"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 in the Frozen Lake toy problem, part of the famous 'Grid World' category of problems. Each of the results below are averaged over a 10 run window.

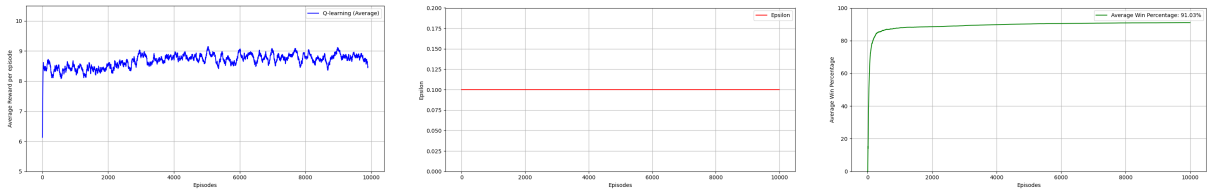
2 Hyperparameter Set 1 - Alpha(0.5), Gamma(0.9), Epsilon(0.1)

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 % over Episodes

Figure 1: Metrics for Hyperparameter Set 1

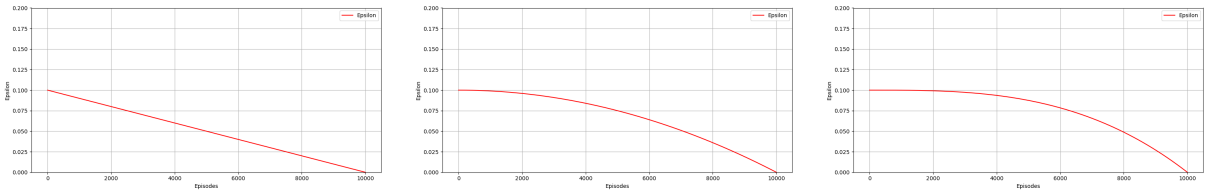
3 Same Hyperparameter Set w/ Epsilon Decay - Linear, Quadratic and Cubic

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).

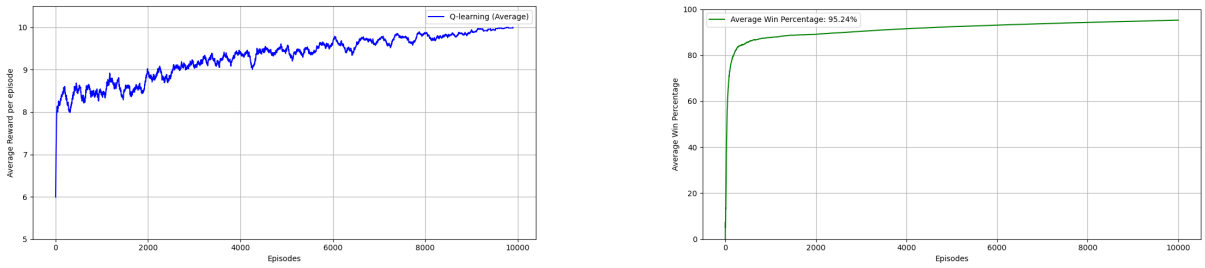
-0.434	0.629	1.81	3.122	4.58
0.0	1.806	3.122	0.0	6.2
1.272	3.122	4.58	6.2	8.0
-1.787	0.0	5.79	8.0	10.0
-2.309	-2.279	0.0	9.982	0.0

Figure 2: Action Value Estimates for Each State



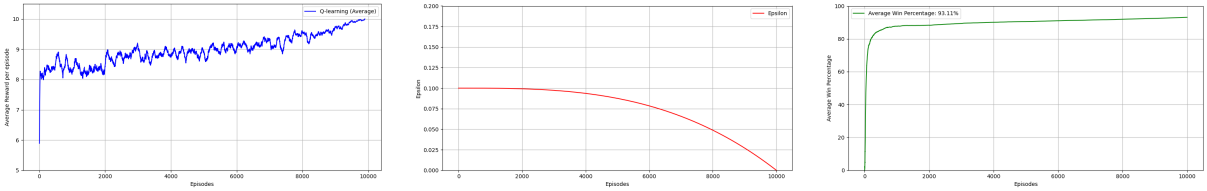
(a) Run Averaged Epsilon Decay (Linear) (b) Run Averaged Epsilon Decay (Quadratic) (c) Run Averaged Epsilon Decay (Cubic)

Figure 3: Metrics for Hyperparameter Set 1 w/ Epsilon Decay



(a) Run Averaged Reward Per Episode (b) Run Averaged Win % over Episodes

Figure 4: Metrics for Hyperparameter Set 1 w/ Epsilon Decay (Quadratic)

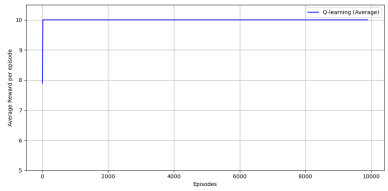


(a) Run Averaged Reward Per Episode (b) Run Averaged Epsilon Decay (c) Run Averaged Win % over Episodes

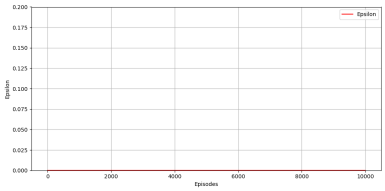
Figure 5: Metrics for Hyperparameter Set 1 w/ Epsilon Decay (Cubic)

4 Hyperparameter Set 3, My Own Choices - Alpha(0.7), Gamma(0.95), Epsilon(0.0)

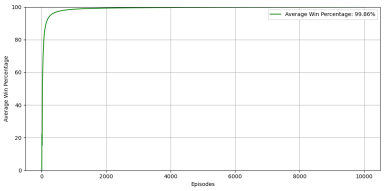
Finally, in this section the hyperparameters used were of my own design.



(a) Run Averaged Reward Per Episode



(b) Run Averaged Epsilon Decay



(c) Run Averaged Win % over Episodes

Figure 6: Metrics for Hyperparameter Set 3

Average Q-values across all runs:

0.95	2.053	3.213	-2.732	-0.464
0.0	-3.12	4.435	0.0	4.945
-3.017	1.439	5.721	3.193	8.311
-3.111	0.0	7.075	8.5	10.0
-3.04	-3.263	0.0	7.0	0.0

Figure 7: Action Value Estimates for Each State