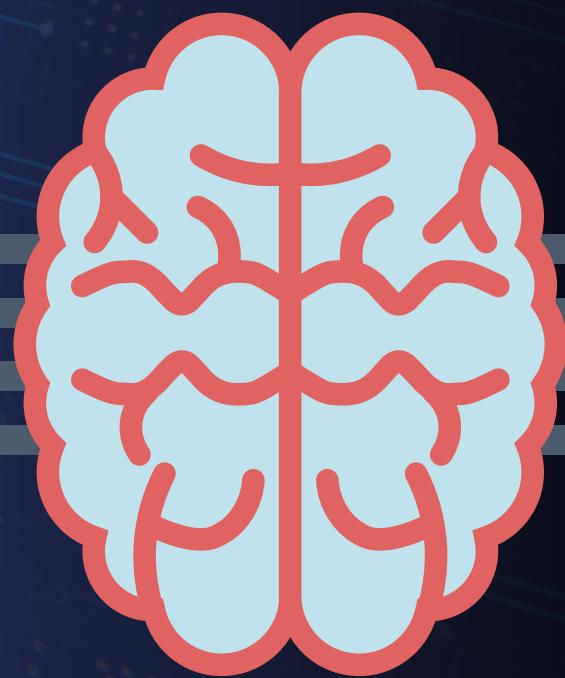


Universidad
Rafael Landívar

Redes Neuronales **Artificiales**

Inteligencia Artificial





Agenda

Sesión 1: Perceptrón

Sesión 2: Perceptrón Multiclasa y Multicapa

Sesión 3: Redes Neuronales Artificiales

- Modelos de Caja Negra
- Deep Learning
- RNA
 - Backpropagation
 - Tipos
 - Arquitecturas





Redes Neuronales: De la Neurona al Perceptrón Multicapa en 9 minutos. El problema X...



Share

REDES NEURONALES



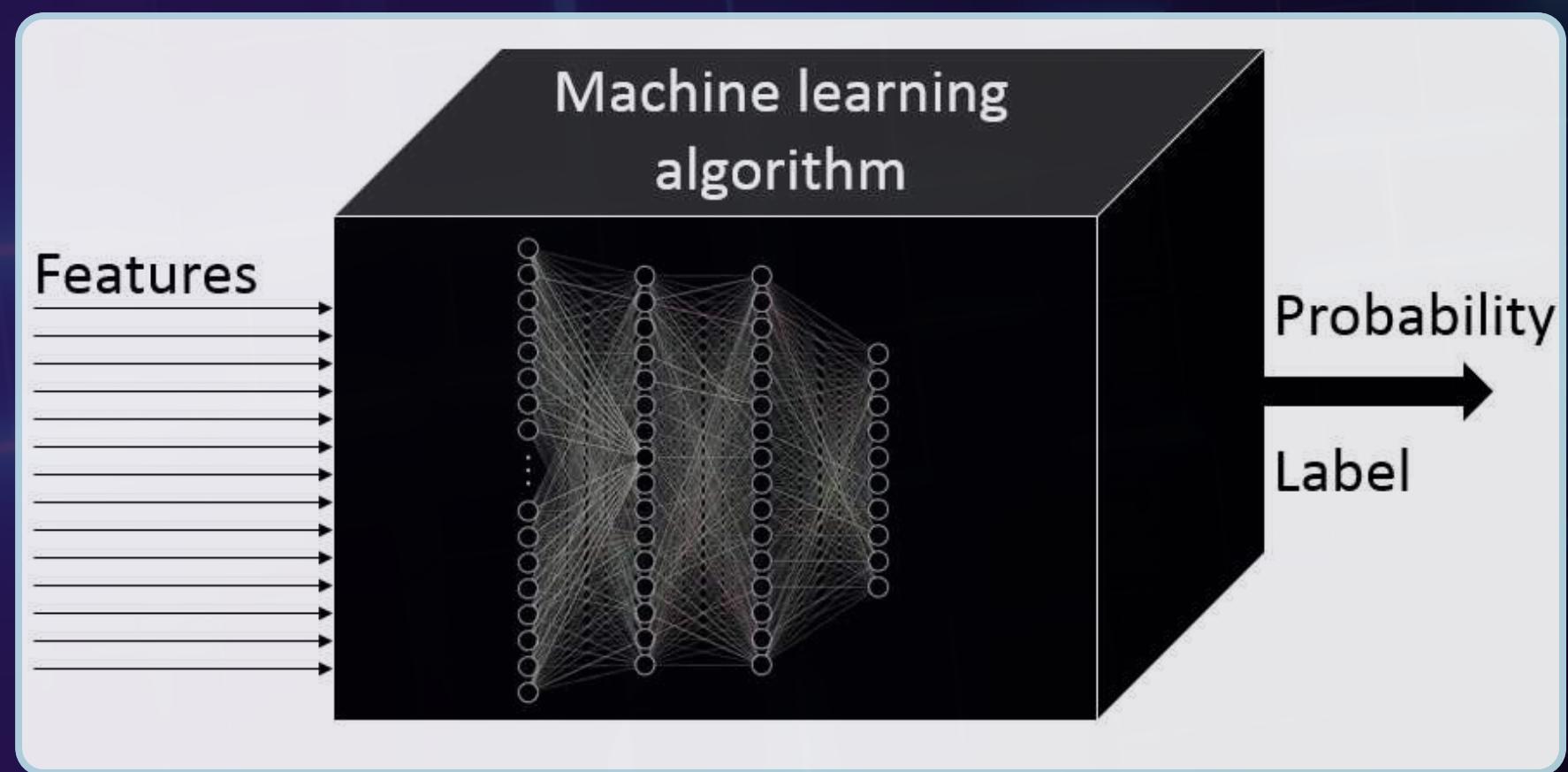
desde CERO

Watch on



Modelos de Caja Negra

Los modelos de caja negra son sistemas computacionales cuyo funcionamiento interno es desconocido o incomprendible para los usuarios, como los utilizados por Netflix o Facebook para recomendaciones

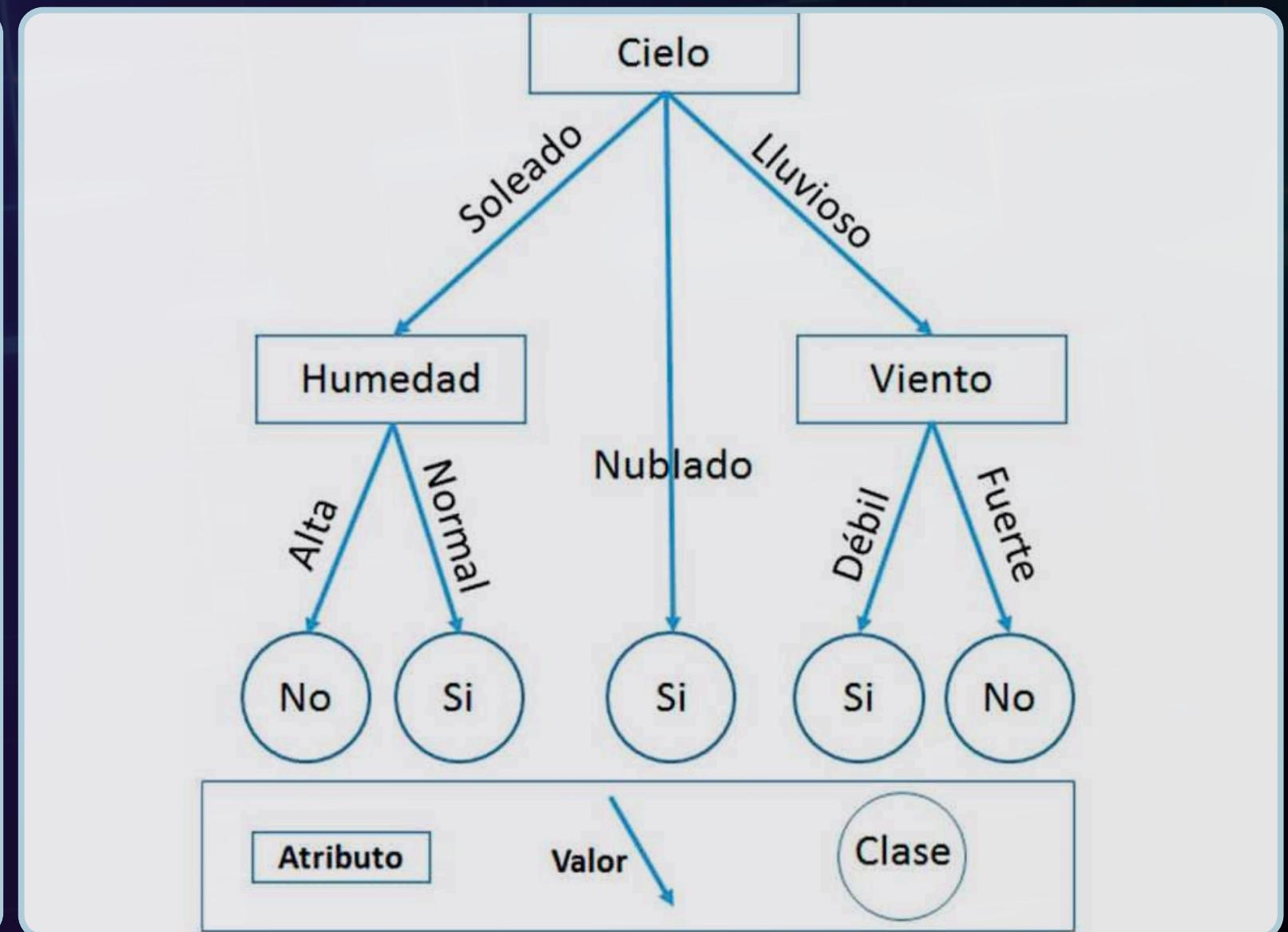
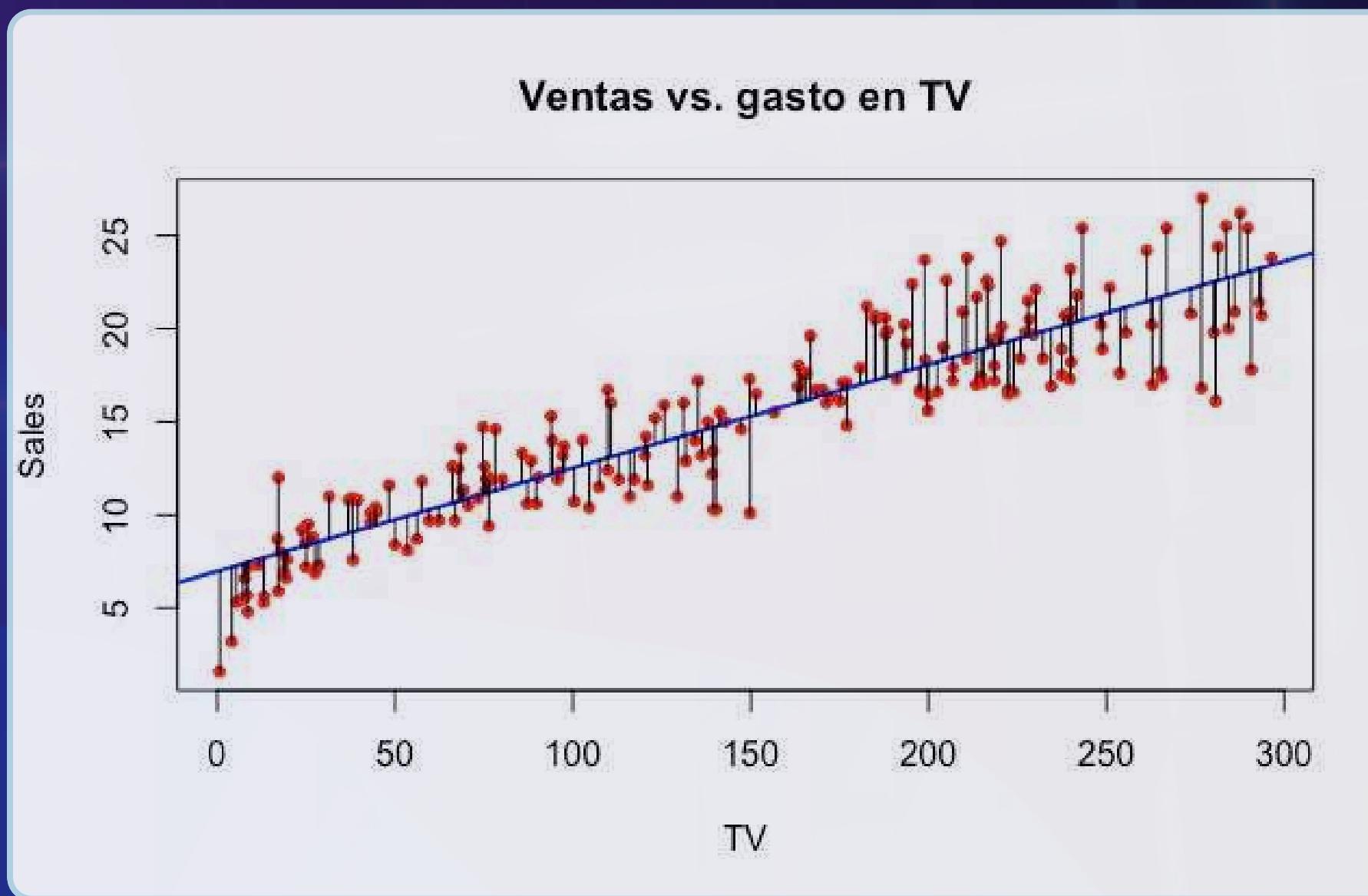


Aunque puede generar predicciones precisas, no es posible interpretar fácilmente cómo llega a sus conclusiones.



Modelos Interpretables

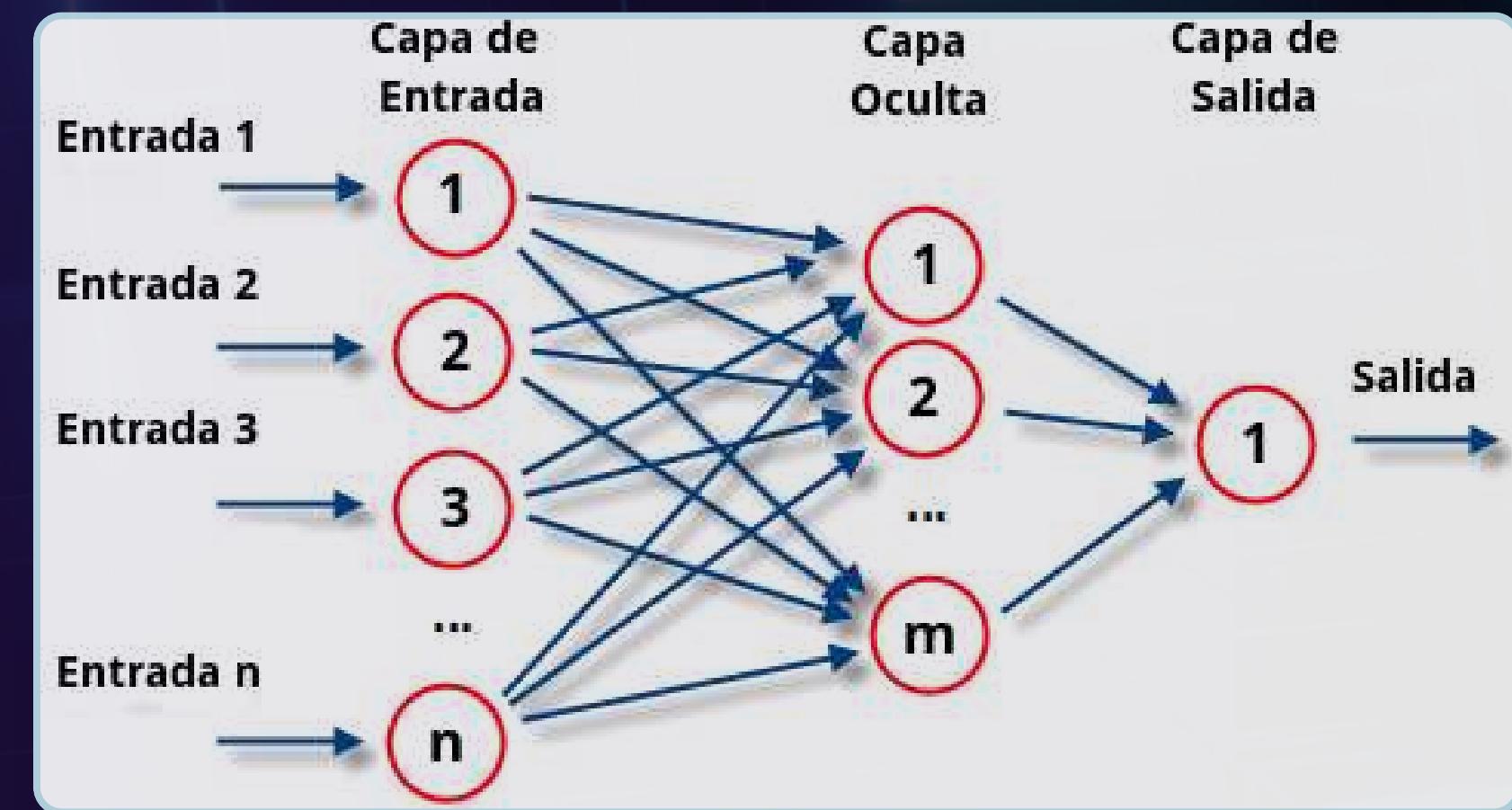
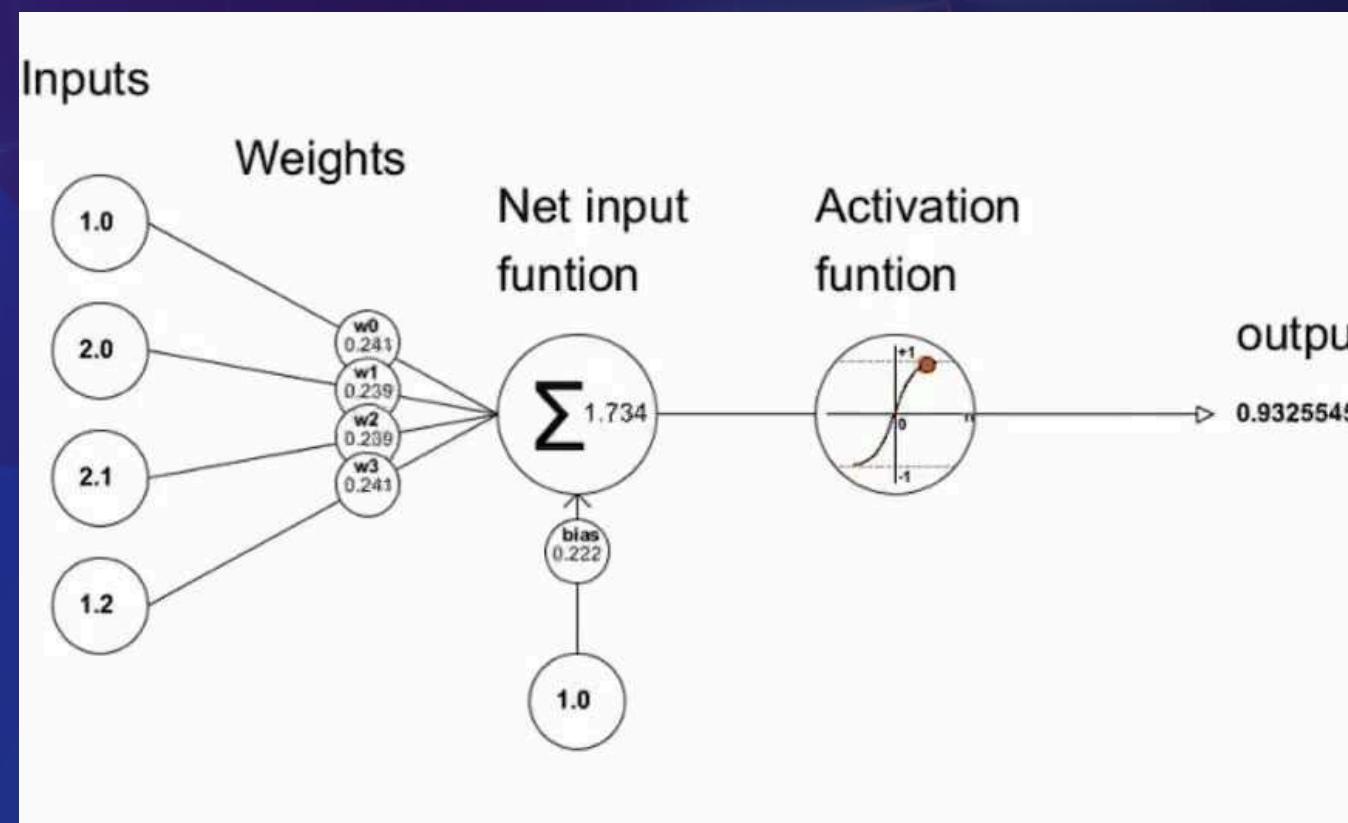
Un modelo interpretable permite a los usuarios entender cómo se toman las decisiones. Ejemplos clásicos son la regresión lineal, la regresión logística y los árboles de decisión, donde cada parámetro tiene una interpretación clara y rastreable.

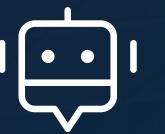


De MLP a redes profundas: ¿Por qué más capas?

El MLP es una red neuronal artificial básica con:

- Una capa de entrada
- Una o más capas ocultas
- Una capa de salida
- Cada capa tiene neuronas que realizan operaciones lineales seguidas de una función de activación (como ReLU, tanh o sigmoid).

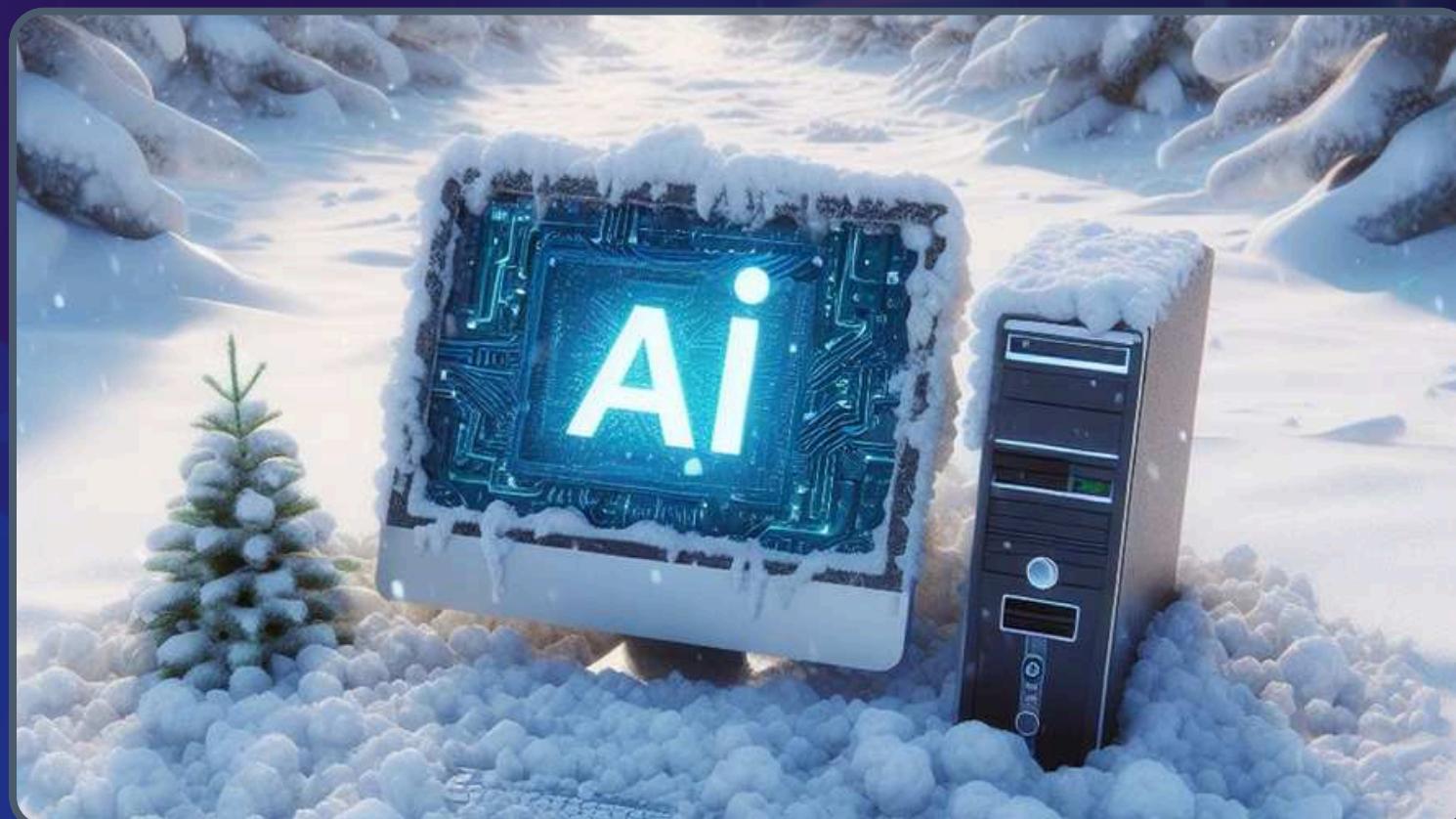




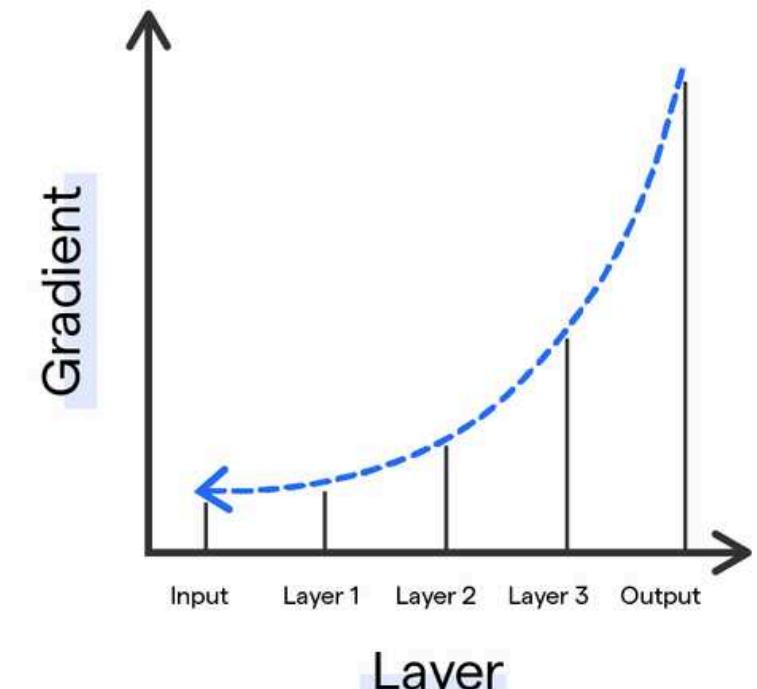
De MLP a redes profundas: ¿Por qué más capas?

Limitaciones del MLP clásico:

- Limitantes descubiertos que generaron el segundo invierno...
- No escala bien a tareas de alta dimensión como imágenes o video.
- Tiene dificultades para representar relaciones jerárquicas o no lineales complejas.
- Necesita una gran cantidad de neuronas si no tiene profundidad.



Vanishing Gradient Problem



Segundo invierno de las redes neuronales

- Período de estancamiento en la investigación y desarrollo
- Finales de los años 80 y principios de los 2000
- Enfrentaron críticas y desinterés debido a varias limitaciones técnicas y conceptuales

¿Qué causó este segundo invierno?

- **Limitaciones del algoritmo de backpropagation:** Algoritmo introducido en los años 70 y popularizado en 1986, presentaba el problema del gradiente desvanecido dificultaba el entrenamiento de redes profundas
- **Falta de poder computacional:** Las computadoras de la época no contaban con la capacidad de procesamiento
- **Competencia de otros métodos:** Durante este período, otros enfoques de aprendizaje automático, como las máquinas de vectores de soporte (SVM) y los métodos estadísticos, demostraron ser más efectivos y eficientes en diversas tareas
- **Escasez de datos**

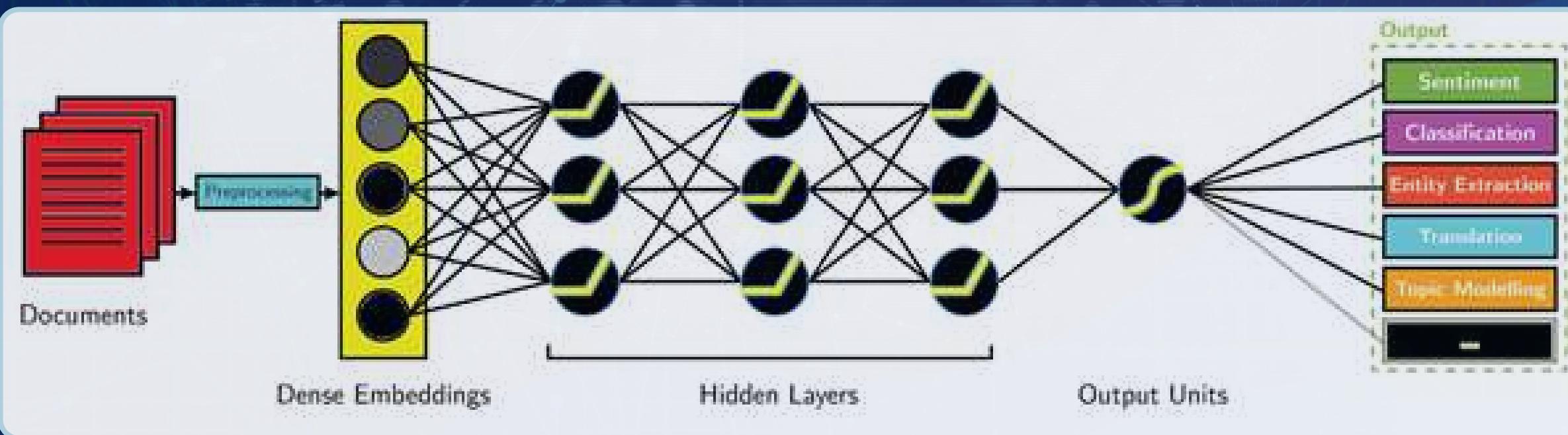


¿Cómo se superó este invierno?

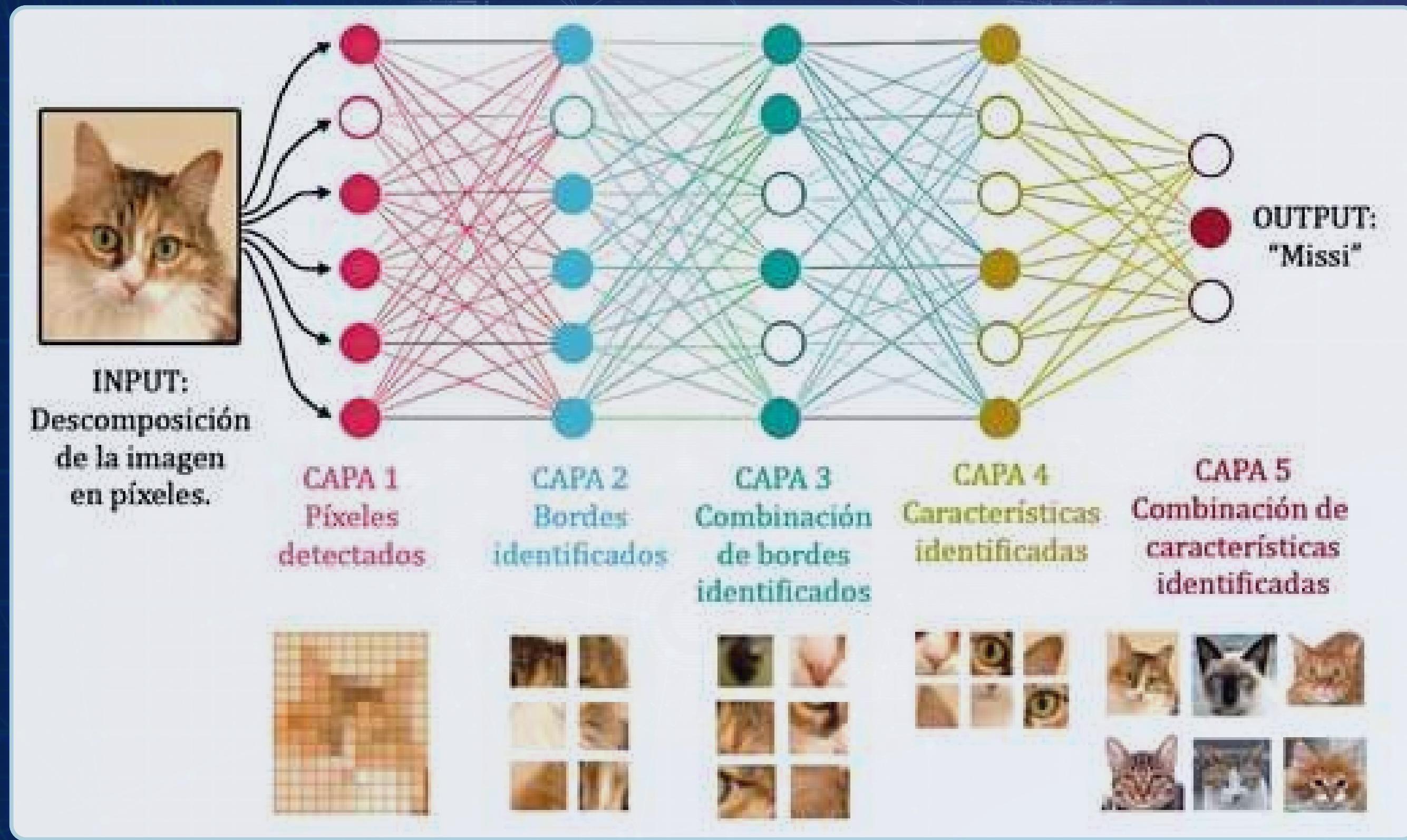
- **Avances en hardware:** El desarrollo de unidades de procesamiento gráfico (GPU)
- **Disponibilidad de grandes conjuntos de datos:** La recopilación y etiquetado de grandes volúmenes de datos
- **Innovaciones en algoritmos:** Se introdujeron nuevas arquitecturas y técnicas, como las redes convolucionales (CNN) y las redes residuales (ResNet), que mitigaron problemas anteriores y mejoraron el rendimiento en tareas específicas.
- **Éxitos prácticos:** Aplicaciones exitosas en reconocimiento de imágenes, procesamiento del lenguaje natural y otros campos demostraron la eficacia de las redes neuronales

De MLP a redes profundas: ¿Por qué más capas?

- Profundidad = más capas ocultas.
- Cada capa aprende una representación más abstracta que la anterior.
- Por ejemplo, las capas jerárquicas en PLN:
 - Capas iniciales – Nivel léxico (Palabras individuales)
 - Capas intermedias – Nivel sintáctico (Relaciones gramaticales)
 - Capas profundas – Nivel semántico (Significado global)
- Las redes profundas procesan texto en niveles: primero entienden las palabras, luego las relaciones, y finalmente el significado completo.
- Esto permite resolver tareas complejas con una arquitectura más eficiente y modular.

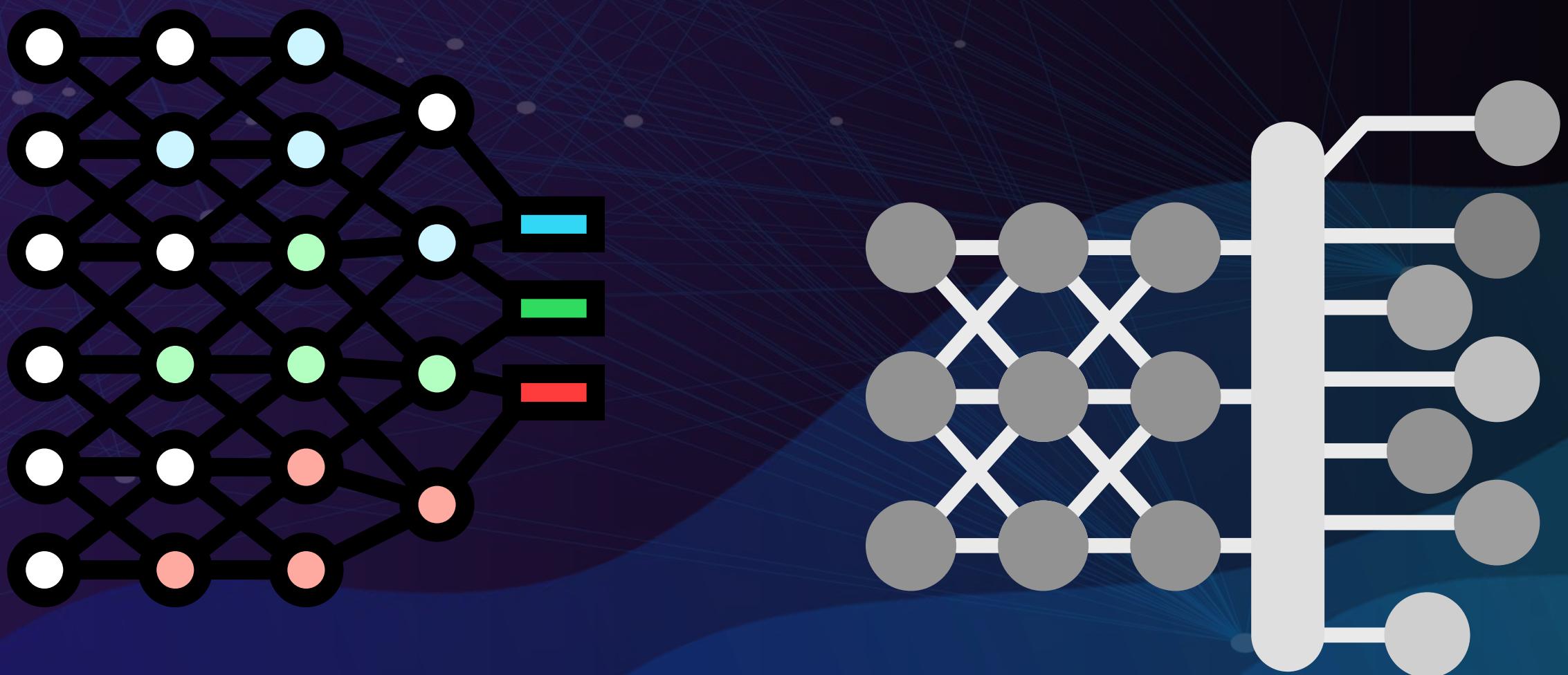


De MLP a redes profundas: ¿Por qué más capas?



Deep Learning

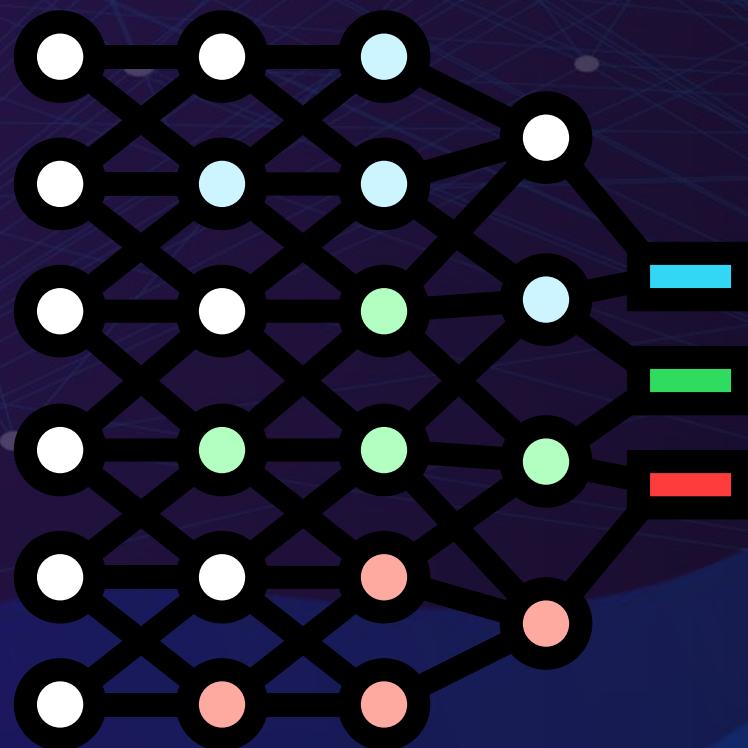
El Deep Learning es una subárea del aprendizaje automático que utiliza redes neuronales con múltiples capas ocultas. Estas redes son capaces de aprender directamente de los datos sin necesidad de diseñar manualmente características o reglas.



Deep Learning

Características principales:

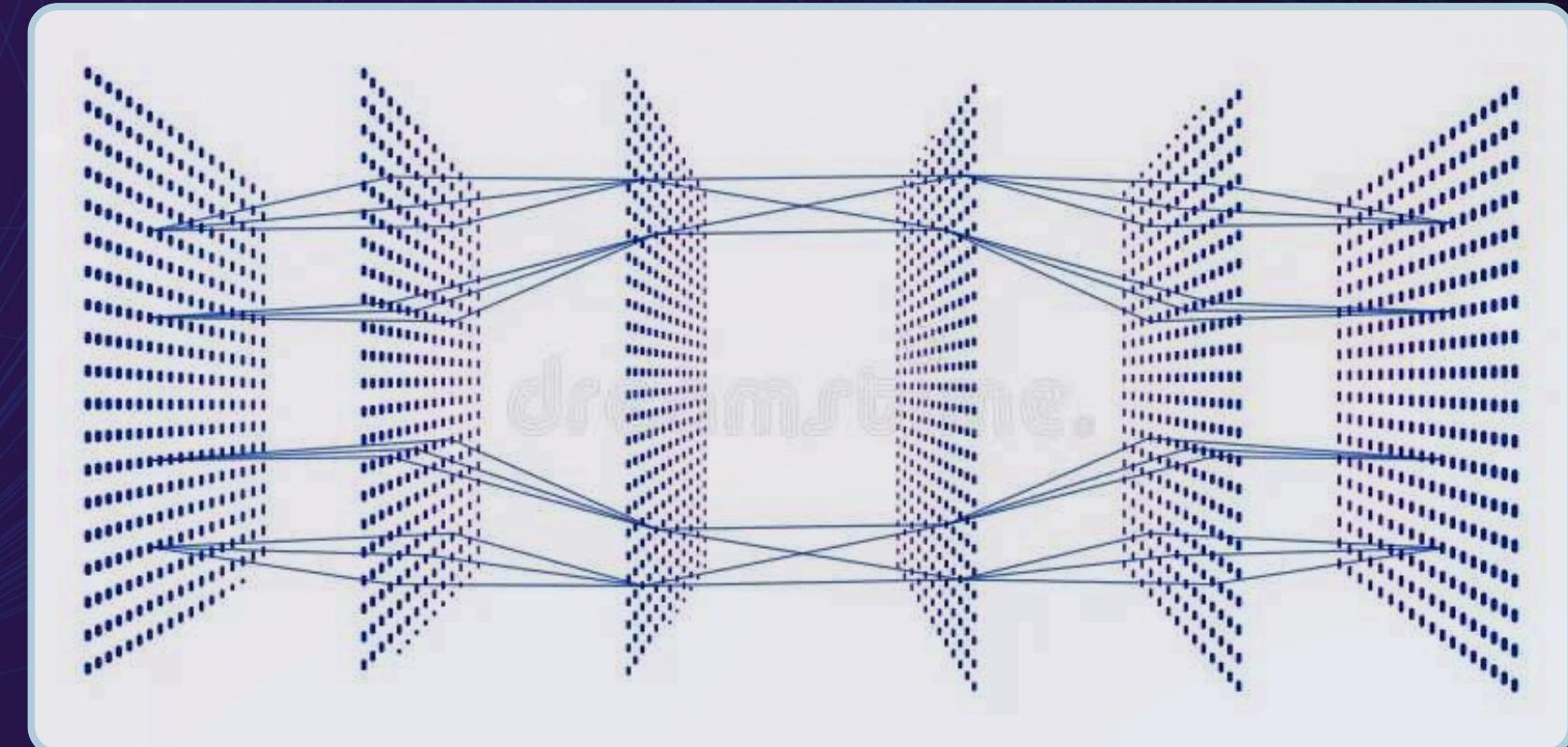
- Composición de múltiples capas no lineales.
- Aprendizaje jerárquico de representaciones: desde lo simple a lo complejo.
- Capacidad de generalización en tareas como clasificación, detección, generación de contenido.
- Requiere una gran cantidad de datos y cómputo (frecuentemente se usan GPUs).



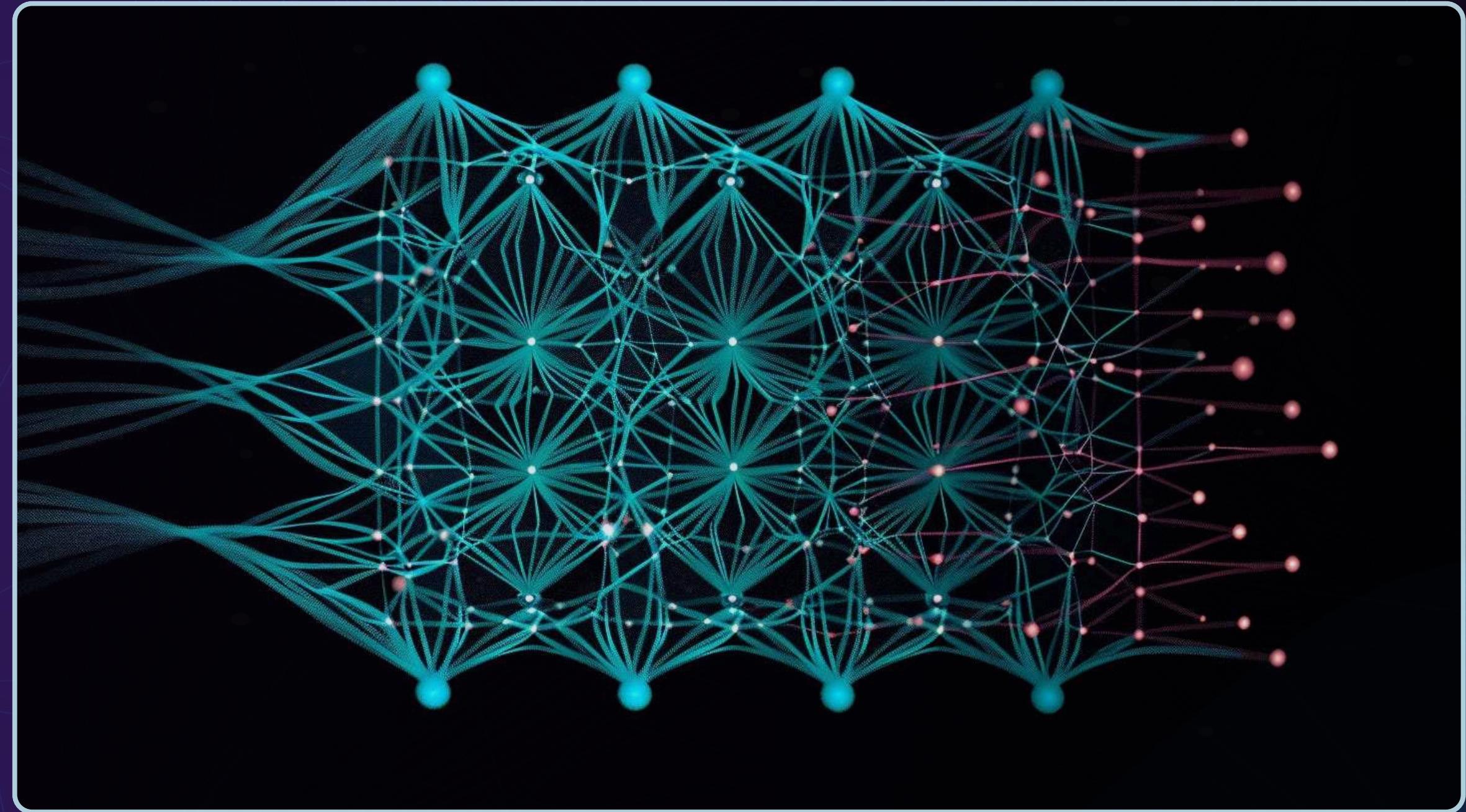
Deep Learning

Las redes neuronales artificiales (RNA) se estructuran en capas de entrada, ocultas y salida, donde cada neurona procesa información mediante funciones matemáticas.

El MLP (Perceptrón Multicapa) clásico muestra limitaciones en tareas de alta dimensionalidad y correlaciones no lineales, debido a su capacidad limitada para abstraer características jerárquicas.



Deep Learning



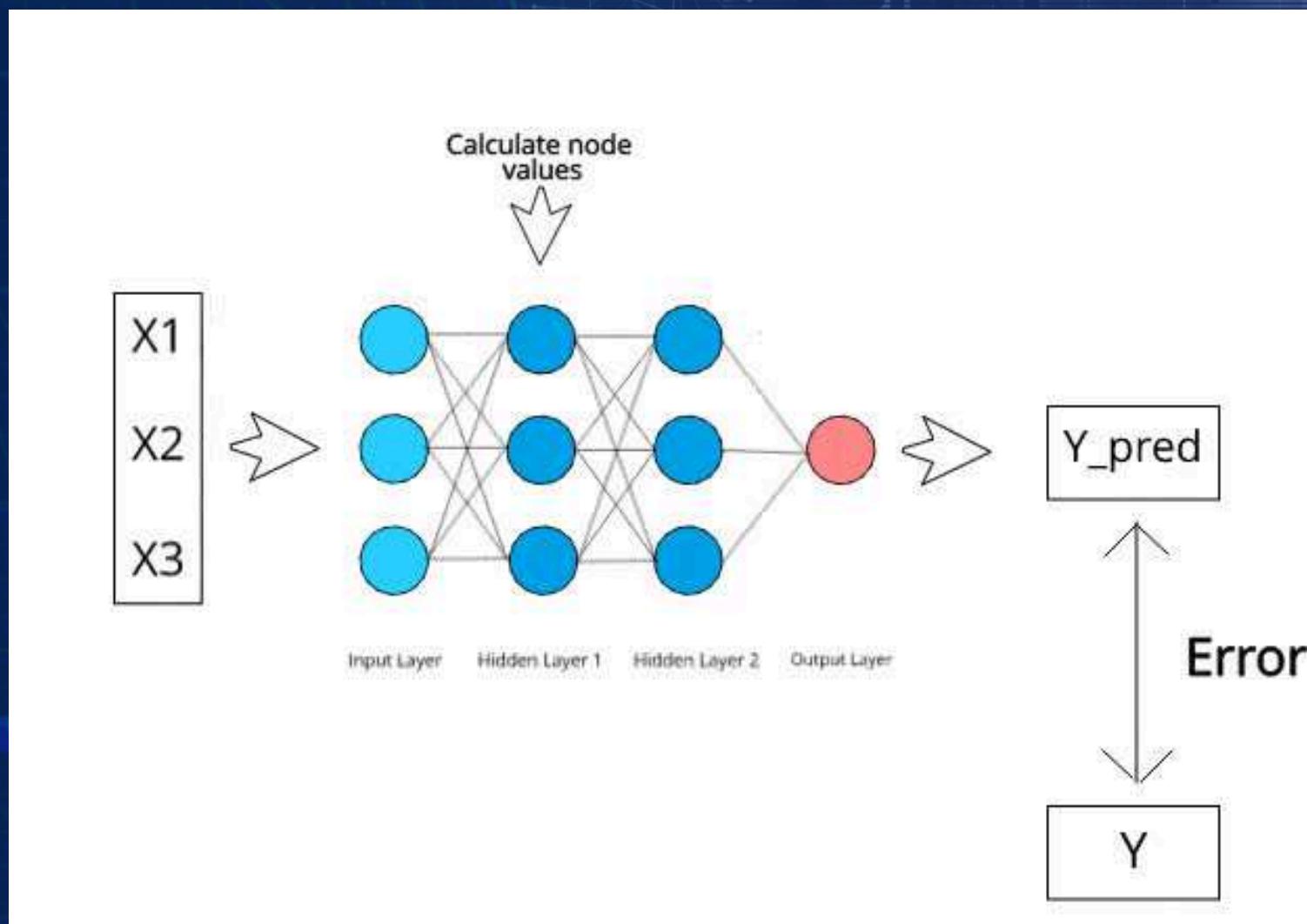
Aumentar la profundidad de la red permite modelar relaciones complejas mediante transformaciones sucesivas.



Ciclo de entrenamiento

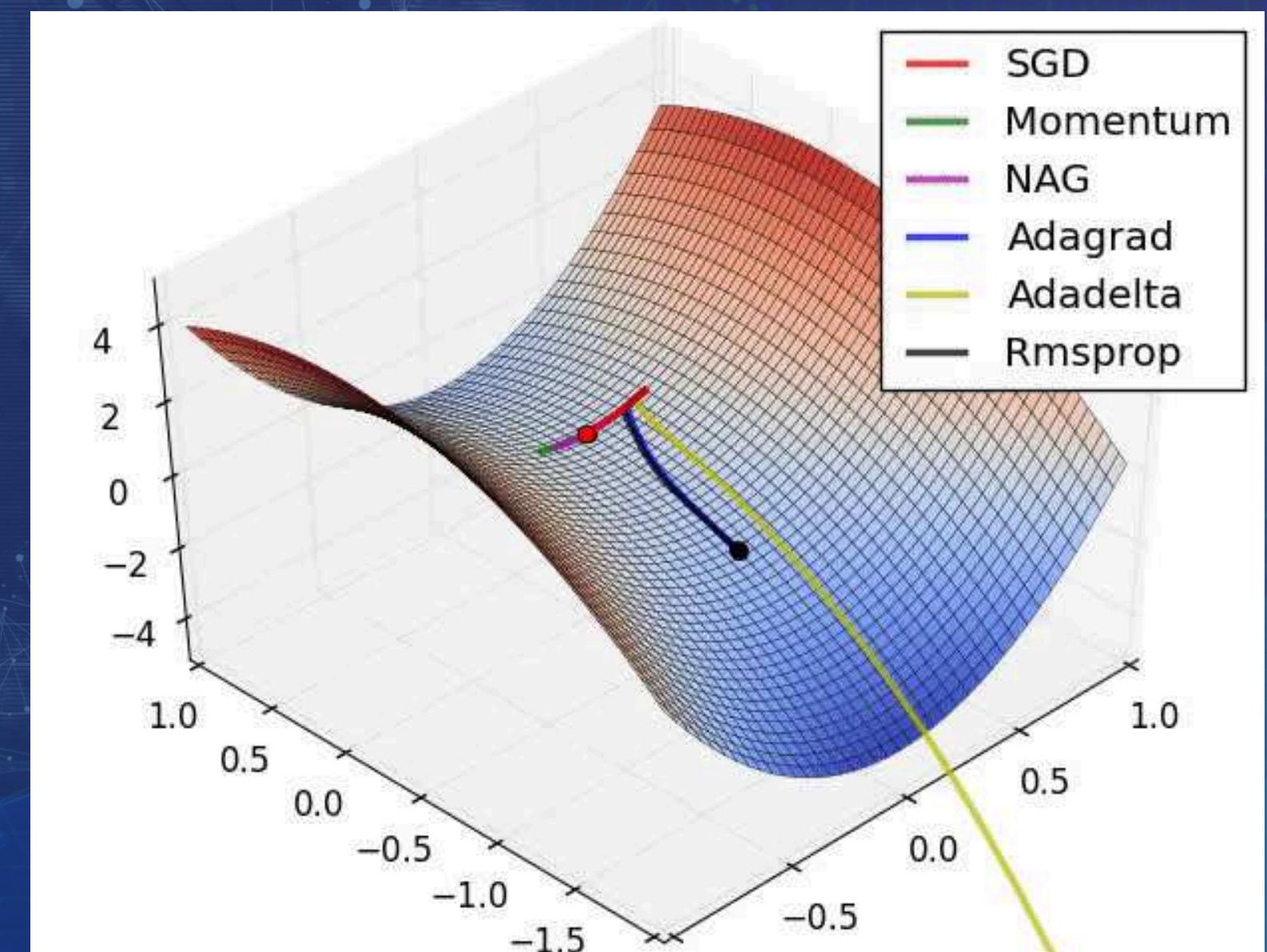
BACKPROPAGATION

El ciclo de entrenamiento es el proceso que permite a una red neuronal aprender a partir de datos etiquetados. Este ciclo se repite una y otra vez durante el entrenamiento, con el objetivo de minimizar el error entre las predicciones de la red y los valores reales.



OPTIMIZACIÓN

Los optimizadores son algoritmos que deciden cómo actualizar los pesos a partir de los gradientes.





¿Qué es una Red Neuronal? Parte 3 : Backpropagation | DotCSV

↗
Share

P A R T E 3 | B A C K P R O P .



Watch on YouTube



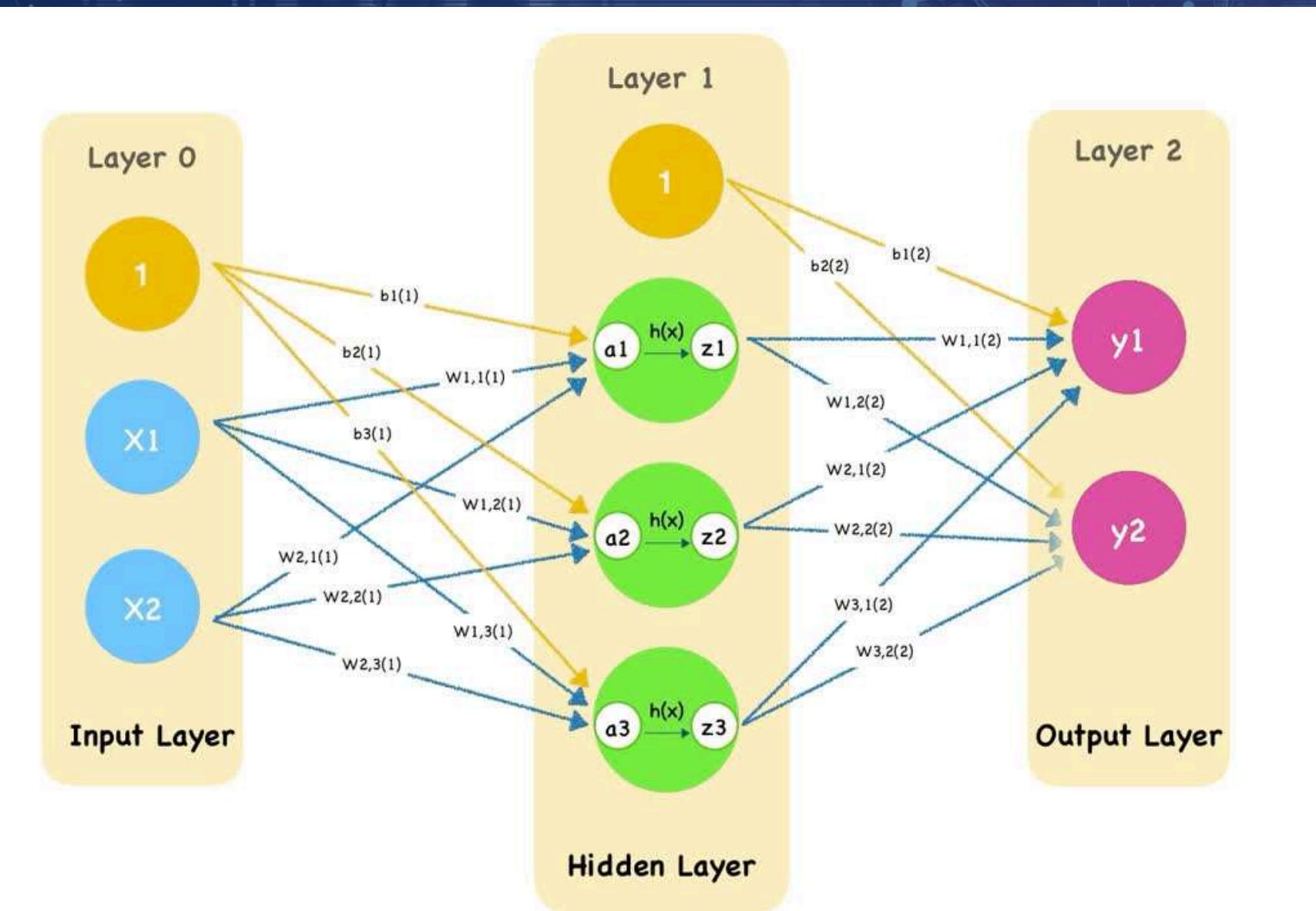
Ciclo de entrenamiento

Paso 1: Hacer una predicción

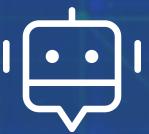
¿Cómo?

(Nombre de la etapa)

FORWARD PROPAGATION



También se conoce de forma coloquial como:
Forward pass o
Feed forward



Ciclo de entrenamiento

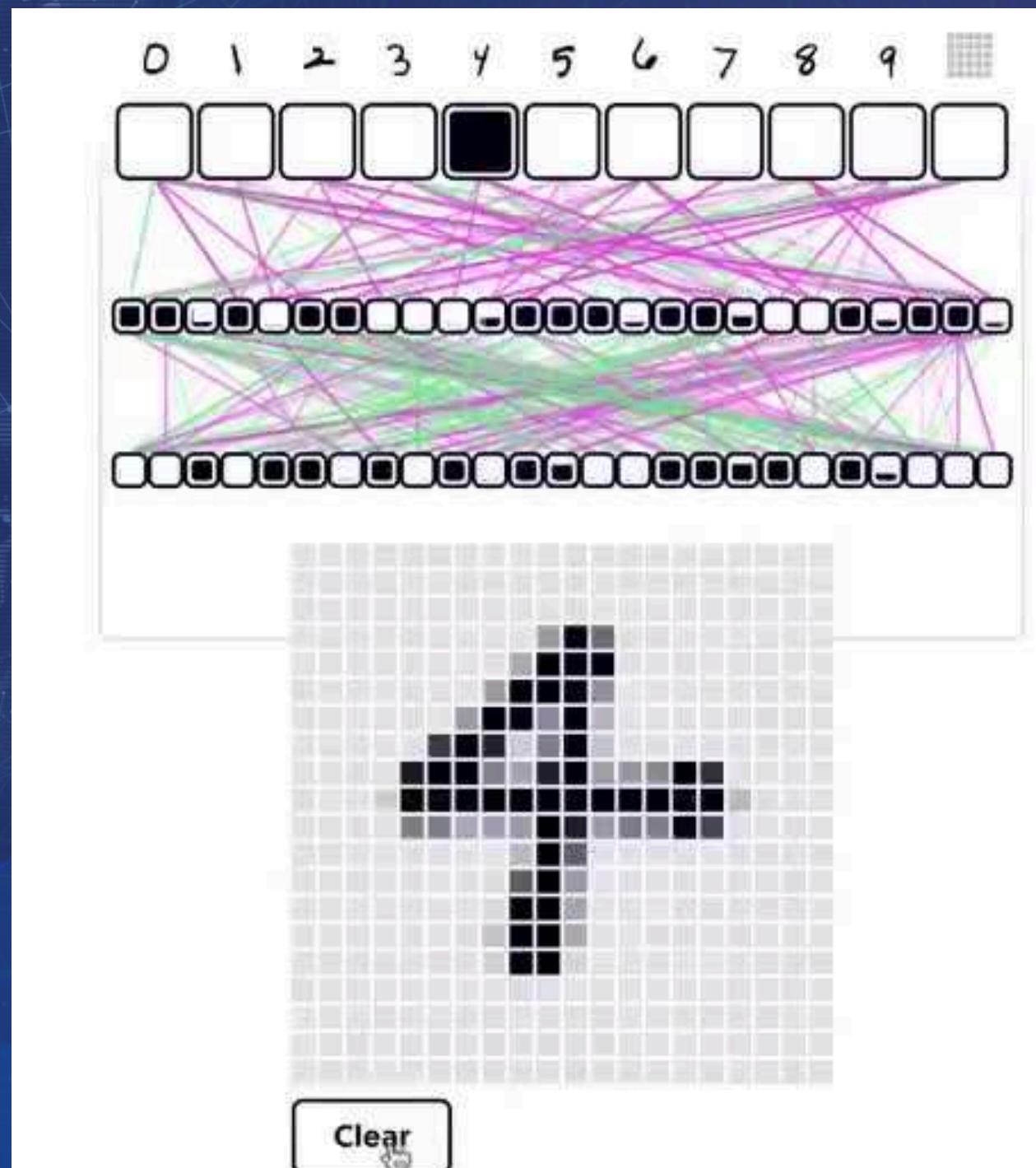
Paso 1: Forward Propagation

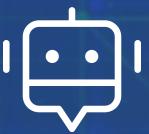
Recorrer los datos desde la entrada hasta la salida de la red neuronal

Ejemplo simple:

Supón que estás entrenando una red para reconocer dígitos escritos a mano (del 0 al 9).

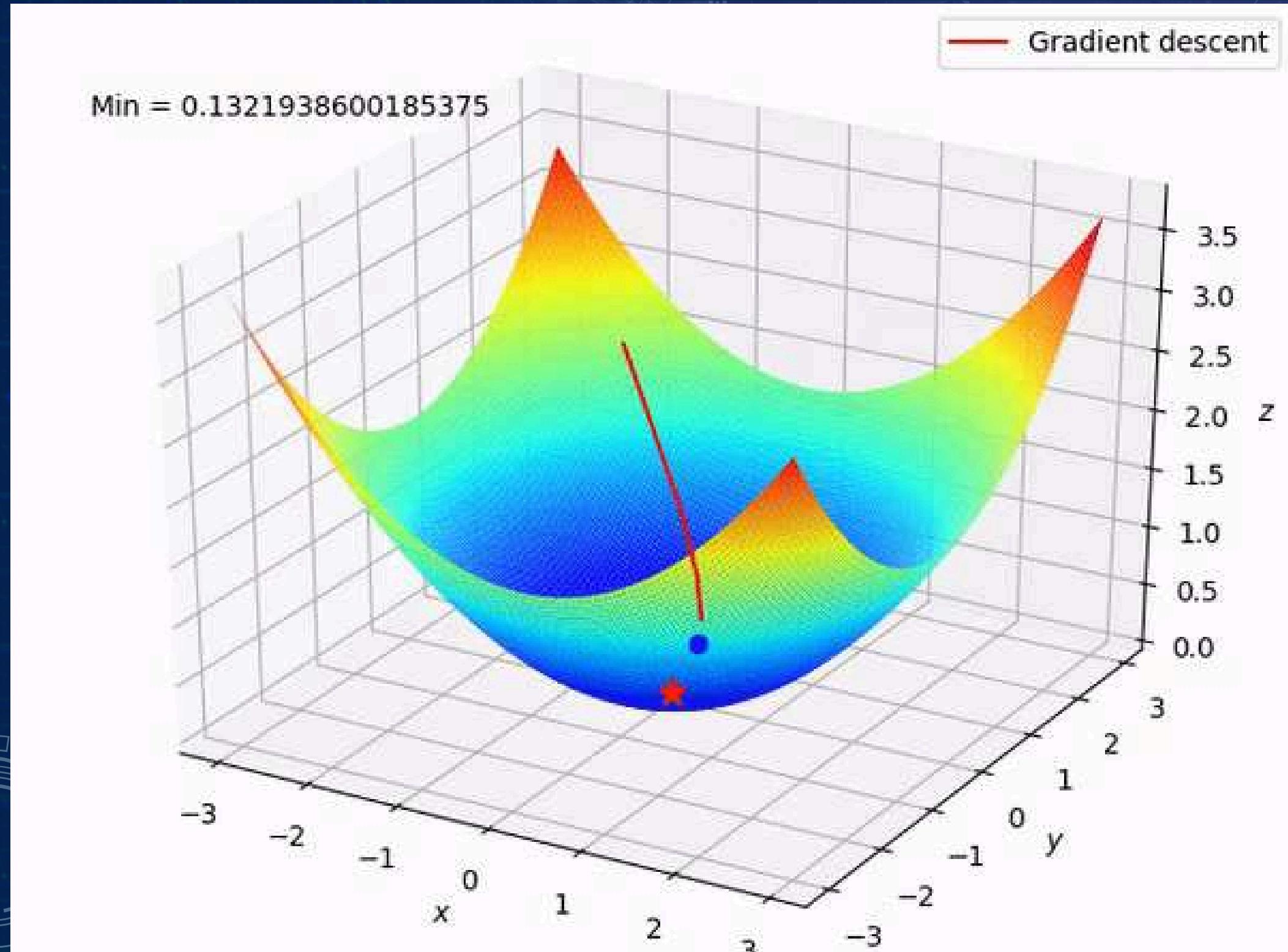
1. Tú le das a la red una imagen
2. La red pasa esa imagen por todas sus capas (aplicando pesos y funciones de activación)
3. Al final, te da una salida como $[0.05, 0.10, 0.8, \dots, 0.01]$ → esto significa que probablemente sea un "2".





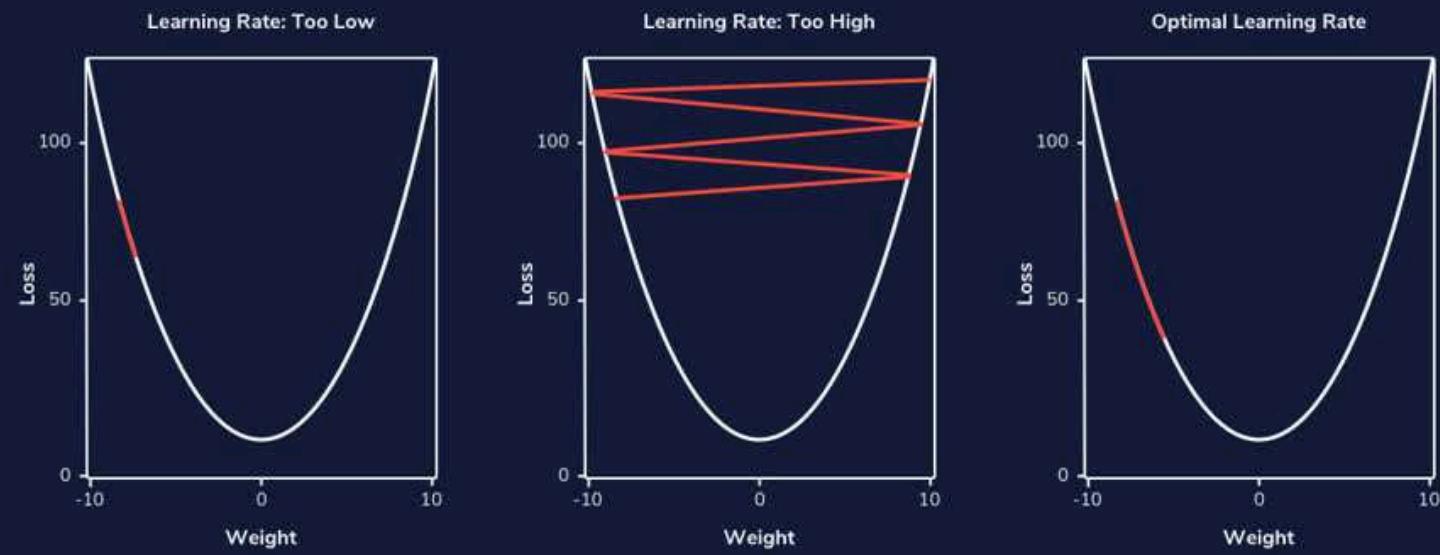
Ciclo de entrenamiento

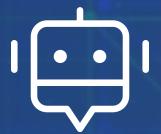
Paso 2: Cálculo del Error



Ejemplo simple:

- Si la red dijo: “Creo que es un 7” pero en realidad es un 5, entonces la función de pérdida nos dice cuánto se equivocó la red.



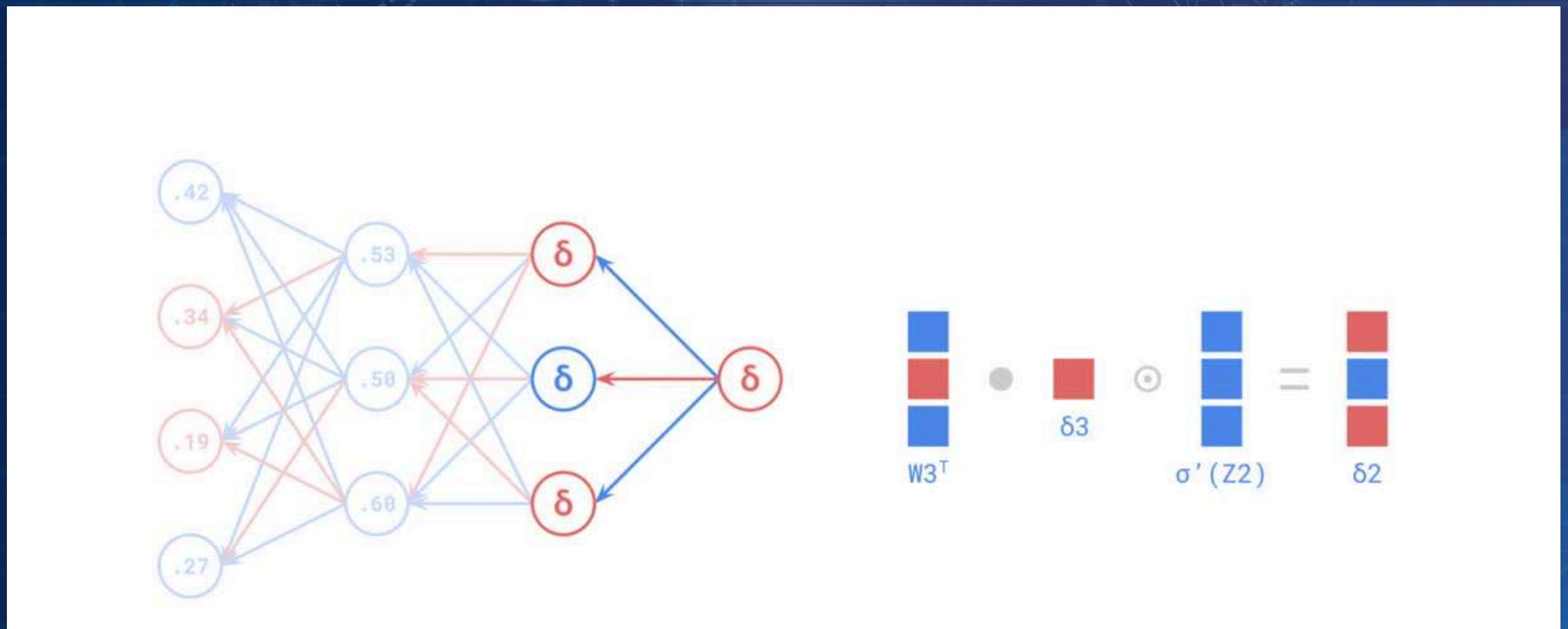


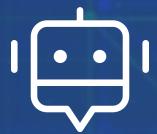
Ciclo de entrenamiento

Paso 3: Aprendiendo de los errores

Backpropagation usa el cálculo diferencial (más específicamente, el gradiente descendente) para calcular cómo afecta cada peso al error total.

1. Calculamos cuánto contribuyó cada peso y sesgo al error.
2. Esto se hace calculando la derivada parcial del loss respecto a cada parámetro .
3. Estas derivadas nos indican hacia dónde debemos mover los pesos para reducir el error.





Ciclo de entrenamiento

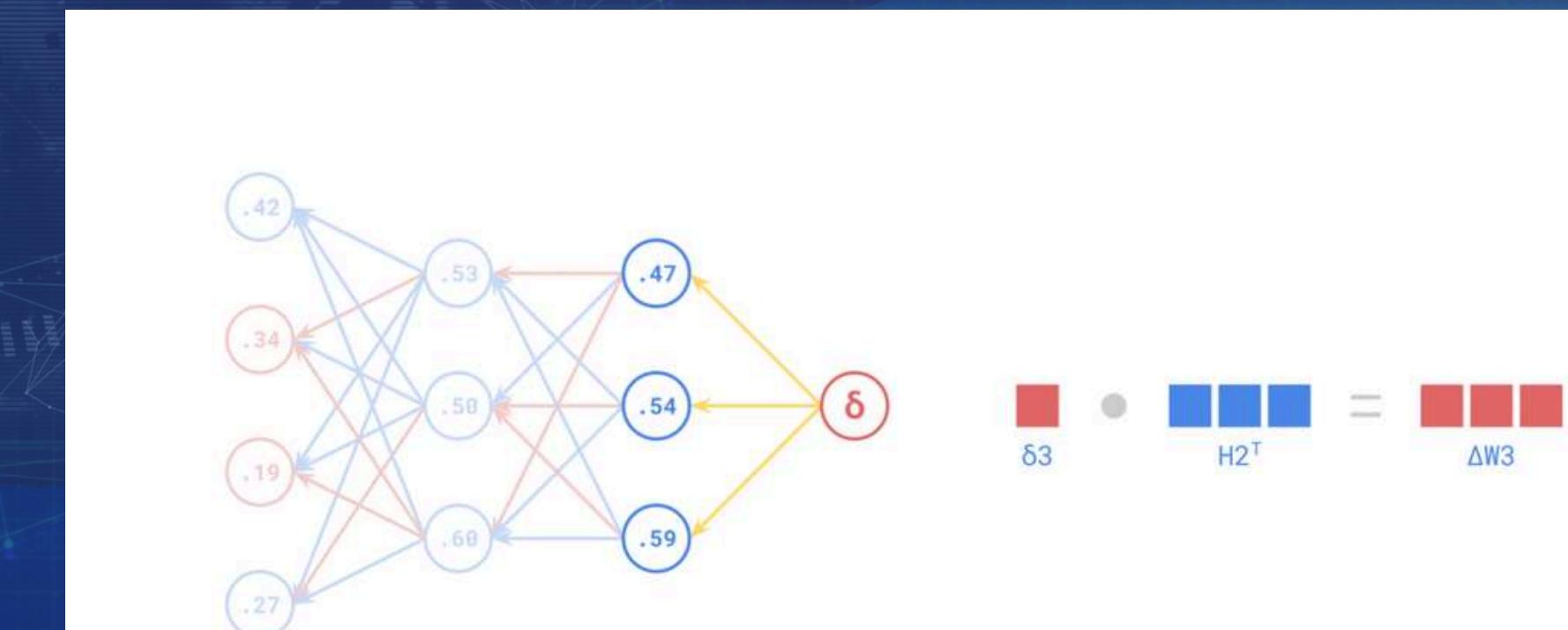
Paso 4: Ajuste de los parámetros

Con los gradientes calculados, ahora actualizamos los pesos y sesgos de la red.

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \eta \cdot \delta_j^{(l)} \cdot a_i^{(l-1)}$$
$$b_j^{(l)} \leftarrow b_j^{(l)} - \eta \cdot \delta_j^{(l)}$$

Donde:

- $w_{ji}^{(l)}$: peso que conecta la neurona i de la capa $l-1$ con la neurona j de la capa l .
- $b_j^{(l)}$: sesgo de la neurona j en la capa l .
- η : tasa de aprendizaje (learning rate).
- $\delta_j^{(l)}$: error delta de la neurona j en la capa l .
- $a_i^{(l-1)}$: salida de la neurona i en la capa $l-1$.

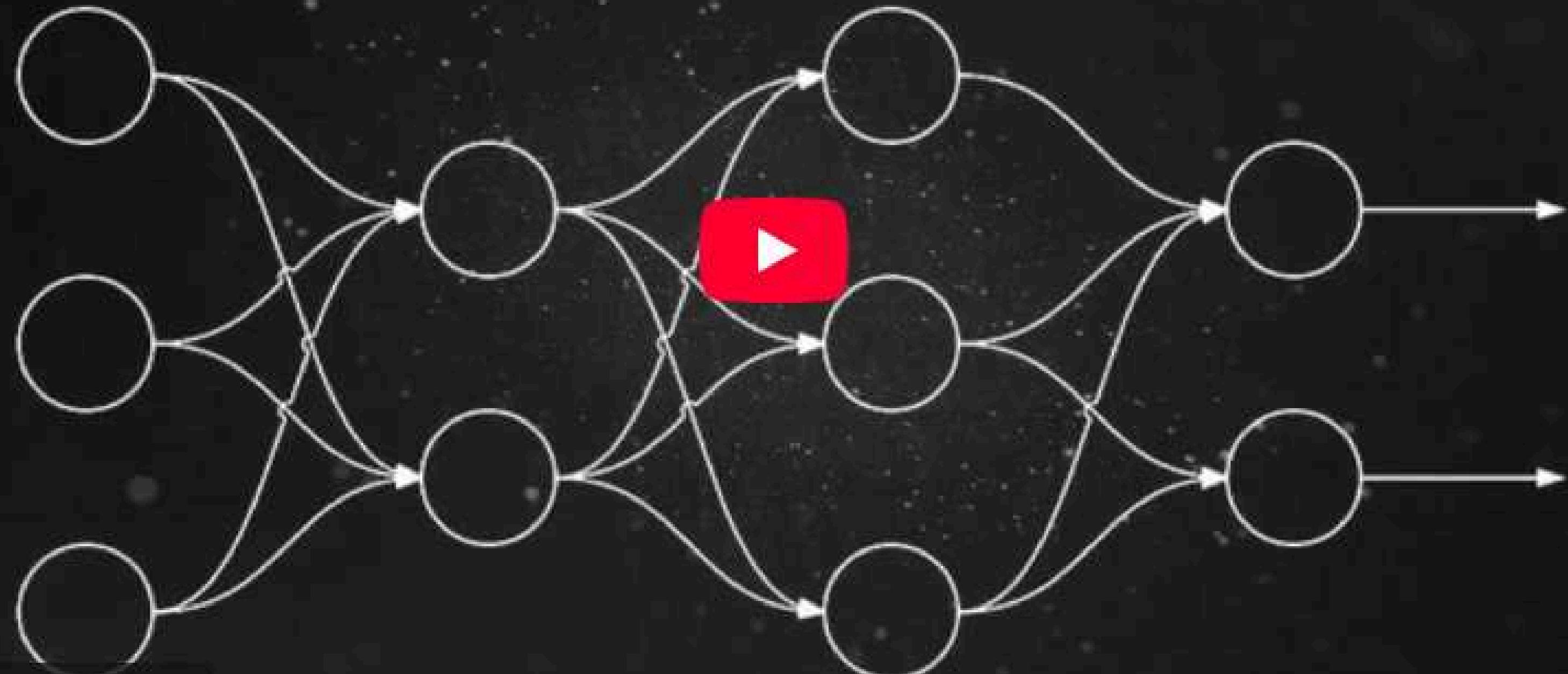




¿Qué es una Red Neuronal? Parte 3.5 : Las Matemáticas de Backpropagation | DotCSV

Share

PARTE 3.5 | FÓRMULAS



Watch on YouTube



Arquitecturas de Deep Learning

- Redes feedforward profundas (DNN)
- Arquitecturas recurrentes
- Arquitecturas convolucionales
- Redes generativas

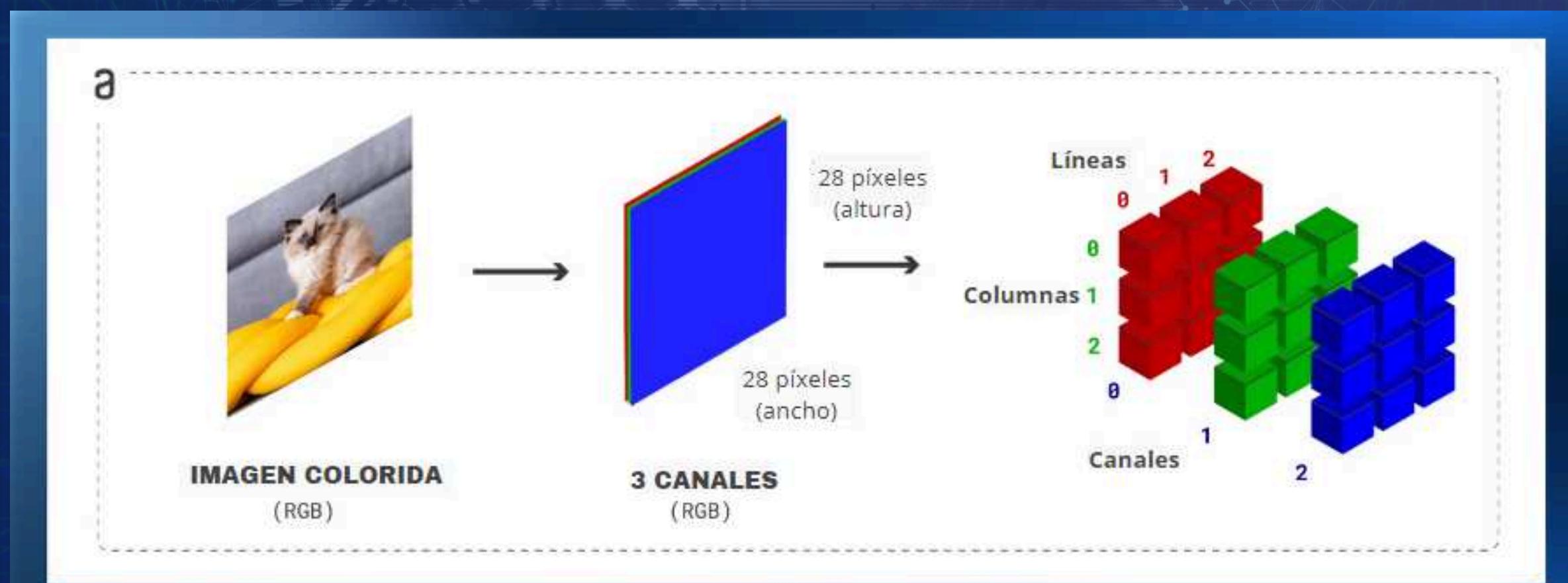


CNN

Las imágenes tienen estructura espacial:

- Anchura
- Altura
- Canales

Para una imagen 28x28 píxeles, una capa totalmente conectada tendría 784 entradas por cada neurona.

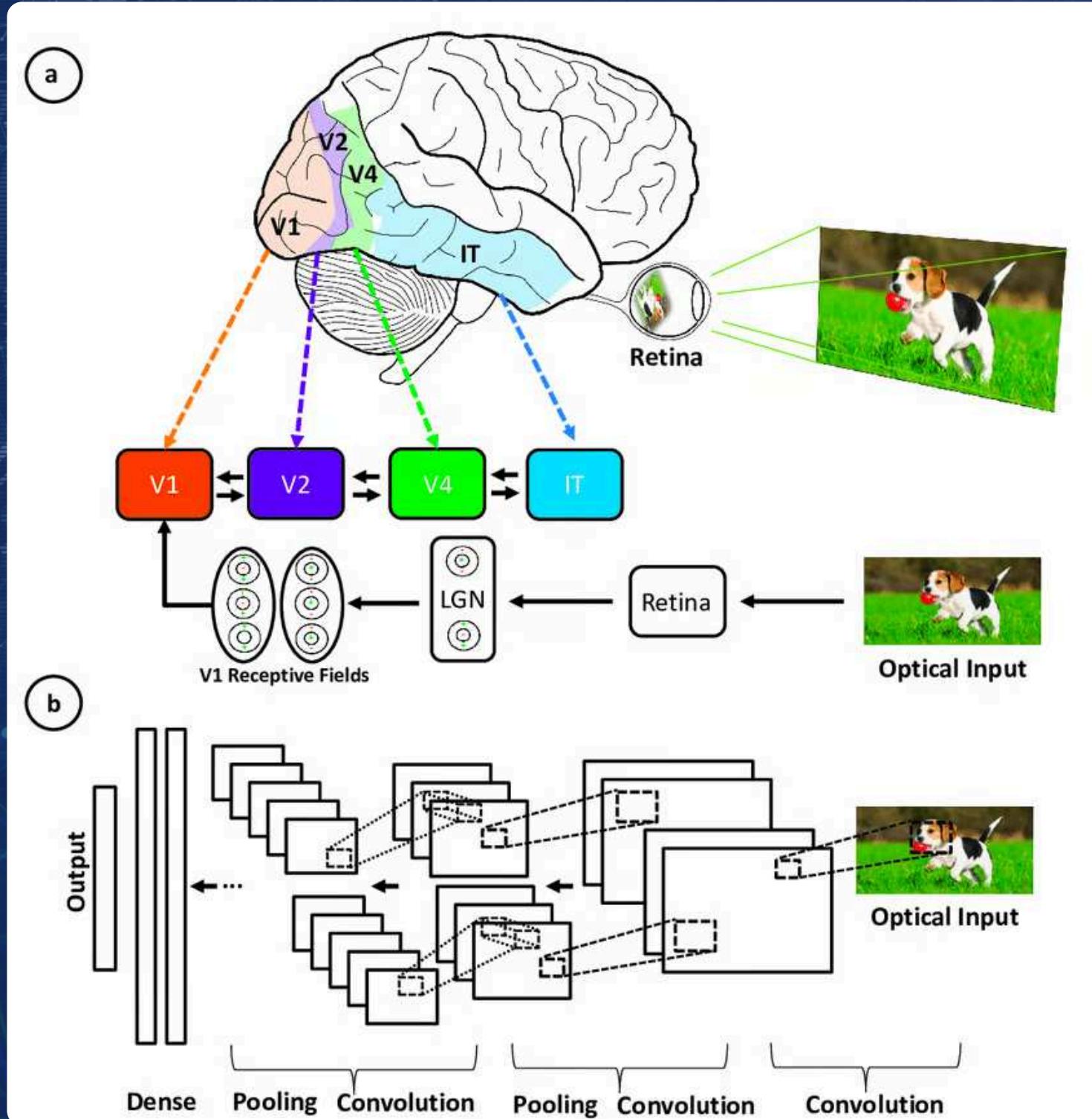




CNN

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son un tipo especializado de red neuronal artificial diseñada para procesar y analizar datos visuales, como imágenes y videos.

Inspiración biológica:
células en la corteza visual
del cerebro que responden
a regiones específicas del
campo visual.



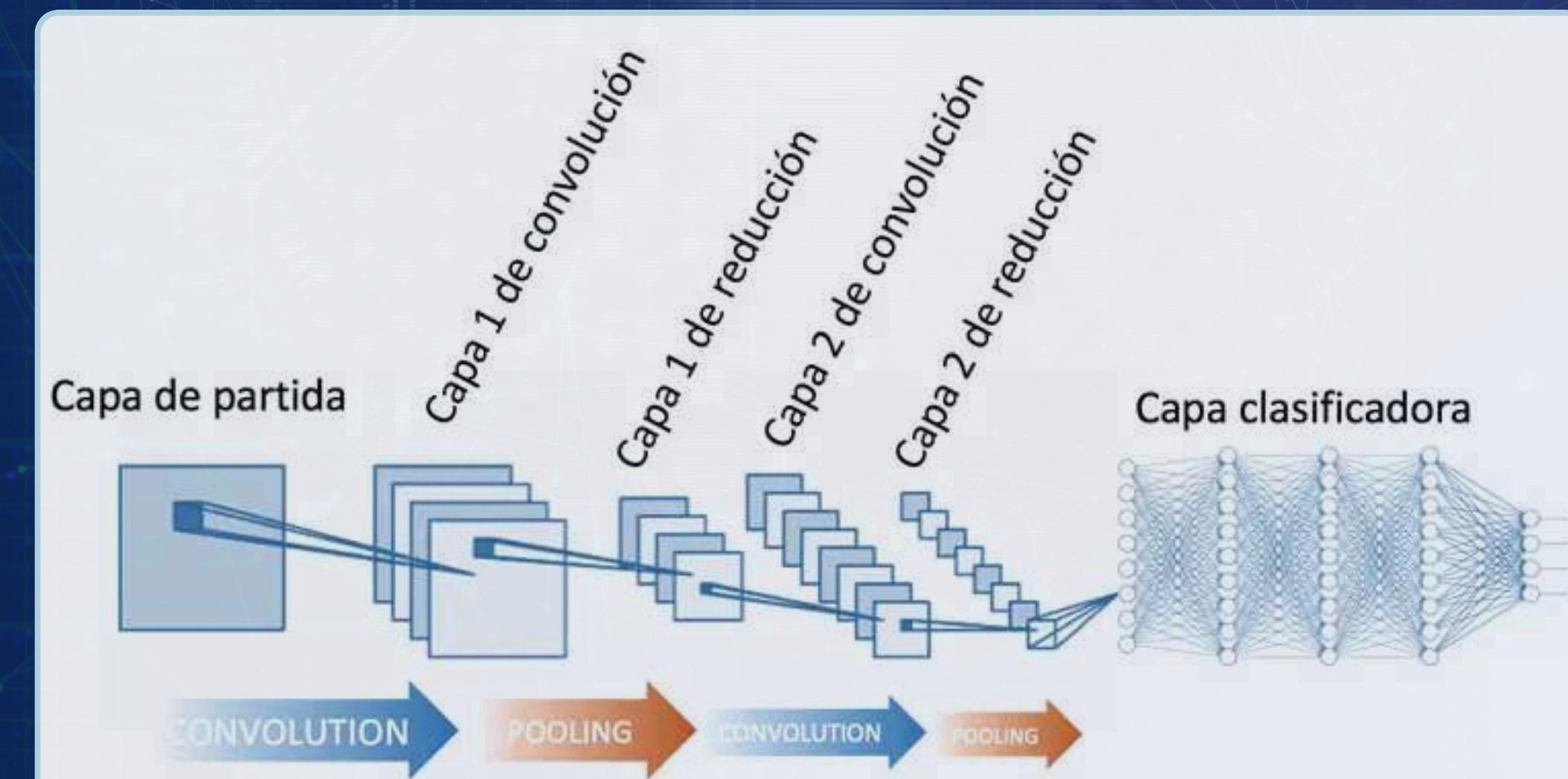
Las CNN procesan imágenes imitando el funcionamiento del ojo humano. Cada neurona de la red trabaja sobre una pequeña región de la imagen (campo receptivo), y a través de varias capas, la red aprende a identificar desde patrones simples (bordes, líneas) hasta patrones complejos (formas, objetos)



CNN

El flujo general es:

- Entrada: Imagen en formato de matriz de píxeles.
- Capas convolucionales: Aplican filtros (kernels) para extraer características locales.
- Capas de activación: Usualmente la función ReLU, que introduce no linealidad.
- Capas de agrupamiento (pooling): Reducen la dimensionalidad y retienen las características más importantes.
- Capas totalmente conectadas: Realizan la clasificación final basándose en las características extraídas

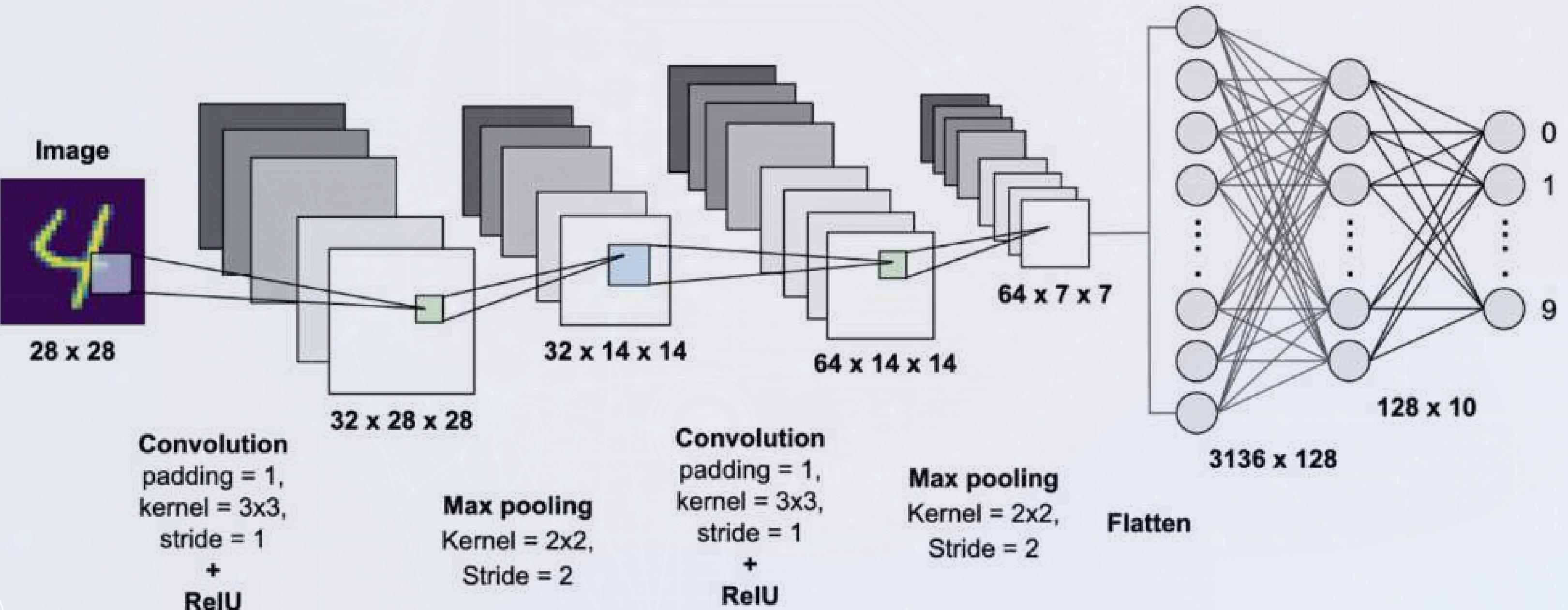


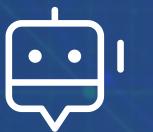


CNN



Keras

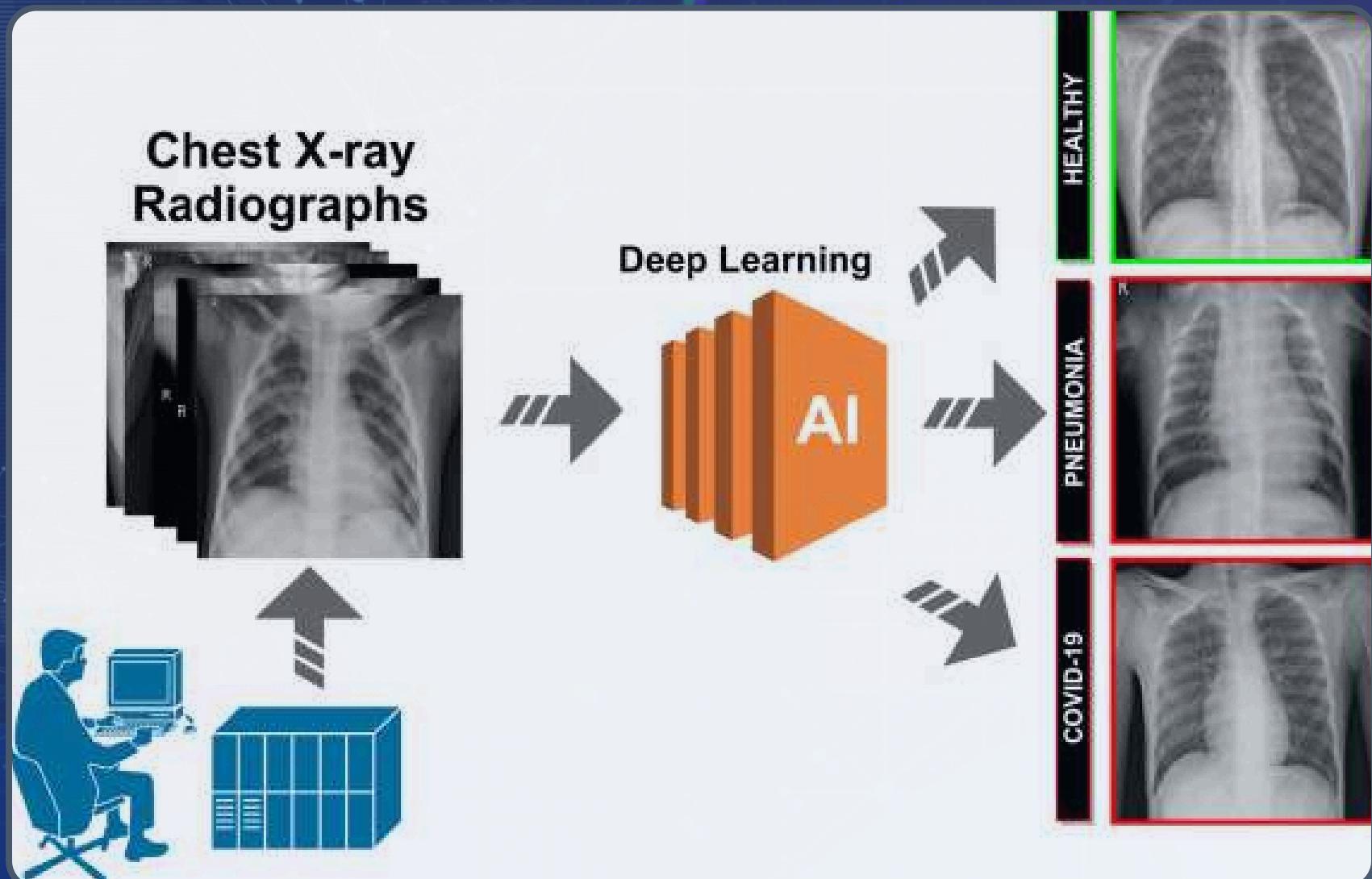
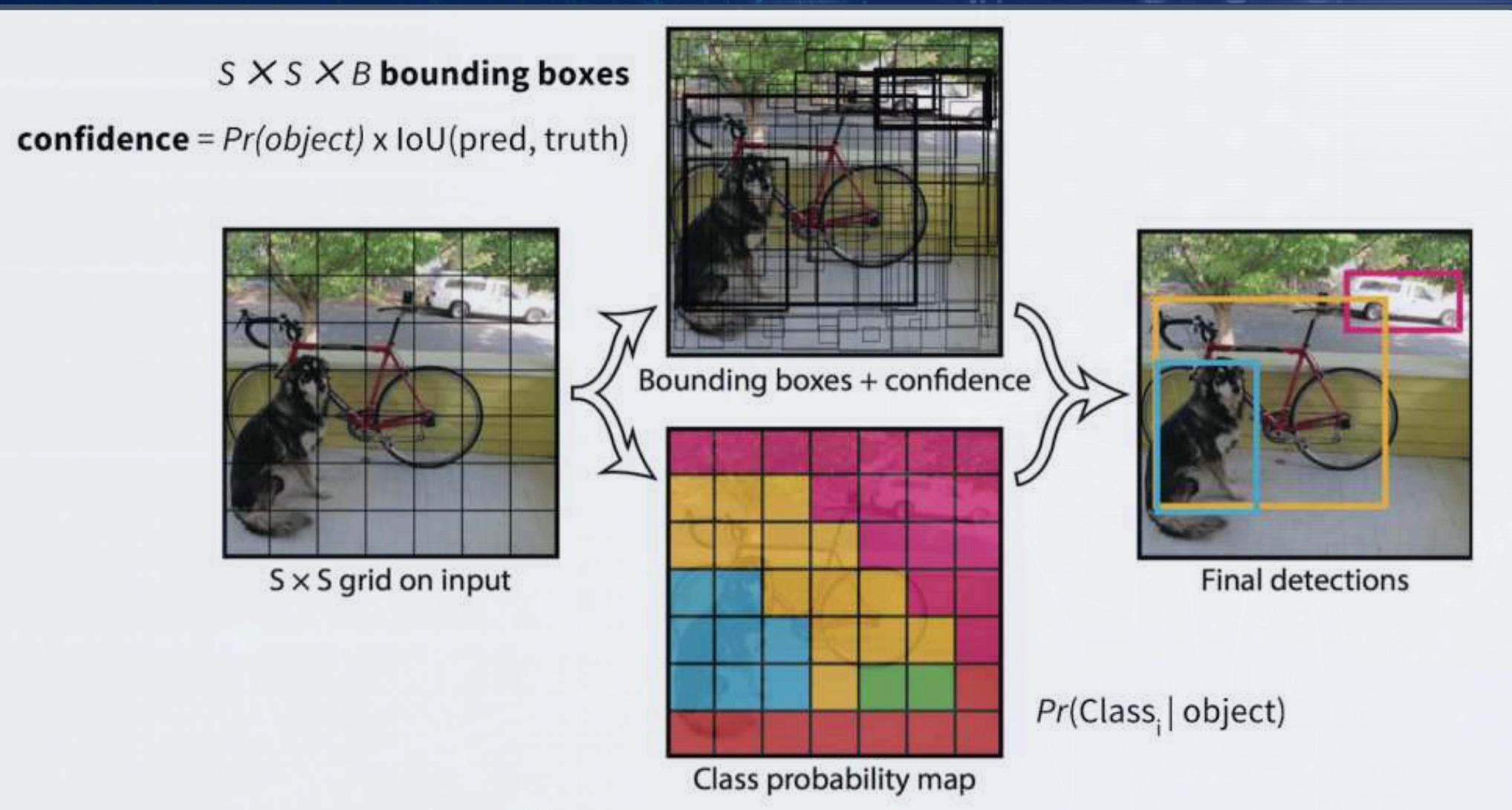




CNN

Aplicaciones:

- Clasificación de imágenes
- Detección de objetos
- Reconocimiento facial
- Reconocimiento de escritura manuscrita
- Diagnóstico médico (detección de tumores, COVID en rayos X)





¡Redes Neuronales CONVOLUCIONALES! ¿Cómo funcionan?

REDES NEURONALES CONVOLUCIONALES

Share



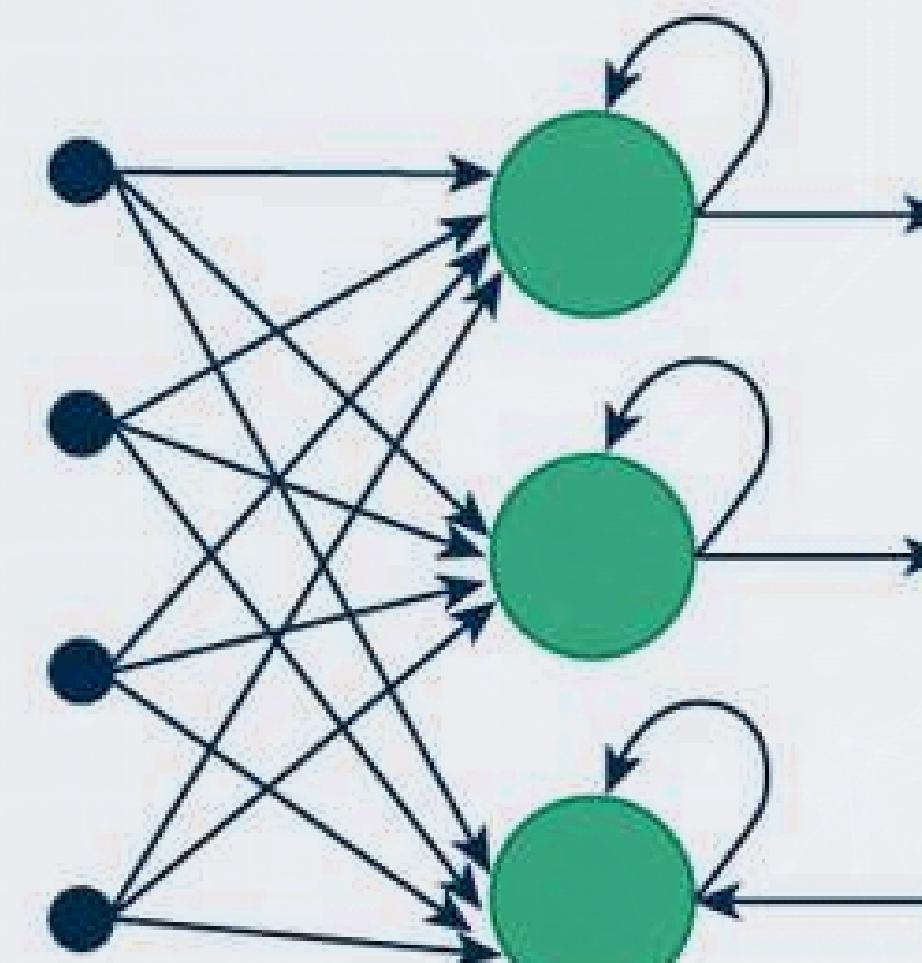
Watch on



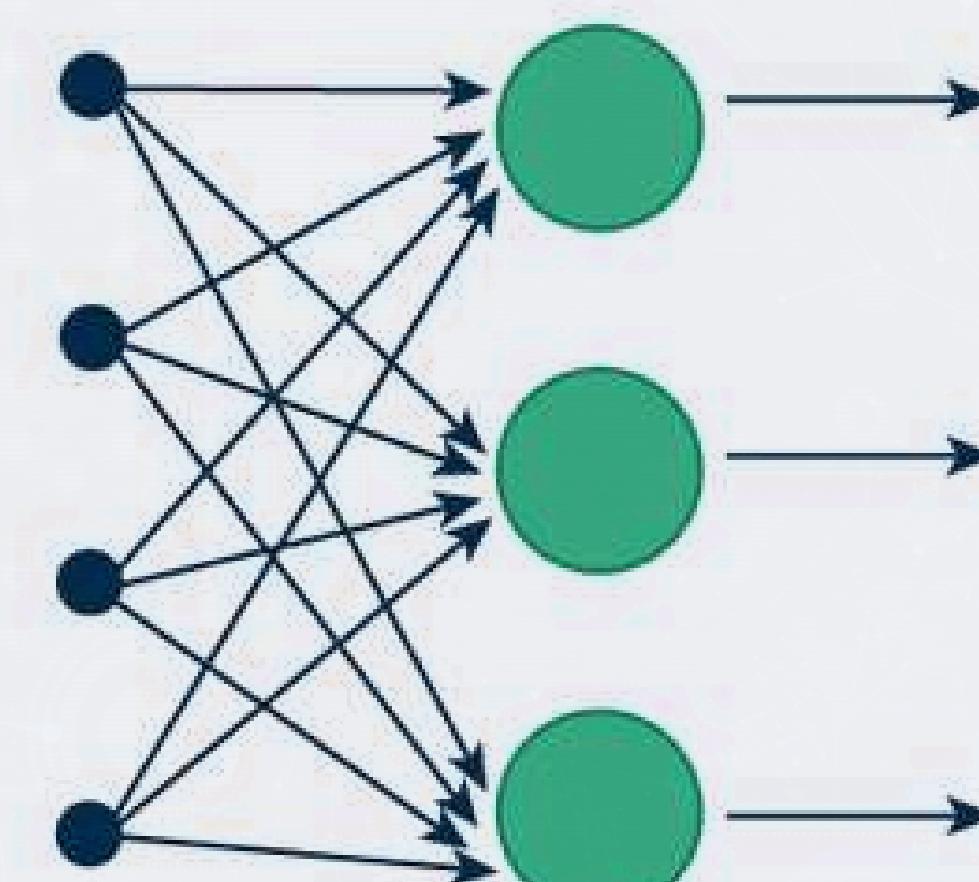


Arquitecturas Recurrentes

Es un tipo de red neuronal diseñada específicamente para procesar datos secuenciales, es decir, aquellos en los que el orden importa, como texto, audio, series temporales, etc.



(a) Recurrent Neural Network



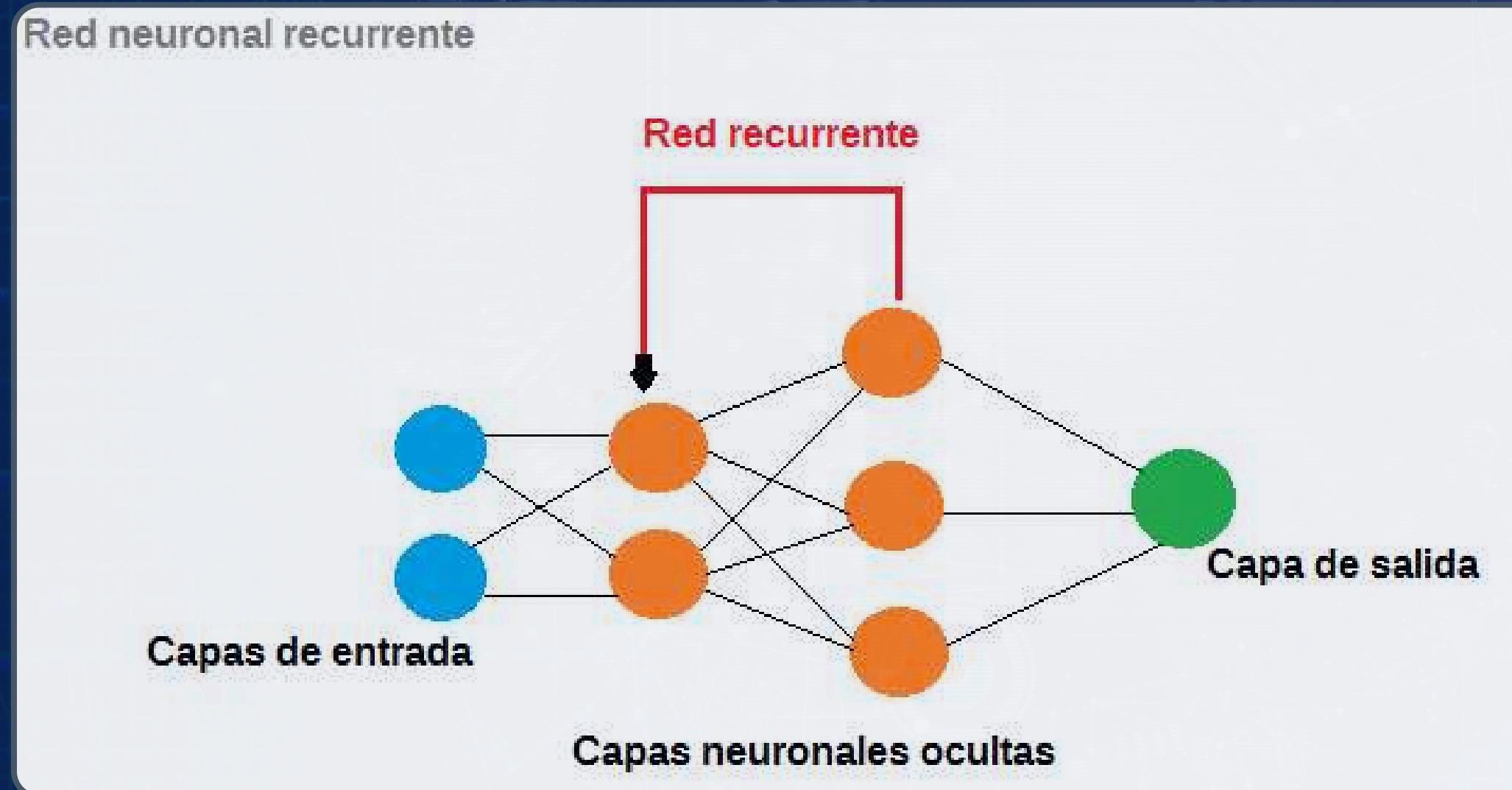
(b) Feed-Forward Neural Network

A diferencia de las redes neuronales tradicionales, las RNN pueden recordar información de entradas anteriores gracias a su arquitectura con conexiones recurrentes, lo que les permite mantener un estado interno o "memoria" a lo largo de la secuencia



Arquitecturas Recurrentes

Funcionamiento

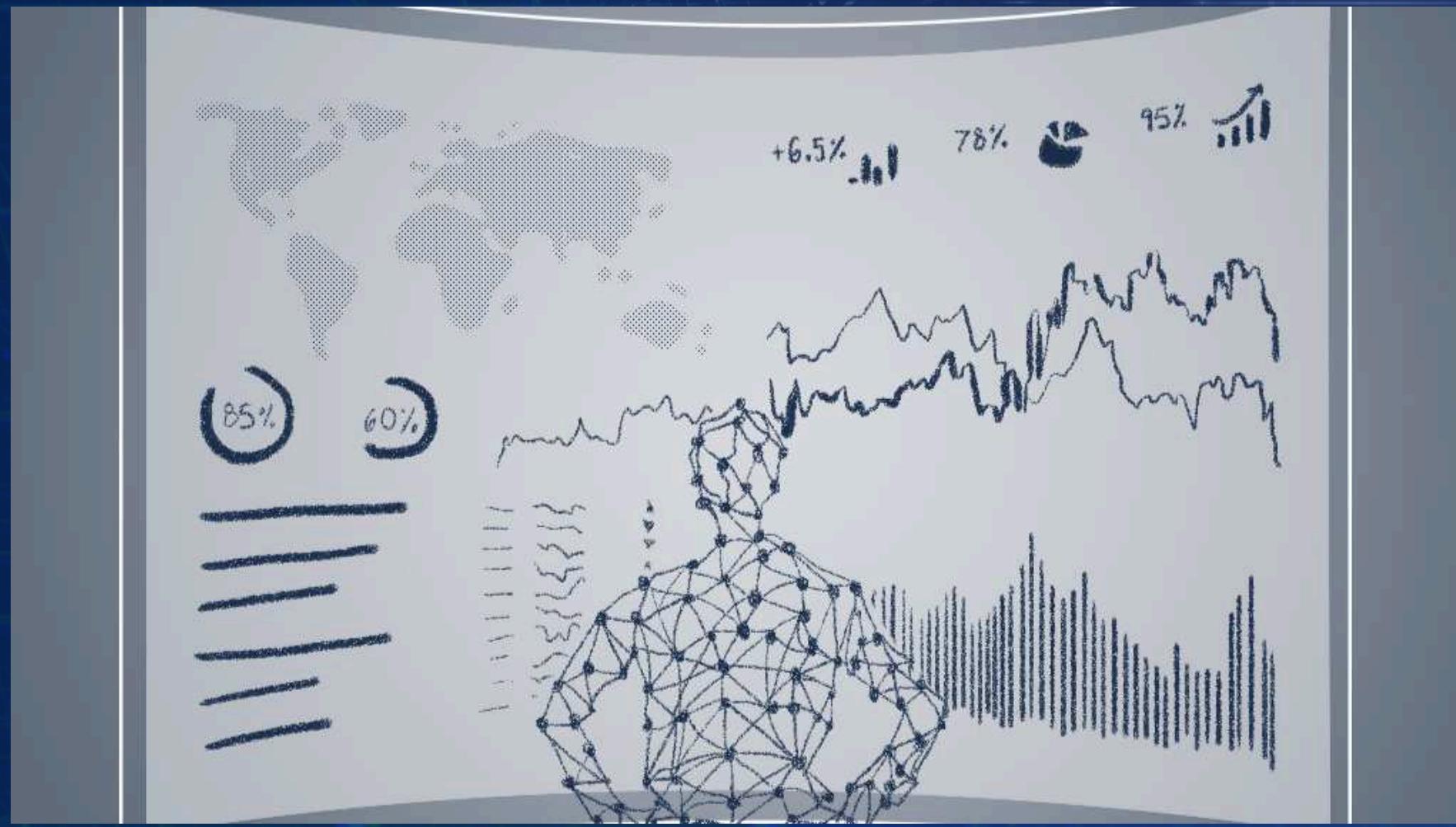


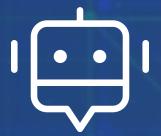
1. La red recibe el primer elemento de la secuencia y lo procesa.
2. El resultado se almacena en el estado interno y se utiliza junto con el siguiente elemento de la secuencia.
3. Este proceso se repite para cada elemento, permitiendo que la red "recuerde" información relevante de pasos anteriores

Arquitecturas Recurrentes

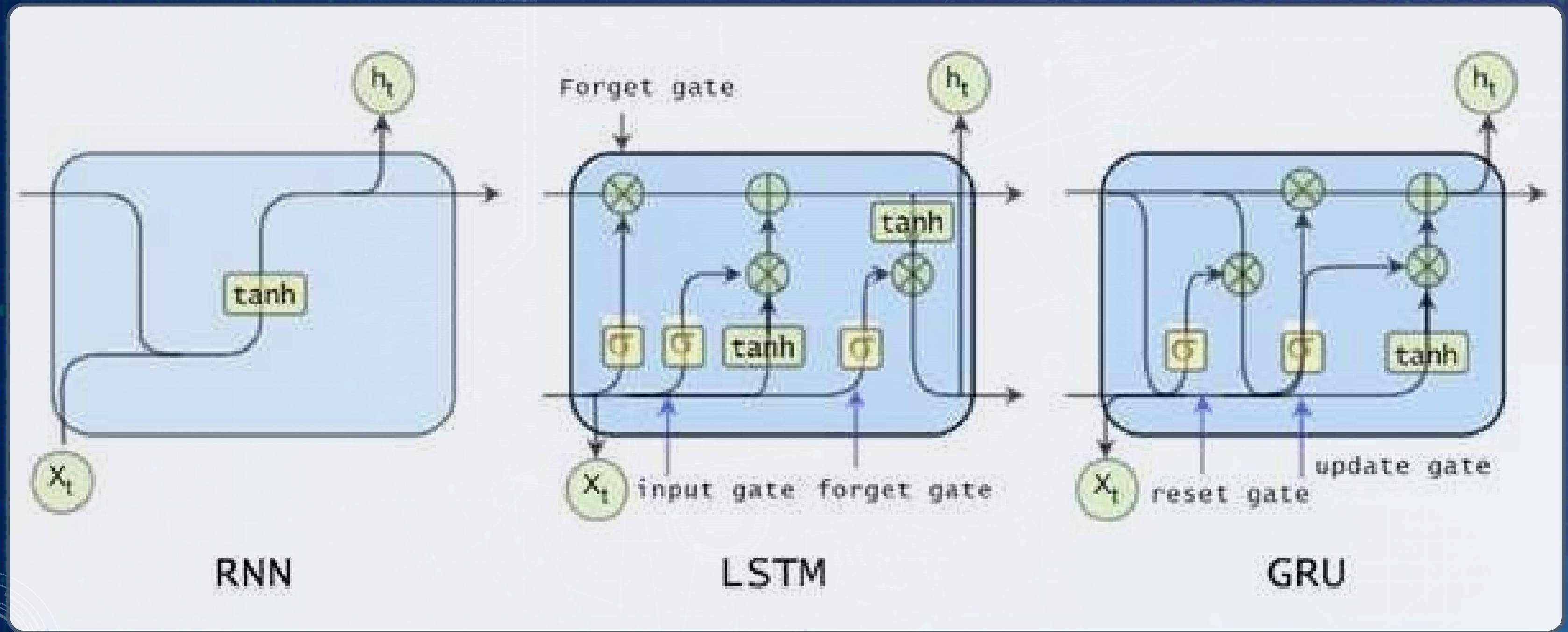
Su memoria de secuencias previas permite aplicarse muy bien en:

1. Procesamiento de lenguaje natural (traducción, generación de texto, análisis de sentimiento)
2. Reconocimiento de voz y síntesis de audio
3. Predicción de series temporales (finanzas, clima)
4. Generación de música y secuencias





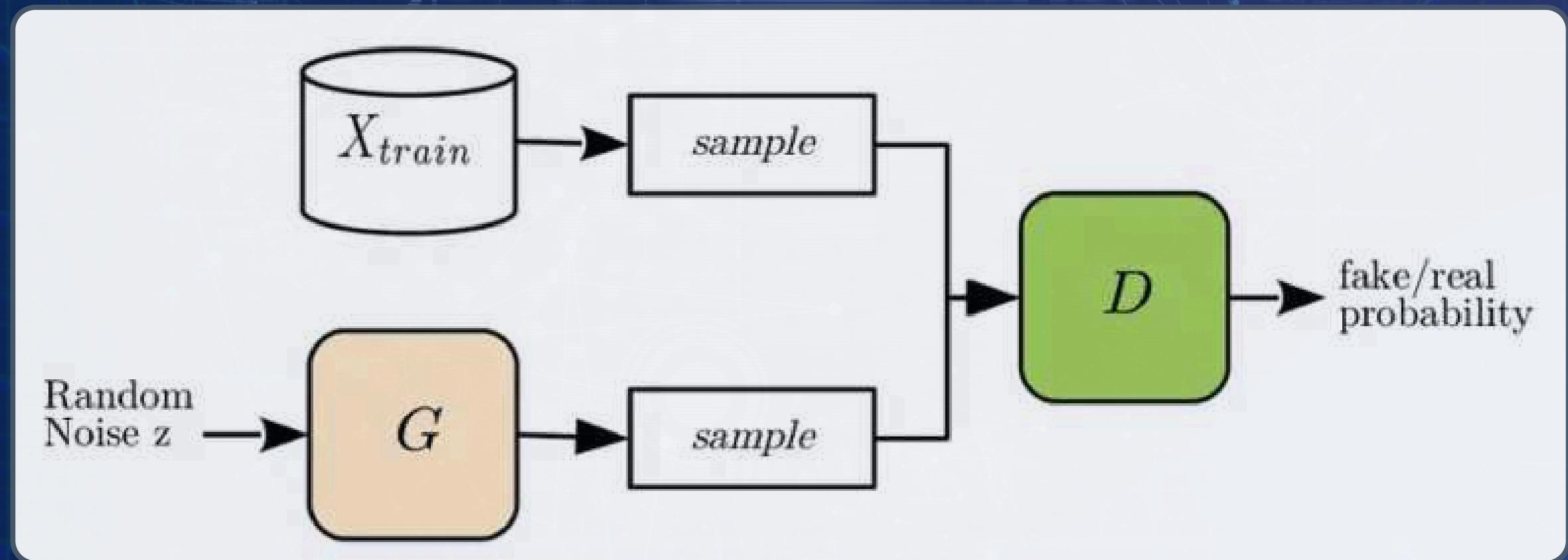
Arquitecturas Recurrentes





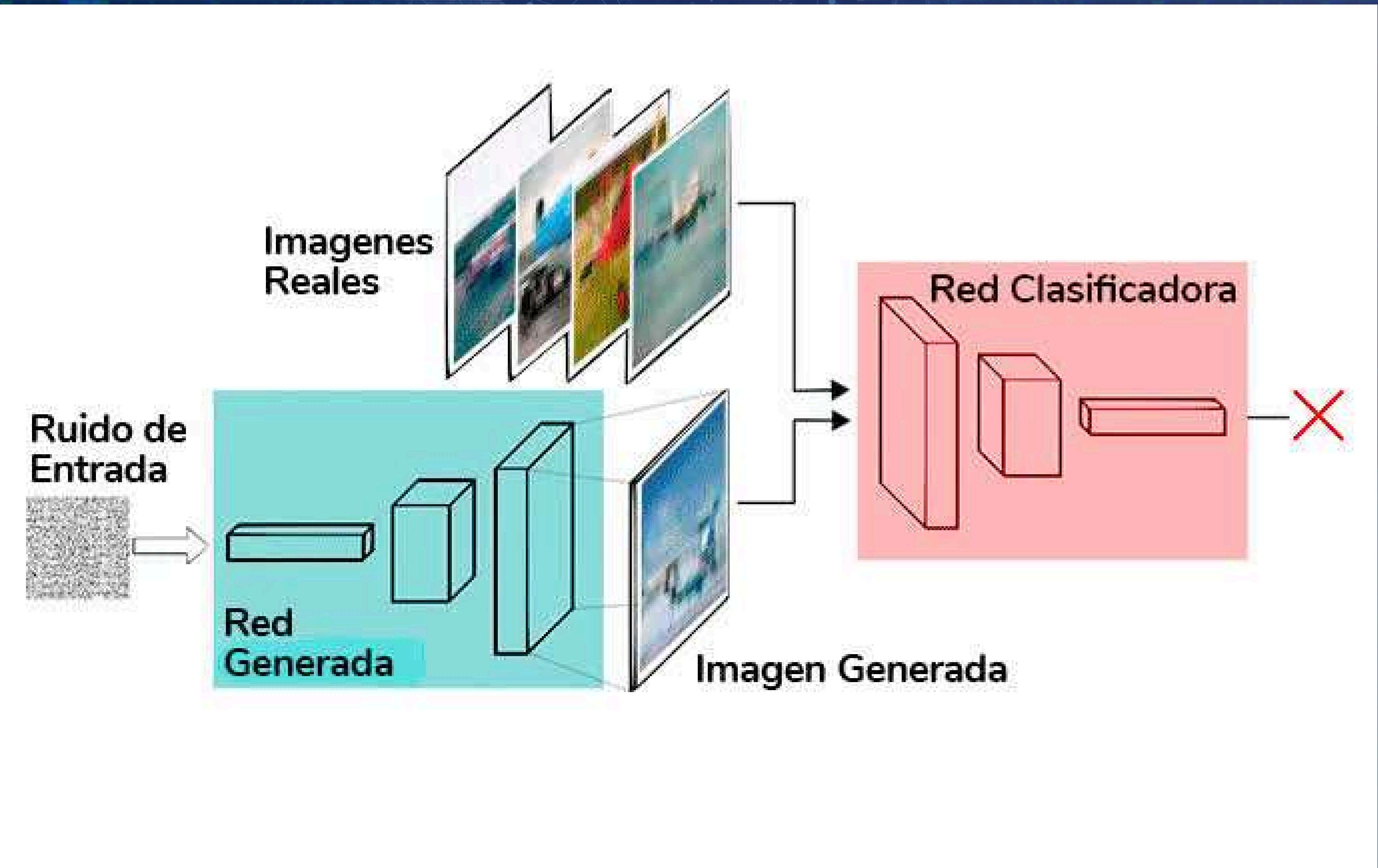
Redes Generativas

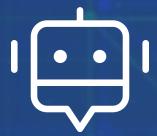
- ¿Qué significa “generar” datos? Crear imágenes, texto, audio o video similares pero nuevos a los del conjunto de entrenamiento.
- Son modelos capaces de aprender la distribución de probabilidad de un conjunto de datos y generar nuevos ejemplos que parezcan provenir de esa distribución.





Redes Generativas





Redes Generativas

- Ejercicio Sugerido: <https://anderfernandez.com/blog/como-crear-una-red-generativa-antagonica-gan-en-python/>

