CC3045 – Inteligencia Aritificial

Proyecto 1

Instrucciones

- Esta es una actividad en grupos de no más de 3 integrantes.
    - Recuerden **unirse al grupo de canvas**
- No se permitirá ni se aceptará cualquier indicio de copia. De presentarse, se procederá según el reglamento correspondiente.
- Tendrán hasta el día indicado en Canvas.
    - No se confíen, aprovechen el tiempo en clase para entender todos los ejercicios y avanzar lo más posible.
- **NOTA**: Limiten el uso de IA generativa. Intenten primero buscar en fuentes de internet y si en verdad necesitan usarla, asegúrense de colcar el prompt que utilizar para cada task donde corresponda, así como una explicación de por qué ese prompt funcionó.

Escenario:

Ustedes están diseñando el software de navegación para un robot de entrega. El robot recibe imágenes satelitales (mapas) donde debe encontrar la ruta más eficiente desde un punto A a un punto B. Sin embargo, el mundo no es binario (pared/camino). Existen diferentes tipos de terreno (carretera, pasto, agua; tierra) representados por colores en el mapa. Usted debe diseñar el cerebro de un robot que:

1. Percibir: Identificar qué tipo de terreno es un píxel basándose en su color RGB usando una Red Neuronal
2. Decidir: Calcular la ruta de menor costo usando A*, donde cruzar "agua" es mucho más costoso (lento/peligroso) que cruzar "carretera".

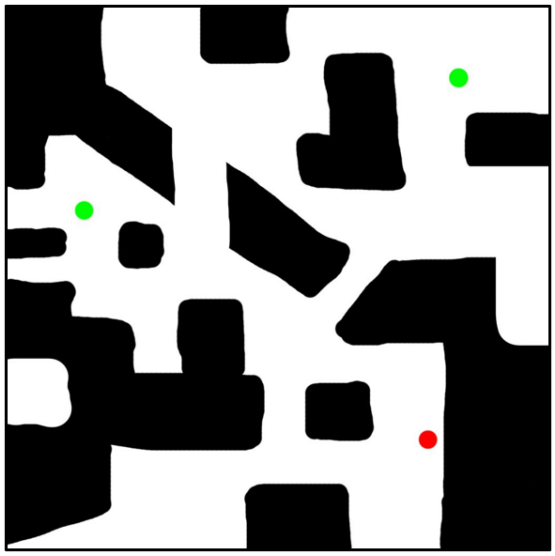
Task 1 – Search Engine

Como primer paso deberá construir el motor de búsqueda que interpreta una imagen con ciertas características y genera rutas.

Para esto, deberá construir un programa que reciba como entrada una imagen cuadrada que representa un laberinto (en formato png o bmp). Con dicha imagen, deberá dibujar en pantalla la solución al mismo. Además, deberá considerar las siguientes restricciones:

- Las dimensiones de entrada podrán variar entre ejecuciones (pero siempre será una imagen cuadrada)
- Las áreas blancas representan caminos libres
- Las áreas negras representan paredes sobre las cuales no se puede pasar
- Las áreas verdes representan la meta (goalTest positivos) (pueden ser varios)
- La área roja representa el punto de inicio (solo podrá haber uno)

Considérese la siguiente imagen como ejemplo:

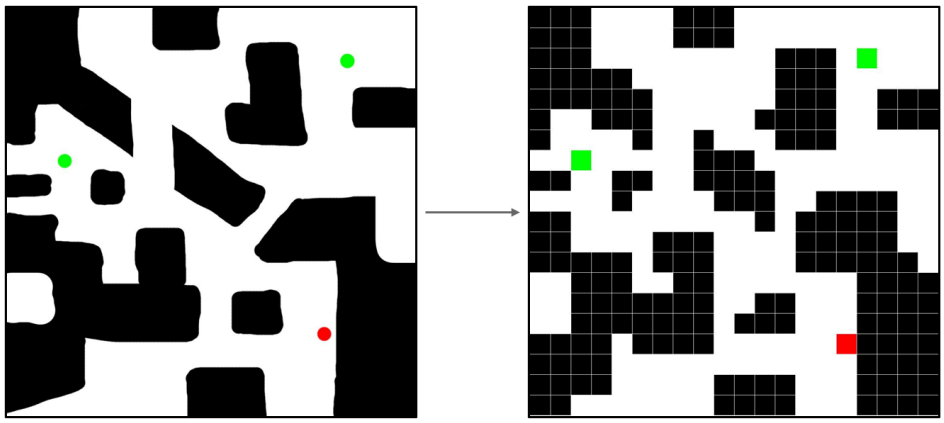


Task 1.1 – Discretización del Mundo

Deberá representar la imagen dada de forma discreta. La cantidad de píxeles a considerar por cuadro queda a discreción del estudiante. Para esto considere:

- Input: El programa recibirá una imagen (.png o .bmp).
- Grid: Discretice la imagen en una matriz de nodos.
    - Nota: No procese píxel por píxel si la imagen es 4K. Agrúpelos en "batches" (tiles) de, por ejemplo, 10x10 o 20x20 píxeles para formar los nodos del grafo.
- Identificación:
    - Encuentre las coordenadas de inicio (Rojo) y meta (Verde).
    - Identifique los obstáculos (Negro absoluto [0,0,0]) que son intransitables.

Considérese el siguiente ejemplo como guía:



Task 1.2 – Búsqueda

Use una interfaz genérica (o clase abstracta) que sirva para representar el framework para definir el problema formal. La implementación de la interfaz para este proyecto deberá recibir como parámetro de construcción la matriz obtenida en el Task 1.1, y con esta deberá deducir las demás funciones del framework (actions(s), stepCost(s,a,s'), etcétera). Debe utilizar efectivamente los conceptos del paradigma de POO.

Con esto, deberá construir el algoritmo genérico (familia de graphSearch. Este recibirá de parámetro una instancia del framework de problemas. Sobre esta familia genérica, implemente los siguientes algoritmos:

- BFS (Breadth-First Search): Encuentre el camino con la menor cantidad de pasos (ignorando costos de terreno).
- DFS (Depth-First Search): Explore caminos rápidamente.
- Visualización: Dibuje una línea sobre la imagen original mostrando la ruta encontrada.

Task 1.3 – Búsqueda A* (A-star)

Implemente A* con una heurística adecuadae. Inicialmente, asuma que moverse a cualquier cuadro que no sea pared tiene costo 1.

Nota: Una vez resuelva el laberinto de entrada, muestre en pantalla gráficamente el camino encontrado. Su representación visual se basará en la matriz discreta, no en la imagen original. Considere la siguiente imagen como ejemplo:

Task 2 – Neural Networks Integration

Ahora, deberán hacer que el mapa tenga "semántica". Los caminos ya no son blancos, sino que tienen colores que representan materiales.

Para esto deberá utilizar el dataset: Associated RGBs and Basic Color Names.

Este dataset contiene columnas R, G, B y una etiqueta (Label) con el nombre del color.

Task 2.1 – Entrenamiento de la Red Neuronal

Debe entrenar una Red Neuronal que clasifique colores RGB.

1. **Preparación:** Descargue el dataset. Mapee las etiquetas de texto a costos numéricos para su robot.
    - Ejemplo de mapeo (Defina esto en su reporte):
        - Green (Grama) -> Costo 3
        - Blue (Agua) -> Costo 10
        - Grey (Pavimento) -> Costo 1
        - Yellow (Arena) -> Costo 5
2. **Modelo "Desde Cero":** Implemente una **Red Neuronal (MLP)** utilizando únicamente numpy (como vimos en la S5).
    - **Arquitectura:** Input Layer (3 neuronas: R, G, B) -> Hidden Layers (a su criterio) -> Output Layer (Clasificación del terreno/color).
    - **Entrenamiento:** Debe usar **Descenso de Gradiente Estocástico (SGD)** y **Backpropagation**.
    - **Métrica:** Reporte el Accuracy de su modelo en un set de prueba (20%).

Task 2.2

Modifique la función de costo stepCost(nodo_actual, vecino) de su algoritmo **A***.

1. **Inferencia en vivo:** Cuando el algoritmo A* esté explorando vecinos, **no** use un costo fijo.
    - Tome el color RGB promedio del nodo/baldosa vecino en la imagen del laberinto.
    - Pase ese RGB por su **Red Neuronal** (Task 2.1).
    - Obtenga la predicción (ej: "Blue") y conviértala a costo (ej: 10).
    - Use ese costo para calcular $g(n)$ y $f(n)$

.

2. **El Experimento (Demo):**
    - Cree (usando Paint o similar) un laberinto donde haya dos caminos posibles hacia la meta:
        - **Camino A:** Corto pero pintado de Azul (Agua, costo alto).
        - **Camino B:** Largo pero pintado de Gris (Pavimento, costo bajo).
    - Ejecute su A*. **El robot debería ser lo suficientemente inteligente para elegir el Camino B (largo pero rápido) en lugar del A.}**
    - Muestre su resultado de la misma manera que en el task 1.

Entregas en Canvas

1. Documento PDF con las respuestas a cada task
2. Archivo .ipynb, o link a repositorio de GitHub (No se acepta entregas en otros medios)
3. Deberá estar listo para presentar el próximo día de clases.

Evaluación

1. [4.0 pt] Task 1
2. [4.0 pt] Task 2

Total 8 pts