Sharif University Of Technology

Advanced Topics in Neuroscience

Simulation 06 (Reinforcement Learning)

Instructor: Dr. Ali Ghazizadeh

Zahra Kavian - 98102121

Contents

Section 1: General Description of the Program

I explain ”full model” completely. Other parts are as same as this only with little difference.

First, place of a platform and cat are assigned and randomly choose one of game board blocks. Then, follow these formula to update each state value and policies to decide which direction will be chosen.

$$V(S_t)_{new} = V(S_t)_{old} + \eta.\delta_t$$

$$\delta_t = r(t) + \gamma \sum_{a} \sum_{S_{t+14}} (\pi(a, S_t)P(S_{t+1}|S_t, a)V(S_{t+1}) - V(S_t))$$

$$\pi(a_i, s_t) = \frac{\exp(\beta m_i)}{\sum_1^n \exp(\beta m_j)}$$

$$\begin{cases} m_i \longrightarrow m_i + \epsilon(1 - \pi(S_t, a_i)) * V(S_{t+1} - r_t), & \text{if action } a_i \text{is chosen} \\ m_i \longrightarrow m_i + \epsilon\pi(S_t, a_i) * V(S_{t+1} - r_t), & \text{if action } a_i \text{is not chosen} \end{cases}$$

This method is repeated 2000-3000 trials. In the final trials, mouse reach the platform in the shortest possible pathway. You can find the game demo here.

Section 2: Question 1

In figure 1, you can compare the path before and after the training. Also, you can watch some related demos here (in demo named ”pathway”, red point is the mouse first location and blue is the target position).

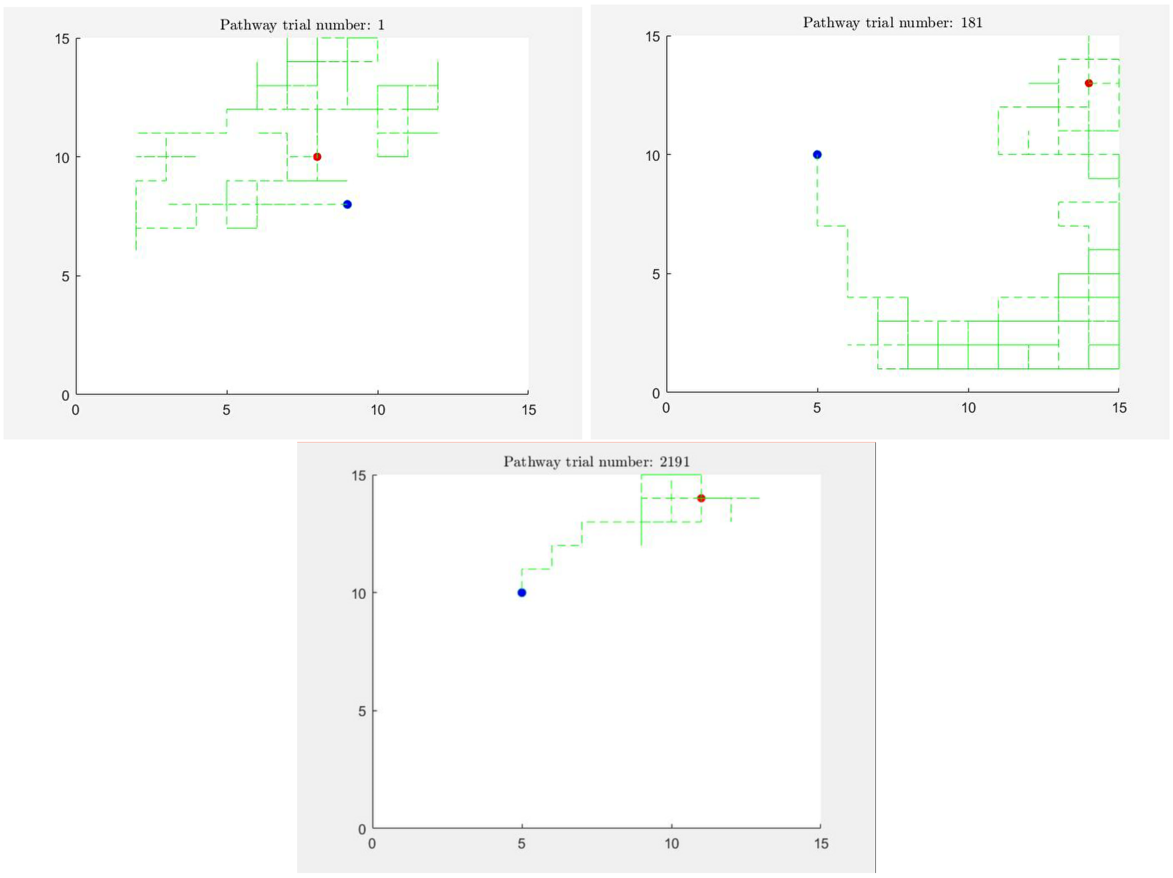

Figure 1: Pathway before and after training

Section 3: Question 2

You can find this part demo here. Three frame is shown in figure 2.

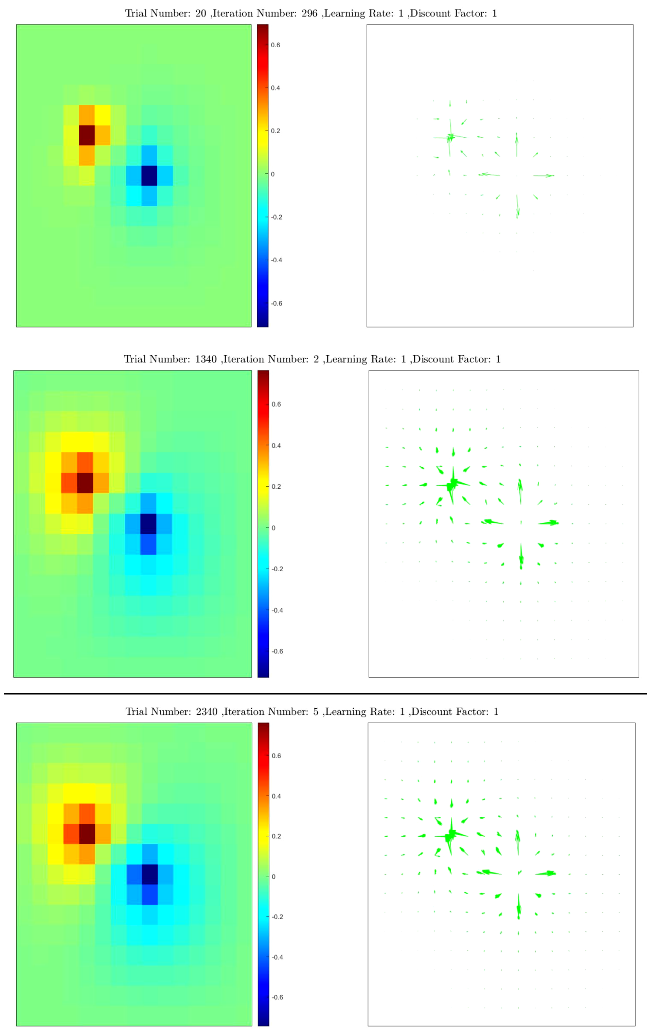

Figure 2: Gradient of learned value before and after training

Section 4: Question 3

In this part, we want to see the effect of learning rate and discount factor on learning procedure. I run the game for different kind of these parameter's value and you can watch them here. In figure 3, logarithm of average steps, which is used in last hundred trials, is shown for each learning rate and discount factor value. Learning procedure is faster for higher learning rate and discount value.

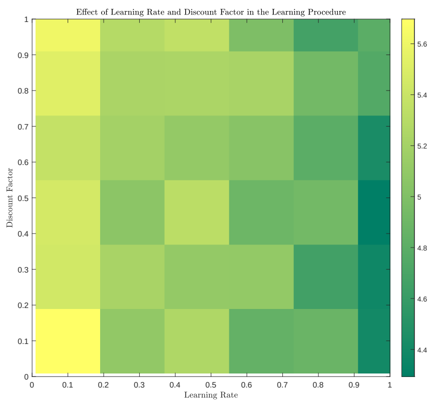

Figure 3: The effect of the learning rate (σ) and discount factor (γ)

Also, you can see demo of gradient of learned value for each learning rate and discount factor here.

Section 5: Question 4

In this part, I use two target with different value. The game is played with different learning rate and discount factor. You can see learning procedure is faster for higher learning rage and discount factor (figure). You can find this part demo here.

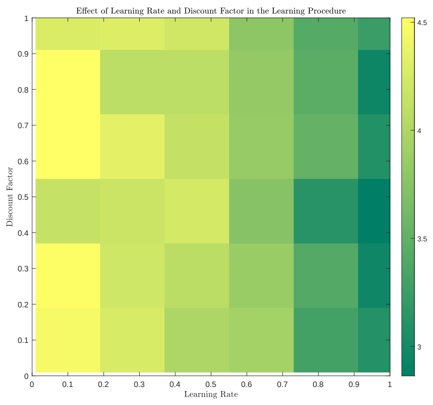

Figure 4: The effect of the learning rate (σ) and discount factor (γ)

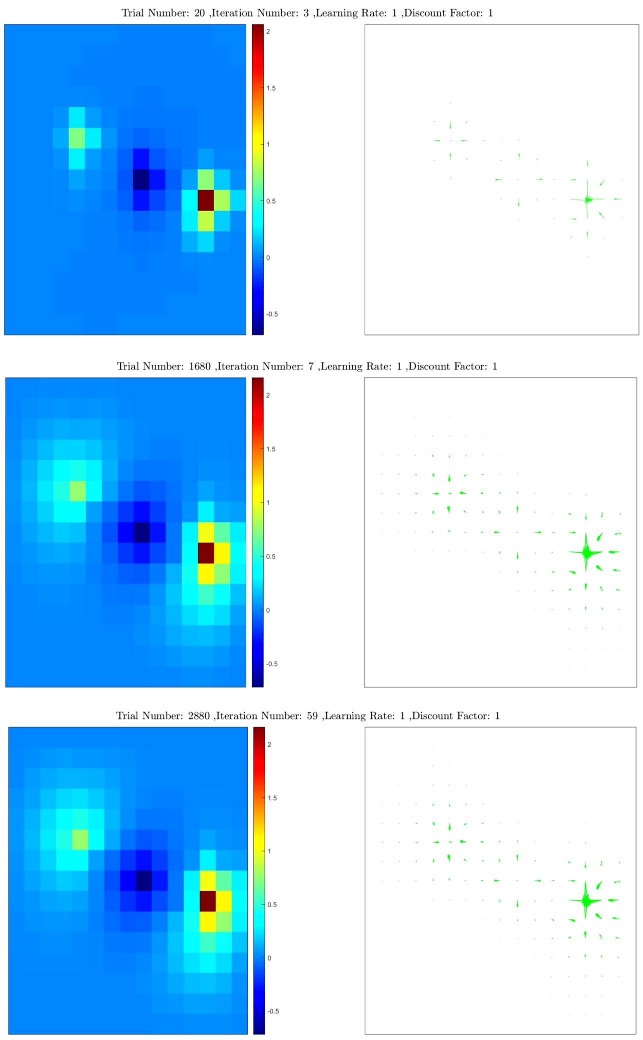

Figure 5: Gradient of learned value before and after training

Section 6: Question 5

In this part, I use TD- lambda, all previous states are updated base on the distance of that state and the target state. This help agent to learn faster and find the best method in each trial.

Three frame of pathway and gradient contour is shown in figure 6:8 . You can find this part demo here.

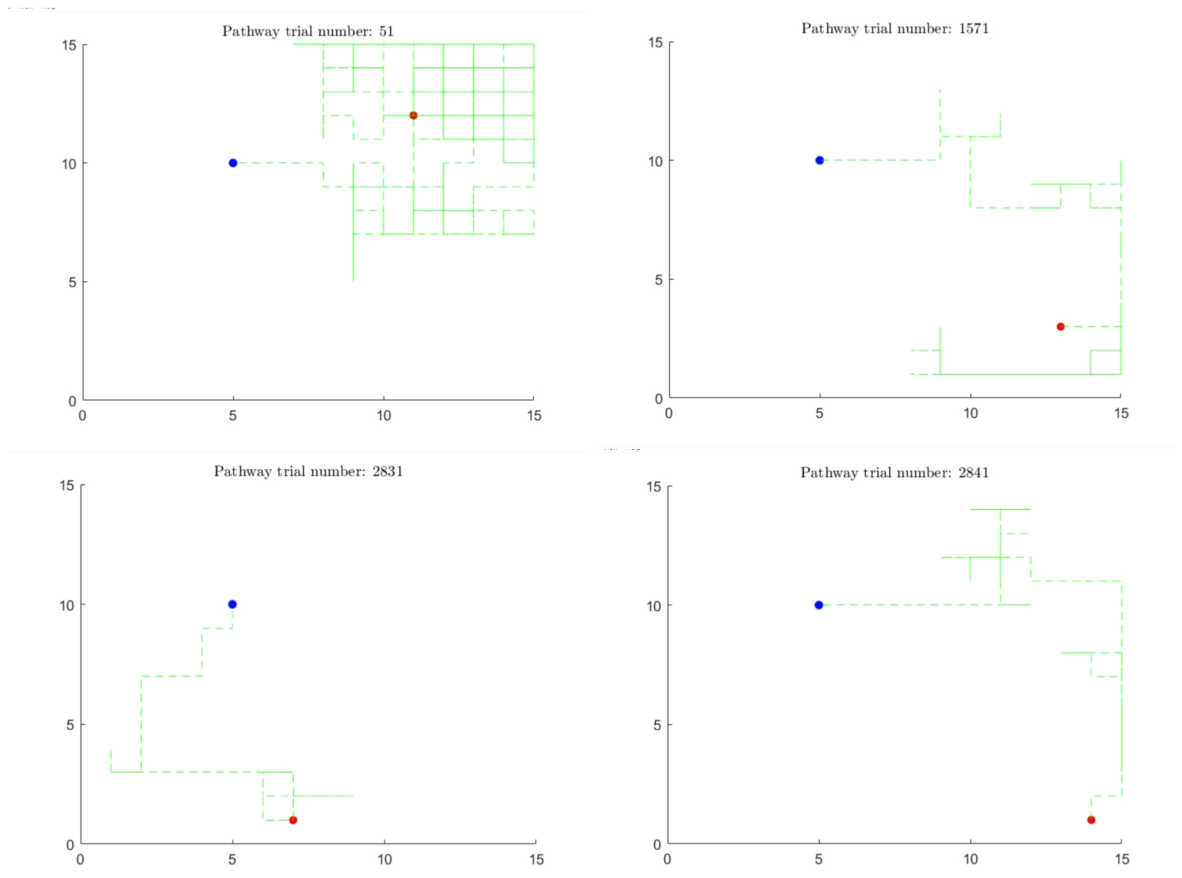

Figure 6: Pathway before and after training

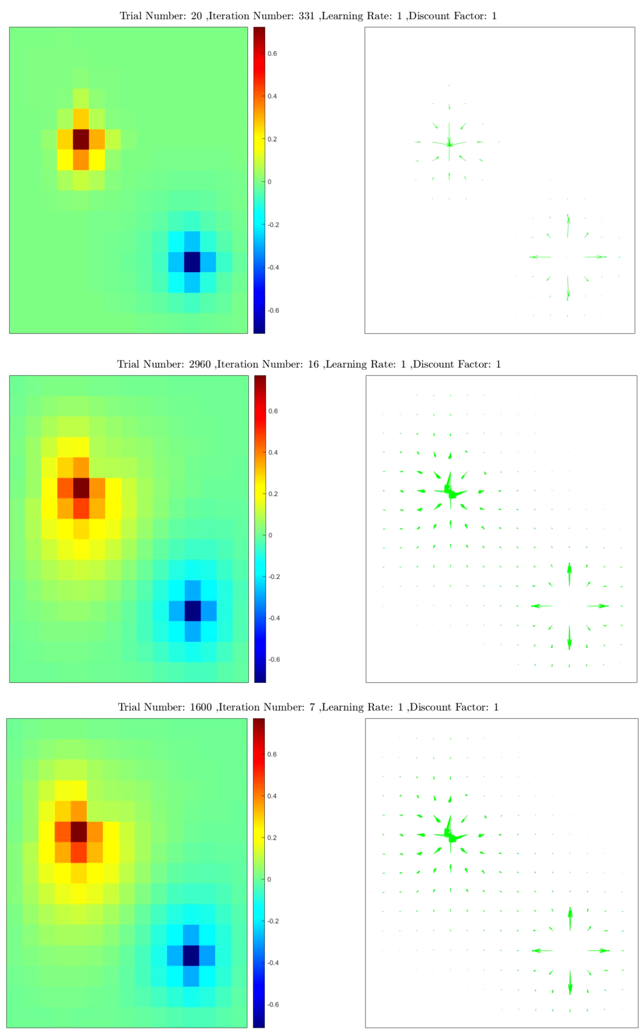

Figure 7: Gradient of learned value before and after training

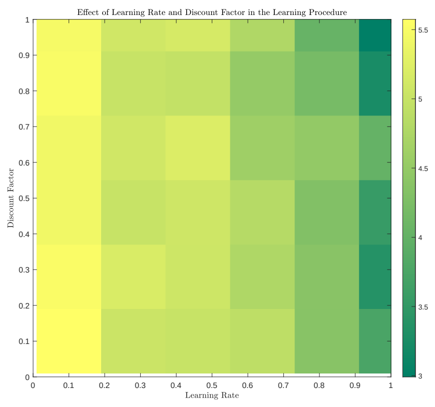

Figure 8: The effect of the learning rate (σ) and discount factor (γ)

Section 7: Greedy Policy (Extra)

As my first try, I update the next state base on the policy which gives me the maximum value. This algorithm is fast but it has no exploration and agent just follow a unique way. You can see the result here.