UVG
UNIVERSIDAD DEL VALLE DE GUATEMALA

Inteligencia Artificial
- Laboratorio 7 -

Instrucciones:

- Esta es una actividad en grupos de no más de 4 integrantes.
    - Este grupo aún no existe en Canvas por lo que deberán unirse a uno con el nombre de **Grupo C [Hasta 4 Integrantes] -**
- Sólo es necesario que una persona del grupo suba el trabajo a Canvas.
- No se permitirá ni se aceptará cualquier indicio de copia. De presentarse, se procederá según el reglamento correspondiente.

Tasks 1 - Teoría

Responda las siguientes preguntas de forma clara y concisa, pueden subir un PDF o bien dentro del mismo Jupyter Notebook.

1. ¿Qué es el temporal difference learning y en qué se diferencia de los métodos tradicionales de aprendizaje supervisado? Explique el concepto de "error de diferencia temporal" y su papel en los algoritmos de aprendizaje por refuerzo
2. En el contexto de los juegos simultáneos, ¿cómo toman decisiones los jugadores sin conocer las acciones de sus oponentes? De un ejemplo de un escenario del mundo real que pueda modelarse como un juego simultáneo y discuta las estrategias que los jugadores podrían emplear en tal situación
3. ¿Qué distingue los juegos de suma cero de los juegos de suma cero y cómo afecta esta diferencia al proceso de toma de decisiones de los jugadores? Proporcione al menos un ejemplo de juegos que entren en la categoría de juegos de no suma cero y discuta las consideraciones estratégicas únicas involucradas
4. ¿Cómo se aplica el concepto de equilibrio de Nash a los juegos simultáneos? Explicar cómo el equilibrio de Nash representa una solución estable en la que ningún jugador tiene un incentivo para desviarse unilateralmente de la estrategia elegida
5. Discuta la aplicación del temporal difference learning en el modelado y optimización de procesos de toma de decisiones en entornos dinámicos. ¿Cómo maneja el temporal difference learning el equilibrio entre exploración y explotación y cuáles son algunos de los desafíos asociados con su implementación en la práctica?

Task 2 - Connect Four

Para este laboratorio deberán hacer una copia de su laboratorio pasado y modificarlo para que este sea capaz de usar temporal difference learning (TD). Si no están familiarizados con el juego, pueden encontrar las reglas aquí.

Recuerden que para el tablero y validación de reglas, pueden usar código de alguna otra fuente (siempre citando), alguna librería (si encuentra, y citándola), generado por una herramienta de IA generativa (siempre citando con el prompt que usaron), o bien programada por ustedes mismos (*kudos* si lo hacen ustedes mismos).

Para programar el agente, deberán modificar su código para que este use un acercamiento de TD, para lo cual pueden considerar (pero esto no significa que sea una guía definitiva) lo siguiente:

- **Defina la representación del estado**: Modifique su programa para representar el estado del juego y el tablero en un formato adecuado para el aprendizaje de TD. Esta representación debe capturar el estado actual del juego, incluidas las posiciones de las piezas en el tablero y cualquier otra información relevante.

  Si usted desea utilizar una acercamiento de Machine Learning, utilice una representación adecuada para el estado del tablero Connect Four. En lugar de utilizar una representación tabular para la función de valor de estado-acción, puede emplear un modelo de aprendizaje automático para aproximar la función de valor según el estado del tablero. Esto podría implicar codificar el estado del tablero como un vector de características.

- **Defina el espacio de acción**: Defina el espacio de acción disponible para el agente en cada estado. En Connect Four, esto implicaría especificar las columnas donde el agente puede dejar su pieza.

Inteligencia Artificial
- Laboratorio 7 -

- **Implemente el algoritmo de aprendizaje TD**: Elija un algoritmo de aprendizaje TD, como Q-learning o SARSA, e impleméntelo dentro de su programa. Estos algoritmos aprenden actualizando estimados de la función de valor basados en las diferencias temporales entre estados y recompensas consecutivos.

  Si usted desea utilizar una acercamiento de Machine Learning, en lugar de actualizar directamente una tabla de valores de estado-acción, utilice un modelo de aprendizaje automático (por ejemplo, una red neuronal) para aproximar la función de valor. En el caso de Connect Four, esto implicaría predecir el valor de cada acción posible dado el estado actual del tablero.

- **Función de actualización de valor:** Actualice su programa para mantener una función de valor que estime el valor de cada par estado-acción. Esta función se actualizará de forma iterativa a medida que el agente interactúe con el entorno y reciba comentarios (recompensas).

  En el caso de usar un acercamiento de Machine Learning, en este punto usted debería entrenar el modelo de aprendizaje automático utilizando actualizaciones de TD learning. Después de cada acción, observe el estado resultante y la recompensa, y utilícelos para actualizar los parámetros del modelo de aprendizaje automático. Por ejemplo, en Q-learning, actualizaría los parámetros del modelo para minimizar el error de diferencia temporal entre los valores predichos y observados.

- **Definir recompensas**: Defina la estructura de recompensas para el juego Connect Four. Se pueden otorgar recompensas en función del resultado del juego (ganar, perder, empatar) o se pueden proporcionar recompensas intermedias para fomentar ciertos comportamientos (por ejemplo, colocar una pieza en una posición ganadora).
- **Implementar estrategia de exploración**: Incorpore una estrategia de exploración para alentar al agente a explorar diferentes acciones y estados durante el aprendizaje. Para este propósito se pueden utilizar técnicas como épsilon-greedy u otras.
- **Ciclo de entrenamiento**: Implemente un bucle de entrenamiento donde el agente juega contra sí mismo o contra un oponente fijo. Durante cada iteración del ciclo, el agente selecciona acciones de acuerdo con su política actual, observa el estado y la recompensa resultantes y actualiza su función de valor en consecuencia.

  En el caso de usar un acercamiento de Machine Learning, en el ciclo de entrenamiento, haga que el agente interactúe con el entorno, seleccione acciones en función de su política actual y actualice la función de valor utilizando el modelo de aprendizaje automático. Las predicciones del modelo sirven como estimaciones de la función de valor y guían el proceso de toma de decisiones del agente.

- **Evalúe y pruebe**: Una vez que se complete el entrenamiento, evalúe el desempeño de su agente de aprendizaje TD contra diferentes oponentes para evaluar su efectividad y ajustar los parámetros según sea necesario.

  Para el caso de machine learning, evalúe el rendimiento del agente de aprendizaje TD basado en aprendizaje automático frente a diferentes oponentes. Supervise métricas como la tasa de éxito, la velocidad de aprendizaje y el comportamiento de convergencia para evaluar la eficacia del enfoque.

- **Fine tuning**: Ajuste los parámetros de su algoritmo de aprendizaje TD, como la tasa de aprendizaje, el factor de descuento y la tasa de exploración, para optimizar el rendimiento y la velocidad de aprendizaje del agente.

Ahora, haga que el agente entrenado con TD learning, juegue contra el agente que usa Minimax, y luego contra el agente de minimax con poda alpha-beta. Haga que estos 3 tipos de juegos sucedan por lo menos 50 veces cada uno, es decir 150 juegos en total. Con el resultado de estos 150 juegos, grafique la cantidad de victorias de cada uno de los agentes y coloquelas en un documento PDF que deberá subir junto con su código en la entrega.

Deberá grabar un video, en el cual deberán mostrar solamente 3 juegos, es decir, uno de cada caso. Para todos los juegos en el video, asegúrense de acelerar lo suficiente para que el video no tome más de 10 minutos en total. En dicho video, también deberá mencionar (siempre dentro del marco de los 10 minutos de tiempo):

- Qué hace su agente entrenando con TD learning a nivel general

- Explique por qué ganó más veces el agente que ganó. ¿Cómo afectó el tener o no esta estrategia al agente que ganó?

Entregas en Canvas

1. Link al repositorio de los integrantes del grupo.
    a. Deberán subir el código también a Canvas por temas de Acreditación
2. Link al video solicitado en las instrucciones.

Evaluación

1. [1.25 pts.] Task 1 (0.25 cada pregunta)
2. [2.25 pts.] Task 2 - Agente
3. [0.75 pts.] Task 2 - Gráficas
4. [0.75 pts.] Task 2 - Video

Total 5 pts.