CS 441

Program 3 Write up

Junyi Feng

This program is implemented in Python by using Pycharm. To run this program, just click 'run' in Pycharm. First, initialize the Q_Matrix as a dictionary. Next, there is a robot class that provides some features to enable the robot to perceive the current state and surrounding state. Action selection is based on the epsilon value(-0.001 every 50 epochs). To select an action, if the randomly generated value is less than epsilon, then choose a random action. Otherwise, get the max value for a state(based on Q_Matrix) as its next action. To perform an action, move the robot by the result from action selection(0: pick up, 1: north, 2: south, 3: east, 4: west), returns the corresponding reward and penalty for this action. For training, I initialize a 12x12 grid(50% probability contain a can for a square and -1 bounds) and a random start location for Robby(coordinate index 1-10) for each episode(N = 5,000 ; M = 200 ; η = 0.2; γ = 0.9). Reward and cans are also initialized to 0 for each episode. In each episode, Robby will perform M(200) actions: get the current state, select and perform an action(if the state is not in Q_Matrix, append it to Q_Matrix), and the most important thing is to update the Q_Matrix in each step. Testing is similar to training except for not updating Q_Matrix and epsilon set to 0. Finally, a plot and results are displayed after training and testing are done. Performance may vary because instead of trying to find the maximum number of cans, Robby tries to find the best action for the current state.

plot:

1.

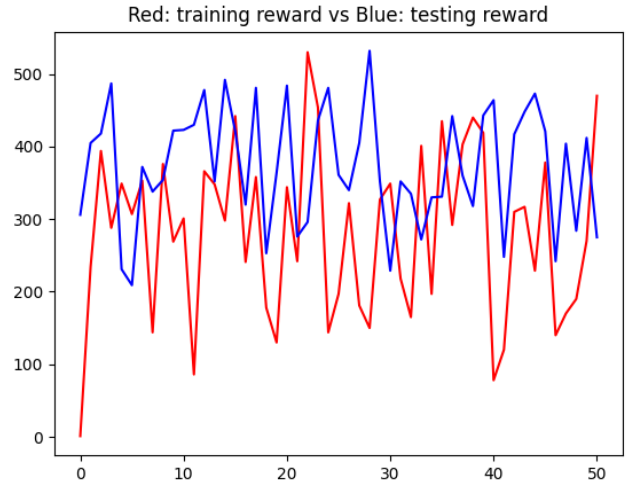
```
Average Reward for Training: 297.2196
Standard Deviation:  112.32440690524237


Average Reward for Testing: 454.6024
Standard Deviation:  52.88654385070318
```

2.

```
Average Reward for Training: 282.2192
Standard Deviation:  107.53996528097441


Average Reward for Testing: 359.3782
Standard Deviation:  90.56216779961224
```



3.

```
Average Reward for Training: 296.0312
Standard Deviation:  115.34828551288751


Average Reward for Testing: 437.2426
Standard Deviation:  53.096049567557365
```