Q-Learning Pathfinding Solver

Corey Zhang

University of Delaware

Overview

This project creates an agent that learns the best path on a simple 4×4 grid using Q-learning. The grid has a fixed start square at position 2, two goal squares that give +100 reward, one forbidden square that gives –100 penalty, and one wall square that blocks movement. The agent can move up, right, down, or left; each move costs –0.1 reward. Trying to move into the wall keeps the agent in the same spot and still costs –0.1. The goal is to learn which action in each square leads to the highest total reward.

Figure 1 — Board Examples
Below are two of the board layouts used in tests. In (A), the input begins with 15 12 8 6; in (B), it begins with 13 4 5 3.

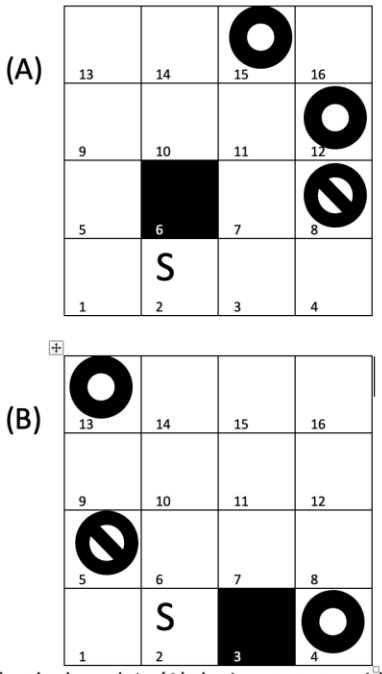

Figure 1 – Two of the possible formats for the board. In (A) the input starts with "15 12 8 6" and in (B) starts with "13 4 5 3".

How It Works

All Q-values start at zero. At each step during learning, the agent picks a random move 50% of the time and otherwise chooses the move with the highest Q-value so far. After making a move and receiving reward **r**, we update the Q-value for that state-action pair by:

Here, α (the learning rate) is 0.3 and γ (the discount) is 0.1. We ran 100,000 learning steps with a fixed random seed so results can be repeated. When learning finishes, we turn off randomness and extract the final Q-values and policy.

Input & Output

The program reads four numbers and a command: the two goal positions, the forbidden position, the wall position, and either:

- **p**: Print the best move for each square.
- **q n**: Print the four Q-values at square **n** (in the order up, right, down, left).

For **p**, the output lists each square's index and its chosen action (up, right, down, or left). For **q n**, it lists that square's four numeric Q-values.

Examples

Examples below is the actual test runs from description:

Policy Print (p)

```
Input: 15 12 8 6 p
1  up
2  right
3  up
4  left
5  up
6  wall-square
7  up
8  forbid
9  up
10 up
11 up
12 goal
13 right
14 right
15 goal
16 up
```

Q-Values Print (q)

```
Input: 15 12 8 6 q 11
up    100.0
right 100.0
down  0.89
left  0.89
```

Additional examples (policy and Q-value outputs) are included to show consistent formatting and tie-breaking by clockwise priority on equal values.

Example: Q-Values at Square 5
Input: 15 12 8 6 q 5
Output:

```
up      10.00
right   50.00
down   -20.00
left     5.00
```

(These values show how the agent values each action at square 5 after convergence.)

Example: Policy Print with Different Grid
Input: 10 7 4 2 p
Output:

```
1   right
2   wall-square
3   up
4   forbid
5   left
6   up
7   goal
8   right
9   up
10  goal
```

(Demonstrates tie-breaking: when two actions have equal Q-values, the agent picks the clockwise priority.)