Student Name: Mohammad Mohammad Beigi
Student ID: 99102189
Subject: Reinforcement Learning

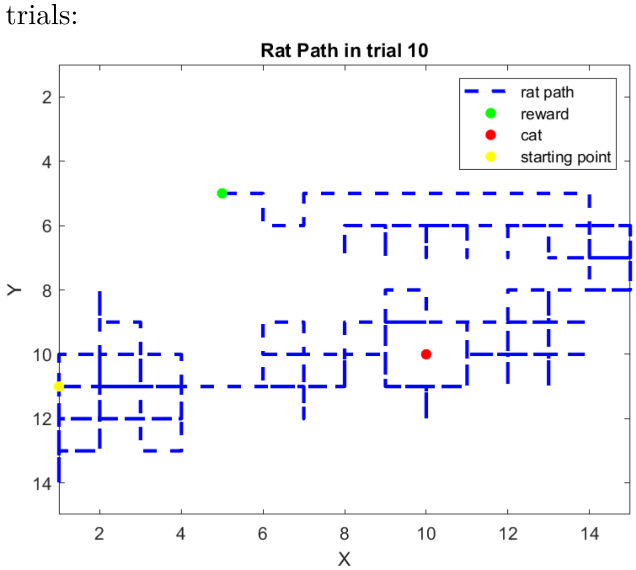
Advanced Topics in Neuroscience - Dr. Ali Ghazizadeh
Assignment 6

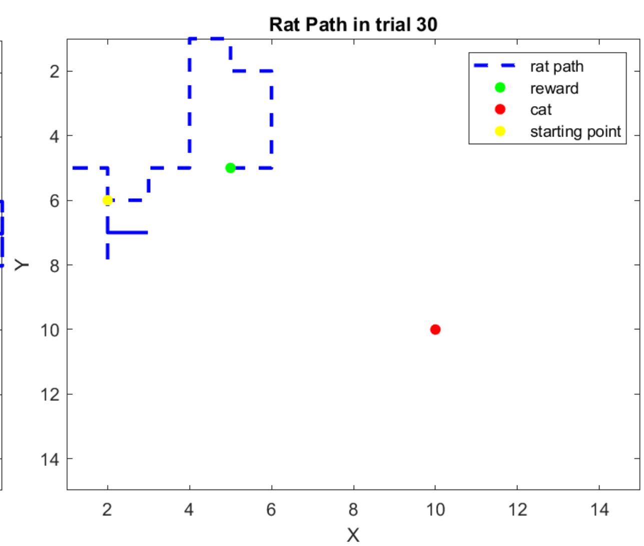
Learning the Water Maze:

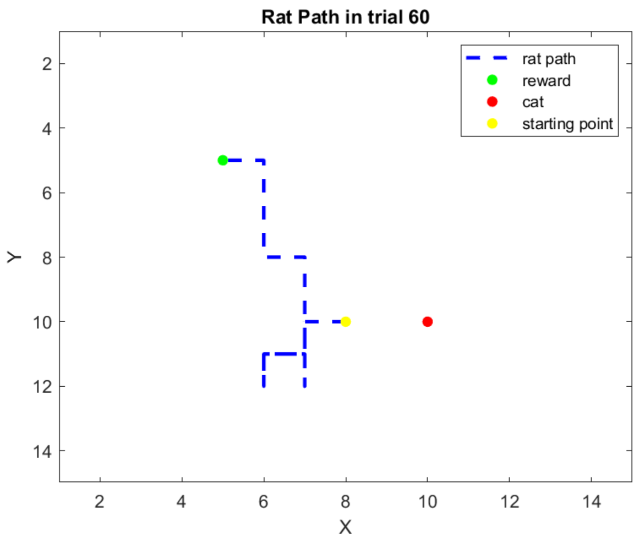
As an example of generalized reinforcement learning, we consider the water maze task. This is a navigation problem in which rats are placed in a large pool of milky water and have to swim around until they find a small platform that is submerged slightly below the surface of the water. The opaqueness of the water prevents them from seeing the platform directly, and their natural aversion to water (although they are competent swimmers) motivates them to find the platform. After several trials, the rats learn the location of the platform and swim directly to it when placed in the water. We are going to simulate a simple model of navigation problem.

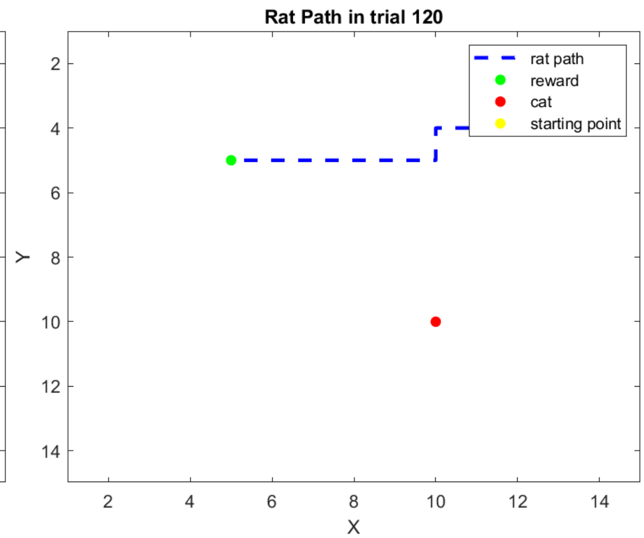
1. Plot the paths before and after training.(assign demo files (video format))

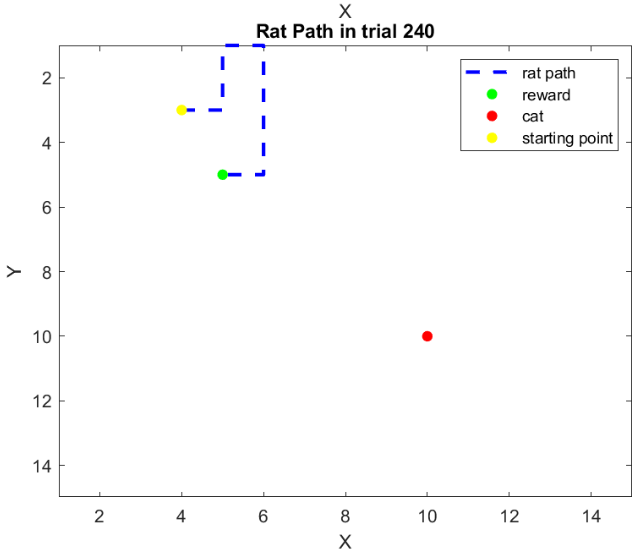
   The video format demo is in zip file. In the figures below rat's path has been plotted in several trials:

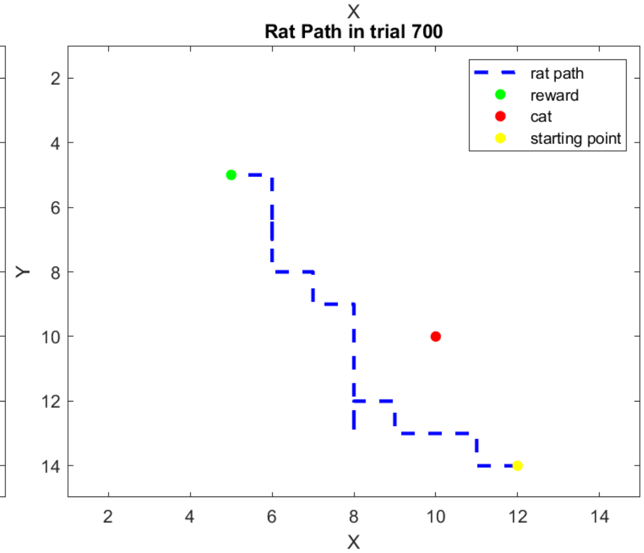


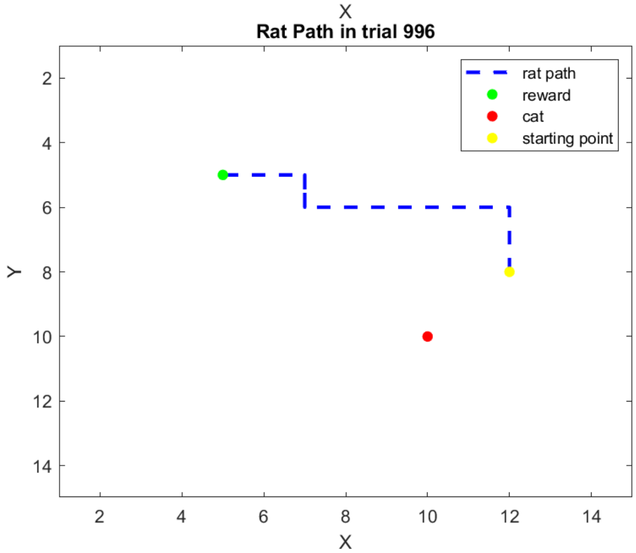


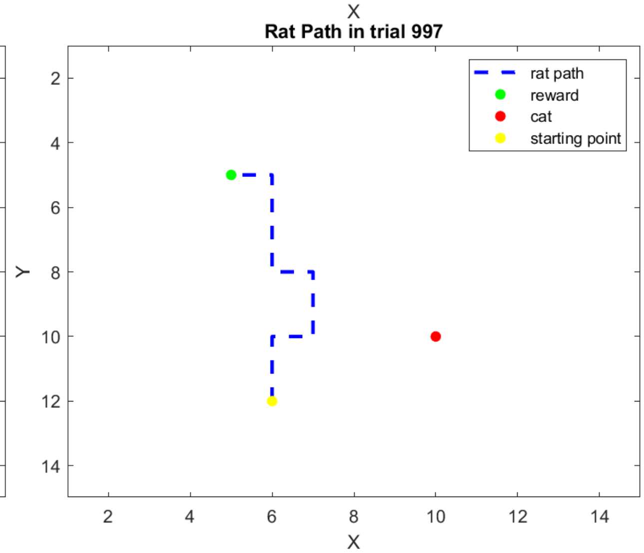

Rat Path in trial 60
rat path
reward
cat
starting point
X
Y

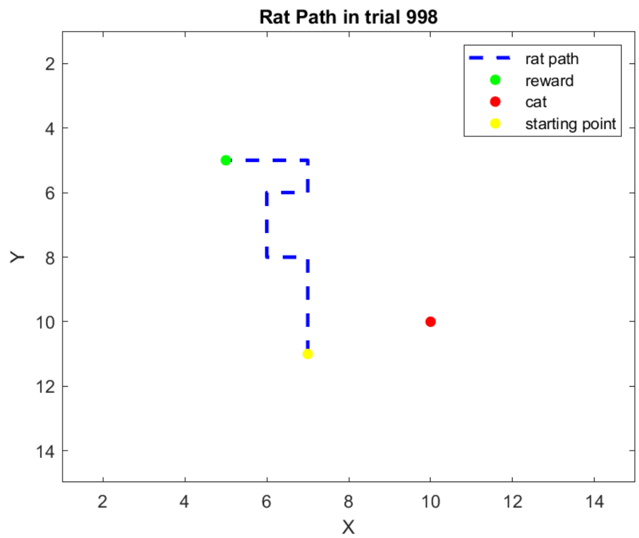

Rat Path in trial 120
rat path
reward
cat
starting point
X
Y

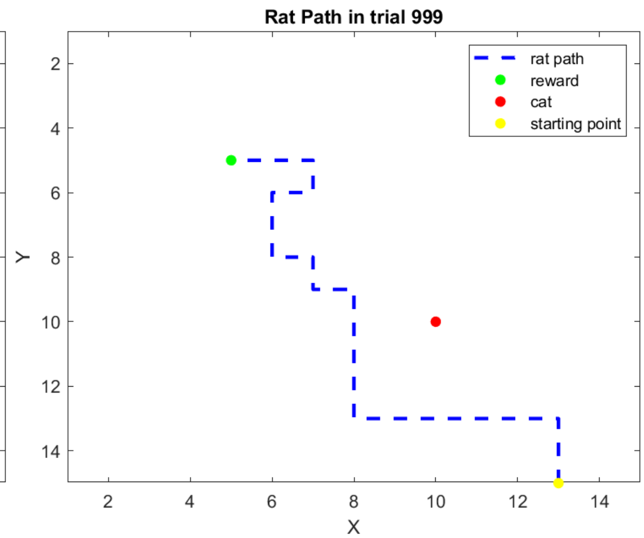

Rat Path in trial 240
rat path
reward
cat
starting point
X
Y

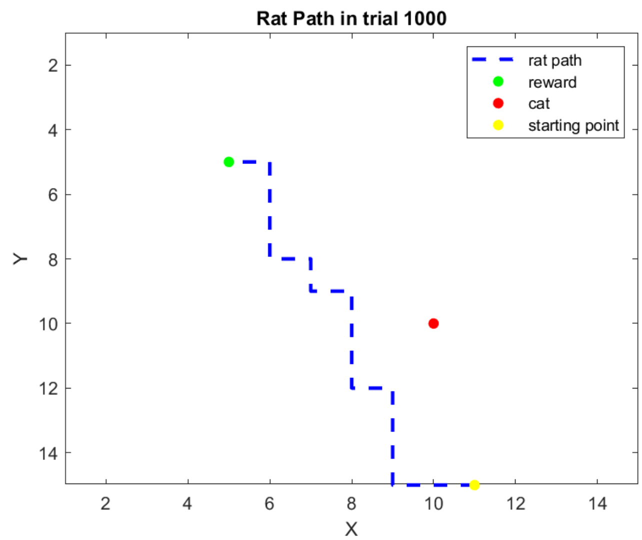

Rat Path in trial 700
rat path
reward
cat
starting point
X
Y


Rat Path in trial 996
rat path
reward
cat
starting point
X
Y

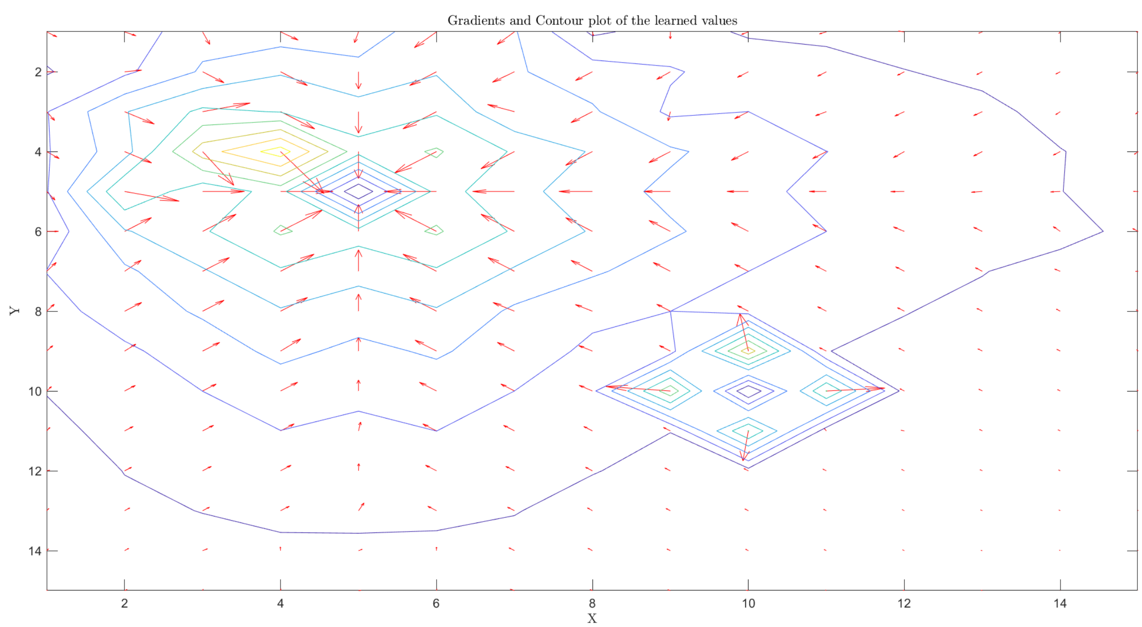

Rat Path in trial 997
rat path
reward
cat
starting point
X
Y





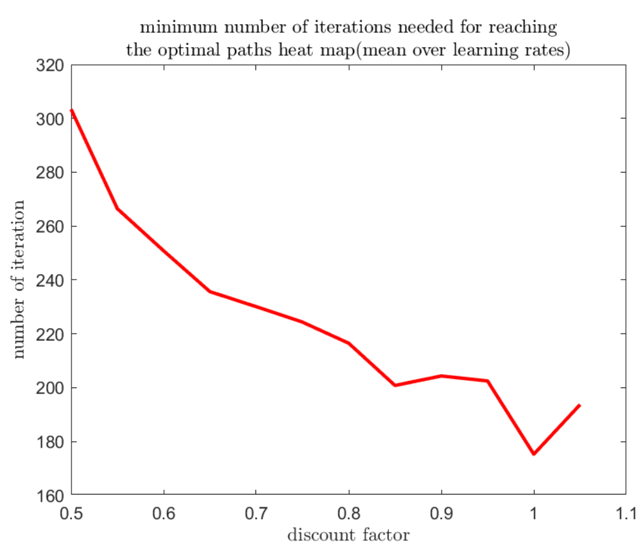


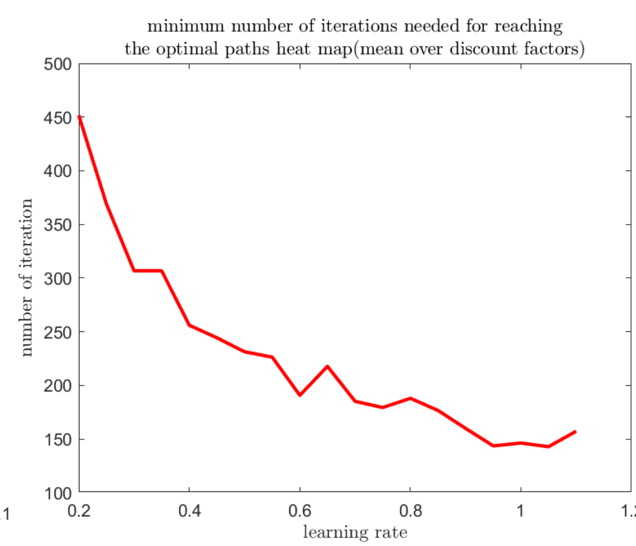
2. Plot the gradients and contour plot of the learned values.



3. What is the effect of the learning rate (α) and discount factor (γ) in this problem?

   We should consider a criterion for the minimum number of iterations needed for reaching the optimal paths. Our criterion is that mean of number of actions of the last 40 iteration should get to less than 14.(14 is chosen by the fact that reward cell is located in [5, 5] and somehow a mean number of actions for the best possible path has been calculated).





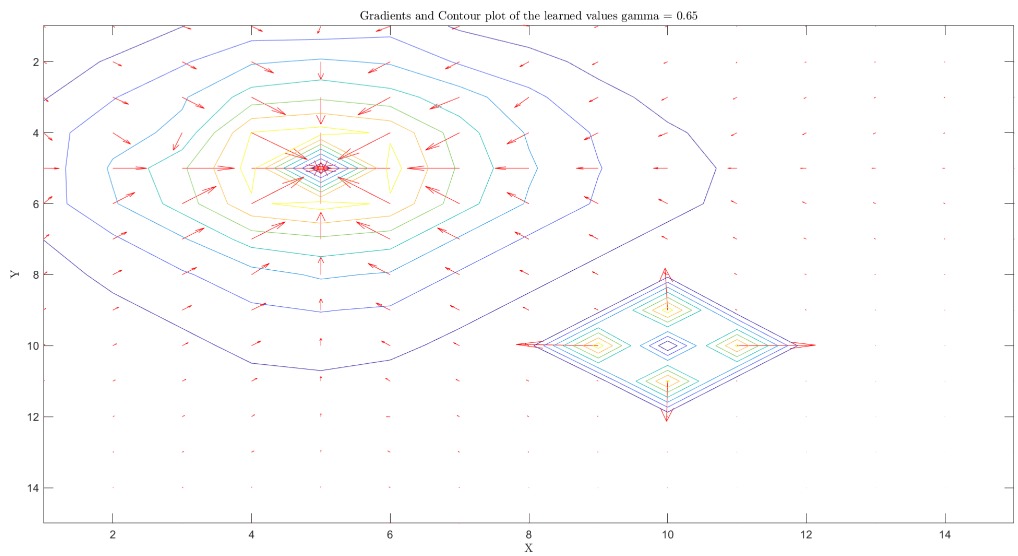
   The learning rate (α) controls how much the Q-value of a state-action pair is updated based on the observed reward. If the learning rate is too high, the algorithm will respond more rapidly to changes in the environment, but may also oscillate or diverge. On the other hand, if the learning rate is too low, the algorithm will take a longer time to converge, which may be acceptable in some cases, but may not be desirable in time-sensitive or critical applications.

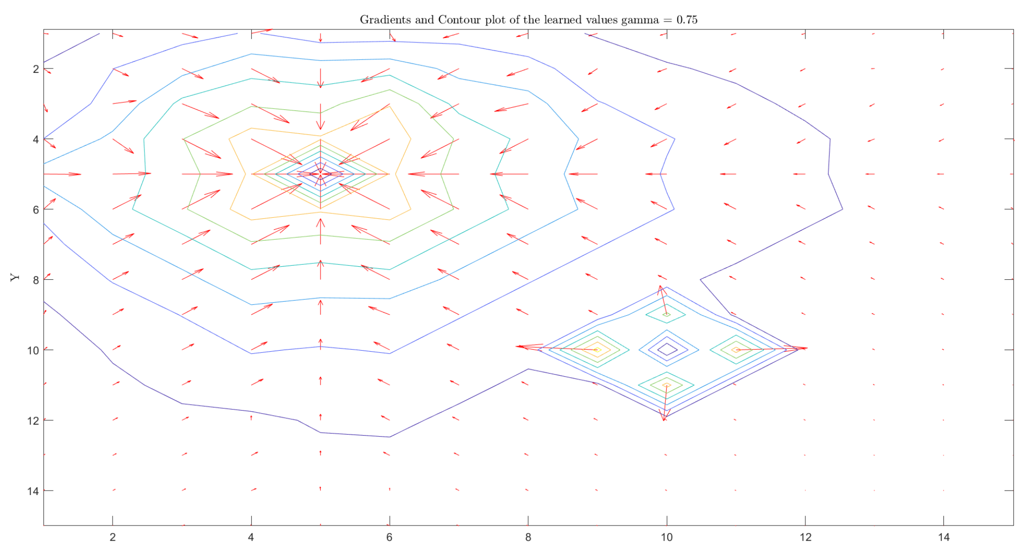
   The discount factor (γ) controls how much importance is placed on future rewards versus immediate rewards. A high discount factor values future rewards more, meaning that the agent will be more forward-looking and focus on actions that lead to higher cumulative rewards over time, whereas low discount factor values immediate rewards more, potentially leading to more myopic policies.

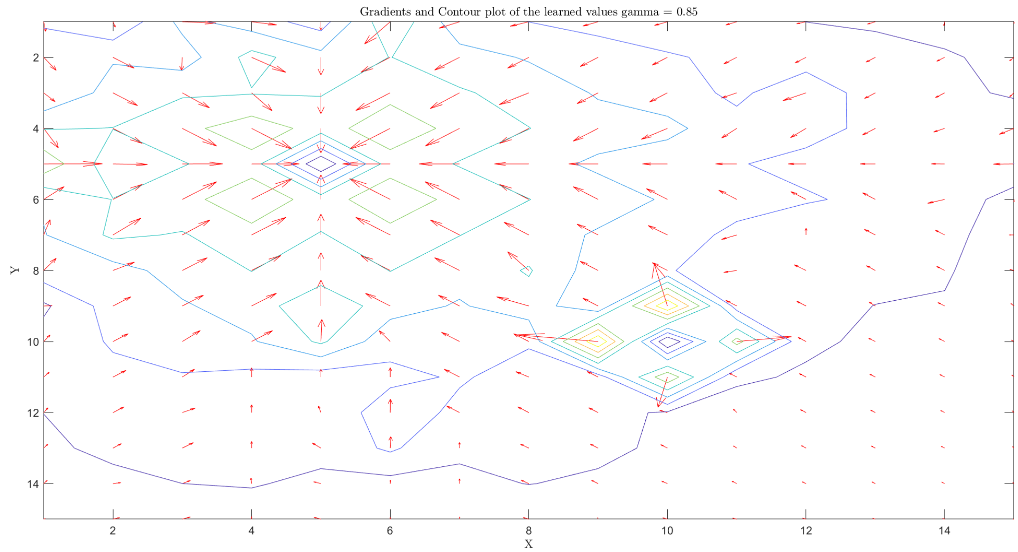
A larger value of the discount factor generally encourages the agent to take longer-term actions that lead to greater rewards. However, setting the discount factor too high can lead to slow learning and convergence issues. Conversely, a smaller value of the discount factor may lead to more short-sighted actions, reducing the overall performance of the algorithm.

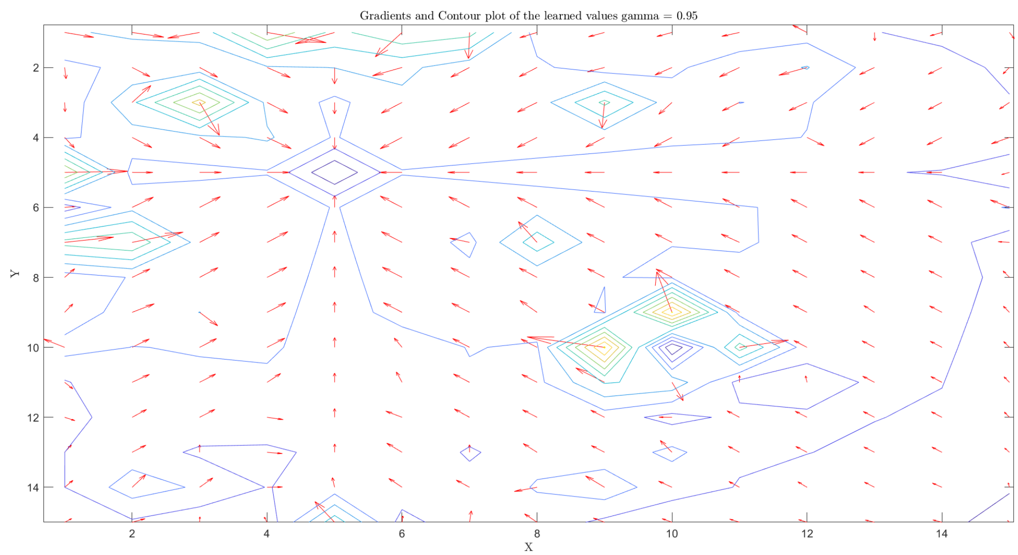
In summary, adjusting the values of the learning rate and discount factor can significantly impact the performance of Q-learning. The optimal values of these parameters may depend on the specific problem and dataset. Therefore, it is important to choose appropriate values for these hyperparameters through experimentation and iterative tuning. In general, decreasing and over time or using adaptive learning rates can help improve the convergence rate of Q-learning and lead to better performance.

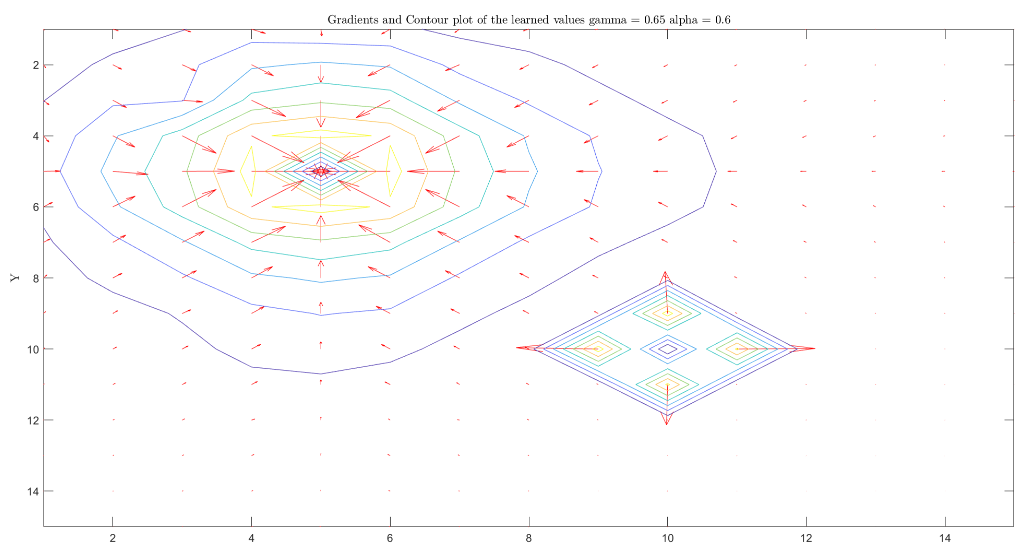
learning rate = 0.8:

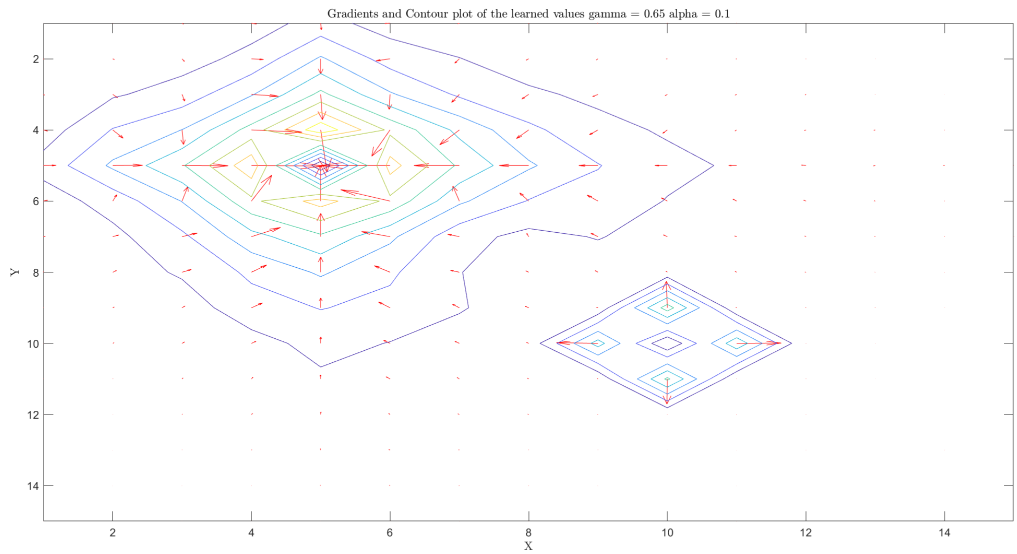





Gradients and Contour plot of the learned values gamma = 0.85
Y
X

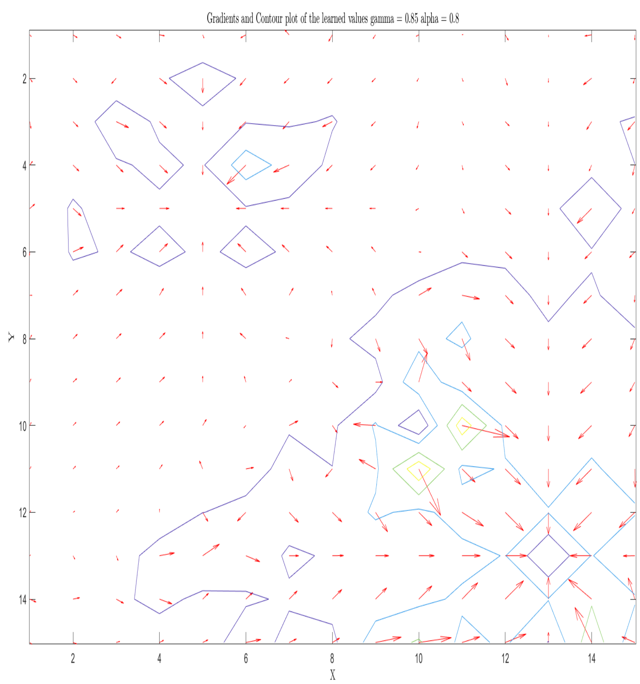

Gradients and Contour plot of the learned values gamma = 0.95
Y
X

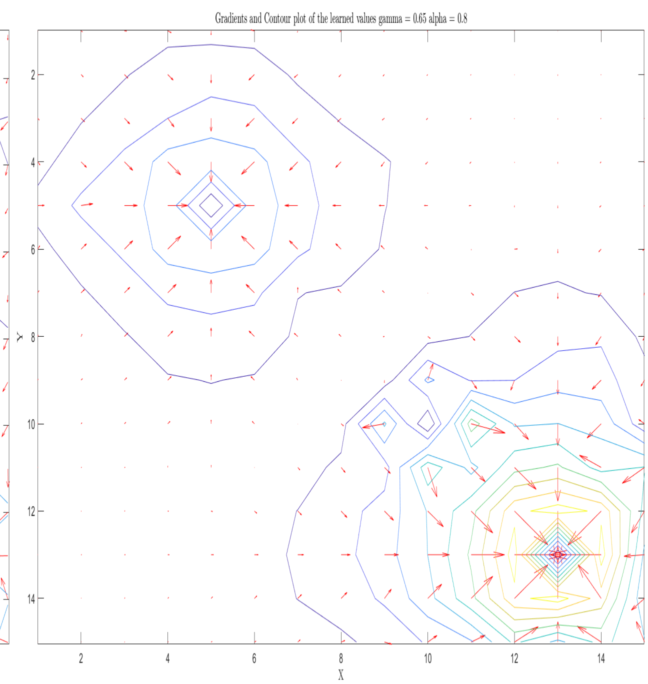

Gradients and Contour plot of the learned values gamma = 0.65 alpha = 0.6
Y
X

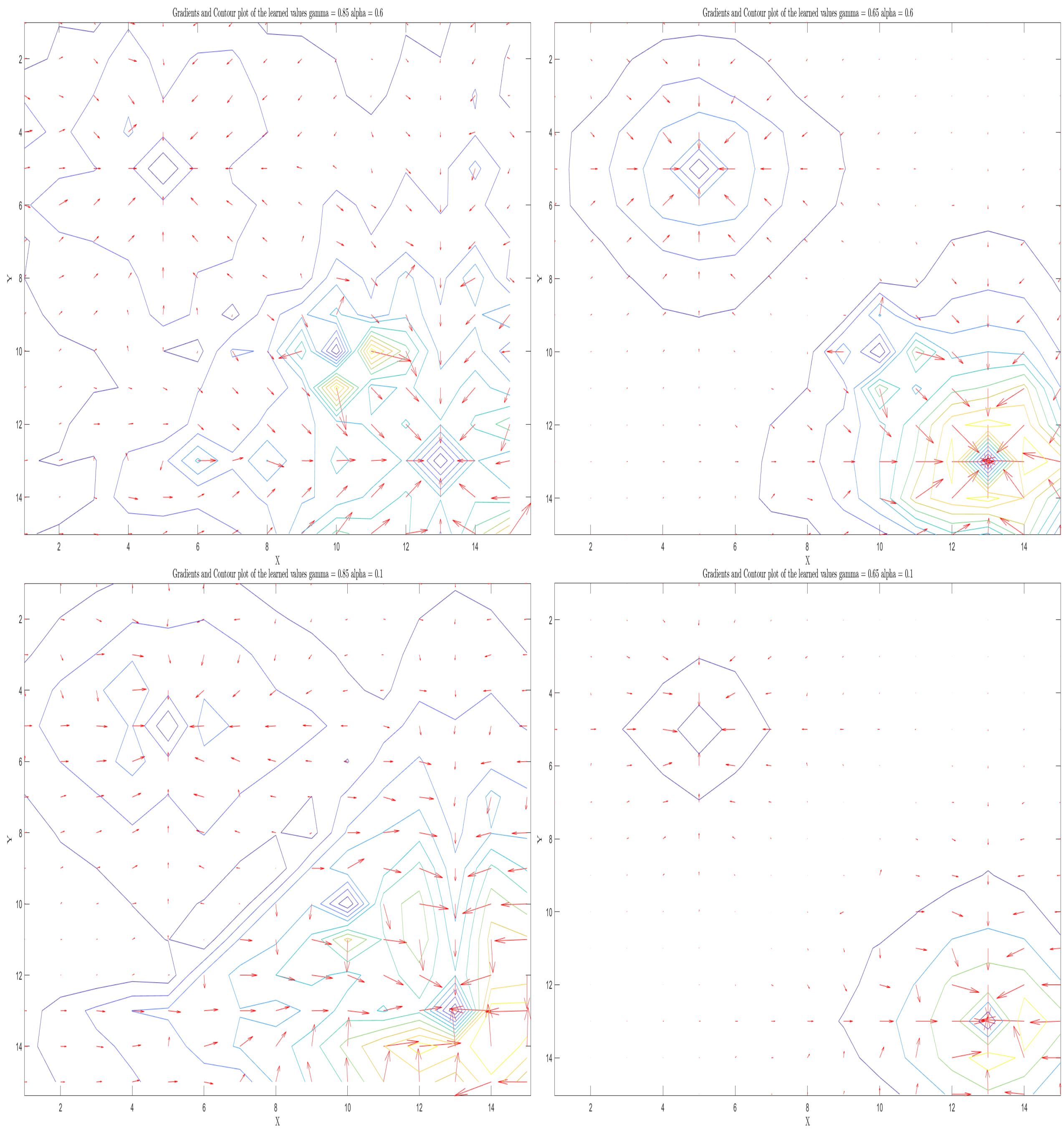


4. Consider two target squares with different positive values; what is the effect of and in the learning procedure?

   We plot gradients and contour plot of the learned values per different amounts of α and γ:







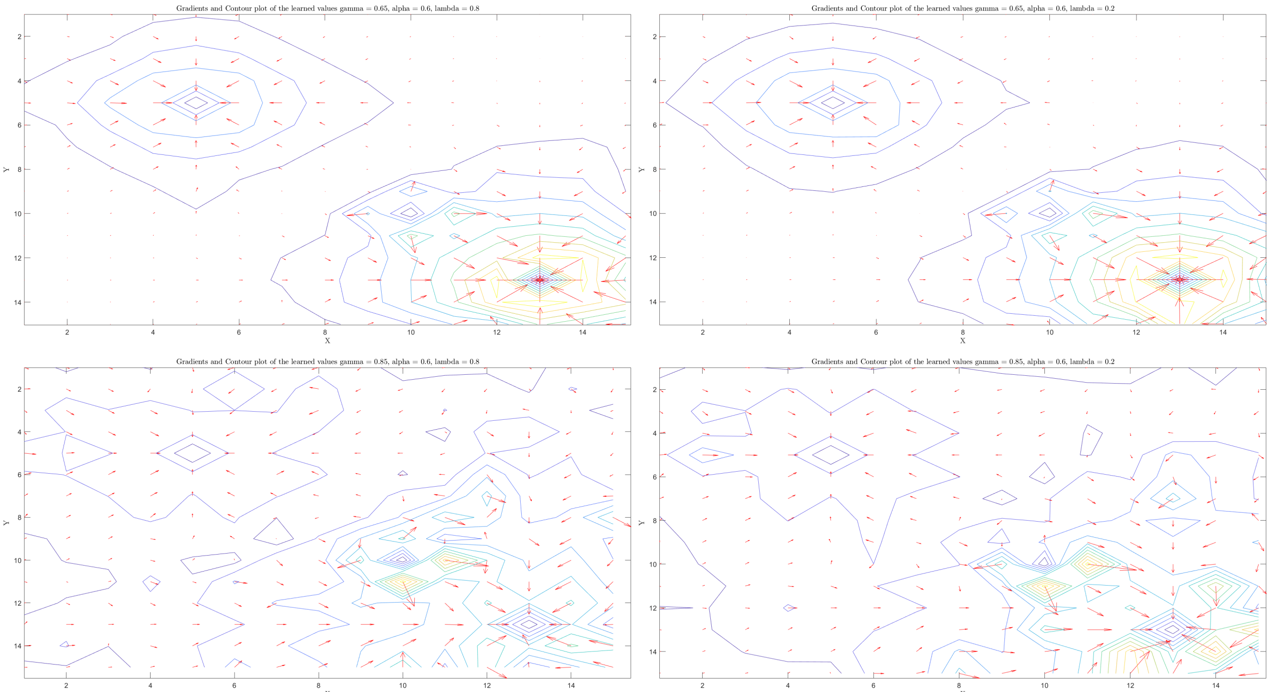
The learning rate controls the adjustment of Q-values based on the observed rewards. Since there are two target cells with different rewards, the learning rate can affect how the agent learns the relative value of each target. A high learning rate can skew the agent towards prioritizing the target with the highest reward, while a low learning rate can increase the exploration and help the agent learn the values of both targets more equally.

The discount factor γ, on the other hand, balances the importance of immediate rewards versus future rewards. When there are two target cells with different rewards, a high discount factor may prioritize the target with the highest long-term reward, while a low discount factor may put more emphasis on the immediate reward. However, the optimal value of γ depends on the rewards' difference between the two targets as a high γ can undermine the importance of low-reward targets, leading to sub-optimal policies.

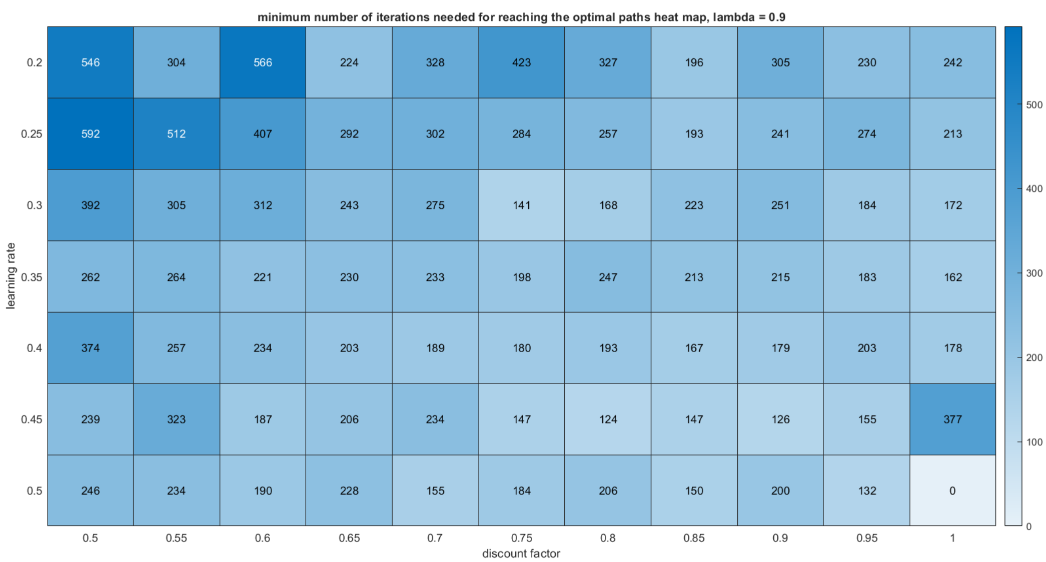
As you see in the figures, the lower discount factor has caused not to find optimal path when the rat is not near the targets(Somehow higher learning rate can compensate this issue). On the other hand lower learning rate has caused to learn the water maze more accurate and based on relative values of targets and cat.

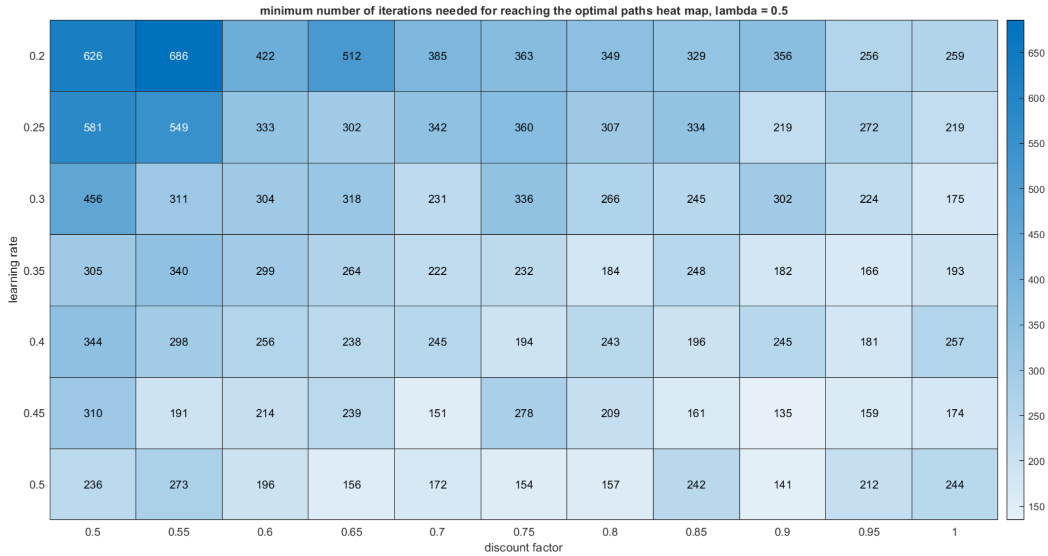
5. Implement TD(λ) algorithm and compare with previous section.

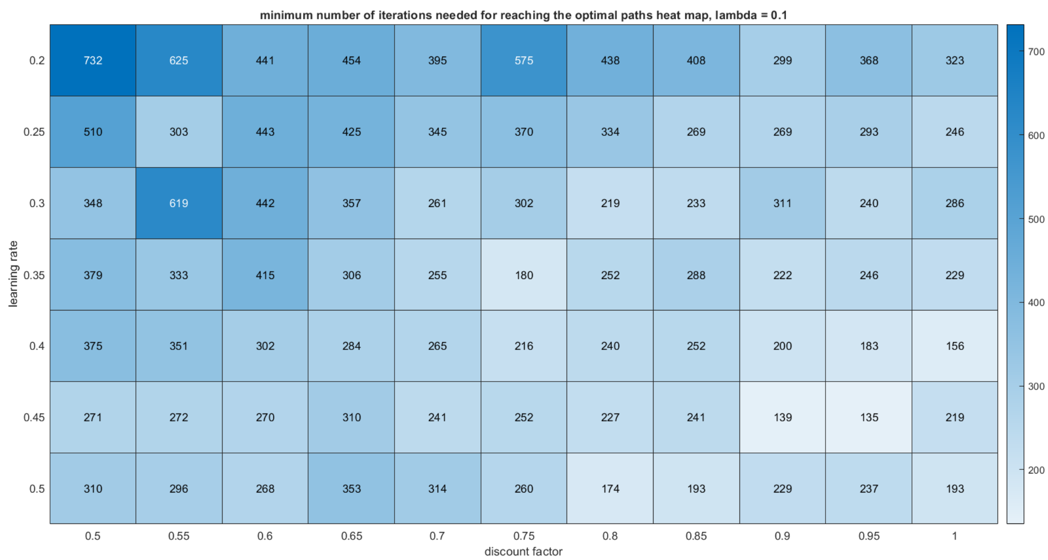
   As you see in the following figure, approximately there is no difference between the final results of values when we have different λs.



   So we check the effect of lambda in number of trials needed to find optimal path(the criterion is that we used in part 3 of this homework).







TD(λ) is used to speed up the learning procedure.

When λ is set high in TD(λ), it means that the eligibility trace will assign more weight to the past TD errors and less weight to the current TD error. This has the effect of making the learning more biased towards long-term estimates, which means that the agent is more likely to consider future rewards when making its decisions. As a result, the agent may be more willing to explore new actions and take a longer time before converging to a policy that maximizes the expected reward.

On the other hand, when λ is set low, the eligibility trace will assign more weight to the current TD error and less weight to the past TD errors. This leads to a more myopic learning approach, where the agent focuses more on short-term estimates of the expected reward, rather than considering future rewards. In this case, the agent may converge to a policy that maximizes the short-term reward faster, but with a higher risk of missing out on potentially larger rewards in the future.