Aplicación de aprendizaje por refuerzo mediante Alpha Zero en el juego de Ultimate Tic-Tac-Toe

David Moisés Aguilar Paredes, *20141933, Ingeniería Mecatrónica,* Ivonne Rocío Heredia León, *20151795, Ingeniería Industrial,* Eduardo Andre Cuya Nizama, *20151192, Ingeniería Industrial,* y Carlos Alberto Sosa Pezo, *20161310, Ingeniería Informática.*

***Resumen*—Ultimate tic-tac-toe es una variante del juego de tres en raya que, a diferencia del juego base, no resulta trivial para algoritmos de decisión como minimax o poda alfa-beta debido a la cantidad de estados posibles. En este artículo, se propone la aplicación del algoritmo de aprendizaje por refuerzos AlphaZero en dicho juego, comparando su desempeño frente a un árbol de búsqueda de Monte Carlo y deep Q-learning. Se espera que AlphaZero alcance un nivel de habilidad semejante o mayor a dichos algoritmos.**

I. Introducción

El presente proyecto busca analizar la performance de tres agentes que utilizan distintos métodos de inteligencia artificial por medio del juego Ultimate Tic Tac Toe, el cual consiste en jugar una partida de Tres en raya donde cada casilla es a su vez un tablero de Tres en raya. Este juego ha sido analizado por diversos investigadores [1], usuarios de Github [2] e incluso existen papers en IEEE [3]; Sin embargo, no se ha realizado una comparación de desempeño entre los algoritmos Alpha Zero, Monte Carlo y deep Q learning. Debido a que existe una constante búsqueda por incrementar el desempeño de estos algoritmos con el objetivo de mejorarlos es que es importante realizar este análisis. Aun con las restricciones de tiempo de entrenamiento y recursos computacionales bajo los que se realiza este experimento, en comparación con la implementación original de AlphaZero por parte de DeepMind [4], se espera que el modelo mejore su rendimiento en el juego de Ultimate tic-tac-toe con los sucesivos entrenamientos y pueda presentar una habilidad semejante o superior a las de otros algoritmos con los que se va a enfrentar. Esto se medirá mediante el número de victorias obtenidas frente a estos algoritmos en diferentes estados de aprendizaje, a los que se espera que venza en por lo menos un 50% de las veces en sus etapas finales.

II. Metodología

A. Tópicos de IA

- Busqueda Adversarial
  Son algoritmos de búsqueda en los que dos o más jugadores con objetivos en conflicto intentan explorar el mismo espacio de búsqueda para hallar la solución óptima. Cada agente debe considerar la acción de otro agente y el efecto de esa acción en su desempeño. Las técnicas que se implementarán serán minimax, poda alfabeta y Monte Carlo Tree Search.
- Aprendizaje por refuerzo
  Es un área de aprendizaje automático relacionada con la forma en que los agentes de software deben tomar medidas en un entorno para maximizar la noción de recompensa acumulativa. Los algoritmos que se implementarán serán Alpha Zero y Deep Q learning.

B. Infraestructura

Se implementará un entorno virtual del juego de Ultimate tic-tac-toe,también conocido como super tic-tac-toe o estratégico tic-tac-toe. El juego está compuesto por nueve tableros de tres en raya dispuestos en una cuadrícula de 3×3. Los jugadores se turnan para jugar en los tableros de tres en raya más pequeños hasta que uno de ellos gane en el tablero de tres en raya más grande. En comparación con el tradicional tic-tac-toe, la estrategia en este juego es conceptualmente más difícil y ha demostrado ser más desafiante para las computadoras.

C. Posibles técnicas base (baselines) para comparación

El uso de técnicas adversariales en Ultimate Tic Tac Toe (UTTT) son numerosas ([1],[5], [6]). La implementación de MiniMax de Amar y Binyamin [5], utiliza profundidad limitada para acelerar la respuesta del algoritmo; asimismo, se utilizaron tres funciones diferentes para evaluar la conveniencia de los estados posibles, una suerte de heurística, la primera de ellas propone la visualización del movimiento anterior en el tablero pequeño para decidir sobre la mejor opción, la cual es dada por una función de pesos creada por los autores; la segunda propone asignar un valor constante muy alto a los estados terminales donde gana Max y un valor negativo en donde gana Min, en caso exista un empate se asignará un valor de cero para ese estado y en caso no sea un estado terminal asigna un valor dado por la suma de los pesos por las celdas ganadas menos la suma de los pesos de las celdas por las celdas del enemigo, por último, la tercera es muy parecida a la segunda pero utiliza una función para asignar los valores más compleja. Chen, Doan y Xu [1] utilizan técnicas como minimax implementado con alfabeta, Monte Carlo Tree Search (MCTS), Deep Q learning y un modelo híbrido entre minimax y MCTS. la función de evaluación de minimax utilizada es la cantidad de mini tableros ganados por Max menos los mini tableros perdidos.

Como baselines, se utilizarán los algoritmos Monte Carlo Tree Search. y el modelo de Deep Q-learning implementado

por Chen,Doan y Xu [1]. Debido a que Minimax y alfabeta tienen una complejidad temporal muy grande.

D. Posibles métricas de evaluación

- Complejidad Espacial
  Esta métrica nos permitirá obtener un estimado del uso de memoria principal expresado mediante una función según el tamaño de la entrada. El resultado es expresado usualmente en notación O grande.
- Complejidad Temporalidad
  Esta métrica nos permitirá obtener un estimado del tiempo de ejecución expresado en función del tamaño de la entrada. El resultado es expresado usualmente en notación O grande.
- Optimalidad
  Comparación del ratio de victorias de un algoritmo frente a otros.

BIBLIOGRAFÍA

[1] P. and Jesse Doan and Edward Xu. (2018). Ai agents for ultimate tic-tac-toe, [Online]. Available: https://web.stanford.edu/~jdoan21/cs221paper.pdf.

[2] Shayakbanerjee. (2017). Reinforcement learning based ultimate tic tac toe player, [Online]. Available: https://github.com/shayakbanerjee/ultimate-ttt-rl.

[3] Sneha Garg, Dalpat Songara, and Saurabh Maheshwari. (2017). The winning strategy of tic tac toe game model by using theoretical computer science, [Online]. Available: https://ieeexplore.ieee.org/document/8003944.

[4] DeepMind. (2017). Mastering chess and shogi by self-play with ageneral reinforcement learning algorithm, [Online]. Available: https://arxiv.org/pdf/1712.01815.pdf.

[5] E. Adi Ben Binyamin. (2017). Ai agent for ultimate tic tac toe game, [Online]. Available: https://www.cse.huji.ac.il/~ai/projects/2013/U2T3P/files/AI_Report.pdf.

[6] Subrahmanya Sista. (2016). Adversarial game playing using monte carlo tree search, [Online]. Available: https : / / etd . ohiolink . edu / !etd . send_file ? accession = ucin1479820656701076&disposition=inline.