

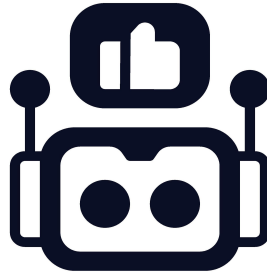


Agentes Inteligentes



Definición

- Es un sistema que toma decisiones para realizar tareas específicas.
- El agente puede percibir su entorno, tomar decisiones basadas en esa información y realizan acciones para lograr sus objetivos.



Agente inteligente: es algo que realiza cosas por sí mismo, tomando decisiones inteligentes 💡

CartPole

- **Objetivo:** Mantener el poste en posición vertical tanto tiempo como sea posible.
- **Entorno:** Un carrito que puede moverse de manera horizontal a lo largo de una pista y un poste montado en el carrito.
- **Acciones:** El agente (el programa o el modelo de ML) puede tomar acciones para mover el carrito hacia la izquierda o hacia la derecha.

CartPole

- **Nota:** El agente recibe información del entorno sobre la posición y la velocidad del carrito, así como de la inclinación y velocidad angular del poste.
- **Recompensas:** El agente recibe una recompensa positiva por cada movimiento que mantenga el poste en posición vertical.

Link: [Juego de cartpole](#)

CartPole

- ¿Qué tipo de agente es?

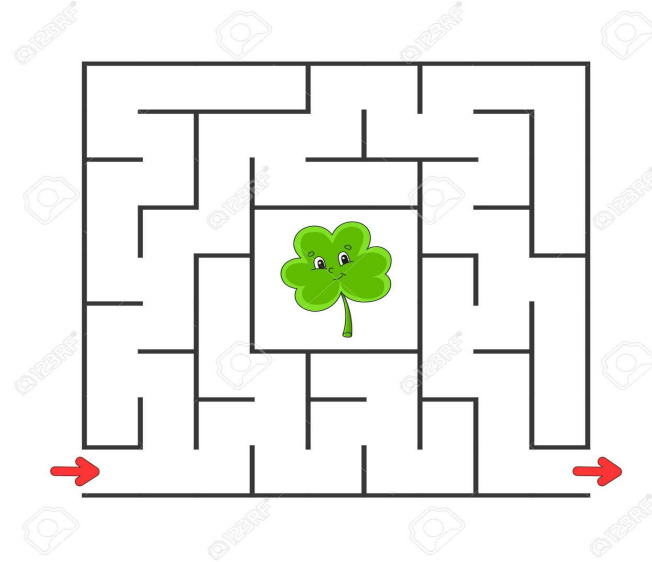
Es un ***agente reactivo simple*** porque sólo reacciona a la percepción actual, es decir, no necesita del historial de percepciones para solucionar el problema.



Práctica: Laberinto

En un laberinto

- ¿Quién es el agente?
- ¿Qué tipo de agente es?



Práctica: Laberinto

Notebook:

<https://colab.research.google.com/drive/1WOz3k8BWYNTLT-ngN9BEIqGnOTSJkBuU?usp=sharing>

Técnica de Backtracking

- Es una estrategia **recursiva** para resolver problemas donde se itera sobre todas las posibles configuraciones del espacio de búsqueda.
- La solución es incremental, así que paso a paso vamos eliminando las soluciones que no nos sirven.

Técnica de Backtracking en laberintos

- Iniciamos en la celda marcada como E (que es el inicio)

E 0 1 0

1 0 1 0

0 0 0 0

1 1 0 S

Técnica de Backtracking en laberintos

- Nos movemos a las celdas adyacentes (arriba, abajo, izquierda, derecha) que no hayan sido visitadas y que no sean paredes

E 0 1 0

1 0 1 0

0 0 0 0

1 1 0 S

Nota: En cada paso verificamos si la celda actual es la salida (S)

Técnica de Backtracking en laberintos

- Si encontramos una celda adyacente válida (es decir, que no sea 1) entonces nos movemos a esa celda, la marcamos como parte del caminito y repetimos el proceso.

E 0 1 0

1 0 1 0

0 0 0 0

1 1 0 S

Nota: En cada paso verificamos si la celda actual es la salida (S)

Técnica de Backtracking en laberintos

E 0 1 0		E 0 1 0		E 0 1 0		E 0 1 0		E 0 1 0		E 0 1 0		E 0 1 0
1 0 1 0		1 0 1 0		1 0 1 0		1 0 1 0		1 0 1 0		1 0 1 0		1 0 1 0
0 0 0 0	->	0 0 0 0	->	0 0 0 0	->	0 0 0 0	->	0 0 0 0	->	0 0 0 0	->	0 0 0 0
1 1 0 S		1 1 0 S		1 1 0 S		1 1 0 S		1 1 0 S		1 1 0 S		1 1 0 S

Nota: En cada paso verificamos si la celda actual es la salida (S)

Técnica de Backtracking en laberintos

- Si desde la celda actual no se puede avanzar más (es decir, no hay celdas adyacentes válidas no visitadas) entonces se retrocede a la celda anterior para explorar otras posibles direcciones que no he encontrado.

E 0 1 0	E 0 1 0	E 0 1 0	E 0 1 0	E 0 1 0	E 0 1 0	E 0 1 0
1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0	1 0 1 0
0 0 0 0	-> 0 0 0 0	-> 0 0 0 0	-> 0 0 0 0	-> 0 0 0 0	-> 0 0 0 0	-> 0 0 0 0
1 1 0 S	1 1 0 S	1 1 0 S	1 1 0 S	1 1 0 S	1 1 0 S	1 1 0 S

Nota: En cada paso verificamos si la celda actual es la salida (S)

Técnica de Backtracking en laberintos

E	0	1	0
1	0	1	0
0	0	0	0
1	1	0	S

Hemos encontrado la salida :D

Nota: En cada paso verificamos si la celda actual es la salida (S)