一个不需要人类知识的对弈系统的探索

姜振兴-107x@163.com

最近几年随着深度学习的发展, 特别是在图像识别和语音识别等领域取得了巨大的成功, 人们也开始尝试在强化学习中使用深度学习模型。2013年的时候DeepMind团队就成功使用DQN来玩Atari游戏。2016年AlphaGo战胜围棋世界冠军李世石，2017年AlphaGo战胜排名世界第一的棋手柯洁，随后DeepMind又公布了最新版的AlphaGo Zero，相比于之前的版本，AlphaGo Zero不需要人类棋手的走棋数据进行训练，并且也不依靠人类总结的走棋特征，只需要输入棋盘这样最原始的数据，从随机状态开始，依靠自我对弈进行学习提升，一段时间后，他的棋力甚至超过了AlphaGo。然后DeepMind将同样的算法应用于其它几种棋类游戏，也取得了成功，称为Alpha Zero。

本文给出了一个不需要人类知识，从随机状态开始训练的走五子棋系统的算法和实现。本系统主要基于DQN和MCTS，对大部分棋类游戏具有一般性，算法类似于Alpha Zero，但不完全相同。项目地址请见附录I。

五子棋是一种我小时候经常下的棋类游戏，棋盘比较小，训练所需的资源也比较少，使用一台服务器或者个人电脑（最好有GPU）就可以训练，方便大家个人进行学习研究。五子棋的介绍请查看附录II。

系统中的强化学习

DQN（Deep QLearning Network)

DQN是深度Q学习网络，简单来说即用一个深度神经网络预测动作的Q值。

卷积神经网络最近在图像识别领域取得了很大的成功，这里我们使用卷积神经网络，可以把棋盘看成一张图片来处理。网络结构方面我使用5个卷积层就可以取得不错的效果，当然也可以使用层数更多，结构更复杂的残差网络，像Alpha Zero中的那样。网络的输入为走某步棋之后的棋局，输出为获胜的概率。

具体的输入如下:

通道	特征	取值	大小
通道1	空白位置	空白:1，否则:0	5 x 5
通道2	己方棋子	是己方棋子:1，否则:0	5 x 5
通道3	对方棋子	是对方棋子:1，否则:0	5 x 5
通道4	是否赢棋	赢棋:1，否则:0	5 x 5
通道5	偏置	全为1	5 x 5

可能有些不太明显，需要特别说明的是，这个输入是走完某步棋（动作）之后形成的棋局，所以这样的输入中包含了动作和游戏规则，但没有人为总结的走棋特征。

MCTS（Monte Carlo Tree Search)

MCTS是一棵搏弈树，用来进行模拟走棋。树的节点为某个棋局(S)，节点的边(E)对应走棋动作。根节点为当前棋局，即从当前棋局开始模拟。

树的每条边中存储有：动作的Q值，动作的概率P，选择的次数N，Q值更新的次数N_0。

$Q = DQN(S)$

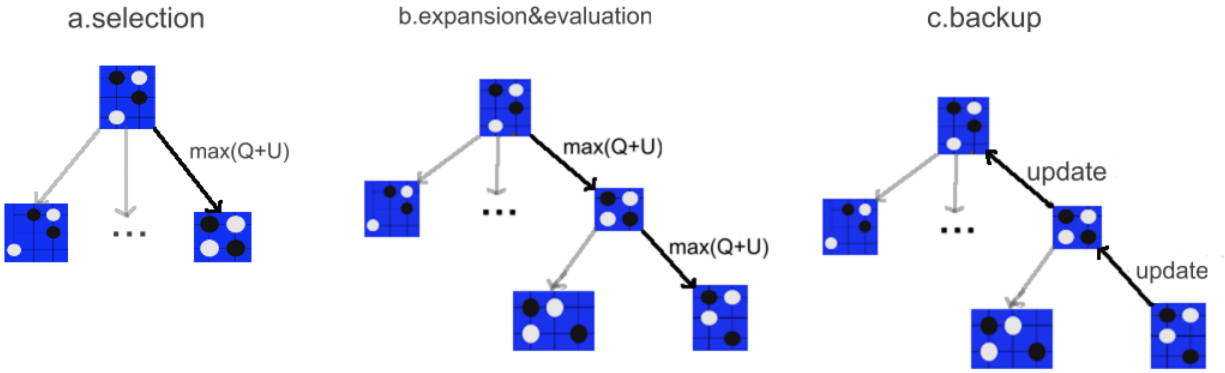
$P = softmax(sigmoid^{-1}(Q))$

奖励值：$U = P / N$

$q = Q + U$

每次模拟走棋时选择q最大的那条边。

MCTS模拟走棋的过程：



a. 选择q=Q+U最大的边。

b. 到达叶子节点时，对叶子节点进行展开，同时对该节点的边的Q值进行评估。

c. 展开叶子节点之后选择q最大的边，然后使用该边的Q值Q'，向后依次更新此次模拟走棋所经过的边。更新公式为：

$N = N + 1$

对与Q'相同选手的边：

$N_0 = N_0 + 1$

$Q = Q + Q'/N_0$（即求Q的平均值）

模拟走棋完成之后选择N最大的边。

训练

方法1. 将DQN随机初始化，从随机状态开始，使用MCTS模拟走棋，模拟完成之后计算概率：$p = \text{softmax}(N^{1/\tau})$，然后按概率选择动作。走棋结束后，将获胜方的棋局及动作标记为1，失败方的棋局及动作标记为0，然后使用这些数据作为训练数据对神经网络进行训练。训练过程中τ逐渐减小，走棋时也逐渐趋向于选择Q值大的动作。

这种训练方法很好，但是需要较多的计算资源，耗费较多的时间。如果大家的计算资源不是很充足的话，使用以下方法也可以取得不错的效果。

方法2. 将DQN随机初始化，从随机状态开始，使用DQN计算各动作的Q值，并以ζ的概率随机选择动作，以1-ζ的概率选择Q最大的动作。走棋结束后，将获胜方的棋局及动作标记为1，失败方的棋局及动作标记为0，然后使用这些数据作为训练数据对神经网络进行训练。训练过程中ζ逐渐减小到0，走棋时也逐渐趋向于选择Q值最大的动作。

循环走棋问题

循环走棋是指从某个局面出发，走了若干步之后又重新回到这个局面（称之为圈）。围棋是落子类游戏，棋子越来越多，所以不易出现重复局面，但是存在打劫问题，所以alphaGo zero的输入中包括了历史局面。

移动棋子类的游戏，特别是当最后棋子较少时容易出现循环走棋的情况，但是圈的长度不确定，且棋的规则里也没有规定不允许出现圈，只是如果没有一方能取胜的话就判为和棋。所以我们在训练的过程中不允许走重复的棋，并且把圈拿出来，将圈里的局面置为和棋。

模型动态调整

在使用MCTS模拟走棋的过程中，对双方都使用我们训练好的同一个模型进行模拟，这样可能并不能准确反映对方选手的走棋特点。所以对于不同的对手，我们可以使用他们走的

棋来对我们的模型进行实时的调整，以更好地预测对手的走棋。但是由于对手走的棋量太少，对已经训练好的模型影响较小，这时可以适当加大学习率，或者直接将对手走的棋保存起来，在模拟对手走棋的过程中，遇到同样的棋局，以一定的概率（如70%）选择对手走过的动作，剩下的情况选择q值最大的动作。

总结

我们使用方法2进行训练，得到的DQN达到了初级以上的水平，使用MCTS走棋时可以达到高手的水平。相信如果使用方法1进行训练的话会有更好的表现。另外像循环走棋问题，模型动态调整问题等还需要更进一步的探讨。

引用

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. *Computer Science*, 2013.

[2] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, Demis Hassabis. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature*, 2016.

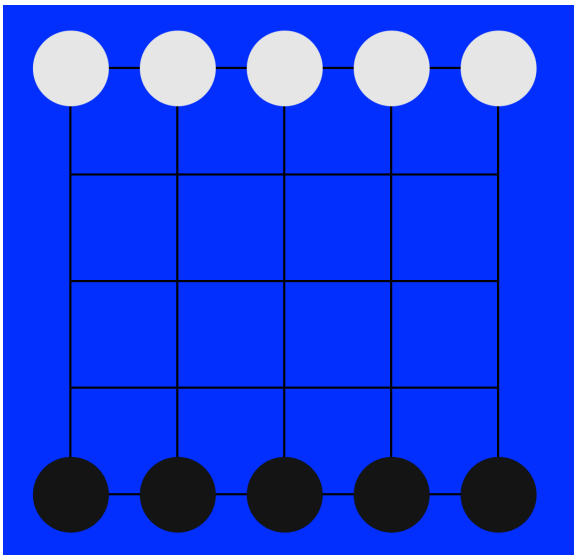
[3] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, Demis Hassabis. Mastering the Game of Go without Human Knowledge. *Nature*, 2017.

附录I. 项目地址

https://github.com/jiangzhenxing/5stone_chess

附录II. 五子棋介绍

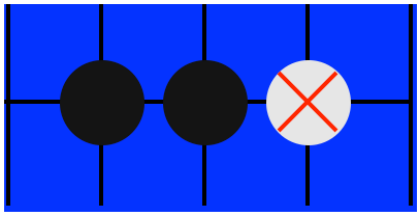
1. 五子棋棋盘如下：



2. 规则。

1) 走棋:一次只能上下左右移动一步

2) 吃子:



如上所示:

a. 任意一个黑子走至当前位置后

b. 形成在一条直线上有两个黑子对一个白子(两个黑子连续，白子在边上)

c. 这三个棋子是连续的

d. 该直线上只有这三个棋子

此时白子被吃掉(横竖直线均可)。

注意：abcd四个条件缺一不可，白子走到当前位置不会被吃。

3) 赢棋:对方棋子少于两个或无路可走时赢棋。

以上信息亦可见于系统中的help。