基于 AI 技术的五子棋状态评估及难度调整

小组介绍：

组长：李岩（基础评分系统的代码编写、论文）

组员：黄少霖（整体框架和 AI 部分的代码编写及后期效率优化、论文）

朱丹（视频报告、论文）

杨泽洋（算法设计、论文）

王奕鸣（思维拓展与延伸、论文）

说明：本次项目的推进建立在多次讨论的基础上，组内每位成员都对项目做出重要贡献。

一、项目介绍

1. 选题背景：

通过 CS103 人工智能导论的学习，小组加深了对 AI 背后逻辑的了解，感觉到与 AI 技术的距离更近了一步。同时，小组非常希望能在完成项目的过程中，进一步对 AI 相关知识进行讨论并尝试创作 AI 作品。由于对 AlphaGo 项目的兴趣，小组决定研究并创造一个棋类 AI。通过阅读文献，小组发现：在目前主流棋类游戏中，五子棋 AI 的开发相对欠缺，而五子棋简洁的规则和由简生繁的棋局特点又很适用小组练习 AI 算法的应用并加深理解。由此，小组将五子棋定为项目方向。经过阅读文献与研讨，小组认为我们应着重关注 AI 算法相关的部分，主要有两处：一是运用 AI 算法对初步形成的五子棋棋局评分系统进行修正和优化，二是运用 AI 技术实现根据对棋手实力的判断动态调整 AI 难度（这一方面在目前已有的棋类软件中鲜少涉及，小组认为这一功能的实现不仅能作为 AI 应用的练习，而且就 AI 产品设计的角度而言也很有意义）。在明确了重点研究部分后，小组将项目题目定为“基于 AI 技术的五子棋状态评估及难度调整”

2. 项目功能：

本项目最终成果为一款五子棋对战软件。主要功能有以下三点：

1)玩家与玩家对战：供两个玩家进行对战的平台

2)玩家与 AI 对战：共两个选项。默认选项是与可动态添加难度的 AI 进行对局，对局期间 AI 会根据对玩家能力强弱程度的判断来动态调整自身的水平。另一选项是与基于当前 AI 训练方法和训练所用对局所形成的最强 AI 进行对局，在对局中，AI 会尽可能下出最优解来战胜玩家。

3)AI 自我对弈：此功能用于提升 AI 的最高水平，同时也能使 AI 在根据玩家水平动态调整的过程中更加精准。通过持续对 AI 投入大量优质对局，让 AI 自己进行模拟对局，能够不断对 AI 的水平进行提升，使其趋向于理论最优状态。

3. 主要算法：监督学习、模拟退火算法、正态分布概率模拟

二、代码组成：

本项目的代码部分主要由以下部分组成：

1. Gomoku.py: 实现五子棋的游戏逻辑及 UI 界面
2. GomokuAI.py: 实现了一个初级五子棋 AI，能根据局面进行落子决策及玩

家水平分评估.

函数组成：

1) Search(): 搜索当前局面下所有可能的落子位置，对落子后的新局面进行评分，将（位置，对应局面评分）以二元组的形式存入可选行为列表中

2) Play(): 获取可选行为列表，按分数排序后根据玩家水平分选择最合适的落子位置

3) UpdateBoard(): 与游戏进程同步更新当前棋盘局面

4) UpdateLevel(): 根据玩家的最新决策与搜索到的所有可行决策更新玩家水平分

5) Save(): 棋局结束时将当前评估出的玩家水平分保存至配置文件

3. Assessor.py: 实现了对于一个棋局局面进行评分的相关功能

函数组成：

1) Get_lines(): 将落子位置沿横竖斜四个方向的相邻位置记录分别为四个由 0，1，2 三种字符组成的字符串

2) Count_type_change(): 统计落子前后各类棋型数量的变化

3) CalcScore(): 根据各类棋型数量计算棋局分数

4) Assess_Game(game): （用于分数训练）传入完整对局，计算全局每一步棋后的局面分数

另于该文件内实现了一个 AC 自动机类，用于加速棋型数量统计

4. Game.py: 读入棋谱文本数据并转换为完整对局
5. ScoreTraining.py: 用于预训练 Assessor 中所使用的各类棋型赋分

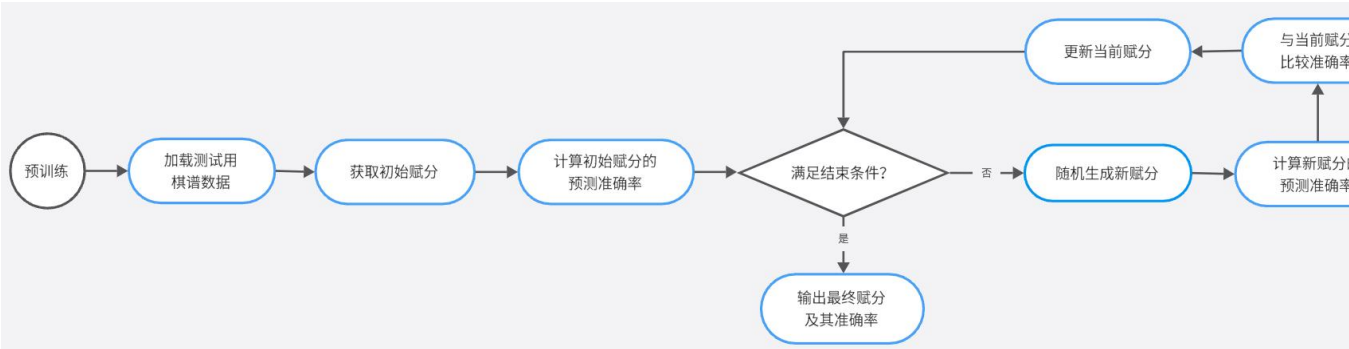
函数组成：

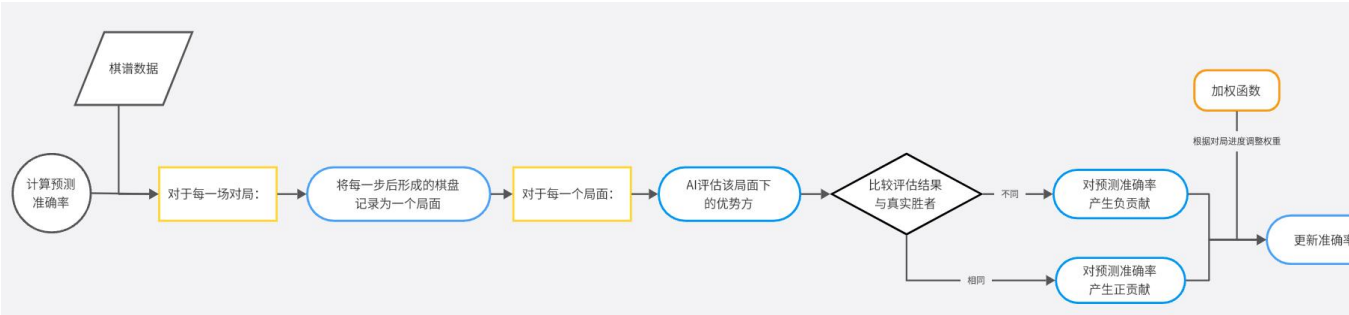
1) Main(): 通过模拟退火算法获得一个较优的评分

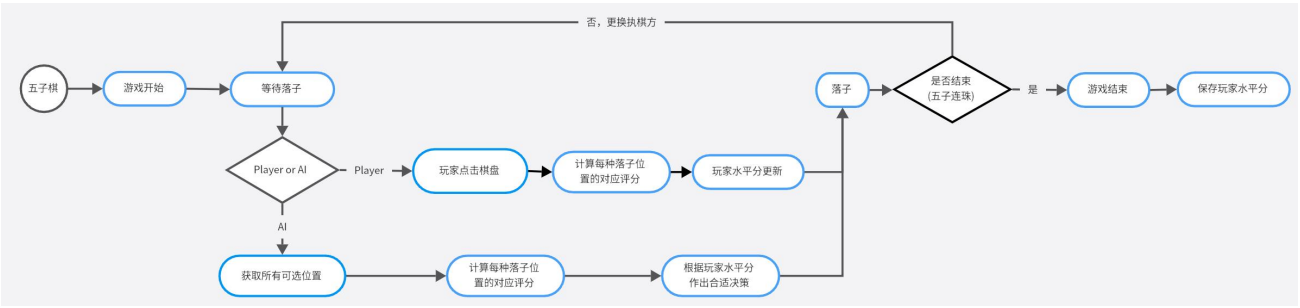
2) Calc(new_score): 使用新评分在棋谱集上进行评分，计算其预测准确率

3) Weigh(progress_rate): 根据游戏进程百分比 分配权重（越接近一局结束权重越大）

Workflow:







三、运行方法：

运行 python Gomoku.py 以进行 PVP 五子棋对局

运行 python Gomoku.py -AI 以进行与带有动态难度调整的 AI 对弈的五子棋对局

运行 python Gomoku.py -AI -n 以进行与无难度调整(固定最高难度)AI 对弈的五子棋对局

修改 configuration.txt 的参数以强制改变 AI 难度

四、效果展示：

1.AI 形成的合理性：

初始阶段，我们基于从公开棋局数据库中获取的数据集来设定评分算法的基准值，但由于这组数据只是单纯地按照倍数增大，其无法给出接近准确分数的答案，导致我们的 AI 常常出现低级的错误判断。对此，我们采用了退火算法作为核心优化技术，以提高系统的预测准确率。通过连续的迭代和优化，我们将系统的准确率从初始的 48%提高至 57%。在五子棋领域，由于棋局的多变性和复杂性，尤其是考虑到禁手规则的影响，每一步棋的下法与棋局的最终结果之间并非总是直接相关，即便是最完美的赋分数据，也难以达到 60%以上的准确率。因此，我们系统所取得的 9%的准确率，几乎达到了对实际棋局进行高度精确评分的水平，我们的 AI 在实际对战中的水平也随之大幅提升。

2.成品 AI 的能力：

它能够根据对局双方的棋力水平动态调整自己的下棋策略和水平。这种智能调整机制旨在保持棋局双方的分数接近，以此提供一个更加平衡且具有挑战性的游戏体验。具体来说，AI 会实时分析棋局，评估双方的策略和棋力，并据此调整自己的行棋策略。核心目标是提供一个既具挑战性又不至于让对

手感到挫败的游戏环境。通过保持双方分数的接近，AI 确保游戏对所有水平的玩家都是富有趣味且旗鼓相当的。

3.对弈情况：

在正常对局情况下，AI 基本符合预期，既不会出现明显的低级错误，也不会下出妙手将我们一下子击败。

此外，为了验证其动态调整功能，我们还与其进行了数场乱下的棋局，结果显示，即便在这种情况下，AI 也能很快反应过来，判断出下棋者的水平异常，并下出与之水平相当的棋子。

五、算法学习

在本次项目中，我们通过阅读文献的方式重点学习了基于值的强化学习以及模拟退火算法的理论及实际应用，使我们的代码有着科学的理论支持，同时有利于此项目后续的进一步发展，文献见文末。

六、参考文献：

[1] 张强 et al. “基于 Q-network 强化学习的超视距空战机动决策.” 空军工程大学学报（自然科学版） 19.6 (2018): 8‐14. Web.
[2] 罗锦彬. “基于人工智能算法的无人机自动控制研究.” 蚌埠学院学报 11.5 (2022): 29‐33. Web.
[3] 程传斌 et al. “改进的动态 A-Q-Learning 算法及其在无人机航迹规划中的应用.” 现代信息科技 5.9 (2021): 1‐9. Web.
[4] 邹科 et al. 基于模拟退火的动态缆线型优化. N.p., 2022. Print.
[5] 陈道蓄. “模拟退火算法.” 中国信息技术教育 3 (2021): 19–23. Print.
[6] 刘娜 et al. “基于模拟退火算法的 Halbach 直线发电机优化设计.” Dian gong ji shu xue bao 36.6 (2021): 1210–1218. Web.
[7] 葛艳, 王爱民, and 叶介然. “基于遗传退火算法的质检扰动应对方法.” 计算机集成制造系统 27.11 (2021): 3159–3171. Web.
[8] 徐小琴 et al. “基于改进遗传退火算法的输配电网协调规划方法.” 电力系统保护与控制 49.15 (2021): 124–131. Web.