Lab Report

--Jinyi WU, Lecheng WANG

Exercise 1

Process of algorithm:

1. Initialize V(s)=0 for every steps. For each step, the policy is uniform distribution.

2 Set the maximum number of episodes; we will terminate the loop either when the maximum episodes is reached.

3 In each episode: we continue do the following until reach ‘timeout’ or get in the final state.

1. Randomly chose the start state: x.
2. Chose the action for state x (u), random sampling according to the distribution of action.
3. Perform action, get next state y, revenue r, and tell if reach terminated or ‘timeout’.
4. Compute delta, update V[x], update pi\_temp to get next pi.
5. Use the state that we already reach as the start of next loop, and ignore a).

4 What data do we store during the process?

a) V\_norms. We observe the growth of V value. It should converge to a certain value.

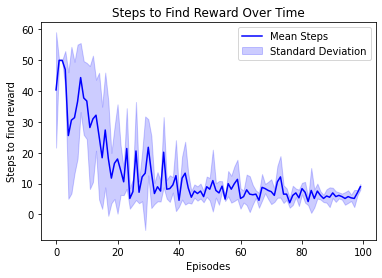
b) Time\_list. This list record how much steps we take in each episode. In the beginning, it’s hard for the agent to reach the final state, so we need to loop until timeout. But as the agent evolute, we found that even the start is random, it takes less steps to find the final state. Which means the policy is more and more effective.

c) Of course, in the end we will return the V value we get.

Parameters that we chose in the EX1:

alpha\_critic = 0.5; alpha\_actor = 0.5

gamma=0.9; timout=50; number of episodes=50

Results we get in EX1: 

The algorithm has succussed in updating V values and provide a useful policy for the maze. With the growth of episodes, the steps it takes to find the reward significantly decrease.

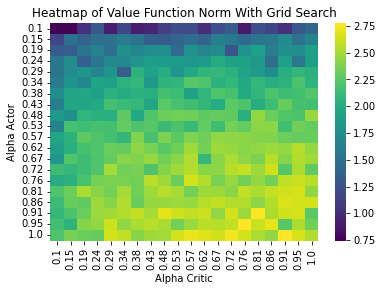
Exercise 2

Grid search method:

* + 1. We generate alpha with np.linspace(0.1,1,20) for both alpha\_critic and alpha\_actor, than we use itertools.product() to produce 400 pairs of parameters for grid search.
    2. During the search, we evaluate parameters by the V value obtain in the end. The lager the better. To avoid coincidence, we run each pair for 5 times and get a average value of V.
    3. After running all pairs, we finally conclude which are the best parameters. In our training, the grid search has shown the pair below is the best (each time can have slightly difference)

alpha\_critic = (0.9526315789473684)

alpha\_actor = (0.7157894736842105)



The value of alpha critic is more stable, mostly over 0.9. However, the value of alpha actor varies more widely, the values that we observed are mostly between 0.6 to 0.9. We guess the main reasons for the phenomena are as follows:

-The method itself has uncertainty. The choice of initial state, the action for each state is chosen randomly (depends on distribution of policy) instead of the largest one.

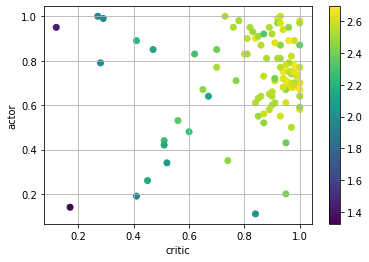
-The evaluation is base on V value, but it is possible that with in 50 episodes, there are a range of alpha can lead the V value get from iteration close enough the true value. In this case, the uncertainty can dominate the difference between different alpha thus hide the actual performance of each pair of parameters.

Bayes optimization method:

In this method, we apply optuna.creat\_study() to conduct bayes search. The frame work is the same as we have done previously in grid search. The only difference is when we chose the next parameter, we let optuna choose for us between 0.1 to 1 instead of fix the parameters in the beginning. The results are:

alpha\_critic = ((0.9636166019026832)

alpha\_actor = (0.6854343269718869)

Draw heatmap: 

It’s easy to get a heat map for Grid Search because the grids are uniformly over the 2 axis. However, we can’t directly draw a heat map for Bayes method because the density of the value we take in 2 dimensions isn’t uniform. So, in order to show the distribution of points it chose and the value of correspond V in the same time, we draw a scatter and use color to represent the V value.

Exercise 3

We use functions provided in the notebook to test the result. The test is aim to detect the difference between hyper parameters. Here are some pictures.

To draw a conclusion, after 50 episodes, we have to admit there are no significant difference between the V value obtained by different alphas. Optimized parameters obtained by 2 different methods slightly better than the naïve parameters when the episode is small. We believe that the naïve alphas isn’t that bad in the first place, the grid search backs our point. So, if we compare the medium parameters with the best parameters, considering the uncertainty of the algorithm, the difference become less and less during the iteration is a reasonable conclusion.