Clase de Prueba: Redes Neuronales

M.Sc. Ing. Victor Hugo Borrayo Herrera

October 24, 2025

CLASE DE PRUEBA

Índice

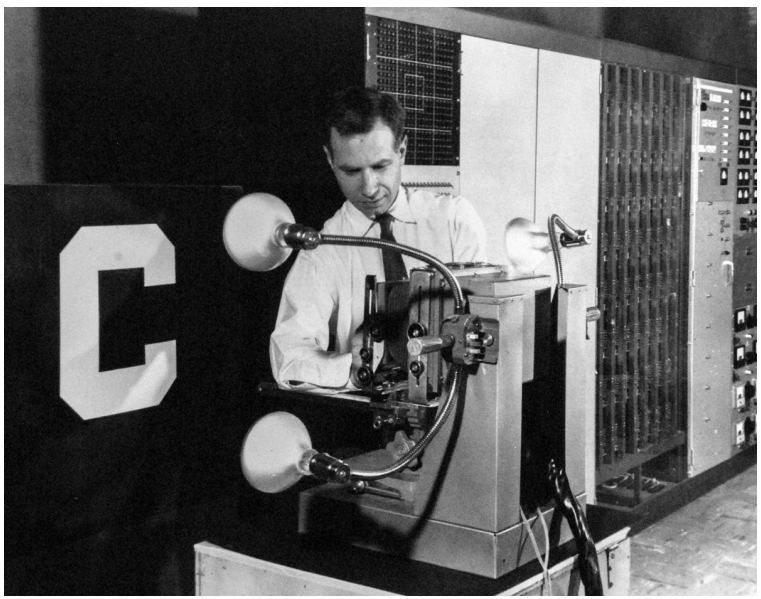
1. El Perceptrón y los Kernels
2. Redes Neuronales Artificiales (ANN)

Repositorio

- Asegura reproducibilidad y acceso al código fuente.
- Permite la colaboración y contribuciones de los estudiantes.

El origen del Perceptrón

- En 1958, Frank Rosenblatt propone el **Perceptrón**, inspirado en el funcionamiento de las neuronas biológicas.
- Representa el primer modelo matemático funcional de una **neurona artificial**.
- Su objetivo era aprender a clasificar patrones mediante un proceso de entrenamiento.



Definición matemática del Perceptrón

- El modelo recibe un vector de entrada:

$$\mathbf{x} = (x_1, x_2, \ldots, x_n)$$

- Cada entrada tiene un peso asociado:

$$\mathbf{w} = (w_1, w_2, \ldots, w_n)$$

- La salida se obtiene aplicando una función de activación a la combinación lineal:

$$y = f(\mathbf{w} \cdot \mathbf{x} + b)$$

- En el Perceptrón original:

$$f(z) = \begin{cases} 1 & \text{si } z \geq 0, \\ 0 & \text{si } z < 0 \end{cases}$$

Algoritmo del Perceptrón

1. Inicializar pesos w_i y sesgo b con valores pequeños aleatorios.
2. Para cada muestra de entrenamiento $(\mathbf{x}_i, y_i)$:
   1. Calcular la salida predicha:

$$\hat{y}_i = f(\mathbf{w} \cdot \mathbf{x}_i + b)$$

   2. Actualizar los parámetros:

$$\mathbf{w} \leftarrow \mathbf{w} + \eta(y_i - \hat{y}_i)\mathbf{x}_i$$

$$b \leftarrow b + \eta(y_i - \hat{y}_i)$$

3. Repetir hasta que el error sea mínimo o no existan cambios.

Donde: η es la tasa de aprendizaje.

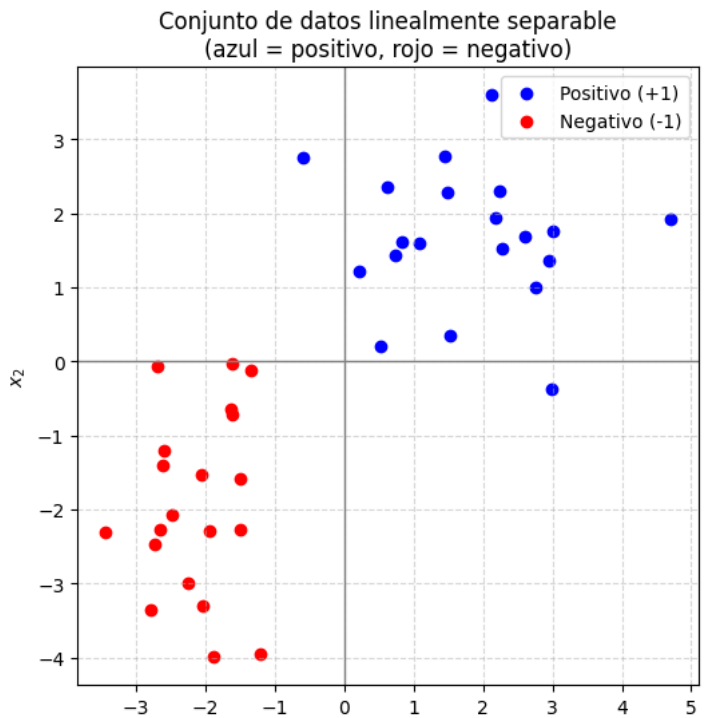
Análisis matemático del Perceptrón

- Cada muestra se representa como un punto en el plano.
- La frontera de decisión es el hiperplano:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

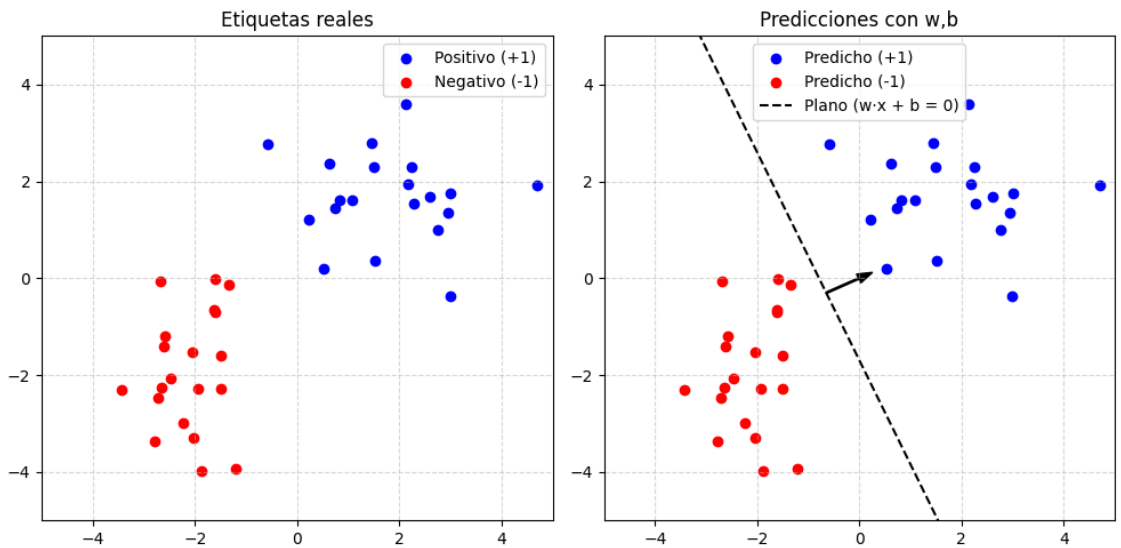
- El signo del resultado define la clase:

$$\text{signo}(\mathbf{w} \cdot \mathbf{x} + b)$$



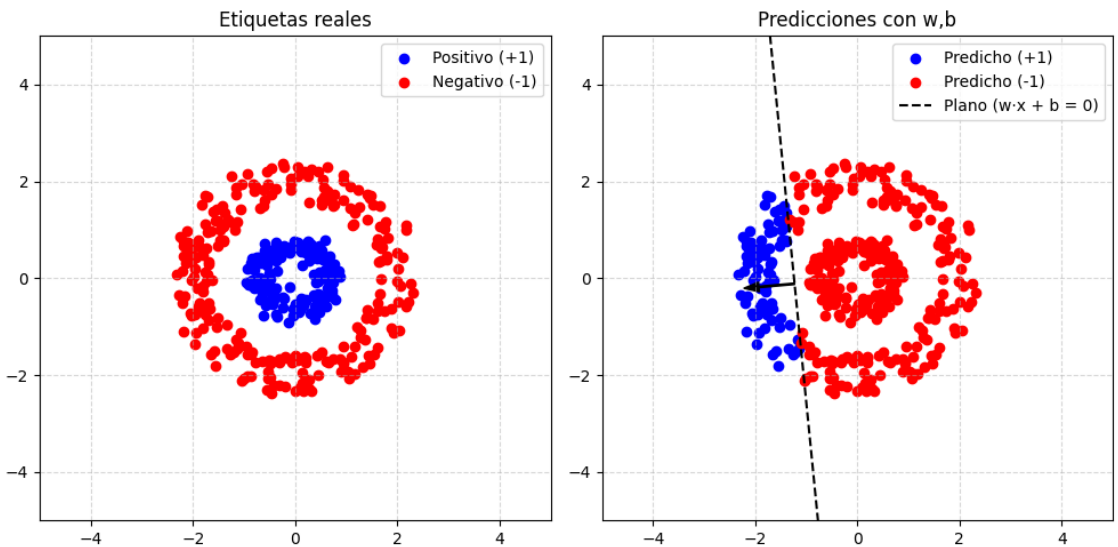
Resultado del entrenamiento

- Luego de varias iteraciones, los pesos convergen a valores que permiten separar las clases linealmente.
- La frontera aprendida define la región de decisión final.
- Este proceso demuestra la capacidad de aprendizaje lineal del Perceptrón.



Caso no linealmente separable

- No todos los conjuntos de datos pueden separarse mediante una frontera lineal.
- El Perceptrón falla en problemas como el clásico ejemplo de círculos concéntricos o XOR.



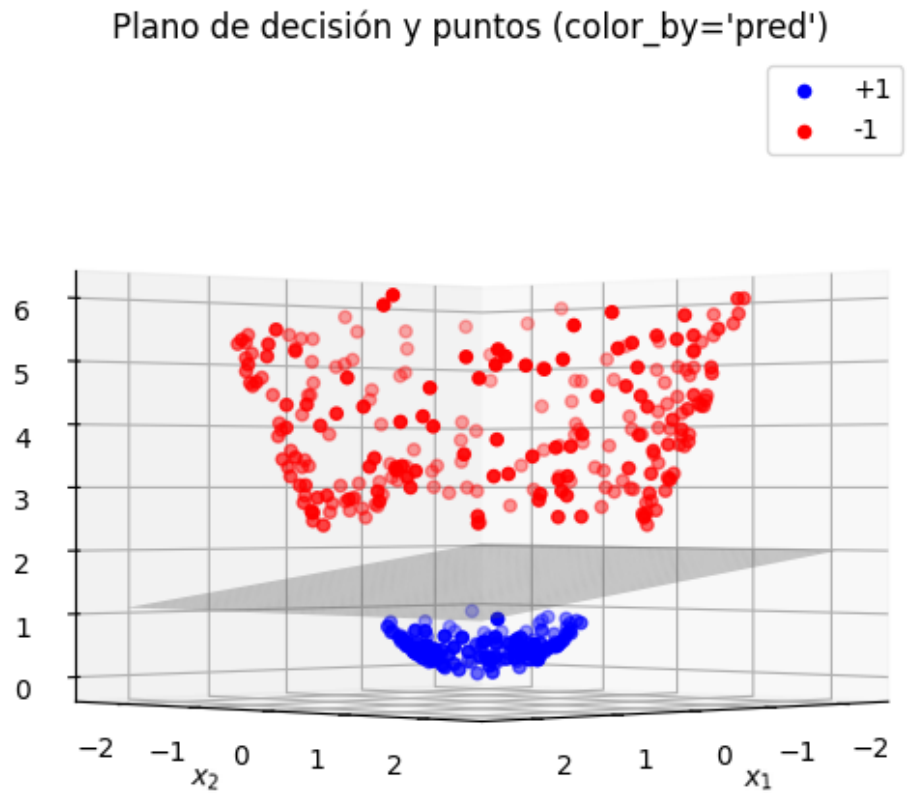
El concepto de Kernel

- Un **Kernel** transforma los datos a un espacio de mayor dimensión donde pueden separarse linealmente.
- En lugar de modificar el algoritmo, se transforma el espacio de representación.
- Esta idea dio origen a los métodos de aprendizaje de gran poder, como las **Máquinas de Soporte Vectorial (SVM)**.

$$\phi : \mathbb{R}^n \to \mathbb{R}^m, \quad m > n$$

Separación en un espacio de mayor dimensión

- Tras aplicar la transformación $\phi(\mathbf{x})$, los datos se vuelven separables en el nuevo espacio.
- El plano de separación en ese espacio se obtiene al aplicar el algoritmo del perceptrón.

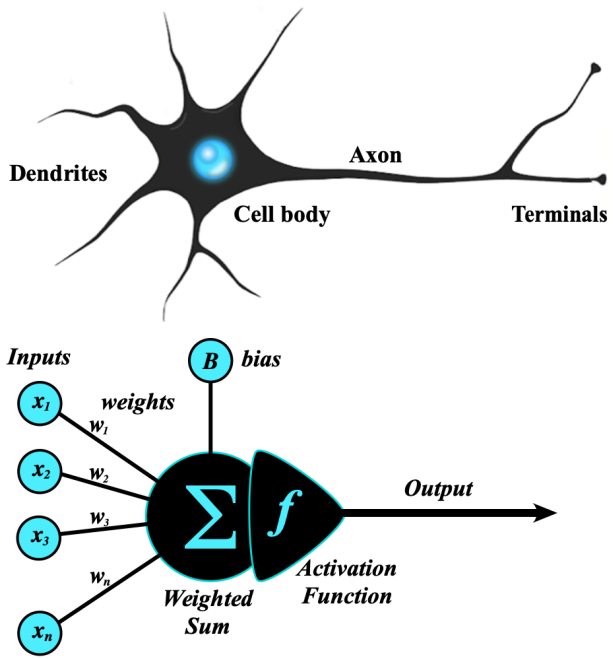


Historia de las Redes Neuronales

- Las redes neuronales modernas surgen como evolución directa del Perceptrón de Rosenblatt.
- En los años 80 se desarrolla el algoritmo de **retropropagación del error**, lo que permite el entrenamiento de redes con múltiples capas.
- En la década de 2010, con la disponibilidad de datos masivos y GPUs, las redes neuronales profundas impulsan el renacimiento del aprendizaje profundo.

Relación entre Perceptrón, Neurona Artificial y Neurona Biológica

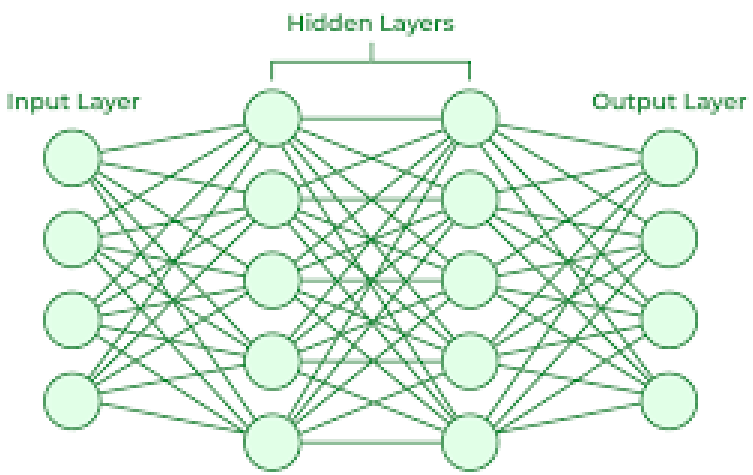
- El Perceptrón representa una versión simplificada de la neurona biológica.
- La neurona artificial combina entradas ponderadas, aplica una función de activación y produce una salida.
- El conjunto de neuronas artificiales interconectadas constituye una red neuronal.



¿Qué es una Red Neuronal?

- Una red neuronal es un modelo compuesto por capas de neuronas artificiales.
- Cada capa transforma la información proveniente de la capa anterior y transmite su resultado a la siguiente.
- Estas transformaciones permiten aprender representaciones jerárquicas de los datos.
- La salida final depende de todos los parámetros de la red:

$$\hat{y} = f(\mathbf{x}; \mathbf{W}, \mathbf{b})$$



Algoritmo de una Red Neuronal

- **Forward propagation:** calcula las salidas de cada capa desde la entrada hasta la salida final.

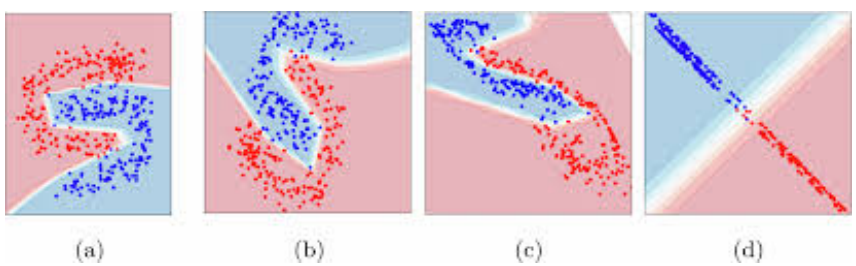
$$\mathbf{a}^{(l)} = f(W^{(l)}\mathbf{a}^{(l-1)} + b^{(l)})$$

- **Función de pérdida:** mide la diferencia entre la salida predicha y el valor real.

$$\mathcal{L}(y, \hat{y})$$

- **Backpropagation:** propaga el error hacia atrás y actualiza los parámetros usando el gradiente de la pérdida.

$$W^{(l)} \leftarrow W^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial W^{(l)}}$$

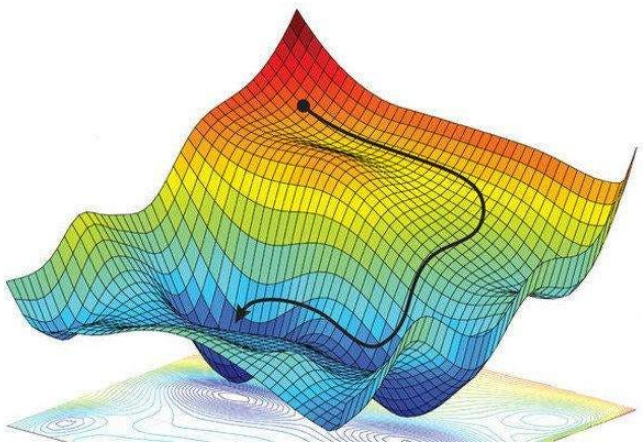


Optimización con Stochastic Gradient Descent (SGD)

- El descenso del gradiente busca minimizar la función de pérdida modificando los parámetros de la red.
- El método estocástico utiliza lotes pequeños de datos para acelerar el proceso y evitar mínimos locales.
- Cada iteración ajusta los pesos en la dirección del gradiente negativo:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$$

- Este proceso continúa hasta alcanzar una configuración óptima de la red.



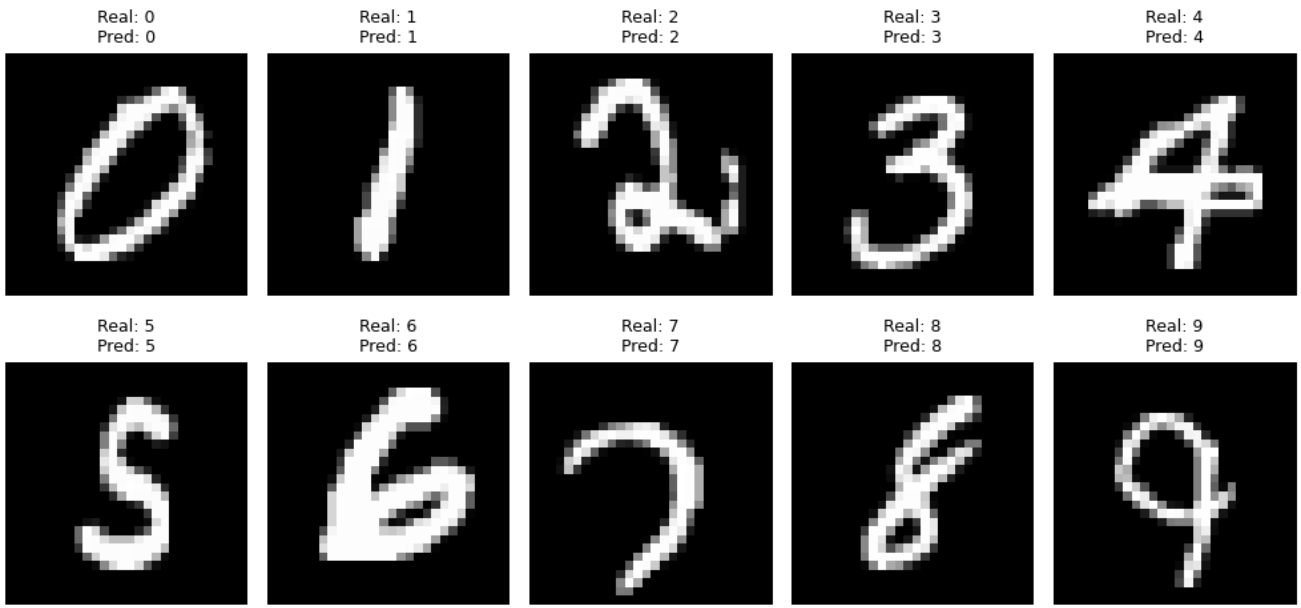
TensorFlow y el entrenamiento de redes neuronales

- **TensorFlow** es una biblioteca desarrollada por Google para construir y entrenar redes neuronales.
- Permite la definición automática de grafos computacionales y la diferenciación automática.
- Facilita el entrenamiento eficiente de modelos en CPU, GPU y TPU.
- Su API de alto nivel (`tf.keras`) simplifica la creación de redes neuronales profundas.

$$\text{modelo} = \texttt{tf.keras.Sequential([...])}$$

Resultados del modelo con MNIST

- El conjunto de datos **MNIST** contiene imágenes de dígitos escritos a mano.
- Una red neuronal completamente conectada puede clasificar correctamente más del 97% de los casos. (Más del 99% con redes convolucionales)
- Este resultado demuestra la capacidad de generalización de las redes neuronales.



¡Gracias!

Ahora veamos cómo se implementan estos algoritmos con código.