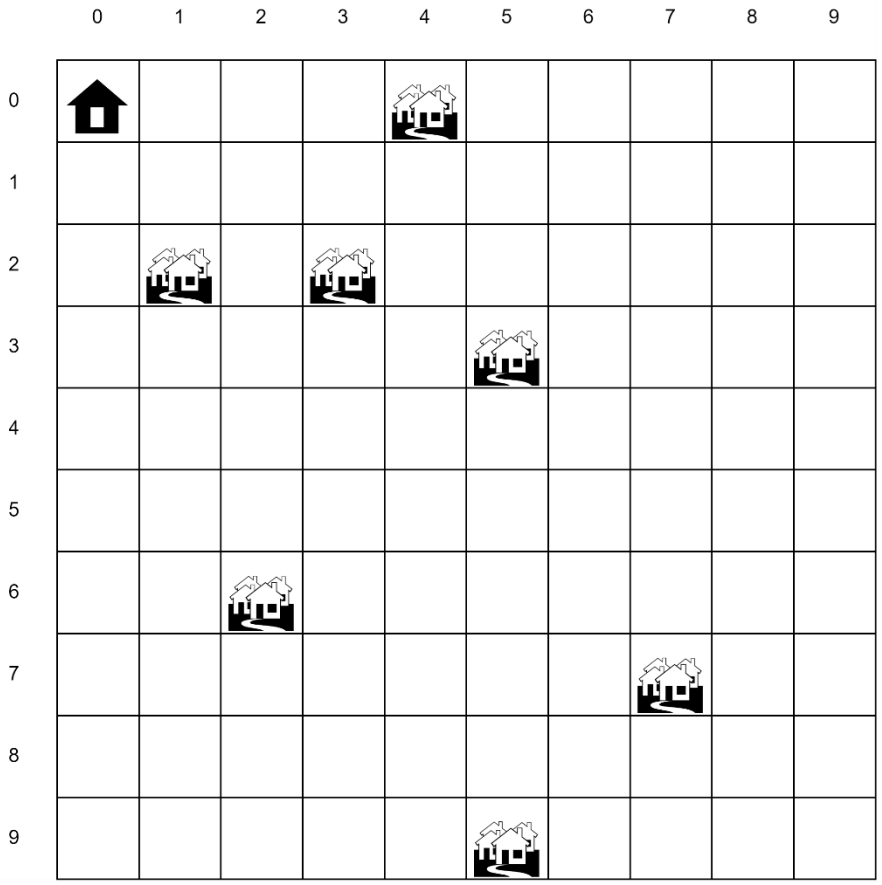
强化学习算法实践 说明文档

项目内容

在如图所示的**方格世界**（Grid World）中使用课上学习过的两类强化学习算法完成对**旅行商问题**（Traveling Salesman Problem）的一个具体实例进行求解。

具体地，旅行商(Agent)从起点城市(🏠位置)出发，求一条途经且仅途经一次所有中间城市(🏘位置)并回到起点城市的**最短路径**。



项目模块

该项目实践主要分为三个部分：**马尔可夫决策过程环境**、**基于值的算法**与**基于策略的算法**。

构建马尔可夫决策过程环境

为了在强化学习的范式下解决该问题，首先需要将该实例建模为**马尔可夫决策过程**环境(Markov Decision Process)。

为此，项目中需要实现一个类（或对象）*Env*，提供 *reset* 与 *step* 两个函数接口：

- ✓ *Env.reset*()：初始化环境并返回环境的初始状态(initial state)。
- ✓ *Env.step*(*action*)：参数为智能体当前时间步执行的动作(action)，环境根据该动作进行更新，并返回更新后的状态(next state)、当前时间步的即时奖励(reward)、环境是否达到终止态(done)。

对于该实例的马尔可夫决策过程环境的要求与提示：

- ✓ 不需可视化该环境，只需实现该环境的内部逻辑并提供上述两个接口供与智能体交互即可。
- ✓ 环境的状态规定为三元向量$(S_{visited}, X_{pos}, Y_{pos})$，其中$S_{visited}$为一个 7 位二进制数，每一位表示一座中间城市是否曾抵达过，X_{pos}与Y_{pos}分别表示智能体当前所处的行与列。这并不包含环境的全部信息，即智能体对环境是部分可感知的(partially observed)。
- ✓ 初始状态始终为$(0,0,0)$，合法状态集的大小为$2^7 \times 10^2$。
- ✓ 每个状态的可行动作集均为$\{0,1,2,3\}$，分别代表智能体向上下左右四个方向移动一格。
- ✓ 有两种情况均视为到达终止态，done 值为真。一是**成功结束**，完成一条途经且仅途经一次所有中间城市并回到起点城市的路径(不要求是最短的，均视为成功)；二是**非法终止**，智能体出界或者到达此前已经到过的城市视为抵达非法状态。其余情况视为未到达终止态，done 值为假。
- ✓ 到达终止态后需要调用 *reset* 接口，开始新一轮交互过程。
- ✓ 当智能体完成一条途经且仅途经一次所有中间城市并回到起点城市的路径后，返回 reward 值为 10000，其他状态转移返回的 reward 值可设为-1，从而保证该环境的最优解是**一条途径且仅途经一次所有中间城市并回到起点城市的最短路径**。
- ✓ **除成功抵达终止态之外，其他状态转移过程中返回的即时奖励可自行调整，但需保证最优解为所求路径。**

学习智能体策略

本次项目要求使用两种**不基于模型的强化学习算法**学习智能体的策略，前面已经提到智能体对环境仅是部分可感知的，因此不能用搜索等传统算法。学习过程包含两个部分：智能体与环境交互收集数据、智能体根据收集到的数据进行策略更新。

Q-learning 算法

Q-learning 算法下智能体的策略为参考 Q 值表进行动作选择，如何根据 Q 值表选择动作可自行设计（如 ε-贪心(epsilon-greedy)等）。

Q-learning 算法大致流程为：根据状态与 Q 值表决定执行的动作、与环境交互、根据交互结果更新 Q 值表，然后往复循环。

Policy Gradient 算法

本次项目中 Policy Gradient 算法要求**不用深层网络进行实现**，具体地，策略的参数是一个大小为$(2^7 \times 10^2, 4)$的矩阵，每个状态对应矩阵的一行。每个状态下，根据矩阵中对应行的 4 个参数值进行动作选择，如何根据值进行动作选择需自行设计。

Policy Gradient 算法大致流程为：收集数据、根据数据计算策略参数的梯度、根据梯度进行策略参数更新，然后循环往复。其中，收集数据过程为循环根据策略参数选择动作、与环境交互。

提交内容

该项目实践需要提交如下内容：

- ✓ 项目代码，包括环境代码、两种算法的学习代码等，代码需要有基本的简单注释。
- ✓ 学习结果，包括 Q-learning 收敛时的 Q 值表、Policy-Gradient 收敛时的参数矩阵、两种算法收敛时智能体在方格世界中的路径、两种算法学习过程中的学习曲线。
- ✓ 项目文档，包括整个代码的流程、值得说明的实现细节、自行设计与探索内容以及对两种算法的分析等。