**什么是 Policy Gradients**

其实在引言部分我们已经介绍了策略梯度的基本思想，就是直接根据状态输出动作或者动作的概率。那么怎么输出呢，最简单的就是使用神经网络啦！  
我们使用神经网络输入当前的状态，网络就可以输出我们在这个状态下采取每个动作的概率，那么网络应该如何训练来实现最终的收敛呢？  
我们之前在训练神经网络时，使用最多的方法就是反向传播算法，我们需要一个误差函数，通过梯度下降来使我们的损失最小。但对于强化学习来说，我们不知道动作的正确与否，只能通过奖励值来判断这个动作的相对好坏。基于上面的想法，我们有个非常简单的想法：

**如果一个动作得到的reward多，那么我们就使其出现的概率增加，如果一个动作得到的reward少，我们就使其出现的概率减小。**

根据这个思想，我们构造如下的损失函数：**loss= -log(prob)\*vt，**式中log(prob)表示在状态 s 对所选动作 a 的吃惊度, 如果概率越小, 反向的log(prob) 反而越大. 而vt代表的是当前状态s下采取动作a所能得到的奖励，这是当前的奖励和未来奖励的贴现值的求和。也就是说，我们的策略梯度算法必须要完成一个完整的eposide才可以进行参数更新，而不是像值方法那样，每一个(s,a,r,s')都可以进行参数更新。如果在prob很小的情况下, 得到了一个大的Reward, 也就是大的vt, 那么-log(prob)\*vt就更大, 表示更吃惊, (我选了一个不常选的动作, 却发现原来它能得到了一个好的 reward, 那我就得对我这次的参数进行一个大幅修改)。

这就是 -log(prob)\*vt的物理意义啦.Policy Gradient的核心思想是更新参数时有两个考虑：如果这个回合选择某一动作，下一回合选择该动作的概率大一些，然后再看奖惩值，如果奖惩是正的，那么会放大这个动作的概率，如果奖惩是负的，就会减小该动作的概率。

策略梯度的过程如下图所示：

