不依赖人类先验知识的象棋博弈理论与应用实现

李浩东　徐国爱　张国涵　徐国胜

摘　要：阿尔法围棋（Alpha Go）是第一个击败人类职业围棋选手、第一个战胜围棋世界冠军的人工智能机器人，由谷歌公司的DeepMind 团队开发，其自 2016 年横空出世，便以 4：1 的战绩战胜了当时的世界围棋冠军李世石。2017 年，DeepMind 团队推出了 Alpha Go 的最强版——Alpha Go Zero。Alpha Go Zero 最大的特点便是不依赖人类的先验知识。象棋属于二人对抗性游戏的一种，其本质是依靠逻辑推理的二人零和游戏。本文希望通过研究 Alpha Go 和 Alpha Go Zero 的实现思路，从而可以将其思路迁移到象棋上，为象棋软件的研发提供一种新的设计思路。

关键词：象棋　人类先验知识　Alpha Go Zero

Abstract： Alpha Go is the first AI robot to defeat human professional go players and the first world champion in go. It was developed by DeepMind team of Google company. Since its birth in 2016, it has defeated Lee Sedol, the world go champion at that time, with a 4:1 record. In 2017, the DeepMind team launched Alpha Go Zero, the strongest version of Alpha Go. The biggest feature of Alpha Go Zero is that it does not rely on human prior knowledge. Chess belongs to a kind of two person antagonistic game, its essence is a two person zero sum game relying on logical reasoning. This paper hopes that by studying the implementation of Alpha Go Zero, we can transfer its idea to chess, and provide a new design idea for the development of chess software.

Keywords： Chinese Chess Prior-Knowledge of human Alpha go Zero

一、引言

(一)象棋

象棋作为一种历史悠久的二人零和博弈游戏，至今在全世界已经有相当多的爱好者。经过近千年的演变和人们的总结，象棋已经形成了例如中炮过河车对屏风马、仙人指路对卒底炮等大量完善的布局体系。在对弈过程中，需要玩家熟悉布局体系，拥有敏锐的棋感以及精确的计算力。

熟悉布局体系需要棋手记忆并理解海量的棋谱；敏锐的棋感需要棋手经过大量的实战训练培养得到并在之后找高手进行复盘；精确的计算力则是对一个人思维敏捷度的考验。而记忆力和计算度恰恰是计算机所擅长的。因此，实现一个可以击败人类顶尖棋手的软件也成为了现代人工智能不断追求的目标之一。

（二）机器学习

自从计算机出现以来，程序员就对人工智能(AI)——在计算机上实现类人行为产生了浓厚的兴趣。而游戏一直也是人工智能研究的热门话题。在个人电脑时代，人工智能已经在跳棋、双陆棋、国际象棋等棋类中超过了人类[1]。

机器学习是实现人工智能的一种方法。传统编程是将清晰的规则应用于结构化数据。开发人员编程实际上是对数据执行一组特定的指令。而机器学习主要是从一系列示例数据中推断程序和算法的技术，而不是直接对数据做处理。因此，

通过机器学习的方法，输入计算机的数据和期望的输出，让机器自己找到一个合适的算法[2]。

机器学习包含很多子领域。例如监督学习是指输入数据和正确的结果，从而不断的拟合准确的模型；无监督学习是指没有先验知识的情况下，根据没有被标记的训练样本解决模式识别中的各种问题；强化学习最大的特点是与环境进行交互，在交互过程中不断最大化奖励值或尽可能实现目标。

机器学习中的强化学习领域特别适合游戏[3]。强化学习主要是反复运行程序并评估它完成任务的情况。使用强化学习训练象棋机器人，只需要令其不断的和自己进行对弈，记录下可以使棋局获胜的招法，不断增加在当前盘面下选择该招法的权重，在计算力和资源无限丰富的情况下，就可以得到最强的象棋机器人。

二．传统象棋软件

(一)象棋软件的发展现状

中国象棋作为中华民族传统文件的精华，对软件的研究起源于上世纪 80 年代。其研究成果借鉴了大量的国际象棋理论，并且在近年同样取得了很不错的成果。现今已经有很多顶尖的象棋软件，例如象棋奇兵、旋风、名手、棋天大圣等等。以上象棋软件均为商业软件，并不开源[4]。现如今的象棋软件大都采用分阶段的设计思路，即开局按照棋谱匹配，中局展开完整的搜索，残局记录必杀招法。采用这种方式的

象棋软件已经具备了相当的智能，可以轻松的战胜人类中的顶尖棋手[5]。

1981 年，台湾张耀腾在《人造智慧在电脑象棋中的应用》中用残局做实验，并且介绍和分析了中国象棋的局面评估函数。在随后的 1982 年，台湾廖嘉成便实现一个完整的中国象棋博弈程序，他也是开创并最先采用分阶段的设计思路[6]。

发展到今天，现今的中国象棋软件已经超越人类中的顶级大师，目前的主流的象棋软件总体水平差不多，但算法不尽相同，另外，这些软件都具有自我学习功能。由于一线程序都是商业软件，并不开源，所以其具体评估函数和搜索方法均未知。

如今传统的象棋软件研究已经相当成熟，主要的研究热点集中在提高搜索效率以及设计更为精确的局面评估函数。

（二）传统象棋软件的设计思路

传统的象棋软件都要依赖于人类的先验知识，即开局阶段依赖开局库。开局库是依靠人类经过近千年的经验所总结的布局体系所建立的数据库，主要做法是搜集棋谱，统计分析各种走法，并通过人类大师不断完善新的布局到数据库中，在实战中快速检索到相应的应招。

中局则会依靠博弈树、搜索算法以及局面评估函数来实现。

1. 博弈树

博弈树是一种特殊的树，它可以表示两名游戏参与者之间的一场博弈(游戏)，双方交替行棋，试图获胜。每一个顶点代表一个局面，其下的分支为行棋方所能落子的所有可能。博弈树的结构如图 2.1 所示，S_i 表示一个盘面，当棋手采用棋招 a_j，就可以导致盘面 S_i 到达下一个盘面 S_k。

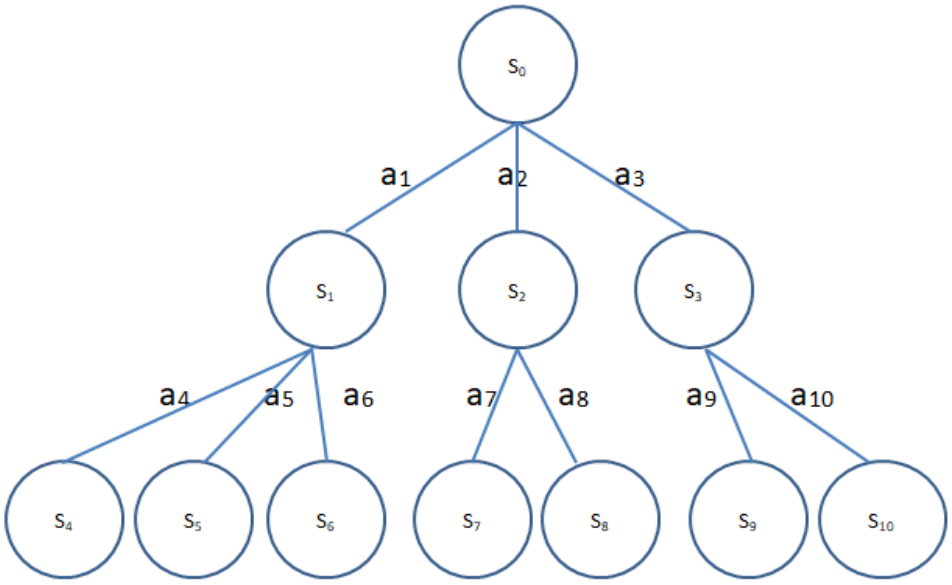

图 2.1 博弈树

2. 搜索算法

前面介绍了博弈树，那么如何可以找到当前盘面下最好的应招？似乎遍历整棵博弈树，就可以立于不败之地。但是，包括围棋、中国象棋、国际象棋在内的各种棋类，想要遍历所有的盘面都是不可能的。因为它们的搜索空间过于庞大。例如，在国际象棋中，一个棋手通常每步有 30 种选择，一

局大约要走 80 步，博弈树的大小大约是 $30^{80} \approx 10^{118}$；围棋通常每个回合约有 250 个合法招法，一局大约会有 150 个回合，因此博弈树的大小约为 $250^{150} \approx 10^{360}$[7]。表 2.1 展示了围棋、中国象棋、国际象棋、五子棋和井字棋的博弈树的复杂度以及空间复杂度。

博弈种类	棋盘大小	博弈树复杂度	空间复杂度
井字棋	3×3	10^5	10^3
国际象棋	8×8	10^{118}	10^{47}
中国象棋	9×10	10^{130}	10^{51}
五子棋	15×15	10^{70}	10^{105}
19 路围棋	19×19	10^{360}	10^{171}

表 2.1 棋类的博弈树复杂度和空间复杂度

在如此庞大的搜索空间中，使用一个简便、有效、快捷的方法找到最好的棋招是十分必要的。因此提出了树搜索算法的概念。在计算机科学中，树搜索算法是一种在许多可能可以在许多决策序列中寻找可以导向最优结果的一个序列算法[8]。有一种搜索算法是极小极大搜索算法(MiniMax)。在该算法中，每个回合中两个互相对立的玩家之间会轮流切换，可以找到完美的落子序列，但它的速度太慢，因此无法应用到复杂游戏中[9]。

3. 局面评估函数

如果可以遍历一个盘面下的所有搜索分支，那么我们就可以承诺在当前盘面下可以选择最优的走法。但是在棋局的早期，如果想要遍历搜索的在人类下棋的过程中，往往会说出“我觉得红方比黑方好”或者“我觉得黑方优于红方”的评论。即使是初学者，也会本能地感觉到他们当前的形势是否优于对手。如果我们可以使得计算机拥有这样的“感觉”，那么在搜索的过程中就可以减少搜索的深度。所以便提出了局面评估函数的概念，局面评估函数就是模仿人类下棋的棋感，可以计算出当前盘面领先多少分的函数[10]。

在许多游戏中，局面评估函数都可以通过游戏知识进行人工制定。例如：

- 国际跳棋：棋盘上的每个棋子都算 1 分，国王算 2 分。用己方的棋子得分减去对手的棋子得分。
- 国际象棋：每个兵算 1 分，每个马或象算 3 分，每个车算 5 分，皇后就是 9 分。用己方棋子的值减去对手棋子的值。

这种评估函数是高度简化的，实际中的国际跳棋和国际象棋会采用更加复杂的局面评估函数。但在这样的局面评估函数下，软件会尽量吃子并保护自己的棋子。此外，它愿意牺牲掉自己分值低的棋子，去吃掉对方一个分值高的棋子。

在商业的象棋软件中，各个版本的软件的局面评估函数是不同的，因此也就造成了各个软件的行棋风格的不同。

残局阶段则使用残局库，残局库的建立有两种方法。第一种是编写大量规则，当盘面符合某些条件则按照既定的规则来落子，同时会编入大量的棋谱或人类大师的研究结果。同样是要依赖人类的先验知识。第二种方法是穷举。因为残局阶段棋子较少，计算机可以尽量穷举出所有的走法和盘面。但这需要构建庞大的数据库，费时费力。

三．Alpha Go 和 Alpha Go Zero 的实现思路

(一) Alpha Go 的实现思路

Alpha Go 包含三个神经网络：两个策略网络以及一个价值网络。第一个策略网络称之为强大的策略网络 (strong policy network)，是使用人类大师的棋谱作为输入，通过监督学习得到的。强大的策略网络的网络结构复杂，体系结构更加深入并且能产生更加精确的结果；第二个策略网络称之为快速策略网络(fast policy network)，同样是使用人类大师的棋谱，通过监督学习得到的。快速策略网络的网络结构较为简单，虽然产生结果不是很准确，但是速度比较快。价值网络(value network)是在强大的策略网络的基础上通过强化学习不断地自对弈得到的[11]。下面分别阐述三个网络的得到的过程。

1. 强大的策略网络(strong policy network)

强大的策略网络是使用人类大师的围棋棋谱通过监督学习得到的。使用盘面作为输入特征(feature)，使用大师在当

前盘面所思考的落子作为标签(label)。强大的策略网络的结构是一个 13 层的卷积网络，所有层都产生一个 19×19 的过滤器并始终保持整个网络的原始棋盘的大小。同时还需要进行相应的零填充。第一层卷积核大小为 5，之后的所有层的卷积核大小为 3。最后一层使用 softmax 激活函数并有一个输出滤波器，前 12 层使用 ReLU 激活函数，每层 192 个输出过滤器[11]。这一落子预测网络是为了准确性而实现的，它拥有极高的准确率，但是速度却很慢

2. 快速走子网络(fast policy network)

快速走子网络是 DeepMind 团队没有公开的部分。我们可以知道它同样使用监督学习，应用大师棋谱作为输入数据，它的网络结构较强大的策略网络要小很多，层数比较少。它的目的不是生成最准确的落子预测，而是为了快速的得到一些比随机落子要好的情况。

3. 价值网络(value network)

价值网络是一个 16 层的卷积神经网络，前 12 层与强策略网络完全相同。第 13 层是一个额外的卷积层，与 2 至 12 层相同。第 14 层是一个具有卷积核大小为 1 的卷积核以及一个输出滤波器的卷积层。网络顶部有两个 Dense 层，一个有 256 个输出和使用 ReLU 激活函数，最后一个有一个输出并且使用 tanh 激活函数。价值网络是在强大的策略网络的基础上通过强化学习得到的。也就是说是使用策略网络进

行自对弈，不断的优化得到了价值网络[12]。图 3.1 展示了得到这三个网络的步骤和流程。

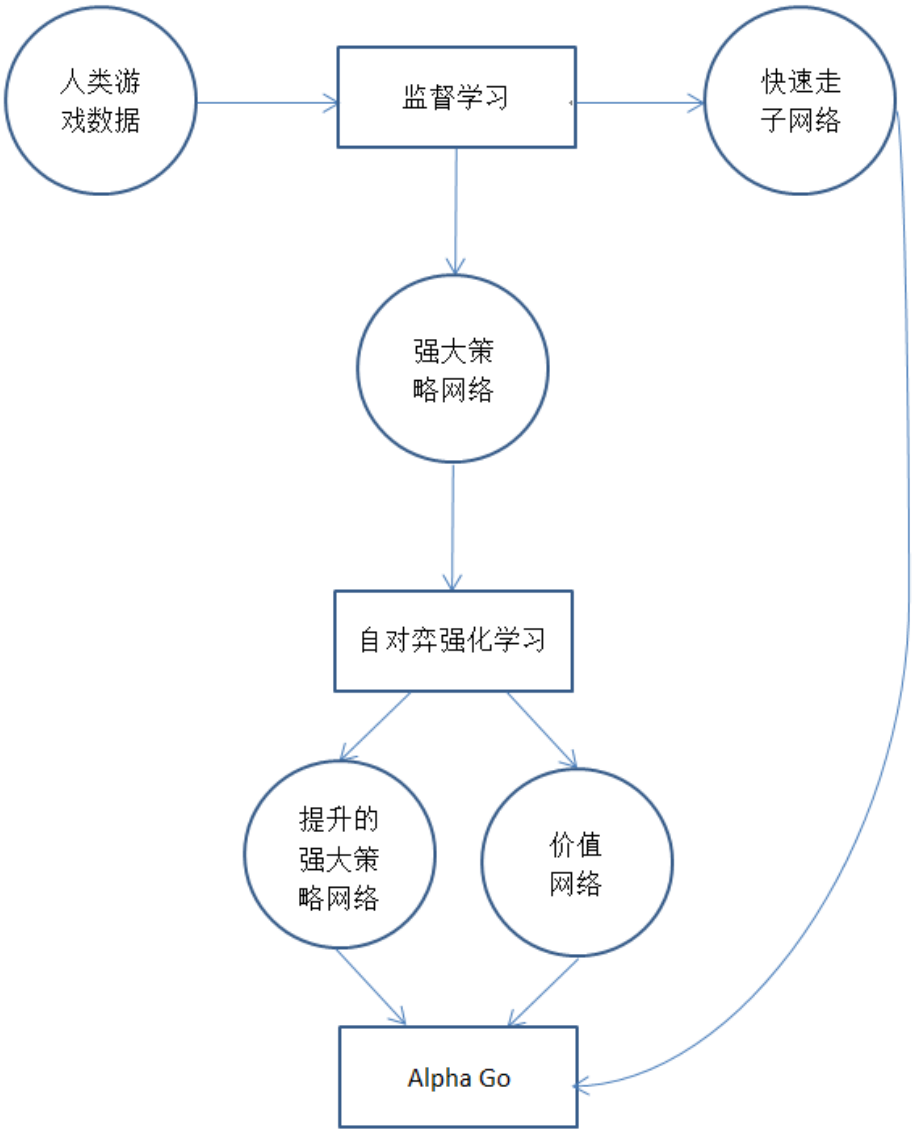

图 3.1 如何训练三个网络

4. 蒙特卡洛搜索算法

第二节中已经介绍过博弈树中搜索算法。搜索算法的目的是为了不必遍历整棵树，就可以尽可能地找到一个较优的落子序列。下面介绍另一种搜索算法——蒙特卡洛树搜索算法(Monte-Carlo Tree Search，MCTS)。在第二节中同样介绍了局面评估函数，局面评估函数可以减少搜索过程中的深度，也就是不必走到终局就可以大概知道当前盘面的优劣。但是如果想给围棋设定一个质量比较好的局面评估函数是非常困难的。蒙特卡洛树搜索提供了一种在没有任何关于游戏的战略知识的情况下评估游戏状态的方法[13]，就是通过大量的模拟棋局，假如从一个盘面开始自对弈一万盘，黑棋赢得次数比白棋多，无论我们选择下棋的策略水平如何都不会对结果产生影响。因为采用的是自对弈策略，对手的水平和己方的水平是一样的。那么我们就有理由认为黑棋处于优势状态。蒙特卡洛的思想就是通过大量的模拟，依靠概率分布来近似估计结果。下面就用蒙特卡洛的思想模拟计算圆周率π。如图 3.2 所示一个正方形内存在一个内切圆。

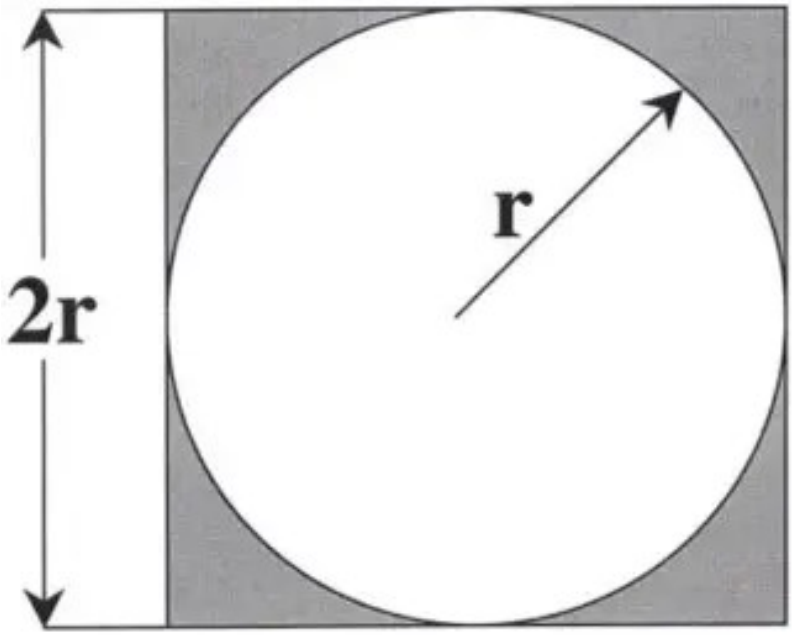

图 3.2 正方形内存在一个内切圆

现在随机在正方形内生成点，并通过计算生成点到圆心的距离来判断该点是否在圆内。随着生成的点的数目越来越多，那么，圆内点的数目/总共生成点的数目逐渐近似等于π/4。

蒙特卡洛树搜索分为四步[13]：

- 选择：依据策略选择一个盘面开始考虑，选择策略我们称为 UCT 公式，下一节会介绍具体的过程。
- 拓展：选择分支直到当前的叶子节点，便在下面随机采用一个合法的落子，之后生成一个盘面作为拓展。
- 模拟：这里使用快速走子网络模拟一盘棋并记录下对弈结果。
- 回溯：将模拟的结果以此回溯到根节点。

之后在循环一定的轮次，得到最终的选择。一轮蒙特卡洛树搜索的过程如图 3.3 所示。图片来源：https://blog.csdn.net/weixin_39878297/article/details/85235694

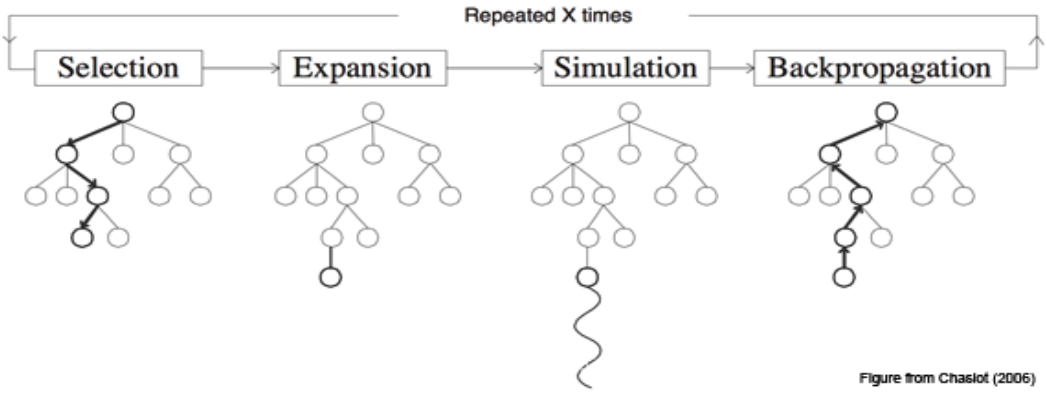

图 3.3 一轮 MCTS 过程

5. 把这些汇总起来就构成了 Alpha GO

不仅要提高软件的落子水平，同时要考虑到软件选择每步棋的时间消耗。如果每步棋都要耗费大量的时间那显然是不正常的。这意味着我们只能执行固定数量的轮次。每一轮 MCTS 模拟都提高了软件对一个合法落子的评估准确度。也就是说每当软件在落子 A 上花费了额外的试验次数，都要相应地在其他落子上减少相应的实验次数。因此急需一个方法来有效且合理的分配有限的模拟轮次。这个方法我们称为树的上限置信区间，或 UCT 算法。UCT 算法在两个相互冲突的目标之间取得平衡[13]。

$$\mathrm{UCT} = \mathrm{argmax}\{\mathrm{Q(s,a)} + c\frac{P(s,a)}{1+N(s,a)}\}$$

- $Q(s,a)$：价值网络给出的当前盘面的评分。
- $P(s,a)$：策略网络给出的当前这招棋的评分。
- $N(s,a)$：该节点的访问次数。

至此，就把三个网络应用 MCTS 结合起来，就形成 AlphaGO。至于 AlphaGO 的水平与策略网络、价值网络还有机器的性能都有很大的关系。当然，MCTS 的轮次越多得到的棋招会越准确，只不过会消耗很多的时间。

(二) Alpha Go Zero 的设计思路

DeepMind 公司推出第一代 AlphaGO 之后，随之推出了名为 Master 第二代版本。尽管 Master 从学习人类棋谱开始，但在强化学习的过程中，它居然能够走出人类棋手难以发现的新招法。

这一现象引出一个明显的问题：如果 Alpha GO 不依赖人类棋手的“启蒙”，而完全使用强化学习会怎样？它能否达到甚至超越人类顶尖棋手的水准，还是仅仅停留在人类初学者的水平？它能否像 Master 一样发现新招法，或者达到人类无法理解的新高度？这些问题的答案都随着 2017 年 Alpha GO Zero(AGZ)的推出而被揭晓。

AGZ 构建于一个优化后的强化学习系统，它在没有任何人类棋谱“启蒙”的情况下从零开始进行自我训练。尽管它在开始的水平比人类棋手要差很多，但是随着训练量的增加，它的水平就很快超越了之前的 Alpha GO。

下面阐述 AGZ 与 Alpha Go 的区别[13]：

- AGZ 使用了一个巨大的神经网络，这个神经网络包含 80 层卷积层，超过 Alpha Go 的神经网络大小的四倍。
- AGZ 使用的是一种创新的强化学习技术。原来的 Alpha Go 独自完成策略网络的训练，之后策略网络被用于优化树搜索。而 AGZ 从一开始就将树搜索与强化学习融合在一起。
- 相比较于 Alpha Go 的三个神经网络，AGZ 只使用一个包含一个输入和两个输出的神经网络。一个输出生成各个选点的概率分布，另一个输出代表当前盘面下，落子方的一个评分。
- 至于 MCTS 的使用，Alpha Go 是在真正的下棋过程中才使用；而 AGZ 在训练和下棋中都使用了[14]。
- AGZ 的树搜索算法与蒙特卡洛树搜索算法类似，其不同之处在于，Alpha Go 树搜索算法只依靠一个策略网络去评估选点，而 AGZ 靠神经网络去指导树搜索。

AGZ 的水平与之前的 Alpha GO 的比赛中取得 100：0 的战绩[15]。并且，在训练中未使用任何手工设计的特征面或者围棋领域的专业知识，仅仅以历史的棋面作为输入，其训练数据全部来自于自对弈。

四．不依赖人类先验知识的象棋博弈理论

(一)不依赖人类先验知识的博弈理论

Alpha Go 的设计思路是完全仿照人类下棋的思路来设计的。当人类面对一个盘面的时候，首先会在头脑中涌现几个“眼光招”，这是依靠人类长时间下棋、复盘的经验总结和积累。这就好比 Alpha Go 中的策略网络，输入一个盘面，就可以得出几个相对较好的选点。这几个选点是强策略网络给出的，也就是“眼光招”。随后人类会在脑中进行推演，思考自己和对手在接下来的数个回合的落子。至于在脑中模拟的回合数，主要取决于个人计算力的差异和下棋时思考的熟练度。这就好比 Alpha Go 中的快速走子网络，虽然不是很精确，但是可以快速地模拟棋局的发展。当最后模拟到相应的盘面时，人类会在脑中给模拟到的盘面一个评价，用来判断当前形势对自己是否有利，也就是说如果选择这招落子，是否可以将自己的形势导向一个比较好的情况；或者说是否会使自己形势变差。而在 Alpha Go 中，我们前面已经说过围棋中是很难建立比较好的局面评估函数，也就是不能给快速模拟之后的盘面一个比较客观的评价。因此 Alpha Go 采用了蒙特卡洛树搜索，经过大量轮次的模拟之后，依靠概率来判断当前盘面的优劣。这也就是大致的将 Alpha Go 和人类下棋的过程进行一个匹配和比较。至于 AGZ 最大的特点

便是不依靠人类的先验知识，也就是不需要人类大师的棋谱，仅仅通过自对弈和强化学习，并可以获得超过人类大师的围棋水平，并且在很多布局和盘面中可以得到人类难以理解并能起到很好效果的落子。

(二)结论与未来工作

参考并结合 Alpha Go 以及 AGZ 的思想，在传统软件的基础上，我们相信可以对传统软件做以下几点改进。

1. 用神经网络替代开局库

在二.(二)节中已经介绍过了开局库。开局库是存储了人类近千年下棋的经验和基础上总结出来的布局体系的数据库。常见的比如中炮过河车对屏风马，飞相局对过宫炮等开局定式。但这是在人类的基础上推进的，传统软件所起到的只是数据库的作用。也就是说，在不脱离布局体系和棋谱的前提下，可以准确无误且迅速的得到人类认为最好的应招。但是开局库的最大的弊端在于当棋局脱离棋谱的记录范围，那么开局库就无法起到作用，也就是软件就要开始进入中局的思考模式。但是，如果像 Alpha Go 的强策略网络的工作原理一般，那么对于棋谱中还未记录的局面，象棋的策略网络依旧可以依据当前局面的特征给出一些比较好的候选落子。

2. 用蒙特卡洛树搜索替代局面评估函数

局面评估函数在传统软件起到了至关重要的作用。一个好的局面评估函数可以直接导致软件性能的优劣。同时，局面评估函数还可以决定软件的下棋风格，比如激进、保守或是喜好攻杀等。蒙特卡洛树搜索虽然不是直接给出局面的评分，但是它可以进行大量轮次地模拟，根据胜率给当前局面一个客观的评价。虽然不能确定蒙特卡洛树搜索比所有的局面评估函数都好，但是可以确定的是，蒙特卡洛树搜索是一个折中的办法，至于蒙特卡洛树搜索的效果如何，还需后面做实验来进一步确定。

3. 不依赖人类先验知识

传统的象棋软件中，人类的先验知识主要体现在开局库、残局库以及局面评估函数。在前面的第 1 小节中已经提出可以用监督学习得到的强策略网络来替代开局库，同时也可以应用在中局的思考和残局中。基于 AGZ 在 Alpha Go 上面的改进，我们认为同样可以迁移到基于策略网络的象棋软件上来。

参考文献

[1] Gelly S , Silver D . Monte-Carlo tree search and rapid action value estimation in computer Go[J]. Artificial Intelligence, 2011, 175(11):1856-1875.

[2]李炜. 机器学习概述[J]. 科技视界, 2017, 000(012):149.

[3]朱航宇. 基于深度强化学习的 3D 游戏的非完备信息机器博弈研究[D]. 哈尔滨工业大学, 2019.

[4]陈继锴.(2013).中国象棋人机博弈系统的研究与实现.(Doctoral dissertation, 厦门大学).

[5]韩卫, 任建敏, 吴瑞芳. 基于数据库技术的中国象棋软件开局库的设计与实现[J]. 科学技术与工程, 2012, 12(003):555-559.

[6]Chen Jikai. (2013). Research and Implementation of the man-machine Chinese Chess Game System (Doctoral dissertation).

[7] Ikeda K , Shishido T , Viennot S . Machine-Learning of Shape Names for the Game of Go[C]// Advances in Computer Games. Springer International Publishing, 2015.

[8]季辉, 丁泽军, JI,等. 双人博弈问题中的蒙特卡洛树搜索算法的改进[J]. 计算机科学, 2018, 01(v.45):149-152.

[9]Satoru, Fujishige. Theory of submodular programs: A fenchel-type min-max theorem and subgradients of submodular functions[J]. Mathematical Programming, 1984, 29(2):142-155.

[10]王一非. 具有自学习功能的计算机象棋博弈系统的研究与实现[D]. 哈尔滨工程大学.

[11] Schrittwieser J , Antonoglou I , Hubert T , et al. Mastering Atari, Go, chess and shogi by planning with a learned model[J]. Nature.

[12] Maddison C J , Huang A , Sutskever I , et al. Move Evaluation in Go Using Deep Convolutional Neural Networks[J]. Computer Science, 2014.

[13] D Silver, Huang A , Maddison C J , et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature.

[14] 刘知青，李文峰.现代计算机围棋基础[M].北京：北京邮电大学出版社，2011.

[15] Silver D , Hubert T , Schrittwieser J , et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play[J]. Science, 362.