中图分类号:O225 文献标识码:A 文章编号:1007-9416(2022)02-0164-03
DOI:10.19695/j.cnki.cn12-1369.2022.02.54

基于CNN模型的亚马逊棋搜索算法设计*

北京信息科技大学计算机学院 李若溪 高铭

亚马逊棋AI搜索算法包括蒙特卡洛搜索算法、改进后的α－β剪枝算法，极大极小搜素算法，经检验发现蒙特卡洛搜索算法取得较好的成果，但原有的搜索算法由于搜索的博弈树层数为固定值，会产生超时或搜索层数过少造成搜索不完全的问题，尝试使用CNN模型优化原有的亚马逊棋搜索算法。设计一个基于CNN模型的优化器，该优化器将当前棋盘的权值作为输入层，进行三层卷积，输出层为当前局面下的最优层数。在同等计算力条件下，采用蒙特卡洛与α－β剪枝算法，令采用CNN模型优化后的搜索算法与固定搜索层数的算法进行对弈，前者的不败率为73.4%。经过实验，发现通过CNN模型自动生成搜索层数的算法对胜率产生了较好的影响。

亚马逊棋发明于1988年，是一种两人棋。一个完整的亚马逊棋局面包括棋盘、棋子、箭，博弈系统包括：落子生成器、落子搜索引擎以及局面得分的评估函数。生成器针对已下的棋局的局面，结合落子的搜索引擎对下一步一定范围内的节点数进行搜索，利用评估函数对每个节点进行评分，返回给落子的生成器进行下棋。综上而言，对胜率产生影响的主要因素是搜索算法与评估函数[1]，亚马逊棋的局面评估函数主要包括以下三个特征的评估值，分别为Territory、Position、Mobility[2]，而搜素算法包括搜索层数与搜索方式，搜索层数的优化为本文的主要研究内容。

最初，大多数强大的亚马逊程序都使用传统的基于极小值[3]的算法，这类程序有Invader[4]和Amazong[5]。这种搜索方法编译简单，容易操作，但是由于亚马逊棋博弈树深度大，遍历的代价极大，即使采取α－β剪枝进行改进，也无法有效解决其耗费大量时间的瓶颈问题。

随后，由于蒙特卡洛的算法在围棋中的优异表现，亚马逊棋也引用了此种技术。蒙特卡洛搜索算法的基本思想是通过迭代的、随机的或者其他更为智能的方法，对每个样本进行随机采样，从而找到最佳走法。但由于蒙特卡洛搜索算法本质是一种随机算法，不可避免的出现过拟合的问题。

而后，人们又对基本的蒙特卡洛算法作出改进，包括向前剪枝和停止蒙特卡洛模拟结束过早的问题。改进步骤包括找到合适的评价功能后逐步拓宽，最终选择正确的评价功能；找寻合适的停止随机模拟的时刻。主要的措施为从UCT树[6]的根开始找到一条路径，通过沿着树向下，选择具有最高的扩张价值的节点来获得叶节点，再从这个叶节点和对应位置运行一个随机模拟装置，将该叶节点的子叶子添加到UCT树中。而是否展开叶节点通常基于该节点被访问的次数，一个节点的展开值等于所有仿真的获胜百分比。

1 基于CNN模型的搜索层数优化器

原有的结合α－β剪枝的蒙特卡洛搜索算法在策略上过于保守，针对于不同的可扩展节点数所遍历的博弈树层数是固定的。上限节点数不但需要根据不同性能的电脑不断调试找到极限值，且由于特定节点数的搜索层数都是固定的为了避免极端情况超时问题的发生，所以需限制搜索层数。在此种情况下的搜索效果会大大降低，以牺牲胜率换取足够时间。针对以上问题，尝试使用CNN模型自动得出当前局面下的最优搜索层数

1.1 CNN输入层

该CNN模型[7]（如图1所示）由输入层，3个卷积

收稿日期:2021－11－17
★基金项目:由北京信息科技大学2021年大学生创新创业训练计划项目资助(5102110805)
作者简介:李若溪,女,河北承德人,本科,研究方向:人工智能、机器博弈。
通讯作者:高铭(1995—),男,河北石家庄人,硕士研究生,实验员,研究方向:机器学习、数据挖掘、虚拟化。

核和输出层构成，神经网络的输入为10×10×1的棋盘数据，输出为一个1×1×1的最优层数值。其中输入的10×10×1的棋盘信息相当于输入一个10维度的方阵，这个矩阵中每个数值的表示方式为：

$$a_{ij}=W_t\times n$$

其中W_t表示当前的时间权值，计算方法为：

$$w_t=1-\frac{t}{900}$$

t表示到目前为止比赛进行的时间，单位为秒；900表示在正规的亚马逊棋博弈中一方的下棋总时长为15分钟，转换为秒数为900s。由于已经对弈所占的时长越长，意味着剩余的时间越短，在较少的时间中为了确保比赛不会出现超时问题，就需要依照时间降低对应的权值。

n表示棋盘在该位置获胜搜索时搜索的层数，采用蒙特卡洛算法进行模拟搜索。蒙特卡洛树搜索是一种基于树的数据结构，可以在搜素空间巨大的情况下仍具有一定效果的启发式算法，其通过选择一个结点后进行扩展，模拟游戏进行直到得到游戏结果，最后通过反向传播返回到达游戏结果搜索的层数n。如果这个位置已经放置了障碍物或存在棋子，则证明这个位置已经搜索完全所以直接令层数n为0。

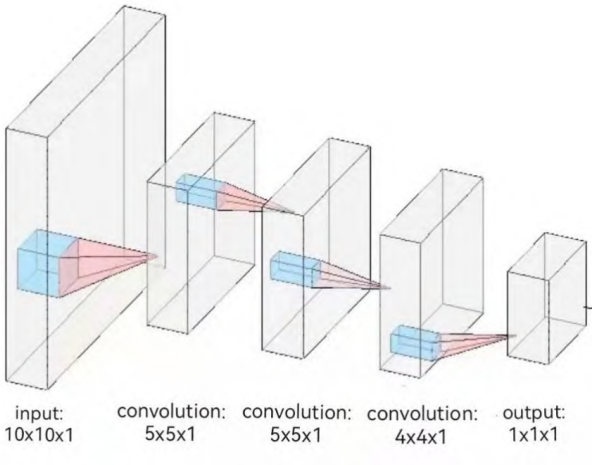

图 1 CNN模型图
Fig.1 CNN model diagram

1.2 CNN卷积层

卷积层是CNN的核心，局部连接和权值共享是卷积层独有特点，该模型卷积层由三个卷积核组成，卷积核大小分别为5×5，5×5以及4×4。为了保证信息的完整性，确保重要信息不被丢失选择Padding为1。卷积核通过在上一层的特征图上不断移动进行卷积运算，从而得到提取出的特征。其中，在同一层的卷积核移动中，权值实现共享。卷积层对应的计算公式如下：

$$x_j^l=f\left(x_i^{l-1}*w_{ij}^l+b_j^l\right)$$

其中x_j^l表示l−1层第i个神经元的输出，x_i^{l-1}表示经过卷积运算后第l层第j个神经元的输出，w_{ij}^l为卷积核，b_j^l为偏置项为激活函数sigmoid。

在训练过程中，不断调整卷积核中权值和偏置，使网络性能达到最优。最后经过三层卷积得到的数值N表示在当前棋盘下的最优搜索层数，由于这个数据是通过卷积神经网络自动生成，能有效避免搜素层数过多出现的超时情况与搜索层数太少而降低胜率的情况。

2 模型训练与测试

由于建立蒙特卡洛搜索树的过程中会产生很多的局面，所以利用程序自对弈的过程，使其不断建立蒙特卡洛搜索树，并产生大量的局面，数据集则来源于整个自对弈过程中所有的局面。采用按比例划分，数据集的总量为81742572条，其中选择85%为训练集，共69,481,186条，剩下的15%为测试集，共12,261,386条。

根据搜索的过程中，扩展节点的层数来划分不同的类别，因为程序的设置，为了保证搜索的精准度和利益最大化，程序设置搜索层数为3以上，3以下暂时不考虑。经过统计可以得出：3到7层的占比分别为：54.48%、26.11%、16.30%、2.91%、0.2%。

其中三层与七层之间的比为272∶1,可以看为数据不平衡，搜索到七层的局面数量太少。而样本不平衡会使得分类模型存在很严重的偏向性[8]，从而使模型产生较严重的误差，所以针对不平衡分类选择以下两种方法解决。

其一为继续增大数据集，因为机器学习是使用现有的数据多整个数据的分布进行估计，因此更多的数据往往能够得到更多的分布信息，以及更好分布估计，但是经过多次再次采样对弈之后发现搜索层数多到第七层的次数更加少，甚至比原本的更加不平衡。

因为第七层的特殊性，对其进行过采样需要耗费大量的人力以及时间，又因为第七层对整体影响较小，所以使用欠采样加人工产生数据样本的方法来解决样本分类的不平衡。使用系统的构造人工数据样本的方法SMOTE[9](Synthetic Minority Over−sampling Technique)。

首先通过删除一半量的三层搜索，以达到趋向平衡的目的，接着进行欠采样。进行完欠采样后，搜索三层和搜索七层的比为136∶1，可见数据集在某种程度上仍然是不平衡的。所以继续使用Smote过采样算法补充第七层的不足。在Python中通过导入Imlbearn库中的过采样方法中的SMOTE接口

数据值无需过大，只需要多层搜索时对模型的影响较小即可最终结果比为10∶1左右。当前数据某种程度上来说是平衡的，所以更新完数据集之后就可以进行下一阶段的模型训练。

最终的数据集3至7层的数量分别为：22267433、21343349、13321549、2378911、2275699。再从61,588,941中分取85%作为训练集，训练数据为52,350,600。

在六核十二线程主频为2.6GHz的电脑下训练，每个Epoch要训练的局面数量是52350600，训练集具有的Batch数目是799次，总的训练时间为27天。

3 实验结果

由于原程序已经拥有着法生成器，搜索器，棋盘生成以及获胜判定等函数，所以为了获取实验数据，只需要使其不断地自对弈[10]，即可获得大量可观测的棋局。

最终，在六核十二线程主频为2.6GHz的电脑下用7241个棋盘的训练集对该卷积神经网络的模型进行训练，通过多次的训练与反复叠加得到卷积核，再选择2000个棋盘的测试集进行测试，总共获胜的棋盘局数为1446，获胜率为72.3%，平局23盘，不败率为73.4%：

经过算法调整和模型训练，使原程序变成并行的算法，预计2000盘中会有10盘因为各种原因导致搜索超时，即便是设置了超时不超时，但依旧留有10场的保守估计，而真实结果则是没有超时现象发生。

在2000个棋盘的测试结果中，模型训练过后的算法相比于原算法来说，胜率达到了72.3%，这说明了训练过后的算法相比于原来的算法能保持着稳定的胜率，而且在平均胜子数目上，经过训练的新算法的平均胜子数为1.65，与预期的2相差不多，有差别的原因是在先后手以及原算法某些特定的落子下，导致训练后的算法某几步思考时间过长，而平均胜子数达到了1.65这说明了算法的提升的，证明了模型的训练是成功的。

4 结论

通过数据显示可以认为，在使用自动生成搜索层数的模型效果优于同等条件下层数固定的模型。但是由于训练集的有限和时间限制可能使训练出的卷积核没有达到较为完美的效果，后续仍存在改进与优化的空间。

这种优化器的思想不仅可以在亚马逊棋中使用，在同样具有盘根错杂的围棋博弈树[11]下也可采用类似的方法，但由于围棋自身的复杂性不建议用蒙特卡洛算法搜索出结果，局面表明我方占有优势即可。

引用

[1] 陈萱华,杨玲.亚马逊棋中评估函数的研究[J].电脑知识与技术,2019,15(8):224-226.

[2] 郭琴琴,李淑琴,包华.亚马逊棋机器博弈系统中评估函数的研究[J].计算机工程与应用,2012,48(34):50-54+87.

[3] 张柳.基于极大极小搜索算法的亚马逊棋博弈系统的研究[D].沈阳:东北大学,2010.

[4] Avetisyan H,Lorentz R J.Selective search in an Amazons program[C].//International Conference on Computers and Games. Springer, Berlin, Heidelberg,2002:123-141.

[5] Lieberum J.An Evaluation Function for the Game of Amazons[J].Theoretical Computer Science,2005,349(22):230-244.

[6] 雷捷维.基于强化学习与博弈树搜索的非完备信息博弈算法的研究与应用[D].南昌:南昌大学,2020.

[7] 石重托,姚伟,黄彦浩,等.基于SE-CNN和仿真数据的电力系统主导失稳模式智能识别[J/OL].中国电机工程学报:1-13[2021-11-08].

[8] 陶立清,黄国满,杨书成,等.一种利用卷积神经网络的干涉图去噪方法[J/OL].武汉大学学报(信息科学版):1-23[2021-11-08].

[9] 张天翼,丁立新.一种基于SMOTE的不平衡数据集重采样方法[J].计算机应用与软件,2021,38(9):273-279.

[10] 裴燕芳.中国象棋自对弈及强化学习系统的设计与实现[D].北京:北京邮电大学,2021.

[11] 谢昌泓.人工智能在围棋研究方面的作用[J].信息系统工程,2019(1):105.