Assignment-1 Report

"Peter and the Wolf": Q-Learning Based Pathfinding Problem

Submitted By:-

SATYAM KUMAR

Roll- 2411AI44

M.Tech(AI)

Overview

This report presents a Q-learning-based solution to a pathfinding problem inspired by the "Peter and the Wolf" fairy tale. The goal is for Peter to navigate a grid environment, collecting apples while avoiding obstacles like wolves and water. The environment is represented as a grid of size 8x8, with various rewards and penalties assigned to each cell, depending on its type (e.g., apple, water, grass, etc.). The solution uses the Q-learning algorithm, a reinforcement learning approach, to train Peter to find the optimal path for collecting all apples.

Library Used

```
1. numpy
2. pandas
3. plotly
4. matplotlib
```

Environment Setup

The environment is represented as an 8x8 grid, where each cell contains one of the following:

- `.` : Empty space (normal movement penalty)
- `G` : Grass (low reward)
- `W` : Water (impassable, large negative reward)
- `A` : Apple (large reward, Peter's goal)
- `D` : Wolf (huge negative penalty, game over)
- `P` : Peter's starting position

Here is the environment layout used in this task:

```
environment = np.array([
    ['.','.','.','.','.','.','.','.'],
    ['.','.','.','.','.','G','.','.'],
    ['W','.','W','A','.','.','.','G'],
    ['W','W','.','.','P','.','.','.'],
    ['W','.','A','.','W','G','.','.'],
    ['.','.','.','.','.','W','.','.'],
    ['.','.','W','.','W','G','.','G'],
    ['.','.','.','W','.','A','.','D'],
])
```

The reward values associated with different items in the grid are as follows:

- **Apple (A)**: +100 points
- **Grass (G)**: +1 point
- **Water (W)**: -10 points (impassable)
- **Wolf (D)**: -100 points (game over)
- **Empty (.)**: -1 point

The primary objective is to navigate the grid, collecting all apples while avoiding the wolf and water cells, which represent danger and obstacles, respectively.

Q-Learning Algorithm

The Q-learning algorithm is employed to train Peter to collect apples efficiently. Q-learning is a reinforcement learning method that seeks to find the optimal action-selection policy for an agent. This is achieved by updating a Q-table that stores the values (expected future rewards) for each state-action pair.

Key parameters used in the Q-learning process:

- **Learning Rate (α)**: 0.3 — This parameter controls how much new information overrides the old information in the Q-table.
- **Discount Factor (γ)**: 0.9 — This controls how much the algorithm considers future rewards versus immediate rewards.
- **Exploration Rate (ε)**: 0.2 — This controls the balance between exploration (choosing random actions) and exploitation (choosing the best-known action).

The Q-table is initialized as a zero matrix with dimensions (8x8x4), corresponding to the grid size (8x8) and the 4 possible actions.

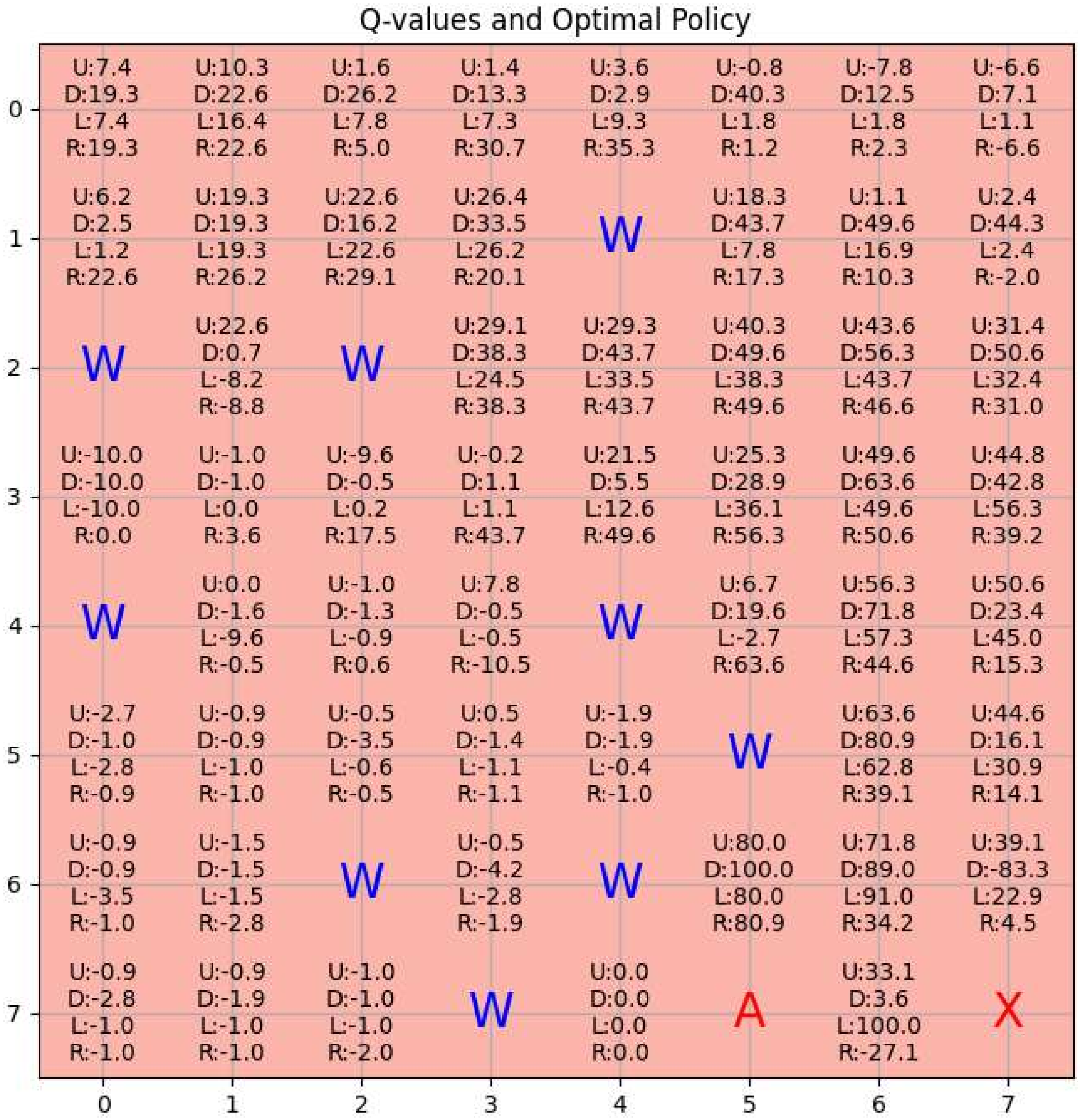
Training Process

The agent undergoes several training episodes, where Peter explores the grid and updates his Q-table based on the rewards received. The Q-value for each state-action pair is updated using the Bellman equation:

[Q(s, a) \leftarrow Q(s, a) + \alpha \left(r + \gamma \max Q(s', a') - Q(s, a) \right)]

Where:

- **(Q(s, a))** is the current Q-value for state **s** and action **a.**
- **(r)** is the reward received after taking action **a.**
- **(\max Q(s', a'))** is the maximum future reward achievable from the next state **(s').**



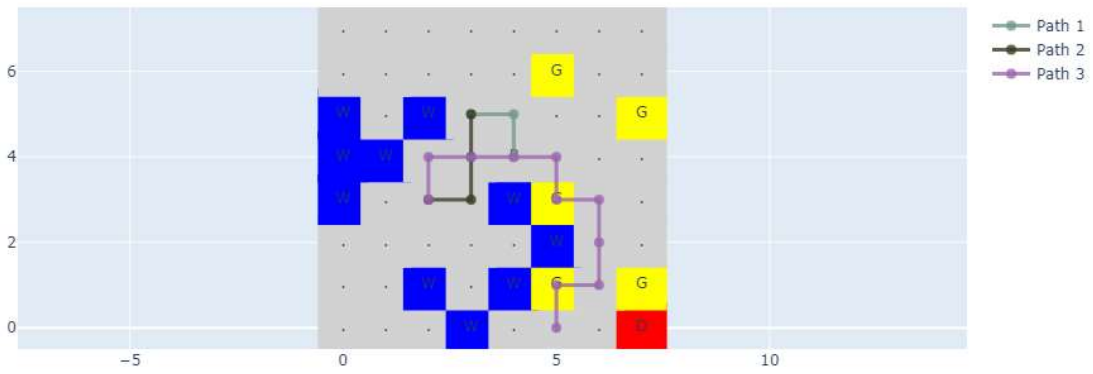
Q-values and Optimal Policy

	0	1	2	3	4	5	6	7
0	U:7.4 D:19.3 L:7.4 R:19.3	U:10.3 D:22.6 L:16.4 R:22.6	U:1.6 D:26.2 L:7.8 R:5.0	U:1.4 D:13.3 L:7.3 R:30.7	U:3.6 D:2.9 L:9.3 R:35.3	U:-0.8 D:40.3 L:1.8 R:1.2	U:-7.8 D:12.5 L:1.8 R:2.3	U:-6.6 D:7.1 L:1.1 R:-6.6
1	U:6.2 D:2.5 L:1.2 R:22.6	U:19.3 D:19.3 L:19.3 R:26.2	U:22.6 D:16.2 L:22.6 R:29.1	U:26.4 D:33.5 L:26.2 R:20.1	W	U:18.3 D:43.7 L:7.8 R:17.3	U:1.1 D:49.6 L:16.9 R:10.3	U:2.4 D:44.3 L:2.4 R:-2.0
2	W	U:22.6 D:0.7 L:-8.2 R:-8.8	W	U:29.1 D:38.3 L:24.5 R:38.3	U:29.3 D:43.7 L:33.5 R:43.7	U:40.3 D:49.6 L:38.3 R:49.6	U:43.6 D:56.3 L:43.7 R:46.6	U:31.4 D:50.6 L:32.4 R:31.0
3	U:-10.0 D:-10.0 L:-10.0 R:0.0	U:-1.0 D:-1.0 L:0.0 R:3.6	U:-9.6 D:-0.5 L:0.2 R:17.5	U:-0.2 D:1.1 L:1.1 R:43.7	U:21.5 D:5.5 L:12.6 R:49.6	U:25.3 D:28.9 L:36.1 R:56.3	U:49.6 D:63.6 L:49.6 R:50.6	U:44.8 D:42.8 L:56.3 R:39.2
4	W	U:0.0 D:-1.6 L:-9.6 R:-0.5	U:-1.0 D:-1.3 L:-0.9 R:0.6	U:7.8 D:-0.5 L:-0.5 R:-10.5	W	U:6.7 D:19.6 L:-2.7 R:63.6	U:56.3 D:71.8 L:57.3 R:44.6	U:50.6 D:23.4 L:45.0 R:15.3
5	U:-2.7 D:-1.0 L:-2.8 R:-0.9	U:-0.9 D:-0.9 L:-1.0 R:-1.0	U:-0.5 D:-3.5 L:-0.6 R:-0.5	U:0.5 D:-1.4 L:-1.1 R:-1.1	U:-1.9 D:-1.9 L:-0.4 R:-1.0	W	U:63.6 D:80.9 L:62.8 R:39.1	U:44.6 D:16.1 L:30.9 R:14.1
6	U:-0.9 D:-0.9 L:-3.5 R:-1.0	U:-1.5 D:-1.5 L:-1.5 R:-2.8	W	U:-0.5 D:-4.2 L:-2.8 R:-1.9	W	U:80.0 D:100.0 L:80.0 R:80.9	U:71.8 D:89.0 L:91.0 R:34.2	U:39.1 D:-83.3 L:22.9 R:4.5
7	U:-0.9 D:-2.8 L:-1.0 R:-1.0	U:-0.9 D:-1.9 L:-1.0 R:-1.0	U:-1.0 D:-1.0 L:-1.0 R:-2.0	W	U:0.0 D:0.0 L:0.0 R:0.0	A	U:33.1 D:3.6 L:100.0 R:-27.1	X

Optimal Pathfinding and Apple Collection

After training, the agent uses the learned Q-table to find the optimal paths to the apples. Once an apple is collected, it is removed from the grid, and the process continues until all apples have been collected.

Peter's Optimal Routes to Collect All Apples.



Results and Discussion

Initially, Peter is at(4,3) and it takes 1st path as Green color to get 1st Apple at(3,2) and after taking 1st apple it assumes the current position as starting point for the next apple and now it takes 2nd path with brown color to get 2nd Apple at(2,4)similarly, it takes 3rd path with purple color to get 3rd Apple at(5,7).