Training a Minesweeper Solver

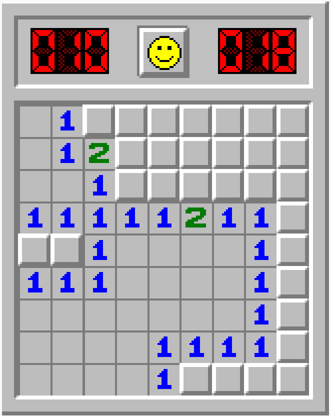
Sayan Dutta (20162097)
Satrajit Datta (20162093)
Swarnadeep Chatterjee(20162094)
Harshveer Singh

International Institute Of Information Technology, Hyderabad

Table of contents

Introduction

WHAT IS MINESWEEPER ??



Minesweeper

- **Minesweeeper**, a puzzle game introduced in 1960’s requires spatial awareness and ability to work with incomplete information. Utilizing different Machine Learning approaches, we implemented solvers that makes use of **Reinforcement Learning**.
- A modified Q-learning algorithm was enhanced by function approximation, which was able to effectively generalize learning of the state space.

Minesweeper

- Playing a game of Minesweeper involves uncovering tiles until the player uncovers a tile containing a mine or uncovers all of the tiles that don’t contain mines.
- As the game progresses, the player is given limited information regarding the location of the mines on the board.
- Using this information the Minesweeper solver should determine which action to take at each stage of the game.

Goal

Goal

The solvers of Minesweeper are implemented using three algorithms :

- Simplified Q-learning
- Modified Q-learning
- Deep Q-learning

Q-Learning

Q-learning Basics

- At each step s, choose the action a which maximizes the function Q(s, a)
- Q is the estimated utility function – it tells us how good an action is given a certain state
- Q(s, a) = Immediate reward for making an action + best utility (Q) for the resulting state

Q-learning

- Here modified version of Q-learning is used to discover the best actions for each given board configuration.
- As we are more interested in the immediate reward rather than the end game result, we have not taken into consideration the final max optimized value.
- We estimate which tile is least likely to have a mine in a given board configuration by finding the tile with the highest Q value

Q-learning :

$$Q(s, a) = r(s, a) + \gamma max_{a'}(Q(s', a'))$$

$$\gamma = Relative\ \ value\ \ of\ \ delayed\ \ vs.immediate\ \ rewards\ \ (0 to 1)$$

r(s,a) = Immediate Reward
s' = The new state after action a

a , a' : Actions
s , s' : States
Selected action:

$$\pi(s) = argmax_a(Q(s, a))$$

The original **Bellman Equation** allows the algorithm to learn not just about the direct reward of the particular action but whether the particular action is more likely to lead to reward in the long-term.

But in **Minesweeper** we are interested in the immediate reward, whether a particular move will uncover a mine on a specific board configuration.

$$Q(s, a) = r(s, a) + Q(s, a)$$

Q-Learning Algorithm

```
Begin probing by selecting a corner
While not game over do
        S <- current state of the board
        Array <- all tiles on frontier
        For tile in Array do
                P(s,a) <- P(s,a)+1
        End for
        Probe random square in Array
End while
```

MODIFIED Q-LEARNING

Modified Q-Learning Algorithm

```
Begin probing by selecting a corner
While not game over do
        S <- current state of the board
        Array <- all tiles on frontier
        For tile in Array do
                P(s,a) <- P(s,a)+1
        End for
        Probe  square in Array with the least probability of being a mine
End while
```

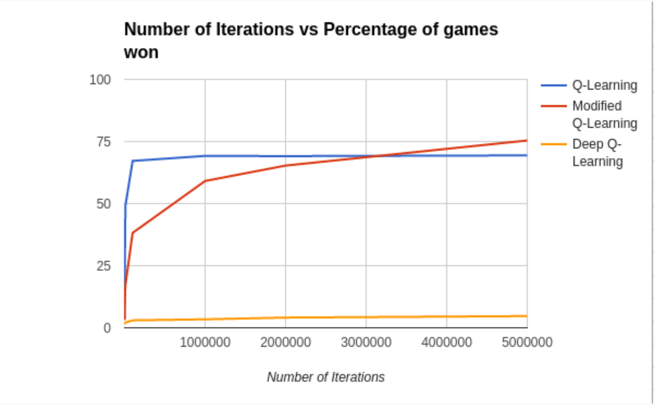
Deep Q-learning

Deep Q-Learning

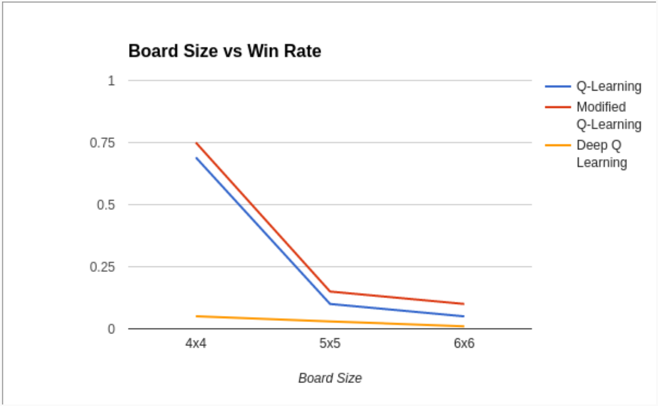
```
Begin probing by selecting a corner
While not game over do
    S <- current state of the board
    Array <- all tiles on frontier
    Choose random tile t from frontier
    Append (S,t) in experience buffer
    If experience buffer is full do
        Train Network on randomly chosen samples from buffer
    End If
End while
```

Analysis

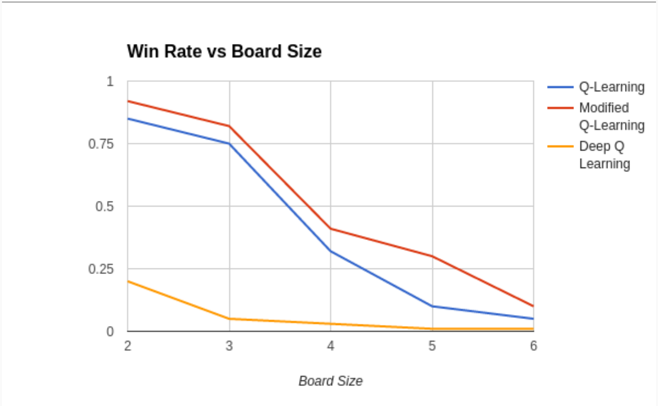
COMPARISONAL ANALYSIS



COMPARISONAL ANALYSIS



COMPARISONAL ANALYSIS



Conclusion

Conclusion

- While Q-learning is good for board sizes, it’s performance decreases drastically for larger boards.
- To solve the above problem, Deep Q-learning has been implemented.
- The version of Deep Q-learning implemented is not giving satisfactory results.

Future Works

Considering a fixed state space for the Neural Network.

DEMO

References

- Kaye, R (2000). Minesweeper in NP- complete The Mathematical Intelligencer, 22, 9-15.
- Alisher Tortay, Oleg Yurchecnko. Solving Minesweeper using NN.
- **Reinforcement Learning**. `http://reinforcementlearning.ai-depot.com/`