Aerial Robotics Kharagpur Documentation Task 5

Agnibha Sinha *

Abstract— **This is the documentation for Task 5, the Tic Tac Toe game**

I. INTRODUCTION

The program predicts the best move for a player in the tic tac toe game using the mini max algorithm. The algorithm is used recursively to calculate the value of all the game states. There are too many states which are calculated repetitively which could have been reduced if a record would have been maintained of the values of the generated states just like the optimization technique used in Task 1.

II. INITIAL ATTEMPTS

As I did not get enough time for this task, the initial attempt was the final attempt although I had a better approach in mind.

III. Final Approach

Minimax Algorithm:

This is a recursive function which traverses all game states according to the condition that the minimizer tries to minimize the score and the maximizer tries to maximize the score to the optimum value and these 2 players play alternatively.

At each state, where ever there is a blank, the function plays it out the game till the end and returns 10 if the maximizer wins and -10 if the minimizer wins and 0 if the game is a draw. The move which has the maximum output is printed.

The minimax function takes three parameters, the board state, the depth which is the number of moves played out and the player who will be playing which will be the opposite of the player playing the previous move.

Implementation:

The board is taken as a 3x3 matrix with x , o and _ as the entries.

The win is evaluated when there are three x or three o in the same row or same column or same diagonal. Accordingly + 10 or - 10 is returned depending on the player winning the game.

In the minimiax function, the whole array is traversed and if a _ is found, the function calls itself recursively by making a move in that cell and incrementing the depth and changing the player for the next move.

The cell giving the maximum output is the best move.

IV. Results and Observation

The program works fine for a 3x3 board. As most of the states lead to a draw, generally the last blank cell which gives a draw is printed.

Even if there is already a win on the board, the program outputs a cell as the maximum value is -10 from all states.

As mentioned before, the running time can be reduced significantly by generating the entire game tree once and storing it somewhere so that there are no common recursive calls to the same state.

V. Future Work

This idea can be expanded to a 3x3x3 board. In that case the number of states would be quite large and applying a min-max algorithm may not work. More advanced algorithms like Q value learning can be implemented there.

CONCLUSION

As there was very little time left for this task, I could not set up the gym tic tac toe environment and have written the code in c++ itself as I am more comfortable in it, Apologies for the same. Still, it was great learning about the minimax algortihm

References

[1] https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-3-tic-tac-toe-ai-finding-optimal-move/

[2] https://www.youtube.com/watch?v=Ntu8nNBL28o&t=640s

[3] https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/