

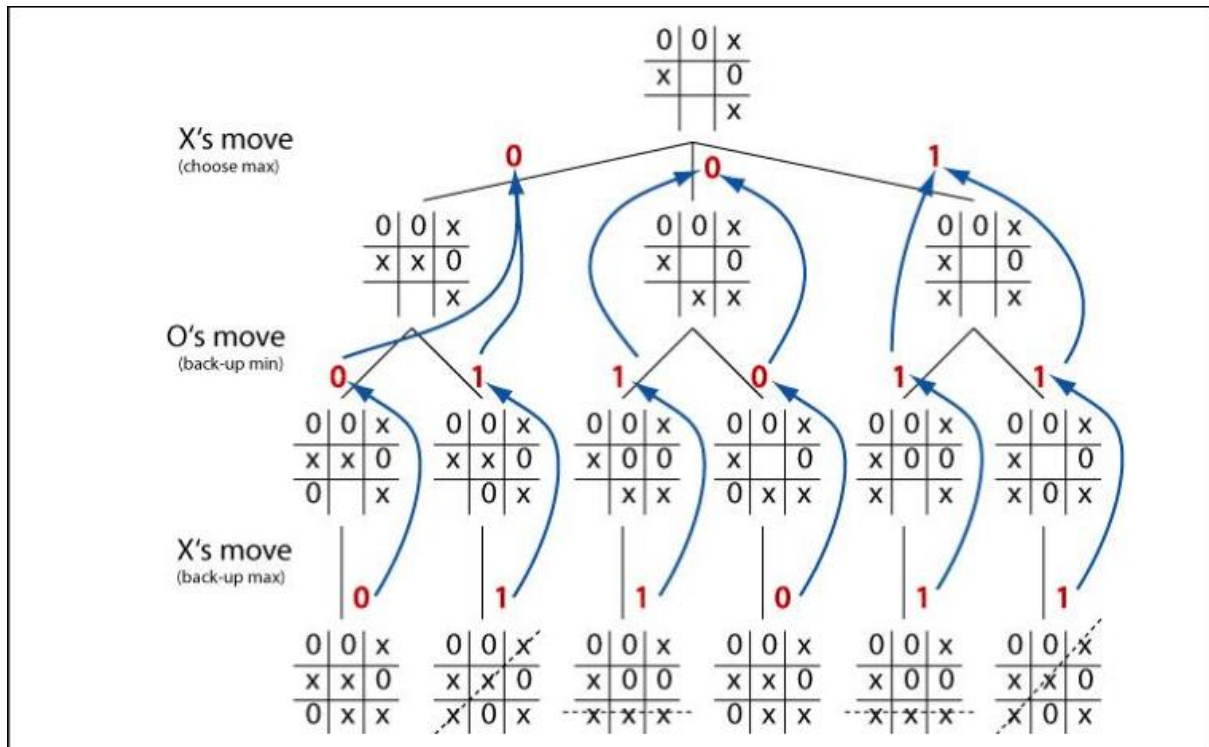
Minimax Algorithm이란?

한국말로 최소극대화. 미니맥스 규칙에서 유래했으며 이를 구현하기 위한 알고리즘. 개념의 시작은 롤스의 '정의론'에서 유래했다. 발생가능한 손실(Maximum)을 최소화(Minimize)하는 규칙이다. 최소극대화 외에 최대극대화, 최소극소화, 최대극소화 규칙도 있다.

게임이론에서 Minimax algorithm을 예시로 풀어 쉽게 설명하자면, 어벤저스에서 닥터 스트레인지가 사용한 방법을 보면 된다. 닥터 스트레인지는 작중 악당 "타노스"와 싸우기 위해서 미래의 1,400만 개의 모든 경우의 수를 살펴보고 그 중에 최선의 수를 알아내려고 하는데, Minimax Algorithm 역시 이와 유사하다.



Tic Tac Toe에서의 Minimax Algorithm의 원리를 다음과 같이 트리의 형태로 도식화할 수 있는데, 이런 식으로 기대되는 결과값을 다른 값으로 저장한 후에 어떤 선택을 했을시에 원하는 결과값을 얻을 수 있는지 계산한다고 보면 된다.



1

¹ https://www.researchgate.net/figure/Game-tree-for-Tic-Tac-Toe-game-using-MiniMax-algorithm_fig1_262672371

```

139     # 미니맥스
140
141     def minimax(self, board, maximizing):
142
143         # terminal case, 최종결과
144         case = board.final_state()
145
146         # player 1 wins +1
147         if case == 1:
148             return 1, None # eval, move
149
150         # player 2 wins -1
151         if case == 2:
152             return -1, None
153
154         # draw 0
155         elif board.isfull():
156             return 0, None
157
158         if maximizing:
159             max_eval = -100
160             best_move = None
161             empty_sqr = board.get_empty_sqr()
162
163             for (row, col) in empty_sqr:
164                 temp_board = copy.deepcopy(board)
165                 temp_board.mark_sqr(row, col, 1)
166                 eval = self.minimax(temp_board, False)[0]
167                 if eval > max_eval:
168                     max_eval = eval
169                     best_move = (row, col)
170
171             return max_eval, best_move

```

플레이어 1이 이기면 +1, 플레이어 2가 이기면 -1, 무승부는 0 (case1,2에 해당하지 않고 board.isfull()이 입력되면 0이 출력되게 만들었다.)

```

171         return max_eval, best_move
172
173     elif not maximizing:
174         min_eval = 100
175         best_move = None
176         empty_sqr = board.get_empty_sqr()
177
178         for (row, col) in empty_sqr:
179             temp_board = copy.deepcopy(board)
180             temp_board.mark_sqr(row, col, self.player)
181             eval = self.minimax(temp_board, True)[0]
182             if eval < min_eval:
183                 min_eval = eval
184                 best_move = (row, col)
185
186         return min_eval, best_move
187

```

참고:

¹ https://www.researchgate.net/figure/Game-tree-for-Tic-Tac-Toe-game-using-MiniMax-algorithm_fig1_262672371

<http://www.saegeoje.com/news/articleView.html?idxno=202785>

<https://youtu.be/Bk9hINZc6sE>

<https://github.com/AlejoG10/python-tictactoe-ai-yt>