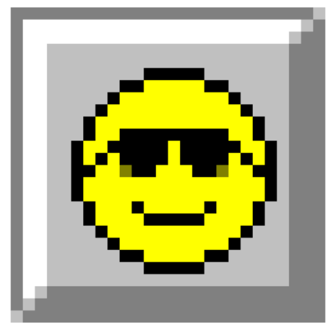
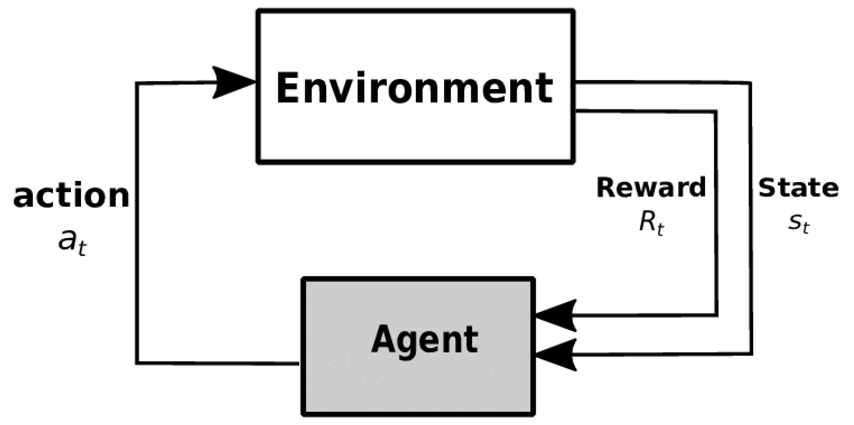
Variations on Q-Learning for Minesweeper



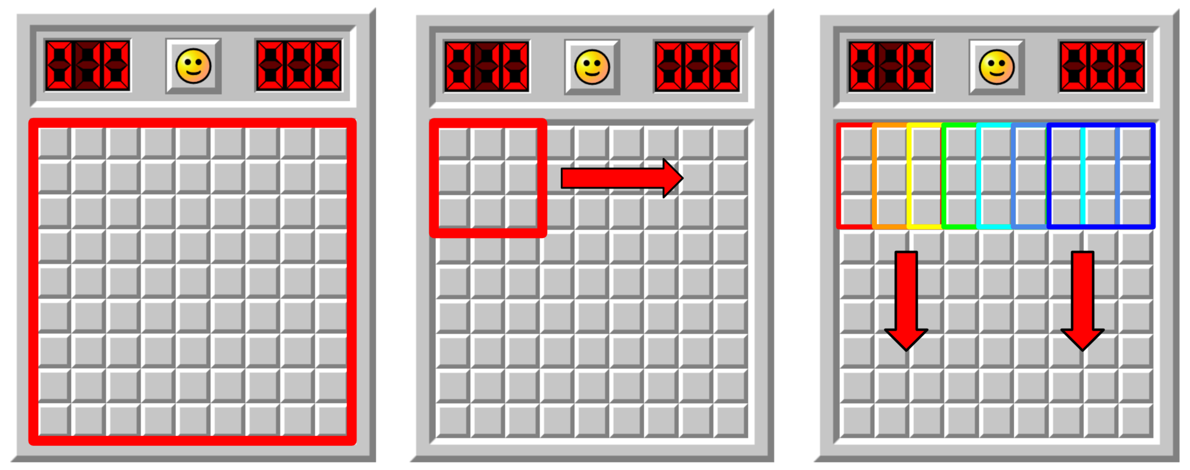
Shane Sawyer

Introduction

With the rising popularity of deep learning and neural networks, other techniques have been less explored. I aimed to implement various approaches to train an agent to play Minesweeper with Q-learning, a popular reinforcement learning algorithm. Reinforcement learning is a machine learning method in which an agent interacts with an environment and learns behavior based off rewards.



Minesweeper's complexity was simplified by using two strategies: a moving 3x3 grid and individual agents for each 3x3 section on the board. These changes allow the agent to handle the game more easily.



Normal | "Sliding Window" | Multi Agent

Q-learning is a method used to figure out the best action to take in each situation. It works by estimating the rewards for each possible action. This estimation is updated using a specific formula:

$$Q(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)]$$

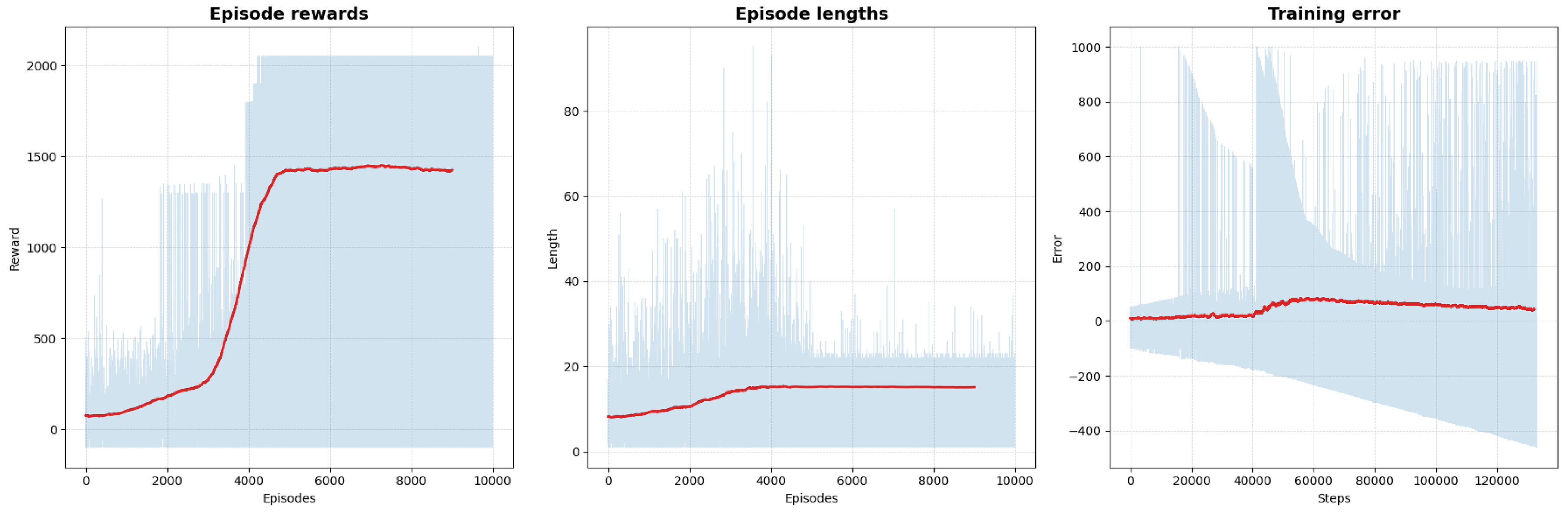
- Q(s, a): estimated reward
- α: learning rate
- R(s, a): actual reward received
- γ: discount factor
- max_a' Q(s', a'): maximum estimated reward of next action
- During training, agent can act randomly to learn rewards

Graphs

The following graphs represent data from training each agent type on 10,000 9x9 boards with 10 mines on 5 different seeds.

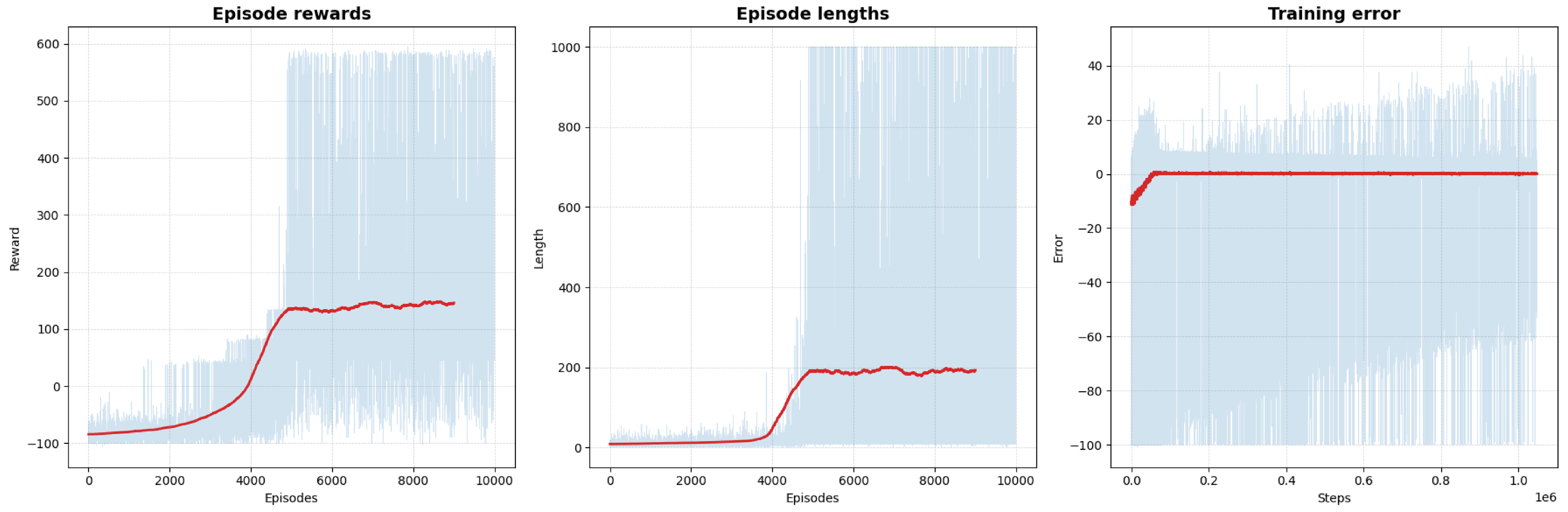
Normal Q-Learning

- Uses entire board as the observation
- Quickly learns how to maximize rewards



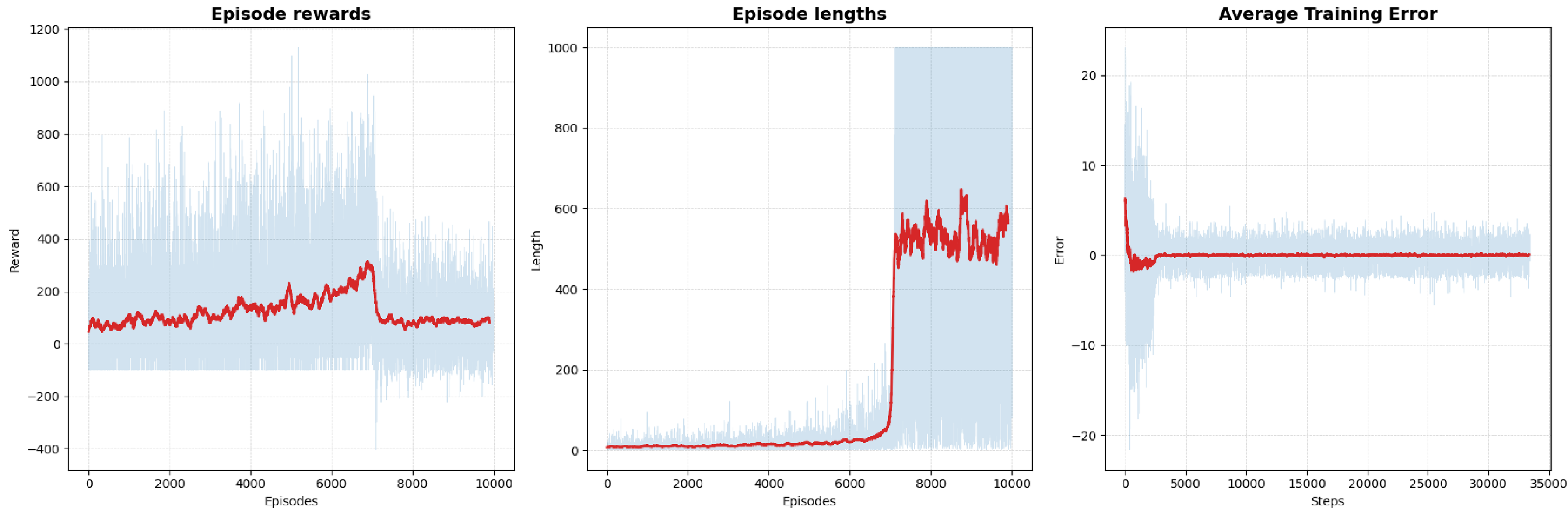
"Sliding Window" Q-Learning

- Uses a 3x3 section as the observation
- 3x3 section is moved after each time agent makes an action and updates
- "No operation" introduced to allow agent to not be forced to make decisions



Multi Agent Q-Learning

- 49 individual agents each on a unique 3x3 section
- Section does not move
- Agents chosen randomly to act
- Also gives agents the option to not act (no operation)
- Heavily restrict exploration strategy to ensure mine is not randomly picked



Results

- Trained on 1,000,000 games
- Percentage games won out of 10,000 after training
- Each board was 9x9 with 10 mines

Agent Type	Win Rate
Normal	0.00%
Sliding Window	1.53%
Multi Agent	4.18%

Conclusion

- The inherent complexity and diverse states of Minesweeper pose a challenge to traditional Q-learning.
- Simplification of the game board ensures manageability and meaningful decision-making.
- This simplification impacts the optimal application of Q-learning to the entire Minesweeper board.
- The compromise allows for faster learning of winning actions, preservation of board variety, and reduced training time.
- Even though the integration of neural networks could enhance performance, strategic simplifications make them non-essential.

Image Credits

1. https://raw.githubusercontent.com/dawsonbooth/pynsweeper/master/assets/png/social.png
2. https://www.researchgate.net/figure/Reinforcement-Learning-Agent-and-Environment_fig2_323867253