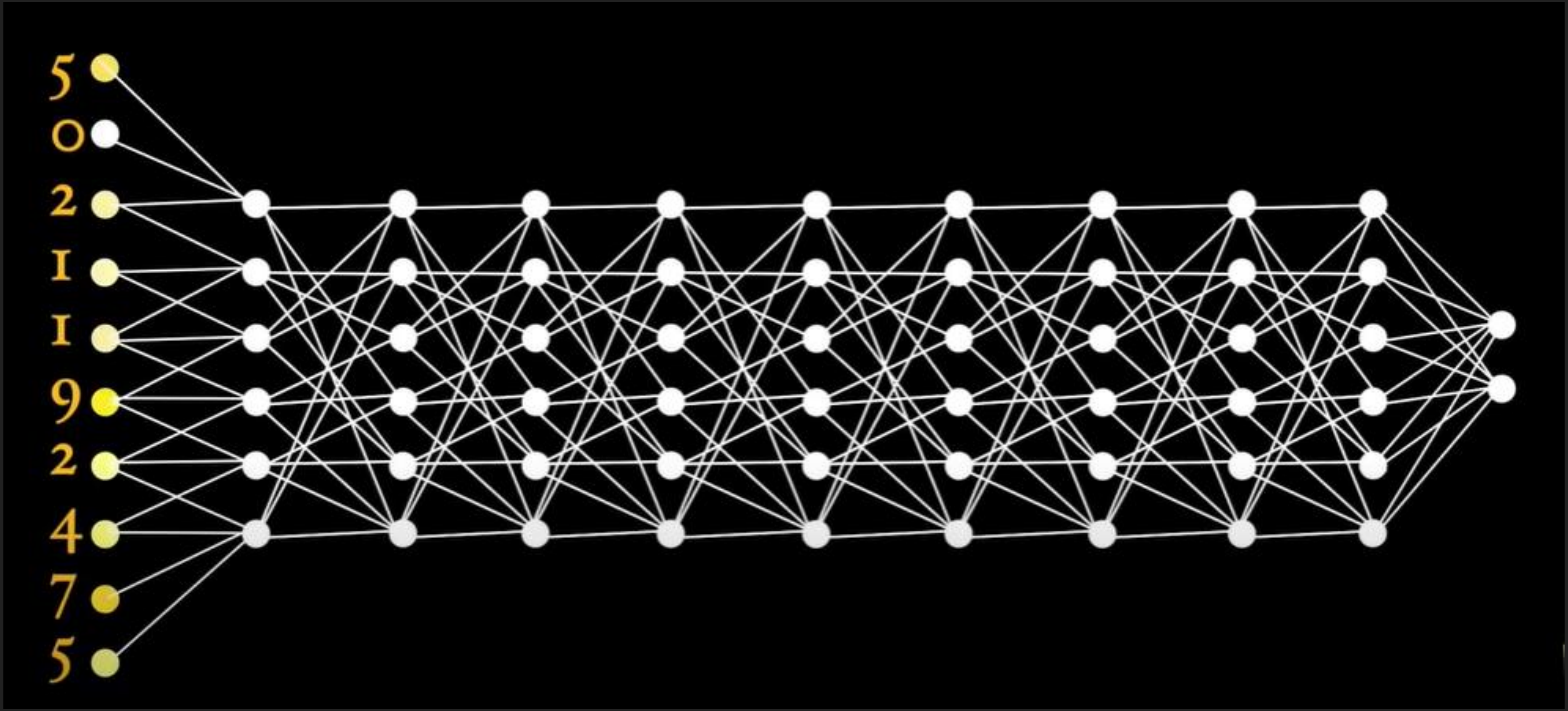
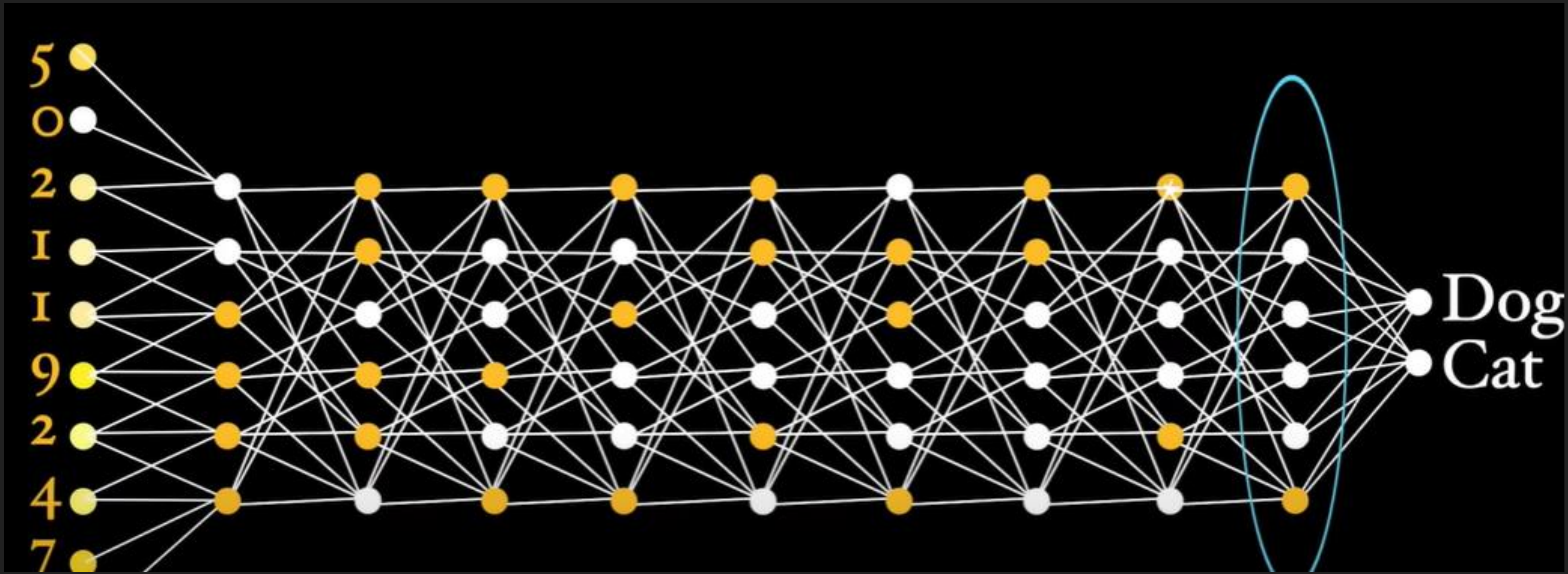
Redes neuronales

Andrés Daniel Godoy Ortiz

Muchos recursos tomados de 3blue1Brown y Emergent Garden

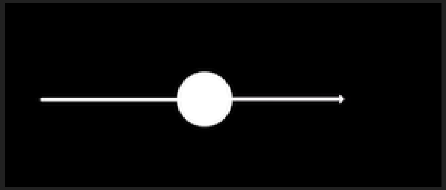


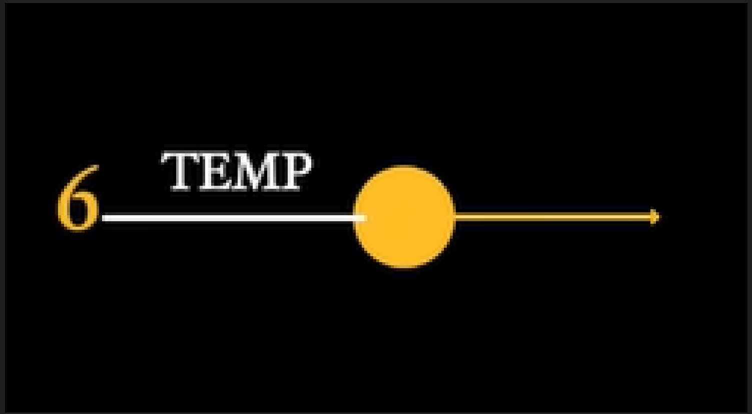
Datos, lista de medidas o Percepciones

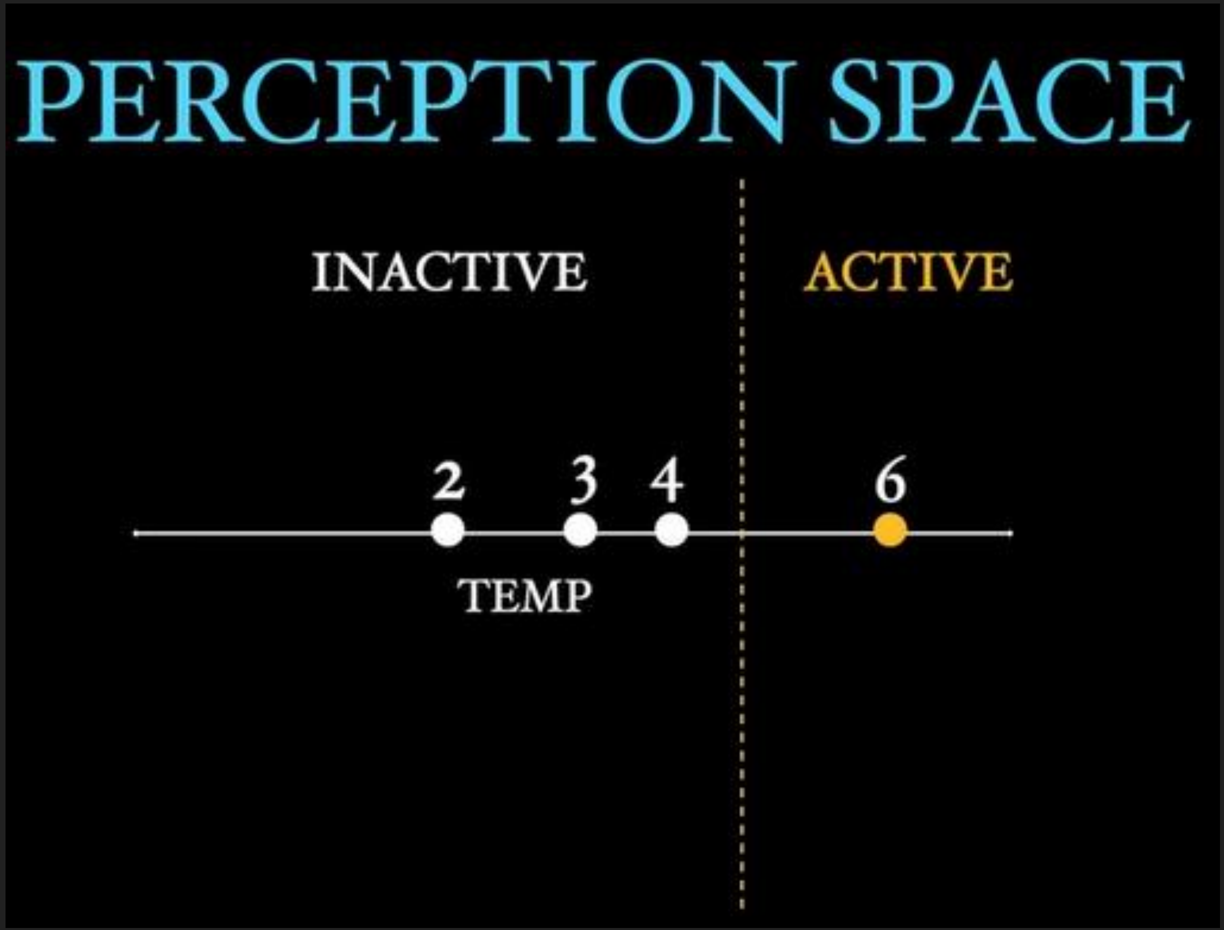


Predicción o conceptos

Una sola neurona, una entrada una salida



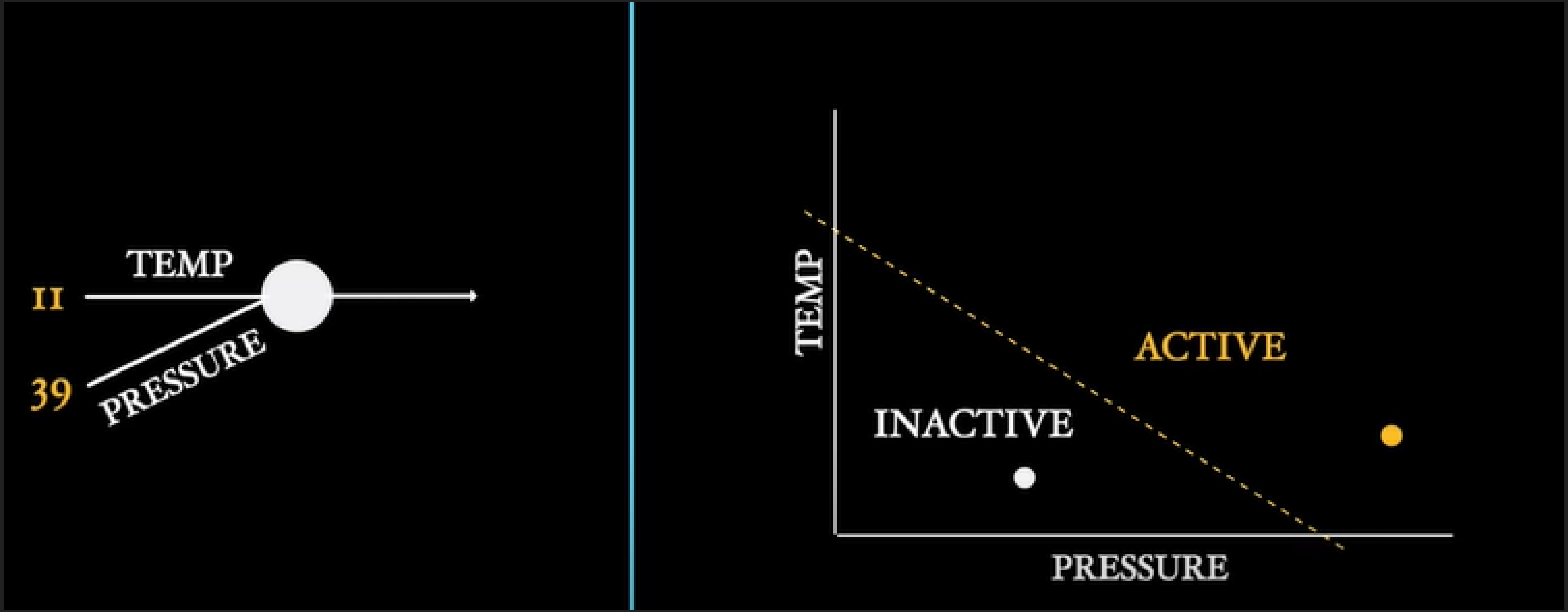


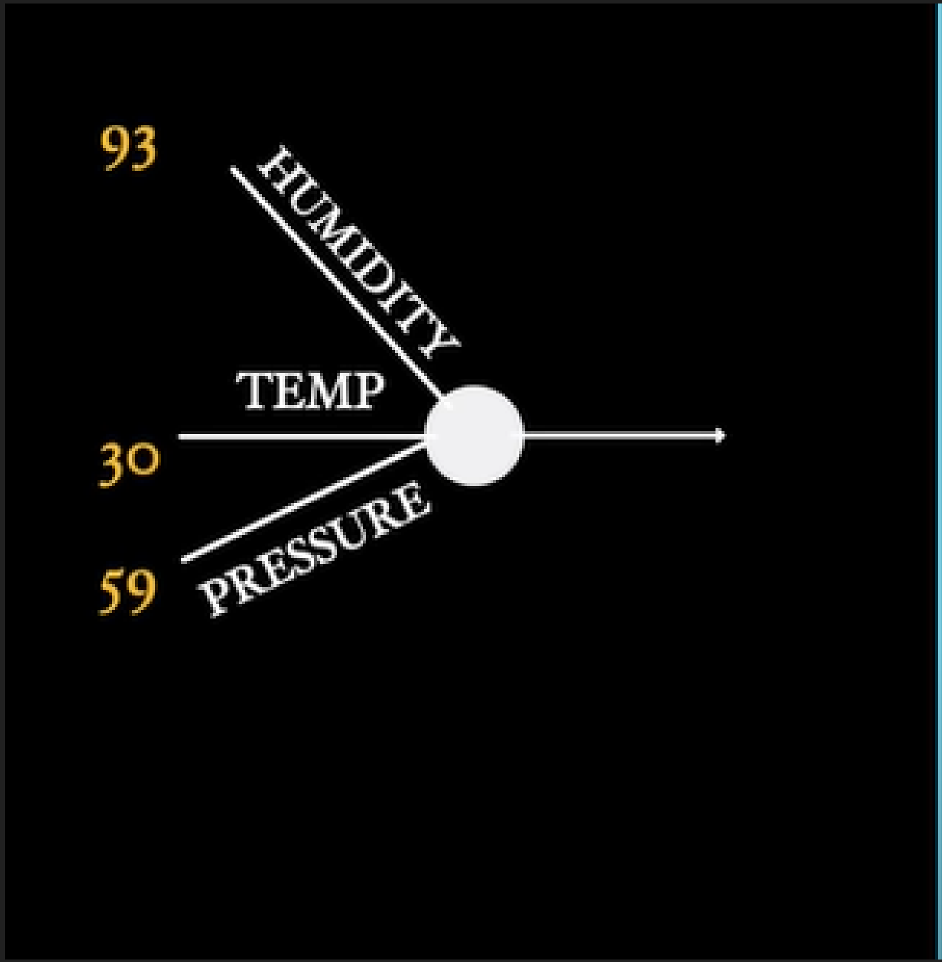


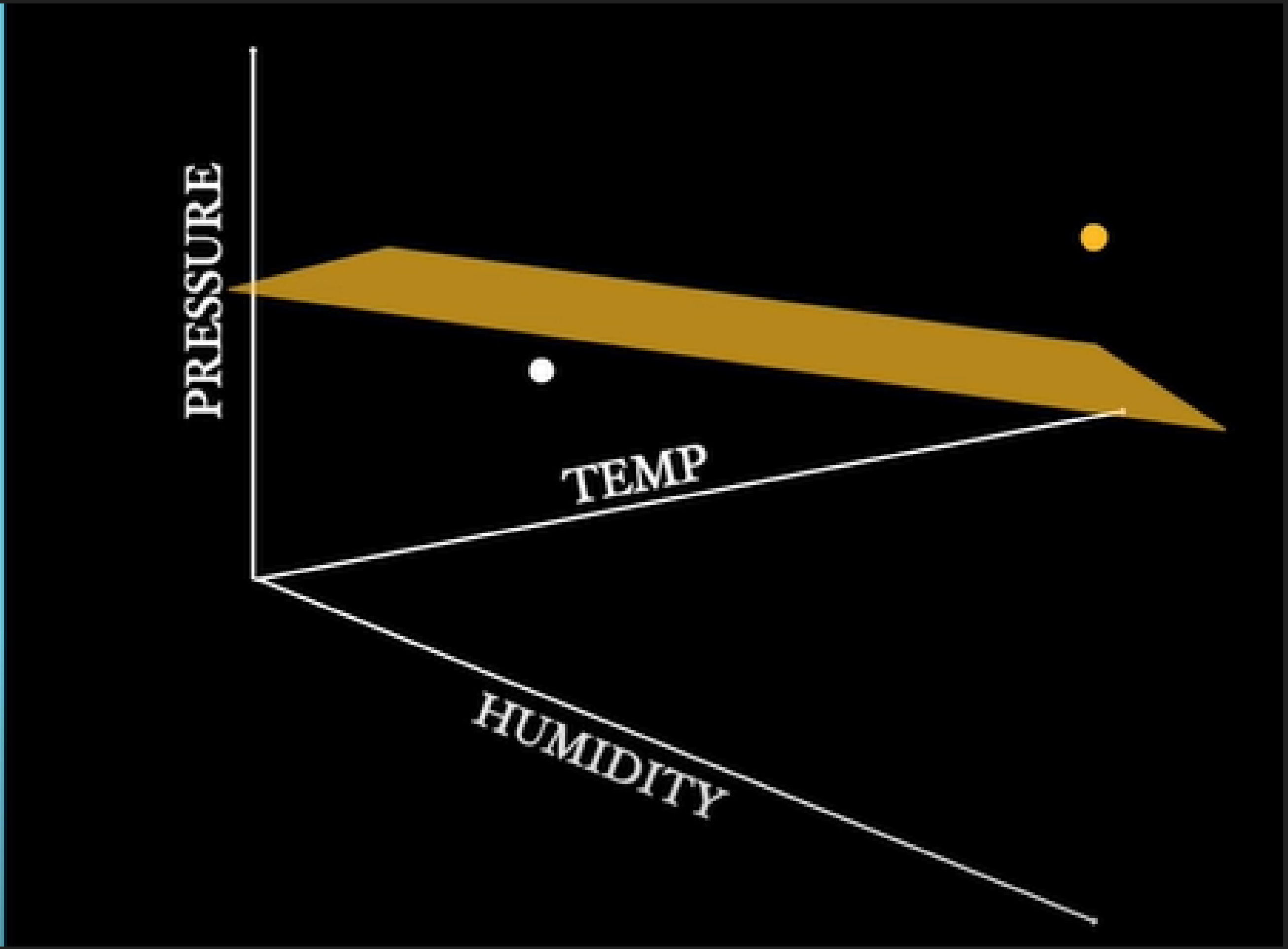
Las mediciones de la realidad (datos) son vectores del espacio de percepciones o características

Entrenar una neurona significa, mover esa línea O hiperplano (olvidémonos por un segundo de la función de activación.

Ajustar la separación del espacio de características.


TEMP
11
39
PRESSURE
TEMP
ACTIVE
INACTIVE
PRESSURE


93
HUMIDITY
TEMP
30
PRESSURE
59


PRESSURE
TEMP
HUMIDITY

¿Un hiperplano? ¿por qué?

¿Cómo lo represento matemáticamente?

En $\mathbb{R}^n$, un **hiperplano** es un subespacio de dimensión $n-1$.

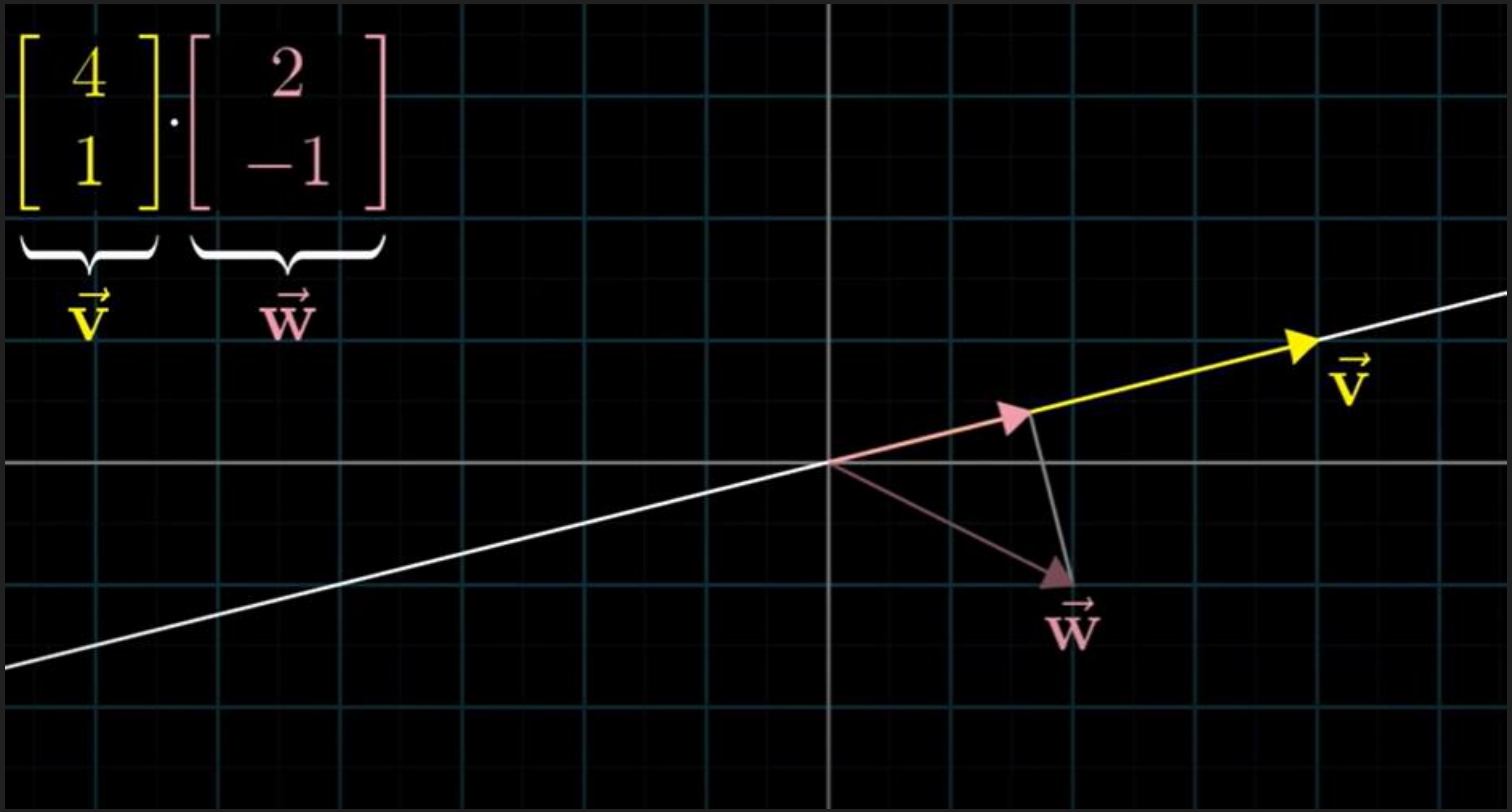
Su ecuación general es:

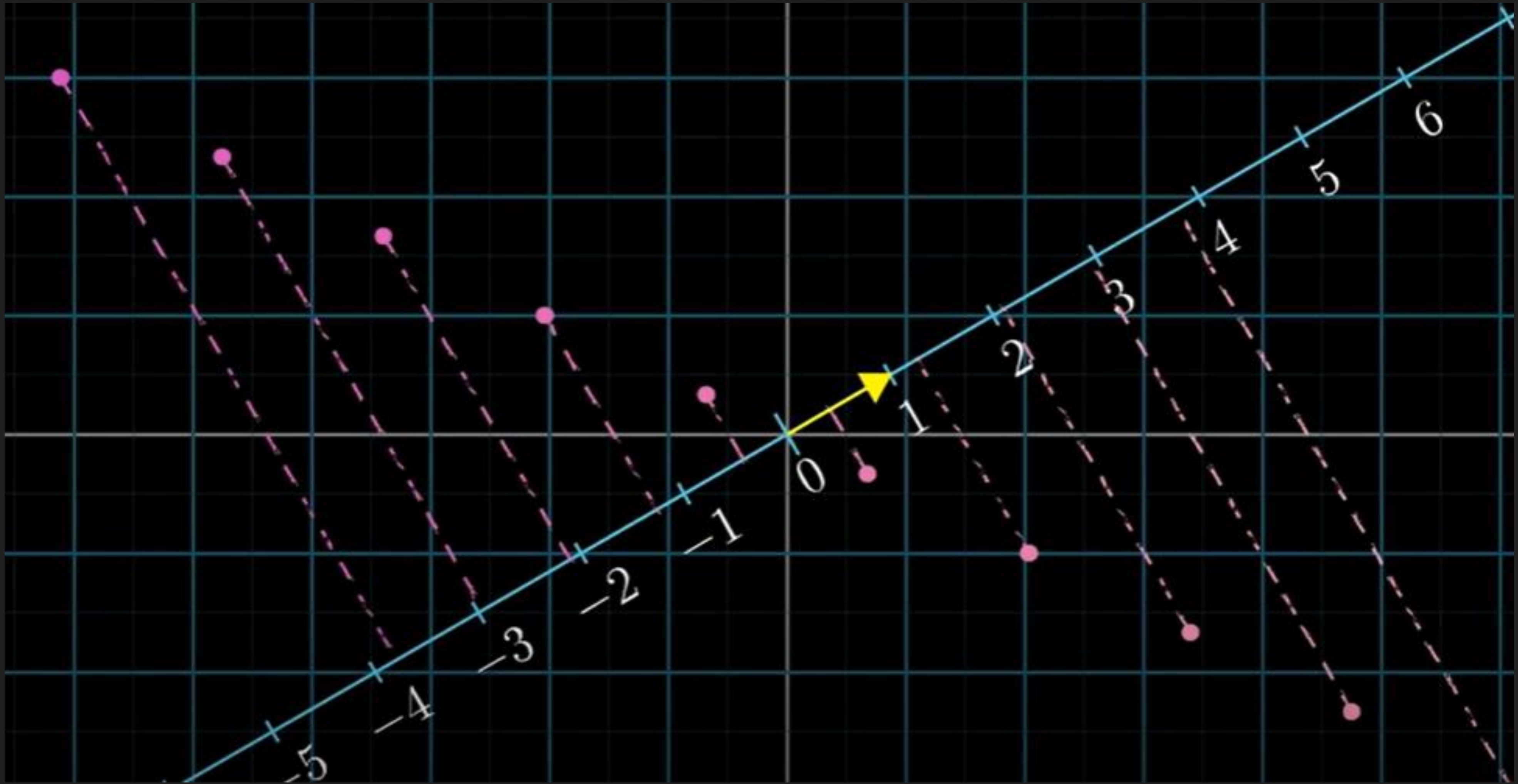
$$\mathbf{w}^\top \mathbf{x} + b = 0$$

Este hiperplano **divide el espacio en dos mitades**:

- Si $\mathbf{w}^\top \mathbf{x} + b > 0$, estás en un lado.
- Si $\mathbf{w}^\top \mathbf{x} + b < 0$, estás en el otro.

Tenemos entonces un producto escalar, es decir, la magnitud de proyectar un vector sobre otro

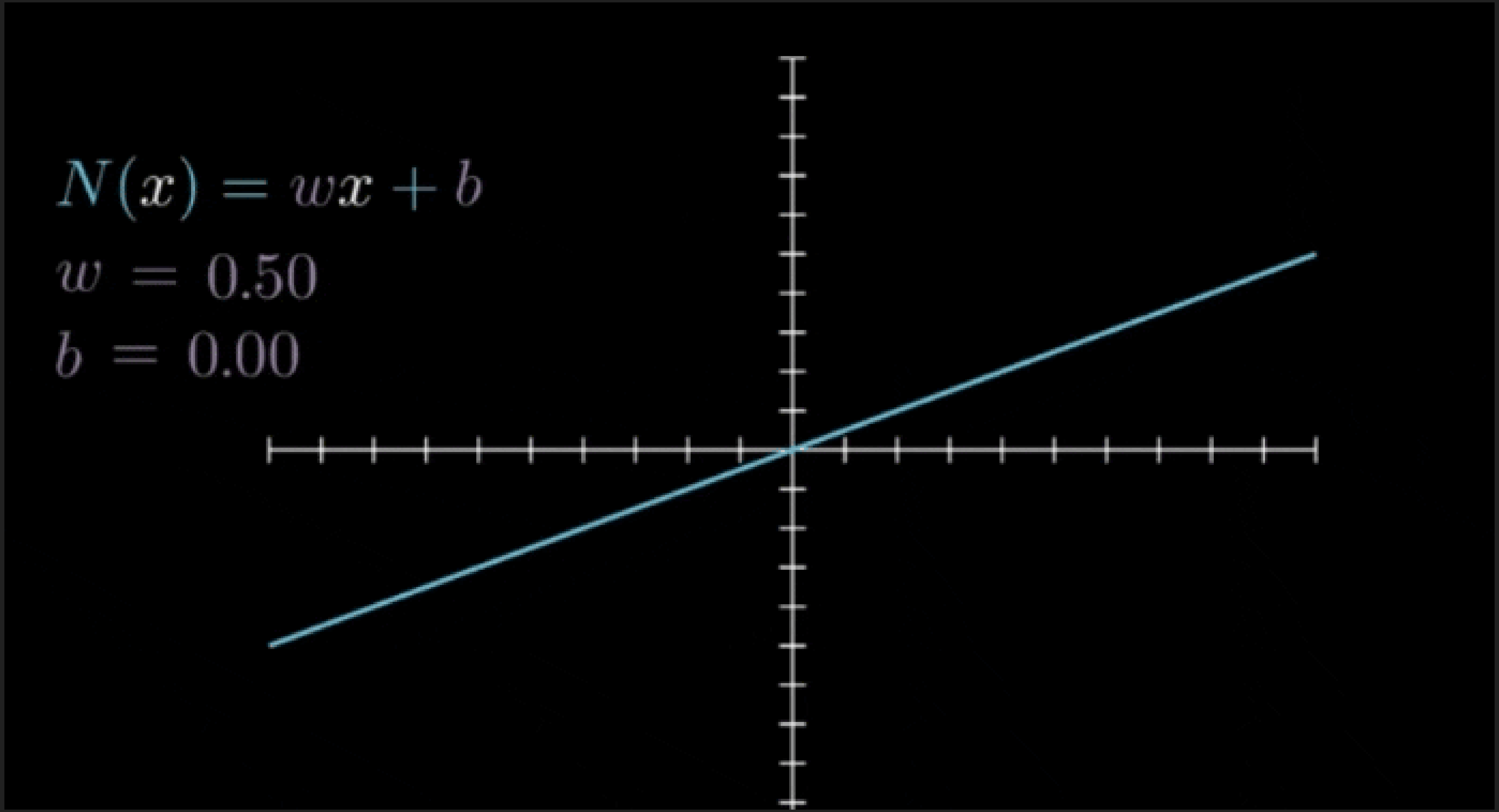

$\underbrace{\begin{bmatrix} 4 \\ 1 \end{bmatrix}}_{\vec{\mathbf{v}}} \cdot \underbrace{\begin{bmatrix} 2 \\ -1 \end{bmatrix}}_{\vec{\mathbf{w}}}$
$\vec{\mathbf{v}}$
$\vec{\mathbf{w}}$



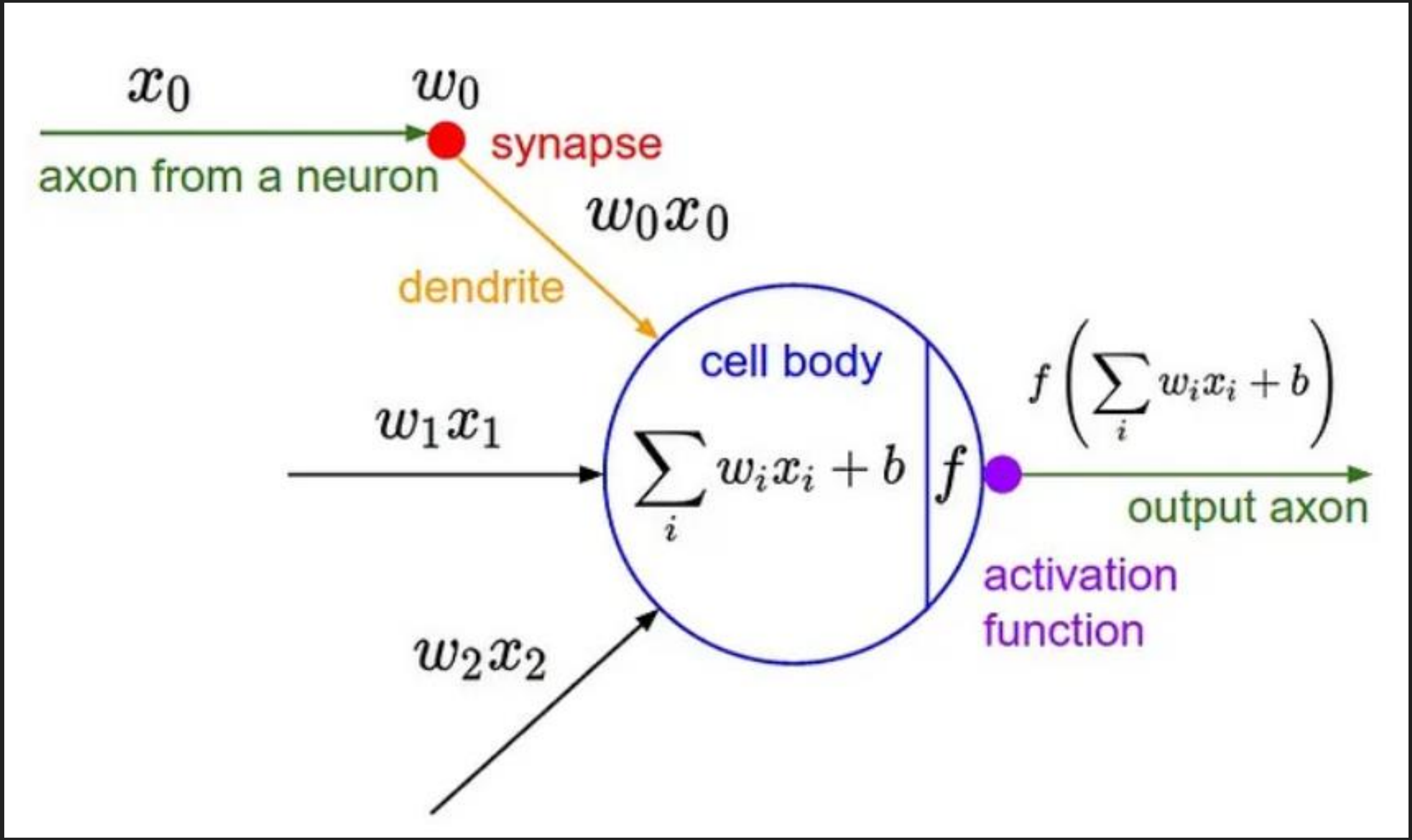
$$\mathbf{w}^\top \mathbf{x} = -b$$

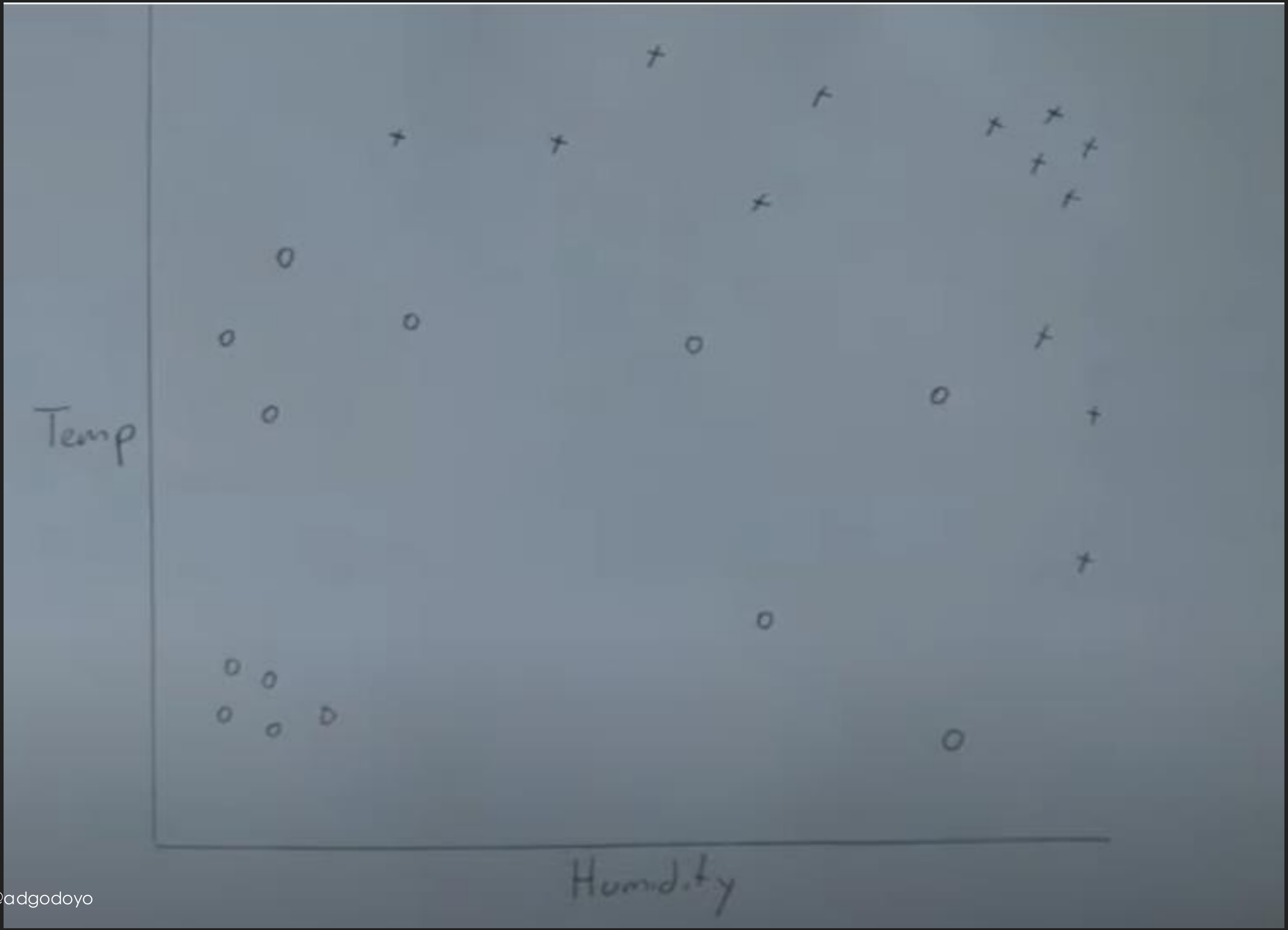
Si ves a wTx como una proyección de x en la dirección de w, entonces estás diciendo: "Todos los puntos x cuya proyección en la dirección de w dan el mismo resultado (-b) están en el hiperplano".

Andrés Daniel Godoy Ortiz - @adgodoyo

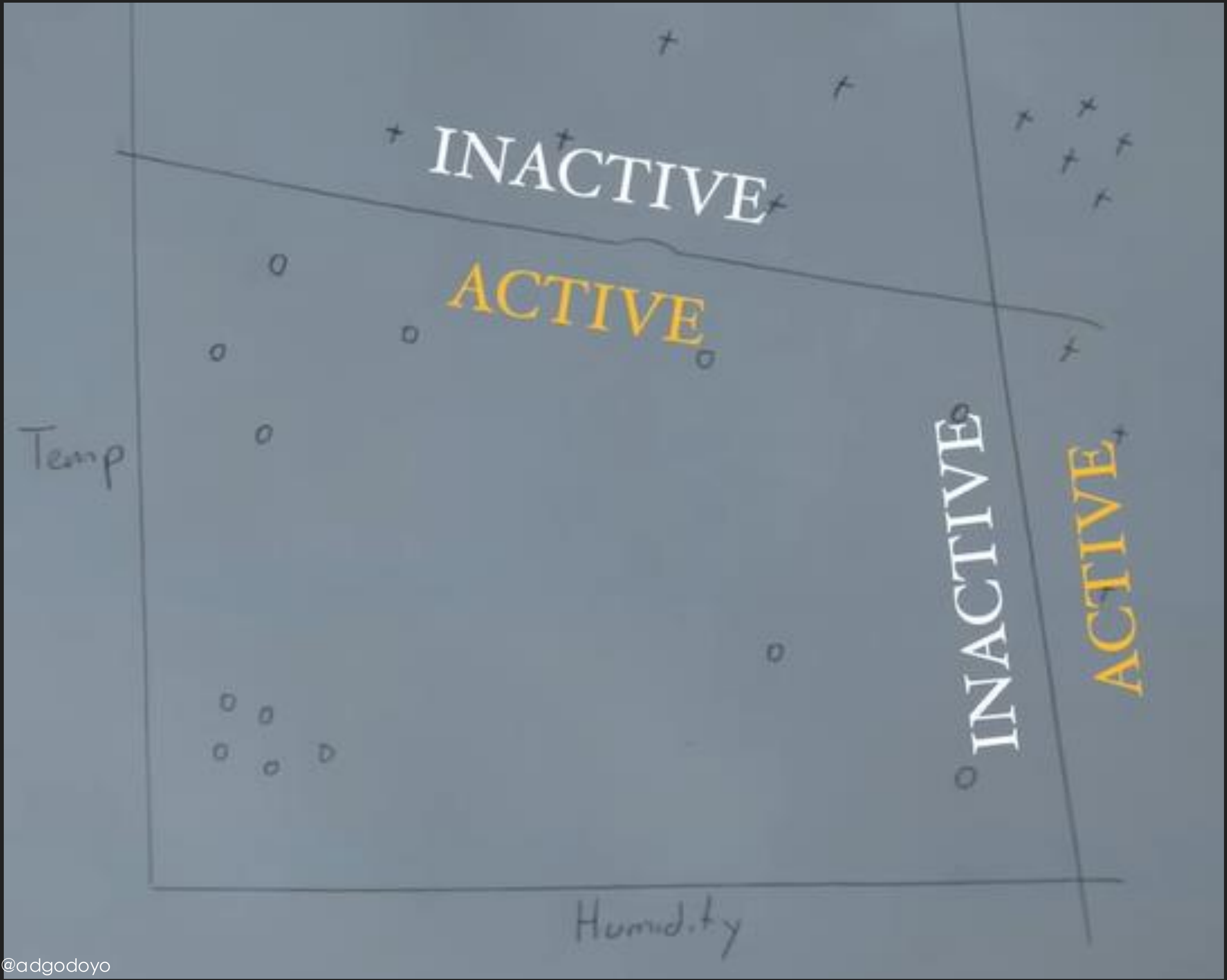

$N(x) = wx + b$
$w = 0.50$
$b = 0.00$

Cada neurona va a ser una función afín dentro una función no línea

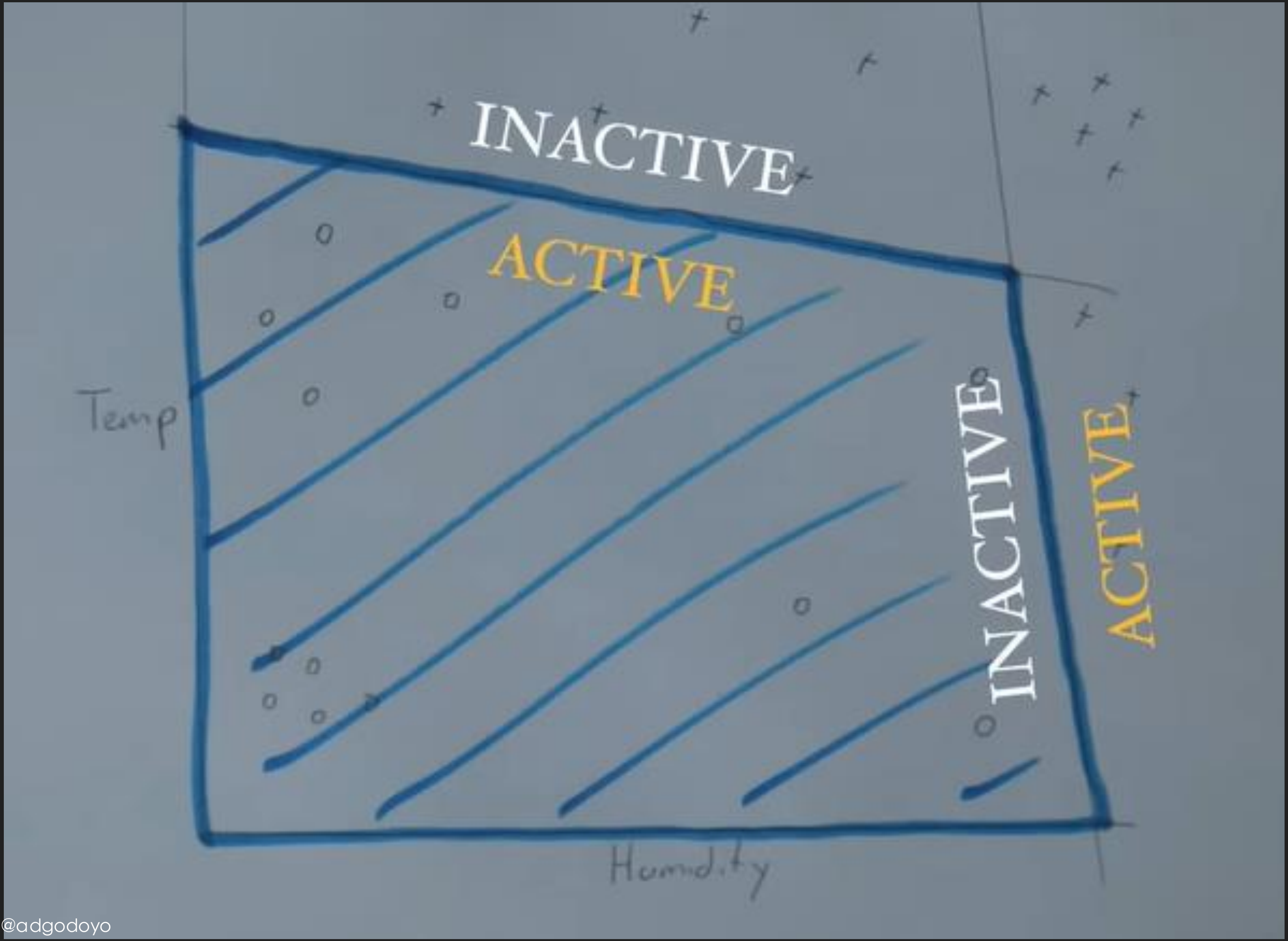



Temp
Humidity

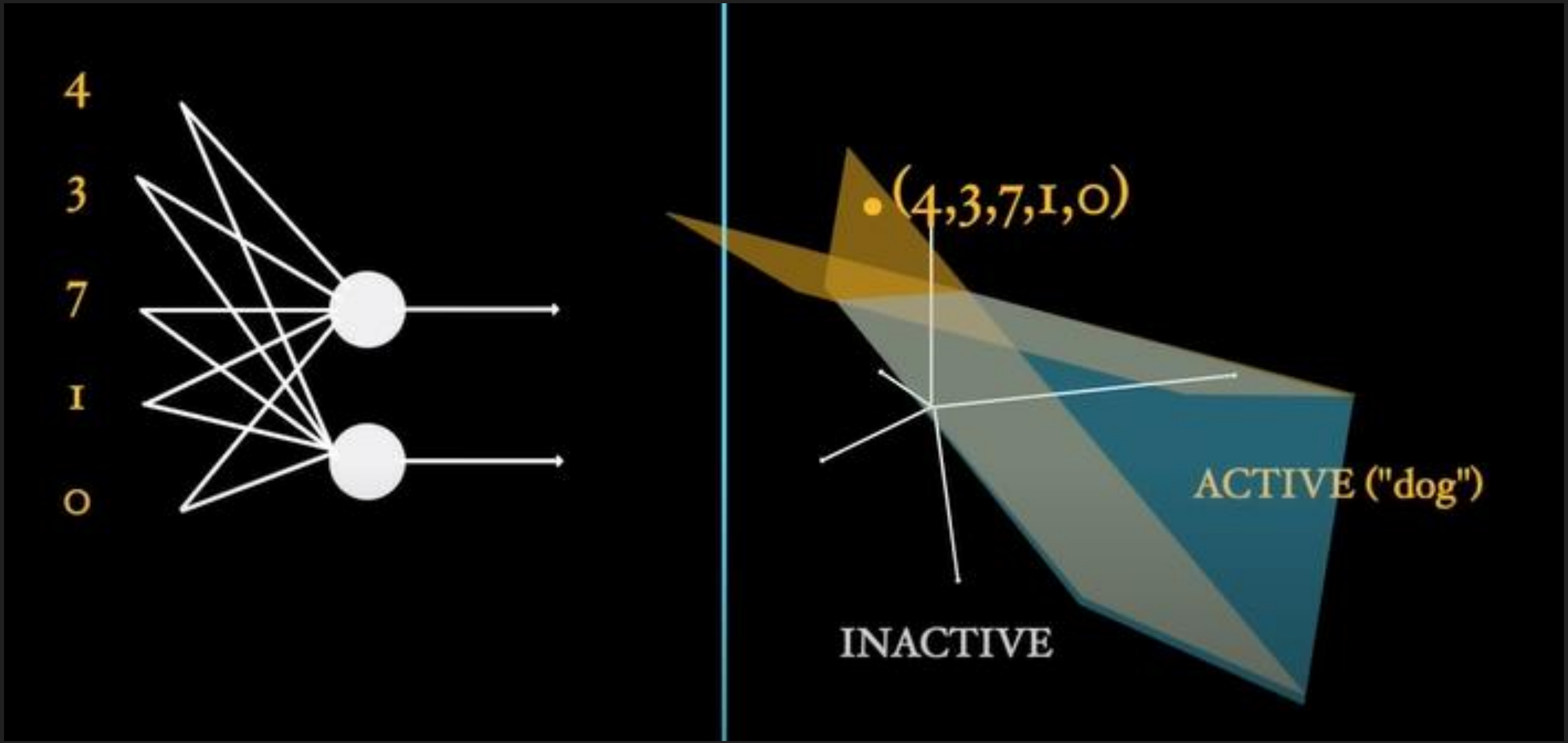
Andrés Daniel Godoy Ortiz



Andrés Daniel Godoy Ortiz



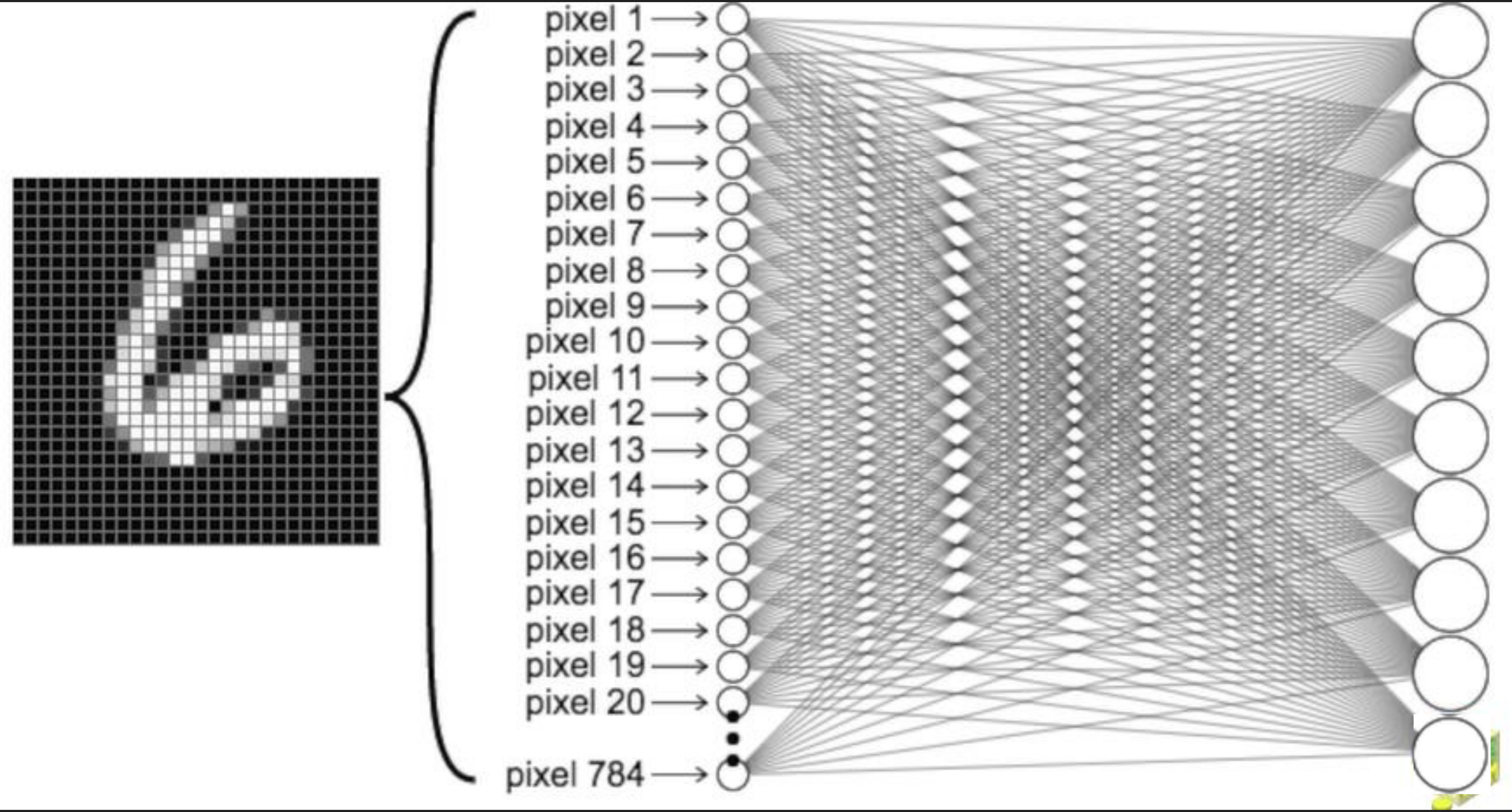
Andrés Daniel Godoy Ortiz


4
3
7
1
0
(4,3,7,1,0)
ACTIVE ("dog")
INACTIVE

Los **conceptos o predicciones** son particiones o regiones del espacio de características, divididas por esa hipersuperficie.

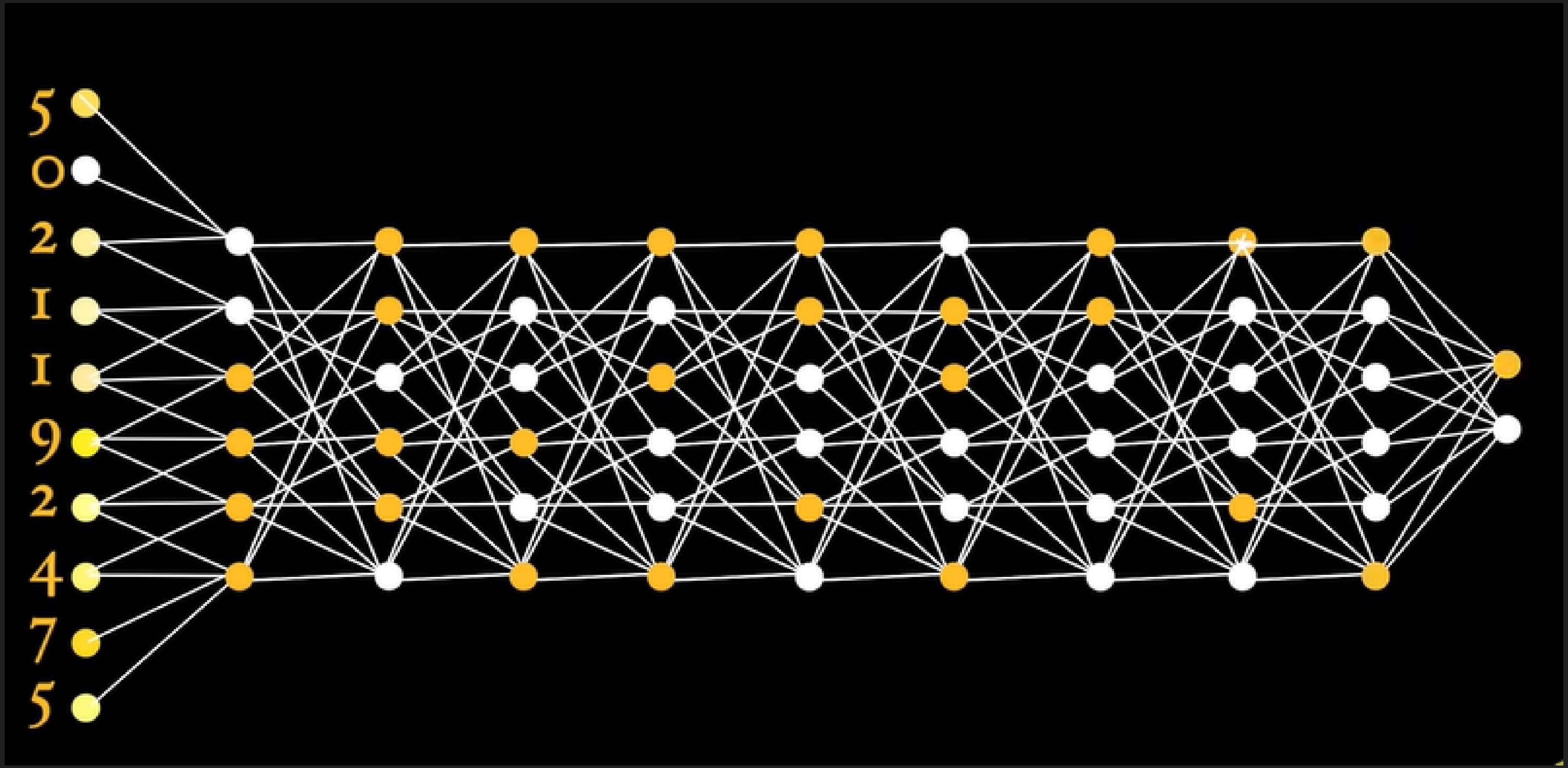
En problemas de regresión, la idea es estar siempre sobre la hipersuperficie (solo hay una neurona al final).

Andrés Daniel Godoy Ortiz

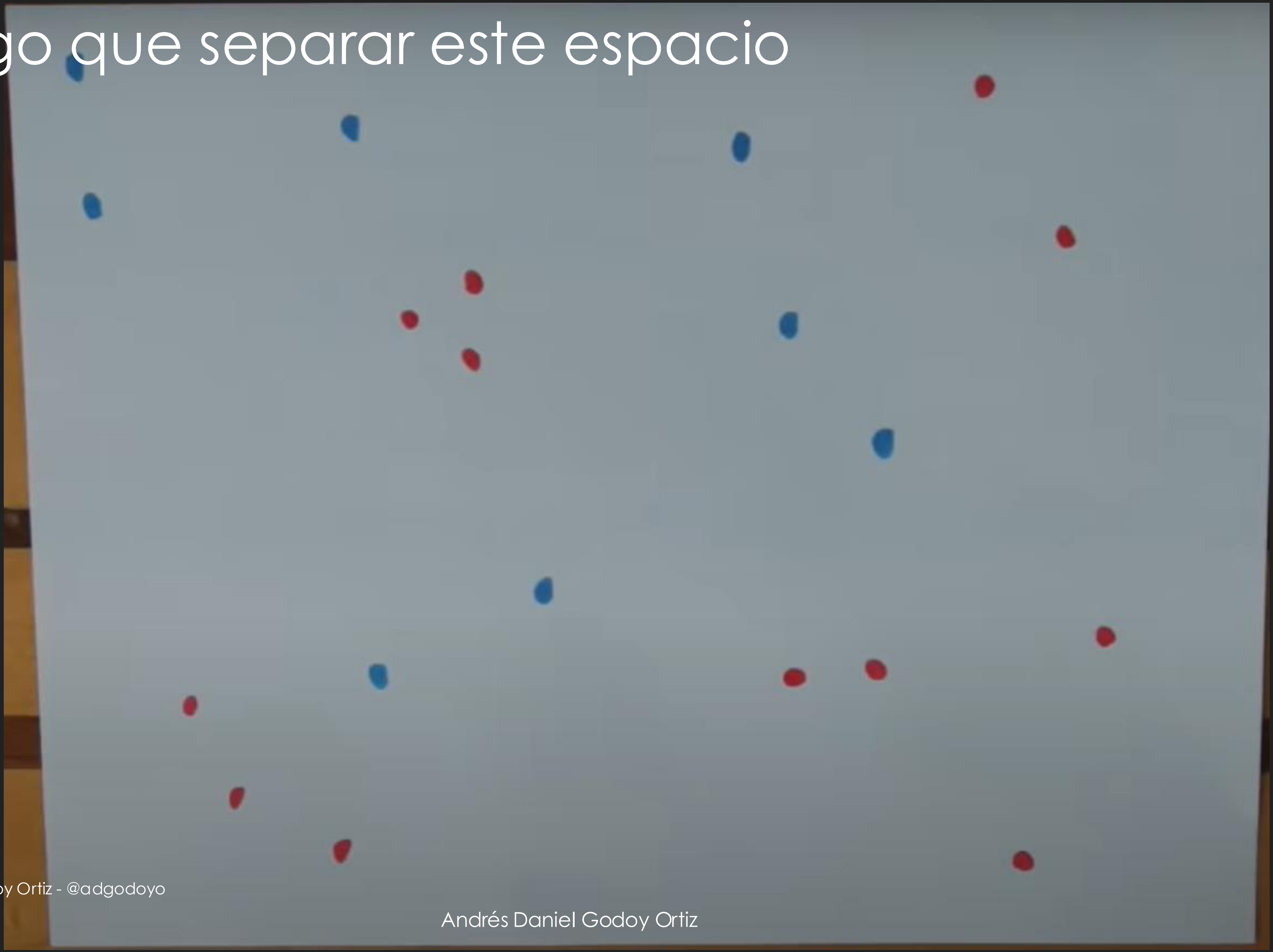

pixel 1
pixel 2
pixel 3
pixel 4
pixel 5
pixel 6
pixel 7
pixel 8
pixel 9
pixel 10
pixel 11
pixel 12
pixel 13
pixel 14
pixel 15
pixel 16
pixel 17
pixel 18
pixel 19
pixel 20
pixel 784

Dividir el espacio con una o varias líneas sería poco exitoso en el ejemplo anterior.

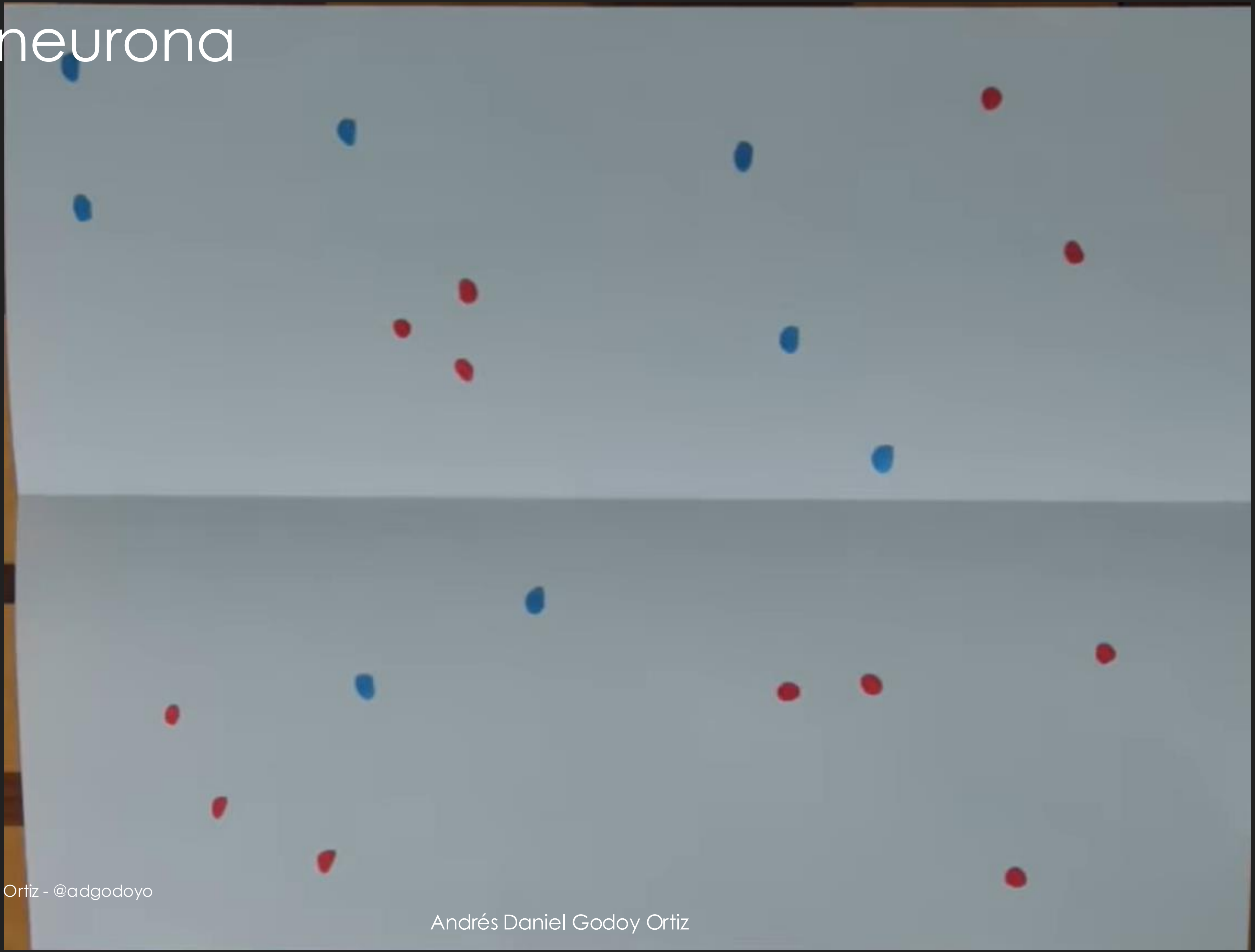
Así que vamos a entrar en la parte MENOS COMPRENDIDA de las redes neuronales: Capas múltiples


5
0
2
1
1
9
2
4
7
5

Tengo que separar este espacio

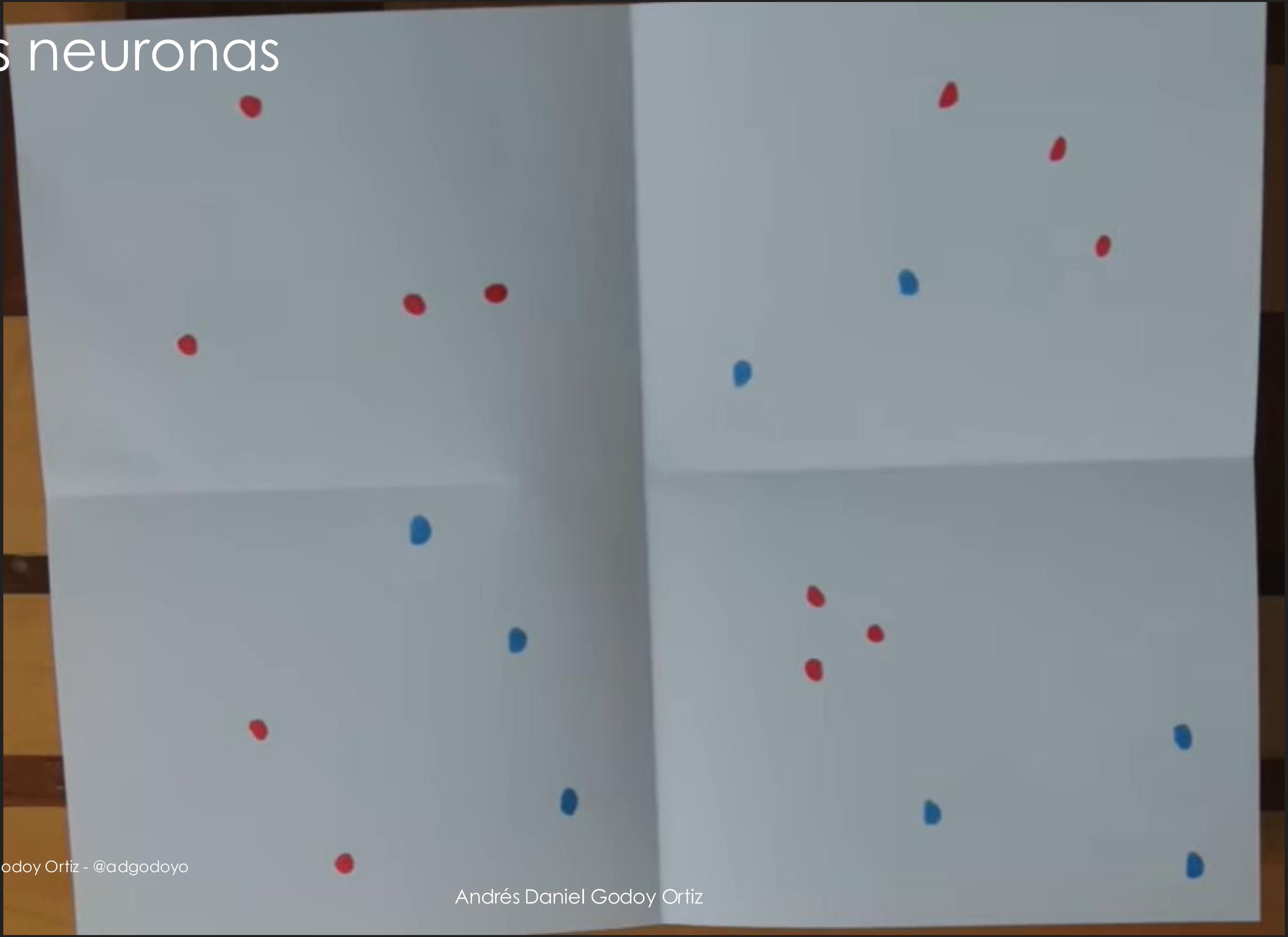


Una neurona

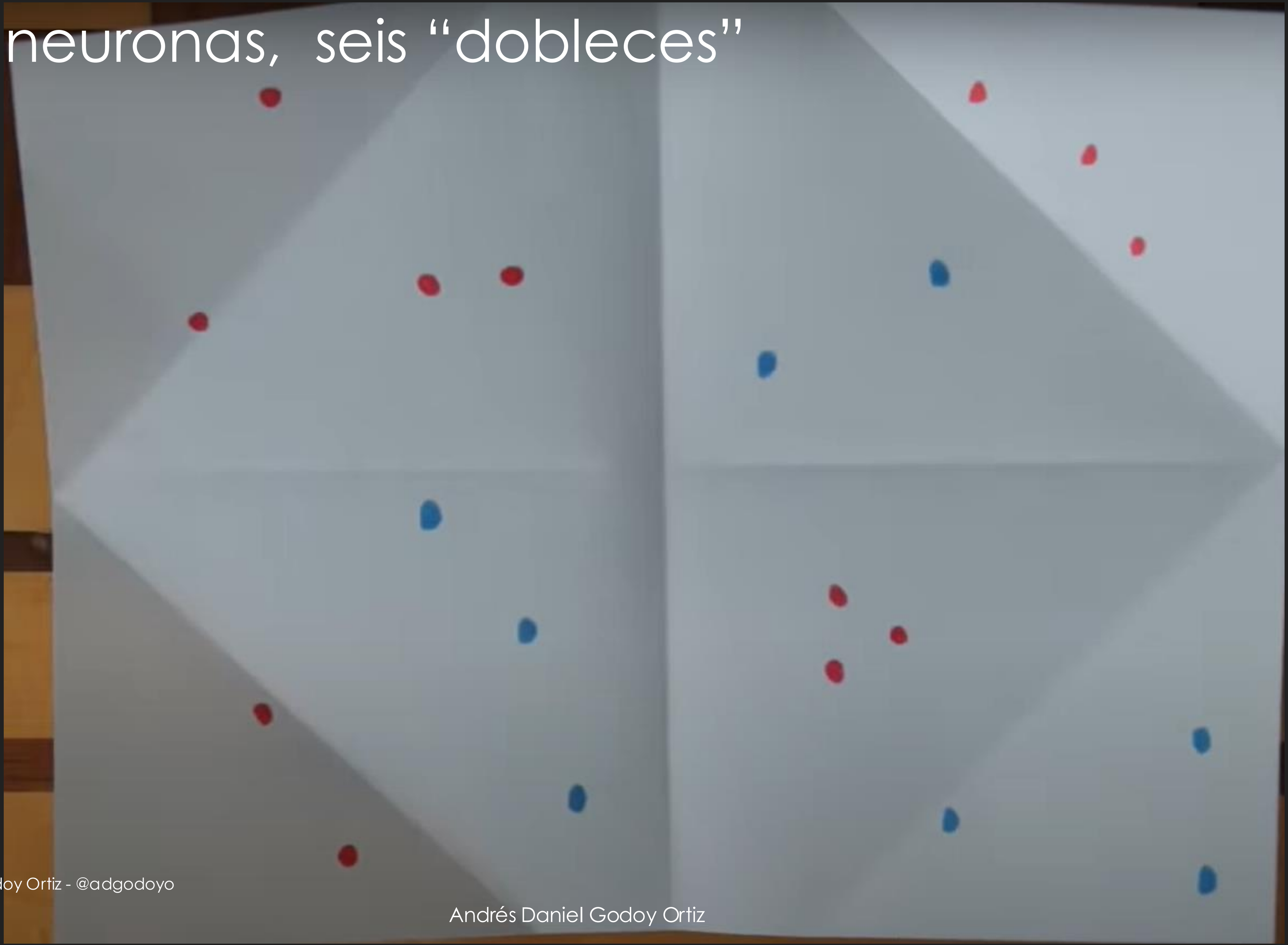


Andrés Daniel Godoy Ortiz

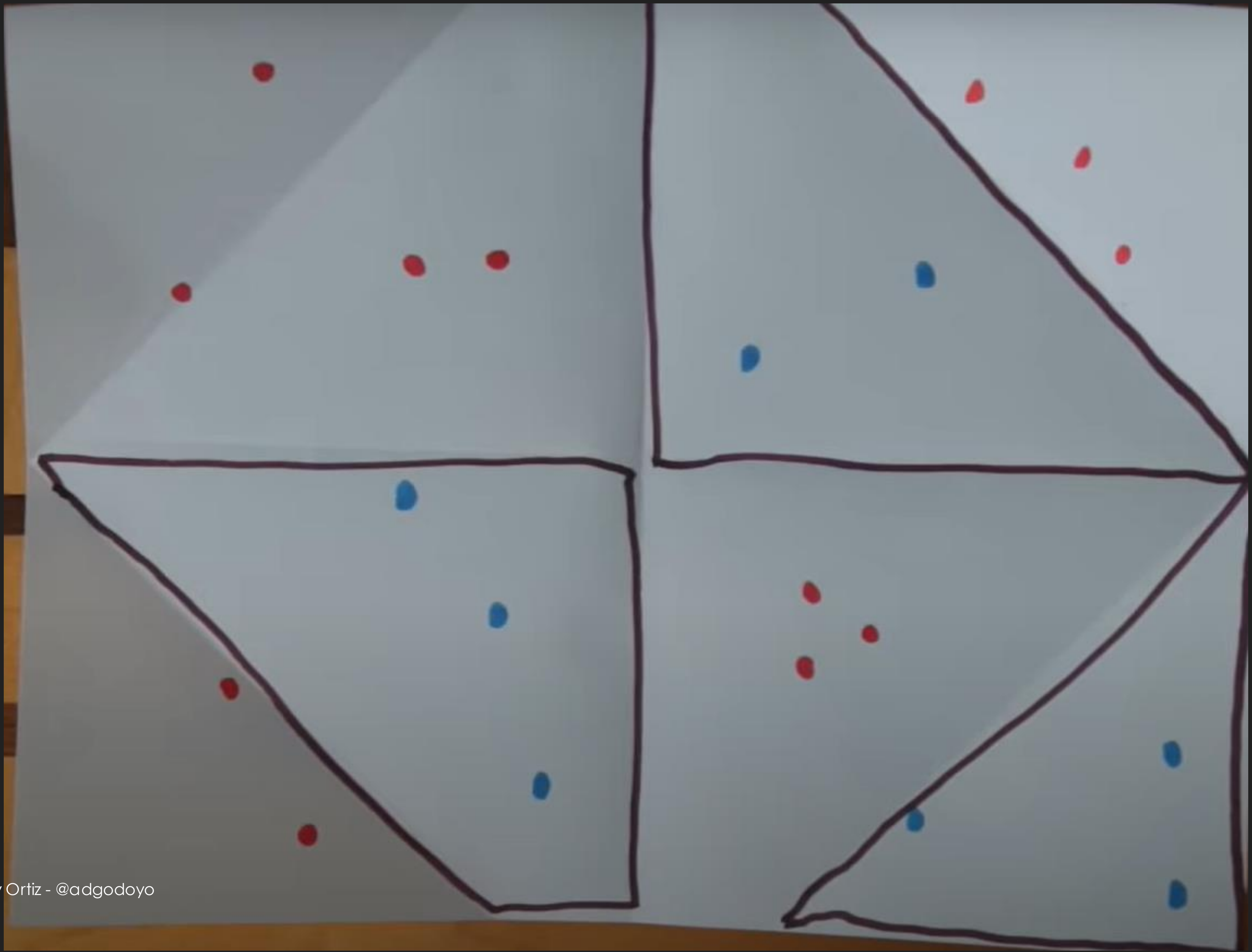
Dos neuronas

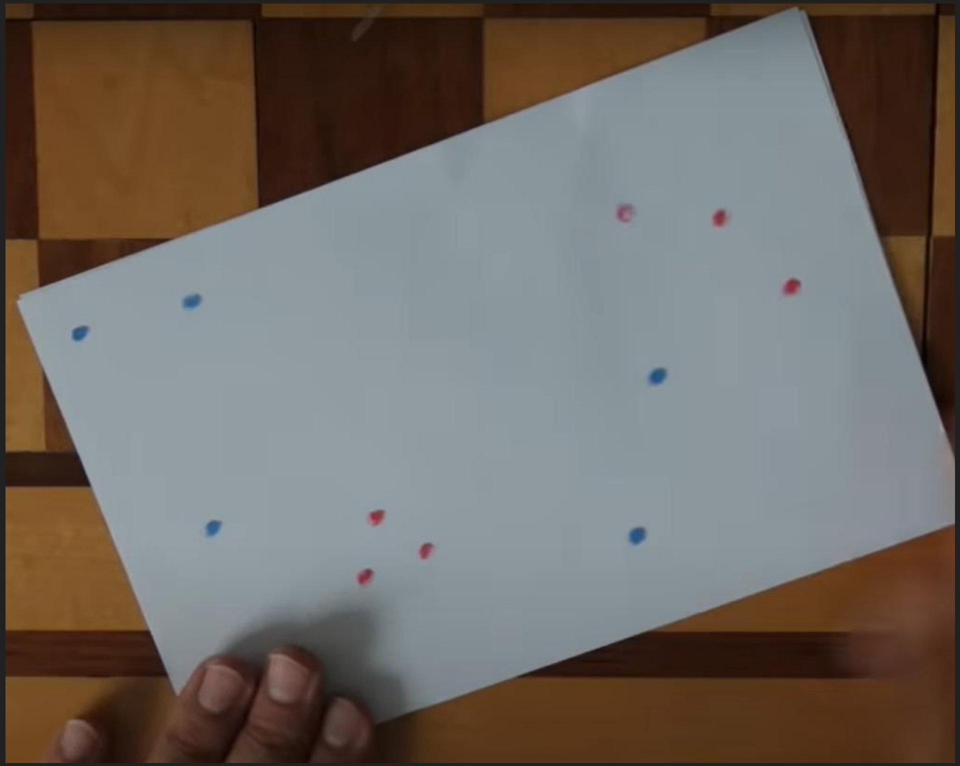


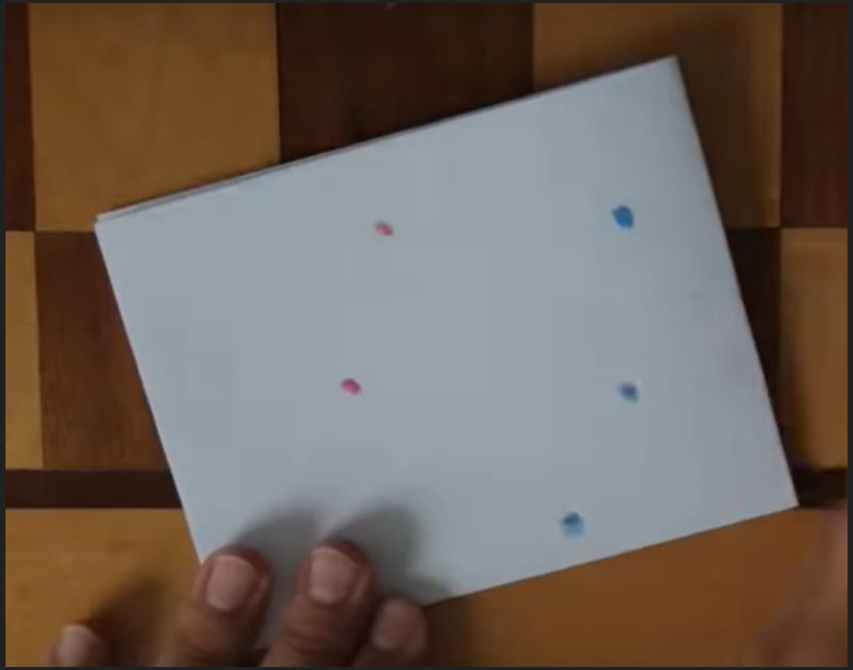
Seis neuronas, seis "dobleces"



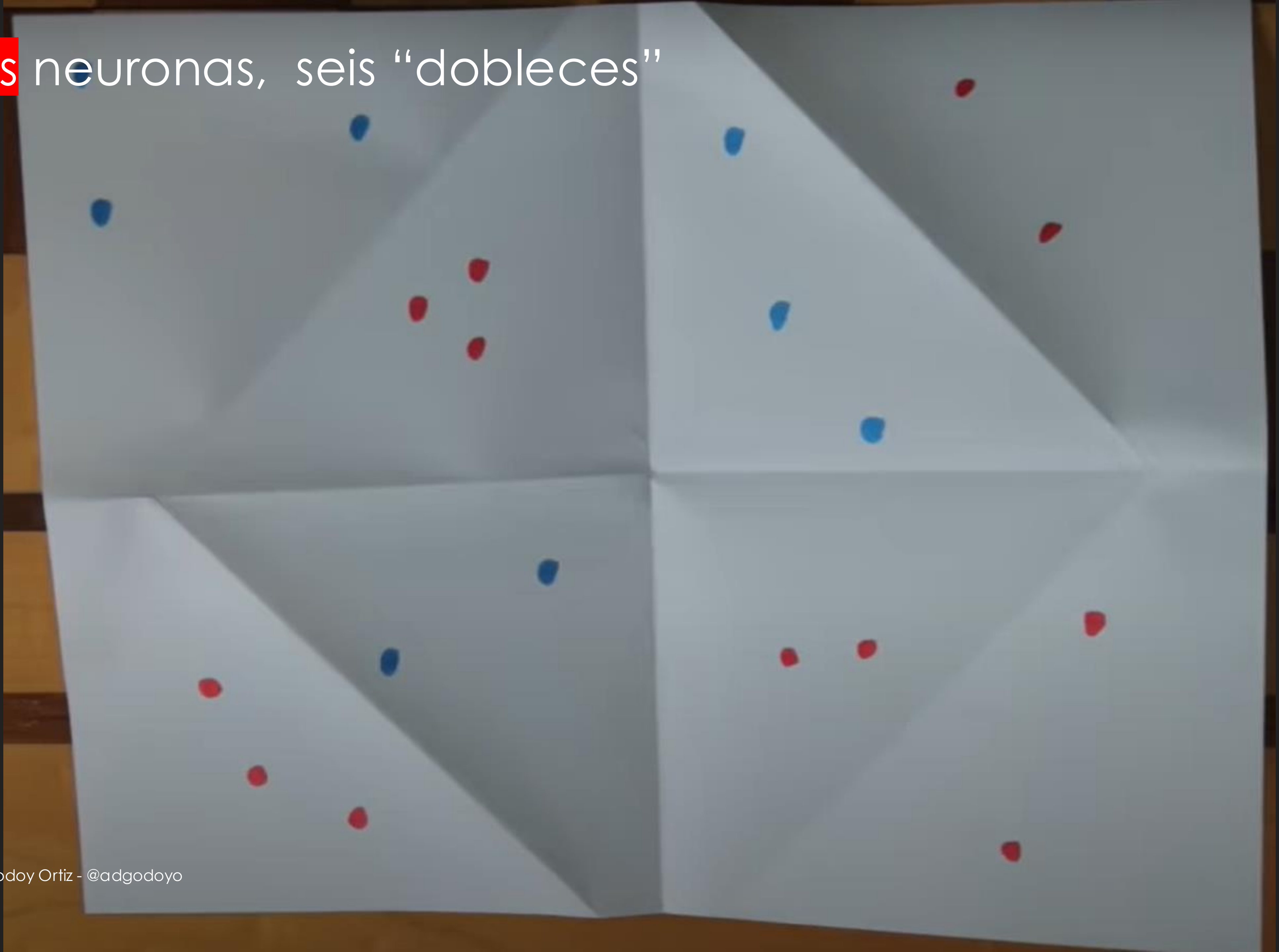
Andrés Daniel Godoy Ortiz





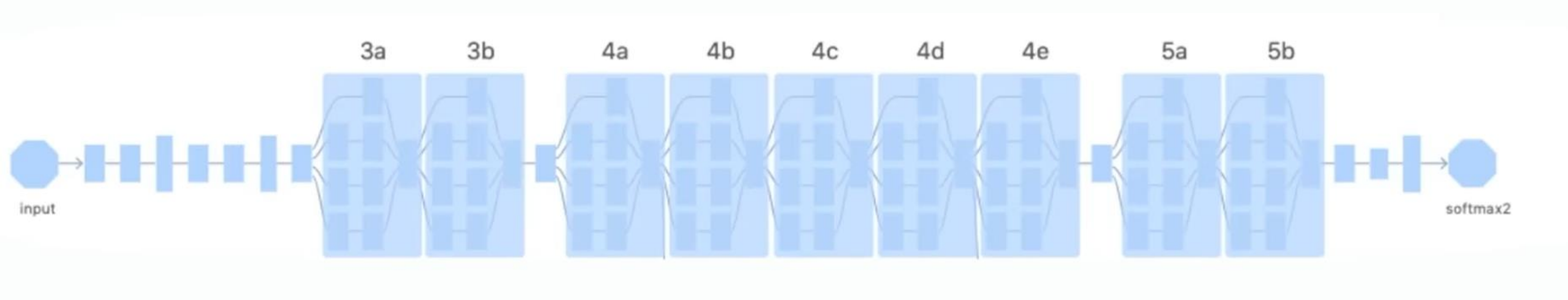


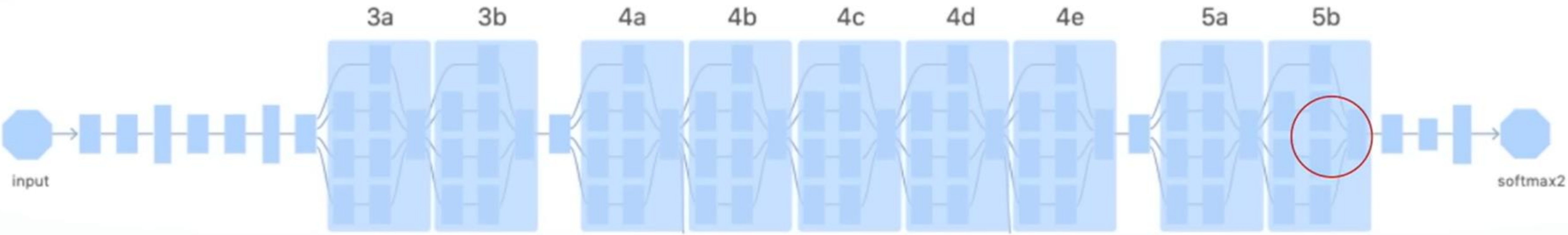
Tres neuronas, seis "dobleces"

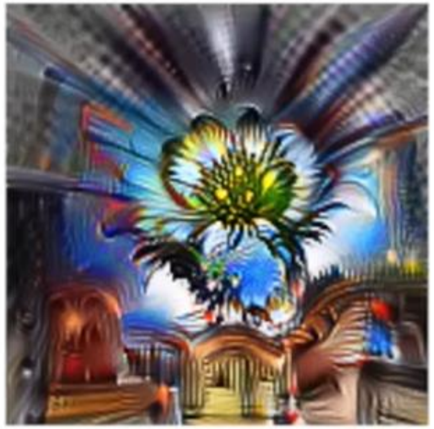


Cuatro “dobleces”, 16 regiones
Cinco “dobleces”, 32 regiones
Seis “dobleces”, 64 regiones y así…crece exponencialmente


Andrés Daniel Godoy Ortiz - @adgodoyo


3a
3b
4a
4b
4c
4d
4e
5a
5b
input
softmax2

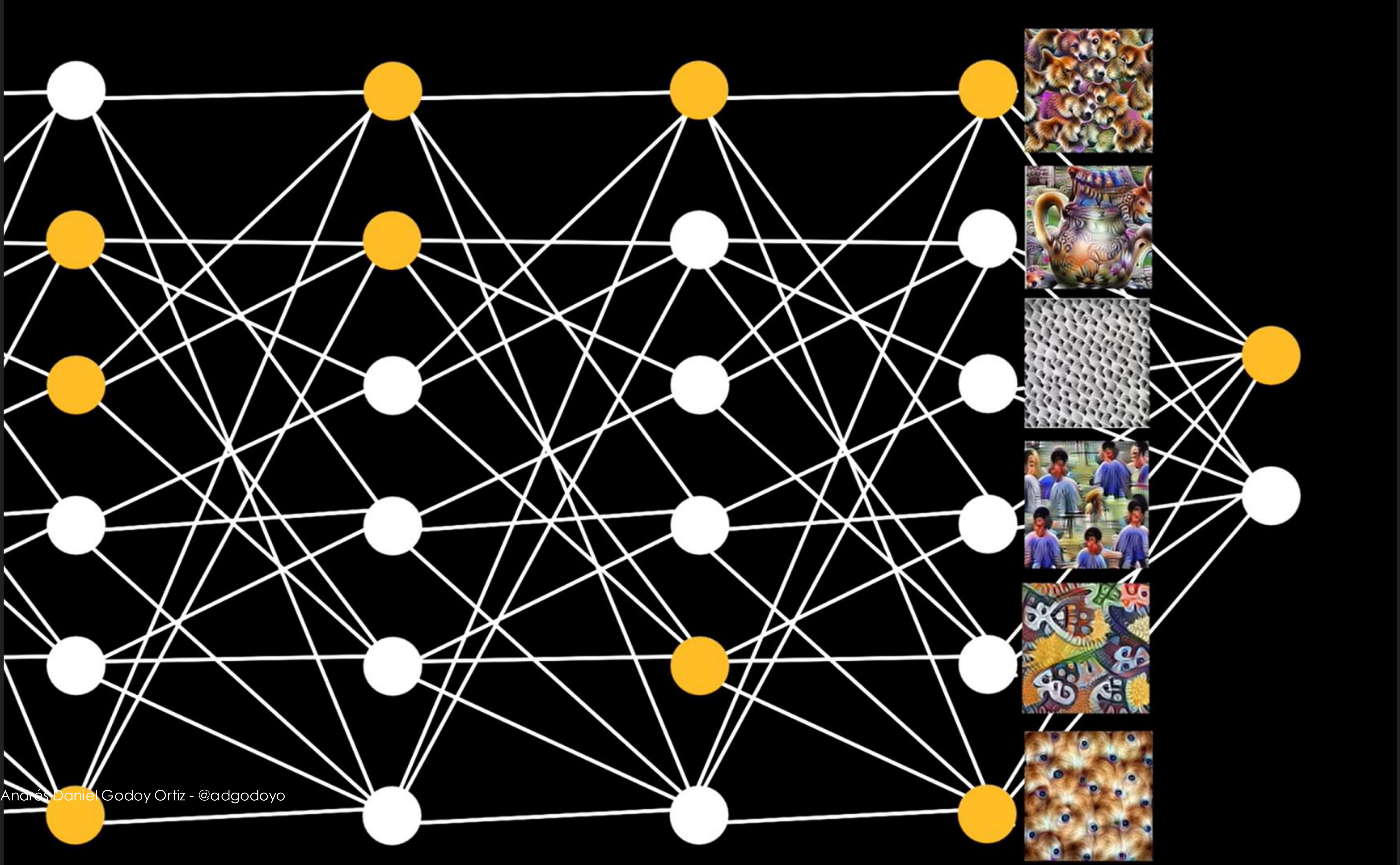



Palm trees

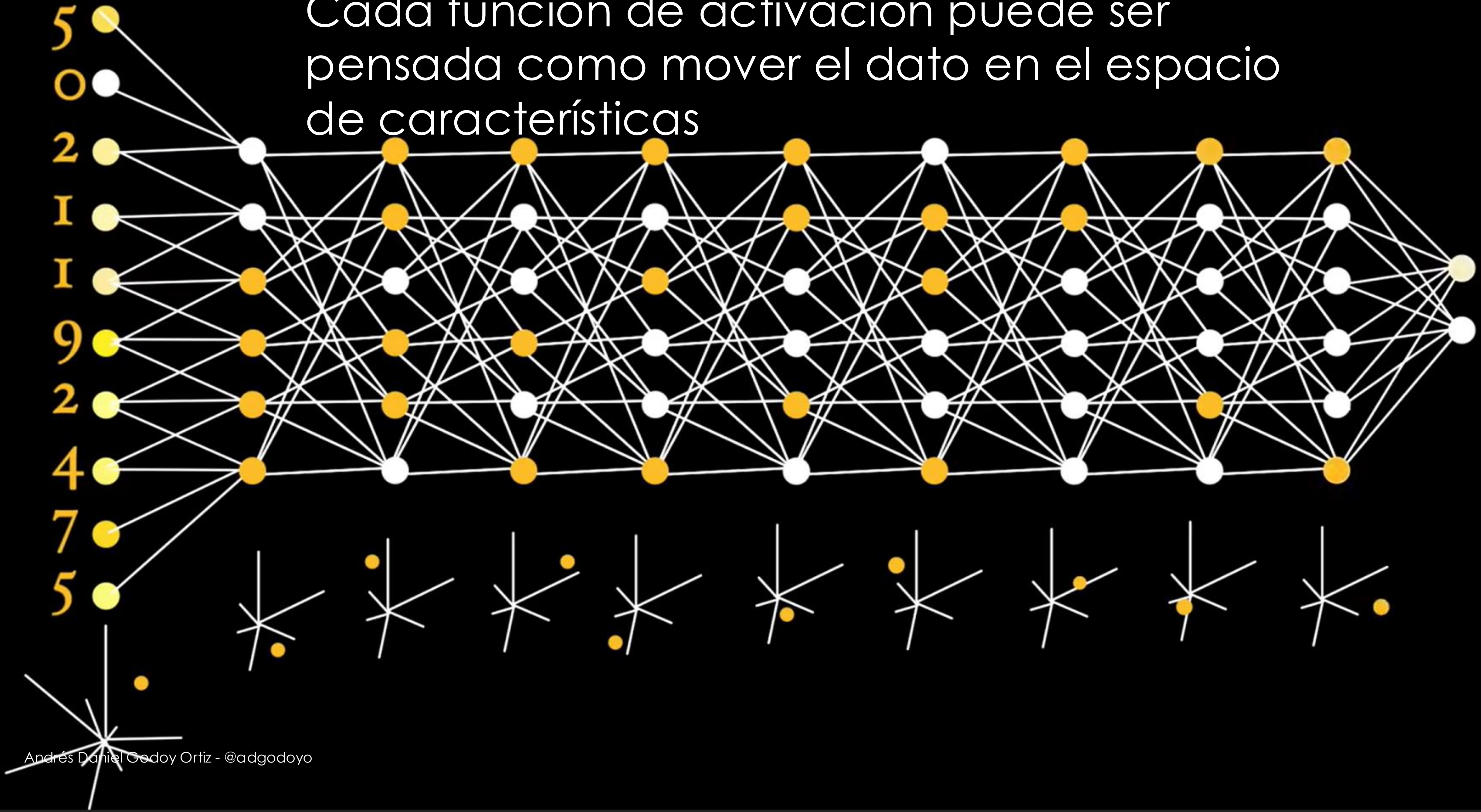

Wheels


Dogs on leash


Houses

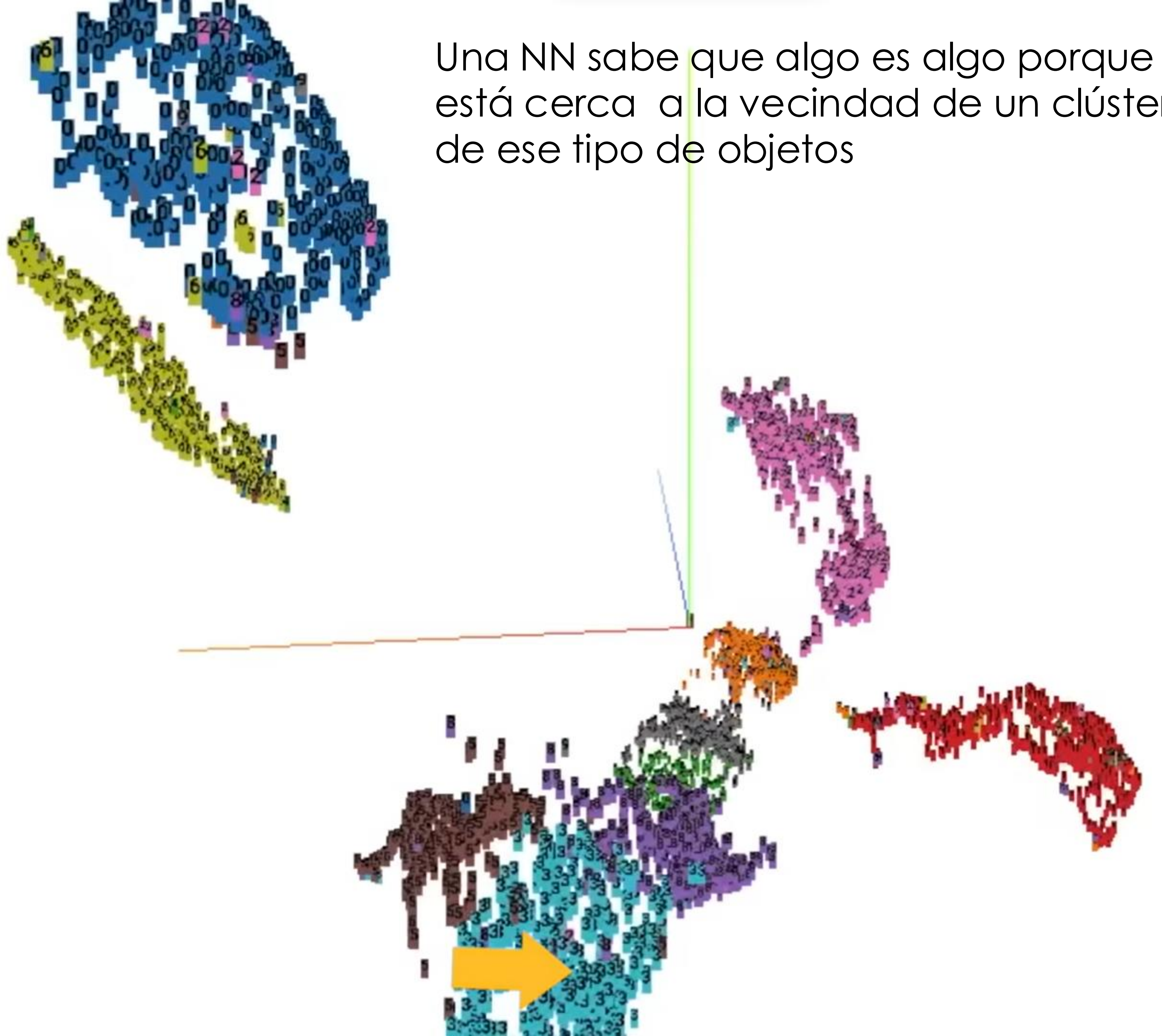


Cada función de activación puede ser pensada como mover el dato en el espacio de características

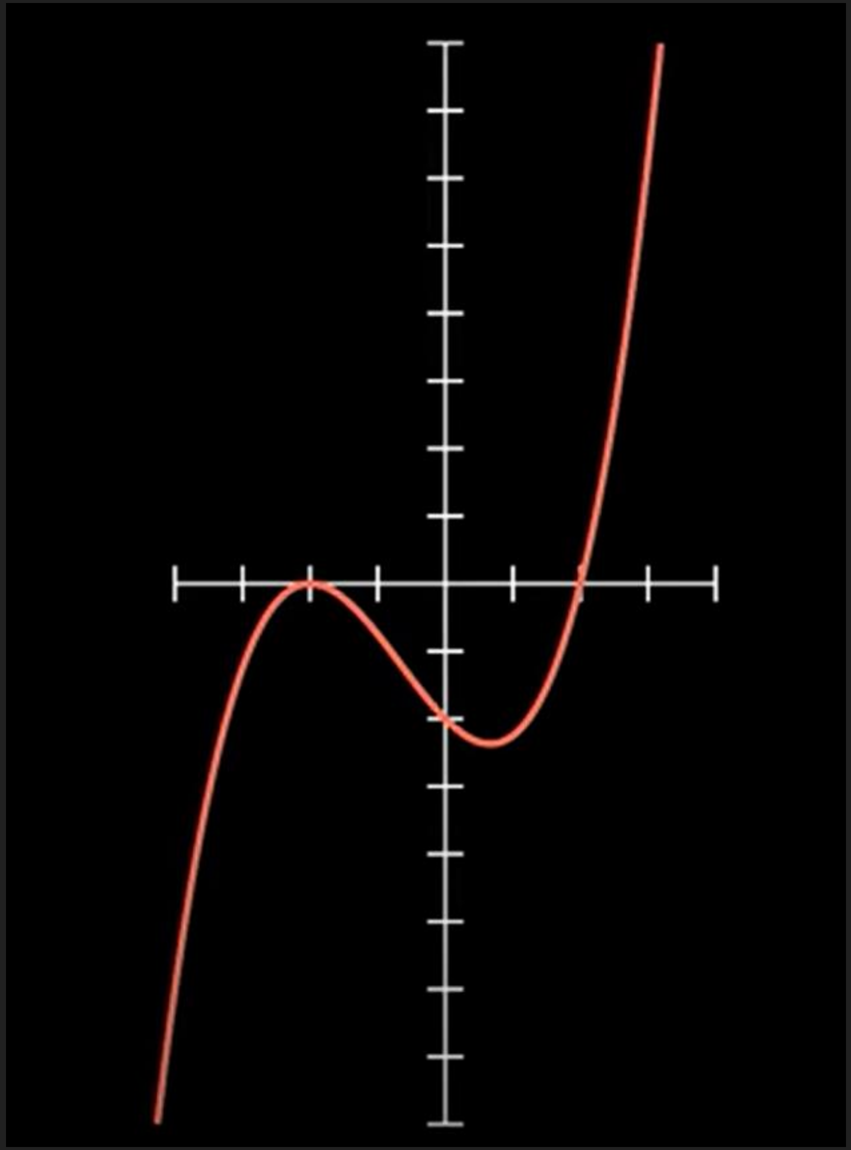


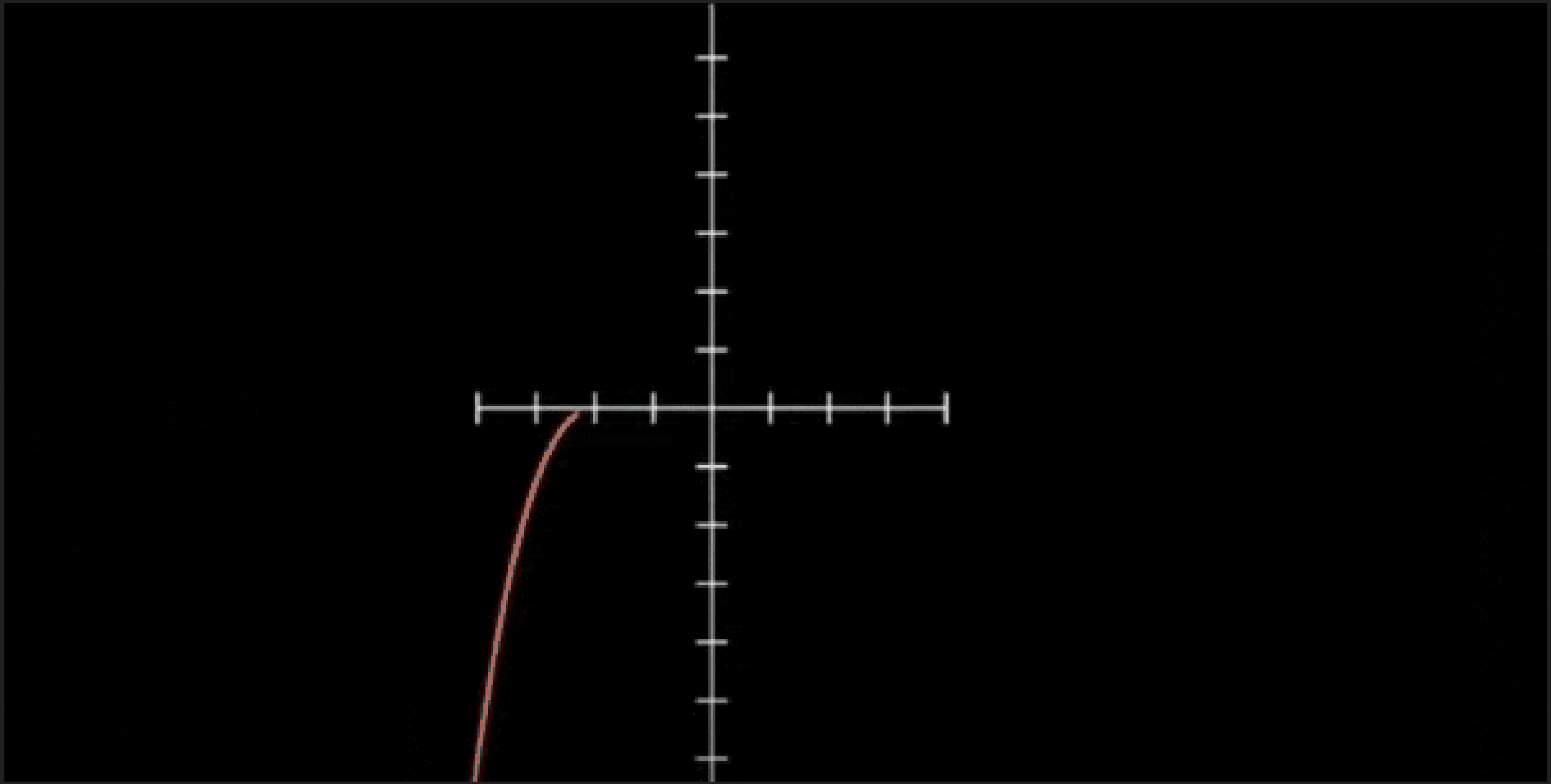
https://distill.pub/2020/grand-tour/

Una NN sabe que algo es algo porque está cerca a la vecindad de un clúster de ese tipo de objetos



Problema de regresión:





Supón una red con varias capas, pero **sin función de activación**, es decir, solo transformaciones lineales:

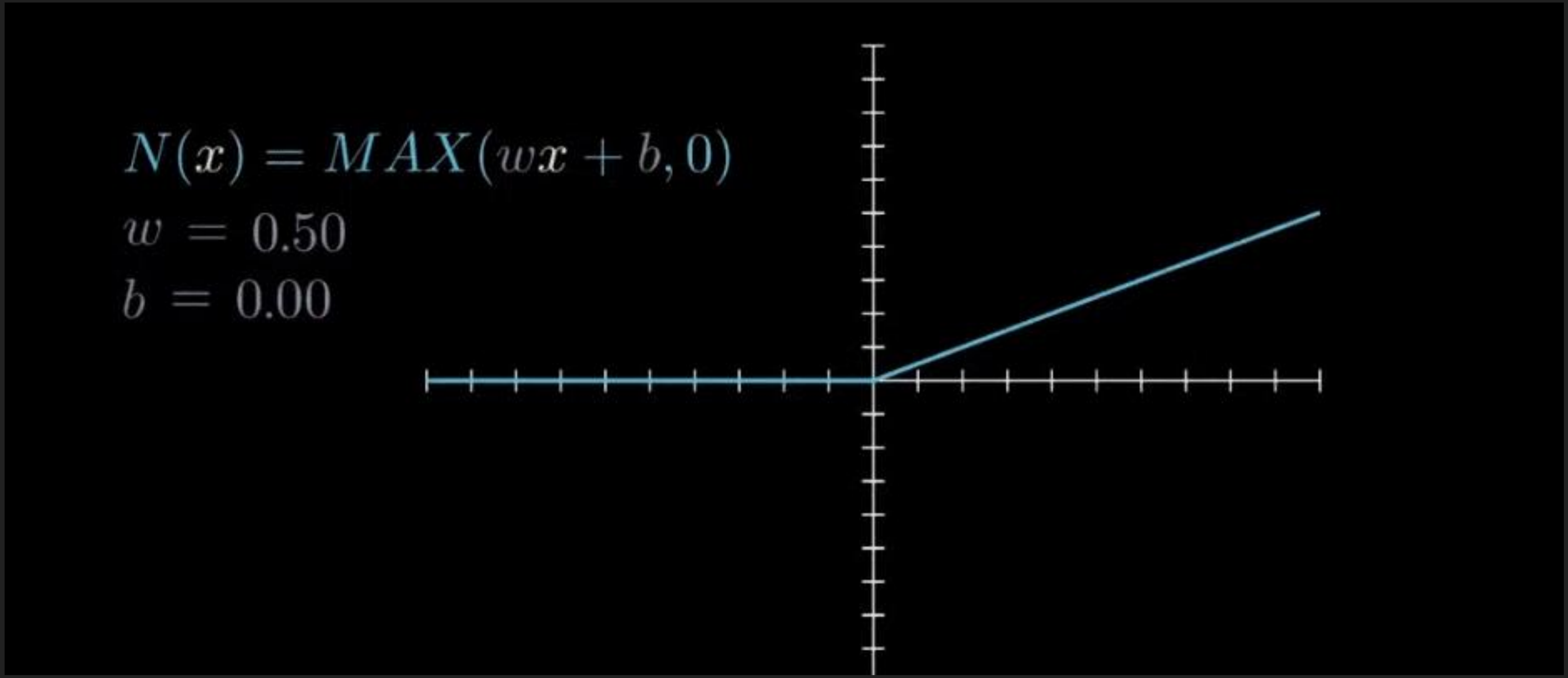
$$f(\mathbf{x}) = \mathbf{W}^{(3)}\left(\mathbf{W}^{(2)}\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}\right) + \mathbf{b}^{(3)}$$

Pero… ¡la composición de funciones lineales es otra función lineal! Entonces:

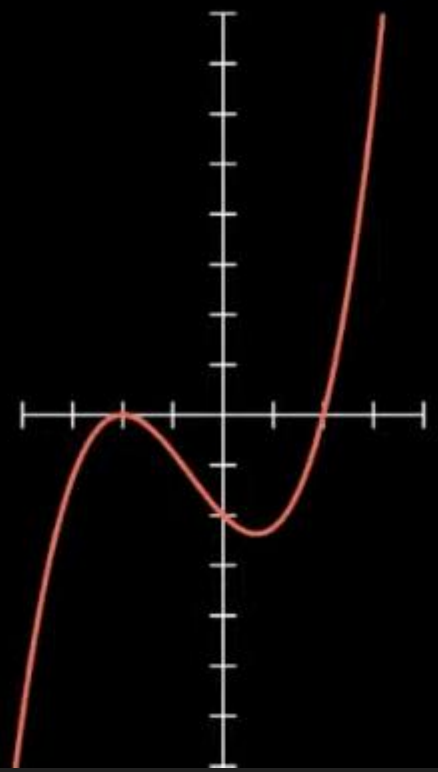
$$f(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$

La función de activación permite transformaciones no lineales

$$ReLU(x) = MAX(x, 0)$$



$$
\begin{aligned}
&-max(-5x + -7.7, 0)+\\
&-max(-1.2x + -1.3, 0)+\\
&-max(1.2x + 1, 0)+\\
&max(1.2x + -0.2, 0)+\\
&max(2x + -1.1, 0)+\\
&max(5x + -5, 0)
\end{aligned}
$$



Dot Product

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

Dot Product

$$\begin{bmatrix} 0.3 \\ 1.8 \\ 0.5 \\ 1.0 \end{bmatrix} \cdot \begin{bmatrix} 0.2 \\ -0.4 \\ 0.6 \\ 0.1 \end{bmatrix} = -0.26$$

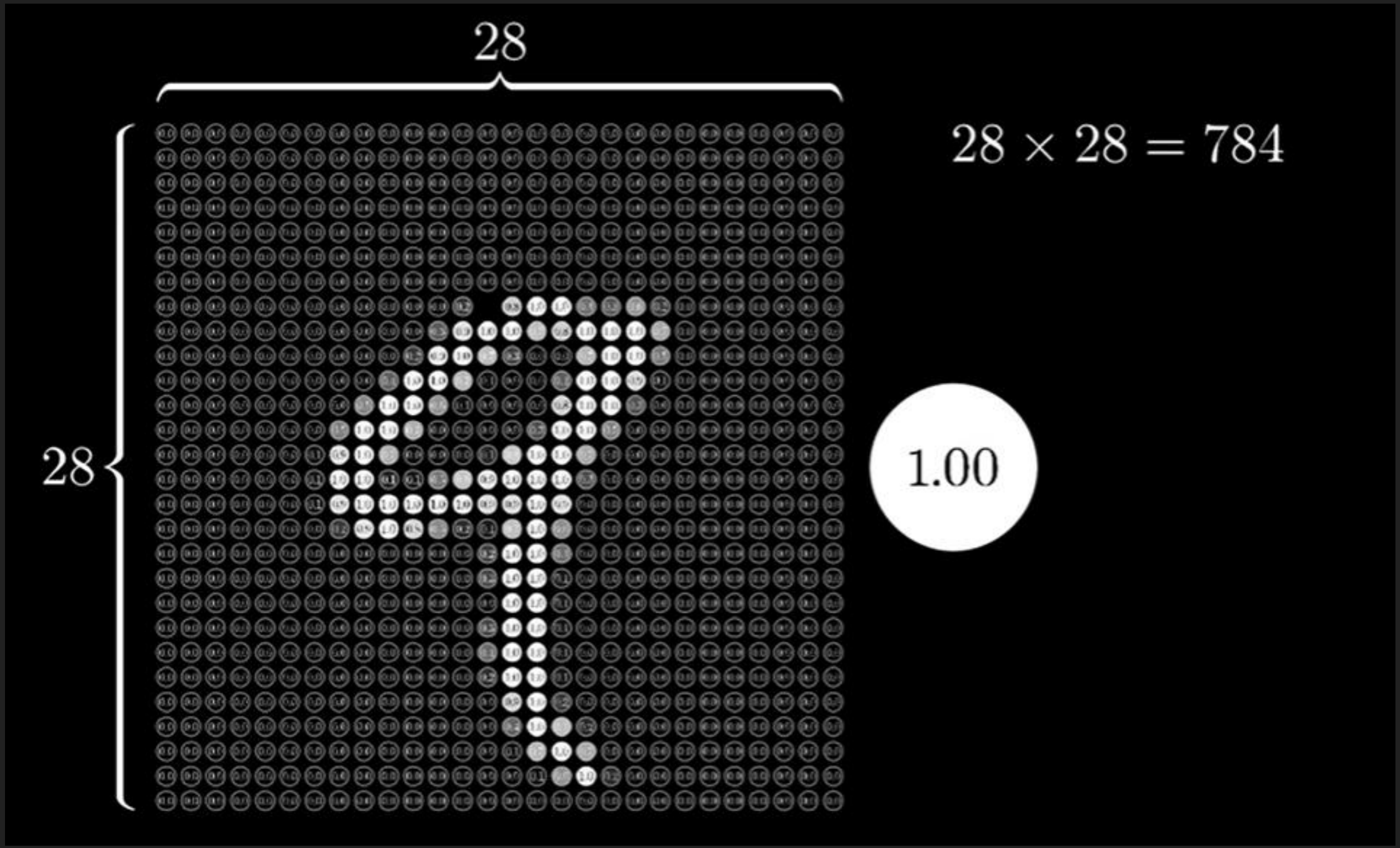
$$\mathrm{ReLU}(-0.26)$$

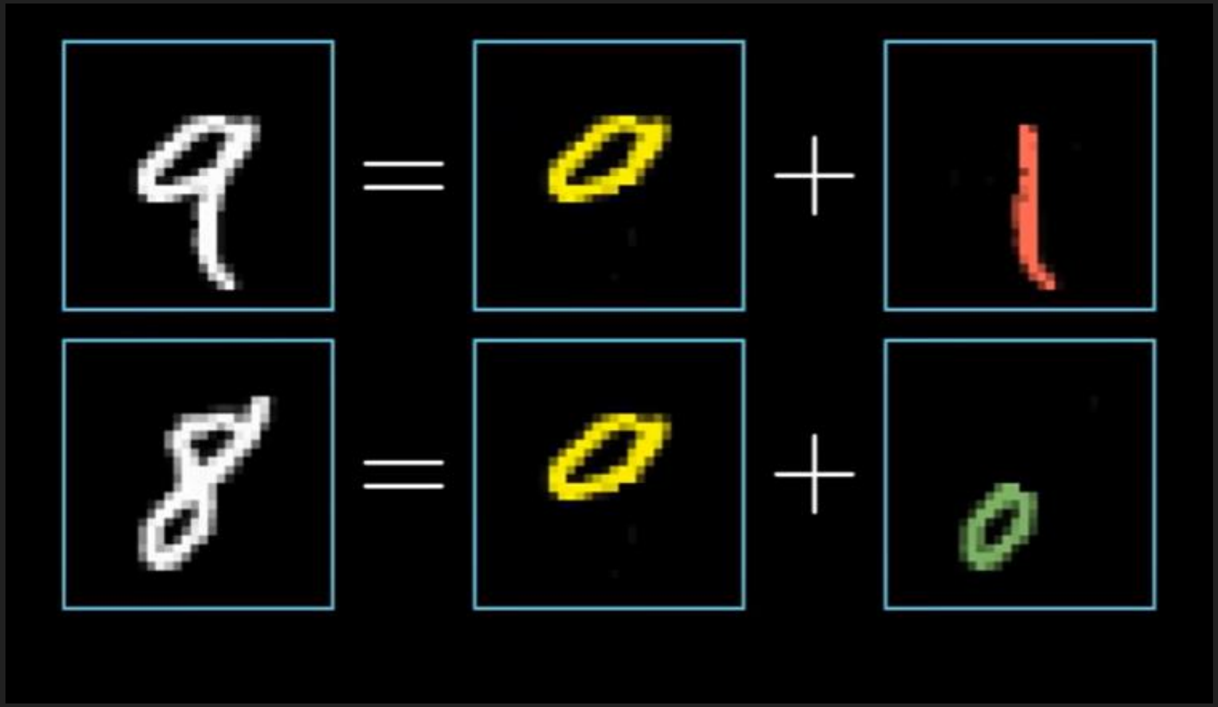
Vamos a pensar entonces que una neurona es un número y ya...

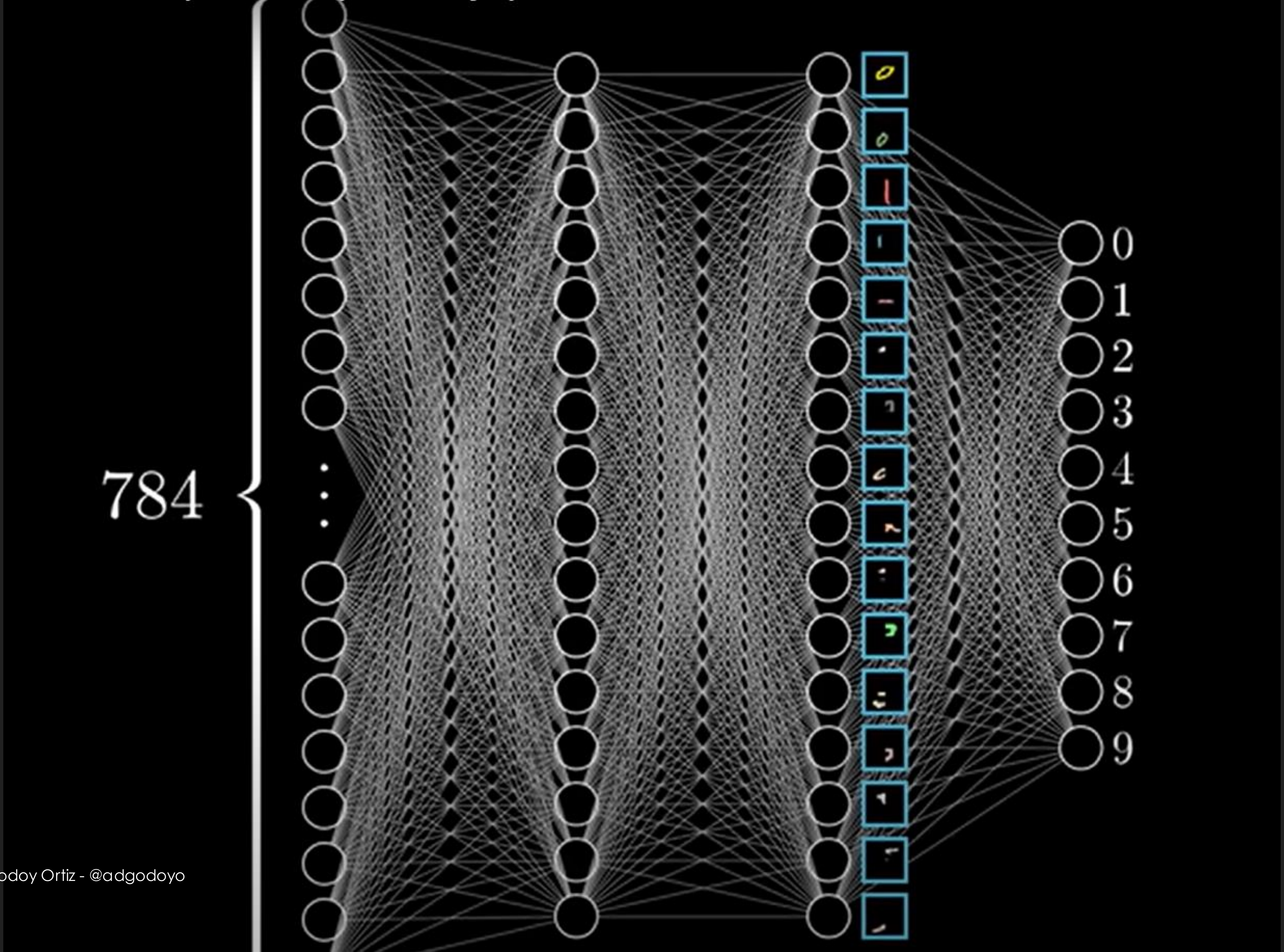
0 0,5 1

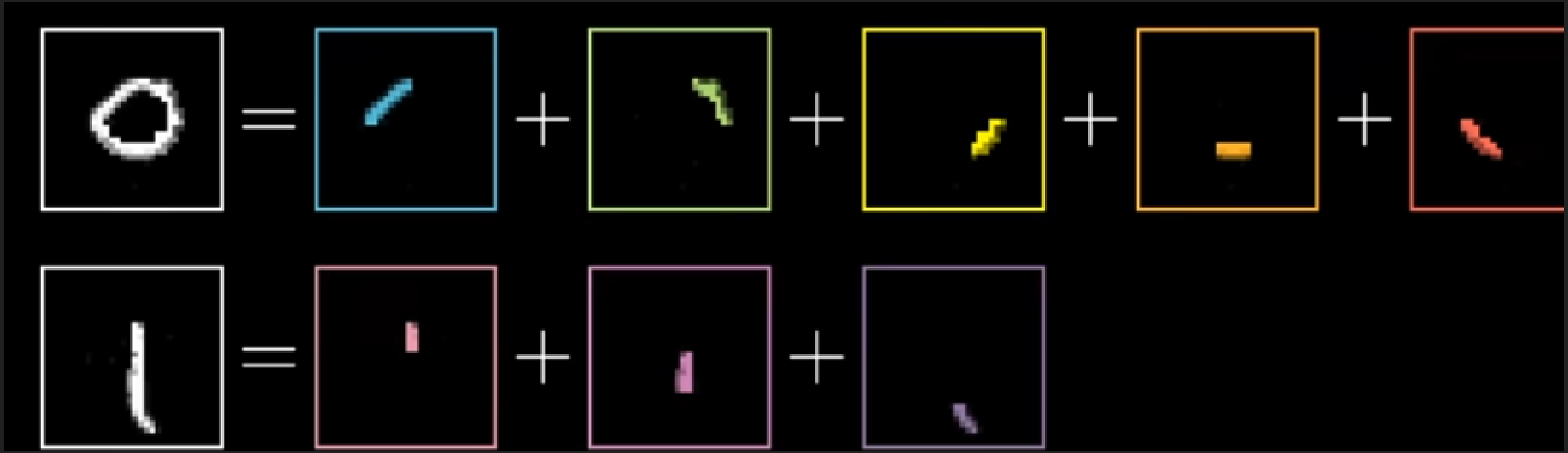
https://playground.tensorflow.org/

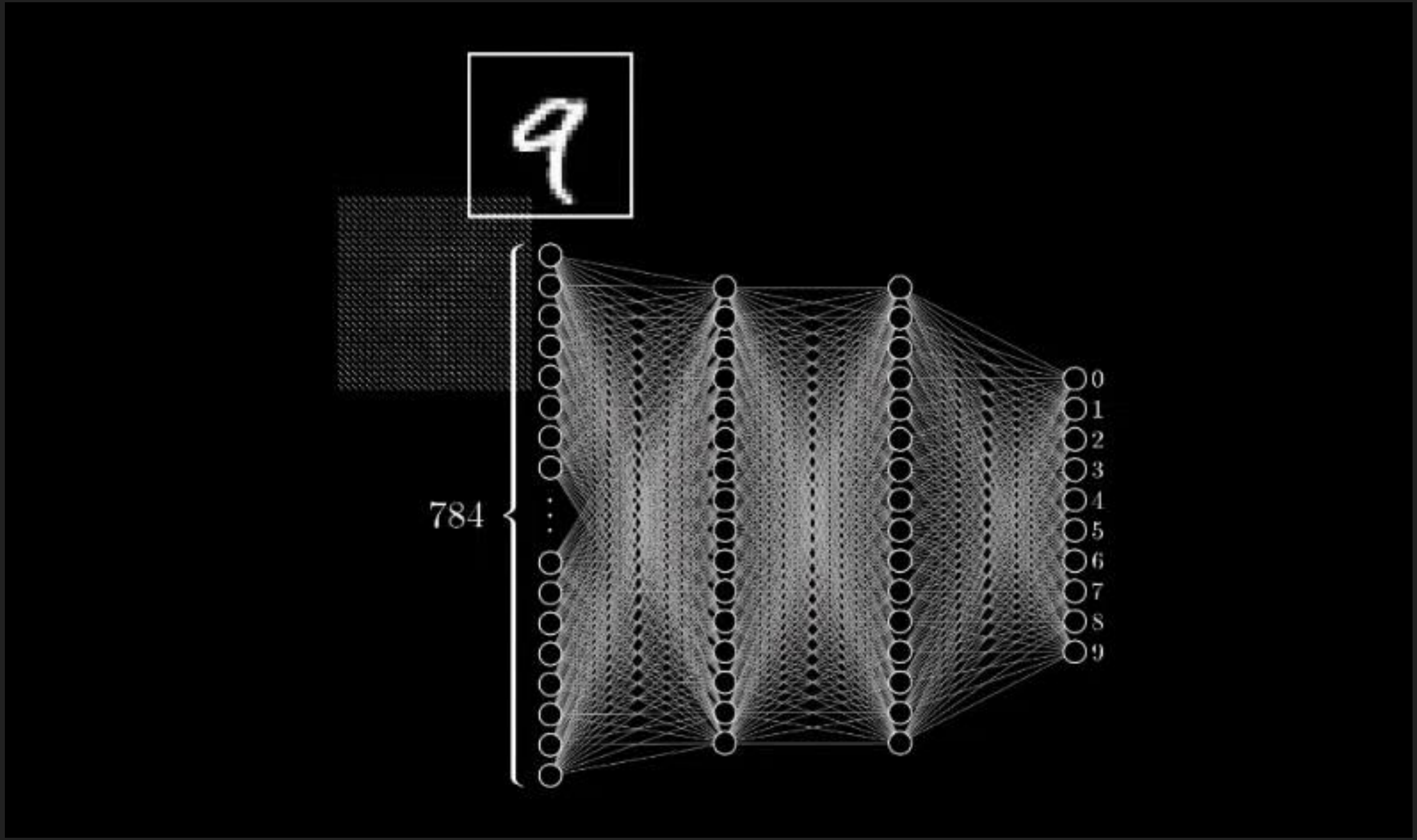
Andrés Daniel Godoy Ortiz


28
28
$28 \times 28 = 784$
1.00



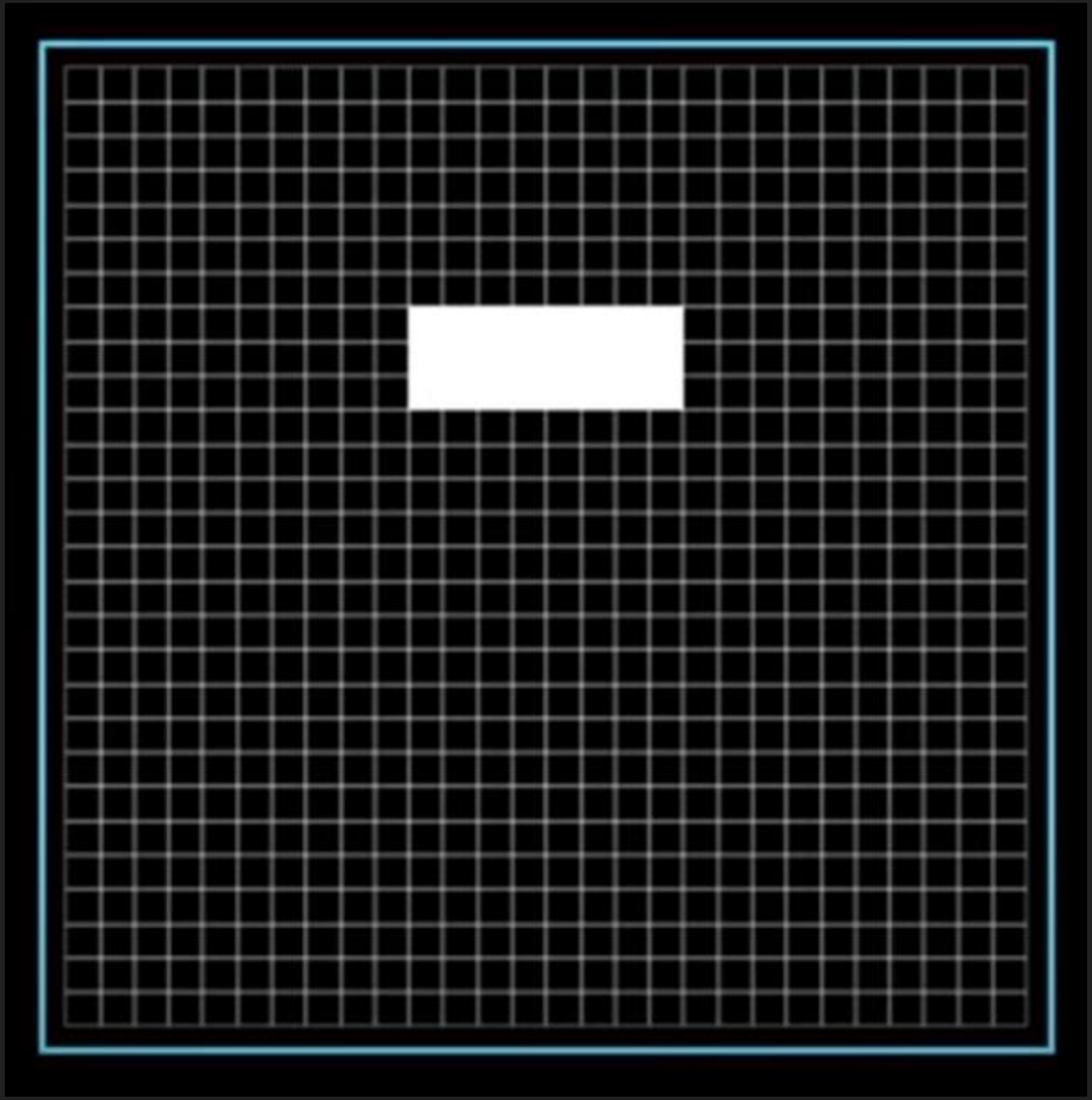

784
0
1
2
3
4
5
6
7
8
9




9
784
0
1
2
3
4
5
6
7
8
9

¿Qué parámetros?

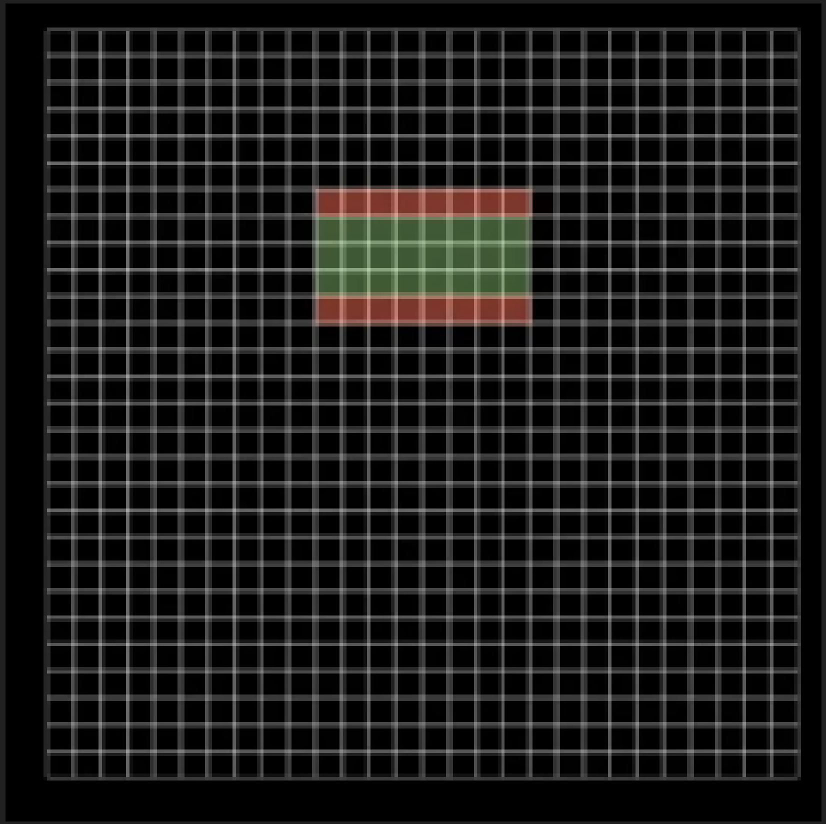
¿Qué pesos debería tener mi red neuronal para detector un objeto como el de la imagen?



Se quisiera que los pixeles en blanco (cuyo valor podemos pensar como 1 y 0 si el pixel es negro), tuvieran un peso muy alto.

Por otro lado, los pixeles circundantes deberían tener un peso incluso negativo.

De forma que justo cuando la red “vea” ese objeto, de como resultado final el valor más alto. SI ve otro objeto, daría un valor bajo.

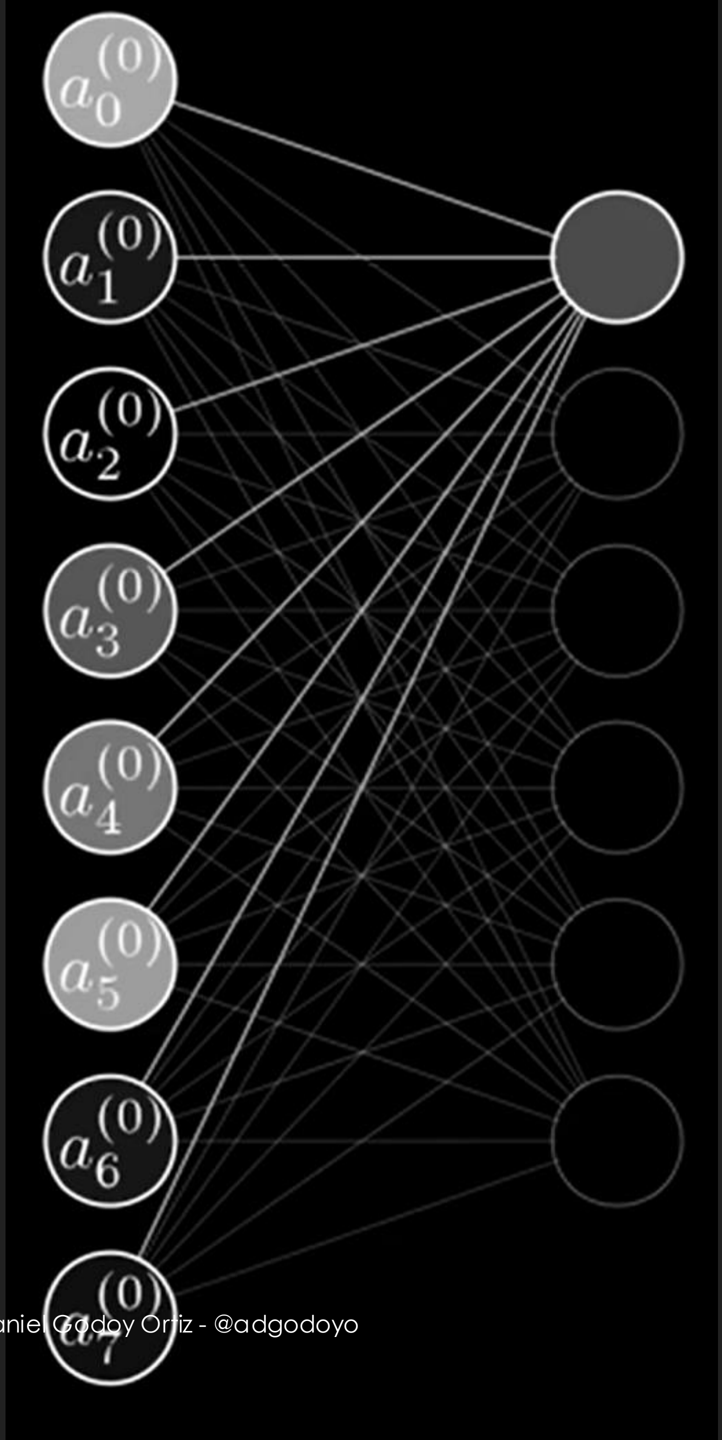


Otra vez, vamos a pensar entonces que una neurona es un número y ya...

0

0,5

1



Una neurona:

$$a_0^{(1)} = \sigma\left(w_{0,0}\; a_0^{(0)} + w_{0,1}\; a_1^{(0)} + \cdots + w_{0,n}\; a_n^{(0)} + b_0 \right)$$

Una capa de neuronas:

$$\sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \right)$$

Notación compacta:

$$\mathbf{a}^{(1)} = \sigma\left(\mathbf{W}\mathbf{a}^{(0)} + \mathbf{b}\right)$$

¿Cómo aprende la red?

Gradiente Descendente

¿Cómo encuentro el gradiente?

Backpropagation

Gracias

adgodoyo@gmail.com