Assignment 2 Report

Submitted By : Ranjan Singh

Tic Tac Toe game : It's a two player game competing for making three consecutive blocks. Producing a complete game tree with every possible node is very lengthy. I put a check on terminal states and thus if it's a terminal state then no child. This will have a high impact on the game playing, and the response from the computer will be very fast, because it has to search fewer possibilities for finding the optimal path.

JSON data : The json data which I have submitted have a length of 5477. Let's see how that can be:

Very naive approximation is that there are 9 boxes, and 3 possibilities for each one.

Total possibilities = $3^9 = 19683$

Since I have not produced the child node after the terminal state. Suppose some branch reaches a terminal state in 5 moves (3 by first player and 2 by second player) then this node will have no child after that. So this has very few nodes than approximation.

Minimax algorithm:

- Develop the complete game tree, by considering all the possible moves for every state.
- Using the terminal node of the tree, update score for every state based on the min max:
  If the parent of the terminal node was max's turn then it will try to maximize its score so choose the maximum of all the child nodes of this parent and update the parent state's score. Else for minimum choose minimum of score and update the parent's score
  Do this process recursively till we reach zero state.
  That's the training.
- After that we can play.

Tic Tac Toe by minimax algorithm:

Since minimax is a deterministic algorithm that means the next step taken by players will be completely predictive if they are playing optimally. For example in the game of chess, there is always a best move and going for that move would never take you to an unpredicted result, so that’s a deterministic game.

Reinforcement Learning algorithm: One of the three paradigms of machine learning. It uses iterative process to train the nodes in the game tree.

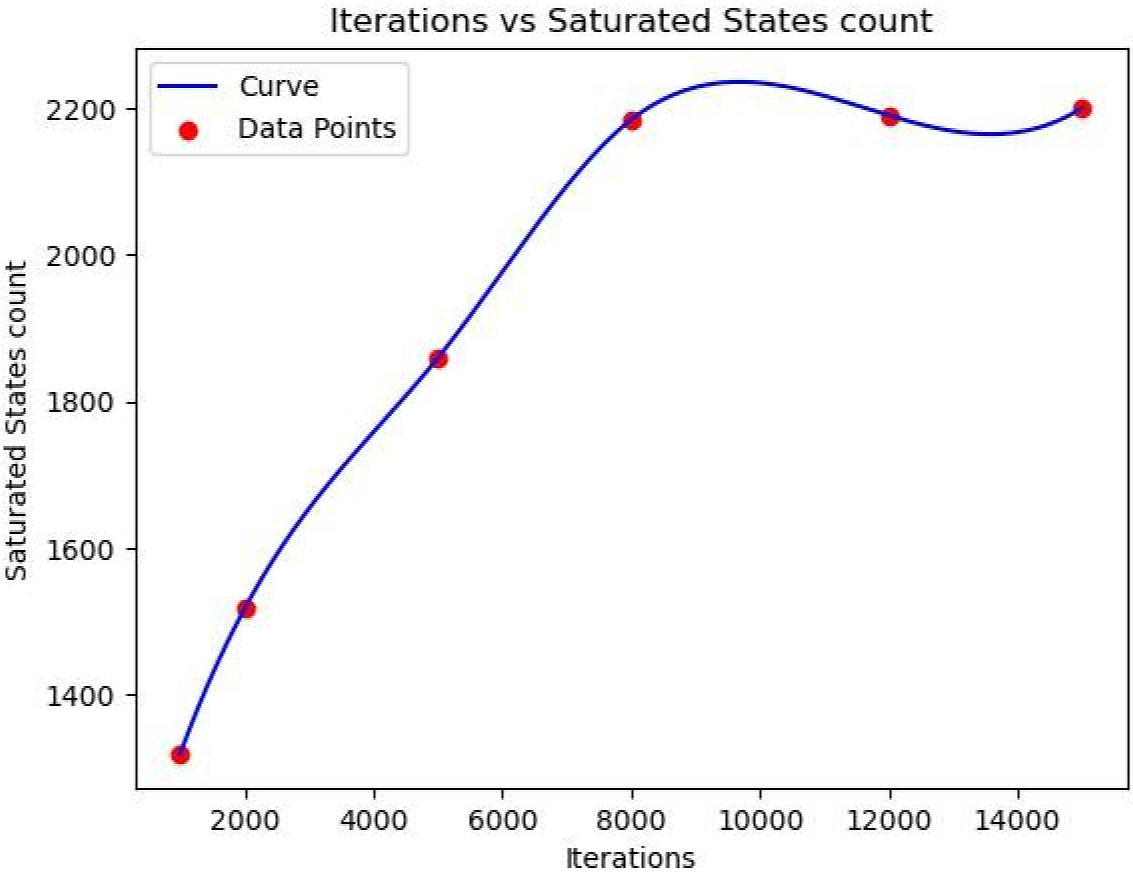
1) It starts from the zeroth level nodes.
2) Now we use this formula for finding the value of current state
   $V(S_t) \ = \ V(S_t) \ + \ \alpha * (V(S_t) - V(S_{t+1}))$
   Here $V(S_t)$ refers to value of the state S_t
   α is the learning rate
   $V(S_{t+1})$ refers to value of randomly chosen child of the state S_t
3) Iterate this process several times till saturation reaches. Here saturation means that after many iteration values of states will start converging to some value.

Tic Tac Toe by Reinforcement Learning:

This algorithm is a probabilistic algorithm. Since it is choosing the child randomly to train the value of parent nodes. So we can’t guarantee the output over the input given.

As we will increase the value of learning rate (α) the algorithm will require higher number of iteration to reach saturation so lower value of (α) is preferrable.

Here is a graph for analyzing that as we increase the number of iteration, more and more states start reaching the saturation point.


Iterations vs Saturated States count
Curve
Data Points
Saturated States count
1400
1600
1800
2000
2200
Iterations
2000
4000
6000
8000
10000
12000
14000