# 人工智能导论第三次报告

——在gym中实现基于深度学习的CartPole

何文宇 2017202012

1. 实验目的

在OpenAiGym中，在先前选取的CartPole游戏环境中实现一个基于深度学习的计算模型，从而在游戏中获得更好的成绩。

1. 环境介绍

CartPole是一个车杆平衡游戏，游戏规定小车只能在原点左右各2.4个单位距离之间移动，且杆摆动的幅度不能超过15°。在游戏中，可以观察到每个状态的小车位置、杆摆动幅度、小车速度以及杆角速度。游戏每持续一步，成绩加一，最高分为200，200分后游戏仍可进行，但成绩不再增长。

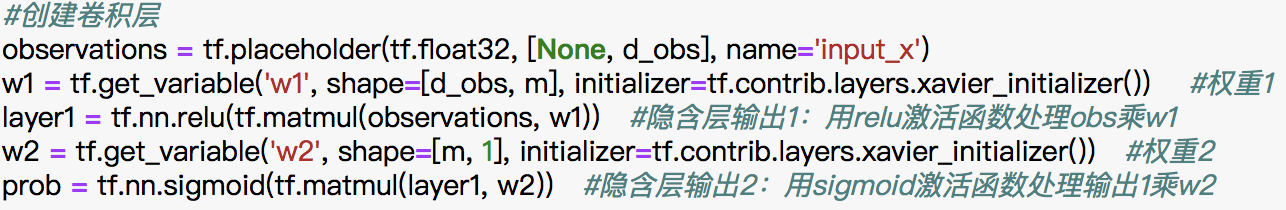
1. 算法思想

建立策略网络，每次训练向模型输入一定的数据，观察所得模型得分。

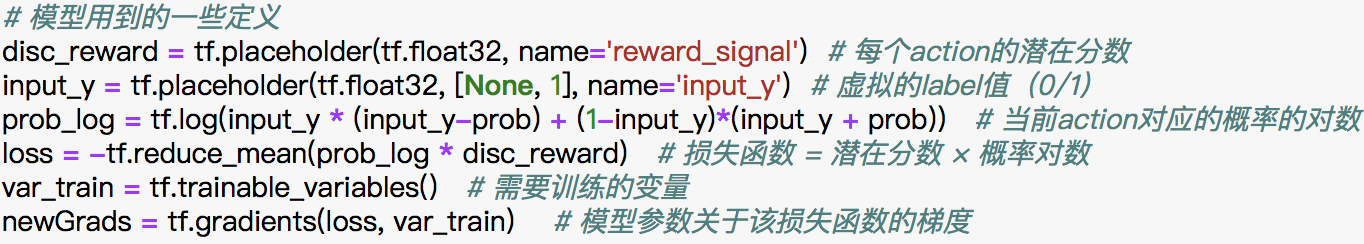
首先设置一些参数。隐藏层节点数m设为50，一次训练所选取的样本数batch\_size 设为25，学习速率learning\_rate 设为0.1，回报的衰减系数decay 设为0.99，要输入的CartPole环境信息的维度d\_obs为4。



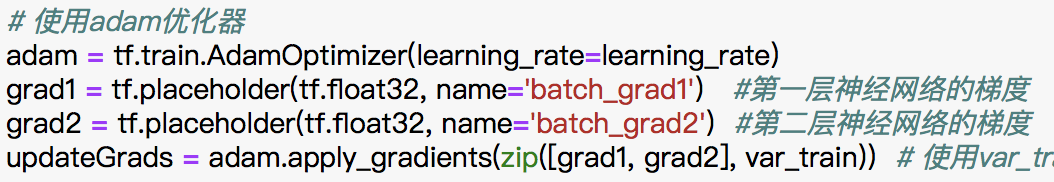
然后开始创建卷积层并定义输出，应输入多组CartPole环境信息observation，输出action向左或向右的概率prob。创建observation输入策略网络的格式observations：行数不定、列数为4的矩阵。创建第一个隐含层输出：observations乘权重w1后用relu激活函数处理。创建第二个隐含层输出即prob：第一个隐含层输出乘权重w2后用sigmoid激活函数处理。因为action可能的值有两个，所以只需输出其中一种action的概率，即可推出另一种action的概率。



创建与action相反的label输入策略网络的格式input\_y：行数不定、列数为1的矩阵，它可以用于对已完成的action进行纠正。创建每个action的潜在分数输入策略网络的格式disc\_reward。求解模型参数关于损失函数的梯度：将损失函数（=潜在分数×action概率的对数）和需要训练的变量var\_train用gradients函数求解。

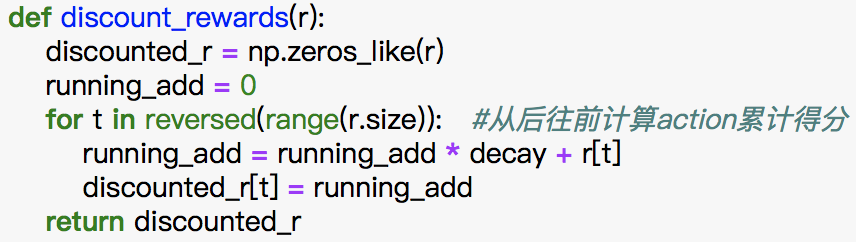


使用Adam算法的优化器来优化模型。用adam.apply\_gradients定义更新模型参数的操作updateGrads，在其中输入两层策略网络参数的梯度和需要训练的变量var\_train。此处的优化器在每个batch中会被使用一次，即积累到一个batch\_size的样本时，再用所得梯度来更新参数。

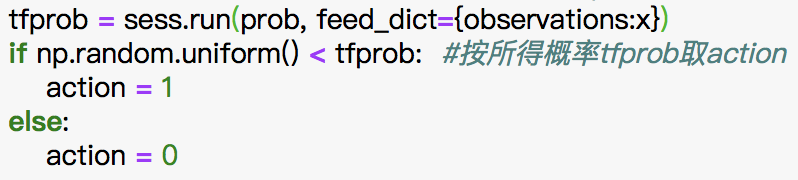


定义函数discount\_rewards来计算当前action的潜在分数，输入r为当前action及之后每一步获得的reward数组，输出discounted\_r是r对应的每一步的潜在分数数组，形如：

[r1, r2+r1\*decay, r3+r2\*decay+r1\*decay\*decay, ...]



开始实验。游戏的每一步中，向神经网络输入环境信息，得到action取值为1的概率，根据该概率随机取action值：

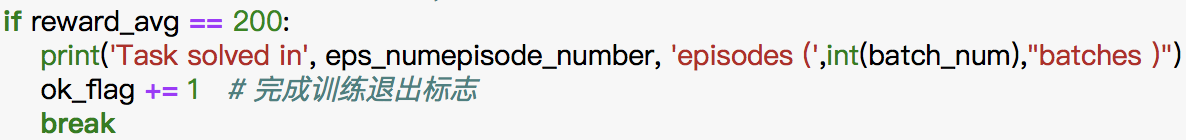


当一轮游戏结束，将这一轮游戏的每一步的环境信息列表、label列表和回报列表分别压入矩阵，并对回报列表矩阵中的潜在分数做标准化处理（减均值后除以标准差）得到新的回报矩阵。用之前定义的newGrads操作和三个矩阵数据得到新的梯度，并加到储存参数梯度的缓冲器gradBuffer上。之后将用其更新模型参数。



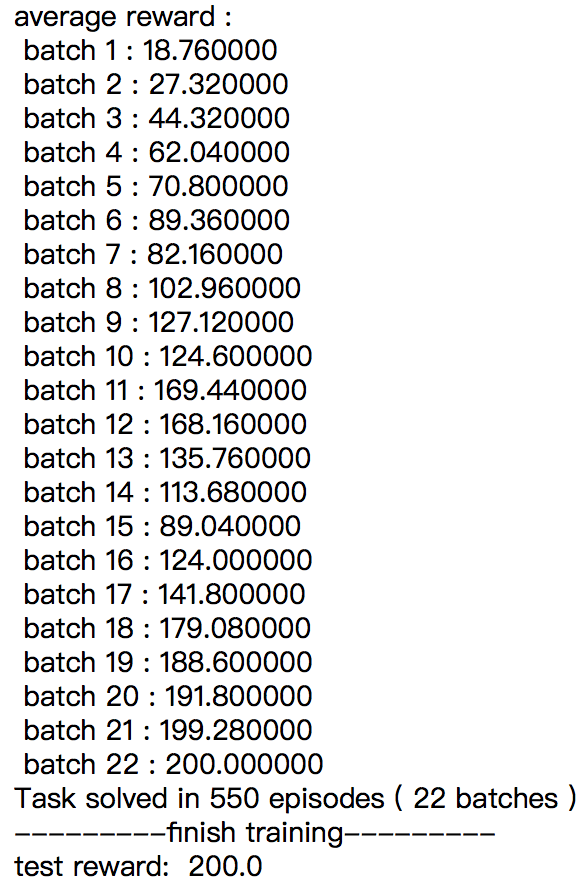
当一个batch结束即做了batch\_size轮游戏之后，整理一次各参数。将参数梯度缓冲器gradBuffer中的参数更新到模型中。计算这个batch中各轮游戏的平均得分，当平均得分已达到满分200时，模型训练完成。

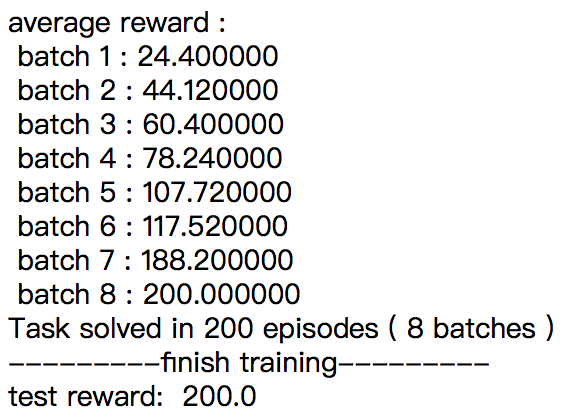


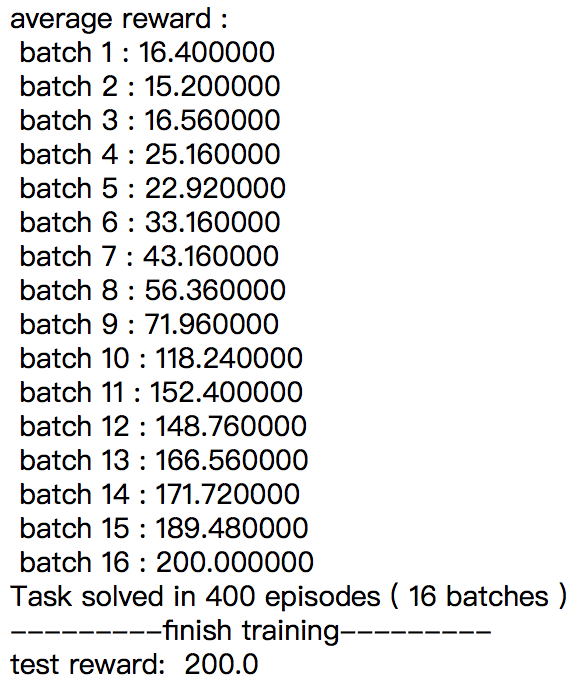


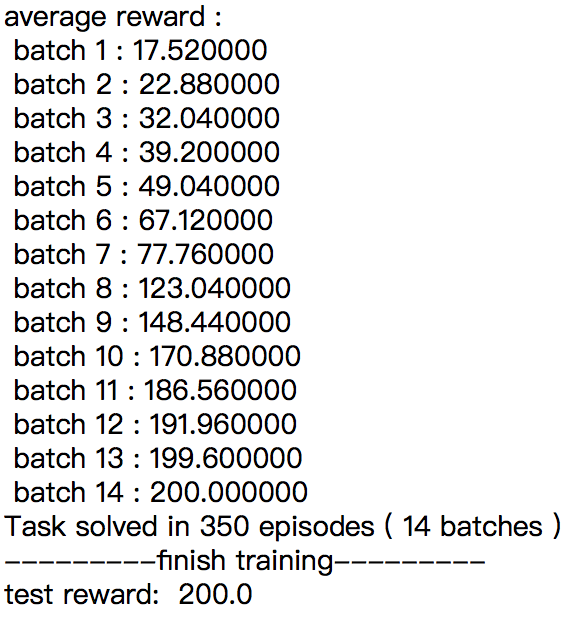
1. 实验结果

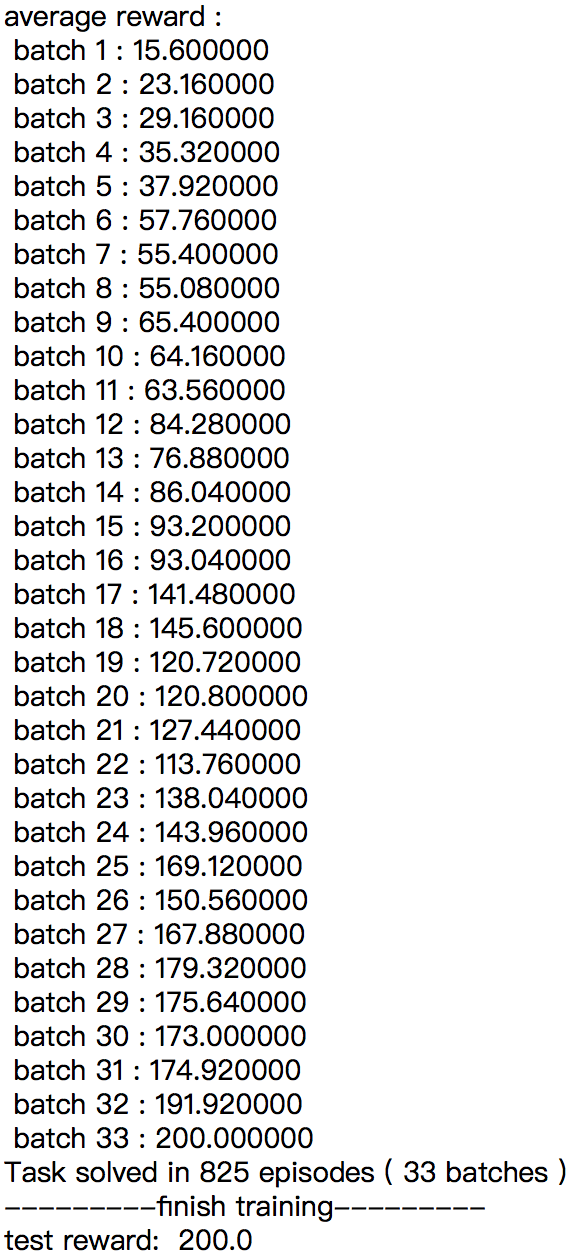
每次的训练过程具有随机性：











在检验中，小车模型完成训练后的测试运行均能达到最高分200，可以看出使用策略网络方法训练的深度学习的模型能较好地进行CartPole游戏。