Tic-Tac-Toe Solver Game : Report

Title: Tic-Tac-Toe Solver Game

Name: Yash Kalra

Roll no:202401100300284

Class: CSE(AI)-D

Date: 11/03/2025

Introduction

Tic-Tac-Toe is a basic two-player game in which players take turns putting their marks ('X' and 'O') on a 3x3 grid. The goal is to have three of one's marks in a row, column, or diagonal.

The task of this problem is to simulate a Tic-Tac-Toe game in which one of the players is served by a machine learning agent (AI) learning to play perfectly using Q-learning, and the other player being a human. The AI has to make a choice based on what it learned, while the user plays through giving their actions.

Methodology

Game Setup:

Tic-Tac-Toe is a 3x3 grid game. One player is AI (X), and the other is a human player (O).

AI learns to play using Q-learning (a type of machine learning).

Q-Learning Basics:

The AI learns from experience by playing many games.

State: The current board configuration.

Action: The AI's move (placing X on the grid).

Training the AI:

The AI plays multiple games against a random player (representing the human player).

It makes its moves based on Q-values (which are kept in a table).

With each game, the AI figures out which actions to take to result in a good outcome.

Game Flow:

User's Move: The user chooses a location on the grid (e.g. firstly row then column).

AI's Move: The AI makes its best move from what it has learned.

The game goes on until a winner (three in a row) or the board is filled (draw).

Winning Conditions:

The game verifies after every move for a winner (three marks in a row, column, or diagonal).

If nobody wins and the board is full, it's a draw.

Code

```
import random


# Function to print the game board
def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 9)


# Function to check if a player has won
def check_winner(board, player):
    for row in board:
        if all(s == player for s in row):
            return True

    for col in range(3):
        if all(row[col] == player for row in board):
            return True

    if all(board[i][i] == player for i in range(3)) or all(board[i][2 - i] == player for i in range(3)):
        return True

    return False


# Function to check if the board is full
```

```
def is_full(board):
    return all(cell != " " for row in board for cell in row)


# Function for the AI to make a move
def ai_move(board):
    empty_cells = [(i, j) for i in range(3) for j in range(3) if board[i][j] == " "]
    return random.choice(empty_cells)


# Main game loop
def tic_tac_toe():
    board = [[" " for _ in range(3)] for _ in range(3)]
    players = ["User", "AI"]
    symbols = {"User": "X", "AI": "O"}
    current_player = "User"

    print("Welcome to Tic Tac Toe!")
    print_board(board)

    while True:
        if current_player == "User":
            # User move
            try:
                row = int(input("Enter row (0-2): "))
                col = int(input("Enter column (0-2): "))
                if board[row][col] != " ":
                    print("Cell already taken, try again.")
                continue
            except (ValueError, IndexError):
                print("Invalid input, try again.")
                continue

        else:  # AI move
            print("AI is making a move...")
            row, col = ai_move(board)

        board[row][col] = symbols[current_player]
        print_board(board)

        if check_winner(board, symbols[current_player]):
            print(f"{current_player} wins!")
            break

        if is_full(board):
            print("It's a tie!")
            break

        # Switch player
        current_player = "AI" if current_player == "User" else "User"

# Start the game
tic_tac_toe()
```

Output

```
Welcome to Tic Tac Toe!
  |   |
---------
  |   |
---------
  |   |
---------
Enter row (0-2): 0
Enter column (0-2): 1
  | X |
---------
  |   |
---------
  |   |
---------
AI is making a move...
O | X |
---------
  |   |
---------
  |   |
---------
Enter row (0-2): 1
Enter column (0-2): 1
O | X |
---------
  | X |
---------
  |   |
---------
AI is making a move...
O | X | O
---------
  | X |
---------
  |   |
---------
Enter row (0-2): 2
Enter column (0-2): 1
O | X | O
---------
  | X |
---------
  | X |
---------
User wins!
```

```
   ---------
   AI is making a move...
   X | X | O
   ---------
   X | O | X
   ---------
   O |   | O
   ---------
   AI wins!
```

```
Enter row (0-2): 0
Enter column (0-2): 2
X | O | X
---------
X | X | O
---------
O | X | O
---------
It's a tie!
```

References

- Python libraries
- Wikipedia
- chatgpt