SESIÓN 02

Deep Learning

Redes neuronales convolucionales

Dr. León Felipe Palafox Novack

lfpalafox@up.edu.mx

01 Invarianza

02 Aumento de datos

03 Características locales

04 Estacionaridad

05 Convolución

06 Pooling

07 CIFAR-10 e ImageNet

02

Redes neuronales convolutivas

02

Todo es incertidumbre

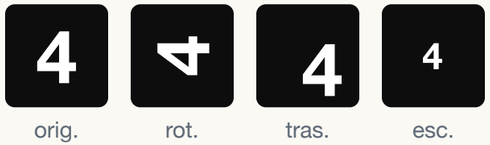
03

Redes neuronales convolucionales

Google aws

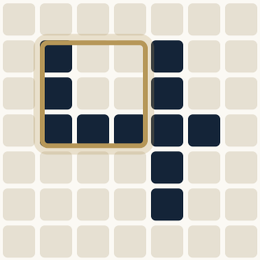
Estado del arte

en detección de patrones: las usan Google y Amazon.



Transformaciones

Toleran la traslación por diseño; la escala y la rotación, en parte.



Características locales

Aprenden rasgos locales en lugar de características generales.

04

Transformaciones de imagen



Original **Rotación** **Traslación** **Escala**

Para nosotros es el mismo 4. Para la computadora, píxel por píxel, son cuatro imágenes muy distintas.

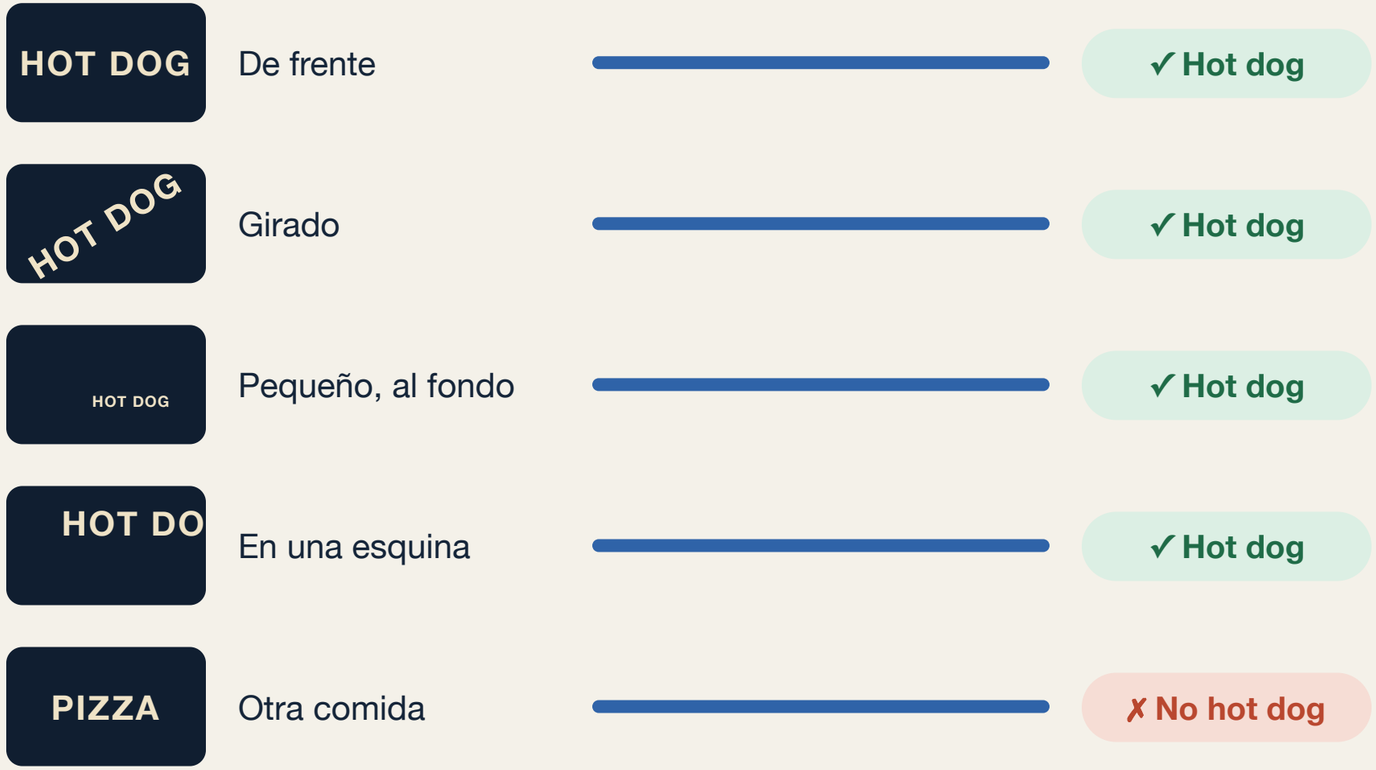
05

¿Hot dog o no hot dog?

Un clasificador binario.

Como la app «Not Hotdog» de la serie *Silicon Valley* (HBO, 2017): solo responde **hot dog** o **no hot dog**.

El reto: reconocerlo aunque esté girado, lejos o en otra posición.



06

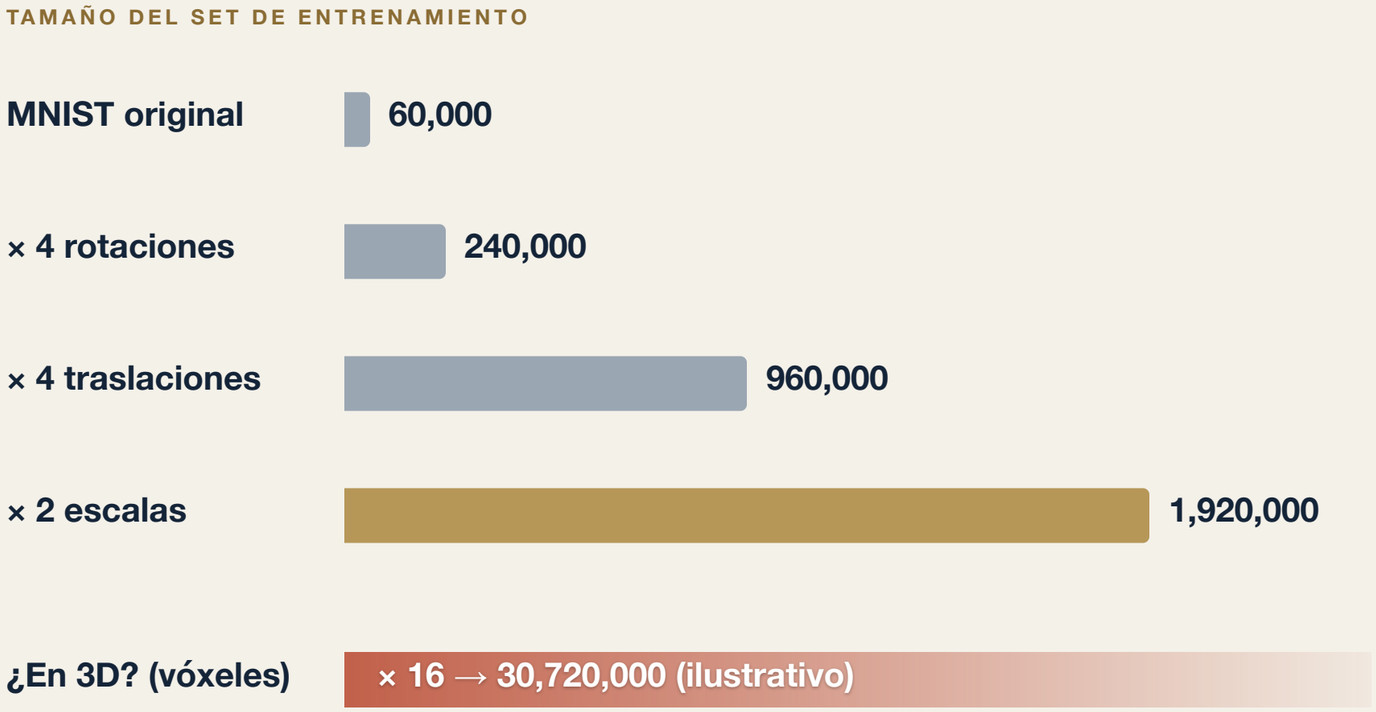
Transformaciones en imágenes

Sin CNN: transformar las imágenes y crear un set de datos artificialmente grande.

Muchos algoritmos solían hacer eso.

Y eso con solo 4 grados de libertad. ¿Qué pasaría con vóxeles?

Hay que utilizar un modelo que aprenda las transformaciones.



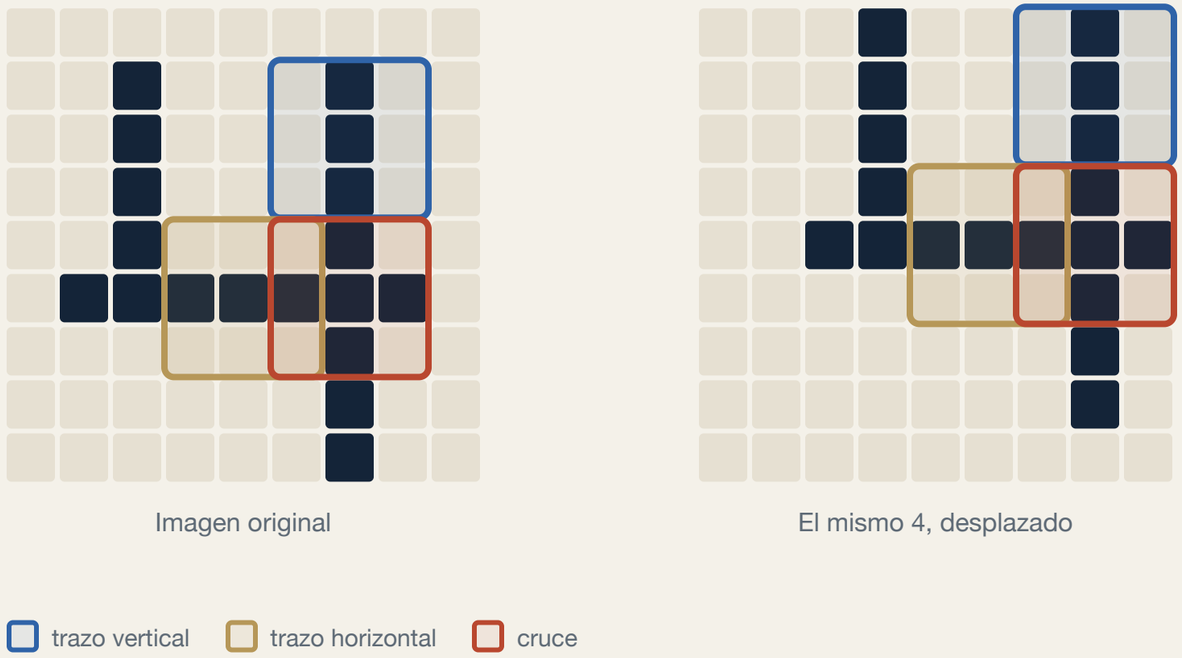
07

Características locales y estacionaridad

Si no describimos la imagen completa, sino sus características…

…podemos modelar las transformaciones dentro del algoritmo.

Pedazos de la imagen tienen las mismas características.



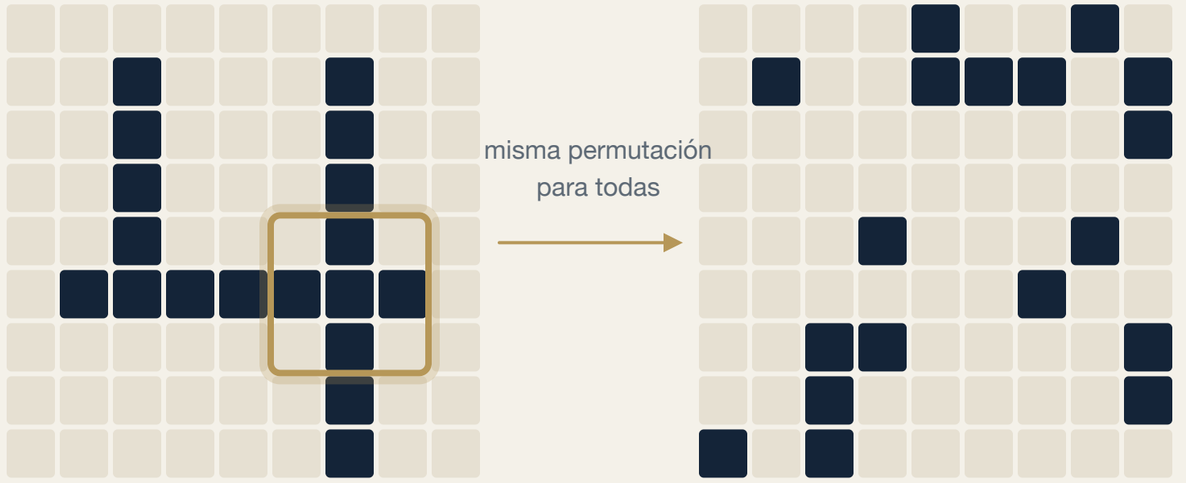
08

Clasificadores y características locales

A la mayoría de los clasificadores no les importa el orden de los píxeles.

Las redes neuronales tradicionales aprenden características entre datos, no entre píxeles.

No es la única solución, pero es de lo mejor que tenemos «out of the box».

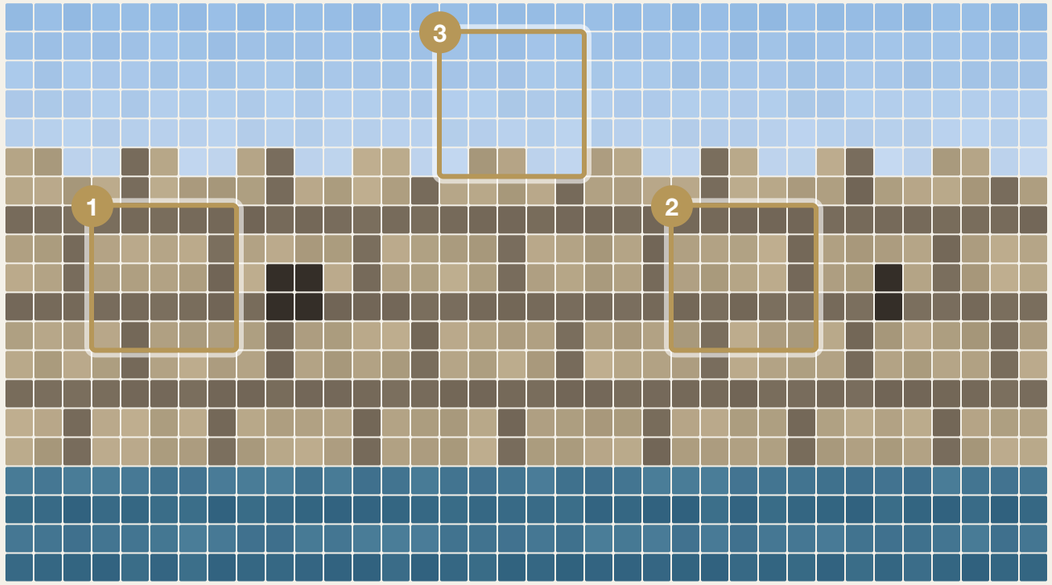


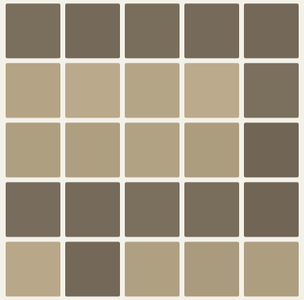
Una CNN sí mira la vecindad de cada píxel

Para una red densa (MLP), ambos problemas son equivalentes

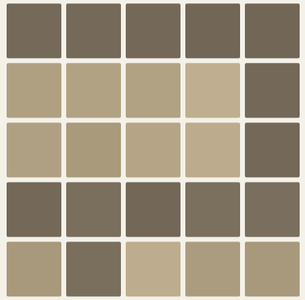
09

Imágenes

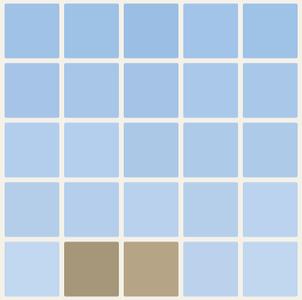




1 **Muro (izquierda)**



2 **Muro (derecha)**



3 **Cielo**

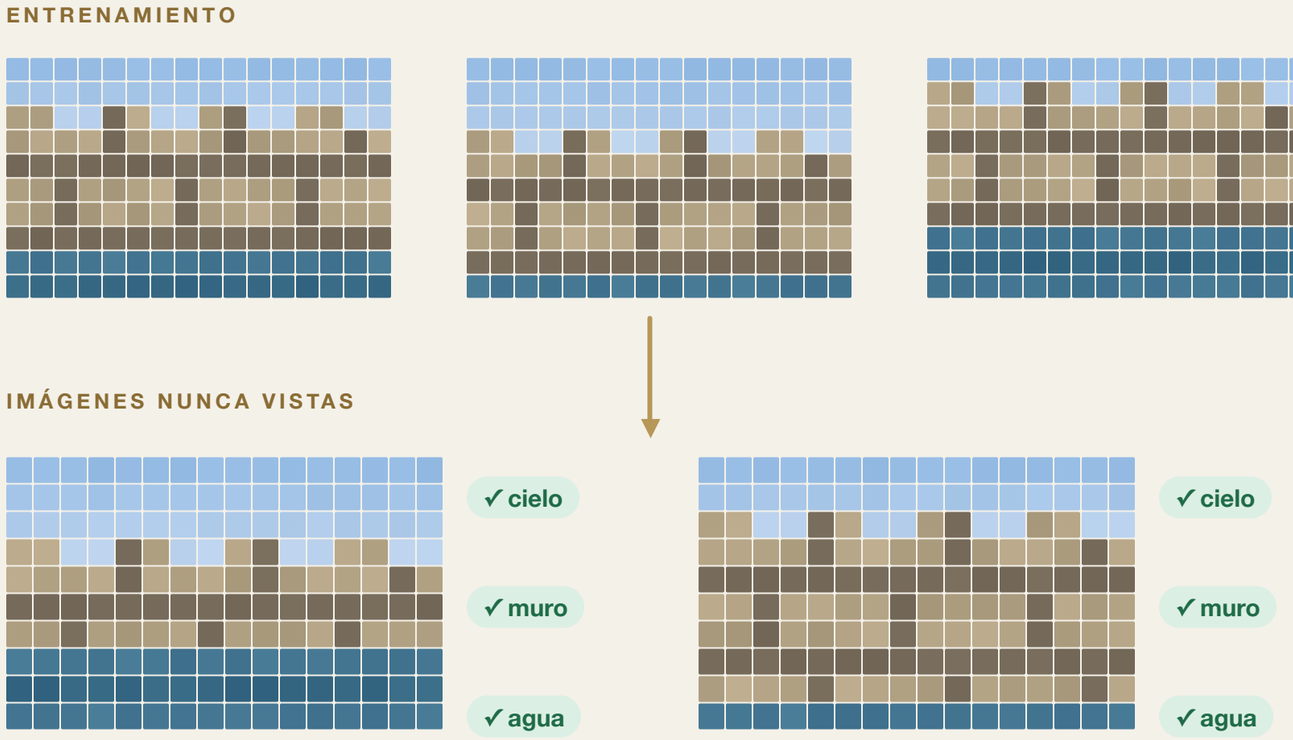
Los parches 1 y 2 están lejos, pero se parecen: tienen la misma estadística local. Eso es la **estacionaridad**.

10

Imágenes y estacionaridad

Pequeñas partes de la imagen tienen las mismas características que partes grandes: paredes, agua, cielo.

Si todas las imágenes tienen la misma naturaleza (RGB, paisajes), podemos entrenar con unas y detectar en otras, nunca vistas.



11

Convolución

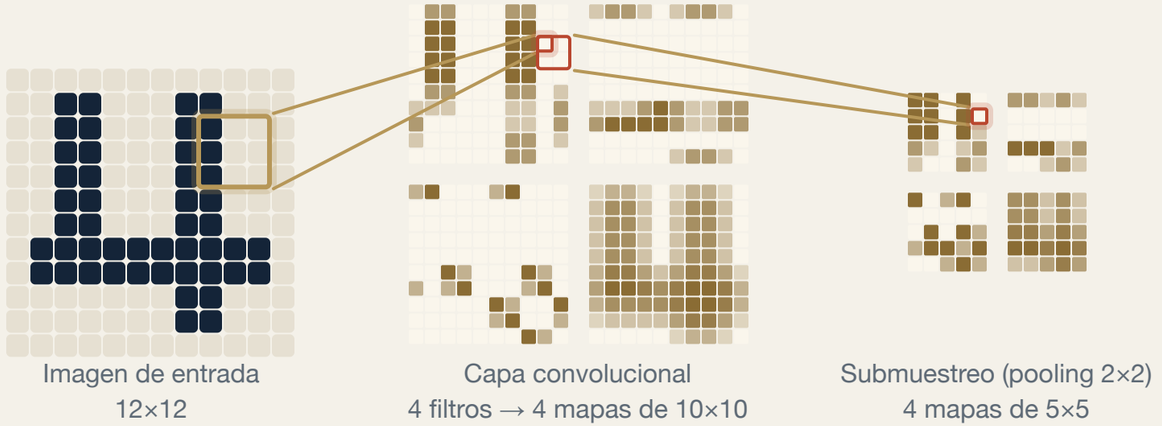
Un mismo filtro recorre toda la imagen

12

Redes neuronales convolucionales

El entrenamiento aprovecha la invarianza de las imágenes.

La red aprende características que no varían, en lugar de características generales.



Cada unidad mira solo una región pequeña (su **campo receptivo**) y todas las de un mapa comparten los mismos pesos.

13

Convolución

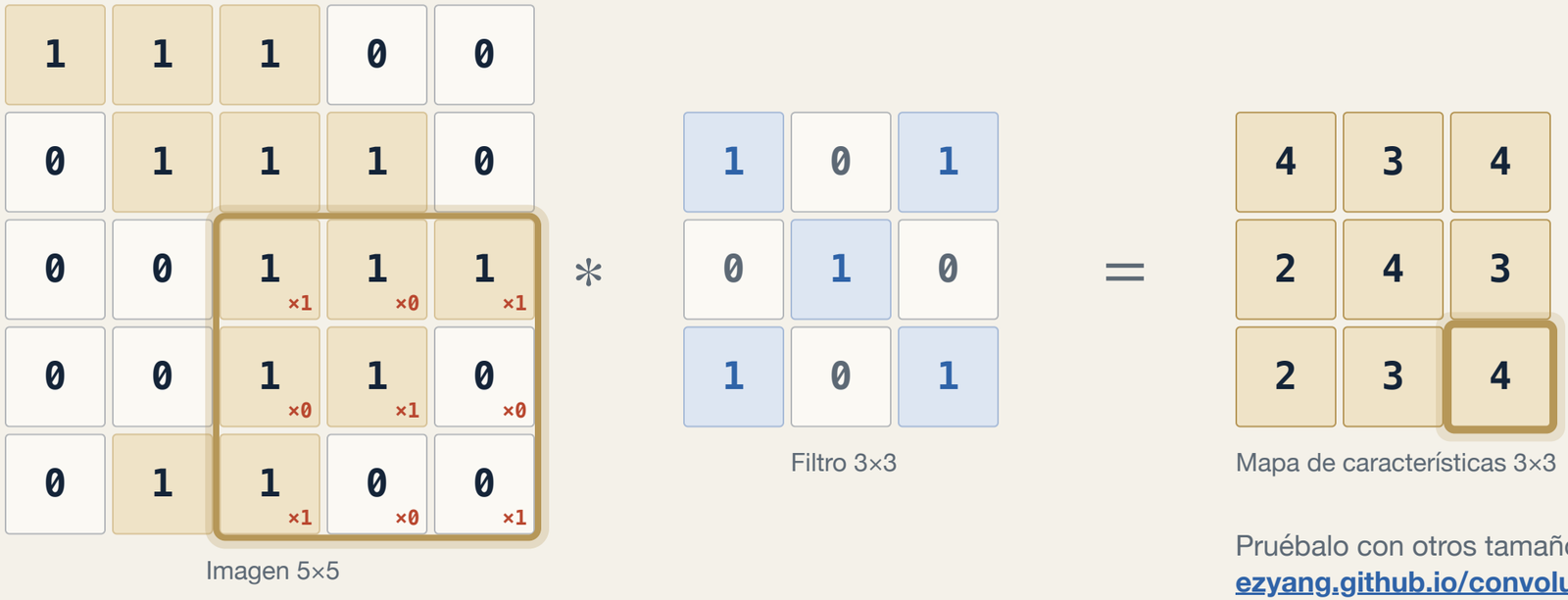


Imagen 5×5

Filtro 3×3

Mapa de características 3×3

$$1\times1 + 1\times0 + 1\times1 + 1\times0 + 1\times1 + 0\times0 + 1\times1 + 0\times0 + 0\times1 = 4$$

Pruébalo con otros tamaños, padding y stride:
ezyang.github.io/convolution-visualizer

14

La convolución como ecuación

$$S(i,j) = (I * K)(i,j) = \sum_{m=0}^{2}\sum_{n=0}^{2} I(i+m, j+n)\cdot K(m,n)$$

i, j: fila y columna donde está la ventana · m, n: fila y columna dentro del filtro

	0	1	2	3	4	5
0	9	8	9	1	0	1
1	8	9	8	0	1	0
2	9	9	9	1	0	0
3	8	9	8	0	0	1
4	9	8	9	1	1	0
5	9	9	8	0	1	0

Imagen I · 6×6 · claro → oscuro

$*$

	0	1	2
0	1	0	−1
1	1	0	−1
2	1	0	−1

Filtro K · bordes verticales

$=$

	0	1	2	3
0	0	24	25	1
1	0	26	24	0
2	0	24	25	1
3	1	25	23	0

Mapa S · 4×4

(6 – 3) / 1 + 1 = 4 posiciones por lado

Ventana en $(i, j) = (3, 3)$

$$S(3,3) = \Sigma_m \Sigma_n\, I(3+m, 3+n)\cdot K(m,n)$$

I(3,3)·K(0,0) 0 · 1	I(3,4)·K(0,1) 0 · 0	I(3,5)·K(0,2) 1 · (−1)
I(4,3)·K(1,0) 1 · 1	I(4,4)·K(1,1) 1 · 0	I(4,5)·K(1,2) 0 · (−1)
I(5,3)·K(2,0) 0 · 1	I(5,4)·K(2,1) 1 · 0	I(5,5)·K(2,2) 0 · (−1)

$S(3, 3) = 0 + 0 - 1 + 1 + 0 + 0 + 0 + 0 + 0 = \mathbf{0}$

En sentido estricto esto es **correlación cruzada**: la convolución matemática gira el filtro 180°. PyTorch y TensorFlow implementan esta versión y la llaman convolución.

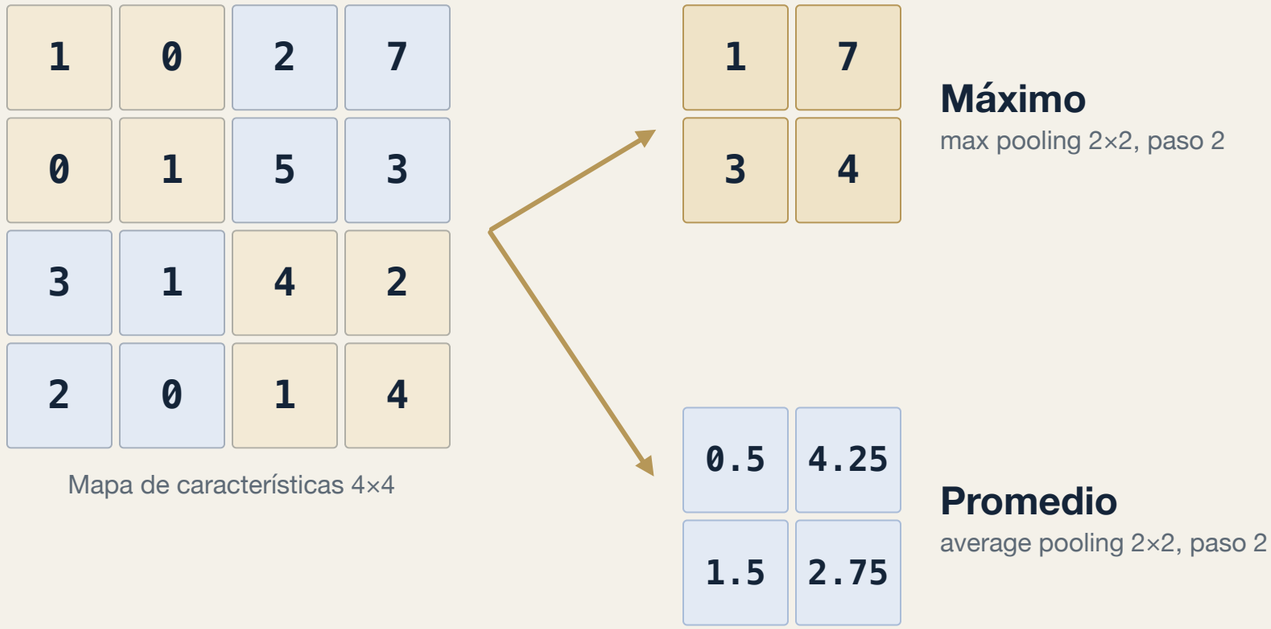
15

Pooling

Una vez que tenemos las convoluciones, hay que aprovechar la localidad.

Tomamos una región local y nos quedamos con el máximo (max pooling) o el promedio (average pooling).

Reducimos el número de características y ganamos robustez ante pequeños cambios de posición.

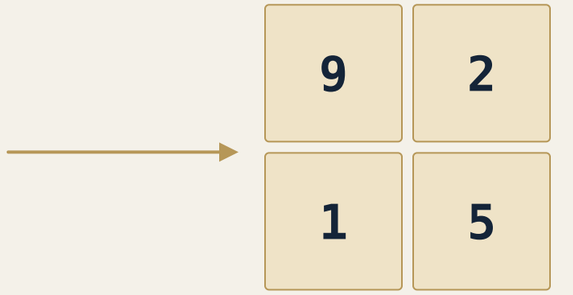


16

Pooling: menos datos, más robustez

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

Mapa de características 8×8 (salida de la convolución)



Pooling máximo 4×4 → 2×2

Desplazamos la entrada 1 píxel a la derecha →

✓ El pooling da exactamente la misma salida: robustez ante pequeñas traslaciones.

17

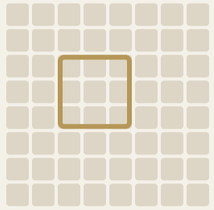
Análisis

Entrenar con una CNN es similar a entrenar con un set de datos artificialmente grande.

El pooling nos ayuda a reducir la dimensionalidad.

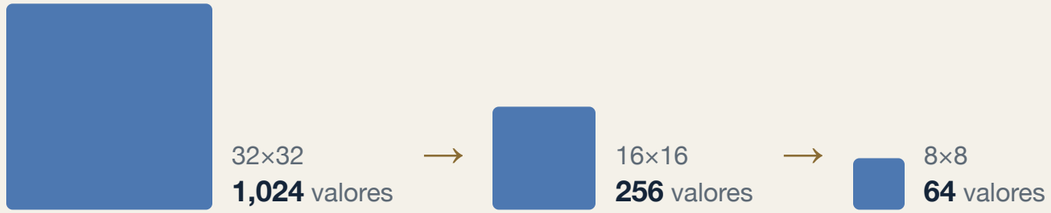
La red comparte pesos para aprovechar los píxeles que están cercanos unos de otros.

PESOS COMPARTIDOS

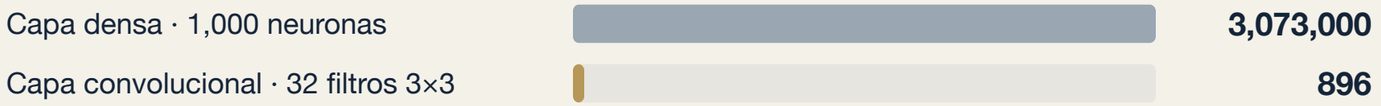


Un filtro 3×3 sobre una imagen de 32×32 se aplica en **900** posiciones con los **mismos 9 pesos**.

POOLING 2×2



PARÁMETROS PARA UNA IMAGEN DE 32×32×3



≈ 3,400 veces menos parámetros

18

Sets de datos: CIFAR-10

CIFAR-10

60,000

imágenes a color de 32×32 píxeles

10

clases · 6,000 imágenes por clase



50,000 entrenamiento — 10,000 prueba

University of Toronto

Alex Krizhevsky, Vinod Nair y Geoffrey Hinton · uno de los benchmarks clásicos de deep learning

avión airplane

automóvil automobile

pájaro bird

gato cat

venado deer

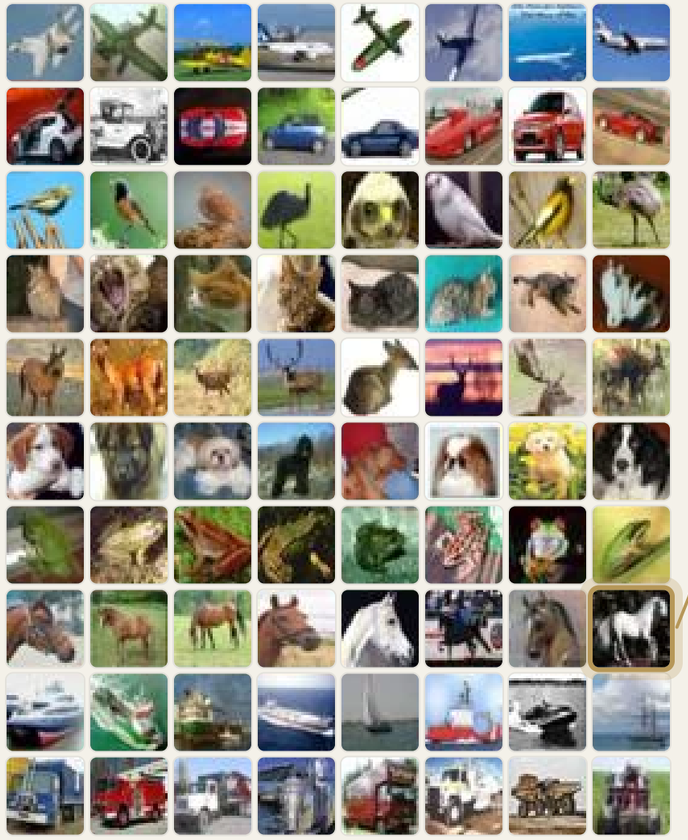
perro dog

rana frog

caballo horse

barco ship

camión truck







tamaño real (32 × 32)

32 × 32 × 3

= **3,072** valores

por imagen: 32×32 píxeles × 3 canales de color (R, G, B)

19

Sets de datos: ImageNet

ImageNet

14,197,122
imágenes etiquetadas a mano

21,841
categorías, organizadas con WordNet

ILSVRC · 1,000 CLASES



1,281,167 entrenamiento — 50,000 val. · 100,000 prueba

A la red entran recortes de **224 × 224 × 3 = 150,528** valores: 49 veces una imagen de CIFAR-10.

Deng, Dong, Socher, Li, Li y Fei-Fei · CVPR 2009 · etiquetado con Amazon Mechanical Turk



jilguero
goldfinch · n01531178



cebra
zebra · n02391049



panda gigante
giant panda · n02510455



estrella de mar
starfish · n02317335



catamarán
catamaran · n02981792



auto deportivo
sports car · n04285008



fresa
strawberry · n07745940



pizza
pizza · n07873807

20

ImageNet (ILSVRC): llegan las CNN

Reto anual de 2010 a 2017: 1,000 clases y 1.28 millones de imágenes de entrenamiento. Se mide el **error top-5**: la clase correcta no está entre las 5 respuestas del modelo.

2012: **AlexNet**, una CNN entrenada en GPUs, gana con 16.4% (15.3% con datos extra) contra 26.2% del segundo lugar.

Desde entonces todos los ganadores son CNN; en 2015 ResNet baja del ≈5.1% estimado para una persona entrenada.

