¿Qué significa ser inteligente?
Según la teoría de las inteligencias múltiples (Garner, 1983), ser inteligente es.
"Poseer y utilizar de manera efectiva una combinación de diversas capacidades específicas que permiten a una persona adaptarse, aprender y resolver problemas en diferentes contextos."

Tipos de Inteligencia:
Lingüística. Habilidad para expresar pensamientos y entender el lenguaje de otros, dominar fonología, semántica y gramática.
Lógico-Matemática. Capacidad para calcular, cuantificar, razonar y realizar operaciones matemáticas complejas.
Espacial. Habilidad para reconocer y representar visualmente el espacio y las formas alrededor.
Corporal-Kinestésica. Capacidad para usar el cuerpo para expresar pensamientos y emociones, y manipular objetos.
Musical. Habilidad para discernir tono, melodía, ritmo y timbre.
Interpersonal. Capacidad para entender e interactuar efectivamente con otros.
Intrapersonal. Capacidad para comprenderse a uno mismo y actuar en consecuencia.
Naturalista. Habilidad para observar y clasificar formas de la naturaleza.

¿Qué es la Inteligencia Artificial?
La primera y más aceptada definición es la propuesta por John McCarthy (en la Conferencia de Dartmouth de 1956)
"La IA se refiere a la capacidad de una máquina (computadoras, teléfonos, u otros dispositivos) de simular el comportamiento inteligente de los humanos con la mayor precisión posible".
Otras definiciones son
Definición de Cambridge
El uso o estudio de sistemas de computadoras o máquinas que tienen algunas de las cualidades del cerebro humano, como la habilidad de interpretar y producir un lenguaje que

parezca humano, reconocer o crear imágenes, resolver problemas y aprender de datos.

Definición de Google Cloud
Campo de la ciencia relacionado con la creación de computadoras y máquinas que pueden razonar, aprender y actuar de una forma que normalmente necesitaría inteligencia humana. A nivel operativo para el uso empresarial, es un conjunto de tecnologías basadas principalmente en aprendizaje automático y aprendizaje profundo.

Definición de IBM
Campo que combina la computación y conjuntos de datos robustos para permitir la resolución de problemas. También abarca subcampos del aprendizaje automático y del aprendizaje profundo, disciplinas compuestas por algoritmos que buscan crear sistemas expertos que realicen predicciones o clasificaciones basadas en datos de entrada.

Enfoques de la IA.
Actuar como humano. Replicar el comportamiento humano. La idea es que si una máquina puede actuar como un humano, entonces puede ser considerada inteligente.
Pensar como humano. Emular los procesos de pensamiento humano. Se inspira en la ciencia cognitiva, investiga cómo funciona la mente humana y se intenta replicar estos procesos en una máquina.
Pensar racionalmente. Aplicar principios de lógica formal y filosofía. Se enfoca en la toma de decisiones correctas y racionales basadas en inferencias y razonamientos.
Actuar racionalmente. Tomar decisiones óptimas para alcanzar objetivos. Considera agentes de IA como sistemas que toman decisiones óptimas para alcanzar sus objetivos, basados en la información disponible y las posibles consecuencias de sus acciones.
¿Qué es un agente?
Es una entidad que percibe su entorno a través de sensores y actúa sobre ese entorno mediante actuadores para lograr ciertos objetivos o tareas. Los agentes pueden ser software, hardware, o una combinación de ambos, y pueden variar en

complejidad desde programas simples hasta sistemas altamente sofisticados.

Tipos de IA
Aprendizaje Supervisado vs. No Supervisado vs. Reforzado
Inteligencia General vs. Inteligencia Estrecha
Por Función Humana

Aprendizaje Supervisado. Se entrena con datos etiquetados.
Aprendizaje No Supervisado. Se enfrenta a datos no etiquetados, busca patrones.
Aprendizaje Reforzado. El agente toma decisiones en un entorno para maximizar recompensas.

Inteligencia Estrecha. Especialización en una tarea específica.
Inteligencia General. Capacidad de comprender, aprender y aplicar conocimientos en diversas áreas.
Superinteligencia. Representaría un nivel de inteligencia que supera significativamente las habilidades cognitivas humanas en todas las áreas.

Clasificación Por Función Humana
Visión por Computadora. Capacidad de las máquinas para interpretar y comprender el mundo visual.
Aprendizaje de Máquina. Desarrollo de algoritmos que permiten a las máquinas aprender patrones a partir de datos.
Procesamiento de Lenguaje Natural. Habilidad de las máquinas para comprender, interpretar y generar lenguaje humano.
Robótica. Combina ingeniería, ciencia de la computación y otras disciplinas para diseñar, construir, operar y utilizar robots.
Optimización. Se centra en encontrar la mejor solución posible para un problema dado, bajo un conjunto de restricciones.

El Autómata Turco
El Autómata Turco fue una máquina creada en 1770 por Wolfgang von Kempelen que simulaba jugar ajedrez de forma autónoma, impresionando al público y derrotando a figuras como Napoleón Bonaparte y Benjamín Franklin. Sin

embargo, era un truco: un maestro de ajedrez oculto dentro del gabinete controlaba los movimientos del "Turco". Aunque fue destruido en 1854, su fama perduró como una fascinante mezcla de tecnología e ilusión, inspirando debates sobre inteligencia artificial y robótica.

Shakey el Robot

Shakey el Robot (1966-1972) fue el primer robot móvil autónomo, desarrollado por el Instituto de Investigación de Stanford. Usaba cámaras, sensores y un software llamado STRIPS para analizar su entorno, planificar rutas y resolver problemas de forma autónoma. Fue pionero en combinar percepción, planeación y acción, sentando las bases de la robótica moderna y la inteligencia artificial. Su legado sigue vivo en tecnologías como los robots y vehículos autónomos.

Test de Turg

El **Test de Turing**. propuesto por Alan Turing en 1950, es una prueba para determinar si una máquina puede exhibir inteligencia similar a la humana. Consiste en que un evaluador humano interactúe mediante texto con una máquina y un humano sin saber cuál es cuál. Si el evaluador no puede distinguir a la máquina del humano, se considera que la máquina ha pasado el test. Es un hito en la discusión sobre inteligencia artificial.

Cuarto chino

El cuarto chino. Propuesto por el filósofo John Searle en su artículo de 1980 "Minds, Brains, and Programs", es un experimento mental que cuestiona la posibilidad de que las máquinas puedan poseer inteligencia genuina.

La IA comenzó en la conferencia de Dartmouth en 1956

En 1956, John McCarthy organizó la Conferencia de Dartmouth en Hanover, New Hampshire. Reunió a investigadores líderes en teoría de la complejidad, simulación del lenguaje, redes neuronales y la relación entre el azar y el pensamiento creativo. Entre los asistentes se encontraban Marvin Minsky, Nathaniel Rochester y Claude Shannon.

Avances del primer verano de la IA (1956-1973)

Probabilidad algoritmica/Inferencia inductiva. Ray Solomonoff introdujo métodos bayesianos universales para la inferencia inductiva y la predicción. Sus ideas sentaron las bases para futuras investigaciones en aprendizaje automático y predicción.

ANALOGY. Thomas Evans creó un programa que permitía a las computadoras resolver problemas de analogía geométrica. Este avance mostró que las máquinas podían entender y manipular conceptos abstractos, acercándose más a la capacidad de razonamiento humano.

UNIMATE. El primer robot industrial creado por la empresa Unimation, trabajó en la línea de ensamblaje de automóviles de General Motors. Este robot fue un hito en la automatización industrial y demostró el potencial de la robótica en la fabricación y otras industrias.

ELIZA. Un programa interactivo desarrollado por Joseph Weizenbaum en el MIT, podía mantener conversaciones en inglés sobre cualquier tema. ELIZA simulaba un psicoterapeuta rogeriano, y aunque sus respuestas eran simples y basadas en patrones predefinidos, su capacidad para interactuar con humanos fue revolucionaria en ese momento.

Causas del primer invierno de la IA (1974-1980)

Expectativas vs. Realidad. La percepción era que los investigadores habían prometido más de lo que podían cumplir

Limitaciones Técnicas. La percepción era que los investigadores habían prometido más de lo que podían cumplir. Las computadoras no podían manejar la complejidad de los problemas planteados. Un sistema de IA para analizar el lenguaje inglés solo podía manejar un vocabulario de 20 palabras debido a limitaciones de memoria.

Cambio en la Financiación (DARPA). DARPA había invertido millones de dólares en IA sin presionar a los investigadores. La Enmienda Mansfield de 1969 cambió esta situación, financiando solo investigaciones orientadas a misiones directas y útiles para la tecnología militar.

Avances del segundo verano de la IA (1981-1987)
La IA se centró en resolver problemas específicos y realizar tareas concretas.
Iniciativa de Computación Estratégica (DARPA, EE. UU.). Enfoque en la supercomputación y la microelectrónica para mantenerse competitivos frente a Japón en la carrera de la IA.
Competencia entre EE. UU. y Japón. Impulsó una mayor inversión en investigación y desarrollo, llevando a avances tecnológicos y metodológicos en la IA.
Iniciativa de Computadoras de Quinta Generación (Japón). Proyecto de 10 años con una inversión de 320 millones de dólares para desarrollar programas/máquinas con capacidades avanzadas como mantener conversaciones y traducir idiomas.

Causas del Segundo Invierno de la IA(1988-2011)
Costo de los Sistemas Expertos. Los sistemas expertos demostraron ser demasiado caros de mantener debido a la necesidad de actualizaciones manuales constantes y la incapacidad de manejar entradas inusuales.
Progreso en Computadoras Personales. Las computadoras personales se volvieron más poderosas y económicas, superando rápidamente a las máquinas Lisp y otros hardware específicos de IA.
Cancelación del Proyecto de Computación Estratégica (DARPA, EE. UU.). En EE. UU., DARPA canceló el proyecto de Computación Estratégica debido a que no cumplió con las expectativas.

Deep Blue
Deep Blue fue una supercomputadora desarrollada por IBM que hizo historia en 1997 al vencer al campeón mundial de ajedrez Garry Kasparov en un torneo oficial. Utilizaba fuerza bruta para calcular millones de jugadas por segundo, pero no tenía "inteligencia" en el sentido humano, solo capacidad para analizar posibles movimientos y elegir el mejor. Fue un hito en la inteligencia artificial aplicada a juegos.

Stanley
Stanley, el vehículo autónomo construido en Stanford, ganó

el DARPA Grand Challenge, una competencia que puso a prueba la capacidad de los vehículos autónomos para navegar de manera segura y eficiente a través de un terreno desértico sin intervención humana.

El resurgimiento de la IA (2011 hasta la actualidad)
Watson (2011). Demostró su capacidad para responder preguntas en lenguaje natural. Watson evolucionó para convertirse en un sistema versátil de aprendizaje automático.
Alexa (2014). Un asistente virtual que permite a los usuarios interactuar con dispositivos mediante comandos de voz.
Redes Generativas Adversarias (GANs) (2014). Introducidas por Ian Goodfellow, permitieron grandes avances en la generación de imágenes, videos, y otros tipos de datos sintéticos.
AlphaFold de DeepMind. Resolvió uno de los mayores desafíos en biología al predecir con precisión la estructura tridimensional de las proteínas a partir de su secuencia de aminoácidos.
Gpt-3(2020). Un chatbot revolucionario para conversaciones automatizadas, utiliza procesamiento del lenguaje natural y aprendizaje profundo para generar texto similar al humano.
Chat GPT(2022). Una iteración de GPT-3, lanzado como un modelo de conversación avanzado capaz de mantener diálogos coherentes y contextualmente relevantes con los usuarios.

Reconocimiento de objetos
El reconocimiento de objetos implica identificar y clasificar objetos dentro de una imagen.

Generación de imágenes
La generación de imágenes implica crear imágenes nuevas a partir de datos de entrada, que pueden ser texto, imágenes base o ruido aleatorio. Esta tarea tiene aplicaciones en arte, entretenimiento, diseño y más. La generación de imágenes ha visto avances extraordinarios con la introducción de modelos como StyleGAN, DALL-E, y MidJourney.

La generación de video es una extensión de la generación de imágenes, donde se crean secuencias de imágenes coherentes que forman videos. Esta tarea es más compleja debido a la necesidad de mantener la coherencia temporal y espacial entre los fotogramas. Avances recientes en esta área incluyen modelos como DeepDream y SORA, que aplican técnicas de visión por computadora para generar secuencias de video surrealistas.

Subtitulación de imágenes

La subtitulación de imágenes implica generar una descripción textual coherente y precisa de una imagen. Este problema combina visión por computadora y procesamiento del lenguaje natural. Los modelos para subtitulación de imágenes, como los basados en arquitecturas CNN-RNN y transformers, han mejorado la precisión y coherencia de las descripciones generadas.

Modelos como CLIP y Flamingo combinan visión por computadora y procesamiento del lenguaje natural para lograr descripciones más precisas.

Medicina

Los algoritmos de IA ahora igualan o superan a los doctores expertos, particularmente cuando el diagnóstico se basa en imágenes. Algunos ejemplos incluyen la enfermedad de Alzheimer, cáncer metastásico, enfermedades oftalmológicas y enfermedades de la piel.

Por ejemplo, el sistema LYNA logra una precisión general del 99.6% en el diagnóstico de cáncer de mama metastásico, mejor que un experto humano no asistido.

Reconocimiento de voz

En 2017, Microsoft mostró que su Sistema de Reconocimiento de Voz Conversacional había alcanzado una tasa de error de palabras del 5.1%, igualando el rendimiento humano en la tarea Switchboard, que implica transcribir conversaciones telefónicas.

Traducción automática

Permiten la lectura de documentos en más de 100 idiomas, incluyendo los idiomas nativos de más del 99% de los humanos, y rinden cientos de miles de millones de palabras por día para cientos de millones de usuarios.

Sistemas de recomendación
Este funciona recomendando lo que podría gustarle a cada usuario basándose en experiencias pasadas y en las de otros usuarios con gustos similares.
El sistema de recomendación esta evolucionando con nuevos métodos de aprendizaje profundo que analizan contenido (texto, música, video) así como historia y metadatos.
El filtrado de spam también puede considerarse una forma de recomendación.

Conducción autónoma
Las primeras demostraciones de conducción autónoma en carretera sin guías especiales ocurrieron en la década de 1980. Pioneros en este campo como Kanade et al. (1986) y Dickmanns y Zapp (1987) llevaron a cabo experimentos iniciales que mostraron el potencial de los vehículos autónomos.

Desafío DARPA
Un hito significativo en el desarrollo de vehículos autónomos fue el Desafío DARPA de 132 millas en 2005: fue una competencia para vehículos autónomos a través del desierto de Mojave.
El equipo liderado por Sebastian Thrun logró completar el recorrido, demostrando que los vehículos podían navegar por terrenos complejos de manera autónoma.
En 2007, el Desafío llevó la prueba a un entorno urbano, donde los vehículos tuvieron que navegar por calles con tráfico, respetar señales de tránsito y evitar obstáculos.

¿Qué es la terminal?
Es una interfaz de línea de comandos que permite a los usuarios interactuar directamente con el sistema operativo mediante la entrada de texto.
A diferencia de las interfaces gráficas, donde se utilizan ventanas, iconos y menús, en la terminal todo se maneja a través de comandos escritos, lo que ofrece una forma poderosa y flexible de controlar el sistema.

Ventajas de usar la terminal
Permite ejecutar comandos y scripts rápidamente sin la

necesidad de navegar por múltiples ventanas y menús.
Proporciona un control más preciso sobre las operaciones del sistema.
Es ideal para usuarios avanzados que necesitan realizar tareas complejas de manera eficiente. Facilita la creación de scripts para automatizar procesos repetitivos o complejos.

Terminales en diferentes sistemas operativos

Windows.

CMD (Command Prompt). CMD, o Símbolo del Sistema.
Aunque es simple y fácil de usar, CMD tiene limitaciones en comparación con terminales más modernas como PowerShell.
PowerShell. Fue introducido por primera vez en Windows XP y Windows Server 2003 como un complemento, y a partir de Windows 7 y Windows Server 2008 R2, viene preinstalado de manera predeterminada en el sistema operativo.
Es una herramienta de línea de comandos avanzada que utiliza un lenguaje de scripting basado en .NET, lo que le permite ejecutar scripts más complejos y realizar tareas de administración y automatización de manera más eficiente que CMD.

MacOS:

La terminal en MacOS proporciona un entorno de línea de comandos similar al de sistemas Linux, utilizando principalmente el shell Bash (hasta MacOS Catalina) y ahora, desde MacOS Catalina en adelante, el shell Zsh como predeterminado.
Esta terminal permite a los usuarios ejecutar comandos Unix, desarrollar scripts, y utilizar herramientas de desarrollo y administración comunes en entornos de sistemas operativos basados en Unix.

Linux:

Bash: Bash, que significa Bourne Again Shell, es el shell predeterminado en la mayoría de las distribuciones de Linux. Es un shell avanzado que deriva de sh (Bourne Shell) y añade características adicionales, como mejoras en la edición de comandos, historial de comandos, y capacidades avanzadas de scripting.

Permite a los usuarios ejecutar comandos del sistema, escribir scripts para automatizar tareas, manipular archivos, y gestionar procesos y redes de manera eficiente.

Microprocesador.

Un microprocesador es un chip o circuito integrado que contiene una CPU en su interior. Es el dispositivo físico que implementa la funcionalidad de la CPU en un solo chip.

Central Processing Unit (CPU).

El procesador central, o CPU (Central Processing Unit), es el cerebro de cualquier sistema computacional.

El CPU esta compuesto de varios elementos que trabajan en conjunto para realizar operaciones de procesamiento de datos.

Unidad Aritmética y Lógica (ALU). Es responsable de realizar operaciones matemáticas y lógicas.

Unidad de control. Gestiona la ejecución de instrucciones dentro del CPU, controlando el flujo de datos entre la ALU, los registros y la memoria.

Registros. Son pequeñas unidades de almacenamiento dentro del CPU que mantienen datos temporales, como operandos y resultados intermedios.

Memoria caché. La caché es una memoria de alta velocidad que almacena datos e instrucciones que se utilizan con frecuencia.

Núcleos de Procesamiento. Un CPU moderno puede tener múltiples núcleos, permitiendo la ejecución de múltiples hilos en paralelo.

La ley de Moore

La ley de Moore dice que el número de transistores en un chip de computadora se duplicaba aproximadamente cada dos años, lo que a su vez duplicaba la capacidad de procesamiento y permitía reducir los costos por transistor.

Altair 8800

Es considerada una de las primeras microcomputadoras personales de la historia y fue lanzada en enero de 1975 por la empresa MITS (Micro Instrumentation and Telemetry Systems).

La ley de Wirth

La ley de Wirth dice que el software tiende a volverse más lento de manera más rápida que la mejora en el hardware.

Graphics Processing Unit (GPU)

La GPU es un tipo de procesador especializado en manejar y acelerar el procesamiento de gráficos y cálculos complejos de manera más eficiente que una CPU .

Originalmente diseñadas para renderizar gráficos en computadoras y consolas de videojuegos, las GPUs han evolucionado para desempeñar un papel crucial en una variedad de aplicaciones más allá de los gráficos, como la inteligencia artificial, el aprendizaje profundo, la simulación científica y la minería de criptomonedas.

Unidades de Procesamiento Tensorial (TPU)

La TPU esta diseñada específicamente para el aprendizaje automático.

A diferencia de las GPU, que tienen miles de núcleos, las TPU cuentan con miles de Unidades Multiplicadoras de Matrices (MXUs) y están optimizadas para operaciones específicas de redes neuronales, como la multiplicación de matrices.

Características de la TPU

Las TPU tienen miles de Unidades Multiplicadoras de Matrices (MXUs).

La TPU original tenía una cuadrícula de multiplicadores de 256 x 256 de 8 bits, realizando un paso clave en muchos algoritmos de aprendizaje automático.

La TPU también tenía 24 MB de RAM en el chip para facilitar los cálculos y 4 MB de registros de 32 bits.

Memoria y almacenamiento

La memoria principal está compuesta principalmente de Memoria de Acceso Aleatorio (RAM).

Toda la memoria en una computadora está numerada, con cada byte teniendo su propia dirección.

Las direcciones suelen representarse en números hexadecimales (base 16), que utilizan los dígitos del 0 al 9 y las letras A, B, C, D, E y F para representar valores. Por ejemplo, en una computadora con direcciones de 32 bits, una dirección de memoria podría ser 0x030FA024.

Memoria RAM (Random Access Memory)
La RAM es un tipo de memoria volátil que permite acceder de manera directa a cualquier ubicación de memoria mediante su dirección, lo que facilita la lectura y escritura de datos.
La RAM permite acceso directo a cualquier byte de la memoria. Existen varios tipos de RAM, como SDRAM, DDR SDRAM, o RDRAM, que difieren en características electrónicas, pero desde el punto de vista de la ciencia de la computación, todos cumplen la misma función.

Memoria de Solo Lectura (ROM)
Es un tipo de memoria no volátil, lo que significa que retiene la información almacenada incluso cuando se apaga el sistema.
La ROM no permite modificar los datos una vez que han sido escritos, o al menos no de manera sencilla o rápida.
Por ejemplo, en una computadora, el BIOS (Basic Input/Output System) se almacena en la ROM, y es el responsable de iniciar el sistema cuando se enciende.

Memoria Mapeada a E/S
Algunas direcciones de la memoria principal están reservadas para interactuar con dispositivos de entrada/salida.
Cuando la CPU escribe en estas direcciones, no está alterando la RAM, sino enviando instrucciones a los dispositivos.

Memoria Cache
Los procesadores modernos contienen varios niveles de caché (L1, L2, L3), que son pequeñas áreas de memoria rápida utilizadas para almacenar datos que la CPU utiliza con frecuencia.
Este almacenamiento reduce el tiempo que la CPU necesita para acceder a la memoria principal.
Las unidades de almacenamiento como los discos duros y SSD también utilizan memoria caché para acelerar las operaciones de lectura y escritura.

Almacenamiento en discos
Discos Duros (HDD). Basados en un sistema magnético de grabación de datos en platos giratorios, los discos duros ofrecen grandes cantidades de almacenamiento a un costo reducido, pero son más lentos que las SSD.
Unidades de Estado Sólido (SSD). Utilizan memoria flash para el almacenamiento de datos. Son más rápidas que los discos duros y permiten acceder a los datos de manera casi instantánea. Son más caras, pero tienen la ventaja de no tener partes móviles, lo que reduce el riesgo de fallas mecánicas.

¿**Qué es un ambiente virtual?**
Un ambiente virtual es un entorno aislado que permite gestionar dependencias, paquetes y configuraciones específicas para un proyecto sin afectar el sistema global o otros proyectos.
Esto garantiza que cada proyecto tenga sus propias versiones de librerías, evitando conflictos y facilitando la reproducibilidad.

Máquina Virtual (VM)
Una maquina Virtual o VM crea una simulación completa de una computadora física. Ejecuta un sistema operativo completo independiente del anfitrión.
Las maquinas virtuales o VM crean un aislamiento total del sistema operativo y recursos (CPU, memoria, almacenamiento, red).
Las maquinas virtuales o VM ocupan un alto consumo de recursos (CPU, RAM, almacenamiento). Cada VM necesita su propio SO y configuraciones.
Las maquinas virtuales o VM requieren software de virtualización (VMware, VirtualBox, Hyper-V, etc.) y la instalación de un sistema operativo.
Las maquinas virtuales o VM requieren más configuración y mantenimiento (instalación de SO, asignación de recursos).
Las maquinas virtuales o VM son de tamaño grande. Incluye

todo un sistema operativo, lo que lo hace más pesado en términos de almacenamiento y recursos.
Por ejemplo las maquinas virtuales ayudan a Ejecutar una aplicación que requiere una versión específica de Windows en un equipo con Linux. - Crear entornos de pruebas controlados.

Ambiente Virtual
Un ambiente virtual es un entorno aislado dentro del sistema operativo para gestionar dependencias y configuraciones de software.
Un ambiente virtual crea un aislamiento parcial enfocado en dependencias y configuraciones de software. Comparte el sistema operativo base.
Un ambiente virtual requiere un consumo de recursos reducido. Solo se aísla el entorno de ejecución del software y las dependencias.
Un ambiente virtual requiere herramientas de desarrollo como venv (Python 3.3+), virtualenv , o conda.
Un ambiente virtual es fácil de crear y manejar para proyectos de software, especialmente con herramientas como venv o conda .
Un ambiente Virtual es de tamaño ligero. Solo incluye dependencias y configuraciones específicas del proyecto.
Un ambiente virtual te ayuda a crear entornos Python aislados para proyectos específicos. - Evitar conflictos entre versiones de librerías en diferentes proyectos.
¿Para qué necesitamos usar ambientes virtuales?
Evita conflictos entre versiones de paquetes utilizados en diferentes proyectos.
Facilita la replicación del entorno de desarrollo en diferentes máquinas o por otros desarrolladores.
Simplifica la instalación, actualización y eliminación de paquetes específicos para cada proyecto.
Reduce riesgos al limitar el alcance de paquetes y dependencias a un entorno específico.
Entornos de cómputo
El término "cómputo" o "computación" se refiere al proceso de convertir información en datos procesables.

Entornos de cómputo El término "cómputo" o "computación" se refiere al proceso de convertir información en datos procesables.
Los entornos de cómputo determinan cómo se desarrollan, ejecutan y optimizan las aplicaciones de IA.
Dependiendo del tipo, cada entorno ofrece capacidades y limitaciones que afectan el rendimiento y la escalabilidad de los sistemas de IA.

Entorno personal o local

Un entorno de computación personal utiliza una computadora, estación de trabajo o servidor local para satisfacer las necesidades de un solo usuario a la vez, donde todos los componentes dehardware están conectados entre sí para realizar tareas personales.
Este tipo de entorno se utiliza en el desarrollo inicial y pruebas de modelos de IA, donde no se requiere una gran capacidad de procesamiento o almacenamiento.

PYTORCH

Es un framework de deep learning desarrollado por Facebook AI Research (FAIR).
Fue creado en 2016 por el equipo de FAIR (Facebook AI Research), liderado por Soumith Chintala.
Estructura basada en tensores y su capacidad de proporcionar cálculos automáticos de derivadas lo hacen ideal para crear redes neuronales. PyTorch permite el uso eficiente de GPUs para acelerar los cálculos. Es popular tanto en investigación como en aplicaciones industriales.

Entorno de computación distribuida

Este tipo de entorno distribuye las tareas computacionales a través de múltiples máquinas, servidores o nodos, lo que permite ejecutar modelos y procesar datos de manera paralela.
Es ideal para manejar grandes volúmenes de datos y modelos complejos.

Hadoop y Spark. Son frameworks utilizados para procesar grandes volúmenes de datos distribuidos en múltiples servidores. Se pueden usar para entrenar modelos de IA que analizan datos masivos.

Entorno de cómputo en la Nube

La computación en la nube ofrece la posibilidad de alquilar recursos computacionales como almacenamiento, procesamiento, redes y análisis, sin necesidad de infraestructura física local.
Esto es esencial para la escalabilidad en IA, ya que permite a los usuarios acceder a recursos masivos bajo demanda.
La computación en la nube ha revolucionado la forma en que se gestionan y procesan los datos. En lugar de depender de hardware físico local, los usuarios pueden acceder a recursos de cómputo (como almacenamiento, procesamiento y software) a través de Internet.

Modelos de servicio en la Nube
IaaS (Infrastructure as a Service) Proporciona infraestructura virtualizada, como servidores y almacenamiento, donde los usuarios tienen control total sobre el sistema operativo y las aplicaciones.
PaaS (Platform as a Service). Proporciona una plataforma que permite a los desarrolladores crear, ejecutar y gestionar aplicaciones sin preocuparse por la infraestructura subyacente.
SaaS (Software as a Service): Los usuarios acceden a aplicaciones alojadas en la nube (como Google Drive o Microsoft 365) directamente a través de Internet, sin necesidad de instalar software en sus dispositivos.
Plataformas de servicios de la nube AWS, Azure, Oracle, Google cloud, etc.

Entorno de cómputo en el borde (Edge computing)
El cómputo en el borde es una arquitectura que lleva el procesamiento de datos más cerca de la fuente de generación, como dispositivos IoT, sensores o smartphones. En lugar de enviar todos los datos a un servidor central, el procesamiento se realiza localmente en los dispositivos.

Cómputo Híbrido
El entorno híbrido combina el cómputo local, la nube y, en algunos casos, el cómputo en el borde para ofrecer una infraestructura flexible. Este modelo es útil cuando se necesita aprovechar las ventajas de varios entornos.
Entornos de Desarrollo Integrado (IDEs)
Un Entorno de Desarrollo Integrado (IDE) es una

herramienta que proporciona a los desarrolladores un conjunto de utilidades para.
Escribir, Depurar y Ejecutar código de manera eficiente.
Los IDEs generalmente combinan un editor de texto, un compilador o intérprete, herramientas de depuración y otras características, en una única interfaz gráfica o de línea de comandos.

Características de un IDE
Editor de código. Permite escribir código con soporte para sintaxis de diferentes lenguajes de programación.
Depurador. Herramienta que facilita encontrar errores en el código mediante la ejecución paso a paso y análisis de variables.
Compilador/Intérprete. Ejecuta el código directamente o lo compila en un formato ejecutable.
Autocompletado y sugerencias. Sugiere palabras clave y corrige errores de sintaxis, mejorando la productividad.
Gestión de proyectos. Facilita la organización de múltiples archivos y dependencias en un solo proyecto.
Integración con sistemas de control de versiones. Como Git, para gestionar cambios en el código de manera colaborativa.

PyCharm
PyCharm, desarrollado por JetBrains, es uno de los IDEs más utilizados para Python.
Ideal para grandes proyectos que requieren pruebas automatizadas, integración continua y trabajo en equipo.
Características de Pycharm
Refactorización de código. PyCharm permite realizar cambios estructurales en el código de forma automática y segura.
Depurador gráfico. Visualiza los procesos y variables en tiempo real, facilitando la corrección de errores.
Integración con Docker y entornos virtuales. Ideal para proyectos de Python que requieren la administración de entornos de ejecución.
Soporte para frameworks web. Como Django y Flask, lo que lo hace muy útil para el desarrollo de aplicaciones web.
Testing integrado: Herramientas como pytest y unittest están completamente integradas.

Visual Studio Code (VSCode)
Visual Studio Code, desarrollado por Microsoft, es un editor de texto con características de IDE, ampliamente utilizado por desarrolladores por su flexibilidad y ecosistema de extensiones.
Ideal para proyectos de inteligencia artificial donde el uso de Jupyter Notebooks es común, por ejemplo, en la implementación de modelos de deep learning o análisis de datos.

Spyder
Spyder es un IDE diseñado específicamente para científicos de datos y programadores que trabajan en proyectos de análisis numérico y visualización en Python.

JupyterLab
JupyterLab es una plataforma que permite crear y compartir documentos que contienen código ejecutable, ecuaciones, visualizaciones y texto narrativo. Es el entorno más común para el trabajo con Jupyter Notebooks.
Es ampliamente utilizado en la academia y en la industria para prototipar modelos de aprendizaje automático y realizar análisis exploratorios de datos.

¿Qué es el Deep Learning?
El Aprendizaje Automático (ML) es un campo dentro de la IA que se enfoca en desarrollar algoritmos que permitan a las computadoras aprender a partir de los datos.
El Deep Learning, en español Aprendizaje Profundo, es una rama del aprendizaje automático que se enfoca en el entrenamiento de algoritmos para aprender a representar datos de manera jerárquica, a través de múltiples capas de procesamiento.
Estas capas forman un modelo de red neuronal artificial, inspirado en la estructura y funcionamiento del cerebro humano.

El perceptrón es un modelo de red neuronal de una sola capa, que podía aprender a clasificar patrones simples. Se utilizó principalmente para problemas de clasificación lineal.

A pesar de su capacidad para resolver problemas lineales, el perceptrón no podía manejar problemas no lineales como el problema XOR, lo que llevó a un período de estancamiento en el desarrollo de redes neuronales.

El perceptrón multicapa, también conocido como red neuronal artificial, permitió el aprendizaje de representaciones no lineales mediante la adición de capas ocultas.

Las redes neuronales profundas, con muchas capas de procesamiento, demostraron ser altamente efectivas en una amplia gama de tareas, desde reconocimiento de voz hasta conducción autónoma.

Estructura del perceptrón simple

Entradas. Representan las características o variables de entrada del problema que se está abordando.
Pesos. Cada conexión entre una entrada y una neurona tiene asociado un peso que representa la fuerza de la influencia de esa entrada en la salida del perceptrón. Estos pesos se ajustan durante el proceso de entrenamiento para minimizar el error de predicción.
Función de activación. Después de que se calcula la suma ponderada de las entradas multiplicadas por los pesos, se aplica una función de activación al resultado. Esta función decide si la neurona se activa o no, y puede introducir no linealidades en el modelo.
Salida. La salida del perceptrón se calcula como la aplicación de la función de activación al resultado de la suma ponderada de las entradas. Dependiendo de la función de activación, la salida puede ser binaria (0 o 1) o continua en un rango específico.

El funcionamiento del perceptrón se puede dividir en dos fases principales: propagación hacia adelante (forward propagation) y ajuste de pesos.
1. Propagación hacia adelante (forward propagation). los datos de entrada se multiplican por los pesos correspondientes y se suman para obtener una suma ponderada. Luego, esta suma se pasa a través de la función

de activación, y la salida del perceptrón se calcula en base al resultado de esta función.
2. Ajuste de pesos (backpropagation) Una vez que se ha calculado la salida del perceptrón, se compara con la salida deseada (etiquetada). Si hay discrepancias entre la salida real y la deseada, los pesos de las conexiones se ajustan utilizando un algoritmo de aprendizaje. El objetivo es minimizar el error de predicción, lo que se logra actualizando los pesos en la dirección que reduce este error.

Limitaciones del perceptrón
Aunque el perceptrón es útil para problemas de clasificación linealmente separables, tiene varias limitaciones
No puede resolver problemas no lineales.
No puede aprender a partir de datos no etiquetados.
Tiene dificultades para aprender relaciones complejas entre las características de entrada y la salida.

Como funciona una red neuronal
División de los datos en batches. Los datos de entrada se dividen en lotes (batches) de un tamaño específico durante el entrenamiento de la red neuronal.
Cada lote consiste en un subconjunto de datos de entrenamiento. Estos datos se propagan a través de la red neuronal en cada iteración durante el entrenamiento, y se utilizan para calcular el gradiente de la función de pérdida con respecto a los parámetros del modelo.

Épocas
Una época es una iteración completa sobre todo el conjunto de datos de entrenamiento. Durante una época, se procesan todos los lotes de datos y se actualizan los parámetros del modelo en función de la pérdida calculada para cada lote.
El número de épocas es un hiperparámetro que se elige antes de entrenar la red neuronal y determina cuántas veces se va a iterar sobre todo el conjunto de datos de entrenamiento.
Es común ejecutar múltiples épocas para permitir que el modelo aprenda de manera más completa y se ajuste mejor a los datos.

Funciones de activación

Las funciones de activación son funciones matemáticas aplicadas a la salida de cada neurona en una red neuronal. Estas funciones son necesarias para introducir no linealidades en la red, permitiendo que la red pueda aprender y modelar relaciones más complejas en los datos.

Funciones de pérdida

Las funciones de pérdida son funciones que cuantifican qué tan bien está realizando la red neuronal en sus predicciones en comparación con las etiquetas verdaderas (objetivo) de los datos de entrenamiento.

El objetivo durante el entrenamiento de la red neuronal es minimizar esta función de pérdida, ajustando los parámetros del modelo para que las predicciones sean lo más cercanas posible a los valores verdaderos.

Algunas funciones de pérdida comunes incluyen

Entropía Cruzada Categórica. utilizada para problemas de clasificación de múltiples clases.

Entropía Cruzada Binaria. utilizada para problemas de clasificación binaria.

Error Cuadrático Medio (MSE). utilizada para problemas de regresión.

Retropropagación del Error (Backpropagation)

Una vez que se ha calculado la salida de la red y se ha evaluado la función de costo, utilizamos el algoritmo de retropropagación para calcular los gradientes de la función de costo con respecto a los pesos y los sesgos de la red. Luego, actualizamos estos parámetros utilizando un método de optimización como el descenso del gradiente.

Tipos de redes neuronales

Una red neuronal perceptrón multicapa (MLP) es simplemente una combinación de muchas de estas neuronas artificiales (tambien llamadas unidades).

En un MLP, las unidades se organizan en un conjunto de capas, y cada capa contiene un número de unidades idénticas.

Cada neurona en una capa está conectada a todas las neuronas en la siguiente capa; decimos que la red está completamente conectada (en implementación,

normalmentenos referimos a este tipo de arquitecturas como capas densas):
Capa de entrada: La capa de entrada recibe los datos de entrada y transmite estos valores a las neuronas de la siguiente capa.
Capas ocultas: Cada neurona en una capa oculta toma las entradas de la capa anterior, realiza una combinación lineal ponderada y aplica una función de activación no lineal.
Capa de salida: La capa de salida produce los resultados de la red neuronal.
La cantidad de neuronas en esta capa depende del problema que se esté resolviendo. Por ejemplo, para problemas de clasificación binaria, podría haber una sola neurona de salida con una función de activación que represente la probabilidad de pertenecer a una clase.
El número de capas se conoce como profundidad y el número de unidades en una capa se conoce como ancho.

Redes Neuronales Profundas
Las redes neuronales profundas consisten en múltiples capas de neuronas interconectadas.
Se les llama profundas debido a la presencia de múltiples capas ocultas entre la capa de entrada y la capa de salida.
Al igual que en las arquitecturas anteriores, ajustan los pesos de las conexiones entre neuronas para minimizar una función de pérdida, lo que permite que la red aprenda a mapear entradas a salidas de manera efectiva.

Perceptrón Simple tiene una sola capa de neuronas sin capas ocultas, con profundidad baja, capacidad de aprendizaje baja, representaciones lineales, ayuda a la clasificación de datos linealmente separables.
Perceptrón Multicapa tiene una o más capas ocultas entre la capa de entrada y la de salida, profundidad moderada, capacidad de aprendizaje moderada, representaciones Jerárquicas y menos complejas, ayuda a el reconocimiento de patrones, clasificación de datos, aproximación de funciones.
Red Neuronal Profunda tiene múltiples capas ocultas entre la capa de entrada y la de salida, profundidad alta, capacidad de aprendizaje alta, representaciones jerárquicas y menos complejas, ayuda al Reconocimiento de imágenes,

procesamiento del lenguaje natural, generación de contenido.

Tipos de Redes Neuronales Profundas
Redes Convolucionales (CNN)
Son especialmente adecuadas para el procesamiento de datos de tipo imagen.
Utilizan operaciones de convolución para aprender automáticamente características espaciales jerárquicas de las imágenes, como bordes, formas y texturas, a medida que las señales se propagan a través de la red.
Esto permite que las CNNs capturen eficazmente la estructura local en las imágenes y generalicen bien a nuevas imágenes.

Capa de convolución (Convolutional Layer)
Estas capas son fundamentales en las redes neuronales convolucionales (CNN).
Realizan operaciones de convolución en los datos de entrada utilizando múltiples filtros (kernels) para extraer características locales.
Los pesos de los filtros son parámetros que se aprenden automáticamente durante el entrenamiento y pueden capturar patrones espaciales en datos como imágenes.
Funcionamiento
Se utilizan múltiples filtros (o kernels), que son pequeñas matrices de pesos, con dimensiones típicamente como 3 ×3 5×5 o .
Cada filtro se desliza sobre la entrada (imagen), multiplicando sus valores por los valores correspondientes de la región de entrada y sumando el resultado. Este proceso se repite en toda la imagen (operación de convolución), lo que genera un conjunto de mapas de características.
Cada mapa resultante resalta una característica específica de la entrada, como bordes, texturas o patrones. Cuantos más filtros tenga la capa de convolución, más características diferentes podrá capturar.
Después de aplicar la convolución, se aplica una función de activación no lineal, como ReLU (Rectified Linear Unit), para introducir no linealidades en la red y permitirle aprender representaciones más complejas de los datos.
Parámetros de la capa: Además del número de filtros, se pueden especificar el stride (paso) y el padding (relleno)

para controlar el tamaño de la salida. El stride determina el desplazamiento del filtro en cada paso, mientras que el padding agrega ceros alrededor de la entrada para controlar el tamaño de la salida.

Redes Recurrentes (RNN)

Las redes recurrentes (RNNs) son una clase especial de redes neuronales diseñadas para procesar datos secuenciales, como series temporales o texto.

Las RNNs tienen conexiones que les permiten mantener una memoria interna de las entradas pasadas (conexiones recurrentes).

Estas conexións permite que la salida de la unidad en un paso de tiempo se reintroduzca como entrada en el siguiente paso de tiempo. Esto crea un bucle de retroalimentación que permite a la red mantener información sobre las entradas pasadas.

Esta capacidad les permite capturar dependencias temporales en los datos y modelar patrones a largo plazo.

Funcionamiento

En una RNN, las capas ocultas se comprimen para formar una sola capa, que se despliega en el tiempo para formar una estructura recurrente.

Esto significa que la salida de la capa anterior se alimenta a la siguiente capa para hacer predicciones, repitiendo la misma estructura, de ahí el nombre de "Recurrent Neural Network" o Red Neuronal Recurrente.

En una RNN, se utilizan los siguientes elementos

X: Entrada.

O: Salida.

h(t): Representa un estado oculto en el tiempo t y actúa como "memoria" de la red.

V: Representa la comunicación de un paso de tiempo a otro.

Redes Generativas Antagónicas (GANs)

Las Redes Generativas Antagónicas (GANs) son un tipo de aprendizaje automático donde el objetivo es modelar la distribución de un conjunto dado de datos.

Modelos generativos

Hay cuatro tipos de modelos generativos profundos

1. Redes Generativas Adversarias
2. Arquitecturas reversibles
3. Modelos autoregresivos
4. Autoencoders variacionales

Las GANs son un tipo de modelo generativo implícito
Las GANs son una red neuronal para producir muestras, lo que define implícitamente una distribución de probabilidad (la distribución de muestras que genera la red).

La arquitectura de un modelo generativo implícito es la siguiente:
1. Muestreamos un vector de ruido de una distribución simple y fija, como una distribución uniforme o una gaussiana estándar N(0, I).
2. Este vector se pasa luego como entrada a un generador determinista G. El generador es produce una salida que se supone que es similar a los datos de entrenamiento.
3. El discriminador es responsable de distinguir entre los datos reales y los datos generados por el generador. Se le proporciona una entrada y debe determinar si la entrada es real o falsa.

Transferencia de Estilo (CycleGAN)
CycleGAN es una arquitectura para hacer transferencia de estilo de imágenes.
Transferencia de estilo es tomar una imagen en un estilo (como una fotografía) y transformarla para que sea un estilo diferente (como una pintura de van Gogh) mientras se preserva el contenido de la imagen (por ejemplo, objetos y sus ubicaciones).