

**Redes Neuronales**

Basada en las neuronas y sus partes

Entradas (simulan el axón)

Cuerpo

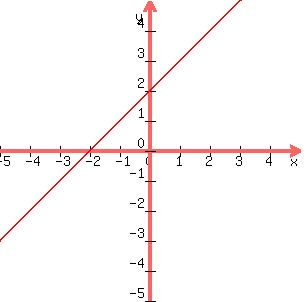
Función de agregación: sumatoria de ( xi \* wi )

Salida

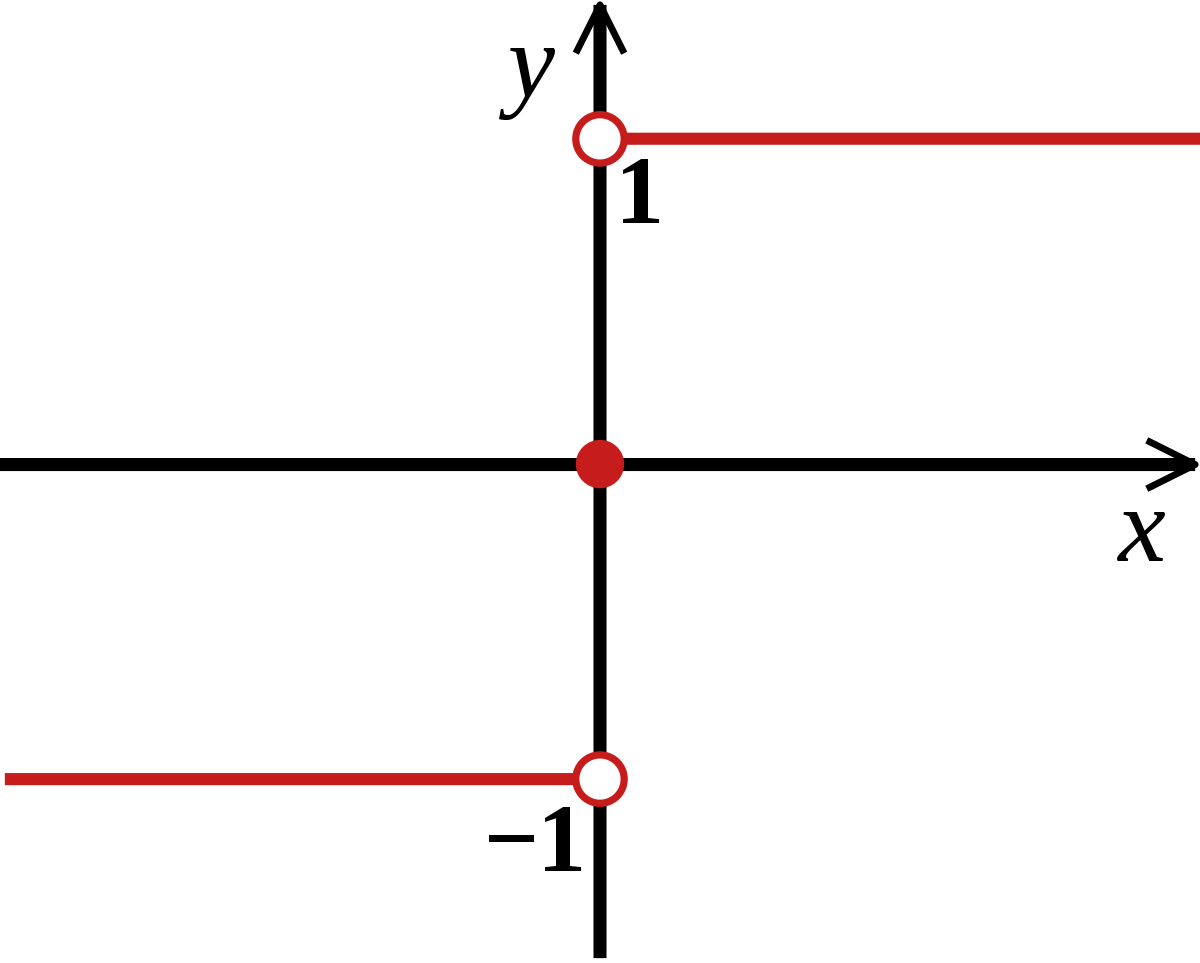
f()

Función de Activación

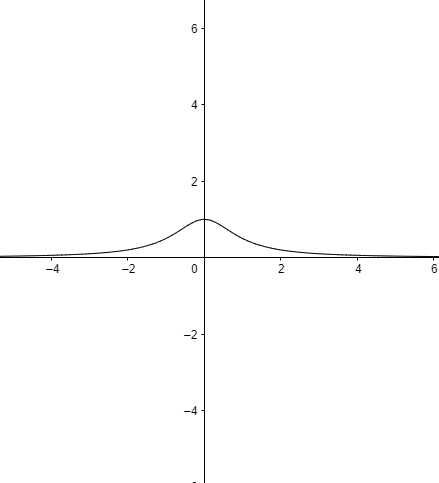
identidad: x = y



y = sin(sigman(x))



y = h(x)



Ejemplo

## Problema de Clasificación/Categorización

C1

C2

D = C1 U C2





Cuál es el b que me indica w2 y w1 para saber a qué lado pertenece el punto (x, y)

Despejando m de **x \* w1 + y \* w2 + b** dado que y = m \* x + b

x \* w1 + y \* w2 + b = 0

y \* w2 = - x \* w1 - b

**y = (- x \* w1 - b) / w2**

x \* w1 + y \* w2 + b = 0

**x = (- y \* w2 - b) / w1**

x \* w1 + y \* w2 + b = 0

**b = - x \* w1 - y \* w2**

y = m \* x + b

**m = (b - y) / x**

**m = [(- x \* w1 - y \* w2) -**

**((- x \* w1 - b) / w2))]**

**/**

**[(- y \* w2 - b) / w1]**

**Entrenamiento**

1. Conjunto de Datos Reales (D)

C1 → {(-2, 2) , (-2, -1)}

C2 → { (4, 6) , (5, 1) , (1, -1) }

Programar

Función de Entrenamiento

Función de clasificación:

Entrada: (w1, w2, b)

Salida: 1 ó -1

2. Generar puntos aleatoriamente

w´1 y w´2 , b ⇒ gen. random

Corrección:

w´1 = w´1 + d \* x0 D → (x0, y0) tomado del conjunto de datos reales

w´2 = w´2 + d \* y0

b = b´ + d

clasificacion??

d = 1 si y = -1 y el punto si pertenece a C2

d =-1 si y = 1 y el punto si pertenece a C1

Nota:

* El entrenamiento termina cuando pasa por todos los elementos de D
* La corrección se hace sólo si falla en cada ciclo
* IA
  + Red Neuronal
    - Perceptón

Deep Learning: Aprendizaje profundo

* Voz/Audio
  + Volumen
  + Pronunciación
  + Contexto
  + Rapidez
  + Dialecto
  + Idioma
  + Discapacidad
* Imágenes/Video
  + Movimiento
  + Profundidad
  + Posición
  + Contexto
  + Percepción
  + Colores
  + Tamaño
  + Iluminación
  + Nitidez
  + Sombras
  + Tipo
  + Rotación
* Reconocimiento Texto
  + Contexto
  + Idioma