主要讲了将policy看成某个参数hetaheta的函数，即将policy形式变成状态和动作的概率分布函数，在policy函数可微的情况下能够通过对参数求导来优化policy。

在第一课的内容中，曾经提到过agent的分类，将value function、policy、model(environment)进行组合可以得到model-based、policy-based、model-free、value-based、actor critic五种类型。其中value-based是说已知policy情况下学习value function，就像第四课的内容。policy-based是指没有显式的值函数形式，而需要学习policy。actor critic则是需要同时学习学习值函数和policy的形式。

带有随机性的policy往往比确定性的policy效果要好，就如第一课中所说的确定性的policy通常会被过利用。Policy-based类型能学习到带有随机性的policy，并且对于高维的action更有效。后者是因为在用value-based的方法时，常常要最大化当前的值函数，会需要大量的计算。

在之前的课程中，policy都是基于greedy或者ϵ−ϵ−greedy的方法直接从值函数中获得。本课主要内容是为policy引入参数，变成在某个状态和某组参数下选择某个动作的概率分布πθ(s,a)=P[a|s,θ]，直接求解策略，从已有的sample的experience中学习如何让policy变得更好。为什么要求解含参数的policy也是跟上节课一样的原因，现实问题中有大量的state和action，无法针对每个state每个action都有一个确定的policy，因此需要一定的泛化能力，面对没有见过的state或者action有一定的决策能力。

将Policy表达成参数θ的目标函数，有如下几种形式，start value是针对拥有起始状态的情况下求起始状态s1获得的reward，average value针对不存在起始状态而且停止状态也不固定的情况，在这些可能的状态上计算平均获得的reward，或者只计算immdiate reward的期望。其中d (s)指状态分布函数。

1. Monte-carlo policy gradient
2. Actor-critic policy gradient