文章编号：2095-2163(2022)11-0138-05 中图分类号：TP391 文献标志码：A

基于 AlphaZero 的不围棋博弈系统研究

高彤彤，丁佳慧，舒文奥，阴思琪
（北京信息科技大学 计算机学院，北京 100101）

摘 要：2017 年，谷歌旗下的 DeepMind 团队公布了 AlphaZero，这是人工智能研究的一个重要里程碑，该算法在不需要专家数据的前提下采用自博弈的方式进行训练，适用于多种棋种。本文以不围棋为载体，将 AlphaZero 算法应用到不围棋博弈系统，较为详细地分析了策略网络、价值网络引导的蒙特卡洛树搜索算法的实现。通过自我对弈学习博弈知识，得到了自我强化，优化了评估函数。

关键词：机器博弈；不围棋；自我对弈；神经网络；蒙特卡洛；AlphaZero；策略网络；价值网络；损失函数

Research on No Go game system based on AlphaZero

GAO Tongtong, DING Jiahui, SHU Wen'ao, YIN Siqi
(Computer School, Beijing Information Science and Technology University, Beijing 100101, China)

【Abstract】 In 2017, Google's DeepMind team announced AlphaZero, which is an important milestone in artificial intelligence research. The algorithm uses self-game training without requiring expert data, which is suitable for a variety of chess games. Taking No Go as a carrier, this paper applies the AlphaZero algorithm to the No Go game system, and analyzes the implementation of the Monte Carlo tree search algorithm guided by the strategy network and the value network in more detail. By learning game knowledge through self-play, self-reinforcing is obtained, and the evaluation function is optimized.

【Key words】 machine game; No Go; self-play; neural network; Monte Carlo; AlphaZero; strategy network; value network; loss function

0 引 言

作为 2012 年出现在大学生博弈大赛[1]中的一种新棋种，不围棋迅速在博弈比赛中流行起来。一般情况下，对围棋的基本理解是消灭敌人取得胜利，而不围棋则与其相反。不围棋的规则不允许有棋子死亡，无论是哪一方自杀、或是吃掉了对方的棋子都会判负。这种规则看似不合理，其实是要求玩家在和平中取胜，最后依然是比较双方占地盘的多少。从某种角度来说，不围棋更符合中华传统文化中“和为贵”的思想。在此背景下，本文提出了基于 AlphaZero 的不围棋博弈系统[2]，通过不断自我学习提高棋力。

1 研究现状

计算机博弈，历来是人工智能的一个重要的研究领域，早期人工智能的研究实践，正是从计算机下棋开始。从 1997 年的“深蓝”，再到 2016 年谷歌公司研发的阿尔法围棋战胜围棋世界冠军，计算机博弈取得了可观的成就。在这期间，蒙特卡洛思想的 UCT(Upper Confidence Bound Apply to Tree)算法曾在围棋人工智能领域主导很长时间。人们围绕蒙特卡洛树搜索(Monte Carlo Tree Search，MCTS)算法始终在做改进和研究，从而不断提高围棋棋力。

不围棋作为研发时间不长的新棋种，相关研究较少。最早对不围棋的研究报道出现在 2011 年，通过对比围棋发现，MCTS、快速评估、UCT 等方法在不围棋中同样有效。文献[3-4]都是利用 MCTS 解决不围棋问题。文献[3]在启动 MCTS 算法时，优先对评分较高的局面进行模拟，通过这种方法来尽可能减少模拟次数。文献[4]为克服 MCTS 计算复杂的问题，利用不围棋博弈本身特点，构建了价值评估模型和函数，递归实现不围棋人工智能。文献[5]提出在对弈过程中进行 UCT 树的重用，可以增加 5%~30%的搜索深度。

本文基于 AlphaZero 对不围棋博弈进行研究，使用深度神经网络和 MCTS 搜索组合形成强化学习框架，不断自我对弈学习博弈知识，优化损失函数，

作者简介：高彤彤(2001-)，女，本科生，主要研究方向：人工智能、计算机博弈；丁佳慧(2001-)，女，本科生，主要研究方向：人工智能、计算机博弈；舒文奥(2001-)，男，本科生，主要研究方向：人工智能、计算机博弈；阴思琪(2001-)，男，本科生，主要研究方向：人工智能、计算机博弈。

收稿日期：2022-08-14

提升不围棋博弈棋力。

2　不围棋及其规则[6]

2.1　棋盘

不围棋使用 9×9 棋盘，分别用字母和数字表示横纵坐标，棋子位置形如 C4、E1。棋盘表示如图 1 所示。

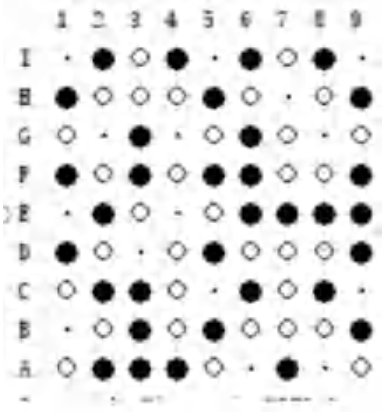

图 1　棋盘表示

Fig. 1　Representation of a chessboard

2.2　棋规

(1)黑棋先行，之后黑白交替落子，落子后棋子不可移动。

(2)对弈的目标不是吃掉对方的棋子，不是像围棋那样围空占领地盘，相反，如果一方落子后吃掉了对方的棋子，则落子一方判负。

(3)如果一方在棋盘上某个交叉点落子后，该棋子将呈现无气状态，相当于自杀，落子自杀一方判负。

(4)不围棋对弈中，禁止空手(pass)，空手一方判负。

(5)如果有时间限制的，超时一方判负。

(6)对弈结果只有胜负，没有和棋。

3　基于 AlphaZero 不围棋博弈系统的设计思想[7]

基于 AlphaZero 不围棋博弈系统主要分为 3 个阶段：自我对战学习阶段，训练神经网络阶段和评估网络阶段。对此拟做研究分述如下。

3.1　自我对战学习阶段

3.1.1　自我对战

自我对战学习阶段主要是蒙特卡洛树搜索进行自我对弈，产生大量棋局样本和胜负关系的过程，由于 AlphaZero 并不使用大师的棋局来学习，而在没有对战数据基础的前提下训练效率不高，因此需要蒙特卡洛树搜索进行自我对弈得到训练数据用于后续神经网络的训练。在自我对战学习阶段，每一步的落子是由 MCTS 搜索来完成的。在 MCTS 搜索的过程中，遇到不在树中的状态，则使用神经网络的结果来更新 MCTS 树结构上保存的内容。而每一次的迭代过程中，在每个棋局当前状态 s 下，每一次移动使用 1 600 次 MCTS 搜索模拟。最终 MCTS 给出最优的落子策略 π，这个策略 π 和神经网络的下一步输出 p 是不一样的，此时的神经网络还没有进行训练。当每一局对战结束后，可以得到在 s 棋局状态下，使用落子策略 π 最终的胜负奖励 z，z 为 1 或者 -1，这取决于游戏的胜负，如此一来，就可以得到非常多的样本 (s,π,z)，这些数据可以用来训练神经网络。

3.1.2　蒙特卡洛树搜索[8]

MCTS 就是用来自对弈生成棋谱的。MCTS 树中保存的数据包括 $N(s,a)$、$W(s,a)$、$Q(s,a)$、$P(s,a)$，分别表示状态 s 下可行动作 a 被选中的次数、总的行动价值、平均行动价值、可行动作 a 的先验概率。搜索过程主要由选择、扩展求值、仿真回溯三部分组成，经过多次模拟后落子。这里对此将给出阐释论述如下。

(1)选择：选择平均行动价值与总行动价值之和 $Q(s,a)+U(s,a)$ 最大的 *action* 搜索分支，$U(s,a)$ 和 $Q(s,a)$ 的计算公式如下所示：

$$U(s,a)=C_{puct}P(s,a)\frac{\sqrt{\sum_b N(s,b)}}{1+N_{(s,a)}} \tag{1}$$

$$Q(s_t,a_t)=W(s_t,a_t)/N(s_t,a_t) \tag{2}$$

其中，s 为搜索树的一个节点代表的棋局状态；a 表示某一个可行的动作；$N(s,a)$ 表示状态 s 下可行动作 a 被选中的次数；$P(s,a)$ 表示状态 s 下的可行动作 a 的先验概率；$Q(s,a)$ 表示状态 s 下可行动作的平均动作价值；$W(s,a)$ 表示状态 s 下可行动作的总动作价值；*puct* 表示一个决定探索程度超参数。

(2)扩展和求值：当棋局还没有结束且当前结点为叶子结点时，就需要进行扩展。扩展的新的结点作为当前结点的子结点，将当前局面输入神经网络得到向量 p 和胜率 v。由此得到的数学公式为：

$$\{N(s_L,a)=0,W(s_L,a)=0,\ Q(s_L,a)=0,P(s_L,a)=p_a\} \tag{3}$$

(3)仿真回溯：如果已被扩展的局面进行选择操作分出了胜负，或者未扩展的局面执行扩展操作，则将胜率回传给上一层，对上一层的值进行更新，被选中的次数加 1，总的行动价值加 v，并重新计算平均行动价值。此时需用到的数学公式分别如以下公

式所示：

$$N(s_t,a_t) = N(s_t,a_t) + 1 \quad (4)$$

$$W(s_t,a_t) = W(s_t,a_t) + v \quad (5)$$

$$Q(s_t,a_t) = W(s_t,a_t)/N(s_t,a_t) \quad (6)$$

其中，s_t 表示搜索树中当次被遍历路径上节点对应的棋局状态；a_t 表示搜索树中当次被遍历路径上节点对应棋局状态下选择的动作；v 表示搜索树中当次被遍历路径上节点的价值，由于搜索树中相邻2层的落子方是不同的，因此相邻2层的节点价值互为相反数。

(4)落子：往棋盘上落一个棋子之前，会进行1 600次模拟，每次模拟都包含上面的 3 个步骤，在此基础上 MCTS 才会做出真正的决策。文中推导得到的公式可表示为：

$$\pi(a \mid s) = \frac{N(s,a)^{1/\tau}}{\sum_b N(s,b)^{1/\tau}} \quad (7)$$

其中，τ 为温度参数，控制探索的程度。τ 越大，不同走法间差异变小，探索比例增大。反之，则更多选择当前最优操作。在零狗中，每一次自我对弈的前 30 步，参数 $\tau = 1$，即早期鼓励探索。游戏剩下的步数，该参数将逐渐降低至 0。如果是比赛，则直接为 0。

3.2 训练神经网络阶段

3.2.1 局面描述

使用 4 层 9 * 9 的二维特征描述当前局面。9 * 9 表示棋盘大小。各层的数学表述具体如下。

(1)第一层：表示当前棋局。

(2)第二层：表示白子当前所占的位置。

(3)第三层：表示黑子当前所占的位置。

(4)第四层：表示哪一方先下棋，如果该下黑子，则矩阵全部等于 1；如果该下白子，则矩阵全部等于 0。

以图 2 局面为例，分析 4 层特征，即如图 3 所示。

图 2 局面描述

Fig. 2 A description of the situation

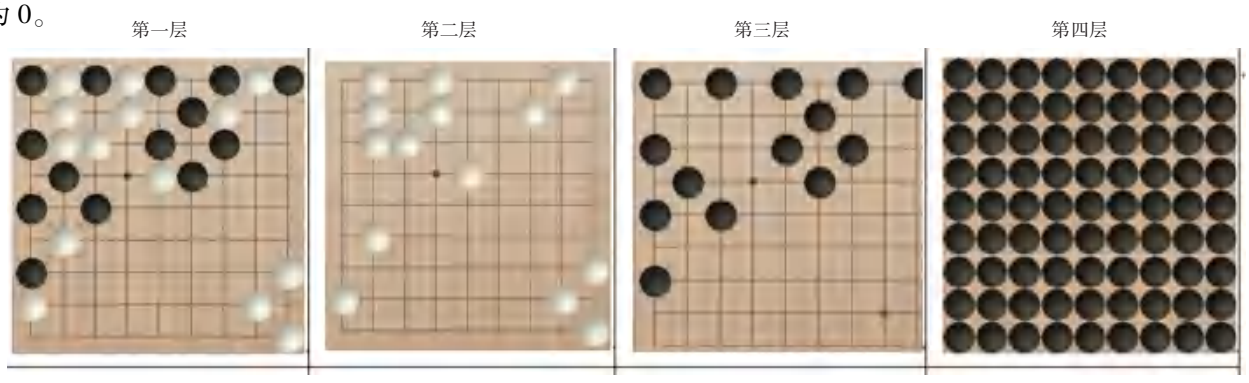

图 3 各层特征详解

Fig. 3 Detailed explanation of the characteristics of each layer

3.2.2 网络结构描述[9]

策略价值网络训练流程如图 4 所示。使用卷积神经网络(Convolutional Neural Network, CNN)进行策略价值网络的训练。CNN 结构比较简单，由公共网络层、行动策略层和状态价值网络层构成。AlphaZero 需要策略网络输出各个动作先验概率以及价值网络评判当前棋局状态的好坏。在 AlphaZero 中策略网络和估值网络共享一部分的卷积层，共享的卷积层为 3 层，分别使用 32、64、128 个 3×3 的 filter，使用 *relu* 激活函数，此后再分成策略 *policy* 和价值 *value* 两个输出。在 *policy* 这一端，先使用 4 个 1×1 的 filter 进行降维，再接一个全连接层、内有 81 个神经元，使用 *softmax* 非线性函数直接输出棋盘上所有可能的走子概率。在 *value* 这一端，先使用 2 个 1×1 的 filter 进行降维，再接一个全连接层、内有 64 个神经元，最后再接一个全连接层，使用 tanh 非线性函数输出局面评分。

该方法既能避免人工设计复杂的静态评估函数，又能较好地解决传统的智能博弈程序中搜索用时巨大、智力水平受程序编写者对博弈技巧理解水平的限制的问题。

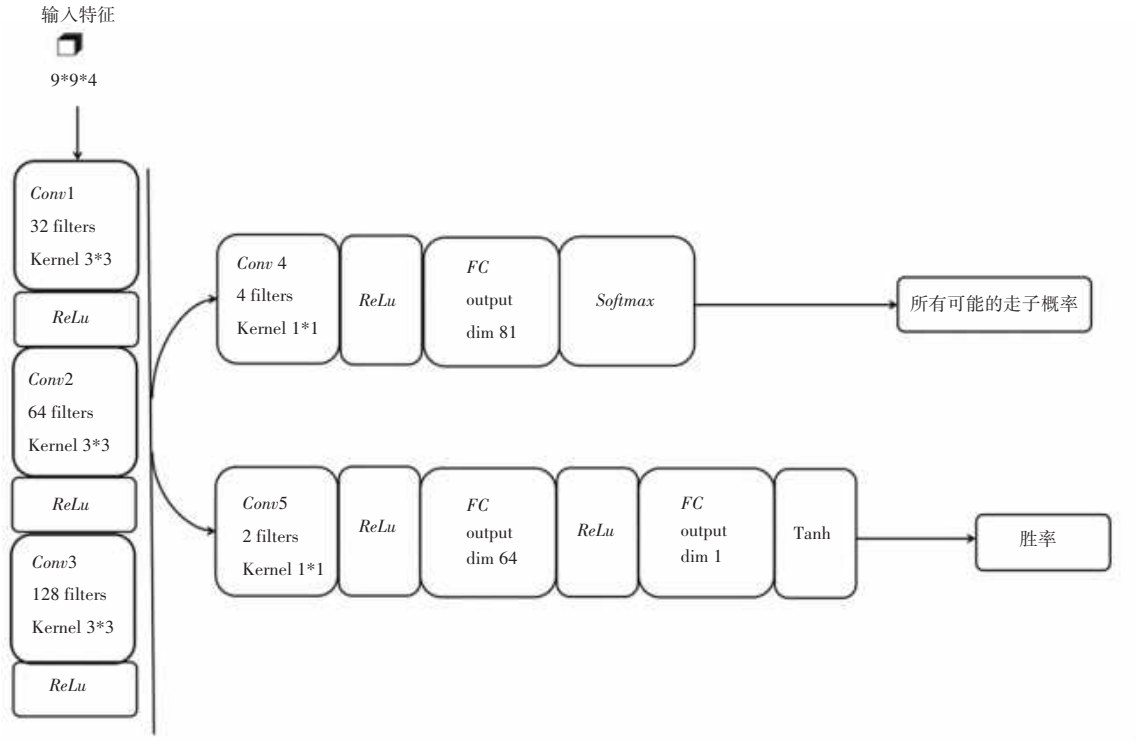

图 4　策略价值网络训练流程图

Fig. 4　Strategy value network training flow chart

3.2.3　最小化损失函数

神经网络的输入为当前的局面 s，输出为下一步行动的概率 p 和对于当前局面胜率的估计 v。在训练神经网络阶段，使用自我对战学习阶段得到的样本集合（s,π,z），训练策略网络和价值网络的模型参数。训练的目标是让策略价值网络输出的当前局面下每一个可行动作的概率 p 更加接近蒙特卡洛树搜索输出的概率 π，让策略价值网络输出的局面评分 v 更加接近真实的对局结果 z。在自我对弈数据集上不断地最小化损失函数，如式（8）所示：

$$L = (z - v)^2 - \pi^{\mathrm{T}}\log(p) + c \| \theta \|^2 \quad (8)$$

其中，z 表示真实的对局结果；v 表示策略价值网络输出的胜率；π 为蒙特卡洛树搜索输出的概率；p 为策略价值网络输出的当前局面下每一个可行动作的概率。式（8）的第三项是用于防止过拟合的正则项。

3.3　评估网络阶段

当神经网络训练完毕后，进行评估阶段，这个阶段主要用于确认神经网络的参数是否得到了优化。这个过程中，自我对战的双方各自使用不同训练程度、不同参数的神经网络指导 MCTS 搜索，并对战若干局，来检验 AlphaZero 在新神经网络参数棋力是否得到了提高。除了神经网络的参数不同外，这个过程和第一阶段的自我对战学习阶段过程是类似的。如果使用新参数后胜率达到 55%，就更新参数，而不再使用旧参数。

4　实验结果与分析

本次研究的不围棋项目结合上文所提到的算法，使用 Python 语言进行编写，在 Windows10 下进行了基于 AlphaZero 的不围棋博弈系统的开发。

实验中，硬件环境设置如下：i7－8750H，主频 2.2 GHz，内存 16 GB，显卡 1060，四核八线程。

表 1 是该算法与 OASE－NoGo 软件的对弈结果及胜率统计，该算法的胜率均在 90% 以上，体现出本文提出算法的可行性和高效性，实现的不围棋博弈有较强的棋力。

表 1　对弈结果统计

Tab. 1　Statistics of game results

测试算法	对手	测试盘数	胜利盘数	胜率/%
本文算法（先手）	OASE－NoGo（后手）	100	97	97
本文算法（后手）	OASE－NoGo（先手）	100	96	96

（下转第 147 页）