Tic-Tac-Toe Simulator

INTRO

In this assignment, a simple simulator for Tic-Tac-Toe that encodes a standard 3x3 grid is implemented to train an artificial agent. The Q-matrix is implemented to conduct Q-learning for the agent. The Q-matrix is initialized to all zeros at the beginning of each run. To note, ϵ decay is limited to a minimum value of 0.0001 to allow for the program to run efficiently.

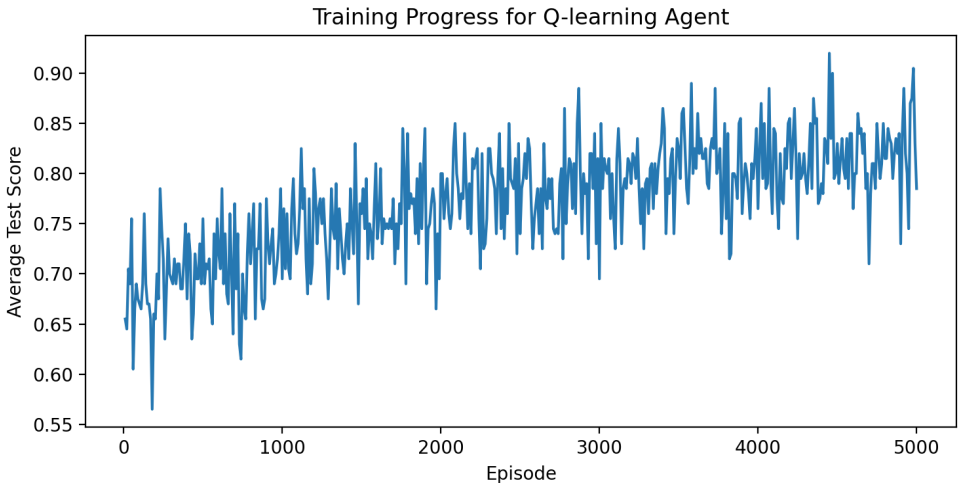
METHOD

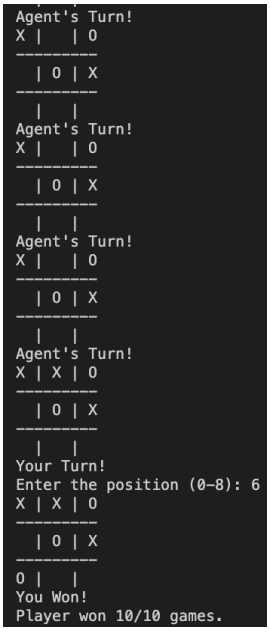
After each training episode, the Q-agent plays 10 games against a baseline random opponent. Scores are awarded as follows : +1 for a win, +0.5 for a draw, and 0 for loss. The average of the 10 games is recorded. In order to smooth out the plot to see a clearer trend, for every 10 episodes, all game averages are summed and averaged. This average of averages is used to create the plot of average test scores (Y-axis) against episodes (X-axis).

EXPERIMENTS

Parameters: $\Delta = 0.001$, $\epsilon = 0.5$, $m = 50$, $episodes = 5000$

Overall game results against random opponent, wins: 36077, draws: 5071, losses: 8852

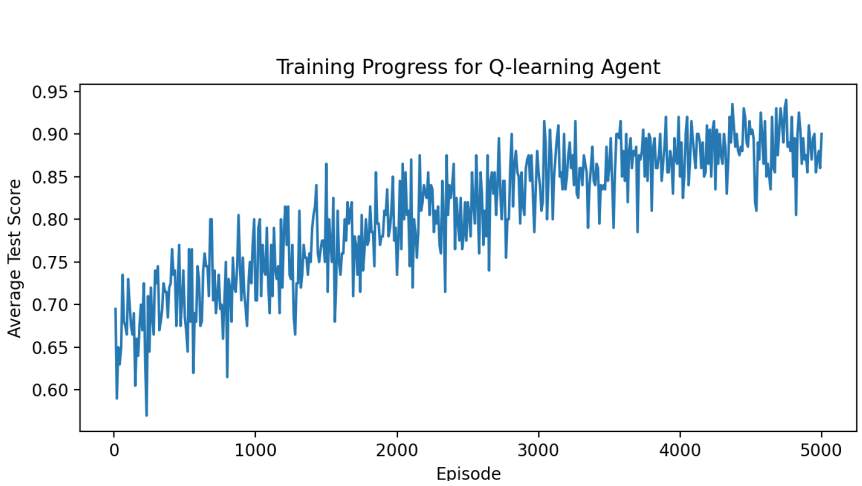




Parameters: $\Delta = 0.003$, $\epsilon = 0.5$, $m = 50$, $episodes = 5000$

Δ reaches 0.0001 at episode 4200

Overall game results against random opponent, wins: 39937, draws: 3089, losses: 6974

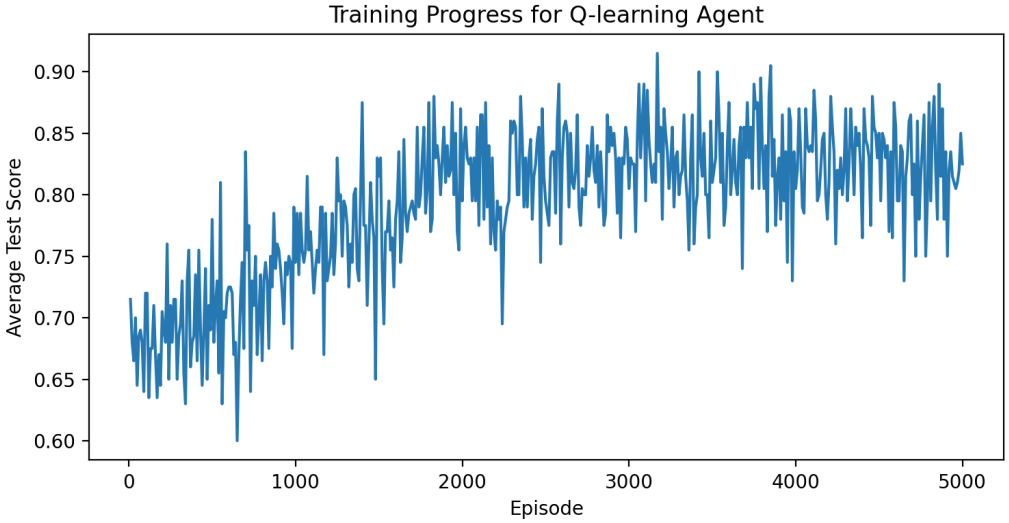



```
X |   | O
Agent's Turn!
X |   |
O | X | O
X |   | O
Agent's Turn!
X |   |
O | X | O
X |   | O
Agent's Turn!
X |   |
O | X | O
X |   | O
Agent's Turn!
X |   | X
O | X | O
X |   | O
Agent Won!
Player won 9/10 games.
```

Parameters: $\Delta = 0.005$, $\epsilon = 0.5$, $m = 50$, $episodes = 5000$

Δ reaches 0.0001 at episode 2500

Overall game results against random opponent, wins: 36722, draws: 5777, losses: 7501




```
Agent's Turn!
X | O |
X |   |
O |   |
Agent's Turn!
X | O | X
X |   |
O |   |
Your Turn!
Enter the position (0-8): 4
X | O | X
X | O |
O |   |
Agent's Turn!
X | O | X
X | O |
O | X |
Your Turn!
Enter the position (0-8): 8
X | O | X
X | O |
O | X | O
Agent's Turn!
X | O | X
X | O | X
O | X | O
It's a Draw!
Player won 7/10 games.
```

OBSERVATIONS

$\Delta = 0.001,\ 0.003,\ 0.005$

0.001:

- Overall Game Results: Wins - 36077, Draws - 5071, Losses - 8852
- Average Test Scores: Started around 0.55, reached a maximum of 0.90 near the end of 5000 episodes.

0.003:

- Overall Game Results: Wins - 39937, Draws - 3089, Losses - 6974
- Average Test Scores: Started around 0.60, reached a maximum of 0.95 near the end of 5000 episodes. Epsilon decayed to the minimum of 0.0001 at episode 4200.

0.005:

- Overall Game Results: Wins - 36722, Draws - 5777, Losses - 7501
- Average Test Scores: Started around 0.60, reached a maximum of around 0.93 at about 3000 episodes. Then the average test scores slightly dropped to around 0.87 till the end of 5000 episodes. Epsilon decayed to the minimum of 0.0001 at episode 2500.

DISCUSSION

Delta and Learning Rate

The parameter delta in this context can be seen as a form of learning rate. Smaller delta values (e.g., 0.001) result in slower changes to epsilon, allowing the agent to explore for longer periods. Larger delta values (e.g., 0.005) cause epsilon to decrease more quickly, potentially making the agent transition to exploitation sooner.

Exploration vs. Exploitation

Smaller delta values (e.g., 0.001) result in slower epsilon decay, which means the agent continues to explore longer. This could lead to more diverse experiences and potentially better learning, resulting in higher average test scores. However, if the decay is too slow, the agent might take longer to converge to optimal strategies.

Epsilon Decay Timing

For delta = 0.003, epsilon decayed later at episode 4200, compared to delta = 0.005 where epsilon decayed earlier at episode 2500. A slower decay (later decay) can allow the agent to explore for a longer time and adapt to different situations.

Performance and Stability

The results show that delta = 0.003 led to the highest average test scores and relatively fewer losses. A balance between exploration and exploitation seems to have been achieved. The fact that it decayed to a minimum epsilon of 0.0001 at episode 4200 suggests stability in learning and exploitation.

Trade-off

There's a trade-off between slower learning with more exploration (smaller delta) and faster convergence with less exploration (larger delta). The agent with delta = 0.003 seems to strike a balance between these two aspects.

In summary, the choice of delta affects how quickly the agent transitions from exploration to exploitation. The optimal delta value might depend on the specific problem, the learning process, and the desired balance between exploration and exploitation. The observed results suggest that `delta = 0.003` yielded a good balance between exploration and exploitation, resulting in higher average test scores and fewer losses against a random opponent.

PLAYING AGAINST Q-AGENT

Q-agent played as “X”, while the player played as “O”, and the Q-agent always started first. Naturally, a common strategy for the first player would be to select the center position:

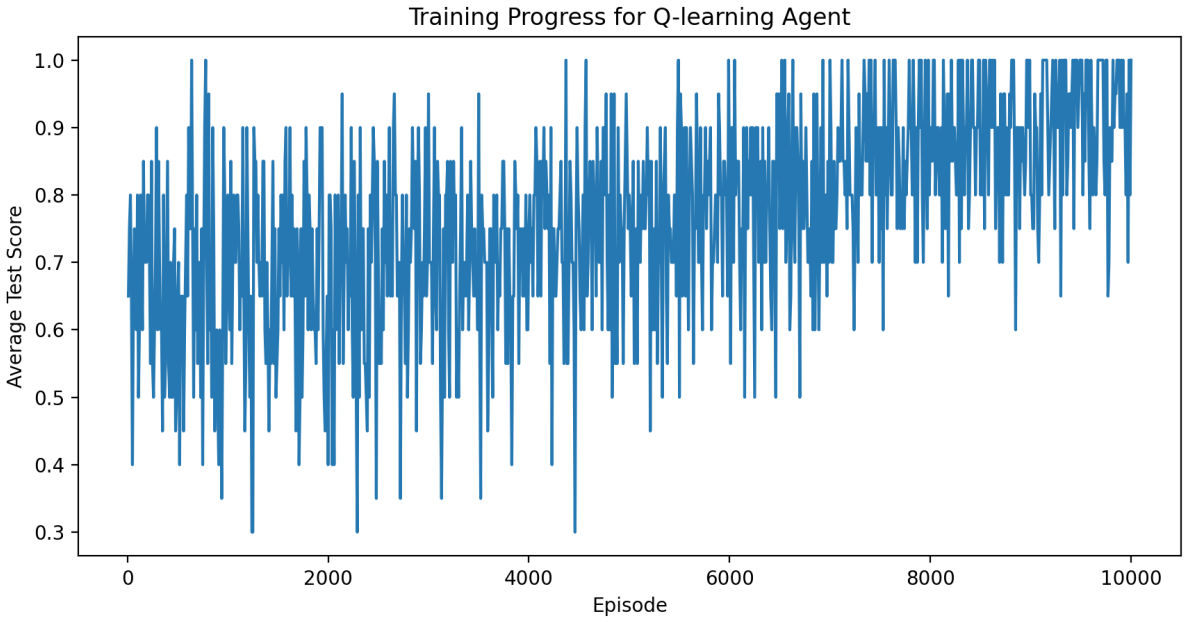
	X	

However, the Q-agent never chooses the center position as its first move. I as the second player would choose the center position. I soon realized that the Q-agent almost always follows a

predictable strategy, though the strategy would vary between experiments. This meant that every game played out with all the same moves, unless I decided to do something different, such as blocking Q-agent's predictable second move when I played my first move.

As long as I chose the center position as the second player, I could always win. However, should I avoid the center position for my first play, the Q-agent could manage a draw when it chooses the center as its second play, whilst if it doesn't choose the center, it is guaranteed to lose.

This behavior suggests overfitting in Q-agent's strategy, and is evident that even though Q-agent performs well against a random opponent, it does not do well against players optimized to win (Minimax comes to mind). Therefore, it would be much more ideal to train Q-agent against a Minimax agent rather than a random opponent.




```
Agent's Turn!
  |   |
---------
  |   |
---------
  | X |
Your Turn!
Enter the position (0-8): 6
  |   |
---------
  |   |
---------
O | X |
Agent's Turn!
  |   |
---------
  | X |
---------
O | X |
Your Turn!
Enter the position (0-8): 1
  | O |
---------
  | X |
---------
O | X |
Agent's Turn!
  | O |
---------
  | X | X
---------
O | X |
Your Turn!
Enter the position (0-8): 3
  | O |
---------
O | X | X
---------
O | X |
Agent's Turn!
X | O |
---------
O | X | X
---------
O | X |
Your Turn!
Enter the position (0-8): 7
Invalid move! Position is already taken.
X | O |
---------
O | X | X
---------
O | X |
Your Turn!
Enter the position (0-8): 8
X | O |
---------
O | X | X
---------
O | X | O
Agent's Turn!
X | O | X
---------
O | X | X
---------
O | X | O
It's a Draw!
```

```
Agent's Turn!
  |   |
---------
  |   |
---------
  | X |
Your Turn!
Enter the position (0-8): 4
  |   |
---------
  | O |
---------
  | X |
Agent's Turn!
  |   |
---------
  | O |
---------
  | X | X
Your Turn!
Enter the position (0-8): 2
  |   | O
---------
  | O |
---------
  | X | X
Agent's Turn!
  |   | O
---------
  | O |
---------
X | X | X
Agent Won!
```

```
Agent's Turn!
  |   |
---------
  |   |
---------
  | X |
Your Turn!
Enter the position (0-8): 4
  |   |
---------
  | O |
---------
  | X |
Agent's Turn!
  |   |
---------
  | O |
---------
  | X | X
Your Turn!
Enter the position (0-8): 6
  |   |
---------
  | O |
---------
O | X | X
Agent's Turn!
  |   |
---------
  | O | X
---------
O | X | X
Your Turn!
Enter the position (0-8): 2
  |   | O
---------
  | O | X
---------
O | X | X
```

```
Agent's Turn!
  |   |
---------
  |   |
---------
  | X |
Your Turn!
Enter the position (0-8): 8
  |   |
---------
  | X | O
Agent's Turn!
  |   |
---------
  | X |
---------
  | X | O
Your Turn!
Enter the position (0-8): 1
  | O |
---------
  | X |
---------
  | X | O
Agent's Turn!
  | O |
---------
X | X |
---------
  | X | O
Your Turn!
Enter the position (0-8): 5
  | O |
---------
X | X | O
---------
  | X | O
Agent's Turn!
X | O |
---------
X | X | O
---------
  | X | O
Your Turn!
Enter the position (0-8): 6
X | O |
---------
X | X | O
---------
O | X | O
Agent's Turn!
X | O | X
---------
X | X | O
---------
O | X | O
It's a Draw!
```