AI - ASSIGNMENT 2 REPORT

SHEKHAR SHRIVAS

Why does the minimax algorithm never lose a match or reach to draw state?

The minimax algorithm is designed to play games optimally, assuming the opponent plays optimally as well.

After playing with minimax I observe that the minimax always tries to increase his score and reduce the opponent's score.

In Tic Tac Toe, with perfect play, the minimax algorithm can ensure at least a draw against any opponent.

X => Human
O => AI
And the number associated with X & O is the move number in a sequence.

Case 1			Case 2			Case 3		
X - 1	X - 5	O - 6	O - 9	O - 8	X - 5	O - 6	X - 5	X - 1
O - 8	O - 2	X - 9	X - 3	X - 1	O - 4	X - 9	O - 2	O - 8
X - 7	O - 4	X - 3	O - 6	X - 7	O - 2	X - 3	O - 4	X - 7

Case 4			Case 5		
O - 8	O - 6	X - 9	X - 2		
X - 1	X - 3	O - 4		X - 4	
O - 2	X - 5	X - 7	O - 3	O - 5	O - 1

In case 5 AI has won the match and all the matches are tied.

When I make changes in code and give a chance to the AI to move first it always starts from the last box or say from the 9 index. I am assuming that you start indexing from 1.

Reinforcement Game play

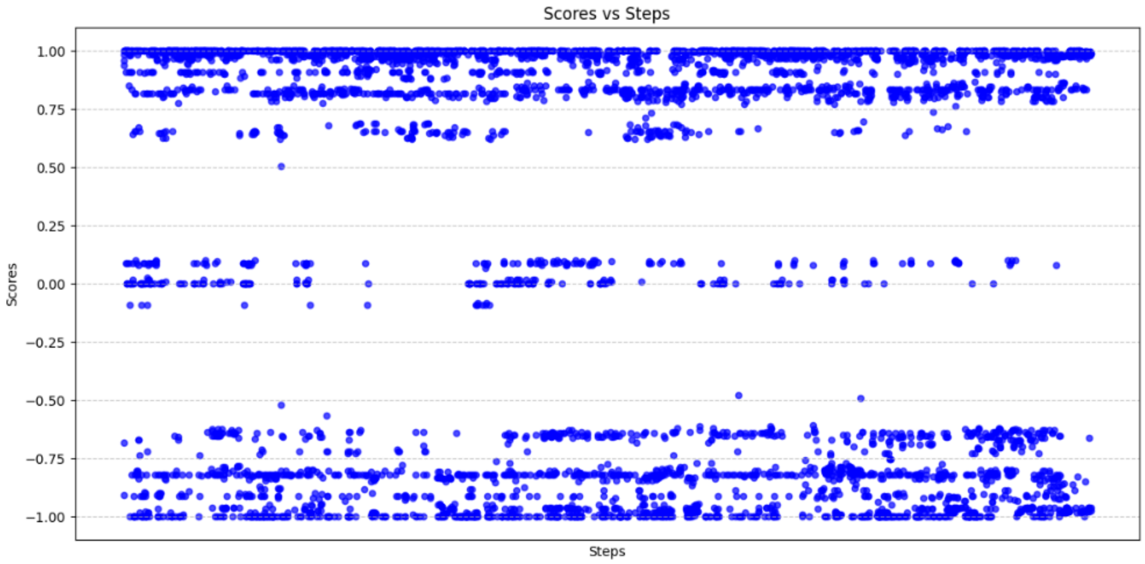
Case 1			Case 2			Case 3		
X - 1		O - 2	0 - 4	X - 5		O - 4		
X - 7	X - 3	X - 5		X - 3		O - 6		X - 1
	O - 6	0 - 4	0 - 1	X - 1		O - 2	X - 3	X - 5

Case 4			Case 5		
O - 8	X - 7	X - 1	O - 6	X - 7	O - 2
X - 5	O - 4	O - 2	X - 1	X - 3	O - 4
X - 3	O - 6	X - 9	X - 9	O - 8	X - 5

In case 3 AI has won the match.
Case 4 & 5 are a draw match.
In case 1 & 2 opponent has won the match.

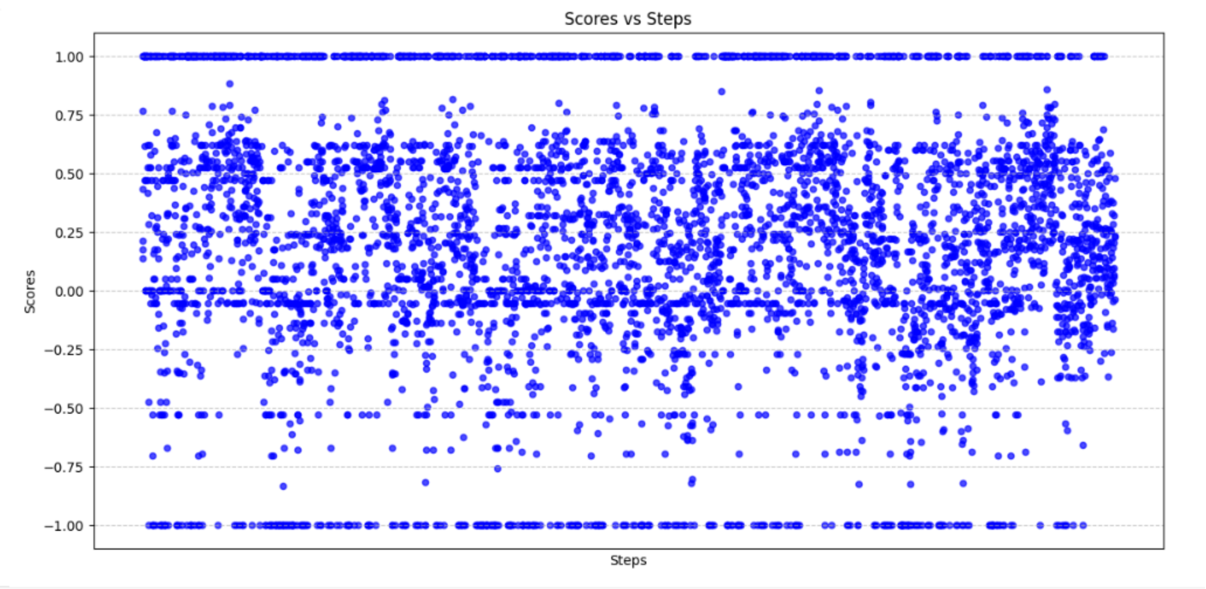
# Reinforcement :-

Initially the value for each state is 0 if the draw or it is not the terminal state. And 1 for if the is win of the opponent or -1 for if the is win of AI.
Alpha = 0.90
Iterations = 110



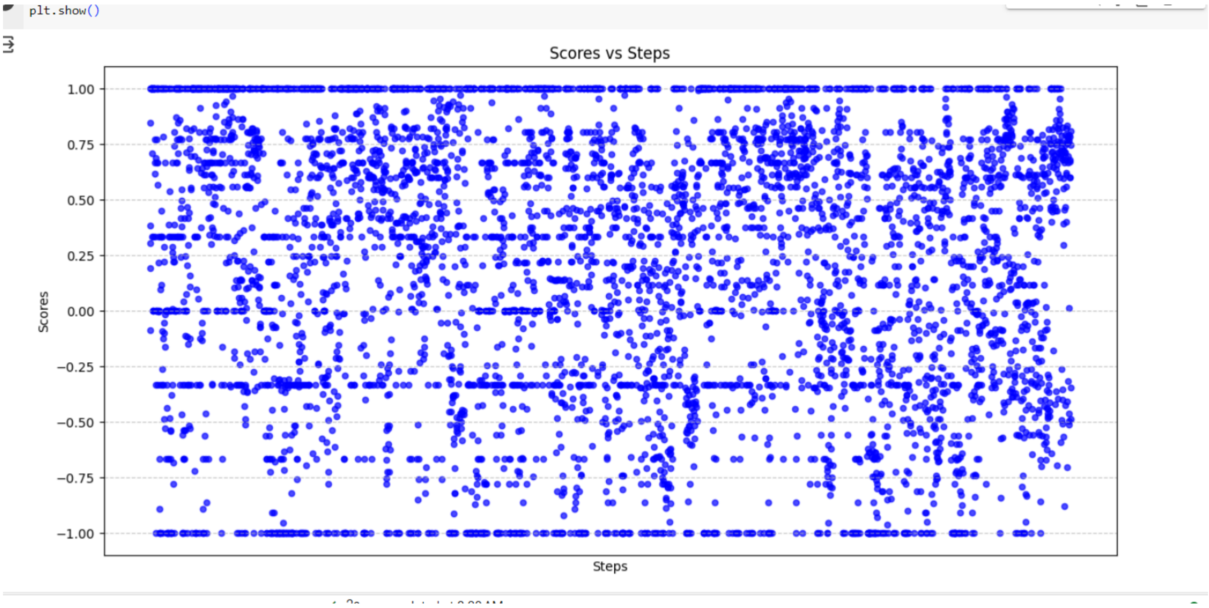
It can be observed that the value has changed but they are mostly around -1, 0 and 1.

Alpha = 0.10
Iterations = 200



On taking the value of Alpha quite low I observed that there was drastic change in the values of the States.

Alpha = 0.50
Iterations = 110



```
plt.show()
```

It can be seen that the values are saturated. I stopped my training after these many iterations.
Reason for stopping the training:

1. The graph was not changing much.
2. All the values were distributed in the graph and were taking multiple values.

# Observation

MiniMax :- When I was playing the game with MiniMax AI I was never able to win from the AI. AI is always choosing the best move and always trying to maximise his own score and reduce my score so that I am not able to reach the win state.

Reinforcement :- When I was playing the game with Reinforcement AI in most of the cases I was able to win with AI. This is different from the MiniMax because in MiniMax there is no chance of winning with AI.
On changing the value of learning rate the experience was changed sometimes it is to play or sometimes it is heard to play.

Observation table

MINIMAX	REINFORCEMENT
It depends on the opponent if the opponent is playing optimally then no one will win. Otherwise AI will win	In my case there are high chances humans will win.
It never makes mistakes so it does not feel like a human.	It feels like a human because sometimes it makes mistakes and gets caught in the trap of the opponent.

It is based on fixed rules and will always perform same	We can modify the learning rate and the moves will be changed
In minmax user never wins the game	In this both user and computer have chances of winning

During my Diwali break I learnt pygame from youtube because I want to work on an online game theme for SUcoders hackathon. So in optional I submitted that Tic-Tac-Toe game using pygame for GUI.

In Reinforcement learning I have created a function called Reinforcement. In this function I have taken help from my friend(Bharat).

<u>THANK YOU</u>