强化学习的套路之一是计算每个状态对应的价值，然后状态价值最大化寻找最优策略。

前提，环境建模即状态转移图的建立，对应状态转移矩阵P。

有限马尔科夫过程：注意是有限马尔科夫过程，本方法只针对有限马尔科夫过程，即有限次后该过程肯定会结束，也是对强化学习求解入门的理解。

马尔科夫决策过程通过状态价值函数和状态动作价值函数，通过贝尔曼方程描述状态之间及动作之间的递推关系，采用倒推法/备份图来倒推整个有限过程的最优路径。我们的目的是找到一条最优路径，时刻线，状态-动作-状态-动作-状态。

通过马尔科夫过程我们可以得到贝尔曼方程，及状态价值函数和状态动作价值函数递推关系，并采用倒推法来进行求解。

在本章中我们将介绍有限马尔科夫过程。这个过程包括评估反馈和在不同的状态下选择不同的动作。MDPs是经典的序列化决策问题，动作不仅影响当前的奖励，也会对后续的状态和奖励产生影响。MDPs包括延时奖励，并且需要在当前和延时奖励之间这种。

MDPs是强化学习问题的理想化数学形式，产生精确的理论化表述。

代理-环境界面

MDPs旨在直接构造交互学习问题来实现目标。学习者和决策者成为agent代理。代理与之交互的，包括代理的外部环境的称作environment环境。代理选择动作并且环境对这些动作做出反应并反馈给代理新的状态。环境对动作给予奖励并且代理努力通过一系列的动作来实现平均奖励最大化。



更具体的来说，代理和环境在一系列离散的时间序列上交互。t=0,1,2…。在每个时间t，代理从环境中接收状态St并在此基础上选择动作At。下一个时间，代理将会受到一个数值奖励Rt+1和一个新的状态St+1，作为动作的结果。MDP和代理产生如下序列

S0，A0，R1，S1，A1，R2，S2，A2，R3，S3…

在有限序列MDP过程中，状态，动作，奖励集合（S,A,R）都有有限元素。在该状况下，随机变量Rt和St可以通过先前的状态和动作很好的被定义概率分布。也就是说，给定先前的状态和动作，下一时刻t的状态和奖赏的的概率可以被定义为



函数p定义的MDP的动态特性dynamics。在等号上方的点表示只是一个定义。动态函数p是一个具有四个自变量的通用确定性函数。满足



在马尔科夫决策过程中，假定当前时刻的状态St和奖赏Rt仅和前一状态的状态St-1和动作At-1有关.。状态转移概率函数







实例。给定马尔科夫决策过程的状态转移图，列出s,a,s’、p和r(s,a,s’)。

奖励和目标机制：代理的目标最大化收益总和，即可能不是最大化当前收益，而是最大化长时间累计收益。根据收益制定目标看似比较局限，但是实际证明了这种方法具有广泛适应性和灵活性。通过收益最大化，代理可以实现我们希望它达到的目标。另外，奖励信号应告诉代理我们需要实现什么，而不是告诉代理如何去实现它的先验知识。

预期收益表达式：





价值函数和贝尔曼方程

马尔科夫过程通过构造贝尔曼方程，通过借这个方程组来求解。

马尔科夫过程

主要要点：贝尔曼方程，状态价值函数和动作价值函数定义及之间关系。

动态规划求解：分两步，预测和控制。主要依靠状态价值函数进行预测和控制。

关键点有两个，一是问题的最优解有若干个小问题的最优解构成，即通过寻找子问题的最优解来寻找问题的最优解；二是可以找到子问题状态之间的递推关系，通过较小的子问题状态递推出较大的子问题的状态。

策略评估：

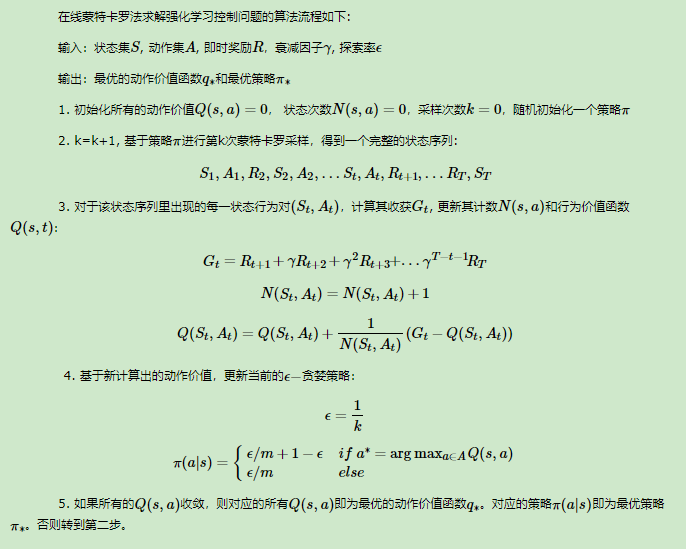
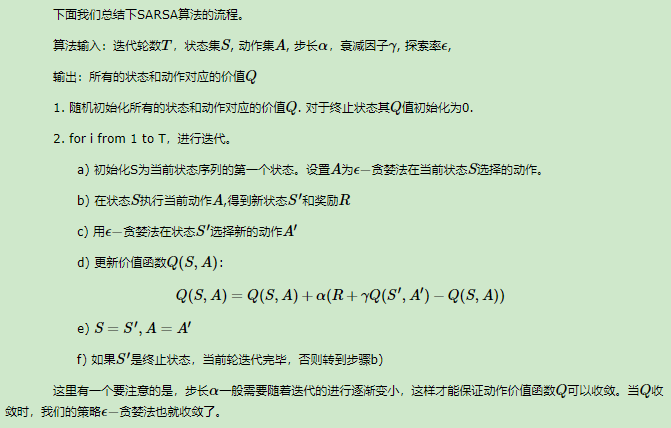
蒙特卡罗方法：基于采样序列的方法，通过采样若干完整的状态序列来估计状态的真实价值，即不基于模型的强化学习方法，前面的马尔科夫决策过程和动态规划方法都需要知晓状态转移概率模型。所谓的经历完整性，就是这个序列必须是达到终点的。有了很多组这样的经历完整性的状态序列，我们就可以近似的估计状态价值，进而求解预测和控制问题了。

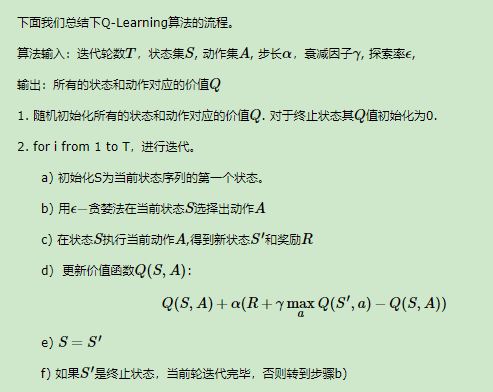
特点，不需要依赖于状态转移概率模型，二是从经历过的完整序列学习，完整经历越多，学习效果越好。

第一个问题是如何从完整序列中学习到每个状态的价值。

从状态价值函数中的定义可以发现，





TD：与MC的区别在于状态价值的计算方式不同。控制策略在线策略与MC采用贪婪方法一致。离线控制策略只不过是只更新价值函数。









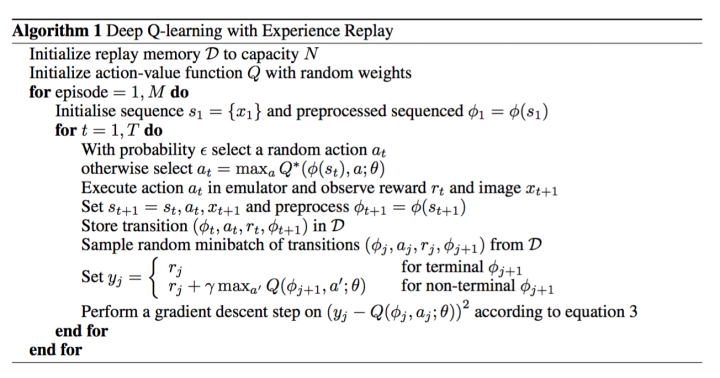


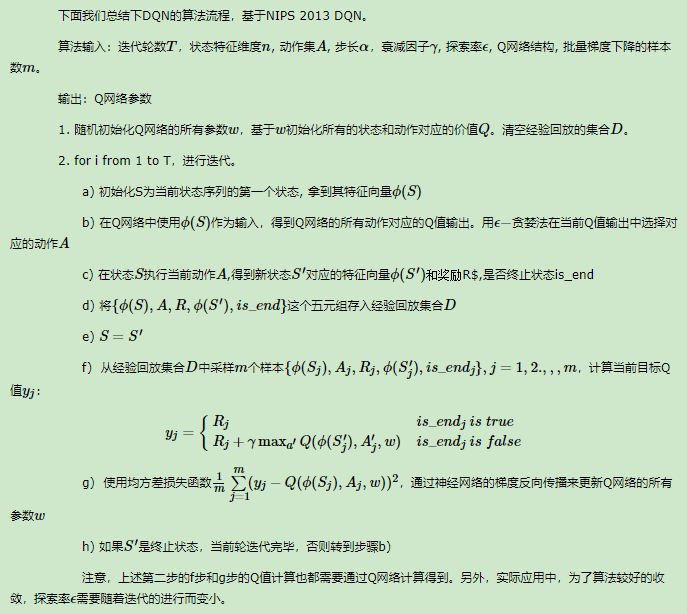






DQN深度Q学习，价值近似函数。经验放回。



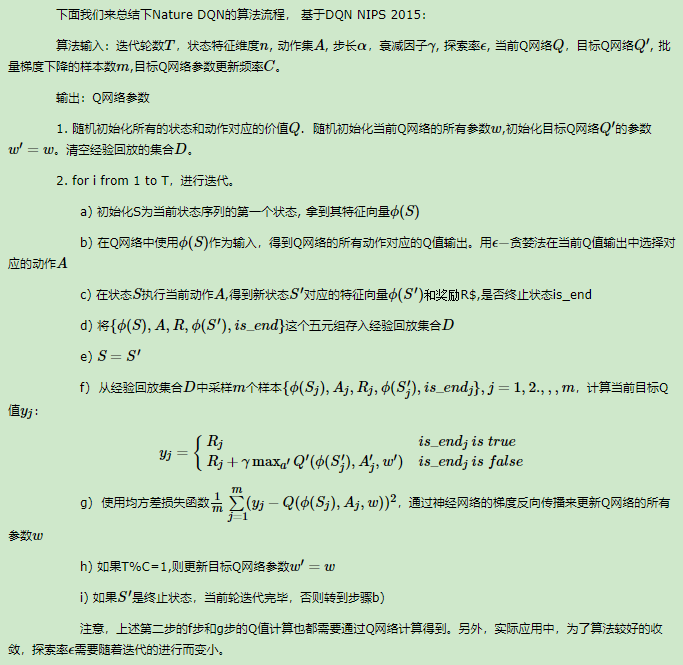


1、任务1，理解DQN算法原理及实现过程，预学习神经网络及其实现，

2、Q-learning编程实现功率分配。

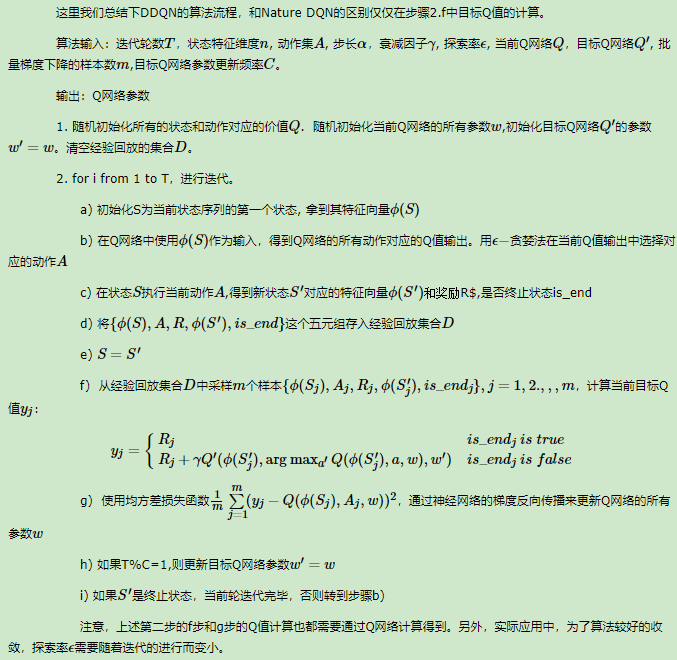
DQN的计算过程中，样本数据的产生中的y值仍然依赖于当前Q网络，相关性较强，所以下面的改进算法需打破这种相关性。

Nature-DQN的思想是采用两个Q网络，一个用来计算Q‘和y值，目标Q网络延迟更新当前Q网络。与DQN不同之处在于f步用的是目标Q网络即Q‘来计算y值，g步中Q用的是经验放回池中的样本输入当前Q网络即Q来得到的Q值，这样做的目的是为了减弱训练的当前网络Q与样本中数据y的相关性，注意到DQN中样本数据的y是依赖于当前网络Q计算得到的，也就是用于训练的样本数据与待训练的网络相关。

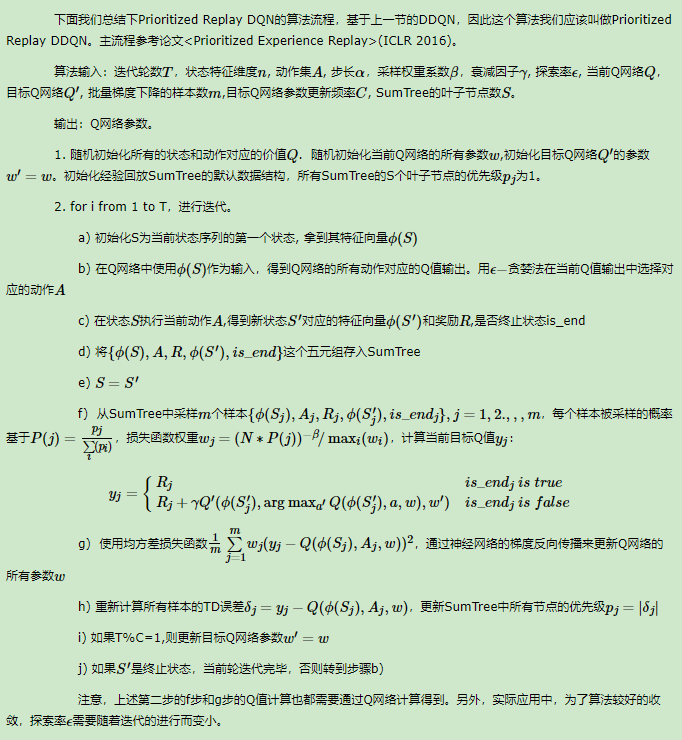


注意到上述y值计算中用到的是输入状态S‘进入Q‘网络后选出的最大输出值，这样做好不好呢？思考一下，毕竟Q‘网络有延迟更新效应，选择S’下获得最大Q值的动作，应该用当前Q网络即Q，选择S’下获得最大Q值的动作a，然后选择在Q‘下状态S’和a对应的Q‘值，这样既能减小y值与Q网络相关性，又能使得选择的Q‘更准确。

DDQN的做法正是基于上面的想法



这样做还有没有缺点呢？答案是肯定有的，从经验放回池中采样方法是随机采样，难道样本之间是没有优先级的吗？肯定有，那下面prioritized Replay DQN就来解决这个问题，定义经验放回池中样本被采样的概率。



总结：

1、强化学习描述的问题/模型

强化学习描述的问题可以归为决策问题。

来看强化学习的模型/学习过程



决策者从环境中获取状态，采取相应的动作，在下一时刻获得奖励（延迟奖励），我们的目标是追求累积奖励最大化，而非延迟奖励最大化。

产生的序列为：

与监督学习、无监督学习的区别？与监督学习最大的区别在于没有预先准备好的数据进行训练，强化学习有延迟奖励，是通过无数次的走一步看一步进行自我经验积累。

模型要素：

环境的状态转移模型：可以用状态转移概率来描述，即在状态s下执行动作a转移到下一个动作的状态

个体的策略：个体依据策略来选择动作，最常见的策略表达式为条件分布，即在状态s下执行动作a的概率

个体在状态s下采取策略后的价值：，用在当前状态s下采取策略后的平均加权累积延时奖励来表示。虽然个体状态s下采取策略会得到一个延时奖励，但是仅仅考虑这个延时奖励是不够的，因为当前延时奖励高，并不代表后续奖励也高。比如下象棋的时候，某个动作可以吃掉对方的车，这个延时奖励高，但是接着我们会输掉棋局，此时在当前状态采取吃车的动作奖励值并不是很高，所以我们要综合当前状态下采取策略后的所有延时奖励。中表示折扣因子，取值在[0,1]之间。如果为0，则为贪婪法，即价值仅由当前延时奖励决定，如果是1，则所有的后续奖励与当前奖励权重一致。大多时候，我们会取(0,1)之间的数值，即当前奖励比后续奖励的权重值要大。

实例：西瓜浇水的例子。

2、有限马尔科夫过程与贝尔曼方程

假设环境状态转移概率与策略仅仅和当前状态有关，与之前状态无关。

即，

所以，价值函数也仅仅依赖于当前状态，其中

