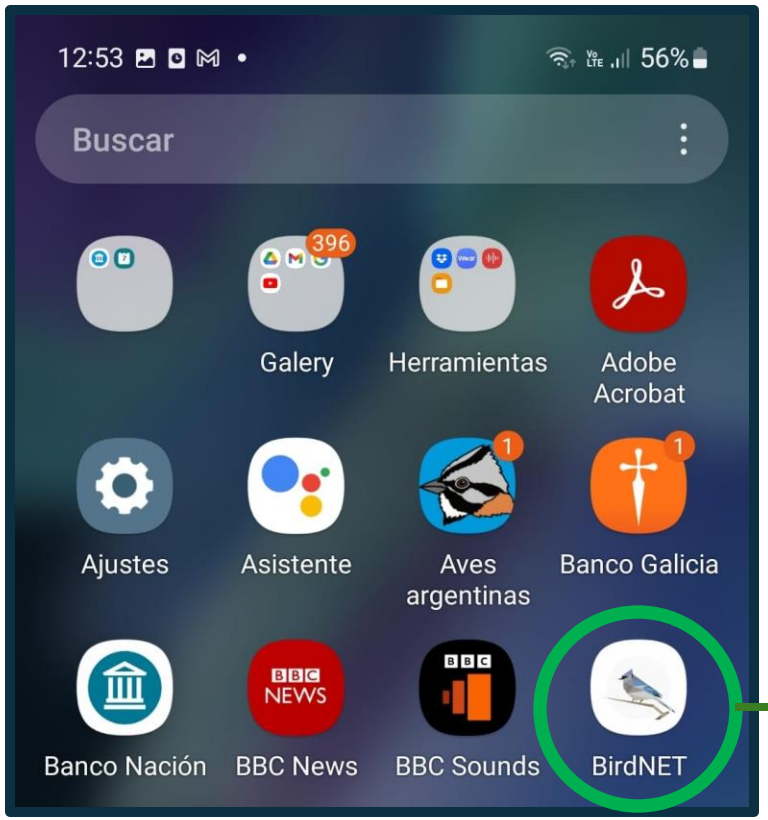
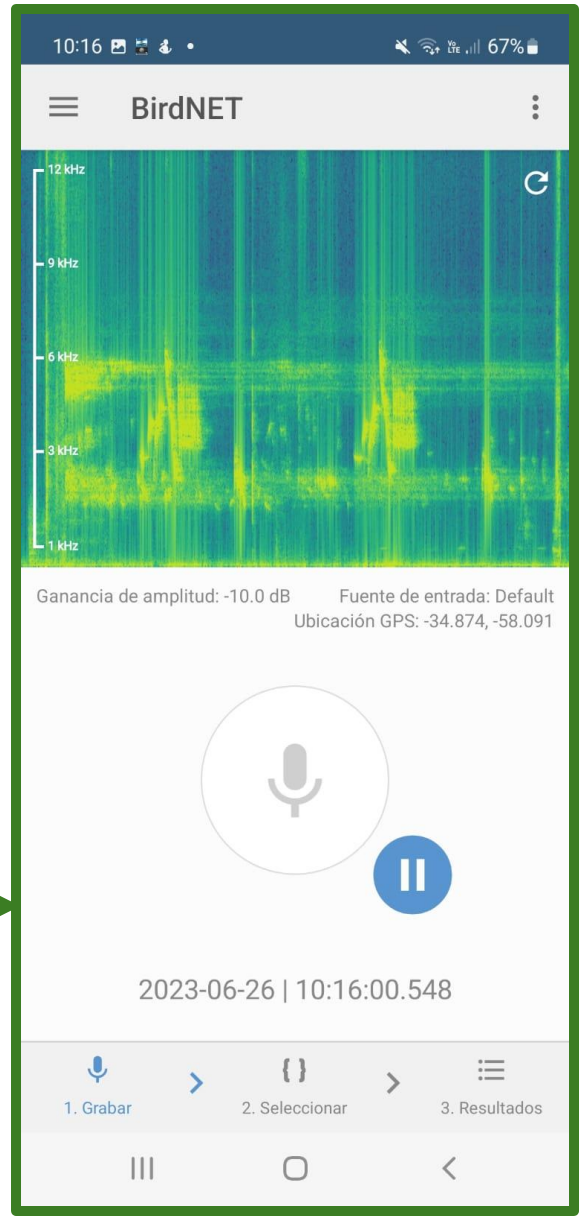
Redes neuronales convolucionales CNN
“El hackeo de la vision”
(Reconocimiento de imagenes)

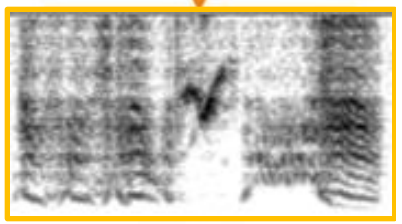




Birdnet translates sound into images
sends the info to a **fitted function**
that was trained to classify,
and the result is sent out
to your phone

What is inside the “brain” that classifies a species in birdnet ?

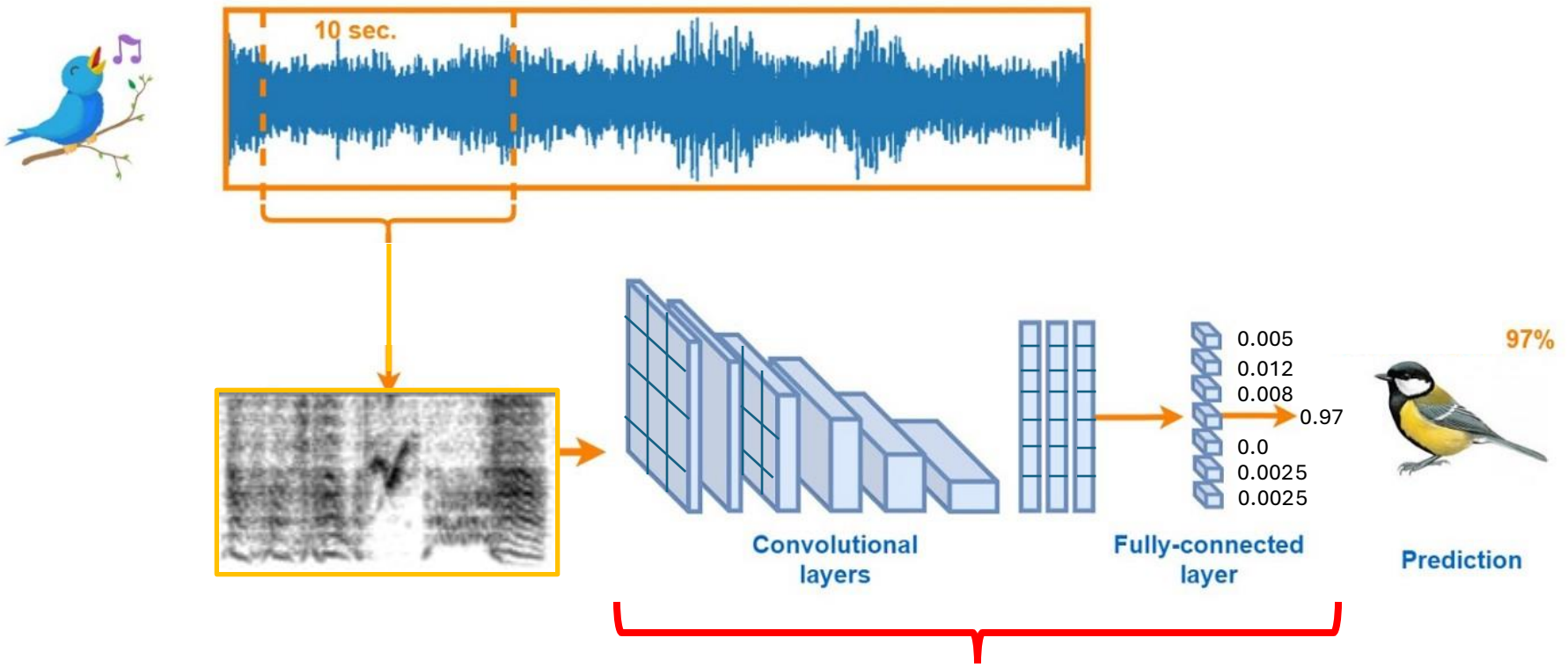
(demystifying AI)



$$f(input) = \begin{bmatrix} \text{probability} \\ \text{of the bird belonging} \\ \text{to a given species} \end{bmatrix}$$

What is inside the “brain” that classifies a species in birdnet ?

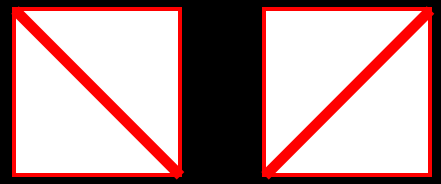
(Demystifying AI)



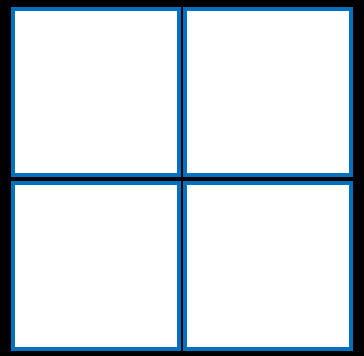
The computational device used to build this function is called

Neural network

Sea un mundo con un alfabeto de dos simbolos

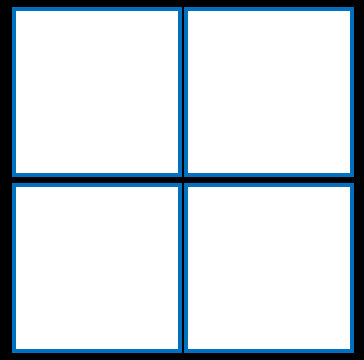


Y tenemos para representarlos, imagenes de 4 pixels



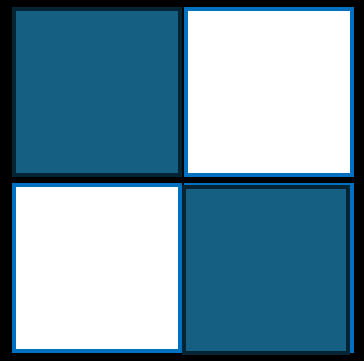
Para representar este simbolo

Con esta pantalla...

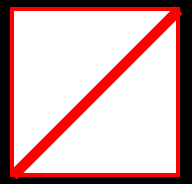


Para representar este simbolo

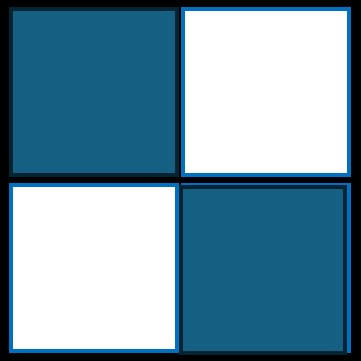
Marcamos los pixeles
de la diagonal

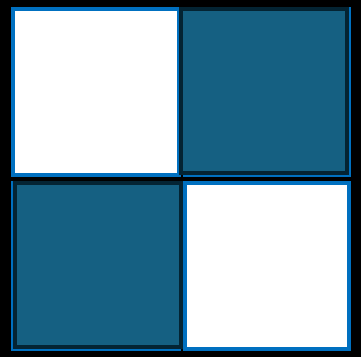


Para representar este simbolo

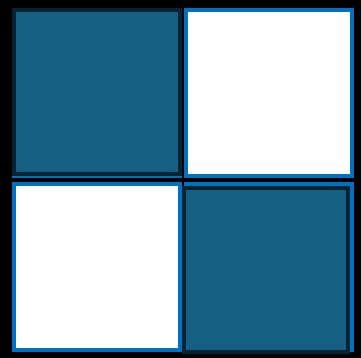


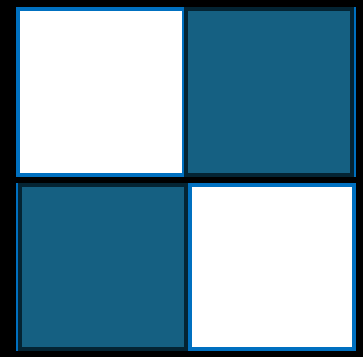
Marcamos los pixeles
de la diagonal

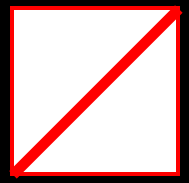


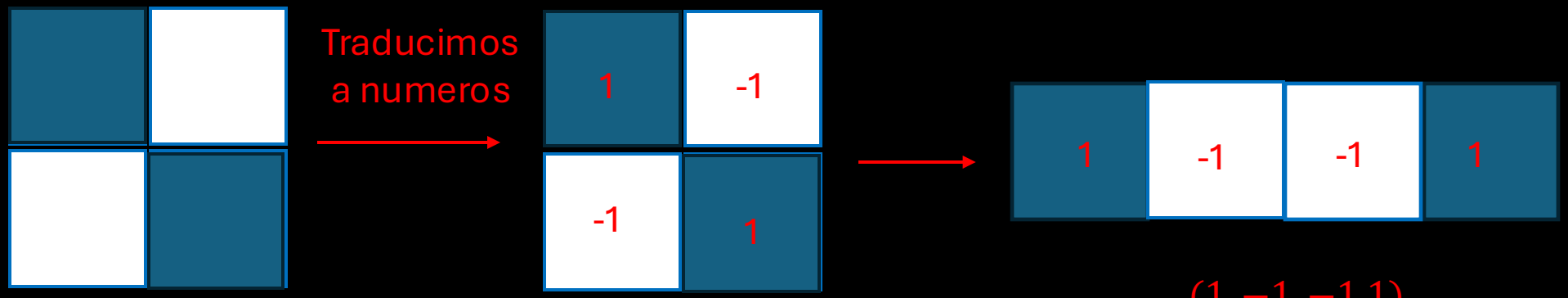


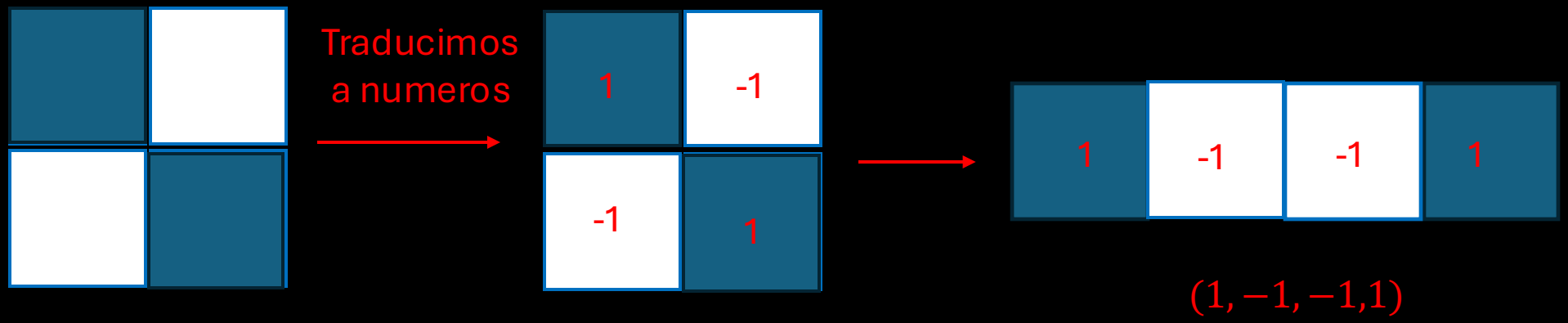
Ahora, queremos un algoritmo que, si le damos las imagenes, Identifiquen al simbolo.

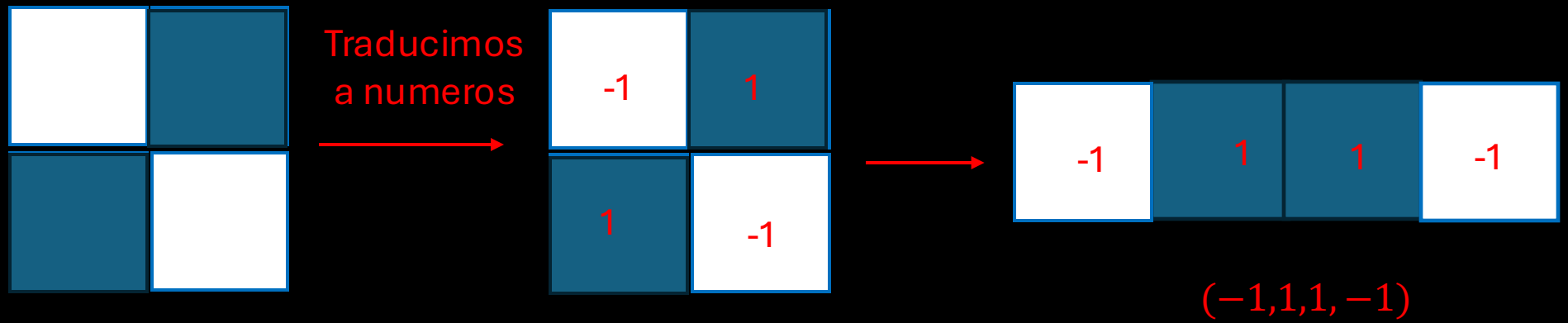




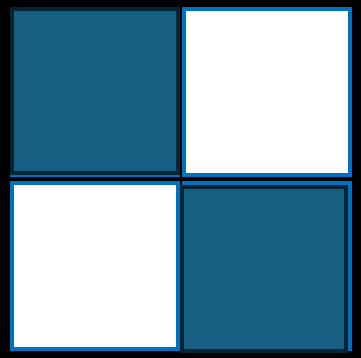



Traducimos
a numeros
1
-1
-1
1
1
-1
-1
1
(1, −1, −1,1)
Traducimos
a un vector

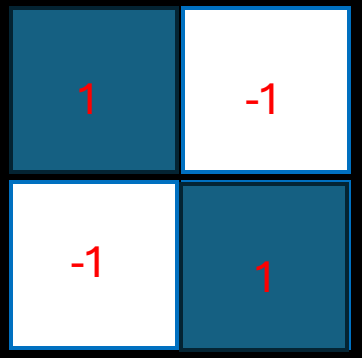

Traducimos
a numeros
1
-1
-1
1
1
-1
-1
1
(1, −1, −1,1)

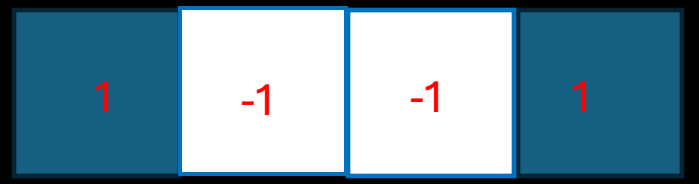

Traducimos
a numeros
-1
1
1
-1
-1
1
1
-1
(−1,1,1, −1)

Podemos definir una operacion sobre estos numeros para poder distinguir/separar los vectores?

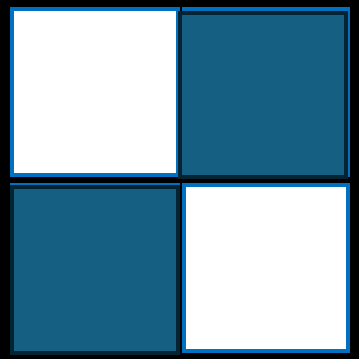


Traducimos a numeros

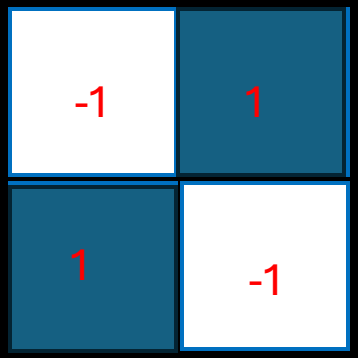


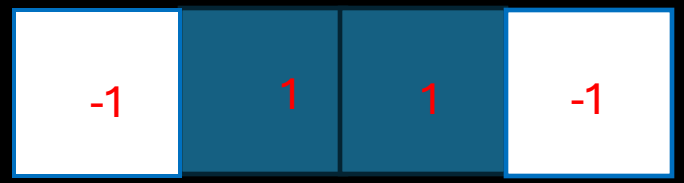


$(1, -1, -1, 1)$



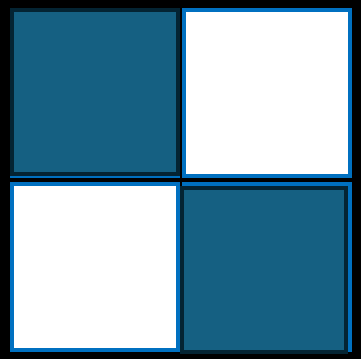
Traducimos a numeros



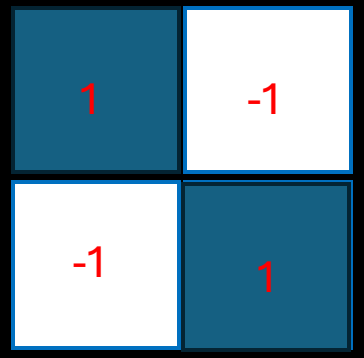


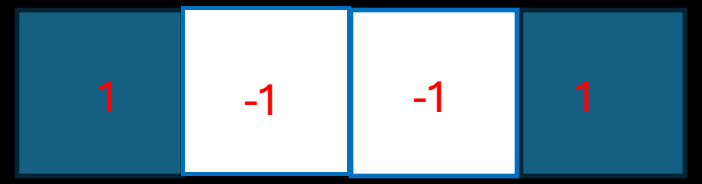
$(-1, 1, 1, -1)$

Podemos definir una operacion sobre estos numeros
para poder distinguir/separar los vectores? Sumar componentes?



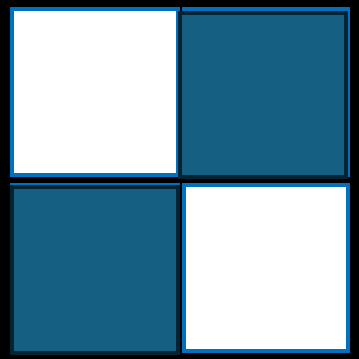
Traducimos
a numeros



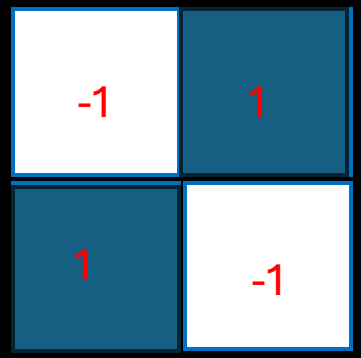


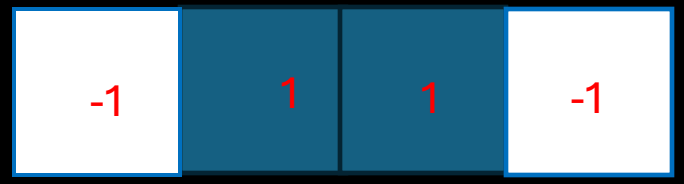
0

$(1, -1, -1, 1)$



Traducimos
a numeros

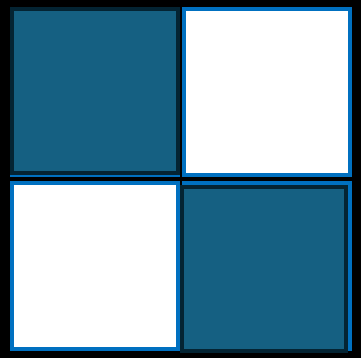




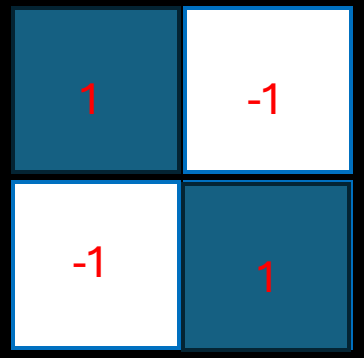
0

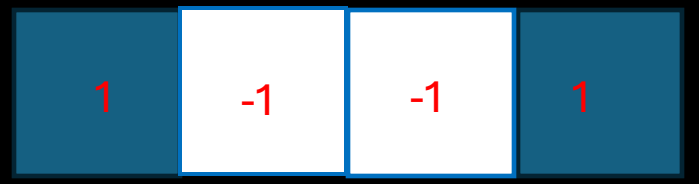
$(-1, 1, 1, -1)$

Podemos definir una operacion sobre estos numeros
para poder distinguir/separar los vectores? multiplicar componentes?



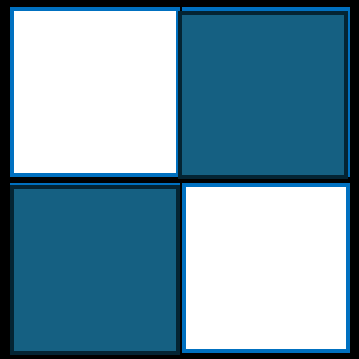
Traducimos
a numeros



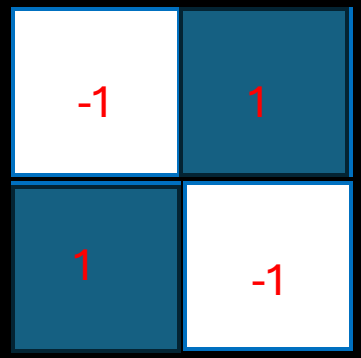


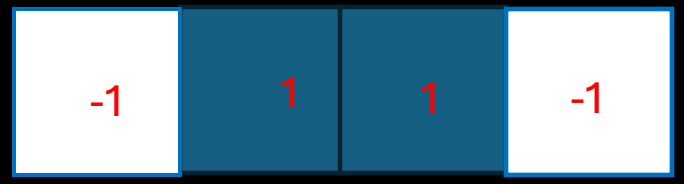
1

$(1, -1, -1, 1)$



Traducimos
a numeros



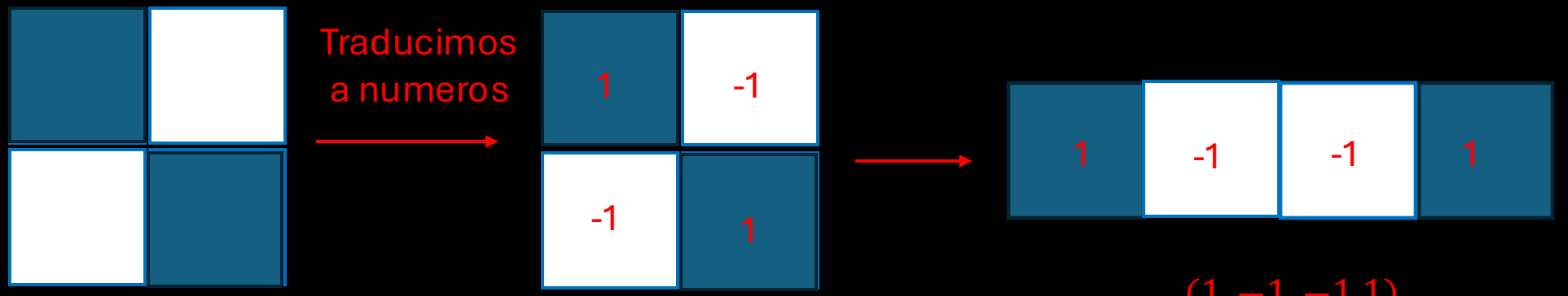


1

$(-1, 1, 1, -1)$

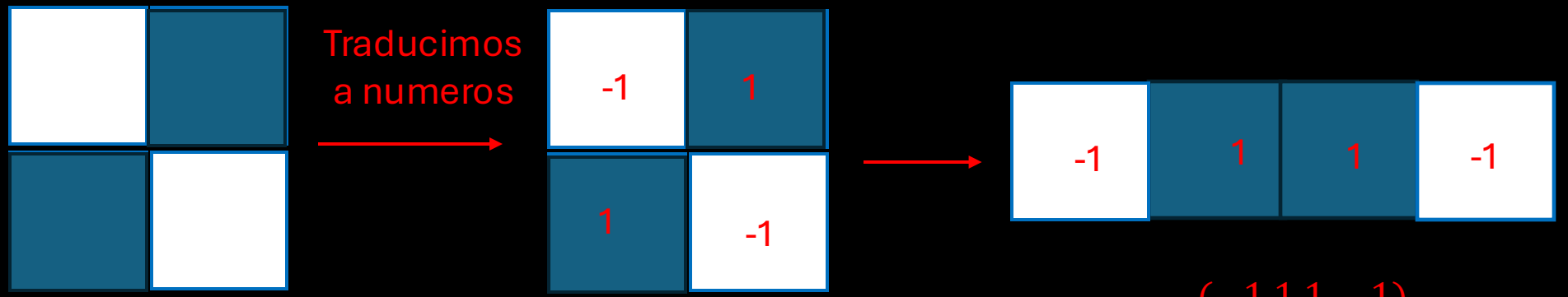
Podemos definir una operacion sobre estos numeros
para poder distinguir/separar los vectores? Producto escalar con un vector particular?

$(1, -1, -1, 1)$



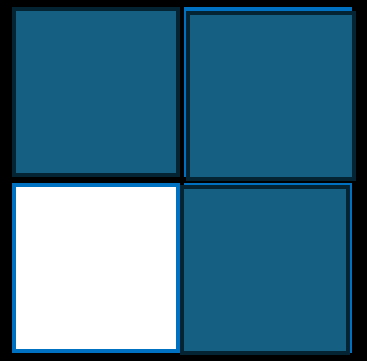
4

$(1, -1, -1, 1)$



-4

$(-1, 1, 1, -1)$



$$\begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

$(1,1,-1,1)$ 2

Un numero “cercano” a 4,
que refleja que el patron es una diagonal
”a la izquierda” mal dibujada

Como elegimos este vector?

$$\begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

Como nos dimos cuenta que con este “filtro”
Ibamos a separar bien a nuestros dos vectores?

Podriamos habe intentado con este

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Pero ya vimos que con ese no se podian distinguir.
Daban cero los productos escalares con ambos vectores
(lo llamamos “sumar las componentes”)

Como se busca el vector ideal para hacer el producto escalar, y separar,
es ir cambiando de a poco
y ver como va cambiando la diferencia de productos escalares

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

Haciendo los productos
con este no va muy bien

$$\begin{pmatrix} -1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$ peor

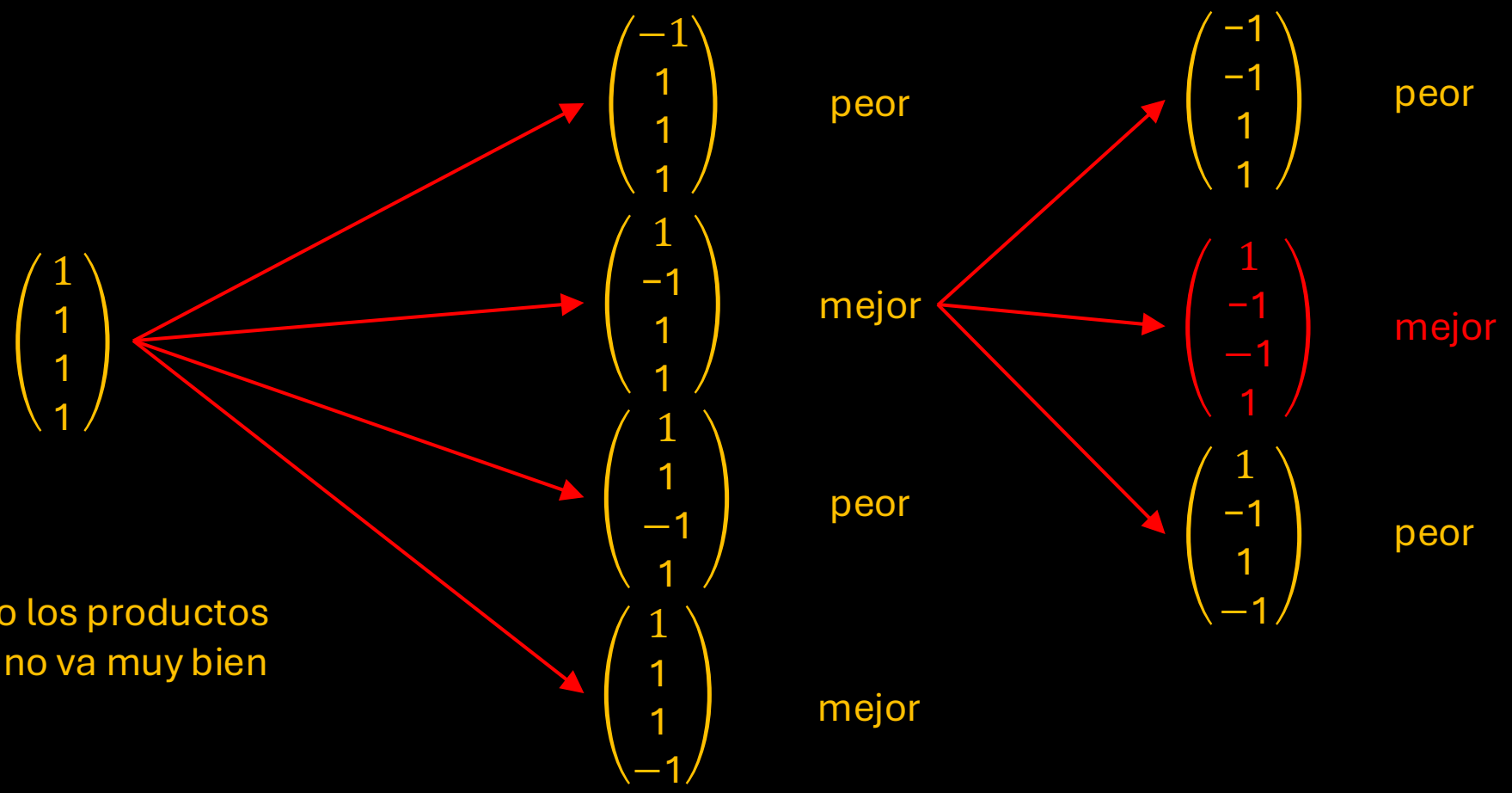
$$\begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$ mejor

$$\begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$$ peor

$$\begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$ mejor

No obvio

Como se busca el vector ideal para hacer el producto escalar, y separar,
es ir cambiando de a poco
y ver como va cambiando la diferencia de productos escalares

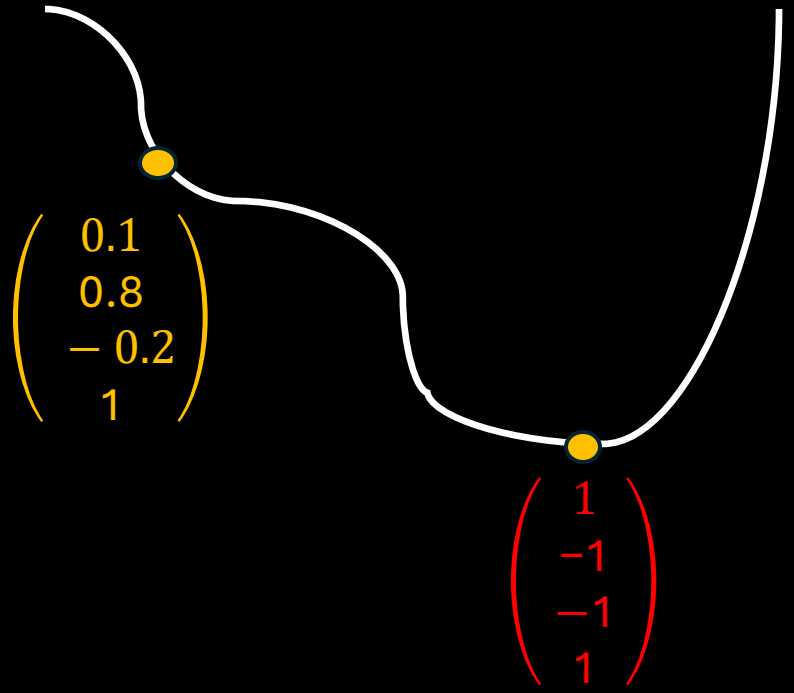


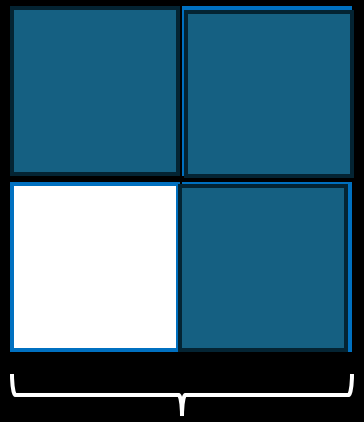
Haciendo los productos
con este no va muy bien

No obvio

Metodo de gradient descent

Errores al intentar separar

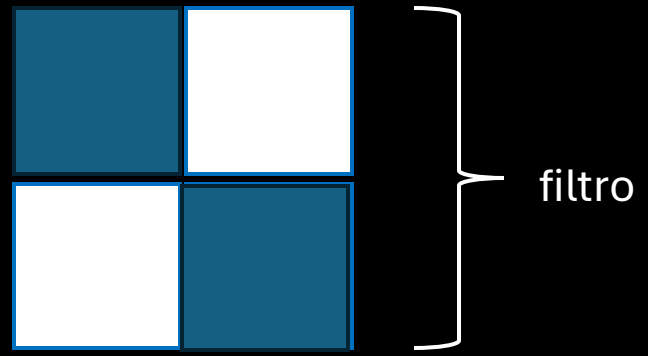




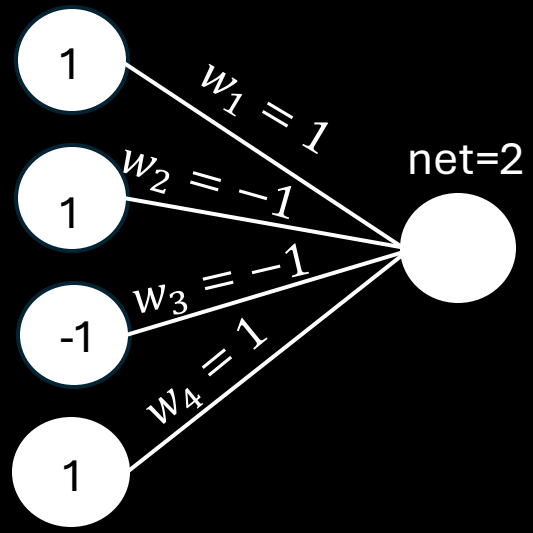
patron

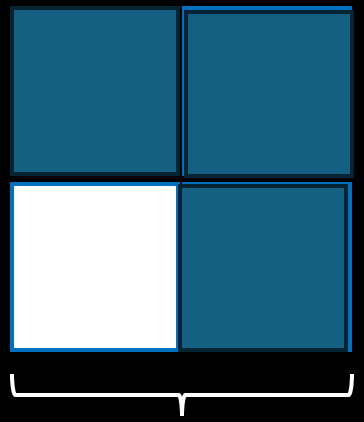
$$\begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$$

$(1,1,-1,1)$



filtro

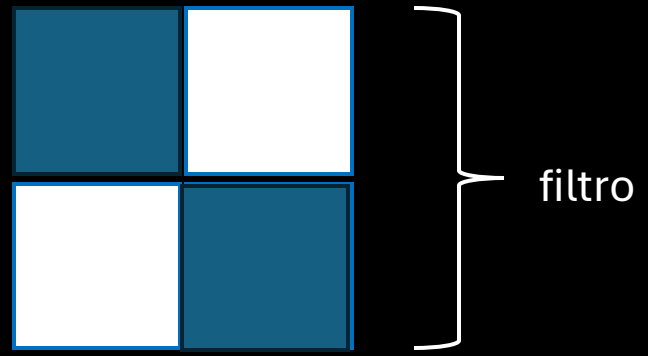




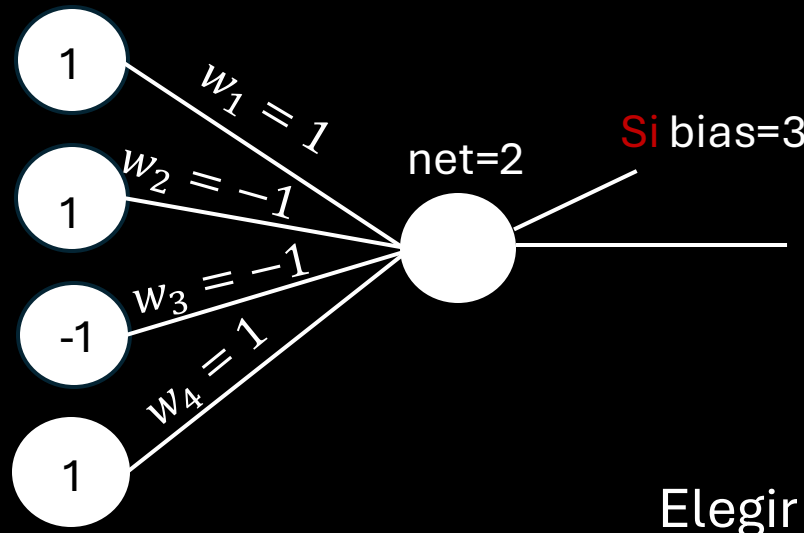
patron

$\begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}$

$(1,1,-1,1)$

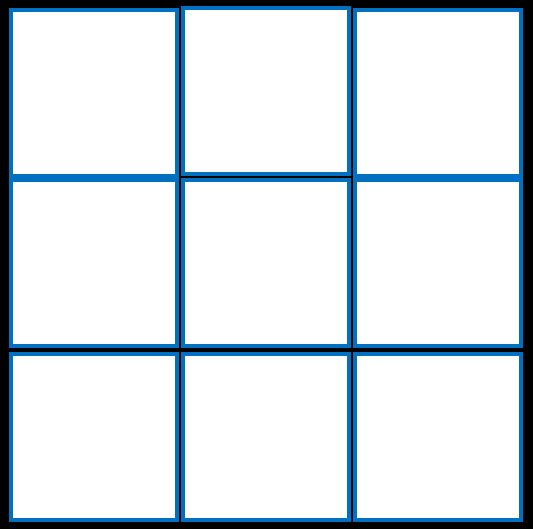


filtro



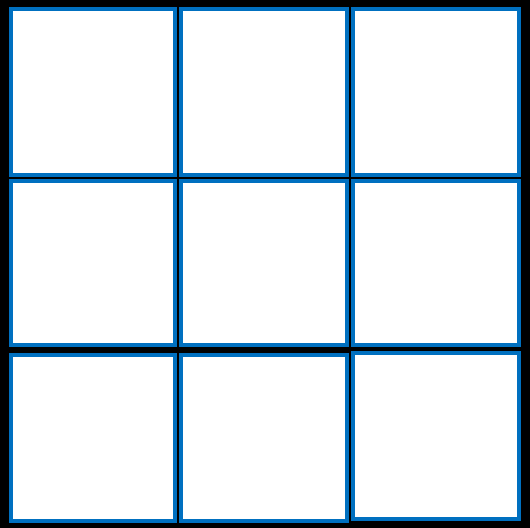
Elegir filtro, es elegir pesos

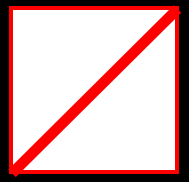
Complejifiquemos el mundo

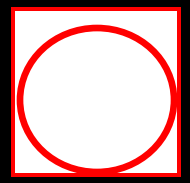


Mas resolucion para las pantallas

Complejifiquemos el mundo

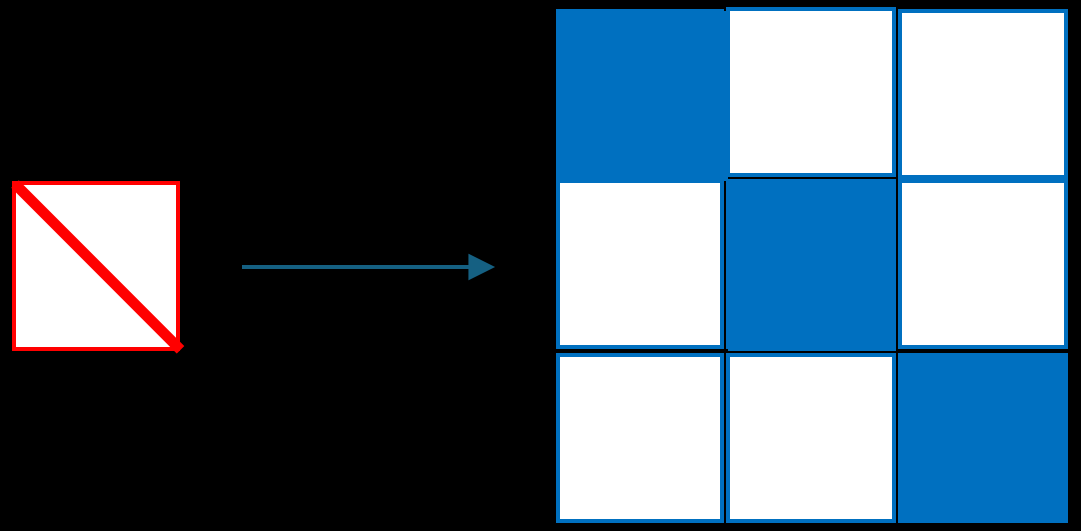


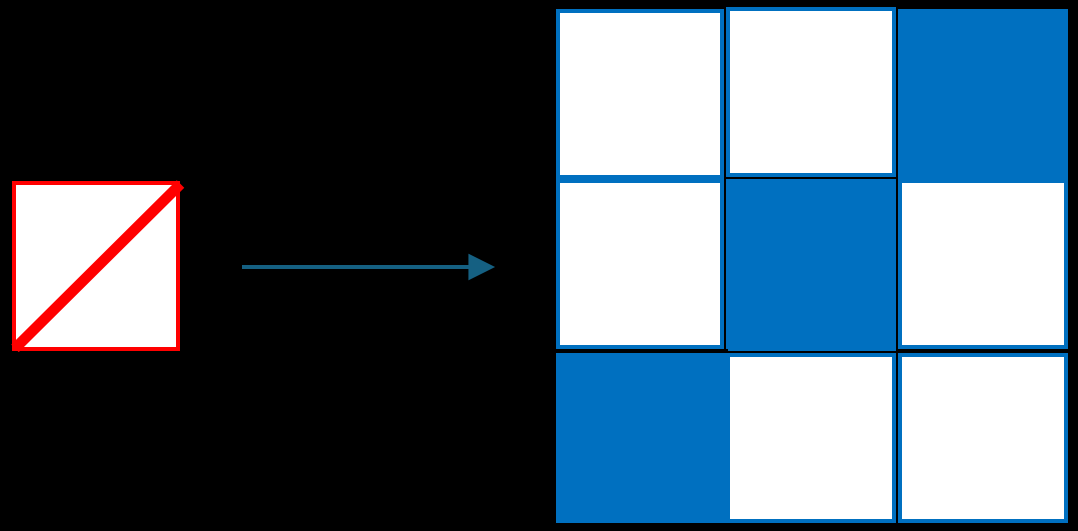


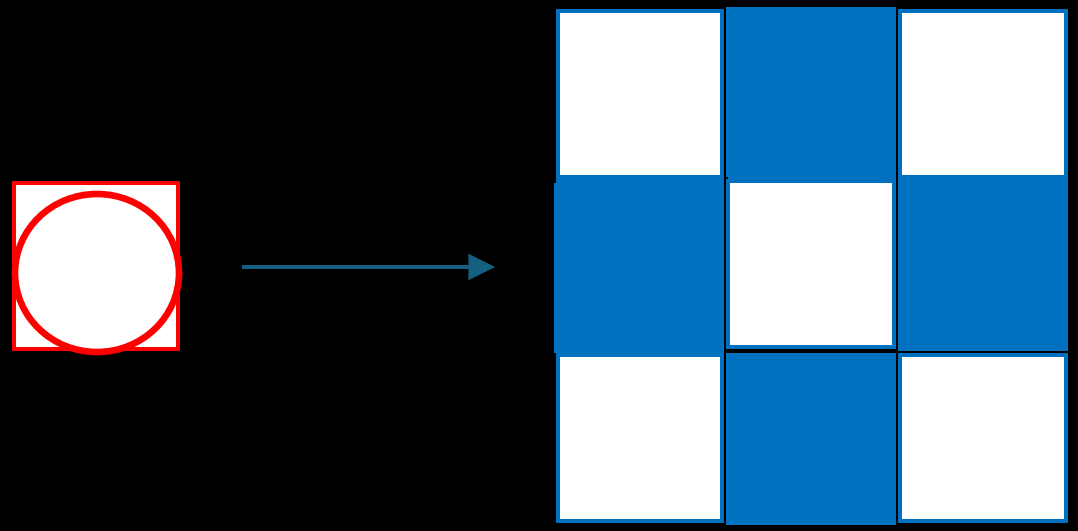


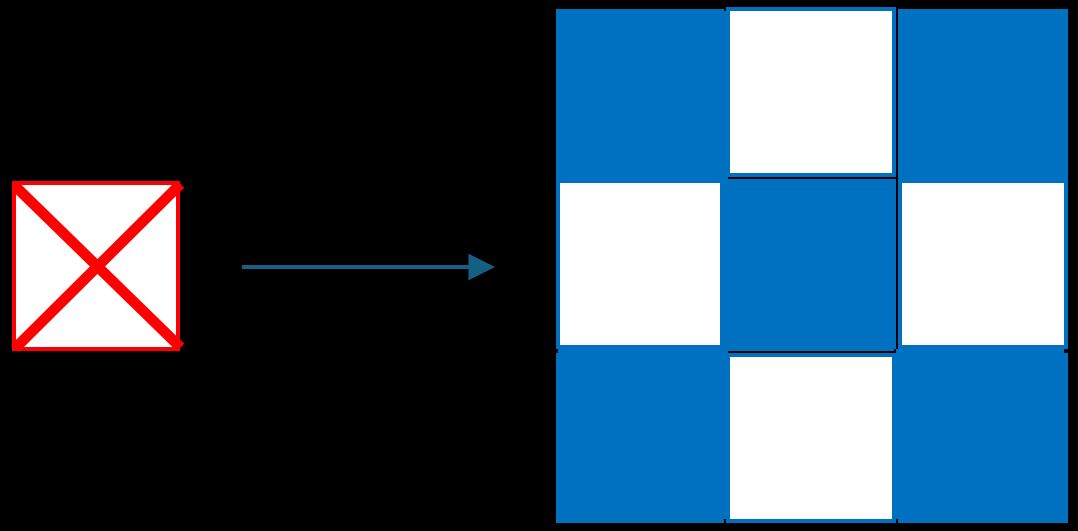
Mas resolucion para las pantallas

Mas simbolos





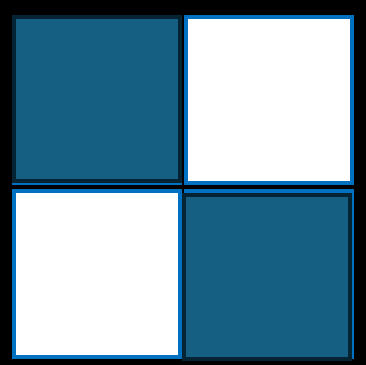


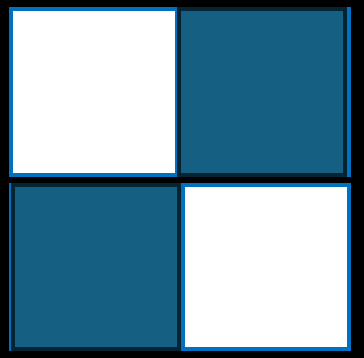


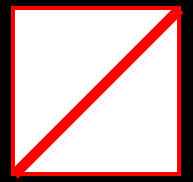
vamos a tratar de resolver el problema inverso,
o sea, dado un patron, reconocer la letra

Y vamos a usar el conocimiento previo para resolver el problema

Vimos como separar estos dos patrones



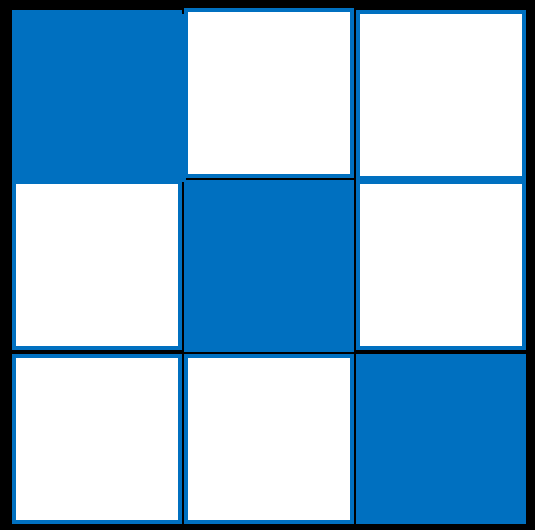


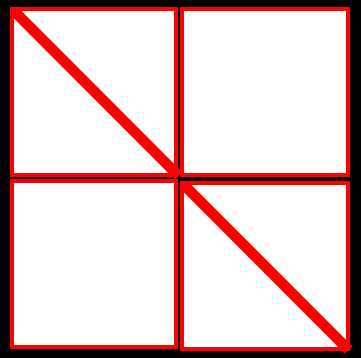


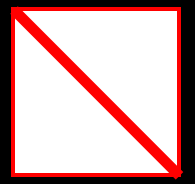
Podemos usarlos para armar los nuevos simbolos?

Primero, la idea general

Reconociendo trocitos de patron



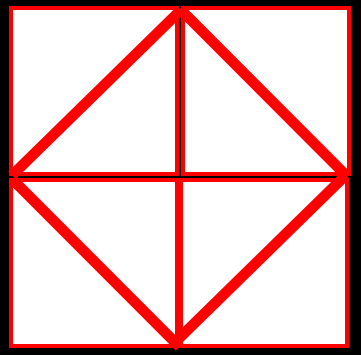


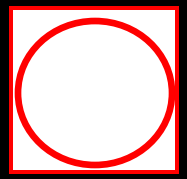


Luego asociaremos este patron
a un simbolo

Primero, la idea general

Reconociendo trocitos de patron

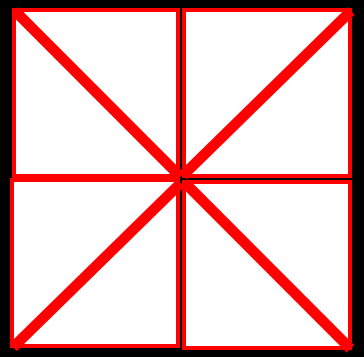




Luego asociaremos este patron
a un simbolo

Primero, la idea general

Reconociendo trocitos de patron

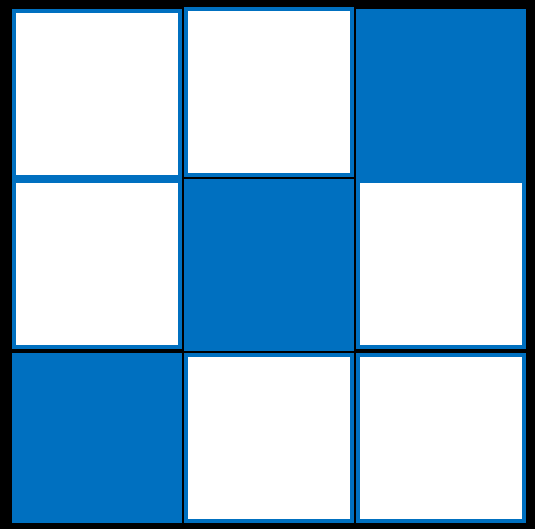


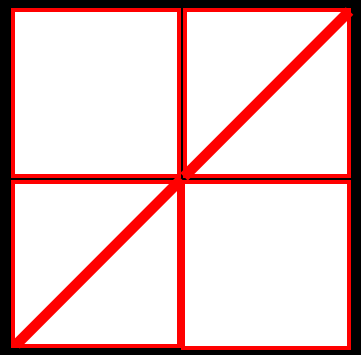
Luego asociaremos este patron
a un simbolo

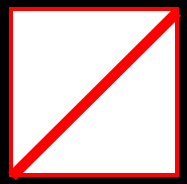
Primero, la idea general

Reconociendo trocitos de patron

(Convolution layer
Y pooling)





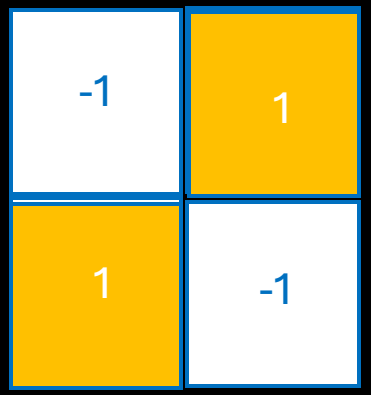


Luego asociaremos este Patrón
a un símbolo

(Fully connected
Layer)

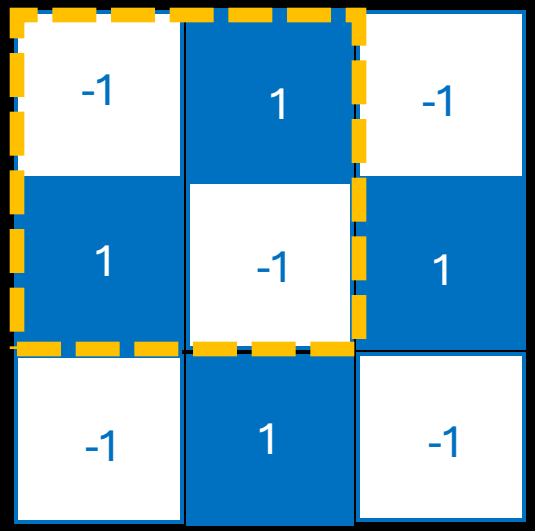
Como lo hacemos?

Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos



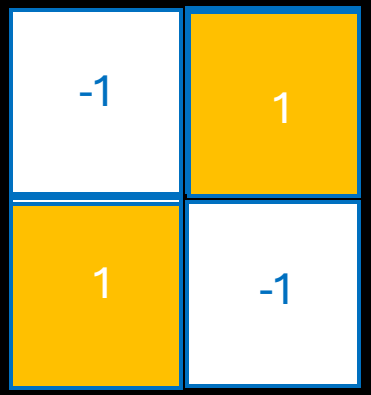
$\rightarrow (-1\ 1\ 1\ -1)$

$(-1\ 1\ 1\ -1) \leftarrow$

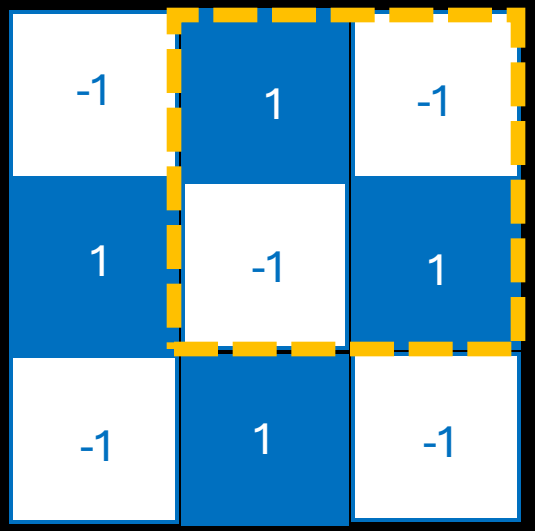


Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos

Como lo hacemos?

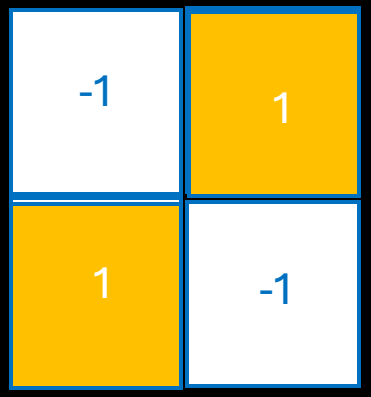


$\rightarrow (-1\ 1\ 1\ -1)$

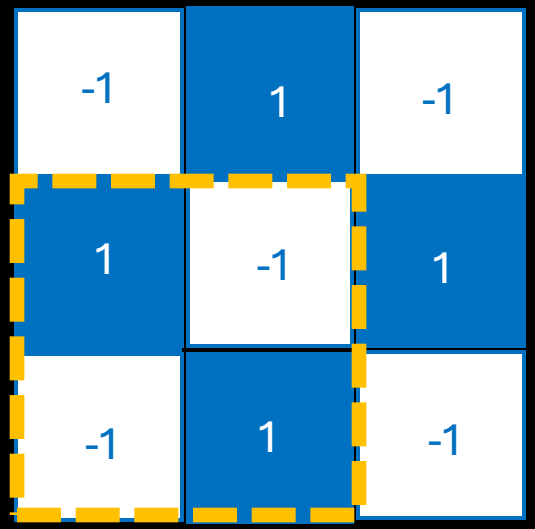


$\rightarrow (1\ -1\ -1\ 1)$

Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos



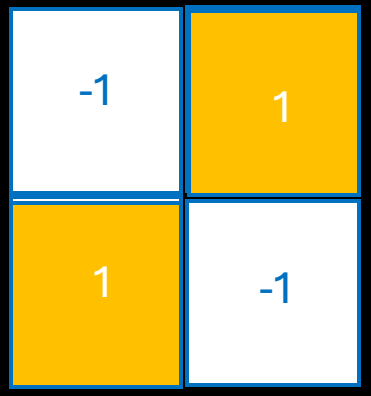
$\rightarrow (-1\ 1\ 1\ -1)$



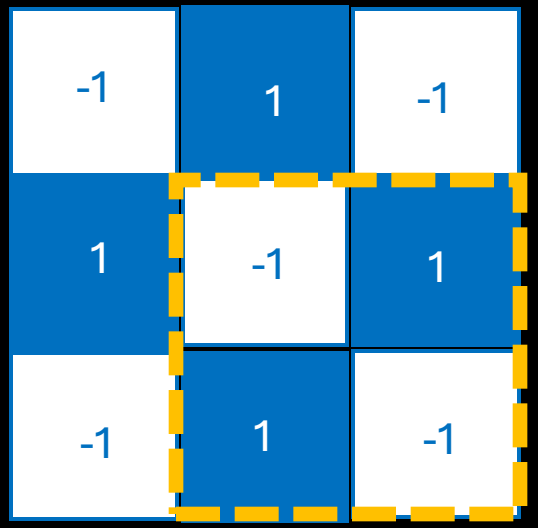
$(1\ -1\ -1\ 1) \leftarrow$

Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos

Como lo hacemos?



$\rightarrow (-1\ 1\ 1\ -1)$

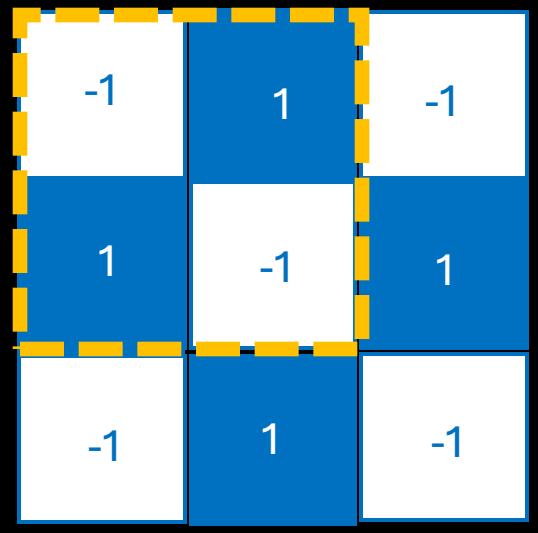


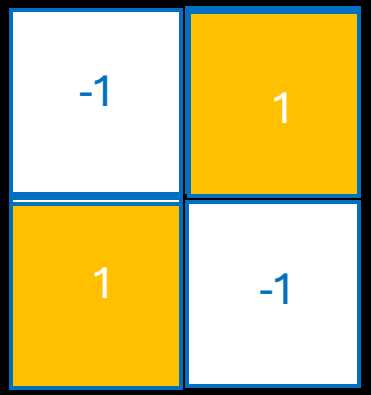
$\rightarrow (1\ -1\ -1\ 1)$

Como lo hacemos?

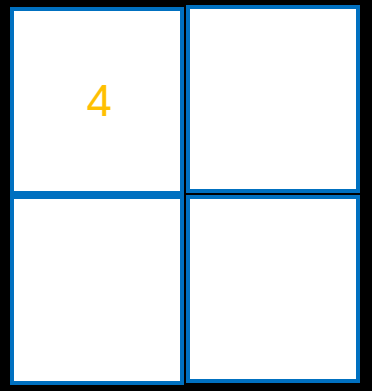
Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos

$(-1\ 1\ 1\ -1) \leftarrow$

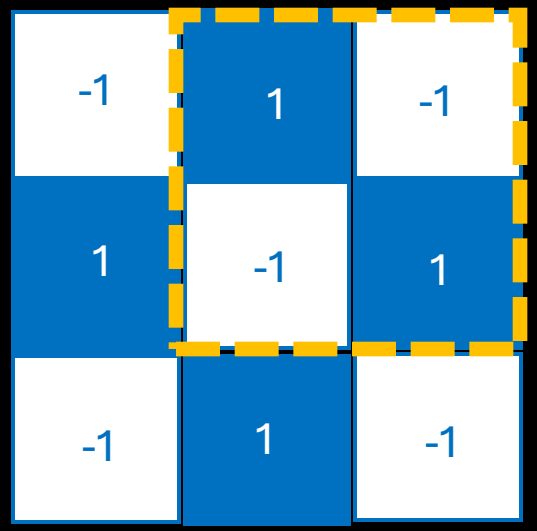




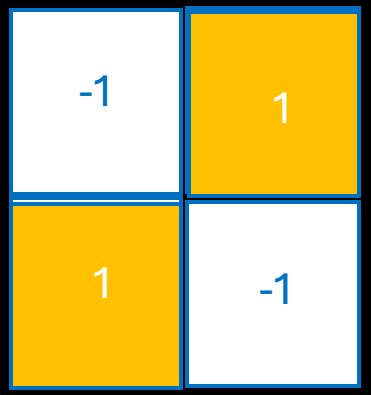
$(-1\ 1\ 1\ -1)\begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$



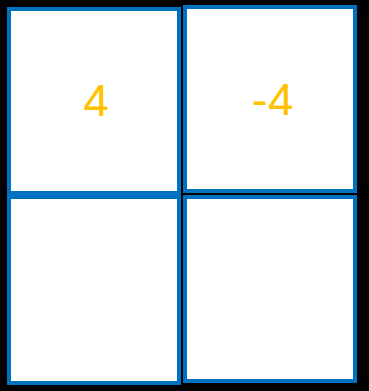
Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos



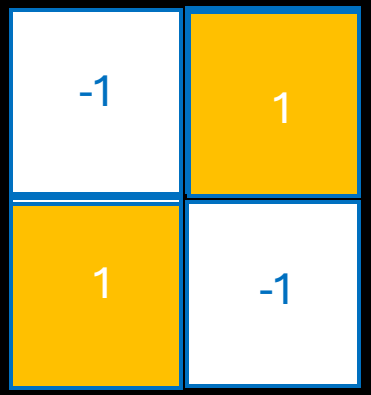
$\rightarrow (1 \ -1 \ -1 \ 1)$

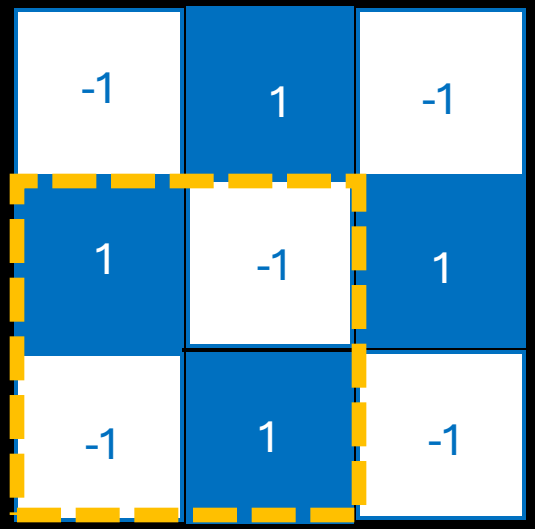


$(1 \ -1 \ -1 \ 1)\begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$



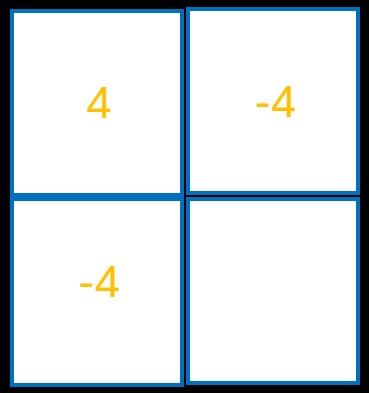
Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos



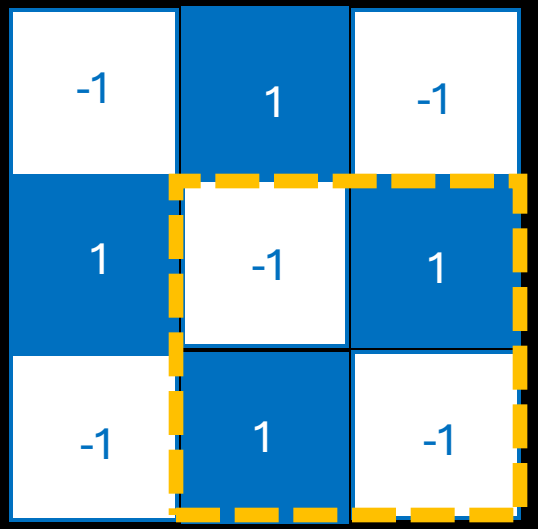


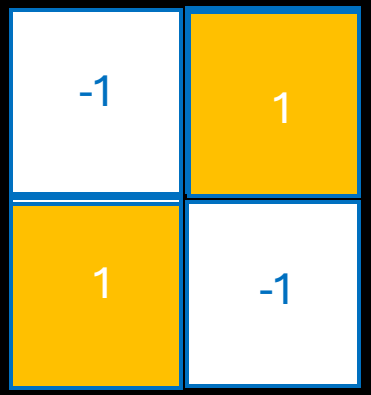
$(1\ -1\ -1\ 1) \leftarrow$

$(1\ -1\ -1\ 1)\begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$

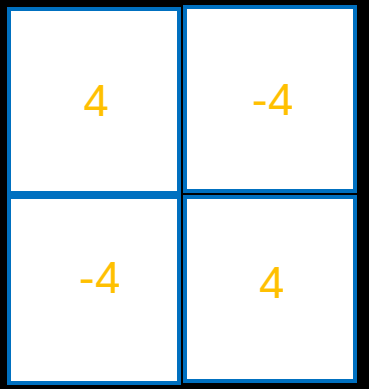


Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos





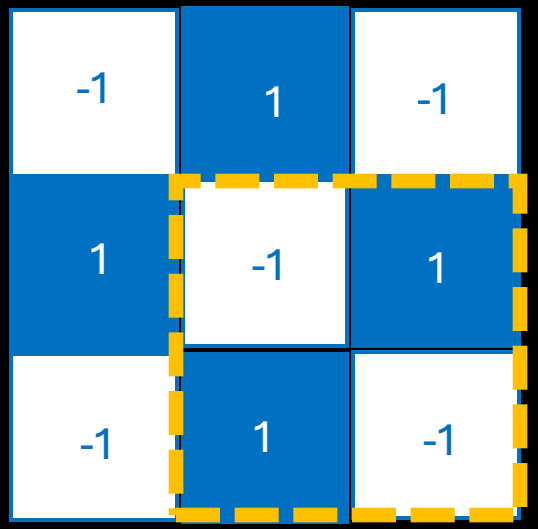
$\rightarrow (1 \ -1 \ -1 \ 1)$

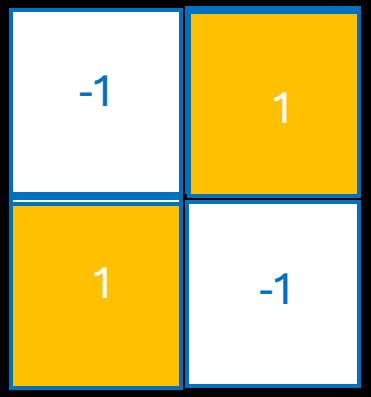


$$(-1\ 1\ 1\ -1)\begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$$

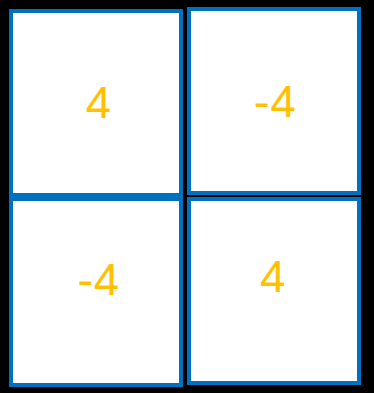
Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos

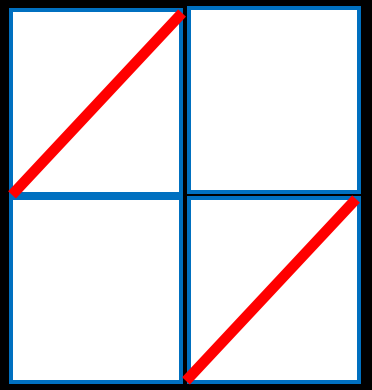
Pongamos threshold en 3,
Para “reconocer” el simbolo





$\rightarrow (1 \ -1 \ -1 \ 1)$



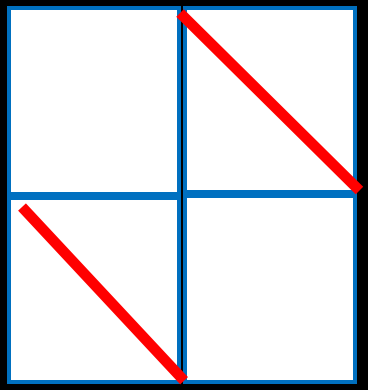


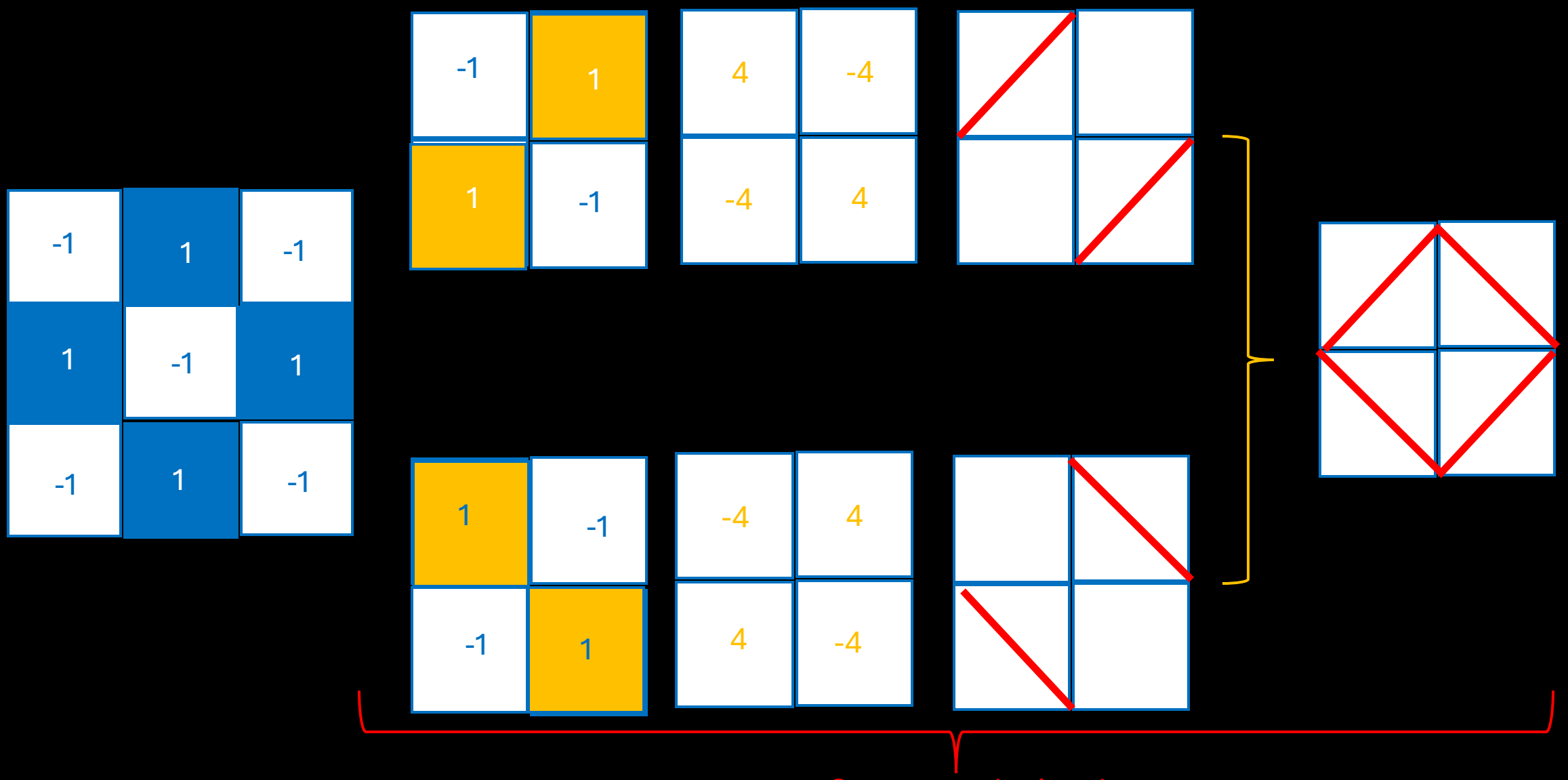
Me fijo si reconozco este simbolo analizando trocitos como analice las cosas en el mundo de dos simbolos

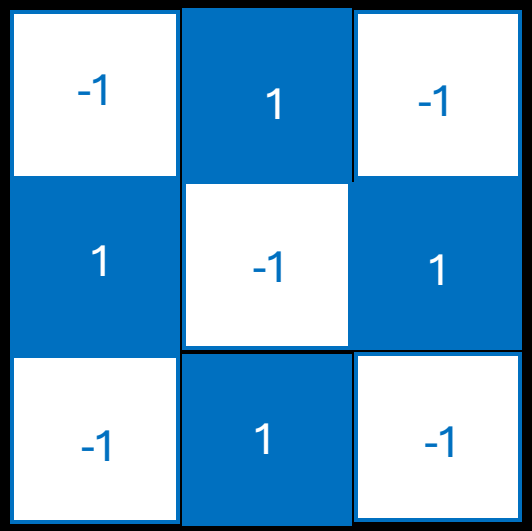
-1	1	-1
1	-1	1
-1	1	-1

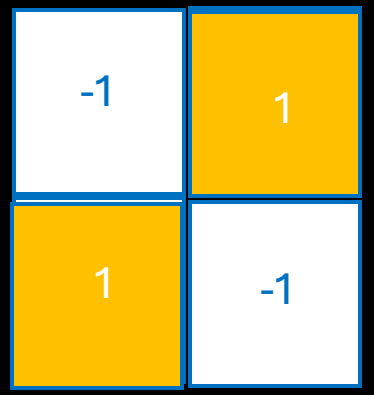
1	-1
-1	1

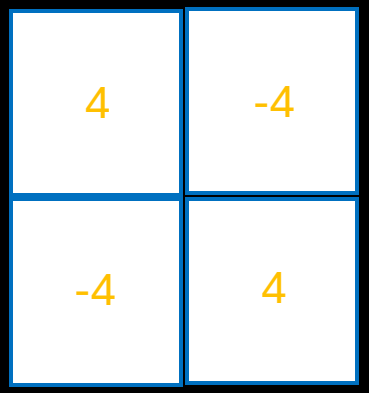
-4	4
4	-4

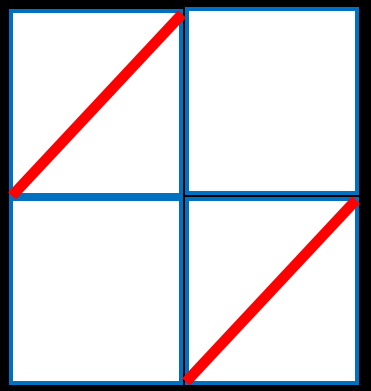


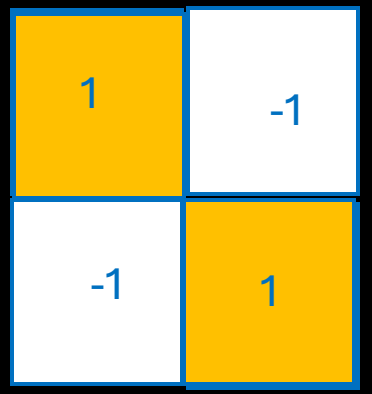

-1
1
-1
1
-1
1
-1
1
-1
-1
1
-1
1
-1
4
-4
-4
4
-4
4
4
-4
Capa convolucional
(producto escalar con filtros)

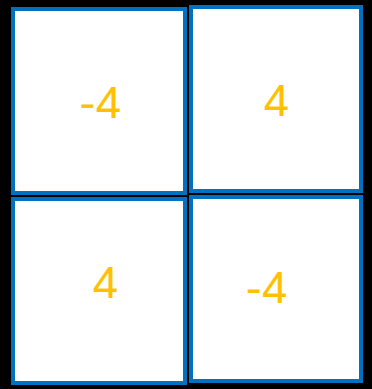


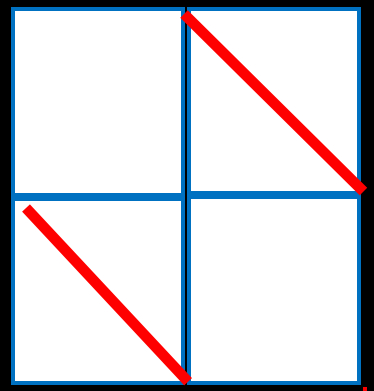


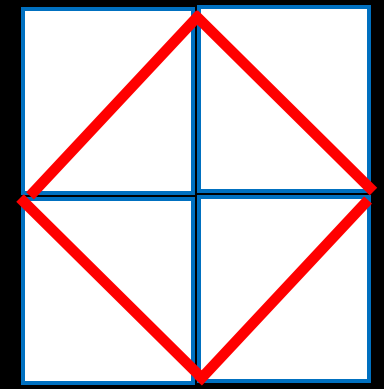




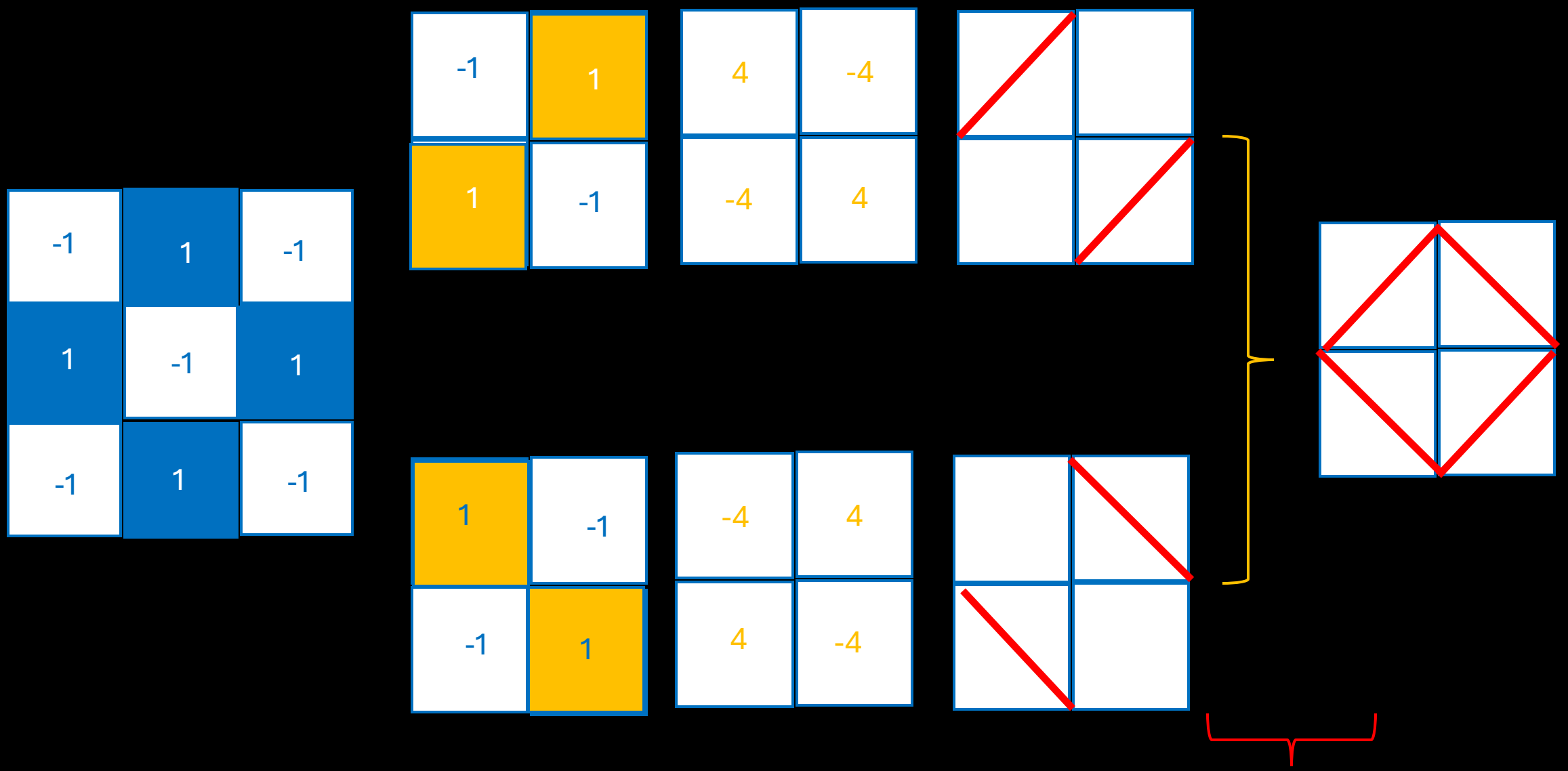




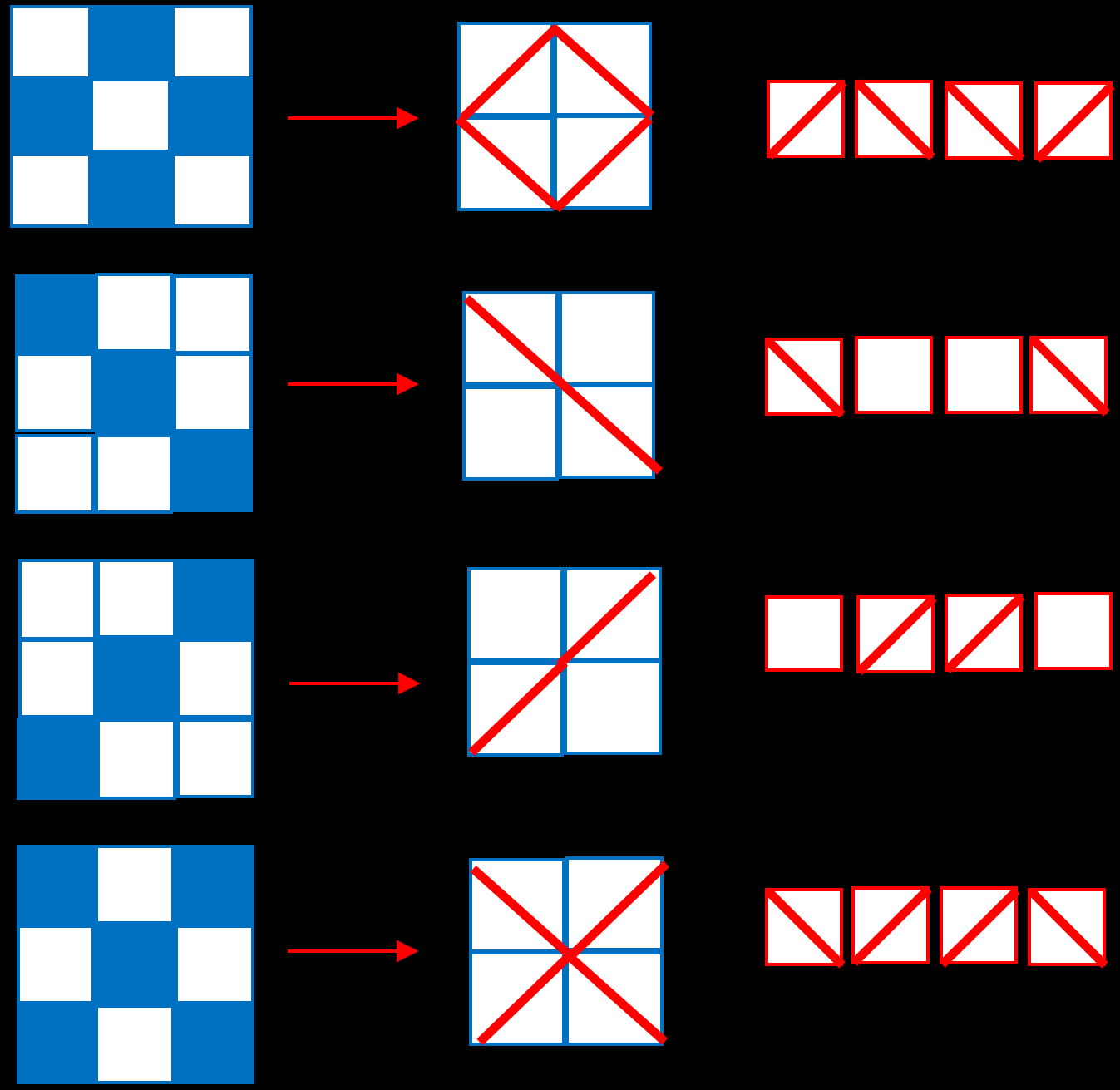




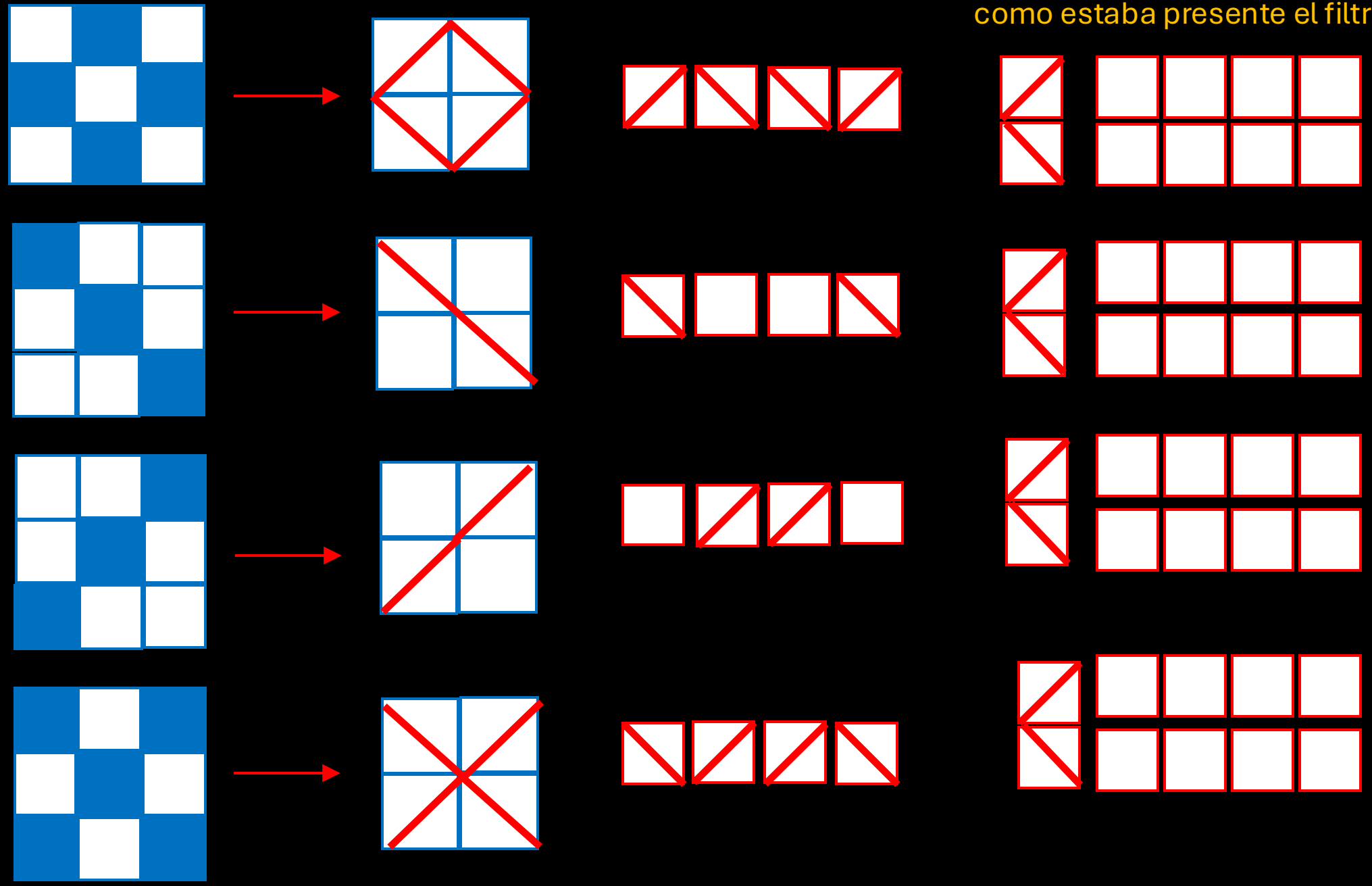
via umbrales declara encontrar
lo que buscaban los filtros


-1
1
-1
1
-1
1
-1
1
-1
-1
1
-1
1
-1
4
-4
-4
4
1
-1
-1
1
-4
4
4
-4
Finalmente se combina todo lo que
Se encontro parcilamente al filtrar

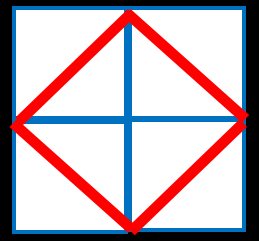
extendidas las podemos representar asi

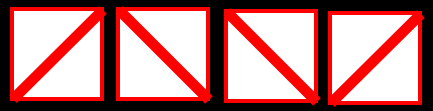


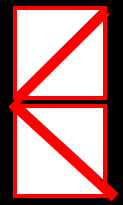
Y matricialmente, en funcion de como estaba presente el filtro



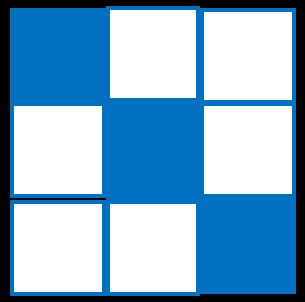
Ponemos 1 si esta el simbolo, -1 si no

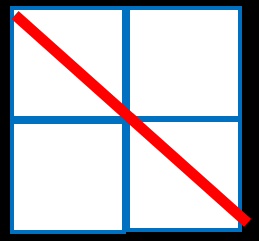


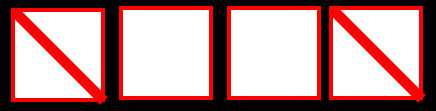


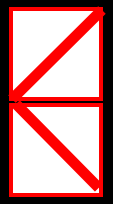


1.	-1.	-1.	1
-1.	1.	1.	-1

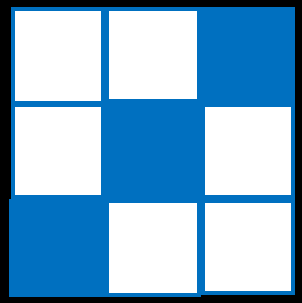


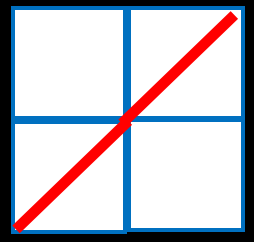


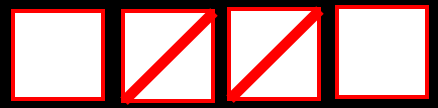


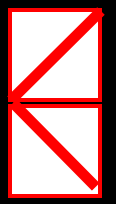


-1.	-1.	-1.	-1
1.	-1.	-1.	1

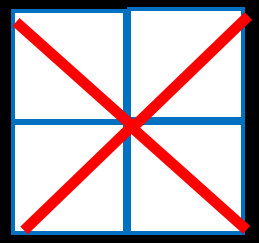


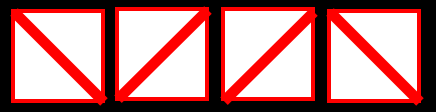


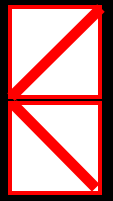




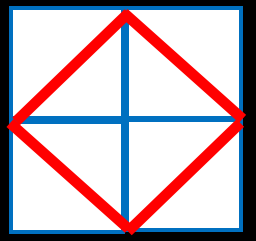
-1.	1.	1.	-1
-1.	-1.	-1.	-1

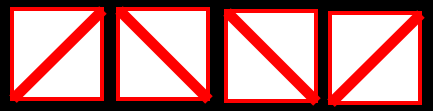


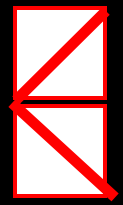




-1.	1.	1.	-1
1.	-1.	-1.	1

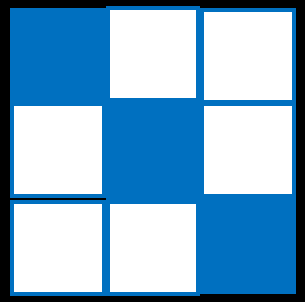


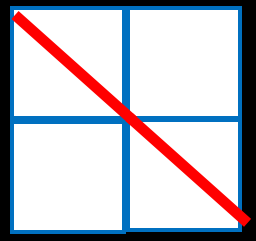


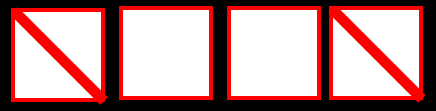


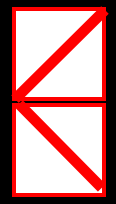
Filtro para la o

1.	-1.	-1.	1
-1.	1.	1.	-1



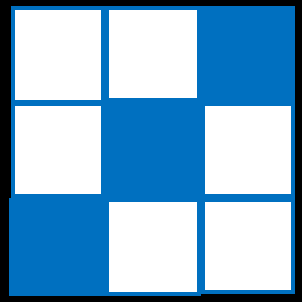


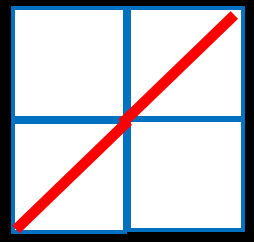


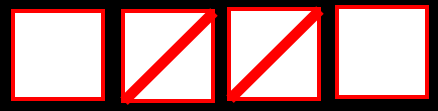


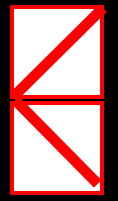
Filtro para la diagonal \

-1.	-1.	-1.	-1
1.	-1.	-1.	1



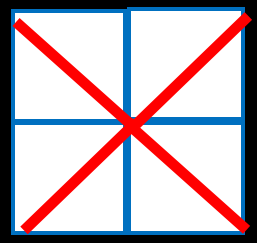


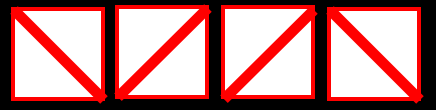


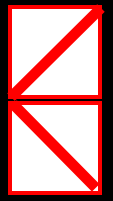


Filtro para la diagonal /

-1.	1.	1.	-1
-1.	-1.	-1.	-1

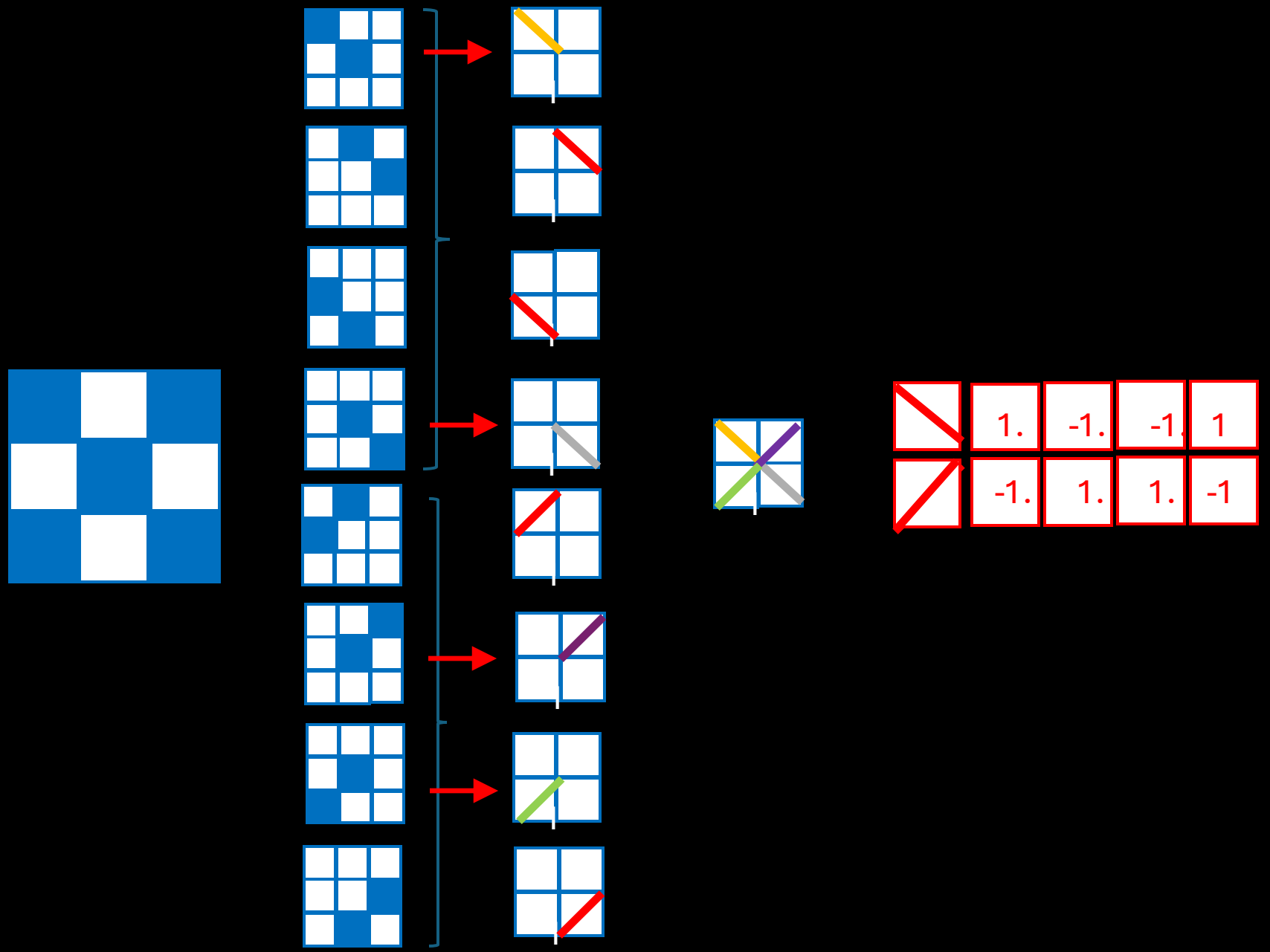


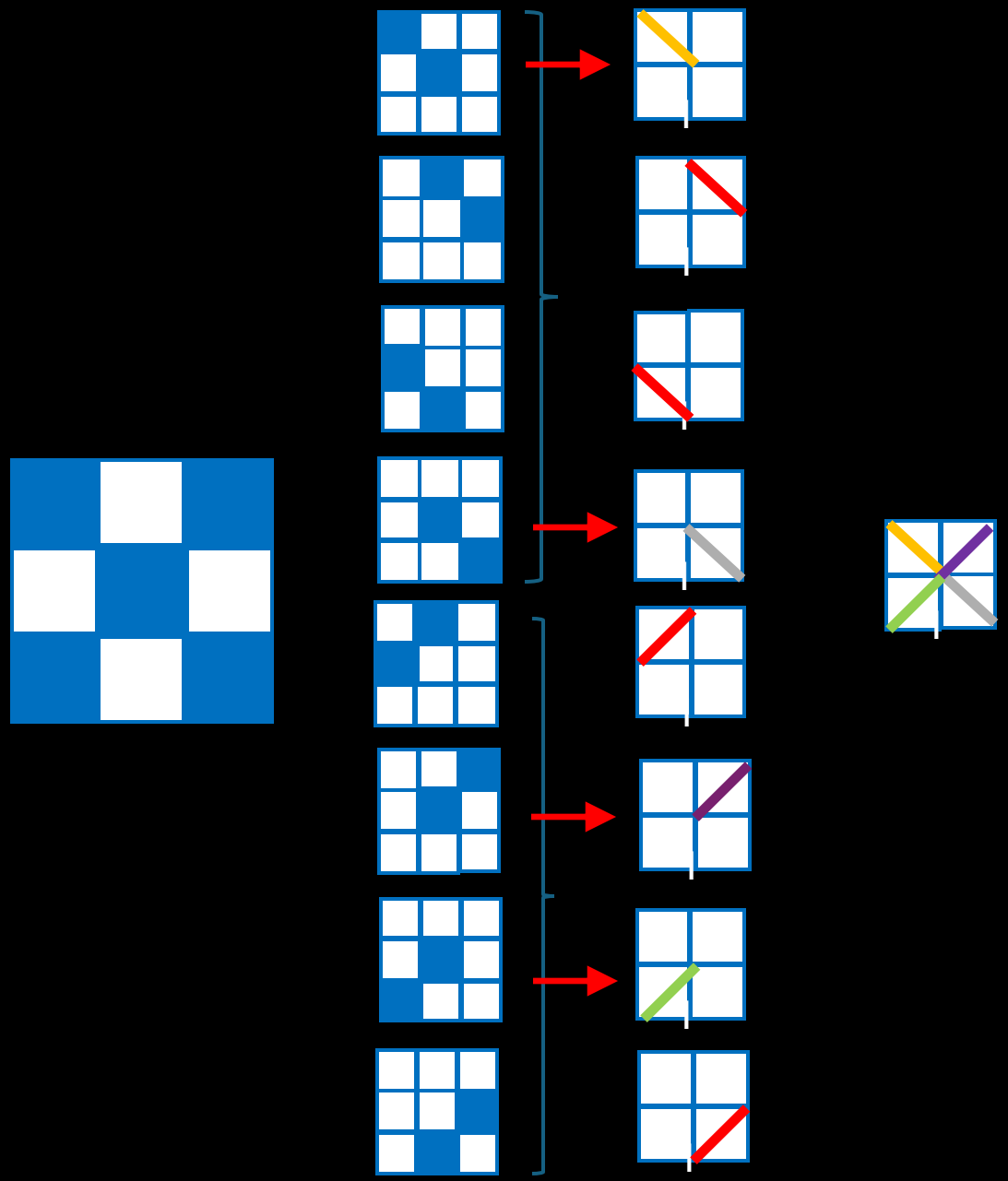




Filtro para la. X

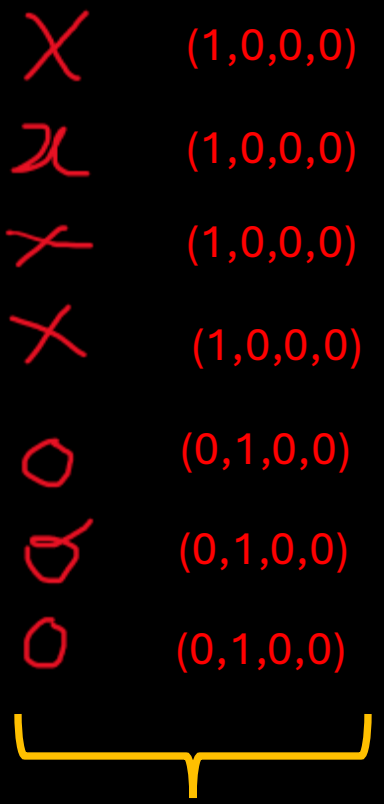
-1.	1.	1.	-1
1.	-1.	-1.	1


1. -1. -1. 1
-1. 1. 1. -1
x 8
-1. 1. 1. -1
1. -1. -1. 1
O -8
-1. -1. -1. 1
1. 1. 1. 1
/ 4
1. -1. -1. 1
-1. -1. -1. -1
\ 4



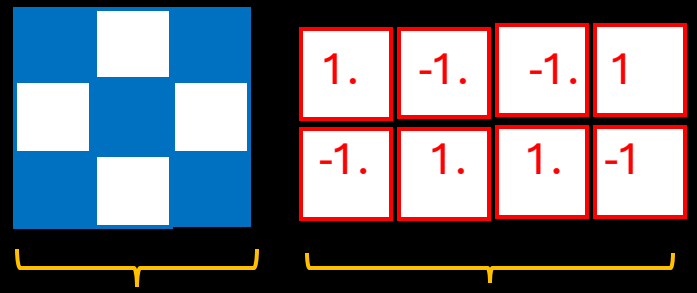
1. -1. -1. 1
-1. 1. 1. -1

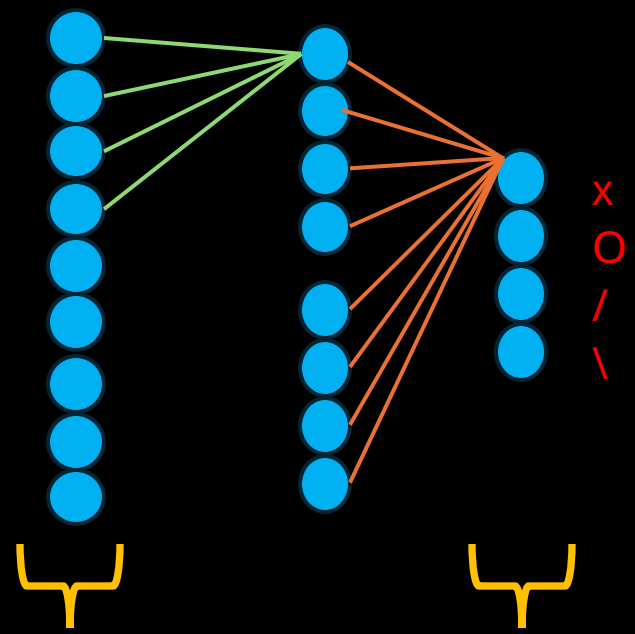
1. -1. -1. 1 -1. 1. 1. -1	x	8
-1. 1. 1. -1 1. -1. -1. 1	O	-8
-1. -1. -1. 1 1. 1. 1. 1	/	4
1. -1. -1. 1 -1. -1. -1. -1	\	4



Conjunto de entrenamiento

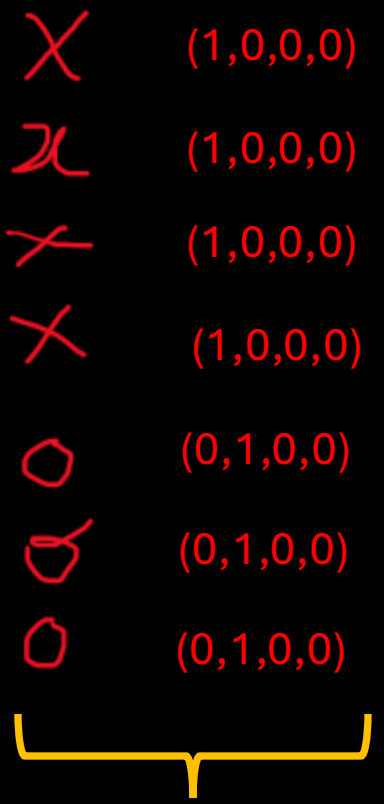
Representacions segun los filtros





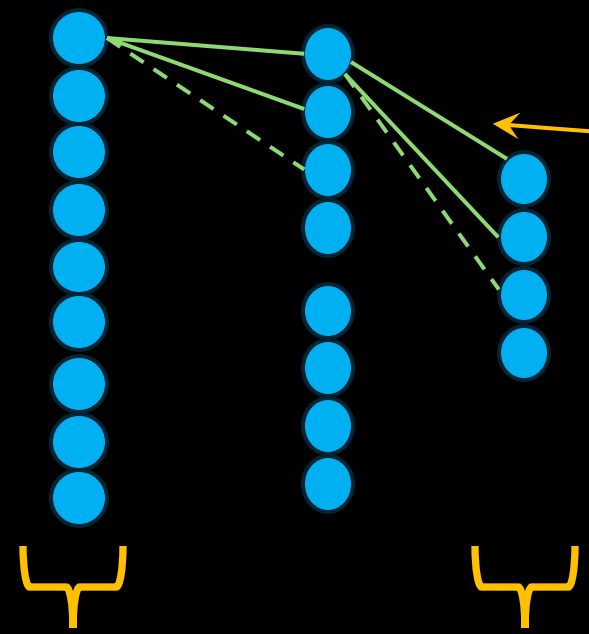
Cada unidad lleva el código de uno de los pixels de la imagen

Cada unidad lleva la probabilidad de que el input corresponda a uno de los simbolos



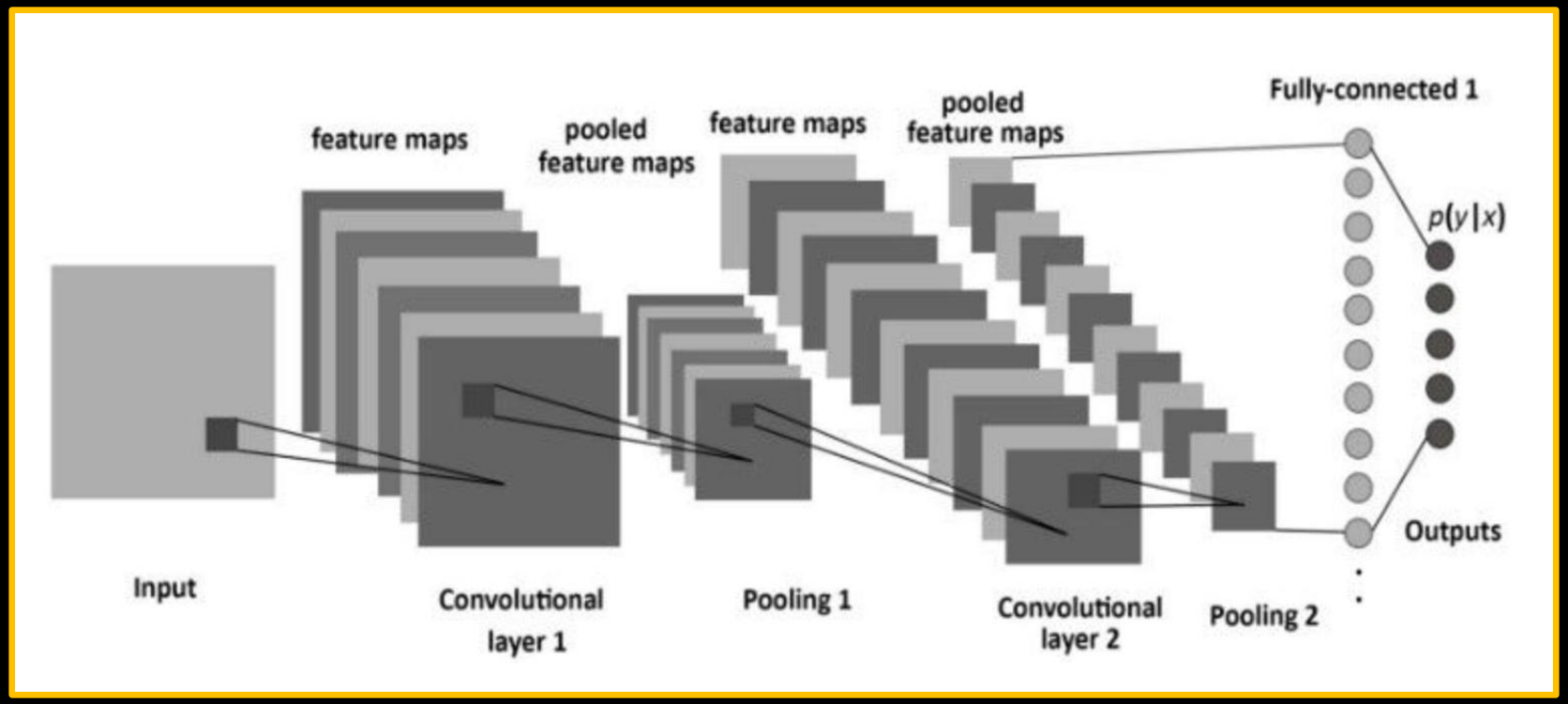
Conjunto de entrenamiento

Los pesos de las conexiones, Inicialmente tienen valores aleatorios. Y se van modificando de modo de poder reproducir las clasificaciones del conjunto de entrenamiento



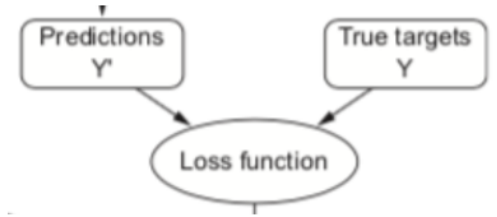
Cada unidad lleva el código de uno de los pixels de la imagen

Cada unidad lleva la probabilidad de que el input corresponda a uno de los simbolos



Lo que hemos analizado es una de las capas que se usan para procesar imagenes complejas

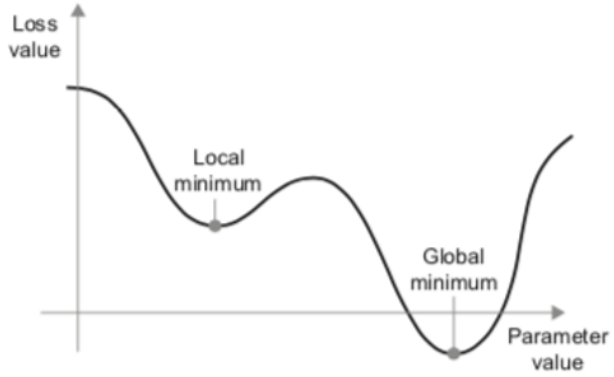
Y el esquema de ajuste (busqueda de parametros), es el mismo para todas estas arquitecturas.



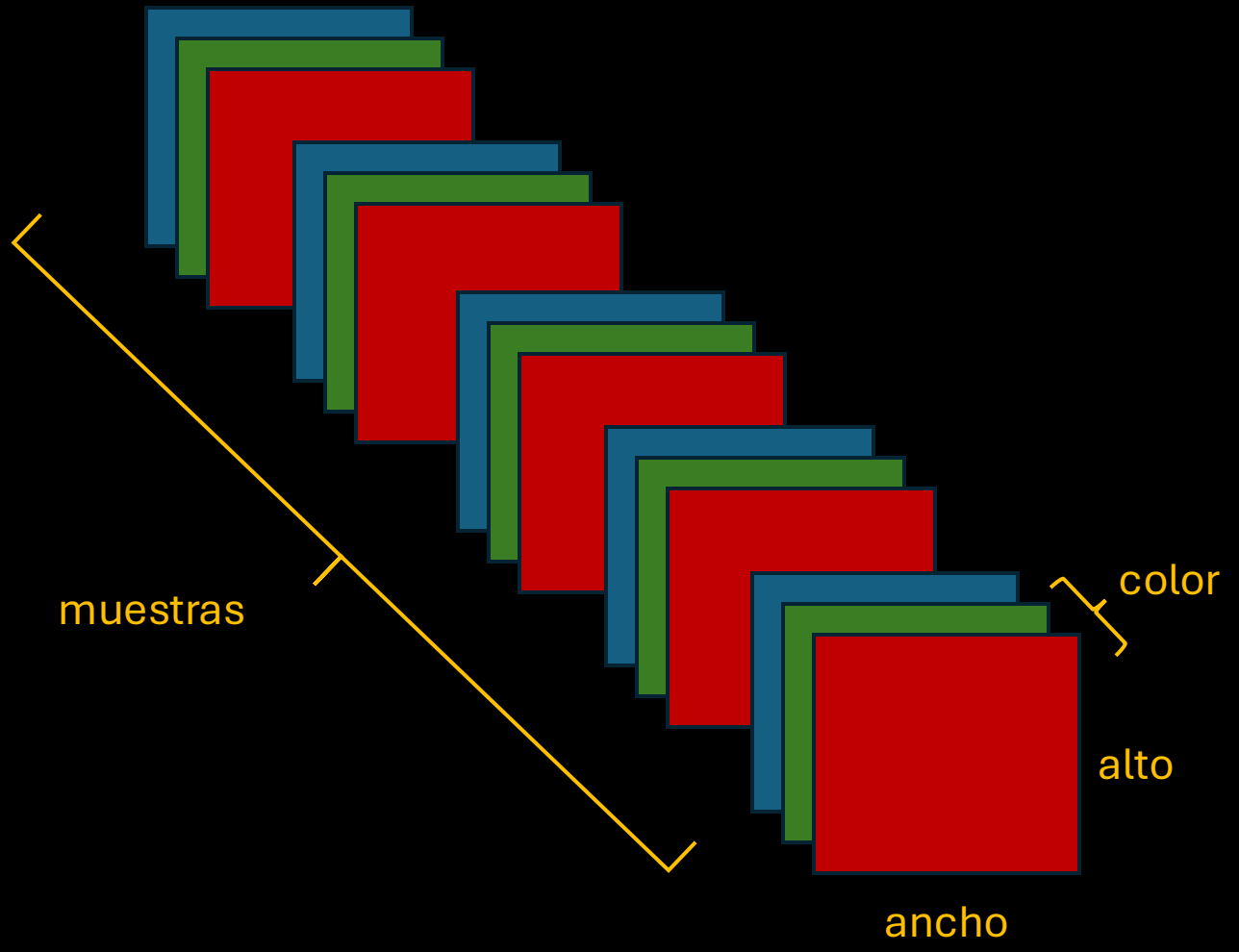
Y lo emplea para calcular los cambios en la red:

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}},$$

Con la idea de encontrar un mínimo en el cual $\Delta W_{ij} = 0$, resulta que



Formatos de los datos



Batch: conjunto de ejemplos de entrenamiento que se procesan juntos antes de actualizar los pesos del modelo

Epoch: Una época es cuando el modelo ha pasado por todo el conjunto de datos una vez.

Tipos de Procesamiento:

- Batch Gradient Descent: Usa todo el dataset para calcular el gradiente antes de actualizar los pesos (lento, pero más preciso).
- Stochastic Gradient Descent (SGD): Usa un solo ejemplo por iteración (rápido, pero más ruidoso).
- Mini-Batch Gradient Descent: Usa pequeños lotes de datos en cada iteración (balance entre precisión y velocidad).

🔷 Formato de los Datos en Redes Convolucionales

En **Keras** (y TensorFlow), las imágenes suelen representarse con la forma:

$$(\text{batch size}, \text{height}, \text{width}, \text{channels})$$

Donde:

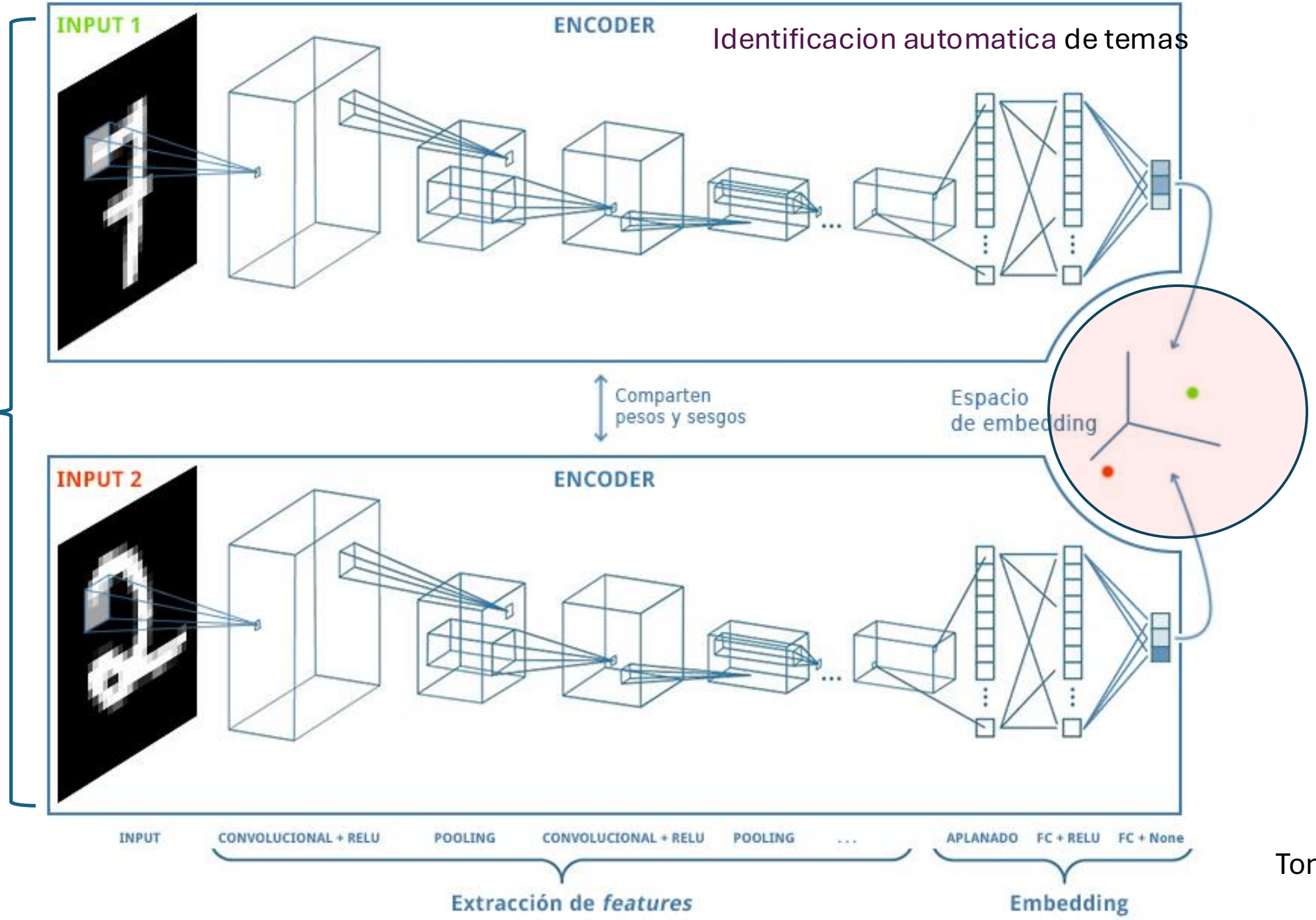
- **batch size**: número de imágenes procesadas simultáneamente en cada paso del entrenamiento.
- **height**: altura de la imagen en píxeles.
- **width**: ancho de la imagen en píxeles.
- **channels**: número de canales de color (1 para blanco y negro, 3 para RGB, etc.).

🔷 ¿Dónde se define el batch size?

El **batch size NO se especifica en la arquitectura de la red**, sino en la función que entrena el modelo, como `model.fit()`, `model.predict()` o `model.evaluate()`.

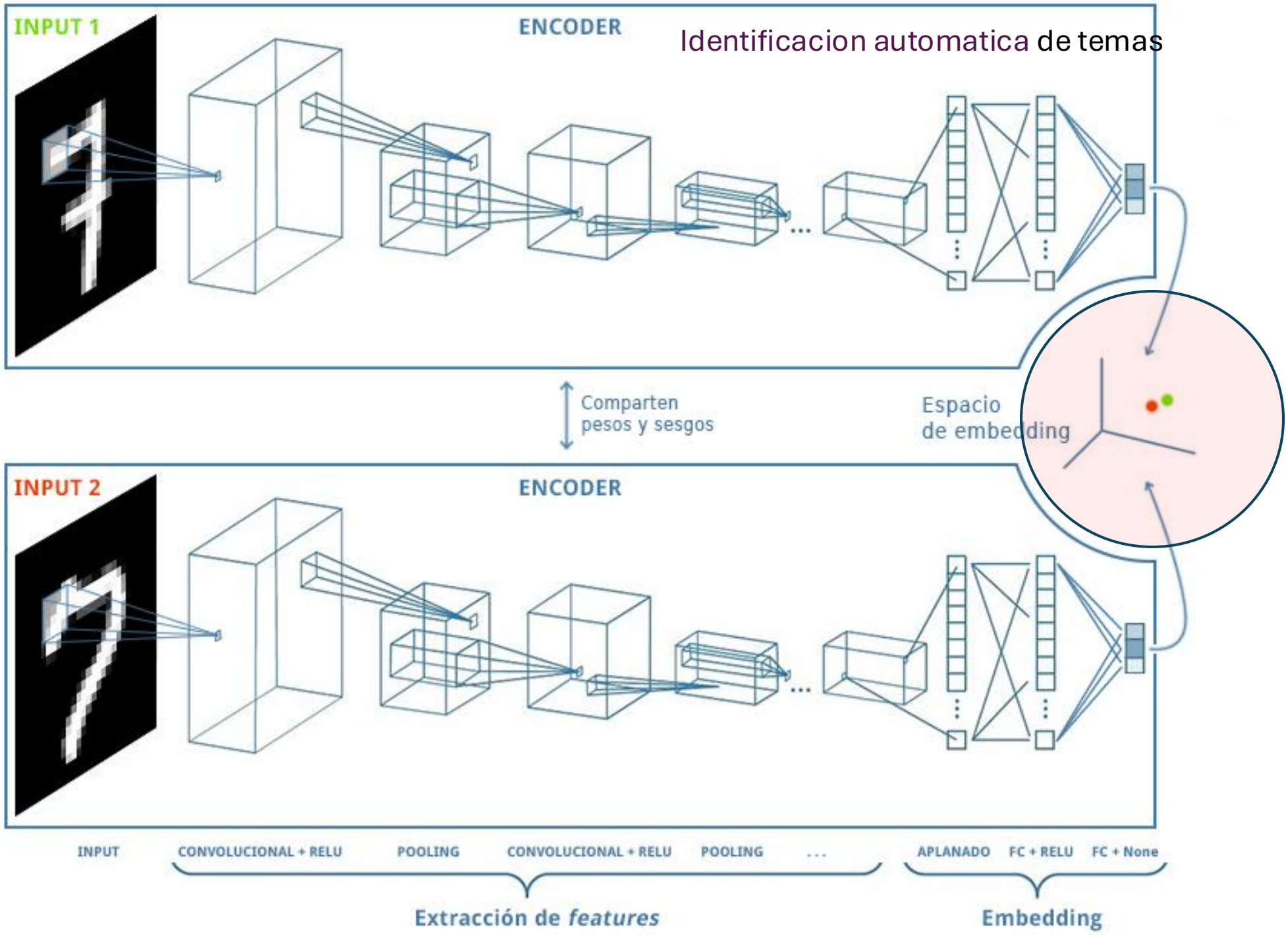
Identificacion automatica de temas

Siamese Networks define a metric



Tomas Bossi

Identificacion automatica de temas



Como la pandemia nos dejo varados cerca del **Parque Pereyra Iraola (reserva de biodiversidad, NU)**

Zonotrichia capensis (oscino, pero que aprende tipicamente un solo canto)

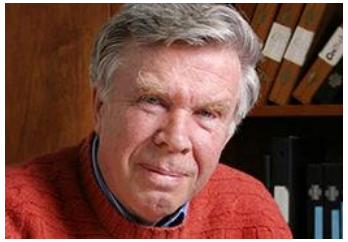
Alex Martinez

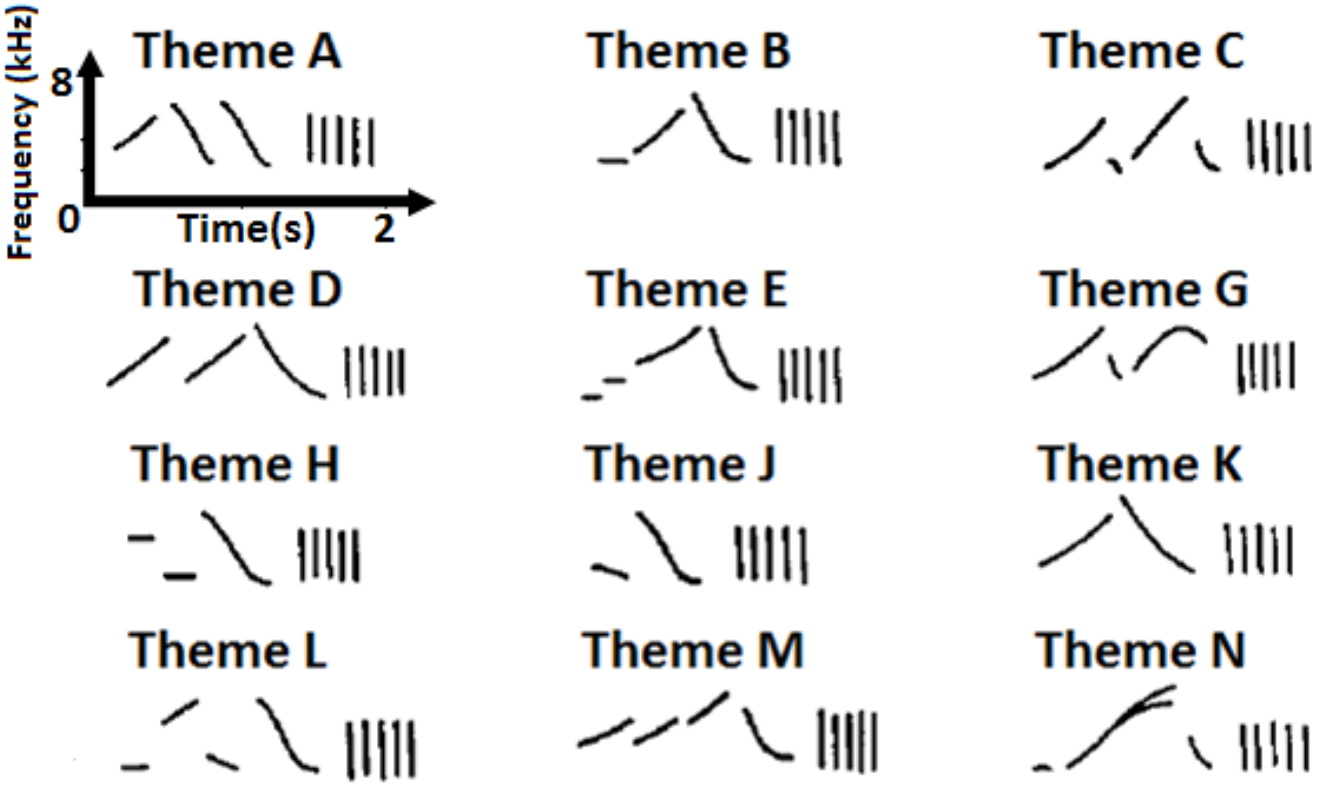

2022
Cno. Gral. Manuel Belgrano
Arroyo San Juan
Cno. Gral. Manuel Belg
C. 403
Con Roberto Bistel
Y Alex Martinez
Google
El Rocio De LAURENT
Imágenes © 2022 CNES / Airbus, Maxar Technologies
100 m


2022
tema trino
Cno. Gral. Manuel Belgrano
Arroyo San Juan
Cno. Gral. Manuel Belg
C. 403
Con Roberto Bistel
Y Alex Martinez
Google
El Rocio De LAURENT
Imágenes © 2022 CNES / Airbus, Maxar Technologies
100 m

Not the first one to get stuck close to the parque Pereyra

Sketched handwritten notes by Nottebohm, in **1966**



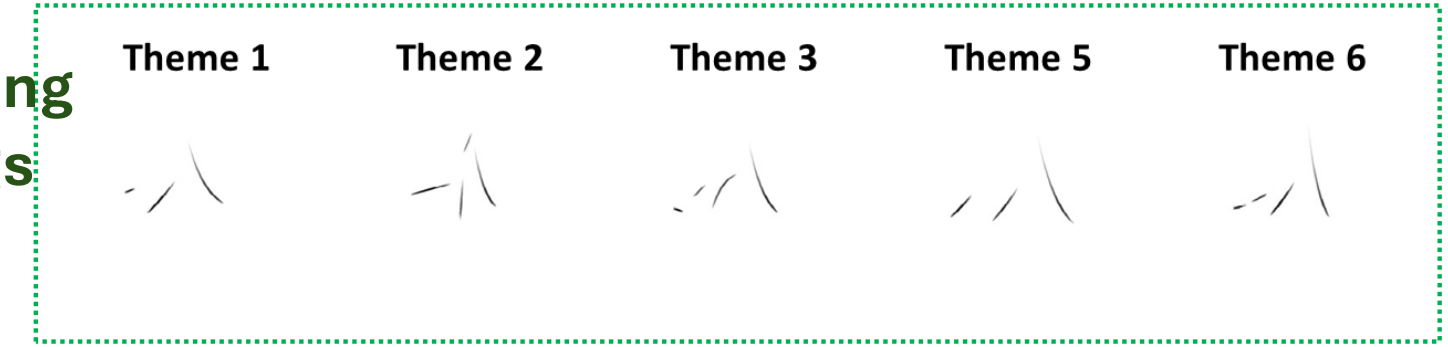


Is it what we find today?

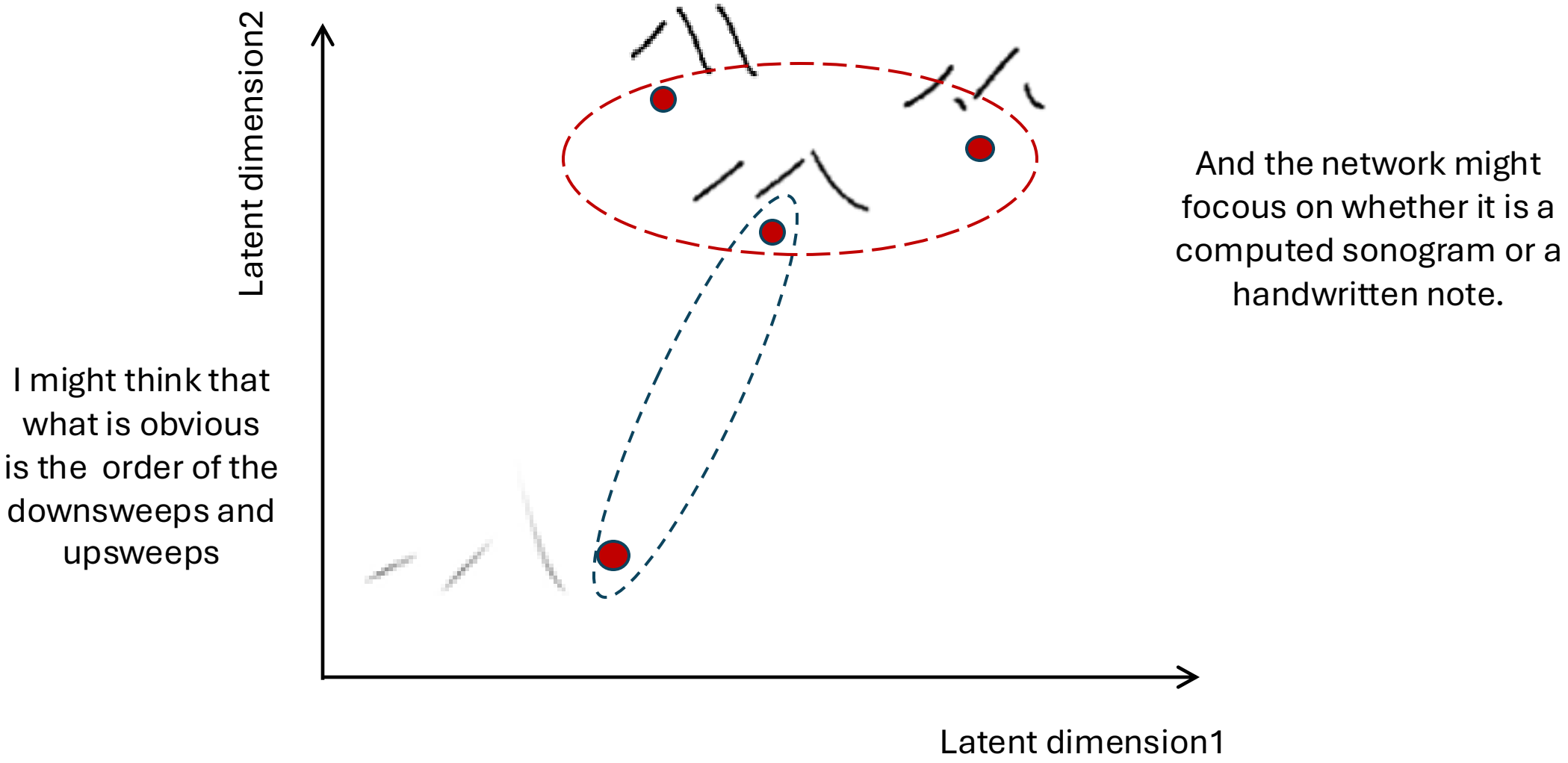
Using dynamics… to generate surrogate synthetic songs and train a network

Synthetic songs that copy what was found in 2020

Synthetic songs bringing back to “life” the songs from the sixties



Why did I re synthesized the songs from the sixties?

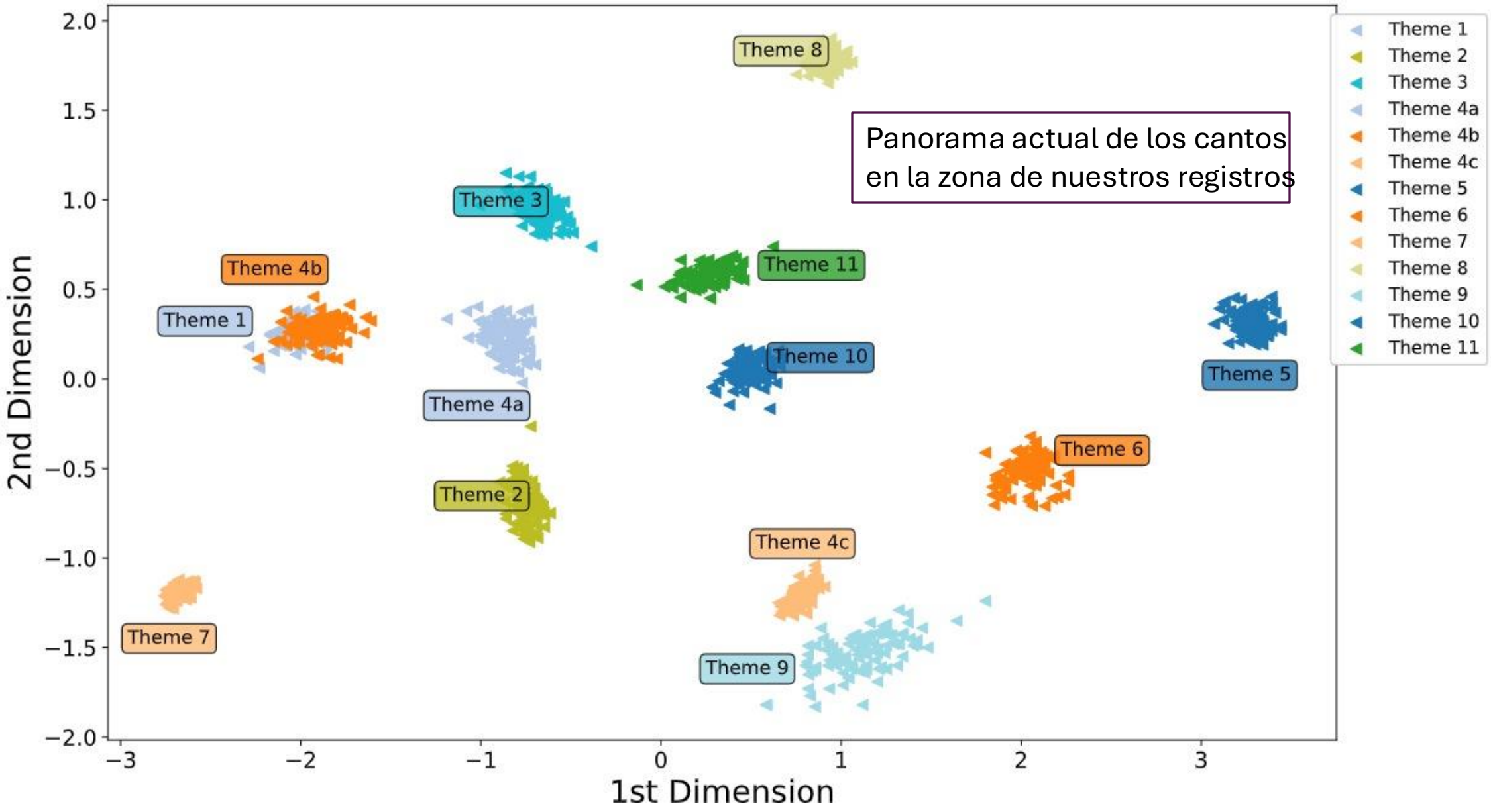


I might think that what is obvious is the order of the downsweeps and upsweeps

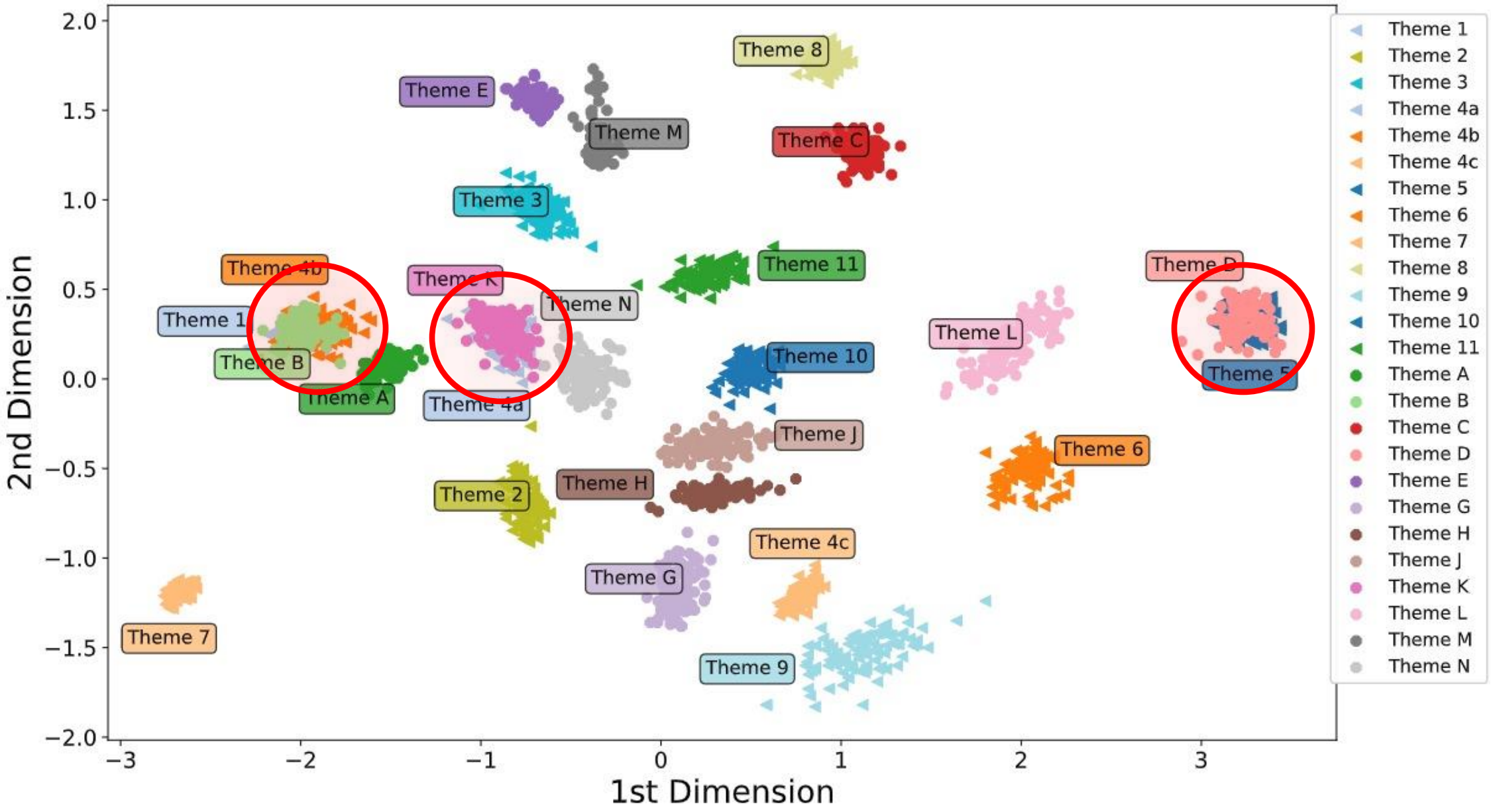
And the network might focous on whether it is a computed sonogram or a handwritten note.

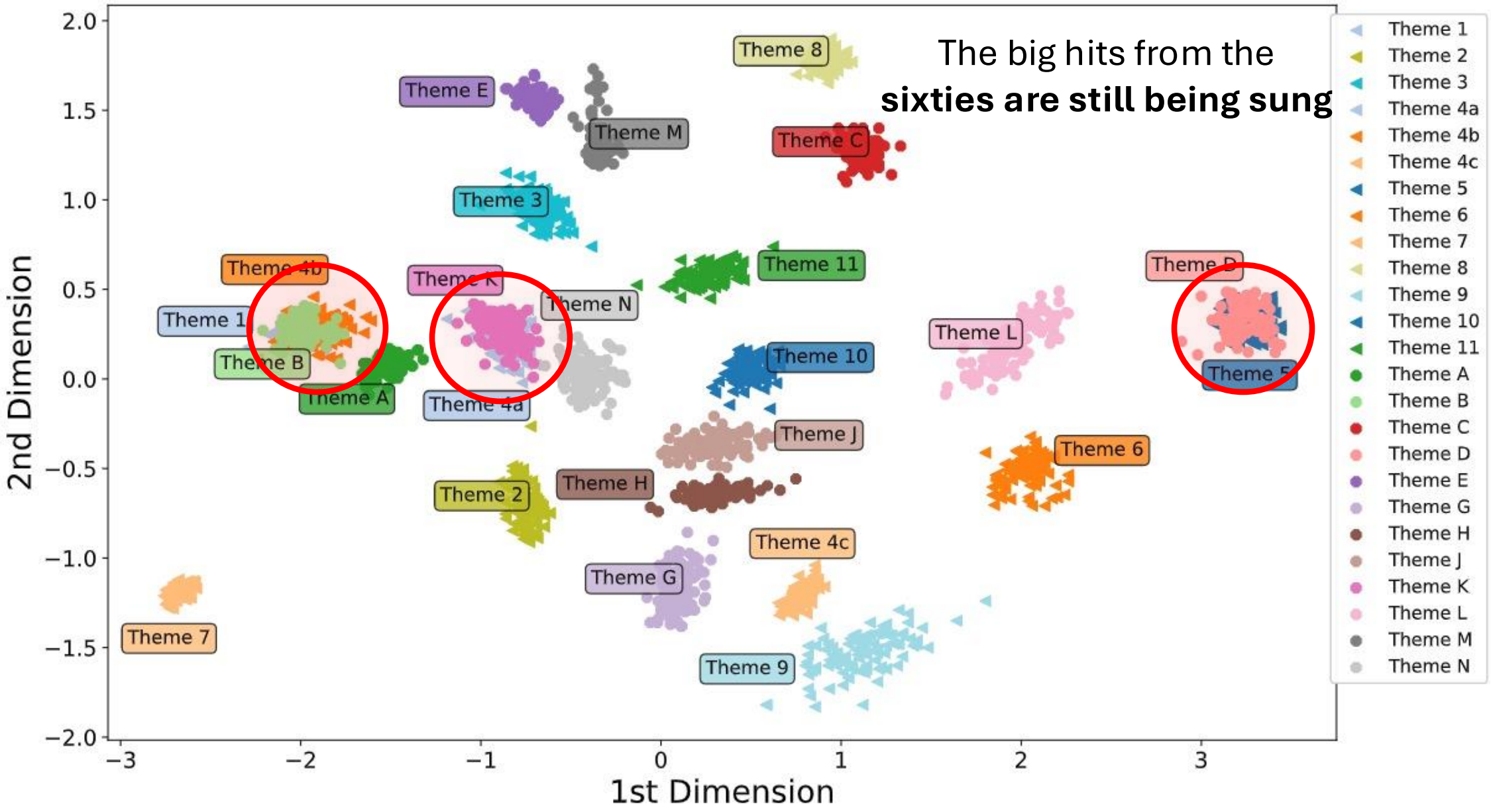
Data preparation, data preparation, data preparation

So we trained a network to distinguish the synthetic songs that copy the 2020 songs



And processed with it the songs from 1966, re synthesized...




The big hits from the
sixties are still being sung
2nd Dimension
1st Dimension
Theme 1
Theme 2
Theme 3
Theme 4a
Theme 4b
Theme 4c
Theme 5
Theme 6
Theme 7
Theme 8
Theme 9
Theme 10
Theme 11
Theme A
Theme B
Theme C
Theme D
Theme E
Theme G
Theme H
Theme J
Theme K
Theme L
Theme M
Theme N