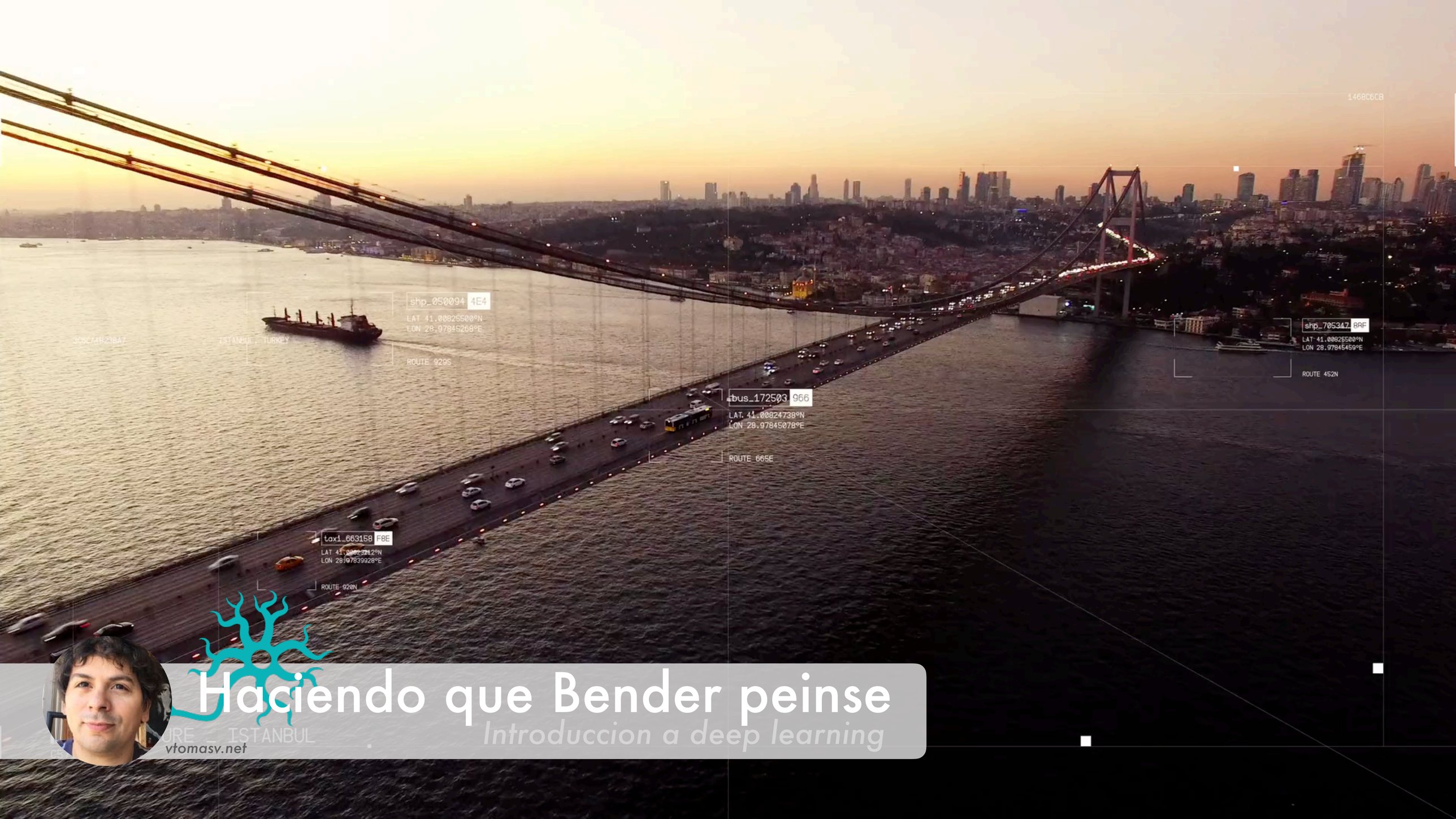


Haciendo que Bender peinse

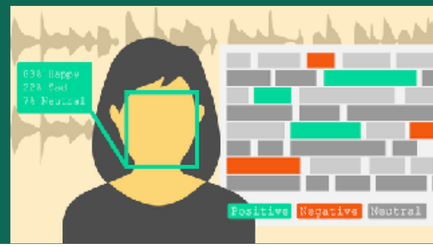
Introduccion a deep learning

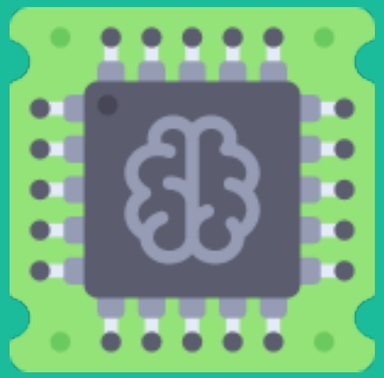
vtomasv.net

Artificial Intelligence

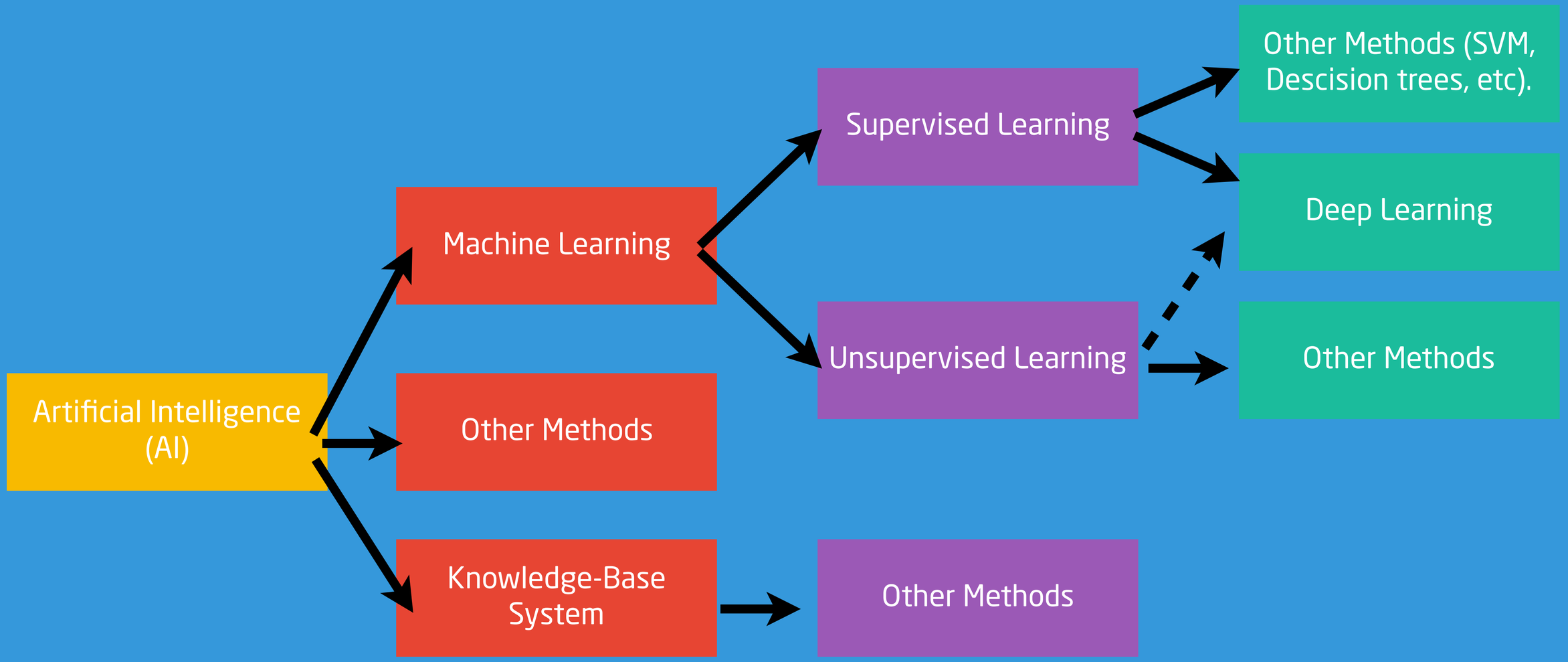
Machine Learning

Deep Learning





1950 1960 1970 1980 1990 2000 2010



De que hablamos cuendo hablamos de Deep Learning

Es una película excelente, si lo que quieres es perder el tiempo y tienes poco cerebro.

¿Inteligencia?

Machine Learning

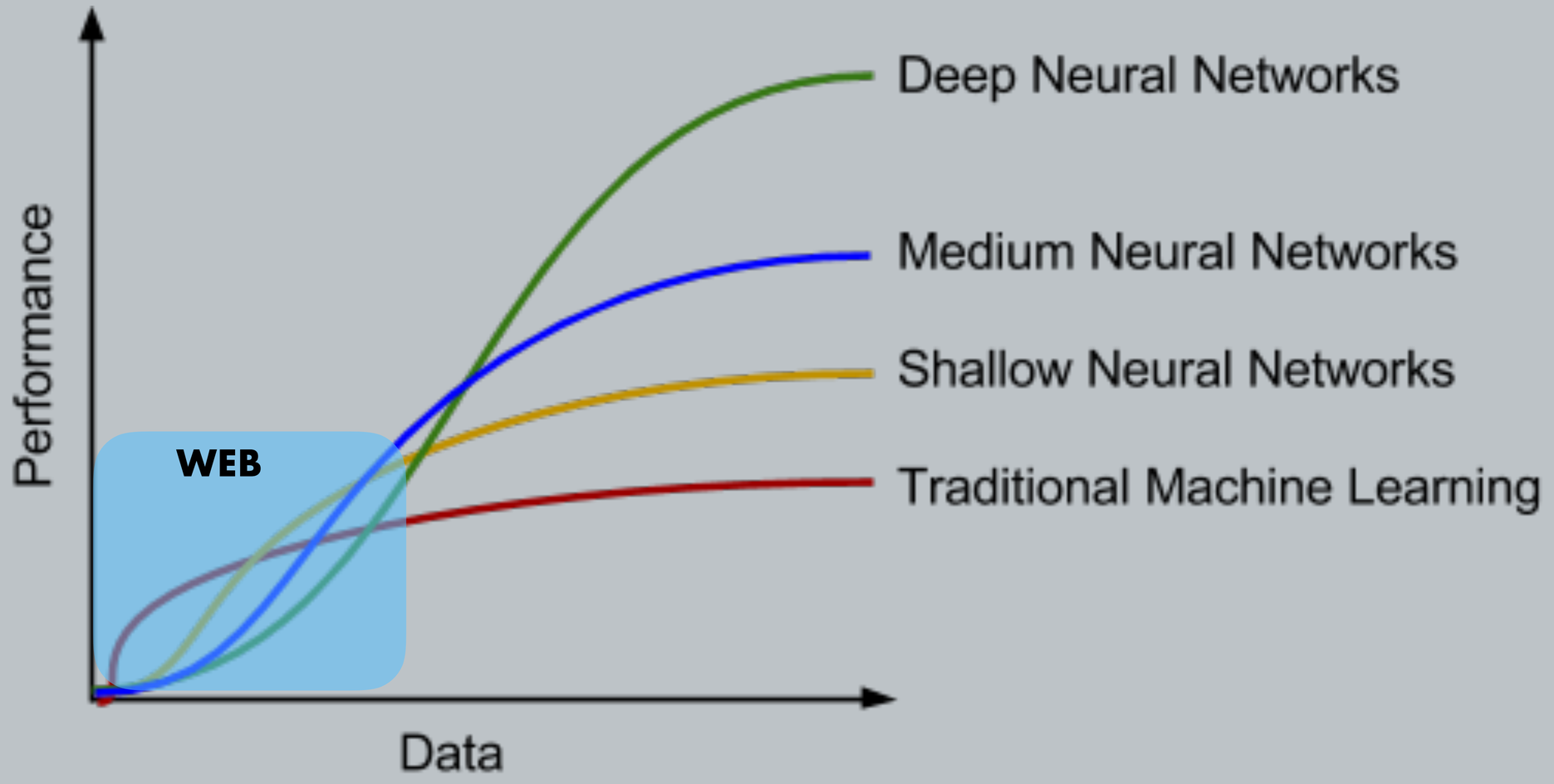
Encontrar características en textos, sentimiento , etc.

Deep Learning

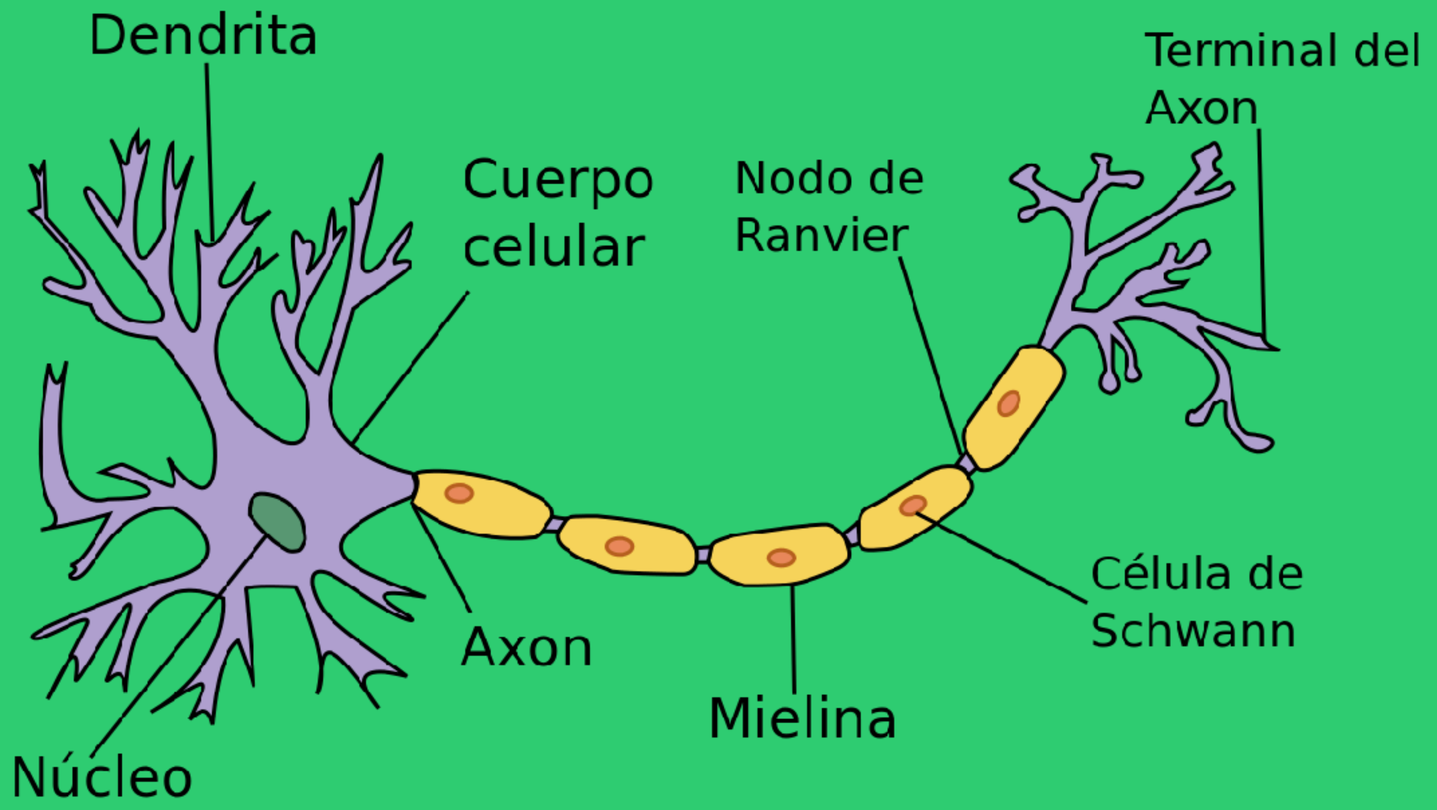
Automatiza el procesamiento y se basa en la clasificación estadística.

Estas técnicas existen desde hace mucho tiempo!

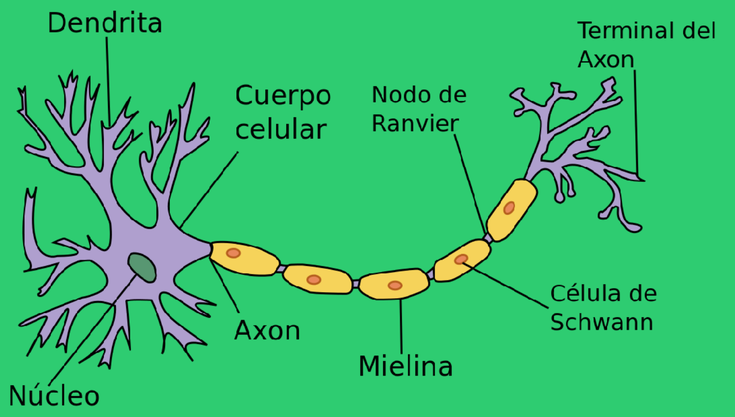
¿Cual es la real diferencia?



¿Cantidad de Datos vs Calidad de Datos?



Redes Neuronales



Bias

W_0

X_1

W_1

X_2

W_2

Entrada

W_3

X_3

$\mathcal{F}$

y

Salida

Función de activación

Perceptron

$$X_1 * W_1$$
$$X_2 * W_2 + W_0 = C$$
$$X_3 * W_3$$

Perceptron

$$y = \mathcal{F}_{(c)}$$

Valor

$$W*X+b = Y$$

Weights

Predicción

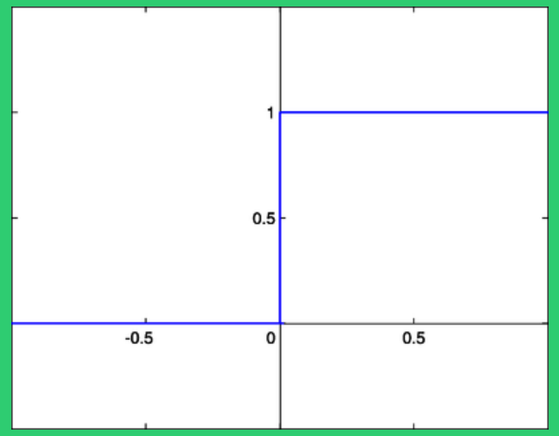
Trained

Bias

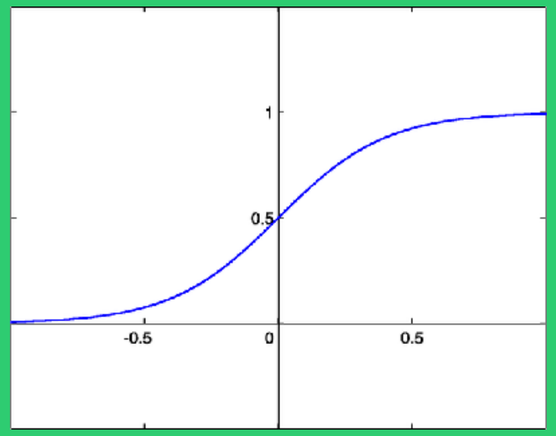
Funciones de activación

$$y = \mathcal{F}_{(c)}$$

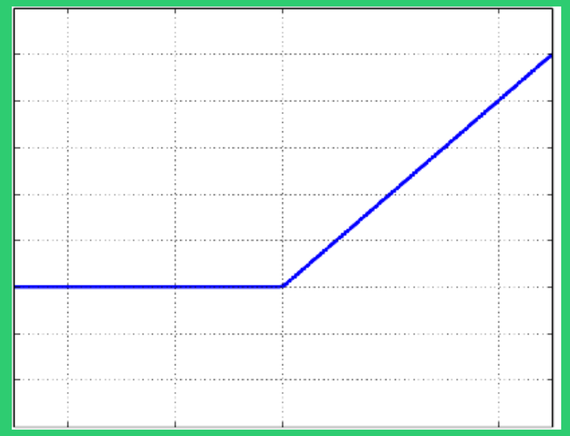
Binary Step



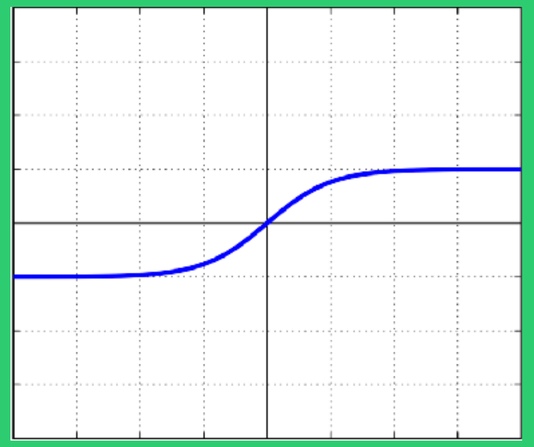
Sigma



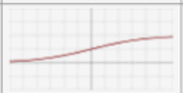
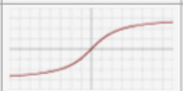
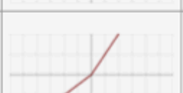
Relu



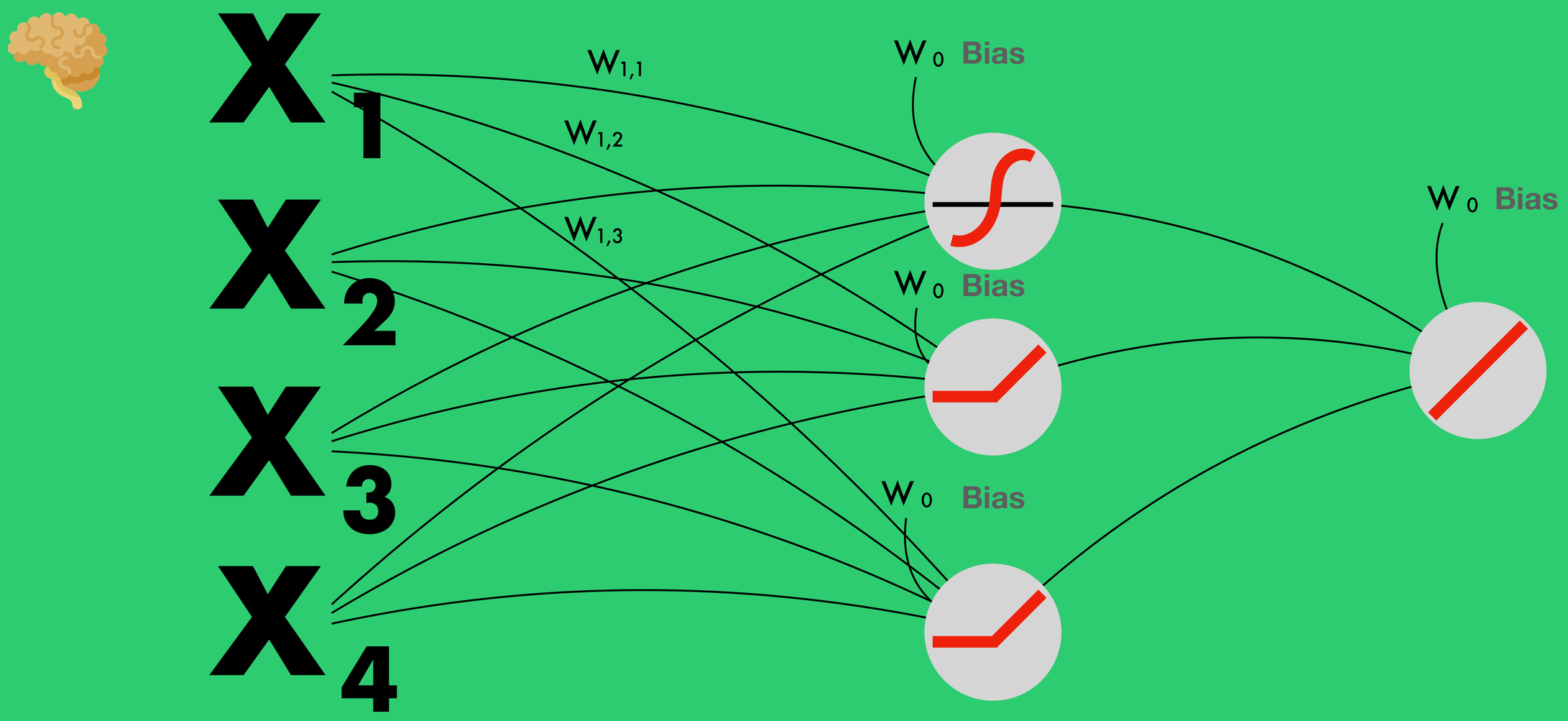
Tanh



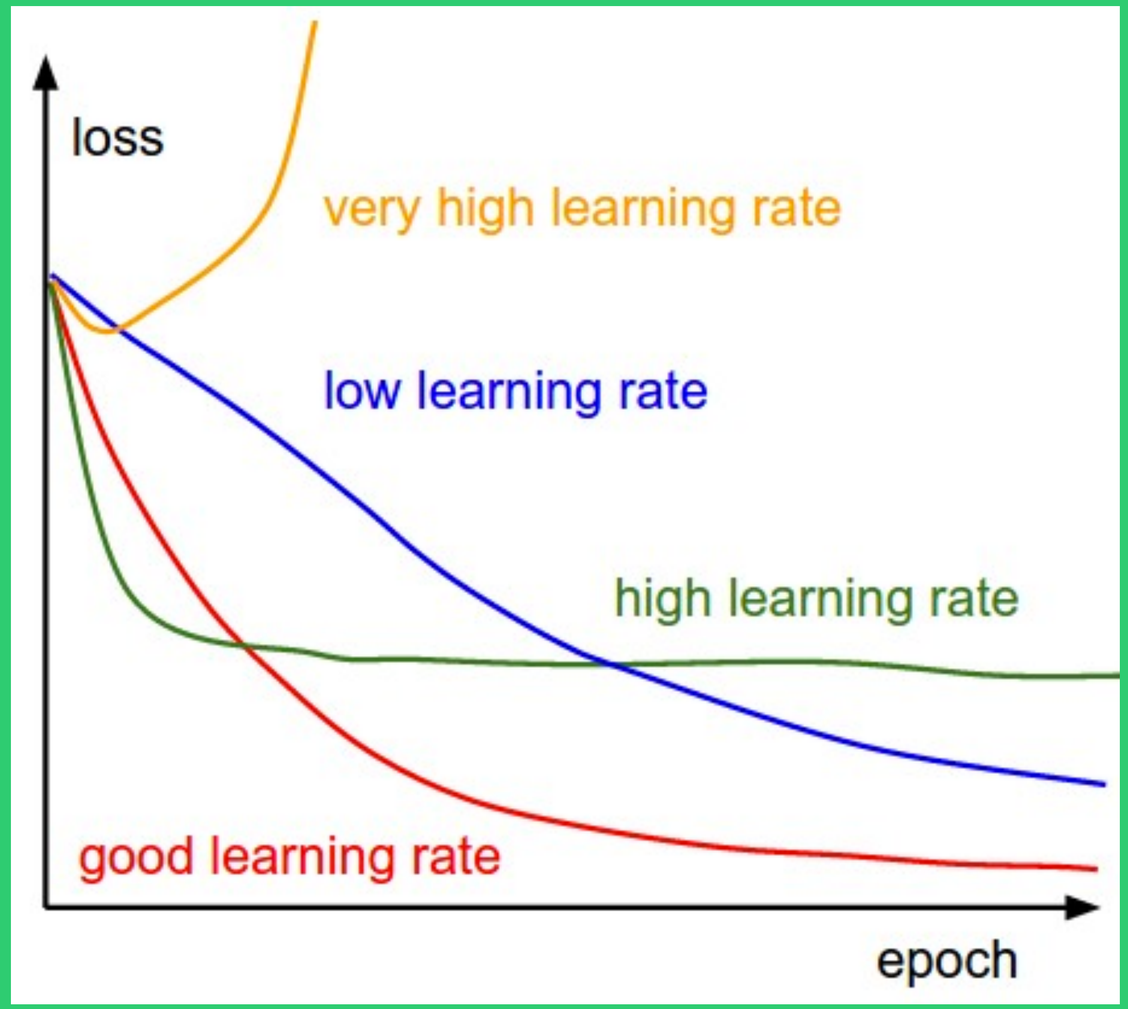
Funciones de activación

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1+e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Funciones de activación



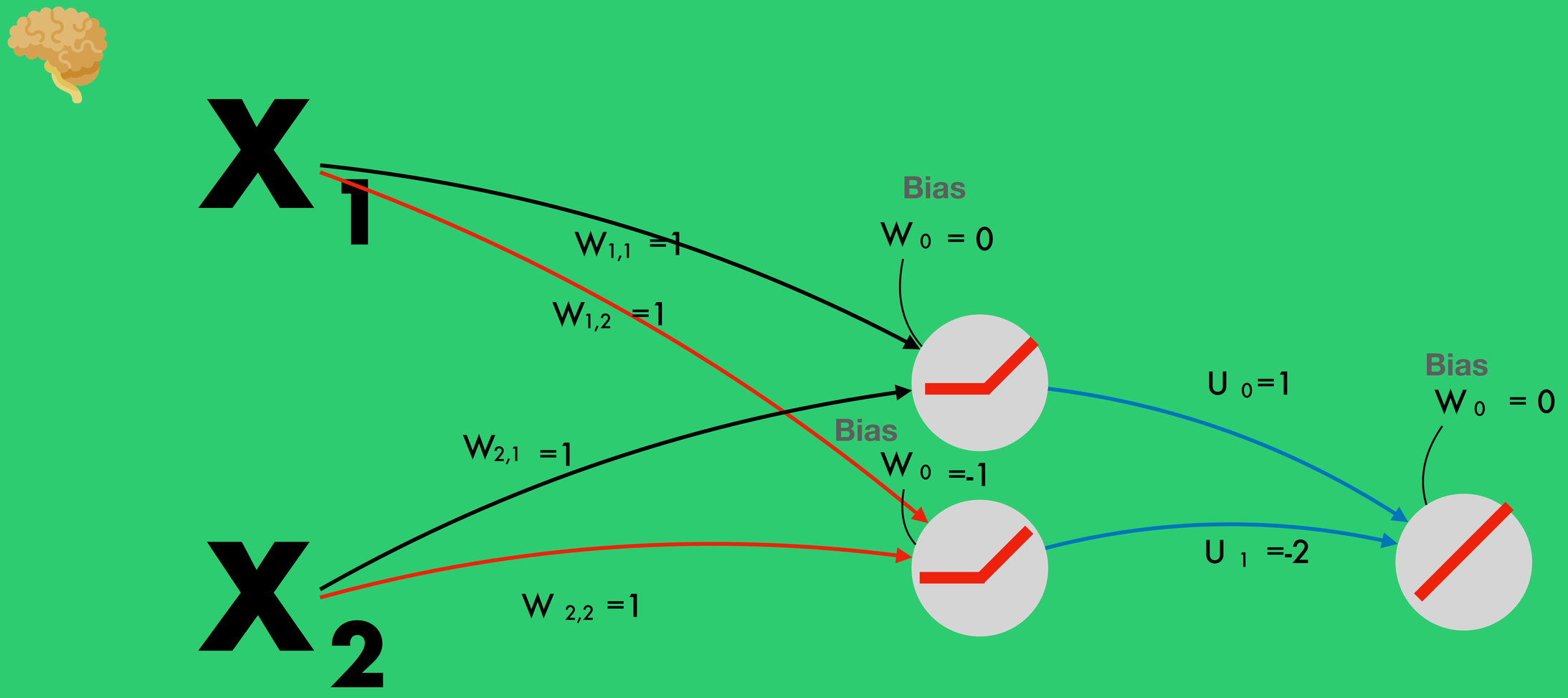
Funciones de activación



La red no puede aprender mas que los datos

A	B	A **XOR** B
0	0	0
0	1	1
1	0	1
1	1	0

XOR



XOR

$$X_1^{(1)} = 0 \quad \text{relu}((0*1)+(0*1) + 0) = 0$$

$$X_2^{(1)} = 0 \quad \text{relu}((0*1)+(0*1) + (-1)) = 0$$

$$\hat{Y} = \text{idem}(0*1 + 0*-2+0) = 0$$

Combinación 1

$$X_1^{(1)} = 1 \quad \text{relu}((1*1)+(0*1)+0) = 1$$

$$X_2^{(1)} = 0 \quad \text{relu}((1*1)+(0*1)+(-1)) = 0$$

$$\hat{Y} = \text{idem}(1*1+0*-2+0) = 1$$

Combinación 2

$$X_1^{(1)} = 0 \quad \text{relu}((0*1)+(1*1)+0) = 1$$

$$X_2^{(1)} = 1 \quad \text{relu}((0*1)+(1*1)+(-1)) = 0$$

$$\hat{Y} = \text{idem}(1*1+0*-2+0) = 1$$

Combinación 3

$$X_1^{(1)} = 1 \quad \text{relu}((1*1)+(1*1)+0) = 2$$

$$X_2^{(1)} = 1 \quad \text{relu}((1*1)+(1*1)+(-1)) = 1$$

$$\hat{Y} = \text{idem}(2*1 + 1*-2+0) = 0$$

Combinación *4*

Capa 1

$$\text{relu}(\ 1 * x_1 + 1 * x_2 + 0)$$

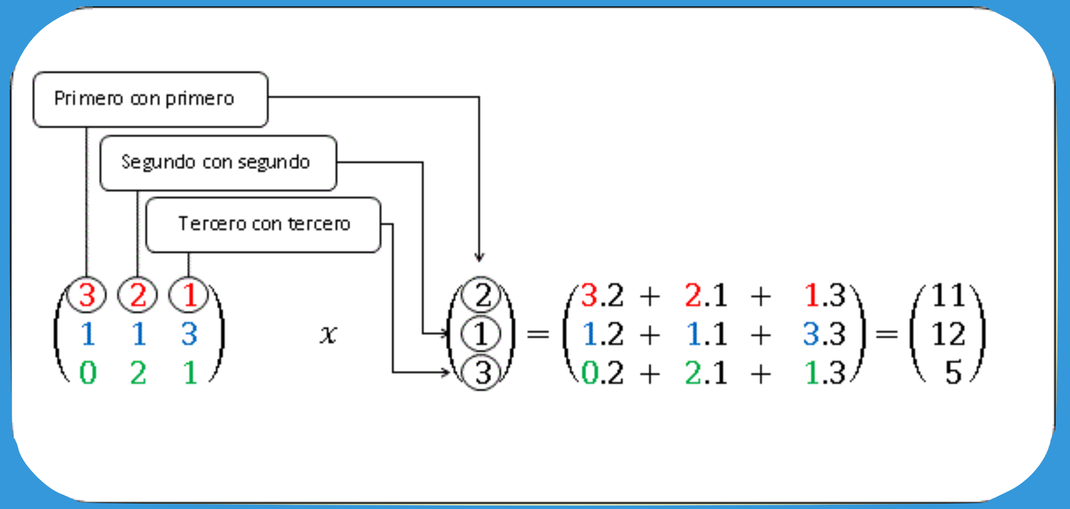
$$\text{relu}(\ 1 * x_1 + 1 * x_2 - 1)$$

$$\text{relu}\left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right)$$

¡ Veamos otro punto de vista !

Capa 2

$$\text{idem}\left(\begin{bmatrix} 1 & -2 \end{bmatrix} * \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right)$$



¡ Veamos otro punto de vista !

Vector

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

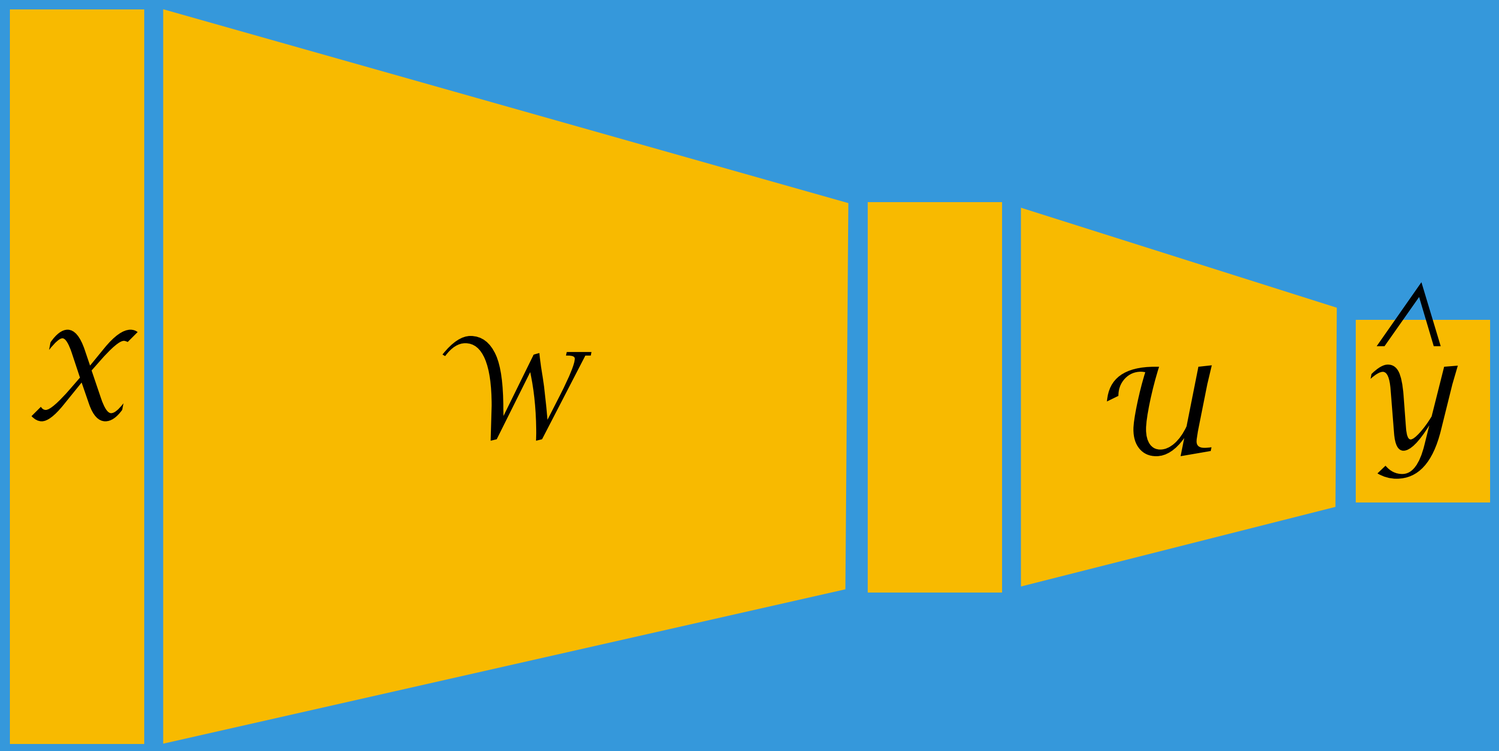
$$v = W * x + b$$

$$u = relu(v)$$

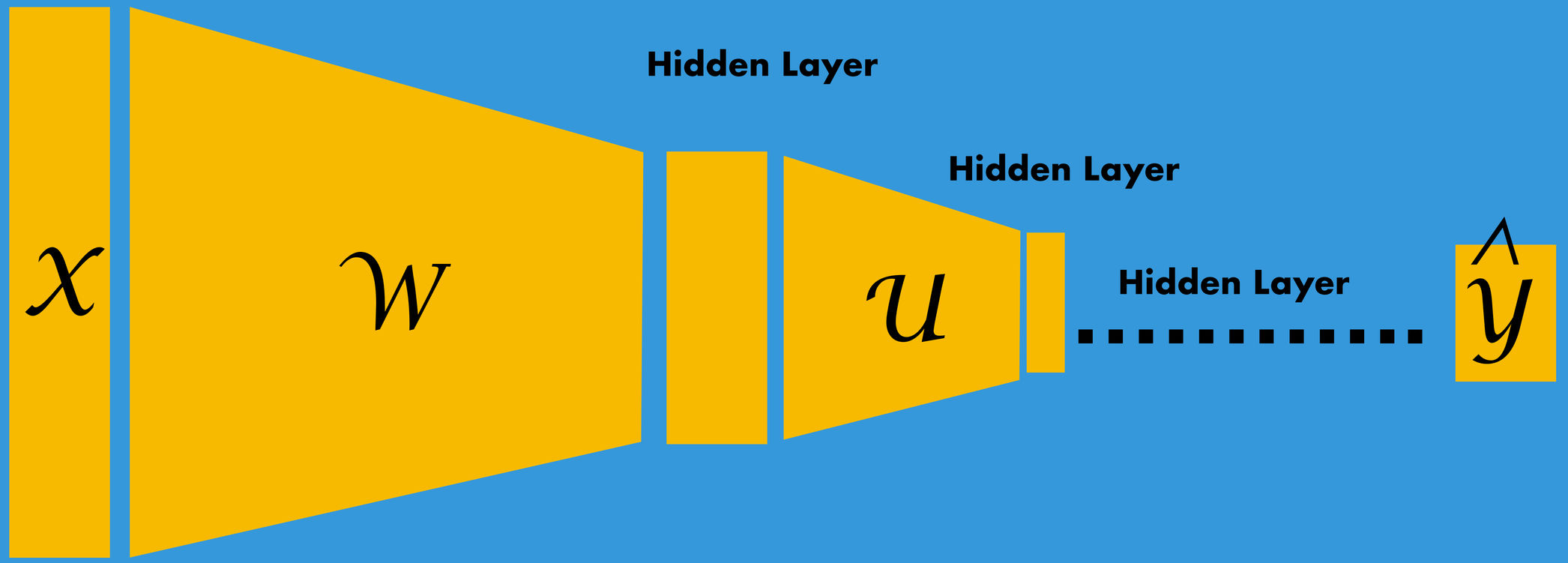
$$\hat{y} = Uu$$

De forma general

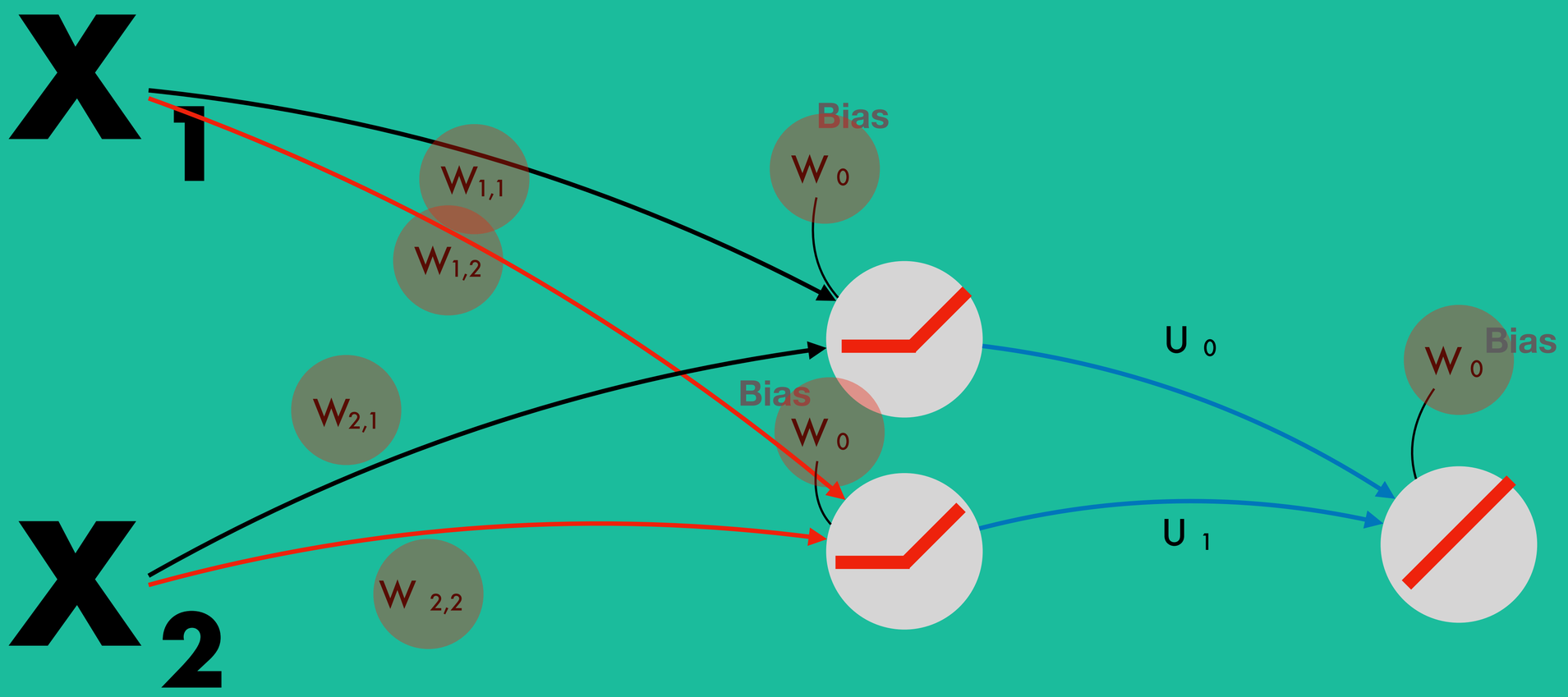
¿Pero que son esas capas?



Visto en capas .....



Visto en capas .....



Hiper parámetros

¿Redes que generan Redes?

Salidas

- **Clasificación Binaria**
- **Clasificación multiclase**

Cada salida es una probabilidad de que se de la clase

Problemas

Funciones

Softmax Function

$$F(X_i) = \frac{Exp(X_i) \quad i = 0, 1, 2, \ldots k}{\sum_{j=0}^{k} Exp(X_j)}$$

VS

Sigmoid Function

$$F(X_i) = \frac{1}{1 + Exp(-X_i)}$$

Como entreno ?

$$((X^{(1)}, Y^{(2)}), \ldots\ldots (X^{(n)}, Y^{(n)}))$$

$$\text{Error}\ (X^{(i)}, Y^{(i)})$$

Para todos los casos

$$\frac{1}{N} \sum \text{Error}\ (X^{(i)}, Y^{(i)})$$

Entrenamiento

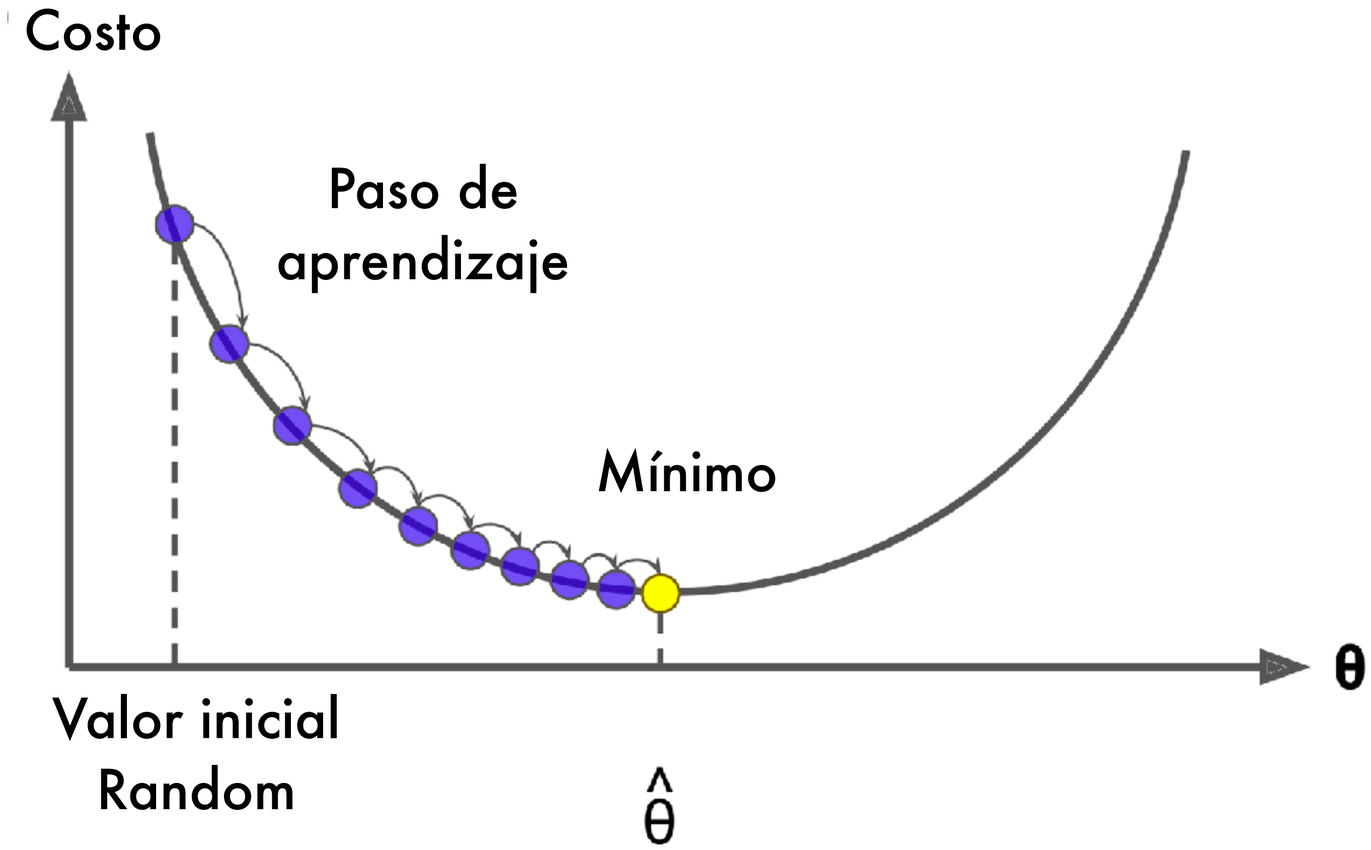
Debemos encontrar todos los valores de

$$W^{(i)} \; U^{(i)} \; \ldots\ldots \; U^{(i+n)} = \theta$$

$$\frac{1}{N} \sum \text{Error}\,(X^{(i)}, Y^{(i)}) = \mathcal{L}$$

Y por donde comenzamos?

Ahora debemos achicar el error



Ahora entrenamos con descenso de gradiente

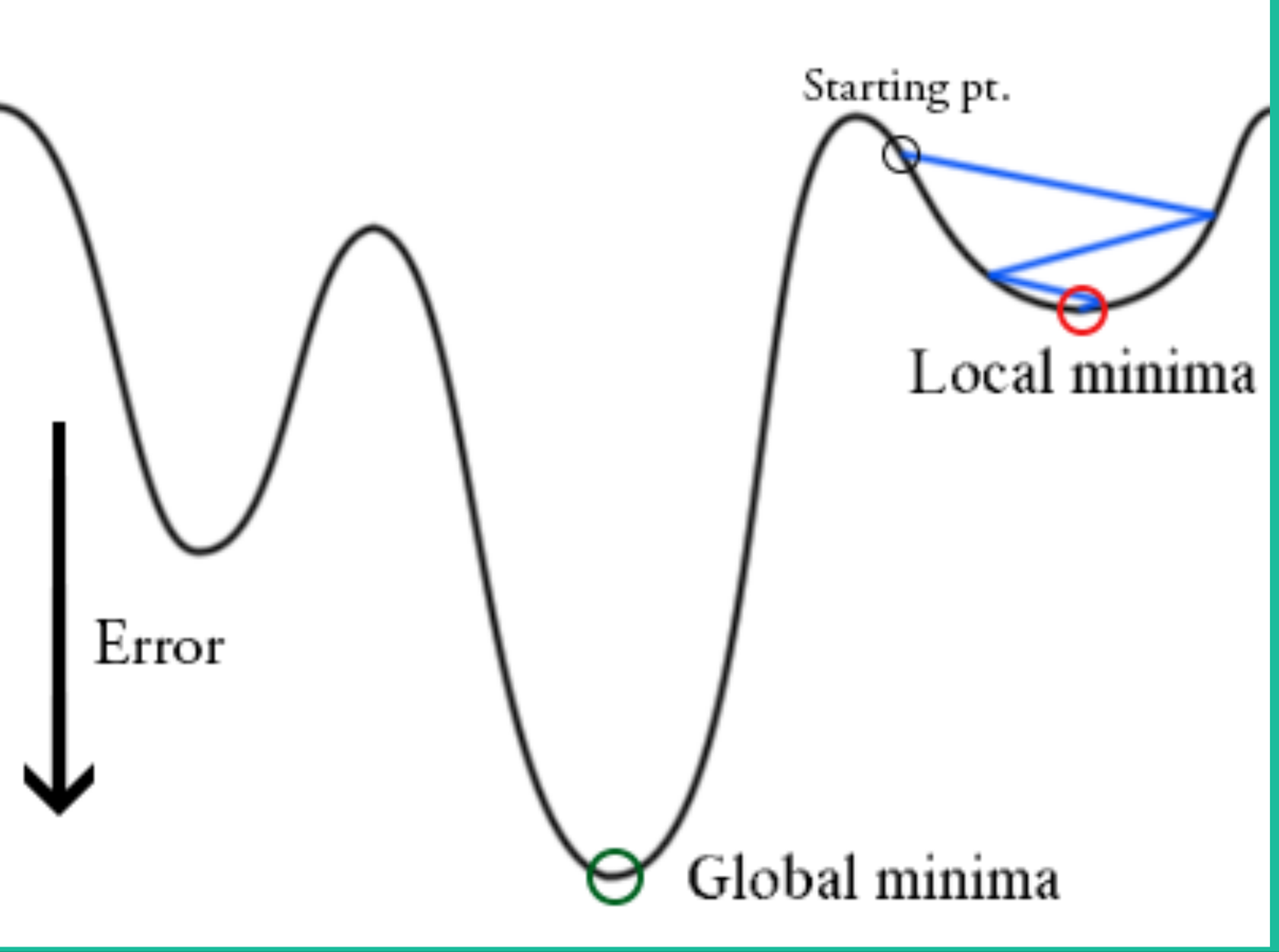
Lo que aprendemos

(Ademas es otro hiperparametro)

Error

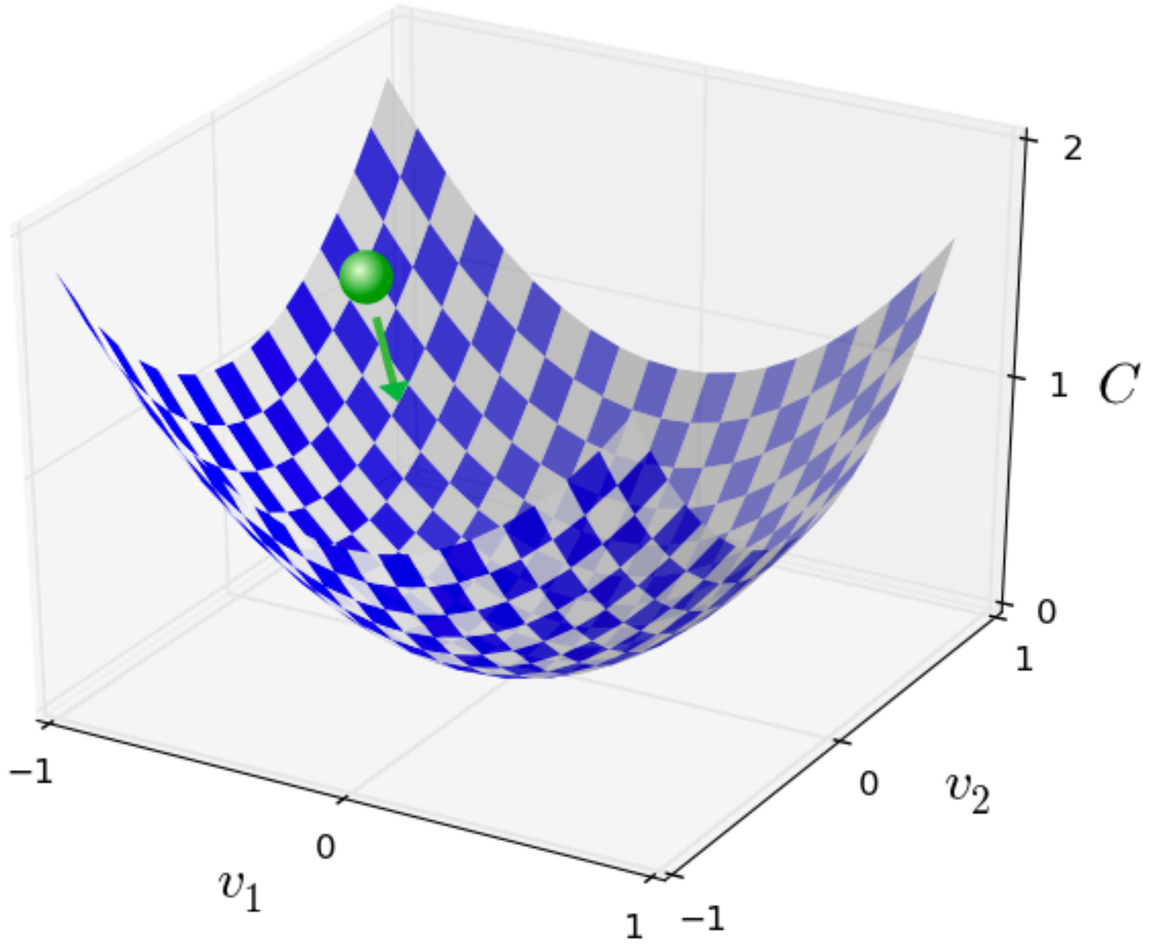
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1, \theta_2)$$

Gradiente = Derivar



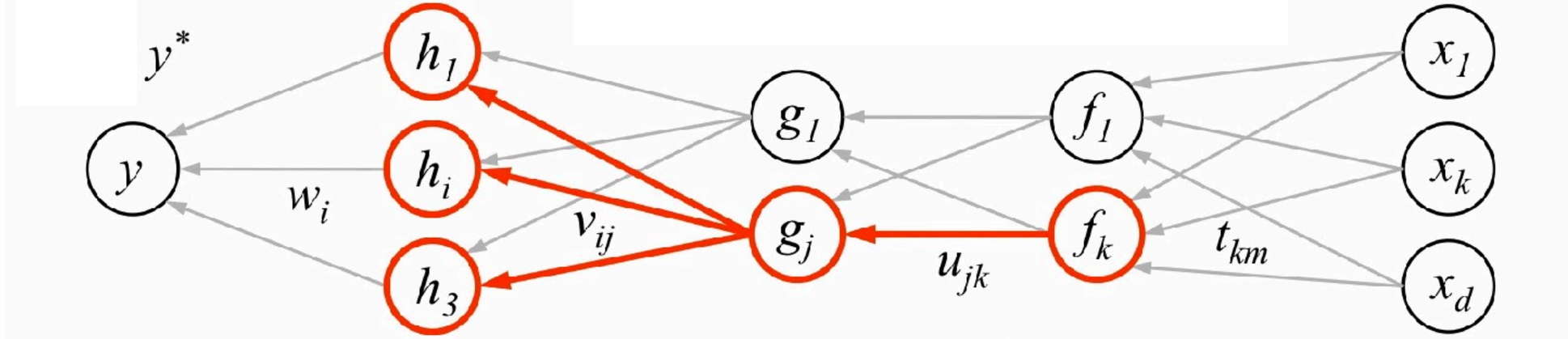
Como evitamos caer en mínimos locales???

Descenso estocastico de gradiente



Tomamos un rango (conjunto) y calculamos el error. Luego pasamos a otro rango y se calcula nuevamente el error

Descenso estocastico de gradiente

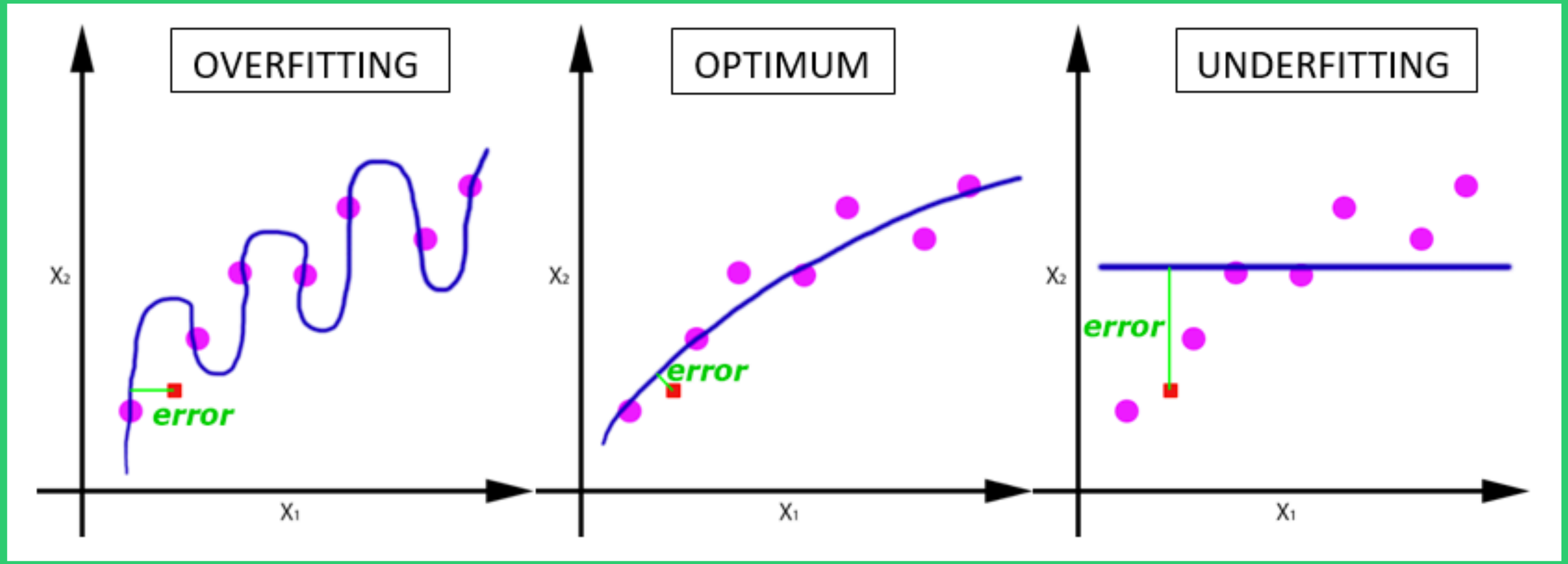


1) Se elige una salida valida y una entrada valida
2) Se hace el calculo al revés en la red , por cada **gj** en cada capa se computa **gj** basado en unidades de **fk** de la capa anterior

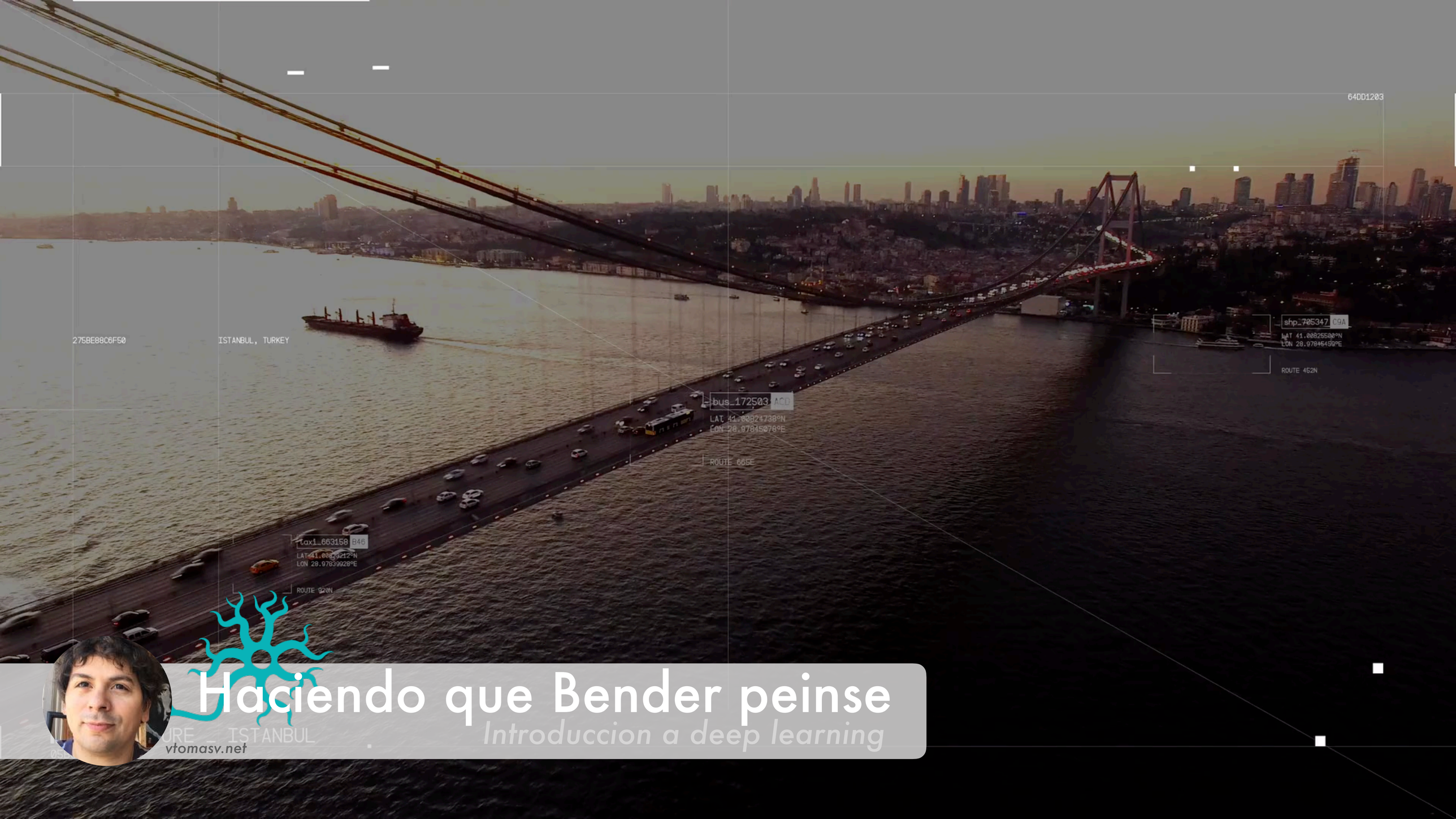
$$g_j = \sigma\left(u_{j0} + \sum_k u_{jk} f_k\right)$$

3) Se obtiene la prediccion y el error
4) Se propaga el error por cada unidad **gj** en cada capa

Back Propagation



Optimizando NN



Haciendo que Bender peinse

Introduccion a deep learning

vtomasv.net