CSCN8020 - Reinforcement Learning Programming Assignment 2

Introduction:
In this assignment, we implemented the Q-Learning algorithm using the Taxi-v3 environment from OpenAI Gymnasium. The objective was to train an agent (taxi) that can efficiently navigate, pick up, and drop off passengers in a small grid world. The learning was guided by a reward system encouraging successful deliveries and discouraging wrong or inefficient moves.

Problem Statement:
Implement the Q-Learning algorithm on the Taxi environment and report metrics for:
1. Total episodes
2. Total steps taken per episode
3. Average return per episode

The algorithm should be trained under multiple configurations to test the impact of different hyperparameters.

Hyperparameters:
Initial Values:
- Learning Rate (α): 0.1
- Discount Factor (γ): 0.9
- Exploration Rate (ε): 0.1

Experimental Values:
- α = [0.01, 0.001, 0.2]
- γ = [0.2, 0.3]

Each configuration was tested independently to observe its impact on the agent’s performance.

Observations & Discussion:
- Lower α values (0.01, 0.001) made learning very slow because the agent updates the Q-table cautiously.
- Very high α (0.2) caused unstable updates as the taxi overreacted to single experiences.
- Lower γ (0.2) made the agent focus only on short-term rewards, ignoring long-term success.
- Higher γ (0.9) encouraged the agent to plan long-term and complete deliveries more efficiently.
- The combination α=0.1 and γ=0.9 produced the best average reward and the fewest steps per episode.

Conclusion:
Through experimentation, we learned how different hyperparameters affect Q-Learning performance. The final trained agent demonstrated optimal decision-making by balancing exploration and exploitation. This exercise reflects real-world reinforcement learning used in autonomous systems like self-driving cars, where agents learn to act intelligently through repeated interaction with their environment.