Assignment 2

Report

Mini-Max algorithm for the Tic-Tac-Toe game:

Q. What is Tic-Tac-Toe game?

Tic-Tac-Toe is a game in which two players seek alternate turns to complete a row, a column, or a diagonal with either three O's or three X's drawn in the spaces of a grid of nine squares.

Q. What is Minimax Algorithm and why should you use it?

Minimax is a decision rule used in artificial intelligence, game theory, decision theory, etc…

Minimax is useful because it leverages the capability of computers evaluating an exponentially growing series of possible scenarios.

There're two utility values called the **min** and **max** value that help the A.I. agent to decide its next optimal move.

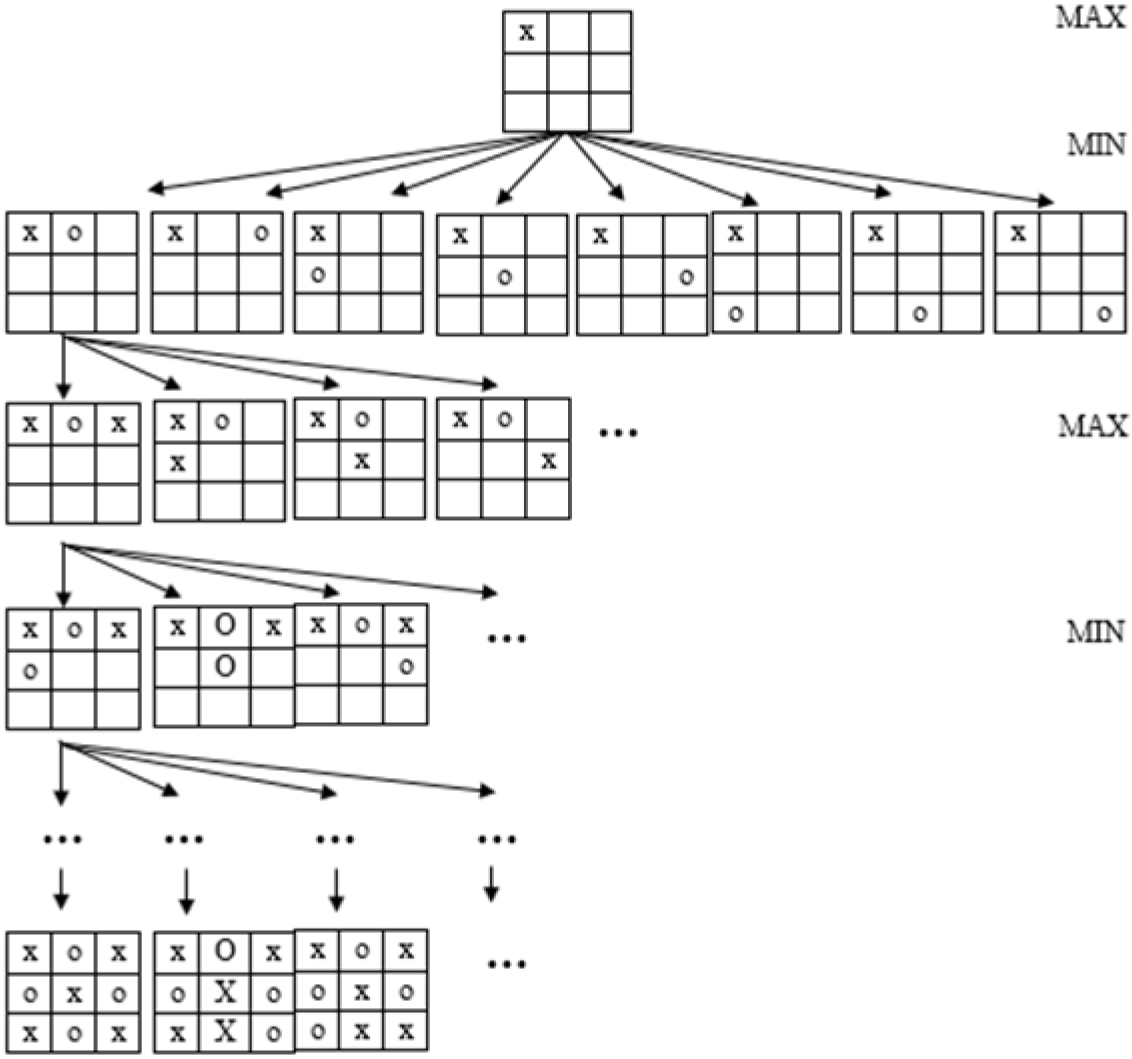
-Min value

This is the value where the A.I. agent seeks to **minimize** the possible loss for a **worst-case scenario.**

-Max value

This is the value where the A.I. seeks to **maximize** the possible gain for a **best-case scenario**.

Min-max is a decision-making algorithm which uses decision theory, game theory, statistics and philosophy to calculate the optimal move It is a two-player game

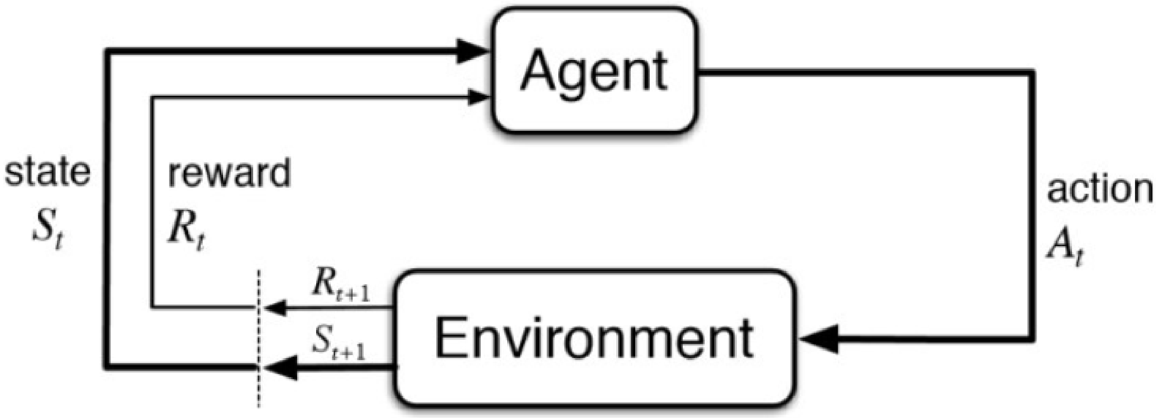


In the Tic-Tac-Toe game, a player tries to ensure two cases:
• Maximize a player's own chances of win.
• Minimize opponent's chances of win.
Maximize profit: The profit can be maximized by either fork or win.
Fork: Initially player will create opportunity where he can win in two ways.
Win: If there two same X or O in a row, then play the third to get three in a row.
Minimize Loss: The loss can be minimized by a block.
Block: If two 'x' or 'o' of the opponent are in a row then block it, or else block opponent's fork.

Note: The Minimax Algorithm is suitable for Tic-Tac-Toe where the AI agent made this game unbeatable. However, tic-tac-toe doesn't fully utilize the potential of Minimax where the chess game might be a better representation.

Reinforcement Learning algorithm for the Tic-Tac-Toe game.

The RL agent is trained by playing the game multiple times and updating the Q-values based on the rewards received for winning, losing, or drawing.



The state of this game is the board state of both the agent and its opponent, so we will initialise a 3x3 board with zeros indicating available positions and update positions with 1 if player 1 takes a move ad -1 if player 2 takes a move. The action is what positions a player can choose based on the current board state. Reward is between 0 and 1 and is only given at the end of the game.

For Tic-Tac-Toe specifically, Mini-Max is a reliable choice due to its guaranteed optimality. However, Reinforcement Learning strengths in adaptable and scalability make it more suitable for more complex games or situations where the optimal strategy is not known in advance.

The best way to compare the efficacy of Mini-Max and Reinforcement Learning is to experiment with both algorithm on a variety of games.

To update value estimation of states, we will apply value iteration which is updated based on the formula below.

$$V(S_t) \leftarrow V(S_t) + \alpha \Big[V(S_{t+1}) - V(S_t) \Big]$$

Q. Compare the efficiency of AI game play against a human player using Mini-max and Reinforcement Learning.

Table summarizing the Strength and weaknesses of Mini-Max and Reinforcement Learning

Algorithm	Strengths	Weaknesses
Mini-Max	Powerful, guaranteed to find the best move	Computationally expensive
Reinforcement Learning	Can learn to play complex games	Slow to learn

Table to summarizing some Feature for compression between Mini-Max and Reinforcement Learning.

Feature	Minimax	Reinforcement Learning
Guarantee of optimality	Yes	No
Computational complexity	High	Low
Learning ability	No	Yes
Scalability	Limited	High
Performance	Optimally	Initially struggle

I have played 10 rounds of the Tic-tac-toe game with each algorithm and noted the result.

Min-Max Algorithm Result:

Rounds	Winner	Remark
1^{th}	Draw	Both have put in equal effort.
2^{nd}	Draw	Both have put in equal effort.
3^{rd}	Draw	Both have put in equal effort.
4^{th}	AI	This time the computer won because the human made a mistake.
5^{th}	Draw	Both have put in equal effort.
6^{th}	Draw	Both have put in equal effort.

7th	AI	This time the computer won because the human made a mistake.
8th	Draw	Both have put in equal effort.
9th	Draw	Both have put in equal effort.
10th	Draw	Both have put in equal effort.

Conclusion:

The AI has won 2 times and Draw 8 times. This shows that a computer is playing optimally because the AI is not going to lose either win or draw. Humans can make mistakes twice or thrice in 10 rounds game but the AI is not going to make mistakes.

Reinforcement Learning Result:

Rounds	Winner	Remark
1th	Human	Human won in 4th move.
2nd	Human	Human won in 4th move.
3rd	AI	The computer won because Human made a mistake.
4th	AI	This time the computer won intelligently.
5th	Draw	There was equal effort from both sides.
6th	Human	Human won in 5th move.
7th	AI	The computer won in 3rd move.
8th	Draw	There was equal effort from both sides.
9th	Draw	There was equal effort from both sides.
10th	Human	Human won in 3th move.

Conclusion:

Human has won 4 times and the AI won 3 times and Draw 3 times. This shows relatively balanced competition between humans and AI.