Newsletter Trimestral

CÁTEDRA

# iDANAE

INTELIGENCIA · DATOS · ANÁLISIS · ESTRATEGIA

## 3T24

# Nuevos paradigmas y sus desafíos en la construcción de redes neuronales


POLITÉCNICA

UNIVERSIDAD POLITÉCNICA DE MADRID

MS Management Solutions
Making things happen

# Introducción

A lo largo de las últimas décadas, el desarrollo de las redes neuronales en el campo de la inteligencia artificial (IA) las ha convertido en una herramienta muy efectiva para resolver problemas complejos. Desde la capacidad de clasificar imágenes hasta traducir idiomas y generar texto, el uso de estas redes ha permitido mejorar el paradigma de cómo las máquinas aprenden y se adaptan al entorno. Detrás de cada gran avance en este campo, existe una red neuronal cuidadosamente diseñada y entrenada. Pero no todas las redes neuronales son iguales. Su arquitectura, complejidad y capacidad de aprendizaje varían drásticamente, lo que influye no solo en su rendimiento, sino también en los recursos necesarios para implementarlas.

A medida que el campo ha evolucionado, han surgido diversas arquitecturas que responden a necesidades específicas. Desde los inicios de las redes multicapa simples, hasta las sofisticadas redes convolucionales, las recurrentes y los últimos modelos transformadores, cada una tiene un propósito específico y requiere una infraestructura adaptada. Esto incluye elementos como la potencia de procesamiento o el almacenamiento, y se han de considerar asimismo el coste energético o la necesidad de la escalabilidad de recursos, lo que hace que cada arquitectura tenga su propio ecosistema.

En este *whitepaper* se exploran las arquitecturas más relevantes de redes neuronales, entendiendo cómo están estructuradas, sus componentes y funcionamiento, y se analizan nuevas tendencias y sus desafíos técnicos.

## Evolución de las redes neuronales y sus arquitecturas

**Las redes neuronales artificiales (RNA)** han recorrido un largo camino desde su origen en la década de 1940 hasta convertirse en una pieza central de la inteligencia artificial moderna. Se puede establecer el inicio de este campo cuando Warren McCulloch y Walter Pitts propusieron un modelo matemático diseñado para simular el funcionamiento de las neuronas biológicas, el cual sentó las bases teóricas para el desarrollo de las RNA [1]. En los siguientes años se introdujeron reglas, como la regla de Hebb, u otros descubrimientos como el ***perceptrón***, desarrollado por Frank Rosenblatt en 1958 [2]. Este se convirtió en uno de los primeros modelos prácticos de red neuronal, capaz de clasificar entradas en dos categorías [3]. Sin embargo, la década de 1980 supuso un primer resurgimiento de la IA, en el que aparecieron las redes neuronales multicapa y el algoritmo de retropropagación del error, que permitió a las redes modelar funciones no lineales de gran complejidad [4], así como grandes avances en el hardware (como la *Connection Machine*).

Un segundo resurgimiento de la IA tuvo lugar a partir de 2010, impulsado por el auge del aprendizaje profundo, por nuevas evoluciones del hardware (como los *clústeres* con aceleradores tipo GPU), y por la liberación de *frameworks* gratuitos (como, por ejemplo, *TensorFlow*). Las **redes neuronales profundas (DNN)** permitieron modelar relaciones jerárquicas complejas gracias a los avances en potencia computacional y la disponibilidad de grandes conjuntos de datos. Modelos como **las redes convolucionales (CNN)** y **las redes neuronales recurrentes (RNN)** revolucionaron la visión por computadora y el procesamiento del lenguaje natural [5].

Hoy en día, las RNA continúan evolucionando, con el desarrollo de modelos avanzados como las **redes generativas adversarias (GAN)** y los ***transformers***. Estos últimos han transformado el procesamiento de lenguaje natural y otras áreas clave de la IA debido a su capacidad para gestionar relaciones complejas y grandes volúmenes de datos de manera eficiente. En el futuro se prevé un crecimiento continuo, consolidando el papel fundamental de las RNA en el avance de sistemas inteligentes.

## *Principios básicos de las redes neuronales*

**Las redes neuronales artificiales (RNA)** representan una de las tecnologías más revolucionarias en el campo de la IA. En términos generales, una RNA define unas funciones matemáticas que contienen unos parámetros ajustables. Estos parámetros se ajustan a partir de los valores de unos datos o muestras de entrenamiento, mediante un proceso de optimización de una función de pérdidas, conceptualmente análogo a la regresión lineal. Para entender su funcionamiento, es útil desglosar sus componentes clave:

- Una **neurona** es la unidad básica de procesamiento, equivalente a la neurona biológica. Cada neurona recibe **entradas**, las procesa a través de una función matemática y genera una salida. Estas entradas están ponderadas por valores numéricos llamados **pesos**, que determinan la importancia de cada entrada en la decisión final de la neurona [5]. Los pesos se ajustan durante el entrenamiento para optimizar el rendimiento del modelo.

- Las **capas** están formadas por neuronas. Cada neurona situada en una capa está conectada a todas las neuronas de la siguiente capa (o a parte de ellas, según cuál sea el patrón de conectividad –denso, convolucional, etc.–), formando una estructura de red. Las capas se dividen en tres tipos:

  - La **capa de entrada** recibe los datos iniciales, como datos procedentes de variables asociadas a cada muestra, o información de imágenes o texto (preprocesados y transformados a través de su representación en valores numéricos), que serán procesados por la red.

  - Las **capas ocultas** transforman los datos aplicando funciones no lineales, extrayendo características de mayor complejidad a medida que los datos recorren la estructura de la red. Estas capas permiten que la red capture patrones profundos en los datos.

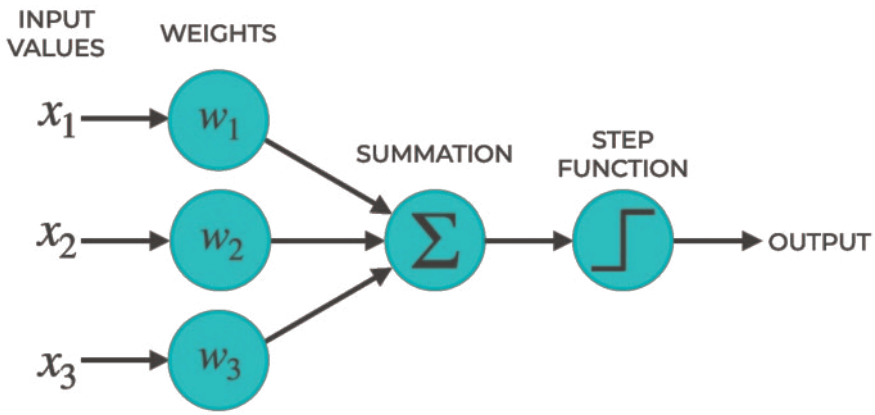
  - La **capa de salida** produce el resultado final, que puede ser una clasificación (como determinar si una imagen contiene un gato o un perro) o un valor continuo en problemas de regresión.

- La **función de activación** forma parte de cada neurona y es el componente que introduce no linealidad en el modelo para que la red pueda aprender relaciones complejas. Algunas funciones como la sigmoide o la tangente hiperbólica son útiles en redes menos profundas o con tareas específicas. En cambio, ReLU es utilizada por su eficiencia en redes profundas.

- La **función de pérdida** define el objetivo a alcanzar durante el entrenamiento y evalúa el rendimiento del modelo midiendo el error. Las más comunes son la entropía cruzada y el error cuadrático medio, utilizados en problemas de clasificación y regresión, respectivamente [6]. Además, esta función de pérdida se puede modificar, añadiendo penalizaciones para establecer procedimientos de regularización, que pueden ser útiles para dotar de mayor capacidad de generalización a la red.

- La **retropropagación** es el proceso de ajuste de los pesos de las conexiones neuronales para minimizar el error. Utiliza el gradiente de la función de pérdida para modificar los pesos en función del error observado entre la salida deseada y la salida predicha [4].

*The structure of a perceptron*




Fuente: Ebner, J. (2023).The Structure of a Perceptron[Fotografía]. Perceptrons, Explained.https://www.sharpsightlabs.com/blog/perceptrons-explained/

- **La optimización** es el proceso de ajuste de parámetros para mejorar el desempeño del modelo. Algoritmos como ***stochastic gradient descent (SGD)***, Adam y RMSprop son muy útiles para encontrar los valores óptimos de los pesos en las conexiones neuronales [7].

En función de la disposición de los elementos de la red, la disposición de los flujos de los datos y los mecanismos de entrenamiento elegidos, se pueden construir distintos tipos de arquitecturas de redes, dando lugar a distintos tipos de redes neuronales. Las arquitecturas clásicas de redes neuronales más comunes son las siguientes (para una descripción detallada de cada una de ellas, véase el ladillo en este documento):

- ***Perceptrón* multicapa (MLP):** múltiples capas de neuronas (capa de entrada, capas intermedias y capa de salida), donde generalmente todas las neuronas de una capa están conectadas con todas las de la siguiente capa.

- **Redes neuronales convolucionales (CNN):** redes neuronales donde en la primera parte de la arquitectura se realizan operaciones de combinación o *convolución* de distintos datos de un *grid*, aplicando filtros para detectar patrones.

- **Redes neuronales recurrentes (RNN):** la arquitectura de estas redes permite analizar secuencias, puesto que se establecen conexiones hacia capas anteriores (o hacia la entrada de la misma capa) que permiten formar ciclos, manteniendo así la información de eventos anteriores, que actúa como una "memoria" en la secuencia.

- **Redes adversarias generativas (GAN):** es una red donde el entrenamiento se realiza a partir de dos componentes: un generador de datos sintéticos similares a los datos reales de entrenamiento, y un discriminador, que trata de clasificar los datos entre los sintéticos creados anteriormente y los reales. Cuando el discriminador no es capaz de distinguir entre datos reales y sintéticos, el generador está ya entrenado para generar nueva información.

No obstante, han surgido nuevas estructuras de redes, como los *transformers* o las redes neuronales de grafos (GNN), que han tenido un impacto muy relevante en el campo de la inteligencia artificial generativa y en el desarrollo de nuevas aplicaciones de IA. Estas redes se describen a continuación.

## Arquitecturas clásicas de redes neuronales

### *Perceptrón* multicapa (MLP)

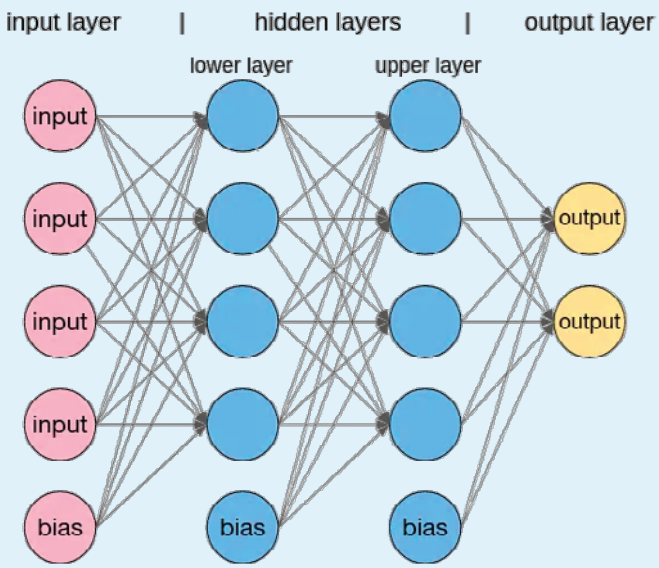
El ***Perceptrón* multicapa (MLP)** surgió como una evolución del *perceptrón* simple, superando sus limitaciones y permitiendo la resolución de problemas no linealmente separables, gracias a la inclusión de las capas ocultas. Esta arquitectura consiste en múltiples capas de neuronas: una capa de entrada, una o más capas ocultas, y una capa de salida. Cada conexión entre nodos tiene un peso asociado, y cada nodo (excepto los de entrada) tiene una función de activación no lineal. Esto permite que la red aprenda representaciones más complejas de los datos, haciendo posible su aplicación en una variedad de tareas, como la clasificación de imágenes, el procesamiento de señales y el análisis de datos tabulares [8].

El aprendizaje en un MLP se realiza mediante un proceso llamado retropropagación, que iterativamente ajusta los pesos de las conexiones entre las neuronas para minimizar la función de pérdida y, como consecuencia, el error en las predicciones. Esta capacidad de ajuste hace que el MLP sea extremadamente versátil y efectivo en diversas aplicaciones [4].

#### Aplicaciones prácticas

- **Clasificación de imágenes.** Aunque las redes neuronales convolucionales (CNN) son más comúnmente utilizadas para tareas de visión por computadora, los MLP también pueden aplicarse a la clasificación de imágenes, especialmente cuando son de baja resolución o cuando los recursos computacionales

*Basic structure of the MLP*




Fuente: An, S. (2021). Basic structure of the MLP [Fotografía]. Introduction to how an Multilayer Perceptron works but without complicated math.https://medium.com/codex/introduction-to-how-an-multilayer-perceptron-works-but-without-complicated-math-a423979897ac

son limitados. Por ejemplo, un MLP puede entrenarse para clasificar imágenes de dígitos en categorías del 0 al 9. En este caso, cada píxel de la imagen actúa como una neurona en la capa de entrada, y el MLP aprende a reconocer patrones en los píxeles que corresponden a cada dígito [9].

- **Procesamiento de datos tabulares.** Predicción de precios de viviendas o la clasificación de clientes en marketing. Cada característica de los datos (como el tamaño de la vivienda, el número de habitaciones, la ubicación, etc.) se introduce en la red como una neurona en la capa de entrada. El MLP procesa estas características a través de sus capas ocultas para predecir una variable objetivo, como el precio de la vivienda o la probabilidad de que un cliente realice una compra [10].

## Redes neuronales convolucionales (CNN)

Las **Redes neuronales convolucionales (CNN)** son un tipo de arquitectura especialmente diseñada para procesar datos que tienen una estructura de tipo *grid*, como las imágenes. Desarrolladas inicialmente para abordar problemas en visión por computadora, las CNN han demostrado ser extremadamente efectivas en tareas como el reconocimiento de imágenes, detección de objetos y segmentación semántica, entre otros [11].

Las CNN son una variante de los MLP, en las que se modifica el patrón de conexión entre capas de neuronas, y se basan en dos conceptos clave:

- La ***convolución*** es una operación matemática que se aplica a la entrada o a capas intermedias para extraer características importantes, como bordes, texturas y patrones. Este proceso se realiza mediante filtros o ***kernels***, que son matrices pequeñas que se deslizan sobre la entrada (como una imagen) para generar un mapa de características [5]. Los filtros en la *convolución* aprenden a detectar características específicas en las imágenes a medida que la red se entrena. Por ejemplo, los filtros de las primeras capas pueden detectar bordes, mientras que los filtros en capas más profundas pueden identificar formas más complejas como ojos, caras o incluso objetos [11].
- El proceso de ***pooling***, en particular el ***max-pooling,*** reduce la dimensionalidad de los mapas de características, conservando la información más importante, lo que ayuda a evitar el sobreajuste y reducir el coste de entrenamiento e inferencia. *Max-pooling* selecciona el valor máximo de una región particular del mapa de características, resumiendo la información de manera eficiente [12]. Las CNN también utilizan funciones de activación no lineales como ReLU (unidad lineal rectificada) para introducir no linealidad en el modelo, permitiendo aprender representaciones más complejas [5].

### Aplicaciones prácticas

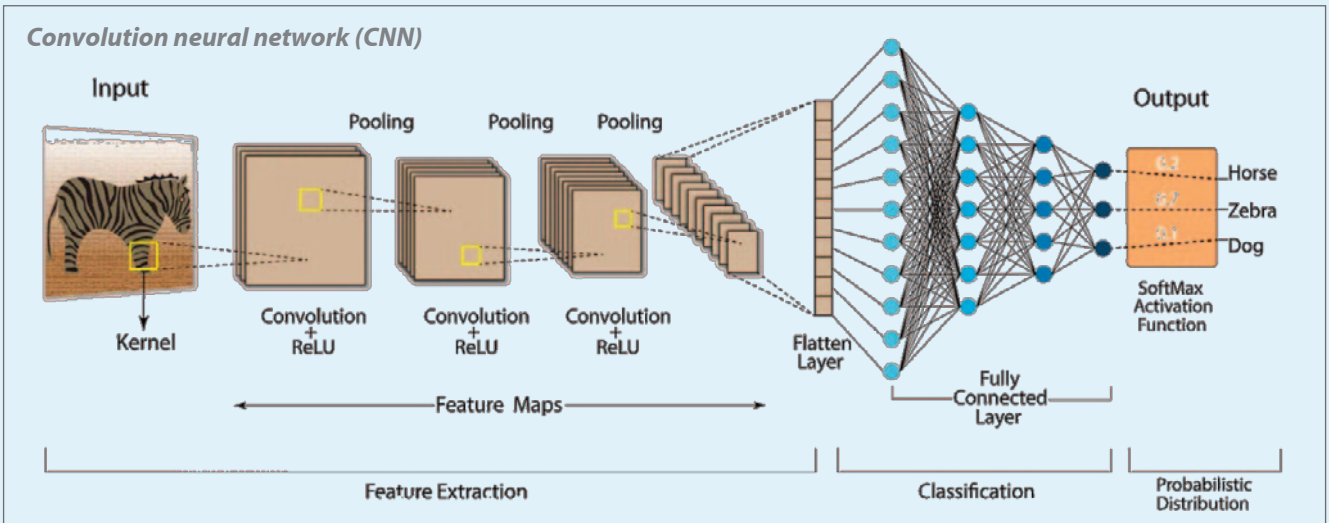
- **Clasificación de imágenes.** Las CNN pueden identificar a qué categoría pertenece una imagen dada, por ejemplo, si una imagen contiene un gato o un perro [11].
- **Visión por computador.** Modelos como R-CNN, YOLO y SSD utilizan arquitecturas basadas en CNN para localizar y clasificar múltiples objetos en una sola imagen, con aplicaciones en sistemas de conducción autónoma y vigilancia, para reconocer señales de tráfico, peatones y otros vehículos [13].
- **Segmentación de imágenes.** En la segmentación de imágenes, la imagen se divide en diferentes regiones, asignando una etiqueta a cada píxel, lo cual es esencial en aplicaciones médicas como la segmentación de tumores en imágenes de resonancia magnética [14].

## Redes neuronales recurrentes (RNN)

Las redes neuronales recurrentes (RNN) son diseñadas específicamente para procesar datos secuenciales, donde la información previa en la secuencia es crucial para la predicción actual. A diferencia de las redes neuronales tradicionales, las RNN tienen conexiones que forman ciclos en su arquitectura, lo que les permite mantener un **"estado"** o **memoria** de eventos anteriores en la secuencia [15]. Esto las hace adecuadas para tareas donde el contexto histórico es importante, como en el procesamiento de lenguaje natural (NLP) y el análisis de series temporales.

El funcionamiento básico de una RNN implica la actualización de su estado oculto en cada paso de la secuencia, utilizando tanto la entrada actual como el estado oculto anterior. Sin embargo, las RNN estándar presentan problemas como **desvanecimiento** o **explosión del gradiente,** lo que dificulta el aprendizaje de dependencias a largo plazo en las secuencias. Para mitigar estos problemas, se han desarrollado variantes más avanzadas como las *long short-term memory* (LSTM) y las *gated recurrent units* (GRU) [8].

- Las **LSTM** fueron introducidas para superar las limitaciones de las RNN tradicionales al incluir mecanismos de puertas ***(gates)*** que controlan el flujo de información, permitiendo a la red recordar u olvidar información de manera más efectiva. Las puertas de entrada, olvido y salida son las principales

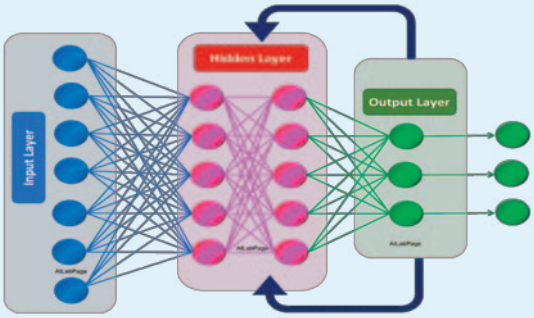



Fuente: Shahriar, N. (2023). Convolution Neural Network (CNN)[Fotografía]. What is Convolutional Neural Network — CNN (Deep Learning).https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5

responsables de la capacidad de las LSTM para manejar dependencias a largo plazo [16].

- Las **GRU** son una simplificación de las LSTM que combinan la puerta de entrada y la puerta de olvido en una sola, lo que reduce la complejidad computacional sin sacrificar significativamente el rendimiento. Las GRU han demostrado ser igualmente efectivas en muchas aplicaciones, y a veces se prefieren por su menor coste computacional [17].

***Recurrent Neural Network***




Fuente: Shahriar, N. (2023). Convolution Neural Network (CNN)[Fotografía]. What is Convolutional Neural Network — CNN (Deep Learning).https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5

### Aplicaciones prácticas

- **Análisis de series temporales.** Las RNN se utilizan para predecir valores futuros basados en datos históricos. Por ejemplo, la predicción del mercado financiero, o la detección de anomalías en señales temporales [18].
- **Procesamiento de lenguaje natural (NLP).** Las RNN se han usado para tareas como traducción automática, análisis de sentimientos, generación de texto y reconocimiento de voz [19].

## Redes adversarias generativas (GAN)

Las **Redes adversarias generativas (GAN)** son una clase innovadora de redes neuronales que han revolucionado el campo de la IA, especialmente en la generación de datos sintéticos, como imágenes. Las GAN destacan por su capacidad de generar datos que son prácticamente indistinguibles de los datos reales [20].

El núcleo de las GAN se basa en un proceso de entrenamiento competitivo entre dos redes neuronales: el generador y el discriminador.

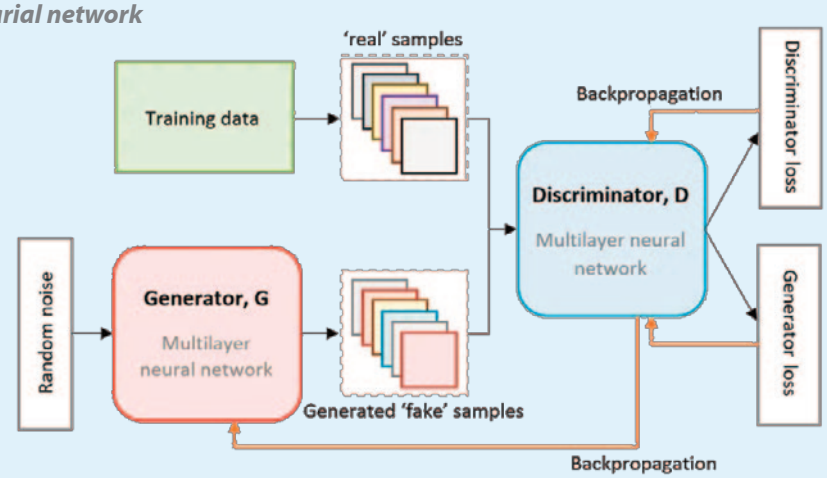
- El **generador** toma una entrada aleatoria y la transforma en datos sintéticos que se asemejan a los datos reales. El objetivo del generador es producir datos tan realistas que el discriminador no pueda distinguirlos de los datos verdaderos. A medida que el generador se entrena, aprende a crear ejemplos cada vez más convincentes [21].
- El **discriminador** actúa como un clasificador binario que intenta diferenciar entre los datos reales y los datos generados por el generador. Durante el entrenamiento, el discriminador recibe ejemplos tanto del conjunto de datos reales como de los datos generados, y su tarea es mejorar continuamente su capacidad para distinguir entre ambos [22].

El equilibrio se alcanza cuando el discriminador ya no puede diferenciar entre los datos reales y los generados con más de un 50% de probabilidad, momento en el que se considera que la GAN ha alcanzado un punto óptimo de aprendizaje [23].

### Aplicaciones prácticas

- **Generación de imágenes fotorrealistas** a partir de entradas aleatorias. Esto incluye la creación de rostros humanos que no existen en la realidad, generación de obras de arte originales y la mejora de la resolución de imágenes borrosas, conocida como superresolución [24].
- ***Deepfakes***. Las GAN generan vídeos y audios falsos que parecen ser reales. Esta tecnología se ha utilizado tanto en la industria cinematográfica para efectos especiales, como para la creación de contenidos engañosos que plantean desafíos éticos y de seguridad [25].
- **Otras aplicaciones.** Las GAN se utilizan en tareas como la generación de música, diseño de moda, simulaciones para entrenamiento de modelos de IA, y la creación de datos sintéticos para entrenar otros modelos cuando los datos reales son escasos o costosos de obtener [26].

***Generative adversarial network***




Fuente: Little, C., Elliot, M. J., Allmendinger, R. & Samani, S.S. (2021). GAN Architecture Diagram[Fotografía]. Generative Adversarial Networks for Synthetic Data Generation: A Comparative Study. CC BY 4.0.

# Nuevas arquitecturas de redes neuronales

## *Arquitectura transformer*

Los ***transformers*** han redefinido radicalmente el campo de la IA, especialmente en el ámbito de la inteligencia artificial generativa, con un gran impacto en aplicaciones como el procesamiento del lenguaje natural (NLP) y, más recientemente, la generación de contenidos multimedia (imagen, vídeos, música, etc.). Esta arquitectura fue propuesta en 2017, y su innovación principal es el mecanismo de **auto-atención,** que permite procesar secuencias de datos en paralelo y de manera eficiente, mitigando las limitaciones de modelos anteriores como las redes neuronales recurrentes (RNN) o las LSTM. Los *transformers* han sido esenciales para el desarrollo de modelos avanzados como BERT y GPT, que han revolucionado el procesamiento de texto, la traducción automática y la generación de lenguaje, entre otras tareas [27].

Los *transformers* poseen varias características que los han hecho sobresalir en la IA:

- **Paralelización**. Los *transformers* son mucho más rápidos en el entrenamiento que las RNN y LSTM debido a que permiten procesar las secuencias de datos de manera paralela, lo que aprovecha mejor las arquitecturas modernas de hardware como las GPU [27].
- **Escalabilidad**. La capacidad de los *transformers* para escalar a modelos masivos, como GPT-4 y GPT-5, ha sido crucial en la creación de redes neuronales con miles de millones de parámetros que pueden aprender de grandes cantidades de datos. Esto es clave para lograr resultados de alta calidad en tareas como la generación de lenguaje [28].
- **Contexto global.** El mecanismo de auto-atención permite a los *transformers* comprender las relaciones a largo plazo en las secuencias, capturando de manera más efectiva la estructura y significado del lenguaje en comparación con los enfoques basados en RNN, que tienen problemas para gestionar dependencias a largo plazo [27] [29].


Como los *transformers* no tienen memoria secuencial como las RNN, utilizan **codificaciones posicionales** para que el modelo entienda la posición relativa de los *tokens* en la secuencia. Estas codificaciones, que se suman a los ***embeddings*** de los *tokens*, introducen información sobre el orden de los elementos en la secuencia, permitiendo al modelo tratar no solo con los datos individuales, sino también con su secuencia [27].

### *La revolución en el ámbito del NLP: desde BERT hasta GPT*

Desde la introducción de los *transformers*, el campo del NLP ha experimentado una revolución sin precedentes. Los modelos como BERT y GPT han llevado las capacidades del lenguaje natural a nuevas alturas, facilitando desde tareas básicas como la traducción automática hasta tareas más avanzadas como la generación de texto y el análisis de sentimiento.

- **BERT *(bidirectional encoder representations from transformers).*** Introducido por Google, BERT ha transformado la forma en que las redes entienden el contexto. A diferencia de los modelos unidireccionales, BERT usa un enfoque bidireccional, es decir, considera tanto el contexto anterior como el posterior para entender el significado de una palabra. Esta capacidad es esencial para tareas como el análisis de sentimientos, donde el sentido de una palabra puede depender de lo que viene antes o después en la oración. BERT ha mejorado el rendimiento en muchas tareas del NLP, como la clasificación de texto y el reconocimiento de entidades [31].
- **GPT *(generative pre-trained transformer).*** Los modelos de la familia GPT, desarrollados por **OpenAI**, son conocidos por su capacidad de generar texto de alta calidad. GPT está basado en un modelo unidireccional, lo que significa que genera texto *token* por *token*, prediciendo la siguiente

palabra a partir del contexto anterior. Esto ha permitido su uso en aplicaciones como *chatbots*, asistentes virtuales y sistemas de recomendación [28].

### *Aplicaciones prácticas*

Existen múltiples aplicaciones que se están desarrollando en la industria. Entre otras, se pueden destacar las siguientes:

- **Traducción automática.** Los *transformers* han mejorado la calidad de las traducciones, permitiendo a modelos como **T5** de **Google *(text-to-text transfer transformer)*** comprender el significado completo de las oraciones y traducirlas de manera más precisa [30].
- **Generación de imágenes a partir de texto.** Modelos como **DALL-E** basados en *transformers* han mostrado capacidades extraordinarias para generar imágenes a partir de descripciones textuales, abriendo nuevas posibilidades en la creación de contenido multimedia y arte generado por IA [29].
- **Aplicaciones médicas.** Los *transformers* también han encontrado aplicaciones en el análisis de grandes volúmenes de datos médicos. Se están utilizando para analizar textos clínicos y generar resúmenes automáticos de investigaciones biomédicas, mejorando la eficiencia en el manejo de datos en la medicina [32].
- **Generación automática de documentación.** El uso de LLM para la generación de documentos está cobrando gran relevancia, dado que permite mejorar la eficiencia de procesos empresariales, reduciendo tanto los tiempos de producción de la documentación como los posibles errores operacionales (actualización, consistencia, etc.).
- **Ayuda para la programación y desarrollo de código.** El uso de esta tecnología permite, entre otros, corregir fallos en el código de una aplicación, generar código a partir de determinadas especificaciones o entender un código desarrollado.

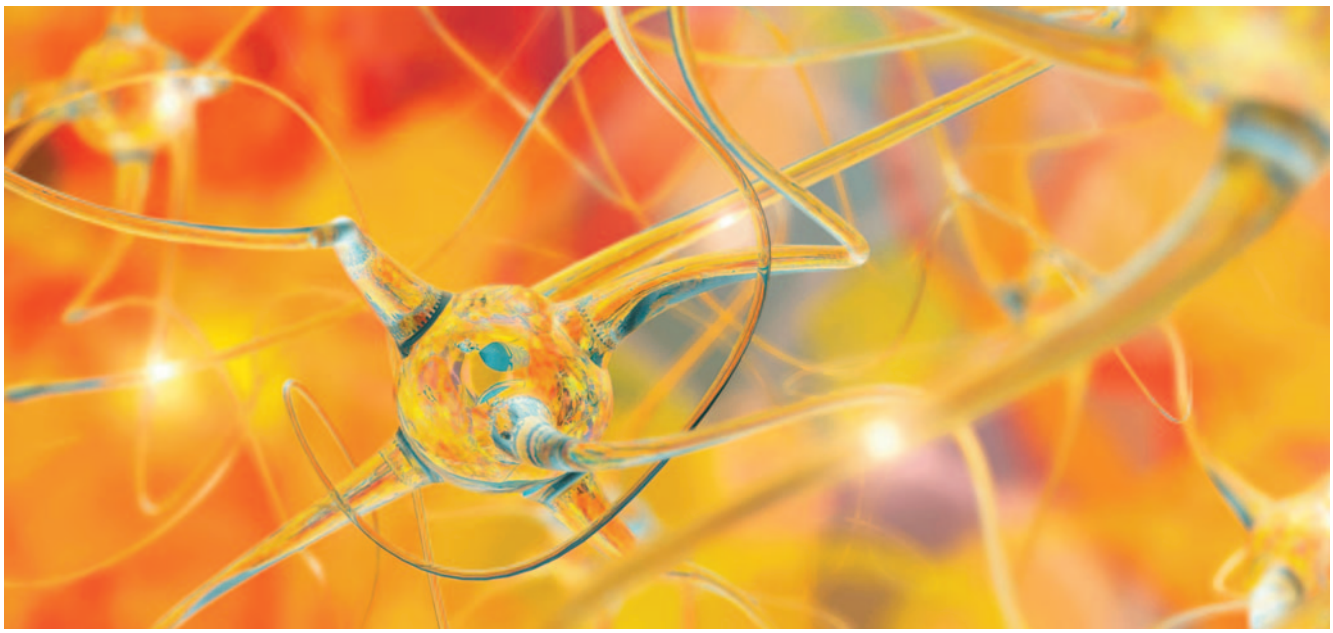
La continua evolución de los *transformers* asegura su papel central en el futuro de la IA, no solo en el procesamiento del lenguaje, sino en múltiples áreas de la ciencia y tecnología.

## *Redes neuronales de grafos (GNN)*

**Las redes neuronales de grafos (GNN)** son un tipo de arquitectura que extiende las capacidades de las redes neuronales tradicionales al trabajar directamente con datos estructurados en forma de grafos. Los grafos son estructuras compuestas de **nodos** (vértices) y **enlaces** (aristas) que representan relaciones entre entidades. Esta arquitectura es especialmente útil para analizar datos con relaciones complejas e interdependientes, como redes sociales, sistemas biológicos o incluso el análisis de moléculas en química y bioinformática [33].

### *Estructura y principios de las GNN*

La estructura fundamental de las GNN se basa en la representación y procesamiento de datos en forma de grafos. A diferencia de las redes neuronales tradicionales, que operan sobre datos tabulares o secuenciales (como el texto o las imágenes), las GNN se diseñan para capturar tanto la información de los nodos individuales como la información de conectividad entre ellos. Las GNN logran esto utilizando un proceso iterativo de agregación de información de los nodos vecinos para aprender representaciones más ricas de los grafos.



Uno de los principios clave detrás de las GNN es el mecanismo de **propagación de mensajes *(message-passing)*.** En cada capa de la red, la información de un nodo se actualiza en función de la información de sus vecinos. El objetivo es que, a través de varias capas, cada nodo logre una representación que incorpore información no solo sobre sí mismo, sino también sobre la estructura local del grafo, es decir, sobre los nodos conectados y sus relaciones.

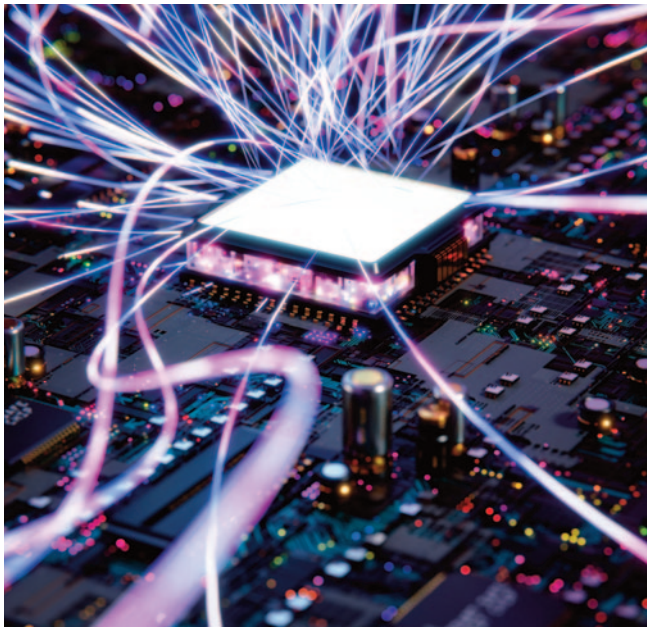
1. **Inicialización.** Cada nodo en el grafo comienza con una representación inicial, que suele basarse en las características o atributos del nodo.

2. **Agregación de información.** En cada iteración, los nodos reciben información de sus nodos vecinos. Este proceso de agregación de la información puede implementarse mediante operaciones como suma, media o concatenación de los vectores de características de los nodos adyacentes.

3. **Actualización.** La nueva representación del nodo se calcula combinando la representación anterior con la información agregada de los vecinos. Esto permite que la representación de cada nodo evolucione para capturar mejor su contexto dentro del grafo.

Este proceso se repite a lo largo de varias capas, permitiendo que las GNN aprendan representaciones ricas que incorporan tanto la información de los nodos individuales como la de su estructura circundante [34].

Al igual que en las CNN (redes neuronales *convolucionales*), las GNN también utilizan operaciones de ***pooling*** o ***readout*** para obtener una representación final de todo el grafo. Después de varias capas de procesamiento, se puede aplicar una función de *pooling* para combinar la información de todos los nodos en una sola representación que resuma el grafo completo. Esto es particularmente útil para tareas donde el objetivo es clasificar un grafo completo, como la predicción de propiedades moleculares en bioinformática [35].



### *Aplicaciones prácticas*

El campo de aplicación de las redes neuronales de grafos es amplio y diverso, dado que los grafos son una representación natural para muchos tipos de datos. Algunas de las aplicaciones más destacadas incluyen:

- **Análisis de redes sociales.** Las redes sociales se pueden modelar como grafos, donde los usuarios son nodos y las relaciones entre ellos son las aristas. Las GNN se utilizan para tareas como:
  - **Detección de comunidades.** Identificar subgrupos de usuarios altamente interconectados dentro de una red.
  - **Predicción de enlaces.** Anticipar futuras conexiones entre usuarios basándose en patrones existentes.
  - **Detección de influencias.** Determinar qué usuarios tienen una influencia desproporcionada dentro de la red [32].
- **Bioinformática.** Las GNN han mostrado ser herramientas valiosas para analizar redes biológicas y moleculares. Las moléculas y proteínas pueden representarse como grafos, donde los átomos son nodos y las uniones químicas son las aristas. Algunas de las aplicaciones incluyen:
  - **Predicción de propiedades moleculares.** Las GNN se han utilizado para predecir propiedades de compuestos químicos, como la actividad biológica o la toxicidad, que son esenciales en el diseño de fármacos.
  - **Interacción proteína-proteína.** En la biología, las interacciones entre proteínas también pueden modelarse como grafos. Las GNN permiten estudiar cómo las proteínas interactúan entre sí y predecir comportamientos basados en estas interacciones [35].
- **Sistemas de recomendación.** Las plataformas comerciales de productos o contenidos audiovisuales pueden modelar sus usuarios y productos como grafos, donde las aristas representan interacciones (como una compra o visualización). Las GNN permiten mejorar los sistemas de recomendación al capturar relaciones más profundas entre usuarios y productos, basadas no solo en las interacciones directas, sino también en las relaciones secundarias dentro del grafo de usuarios y productos.

- **Modelado del conocimiento.** En los sistemas de IA, las bases de conocimiento también pueden representarse como grafos, donde las entidades (como conceptos u objetos) son nodos y las relaciones entre ellas son las aristas. Las GNN se utilizan para tareas como el razonamiento lógico y la inferencia sobre datos en estos grafos, mejorando la capacidad de los sistemas para aprender relaciones complejas entre conceptos y generar inferencias más precisas.

En resumen, las GNN son un avance revolucionario en la IA, proporcionando un enfoque robusto para trabajar con datos no estructurados y relaciones complejas. Su capacidad para modelar redes y grafos ha abierto nuevas posibilidades en campos como las redes sociales, la bioinformática, y los sistemas de recomendación, donde la estructura subyacente de los datos es fundamental para el análisis.

## Mecanismo de auto-atención y su funcionamiento

La arquitectura de los *transformers* está compuesta por bloques ***encoder*** y ***decoder***, donde el *encoder* recibe la entrada y el *decoder* genera la salida, si bien algunos diseños solo incluyen uno de estos dos bloques. Sin embargo, el núcleo de su éxito radica en su enfoque a la **auto-atención.**

El mecanismo de auto-atención es fundamental para el éxito de los *transformers*. En lugar de tratar los datos secuenciales de forma estricta y en orden, como lo hacen las RNN, la auto-atención permite que cada elemento (o *token*) en una secuencia "preste atención" a todos los demás elementos de la misma. Esto se logra calculando la relevancia de cada *token* dentro del contexto de los otros mediante operaciones matriciales entre tres vectores:

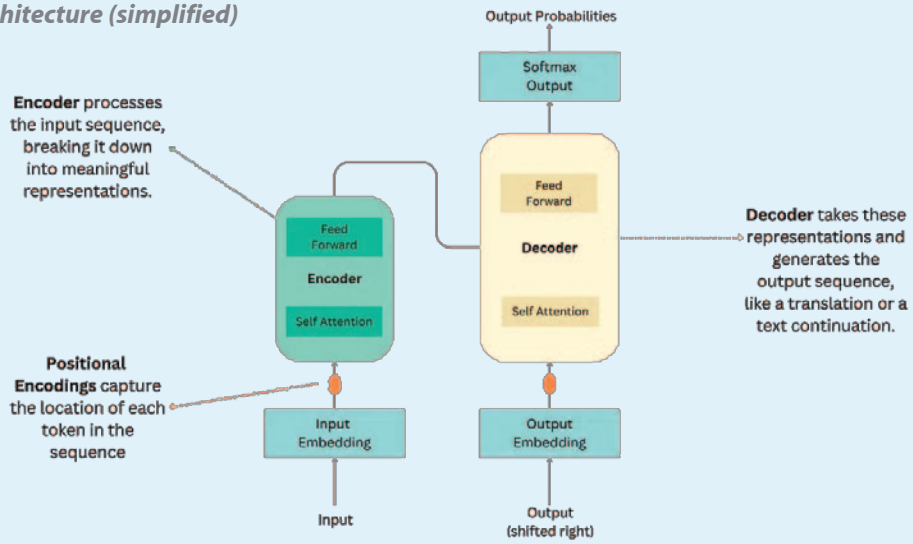
- **Vectores clave:** capturan las características de cada *token*.
- **Vectores consulta:** representan el *token* en cuestión que necesita "atención" sobre otros.
- **Vectores valor:** representan la información real a la que se prestará atención.

El proceso consiste en calcular la similitud entre la consulta y las claves mediante el producto escalar, determinando qué elementos de la secuencia son más importantes para el *token* actual. Esto produce una puntuación de atención, que se aplica a los valores para obtener una representación ponderada de la secuencia.

Una característica clave del mecanismo de auto-atención es el uso de ***multi-head attention,*** que permite que la red aprenda múltiples relaciones semánticas simultáneamente entre los *tokens*. Cada "cabeza" de atención se encarga de capturar diferentes tipos de relaciones en los datos, como la relación entre palabras a corta o larga distancia en un texto. Esto ofrece un contexto global que mejora significativamente el rendimiento de la red en tareas como la comprensión del lenguaje y la generación de texto, además de permitir el procesamiento paralelo, lo que reduce enormemente los tiempos de cómputo.

Por ejemplo, en tareas de traducción automática, el modelo puede captar el significado de una palabra dependiendo del contexto de otras palabras, incluso si están muy alejadas en la oración. Este enfoque supera a las RNN y LSTM, que luchan por mantener la coherencia en secuencias largas [27][30].Después de la capa de atención, los *transformers* utilizan una ***red feed-forward*** para procesar las representaciones de los *tokens*. Cada *token* pasa por una red densa totalmente conectada que introduce complejidad no lineal al modelo. Entre las capas de atención y *feed-forward,* se añade una **capa de normalización** que estabiliza el entrenamiento y ayuda a controlar la propagación de los gradientes, evitando que se amplifiquen o desaparezcan, un problema común en las redes profundas [27].

*Transformer architecture (simplified)*




Fuente: Tech Gumptions. (s.f.). (2023)Transformer Architecture[Fotografía]. Transformer Architecture Simplified.https://medium.com/@tech-gumptions/transformer-architecture-simplified-3fb501d461c8

# Desafíos actuales y futuro de las arquitecturas de redes neuronales

El avance de las redes neuronales ha impulsado enormes desarrollos en IA. Sin embargo, el futuro de estas tecnologías también viene acompañado de desafíos clave, que incluyen la escalabilidad y eficiencia, la interpretabilidad y explicabilidad, la privacidad y seguridad, o la ética. Estos problemas necesitan soluciones tanto en términos de investigación técnica como de implementación práctica para mantener su crecimiento y aceptación.

## Escalabilidad y eficiencia

Con el crecimiento masivo de los modelos neuronales, la escalabilidad y eficiencia se han convertido en temas prioritarios. Modelos como GPT-4 o BERT, que poseen miles de millones de parámetros, requieren infraestructuras avanzadas, lo que resulta en altos costos computacionales y un consumo energético considerable [36].

### *Desafíos en el entrenamiento de modelos a gran escala.*

- **Crecimiento exponencial de los parámetros.** A medida que los modelos crecen en tamaño, la necesidad de recursos aumenta significativamente. Los modelos actuales como **GPT-4,** con más de un billón de parámetros, requieren semanas de entrenamiento en grandes *clústeres* de **GPU** o **TPU**, lo que implica un enorme consumo energético y costes financieros [36]. Según estimaciones, el entrenamiento de estos modelos puede tener una huella de carbono significativa [36].

- Almacenamiento y velocidad de entrenamiento. A medida que los modelos aumentan en tamaño, también lo hace la cantidad de datos que deben procesar, lo que ralentiza el entrenamiento y crea cuellos de botella en los sistemas de almacenamiento y procesamiento de datos. Para mitigar estos problemas, los avances incluyen diferentes técnicas de procesamiento paralelo como ***model parallelism*** (división de las capas del modelo en múltiples máquinas) y ***pipeline parallelism*** (entrenamiento simultáneo de diferentes partes del modelo, donde cada una procesa diferentes fragmentos del *dataset*) [36].

### *Eficiencia en el uso de recursos computacionales*

A medida que las redes neuronales crecen en tamaño y complejidad, la eficiencia en el uso de recursos computacionales se vuelve un reto crítico. Existen varias técnicas y enfoques para abordar esta problemática:

- **Reducción de la complejidad mediante técnicas de compresión de modelos:** métodos como la ***cuantización***, ***pruning*** y ***distillation*** permiten reducir la cantidad de parámetros y operaciones requeridas por una red neuronal sin perder demasiado en precisión. La *cuantización*, por ejemplo, convierte los pesos de la red en formatos de menor precisión (por ejemplo, de 32 a 8 bits), reduciendo el uso de memoria y mejorando la eficiencia computacional [37]. El ***pruning*** elimina conexiones y neuronas innecesarias, mientras que la ***distillation*** permite entrenar redes más pequeñas y eficientes a partir de redes más grandes [38].

- **Algoritmos optimizados para hardware especializado.** Para mejorar la eficiencia, muchas empresas e instituciones están desarrollando **hardware especializado** para redes neuronales, como **TPUs *(tensor processing units)* y chips dedicados.** Estos chips permiten realizar las operaciones matemáticas que subyacen en el entrenamiento de redes de manera mucho más eficiente que las CPU o GPU tradicionales [8]. Un aspecto clave para conseguir una elevada eficiencia consiste en minimizar el movimiento de datos. Una de las formas más efectivas de conseguirlo es reutilizando cada dato leído de memoria en muchas operaciones consecutivas.

- Además, existen esfuerzos en **optimización algorítmica,** como el uso de operaciones matriciales más eficientes, enfoques distribuidos para el entrenamiento, o un uso eficiente de la memoria.

- Asimismo, se pueden plantear estrategias de búsqueda de hiperparámetros y configuración óptima de la arquitectura de la red. Se ha observado que una configuración incorrecta de los hiperparámetros de entrenamiento puede llegar a consumir hasta 5 veces más energía que la configuración óptima [39]. Por tanto, ajustar los hiperparámetros correctos puede generar mayor eficiencia en el entrenamiento y uso del modelo. Asimismo, la aplicación de estrategias ***neural architecture search*** (NAS) para hacer

más eficiente la arquitectura de la red, así como mejorar su eficiencia energética [40].

- **Inferencia en dispositivos de borde *(Edge AI)*.** Para aplicaciones que requieren procesamiento en tiempo real y sin acceso constante a grandes centros de datos, se ha vuelto esencial realizar la **inferencia** (predicción) en dispositivos locales como teléfonos móviles o dispositivos IoT. Estos dispositivos suelen tener capacidades computacionales y de memoria limitadas, por lo que se utilizan versiones comprimidas de los modelos (como en el caso de redes ligeras como **MobileNet** y **EfficientNet**) para garantizar que el proceso sea rápido y eficiente sin sacrificar demasiado la precisión [41].

### *Proyectos y evoluciones futuras*

- La **optimización de recursos computacionales** es uno de los principales proyectos. Se están desarrollando nuevos algoritmos y hardware especializado para mejorar la eficiencia energética y computacional de los grandes modelos. Métodos como la ***cuantización*** y ***pruning*** reducen el número de parámetros y aceleran el procesamiento, al eliminar neuronas y conexiones innecesarias sin sacrificar la precisión [37].

- Uno de los avances más prometedores para mejorar la escalabilidad es el ***federated learning.*** Consiste en entrenar los modelos localmente en dispositivos individuales y, posteriormente, combinarlos en uno más grande [38].

- Otro enfoque muy prometedor consiste en redefinir las redes neuronales de modo que se consigue un comportamiento no lineal a partir de un modelo lineal, a base de seleccionar un subconjunto diferente del modelo lineal para cada muestra. Gracias a la linealidad del modelo, es posible combinar fácilmente diferentes copias de un modelo entrenadas con conjuntos de datos diferentes. Esto permite realizar reentrenamientos incrementales (incluso en *federated learning*), procesando solo las muestras nuevas y sin que se olvide el conocimiento previamente adquirido [42].

## *Interpretabilidad y explicabilidad*

En aplicaciones críticas como la medicina o las finanzas, es esencial que las redes neuronales no solo ofrezcan predicciones precisas, sino que también expliquen cómo llegaron a esas decisiones. La **interpretabilidad** es fundamental para generar confianza en los modelos, especialmente en sistemas que requieren la toma de decisiones sensibles [43].

### *Métodos para entender el comportamiento de redes complejas*

Las redes neuronales son sistemas complejos y su comportamiento se ve como una caja negra. Para comprender cómo una red neuronal procesa la información, los investigadores utilizan **técnicas de visualización** que muestran qué partes de una entrada activan ciertas neuronas [43].

Por otra parte, se están utilizando técnicas como SHAP y LIME para hacer más comprensibles los resultados de las redes neuronales. **SHAP *(shapley additive explanations)*** y **LIME *(local interpretable model-agnostic explanations)*** son métodos que ayudan a desglosar las predicciones complejas en características más simples, mostrando cómo cada característica afecta al resultado final del modelo [44].

### *Aplicaciones en áreas sensibles como medicina o finanzas*

En medicina o finanzas, es crucial que los modelos sean explicables. Los sistemas de diagnóstico basados en redes neuronales, que se usan para identificar enfermedades como el cáncer o evaluar la concesión de créditos, deben ser capaces de justificar sus predicciones [45]. Las redes neuronales de grafos (GNNs) están siendo usadas para identificar interacciones, donde la explicabilidad permite entender mejor las relaciones entre los diferentes nodos de una red [46].

Una alternativa tradicional para conseguir buena interpretabilidad consiste en emplear sistemas fáciles de entender, tales como la regresión lineal, los árboles de decisión o las redes bayesianas. Pero, en general, resultan insuficientes para problemas muy complejos. El sistema antes mencionado [42], por estar basado en un modelo globalmente lineal, también genera un modelo lineal equivalente para cada muestra, lo que implícitamente proporciona una excelente interpretabilidad.

## *Privacidad y Seguridad*

Con el crecimiento del uso de redes neuronales en diversas aplicaciones, uno de los desafíos más importantes es garantizar la **privacidad de los datos** y la **seguridad** de las predicciones generadas por los modelos. El manejo de datos sensibles y la responsabilidad de evitar que los modelos revelen información no deseada son áreas de investigación clave en la actualidad.

### *Amenazas como ataques adversariales*

Uno de los mayores riesgos asociados con modelos avanzados es que podrían generar contenido perjudicial o inseguro. Por ejemplo, si un modelo de lenguaje como GPT está expuesto a datos sensibles o maliciosos, podría generar respuestas indeseadas, como explicar cómo fabricar una bomba nuclear o participar en actividades ilícitas. Para abordar estas amenazas, se implementan diversas opciones:

- **Filtros de contenido y entrenamiento controlado** que evalúan las salidas generadas por la red antes de entregarlas al usuario. Estos filtros están diseñados para detectar y bloquear contenido potencialmente peligroso. Además, durante el proceso de entrenamiento, se utilizan conjuntos de datos curados que excluyen información peligrosa o ilegal, reduciendo la probabilidad de que el modelo aprenda este tipo de contenido [47].

- Los **ataques *adversariales*** manipulan las entradas a los modelos para engañarlos y hacer que generen salidas incorrectas o perjudiciales. Para proteger los modelos contra estos ataques, se emplean técnicas como ***defensive distillation y adversarial training*,** que refuerzan la robustez del modelo ante entradas maliciosas [48]. Estas técnicas son fundamentales para garantizar la seguridad de las aplicaciones de IA en campos críticos como la defensa y la salud.

### *Técnicas de preservación de privacidad: federated learning, differential privacy*

A medida que las redes neuronales se entrenan en conjuntos de datos masivos, es esencial garantizar que no se exponga información sensible o privada, especialmente en campos como la medicina, finanzas o educación. Para abordar este reto, se han desarrollado varias técnicas que permiten preservar la privacidad de los datos sin sacrificar la calidad del modelo:

- ***Federated learning.*** Como se ha indicado, este enfoque distribuye el proceso de entrenamiento a nivel local, lo que se puede realizar incluso en dispositivos individuales (móviles o dispositivos IoT). Los datos nunca abandonan el dispositivo del usuario; en su lugar, se entrena un modelo localmente y luego se comparten únicamente los pesos del modelo con un servidor central, donde se combinan para mejorar el modelo global [38].

- ***Differential privacy.*** Este es otro enfoque clave para la preservación de la privacidad, que asegura que las salidas de un modelo no revelen información específica sobre individuos en los datos de entrenamiento. Se añaden ruidos matemáticos a los datos durante el entrenamiento para garantizar que los resultados no puedan ser rastreados hasta individuos concretos, protegiendo así la privacidad [49]. Esto es crucial para evitar que los modelos expongan datos sensibles de usuarios o pacientes.

- ***Synthetic data.*** Las redes adversarias generativas pueden utilizarse para fabricar datos sintéticos con idéntica distribución a los datos originales, de modo que los datos sintéticos generados pueden ser utilizados para entrenar redes neuronales, sin comprometer la privacidad de los datos originales.

## *Ética*

El desarrollo y el uso de estas técnicas ha de realizarse incorporando una dimensión ética:

- Por una parte, durante la fase de entrenamiento hay aspectos que pueden introducir sesgos en los modelos. Por ejemplo, se han de analizar los datos utilizados para el entrenamiento, para entender si se requiere algún preprocesado para que no haya sesgos (por raza, género, etc.). Cuando se utilizan datos históricos se pueden estar perpetuando patrones no deseables, y esto se puede evitar a través de la modificación de las muestras de entrenamiento, aunque ello puede impactar el poder predictivo del modelo.

- Por otra parte, se ha de garantizar que se realiza un uso responsable y seguro de la IA (por ejemplo, limitando su uso en actividades lúdicas, reduciendo el uso innecesario de los modelos que tengan un excesivo impacto medioambiental, evitar usos maliciosos como la elaboración de *fake news* que parezcan verdaderas, o asegurar que se respetan los códigos y marcos de confidencialidad de las empresas).

# Conclusiones

Las redes neuronales han demostrado ser fundamentales para el desarrollo de soluciones avanzadas en IA, evolucionando constantemente para abordar desafíos cada vez más complejos. A lo largo de este *whitepaper* se han explorado diversas arquitecturas y tendencias clave que marcan el rumbo de esta tecnología.

Las arquitecturas clásicas como el *perceptrón multicapa* (MLP), las redes *convolucionales* (CNN) y las redes recurrentes (RNN) continúan siendo relevantes en múltiples aplicaciones, desde la clasificación de imágenes hasta el análisis de series temporales. Sin embargo, las arquitecturas emergentes como los *transformers* y las redes neuronales de grafos (GNN) han revolucionado campos como el procesamiento de lenguaje natural y el análisis de datos estructurados, respectivamente, ofreciendo nuevas capacidades en términos de eficiencia, escalabilidad y manejo de datos complejos.

A su vez, desafíos como la escalabilidad, la interpretabilidad y la privacidad siguen siendo puntos críticos por resolver. Las técnicas de compresión de modelos, el uso de hardware especializado y el aprendizaje federado surgen como soluciones prometedoras para enfrentar la creciente demanda de recursos computacionales y garantizar la seguridad y privacidad de los datos. Además, la interpretabilidad de las redes neuronales es cada vez más importante en aplicaciones sensibles, donde es crucial comprender cómo y por qué se generan determinadas predicciones.

En resumen, las redes neuronales continúan evolucionando a gran velocidad, con avances significativos en su arquitectura y aplicaciones. El futuro de estas tecnologías está marcado por su capacidad de escalar, mejorar en eficiencia y garantizar la transparencia y seguridad en sus aplicaciones. Esto promete seguir transformando industrias y redefiniendo el futuro de la IA.

# Bibliografía


1. McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics, 5(4), 115-133.*

2. Hebb, D. O. (1949). *The Organization of Behavior.* Wiley.

3. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65(6), 386-408.*

4. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323(6088), 533-536.*

5. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521(7553), 436-444.*

6. Minsky, M., & Papert, S. (1969). Perceptrons: An Introduction to Computational Geometry. MIT Press.

7. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

8. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

9. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

10. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

11. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*.

12. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

13. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

14. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431-3440).

15. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780.

16. Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural Computation, 12*(10), 2451-2471.

17. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

18. Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer.

19. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems* (pp. 3104-3112).

20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems* (pp. 2672-2680).

21. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

22. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training GANs. In *Advances in Neural Information Processing Systems* (pp. 2234-2242).

23. Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein GAN. *arXiv preprint arXiv:1701.07875*.

24. Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4401-4410).

25. Tolosana, R., Vera-Rodriguez, R., Fierrez, J., Morales, A., & Ortega-Garcia, J. (2020). Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion, 64*, 131-148.

26. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing, 321*, 321-331.

27. *Attention is All You Need*, Vaswani et al., 2017.

28. *Generative Pre-trained Transformer 3 (GPT-3)*, Brown et al., 2020.

29. *Text-to-Text Transfer Transformer*, Raffel et al., 2019.

30. Cátedra iDanae (2023). Newsletter 2T23. Large Language Models: una nueva era en la inteligencia artificial.

31. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, Devlin et al., 2018.

32. *Transformers for Biomedical NLP*, Lee et al., 2020.

33. Cátedra iDanae (2023). Newsletter 2T24. Teoría de grafos y redes neuronales en la industria farmacéutica.

34. *Semi-Supervised Classification with Graph Convolutional Networks*, Kipf & Welling, 2017.

35. *Neural Message Passing for Quantum Chemistry*, Gilmer et al., 2017.

36. Strubell, E., Ganesh, A., & McCallum, A. (2019). *Energy and Policy Considerations for Deep Learning in NLP*. ACL.

37. Cátedra iDanae (2024). Newsletter 1T24. Hacia una inteligencia artificial sostenible.

38. McMahan, B., Ramage, D., Talwar, K., & Zhang, L. (2017). *Learning Differentially Private Recurrent Language Models*. ICLR.

39. Geißler, D. et al. (2024). The Power of Training: How Different Neural Network Setups Influence the Energy Demand. https://arxiv.org/pdf/2401.01851.

40. SustainML Project (2023). Carbon footprint based model optimization tool. Deliverable D3.1.

41. Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

42. Duato, J. et al. (2023). GreenLightningAI: An Efficient AI System with Decoupled Structural and Quantitative Knowledge. https://arxiv.org/abs/2312.09971.

43. Selvaraju, R. R., et al. (2017). *Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization*. ICCV.

44. Hamilton, W. L., Ying, Z., & Leskovec, J. (2017). *Inductive Representation Learning on Large Graphs*. NeurIPS.

45. Lundberg, S. M., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*. NIPS.

46. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *"Why should I trust you?": Explaining the Predictions of Any Classifier*. KDD.

47. Radford, A., et al. (2019). Language Models are Unsupervised Multitask Learners. OpenAI.

48. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and Harnessing Adversarial Examples. arXiv preprint arXiv:1412.6572.

49. Dwork, C., & Roth, A. (2014). *The Algorithmic Foundations of Differential Privacy*. TCS.

## *Autores*

Ernestina Menasalvas (UPM)
Manuel Ángel Guzmán (Management Solutions)
Segismundo Jiménez (Management Solutions)
Guillermo Fragio (Management Solutions)
Almudena Balmont (Management Solutions)

Agradecimiento:
José Duato, por su exhaustiva revisión y comentarios.

UNIVERSIDAD POLITÉCNICA DE MADRID

La Universidad Politécnica de Madrid es una Entidad de Derecho Público de carácter multisectorial y pluridisciplinar, que desarrolla actividades de docencia, investigación y desarrollo científico y tecnológico.

**www.upm.es**

Management Solutions es una firma internacional de consultoría, centrada en el asesoramiento de negocio, finanzas, riesgos, organización, tecnología y procesos, que opera en más de 50 países con un equipo de cerca de 4.000 profesionales que trabajan para más de 2.000 clientes en el mundo.

**www.managementsolutions.com**

**Para más información visita**

# blogs.upm.es/catedra-idanae/