(Part I): Solving Tic-Tac-Toe Using Minimax ¶

BY SIRISH PRABAKAR, DANE JEW

In [17]:
```
import random
import time

current = []


def show(board, player):
    print(player, ":\n")
    for i in range(3):
        for j in range(9):
            if j // 3 == i:
                print(" ", board[j], end="  |")
        print()
        print("------------------")


def check_line(char, pos1, pos2, pos3):
    return pos1 == pos2 == pos3 == char


def check_all(board, char):
    if check_line(char, board[0], board[1], board[2]):
        return True
    elif check_line(char, board[3], board[4], board[5]):
        return True
    elif check_line(char, board[6], board[7], board[8]):
        return True
    elif check_line(char, board[0], board[3], board[6]):
        return True
    elif check_line(char, board[1], board[4], board[7]):
        return True
    elif check_line(char, board[2], board[5], board[8]):
        return True
    elif check_line(char, board[0], board[4], board[8]):
        return True
    elif check_line(char, board[2], board[4], board[6]):
        return True
    else:
        return False


def check_draw(board):
```

```
    # check if all the night cells have been used on the board (i.e., there is no number left)
    return set(board) == {"o", "x"}


# check if the game is over.   If that is the case, return the score
def check_game_over(board):
    if check_all(board, "x"):
        return 10
    elif check_all(board, "o"):
        return -10
    elif check_draw(board):
        return 0
    else:
        return False



def minimax(board, player):    # return the minimax score of a node
    global current
    current_score = check_game_over(board)

    if current_score is not False:
        return current_score

    # if the game is not over, do the following
    scores = []
    moves = []
    x_win = False
    o_win = False
    for i in range(9):
    # check all possible moves.
```

```
if player == "x":
# Find the move with the highest score.   Add that move to current and return that score.



elif player == "o":
# Find the move with the lowest score.   Add that move to current and return that score.
```

Optimal vs Optimal

In [18]:
```
def optimal_vs_optimal():
    global current
    board = [0, 1, 2, 3, 4, 5, 6, 7, 8]
    current = []
    print("Player x and Player o Both play optimally.\n")

    show(board, "Board")
    print()
    curr = ["x", "o"]
    i = 0


    while True:

        minimax(board, curr[i])
        show(current[len(current) - 1], curr[i])
        board = current[len(current) - 1]

        print()

        if check_all(board, curr[i]):
            print(curr[i] + " Wins!")
            return curr[i]
        elif check_draw(board):
            print("Draw!")
            return "Draw"
        i = (i + 1) % 2

#start_time = time.time()
#optimal_vs_optimal()
#print("\nSeconds Elapsed:", time.time() - start_time)
```

In [19]:
```
optimal_vs_optimal()
```

```
Player x and Player o Both play optimally.

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------

x :

  x  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------

o :

  x  |  1  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  8  |
------------------

x :

  x  |  x  |  2  |
------------------
  3  |  o  |  5  |
------------------
  6  |  7  |  8  |
------------------

o :

  x  |  x  |  o  |
------------------
  3  |  o  |  5  |
```

```
-----------------
  6  |  7  |  8  |
-----------------

x :

  x  |  x  |  o  |
-----------------
  3  |  o  |  5  |
-----------------
  x  |  7  |  8  |
-----------------

o :

  x  |  x  |  o  |
-----------------
  o  |  o  |  5  |
-----------------
  x  |  7  |  8  |
-----------------

x :

  x  |  x  |  o  |
-----------------
  o  |  o  |  x  |
-----------------
  x  |  7  |  8  |
-----------------

o :

  x  |  x  |  o  |
-----------------
  o  |  o  |  x  |
-----------------
  x  |  o  |  8  |
-----------------

x :

  x  |  x  |  o  |
```

```
-----------------
  o  |  o  |  x  |
-----------------
  x  |  o  |  x  |
-----------------

Draw!
```

Out[19]: 'Draw'

Random vs Optimal

In [42]:
```
def random_vs_optimal():
    global current
    board = [0, 1, 2, 3, 4, 5, 6, 7, 8]
    current = []
    print("Player x plays randomly, and Player o plays optimally.\n")
    show(board, "Board")
    print()
    curr = ["x", "o"]
    i = 0

    while True:
        print()
        if curr[i] == "x":
            valid_choice = [i for i in range(9) if board[i] != "x" and board[i] != "o"]
            ran = valid_choice[int(random.random() * len(valid_choice))]
            board[ran] = "x"
            show(board, "x")

        elif curr[i] == "o":
            minimax(board, "o")
            show(current[len(current) - 1], "o")
            board = current[len(current) - 1]

        print()

        if check_all(board, curr[i]):
            print(curr[i] + " Wins!")
            return curr[i]
        elif check_draw(board):
            print("Draw!")
            return "Draw"
        i = (i + 1) % 2

#start_time = time.time()
#random_vs_optimal()
#print("\nSeconds Elapsed:", time.time() - start_time)
```

In [43]:
```
random_vs_optimal()
```

```
Player x plays randomly, and Player o plays optimally.

Board :

  0  |  1  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


x :

  x  |  1  |  2  |
-----------------
  3  |  4  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


o :

  x  |  1  |  2  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


x :

  x  |  1  |  x  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


o :
```

```
  x  |  o  |  x  |
-----------------
  3  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


x :

  x  |  o  |  x  |
-----------------
  x  |  o  |  5  |
-----------------
  6  |  7  |  8  |
-----------------


o :

  x  |  o  |  x  |
-----------------
  x  |  o  |  5  |
-----------------
  6  |  o  |  8  |
-----------------

o Wins!
```

Out[43]: 'o'

You vs Optimal

In [44]:
```
def you_vs_optimal():
    global current
    board = [0, 1, 2, 3, 4, 5, 6, 7, 8]
    current = []
    print("You play as Player x, can you win the game?\n")
    show(board, "Board")
    print()
    curr = ["x", "o"]
    i = 0
    while True:
        if curr[i] == "x":
            valid_choice = [str(i) for i in range(9) if i in board]
            while True:
                cell = input("Please enter a valid cell (" + ", ".join(valid_choice) + "): ")
                if cell in valid_choice:
                    break
            print()
            board[int(cell)] = curr[i]
            show(board, "x")
        elif curr[i] == "o":
            minimax(board, "o")
            show(current[len(current) - 1], "o")
            board = current[len(current) - 1]
        print()
        if check_all(board, curr[i]):
            print(curr[i] + " Wins!")
            return curr[i]
        if check_draw(board):
            print("Draw!")
            return "Draw"
        i = (i + 1) % 2

you_vs_optimal()
```

```
You play as Player x, can you win the game?

Board :

  0  |  1  |  2  |
------------------
  3  |  4  |  5  |
------------------
  6  |  7  |  8  |
------------------

Please enter a valid cell (0, 1, 2, 3, 4, 5, 6, 7, 8): 5

x :

  0  |  1  |  2  |
------------------
  3  |  4  |  x  |
------------------
  6  |  7  |  8  |
------------------

o :

  0  |  1  |  o  |
------------------
  3  |  4  |  x  |
------------------
  6  |  7  |  8  |
------------------

Please enter a valid cell (0, 1, 3, 4, 6, 7, 8): 4

x :

  0  |  1  |  o  |
------------------
  3  |  x  |  x  |
------------------
  6  |  7  |  8  |
------------------

o :
```

```
  0  |  1  |  o  |
-----------------
  o  |  x  |  x  |
-----------------
  6  |  7  |  8  |
-----------------

Please enter a valid cell (0, 1, 6, 7, 8): 3
Please enter a valid cell (0, 1, 6, 7, 8): 8

x :

  0  |  1  |  o  |
-----------------
  o  |  x  |  x  |
-----------------
  6  |  7  |  x  |
-----------------

o :

  o  |  1  |  o  |
-----------------
  o  |  x  |  x  |
-----------------
  6  |  7  |  x  |
-----------------

Please enter a valid cell (1, 6, 7): 7

x :

  o  |  1  |  o  |
-----------------
  o  |  x  |  x  |
-----------------
  6  |  x  |  x  |
-----------------

o :

  o  |  o  |  o  |
```

```
-----------------
  o  |  x  |  x  |
-----------------
  6  |  x  |  x  |
-----------------

o Wins!
```

Out[44]: 'o'