Phase II Report: Application of Reinforcement Learning

Team Members:
Mohamed Tarek Mohamed Ali: s-mohamed-tarek@zewailcity.edu.eg :202201647
Abdalrahman Khaled : s-abdalrahman.abdalrahman@zewailcity.edu.eg : 202201655
Ahmed Gamal Abdelfattah : s-ahmed.sharaf@zewailcity.edu.eg : 202201638
Habiba Ayman Amin : s-habiba.ibrahim@zewailcity.edu.eg : 202202088

1. Introduction

This report outlines the implementation and application of Q-learning to solve a reinforcement learning task in a grid environment. The problem involves navigating an agent from various starting points to a predefined goal state within a 40x40 grid. The agent is trained to optimize its path and reward accumulation. This report provides details about the design, Q-learning setup, results, and analysis of the agent's performance.

---

2. Problem Design

State Space Representation :

- **State Definition**: The environment is represented as a 40x40 grid. Each cell corresponds to a unique state defined by its position `(x, y)` on the grid.
- **Initial States**: The agent starts at one of the specified starting points: `(0, 0)`, `(10, 10)`, `(30, 30)`, or `(39, 39)`.
- **Goal State**: The goal is located at `(39, 39)`.

Action Set:

The agent has four possible actions:

- **Up**: Move one step upward.
- **Down**: Move one step downward.
- **Left**: Move one step left.
- **Right**: Move one step right.

Reward Function:

- **Goal Reward**: The agent receives a reward of `100` upon reaching the goal state.
- **Step Penalty**: A penalty of `-1` is applied for each step taken to encourage faster goal-reaching.

3. Q-Learning Setup

Algorithm Configuration:

- **Learning Rate (α)**: `0.1` – This controls the rate at which the agent updates its Q-values.
- **Discount Factor (γ)**: `0.99` – Encourages long-term rewards while factoring in immediate rewards.
- **Exploration-Exploitation Tradeoff**:
  - **Epsilon-Greedy Strategy**: The agent starts with `ε = 1.0` (full exploration) and gradually decays to `ε = 0.01` to focus on exploiting learned policies in later episodes.

Q-Table Initialization:

The Q-table is initialized to zeros for all state-action pairs, representing no prior knowledge of the environment.

Training Process:

- **Number of Episodes**: The agent is trained for `100,000` episodes.
- During each episode:
  1. The agent begins at a starting position.
  2. It selects an action using the epsilon-greedy strategy.
  3. It updates its Q-values using the formula:
     - q_table[state[0], state[1], action_index] = q_table[state[0], state[1], action_index] + alpha * (reward + gamma * max_next_q - q_table[state[0], state[1], action_index])
  4. Exploration decays gradually with each episode.

---

4. Results and Analysis

Training Results

- **Total Rewards Over Time**: The agent demonstrated rapid improvement, with rewards stabilizing as it learned an optimal policy. Specific milestones include:
  - **Episode 0**: Total Reward = `-16,815`
  - **Episode 50,000**: Total Reward = `21`
  - **Episode 95,000**: Total Reward = `23`

Policy Evaluation

The agent successfully learned optimal paths from all specified starting points:

- **Start (0, 0)** → Goal: Path length = `78`
- **Start (10, 10)** → Goal: Path length = `58`
- **Start (30, 30)** → Goal: Path length = `20`
- **Start (39, 39)** → Goal: Path length = `0`

Q-Values

- **Near Goal**: `[Up: 0, Down: 0, Left: 0, Right: 0]` – The agent correctly learned to stay in the goal state.
- **Near Start**: `[-8.68, -7.76, -8.68, -9.59]` – Reflects early-stage exploration and suboptimal movements in the starting region.

Performance Monitoring

- **Reward Plot**: The plot illustrates a rapid improvement in rewards during early training, followed by stabilization near optimal performance.
- **Path Visualization**: The visual confirms efficient navigation by the agent toward the goal.

---

5. Conclusions

Design Choices

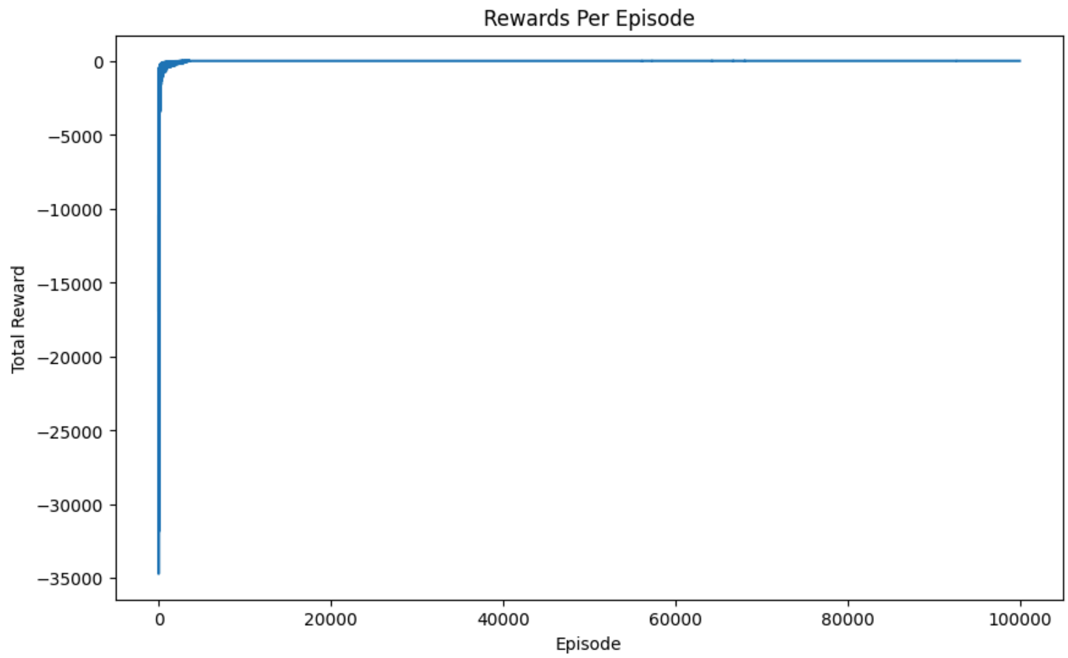
- The reward function effectively balanced exploration and exploitation while encouraging efficient navigation.
- The epsilon-greedy strategy ensured sufficient exploration during initial training phases while focusing on exploitation in later episodes.

Key Observations

- The agent successfully achieved the goal from all starting points after training, with efficient path lengths.
- The convergence of Q-values and rewards demonstrates the effectiveness of the learning setup.

Appendix: Visuals

Reward Plot:



Agent Path: