

Solving Frozen Lake Problem with Q-learning

Ke Yan

Abstract

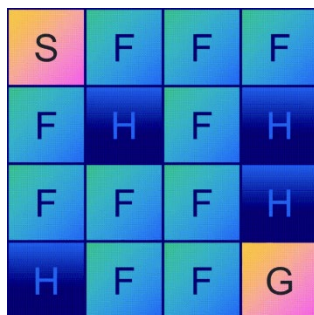
Q-learning is a model-free reinforcement algorithm to learn the value of an action in a particular state. We implemented q-learning algorithm to solve a 4x4 frozen lake and a complex 8x8 frozen lake. We found that it is important to balance exploration and exploitation during training. Specifically, among methods of exploration-exploitation tradeoff, exponential decay performs better than linear decay. It can be used to speed up the training process.

Introduction

Reinforcement Learning is the science of making optimal decisions using experiences. It has five elements: agent, environment, states, actions and rewards. The process of Reinforcement Learning involves these simple steps: Observation of the environment, Deciding how to act using some strategy, Acting accordingly, Receiving a reward or penalty, Learning from the experiences and refining our strategy and Iterate until an optimal strategy is found. Q-learning is a model-free reinforcement algorithm to learn the value of an action in a particular state. Usually, the environment is divided into cells, each cell has a reward. The agent has four actions: left, down, right and up. Q-learning maintains a q-table, which is updated every episode, and the agent will choose actions based on such a q-table.

We plan to implement q-learning algorithm to solve the frozen lake problem in 4x4 grids and expand it to 8x8 grids. In addition, we are going to discuss the importance of exploration-exploitation tradeoff. Specifically, we will compare linear and exponential methods in exploration-exploitation tradeoff.

Method



S	F	F	F
F	H	F	H
F	F	F	H
H	F	F	G

Fig 1. 4x4 Frozen Lake

Firstly, we construct the environment as a 4x4 frozen lake. S is the start point, G is the goal, F means frozen, which is safe area and H means hole. We are going to use q-learning algorithm to find an optimal path from Start to Goal, without falling into Hole. Since the 4x4 grids represent 16 states and the agent has four actions for each cell, we should maintain a 16x4 q-

table which consists of $Q(s_t, a_t)$ values. All $Q(s_t, a_t)$ values are set to be zero at the beginning. The agent only gets a reward by 1 if it reaches the goal, then the values in q-table will be updated based on such equation:

$$Q_{new}(s_t, a_t) = Q(s_t, a_t) + \alpha \cdot (reward + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$$

Where,

- α is the learning rate with domain $[0,1]$, which is how much we should change the original $Q(s_t, a_t)$ value. If $\alpha = 0$, the value never changes, but if $\alpha = 1$, the value changes extremely fast. In our attempt, we define the learning rate as 0.5.
- γ is the discount factor with domain $[0,1]$, which determines how much the agent cares about future rewards compared to immediate ones. If $\gamma = 0$, the agent only focuses on immediate rewards, but if $\gamma = 1$, any potential future reward has the same value as current ones. For frozen lake, we set the discount factor as 0.9.

With the learning rate α and discount factor γ , the q-learning algorithm can already solve the frozen lake problem. However, it might not be the optimal solution. Notably, the agent always chooses the action with the highest value. The other actions with lower value will never be taken, which means their values will never be updated. This situation leads us to the exploration-exploitation tradeoff. Exploration means choosing the actions randomly, and exploitation means taking action with the highest value. On the one hand, if the agent only focuses on exploration, the training is pointless. On the other hand, if the agent only focuses on exploitation, it can never try new solutions.

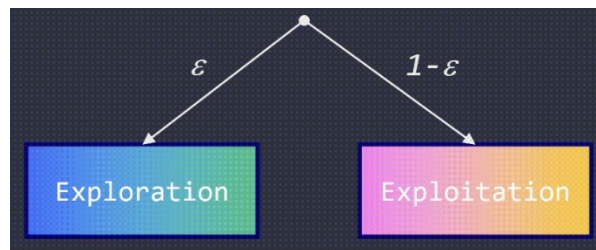


Figure 2. Epsilon Decay

We introduced epsilon decay to balance exploration and exploitation. In the beginning, we want the agent to explore the environment as much as possible. But as the training goes on, the agent will be more and more focused on exploitation. When the agent takes action, it has a probability of ϵ of choosing a random one, and a probability $1-\epsilon$ of choosing the one with the highest value. There are two kinds of epsilon decay methods one is linear decay, and another is exponential decay. Linear decay will decrease the epsilon with a fixed amount at the end of each episode, while exponential decay will multiply epsilon by a decay rate. The formula below shows how the decay rate is calculated for each episode:

$$\epsilon_{new} = \epsilon \cdot \lambda^{episode}$$

Where,

$$\lambda = \frac{total\ episodes}{\sqrt{0.001}}$$

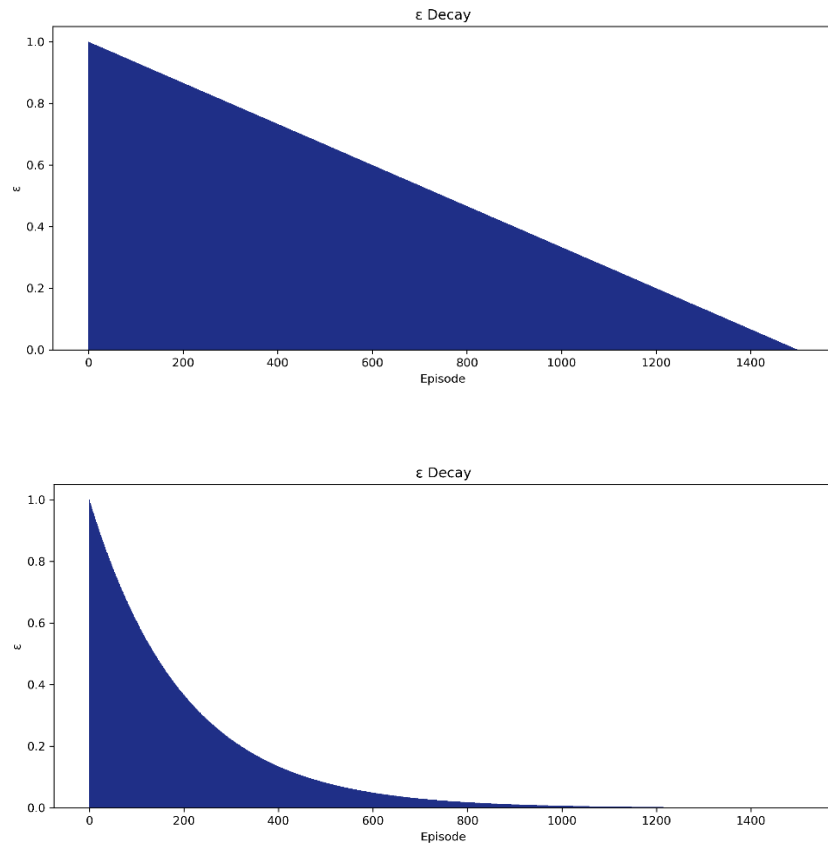


Figure 3. Linear Decay (Up) & Exponential Decay (Below)

Figure 3 shows how the epsilon changes with each episode for linear method and exponential method. Next, we will compare the results of Training without Exploration-Exploitation Tradeoff (i.e. without ϵ decay), Training with Linear Decay and Training with Exponential Decay.

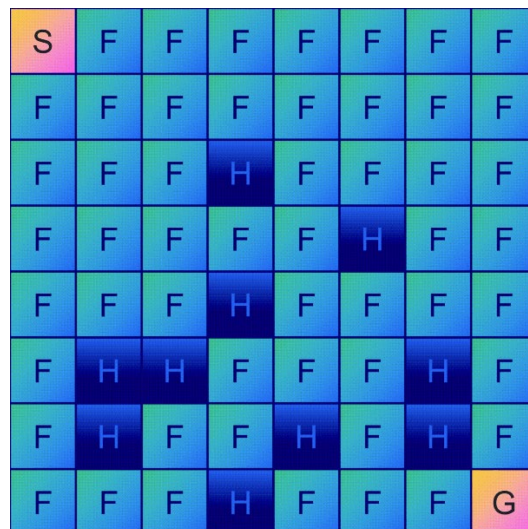
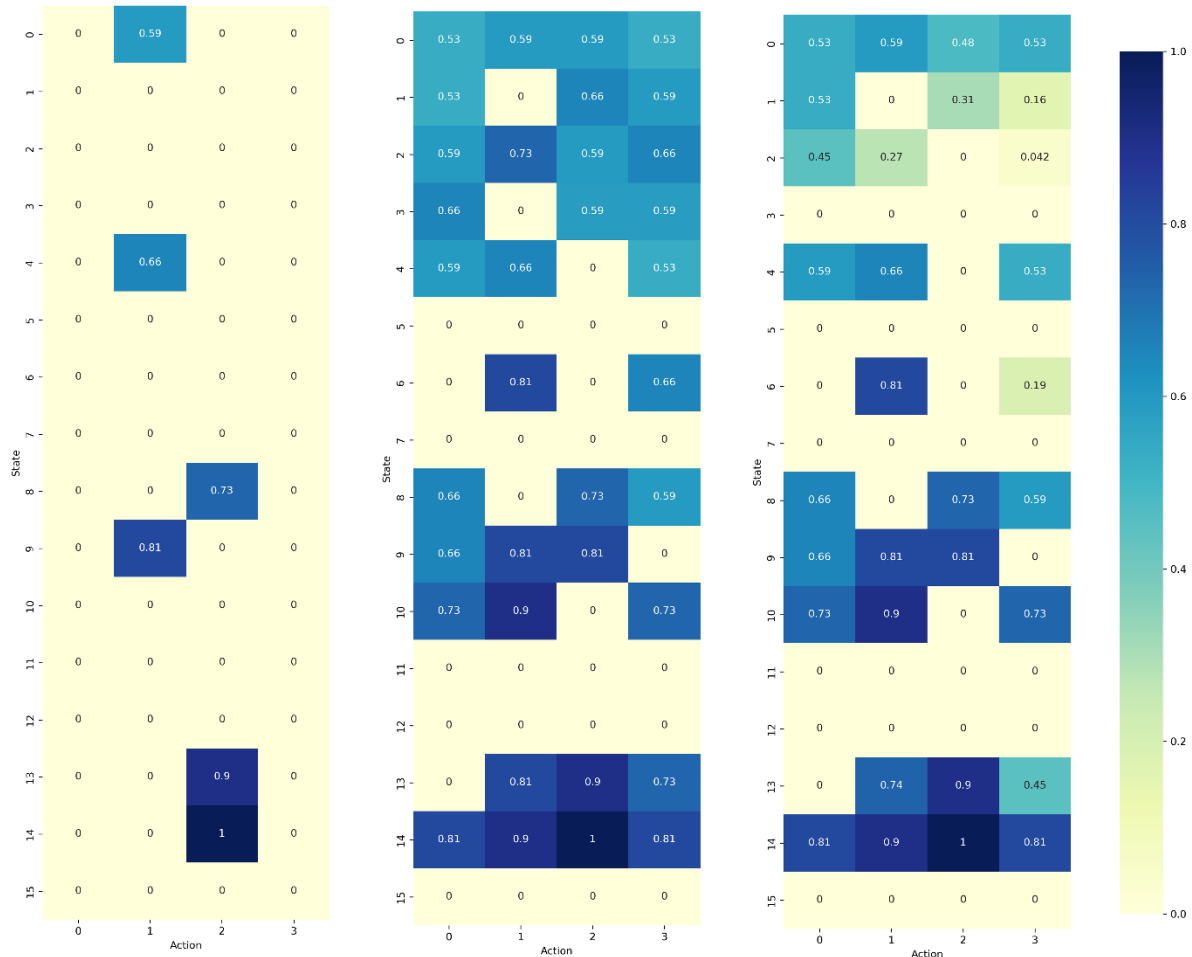


Figure 4. 8x8 Frozen Lake

Then, we will expand the frozen lake to a more complex environment. This is an 8x8 grid, which means the chance to reach the goal is much smaller than the chance for a 4x4 grid. At the same time, our q-table also expand to 64x4. Therefore, we plan to use 250,000 episodes to train our agent.

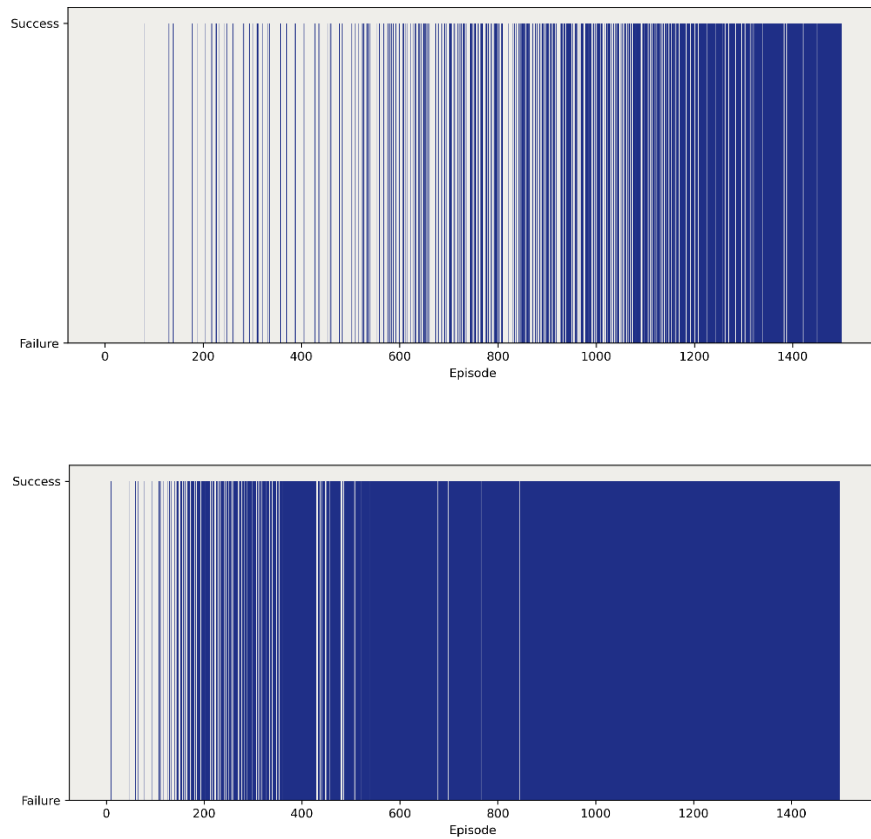
Results

For coding purposes, we map actions to integers. Here, 0 = Left, 1 = Down, 2 = Right, 3 = Up.



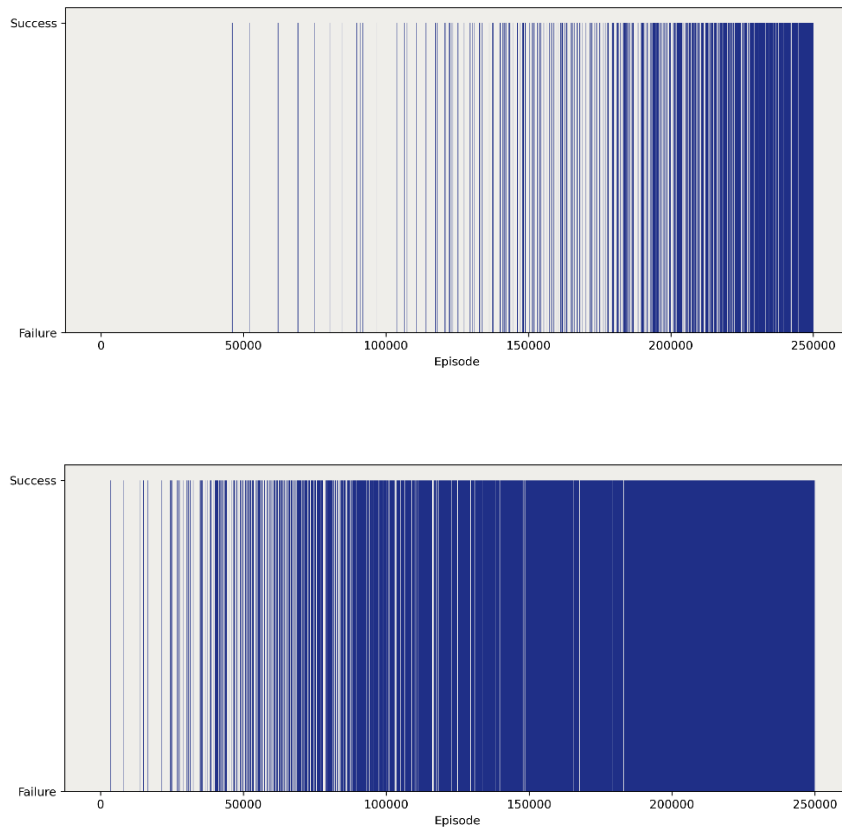
**Figure 5. Q-table for No Exploration-Exploitation Tradeoff (Left), Linear Decay (Mid), and Exponential Decay (Right).
4x4 Frozen Lake**

From the q-tables, we can tell that the introduction of exploration-exploitation tradeoff is remarkable. The q-table without exploration-exploitation tradeoff is very sparse. Both q-tables with linear decay and exponential decay show good exploration of the environment. Therefore, the exploration-exploitation tradeoff is important. It helps the agent understand the environment much better and have the ability to find multiple optimal solutions.



**Figure 6. Outcomes for Linear Decay (Up) and Exponential Decay (Below)
4x4 Frozen Lake**

Figure 6 shows the training outcomes for linear decay method and exponential decay method. Each solid blue line corresponds to a success, so we can see that both methods had a hard time finding the goal at the beginning. But as the training goes on, the exponential decay method converges much faster than linear decay, which means the exponential decay could reach even better results with smaller training episodes. Therefore, our conjecture is that we can use exponential decay to speed up the training process. To support our idea, we also compared linear decay and exponential decay in a complex 8x8 environment. When the training episodes are very large, the Q-table for linear decay and exponential decay are almost the same (shown in the appendix). The outcomes of each episode are shown in Figure 7. Not surprisingly, exponential decay still shows better performance. Actually, there is not enough related literature to support our idea. However, the result is very obvious. We guess what makes the exponential decay method perform better is that it decays faster in the middle of training. This makes the agent have a higher probability of exploitation. Perhaps we don't have to spend too much time exploring, and exponential decay method allows us to start exploitation earlier.

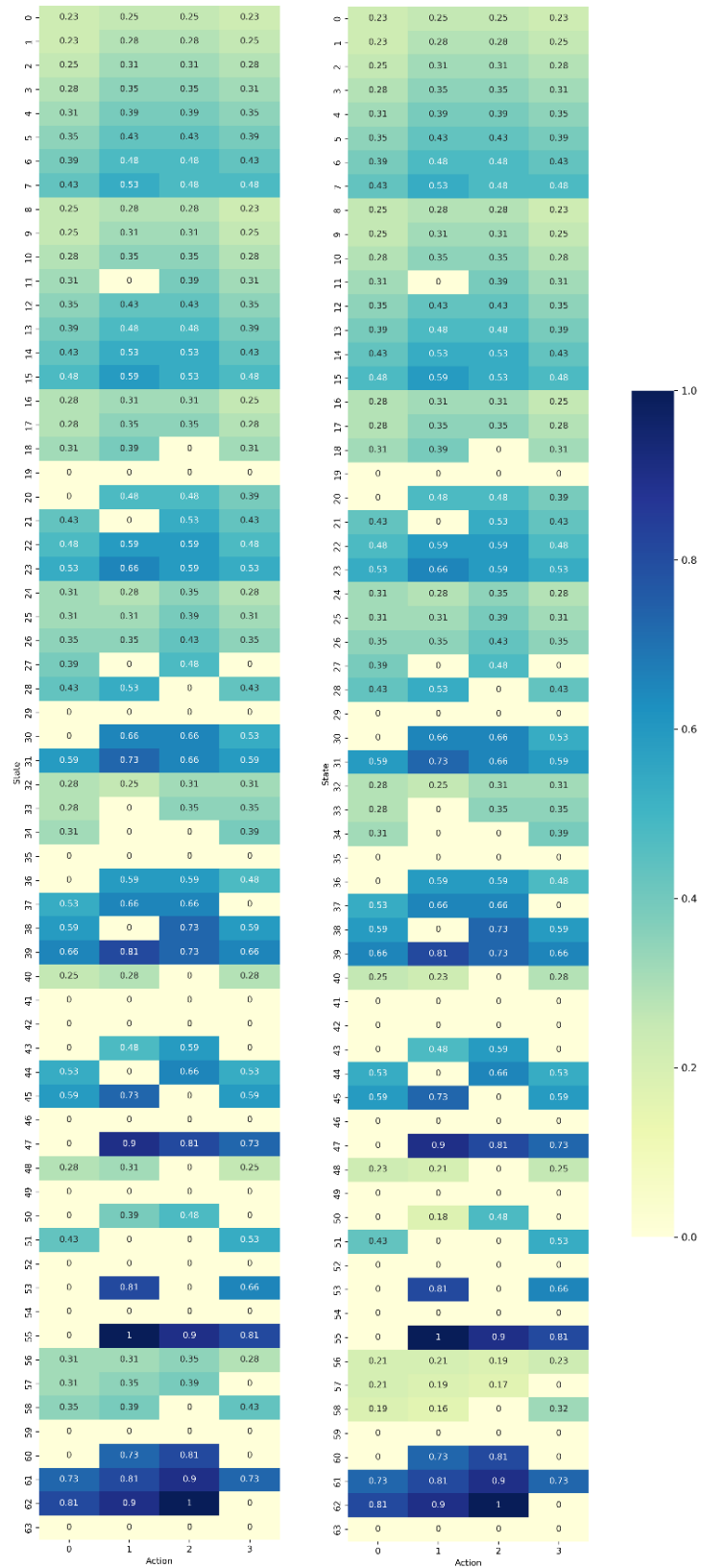


**Figure 6. Outcomes for Linear Decay (Up) and Exponential Decay (Below)
8x8 Frozen Lake**

Conclusion

We have implemented q-learning algorithm to solve the frozen lake problem in 4x4 grids and 8x8 grids. We found that exploration - exploitation tradeoff is important. It helps the agent understand the environment much better and is able to find multiple optimal solutions. Specifically, we compared linear and exponential methods in exploration-exploitation tradeoff. The results show that exponential decay performs better than linear decay. It can be used to speed up the training process.

Appendix



**Q-table for Linear Decay (Left) and Exponential Decay (Right).
8x8 Frozen Lake**