"Reinforcement Learning Assignment: FrozenLake"
Agents, Multi-Agent Systems and Reinforcement Learning
MSc, Artificial Intelligence, 2025

Michael Rice

March 15, 2025

1 Introduction

This report will discuss the results from the second assignment of the CT5134 module. This assignment tackles Reinforcement Learning (RL) via the Q-Learning algorithm on the Frozen Lake toy problem, part of the famous 'Grid World' category of problems. Each of the results below are averaged over a ten run window.

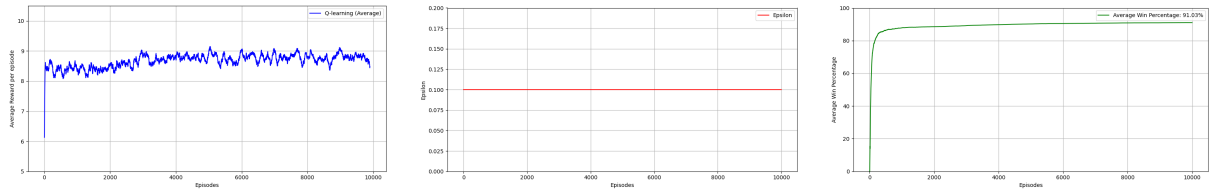
2 Hyperparameter Set 1 - Alpha(0.5), Gamma(0.9), Epsilon(0.1)

The first set of hyperparameters to test were provided in the assignment and are as stated above. This test can be viewed as being designed to be a baseline from which to compare performance to later in the assignment. Below are three figures, each representing a different metric of the system's performance/learning.

Firstly, the leftmost graph below represents the algorithm's reward earned per episode. This can be helpful in understanding the algorithms iterative improvements over the episodes. As is evident here, the system approaches convergence relatively quickly, then proceeds to oscillate around the level of convergence for the rest of the episodes.

Secondly, the plot in the center shows the epsilon value over the length of the experiment. This value is kept constant here but will be varied, and therefore rendering this plot more relevant, further into these experiments.

Finally, the rightmost graph depicts the percentage of episodes, averaged again, that reach the 'Win' state. Again, as in the first graph, this plot shows early convergence followed by stagnation, reaching a peak win rate of just over **91%**.



(a) Run Averaged Reward Per Episode (b) Run Averaged Epsilon (c) Run Averaged Win %

Figure 1: Metrics for Hyperparameter Set 1

The figure below shows the Q-value results from this first foray into the Frozen Lake problem. As can be seen, each of the predefined 'hole' states, as well as the 'win' state, end the process with a Q-value of 0, as they are terminal states. Other states that are hole adjacent tend to end with negative Q-values due to the fact that they do not explicitly provide a worse reward by moving to that state, but choosing to move to them negatively effects overall episodic reward but placing us in proximity to such negative termini. Remaining states increase in value the further right and down they are, as this means they are closer to the goal state, and thus a substantial reward. There are also other notable artifacts that occur due to the structure of this problem, such as the fact that non-terminal states are more likely to have a higher reward if they are above the diagonal than below. This is likely a result of the placement of the hole states, meaning being below the diagonal is generally less likely to result in an optimal path.

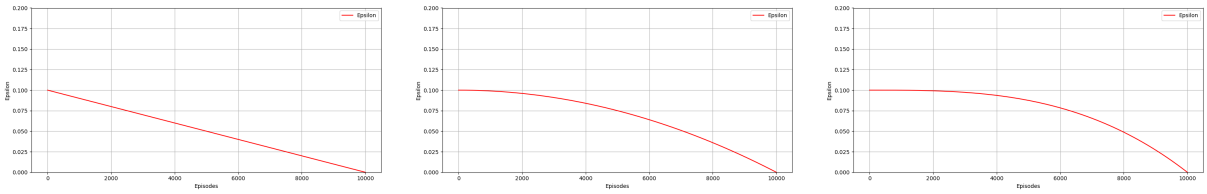
-0.434	0.629	1.81	3.122	4.58
0.0	1.806	3.122	0.0	6.2
1.272	3.122	4.58	6.2	8.0
-1.787	0.0	5.79	8.0	10.0
-2.309	-2.279	0.0	9.982	0.0

Figure 2: Action Value Estimates for Each State

3 Same Hyperparameter Set w/ Epsilon Decay - Linear, Quadratic and Cubic

In this portion of the experiment, the starting hyperparameters remained the exact same as the section previous. However, this time throughout each run, the epsilon value would be decayed in order to alter the exploration/exploitation balance to aid in the discovery of an optimal path through the grid.

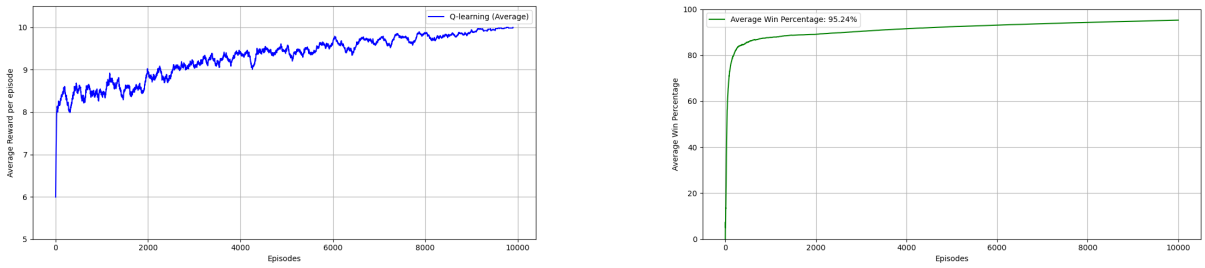
Three different methods of epsilon decay were tried during this section of the experiment; Linear, Quadratic and Cubic, each starting at a value of 0.1 as before, and ending each run at 0. As can be seen in the plots below, the Linear method retains a constant level of decay throughout each run, while both the Quadratic and Cubic methods keep larger values of epsilon for longer. This means both of these non-linear methods are more explorative at the beginning of the runs, with a faster decay at the end (exploitation).



(a) Linear Decay (b) Quadratic Decay (c) Cubic Decay

Figure 3: Epsilon Decay Schedules

The best performing method of epsilon decay was Linear, perhaps indicating that this problem favored a less explorative approach, as linear was the least explorative method tested. The reward per episode and win percentage graphs are shown below, with a clear improvement in the trend of the reward per episode, in comparison to the initial test. The final win percentage was also improved from the first test, to **95.24%**, a gain of over 4%.



(a) Reward Per Episode (b) Win Percentage

Figure 4: Performance with Linear Decay

Finally, for this section, the Q-Values exhibit most of the same characteristics as the Q-Values in the initial experiment, with one major notable change. As mentioned in the first section, the Q-Values above the diagonal tended to be higher than those below due to the presence of more 'hole' states below the diagonal. However, due to the increased exploitation by the linear epsilon delay versus a constant epsilon in this attempt, positive values appear to be more even, aside from the obvious exception of the bottom left of the state space, which is a trap when it comes to improving episodic rewards.

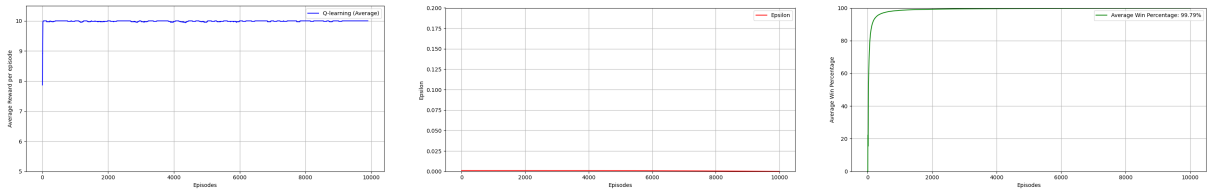
-0.434	0.629	1.81	2.673	4.485
0.0	1.701	3.122	0.0	6.2
1.341	3.122	4.58	6.2	8.0
-1.813	0.0	6.2	8.0	10.0
-2.151	-1.908	0.0	10.0	0.0

Figure 5: Action Value Estimates

4 Hyperparameter Set 3 - Alpha(0.7), Gamma(0.95), Epsilon(0.001)

Finally, using the learnings taken from the previous experiments, a new set of hyperparameters were to be designed. As previously mentioned, these previous tests pointed to increased exploitation improving the algorithm's performance, therefore this was the chosen direction when selecting these hyperparameters. After a number of tests, values of Alpha=0.7, Gamma=0.95 and Epsilon=0.001. This approach maximised exploitation, with little to no randomness/exploration.

As is evident in the reward per episode and win percentage plots below, the algorithm converges towards the optimal solution much quicker with these parameters. Furthermore, the overall percentage of episodes in which the win state was reached was again improved, reaching **99.79%**, another sizeable improvement of over 4%.



(a) Run Averaged Reward (b) Epsilon Decay (c) Win Percentage

Figure 6: Metrics for Custom Hyperparameters

When comparing the eventual Q-Values for these hyperparameters to the previous two experiments, it is evident that several changes took place. Firstly, these parameters punish the bottom left corner of the state space even more heavily than before, this time including the (2,0) state, which held a positive value for both previous sets of arguments. Secondly, the top right corner of the state space is now also actively discouraged, without having any hole states, which had not occurred previously. This shows that the algorithm is learning to choose the quickest possible path through the world, considering both hole states and overall step count. The hole state of (1,3) is likely responsible for this, as having to go around it increases overall step count, thus lowering reward, which leads to the states which would be involved in travelling around this hole state to be discouraged. Finally, the chosen path is much more evident in this section, with the algorithm clearly favouring repetitions of the (Right,Down) action pair until the goal state is reached. This is the reason why state (4,3) has a lower value than (3,4), despite both being win state adjacent. As a final note, none of the epsilon decline methods had any major effect on the result of this experiment, meaning any of them, or none, could be used.

Average Q-values across all runs:

0.95	1.226	2.545	-0.461	-0.46
0.0	3.177	4.435	0.0	4.945
-3.012	3.709	5.721	6.968	8.5
-3.111	0.0	7.075	8.5	10.0
-3.04	-3.263	0.0	9.33	0.0

Figure 7: Action Value Estimates