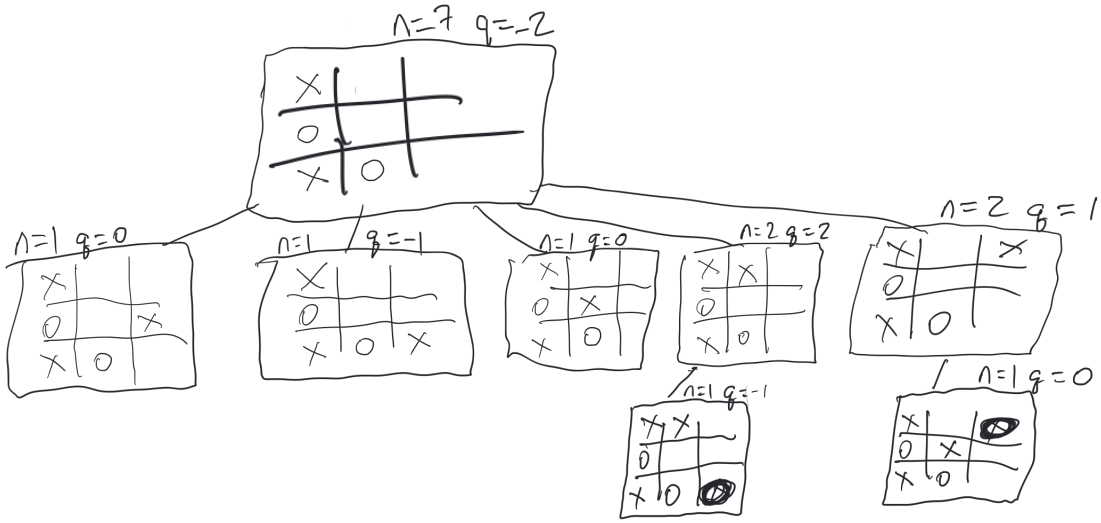
Reinforcement Learning—HW 9 Solution

April 13, 2020

Here's a partially-expanded Monte Carlo Search Tree for determining the next move in Tic-Tac-Toe:



Assume that c, the constant used for UCT (Upper Confidence Bound applied to Trees) is 0.5.

1. In this Tic-Tac-Toe game, whose turn is it (X or O)?

> **Solution:** Looking at the root node, we see that there are the same number of Xs and Os. Since X goes first, it is X's turn.

2. From your knowledge of Tic-Tac-Toe, what is the best move for that player?

Solution: The best solution would be either the center square or the top-right square. That sets up two in a row two times. O can block one of them, but that'll leave the other open so that X wins on the next turn.

3. For this question, you'll do one iteration of the MCST algorithm (Selection/Expansion/Simulation/Backup)

   (a) *Selection* What leaf node is the next to be selected (show your work)?

Solution: We start at the root node. Since it is fully expanded (five possible actions, five children), we must use the UCT formula on all its children and navigate to the highest UCT child.

The first three children all have the same value of n, and so all have the same exploration bonus in the UCT formula. Since $\frac{q(v)}{n(v)}$ is highest among the first and third child, those two will have the highest (and equal) UCT values among the first three children. The last two children also have equal values for n, but the second-from-the-right has the higher exploitation value. So, we must calculate two UCT values: for the first/third children, and for the fourth child.

For the first and third:

$$\begin{aligned} UCT(v) &= \frac{q(v)}{n(v)} + c\sqrt{\frac{\log n(v.parent)}{n(v)}} \\ &= \frac{0}{1} + 0.5\sqrt{\frac{\log 7}{1}} \\ &= 0 + .46 = .46 \end{aligned}$$

For the fourth child:

$$\begin{aligned} UCT(v) &= \frac{q(v)}{n(v)} + c\sqrt{\frac{\log n(v.parent)}{n(v)}} \\ &= \frac{2}{2} + 0.5\sqrt{\frac{\log 7}{2}} \\ &= 1 + .33 = 1.33 \end{aligned}$$

The second child has the highest UCT so is the next node. The second child is *not* fully expanded, so it is the selected node.

   (b) *Expansion* What child node is expanded?

Solution: There are three unexpanded children of the selected node. We could pick any one of them but happen to choose an action of placing an O in the middle square. The new child is:

```
X X _
O O _
X O _
```

(c) *Simulation* Do a rollout (simulation). Use some form of randomization (coin flips, random number generation in Excel or on a website, coin toss, etc.) to do the simulation. What is the result? If the result is a tie, rerun the simulation until you get a decisive winner.

Solution: I roll a three-sided die and it tells me to place an X at the bottom-right. Then, I flip a coin and it tells me to place an O at the middle-right. This is now a terminal state (O wins). The result value is -1 (since O wins).

(d) *Backup* Backup the result through the MCST.

Solution: We skip the rollout states and go first to the newly-expanded node. We set that to n=1 and q=zero minus the result (since the expanded node is a result of the O player going, the -1 result is a loss for X, but a win for O).

The newly-expanded node's parent (the root's fourth child) gets updated: n is incremented from 2 to 3, and q gets the result added to it (so q goes from 2 to 1).

That node's parent (the root) gets updated: n is incremented from 7 to 8, and q gets the result subtracted from it (again, every other level gets the result added to q or subtracted from q). q goes from -2 to -1.

(e) If this were the last simulation of the MCST, what action would be chosen and why?

Solution: There are two valid ways to choose the final action from the root:

- The action that leads to the child with the highest $\frac{q(v)}{n(v)}$. In this case, that'd be the fifth child (q/n = 1/2).
- The action that leads to the child with the highest $n(v)$ value. In this case, that'd be the fourth child (n=3). Theory here is this is the child that has had the most rollouts, so the one that has been most promising.