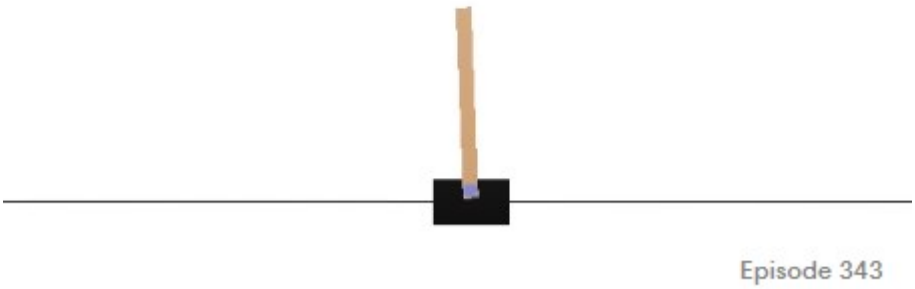
任务说明

环境配置：

1. 安装Python环境（推荐安装Anaconda python 3.6版本）
2. 安装OpenAI gym （http://gym.openai.com/）
3. 运行提供的代码，即执行 `python main.py` ，确保可以运行

任务1

在环境配置的第3步运行 `python main.py` 时，如果环境配置正常，我们会看到如下图所示的CartPole环境（详细说明参见：http://gym.openai.com/envs/CartPole-v1/）



CartPole是一个经典的控制问题，简单来说，其目标就是根据当前观察到的黑色小车及杆子的状态，控制小车向左或者向右，以保持杆子处于直立状态。CartPole环境中具体可以观察到的Observation以及可以采取的action如下：

```
Observation:
    Type: Box(4)
    Num Observation                Min         Max
    0   Cart Position             -4.8          4.8
    1   Cart Velocity             -Inf          Inf
    2   Pole Angle                 -24°         24°
    3   Pole Velocity At Tip      -Inf          Inf

Actions:
    Type: Discrete(2)
    Num Action
    0   Push cart to the left
    1   Push cart to the right
```

请查阅相关资料，了解强化学习的相关概念，并实现至少一种强化学习算法 (实验多个算法有加分)，解决CartPole问题。
注1：RL算法流程本身需要自己实现，不能直接调用现有的强化学习算法包，但实现中可以调用其他任意python包，如numpy，tensorflow等。
注2：连续20个episode获得200以上的reward（即保持杆子直立200个timesteps以上）即视为解决。

任务2

现在假设因为某种原因，我们无法再观察到observation中关于速度的部分，即我们只能观察到如下Observation：

```
Observation:
    Type: Box(2)
    Num Observation                 Min         Max
    0   Cart Position             -4.8            4.8
    1   Pole Angle                 -24°           24°
```

main.py中提供了创建这个部分观测版CartPole环境的代码，即：

```
env = ReduceObs(gym.make('CartPole-v1'))
```

首先尝试直接使用任务1中实现的RL算法，看是否能够解决当前这个部分观测版本的CartPole环境。
如果可以，请比较该算法在两个环境中的效果差异。
如果不能直接解决，请尝试修改任务1中实现的算法，或者实现新的算法，以解决部分观测版的CartPole环境。

任务3 （Bonus）

首先确认任务2中实现的RL算法是否满足如下要求：

> 运行时每一个step的输入只有当前step观测到的原始Observation中的2维信息。

如果满足，则任务3达成。
如果不满足，请尝试实现满足该要求的RL算法，来解决任务2中部分观测版本的Cartpole环境，并分析比较和任务2中实现算法的效果差异。

提交内容

1. 可以直接运行并复现结果的代码（包含必要的注释，并简单说明代码的运行方法）
2. 一份文档，简要描述选择的工具，实现的算法，训练、调参以及评估的过程等，并包括训练过程中episode reward随训练步数变化的曲线。

提交方式：所有内容打包发送到 sjx@inspirai.com, xj@inspirai.com