阿尔法狗模型原理探究

1. 摘要

阿尔法狗横空出世，战胜当今世界排名第一的棋手，其背后精妙的算法功不可没。本文从阿尔法狗下棋的基本思路谈起，从模仿学习、加强学习与蒙特卡洛树搜索三个角度简单分析其成功背后的奥秘。

二、研究背景

中国围棋历史悠久，无数围棋大师究其一生钻研，尚只能领悟一二。纵横十九线交出三百六十一个交点，一场博弈大戏就在这棋盘上展开。外行人只听说围棋深奥，或闻聂卫平、李世石等围棋大师美名，却未曾深入了解过它。然而，就在几年前，阿尔法狗横空出世，在这以深奥著称的项目中碾压一众大师，以4：1的战绩战胜大师李世石，并且在棋类网站上以Master为名挑战天下大师棋手，无一败绩。同期棋坛冉冉升起的少年天才柯洁放话能够打败阿尔法狗，最终也以失败告终。阿尔法狗阐述了一种新方法去处理围棋，其背后的神经网络、深度学习、蒙特卡洛搜索树等技术延伸价值极高。

三、下棋思路分析

1. 围棋基本规则：

1. 围棋棋盘是19×19路，由若干条横线和竖线交织组成的“田”字形网格，共有361个交叉点。
2. 围棋落子黑先白后，交替着一子于棋盘的点上，允许任何一方放弃落子权而使用虚着。
3. 与直线紧邻的空点称“气”，直线紧邻的点上若有同色棋子存在，则其相互连接成一个不可分割的整体。直线紧邻的点上如果有异色棋子存在，此处的气便不存在。若棋子失去所有的气，就不能在棋盘上存在。
4. 落子后，对方棋子无气，应立即清理对方无气之子；若双方棋子都呈无气状态，应立即清理对方无气之子。
5. 棋盘上的任何一点，若某方落子后，该子立即呈无气状态，但此时又不能提取对方的棋子，则此点称“禁着点”。
6. 着子后不得使对方重复面临曾出现过的局面。
7. 棋局下到双方一致确认着子完毕或对局中有一方中途认输时。
8. 着子完毕的棋局，采用[数子法](https://baike.baidu.com/item/%E6%95%B0%E5%AD%90%E6%B3%95/15089923?fromModule=lemma_inlink)计算胜负。将双方死子清理出盘[外后](https://baike.baidu.com/item/%E5%A4%96%E5%90%8E/18159824?fromModule=lemma_inlink)，对任意一方的不可被提取的棋和此类棋围住的点以子为单位进行计数。双方不可被提取的棋之间的空点各得一半。黑棋得185点及以上为胜，白棋得177点及以上为胜
9. 阿尔法狗下棋基本思路：

根据棋盘特点可知，可用一个361 \* n维的向量来表示一个棋盘的状态：

将一个棋盘状态向量记为s，在此状态下若暂不考虑无法落子的地方，下一步的落子的行动亦可用361维的向量来表示，并记为a。基于此，该问题就转换成了：任意给定一个s状态，寻找最好的应对策略a，让程序按此策略执行，最后占据棋盘上最大的地盘。

阿尔法狗通过深度强化学习算法和人工神经网络模型来学习下棋，其基本原理包含以下数学模型：

状态表示模型：阿尔法狗可用一个矩阵来表示当前的围棋棋盘状态，其中每个元素表示棋盘上一个交叉点的状态。

策略评估模型：阿尔法狗需要评估每个落子位置的价值，以决定下一步应该落在哪里。该模型可用深度卷积神经网络实现，接收围棋棋盘状态作为输入，并输出每个落子位置的得分。

强化学习模型：阿尔法狗通过强化学习算法来优化神经网络模型，以最大化预期收益。它使用自我对弈来收集数据，并利用这些数据来更新神经网络的参数。与此同时，阿尔法狗还会使用蒙特卡罗树搜索算法来评估每个落子位置的价值，并将这些价值作为奖励信号来指导神经网络的更新。

综上所述，阿尔法狗下棋的关键思路是利用MCTS算法来搜索最优策略，并使用神经网络来评估落子位置的得分。通过强化学习算法来优化神经网络模型，并使用自我对弈来收集数据。

1. 阿尔法狗下棋流程：
2. 基于SL Policy Network 预测未来下一步走法，直到L步，到一个叶节点，若该节点访问次数大于某阈值，则使用Tree Policy拓展下一节点(Tree Policy比Rollout Policy多一些特征)。
3. 结合Value Network与Rollout Network对L未来走势进行评估。
4. 将评估结果作为当前棋局下的下一步走法Q值，Q值越大，之后模拟选择次数越多。
5. 结合下一步走法Q值与SL Policy Network再一次模拟，若出现走法相同的情况则对走法Q值取平均。
6. 重复上述步骤多次后选择选择次数最多的走法为下一步走法。

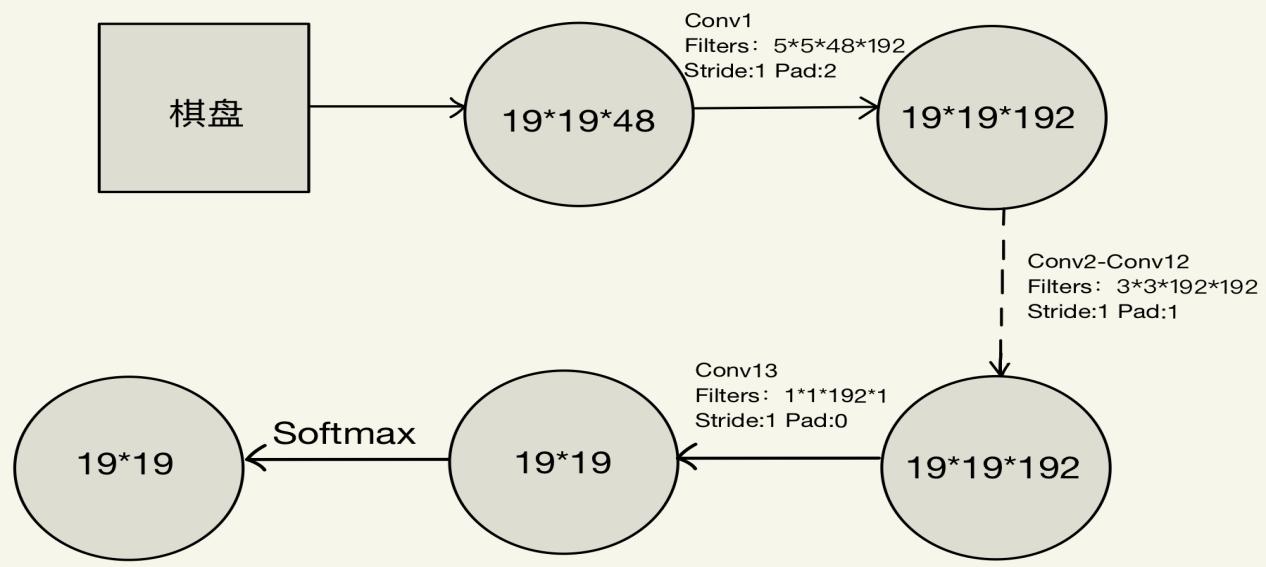
四、应用模型与技术：

1. 模仿学习阶段：

阿尔法狗从网上的围棋对战平台获得人类选手的围棋对弈记录。观察这些棋局，每一个状态s对应一次落子行动a，如此可以得到大量<s,a>样本。将s看做一个19×19×n的二维图像（n是棋盘其他特征），输入一个卷积神经网络进行分类，分类的目标就是落子向量a，通过不断地训练网络，可尽可能的让计算机得到的a接近人类棋手的落子结果。

·SL Policy Network：

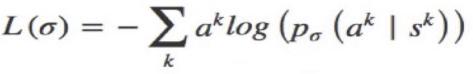
SL Policy Network采用卷积神经网络和ReLu激活函数，用随机梯度上升去最大化其与人类棋手的落子相似度，最后一层使用softmax输出落子概率，其结构如图一。输入策略网络中的状态是一个棋盘状态的表达，输出为每个位置的选择概率，棋盘特征维度如图二。

图一 Policy Network CNN结构



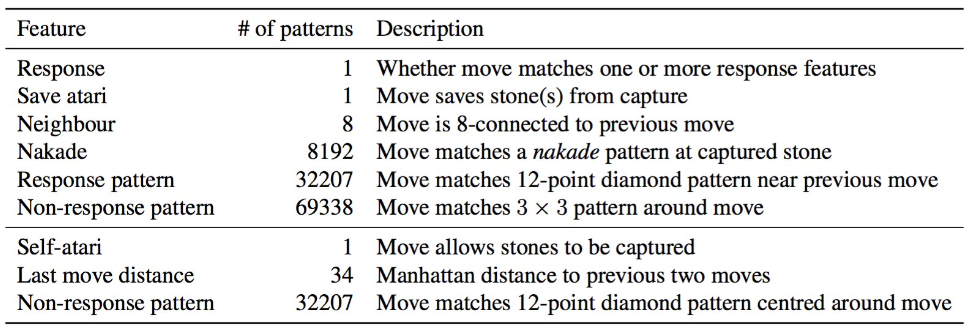
图二 棋盘特征维度

目标：最大化正确动作概率值即最大化对数似然函数。

实际操作：最小化

梯度

·Rollout policy：

Rollout policy与SL policy network的功能一样，但Rollout policy网络简单，落子更快，准确率较低。它以人工提取的围棋特征为输入，输出落子选择，输入特征如图三。

图三 输入特征

1. 增强学习阶段：

·RL Policy Network：

RL Policy Network的网络结构与SL Policy Network的网络结构相同，并且其通过拷贝SL网络的参数进行网络权重的初始化。RL policy network将进行自我对弈，在对弈的结果里面依据最终的胜负来修正权重。

训练方法：

1. 复制SL网络作为增强学习的初始网络。
2. 在当前版本下随机选择之前某版本对弈，得到棋局与输赢结果。
3. 根据棋局与输赢结果利用REINFORCE算法更新参数最大化期望结果。
4. 每500次迭代复制一次当前网络参数至步骤2对手池。

目标：给定策略网络Πθ(s,a)，寻找最优参数θ

评判策略好坏标准：价值Value

回报Return：

价值Value：

反馈Reward：平0，胜1，输-1

目标函数：

梯度下降：

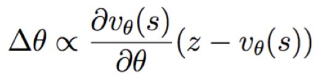
·Value Network：

用RL Policy Network和SL Policy Network生成的棋局数据来训练价值网络。输入是一个棋面，输出是该棋面的胜率。

输出：

数据集：

1. 随机计算时间点U
2. 第1到U-1步使用SL Policy Network生成
3. 第U步随机
4. 第U到T步，使用RL Policy Network生成
5. 计算输赢，将(SU+1，ZU+1)放入数据库

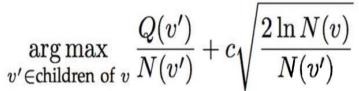
梯度下降：

1. 蒙特卡洛树搜索：

解决问题：将围棋每一步的所有可能选择都作为树的节点，第零层只有1个根节点，第一层有361个节点，第二层有360个节点，需在每一层树节点中搜索出赢概率最大的节点即落子方法。

MCTS算法步骤：

1. 选择（selection）：选择指算法从搜索树的根节点开始，向下递归选择子节点，直至到达叶子节点或者到达具有还未被扩展过的子节点的节点L。该过程可由UCB算法来实现，并记录下每个节点被选择次数和每个节点得到的奖励均值。
2. 扩展（expansion）：若节点 L 不是一个终止节点（或对抗搜索的终局节点），则随机扩展它的一个未被扩展过的后继边缘节点M。
3. 模拟（simulation）：从节点M出发，模拟扩展搜索树，直到找到一个终止节点。
4. 反向传播（Back Propagation）：用模拟所得结果（终止节点的代价或游戏终局分数）回溯更新模拟路径中M以上（含M）节点的奖励均值和被访问次数。

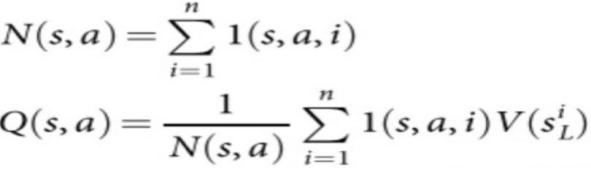
UCB算法：

其中V’表示当前树节点，V表示父节点，Q表示该树节点的累计quality值，N表示该树节点的visit此时，C为常量参数，可控制exploitation和exploration权重。

该公式兼顾exploitation和exploration，表示对每一个节点求一值用于后面的选择，公式左半部分为该节点平均收益值，该值越高表示该节点越值得选择，与exploitation相关。公式右半部分为父节点的总的访问次数除以子节点的访问次数，该值越大表示该节点访问次数越少越值得选择，与exploration相关。

每条蒙特卡洛路径上面都有四个值，Q、N、P、V。Q代表一条路径的好坏，若探索某路径获胜了，那么该路径的Q值就会提升，失败则Q值下降。N表示模拟走子经过这条路径的次数。Q、P初始值都为0，P代表Pα(SL Policy Network），是人类棋手在这条边落子的概率(后面随着网络的改进就不再代表人类棋手)。最开始时没有子节点，随着探索增加逐渐加入子节点。

节点选择方法：

101032850401762058