**cart-pole实验报告**

2016202106

张艺邻

1. **背景介绍**
   1. **cartpole**

cartpole-v0和cartpole-v1是openaigym上的一个系列的小车平衡游戏。小车上有一个可摇晃的杆子，小车可被操纵向左移或向右移，在杆子倾斜角度大于十五度或小车距离绝对值大于2.4时，游戏失败；否则，玩家获得点数为1的回报值，游戏继续。

这个平台的接口为玩家提供了方便的接口，其中env.step(parameter)接受action（值为0或1）作为参数，返回observation，reward，done，info。

observation是一个大小为4的一维数组，值分别是小车当前的位置，速度，旋转角度，和旋转速度；reward是action之后的报酬，若小车未失去平衡则为1；done为1则小车失去平衡，反之为0；info为调试信息，通常不用。

observation

| **Num** | **Observation** | **Min** | **Max** |
| --- | --- | --- | --- |
| 0 | Cart Position | -2.4 | 2.4 |
| 1 | Cart Velocity | -Inf | Inf |
| 2 | Pole Angle | ~ -12° | ~ 12° |
| 3 | Pole Velocity At Tip | -Inf | Inf |

action

| **Num** | **Action** |
| --- | --- |
| 0 | Push cart to the left |
| 1 | Push cart to the right |

玩家的任务即是通过观察值observation，制定策略决定左移（action = 0）或右移（action = 1），使得小车越长时间保持平衡越好

**2.pytorch**

Pytorch由Facebook开源的神经网络框架，Torch 是一个经典的对多维矩阵数据进行操作的张量（tensor ）库，在机器学习和其他数学密集型应用有广泛应用。与Tensorflow的静态计算图不同，pytorch的计算图是动态的，可以根据计算需要实时改变计算图。作为经典机器学习库 Torch 的端口，PyTorch 为 Python 语言使用者提供了舒适的写代码选择。

PyTorch的设计追求最少的封装，设计遵循tensor→variable(autograd)→nn.Module 三个由低到高的抽象层次，分别代表高维数组（张量）、自动求导（变量）和神经网络（层/模块），而且这三个抽象之间联系紧密，可以同时进行修改和操作。

在本次实验中，主要利用pytorch完成了机器学习的线性回归，逻辑回归和深度强化学习

1. **普通方法**

**1.比例选择法**

很容易认识到，游戏的结束与否，取决于位置和木杆角度两个变量，而这两个变量与action的关系是很直接的，所以我决定根据这两个变量的比例选择action。

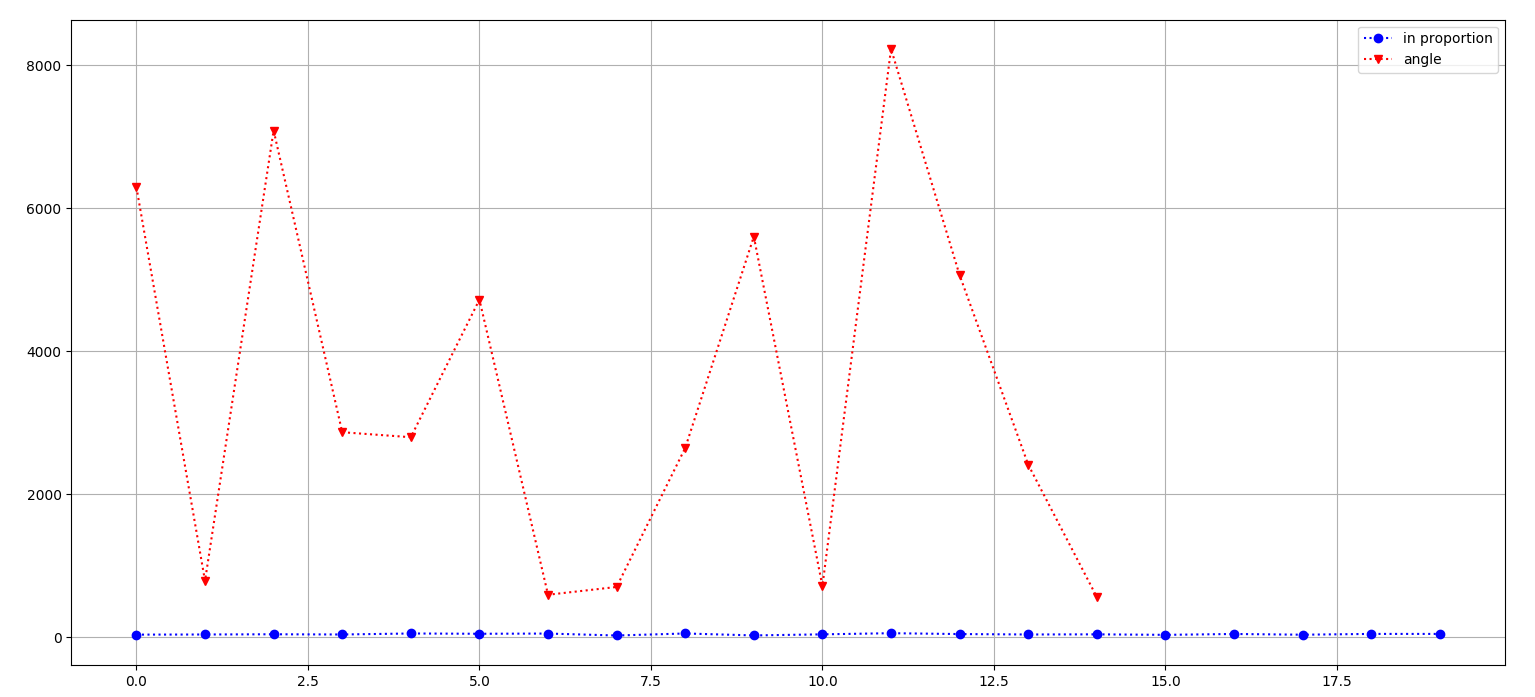
比例，即是它们在取值范围内的比例。比如，当位置为2，角度为0.1时，它们分别的比例为2 / 2.4 = 0.83, 0.1/cos(78) = 0.48,因为0.83 > 0.48所以优先选择能使距离绝对值变小的操作，即action = 0。

**2.维持角度法**

维持角度法的想法很朴素，就是木杆向右斜，就把小车向右移，木杆向左斜，就把小车向左移。为了不让小车驶出范围，我们额外考虑了小车的位置，太右则左移，太左则右移



3.**两者比较**



蓝色为根据比例选择，红色为根据角度选择。

显然，根据角度的算法远远超过了根据比例的算法

但同时，根据算法的问题是，它虽然能保持木杆的平衡，但小车却每次都以微小的速度向某一端移动直至到达端点，游戏失败。