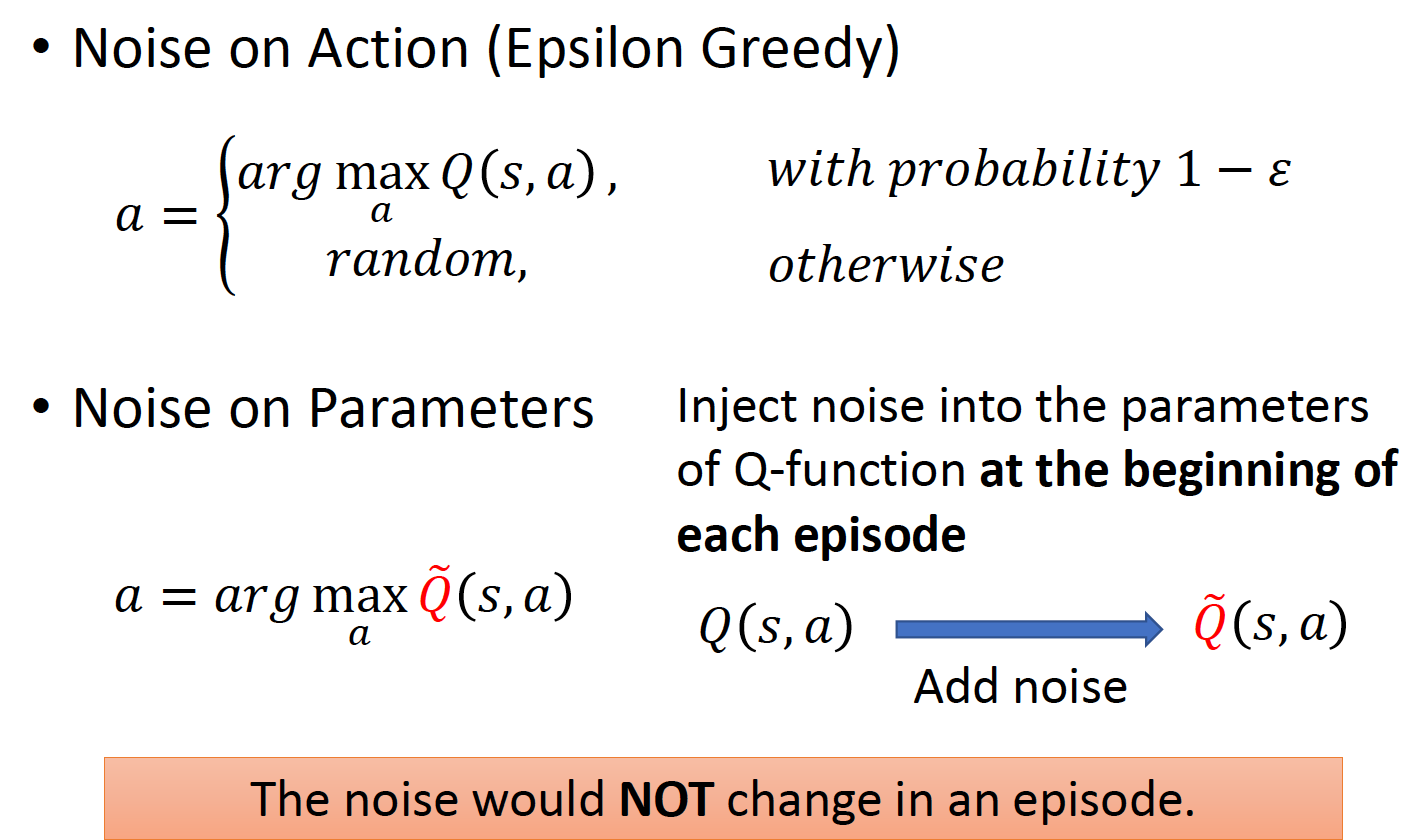
训练小技巧1：



训练技巧2：multi-step 蒙特卡洛和时间差分的平衡

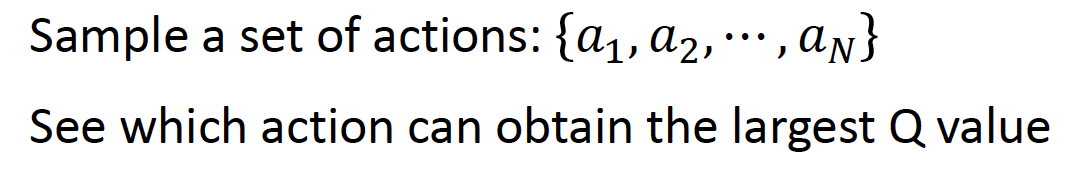


训练技巧3：给网络加点噪音



如何处理连续的action？也就是action a是一个连续的向量

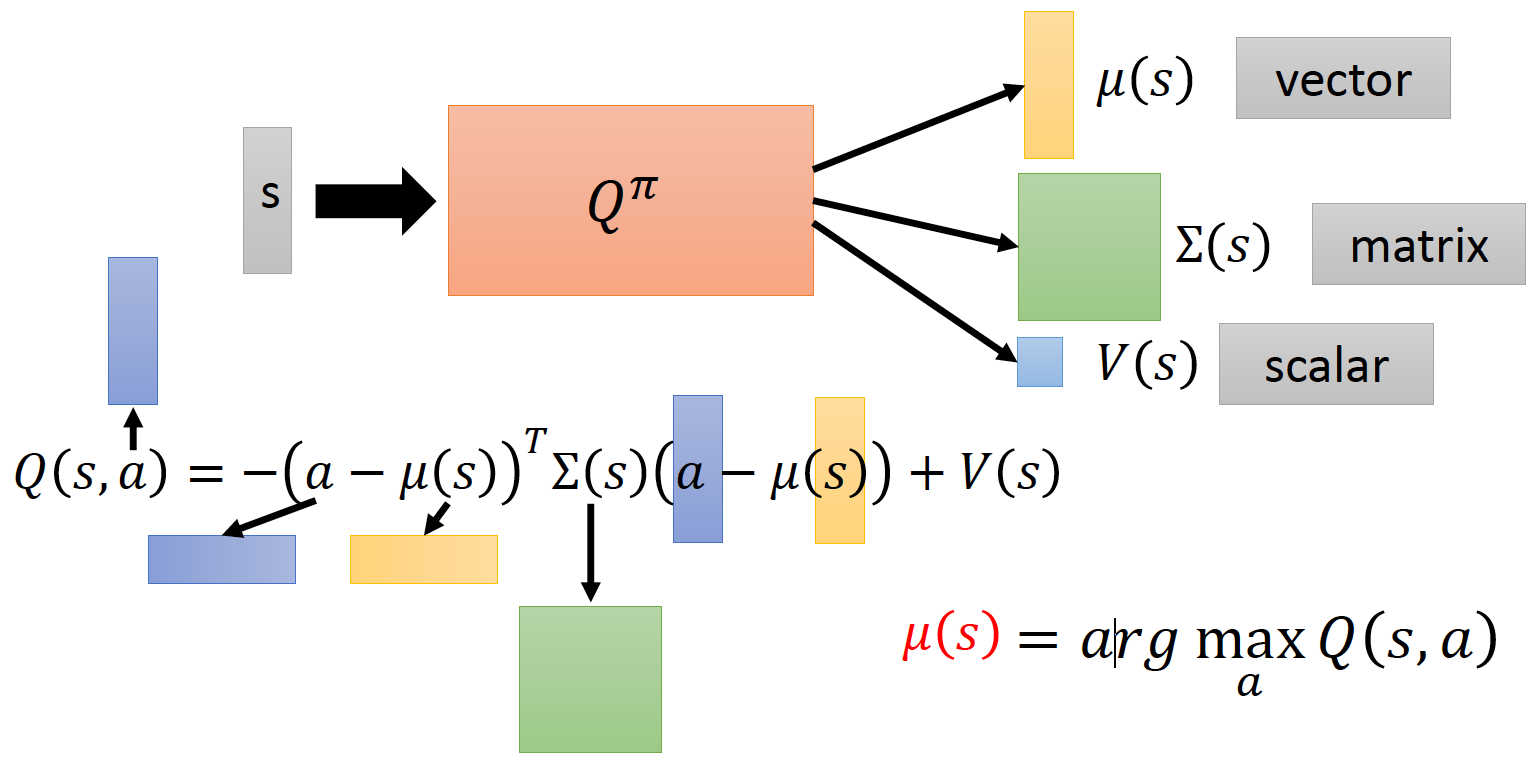
方法1：



方法2：

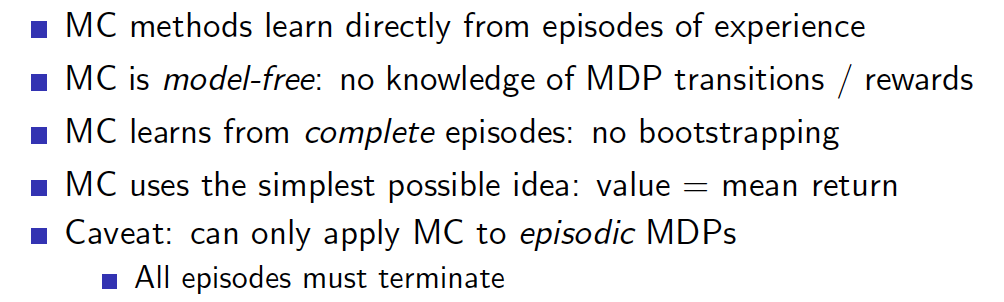
使用梯度下降来求解最优化问题

方法3：设计一个能使得优化容易的网络

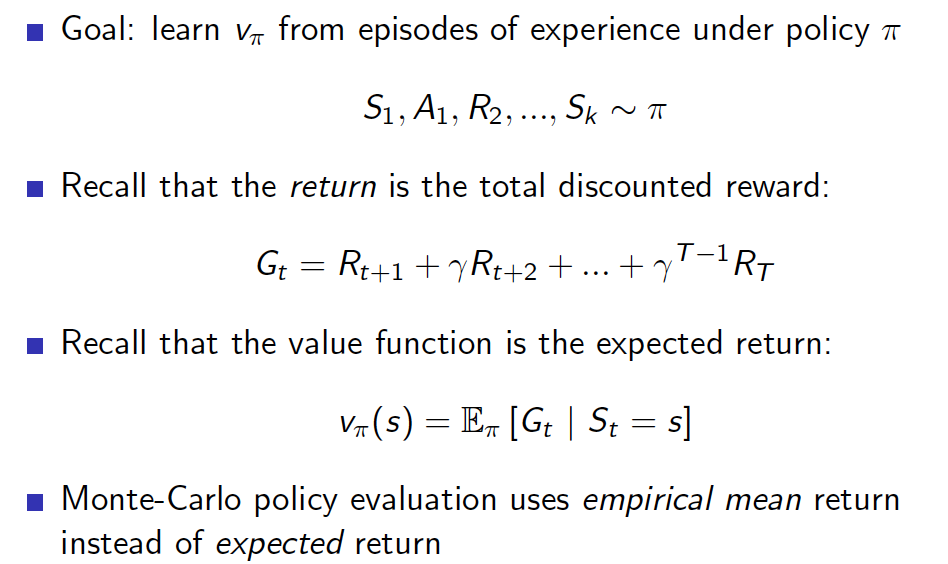


方法4：不适用Q-learning

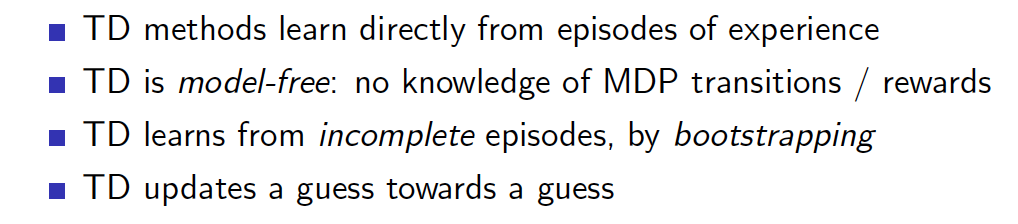
Monte-carlo算法的特点：



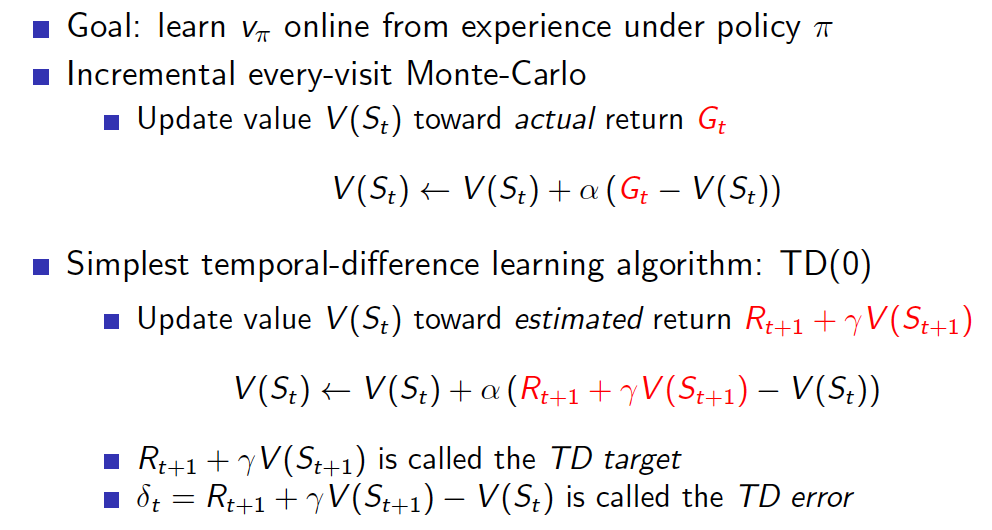
算法详解：

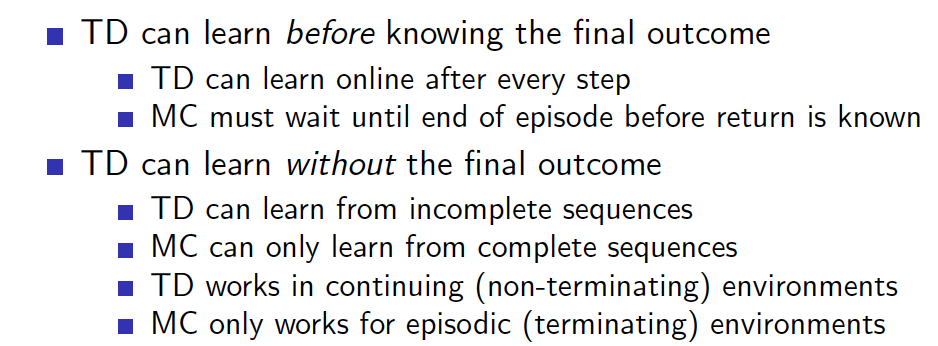


Temporal-difference learning：

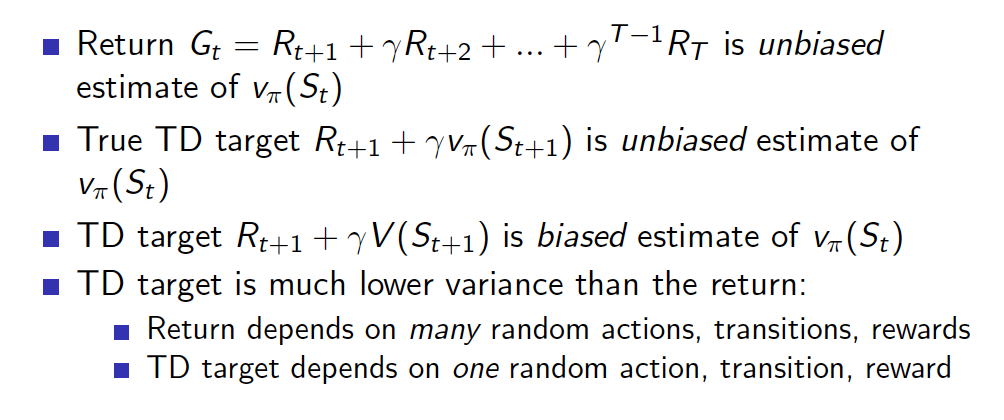
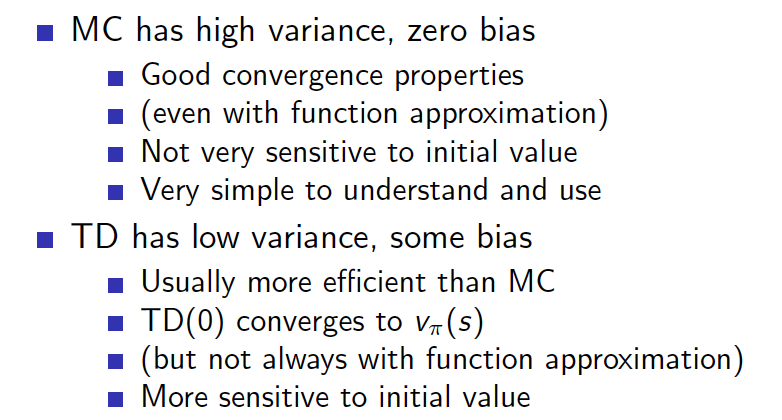


MC VS TD:

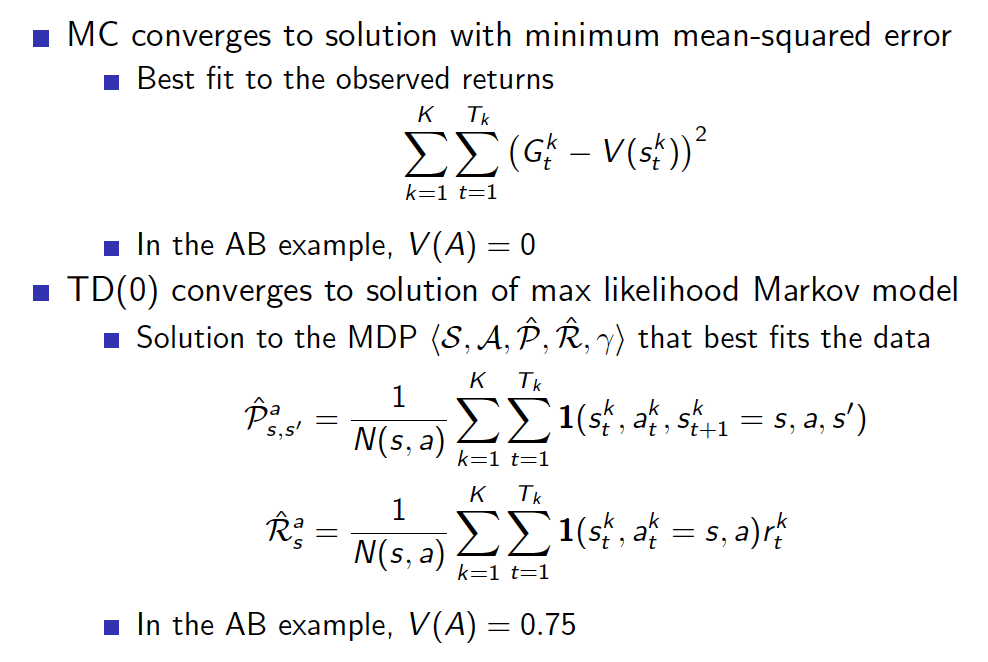




MC的方差大

实际例子：



TD exploits Markov property -》Usually more effcient in Markov environments

MC does not exploit Markov property -》Usually more effective in non-Markov environments

蒙特卡洛算法是什么？蒙特卡罗方法 不需要对环境的完整知识，仅仅需要经验就可以求解最优策略，这些经验可以在线获得或者根据某种模拟机制获得。经验其实就是训练样本，比如在初始状态s，遵循策略π，最终获得了总回报R，这就是一个样本。如果我们有许多这样的样本，就可以估计在状态s下，遵循策略π的期望回报，也就是状态值函数Vπ(s)了。蒙特卡罗方法就是依靠样本的平均回报来解决增强学习问题的。要注意的是，我们仅将蒙特卡罗方法定义在episode task上，所谓的episode task就是指不管采取哪种策略π，都会在有限时间内到达终止状态并获得回报的任务。比如玩棋类游戏，在有限步数以后总能达到输赢或者平局的结果并获得相应回报。

蒙特卡洛算法需要采用完成一个轨迹之后，才能进行值估计。

时序差分算法是什么？利用马尔科夫和动态规划的思想，效率更高的无模型学习。

马尔科夫的本质是通过多次采样后计算平均作为期望累计奖赏的近似，其在一个轨迹采用完成后才对各个状态-动作对进行更新，属于批处理方式。