Aprendizaje por refuerzo: Juego de DAMAS 4x4 con algoritmo Q-Learning

Reinforcement learning: 4x4 CHECKERS game with Q-Learning algorithm

Autor: Sebastián Landaeta
sebastianlandaeta12@gmail.com
Docente: Manuel Paniccia
Universidad Nacional Experimental de Guayana
Ingeniería en Informática
Febrero de 2025

Resumen

Este proyecto es una aplicación práctica del aprendizaje por refuerzo (RL), una rama del Machine Learning (ML) que permite a las máquinas guiar su propio aprendizaje a través de recompensas y castigos. Se desarrolló una versión simplificada del juego de DAMAS, utilizando un tablero de 4x4 con dos fichas para cada jugador. El algoritmo de RL empleado fue Q-Learning, el cual se enfoca en encontrar la mejor estrategia o conjunto de acciones para maximizar la recompensa final. Este algoritmo se implementó para que la IA aprenda a jugar y pueda competir contra un oponente humano.

Después de varias partidas, en las que la IA aprendió dinámicamente a través de la interacción con el entorno, se observó que el agente es capaz de vencer a un humano, aunque su porcentaje de victorias sigue siendo inferior a las de este. Esto demuestra Q-Learning es un algoritmo superior al estudiado anteriormente, pero sigue sin ser perfecto. El proyecto no solo ilustra los fundamentos del aprendizaje

por refuerzo, sino que también sirve como base para explorar aplicaciones más avanzadas en entornos más complejos.

Palabras clave: Machine Learning, Aprendizaje por refuerzo, Q-Learning.

Abstract

This project is a practical application of reinforcement learning (RL), a branch of Machine Learning (ML) that enables machines to guide their own learning through rewards and punishments. A simplified version of the game of CHECKERS was developed, using a 4x4 board with two pieces for each player. The RL algorithm used was Q-Learning, which focuses on finding the best strategy or set of actions to maximize the final reward. This algorithm was implemented so that the AI could learn to play and compete against a human player.

After several games, in which the AI learned dynamically through interaction with the environment, it was observed that the agent is able to beat a human, although its percentage of victories is still lower than that of the latter. This shows that Q-Learning is a superior algorithm to the one previously studied, but it is still not perfect. The project not only illustrates the fundamentals of reinforcement learning, but also serves as a basis for exploring more advanced applications in more complex environments.

Keywords: Machine Learning, Reinforcement Learning, Q-Learning.

I. Introducción

El aprendizaje automático (Machine Learning, ML) ha revolucionado la forma en que las máquinas pueden aprender y adaptarse a entornos complejos sin necesidad de ser programadas explícitamente para cada tarea. Dentro de este campo, el aprendizaje por refuerzo (Reinforcement Learning, RL) se destaca como una técnica que permite a los agentes aprender a través de la interacción con su entorno, utilizando un sistema de recompensas y castigos para guiar su comportamiento.

Este trabajo se enfoca en la aplicación práctica del algoritmo Q-Learning en

un entorno simplificado pero representativo: un juego de DAMAS en un tablero de 4x4. A través de este proyecto, se busca ilustrar cómo un agente autónomo (IA) puede aprender a jugar y mejorar su estrategia compitiendo contra un jugador humano. El equilibrio entre exploración y explotación, junto con el diseño de un sistema de recompensas efectivo, son aspectos clave que se exploran para demostrar la eficacia del Q-Learning en la toma de decisiones estratégicas.

En las siguientes secciones, se detallará la implementación del Q-Learning en el juego de damas y se analizarán los resultados obtenidos. Este estudio no solo sirve como una introducción accesible al ML, sino que también sienta las bases para entender conceptos más avanzados, como el Deep Learning o las redes neuronales.

II. Bases teóricas

Aprendizaje Automático (Machine Learning)

"El Aprendizaje Automático es una rama de la Inteligencia Artificial que estudia sistemas capaces de aprender a realizar una tarea a partir de datos de ejemplo" (Quiroga & Lanzarini, 2019). El aprendizaje automático es una rama de la inteligencia artificial que se enfoca en desarrollar algoritmos y modelos que permiten a las máquinas aprender a partir de datos, sin ser programadas explícitamente para realizar una tarea específica. En lugar de seguir instrucciones paso a paso, los sistemas de aprendizaje automático identifican patrones en los datos y utilizan ese conocimiento para hacer predicciones, tomar decisiones o mejorar su desempeño en una tarea determinada.

El Machine Learning distingue 3 tipos principales de aprendizaje, el supervisado, el no supervisado, y el reforzado, pero para este proyecto, nos centraremos en el último mencionado.

Aprendizaje por Refuerzo (Reinforcement Learning)

"El AR es un enfoque de la IA en el que los agentes aprenden a partir de su interacción con el ambiente" (Fonseca, Martínez & Nowé, 2018). Se inspira en la psicología conductista y se centra en

cómo un agente debe tomar decisiones en un entorno para maximizar una noción de recompensa acumulativa. El aprendizaje reforzado se diferencia de otros paradigmas, como el aprendizaje supervisado, en que este no requiere pares de entrada/salida etiquetados ni correcciones explícitas de acciones subóptimas. En cambio, se enfoca en el equilibrio entre la **exploración** de nuevas acciones y la **explotación** del conocimiento existente para optimizar las decisiones tomadas por el agente.

Existen múltiples algoritmos basados en este tipo de aprendizaje, pero para este proyecto, usaremos uno llamado Q-Learning.

Q-Learning

Es un algoritmo que tiene como objetivo hacer que el agente aprenda una política, es decir, una estrategia que le dice al agente qué acción tomar en cada estado para maximizar su recompensa total a largo plazo. Este algoritmo se basa en un sistema de recompensas y castigos: Lo que busca el agente es maximizar recompensas, y minimizar castigos.

El algoritmo Q-Learning se compone de los siguientes componentes:

- Estado: Representa la situación actual del agente.

- Acción: Es una decisión que el agente puede tomar en un estado.

- Recompensa: Es el feedback recibido después de tomar una acción acertada en un estado.

- Castigo: Es el feedback recibido después de tomar una acción errónea en un estado.

- Función Q: Es una tabla o matriz que asocia cada par (s, a) (estado s y acción a) con un valor $Q(s, a)$. Este valor representa la recompensa acumulativa esperada que el agente puede obtener si toma la acción a en el estado s y luego sigue la política óptima a partir de ese momento.

  El Q-Learning actualiza la función Q utilizando la ecuación de Bellman:

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma max_{a'} Q(s',a') - Q(s,a)]$$

Donde:

s es el estado anterior.

a es la acción tomada.

r es la recompensa recibida.

s' es el nuevo estado.

a' es la nueva acción.

γ es el factor de descuento (controla la importancia de las recompensas futuras).

α es la tasa de aprendizaje (controla cuánto se actualiza $Q(s, a)$.

III. Funcionamiento

Para el programa que se realizó, se utilizó este algoritmo para entrenar a una IA y que sea capaz de jugar a las DAMAS contra un jugador humano. Cabe aclarar, que estamos trabajando una versión simplificada del juego, en el que la tabla es 4x4, las fichas pueden moverse diagonalmente hacía adelante y hacia atrás, sin necesidad de ser fichas reina, y cada jugador solo cuenta con 2 fichas.

A continuación, se explica cómo se implementa el Q-Learning en el código y cómo funciona en el contexto del juego:

1. Inicialización de la tabla Q

La tabla Q se inicializa como un diccionario vacío al comienzo del programa. Esta tabla almacenará los valores Q para cada par estado-acción.

- Estado: Representa la configuración actual del tablero, codificada como una tupla de tuplas.

- Acción: Representa un movimiento válido de una ficha, codificado como una tupla (x, y, i, j). Donde (x, y) es la posición actual de la ficha y (i, j) es la nueva posición.

2. Selección de acciones (Exploración vs. Explotación)

Las acciones se seleccionan a partir de una política ϵ-greedy, la cual es una estrategia fundamental en el aprendizaje por refuerzo que equilibra dos aspectos clave: la exploración y la explotación.

Exploración: Se refiere a la acción de probar nuevas opciones o estrategias que el agente no ha experimentado antes. El objetivo es descubrir información nueva sobre el entorno,

como recompensas potenciales o estados desconocidos.

Explotación: Se refiere a la acción de utilizar el conocimiento que el agente ya ha adquirido para maximizar las recompensas inmediatas. Es decir, el agente elige la acción que, según su experiencia, le dará la mayor recompensa.

ϵ-greedy equilibra esos aspectos utilizando un parámetro épsilon (ϵ), que es un valor entre 0 y 1.

- Con probabilidad ϵ, el agente explora (elige una acción aleatoria).

- Con probabilidad $1 - \epsilon$, el agente explota (elige la acción con el mayor valor Q conocido).

En el código, ϵ empezará teniendo un valor de 0.5, o sea que al comienzo habrá una probabilidad del 50% de que explore y 50% de probabilidad de que explote. Pero conforme la IA vaya jugando partidas, ese valor irá decreciendo gracias a una épsilon decay (ϵ-decay), Esto permite que el agente explore más al principio (cuando no sabe casi nada) y explote más al final (cuando ya ha aprendido una buena estrategia). En este caso, el ϵ-decay es de 0.995.

3. Actualización de la tabla Q

Se utiliza la ecuación de Bellman para actualizar los valores Q, aunque en este caso, la ecuación se escribió de esta forma:

$$Q(s,a) = (1-\alpha) * Q(s,a) + \alpha(r + \gamma max_{a'} Q(s',a'))$$

4. Cálculo de recompensas

- Si la IA captura una ficha del rival, gana +5 puntos por cada ficha capturada. En cambio, si la IA pierde una ficha, pierde -5 puntos por cada ficha perdida.

- La IA recibe una pequeña recompensa por moverse hacia adelante en el tablero. Eso evita que las partidas se hagan muy largas y que la IA entre en un bucle constante de avanzar y retroceder. Cada ficha de la IA gana $(3 - i) * 0.5$ puntos, donde i es la fila actual de la ficha.

- Si la IA gana la partida (es decir, elimina todas las fichas del

humano), obtiene una recompensa final de +20 puntos. En caso contrario, pierde -20 puntos.

5. Persistencia de la tabla Q

La tabla Q se guarda y carga desde un archivo binario para conservar el aprendizaje entre sesiones.

IV. Conclusiones

1. Mejoras en el rendimiento

En las primeras partidas, la IA realizaba movimientos aleatorios debido a su alta tasa de exploración ($\epsilon = 0.5$). Durante esta fase, el agente cometía errores frecuentes, como mover fichas sin un propósito claro o perder oportunidades de capturar fichas enemigas. Después de aproximadamente 20-30 partidas, la IA comenzó a mostrar un comportamiento más estratégico. La tasa de exploración se redujo gradualmente (ϵ decayó a 0.1), y el agente priorizó acciones que maximizaban las recompensas, como capturar fichas enemigas y proteger sus propias fichas.

2. Tasa de victorias

Luego de varias partidas, se llegó a la conclusión de que porcentaje de victorias de la IA sigue siendo menor que el del humano, pero sin dudas juega mucho mejor con este algoritmo que con el algoritmo Minimax u otros menos avanzados.

V. Bibliografía

Quiroga, F., & Lanzarini, L. C. (2019). *APRENDIZAJE AUTOMÁTICO. APLICACIONES EN RECONOCIMIENTO DE GESTOS, ACCIONES Y SEÑAS*. Investigación Joven. Universidad Nacional de la Plata. Recuperado de Revistas UNLP

Fonseca Reyna, Y. C., Martínez Jiménez, Y., & Nowé, A. (2018). *Aprendizaje reforzado aplicado a la programación de tareas bajo condiciones reales*. Ingeniería Industrial. Recuperado de SciELO