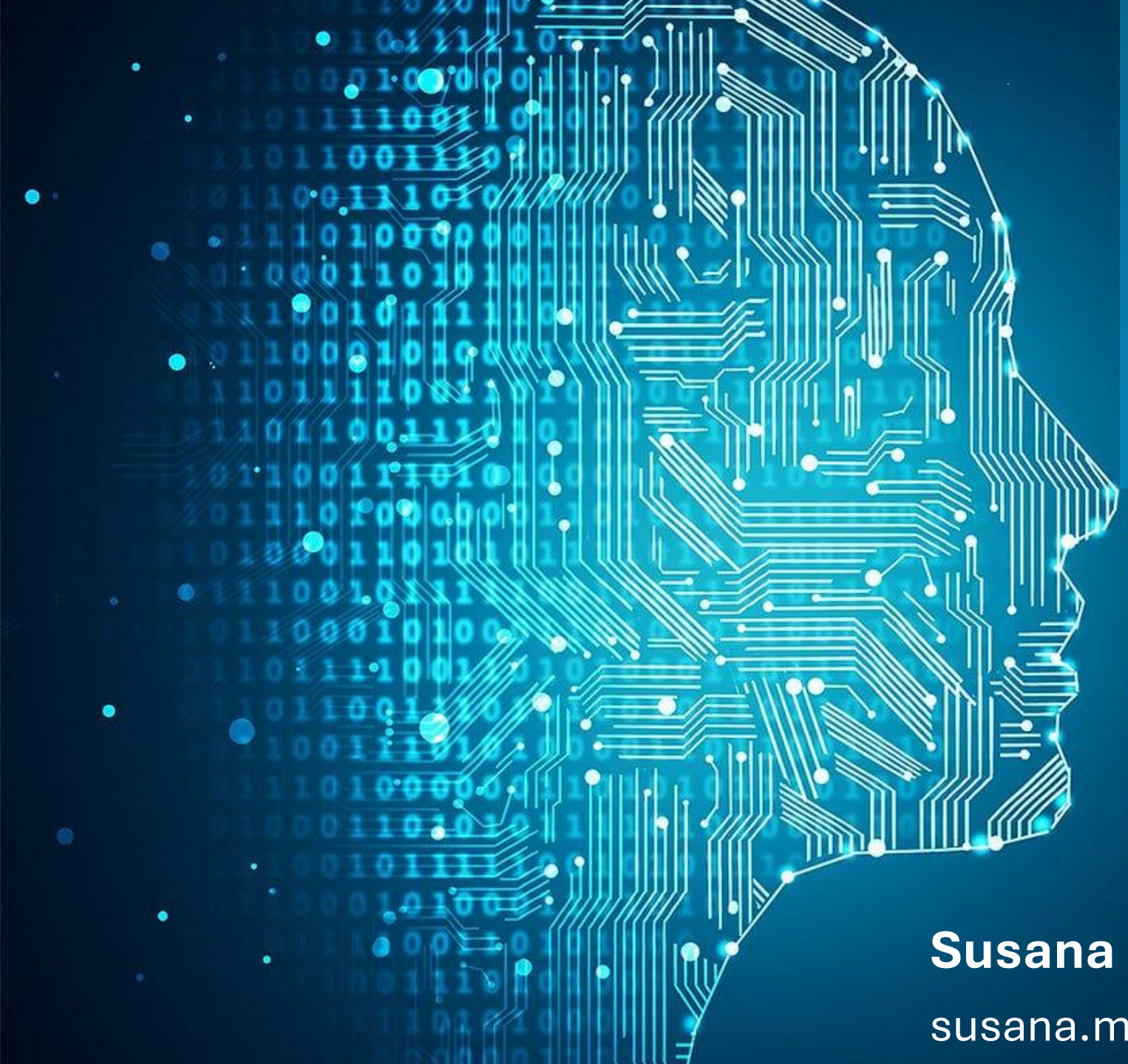


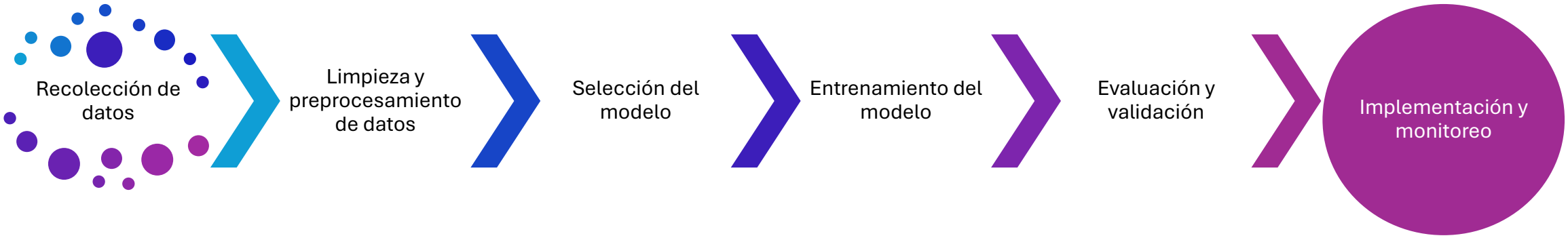
Machine Learning

Susana Medina Gordillo

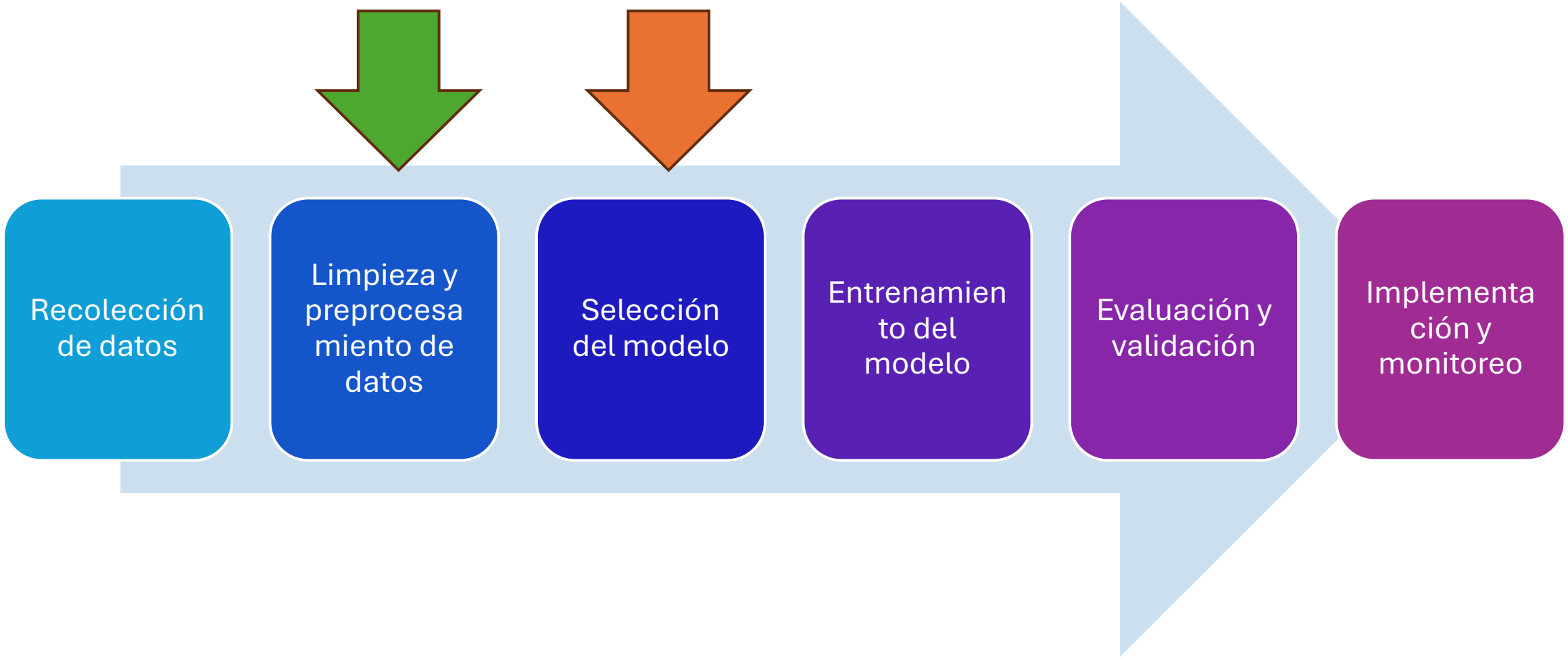
susana.medina@correounivalle.edu.co

Redes Neuronales Artificiales y Deep Learning

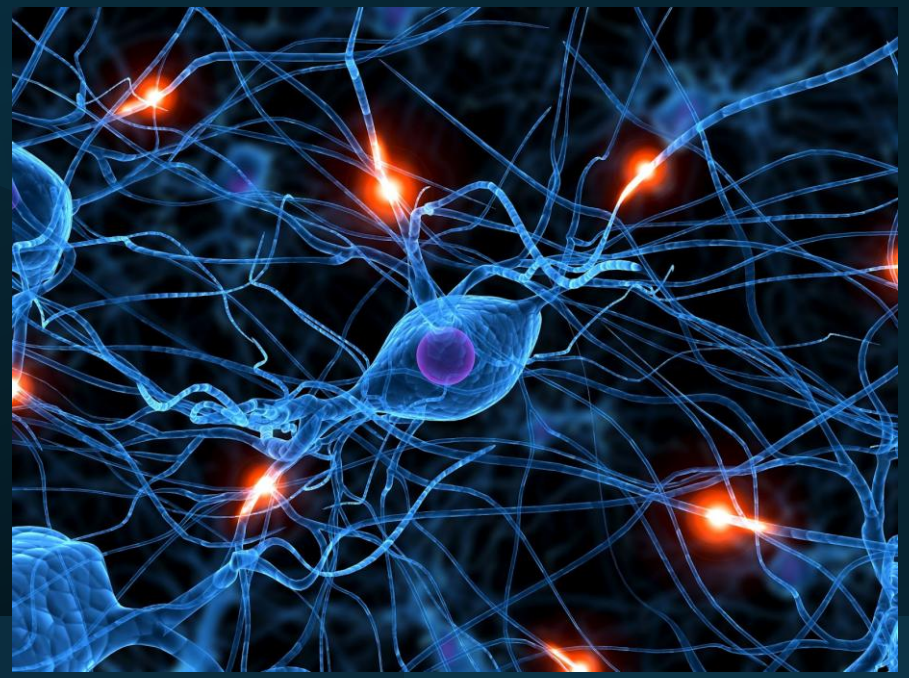
Flujo de trabajo en Machine Learning



Flujo de trabajo en Machine Learning



Redes Neuronales Artificiales y Deep Learning

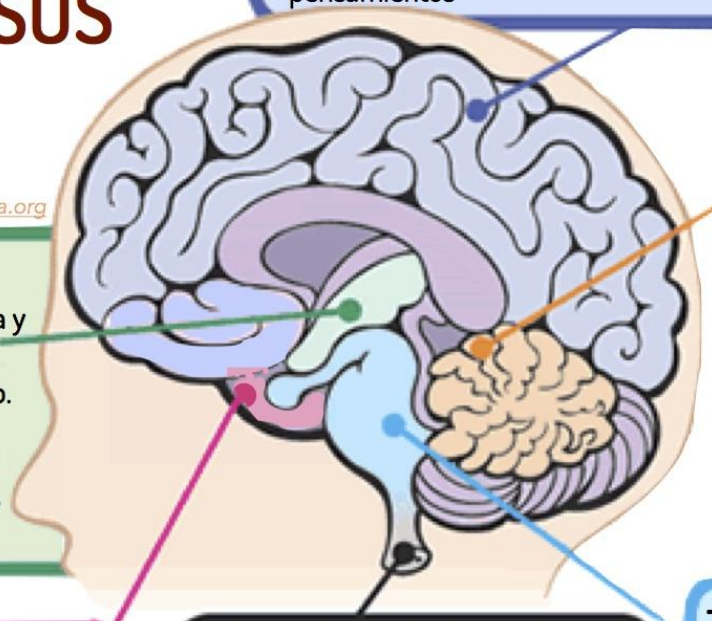


Introducción a las Redes Neuronales Artificiales

El Cerebro

PARTES DEL SISTEMA NERVIOSO CENTRAL Y SUS FUNCIONES

www.neuropediatra.org



Corteza cerebral: cubre la superficie cerebral.
Rige las funciones superiores, de las que somos conscientes.
- percepción sensorial - los 5 sentidos
- movimiento voluntario
- lenguaje
- emociones
- pensamientos

Cerebelo:
Centro de coordinación.

Integra la información que recibe de los 5 sentidos y la cerebral.

Hace que el movimiento sea fluidos y coordinado.

Tálamo y núcleos grises:
Estación intermedia entre corteza y tronco cerebral.
Control del movimiento y del tono.

Hipotálamo:
Control de supervivencia: ingesta, temperatura, defensa, sexual...

Hipocampo y sistema límbico:
Sede principal de la memoria y el aprendizaje.

Forma parte del sistema límico, el principal rector de las emociones

Médula espinal:
Comunica el cerebro y los nervios periféricos
- conduce las sensaciones al cerebro
- lleva los impulsos del movimiento voluntario e involuntario

Tronco encefálico:
Controla las funciones vitales, latido cardíaco y respiración.
El ritmo sueño / vigilia
Núcleos nerviosos de los sentidos, los movimientos de la cabeza y del cuello.

¿Qué son las Redes Neuronales Artificiales (RNA)?

- Inspiradas en la estructura y función del **cerebro biológico**.
- Modelos computacionales compuestos por **nodos** (neuronas) interconectados.
- Diseñadas para aprender **patrones** complejos a partir de datos.
- Bloques fundamentales del **Deep Learning**.

El Perceptrón - La Neurona Artificial Simple

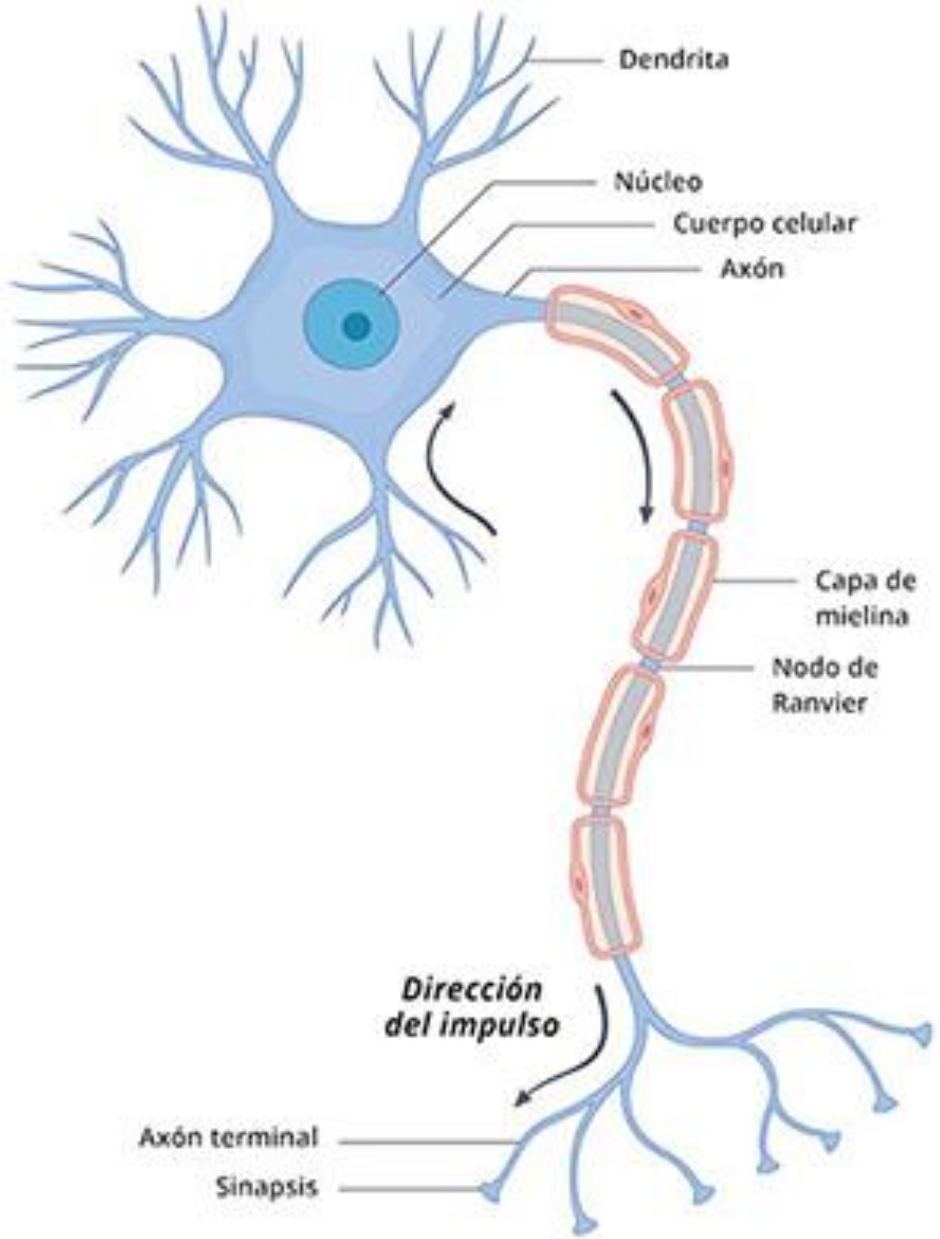
Analogía Biológica:

- ✓ Soma (cuerpo celular)
- ✓ Dendritas (entradas)
- ✓ Axón (salida)
- ✓ Sinapsis (conexiones)

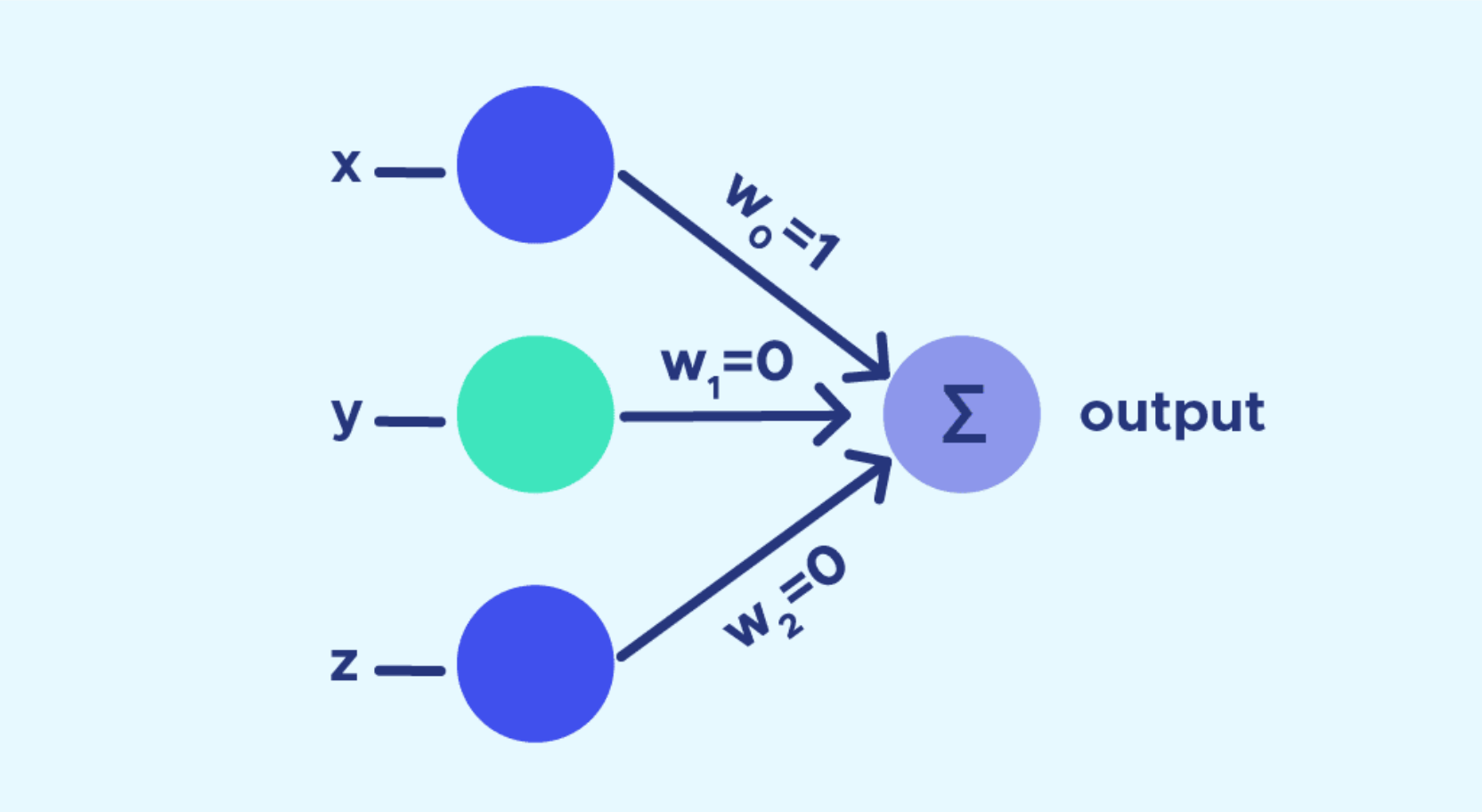
Modelo Matemático:

- ➢ Entradas ($x_1, x_2, \ldots, x_n$)
- ➢ Pesos ($w_1, w_2, \ldots, w_n$) - Importancia de cada entrada.
- ➢ Suma ponderada: $z = \sum_{i=1}^{n} w_i x_i + b$ (donde b es el *bias* / sesgo).
- ➢ Función de activación (σ) - Introduce no linealidad.
- ➢ Salida: $a = \sigma(z)$

Neurona


Dendrita
Núcleo
Cuerpo celular
Axón
Capa de mielina
Nodo de Ranvier
Dirección del impulso
Axón terminal
Sinapsis

El Perceptrón - La Neurona Artificial Simple



Limitaciones del Perceptrón Simple

⊗Solo puede aprender funciones linealmente separables (ej. compuertas **AND**, **OR**).

⊗Incapaz de resolver problemas no lineales (ej. compuerta **XOR**).

⊗Necesidad de arquitecturas más complejas para problemas del mundo real.

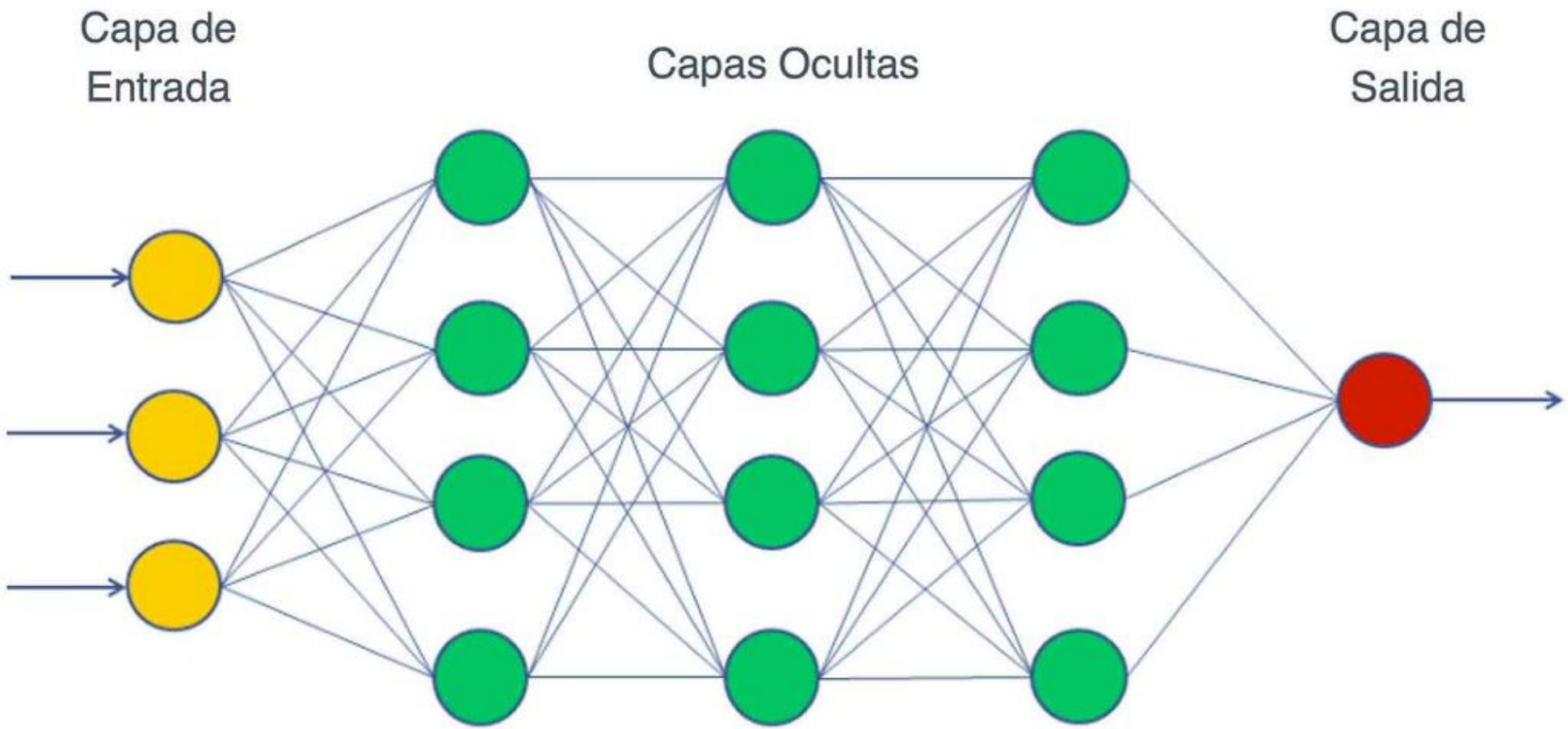
El Perceptrón Multicapa (Multi Layer Perceptron) - La Solución

Consiste en **múltiples capas** de perceptrones interconectados.

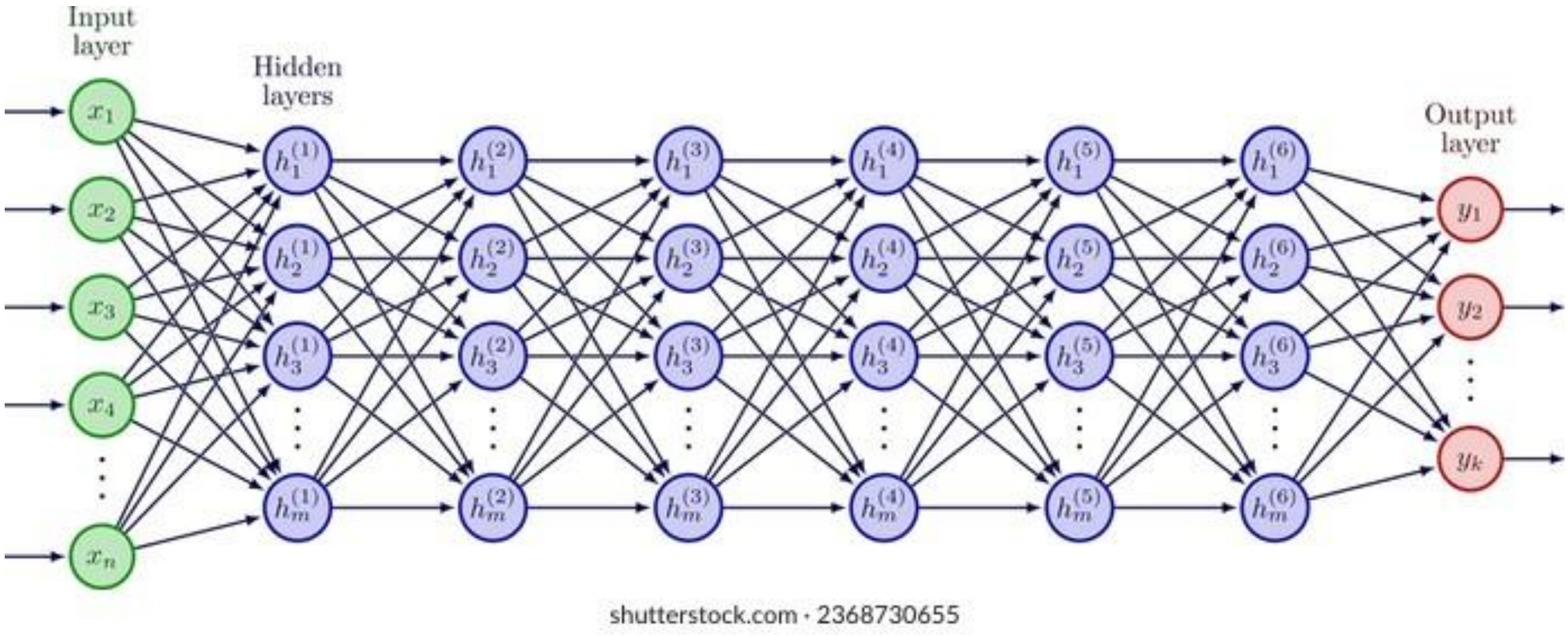
- **Capa de Entrada**: Recibe los datos *brutos*.
- **Capas Ocultas:** Realizan **transformaciones no lineales** de las entradas. Puede haber una o varias.
- **Capa de Salida:** Produce la predicción final (puede tener una o varias neuronas).

Las conexiones entre capas tienen **pesos** asociados.

El Perceptrón Multicapa (MLP) – 1 salida



El Perceptrón Multicapa (MLP) – k Salidas



Funciones de Activación: Introducción a la No Linealidad

Elemento **crucial** para que las Redes Neuronales Artificiales (RNA) aprendan funciones complejas.

Transforman la *señal de entrada* de un nodo de una red neuronal en una *señal de salida* que pasa a la capa siguiente.

Si no hubiera funciones de activación no lineales, un Perceptrón Multicapa (MLP) se comportaría como una regresión lineal.

Funciones de Activación Comunes

Funciones de Activación Comunes: básicas

Sigmoide

- ✓Fórmula: $\sigma(z) = \frac{1}{1+e^{-z}}$
- ✓Salida entre 0 y 1 (útil para probabilidades en clasificación binaria).
- ✓Problemas: Desvanecimiento del gradiente en valores extremos.

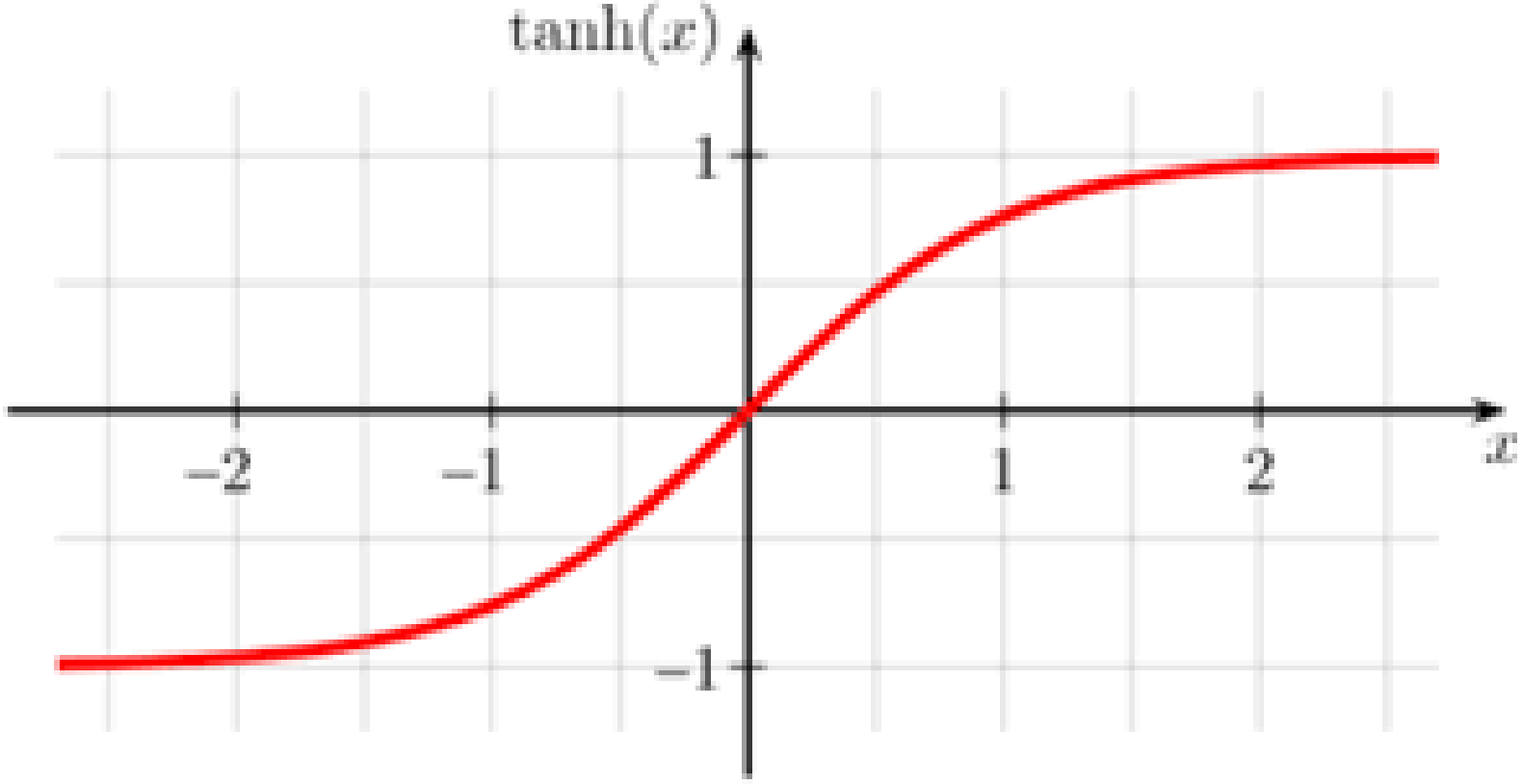
Tangente Hiperbólica (tanh)

- Fórmula: $\tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$
- Salida entre -1 y 1 (centrada en cero).
- Problemas: desvanecimiento del gradiente.

Sigmoide

$$f(x) = \frac{1}{1 + e^{-x}}$$

Tangente Hiperbólica



Rectified Linear Unit (ReLU)

Fórmula: $ReLU(z) = \max(0, z)$

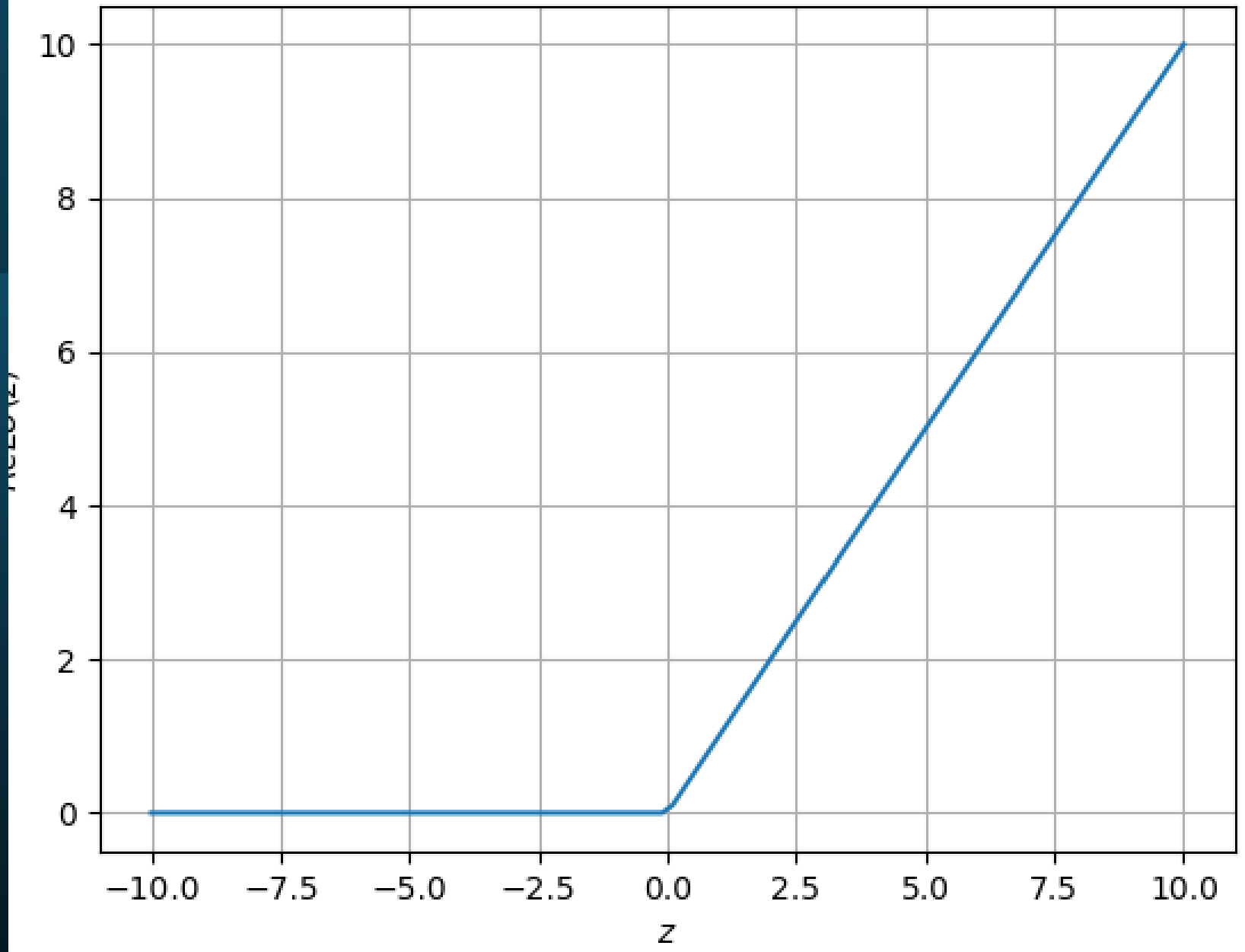
Salida es 0 si la entrada es negativa, y *la entrada* si es positiva.

✓Soluciona **parcialmente** el problema del desvanecimiento del gradiente para valores positivos.

✓Computacionalmente eficiente.

Problema: "Neuronas muertas" si la entrada siempre es negativa.

Rectified Linear Unit (ReLU)



Leaky ReLU

Fórmula: $LeakyReLU(z) = \begin{cases} z & si\ z > 0 \\ \alpha z & si\ z \leq 0 \end{cases}$

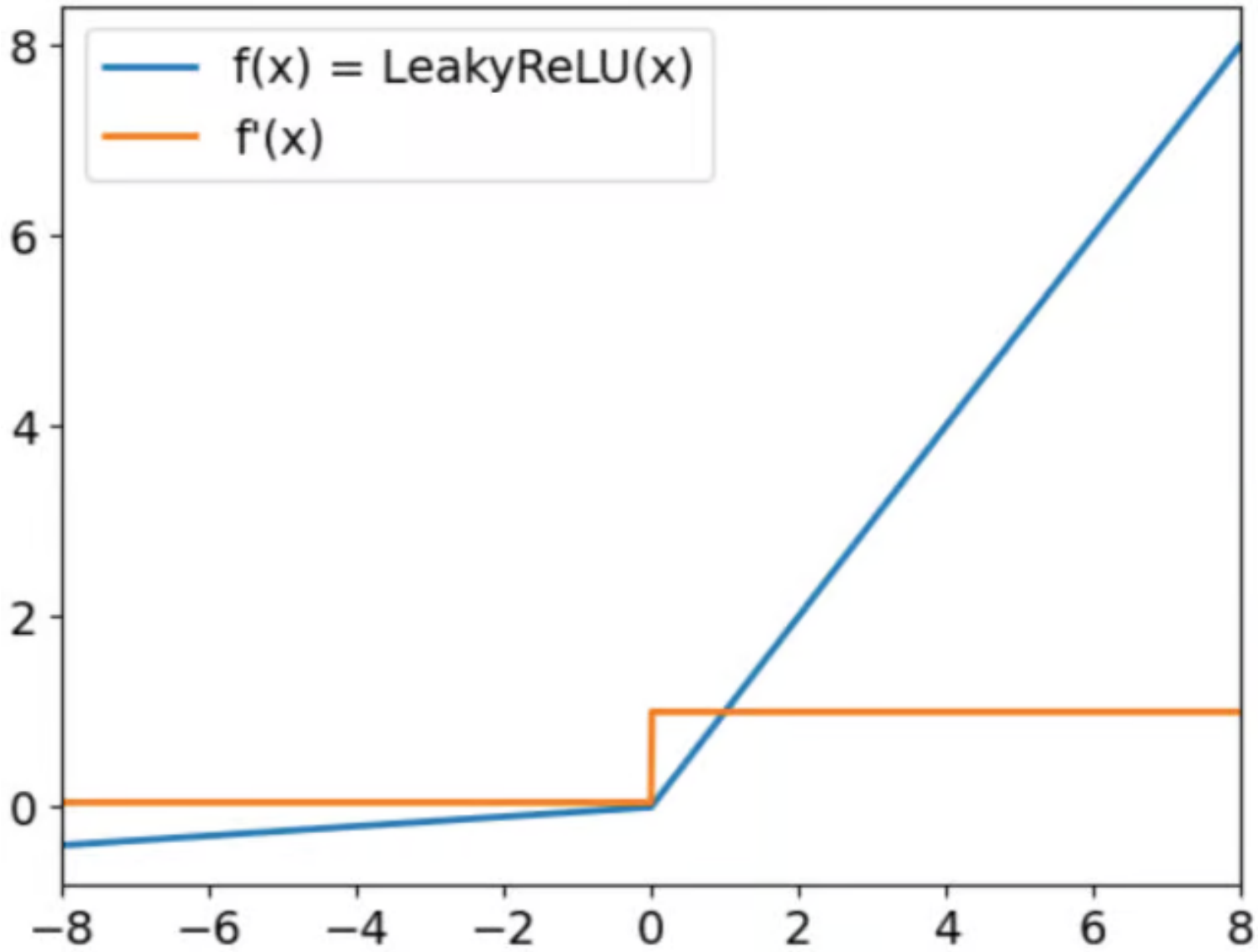
(donde α es un pequeño valor positivo).

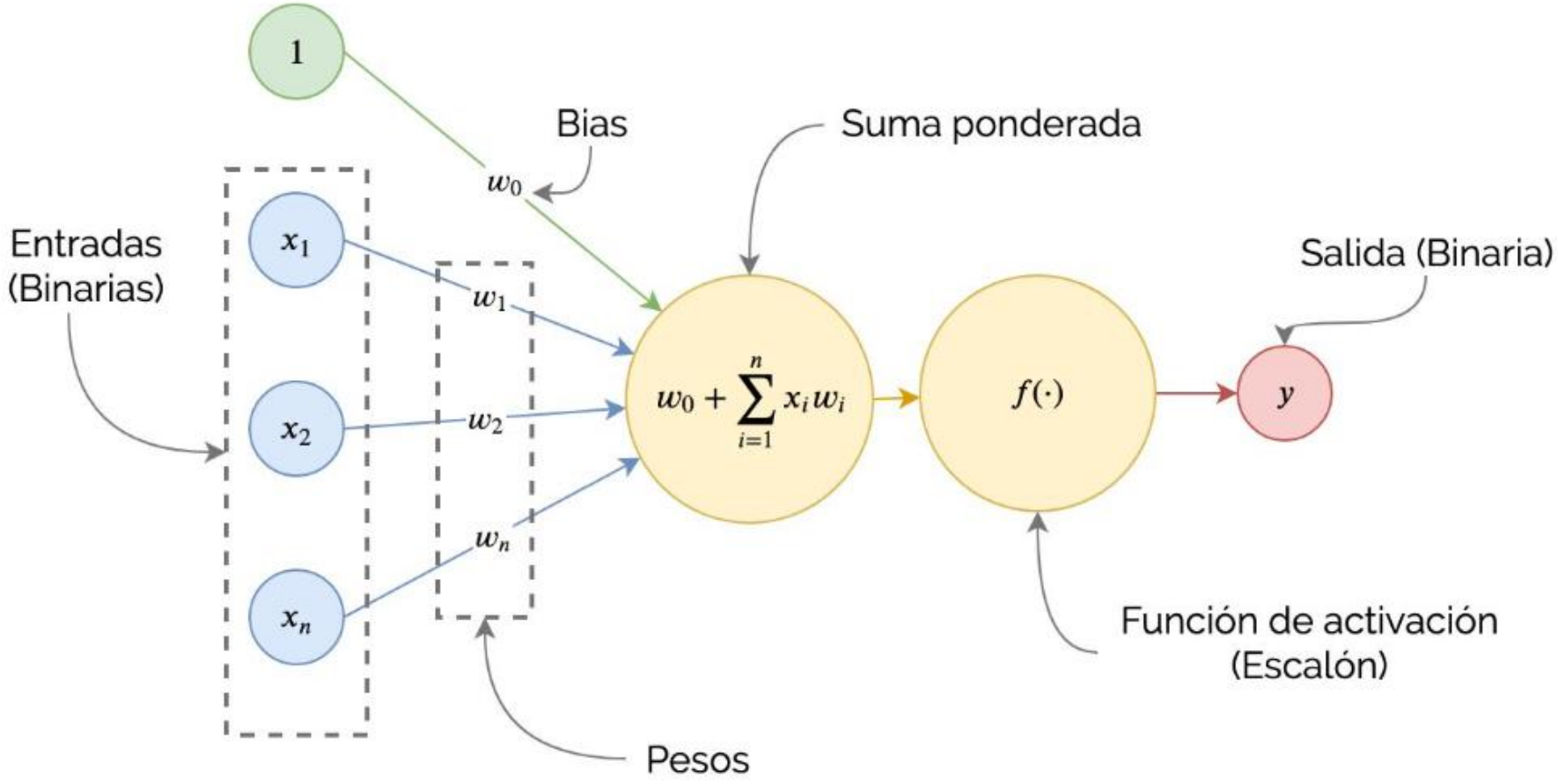
Intenta solucionar el problema de las neuronas moribundas.

ReLU siempre da valores nulos para las entradas negativas.

Cuando esto ocurra, los gradientes de los nodos con valores negativos se pondrán a cero durante el resto del entrenamiento, lo que impedirá que este **parámetro aprenda**.

Leaky ReLU





Perceptrón con función de activación

Elección de la función de activación adecuada

Para la clasificación binaria

Utiliza la función de activación sigmoide en la capa de salida. Reducirá las salidas a un valor entre 0 y 1, que representa las probabilidades de las dos clases.

Para la clasificación multiclase

Utiliza la función de activación softmax en la capa de salida. Dará como resultado distribuciones de probabilidad sobre todas las clases.

Si no estás seguro

Utiliza la función de activación ReLU en las capas ocultas. ReLU es la función de activación predeterminada más común y suele ser una buena elección.

En resumen

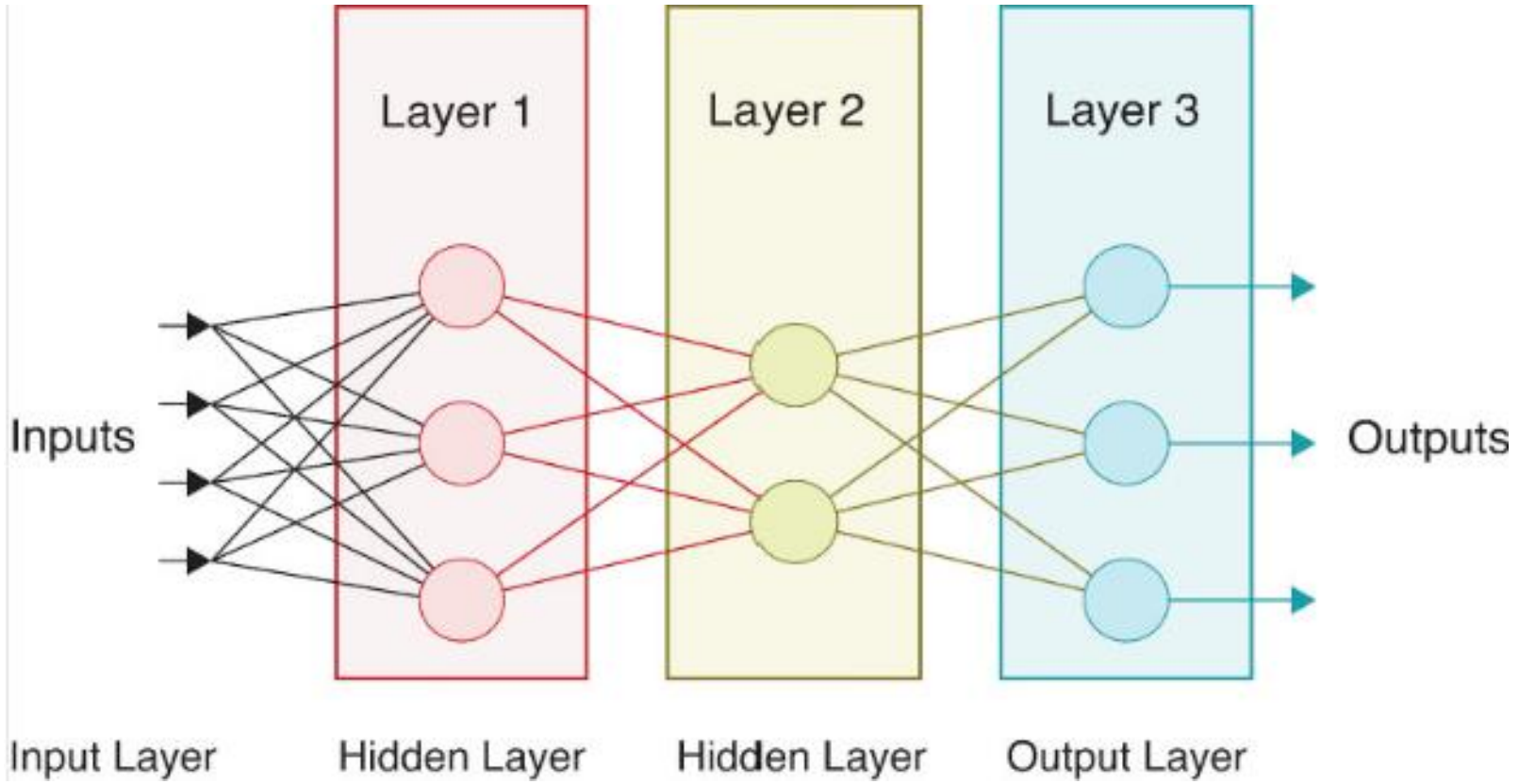
- Introducción al concepto de Redes Neuronales Artificiales.
- El Perceptrón como la unidad básica.
- Limitaciones del perceptrón simple y la necesidad del MLP.
- Importancia de las funciones de activación para la no linealidad.
- Exploración de funciones de activación comunes (Sigmoide, Tanh, ReLU, Leaky ReLU).

Forward Propagation y Backpropagation

Forward Propagation:
La Propagación Hacia Adelante

- ✓El proceso de calcular la salida de la red para una entrada dada.
- ✓Los datos de entrada se pasan a través de la red, capa por capa.
- ✓En cada neurona, se calcula la suma ponderada de las entradas y se aplica la función de activación.
- ✓La salida de una capa se convierte en la entrada de la siguiente capa.
- ✓El proceso continúa hasta llegar a la capa de salida, donde se obtiene la predicción.

Forward Propagation:
Ejemplo Simplificado (MLP de 2 capas)



Backpropagation - Aprendiendo de los Errores

El algoritmo fundamental para **entrenar** redes neuronales.

Basado en la regla de la cadena del **cálculo diferencial**.

Objetivo: Ajustar los pesos de las conexiones para minimizar la diferencia entre las predicciones de la red y los valores reales (función de pérdida).

Pasos del Backpropagation

Forward Pass: Se pasa una entrada a través de la red y se calcula la salida.

Cálculo del Error: Se calcula la función de pérdida (error) entre la salida predicha y la salida real.

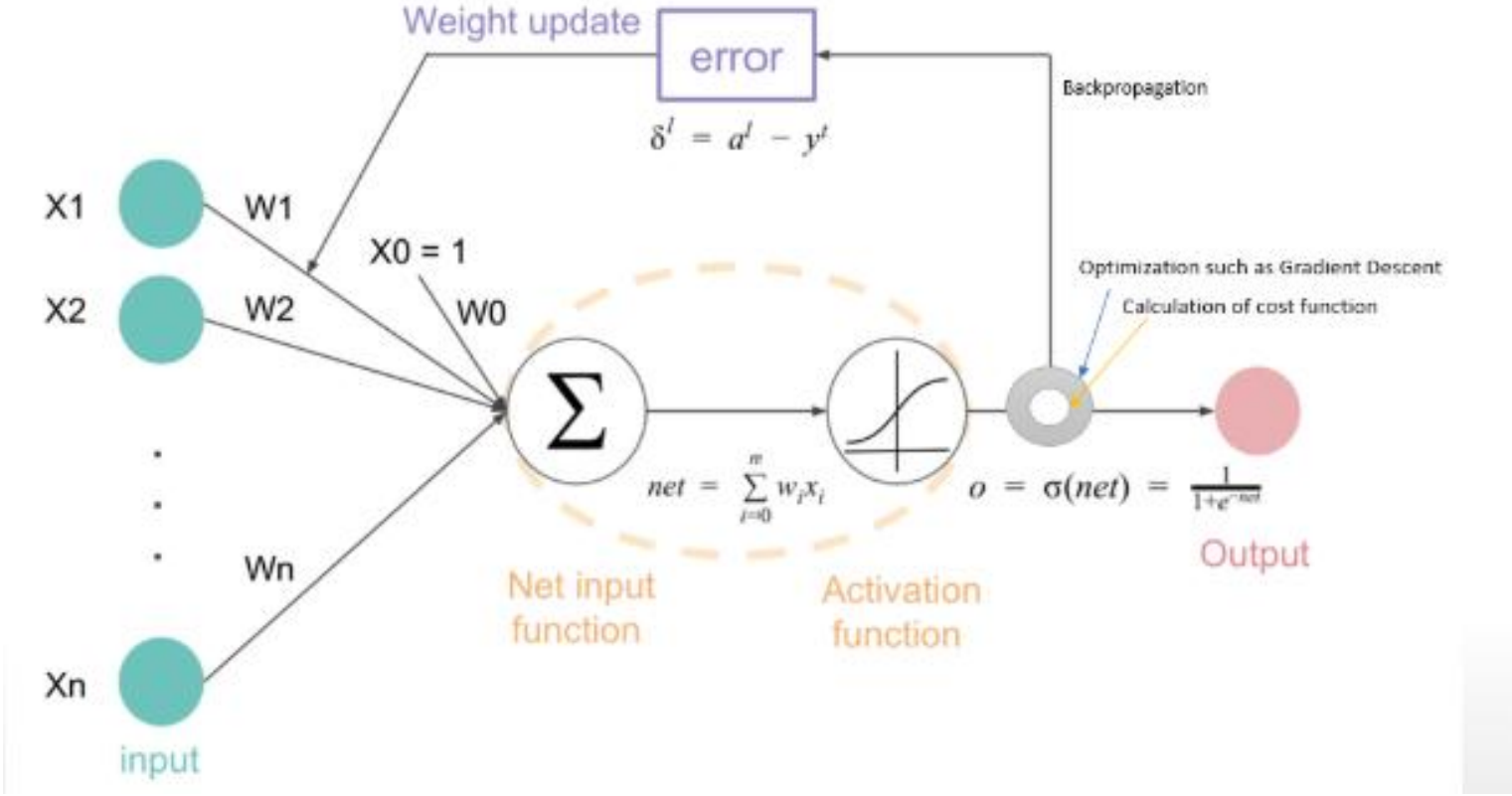
Backward Pass: El error se propaga hacia atrás a través de la red, capa por capa.

Cálculo de los Gradientes: Se calcula el gradiente de la función de pérdida con respecto a cada peso y bias en la red. El gradiente indica la dirección y magnitud del cambio necesario para reducir el error.

Actualización de los Pesos: Los pesos y biases se actualizan utilizando un algoritmo de optimización (ej. Descenso de Gradiente) en la dirección opuesta al gradiente, multiplicado por una tasa de aprendizaje (α).

- wnuevo=wviejo−α∂w∂Pe´rdida

Backpropagation - Visualización del Flujo del Error)



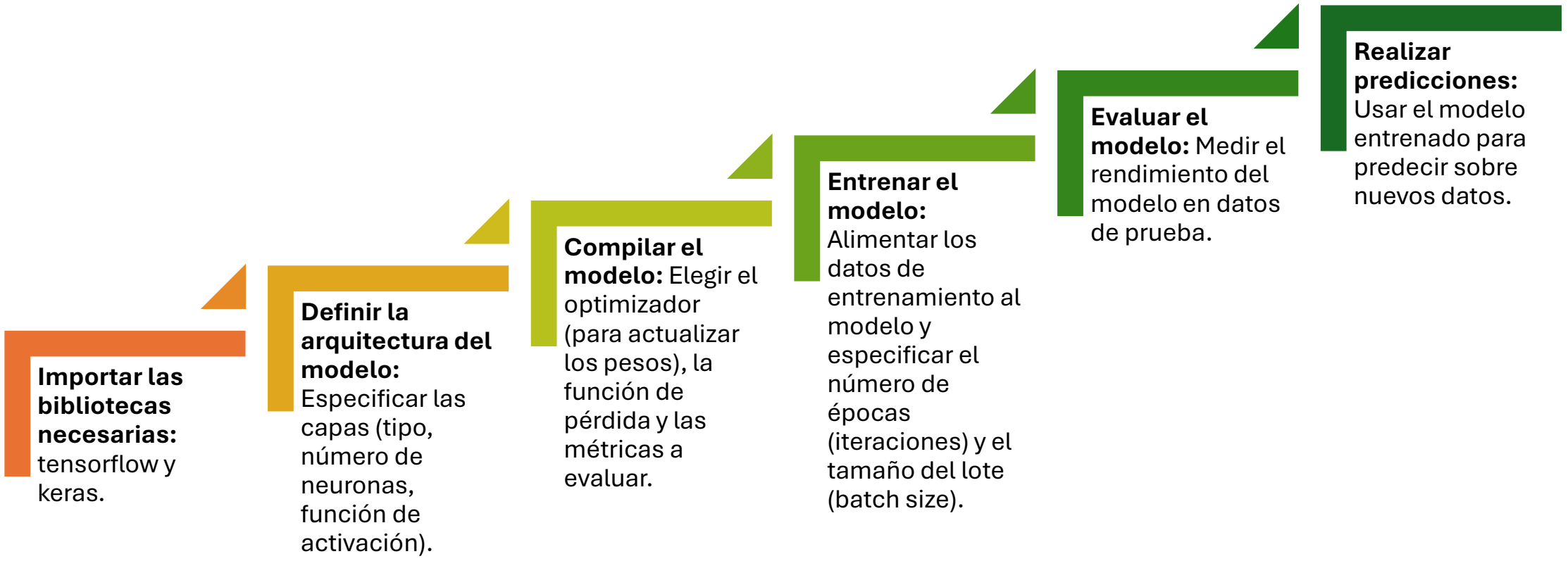
Construcción de una Red Neuronal Básica con TensorFlow/Keras

Introducción a TensorFlow y Keras

TensorFlow: Una biblioteca de código abierto para computación numérica utilizando grafos de flujo de datos. Ampliamente utilizada para Machine Learning y Deep Learning.

Keras: Una API de alto nivel para construir y entrenar modelos de redes neuronales. Puede correr sobre TensorFlow (u otros backends). Facilita la creación rápida de prototipos.

Pasos para Construir una Red Neuronal con Keras



En resumen

Introducción a TensorFlow y Keras como herramientas para Deep Learning.

Pasos fundamentales para construir una red neuronal con Keras.

Ejemplos de código para definir, compilar, entrenar, evaluar y usar un modelo.

Construcción de una Red Neuronal Básica con TensorFlow/Keras

scikit-learn

Machine Learning in Python

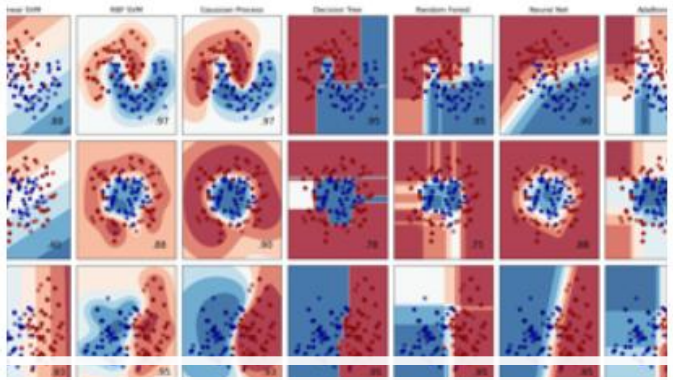
Getting Started | Release Highlights for 1.6

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...

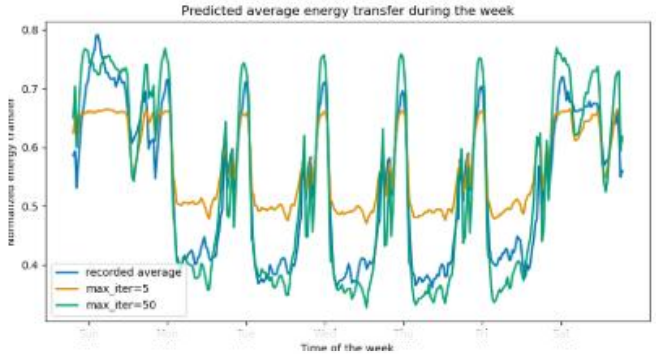


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...

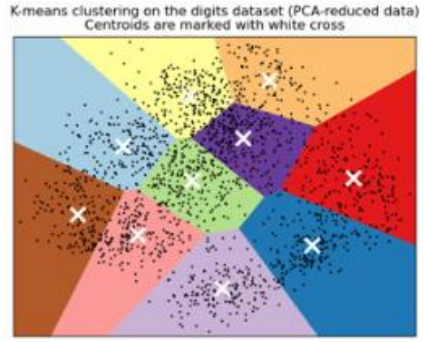


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, grouping experiment outcomes.
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, increased efficiency.

Model selection

Comparing, validating and choosing parameters and models.

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for

Introducción al Deep Learning

¿Qué es el Deep Learning?

Un subcampo del **Machine Learning** que utiliza redes neuronales con **múltiples capas** (redes neuronales profundas) para aprender representaciones jerárquicas de los datos.

Inspirado en la forma en que el cerebro procesa la información a través de **múltiples etapas**.

Capaz de **aprender características complejas** y **abstractas** automáticamente a partir de datos sin procesar.

La "Profundidad" en Deep Learning

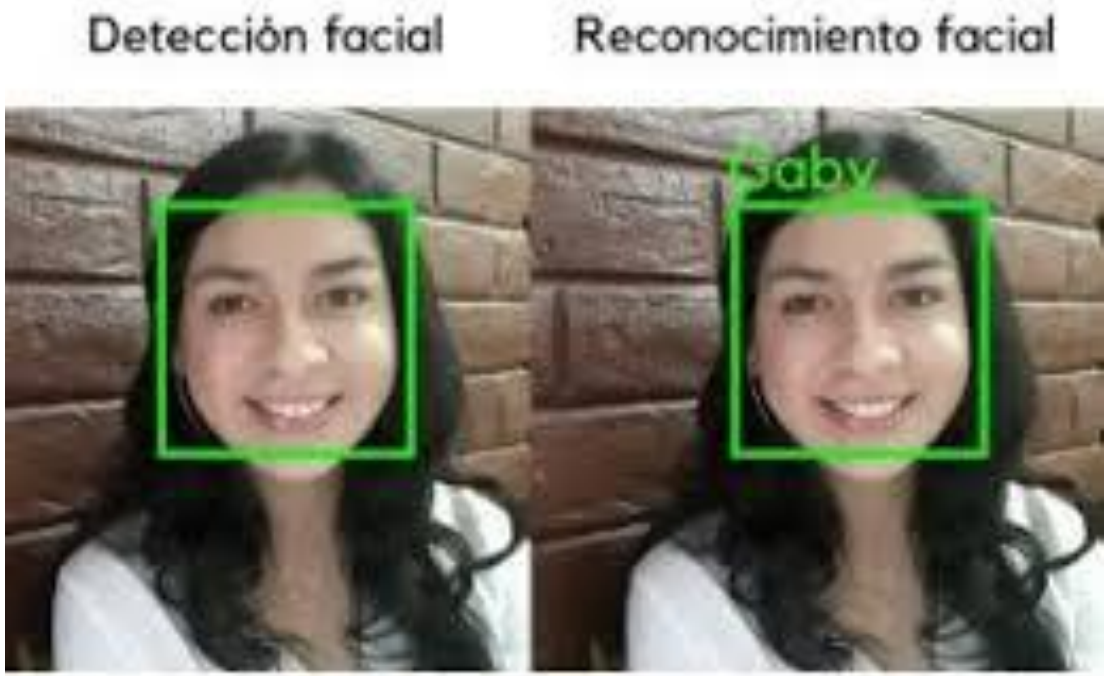
- ✓Se refiere al número de capas ocultas en la red neuronal.
- ✓Las redes neuronales tradicionales pueden tener **una** o pocas capas ocultas.
- ✓Las redes de Deep Learning pueden tener decenas o incluso cientos de capas.
- ✓Cada capa aprende representaciones de características en un nivel diferente de abstracción.

Aprendizaje Jerárquico de Características Ejemplo con Imágenes

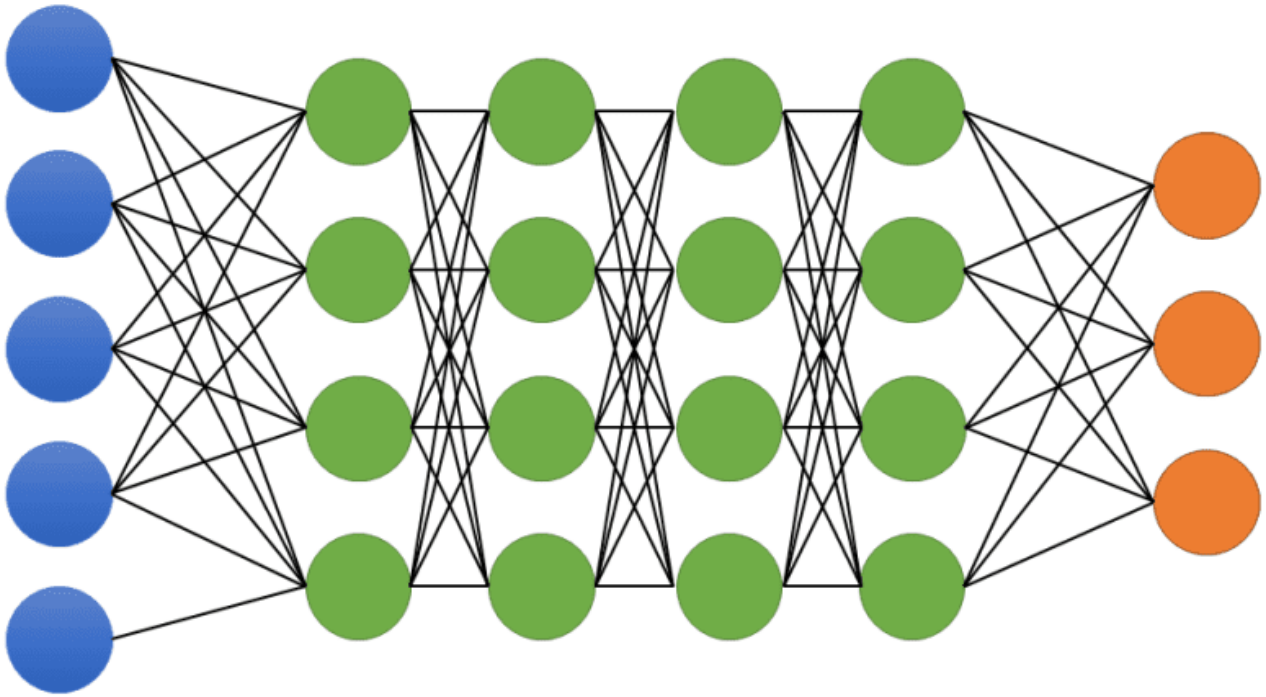
Capa 1: Puede aprender bordes y esquinas.

Capa 2: Puede aprender formas y texturas.

Capas más profundas: Pueden aprender partes de objetos y, finalmente, objetos completos.



Aprendizaje Jerárquico de Características

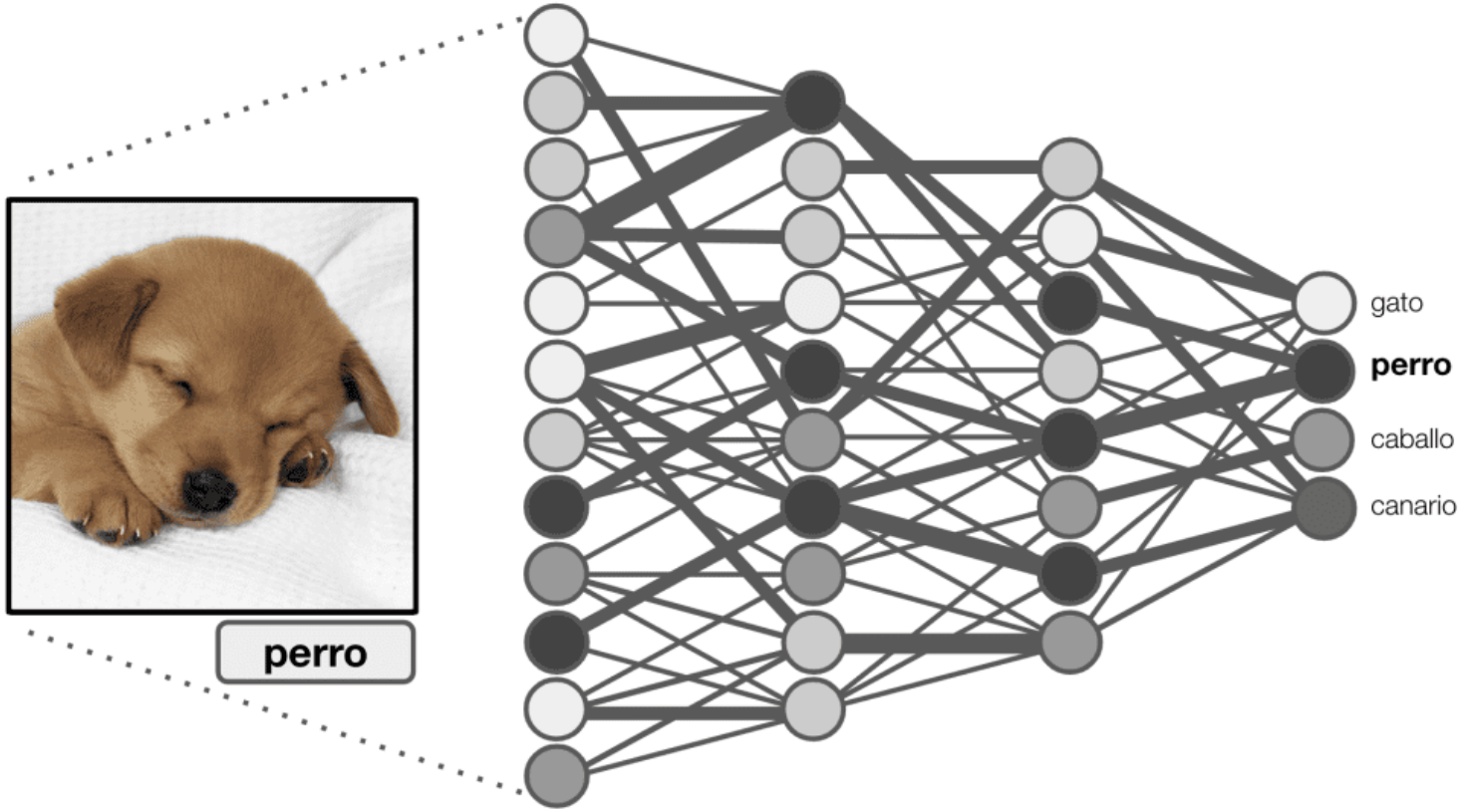


Capa inicial
(Atributos)

Capas ocultas
(Neuronas)

Capa final
(Categorías)

Aprendizaje Jerárquico de Características
Ejemplo con Imágenes



¿Por qué el Deep Learning ha tenido tanto éxito recientemente?

Disponibilidad de grandes cantidades de datos: Las redes profundas requieren muchos datos para aprender de manera efectiva.

Avances en el hardware: La potencia de cómputo (*GPUs*) ha aumentado significativamente, permitiendo entrenar redes más grandes en tiempos razonables.

Nuevas arquitecturas y técnicas: Desarrollo de arquitecturas innovadoras (CNNs, RNNs, Transformers) y técnicas de entrenamiento más efectivas (ej. optimizadores avanzados, regularización).

Aplicaciones del Deep Learning – Parte 2

Reconocimiento de Voz: Transcripción de audio a texto.

Sistemas de Recomendación: Sugerir productos o contenido a usuarios.

Juegos: IA para agentes inteligentes en videojuegos (ej. AlphaGo).

Medicina: Diagnóstico de enfermedades, descubrimiento de fármacos.

Conducción Autónoma: Percepción Ambiental Avanzada (CNNs), Planificación y Toma de Decisiones (RNNs y Deep Reinforcement Learning - DRL)

Tipos de redes neuronales: estructura

Prealimentadas

Procesan los datos en una dirección, desde el nodo de entrada hasta el nodo de salida. Todos los nodos de una capa están conectados a todos los nodos de la capa siguiente. Una red prealimentada utiliza un proceso de retroalimentación para mejorar las predicciones a lo largo del tiempo.

Retropropagación

Aprenden de forma continua mediante el uso de bucles de retroalimentación correctivos para mejorar su análisis predictivo. Los datos fluyen desde el nodo de entrada hasta el nodo de salida a través de muchos caminos diferentes. Solo un camino es el correcto: el que asigna el nodo de entrada al nodo de salida correcto. Para encontrarlo, la red neuronal utiliza un bucle de retroalimentación.

Convolucionales - CNN

Las capas ocultas realizan funciones matemáticas específicas, como la síntesis o el filtrado, denominadas convoluciones. Son muy útiles para la clasificación de imágenes porque pueden extraer características relevantes de las imágenes que son útiles para el reconocimiento y la clasificación de imágenes. Cada capa oculta extrae y procesa diferentes características de la imagen, como los bordes, el color y la profundidad.

Deep Learning = Aprendizaje por Refuerzo?

Deep Learning = Aprendizaje por Refuerzo?

Aprendizaje por Refuerzo (RL)

Es un paradigma de aprendizaje automático donde un **agente** aprende a tomar **secuencias de decisiones** en un **entorno** para maximizar una **recompensa** acumulada.

A diferencia del aprendizaje supervisado, no se le proporcionan etiquetas explícitas de qué acción tomar.

En cambio, el agente aprende a través de la **interacción** con el entorno, recibiendo **señales de recompensa (o castigo)** después de cada acción.

Componentes clave del RL

Agente: El aprendiz que toma las decisiones.

Entorno: El mundo con el que el agente interactúa.

Estado: Una representación de la situación actual del entorno.

Acción: Una decisión que el agente puede tomar en un estado dado.

Recompensa: Una señal numérica que indica la consecuencia inmediata de una acción.

Política: La estrategia del agente para elegir acciones en función del estado actual.

Valor: Una estimación de la recompensa total futura que se espera recibir al estar en un estado particular (o al tomar una acción en un estado particular).

Clasificación de imágenes con RNA

developers.google.com

Clasificación de imágenes

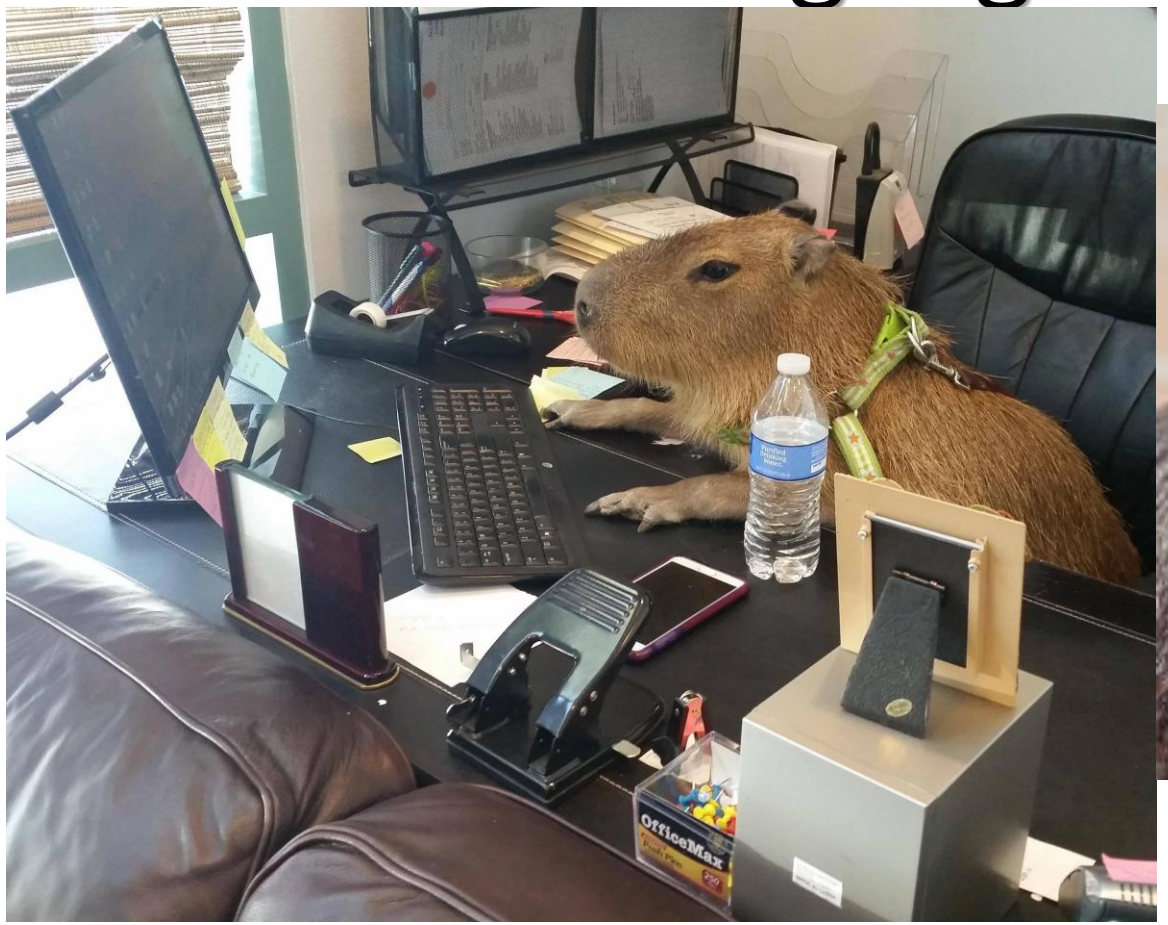
Leer la sección y realizar el quiz al final (Verifica tu comprensión)

Conclusiones...

- Deep learning usa redes neuronales profundas con muchas capas.
- Estas redes aprenden representaciones complejas de los datos.
- Requieren grandes cantidades de datos y alta capacidad de cómputo.
- Impulsan avances en visión, lenguaje y muchas otras áreas.

Ejercicio práctico

colab.research.google.com

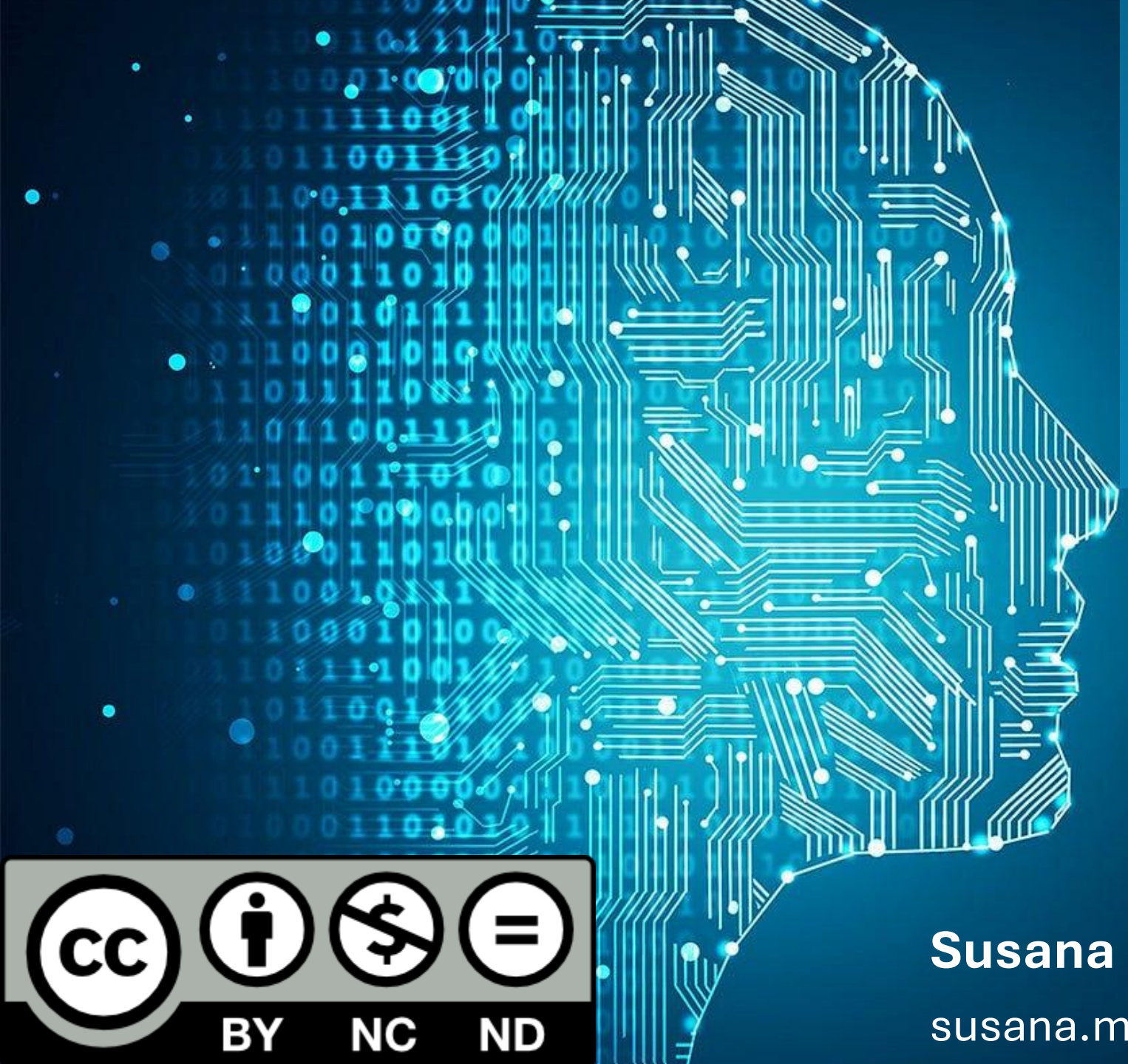


Ejercicio práctico: Google Colaboratory (*Colabs*)

- Página oficial: https://colab.google/
- Abrir Colab (incluye tutorial): https://colab.research.google.com/
- Guía para EDA: https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Exploratory_data_Analysis.ipynb
- Guía / tutorial para Selección de características con **scikit-learn**: https://www.datacamp.com/tutorial/feature-selection-python

Referencias

- “Introducción a las funciones de activación en las redes neuronales”. Consultado: el 13 de mayo de 2025. [En línea]. Disponible en: https://www.datacamp.com/tutorial/introduction-to-activation-functions-in-neural-networks [
- “Propagación hacia delante en redes neuronales: Guía completa”. Consultado: el 13 de mayo de 2025. [En línea]. Disponible en: https://www.datacamp.com/tutorial/forward-propagation-neural-networks
- “Redes Neuronales”. Consultado: el 13 de mayo de 2025. [En línea]. Disponible en: https://rstudio-pubs-static.s3.amazonaws.com/599210_4306c5d3d6a34a6c829566ca11b7e27a.html#11
- “¿Quién es quién en el sistema nervioso?”, neuronas en crecimiento. Consultado: el 13 de mayo de 2025. [En línea]. Disponible en: https://neuropediatra.org/2013/06/15/quien-es-quien-en-el-sistema-nervioso/
- Administrador, ? “Reconocimiento Facial ? | Python - OpenCV » omes-va.com”, OMES. Consultado: el 13 de mayo de 2025. [En línea]. Disponible en: https://omes-va.com/reconocimiento-facial-python-opencv/
- H. Palomares, “¿Para qué sirven las redes neuronales?”, Innovación con Hilmer. Consultado: el 13 de mayo de 2025. [En línea]. Disponible en: https://hilmer.vip/redes-neuronales/
- admin, “Cómo Funcionan las Redes Neuronales: Explicación Detallada”. Consultado: el 10 de mayo de 2025. [En línea]. Disponible en: https://data-universe.org/como-funcionan-las-redes-neuronales-explicacion-detallada/
- “¿Qué es una red neuronal y cómo funciona?”, Google Cloud. Consultado: el 10 de mayo de 2025. [En línea]. Disponible en: https://cloud.google.com/discover/what-is-a-neural-network
- “¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS”, Amazon Web Services, Inc. Consultado: el 10 de mayo de 2025. [En línea]. Disponible en: https://aws.amazon.com/es/what-is/neural-network/
- “¿Qué es una red neuronal? | IBM”. Consultado: el 10 de mayo de 2025. [En línea]. Disponible en: https://www.ibm.com/es-es/think/topics/neural-networks
- Daniel, “Perceptrón: ¿qué es y para qué sirve?”, DataScientest. Consultado: el 10 de mayo de 2025. [En línea]. Disponible en: https://datascientest.com/es/perceptron-que-es-y-para-que-sirve
- “Guía para principiantes sobre la unidad lineal rectificada (ReLU)”. Consultado: el 10 de mayo de 2025. [En línea]. Disponible en: https://www.datacamp.com/blog/rectified-linear-unit-relu
- “La Función de Activación”, Codificando Bits. Consultado: el 10 de mayo de 2025. [En línea]. Disponible en: https://codificandobits.com/blog/funcion-de-activacion/
- “¿Cuáles son las partes del sistema nervioso? | NICHD Español”. Consultado: el 10 de mayo de 2025. [En línea]. Disponible en: http://espanol.nichd.nih.gov/salud/temas/neuro/informacion/partes
- “scikit-learn: Machine Learning in Python”. Consultado: el 20 de febrero de 2025. [En línea]. Disponible en: https://scikit-learn.org/stable/
- Información e ideas presentadas basadas en el conocimiento general de modelos de lenguaje de IA. Gemini 2.9 Flash. Consultado: el 9 de mayo de 2025. [En línea].



Machine Learning

Susana Medina Gordillo

susana.medina@correounivalle.edu.co