

## SESIÓN LA RED NEURONAL ARTIFICIAL

### CONTENIDOS:

- Qué es una red neuronal artificial.
- Problemas que se pueden resolver con redes neuronales.
- Elementos de una red neuronal:
  - a. El perceptrón.
  - b. Capas de entrada, salida, capas ocultas.
  - c. Función de activación.
  - d. Pesos y sesgos.
- Arquitectura de una red neuronal densa.
- Entrenamiento de una red neuronal: loss-function.
- Forward pass y Back propagation.

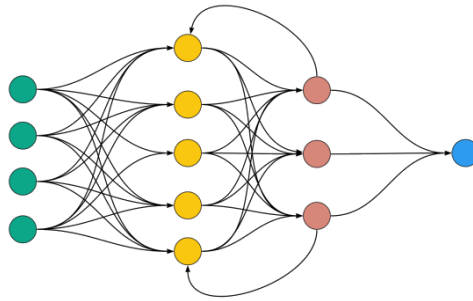
### QUÉ ES UNA RED NEURONAL ARTIFICIAL

Una **red neuronal artificial (RNA)** es un modelo computacional inspirado en la estructura y el funcionamiento del cerebro humano. Está compuesta por **neuronas artificiales**, organizadas en capas, que procesan información mediante conexiones ponderadas. Su objetivo es aprender patrones a partir de datos, ajustando los pesos de las conexiones a través de un proceso de entrenamiento.

Las redes neuronales se componen de tres tipos de capas principales:

- **Capa de entrada:** recibe los datos de entrada.
- **Capas ocultas:** procesan la información mediante funciones de activación.
- **Capa de salida:** genera la respuesta final del modelo.

El aprendizaje en una red neuronal se basa en la **retropropagación del error**, utilizando algoritmos como el **descenso del gradiente** para ajustar los pesos y mejorar la precisión del modelo.



*Ilustración 1 Esquema redes neuronales. OpenWebinars*

## PROBLEMAS QUE SE PUEDEN RESOLVER CON REDES NEURONALES

Las redes neuronales artificiales se utilizan para resolver una gran variedad de problemas en distintos campos, como:

1. **Clasificación:** asignación de etiquetas a datos (ej. detección de spam en correos electrónicos, reconocimiento de imágenes).
2. **Regresión:** predicción de valores continuos (ej. predicción de precios de viviendas, análisis financiero).
3. **Procesamiento de imágenes:** reconocimiento facial, segmentación de imágenes médicas, detección de objetos.
4. **Procesamiento de lenguaje natural (NLP):** traducción automática, análisis de sentimientos, chatbots.
5. **Series temporales y predicciones:** predicción del clima, detección de fraudes, análisis de tendencias de mercado.
6. **Sistemas de recomendación:** recomendaciones de productos en plataformas como Netflix, Amazon o Spotify.
7. **Control y automatización:** conducción autónoma, robótica, sistemas de control en la industria.

## ELEMENTOS DE UNA RED NEURONAL

Las redes neuronales artificiales están compuestas por varios elementos fundamentales que permiten su funcionamiento y aprendizaje. Entre ellos se destacan:

### El perceptrón

El **perceptrón** es la unidad básica de una red neuronal artificial. Fue desarrollado por **Frank Rosenblatt** en 1958 y representa el modelo más simple de una neurona artificial. Su funcionamiento se basa en los siguientes pasos:

1. **Recepción de entradas:** recibe múltiples valores de entrada  $x_1, x_2, \dots, x_n$ .
2. **Asignación de pesos:** cada entrada tiene un peso asociado  $w_1, w_2, \dots, w_n$  que indica su importancia.
3. **Cálculo de la suma ponderada:** se multiplica cada entrada por su peso y se suma con un valor adicional llamado **sesgo** ( $b$ ).
4. **Aplicación de la función de activación:** el resultado se pasa por una función matemática que determina si la neurona se activa o no.
5. **Salida del perceptrón:** se obtiene un valor de salida  $y$ .

**Fórmula del perceptrón:**

$$y = f \left( \sum (w_i \cdot x_i) + b \right)$$

*Ilustración 2 Fórmula del perceptrón*

Donde  $f$  es la función de activación.

## Capas en una Red Neuronal

Las redes neuronales están organizadas en distintas capas:

1. **Capa de entrada:** recibe los datos de entrada y los transmite a la siguiente capa. No realiza cálculos, solo distribuye los valores.
2. **Capas ocultas:** son capas intermedias entre la entrada y la salida. Aquí ocurre el procesamiento mediante la combinación de pesos, sesgos y funciones de activación. A mayor número de capas ocultas, más complejidad y capacidad de aprendizaje tiene la red.
3. **Capa de salida:** genera la respuesta final del modelo. El número de neuronas en esta capa depende del problema; por ejemplo, en clasificación binaria hay una sola neurona, mientras que en clasificación multiclase hay tantas neuronas como categorías.

### Modelo Multicapa ( $L$ capas)

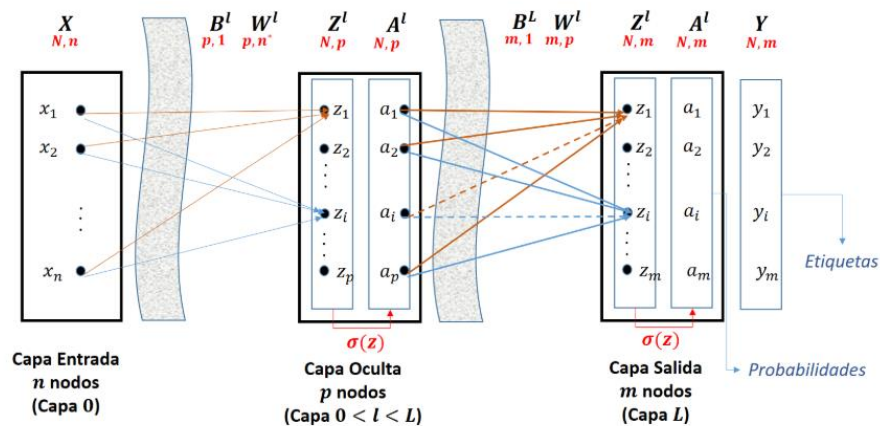


Ilustración 3 Modelo Multicapa. Universidad Politécnica de Madrid

## Función de activación

Las funciones de activación son operaciones matemáticas que introducen no linealidad en la red, permitiendo resolver problemas complejos. Algunas de las más utilizadas son:

- **Sigmoide:** útil en problemas de clasificación binaria.

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

*Ilustración 4 Función sigmoide*

- **ReLU (Rectified Linear Unit):** se usa en redes profundas debido a su eficiencia computacional.

$$f(x) = \max(0, x)$$

*Ilustración 5 Función ReLu*

- **Tangente hiperbólica (tanh):** similar a la sigmoide, pero con valores entre -1 y 1.

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

*Ilustración 6 Función Tangente hiperbólica*

- **Softmax:** ideal para clasificación multiclase, ya que convierte los valores en probabilidades.

### **Pesos y sesgos**

Los **pesos (w)** determinan la importancia de cada conexión en la red. Se ajustan durante el entrenamiento para mejorar la precisión del modelo.

El **sesgo (b)** es un valor adicional que permite desplazar la función de activación, lo que ayuda a la red a aprender patrones más complejos.

## ARQUITECTURA Y ENTREGAMIENTO DE UNA RED NEURONAL

Las redes neuronales artificiales requieren una estructura bien definida y un proceso de entrenamiento que les permita aprender de los datos. A continuación, se explican los conceptos clave:

### Arquitectura de una red neuronal densa

Una **red neuronal densa** (o totalmente conectada) es un tipo de red en la que cada neurona de una capa está conectada a todas las neuronas de la capa siguiente. Su estructura básica incluye:

1. **Capa de entrada:** recibe los datos iniciales y los transmite sin modificaciones.
2. **Capas ocultas:** procesan los datos aplicando pesos, sesgos y funciones de activación.
3. **Capa de salida:** genera el resultado final del modelo.

Cada neurona en una capa toma las salidas de la capa anterior, les aplica pesos y sesgos, y las pasa por una función de activación.

Ejemplo de una red neuronal densa con **tres capas**:

- **Entrada:** 3 neuronas (por ejemplo, características de una imagen: altura, ancho, color).
- **Ocultas:** 4 neuronas con activación ReLU.
- **Salida:** 2 neuronas con activación Softmax (por ejemplo, clasificación en dos clases).

**Representación matemática de una capa densa:**

$$h = f(W \cdot X + b)$$

*Ilustración 7 Representación matemática capa densa*

donde:

- W son los pesos,
- X son las entradas,
- b es el sesgo,
- f es la función de activación.

### Entrenamiento de una red neuronal

El entrenamiento de una red neuronal es el proceso mediante el cual el modelo ajusta sus pesos y sesgos para minimizar el error en sus predicciones. Se lleva a cabo en tres fases principales:

1. **Loss Function** (Función de Pérdida)
2. **Forward Pass** (Propagación hacia adelante)
3. **Backpropagation** (Retropropagación del error)

### Loss-function (Función de Pérdida)

La **función de pérdida** es una medida del error entre la salida predicha por la red  $\hat{y}$  y el valor real  $y$ . Su objetivo es proporcionar una métrica que se pueda minimizar durante el entrenamiento.

### Tipos de funciones de pérdida

- **Para problemas de regresión:**
  - **Error cuadrático medio (MSE)**

$$L = \frac{1}{n} \sum (y_{\text{real}} - y_{\text{predicho}})^2$$

*Ilustración 8 Error cuadrático medio*

- **Error absoluto medio (MAE)**

$$L = \frac{1}{n} \sum |y_{\text{real}} - y_{\text{predicho}}|$$

*Ilustración 9 Error absoluto medio*

➤ **Para problemas de clasificación:**

- **Entropía cruzada binaria (para dos clases)**

$$L = -\frac{1}{n} \sum (y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

*Ilustración 10 Entropía cruzada binaria*

- **Entropía cruzada categórica (para múltiples clases)**

$$L = -\sum y_i \log(\hat{y}_i)$$

*Ilustración 11 Entropía cruzada categórica*

El objetivo del entregamiento es minimizar esta función de pérdida, ajustando los pesos de la red neuronal.



### Forward pass (Propagación hacia adelante)

El forward pass es la fase en la que los datos de entrada pasan a través de la red hasta producir una salida.

#### Pasos del forward pass:

1. Se toma una muestra de datos de entrada  $X$ .
2. Se multiplica por los pesos  $W$  y se suma el sesgo  $b$ .
3. Se aplica una **función de activación**  $f$ .
4. Se obtiene la salida  $\hat{y}$  de la red.

#### Ejemplo matemático de una sola capa:

$$\hat{y} = f(WX + b)$$

*Ilustración 12 Ejemplo 1 sola capa*


Cada capa de la red transforma los datos hasta llegar a la predicción final.

### Backpropagation (Retropropagación del error)

Después del forward pass, la red compara su salida  $\hat{y}$  con la salida real  $y$  y calcula el error utilizando la función de pérdida. **Backpropagation** es el proceso que permite ajustar los pesos para minimizar este error.

#### Pasos de backpropagation:

1. Se calcula la **derivada del error** respecto a cada peso (gradientes).
2. Se propaga este error **de la capa de salida a la capa de entrada**.
3. Se ajustan los pesos con el **descenso de gradiente**:


$$W_{\text{nuevo}} = W - \eta \cdot \frac{\partial L}{\partial W}$$

*Ilustración 13 Descenso de gradiente*

donde  $\eta$  es la **tasa de aprendizaje**.

Este proceso de forward pass + backpropagation se repite durante muchas **épocas** hasta que la red aprende correctamente.