实验报告：基于强化学习的迷宫最短路径求解

摘要：

本实验旨在利用强化学习中的 Q-learning 算法求解迷宫的最短路径问题。通过对迷宫图像的预处理，将其转化为适合算法处理的矩阵形式，并设计了一套奖励机制以引导智能体高效探索路径。此外，实验还实现了一个交互式界面，允许用户通过鼠标指定终点，实时展示路径或提示无法到达。

1. 模型描述：

本模型的实现主要分为三个步骤：

1.1 迷宫图的预处理： 首先，对输入的迷宫图像进行预处理，以便于后续的算法处理。如 maze.jpg 所示，迷宫图的边界需要作出修剪，同时需要对迷宫图做作灰度化处理和二值化处理，以便将迷宫转化为 0/1 矩阵，同时，还需要对墙壁和通路的黑白马赛克块合并成单个像素块，降低迷宫的复杂度。图1显示了预处理后的迷宫图。

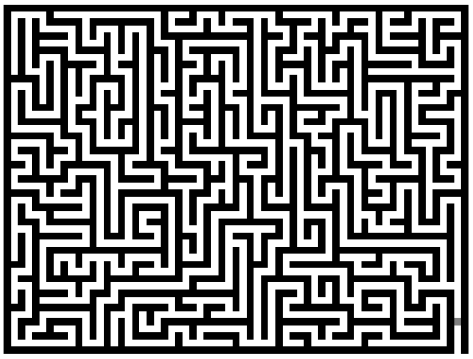

图 1: 预处理后的迷宫图

1.2 Q-learning 算法求解: 在预处理后的迷宫矩阵上应用 Q-learning 算法以求解最短路径。具体实现包括：通过将迷宫的二维位置 (x, y) 转化为一维状态 state = x * columns + y, 从而将 Q 表从三维降至二维，减少模型的复杂度，提高计算效率。同时设计有效的奖励函数以引导智能体寻找最优路径，具体实现请见后文。

1.3 交互式场景创建： 为了增强用户体验，我们构建了一个交互式界面，其中，初始起点由程序预设，终点可由用户通过鼠标点击任意通路位置设定，如点击墙壁位置，则会提醒” 点击位置是墙，请选择一个通路位置”，如点击迷宫界外位置，则会提醒” 点击位置超出迷宫范围，请重新选择”。根据 Q-learning 算法的结果，实时绘制最短路径并给出最短路径坐标，若路径不可达，系统将会提示用户” 无法到达终点”。

2. 训练方式：

2.1 Q-learning 算法：采用 Q-learning 算法进行训练，核心更新公式如下：

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

其中，$Q(s,a)$ 为状态 s 下动作 a 的 Q 值，α 为学习率，γ 为折扣因子，r 为即时奖励，s' 为执行动作 a 后的新状态，a' 为新状态下的所有可能动作。

2.2 奖励函数设计：为了有效引导智能体，设计了以下奖励机制：

$$\begin{cases} \text{达到终点：} r = +1000 \\ \text{超出边界：} r = -1000 \\ \text{移动至通路：} r = -1 \\ \text{撞墙：} r = -\frac{n_{states}}{2} \end{cases}$$

其中，$n_{states} = rows \times columns$, $\frac{n_{states}}{2}$ 是我们对通路像素个数的粗略估计，确保其撞墙的惩罚足够大，促使其通过绕路到达终点而不是穿过墙壁，同时也不至于将其设置的过于大，以免影响模型的收敛性。

2.3 Q 表的优化：传统 Q 表为三维矩阵 (position[0], position[1], len_actions), 本实验将其转化为二维矩阵 (state, len_actions), 其中 state = position[0] * columns + position[1]. 此优化不仅减少了模型的复杂度，还显著提高了训练和推理的运行速度。

2.4 参数设置：

$$\begin{cases} \text{训练轮数} : 20000 \\ \text{学习率}(\alpha) : 0.1 \\ \text{折扣因子}(\gamma) : 0.98 \\ \text{探索率}(\epsilon_{initial}) : 1 \end{cases}$$

其中，ϵ 的初始值为 1, 采用衰减率 $decay\_rate = 0.995$, 并设置最小探索率 $min\_epsilon = 0.01$, 以优化算法性能。

3. 测试实验设置及其结果

我们选取多个不同的终点进行测试，对于题中给定的起点，所有通路均可到达，可通过手动添加障碍测试不可到达的情况，以下为部分测试示例：

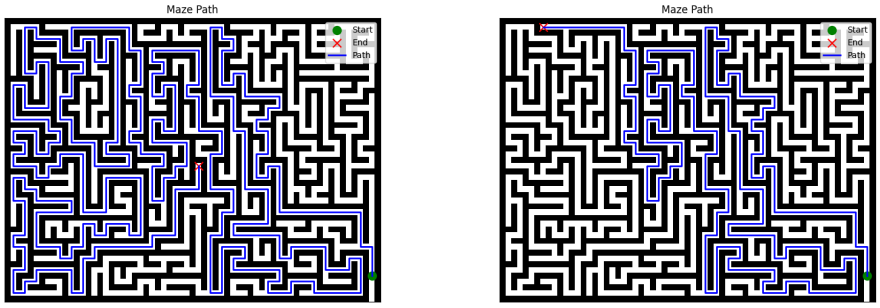

图 2: 终点（左）: (25, 33)，　终点（右）: (1，7)

另外，为了测试终点无法到达的情况，我们在起点下方设置了一个墙壁，并且将终点设置为墙壁下方，最终程序输出："无法到达终点。"

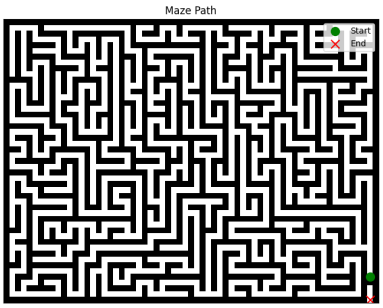

图 3: 终点 (48, 63)

4. 相关讨论：

本实验在传统 Q-learning 算法的基础上进行了多项优化：

状态表示优化：将二维位置转化为一维状态，有效减少了 Q 表的维度，从而降低了内存消耗和计算复杂度，提升了训练和推理的速度。并且经过实验测试，使用二维位置的 Q 表无法做到实时交互，在没有 gpu 的情况下训练缓慢。

奖励函数设计：通过合理设定不同情境下的奖励值，引导智能体更有效地探索和利用路径。特别是对撞墙行为的惩罚设计，使得智能体能够更快地学习避开障碍物。

探索率衰减策略：采用指数衰减的方式逐步减少探索率，确保智能体在训练初期广泛探索，在后期集中利用已学得的知识，提高了算法的收敛速度和稳定性。

折扣因子设置：在实验初期，设置折扣因子 $\gamma = 0.9$ 常常寻找路径失败，或者在能够到达的地点显示无法到达，这意味着算法对迷宫全局的掌握程度不佳，经上调折扣因子 $\gamma = 0.98$ 后，搜索能力明显提升。

5. Github 链接：

https://github.com/JunchengZhong/RL-Maze_Path_Search.git