INSTITUTO POLITÉCNICO NACIONAL

ESCOM

INTELIGENCIA ARTIFICIAL

LAB 7: MINIMAX Y PODA ALFA BETA

MARTÍNEZ CHÁVEZ JORGE ALEXIS

6CV3

24 OCTUBRE 2024

Introducción

El fin de usar MINIMAX es para poder hacer la recursividad en toda se exploran todas las jugadas posibles en el árbol decisiones que puede tomar el jugador y la IA. El objetivo del jugador "maximizador" (generalmente la IA) es obtener el puntaje más alto posible, mientras que el jugador "minimizador" (el oponente) trata de minimizar este puntaje. El Minimax recorre el árbol de decisiones completo para encontrar el movimiento que asegura el mejor resultado posible para la IA, suponiendo que el oponente también está jugando para obtener su mejor resultado.

El fin de implementarlo junto con la poda alfa beta, es poder optimizar el algoritmo MINIMAX que reduce significativamente el número de nodos evaluados en el árbol de decisiones, lo que mejora la eficiencia sin afectar la precisión del resultado final. La poda se basa en dos parámetros:

Alfa: Es el valor máximo que el jugador “maximizador” está garantizado a obtener. Al principio se establece como infinito negativo.

Beta: Es el valor mínimo que el jugador minimizador está dispuesto a permitir. Se inicializa como infinito positivo.

La poda Alfa-Beta funciona descartando ramas del árbol que no pueden influir en la decisión final, es decir, ramas que ya se ha determinado que no proporcionarán un mejor resultado que el ya garantizado por Alfa o Beta.

Desarrollo

Para poder hacer el juego del gato primero debemos de crear el tablero, para ello definimos el tamaño que tendrá que será de 4x4, para poder crearlo hacemos dos “for” anidados para poder hacer la creación de la matriz 4x4.

```
# Tamaño del tablero
TAMANO_TABLERO = 4

# Inicialización del tablero 4x4
def inicializar_tablero():
    return [[VACIO for _ in range(TAMANO_TABLERO)] for _ in range(TAMANO_TABLERO)]
```

Ahora debemos de definir las “fichas” que tendra cada jugador, la para el humano le asignaremos la “x” y para la computadora le asiganremos el “o”, para los espacios vacios los representaremos con “-“, en caso de que se el juego de humano vs humano o IA vs IA el primero en tirar sera al que se le asigne la “x”.

```
# Representación de jugadores
HUMANO = 'X'
IA = 'O'
VACIO = '-'
```

Ahora creamos la función de jugar, para poder tener el menu entre que tipo de juego deseamos si “humano vs humano”, “humano vs IA” ó “IA vs IA” y para ello mandamos a inicialziar el tablero de juego sin importar la opción seleccionada.

```
# Modalidades del juego
def jugar():
    print("Elige una modalidad:")
    print("1. Humano vs Humano")
    print("2. Humano vs IA")
    print("3. IA vs IA")
    opcion = int(input("Ingresa el número de la modalidad: "))

    tablero = inicializar_tablero()
    imprimir_tablero(tablero)
```

Juego humano vs humano

Para el juego de humano vs humano, hacemos un ciclo en donde mientras no haya un ganador o no haya un empate se seguira repitiendo ese ciclo. Primero imprime el turno de quien va, para eso el segundo jugador toma el turno de la IA, es decir, el segundo jugador se va a llamar IA, depues hara la petición de que se ingrese la posición en la

que desea tirar y se imprimira el resultado del tablero despues de una tirada. Todo esto se reptira hasta que exista un ganador o hasta que haya un empate.

```
# Modalidad Humano vs Humano
turno = HUMANO
while not hay_ganador(tablero, HUMANO) and not hay_ganador(tablero, IA) and not hay_empate(tablero):
    print(f"Turno de {turno}")
    fila, columna = validar_entrada(tablero)
    tablero[fila][columna] = turno
    turno = IA if turno == HUMANO else HUMANO
    imprimir_tablero(tablero)
```

Para poder comprobar que la posición que ingreso cualquier jugador es correcta, se manda a una función llamada validar entrada, en donde se hace un ciclo en donde se repite cuando el jugador ingresa una posición no valida, ya sea que este fuera de los rangos o que sea una posición que ya ha sido ingresada.

```
# Función para validar la entrada del usuario
def validar_entrada(tablero):
    while True:
        try:
            fila = int(input("Ingresa la fila (0-3): "))
            columna = int(input("Ingresa la columna (0-3): "))
            if 0 <= fila < TAMANO_TABLERO and 0 <= columna < TAMANO_TABLERO:
                if tablero[fila][columna] == VACIO:
                    return fila, columna
                else:
                    print("La posición ya está ocupada. Intenta de nuevo.")
            else:
                print("Posición fuera de rango. Intenta de nuevo.")
        except ValueError:
            print("Entrada inválida. Ingresa un número entre 0 y 3.")
```

Un ejemplo del juego de humano vs humano seria el siguiente:

```
2. Humano vs IA
3. IA vs IA
Ingresa el número de la modalidad: 1
- - - -
- - - -
- - - -
- - - -

Turno de X
Ingresa la fila (0-3): 0
Ingresa la columna (0-3): 0
X - - -
- - - -
- - - -
- - - -

Turno de O
Ingresa la fila (0-3): 1
Ingresa la columna (0-3): 1
X - - -
- O - -
- - - -
- - - -

Turno de X
Ingresa la fila (0-3): 0
Ingresa la columna (0-3): 1
X X - -
- O - -
- - - -
- - - -

Turno de O
Ingresa la fila (0-3): 1
Ingresa la columna (0-3): 2
X X - -
- O O -
- - - -
- - - -
```

```
Turno de X
Ingresa la fila (0-3): 0
Ingresa la columna (0-3): 4
Posición fuera de rango. Intenta de nuevo.
Ingresa la fila (0-3): 2
Ingresa la columna (0-3): 2
X X - -
- O O -
- - X -
- - - -

Turno de O
Ingresa la fila (0-3): 3
Ingresa la columna (0-3): 3
X X - -
- O O -
- - X -
- - - O

Turno de X
Ingresa la fila (0-3): 0
Ingresa la columna (0-3): 0
La posición ya está ocupada. Intenta de nuevo.
Ingresa la fila (0-3): 0
Ingresa la columna (0-3): 2
X X X -
- O O -
- - X -
- - - O

Turno de O
Ingresa la fila (0-3): 3
Ingresa la columna (0-3): 1
X X X -
- O O -
- - X -
- O - O
```

```
Turno de X
Ingresa la fila (0-3): 0
Ingresa la columna (0-3): 3
X X X X
- O O -
- - X -
- O - O

¡El Humano ganó!
PS C:\Users\jorge\OneDrive\Documentos\JORGE\ESCOM\9_SEMESTRE>
```

Juego humano vs IA

Para el caso de juego de humano vs IA, primero debe de tirar el humano, al igual que antes se hace un ciclo para poder comprobar que la posición que se ingresa es correcta.

```python
opcion == 2:
# Modalidad Humano vs IA
turno = HUMANO
while not hay_ganador(tablero, HUMANO) and not hay_ganador(tablero, IA) and not hay_empate(tablero):
    if turno == HUMANO:
        print("Turno del Humano")
        fila, columna = validar_entrada(tablero)
        tablero[fila][columna] = HUMANO
        turno = IA
```

Despues el el turno de la IA, en donde se manda a llamar la función de movimiento de ia y se le envie el tablero actual, Esta función lo que hace es buscar el mejor movimiento de la IA, en lo que se hace esta busqueda imprime un mensaje de que la IA esta pensando, para buscar el mejor movieminto crearemos 2 variables, la primera para hacer el calculo de mejor puntaje, y la segunda para guarda el mejor movimiento.

```python
# Función para que la IA haga su movimiento óptimo con impresión de diagnóstico
def movimiento_ia(tablero):
    print("IA está pensando...")
    mejor_puntaje = -math.inf
    mejor_movimiento = (-1, -1)
```

Ahora crearemos dos for anidados para poder recorrer todo el tablero y la IA hara un movimiento temporal en el espacio vacio que encuentre y mandara llamar la función de minimax para que busque evalue el puntaje del movimiento que se realizo, para ello debe de enviar el tablero con el movimiento que se realizo, y como no se guardan todas

las posibles soluciones, se le asigno una profundidad maxima de 4.

```
for i in range(TAMANO_TABLERO):
    for j in range(TAMANO_TABLERO):
        if tablero[i][j] == VACIO:
            # La IA realiza un movimiento temporal
            tablero[i][j] = IA
            # Calcula el puntaje del movimiento con Minimax
            puntaje_actual = minimax(tablero, 0, -math.inf, math.inf, False, limite_profundidad=4)
            # Deshace el movimiento temporal
            tablero[i][j] = VACIO
```

Si el puntaje que se esta evaluando es mejor que el puntaje actual, entonces actualiza las variables de forma que ahora las que se evaluaron son las mejores. Y se regresa el mejor movimiento encontrado.

```
            # Actualiza el mejor puntaje y movimiento si es necesario
            if puntaje_actual > mejor_puntaje:
                mejor_puntaje = puntaje_actual
                mejor_movimiento = (i, j)

print(f"IA elige la posición: {mejor_movimiento}")
return mejor_movimiento
```

Para hacer la evaluación del puntaje, la función de minimax recibe el tablero, la profundida maxima y el alfa y el beta, primero se manda a evaluar el tablero, en donde se regresa un 10 si gana la IA, un -10 si gana el humano y un 0 si hay un empate.

```
# Función de evaluación para el estado actual del tablero
def evaluar_tablero(tablero):
    if hay_ganador(tablero, IA):
        return 10  # Victoria de la IA
    elif hay_ganador(tablero, HUMANO):
        return -10  # Victoria del Humano
    else:
        return 0  # Empate o sin ganador

# Implementación del algoritmo Minimax con poda Alfa-Beta y límite de profundidad
def minimax(tablero, profundidad, alfa, beta, es_maximizador, limite_profundidad=4):
    puntaje = evaluar_tablero(tablero)
```

Despues se comprueba si ya hay uno de estos ganadores, o si se ha llegado al limite de la profundida asignada o si hay un empate y regresa sea cual sea el resultado.

```
# Si el juego termina o se alcanza el límite de profundidad, devolver el puntaje
if puntaje == 10 or puntaje == -10 or profundidad == limite_profundidad:
    return puntaje

if hay_empate(tablero):
    return 0
```

Despues haremos la funcion de minimax, en donde se recorres todas las opciones posibles, usando la poda alfa beta para poder mejorar la eficiencia, de forma que se van descartando las ramas que no afecten el resultado final. Hacemos los dos for para ir recorriendo el tableto y hacer las posibles jugadas

```
# Maximizar la IA
if es_maximizador:
    mejor_puntaje = -math.inf

    for i in range(TAMANO_TABLERO):
        for j in range(TAMANO_TABLERO):
            if tablero[i][j] == VACIO:
                tablero[i][j] = IA
```

Despues de forma iterativa debemos de volver a evaluar el mejor puntaje mandando a llamar nuevamente la función de minimax pero se le suma un 1 al valor de la profundida para cuando lleguemos al limite de la misma. El mejor puntaje lo guardamos en una variable, y a alfa lo elegimos entre el mejor valor ente el mejor puntaje y el valora actual de alfa y se elimina la tirada para limpiar el movimiento.

```
puntaje_actual = minimax(tablero, profundidad + 1, alfa, beta, False, limite_profundidad)
mejor_puntaje = max(mejor_puntaje, puntaje_actual)
alfa = max(alfa, mejor_puntaje)
tablero[i][j] = VACIO
```

Este ciclo de seguir llamanod iteretivamente la función de minimax, la romperemos cuando la beta sea menor que la alfa, entonces regresaremos el valor de mejor puntaje.

```
            if beta <= alfa:
                break
return mejor_puntaje
```

El ejemplo de ejecución de la modalidad de humano vs ia es el siguiente:

```
PS C:\Users\jorge\OneDrive\Documentos\JORGE\ESCOM\9_SEMESTRE> & C:/Users/jorge/AppData/Local/Programs/Python/Python310/pytho
n.exe c:/Users/jorge/OneDrive/Documentos/JORGE/ESCOM/9_SEMESTRE/IA/LABORATORIO/7_MINIMAX_PODA_ALFA/gato4p4.py
Elige una modalidad:
1. Humano vs Humano
2. Humano vs IA
3. IA vs IA
Ingresa el número de la modalidad: 2
- - - -
- - - -
- - - -
- - - -

Turno del Humano
Ingresa la fila (0-3): 0
Ingresa la columna (0-3): 0
X - - -
- - - -
- - - -
- - - -

Turno de la IA
IA está pensando...
IA elige la posición: (0, 1)
X O - -
- - - -
- - - -
- - - -

Turno del Humano
Ingresa la fila (0-3): 1
Ingresa la columna (0-3): 0
X O - -
X - - -
- - - -
- - - -
```

```
Turno de la IA
IA está pensando...
IA elige la posición: (0, 2)
X O O -
X - - -
- - - -
- - - -

Turno del Humano
Ingresa la fila (0-3): 2
Ingresa la columna (0-3): 0
X O O -
X - - -
X - - -
- - - -

Turno de la IA
IA está pensando...
IA elige la posición: (3, 0)
X O O -
X - - -
X - - -
O - - -

Turno del Humano
Ingresa la fila (0-3): 3
Ingresa la columna (0-3): 3
X O O -
X - - -
X - - -
O - - X

Turno de la IA
IA está pensando...
IA elige la posición: (0, 3)
X O O O
X - - -
X - - -
O - - X
```

```
Turno del Humano
Ingresa la fila (0-3): 2
Ingresa la columna (0-3): 2
X O O O
X - - -
X - X -
O - - X

Turno de la IA
IA está pensando...
IA elige la posición: (1, 1)
X O O O
X O - -
X - X -
O - - X

Turno del Humano
Ingresa la fila (0-3): 2
Ingresa la columna (0-3): 1
X O O O
X O - -
X X X -
O - - X

Turno de la IA
IA está pensando...
IA elige la posición: (2, 3)
X O O O
X O - -
X X X O
O - - X

Turno del Humano
Ingresa la fila (0-3): 1
Ingresa la columna (0-3): 3
X O O O
X O - X
X X X O
O - - X
```

```
Turno de la IA
IA está pensando...
IA elige la posición: (1, 2)
X O O O
X O O X
X X X O
O - - X

Turno del Humano
Ingresa la fila (0-3): 3
Ingresa la columna (0-3): 1
X O O O
X O O X
X X X O
O X - X

Turno de la IA
IA está pensando...
IA elige la posición: (3, 2)
X O O O
X O O X
X X X O
O X O X

¡Empate!
PS C:\Users\jorge\OneDrive\Documentos\JORGE\ESCOM\9_SEMESTRE>
```

Juego IA vs IA

Para la modalidad de IA vs IA, a una de las IA, se le asignará el valor del humano, para ello debemos de hacer el ciclo en donde se ira alternando entre IA e IA, al igual que en la segunda modalidad, la tirada de la IA es a través de mandar llamar la función de movimiento de IA, la única diferencia es que en la función de minimax, debemos de crear las 2 opciones ya que cada una de las IA debe de buscar su mejor movimiento. Esto debido que uno sera el que se busque como jugador maximizador y otro como minimizador.

```
# Maximizar la IA
if es_maximizador:
    mejor_puntaje = -math.inf

    for i in range(TAMANO_TABLERO):
        for j in range(TAMANO_TABLERO):
            if tablero[i][j] == VACIO:
                tablero[i][j] = IA
                puntaje_actual = minimax(tablero, profundidad + 1, alfa, beta, False, limite_prof
                mejor_puntaje = max(mejor_puntaje, puntaje_actual)
                alfa = max(alfa, mejor_puntaje)
                tablero[i][j] = VACIO

                if beta <= alfa:
                    break
    return mejor_puntaje

# Minimizar el jugador humano
else:
    mejor_puntaje = math.inf

    for i in range(TAMANO_TABLERO):
        for j in range(TAMANO_TABLERO):
            if tablero[i][j] == VACIO:
                tablero[i][j] = HUMANO
                puntaje_actual = minimax(tablero, profundidad + 1, alfa, beta, True, limite_profu
                mejor_puntaje = min(mejor_puntaje, puntaje_actual)
                beta = min(beta, mejor_puntaje)
                tablero[i][j] = VACIO

                if beta <= alfa:
```

Un ejemplo de ejecución de juego es el siguente:

```
PS C:\Users\jorge\OneDrive\Documentos\JORGE\ESCOM\9_SEMESTRE> & C:/Users/jorge/AppData/Local/Programs/Python/Python310/pytho
n.exe c:/Users/jorge/OneDrive/Documentos/JORGE/ESCOM/9_SEMESTRE/IA/LABORATORIO/7_MINIMAX_PODA_ALFA/gato4p4.py
Elige una modalidad:
1. Humano vs Humano
2. Humano vs IA
3. IA vs IA
Ingresa el número de la modalidad: 3
- - - -
- - - -
- - - -
- - - -
```

```
Turno de la IA (O)
IA está pensando...
IA elige la posición: (0, 0)
O - - -
- - - -
- - - -
- - - -

Turno de la IA (X)
IA está pensando...
IA elige la posición: (0, 1)
O X - -
- - - -
- - - -
- - - -

Turno de la IA (O)
IA está pensando...
IA elige la posición: (0, 2)
O X O -
- - - -
- - - -
- - - -

Turno de la IA (X)
IA está pensando...
IA elige la posición: (0, 3)
O X O X
- - - -
- - - -
- - - -

Turno de la IA (O)
IA está pensando...
IA elige la posición: (1, 0)
O X O X
O - - -
- - - -
- - - -
```

```
Turno de la IA (X)
IA está pensando...
IA elige la posición: (1, 1)
O X O X
O X - -
- - - -
- - - -

Turno de la IA (O)
IA está pensando...
IA elige la posición: (1, 2)
O X O X
O X O -
- - - -
- - - -

Turno de la IA (X)
IA está pensando...
IA elige la posición: (1, 3)
O X O X
O X O X
- - - -
- - - -

Turno de la IA (O)
IA está pensando...
IA elige la posición: (2, 0)
O X O X
O X O X
O - - -
- - - -

Turno de la IA (X)
IA está pensando...
IA elige la posición: (2, 1)
O X O X
O X O X
O X - -
- - - -
```

```
Turno de la IA (O)
IA está pensando...
IA elige la posición: (3, 0)
O X O X
O X O X
O X - -
O - - -

¡La IA ganó!
```

Enlace

Se adjunta el link del repositorio en donde se encuentran todas las practica, la actual se encuentra en la carpeta de LAB_7_MINIMAX_PODA_ALFA_BETA: https://github.com/Jorge300403/IA_6CV3_MartinezChavez/tree/main/LAB_7_MINIMAX_PODA_ALFA_BETA