Project2: Q-Learning for World Grid Navigation

Reinforcement learning

Name :　　LUO ZIJIAN

Matric.No：A0224725H

Email:　　luozijian@u.nus.edu

Module：　NEURAL NETWORKS（EE5904）

Task 1:

The goal of this task is to find the optimal way from initial state (1) to reach the goal state (100), with the given reward matrix.

Algorithm Description:

In total algorithm, run Q-learning algorithm 10 times. For each run, and there are at most 3000 trials. For each trial, we use Exploitation and Exploration two methods to decide the next step. Different action will get different Q-value and different reward from current policy. I use the rand function to make a random number, and compare it with the exploration probability to decide to choose which method.

As for the stop condition of each run, I set a threshold for the convergence of Q-matrix, if this condition is satisfied, we just justify this Q-learning get optimal result.

```
if  mean((Q(:) - Q0(:)) .^ 2)<0.001  %%Q_function
reach the optimal result
    break
end
```

Finally, in each run, l will maximize the Q-value table to find an optimal way. The optimal route shown in Fig.1, and the total reward = 2294.869539

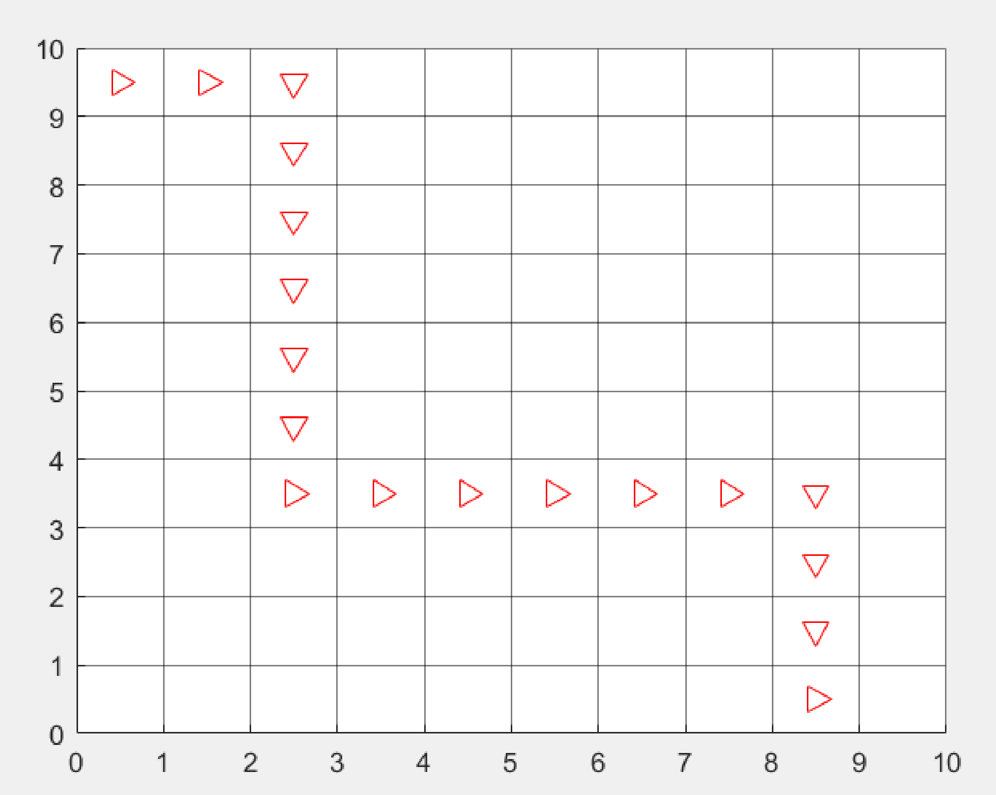


Figure.1 The optimal route

Based on the requirement from the project, I justify the performance of different parameters in this algorithm.

Table.1 The records of performance

ε_k, α_k	No. of goal-reached runs		Execution time(sec.)	
	$\gamma = 0.5$	$\gamma = 0.9$	$\gamma = 0.5$	$\gamma = 0.9$
$\frac{1}{k}$	0	0	--	--
$\frac{100}{100+k}$	0	10	--	12.042998
$\frac{1+\log(k)}{k}$	0	0	--	--
$\frac{1+5\log(k)}{k}$	0	6	--	11.005661

When we focus on the learning rate. According to the result, I can notice that there are only 2 situations that we can get an optimal path from this Q-learning algorithm. But only for $\frac{100}{100+k}$, we can the best result. As for the discount factor, all the situations in this algorithm can not get the optimal result when the discount factor equals to 0.5. Thus, I make an analysis about the impact of parameters.

1. For learning rate, the reason why $\frac{100}{100+k}$ can get best result is that this learning rate guarantee a large value more than threshold (0.005), when the step is too large. So, the farther step can provide an appropriate figure to update the Q-value. For other parameters, with the increase of k, their value approach 0 very fast. Thus, they provide very limited help for far step. If we want to get enough steps to get the optimal result, the optimal learning rate should be like $\frac{100}{100+k}$.
2. For discount factor. The larger the discount factor is, the more effect the next step has. It helps us to find the larger reward given by next step. When the discount factor increases, the total reward is more. As for 0.5, this parameter could not get enough farsighted step in total process, so it can not reach the state 100.
3. For exploration probability, the larger exploration probability is, the more random the choice of Q-value is. But in this project, we just let this parameter equals the learning rate.

Task 2:

In order to choose a learning rate, discount rate and exploration probability wisely, I try different value of these three parameters.

In this task, I suppose that the total executing time is the metrics that can detect how wisely of different groups. Because, less time means it can reach the optimal result more wisely.

Firstly, I set the gamma equals 0.9, I try different parameters like $\frac{100}{100+k}$, and record the average executing time.

Table.2 Gamma=0.9

Learning rate	Task1.mat(sec)	Own.mat(sec)
$\frac{80}{80+k}$	12.854521	13.310610
$\frac{100}{100+k}$	**12.042998**	**12.256394**
$\frac{150}{150+k}$	13.222763	14.868959
$\frac{200}{200+k}$	15.841137	15.840950
$\frac{300}{300+k}$	16.723902	16.958095

Through this result, I notice $\frac{100}{100+k}$ is still the best result we can get in my algorithm, so we set $\frac{100}{100+k}$ as the learning rate.

And then, I set the learning rate $\frac{100}{100+k}$, I try different discount factor, and record the average executing time.

Table.3 Learning rate= $\frac{100}{100+k}$

Discount factor	Task1.mat(sec)	Own.mat(sec)
0.6	--	--
0.7	**9.128899**	**9.499199**
0.8	11.574196	11.918288
0.9	14.868959	13.161320
0.95	15.238756	15.760052
0.97	15.905748	16.457075
0.99	--	--

Based on this result, we can find that less discount factor, less time it needs. But considering the issues from real world, we know less discount factor, the total reward is also less. Combining all the information, in my task2, I should set the discount factor 0.7 to satisfy all the requirements.

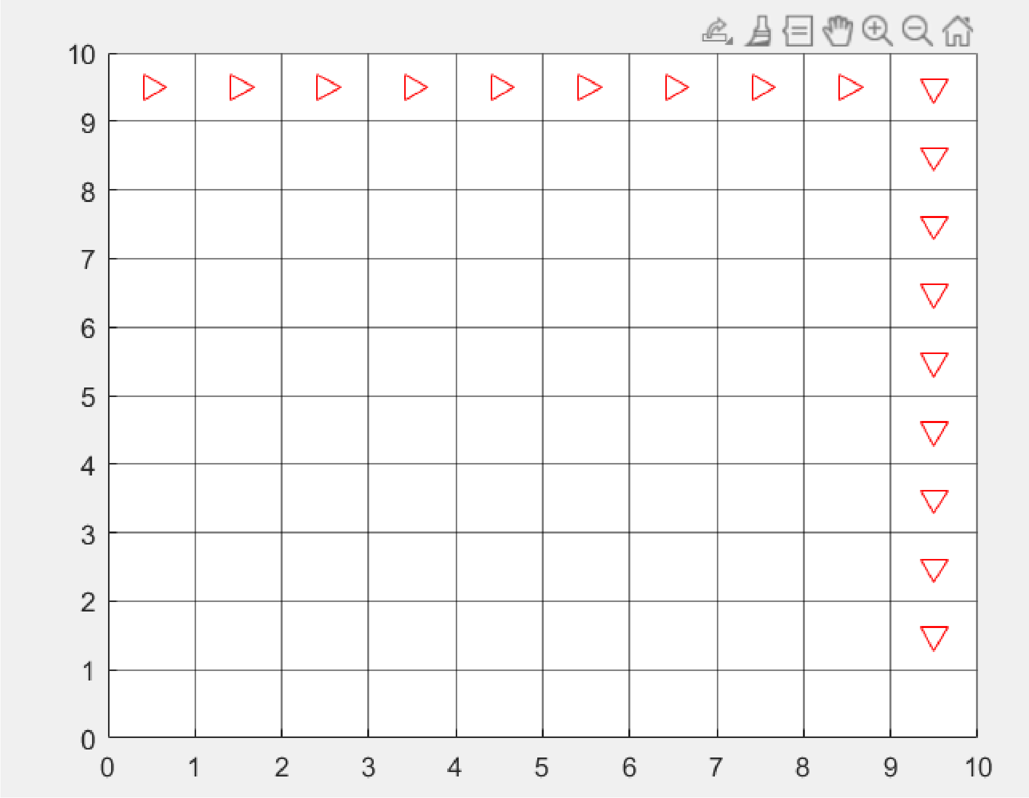
Finally, the exploration probability equals to the learning rate, so there is no record of different parameters.

Totally, I set the parameters as below.

Table.4 Parameters setting

Parameters	Value
Learning rate	$\frac{100}{100+k}$
Discount factor	0.7
Exploration probability	$\frac{100}{100+k}$

In my own reward data, I can get the best path like this.



10
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9
10