作业5: Mini AlphaGo

张祎扬 （181840326 181840326@smail.nju.edu.cn）

（南京大学 匡亚明学院，南京 210046）

1. 引言

本次作业基于2016年的nature，实现Alpha Go的框架。关键在于对论文内容以及所提及的算法的理解。

2. 实验内容

2.1 阅读AlphaGo原文和代码

首先我阅读了框架代码里提供的两个demo，发现它的整个框架都是基本一致的，只是使用的agent不同。所以我觉得在实现MCTS方法时最好也封装成agent。

在提供的demo中，首先是对agent进行了初始化，然后对agent进行训练，并且将训练得到的模型保存，然后对训练好的agent进行评估。

2.1.0 论文分析和算法介绍

在开始实验之前，我差不多花了整整一天的时间去研读论文，试图完全弄清楚论文中的思路和算法，在这里做一个简单的梳理。

蒙特卡洛树搜索分为四步。第一步是selection，选择Q值和u值最大的一边，所以这两个值都必须存储在结点中，同时u值的计算和先验概率P和访问次数nvisits都有关，所以这些值都要存储在树结点中。在alphago的文章中，使用的是如下公式：

$$action = argmax_a(Q(s,a)+u(s,a))$$

$$u(s,a) = c_{puct}P(s,a)\frac{\sqrt{\sum_b N_r(s,b)}}{1+N_r(s,a)}$$

而其中的$P(s,a)$则是利用策略网络进行计算的。而针对$Q(s,a)$的计算公式为

$$V(s_L) = (1-\lambda)v_\theta(s_L) + \lambda z_L Q(s,a) = \frac{1}{N(s,a)}\sum_{i=1}^{n} 1(s,a,i)V(s_L^i)$$

而其中的$v_\theta(s_L)$是利用价值网络进行计算的，λz_L是根据快速走子网络进行计算的。

第二步是expansion，扩张一个叶结点，并且用policy network处理，把输出的概率存储作为该结点的先验概率P。第三步是evaluation，用两种方法来对叶结点进行评估：用value network和通过rollout到游戏结束，然后用函数r计算获胜者。第四步是用子树下存储的数据更新Q值。

上面提到了三个网络训练，分别是监督学习的策略网络 SL policy network,快速走子网络Rollout policy和价值网络 Value network,整体网络训练分为三个部分：classification,reinforcement 和 regression.在classification部分，主要是通过前人的棋谱来有监督的让模型进行学习如何采取action，主要学习出一个准确率较低但是速度很快的快速走子网络和一个准确率相对较高但是较为复杂的SL policy network。在reinforcement部分，利用classification阶段学习出来的网络权重来初始化此时的网络权重，同时通过自我博弈，也就是self play，和自己的历史版本进行对抗，来不断进一步学习。在regression部分，主要需要利用强化学习训练得出的策略网络来初始化权重并且生成训练数据。

2.1.1 在围棋环境中实现MCTS方法

按照讲义提示，可以使用RL算法和Uniform Random对手博弈。需要训练得到两个不同深度的网络结构，较深的是初始的policy nets，较浅的是rollout policy nets。框架代码已经提供了两种方法，分别是DQN和polcy gradient。为了实现mcts算法，我在同样的文件夹下新建文件mcts.py.因为蒙特卡洛是树搜索，所以先新建一个类treenode，然后在该类的基础上实现mcts算法，再把它封装成agent。

具体代码实现让我一筹莫展了非常久，我尝试在github上搜索可供参考的代码，但是能够找到的资源非常有限。最终参考了https://github.com/Trussin/Chinese-Chess-AI/blob/master/MCTS.py和https://github.com/AIDefender/Intro_AI/tree/master/PA5/mini_go/MCTS.

首先是定义了类TreeNode，并且在其中存储了父结点，子结点，p值，Q值，u值等，同时定义了一些基本的方法，比如扩张叶结点，判断是否为叶结点或根结点之类的。然后定义类MCTS，并且定义了evaluate，get_move，update等方法，然后封装成MCTSAgent。在博弈时基本可以参考demo的实现，让MCTSAgent和random agent进行博弈。

2.1.2 实现对手池方法

2.1.3 用训练完成后的AI vs 用均匀随机rollout的MCTS AI

2.2 尝试修改参数得到更好的学习性能

3. 结束语

在这之前，我对tensorflow完全没有一点接触，所以在阅读框架代码的时候非常吃力，有很多不懂的地方，所以我认为在完成这次作业之前首先应该系统学习一下tensorflow（据我所知人工智能院的同学在大一暑期学校是学过的，但是我们院没有）。在开始作业之前，我就告诉自己认真阅读论文并且理解论文算法思想的重要性，于是我花了一整天阅读论文，但还是发现对于目前的我来说阅读论文只能让我形成一个大概的框架，还有很多细节的地方并不能搞清楚。所以我参考了网上和知乎上的很多资料，但是感觉它们都有说得不到位的地方。

在对算法有了一个大概的理解以后，将它实现成代码又是另一个难题，感觉由于自身水平的局限，苦苦思考了很久但是却没有一点思路，完全无法下手。参考了一些大神的代码，发现缺少了tensorflow基础和一些python库的知识，理解起来也非常困难，更别提自己去实现了。

考试周连续的熬夜让我到现在还没有缓过来，作业好几天一筹莫展，眼看着要解脱了，但是又延期了？……

老师说其实考核不是重点，真正学到东西才是。我觉得这话是没错的。对于我来说，作业5是一个难度非常大，远远超出了我目前水平的任务。我觉得想要思路清晰地完成它，首先得自学一些够用的tensorflow和python，而不是在看框架代码的时候现场去查，这样是不能形成完整的知识体系的，所以我觉得不如先花时间学习提升自己的基础能力。无论是现在截止，还是接下来的短短七天，倘若只是为

了作业上更好看而逼迫自己不断完成这些代码，最后也就演变成对大佬的代码的搬运了，也许到最终自己还是一知半解。当自身的知识储备还不够的时候，强行去够目标，为了完成作业而很有针对性地吸取知识，其实是不成体系的。所以我决定在接下来的时间甚至整个寒假系统地去学习一些东西，填补自己的知识漏洞，提高能力。至于七天后自己能不能交上一份有改动的作业，其实这个可能性是不太大的……说实话这个作业的难度再给我七天也帮助不大（

所以还是放宽心，利用假期潜心学习吧。非常感谢这门课在短短的一学期里让我接触并了解了这么多算法和知识，同时也通过一些作业任务让我对学习的意义有了更深的思考。我开始不在意分数而去思考一些能力层面的东西，这大概是这门课带给我最宝贵的财富吧。同时，这些作业（尤其是这次的）也让我深刻认识到了自己的菜，从而更有动力去努力地提升自己。

致谢：

References:

[1]https://www.zhihu.com/question/41331593/answer/954760617

[2]https://github.com/Trussin/Chinese-Chess-AI/blob/master/MCTS.py

[3]https://github.com/AIDefender/Intro_AI/tree/master/PA5/mini_go/MCTS