

# Apuntes de Clase Inteligencia Artificial IC-6200

## Semana 8 — Jueves 04 de Abr del 2024

Profesor: MSc Steven Pacheco Portuguez

1<sup>nd</sup> Sebastián Chaves Zumbado  
*Ingeniería en Computación*  
*Tecnológico de Costa Rica*  
Cartago, Costa Rica  
sebas04@estudiantec.cr

### I. REDES NEURONALES

Las redes neuronales son modelos computacionales inspirados en la estructura y el funcionamiento del cerebro humano, diseñados para simular la manera en que los humanos aprenden. Estas se componen de nodos, o "neuronas", organizados en capas que procesan la información mediante conexiones ponderadas. Los pesos de estas conexiones se ajustan durante el proceso de aprendizaje, permitiendo a la red realizar tareas complejas, como clasificación y predicción, mediante la identificación de patrones en los datos de entrada.

### II. EL PERCEPTRON [1]

En 1957 Frank Rosenblatt inventó el perceptrón en el laboratorio aeronáutico de Cornell. Basándose en los primeros conceptos de neuronas artificiales, propuso la "regla de aprendizaje del perceptrón".

El perceptrón recibe múltiples señales de entrada. Si la suma de las señales supera un umbral determinado, se produce una señal o, por el contrario, no se emite ningún resultado. Si estructura se muestra en la Figura 1.

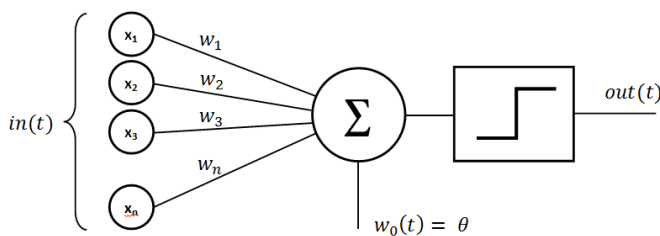


Fig. 1. Formula del perceptrón

Por primera vez, se tenía una máquina que podía aprender de la experiencia. El perceptrón fue diseñado para ajustar automáticamente sus pesos en respuesta a los errores de predicción, un concepto que era revolucionario para la época.

Sin embargo, la publicación del libro "Perceptrons" de Marvin Minsky y Seymour Papert en 1969 demostró

matemáticamente que los perceptrones simples no podían resolver problemas no linealmente separables, como la operación lógica XOR.

Esto fue un duro golpe para la comunidad de IA y redujo el entusiasmo y la financiación en la investigación de las redes neuronales durante años, en un período conocido como el "invierno de la IA".

### A. Redes Neuronales Multicapa [2]

La solución al problema fue implementar perceptrones anidados que consisten en múltiples capas de neuronas, con al menos una capa oculta entre la entrada y la salida. Cada neurona en una capa está conectada con todas las neuronas de la siguiente capa, formando una red densamente conectada.

Este proceso se inspiró biológicamente en como las neuronas de nuestro sistema están conectadas. Similitud representada en la Figura 2.

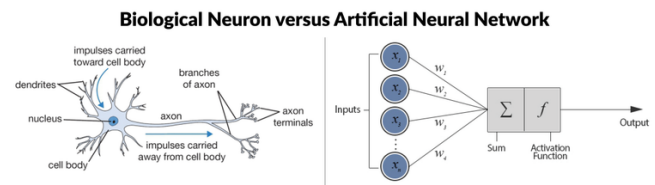


Fig. 2. Representación Nuerona — Red Neuronal Artificial

Similar a las dentritas una red neuronal recibe N entradas las cuáles son procesados por una función lineal y similar al axioma, la **función de activación** es la que de acuerdo al resultado deja pasar la información.

### III. DEFINICIÓN FUNCIÓN DE ACTIVACIÓN

Se le suele conocer como función no-lineal (sigmoid) en la regresión logística. Es la que se encuentra luego de la regresión lineal. Pueden existir varias funciones de activación.

Se utilizan principalmente para controlar el flujo del paso de la información si se le deja pasar o no, e incluso hacerle alguna transformación para que tenga otros valores que mejoren el

desempeño de la red. Nos permiten dar un comportamiento eficiente a la red y controlar el rango de salida.

#### IV. PERCEPTRÓN MULTICAPA (MLP) [3]

Es una multiplicación de matrices en donde todos los  $X$  (inputs) y los multiplico por los pesos  $W$ . Cada capa le pasa la información resultante a todas la siguientes capas como se presenta en la Figura 3.

**Capa de entrada (Input layer):** Su tamaño es la dimensión de  $X$ .

**Capas ocultas (Hidden Layers):** Pueden haber múltiples capas ocultas. Las capas ocultas toman su entrada de la capa de entrada o de otras capas ocultas. Las redes neuronales artificiales pueden tener una gran cantidad de capas ocultas. Cada capa oculta analiza la salida de la capa anterior, la procesa aún más y la pasa a la siguiente capa.

**Capa de Salida (Output Layer):** Es la última capa en la que termina el modelo. La capa de salida proporciona el resultado final de todo el procesamiento de datos que realiza la red neuronal artificial.

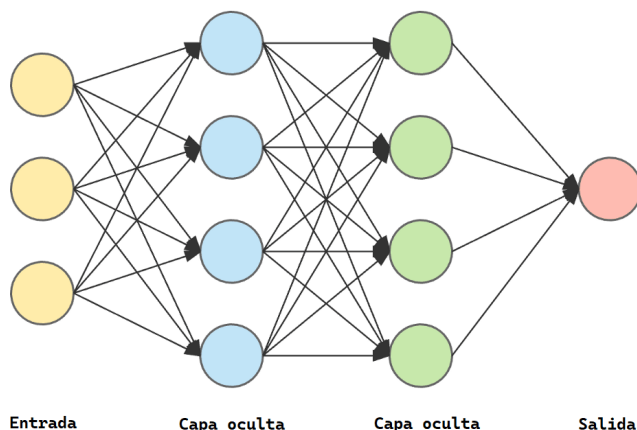


Fig. 3. Red Neuronal

1) *Maldición de la dimensionalidad* [4]: Cuando tenemos muchas dimensiones surgen problemas a nivel estadístico. Deseamos siempre que nuestro modelo sea lo más preciso posible, pero al tener tantas variables algunas serán características poco importantes y otras pueden ser meramente arbitrarias, esto lleva a que la precisión del modelo caiga

A mayor cantidad de dimensiones, aumenta la complejidad.

- **Problemas computacionales** Algoritmos de aprendizaje automático se vuelven más costosos computacionalmente a medida que aumenta la dimensionalidad. Esto puede hacer que la convergencia de los modelos se vuelvan más lentos.

- **Dificultades de Visualización:** La visualización de datos multidimensionales es compleja, lo que hace que sea más difícil explorar los datos y encontrar patrones.

2) *Cómo abordar la maldición de la dimensionalidad:*

- **Selección de características:** Reducir el número de dimensiones relevantes mediante técnicas como la selección de características o la reducción de dimensionalidad.

- **Regularización:** Aplicar técnicas de regularización para evitar el sobreajuste, como la regularización L1 o L2.

3) *Comportamiento Jerárquico:* El comportamiento jerárquico implica el procesamiento de información en capas, donde las capas elementales se ocupan de características básicas y operaciones de bajo nivel, ascendiendo gradualmente hacia capas de alto nivel que sintetizan conocimiento y realizan inferencias más complejas.

#### V. TIPOS DE FUNCIÓN DE ACTIVACIÓN

##### A. Función Lineal

Esta función no resulta muy útil debido a que no permite el cálculo de regresiones no lineales. En esta función el gradiente siempre va a ser el mismo ya que la derivada no depende de la entrada. Representación en Figura 4

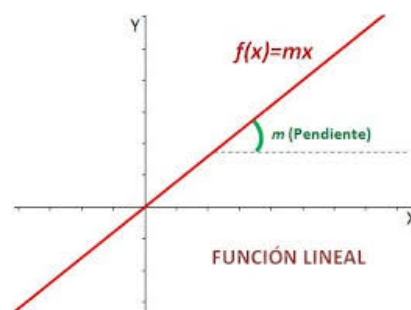


Fig. 4. Función Lineal y Derivada

1) *Sigmoide:* Se representa en la Figura 5.

- Rango:  $[0, 1]$
- Positiva, acotada, crecimiento estricto
- Es una función estrictamente creciente,

A pesar de sus ventajas, la función sigmoide puede presentar un problema con su pendiente, ya que puede ser muy cercana a cero en regiones lejanas de cero, lo que puede provocar que:

- El entrenamiento puede ser muy lento o en el peor de los casos este no converger.
- Vanishing gradient, lo cual es la principal por la que el entrenamiento se estanque.
- Gradiente pequeño al final de la función

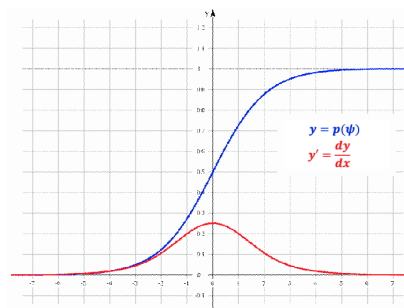


Fig. 5. Función Sigmoide y Derivada

### B. Tangente Hiperbólica

Se representa en la Figura 6.

- Rango de activación  $[-1,1]$
- Positivo y negativo
- Estrictamente creciente

**Ventajas:** mayor rango, simetría, menos propensa a vanishing gradient. \*Presenta los mismo problemas que la sigmoide.

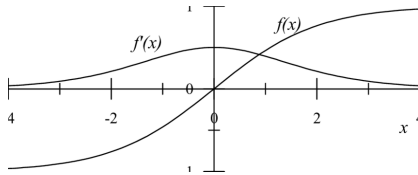


Fig. 6. Tangente Hiperbólica y Derivada

### C. Rectifier Linear Unit (ReLU)

- Acotada debajo de 0
- Estrictamente creciente
- Inactiva las neuronas con valores negativos, generando una salida cero.
- $g(x) = \max(x,0)$

Es muy eficiente en el Deep Learning. A pesar de acelerar la convergencia del entrenamiento gracias a su simplicidad, puede llevar a "neuronas muertas" durante el proceso si los pesos se ajustan de manera que la salida de la neurona sea siempre negativa. Se representa en la Figura 7.

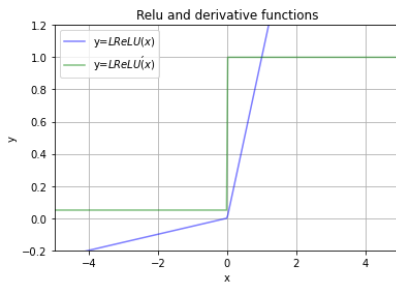


Fig. 7. Función ReLU y Derivada

### D. Leaky ReLU

Se representa en la Figura 8.

- Similar a ReLU, pero permite una pequeña pendiente positiva para valores negativos, evitando neuronas muertas.
- No está acotada y presenta una función creciente.
- Ayuda a mantener activo el flujo de gradientes durante el retroceso, lo que puede mejorar el entrenamiento en redes profundas.

Evita Dying ReLU, mantiene ventajas de ReLU

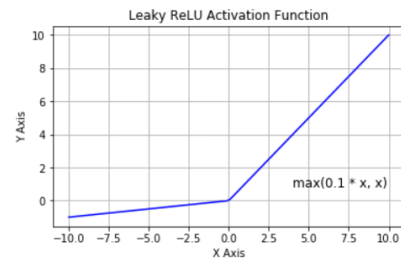


Fig. 8. Función Leaky ReLU

### E. Parametric ReLU

Se representa en la Figura 9.

- Es una variante de Leaky ReLU donde la pendiente para los valores negativos es aprendida durante el entrenamiento, lo que añade un parámetro ajustable a la neurona.
- No está acotada y retiene la propiedad creciente.
- Ofrece flexibilidad y puede adaptarse mejor a los datos específicos del problema.

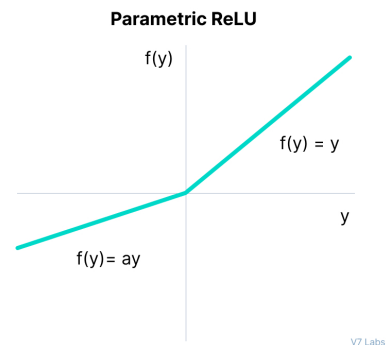


Fig. 9. Función Parametric ReLU

### F. Softmax

Softmax se emplea comúnmente en la última capa de los modelos de clasificación para convertir los valores de salida en probabilidades interpretables. Representada en la Figura 10.

- Produce un conjunto de valores que son todos positivos y cuya suma total es 1, lo cual equivale a una distribución de probabilidad entre varias categorías.
- A diferencia de funciones crecientes típicas, la salida de Softmax depende de la relación relativa entre las entradas debido a su naturaleza normalizadora.

## VI. CROSS-ENTROPY LOSS

Es una función de coste ampliamente utilizada en problemas de clasificación, especialmente cuando las salidas del modelo son probabilidades producidas por una función softmax en la última capa. Para un caso de clasificación multiclase, la entropía cruzada mide la diferencia entre la distribución de probabilidades predicha por el modelo y la distribución de probabilidad real.

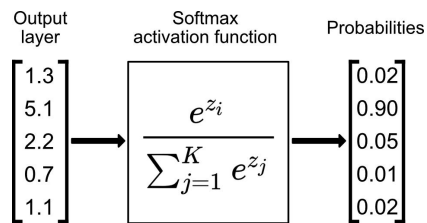


Fig. 10. Función Softmax

## REFERENCES

- [1] Daniel, "Perceptrón: ¿qué es y para qué sirve?," Formación en ciencia de datos — DataScientest.com, Mar. 07, 2022. <https://datascientest.com/es/perceptron-que-es-y-para-que-sirve> (accessed Apr. 12, 2024).
- [2] S. Jaiswal, "Multilayer Perceptrons in Machine Learning: A Comprehensive Guide," Datacamp.com, Feb. 07, 2024. <https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning> (accessed Apr. 12, 2024).
- [3] "¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS," Amazon Web Services, Inc., 2023. <https://aws.amazon.com/es/what-is/neural-network/> (accessed Apr. 12, 2024).
- [4] A. Landa, "La maldición de la dimensionalidad - Nicolás Arrijoja Landa Cosio - Medium," Medium, Oct. 11, 2021. <https://medium.com/@nicolasarrijoja/la-maldici>