Openai gym 第二阶段 爬山算法的实现

1. 游戏规则：

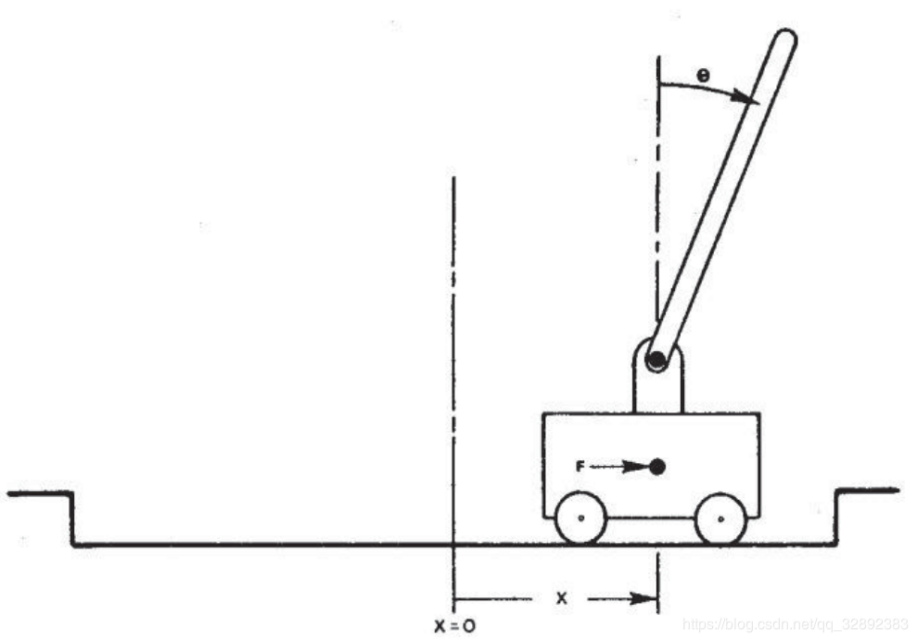
Car游戏里面有一个小车，上有竖着一根杆子，每次重置后的初始状态会有所不同。小车需要左右移动来保持杆子竖直，为了保证游戏继续进行需要满足以下两个条件：

杆子倾斜的角度*θ*

θ不能大于15°

小车移动的位置*x*

x需保持在一定范围（中间到两边各2.4个单位长度）



**动作**：

左移（0）

右移（1）

**状态变量**：

x：小车在轨道上的位置

θ：杆子与竖直方向的夹角

x˙：小车速度

θ˙：角度变化率

1. 算法思想；

简单来说，爬山算法分为

1.随机选择一个符合条件的解

2.按某一个间隔n递增

3.得出一个解集，找出解集中的最优解

4.将这个最优解依据上面的方法再构造一个解集，再求最优解

5.直到前一次的最优解和后一次的最优解相同才结束爬山

这五个步骤。

对于CartPole倒立摆这个具体问题，其控制输出a很简单，就是向左或者向右。输入p就是车的位置，速度，杆的角度，角速度。首先需要构造一个从输入到输出的映射:a=f（p）

那么这里显然我们可以使用最简单的映射，就是线性映射： a=wp，w即为参数向量。然后我们就模拟修改这些参数，从而得到一个新的“点”。

最后的问题就是如何判断w好坏了，想法也很简单，每次运行一次episode即一次实验，累加计算得到的reward值，用Score得分表示，代表这个参数效果更好。

显然，这是一个局部最优的算法，无法保证结果最优