

Taller MLP

Profesor: Angel Augusto Agudelo Zapata

Andres Felipe Gordillo Guerrero

1088348241

a.gordillo@utp.edu.co

Inteligencia Artificial

Universidad tecnológica de pereira

2024 - Semestre II

Documentación del Código: Juego de Tic-Tac-Toe con Aprendizaje

Este documento describe el código que implementa un juego de Tic-Tac-Toe con la capacidad de leer datos de entrenamiento, jugar partidas aleatorias, registrar resultados y graficar el aprendizaje de los jugadores.

Este código es funcional de un juego de Tic-Tac-Toe que no solo juega el juego, sino que también puede aprender de las partidas anteriores. Las capacidades de aprendizaje y visualización de resultados lo convierten en una herramienta interesante para analizar estrategias y rendimiento en el juego.

Clase `TicTacToeGame`

La clase `TicTacToeGame` representa el juego de Tic-Tac-Toe y maneja la lógica del juego, el registro de resultados y la visualización.

Atributos

- **`trainings`:**
 - **Tipo:** `list`
 - **Descripción:** Almacena los datos de entrenamiento, cada uno representando un movimiento y el estado del tablero.
- **`board`:**
 - **Tipo:** `list[list[str]]`
 - **Descripción:** Matriz 3x3 que representa el tablero del juego, inicializada con '-' (celdas vacías).
- **`current_player`:**
 - **Tipo:** `str`
 - **Descripción:** Representa al jugador actual, que puede ser 'X' o 'O'.
- **`scores`:**
 - **Tipo:** `dict`
 - **Descripción:** Diccionario que lleva un registro de las victorias de cada jugador y los empates.

Métodos

1. `__init__(self)`

- **Descripción:** Constructor que inicializa los atributos de la clase.
- **Acciones:**
 - Inicializa `trainings` como una lista vacía.
 - Inicializa `board` como una matriz 3x3 de celdas vacías.

- Establece `current_player` como 'X'.
- Inicializa `scores` para contar las victorias y empates.

2. `read_training_file(self, filename)`

- **Descripción:** Lee un archivo de texto que contiene datos de entrenamiento y los almacena en `trainings`.
- **Parámetros:**
 - `filename`: Ruta al archivo que se desea leer.
- **Acciones:**
 - Analiza cada línea para extraer movimientos y el estado del tablero.
 - Maneja errores relacionados con archivos no encontrados o formatos incorrectos.

3. `print_trainings(self)`

- **Descripción:** Imprime en consola los datos de entrenamiento almacenados en `trainings`.
- **Acciones:**
 - Si no hay datos de entrenamiento, informa al usuario.
 - Para cada entrada en `trainings`, imprime el movimiento y el estado del tablero.

4. `print_board(self)`

- **Descripción:** Muestra el estado actual del tablero en la consola.
- **Acciones:**
 - Imprime cada fila del tablero en la consola.

5. `check_winner(self)`

- **Descripción:** Verifica si hay un ganador en el juego.
- **Retorna:**
 - `str` o `None`: Devuelve el símbolo del ganador ('X' o 'O') o `None` si no hay ganador.
- **Acciones:**
 - Comprueba las filas, columnas y diagonales del tablero en busca de un ganador.

6. `play_game(self)`

- **Descripción:** Inicia una partida de Tic-Tac-Toe.
- **Retorna:**
 - `str`: El ganador ('X', 'O') o 'draw' si hay empate.
- **Acciones:**
 - Alterna entre jugadores ('X' y 'O') y realiza movimientos aleatorios.
 - Actualiza el puntaje según el resultado.

7. `get_random_move(self)`

- **Descripción:** Genera un movimiento aleatorio para el jugador actual.
- **Retorna:**
 - `tuple[int, int]`: Una tupla con la fila y columna del movimiento seleccionado.
- **Acciones:**
 - Selecciona aleatoriamente una celda vacía en el tablero.

8. `plot_results(self, iterations)`

- **Descripción:** Imprime los resultados finales después de un número determinado de partidas y grafica el aprendizaje.
- **Parámetros:**
 - `iterations`: Número total de partidas jugadas.
- **Acciones:**
 - Imprime las victorias de cada jugador y el número de empates.
 - Utiliza `matplotlib` para graficar las victorias y la tasa de aprendizaje (promedio móvil) a lo largo de las partidas.

Función `main()`

La función `main()` es la entrada principal del programa, que realiza las siguientes acciones:

- Crea una instancia de `TicTacToeGame`.
- Lee los datos de entrenamiento desde un archivo.
- Ejecuta un número máximo de partidas, verificando empates consecutivos.
- Llama al método `plot_results` para mostrar los resultados finales.

Ejecución del Programa

La línea `if __name__ == "__main__":` asegura que la función `main()` se ejecute cuando el script se corre directamente, permitiendo iniciar el juego.

Dependencias

Este código requiere las siguientes bibliotecas:

- `random`: Para generar movimientos aleatorios.
- `matplotlib.pyplot`: Para crear gráficas de resultados.
- `numpy`: Para operaciones matemáticas, como el promedio móvil.