文章编号：2095-2163(2020)03-0142-04 中图分类号：TP18 文献标志码：A

基于强化学习的海克斯棋博弈算法研究与实现

张芃芃，孟 坤，杨震栋
（北京信息科技大学 计算机学院，北京 100192）

摘 要：本文旨在研究如何将强化学习模型合理地应用在海克斯棋博弈算法中，并给出程序实现方案。以蒙特卡洛树搜索生成数据集训练卷积神经网络的方式，使得模型能够在不断自我对弈的过程中，修正自身选择动作的策略，更新模型参数，从而达到提升棋力的目的。实验结果表明，通过强化学习算法能够准确地评估海克斯棋的局面，并有效地选择有利的落子位置，使得海克斯棋博弈系统获得高质量的决策能力。

关键词：强化学习；蒙特卡洛树搜索；海克斯棋；计算机博弈

Research and implementation of Hex game based on reinforcement learning

ZHANG Pengpeng, MENG Kun, YANG Zhendong
(Computer School, Beijing Information Science & Technology University, Beijing 100192, China)

【Abstract】 The purpose of this paper is to study how to apply reinforcement learning model to the algorithm of Hex game reasonably, and give the program implementation scheme. In this way, the convolution neural network can be trained by using the data set generated by the Monte Carlo tree search, so that the model can enhance chess skills by modifying the strategy of its own choice of action and updating the model parameters in the process of continuous self playing. The experimental results show that the reinforcement learning algorithm can accurately evaluate the situation of Hex game, and effectively select a favorable moves, so that Hex game system gains high-quality decision-making ability.

【Key words】 reinforcement learning; Monte-Carlo tree search; Hex game; computer game

0 引 言

随着人工智能的兴起，人们对计算机博弈的研究日趋深入，计算机博弈算法也已越来越多地被应用在各棋种上。海克斯棋是近年来比较流行的计算机博弈棋种之一，现已成为中国大学生计算机博弈大赛的竞技项目[1]。其规则很简单：博弈的双方依次在菱形的棋盘上落子，当任意一方最先将自己的两条边界用己方的棋子连接起来，则该方获胜。

强化学习也称增强学习[2]，是一类在自身智能体不断摸索和尝试的过程中，依靠环境带来的反馈更新自身决策方式的机器学习算法。当智能体模型做出某种动作后产生了有利的状态，则对模型进行奖励，反之则进行惩罚。以此不断迭代，最终使模型具有高质量的决策能力。本文将强化学习模型合理地应用在海克斯棋博弈算法中，使得模型能够通过不断自我对弈，提升棋力。

1 强化学习模型算法设计

海克斯棋是一种完全信息博弈，能够通过模拟大量对局来学习优良的落子选择方法。受 AlphaGo Fan[3] 和 AlphaGo Zero[4] 的算法启发，在模型的核心部分使用价值-策略网络二合一的卷积神经网络，使其输入原始棋盘，输出该局面输赢概率作为价值评估，同时输出每个落子位置获胜的概率分布作为策略评估。首先采取随机落子的方式，使用蒙特卡洛树搜索[5]生成大量对局样本，用监督学习的方式，根据最后的胜负结果、棋面状态以及走法的模拟来训练神经网络。以局面最终输赢训练价值网络的同时，用局面每个落子位置获胜频率分布训练策略网络。然后将这个模型加入到新建立的对手池中，并在从对手池中随机选择一个模型和最新模型进行对弈的过程中，同样通过蒙特卡洛树搜索模拟生成对局，从而产生对弈数据，对神经网络进行训练。并在将训练完成的模型加入到对手池中后，再利用新的模型继续模拟对弈，继而持续进行从对手池中选择模型对弈的过程，以此迭代出一个效果最佳的模型。

1.1 神经网络的设计

使用神经网络的目的在于能够对于给定的海克

基金项目：北京信息科技大学 2019 年促进高校内涵发展-大学生科研训练项目(5101923400)。

作者简介：张芃芃(1998-)，男，本科生，主要研究方向：计算机博弈；孟 坤(1980-)，男，博士后，副教授，主要研究方向：随机模型与网络性能评价、计算智能、网络安全等；杨震栋(1998-)，男，本科生，主要研究方向：计算机博弈。

收稿日期：2019-11-25

斯棋局面做出准确估价的同时，给出最佳的落子位置。因此选择多输出的卷积神经网络，使其同时具有价值输出和策略输出。

对于一个大小为 11×11 的海克斯棋局面来讲可以抽象成一个 11×11×2 的三维张量，作为模型的输入。其中，最低的维度用来区分棋盘上每一个位置的落子情况。考虑到 3×3 大小的卷积核刚好能够覆盖到任何一个方向上的双桥[6]，使得神经网络能够描述和考虑对局面影响较大的特定布局模式。因此在样本输入网络后，首先通过一个含有 32 个 3×3 卷积核的卷积层，再依次通过 2 个含有 64 个 3×3 卷积核的卷积层，然后通过一个含有 96 个 3×3 卷积核的卷积层，接着通过含有 128 个神经元的全连接层，最后通过一个神经元输出价值评估，通过一个含有 121 个神经元的全连接层输出策略评估。另外，所有的卷积层均使用线性整流函数作为激活函数，以保证在训练过程中误差反向传播的高效进行。神经网络结构图如图 1 所示。

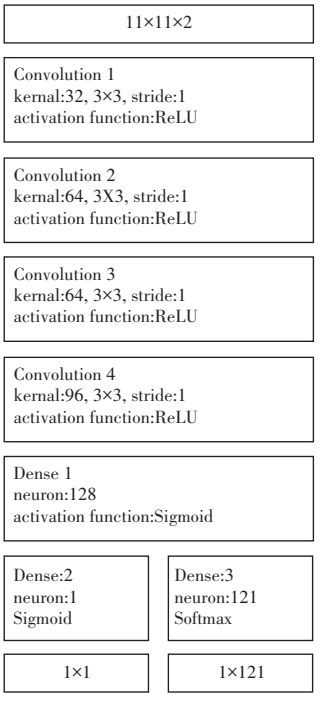

图 1　神经网络

Fig. 1　Neural network

网络的训练过程使用均方差作为价值输出 V 的损失函数 L_V，并将多分类交叉熵作为策略输出 P 的损失函数 L_P，用 RMSprop 优化器在 2 个输出的总损失函数 L 上梯度下降，从而使得模型的参数 θ 收敛，计算公式如下：

$$L_V = (V(S,\theta) - z)^2, \tag{1}$$

$$L_P = -\sum_{a \in A} \pi(S,a)\log_2 P(S,a,\theta), \tag{2}$$

$$L = (1-\alpha)L_V + \alpha L_P + \beta \| \theta \|^2. \tag{3}$$

其中，S 为输入的棋局状态；z 是局面最终输赢；A 是对棋局所有能够触发的动作集合；a 是 A 中的一个元素；π 是用蒙特卡洛树搜索逼近的落子获胜的概率分布；α 为加权系数，用于调节训练时对价值和策略的重视程度；β 是 L2 权重正则化水平的系数，用于防止过拟合。

1.2　数据集的生成

用于训练网络的数据样本含有价值标签和策略标签，其生成过程分为 2 种。第一种在随机对弈的过程中生成，第二种是在模型自我对弈的过程中生成，2 种生成方式在标注方法上完全相同。

对于价值标签，使用局面最终的胜负结果进行标注，若该局面最终胜利，则将其标注为“1”，反之则标注为“0”。

对于策略标签，则通过引入 UCB 公式的蒙特卡洛树搜索，即 UCT 算法，模拟出每一个可落子位置获胜的概率，再转化成分布得到。对此可阐述为：首先对于给定的局面随机选择一个未被探索过的子节点，若所有子节点均被探索过，则根据 UCB 公式选择一个值最大的节点，公式如下：

$$UCB = \frac{W}{N} + C\sqrt{\frac{2\ln N'}{N}}. \tag{4}$$

其中，N 表示子节点访问次数；N' 表示父节点访问次数；W 为子节点获胜次数，C 为平衡系数，以此调节选择当前最优和具有优势潜力节点的偏向程度即平衡探索与利用。

接着，将选择出来的节点进行拓展，然后双方随机落子直到对局结束。若该对局己方胜利，则在沿叶子节点回溯到根节点的过程中，给每个节点访问次数加一的同时，获胜次数也加一。反之，则仅将每个节点的访问次数加一。以此迭代这个过程，在模拟大量对局后，取出根节点，将其每个子节点获胜频率转化成分布后作为根节点局面的策略标注。

1.3　模型的决策算法

模型的决策算法不只依赖于神经网络的输出，而是根据蒙特卡洛树搜索算法，对给定的局面大量模拟，每次选择节点时才使用网络的输出，计算出每个可选择节点的有利程度，在此基础上选择最有利的节点进行拓展。对于每个给定的输入局面 S'，其在动作 a 执行后的局面记为 S，则模型所输出的动作 $\hat{a}$ 满足下式：

$$\hat{a} = \arg\max_{a \in A}(Q(S,\theta) + u(S',a,\theta)), \quad (5)$$

$$Q(S,\theta) = (1-\lambda)V(S,\theta) + \frac{W(S)}{N(S)}, \quad (6)$$

$$u(S',a,\theta) = \eta \frac{P(S',a,\theta)}{1+N(S')}. \quad (7)$$

其中，λ 为混合系数，用来控制网络的价值输出和模拟胜率的选择倾向程度。η 为调节系数，用来控制网络价值输出和策略输出对决策的影响程度。

通过计算 $\hat{a}$ 选择出最优的节点来拓展，不断模拟后，再通过上式求出最佳动作后作为模型的输出。模型决策的数据流图如图 2 所示。

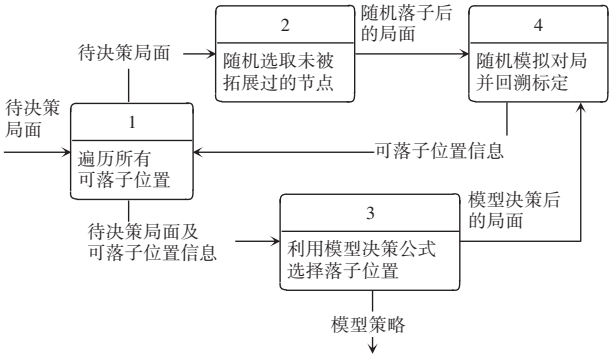

图 2 模型决策的数据流图

Fig. 2 Data flow diagram of model decision

1.4 自我对弈的训练过程

模型决策效果的提升主要依赖于自我对弈生成数据集来训练网络的过程，这需要维护一个对手池，用来存放模型的历代版本。由于不同模型仅在网络参数上存在区别，因此在对手池中存放模型等价于存放网络参数。故首先将监督学习训练完成的网络参数放入对手池中，每次从中随机选取一个网络参数组成模型后作为最新加入对手池的网络参数所组成模型的对手，进行多次双循环对弈，将对弈过程中的局面用其胜负结果标注其价值，并使用蒙特卡洛树搜索获得其策略标签。再用新生成的数据集训练当前的网络，以此获得新一代的网络参数，并放入对手池中，继续自我对弈。模型自我对弈过程的数据流图如图 3 所示。

2 系统界面及算法实现

2.1 海克斯棋界面的实现

根据海克斯棋的规则和特点，使用 PyQt5 实现海克斯棋人机对弈软件，其界面如图 4 所示。该软件具备单步计时、输出棋谱、悔棋等常用功能，能够轻松与博弈比赛的打谱软件对接，具有较强的可扩展性。

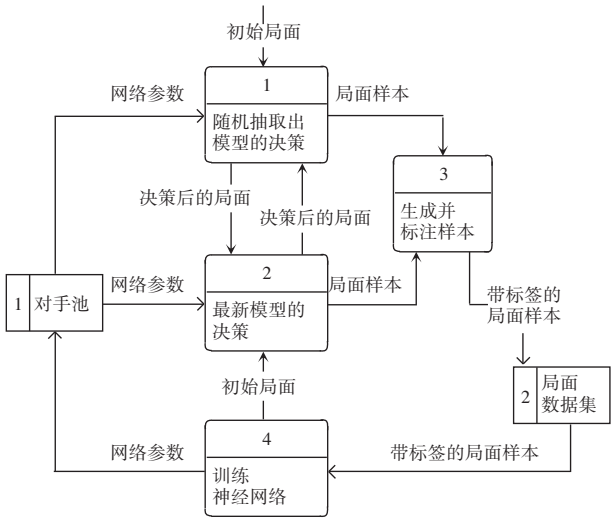

图 3 模型自我对弈过程的数据流图

Fig. 3 Data flow diagram of model self playing process

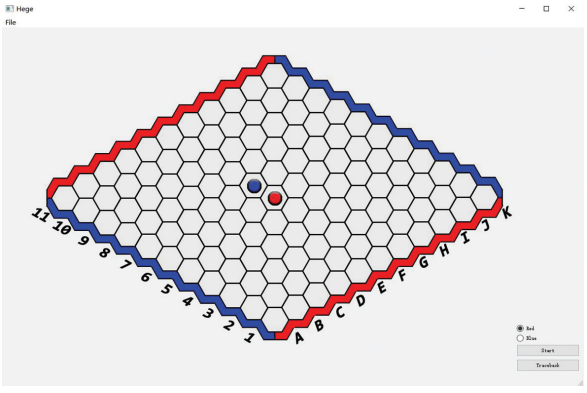

图 4 海克斯棋界面

Fig. 4 GUI of Hex game

2.2 卷积神经网络的实现

考虑到生成模型的过程中需要对卷积神经网络进行训练，因此选择 TensorFlow 框架进行编程，并使用 CUDA 和 cuDNN 实现 GPU 加速。神经网络的 TensorFlow 实现如图 5 所示。

2.3 蒙特卡洛树搜索的实现

在生成数据集和模型决策时均要使用蒙特卡洛树搜索，二者只在每次选择节点的方式上存在区别，因此蒙特卡洛树搜索算法的实现仅需完成树搜索的框架，留出节点评估更改接口即可。

3 实验结果与分析

模型的效果需要从神经网络的训练效果和模型的对弈胜率两个角度衡量。本实验在 Windows 系统下进行，开发环境为 Python 3.6。

神经网络的训练方面，统计监督学习过程中每轮训练时在验证集下的损失函数和轮数的关系，并绘制 loss 曲线，如图 6 所示。

Layer(type)	Output Shape	Param #	Connected to
chess_table_input (InputLayer)	(None, 11, 11,2)	0	
conv2d(Conv2D)	(None, 9, 9,32)	608	chess_table_input [0][0]
conv2d_1(Conv2D)	(None, 7, 7,64)	18496	conv2d[0][0]
conv2d_2(Conv2D)	(None, 5, 5,64)	36928	conv2d_1[0][0]
conv2d_3(Conv2D)	(None, 3, 3,96)	55392	conv2d_2[0][0]
flatten(Flatten)	(None, 864)	0	conv2d_3[0][0]
dense(Dense)	(None, 128)	110720	flatten[0][0]
value_output_layer (Dense)	(None, 1)	129	dense[0][0]
policy_output_layer (Dense)	(None, 121)	15609	dense[0][0]
Total param:237,882 Trainable parames:237,882 Non-trainable params:0			

图 5　神经网络的 TensorFlow 实现

Fig. 5　Implementation of neural network by Tensorflow

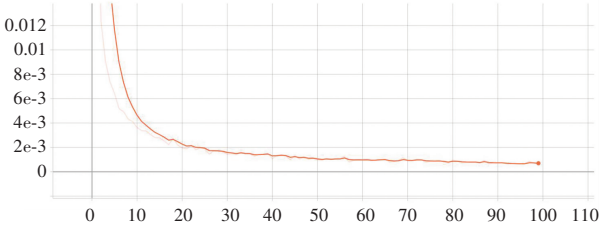

图 6　loss 曲线

Fig. 6　loss curve

由图 6 可知，神经网络的损失函数在训练到第 100 轮左右时已基本收敛，其值大概在 6.59×10^{-4} 左右。经过分析后发现该训练过程较为有效。

模型的对弈方面，使用经过 10 次以内（含第 10 次）自我对弈训练过的网络参数分别组成共计 11 个强化学习模型，与仅使用 UCT 算法下的模型分先后手，分别对局 10 次。即每个强化学习模型与 UCT 算法模型进行 20 次对局，统计胜负情况，得到强化学习模型胜率与其网络训练次数的关系，如图 7 所示。

由图7可以看出，强化学习模型在自我对弈生成数据集并训练的过程中，与UCT算法模型对弈的胜率具有显著提升，并在第 10 轮左右基本收敛。而且最后一次训练出来的网络参数所组成的模型在和 UCT 算法模型对弈时，已经能够赢得 80%的对局。

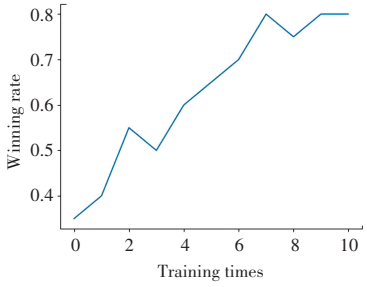

图 7　胜率与训练次数的关系

Fig. 7　The relationship between winning rate and training times

综上可知，神经网络的训练效果和模型的对弈胜率均说明通过强化学习算法能够准确地评估海克斯棋的局面，并有效地选择有利的落子位置，使得海克斯棋博弈系统具有高质量的决策能力。

4　结束语

本文针对海克斯棋的强化学习算法展开研究，给出强化学习模型所需要的各个组成部分的算法流程和计算公式，包括神经网络结构，蒙特卡洛树搜索和模型决策算法等。最后通过实验证明强化学习模型可以通过不断自我对弈，提升海克斯棋博弈系统的智能程度。

参考文献

[1] 中国人工智能学会机器博弈专业委员会. 2018 中国计算机博弈大赛项目规则与棋牌谱规范[Z]. 沈阳：中国人工智能学会机器博弈专业委员会，2018.

[2] 周志华. 机器学习[J]. 航空港，2018(2)：94.

[3] SILVER D，HUANG A，MADDISON C J，et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature，2016，529(7587)：484-489.

[4] SILVER D，SCHRITTWIESER J，SIMONYAN K，et al. Mastering the game of Go without human knowledge[J]. Nature，2017，550(7676)：354.

[5] ARNESON B，HAYWARD R B ，HENDERSON P. Monte Carlo tree search in Hex [J]. IEEE Transactions on Computational Intelligence and AI in Games，2010，2(4)：251.

[6] ANSELEVICH V. The game of Hex：An automatic theorem proving approach to game programming [C]//AAAI/IAAI. Austin，TA，USA：dblp，2000：189.

（上接第 141 页）

[10] 张可，杨恒占，钱富才. 基于动态故障树的卫星电源系统可靠性分析[J]. 计算机与数字工程，2016，44(3)：400.

[11] 杨超. 动态故障树及其在民机复杂系统安全性评估中的应用研究[D]. 南京：南京航空航天大学，2011.

[12] 吴天魁，王波，周晓辉. 基于模糊故障树的鳞板输送机可靠性分析[J]. 科技通报，2014，30(9)：157.

[13] 陶勇剑 ，董德存，任鹏. 故障树分析的二元决策图方法[J]. 铁路计算机应用，2009，18(9)：4.

[14] 李希灿. 模糊数学方法及应用[M]. 北京：化学工业出版社，2017.

[15] GE Daochuan，LIN Meng，YANG Yanhua，et al. Reliability analysis of complex dynamic fault trees based on an adapted K.D. Heidtmann algorithm [J]. Proceedings of the Institution of Mechanical Engineers，Part O：Journal of Risk & Reliability，2015，229(6)：576.