**Gobang Based on Monte Carlo Tree Search**

陈戌 1612839

1. **实验目的**

围棋起源于3000多年前的中国，赢得这场棋盘游戏需要玩家有多层次的战略思维。围棋的规则是两名对弈者分别执白子和黑子，轮流将它们放置在19✖️19的棋盘上，去包围和捕获对手的棋子或战略性地创造领土空间，一旦所有可能的动作都进行了，棋盘上的棋子和空点都会被计算在内，点数高者获胜。虽然围棋规则看似简单，但它的可能步数超过了已知宇宙中的所有原子数量，这使得围棋游戏比国际象棋更复杂，它也因此被称为**人工智能最具挑战性的经典游戏**。

AlphaGo是第一个击败专业人类围棋选手的计算机程序、第一个击败围棋世界冠军的计算机程序，可以说是历史上最强的围棋玩家。其算法核心在于采用了高级搜索方法和两个神经网络（策略网络与价值网络）去达到强化学习的效果，其中的高级搜索方法就是最通用，功能最强大且使用最广泛的树搜索算法之一：**蒙塔卡洛树**（Monte Carlo Tree Search）**搜索**。

基于MCTS的广泛适用性，本实验旨在以蒙特卡洛树搜索算法为核心去实现一种世界上广为流传的连棋游戏：**五子棋**。五子棋是一种两人对弈的纯策略型棋类游戏，通常双方分别使用黑白两色的棋子，轮流下在棋盘直线与横线的交叉点上，先在横线、直线或斜对角线上形成5子连线者获胜。原始规则的15✖️15五子棋已被证明先手必胜，前提是不引入禁手，由此可以看出五子棋在搜索空间上比围棋小不少，所以在普通民用电脑上使用MCTS实现一个可与人类对决的五子棋程序是值得一试并且有意义的。为带来良好用户体验，该程序应同时拥有**交互性良好的UI界面**与**控制台输出**。

1. **实验原理**

蒙特卡洛树搜索是一种对抗搜索，是利用（Exploitation）与探索（Exploration）在游戏博弈树上的有机协调，谈到他就必须谈起单一状态的蒙特卡洛规划：**多臂赌博机**（Multi-armed Bandits）。

在摇臂赌博机问题中，每次以随机采样形式采取一种行动𝑎, 好比随机拉动第𝑘个赌博机的臂膀，得到𝑅(𝑎𝑘) 的回报，问题是：下一次需要拉动那个赌博机的臂膀，才能获得最大回报呢？这种序列决策问题需要在利用（Exploitation）和探索（Exploration） 之间保持平衡，**利用保证在过去决策中得到最佳回报**，意即过去抽到奖的老虎机可能会更大概率地带来好的收益；**探索寄希望在未来能够得到更大回报**，也就是说从其他老虎机碰碰运气可能会有意想不到的收获。

如果有𝑘个赌博机，这𝑘个赌博机产生的操作序列为𝑋𝑖,1, 𝑋𝑖,2, … *(i* = 1*,* 2*…, k)*，在时刻𝑡 = 1, 2…，选择第𝐼𝑡个赌博机后，可得到奖赏𝑋𝐼𝑡,𝑡，则在𝑛次操作𝐼1, …, 𝐼𝑛后，可如下定义悔值函数：

𝑅𝑛 =

悔值函数表示了如下意思：在第𝑡次对赌博机操作时，假设知道哪个赌博机能够给出最大奖赏（虽然在现实生活中这是不存在的），则将得到的最大奖赏减去实际操作第𝐼𝑡个赌博机所得到的奖赏。将𝑛次操作的差值累加起来，就是悔值函数的结果。很显然，一个良好的多臂赌博机操作的策略是在不同人进行了多次玩法后，能够让悔值函数的方差最小。

在多臂赌博机的研究过程中，上限置信区间（Upper Confidence Bound, UCB）成为一种较为成功的策略学习方法，因为其在探索-利用（exploration-exploitation）之间取得平衡。我们使𝑋𝑖,𝑇\_𝑖来记录第𝑖个赌博机在过去 𝑡 − 1时刻内的平均奖赏，则在第𝑡时刻，选择使如下具有最佳上限置区间的赌博机：

𝑅𝑛 =

其中𝑐𝑡,𝑠取值定义如下：

𝑐𝑡,𝑠 =

也就是说，在第𝑡时刻，UCB算法一般会选择具有如下最大值的第𝑗个赌博机：

Kocsis和Szepesvari在2006年提出将上限置信区间算法UCB应用于游戏树的搜索方法，这就诞生了蒙特卡洛树搜索，其包括了四个步骤：选择(selection)，扩展(expansion)，模拟(simulation)，反向传播(back-propagation)：

* 选择

从根节点 R 开始，向下递归选择子节点，直至选择一个叶子节点L，具体来说，通常用UCB（Upper Confidence Bound，上限置信区间）选择最具“潜力”的后续节点

* **扩展**

如果 L 不是一个终止节点（即游戏未结束），则随机创建其后的一个未被访问节点，选择该节点作为后续子节点C

* **模拟**

从节点 C出发，对游戏进行模拟，直到游戏结束

* **反向传播**

用模拟所得结果来回溯更新导致这个结果的每个节点中获胜次数和访问次数。

从中我们可以看到MCTS有两种策略学习机制，**搜索树策略**与**模拟策略**。前者需要在利用和探索之间保持平衡；而后者从非叶子结点出发模拟游戏，得到游戏仿真结果。

图片包含 屏幕截图

描述已自动生成

所以蒙特卡洛树搜索是基于采样来得到结果，而非穷尽式枚举。

1. 编程环境

本游戏全部基于**Python 3.7.4**完成，开发及测试环节都基于MacOS Catalina系统10.15.1，以下是项目详情以及*Pylint*静态分析代码的报告

|  |  |  |
| --- | --- | --- |
| ***directory*** | ***requirements.txt*** | ***External dependencies*** |
| *├── Board.py*  *├── Node.py*  *├── README.md*  *├── gobang.py*  *├── mcts.py*  *└── requirements.txt* | *numpy==1.17.3* | *Board (gobang,mcts)*  *Node (mcts)*  *mcts (gobang)* |

***Statistics by types***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **type** | **number** | **old number** | **difference** | **%documented** | **%badname** |
| *module* | *4* | *NC* | *NC* | *0.00* | *50.00* |
| *class* | *3* | *NC* | *NC* | *33.33* | *0.00* |
| *method* | *22* | *NC* | *NC* | *54.55* | *0.00* |
| *function* | *6* | *NC* | *NC* | *0.00* | *0.00* |

***Raw metrics***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **type** | **number** | **%** | **previous** | **difference** |
| *code* | *322* | *77.78* | *NC* | *NC* |
| *docstring* | *31* | *7.49* | *NC* | *NC* |
| *comment* | *7* | *1.69* | *NC* | *NC* |
| *empty* | *54* | *13.04* | *NC* | *NC* |

***Duplication***

|  |  |  |  |
| --- | --- | --- | --- |
|  | **now** | **previous** | **difference** |
| *nb duplicated lines* | *0* | *NC* | *NC* |
| *percent duplicated lines* | *0.000* | *NC* | *NC* |

***Messages by category***

|  |  |  |  |
| --- | --- | --- | --- |
| **type** | **number** | **previous** | **difference** |
| *convention* | *73* | *NC* | *NC* |
| *refactor* | *6* | *NC* | *NC* |
| *warning* | *16* | *NC* | *NC* |
| *error* | *7* | *NC* | *NC* |

***% errors / warnings by module***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **module** | **error** | **warning** | **refactor** | **convention** |
| *gobang* | *42.86* | *31.25* | *66.67* | *46.58* |
| *mcts* | *28.57* | *56.25* | *16.67* | *24.66* |
| *Board* | *14.29* | *6.25* | *16.67* | *21.92* |
| *Node* | *14.29* | *6.25* | *0.00* | *6.85* |

1. 测试方法
2. 打开终端/命令行程序，复制输入以下内容：

***git clone*** [***https://github.com/cnarutox/Gobang***](https://github.com/cnarutox/Gobang)***; cd Gobang***

1. 执行***pip3 install -r requirements.tx***t去安装依赖库
2. 执行***python3 gobang.py***，游戏界面弹出，点击左上方的Start开始游戏

简洁复制版：

***git clone https://github.com/cnarutox/Gobang ; cd Gobang ; pip3 install -r requirements.txt ; python3 gobang.py***

注：如果执行后控制台输出以下warning

*EPRECATION WARNING: The system version of Tk is deprecated and may be removed in a future release. Please don't rely on it. Set TK\_SILENCE\_DEPRECATION=1 to suppress this warning*

说明系统安装的***Tcl/Tk***与*Python*[版本不契合](https://www.python.org/download/mac/tcltk/#built-in-8-6-8)，需要根据自己的*Python*版本去安装指定版本的[***Tcl/Tk***](https://www.activestate.com/products/tcl/downloads/)，否则UI界面会出现色差与位移。

1. 结果展示

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

1. 实验源码（点击对应py文字可去查看源码）