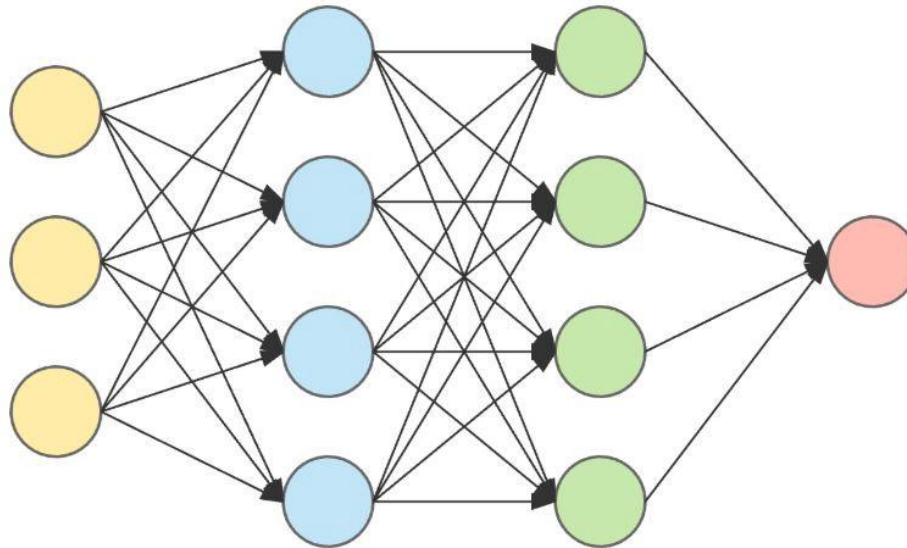


# Aprendizaje automático

## Introducción a las redes neuronales

# Redes neuronales artificiales

- Las redes neuronales artificiales (ANN) son modelos de aprendizaje automatizado **inspirados vagamente** por las redes de neuronas biológicas que constituyen nuestros cerebros.
- Una ANN está conformada por unidades computacionales llamadas **neuronas** interconectadas en una **red dirigida** (grafo dirigido).

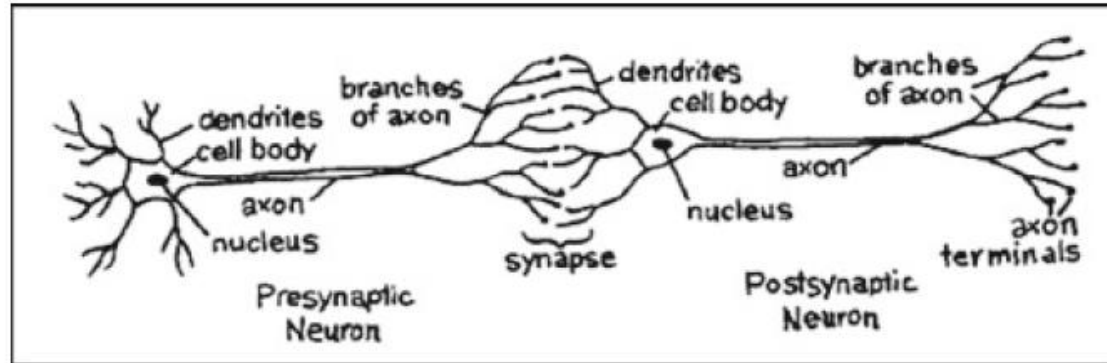


# Aplicaciones de redes neuronales

- Regresión
- Clasificación
- Agrupamiento
- Análisis de componentes y descomposición en componentes más simples
- Reducción de ruido y dimensionalidad
- Estimación de distribuciones de probabilidad
- Análisis de series de tiempo
- Sistemas dinámicos y de control
- Cualquier tarea de aprendizaje automatizado

# Inspiración biológica

- En cerebros biológicos, las neuronas están interconectadas unas con otras a través de **axones** y **dendritas**.

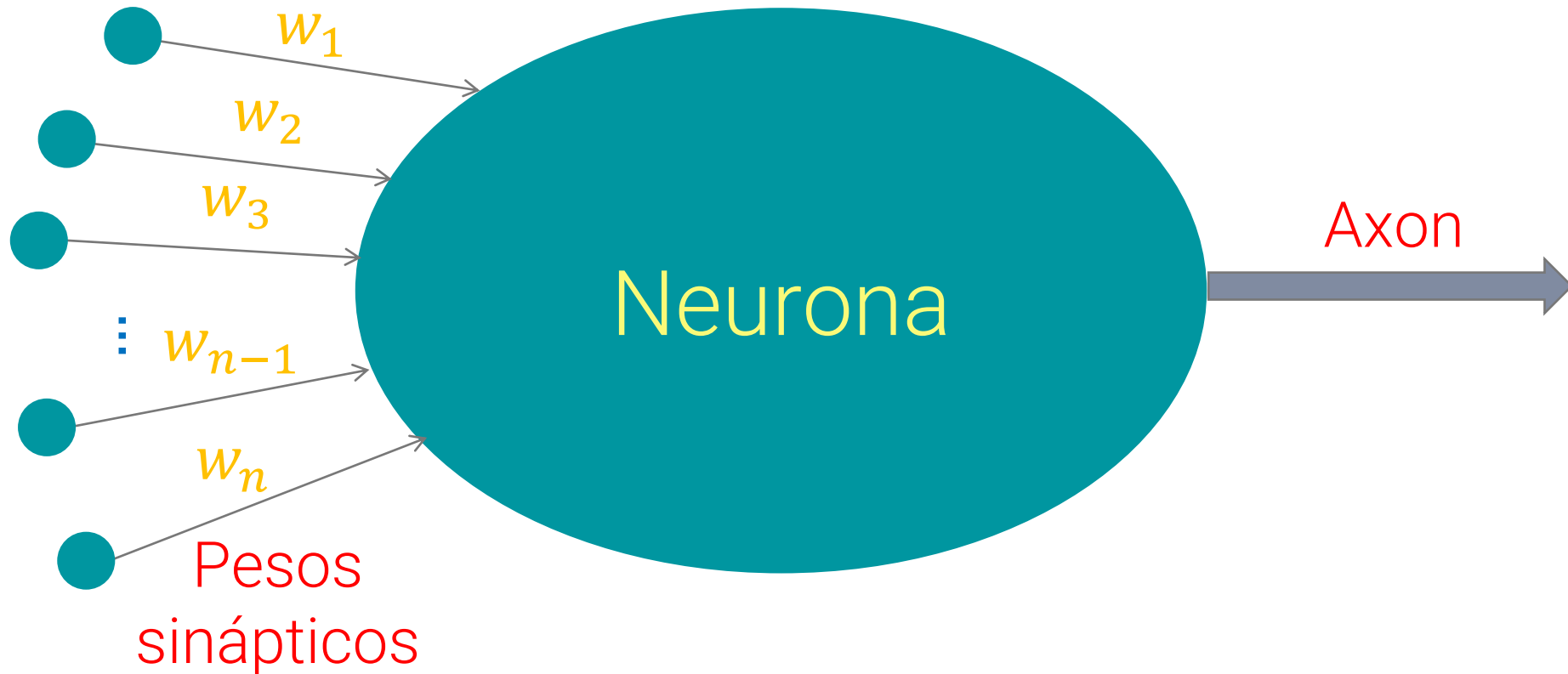


Aggarwal, 2023

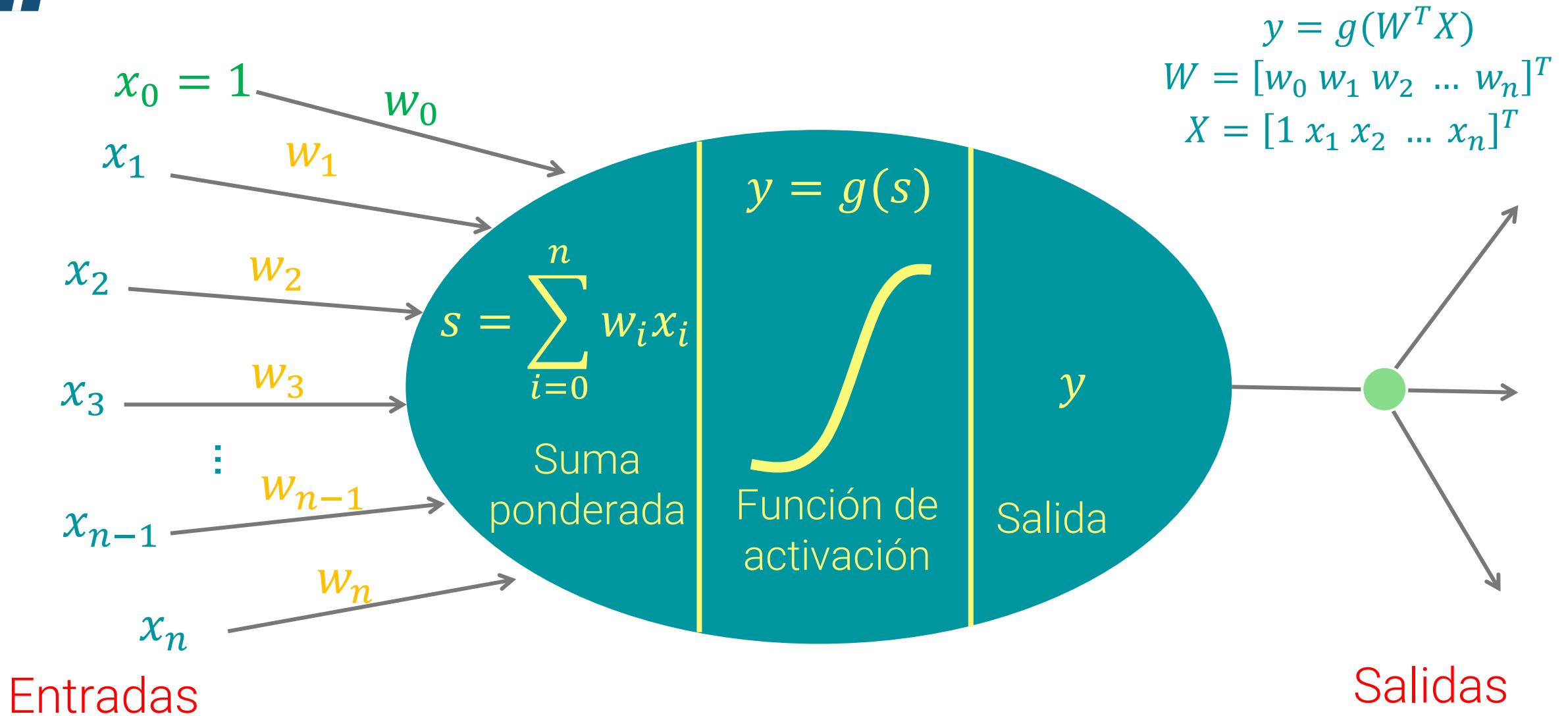
- Las **dendritas** son extensiones con muchas bifurcaciones. En estas estructuras una neurona recibe información de otras neuronas.
- El **axón** es una proyección similar a un cable que transmite señales nerviosas fuera de una neurona.
- Las **terminales del axón** (bifurcaciones del axón) se encuentran en el extremo del axón y contienen las sinapsis.
- Una **sinapsis** es una estructura que permite a una neurona transmitir señales eléctricas o químicas.

# Neuronas artificiales

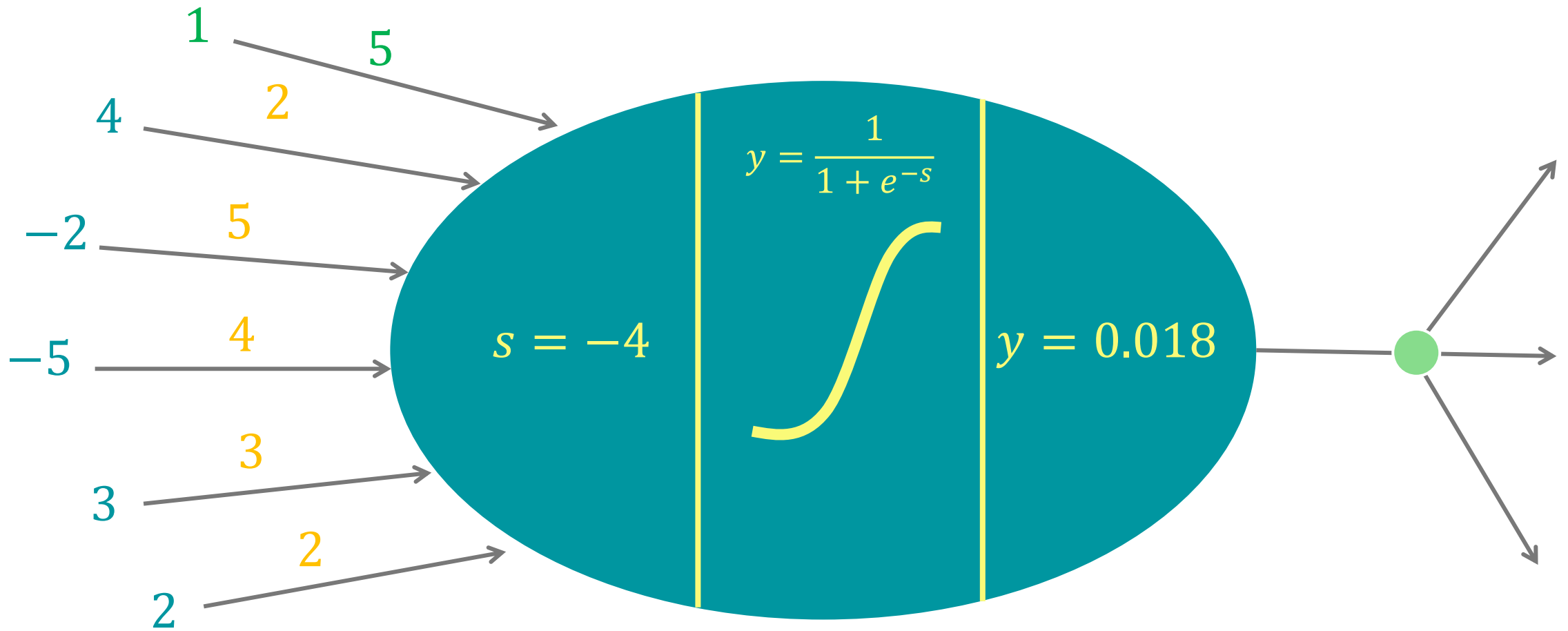
- En una red neuronal, típicamente una neurona es una **unidad de procesamiento** que transforma una **suma ponderada** de sus **entradas** para producir una respuesta de **salida**.



# Neuronas artificiales



# Neuronas artificiales



$$s = 1(5) + 4(2) - 2(5) - 5(4) + 3(3) + 2(2) = -4$$

# Neuronas artificiales

- Una simple neurona funciona como un modelo lineal.

$$y = g(w_0 + w_1x_1 + w_2x_2 \dots + w_nx_n)$$

- El **término constante** (**bias**) se incluye para representar el **intercepto** de un modelo lineal. Puede ser omitido en algunos casos, tal como se hace en el caso de regresión lineal.

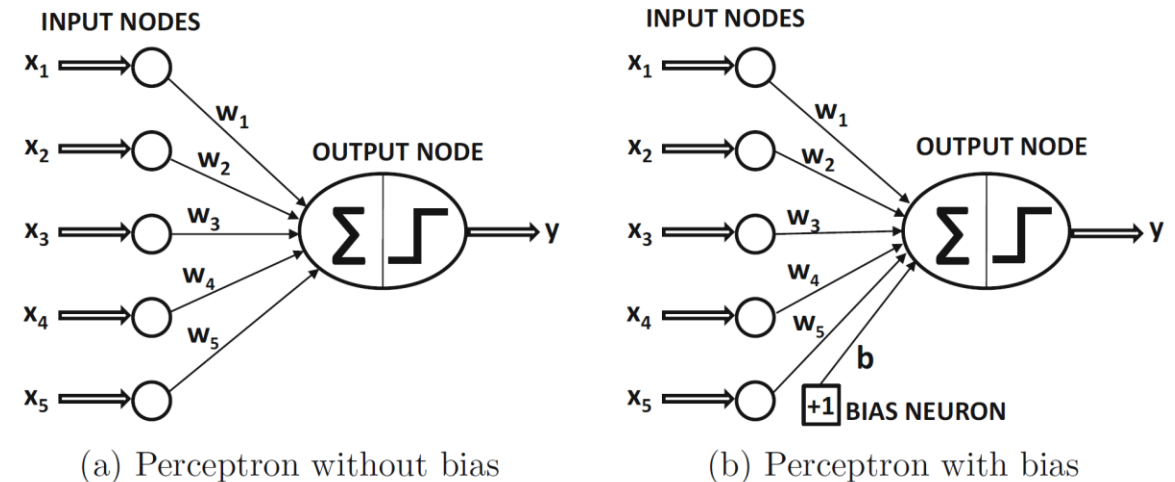


Figure 1.3: The basic architecture of the perceptron

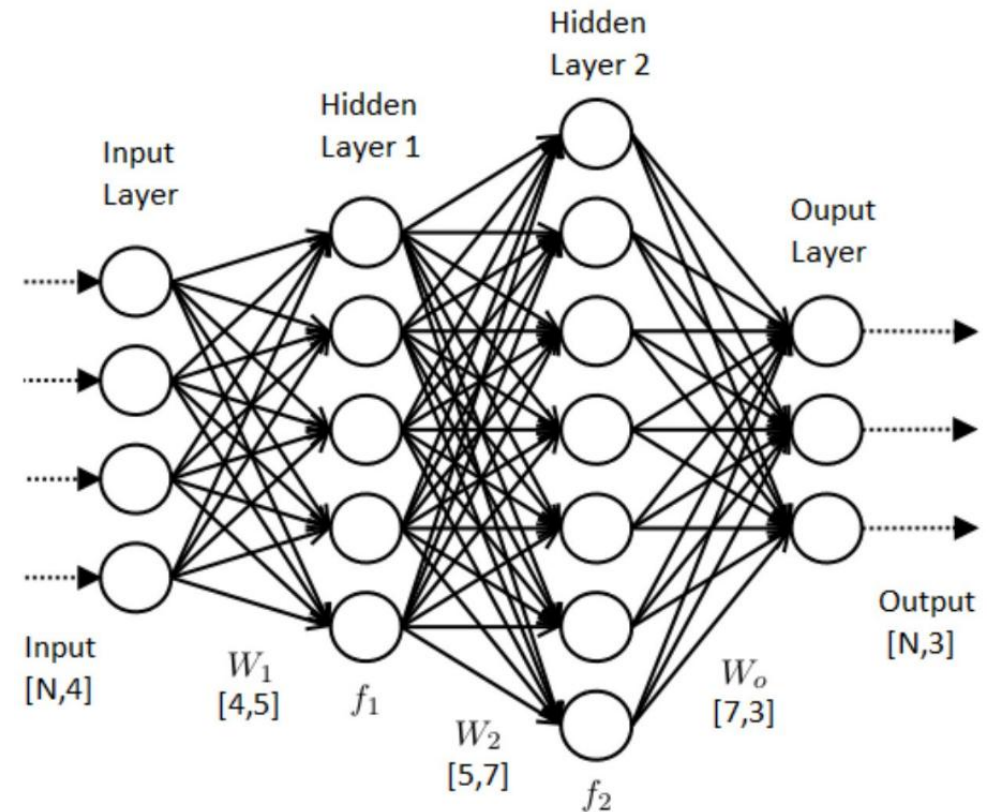
*Aggarwal, 2023*

- Si la función de activación es la **función umbral**, la neurona es llamada **perceptrón**, i.e., un perceptrón es un **clasificador lineal**.

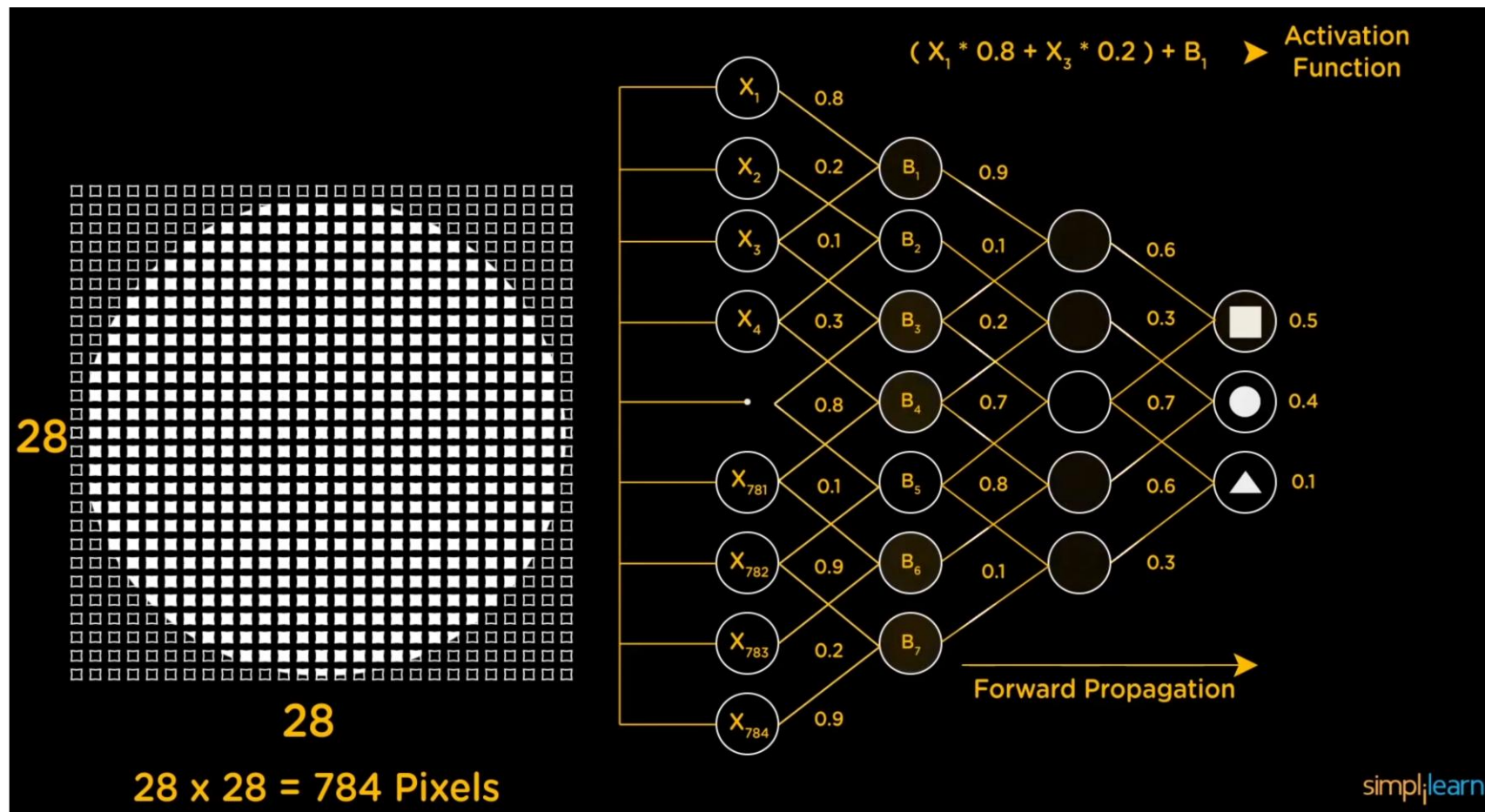


# Redes neuronales artificiales

- Una red neuronal es una colección de unidades de procesamiento.
- En la **capa de entrada**, se evalúan las variables de entrada (predictores).
- En las **capas ocultas**, se evalúan los resultados de las capas anteriores de manera secuencial.
- Se continúa con la evaluación de cada capa, hasta alcanzar las neuronas de salida (**capa de salida**).



# Redes neuronales



---

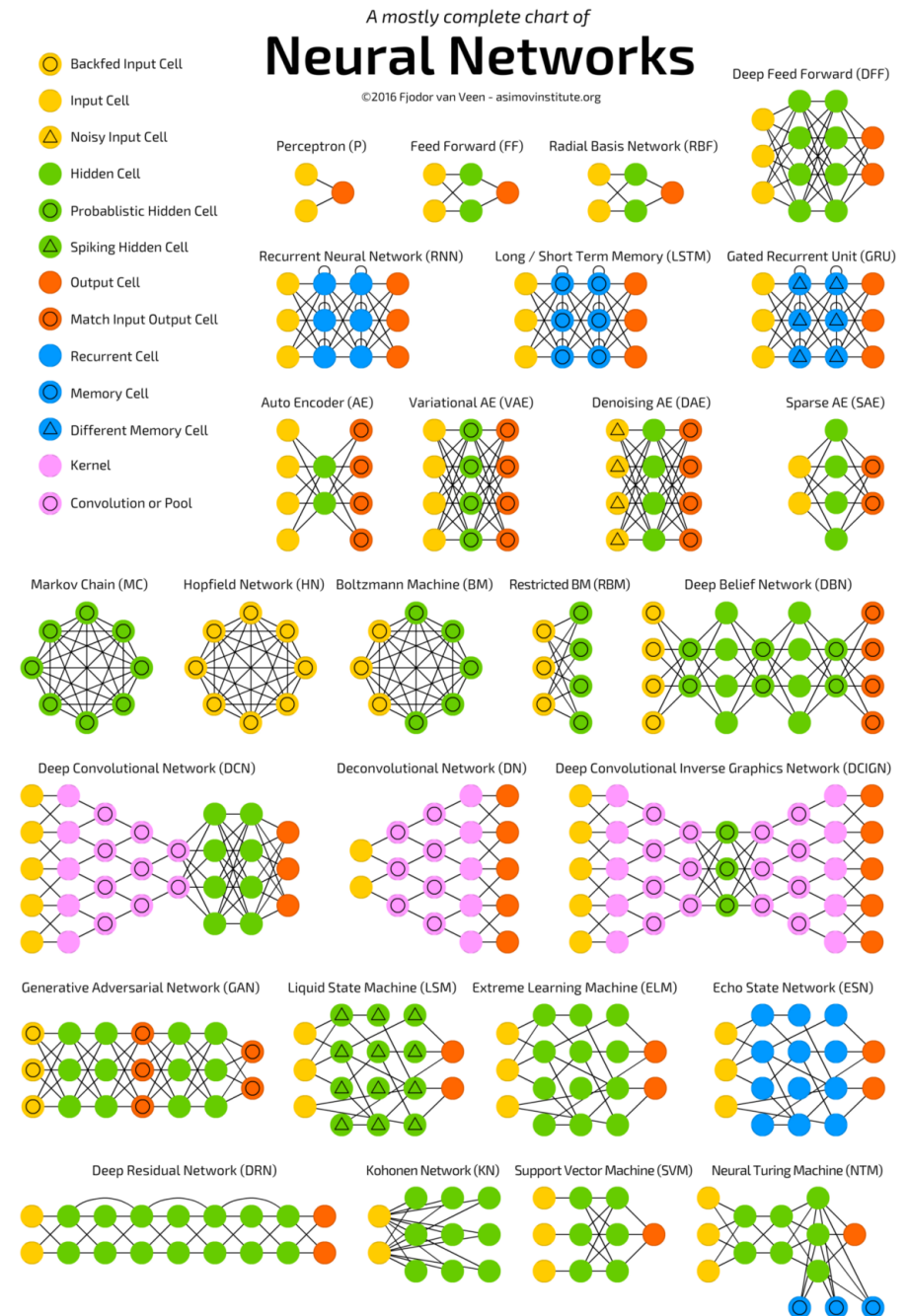
En la **etapa de entrenamiento**, se calculan los pesos de cada unidad de procesamiento de tal forma que se reduzca el error de predicción para un conjunto de datos de entrenamiento (**problema de optimización**).

---

# Elementos de una arquitectura de redes neuronal

- Comportamiento de las neuronas individuales:
  - Tipo de neurona
  - Función de activación
- Topología de la red (forma como se conectan las neuronas)
- Número de capas y neuronas
  - Redes poco profundas (**shallow networks**)
  - Redes profundas (**deep networks**)
- El método con el que se **entrenan** los pesos de la red neuronal (**función de pérdida y algoritmo de optimización**).

# Arquitecturas comunes



# Funciones de activación

---

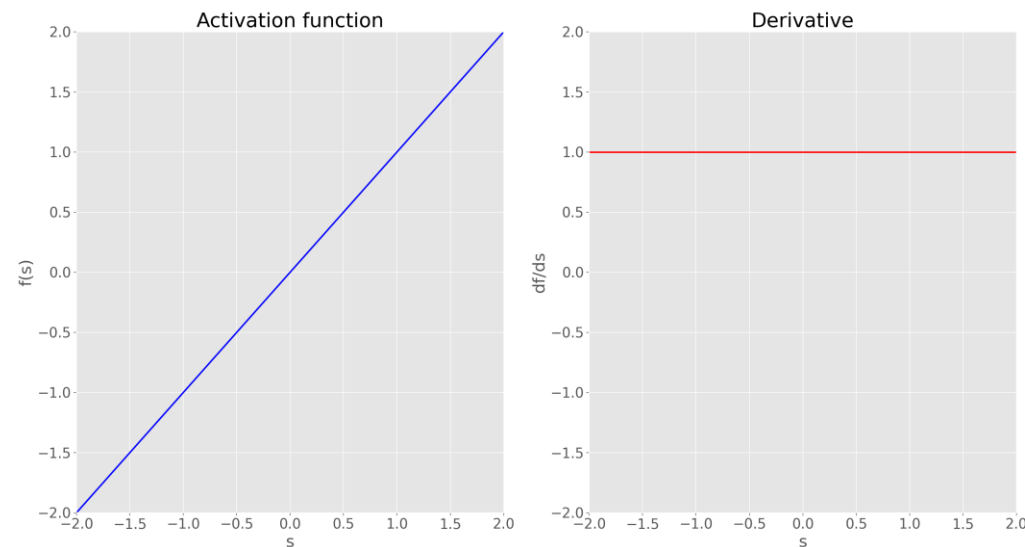


# Funciones de activación

- Función identidad

$$f(s) = s$$

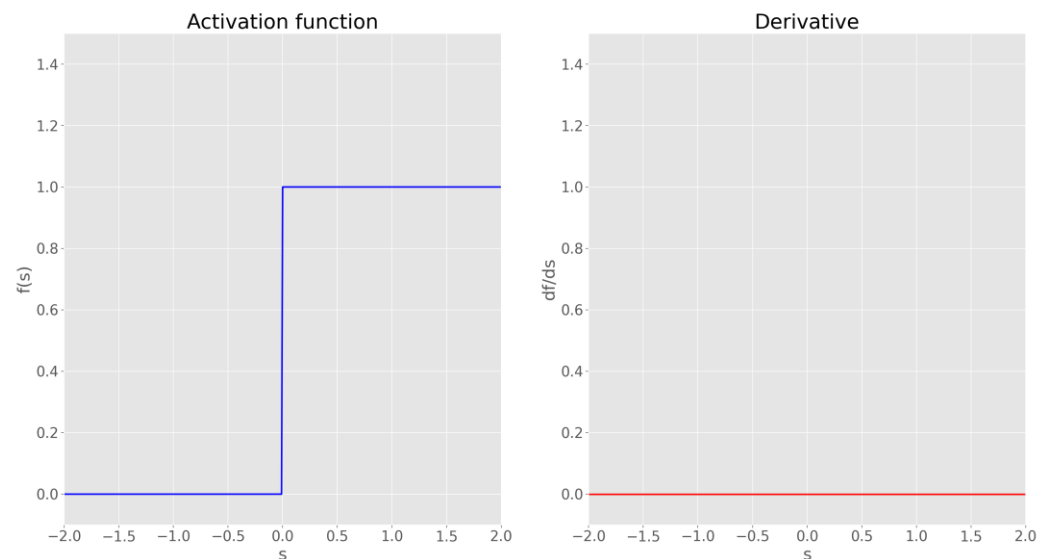
$$\frac{df}{ds} = 1$$



- Función escalón

$$f(s) = \begin{cases} 0 & \text{para } s < 0 \\ 1 & \text{para } s \geq 0 \end{cases}$$

$$\frac{df}{ds} = 0$$





# Funciones de activación

- Función logística (sigmoide) :

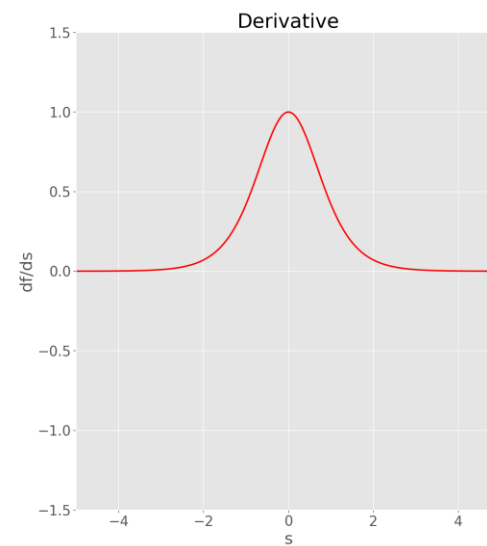
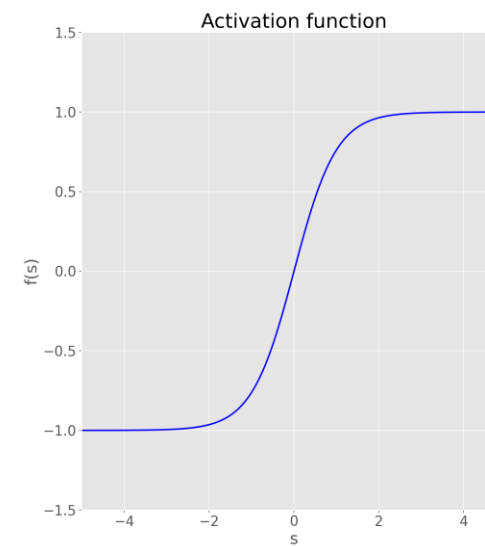
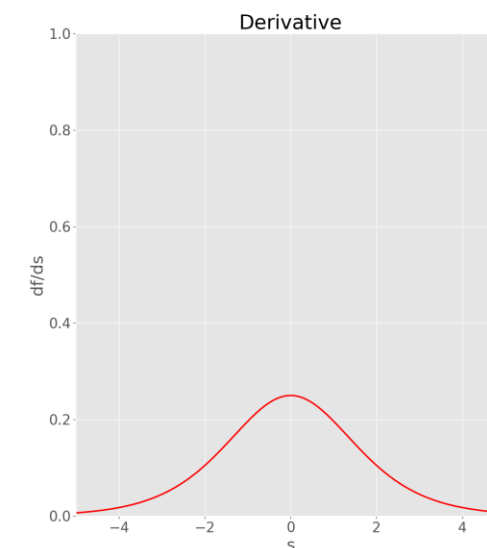
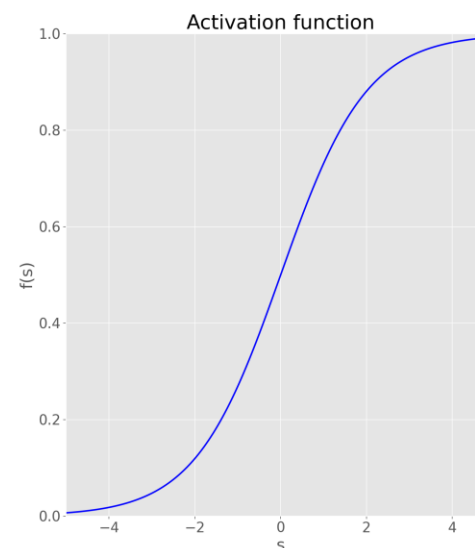
$$f(s) = \frac{1}{1 + e^{-s}}$$

$$\frac{df}{ds} = f(s)(1 - f(s))$$

- Función tanh:

$$f(s) = \tanh(s) = \frac{2}{1 + e^{-2s}} - 1$$

$$\frac{df}{ds} = 1 - \tanh^2(s)$$





# Funciones de activación

- Función unidad rectificada lineal (ReLU):

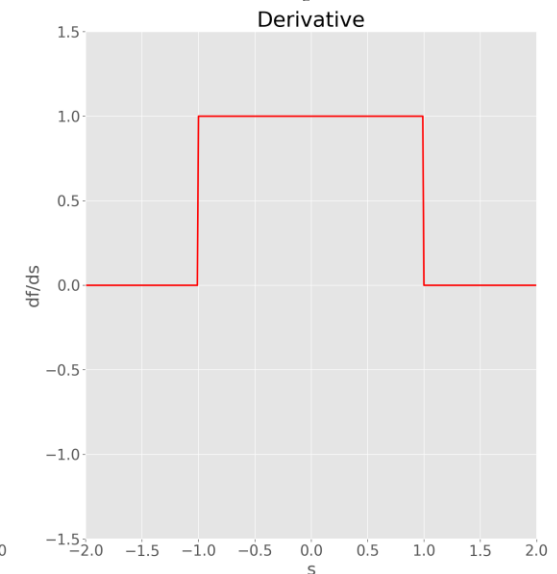
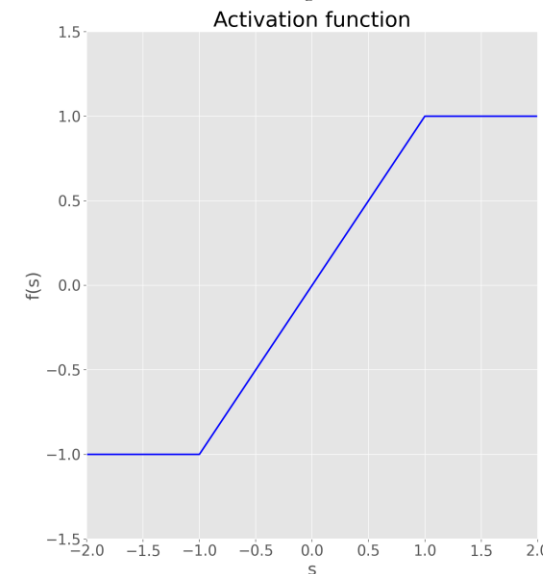
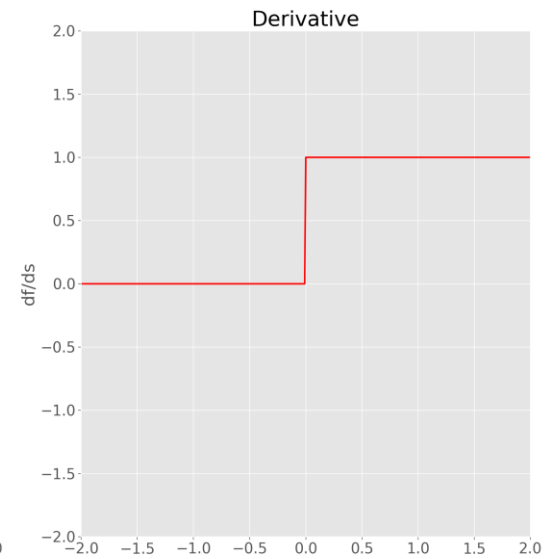
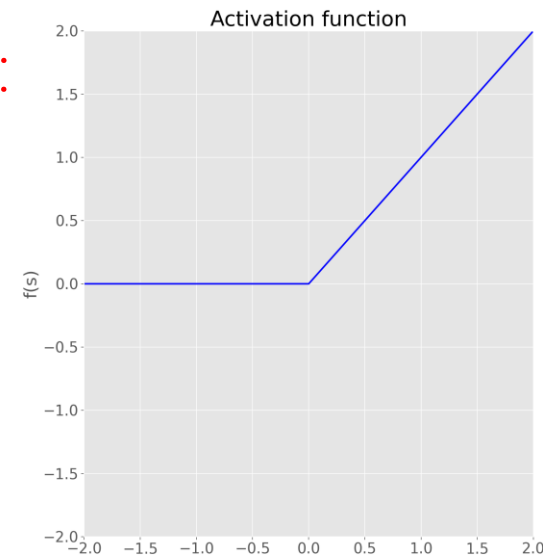
$$f(s) = \begin{cases} 0 & \text{para } s < 0 \\ s & \text{para } s \geq 0 \end{cases}$$

$$\frac{df}{ds} = \begin{cases} 0 & \text{para } s < 0 \\ 1 & \text{para } s \geq 0 \end{cases}$$

- Función **tanh** dura (*hard* – tanh):

$$f(s) = \max(\min(s, 1), -1)$$

$$\frac{df}{ds} = \begin{cases} 0 & \text{para } s < -1 \\ 1 & \text{para } -1 \leq s \leq 1 \\ 0 & \text{para } s > 1 \end{cases}$$



# Bibliografía

- Aggarwal , C. C. (2023). *Neural networks and deep learning* (2da ed.). Springer.
  - Capítulo 1
- James , G., Witten, D., Hastie, T. & Tibshirani, R. (2023). *An introduction to statistical learning: with applications in Python* (2da ed.). Springer.
  - Capítulo 10
- Hastie , T., Tibshirani , R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2da ed.). Springer.
  - Capítulo 11
- Russell, S. J. & Norvig, P. (2021). *Artificial intelligence: A modern approach (global edition)* (4ta ed.). Person.
  - Capítulo 22