Q-Learning Agent for Taxi-v3: Hyperparameter Experimentation Report

Name: Aayush Rautela & Cagla Kuleasan (Group 29)

Lab Variant: 4 (Taxi-v3)

1. Introduction

In this report, we detail our systematic experimentation with hyperparameters for a Q-Learning agent designed to solve the "Taxi-v3" environment. Our goal with these experiments was to understand the impact of different learning rates (α), discount factors (γ), epsilon decay rates, and Q-table initialization strategies on the agent's learning performance and to identify configurations that yield better results.

The Q-Learning algorithm, with its update rule, and an epsilon-greedy exploration strategy, formed the basis of our experiments.

2. Methodology

We conducted a series of experiments using the `q_learning_taxi_experimenter.py` script. Each experiment involved training our agent for **10,000** episodes (as per `BASE_TOTAL_EPISODES` in the script) and then evaluating its learned policy over **5** episodes with exploration turned off. We performed the following sets of experiments:

1. **Varying Learning Rates (α):** We tested values **[0.01, 0.05, 0.1, 0.3, 0.5]** while keeping other baseline parameters constant (γ=**0.99**, decay=**0.0005**, Q-init="zeros").
2. **Varying Discount Factors (γ):** We tested values **[0.9, 0.95, 0.99, 0.999]** while keeping other baseline parameters constant (α=**0.1**, decay=**0.0005**, Q-init="zeros").
3. **Varying Epsilon Decay Rates:** We tested values **[0.0001, 0.0005, 0.001, 0.005]** while keeping other baseline parameters constant (α=**0.1**, γ=**0.99**, Q-init="zeros").

4. **Varying Q-Table Initialization:** We tested strategies **["zeros", "random"]** while keeping other baseline parameters constant (α=**0.1**, γ=**0.99**, decay=**0.0005**).

Performance was primarily assessed by the average reward our agent achieved during the evaluation phase and the average number of steps taken. We generated training plots (average reward and episode length vs. episodes) for each run.

Baseline Parameters (unless otherwise specified for an experiment):

- Learning Rate (α): **0.1**
- Discount Factor (γ): **0.99**
- Epsilon Decay Rate: **0.0005**
- Q-Table Initialization: "zeros"
- Total Training Episodes: **10,000**

3. Experimental Results and Analysis

The following sections detail the outcomes for each set of our experiments, based on the provided summary table. The "Eval Rew" (average evaluation reward) is the primary metric we used for identifying the best performer in each category.

3.1. Impact of Learning Rate (α)

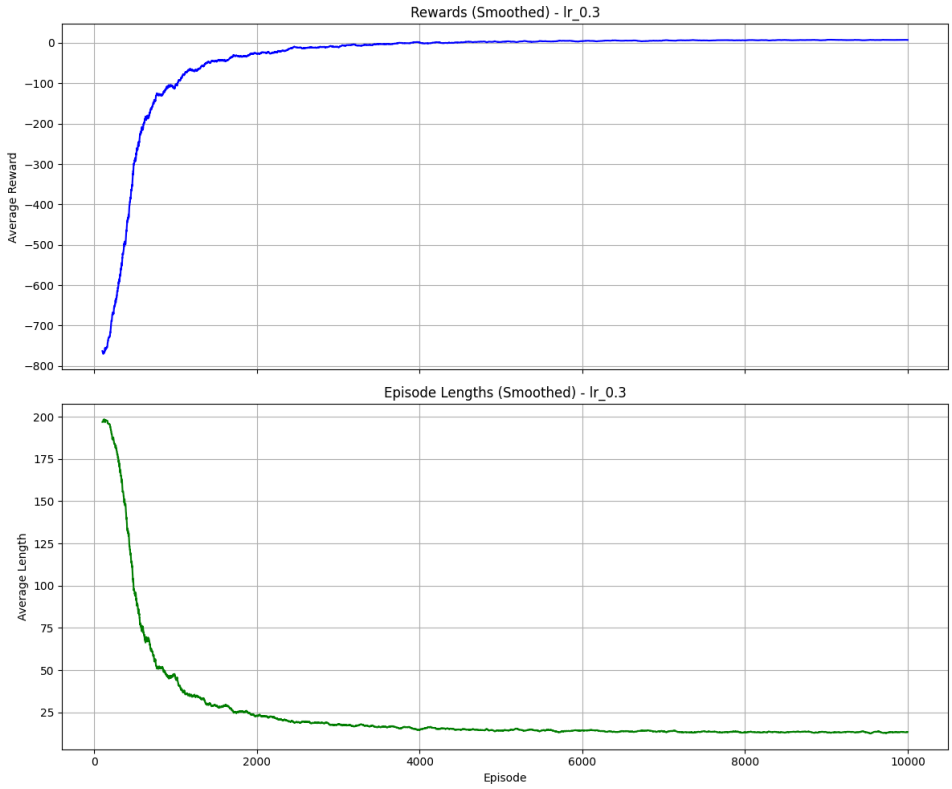
The learning rate determines how much new information overrides old information in the Q-table.

Tested Values: **0.01**, **0.05**, **0.1**, **0.3**, **0.5**

Observations from Summary Table:

- We observed that a very low LR (**0.01**) resulted in poor performance (Eval Rew: **-75.20**).
- Performance significantly improved with LRs of **0.05** and **0.1**.
- The best evaluation reward (**8.20**) in this set was achieved with LR = **0.3**.
- We noted that LR = **0.5** showed a slight decrease in performance compared to **0.3**.

Plot for Best Performing Learning Rate (α = **0.3**):



3.2. Impact of Discount Factor (γ)

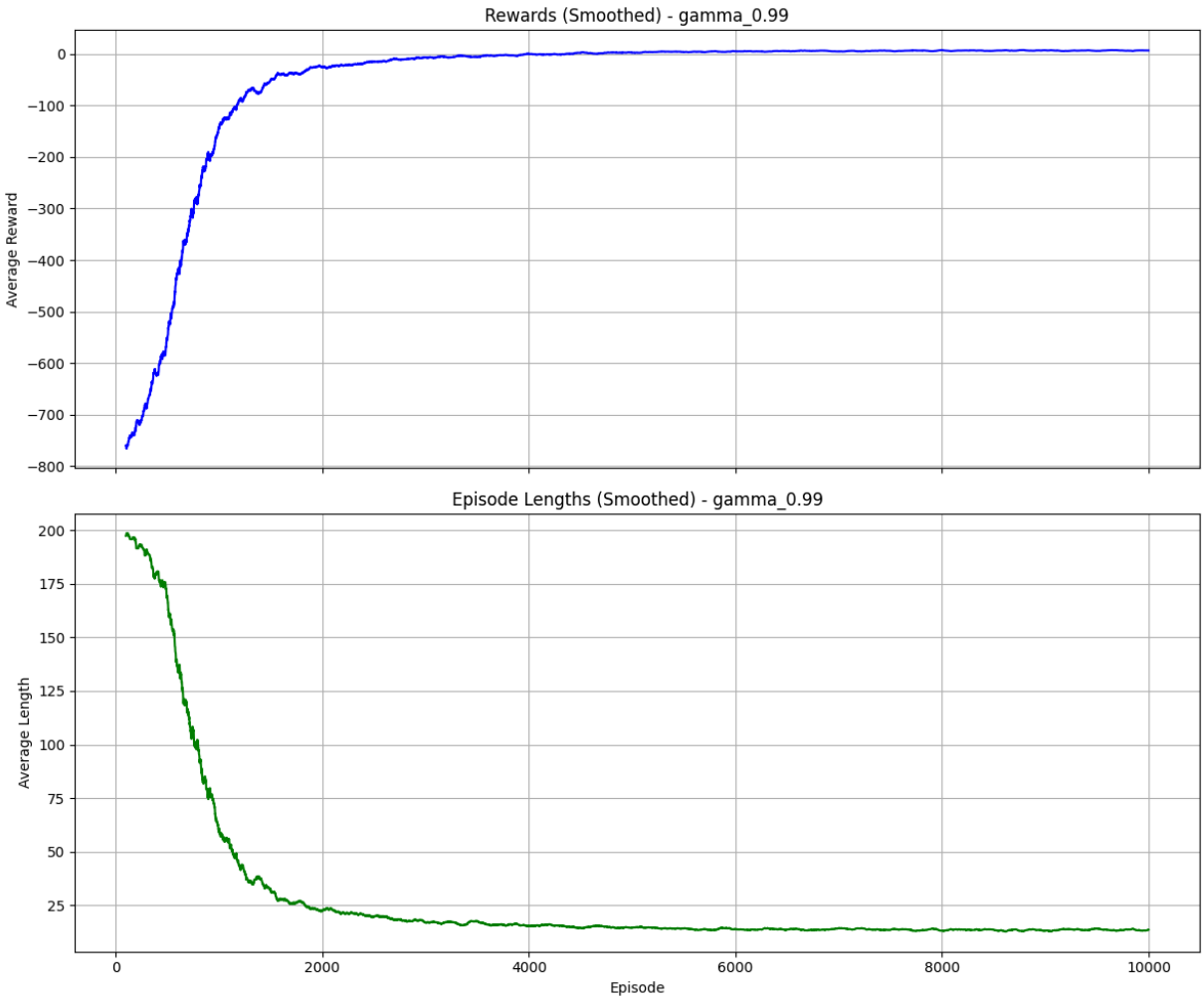
The discount factor determines the importance of future rewards. A value closer to **1** gives more weight to long-term rewards.

Tested Values: **0.9**, **0.95**, **0.99**, **0.999**

Observations from Summary Table:

- Generally, we found that higher gamma values led to better evaluation rewards.
- The best evaluation reward (**9.00**) in this set was achieved with γ = **0.99**.
- We observed that γ = **0.999** performed well but slightly less than **0.99** in this specific run.

Plot for Best Performing Discount Factor (γ = **0.99**):


Rewards (Smoothed) - gamma_0.99
Average Reward
0
−100
−200
−300
−400
−500
−600
−700
−800
Episode Lengths (Smoothed) - gamma_0.99
Average Length
200
175
150
125
100
75
50
25
0
2000
4000
6000
8000
10000
Episode

3.3. Impact of Epsilon Decay Rate

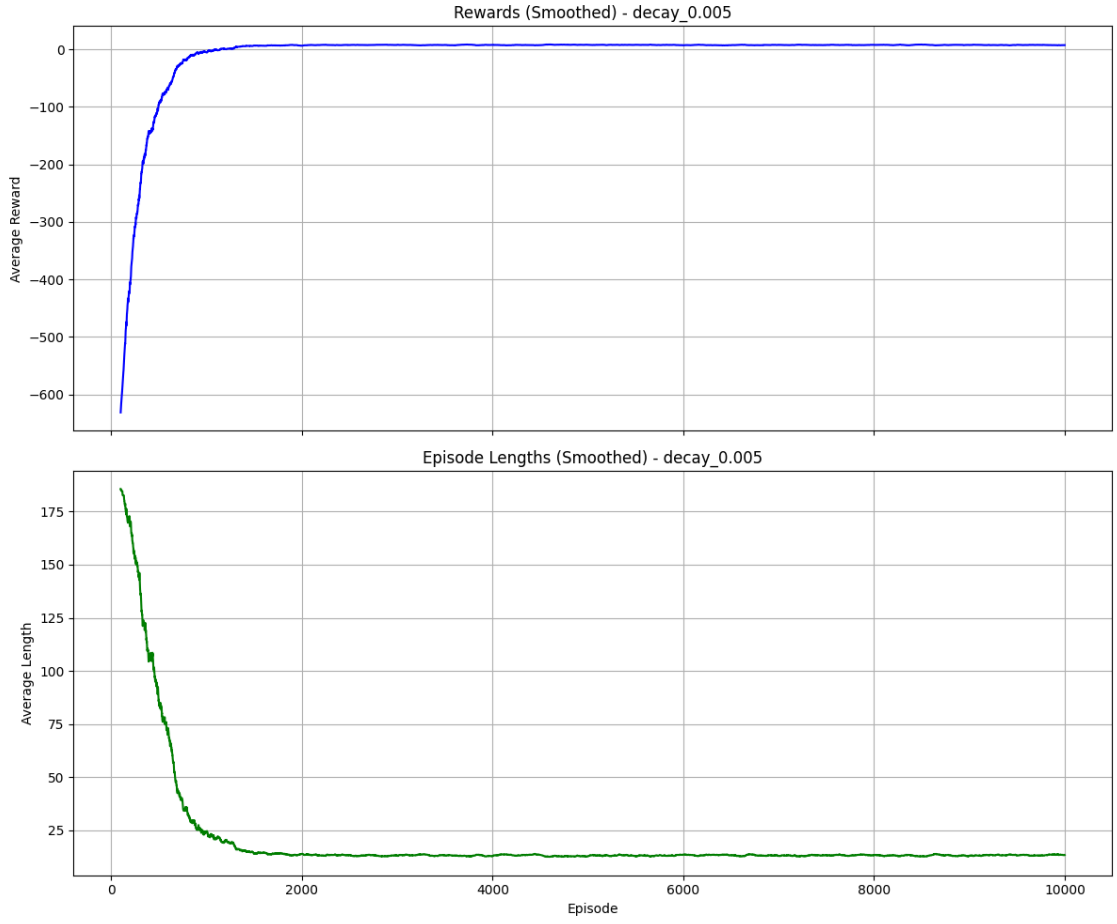
The epsilon decay rate controls how quickly the agent shifts from exploration to exploitation.

Tested Values: **0.0001**, **0.0005**, **0.001**, **0.005**

Observations from Summary Table:

- We noted that a very slow decay (**0.0001**) showed poor average training reward but a surprisingly good evaluation reward (**8.20**), suggesting late learning.
- Increasing the decay rate (faster transition to exploitation) generally improved evaluation performance in this set.
- The best evaluation reward (**10.00**) and fewest steps (**11.00**) across all our experiments were achieved with a decay rate = **0.005**.

Plot for Best Performing Epsilon Decay Rate (decay = **0.005**):



3.4. Impact of Q-Table Initialization

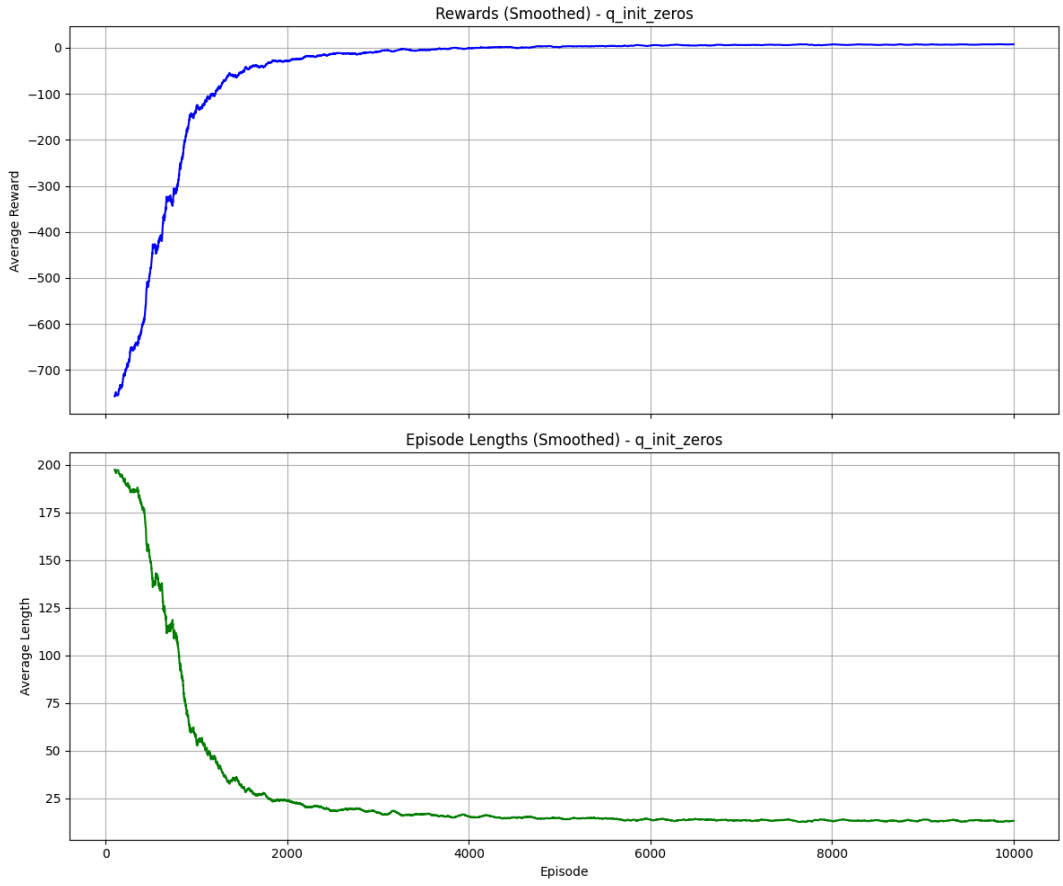
This experiment compared initializing the Q-table with all zeros versus small random values.

Tested Strategies: "zeros", "random"

Observations from Summary Table:

- We found that initializing with "zeros" yielded a slightly better evaluation reward (**6.80**) compared to "random" (**6.40**) in this specific experimental setup with baseline parameters.
- The difference was not dramatic, suggesting that for this problem and with sufficient training, the initial Q-table values (at least between these two strategies) might not be the most critical factor when other hyperparameters are reasonably set.

Plot for Q-Table Initialization with Zeros (best in this pair):
Bbaseline LR=**0.1**, gamma=**0.99**, decay=**0.0005**. The plot name would reflect the experiment name `q_init_zeros`)



4. Overall Discussion and Summary

Based on the experiments conducted and the provided summary table:

- **Optimal Learning Rate:** A learning rate in the range of **0.1** to **0.3** appears to be effective for the Taxi-v3 problem with 10,000 training episodes. Too low (**0.01**) prevents effective learning, while very high rates (though not extensively tested beyond **0.5**) could risk instability.
- **Importance of Future Rewards:** A high discount factor (**γ = 0.99**) consistently performed well in our tests, emphasizing the need for the agent to consider long-term consequences in this environment.
- **Exploration-Exploitation Balance:** The epsilon decay rate had a significant impact in our experiments. A faster decay rate (**0.005**) led to the best overall performance in this set of experiments, suggesting that quickly transitioning to exploiting learned knowledge was beneficial within the 10,000 episode limit. The run with a very slow decay (**0.0001**) showed an interesting discrepancy between poor training average and good evaluation, highlighting that average training metrics might not always tell the full story if learning occurs very late.
- **Q-Table Initialization:** While initializing with "zeros" performed slightly better than small random numbers in the specific baseline configuration we tested, the impact of initialization seemed less critical than other hyperparameters like learning rate or decay rate.

The best overall performing configuration from our summary table was:

- Learning Rate (α): **0.1**
- Discount Factor (γ): **0.99**
- Epsilon Decay Rate: **0.005**
- Q-Table Initialization: "zeros" (as per the `decay_0.005` run's baseline)

This configuration achieved an average evaluation reward of **10.00** with an average of **11.00** steps.

5. Conclusion

Our hyperparameter tuning process revealed that the performance of the Q-Learning agent on the Taxi-v3 environment is highly sensitive to the choice of learning rate, discount factor, and particularly the epsilon decay strategy.

For the configurations tested over 10,000 episodes:

- A learning rate of **0.1-0.3**, a discount factor of **0.99**, and a relatively fast epsilon decay rate of **0.005** yielded the most promising results.
- Q-table initialization with zeros was slightly favored over small random numbers but was not as impactful as other parameters.

These experiments underscore the iterative nature of developing effective reinforcement learning agents. The optimal parameters we identified from this set provide a strong starting point for this environment, but we believe further fine-tuning or testing with more episodes could potentially yield even better performance. The plots for each experimental run (especially the best performers in each category) provide visual confirmation of these learning trends.