Maze\_env.py:

Class Maze 主要用于定义交易环境，一共包含 3 个function:

1. init 函数，主要用于初始化 交易环境所需要的变量。
2. build\_maze 函数，主要在创建Maze 变量以后，对Maze 环境变量就进行赋值。
3. reset 函数，主要是在一次episode 结束以后，对环境变量进行重置，利用当前weights 参数更新后的 DQN 再进行下一轮episode 的。

RL\_brain.py:

Class DeepQNetwork 主要用于 从 Class Maze 中接收 observation， 并作出决策 action， 一共包含6个function：

1. init 函数， 主要用于初始化 DeepQNetwork 的变量。
2. build\_net 函数， 该函数通过 TensorFlow 搭建了Deep Q-Network 的结构。
3. store\_transition 函数， 用于接收 旧的state, action, reward，以及 新的state。 用于储存transition 数据
4. choose\_action函数，主要用于 接收 observation ， 并返回一个 action 的数值，以便 Maze\_env.py 中的step 函数能通过这个action 数值 与环境变量进行交互。
5. learn 函数， 是Deep Q-Network的 学习函数，用于更新Deep QNetwork 的参数。
6. plot\_cost 函数，用于画cost 函数曲线图。

Run\_this.py:

run\_maze 函数，重复 5轮 episode，

初始化maze 环境，

重置 maze 环境变量。

在每轮episode中，使用 DQN 的choose\_action 函数 接收 observation 并产生 action。

使DeepQNetwork 产生的 action 作用于 trading environment 上，并更新 observation, reward， done 这三个环境变量。将其作为 transition 储存在 DeepQNetwork 中。

让DeepQNetwork 每当 Step 能被5整除时对Network 的 weights进行更新。