

Practica 1. Agentes Inteligentes

Un **agente inteligente** es un sistema que percibe su entorno mediante sensores, procesa la información recibida y toma decisiones o realiza acciones de manera autónoma para alcanzar sus objetivos. Estos agentes pueden ser físicos, como robots, o virtuales, como los sistemas de inteligencia artificial que se encuentran en programas de software.

Existen diferentes tipos de agentes inteligentes, algunos más simples y otros más complejos, dependiendo de su capacidad para aprender, adaptarse, interactuar con su entorno y predecir resultados futuros.

Ejemplos clásicos de **agentes inteligentes**:

1. **Termostato inteligente:** Un termostato puede ajustarse automáticamente en función de la temperatura percibida en su entorno, activando o desactivando el sistema de calefacción o aire acondicionado para mantener una temperatura deseada.
2. **Robot aspiradora:** Robots como Roomba utilizan sensores para percibir su entorno, navegar por el espacio, esquivar obstáculos y limpiar áreas de manera autónoma.
3. **Sistemas de recomendación:** Plataformas como Netflix o Amazon utilizan agentes inteligentes para analizar el comportamiento del usuario (historial de visualización o compras) y recomendar contenido relevante.
4. **Asistentes virtuales:** Sistemas como Siri, Alexa o Google Assistant, que utilizan procesamiento del lenguaje natural para interactuar con los usuarios, responder preguntas, realizar tareas y aprender de sus interacciones.
5. **Agentes de juegos:** Los programas que juegan ajedrez o Go, como el caso de *AlphaGo*, que utilizan inteligencia artificial para analizar movimientos posibles y realizar jugadas estratégicas en tiempo real.
6. **Vehículos autónomos:** Coches como los desarrollados por Tesla o Waymo utilizan sensores y algoritmos avanzados para percibir el entorno, tomar decisiones de navegación y moverse de manera autónoma en carreteras.

Arquitectura

Una **arquitectura general** para un agente inteligente suele constar de varios componentes esenciales que trabajan juntos para percibir el entorno, tomar decisiones y actuar. A continuación se presenta un esquema básico de dicha arquitectura:

1. Sensores (Percepción):

- Capturan información del entorno. Dependiendo del tipo de agente, estos pueden ser cámaras, micrófonos, sensores de temperatura, sensores de proximidad, etc.
- Ejemplos: Cámaras en un robot, micrófonos en un asistente virtual.

2. Procesamiento de Percepción:

- Convierte los datos crudos de los sensores en una forma útil para el procesamiento. Aquí se filtra y procesa la información para obtener una representación significativa del entorno.
- Ejemplos: Reconocimiento de objetos, detección de patrones en imágenes o sonidos.

3. Base de Conocimiento:

- Es un almacén de información que el agente utiliza para tomar decisiones. Puede incluir reglas predefinidas, hechos, modelos aprendidos o experiencias previas.
- Ejemplos: Conocimiento del mapa en un robot aspirador o una base de datos de enfermedades en un sistema de diagnóstico médico.

4. Módulo de Toma de Decisiones (Razonamiento):

- Este componente recibe la información procesada por los sensores y la compara con su base de conocimiento. Utiliza algoritmos para tomar decisiones sobre las acciones que el agente debe realizar.
- Ejemplos: Algoritmos de optimización, sistemas de reglas, planificación de acciones.

5. Aprendizaje (Opcional):

- En agentes más avanzados, este módulo permite al sistema mejorar sus decisiones a través de la experiencia. El aprendizaje puede ser supervisado (con etiquetas) o no supervisado (explorando patrones en los datos).
- Ejemplos: Algoritmos de aprendizaje automático que ajustan los parámetros del modelo basándose en datos históricos.

6. Módulo de Acción (Actuadores):

- Ejecuta las decisiones del agente en el mundo físico o virtual. Los actuadores pueden ser motores en un robot, comandos de texto en un asistente virtual, o una señal para activar algún dispositivo.
- Ejemplos: Brazos robóticos que mueven objetos, altavoces que emiten respuestas en voz alta.

7. Retroalimentación:

- Los agentes más sofisticados pueden recibir retroalimentación de sus acciones y ajustar su comportamiento. La retroalimentación permite mejorar la eficacia del agente con el tiempo.
- Ejemplos: Un robot que ajusta su trayectoria al recibir información sobre obstáculos, o un sistema de recomendación que ajusta sus sugerencias según las reacciones del usuario.

Practica para el alumno:

- **Robot aspiradora:** Los sensores detectan obstáculos (percepción), los datos se procesan para crear un mapa del entorno, el sistema toma decisiones sobre la dirección a seguir (razonamiento), y los motores actúan (actuadores). Si el robot queda atascado, puede aprender a evitar esa área en el futuro (aprendizaje).

Esta arquitectura es adaptable y se puede personalizar según el tipo de agente inteligente y su nivel de complejidad.

En este caso, construiremos un agente simple que navega en una cuadrícula evitando obstáculos para llegar a un objetivo. Este agente utilizará una arquitectura con sensores, procesamiento de percepción, toma de decisiones y actuadores.

Escenario:

- El agente está en una cuadrícula de 5x5.
- Su objetivo es moverse desde una posición inicial hasta un objetivo específico.
- La cuadrícula tiene algunos obstáculos que el agente debe evitar.
- El agente utiliza sensores para detectar si hay un obstáculo en una posición adyacente.

Capture y corra en siguiente código. Analice el software y de a su profesor respuesta a sus cuestionamientos.

```
clc;
clear;

% Definición del entorno (5x5 grid)
gridSize = 5;
grid = zeros(gridSize); % 0: libre, 1: obstáculo

% Posicionar obstáculos
grid(2, 2) = 1;
grid(4, 3) = 1;

% Definir posición inicial del agente y objetivo
agentPos = [1, 1]; % Inicio en la esquina superior izquierda
goalPos = [5, 5]; % Meta en la esquina inferior derecha

% Función para mostrar la cuadrícula
function showGrid(grid, agentPos, goalPos)
    clf;
    imagesc(grid);
    colormap(gray);
    hold on;
    plot(agentPos(2), agentPos(1), 'bo', 'MarkerSize', 15, 'LineWidth', 3); %
    Agente
```

```

    plot(goalPos(2), goalPos(1), 'ro', 'MarkerSize', 15, 'LineWidth', 3); %
Objetivo
    title('Agente navegando en la cuadrícula');
    drawnow;
end

% Mostrar la cuadrícula inicial
showGrid(grid, agentPos, goalPos);

% Definir las posibles acciones (movimientos)
actions = [0, 1; 0, -1; 1, 0; -1, 0]; % Derecha, Izquierda, Abajo, Arriba

% Mover al agente hasta alcanzar la meta o quedar atrapado
while ~isequal(agentPos, goalPos)
    % Mostrar la cuadrícula
    showGrid(grid, agentPos, goalPos);

    % Leer los sensores (percibir el entorno)
    % Checar si los movimientos posibles están bloqueados por obstáculos o
límites del grid
    validMoves = [];
    for i = 1:size(actions, 1)
        newPos = agentPos + actions(i, :);
        if newPos(1) >= 1 && newPos(1) <= gridSize && newPos(2) >= 1 &&
newPos(2) <= gridSize
            if grid(newPos(1), newPos(2)) == 0 % No hay obstáculo
                validMoves = [validMoves; actions(i, :)];
            end
        end
    end
end

% Decisión (elegir un movimiento aleatorio válido)
if isempty(validMoves)
    disp('El agente está atrapado.');
```

break;

```

else
    % Tomar una decisión aleatoria basada en los movimientos válidos
    move = validMoves(randi(size(validMoves, 1)), :);
    agentPos = agentPos + move; % Actualizar la posición del agente
end

    pause(0.5); % Pausa para visualizar el movimiento
end

% Verificar si el agente llegó a la meta
if isequal(agentPos, goalPos)
    disp('El agente alcanzó su objetivo.');
```

else

```

    disp('El agente no pudo llegar a su objetivo.');
```

end

Explicación:

1. **Cuadrícula:** Representamos el entorno como una matriz de 5×5 . Los valores en la matriz indican si una celda está libre (0) o bloqueada por un obstáculo (1).
2. **Posiciones iniciales:** El agente comienza en la posición (1, 1) y su objetivo es alcanzar la posición (5, 5).
3. **Sensores:** El agente "lee" su entorno verificando las posiciones adyacentes para detectar si hay obstáculos o si está fuera de los límites de la cuadrícula.
4. **Decisiones:** Basado en la información de los sensores, el agente selecciona un movimiento válido aleatorio de entre las direcciones disponibles (arriba, abajo, izquierda, derecha).
5. **Actuación:** El agente ejecuta la acción seleccionada, moviéndose a una nueva posición en la cuadrícula.
6. **Ciclo de vida:** El bucle principal continúa hasta que el agente alcanza su objetivo o queda atrapado sin movimientos válidos.