MLDS 422 - Intro to Python
Lab 7

Sungsoo Lim

November 9, 2023

NORTHWESTERN
UNIVERSITY

Today's Lab Materials

- Monte Carlo Tree Search
  - Algorithm Background
  - Implementation Details

Monte Carlo Tree Search

Figure: AlphaZero

- https://www.analyticsvidhya.com/blog/2019/01/monte-carlo-tree-search-introduction-algorithm-deepmin

Monte Carlo Tree Search

- With Go, the number of possible moves given a board position is very large
- AlphaZero simulates ahead from a given board position to the end of game (sample all possible next moves of all future board positions)
- The algorithm selectively selects from the possible next moves - 'promising ones' that lead to victory
- Based on the simulations, it calculates the win probabilities from the given position for each possible move and takes the highest move based on win probability

Monte Carlo Tree Search

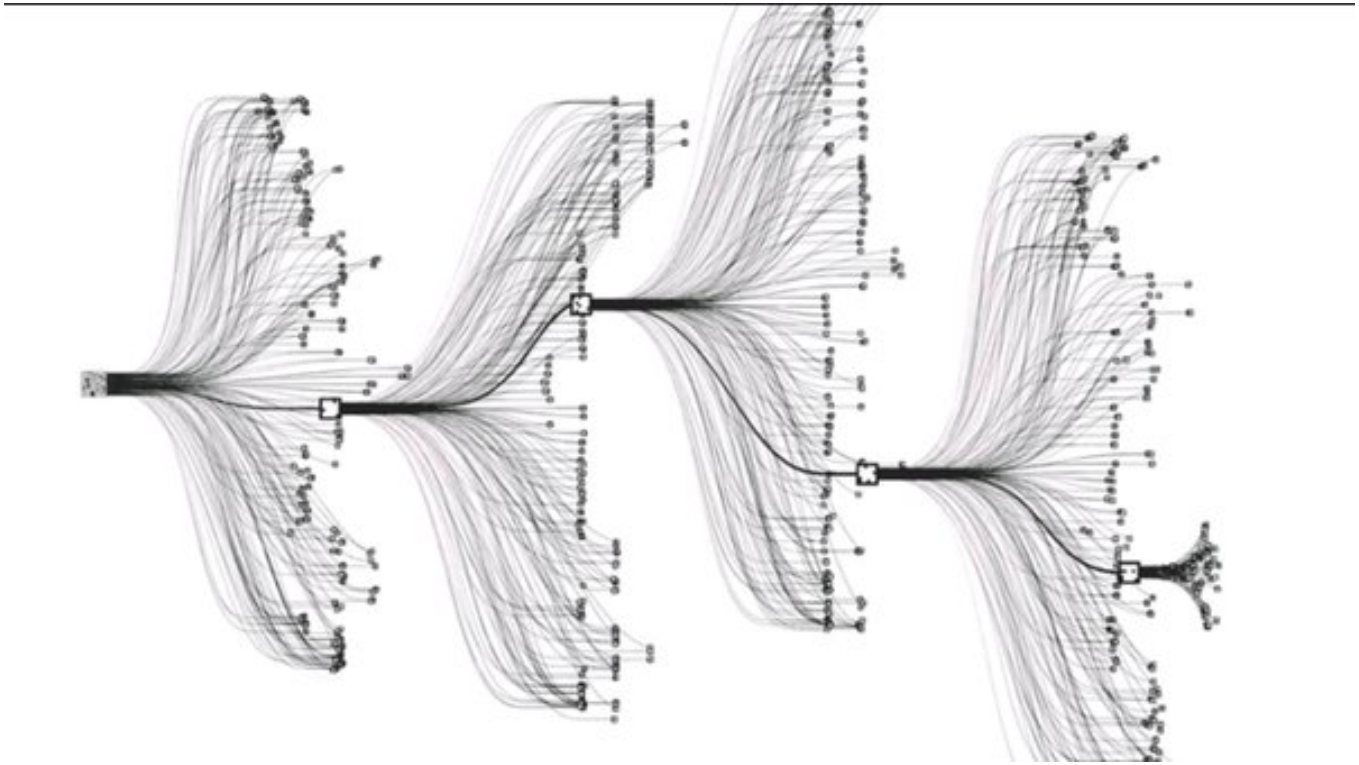

Figure: https://www.quora.com/What-does-it-mean-that-AlphaGo-relied-on-Monte-Carlo-tree-searc

Monte Carlo Tree Search

- Selection - from the root node, select a leaf node (this step needs to be selective - more promising moves are selected more often)
- Expansion - from the selected leaf node, add all its child nodes to the tree
- Simulation - from one of the child nodes, simulate to end of game
- Backpropagation - for the path from the root node to the child node, update their statistics
- Statistics - for the case of Go - did the game end in a win, tie, or loss (1, 0, -1). How many times did the path get visited?

Monte Carlo Tree Search

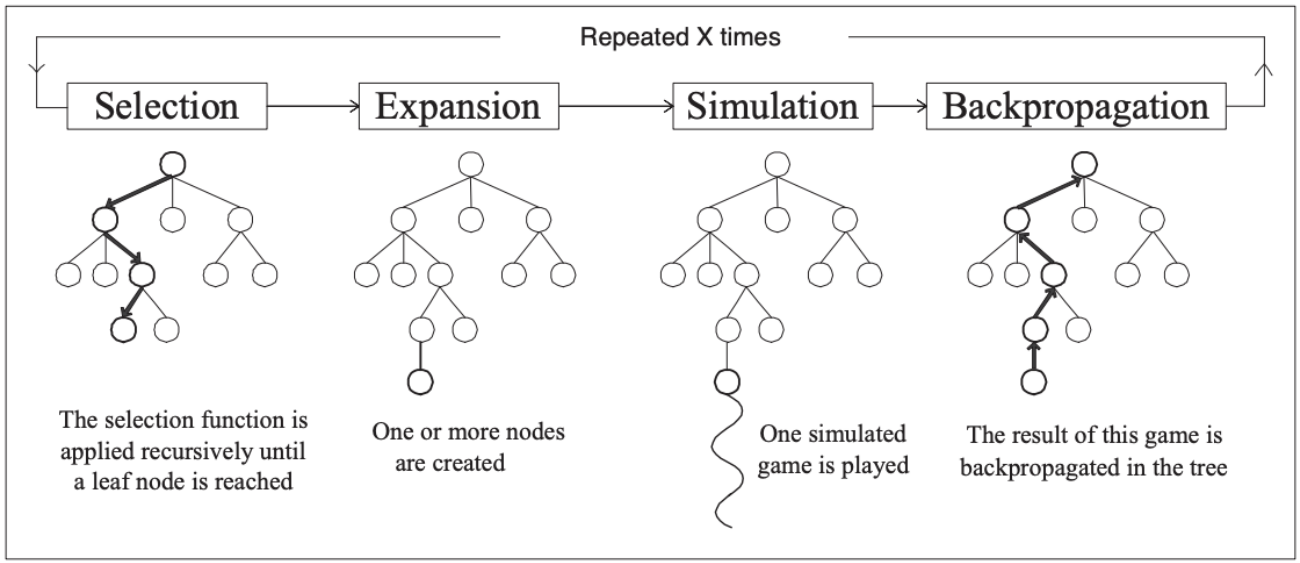

Figure 1: Outline of a Monte-Carlo Tree Search.

Figure: Sutton and Barto 2009

Monte Carlo Tree Search - Selection

- Upper Confidence Bounds algorithm (Auer *et al.* 2002)
- a - action (which move should be done to select the leaf node from the root node)
- A_s - the set of all possible moves from the root node s
- $Q(s, a)$ - the statistics we store (win, tie or loss) - if we select a from s, did it lead to win, tie or loss?
- c - a hyperparameter to balance the first term and the second term
- n_s - how many times the give root node was visited
- $n_{s,a}$ - how many times the move a was selected from s

$$a^* = \arg\max_{a \in A_s} \left[Q(s, a) + c\sqrt{\ln(n_s)/n_{s,a}} \right],$$