

SESIÓN DEEP LEARNING

CONTENIDOS:

- Qué es deep learning.
- Diferencias entre inteligencia artificial, machine learning y deep learning.
- Aplicaciones de deep learning: arquitecturas de redes neuronales.
 - Arquitecturas más utilizadas.
 - Principales características y Casos que resuelven.
- Frameworks para deep learning y sus principales características: tensorflow. Keras. Py-torch.

QUÉ ES DEEP LEARNING

El Deep Learning o aprendizaje profundo es una rama del aprendizaje automático (Machine Learning) que se basa en redes neuronales artificiales con múltiples capas (de ahí el término "profundo"). Estas redes son capaces de aprender representaciones de datos en niveles jerárquicos, lo que permite abordar tareas complejas como el reconocimiento de imágenes, el procesamiento de lenguaje natural, la visión por computadora, y más.

En Deep Learning, las redes neuronales están estructuradas en capas que transforman los datos de entrada en una salida deseada, aprendiendo automáticamente las características relevantes sin necesidad de intervención humana. Este proceso es altamente eficaz para tareas donde los métodos tradicionales de programación o análisis de datos fallan o no son lo suficientemente precisos.

DIFERENCIAS ENTRE INTELIGENCIA ARTIFICIAL, MACHINE LEARNING Y DEEP LEARNING

Inteligencia Artificial (IA)

La IA es un campo amplio que busca crear máquinas capaces de realizar tareas que, de ser realizadas por seres humanos, requerirían inteligencia, como entender el lenguaje natural, tomar decisiones o reconocer patrones. Algunos ejemplos son: Sistemas expertos, chatbots, asistentes virtuales.

La IA abarca una variedad de enfoques, incluidos los algoritmos tradicionales y el aprendizaje automático.

1.- Machine Learning (Aprendizaje Automático)

El Machine Learning es una subdisciplina de la IA que se enfoca en el desarrollo de algoritmos que permiten a las máquinas aprender de los datos y mejorar con el tiempo sin ser programadas explícitamente para cada tarea.

- **Ejemplos:** Algoritmos de clasificación, regresión, agrupación, y sistemas de recomendación.
- **Enfoque:** El Machine Learning se basa en el uso de datos para entrenar modelos y hacer predicciones o tomar decisiones.

1.2.- Deep Learning (Aprendizaje Profundo)

El Deep Learning es una subcategoría del Machine Learning que se centra en el uso de redes neuronales profundas (con múltiples capas) para modelar datos complejos y no estructurados. Las redes neuronales profundas tienen la capacidad de aprender representaciones de alto nivel a partir de grandes cantidades de datos.

- **Ejemplos:** Reconocimiento de voz, visión por computadora, traducción automática.
- **Enfoque:** El Deep Learning es particularmente poderoso para tareas de procesamiento de datos en grandes volúmenes y donde las características complejas deben ser aprendidas de manera automática.

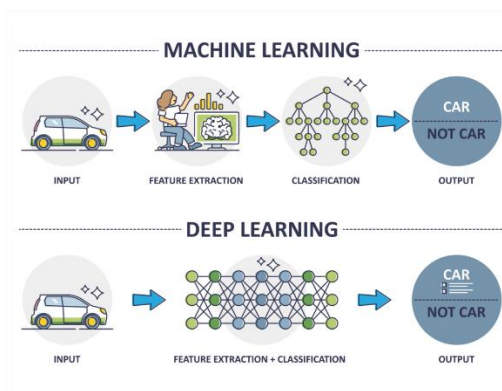


Ilustración 1 Gráfico diferencias Machine Learning, Deep Learning . Codigospython.com

APLICACIONES DE DEEP LEARNING: ARQUITECTURAS DE REDES NEURONALES

Las arquitecturas de redes neuronales son fundamentales en Deep Learning, ya que permiten abordar tareas complejas en áreas como visión por computadora, procesamiento de lenguaje natural y reconocimiento de voz. A continuación, se describen algunas de las arquitecturas más utilizadas, sus características y los casos que resuelven.

1. Redes Neuronales Convolucionales (CNN - Convolutional Neural Networks)

Características:

- **Capas Convolucionales:** Utilizan filtros (kernels) que realizan convoluciones sobre la entrada para extraer características locales, como bordes o texturas, sin necesidad de una intervención manual.
- **Capas de Pooling:** Reducción de la dimensionalidad de las características extraídas, manteniendo solo la información más relevante (max pooling o average pooling).
- **Capas Completamente Conectadas:** Típicamente al final de la red, estas capas realizan la clasificación de las características extraídas.
- **Extracción Automática de Características:** No es necesario preprocesar o etiquetar explícitamente los datos antes de entrenar la red.

Casos que Resuelve:

- **Visión por Computadora:** Como el reconocimiento de objetos, clasificación de imágenes y detección de caras.
- **Diagnóstico Médico:** Análisis de imágenes médicas como radiografías, resonancias magnéticas o tomografías computarizadas.
- **Automóviles Autónomos:** Detección de obstáculos y tráfico a través de cámaras y sensores.

2. Redes Neuronales Recurrentes (RNN - Recurrent Neural Networks)

Características:

- **Conexiones Recurrentes:** A diferencia de las redes neuronales tradicionales, las RNN tienen conexiones que permiten que la salida de una capa dependa de su entrada y de las salidas previas, lo que las hace útiles para procesar secuencias de datos.

- **Memoria de Corto Plazo:** Retienen información temporal de las entradas anteriores, lo que les permite manejar secuencias de longitud variable (texto, sonido, etc.).
- **Celdas de Activación:** Utilizan celdas de activación como **LSTM (Long Short-Term Memory)** o **GRU (Gated Recurrent Unit)**, que ayudan a resolver problemas de desvanecimiento de gradientes en secuencias largas.

Casos que Resuelve:

- **Procesamiento de Lenguaje Natural (NLP):** Traducción automática, análisis de sentimientos, generación de texto.
- **Reconocimiento de Voz:** Conversión de voz a texto y entendimiento del lenguaje hablado.
- **Predicción de Series Temporales:** Como la predicción del precio de acciones o el comportamiento de usuarios.

3. Redes Generativas Antagónicas (GAN - Generative Adversarial Networks)

Características:

- **Dos Redes en Competencia:** Las GAN consisten en dos redes, el **Generador** y el **Discriminador**, que compiten entre sí. El generador crea datos falsos (por ejemplo, imágenes) mientras que el discriminador evalúa si los datos generados son reales o falsos.
- **Entrenamiento en Dos Fases:** Mientras una red mejora en generar datos más realistas, la otra mejora en distinguir los datos reales de los generados.
- **Generación de Datos Sintéticos:** Son muy buenas para crear datos nuevos a partir de ejemplos existentes.

Casos que Resuelve:

- **Generación de Imágenes:** Creación de imágenes realistas a partir de ruido aleatorio o de ejemplos existentes (por ejemplo, arte, moda, fotos falsas).
- **Mejora de Imágenes:** Restauración de imágenes de baja resolución o eliminación de ruido.
- **Modelado de Datos:** En campos como la biología y la química, las GAN se usan para generar moléculas o compuestos que podrían tener propiedades deseables.

4. Redes Neuronales de Transformadores (Transformers)

Características:

- **Atención:** En lugar de procesar datos secuenciales de manera lineal como las RNN, los transformers utilizan un mecanismo de **atención**, que permite al modelo centrarse en diferentes partes de la entrada al procesar cada elemento de la secuencia.
- **Paralelización:** A diferencia de las RNN, los transformers pueden procesar las secuencias en paralelo, lo que mejora significativamente el tiempo de entrenamiento.
- **Escalabilidad:** Son muy eficientes en el manejo de grandes volúmenes de datos y secuencias largas.

Casos que Resuelve:

- **Procesamiento de Lenguaje Natural:** Modelos como **BERT** y **GPT** (como el que estamos usando) se basan en arquitecturas de transformers para tareas como la traducción automática, resumen de texto, generación de texto, etc.
- **Traducción de Lenguaje:** Traducción de un idioma a otro con gran precisión.
- **Reconocimiento de Intentos:** En asistentes virtuales y chatbots, donde se debe entender la intención detrás de un texto dado.

5. Redes Neuronales de Autoencoders (Autoencoders)

Características:

- **Codificación y Decodificación:** Un autoencoder consta de dos partes: un **codificador** que reduce la dimensionalidad de la entrada y un **decodificador** que reconstruye los datos a partir de esta representación comprimida.
- **Aprendizaje No Supervisado:** El objetivo es aprender una representación densa y eficiente de los datos.
- **Reducción de Ruido y Compresión:** Pueden ser utilizados para eliminar ruido de los datos o comprimir imágenes y señales.

Casos que Resuelve:

- **Reducción de Dimensionalidad:** Comprimir datos complejos y reducir su tamaño.
- **Eliminación de Ruido:** Filtrar ruido de imágenes o señales de audio.
- **Generación de Nuevas Muestras:** Generar nuevos ejemplos de datos basados en los patrones aprendidos.

Resumen de arquitecturas y casos de uso

Arquitectura	Características Clave	Casos de Resuelve
CNN	Convolución, Pooling, Capa densa	Reconocimiento de imágenes, diagnóstico médico, vehículos autónomos.
RNN	Conexiones recurrentes, secuencias	Procesamiento de texto, reconocimiento de voz, predicción de series temporales.
GAN	Generador vs. Discriminador	Generación de imágenes, restauración de imágenes, modelado de datos.
Transformers	Atención, paralelización	Traducción, generación de texto, comprensión de lenguaje.
Autoencoders	Codificación y decodificación	Reducción de ruido, compresión de datos, eliminación de ruido.

FRAMEWORKS PARA DEEP LEARNING Y SUS PRINCIPALES CARACTERÍSTICAS

El desarrollo de modelos de Deep Learning se facilita enormemente con el uso de frameworks especializados. Tres de los frameworks más populares son **TensorFlow**, **Keras** y **PyTorch**, cada uno con características únicas que lo hacen adecuado para diferentes tipos de tareas.

TensorFlow

Características Principales de TensorFlow:

- **Desarrollado por Google:** TensorFlow es un framework de código abierto desarrollado por Google para la construcción y entrenamiento de redes neuronales y modelos de aprendizaje automático.
- **Escalabilidad:** TensorFlow es altamente escalable, lo que significa que puede manejar grandes volúmenes de datos y entrenar modelos complejos tanto en CPUs como en GPUs, y en múltiples dispositivos.
- **Compatible con Python y otros lenguajes:** Aunque su API principal es para Python, TensorFlow también tiene interfaces para otros lenguajes como JavaScript, Java y C++.
- **Ecosistema Completo:** Ofrece un ecosistema completo para Deep Learning, incluyendo herramientas de visualización como **TensorBoard**, y soporte para la implementación de modelos en producción con **TensorFlow Serving**.
- **TensorFlow Lite y TensorFlow.js:** Permite implementar modelos en dispositivos móviles (TensorFlow Lite) y en navegadores (TensorFlow.js), lo que facilita su despliegue en aplicaciones del mundo real.
- **TensorFlow Hub:** Proporciona modelos preentrenados que pueden ser reutilizados en nuevas aplicaciones para ahorrar tiempo y recursos en el entrenamiento.



Ilustración 2 Logo TensorFlow

Casos de uso comunes:

- **Visión por Computadora:** Reconocimiento de imágenes, clasificación y segmentación.
- **Procesamiento de Lenguaje Natural:** Traducción automática, generación de texto, análisis de sentimientos.
- **Predicción en Series Temporales:** Para aplicaciones como predicción financiera o de demanda en el mercado.

Keras

Características Principales de Keras:

- **API de Alto Nivel:** Keras es una API de alto nivel para construir y entrenar redes neuronales profundas. Aunque fue desarrollado originalmente como un framework independiente, ahora está integrado en TensorFlow.
- **Facilidad de Uso:** Keras está diseñado para ser fácil de usar y rápido para prototipado, lo que lo hace ideal para principiantes y para quienes buscan construir modelos rápidamente sin complicarse demasiado con los detalles.
- **Modularidad:** La arquitectura de Keras permite agregar, quitar y modificar capas de manera fácil y flexible. Esto hace que el diseño de modelos sea intuitivo.
- **Compatibilidad con TensorFlow:** Aunque Keras puede ser usado de manera independiente, generalmente se utiliza como una interfaz de alto nivel para TensorFlow, lo que le permite aprovechar la potencia y escalabilidad de TensorFlow.
- **Soporte para Redes Neuronales Convolucionales y Recurrentes:** Keras permite la implementación fácil de CNN y RNN, y tiene soporte para LSTM y GRU.
- **Compatibilidad con Modelos Preentrenados:** Keras proporciona acceso a modelos preentrenados para tareas comunes como clasificación de imágenes y procesamiento de texto.



Ilustración 3 Logo Keras

Casos de Uso:

- **Prototipado Rápido:** Crear modelos de Deep Learning rápidamente sin perder tiempo en configuraciones complejas.
- **Modelos de Visión por Computadora:** Convolución y clasificación de imágenes.
- **Modelos de Lenguaje Natural:** Generación de texto, clasificación y análisis de sentimientos.

PyTorch

Características Principales de PyTorch:

- **Desarrollado por Facebook:** PyTorch es un framework de Deep Learning de código abierto creado por Facebook. Es ampliamente utilizado tanto en investigación como en aplicaciones industriales.
- **Tensor Computation:** Al igual que TensorFlow, PyTorch permite realizar cálculos con tensores, pero su enfoque es más intuitivo y flexible.
- **Dynamic Computational Graph:** PyTorch utiliza un **grafo computacional dinámico**, lo que significa que las redes se definen en tiempo de ejecución. Esto permite mayor flexibilidad al momento de modificar los modelos, lo cual es útil para tareas de investigación y prototipado rápido.
- **Facilidad para Depuración:** Gracias a su integración con Python, PyTorch es fácil de depurar utilizando herramientas estándar de Python, como el depurador integrado y el entorno interactivo.

- **Integración con Otras Bibliotecas:** PyTorch se integra bien con otras bibliotecas populares de Python como **NumPy**, **SciPy** y **Matplotlib**, lo que lo hace útil para trabajos científicos y experimentales.
- **Soporte para GPU:** Tiene un excelente soporte para la ejecución en GPU, lo que mejora significativamente la eficiencia en el entrenamiento de redes neuronales complejas.



Ilustración 4 Logo PyTorch

Casos de Uso:

- **Investigación Académica:** Gracias a su flexibilidad y facilidad para modificar modelos, PyTorch es muy popular en la comunidad de investigación.
- **Visión por Computadora y Procesamiento de Lenguaje Natural:** Implementación de modelos avanzados para tareas como la segmentación de imágenes, análisis de sentimientos, y generación de texto.
- **Entrenamiento de Modelos Complejos:** Usado en investigación avanzada en Deep Learning y en aplicaciones como la simulación de sistemas y la optimización.