# 第一次实验报告

**实验目的**：基于规则完成游戏设计

**实验准备**：python环境以及gym库安装

**实验过程：**第一次的实验要求是基于规则的实验因为没有太多的约束，因此我在看游戏规则时想出来一个简单的执行规则。

**游戏执行规则**；首先是通过对角度进行判断，当角度在一定范围之内，即木棒接近垂直的时候左右两边随意晃动，当木棒角度超出范围的时候，对角度，角度变化率以及车的速度进行第一次方向上的判断，如果角度和角度变化率不同号的时候则保持原来的速度方向，当两者同向的时候，则对速度的方向进行更改。之后的速度方向就和第一次判断的相同，当角度变小到一定范围时候就重复之前的步骤。这个规则简单也易于实现，但是结果十分的不稳定从十几分到九十多分都有可能，平均下来接近50左右。通过对数据的观察发现，失败的情况一般是由于角度变化率过大导致在调整放方向的时候时间不足使得当我们将角度变化率于角度异号的时候角度已经增加到无法挽回的地步，游戏就失败了。因为发现了角度变化率的重大影响，因此在随机选取方向的时候自己又简单的加了一个约束条件，就是变化率过大的时候将速度方向与角度反向。但这样效果并不明显，甚至可能并没有任何帮助。

**代码：（基于上述规则写的）**

import gym

import random

env = gym.make('CartPole-v0')

n=0

x=0

for i\_episode in range(20):

observation = env.reset()

for step in range(100):

env.render()

#print(observation)

dis=float(observation[0])

v=float(observation[1])

degree=float(observation[2])

change=float(observation[3])

if(degree>-0.1 and degree<0.1):

action = random.randint(0,1)

n=0

if(abs(change)>0.8):

if(v>=0):

action=0

else:

action=1

else:

if(n==0):

n=n+1

last=change=float(observation[2])

if(degree>0 and change>0):

if(v>=0):

action=0

else:

action=1

elif(degree<0 and change<0):

if(v>=0):

action=0

else:

action=1

else:

if(v>=0):

action=1

else:

action=0

lastaction=action

else:

action=lastaction

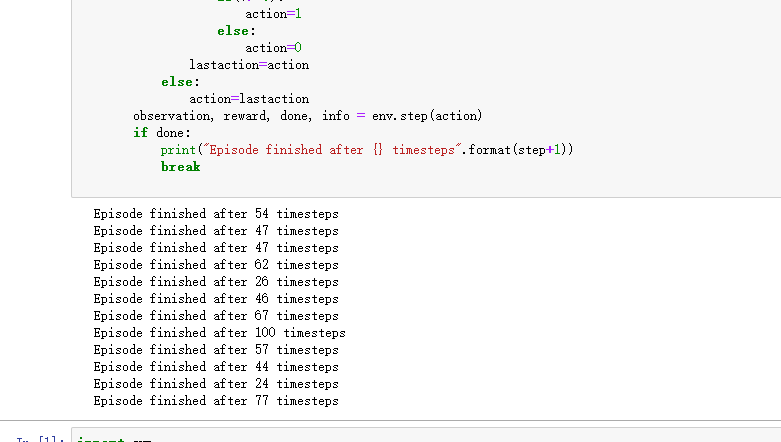
observation, reward, done, info = env.step(action)

if done:

print("Episode finished after {} timesteps".format(step+1))

Break

**实验截图：**



可以看到实验的结果波动很大，但平均在50次左右。原因大部分是由于角度变化速率过大，还有一部分的原因是由于，一些决策的失误，这些都是规则不够完善所导致的。所以不管从实现角度还是实现手段都还有很大的提升空间。可能简单的规则并不能让游戏完成还需要取使用更多的方法去实现该实验。

**另一个实现途径（模拟退火）**

该思路来源网上，是在搜索gym出现问题时看到的这是一个较为简单的实现办法效果远比自己的规则好的多，而且更加稳定。思路：对于CartPole倒立摆，其控制输出A很简单，就是向左或者向右。输入O

就是车的位置，速度，杆的角度，角速度。

为了应用SA，首先需要构造一个从输入到输出的映射:

A=f(O)

那么这里显然我们可以使用最简单的映射，就是线性映射：

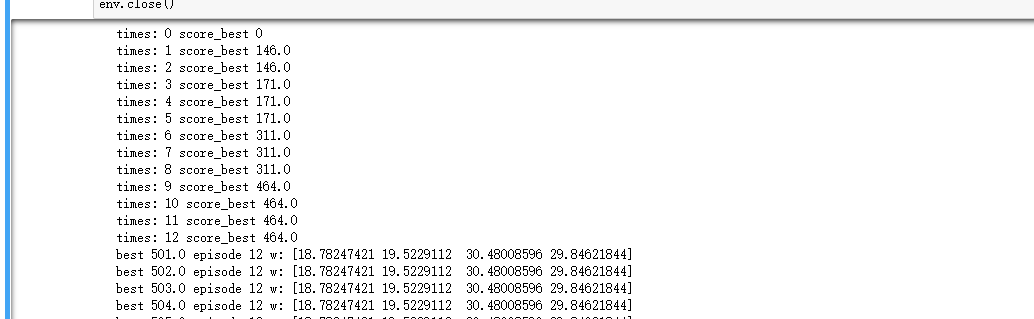
A=wO

w

即为参数向量。模拟退火就来改这些参数，从而得到一个新的“点”。

再下来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

**实验截图**



可以看到该方法随着试验次数增加效果也越好，也是基于规则实现该问题的一个简单有效方法。

**实验中的问题：**

在运行时很多情况下会出下突然卡住的情况之后jupyter会显示kernel died，这问题出现了多次，暂时没找到很好的解决办法，只好在后期实验慢慢改进