

Redes Neuronales

Sistemas de Inteligencia Artificial

Primer Cuatrimestre 2025

Alan Pierri
Rodrigo Ramele
Eugenia Piñeiro
Marina Fuster
Luciano Bianchi
Santiago Reyes
Marco Scilipoti
Paula Oseroff
Joaquín Girod



CLAUDE AI



ALPHAGO



 SORA



¿Qué
entienden
por red
neuronal?





“Una red neuronal es un tipo de algoritmo de aprendizaje automático que está modelado según la estructura y función del cerebro humano.”



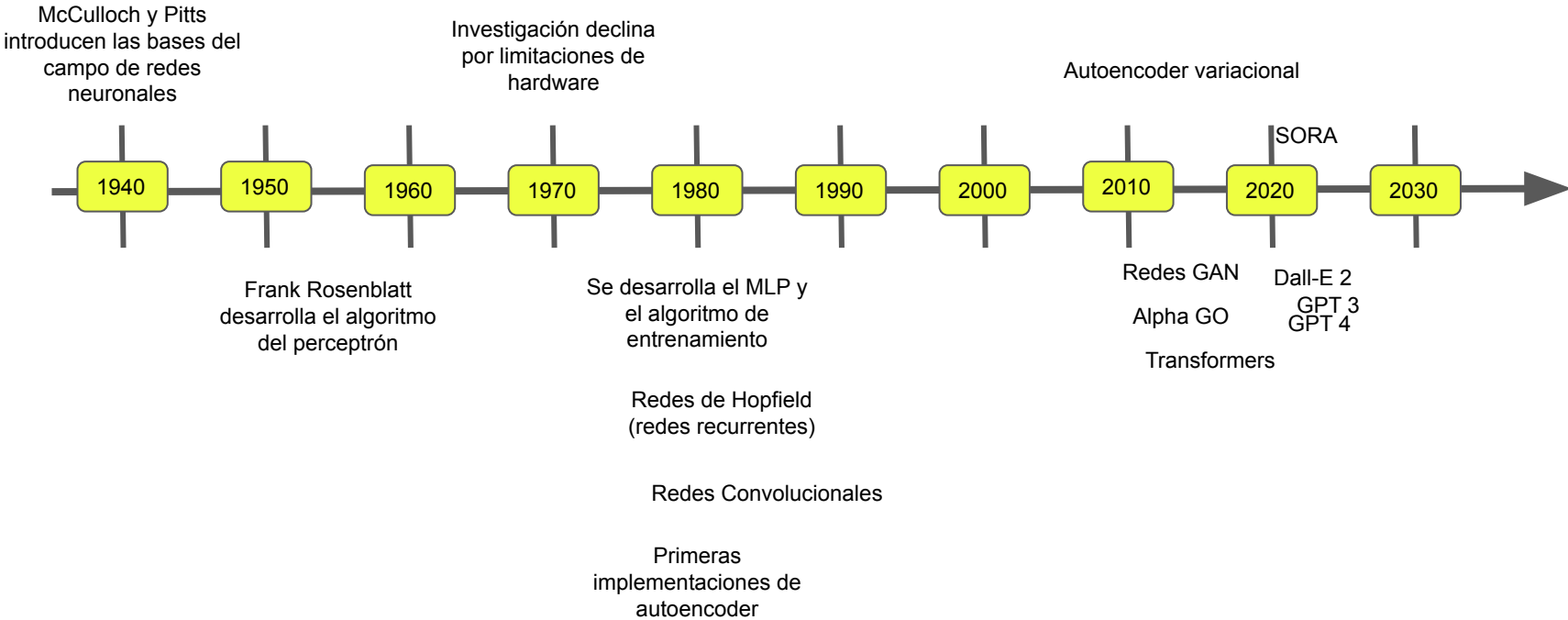
Consiste en una gran cantidad de nodos de procesamiento interconectados (neuronas) que trabajan juntos para procesar y analizar datos complejos.



Cada neurona recibe uno o más inputs, los procesa utilizando un conjunto de pesos aprendidos y produce una salida que se transmite a otras neuronas en la red.”



EVENTOS EN EL ÁREA DE REDES NEURONALES

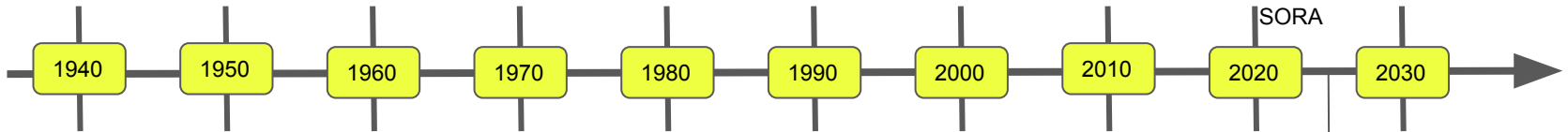


EVENTOS EN EL ÁREA DE REDES NEURONALES

McCulloch y Pitts
introducen las bases del
campo de redes
neuronales

Investigación declina
por limitaciones de
hardware

Autoencoder variacional



Frank Rosenblatt
desarrolla el algoritmo
del perceptrón

Se desarrolla el MLP y
el algoritmo de
entrenamiento

Redes de Hopfield
(redes recurrentes)

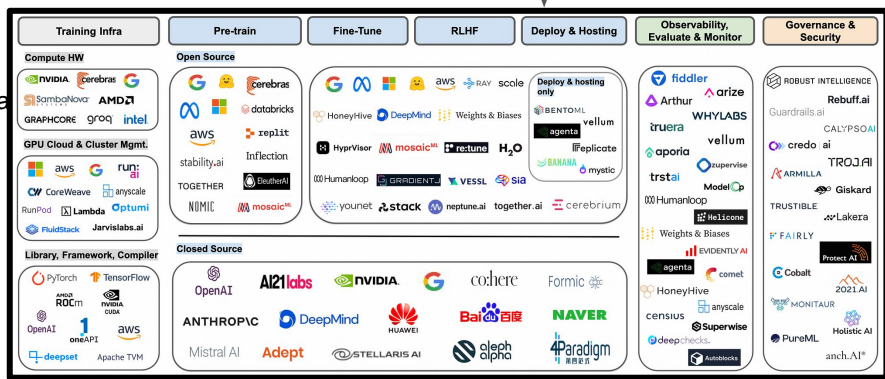
Redes Convoluciona

Primeras
implementaciones de
autoencoder

Redes GAN
Alpha GO
Transformers

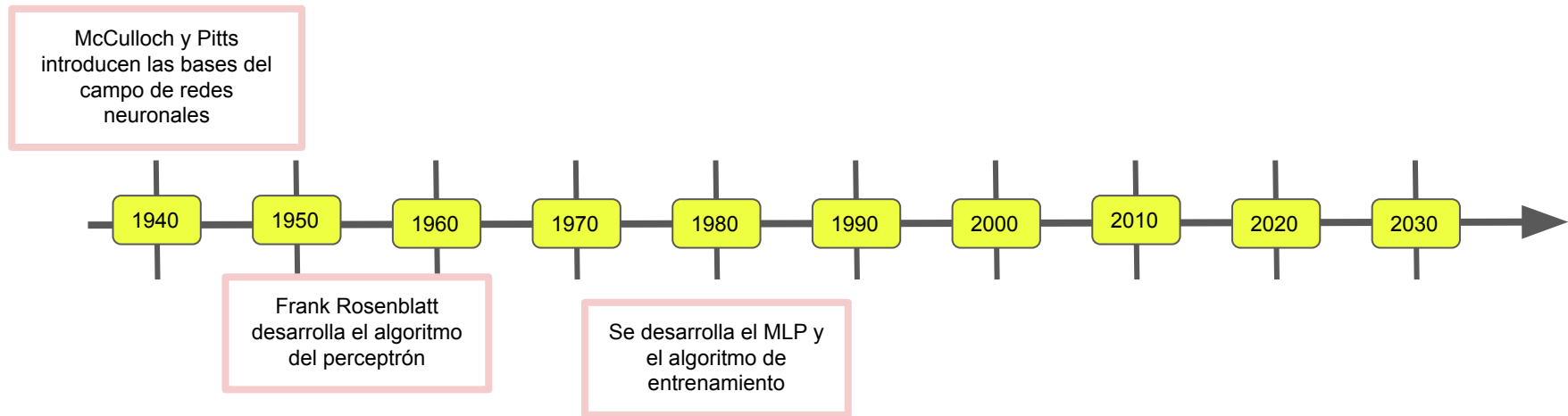
SORA
Dall-E 2
GPT 3
GPT 4

LLM BOOM



LLM Explained: The LLM Training Landscape

¿QUÉ IDEAS ESTUDIAMOS DURANTE EL TP3?



- Por el momento no vamos a profundizar el área de “*Deep Learning*” (se ve más adelante en la materia)
- Objetivo: entender qué problemas se pueden resolver, los conceptos básicos que componen a una red neuronal y su entrenamiento

Clase 1: modelo básico
de neurona (capítulo 6)

Clase 2: modelo básico
de red neuronal
(capítulo 7)

Clase 3: métricas de
evaluación y sobreajuste
(capítulo 8), métodos de
optimización para RN
(capítulo 7) + consultas
hora extra

Clase 4: Backpropagation
II
consultas (presenciales?)

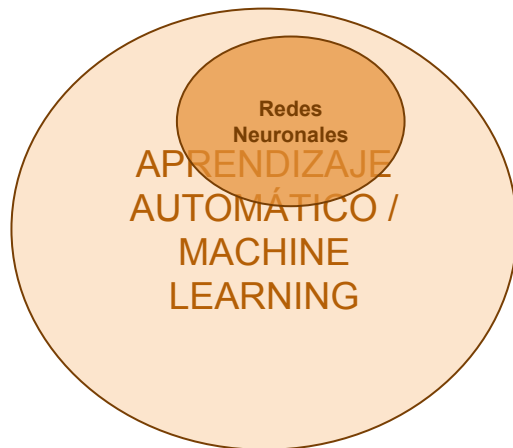
DE LA CLASE DE OPTIMIZACIÓN



9.1 Redefiniendo Aprendizaje

El área de redes neuronales cae dentro de la frontera de lo que se denomina **Aprendizaje Automático** o *Machine Learning*. Puede definirse como:

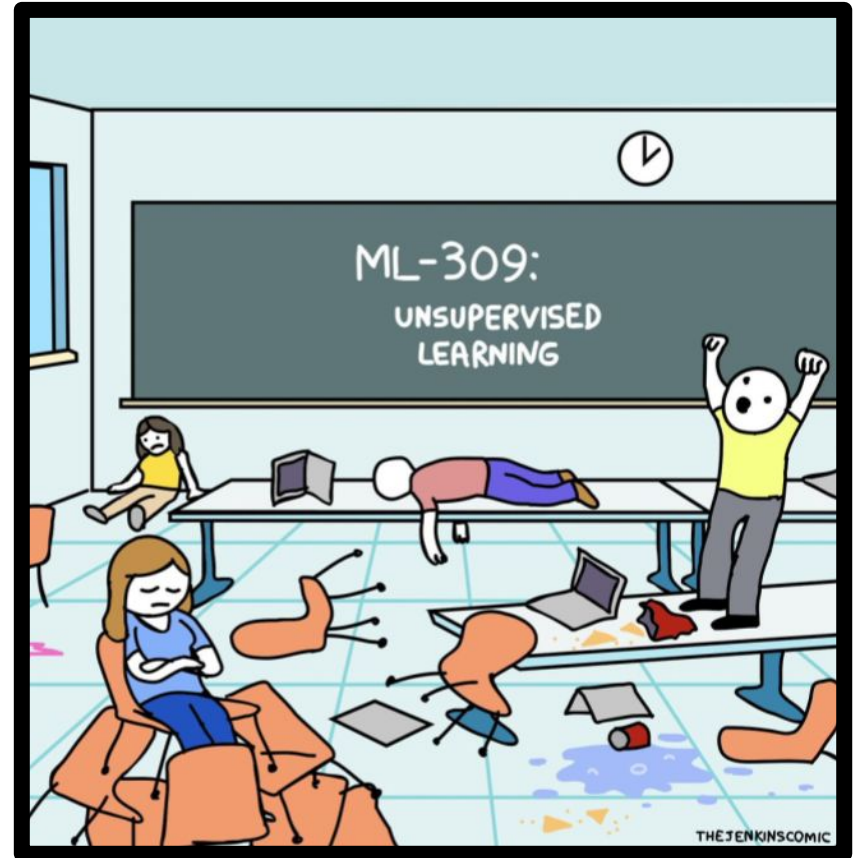
Definition 9.1.1 — Aprendizaje Automático. Técnicas para identificar relaciones entre datos (X) o mapeos ($X \rightarrow Y$) basadas en algoritmos libres de modelo que dada una serie de parámetros libres que estructuran esas relaciones o mapeos, pueden ajustarse mediante algún proceso de optimización matemática [23].



APRENDIZAJE SUPERVISADO



vs. APRENDIZAJE NO SUPERVISADO



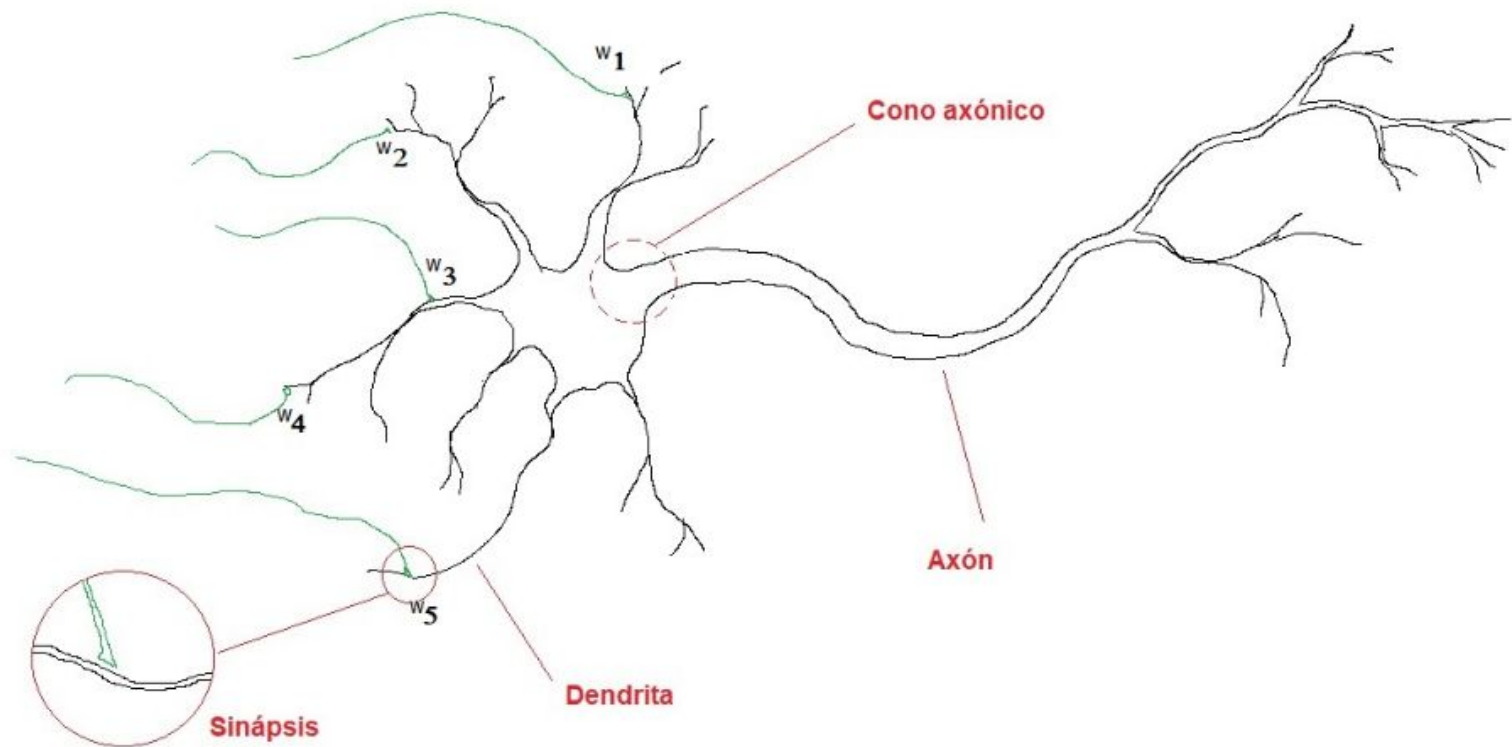
Perceptrón Simple

Sistemas de Inteligencia Artificial

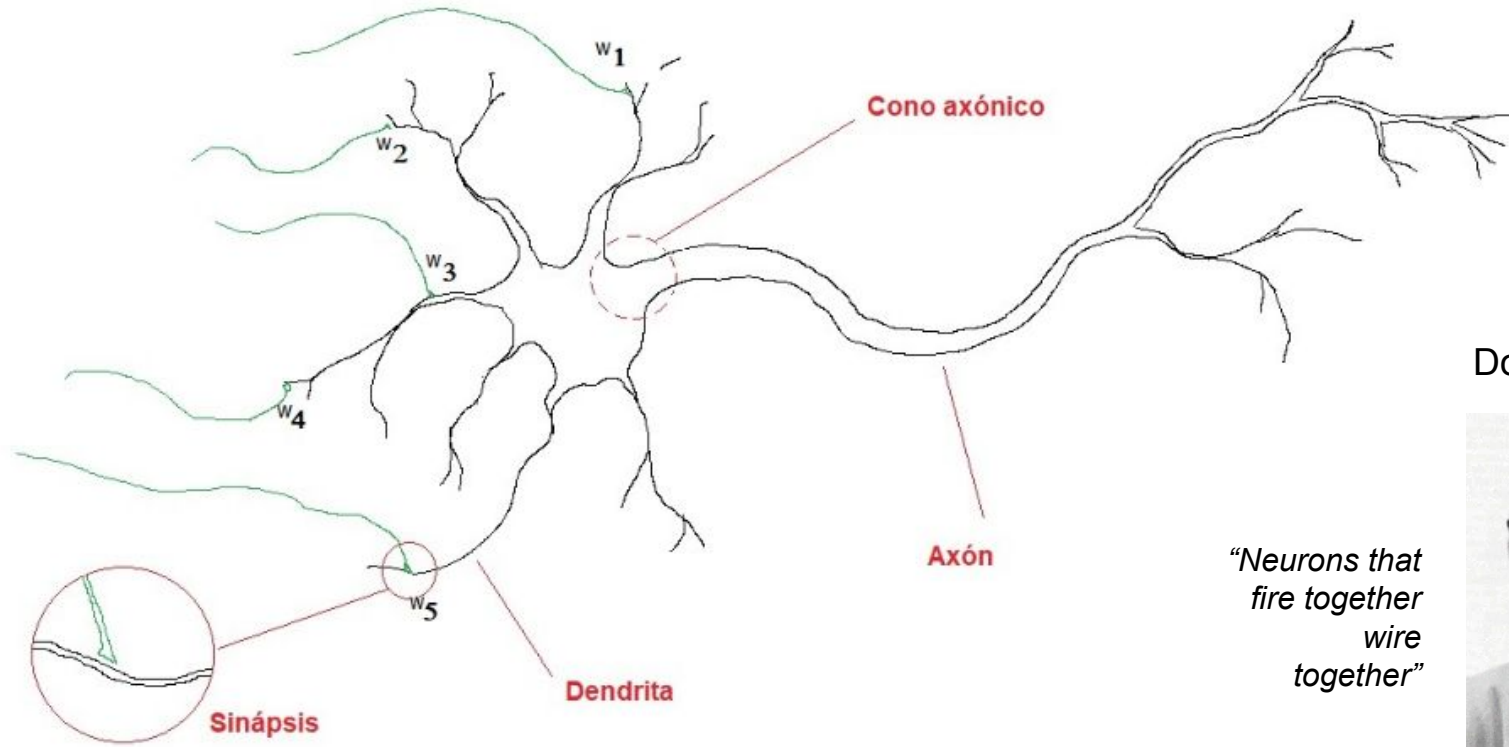
Segundo Cuatrimestre 2024

Alan Pierri
Rodrigo Ramele
Eugenia Piñeiro
Marina Fuster
Luciano Bianchi
Santiago Reyes
Marco Scilipoti
Paula Oseroff

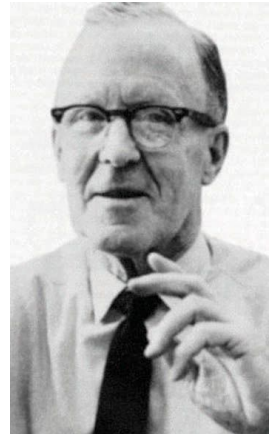
MODELO DE NEURONA - 1943



MODELO DE NEURONA - 1943

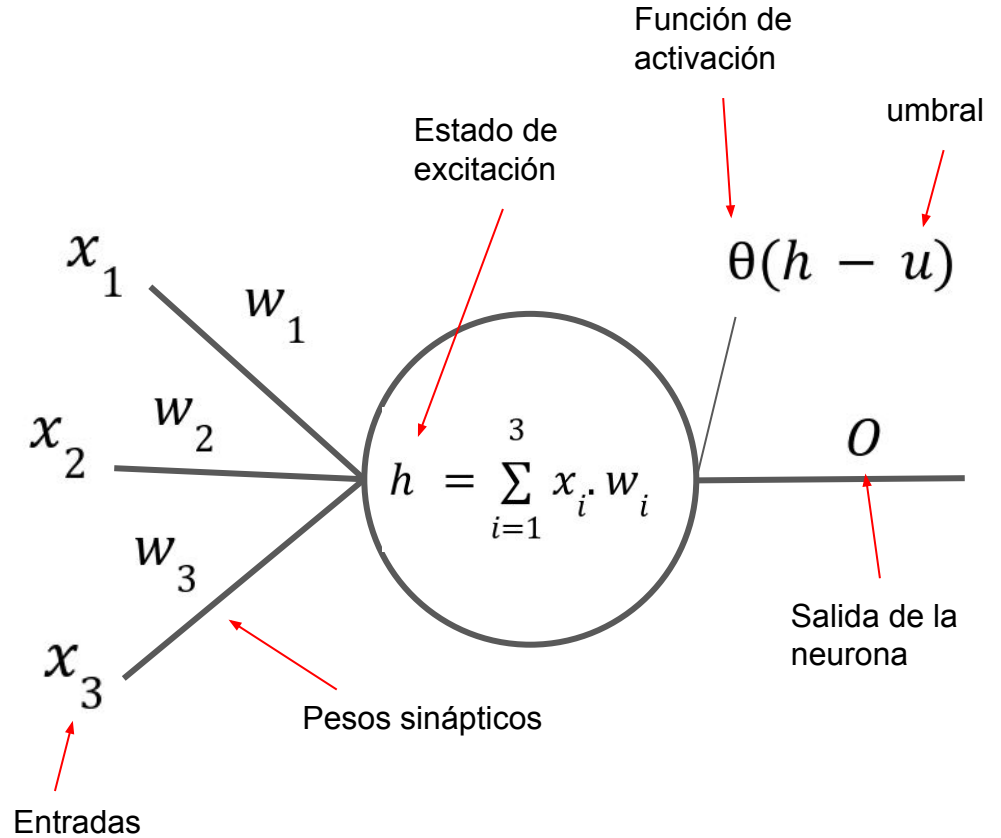
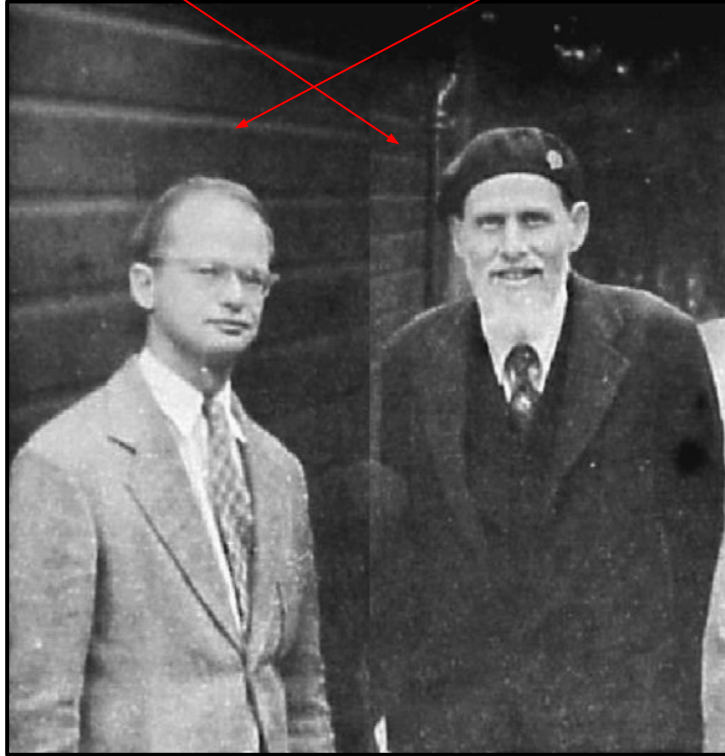


Donald Hebb



*"Neurons that
fire together
wire
together"*

Warren McCulloch & Walter Pitts 1943

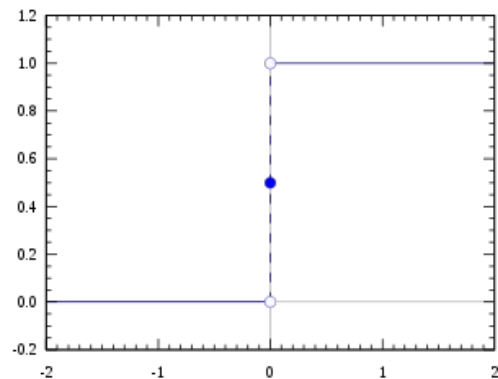


FORMALIZACIÓN MATEMÁTICA

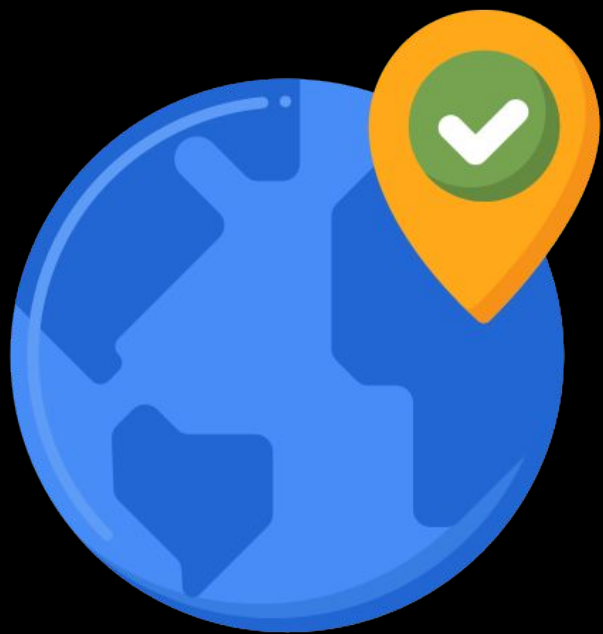
$$O = \theta\left(\sum_{i=1}^n x_i \cdot w_i - u\right)$$

$$\theta(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{en otro caso} \end{cases}$$

Función de activación **escalón** o **signo**.







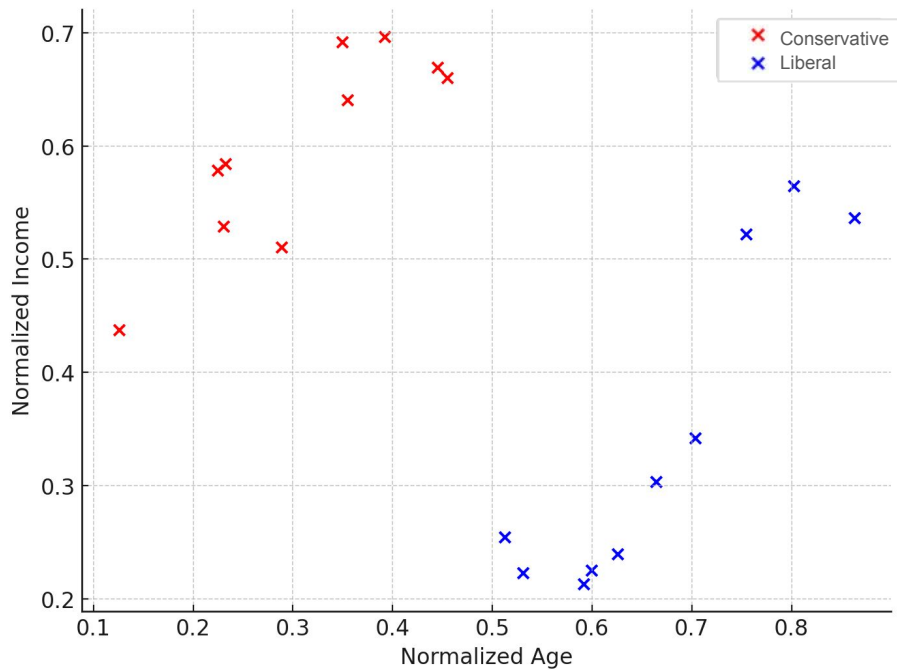
Un cliente acaba de expandir su negocio a Canadá. Con todo el quilombo de tarifas entre Canadá y Estados Unidos, le interesa saber qué candidato se encuentra más favorecido en las encuestas, ya que la votación de Abril probablemente afecte su negocio.

Los datos de los ciudadanos de los cuales dispone son:

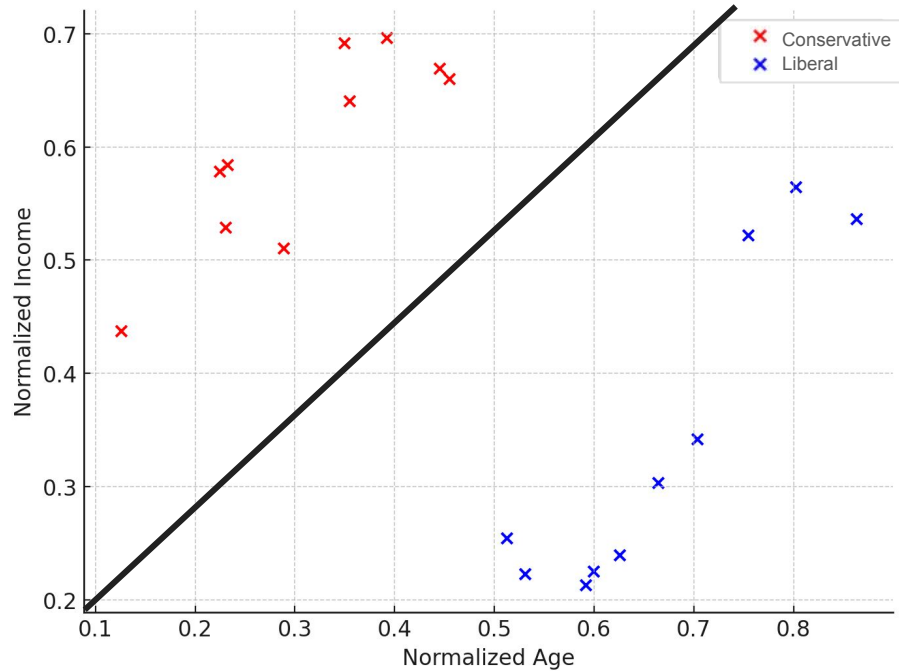
- Edad
- Ingreso económico



Carney vs Poilievre



Age	Income	Party
0.445	0.669	1.0
0.349	0.692	1.0
0.232	0.585	1.0
0.125	0.438	1.0
0.224	0.579	1.0
0.23	0.529	1.0
0.392	0.696	1.0
0.355	0.641	1.0
0.455	0.66	1.0
0.289	0.51	1.0
0.513	0.255	-1.0
0.755	0.522	-1.0
0.626	0.24	-1.0
0.703	0.342	-1.0
0.863	0.536	-1.0
0.6	0.225	-1.0
0.664	0.303	-1.0
0.802	0.565	-1.0
0.592	0.213	-1.0
0.531	0.223	-1.0

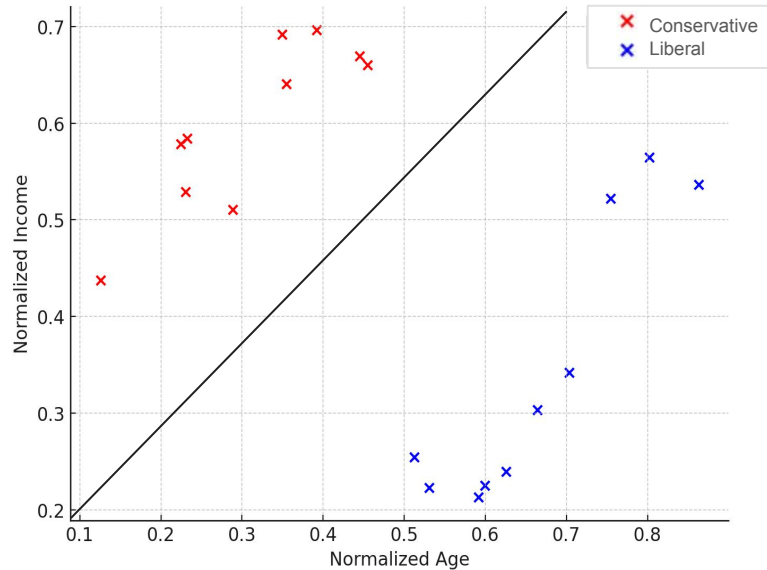


Age	x_1	Income	x_2	Party	ζ
0.445		0.669		1.0	
0.349		0.692		1.0	
0.232		0.585		1.0	
0.125		0.438		1.0	
0.224		0.579		1.0	
0.23		0.529		1.0	
0.392		0.696		1.0	
0.355		0.641		1.0	
0.455		0.66		1.0	
0.289		0.51		1.0	
0.513		0.255		-1.0	
0.755		0.522		-1.0	
0.626		0.24		-1.0	
0.703		0.342		-1.0	
0.863		0.536		-1.0	
0.6		0.225		-1.0	
0.664		0.303		-1.0	
0.802		0.565		-1.0	
0.592		0.213		-1.0	
0.531		0.223		-1.0	

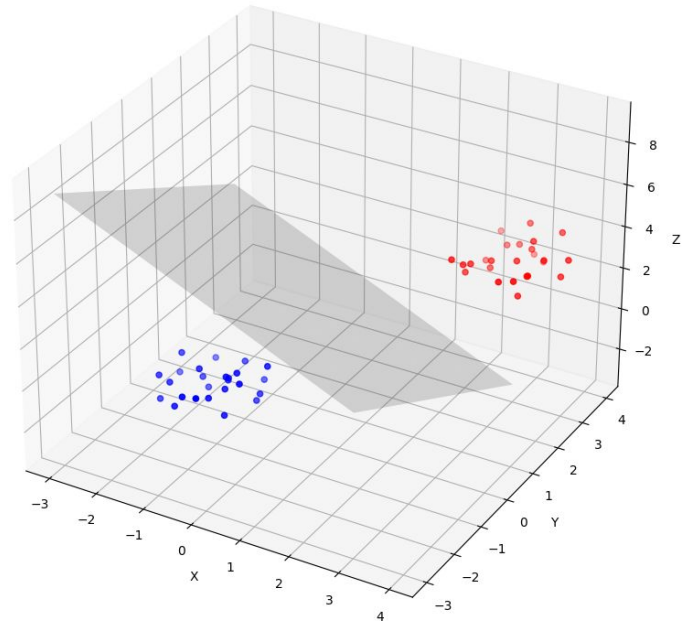
HIPERPLANO DE SEPARACIÓN

Problemas linealmente
separables

$R^2 \rightarrow$ *recta de separación*



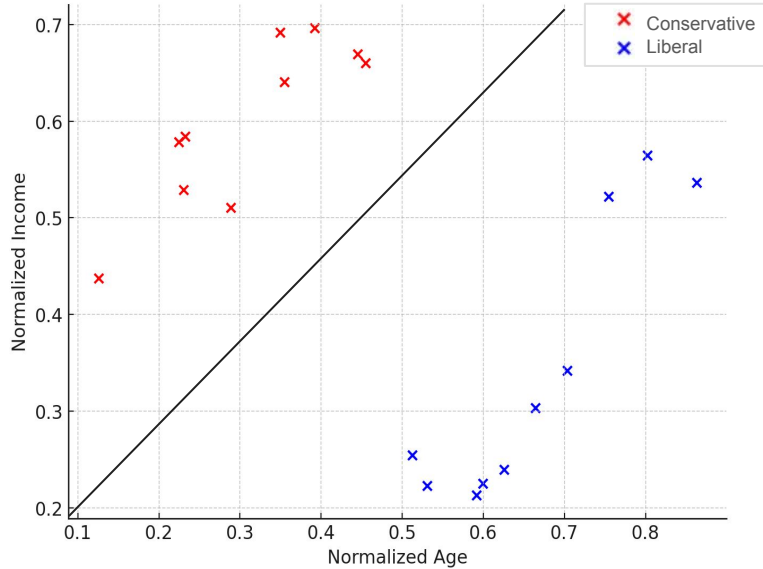
$R^3 \rightarrow$ *plano de separación*



$R^n \rightarrow$ *hiplerplano de separación*

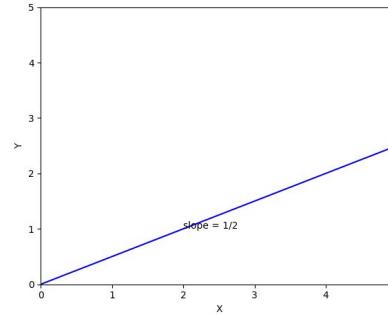
HIPERPLANO DE SEPARACIÓN

$R^2 \rightarrow$ *recta de separación*



$R^n \rightarrow$ *hiplerplano de separación*

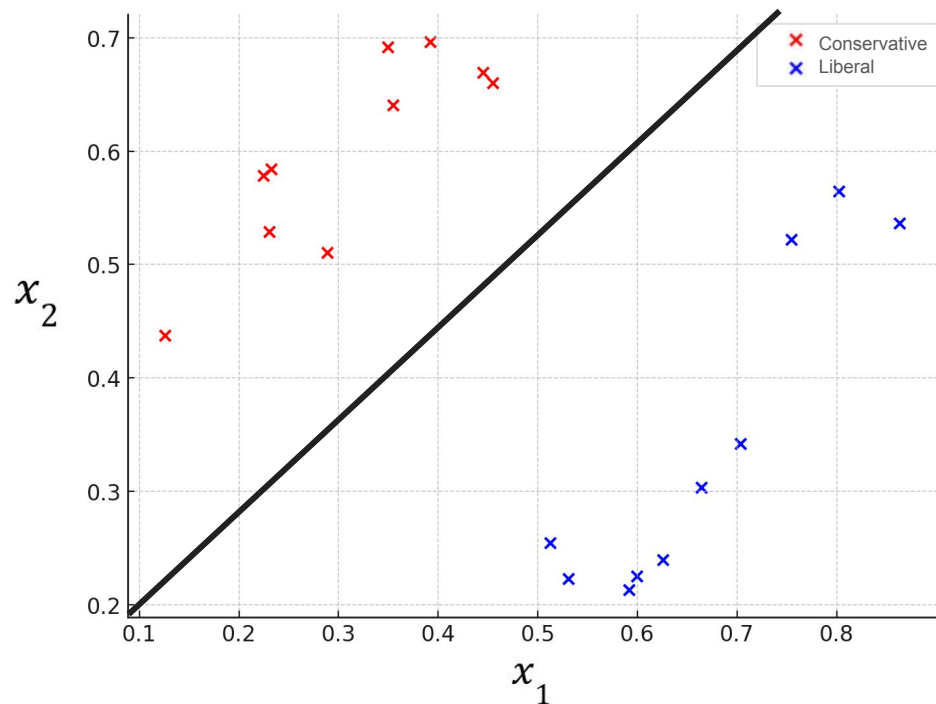
REPRESENTACIÓN DEL HIPERPLANO



$$y = mx + b$$

$$y = \frac{1}{2}x$$

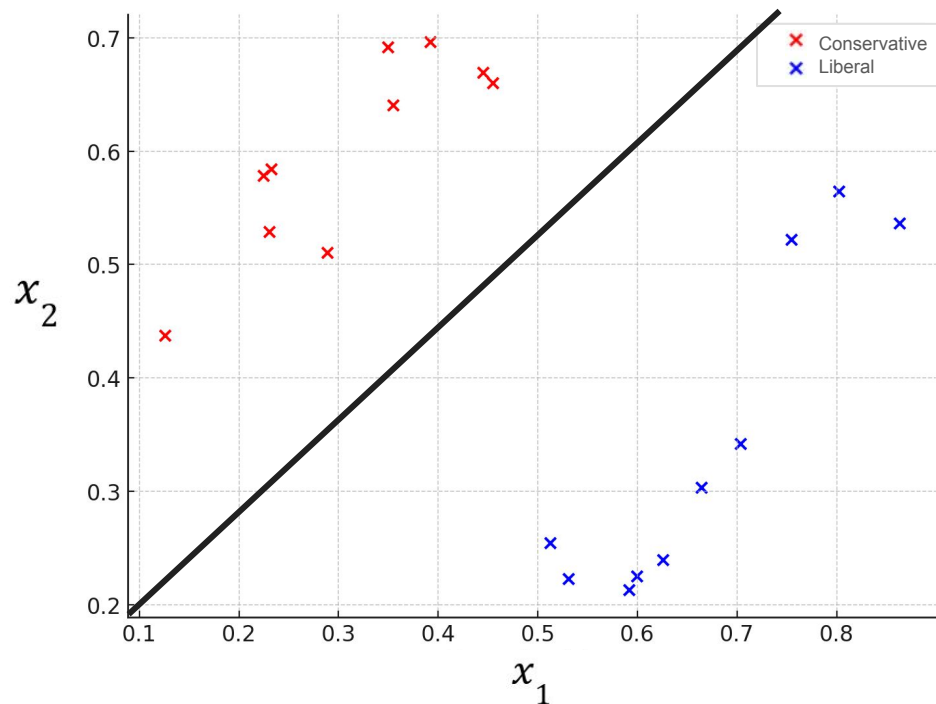
HIPERPLANO DE SEPARACIÓN



Proyección de un punto sobre el hiperplano, correspondiente a la recta que pasa por el origen:

$$proyección = \sum_{i=1}^2 x_i \cdot w_i$$

HIPERPLANO DE SEPARACIÓN

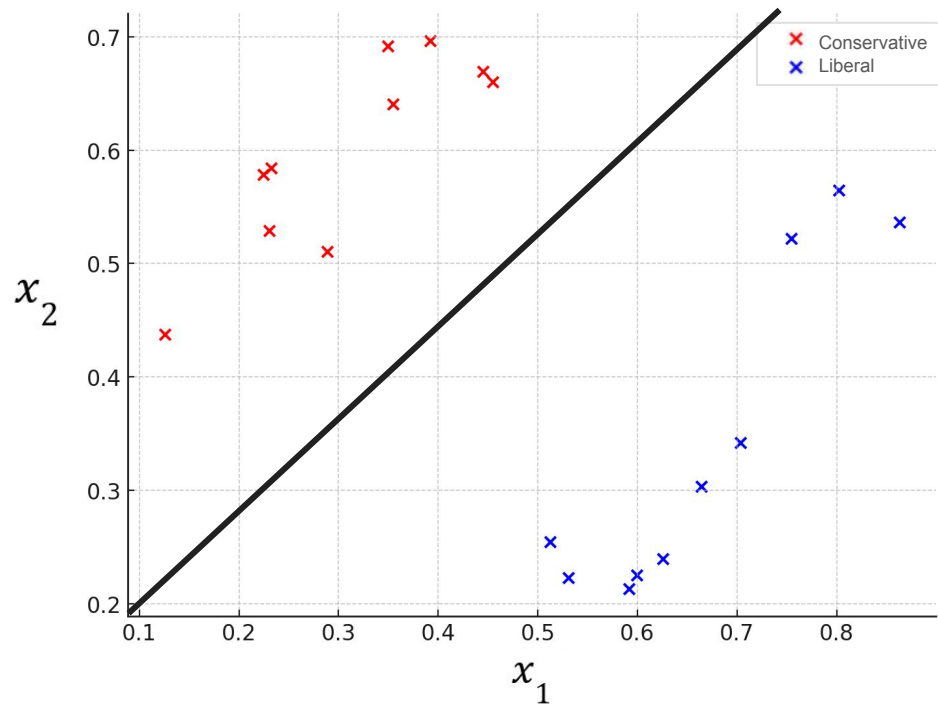


Proyección de un punto sobre el hiperplano, correspondiente a la recta que pasa por el origen:

$$proyección = \sum_{i=1}^2 x_i \cdot w_i$$

Si la proyección es negativa, la clase del dato será **(-1)**, mientras que si es positiva la clase será **(1)**

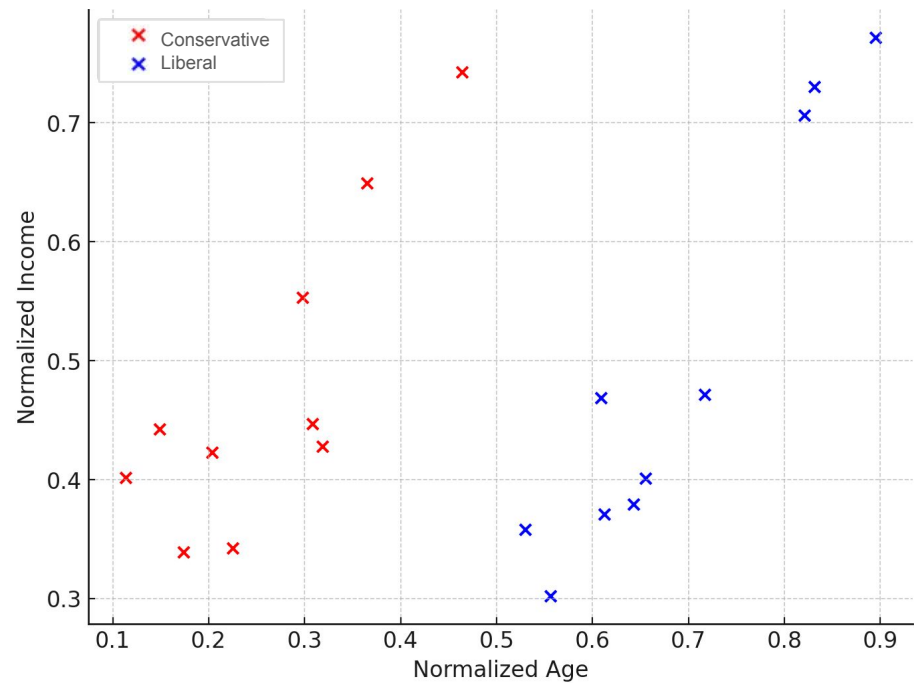
HIPERPLANO DE SEPARACIÓN



$$clase = \theta\left(\sum_{i=1}^n x_i \cdot w_i\right) \quad n = 2$$

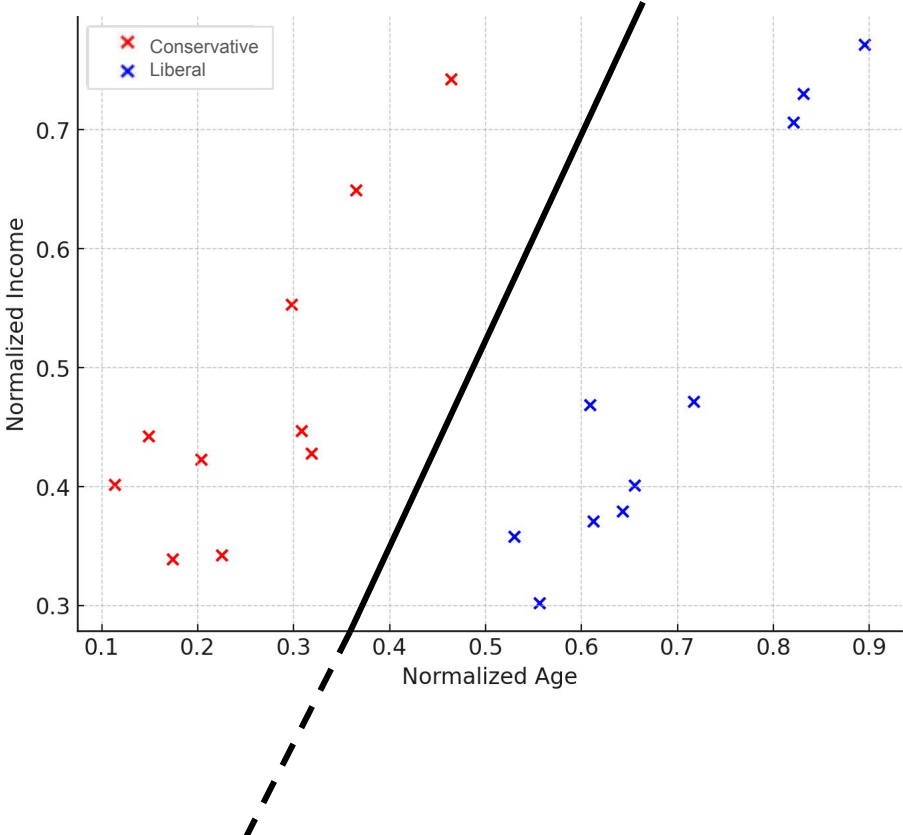
$$\theta(x) = \begin{cases} 1 & x \geq 0 \\ -1 & \text{en otro caso} \end{cases}$$

¿QUÉ SUCEDE SI MODIFICAMOS EL CONJUNTO DE DATOS?



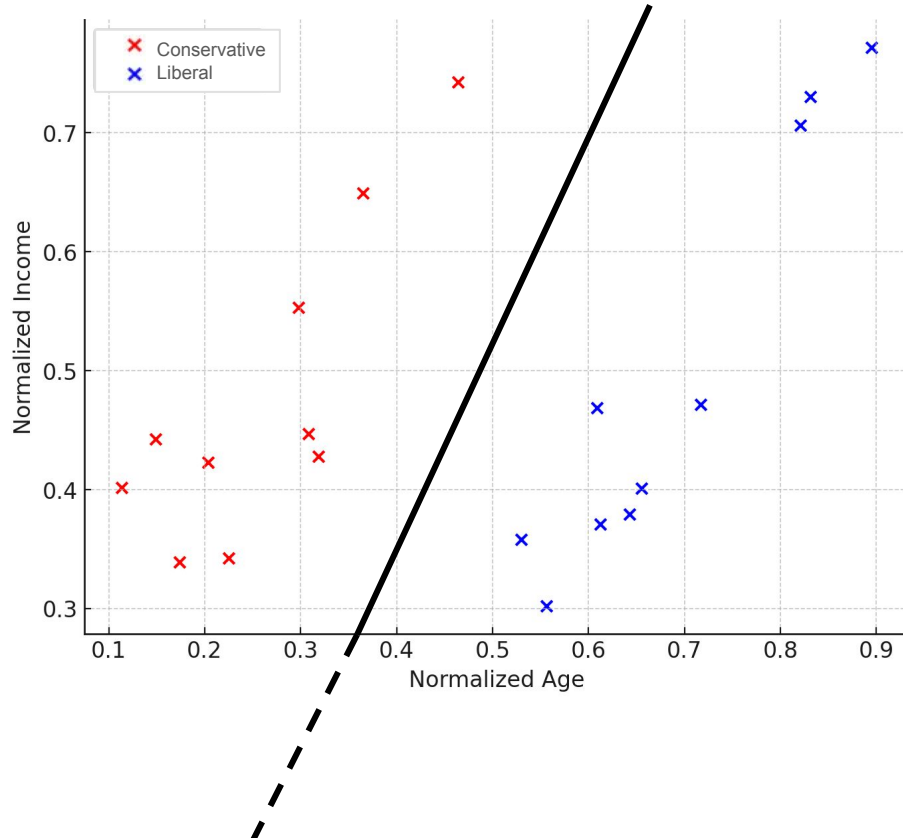
Age	x_1	Income	x_2	Party	ζ
0.149		0.443		1.0	
0.298		0.553		1.0	
0.114		0.402		1.0	
0.464		0.743		1.0	
0.204		0.423		1.0	
0.365		0.649		1.0	
0.225		0.342		1.0	
0.308		0.447		1.0	
0.319		0.428		1.0	
0.174		0.339		1.0	
0.655		0.401		-1.0	
0.609		0.469		-1.0	
0.831		0.73		-1.0	
0.643		0.38		-1.0	
0.612		0.371		-1.0	
0.717		0.471		-1.0	
0.556		0.302		-1.0	
0.821		0.706		-1.0	
0.53		0.358		-1.0	
0.895		0.772		-1.0	

¿QUÉ SUCEDE SI MODIFICAMOS EL CONJUNTO DE DATOS?



Age	x_1	Income	x_2	Party	ζ
0.149		0.443		1.0	
0.298		0.553		1.0	
0.114		0.402		1.0	
0.464		0.743		1.0	
0.204		0.423		1.0	
0.365		0.649		1.0	
0.225		0.342		1.0	
0.308		0.447		1.0	
0.319		0.428		1.0	
0.174		0.339		1.0	
0.655		0.401		-1.0	
0.609		0.469		-1.0	
0.831		0.73		-1.0	
0.643		0.38		-1.0	
0.612		0.371		-1.0	
0.717		0.471		-1.0	
0.556		0.302		-1.0	
0.821		0.706		-1.0	
0.53		0.358		-1.0	
0.895		0.772		-1.0	

¿QUÉ SUCEDE SI MODIFICAMOS EL
CONJUNTO DE DATOS?



$$ax + by + c = 0$$

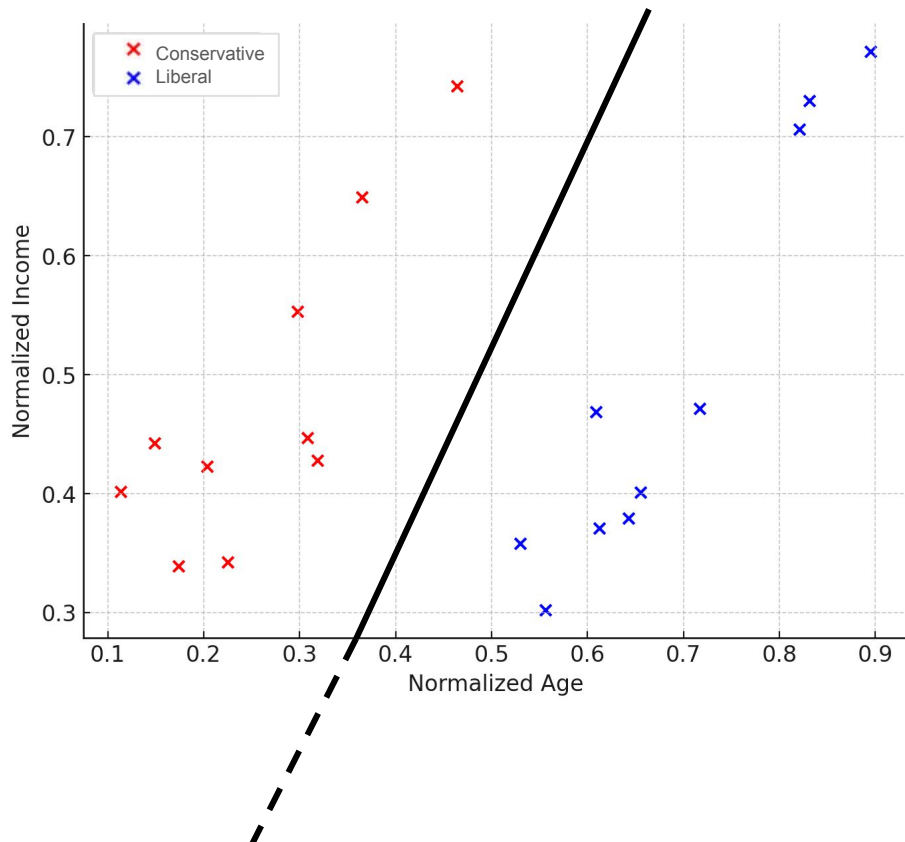


$$w_1 x_1 + w_2 x_2 + w_0 = 0$$



$$\sum_{i=1}^n x_i \cdot w_i + w_0 = 0$$

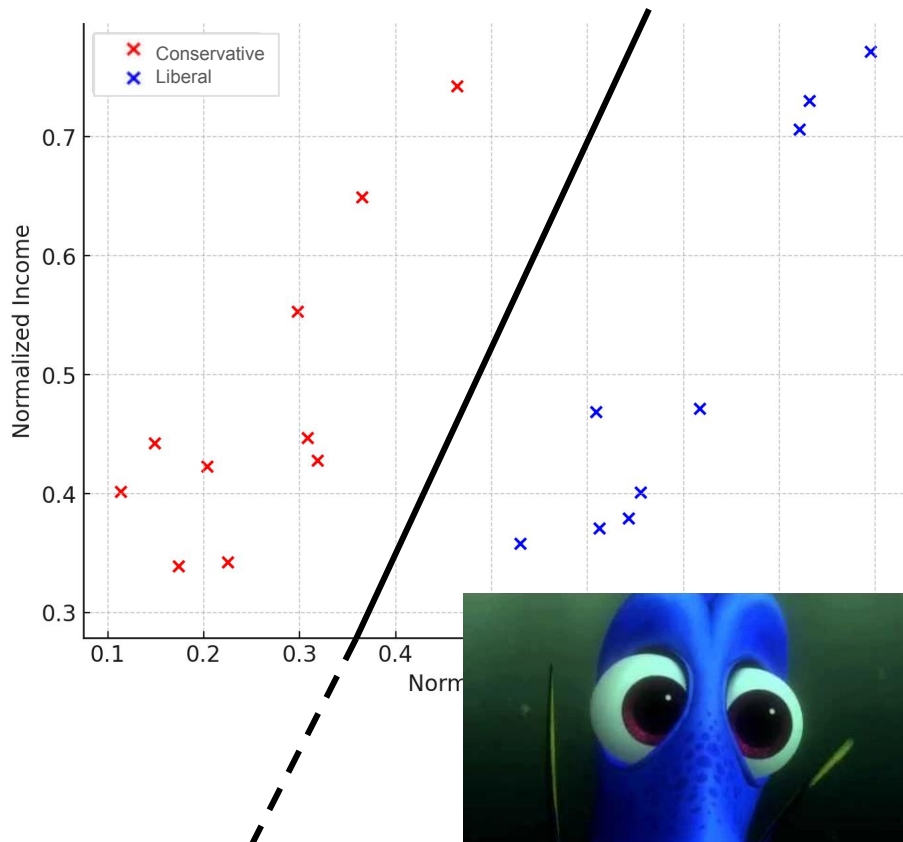
¿QUÉ SUCEDE SI MODIFICAMOS EL CONJUNTO DE DATOS?



$$clase = \theta\left(\sum_{i=1}^n x_i \cdot w_i + w_0\right)$$

$$\theta(x) = \begin{cases} 1 & x \geq 0 \\ -1 & \text{en otro caso} \end{cases}$$

¿QUÉ SUCEDE SI MODIFICAMOS EL CONJUNTO DE DATOS?



$$clase = \theta\left(\sum_{i=1}^n x_i \cdot w_i + w_0\right)$$

$$\theta(x) = \begin{cases} 1 & x \geq 0 \\ -1 & \text{en otro caso} \end{cases}$$

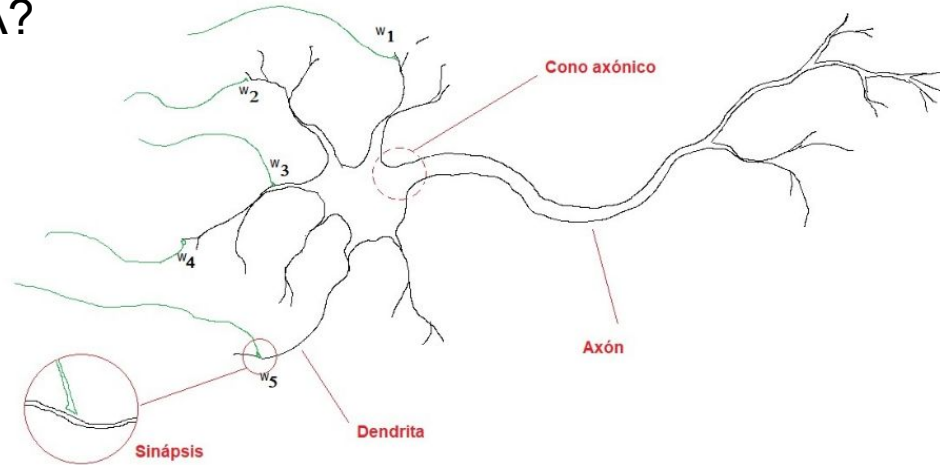
Formalización del perceptrón

$$O = \theta\left(\sum_{i=1}^n x_i \cdot w_i - u\right)$$

¿QUÉ TENEMOS HASTA AHORA?

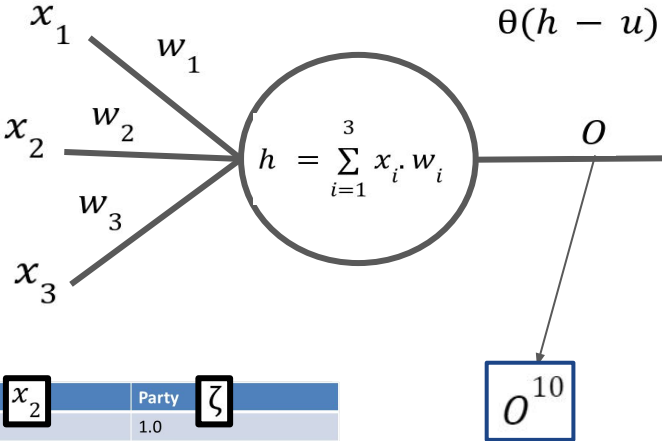
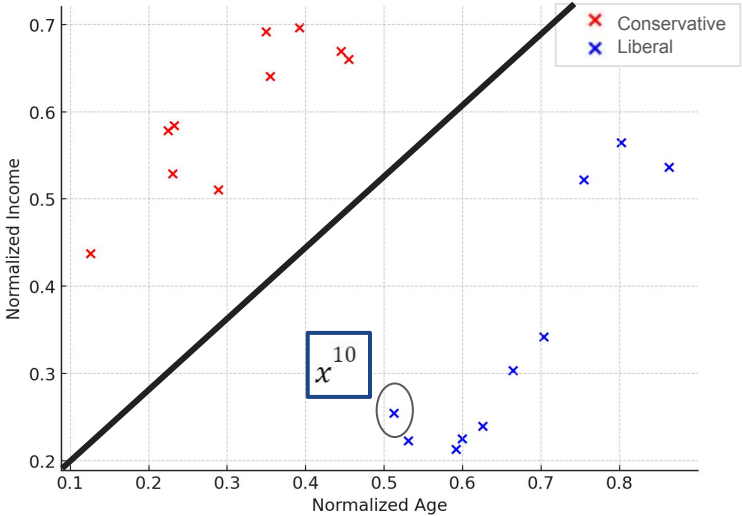
- Modelo matemático de neurona biológica
- Herramienta para resolver problemas linealmente separables en \mathbb{R}^n

¿QUÉ NOS FALTA?



REPASO

p es la cantidad de datos. μ representa el dato $(0 \dots p-1)$



Age	x_1	Income	x_2	Party	ζ
0.445		0.669		1.0	
0.349		0.692		1.0	
0.232		0.585		1.0	
0.125		0.438		1.0	
0.224		0.579		1.0	
0.23		0.529		1.0	
0.392		0.696		1.0	
0.355		0.641		1.0	
0.455		0.66		1.0	
0.289		0.51		1.0	
0.513		0.255		-1.0	
0.755		0.522		-1.0	
0.626		0.24		-1.0	
0.703		0.342		-1.0	
0.863		0.536		-1.0	
0.6		0.225		-1.0	
0.664		0.303		-1.0	
0.802		0.565		-1.0	
0.592		0.213		-1.0	
0.531		0.223		-1.0	

Annotations: A dashed blue box highlights the row with Age 0.513. Arrows point from this row to boxes labeled x_1^{10} , x_2^{10} , and ζ^{10} .

Frank Rosenblatt 1957



Cada vez que la neurona recibe un estímulo, los pesos sinápticos pueden actualizarse (proceso iterativo):

$$w^{nuevo} = w^{anterior} + \Delta w$$

¿Cómo calculamos el delta?

$$\Delta w = 2\eta x^\mu \zeta^\mu$$

Diagram illustrating the calculation of the weight update Δw using the perceptron learning rule. Red arrows point from the text labels to the corresponding terms in the equation:

- Red arrow from "tasa de aprendizaje (escalar)" to 2η
- Red arrow from "entrada o dato" to x^μ
- Red arrow from "salida esperada" to ζ^μ

$$o^\mu \neq \zeta^\mu$$

Diagram illustrating the condition for weight update. A red arrow points from the text "salida obtenida" to the term o^μ in the equation.

ACTUALIZACIÓN DE LOS PESOS SINÁPTICOS

¿Cómo calculamos el delta?

$$\Delta w = \begin{cases} 2\eta x^\mu \zeta^\mu & O^\mu \neq \zeta^\mu \\ 0 & \text{en otro caso} \end{cases}$$

Si consideramos que la salida puede ser 1 o -1, podemos reformular:

$$\Delta w = \eta(\zeta^\mu - O^\mu)x^\mu$$

donde la salida de la neurona está representada por

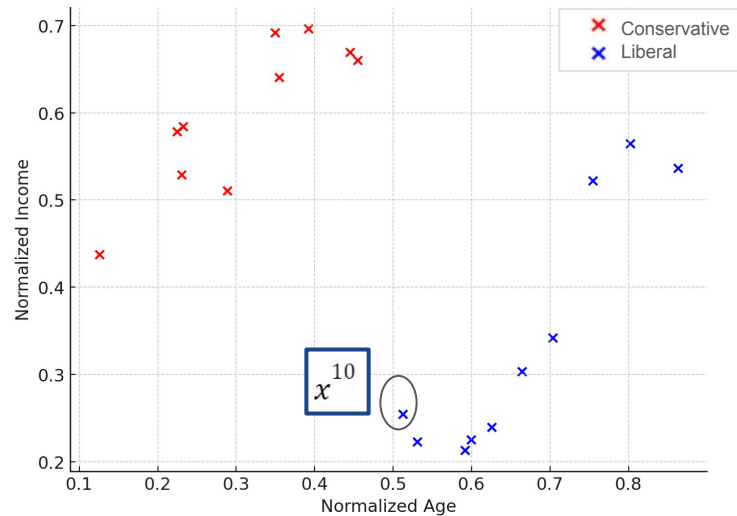
$$O(x) = \sum_{i=1}^n x_i \cdot w_i + w_0$$

EJEMPLO NUMÉRICO

$$(x_1^{10}, x_2^{10}) = (0.513, 0.255) \quad \zeta^{10} = (-1)$$

$$\Delta w = \eta(\zeta^\mu - o^\mu)x^\mu$$

$$w^{\text{nuevo}} = w^{\text{anterior}} + \Delta w$$



Age	x_1	Income	x_2	Party	ζ
0.445		0.669		1.0	
0.349		0.692		1.0	
0.232		0.585		1.0	
0.125		0.438		1.0	
0.224		0.579		1.0	
0.23		0.529		1.0	
0.392		0.696		1.0	
0.355		0.641		1.0	
0.455		0.66		1.0	
0.289		0.51		1.0	
0.513		0.255		-1.0	
0.755		0.522		-1.0	
0.626		0.24		-1.0	
0.703		0.342		-1.0	
0.863		0.536		-1.0	
0.6		0.225		-1.0	
0.664		0.303		-1.0	
0.802		0.565		-1.0	
0.592		0.213		-1.0	
0.531		0.223		-1.0	

EJEMPLO NUMÉRICO

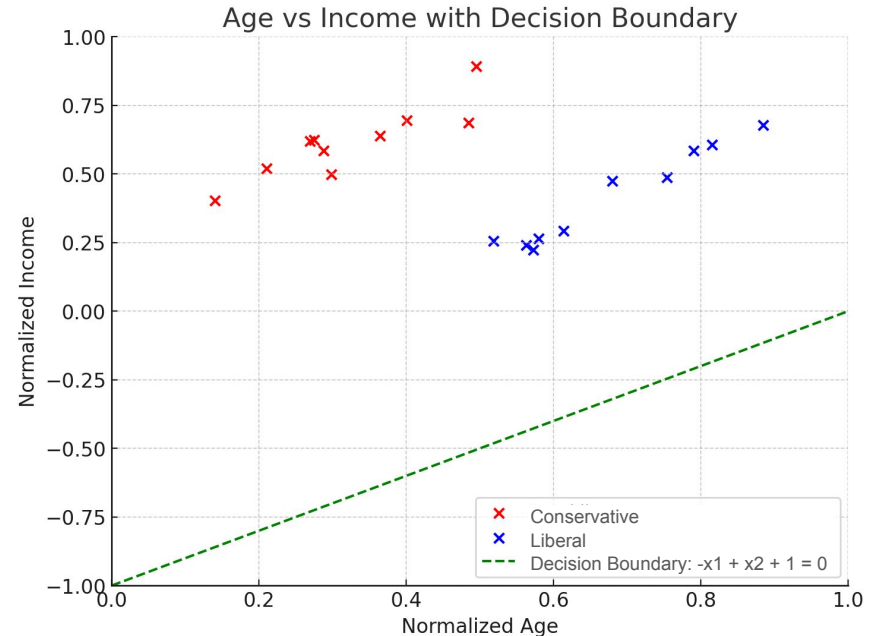
$$(x_1^{10}, x_2^{10}) = (0.513, 0.255) \quad \zeta^{10} = (-1)$$

$$(w_1, w_2) = (-1, 1) \quad w_0 = 1$$

$$\eta = 0.1$$

$$\Delta w = \eta(\zeta^\mu - o^\mu)x^\mu$$

$$w^{\text{nuevo}} = w^{\text{anterior}} + \Delta w$$



EJEMPLO NUMÉRICO

$$\Delta w = \eta(\zeta^\mu - O^\mu)x^\mu$$

$$w^{nuevo} = w^{anterior} + \Delta w$$

$$(x_1^{10}, x_2^{10}) = (0.513, 0.255) \quad \zeta^{10} = (-1)$$

$$(w_1, w_2) = (-1, 1) \quad w_0 = 1$$

$$O^{10} = \theta(x_1^{10} \cdot w_1 + x_2^{10} \cdot w_2 + w_0)$$

$$O^{10} = \theta(-0.513 + 0.255 + 1) = \theta(0.742) = 1$$

EJEMPLO NUMÉRICO

$$\Delta w = \eta(\zeta^\mu - O^\mu)x^\mu$$

$$w^{nuevo} = w^{anterior} + \Delta w$$

$$(x_1^{10}, x_2^{10}) = (0.513, 0.255) \quad \zeta^{10} = (-1)$$

$$(w_1, w_2) = (-1, 1) \quad w_0 = 1$$

$$O^{10} = \theta(-0.513 + 0.255 + 1) = \theta(0.742) = 1$$

El valor esperado es (-1)

$$\Delta w = 0, 1 \cdot (-2) \cdot (0.513, 0.255)$$

$$\Delta b = 0, 1 \cdot (-2)$$

EJEMPLO NUMÉRICO

$$(x_1^{10}, x_2^{10}) = (0.513, 0.255) \quad \zeta^{10} = (-1)$$

$$\Delta w = 0, 1 \cdot (-2) \cdot (0.513, 0.255)$$

$$\Delta b = 0, 1 \cdot (-2)$$

$$w_{nuevo} = (-1, 1) + (-0.1026, -0.051)$$

$$w_{nuevo} = (-1.1026, 0.949)$$

$$b_{nuevo} = 1 - 0.2 = 0.8$$

$$\Delta w = \eta(\zeta^\mu - O^\mu)x^\mu$$

$$w^{nuevo} = w^{anterior} + \Delta w$$

EJEMPLO NUMÉRICO

$$(x_1^{10}, x_2^{10}) = (0.513, 0.255) \quad \zeta^{10} = (-1)$$

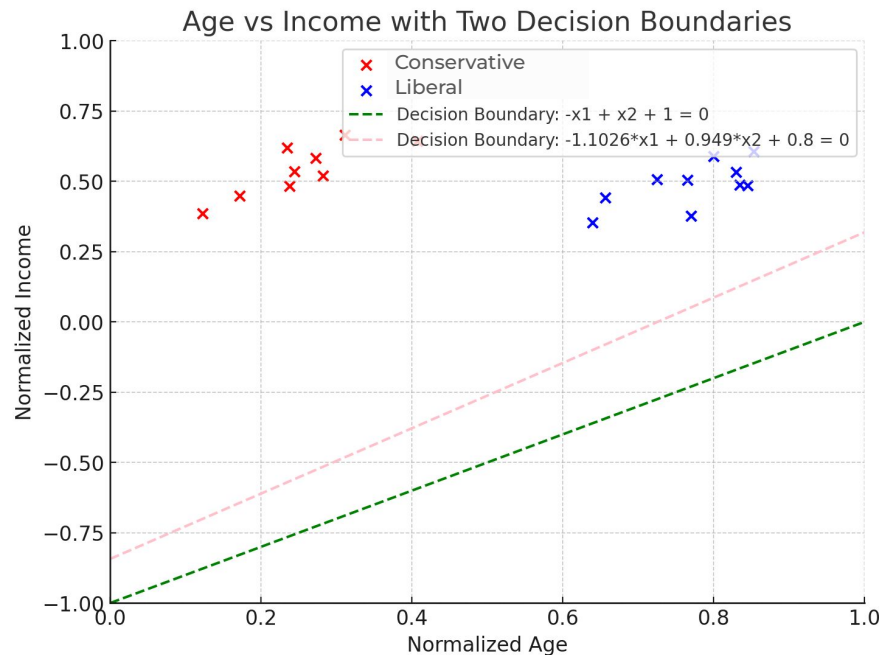
$$(w_1, w_2) = (-1.1026, 0.949) \quad w_0 = 0.8$$

$$o^{10} = \theta(-0.565 + 0.242 + 0.8)$$

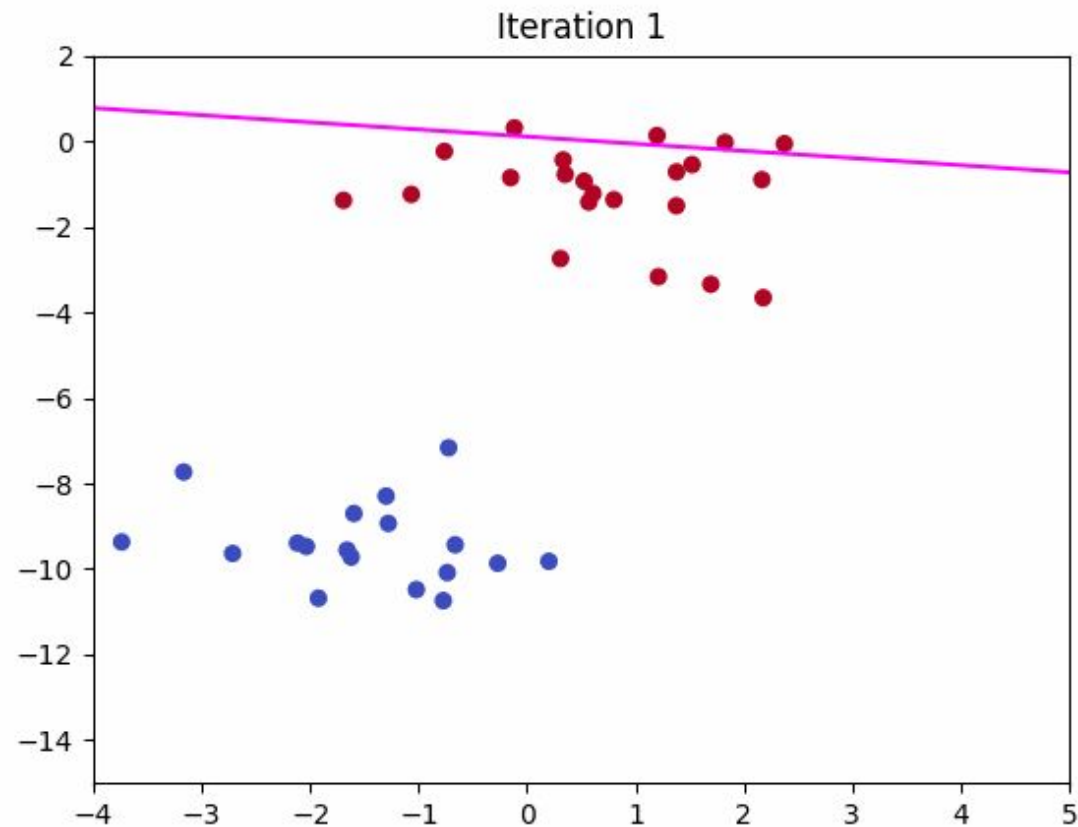
$$o^{10} = \theta(0.477) = 1$$

$$\Delta w = \eta(\zeta^\mu - o^\mu)x^\mu$$

$$w^{\text{nuevo}} = w^{\text{anterior}} + \Delta w$$



GRAFICAR ACTUALIZACIÓN DE LOS PESOS SINÁPTICOS



ALGORITMO DE ENTRENAMIENTO DEL PERCEPTRÓN SIMPLE

```
Initialize weights w to small random values
Initialize bias b to a small random value
Set learning rate  $\eta$ 

for a fixed number of epochs:
    For each training example  $\mu$  in the dataset:
        1. Calculate the weighted sum:
            
$$h^\mu = w_1 * x^\mu_1 + w_2 * x^\mu_2 + \dots + w_n * x^\mu_n + b$$


        2. Compute activation given by  $\Theta$ :
            
$$O(h^\mu) = \Theta(h^\mu) = 1 \text{ if } h^\mu > 0 \text{ else } 0$$


        3. Update the weights and bias:
            For each weight  $w_i$ :
                
$$w_i = w_i + \eta * (y - \text{output}) * x^\mu_i$$

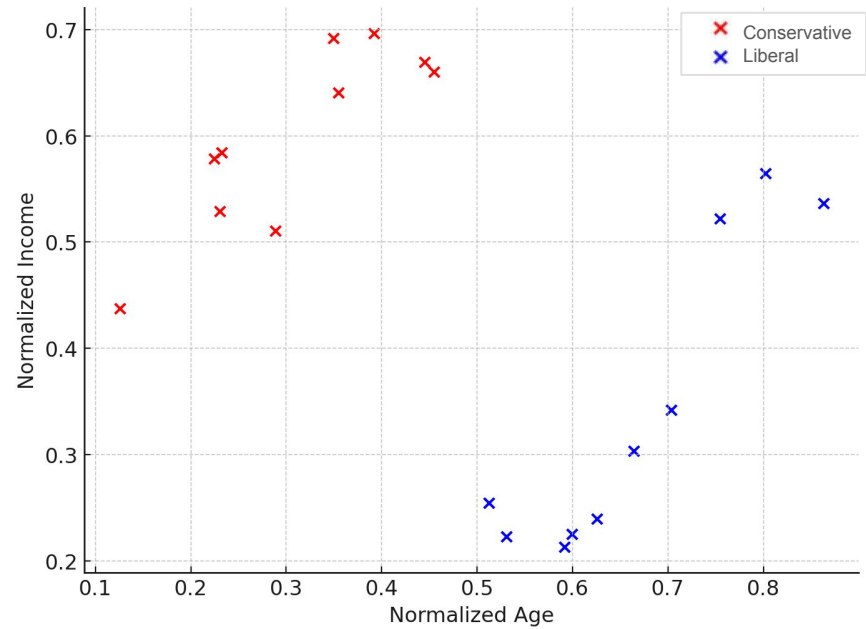
            Update bias:
                
$$b = b + \eta * (y - \text{output})$$


        4. Calculate perceptron error:
            
$$\text{error} = f(x^\mu_1, x^\mu_2, \dots, x^\mu_n)$$

            convergence = True if error <  $\varepsilon$  else False
            if convergence: break
```

End

AGREGAR EL UMBRAL/BIAS



x0	Age	Income	Party
1.0	0.445	0.669	1.0
1.0	0.349	0.692	1.0
1.0	0.232	0.585	1.0
1.0	0.125	0.438	1.0
1.0	0.224	0.579	1.0
1.0	0.23	0.529	1.0
1.0	0.392	0.696	1.0
1.0	0.355	0.641	1.0
1.0	0.455	0.66	1.0
1.0	0.289	0.51	1.0
1.0	0.513	0.255	-1.0
1.0	0.755	0.522	-1.0
1.0	0.626	0.24	-1.0
1.0	0.703	0.342	-1.0
1.0	0.863	0.536	-1.0
1.0	0.6	0.225	-1.0
1.0	0.664	0.303	-1.0
1.0	0.802	0.565	-1.0
1.0	0.592	0.213	-1.0
1.0	0.531	0.223	-1.0

ALGORITMO DE ENTRENAMIENTO DEL PERCEPTRÓN SIMPLE

Initialize weights w to small random values

Set learning rate η

for a fixed number of epochs:

For each training example μ in the dataset:

1. Calculate the weighted sum:

$$h^\mu = w_0 * x_0^\mu + w_1 * x_1^\mu + w_2 * x_2^\mu + \dots + w_n * x_n^\mu$$

2. Compute activation given by θ :

$$o(h^\mu) = \theta(h^\mu) = 1 \text{ if } h^\mu > 0 \text{ else } 0$$

3. Update the weights and bias:

For each weight w_i :

$$w_i = w_i + \eta * (y - \text{output}) * x_i^\mu$$

4. Calculate perceptron error:

$$\text{error} = f(x_1^\mu, x_2^\mu, \dots, x_n^\mu)$$

convergence = True if error < ϵ else False

if convergence: break

End

Calcular el error del perceptrón

Distintas formas:

- La suma del valor absoluto de los errores devuelve cero
- Accuracy es del 100% (para problemas de clasificación)
- ...

¿PARA QUÉ NOS SIRVE ESTA HERRAMIENTA?

Aprendizaje

Refiere al proceso de entrenar un **perceptrón** sobre un conjunto de datos, con el objetivo de minimizar el error o la función de costo de la red sobre las entradas del conjunto de datos.

Generalización

Refiere a la habilidad del **perceptrón** de desempeñarse correctamente sobre datos que no fueron alimentados durante el entrenamiento

RESUMEN

- McCulloch y Pitts sientan las bases del modelo de neurona que se utiliza en el área de redes neuronales. Este modelo se denomina **Perceptrón**.
- El modelo de McCulloch y Pitts permite resolver problemas linealmente separables.
- Rosenblatt provee el mecanismo que permite obtener los pesos del perceptrón de manera iterativa
- No es lo mismo aprendizaje que generalización

BIBLIOGRAFÍA

Rodrigo Ramele (2024) *Reglamento y Apuntes de Sistemas de Inteligencia Artificial*, Capítulo 6.

McCulloch, W.S., Pitts, W. *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics 5, 115–133 (1943).
<https://doi.org/10.1007/BF02478259>

Rosenblatt, F. (1958). *The perceptron: A probabilistic model for information storage and organization in the brain*. Psychological Review, 65(6), 386–408.
<https://doi.org/10.1037/h0042519>

Łukasz Gebel (2021), *Why We Need Bias in Neural Networks*.
<https://towardsdatascience.com/why-we-need-bias-in-neural-networks-db8f7e07cb98>

