

# Funciones de Activación y Redes Neuronales

Andrés Martínez Vargas  
Instituto Tecnológico de Costa Rica  
martinezandres@estudiantec.cr

**Abstract**—Este documento presenta una revisión integral de las funciones de activación utilizadas en redes neuronales, así como conceptos clave relacionados con la arquitectura y evolución de las redes neuronales artificiales. Se detallan funciones como ReLU, Sigmoides, Softmax y otras, incluyendo su justificación matemática, así como una introducción al perceptrón, redes multicapa y problemáticas como el problema XOR y la maldición de la dimensionalidad.

## I. RESPUESTAS DE QUIZ

### A. Fórmulas de la Normalización y la Estandarización

#### Normalización (Min-Max Scaling):

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (1)$$

#### Estandarización (Z-Score Scaling):

$$x' = \frac{x - \mu}{\sigma} \quad (2)$$

Donde  $\mu$  es la media de los datos y  $\sigma$  la desviación estándar.

### B. La derivada de la función sigmoide

La función sigmoide es:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

Su derivada es:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \quad (4)$$

### C. Fórmulas para calcular el Accuracy, Recall, Precision y F1-Score

- **Accuracy:**  $\frac{TP + TN}{TP + TN + FP + FN}$
- **Recall (Sensibilidad):**  $\frac{TP}{TP + FN}$
- **Precision:**  $\frac{TP}{TP + FP}$
- **F1-Score:**  $2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$

Donde: TP = Verdaderos Positivos, TN = Verdaderos Negativos, FP = Falsos Positivos, FN = Falsos Negativos.

### D. Desarrollo de la log-verosimilitud en 3 pasos

- 1) Se parte de la función de verosimilitud para clasificación binaria:

$$\mathcal{L}(\theta) = \prod_{i=1}^n \hat{y}_i^{y_i} (1 - \hat{y}_i)^{1-y_i} \quad (5)$$

- 2) Se aplica el logaritmo para obtener la log-verosimilitud:

$$\log \mathcal{L}(\theta) = \sum_{i=1}^n y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \quad (6)$$

- 3) Se cambia el signo negativo para obtener la función de pérdida logarítmica o entropía cruzada:

$$L = - \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (7)$$

## II. FUNCIONES DE ACTIVACIÓN

Las funciones de activación son fundamentales para introducir no linealidad en una red neuronal, permitiendo la aproximación de funciones complejas.

### A. Función Lineal

$$f(x) = x \quad (8)$$

Es diferenciable en todo punto, pero no introduce no linealidad, por lo que no es útil en redes profundas.

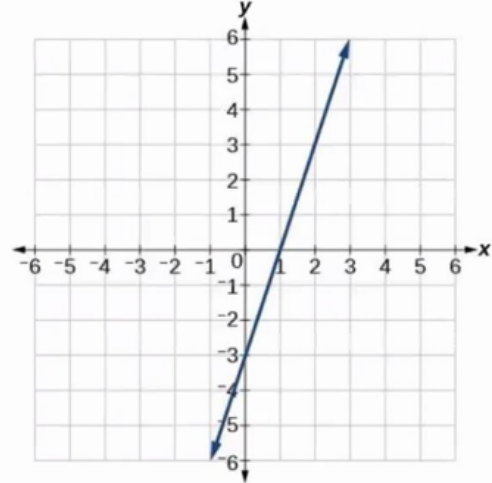


Fig. 1. Función de Activación Lineal

### B. Función Sigmoides

$$f(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

Limita el valor de salida entre 0 y 1, útil en modelos probabilísticos, pero sufre de *vanishing gradient*.

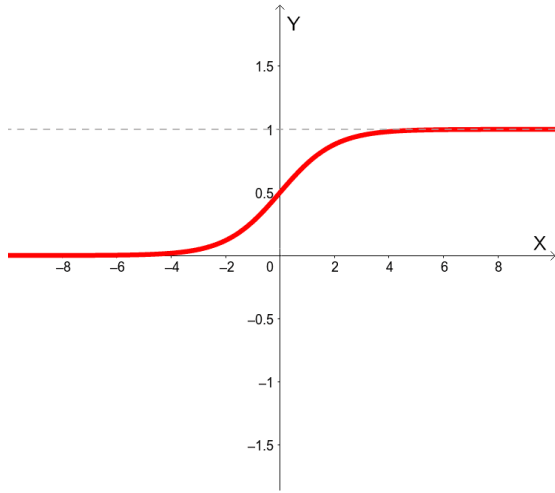


Fig. 2. Función Sigmoide

### C. Tangente Hiperbólica

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (10)$$

Similar a la sigmoide, pero centrada en cero. Puede sufrir también de *vanishing gradient*.

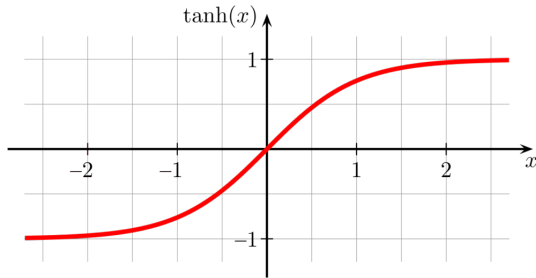


Fig. 3. Tangente Hiperbólica

### D. ReLU

$$f(x) = \max(0, x) \quad (11)$$

Simple y eficiente. Tiene el problema del “moribundo ReLU” si los gradientes son cero para valores negativos.

### E. Leaky ReLU

$$f(x) = \begin{cases} x, & \text{si } x > 0 \\ \alpha x, & \text{si } x \leq 0 \end{cases} \quad (12)$$

Con  $\alpha$  pequeño, típicamente 0.01, permite gradientes para valores negativos.

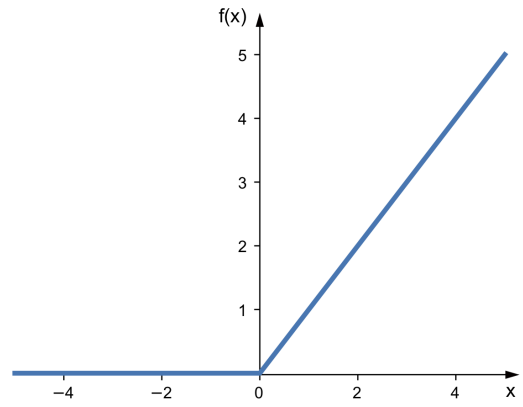


Fig. 4. Función ReLU

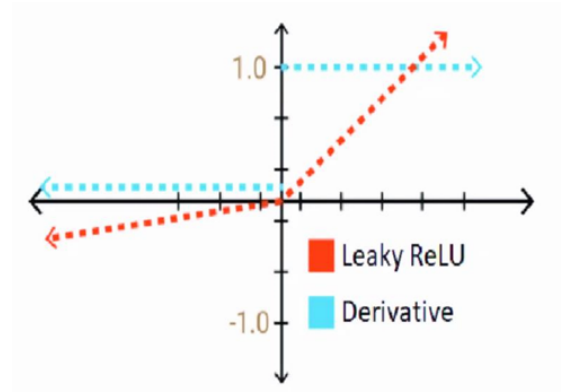


Fig. 5. Función Leaky ReLU

### F. Parametric ReLU (PReLU)

$$f(x) = \begin{cases} w, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (13)$$

Similar a Leaky ReLU pero con  $w$  como parámetro entrenable.

### G. Softmax

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (14)$$

Convierte vectores en distribuciones de probabilidad. Utilizada comúnmente en la capa de salida para clasificación múltiple.

### H. ¿Por qué $e^x$ ?

La exponencial  $e^x$  es continua, suave y nunca negativa, lo que la hace útil en funciones como la sigmoide y softmax por su capacidad de resaltar diferencias relativas.

### I. Cross-Entropy Loss

$$L = - \sum_{i=1}^n y_i \log(\hat{y}_i) \quad (15)$$

Función de pérdida común para clasificación. Mide la disimilitud entre las distribuciones verdadera  $y_i$  y predicha  $\hat{y}_i$ .

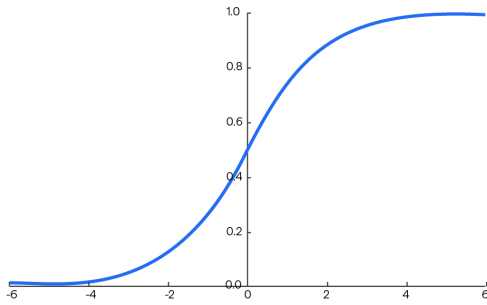


Fig. 6. Función Softmax

### III. REDES NEURONALES

#### A. El Perceptrón

Propuesto por Frank Rosenblatt, el perceptrón es un modelo lineal binario:

$$y = f\left(\sum w_i x_i + b\right) \quad (16)$$

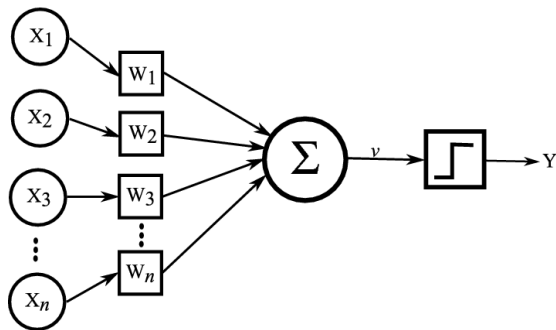


Fig. 7. Modelo de Perceptrón

#### B. Invierno de la IA

Periodo durante el cual se redujo el interés y financiamiento en IA debido a expectativas no cumplidas.

#### C. Predicción de Compuertas Lógicas

El perceptrón puede predecir compuertas AND, OR, pero no XOR.

#### D. Problema del XOR

No puede resolverse con modelos lineales. Requiere redes con múltiples capas.

#### E. Inspiración Biológica

Las redes neuronales se inspiran en la estructura del cerebro, con neuronas conectadas mediante sinapsis.

#### F. Función de Activación en Redes Neuronales

Introduce no linealidad, permite resolver problemas complejos.

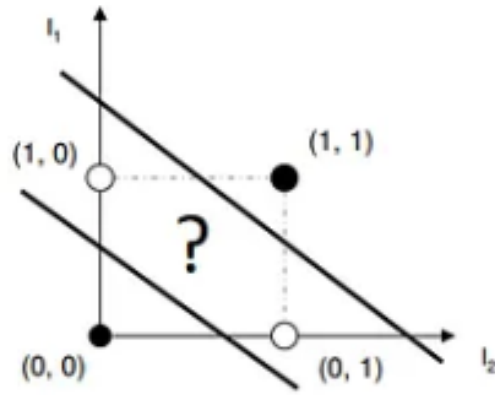


Fig. 8. Problema del XOR

#### G. Perceptrón Multicapa (MLP)

Extiende el perceptrón añadiendo capas ocultas. Permite modelar relaciones no lineales.

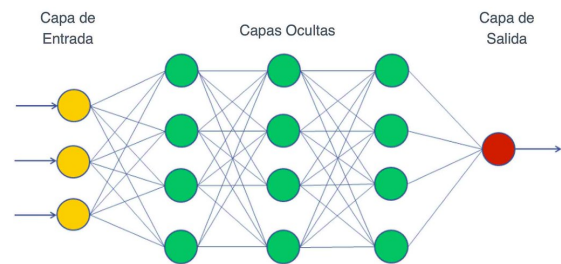


Fig. 9. Arquitectura de un Perceptrón Multicapa (MLP)

#### H. Salida Independiente y Distribución

Cada salida puede representar una variable diferente. La distribución puede ser categórica (softmax) o continua (regresión).

#### I. Capa de Salida

La función de activación depende del tipo de tarea (clasificación binaria, múltiple o regresión).

#### J. Función de Costo

Función matemática que mide el error del modelo. Se minimiza durante el entrenamiento.

#### K. Maldición de la Dimensionalidad

A medida que aumentan las dimensiones, los datos se dispersan, dificultando el aprendizaje.

#### L. Comportamiento Jerárquico

Las redes profundas aprenden representaciones desde lo simple (bordes) a lo complejo (formas, conceptos).

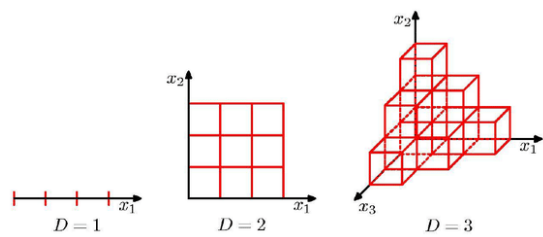


Fig. 10. Ejemplo de Maldición de la Dimensionalidad

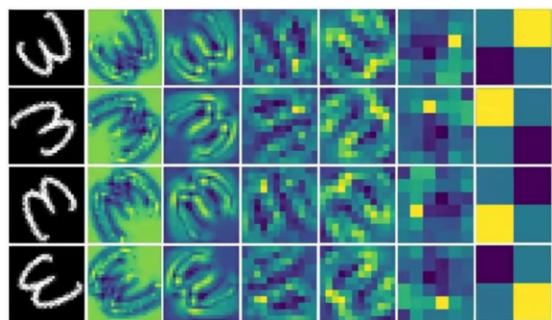


Fig. 11. Jerarquía de Representaciones en Redes Neuronales