Guía de estudio de Inteligencia Artificial

1. **Inteligencia artificial generativa**
   Se enfoca en crear contenido nuevo al aprender patrones de datos existentes
   - Texto
   - Imágenes
   - Videos
   - Música
2. **Inteligencia artificial**
   Centrado en crear sistemas que pueden realizar tareas que normalmente requieren inteligencia humana
3. **Árbol de decisiones**
   Algoritmo de aprendizaje supervisado para tomar decisiones basadas en datos.
4. **¿Por qué la IA es buena en los juegos?**
   Entradas y salidas que se pueden representar matemáticamente
5. **Machine learning**
   Se le proporciona a la computadora datos y un algoritmo que la ayuda a aprender patrones y relaciones dentro de los datos para hacer predicciones, clasificaciones o tomar decisiones.
6. **Reinforcement learning**
   - Se centra en entrenar un agente para que tome decisiones en un entorno, con el objetivo de maximizar una recompensa acumulativa.
   - El agente aprende por prueba y error, explorando acciones y observando las consecuencias en recompensas o penalizaciones.
7. **Supervised learning**
   Técnica de aprendizaje automático donde un modelo aprende a partir de datos etiquetados para predecir o clasificar correctamente nuevos datos.
8. **Unsupervised learning**
   - Tipo de aprendizaje automático en el que se entrena un modelo con datos no etiquetados.
   - Descubrir patrones, relaciones o estructuras en los datos sin guía de etiquetas o resultados explícitos.
9. **Deep learning**
   Consigue que un ordenador termine aprendiendo por cuenta propia y realizando tareas similares a los humanos
10. **Redes neuronales**

Procesan datos y parámetros para aprender patrones que, dependiendo del input, generen una respuesta precisa.

11. Large language models

- ChatGPT
- Gemini
- DeepSeek

12. Generative Pre-trained Transformers (GPT)

La IA no "sabe" la respuesta correcta en sentido humano, calcula la probabilidad de que ciertas palabras o frases sean apropiadas en función del contexto.

13. Agente

Entidad que percibe su entorno y actúa sobre este.

14. Estado

Configuración del agente y su entorno.

15. Estado inicial

Estado en el que comienza el agente.

16. Acciones

Opciones que se pueden tomar en un estado.

17. Modelo de transición

Descripción del estado que resulta de realizar cualquier acción aplicable en cualquier estado.

18.Espacio de estados

Conjunto de todos los estados alcanzables desde el estado inicial mediante cualquier secuencia de acciones.

19. Nodo

Estructura de datos que realiza seguimiento de:

- Estado
- Padre
- Acción
- Costo de ruta

20. Stack

Ultimo en entrar, primero en salir.

21. Queue

Primero en entrar, primero en salir.

22. Depth-First Search

- Búsqueda en amplitud.
- Algoritmo de búsqueda que siempre expande el nodo mas profundo en la frontera.

- Usa stack.

23. Función DFS

- Cuando se encuentra un vertice sin vecinos no visitados.
- El algoritmo retrocede a lo largo del camino hasta encontrar un vertice con vecinos no visitados y continua la exploración desde allí.
- El proceso se repite hasta que se visitan todos los vértices alcanzables desde el vertice origen.

24. Breadth-First Search

- Búsqueda en amplitud.
- Algoritmo de búsqueda que siempre expande el nodo mas superficial en la frontera.
- Usa queue.

25. Función BFS

- Empieza en un nodo raíz
- Visita todos los nodos adyacentes al nodo raíz
- Luego visita los nodos adyacentes a los nodos visitados en el paso anterior.
- Continua hasta que todos los nodos estén visitados o hasta que se haya encontrado el nodo objetivo.

26. Greedy best-first search

- Algoritmo de búsqueda que expande el nodo mas cercano al objetivo, según lo estimado por una función heurística.
- Este algoritmo siempre expandirá el nodo hacia donde crea que está más cerca la meta.

27. A* Search

Algoritmo de búsqueda que expande el nodo con el valor mas bajo de g(n) + h(n)

- g(n) = costo para alcanzar el nodo
- h(n) = costo estimado para alcanzar la meta

28. Minimax

Algoritmo utilizado para resolver problemas de juegos competitivos. Ayuda al jugador a anticipar las jugadas de su oponente y tomar decisiones optimas.

- Max: Tiene como objetivo maximizar la puntuación.
- Min: Tiene como objetivo minimizar la puntuación.

29. Poda Alfa-Beta

- Optimización del algoritmo minimax que acelera la toma de decisiones evitando evaluar ramas innecesarias del árbol de búsqueda cuando se sabe que no influirán en la decisión final.

- Se necesita cumplir la condición α >= β para poder determinar si un nodo no es necesario que sea recorrido, mientras esta condición no se cumpla, se seguirá explorando los nodos hasta que se cumpla la condición.
- α = MAX
- β = MIN

30. Depth-Limited Minimax

Limita la profundidad de búsqueda en el árbol de decisiones. En lugar de explorar todos los estados finales del juego, detiene la búsqueda en un nivel especifico y evalúa los nodos con una función heurística.