Redes convolucionales

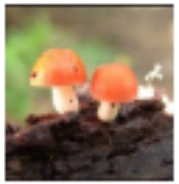
Aprendizaje Automático

Docente: Juan David Martínez Vargas
jdmartinev@eafit.edu.co

Agenda

- Redes Neuronales Convolucionales (CNNs)
- Por qué no utilizar feed forward NNs?
- Convolución
- Pooling
- Stride
- Ejemplos

Clasificación de imágenes

Image	Category
	mushroom
	cherry
...	...

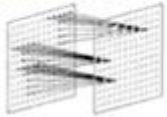
Feed-forward NNs

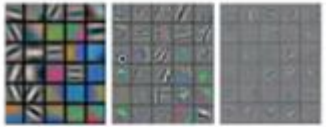
input

layer 1

Propiedades de señales naturales

- Main assumption :
  - Data (images, videos, speech) is compositional, it is formed of patterns that are:
    - Local (Hubel-Wiesel 1962)
    - Stationary (shared patterns)
    - Hierarchical (multi-scale)





- ConvNets leverage the compositionality structure :
  - They extract compositional features and feed them to classifier, recommender, etc (end-to-end systems).

Computer Vision

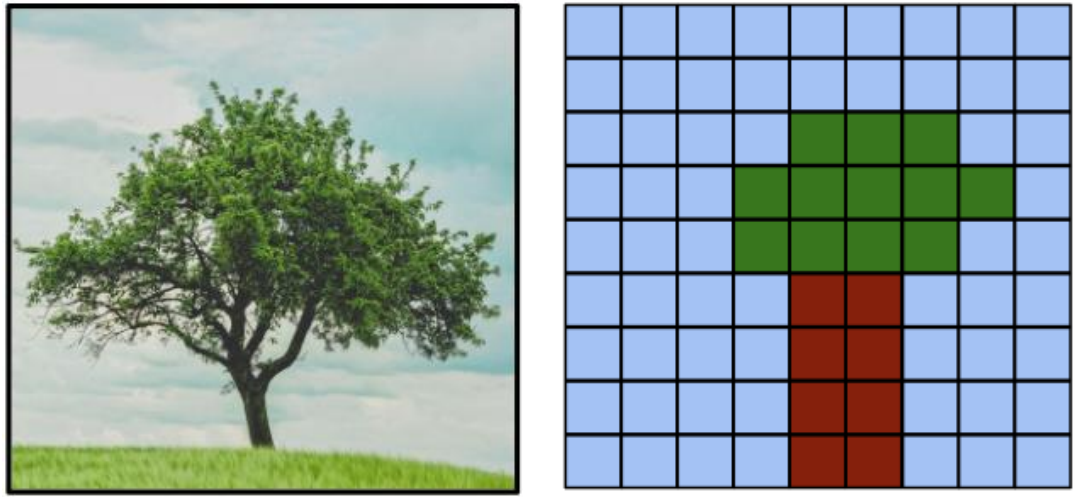
NLP

Speech

Game of Go

https://www.youtube.com/watch?v=Iiv9R6BjxHM&list=PLLHTzKZzVU9eaEyErdV26ikyolxOsz6mq&index=24

Propiedades de señales naturales



A digital image is a 2D **grid of pixels.**

https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L3%20-%20UUCLxDeepMind%20DL2020.pdf

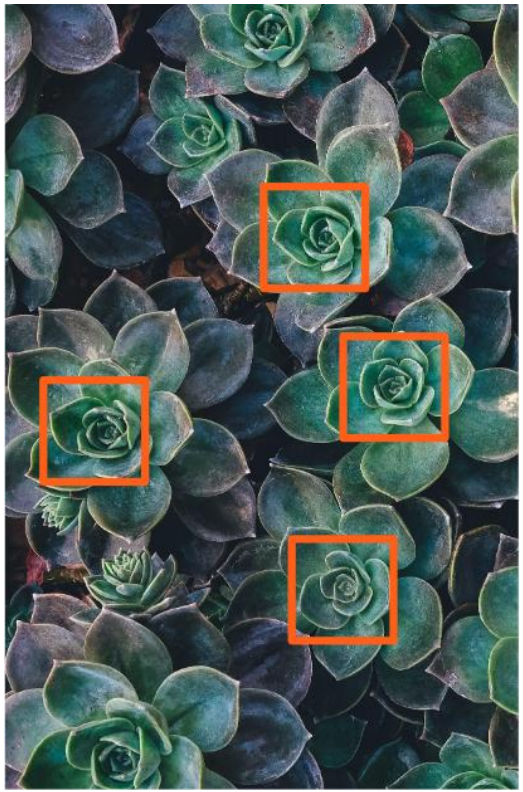
Localidad y estacionariedad

Locality: nearby pixels are more strongly correlated

Translation invariance: meaningful patterns can occur anywhere in the image

Estacionariedad: Translation invariance

Utilizando la estructura de las imágenes

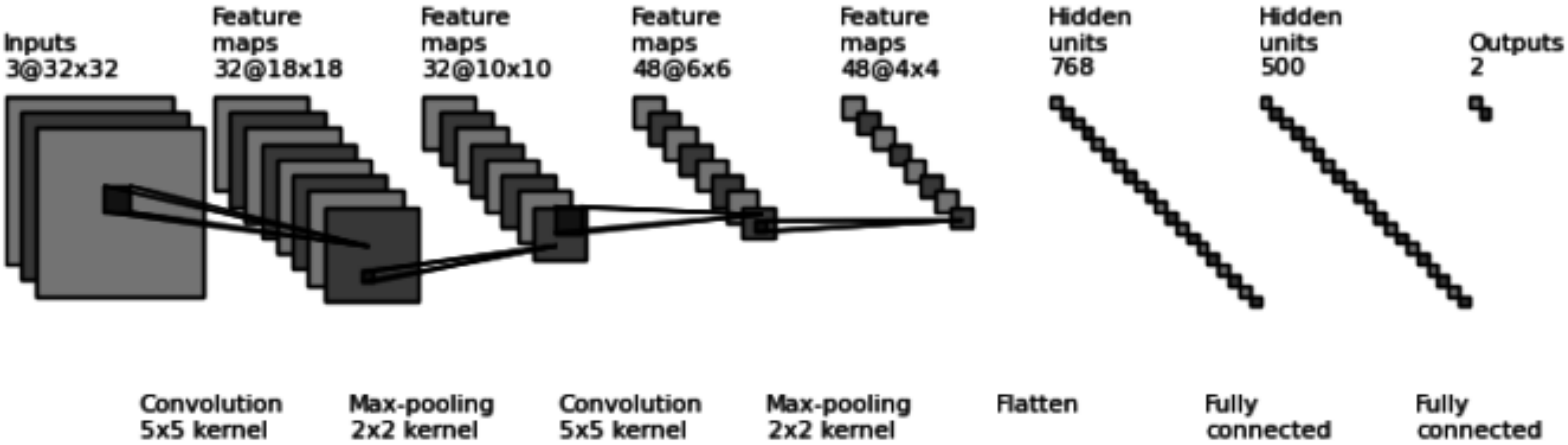


Weight sharing: use the same network parameters to detect local patterns at many locations in the image

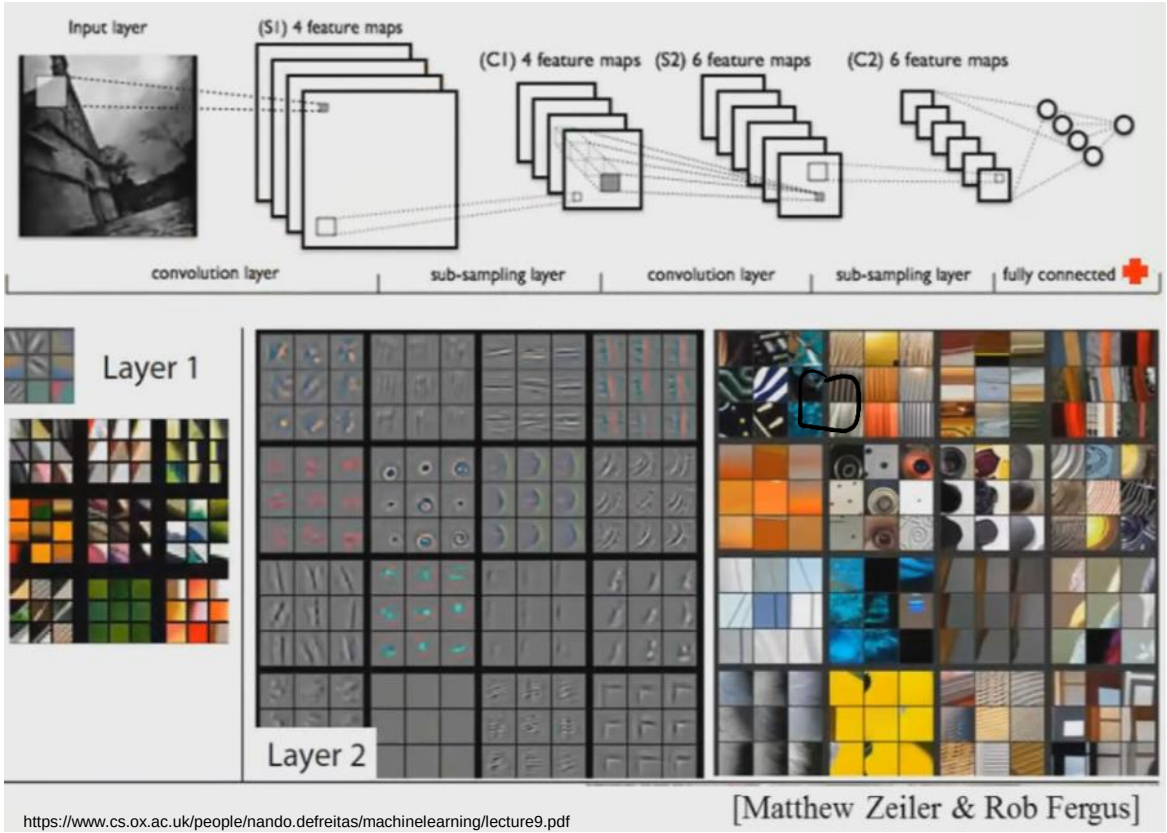
Utilizando la estructura de las imágenes

Hierarchy: local low-level features are composed into larger, more abstract features

Estructura de la red convolucional

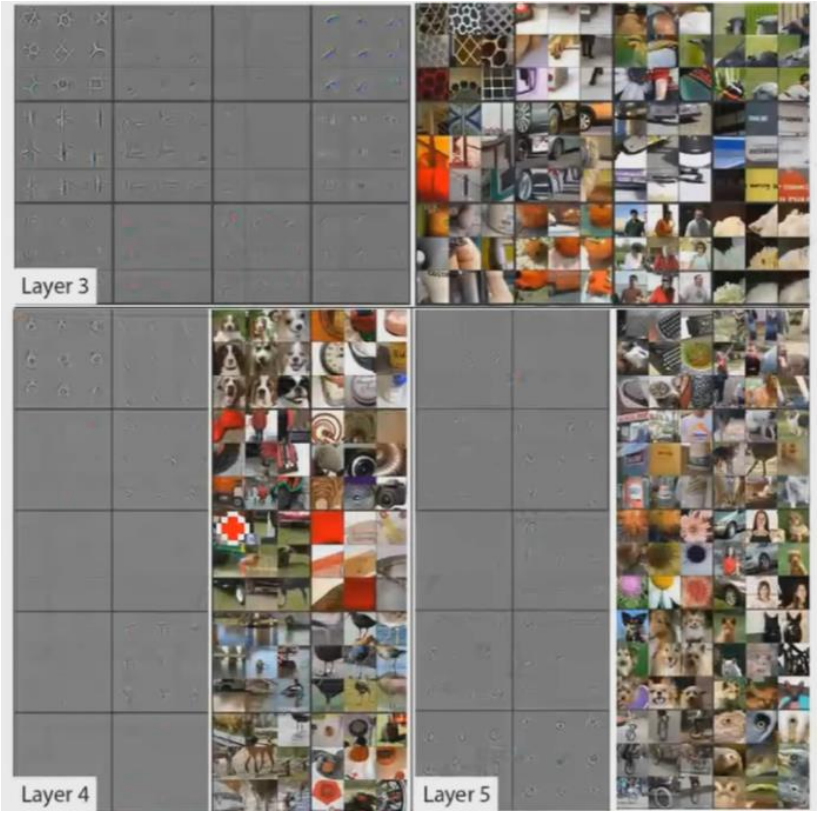


Ejemplos

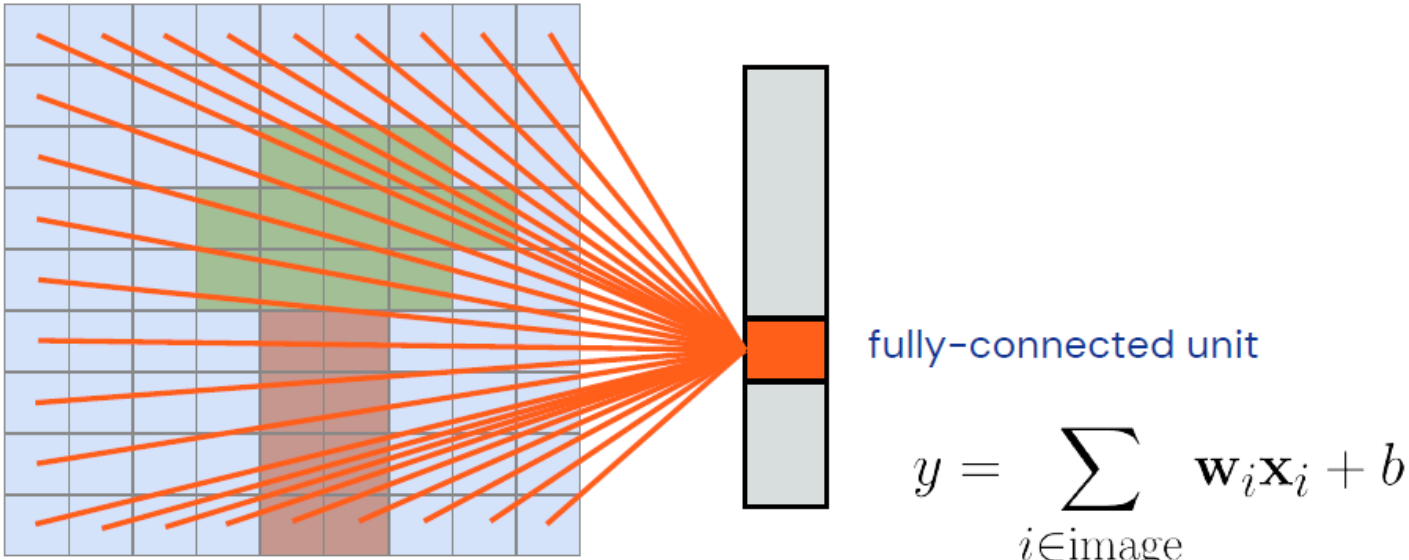


https://www.cs.ox.ac.uk/people/nando.defreitas/machinelearning/lecture9.pdf

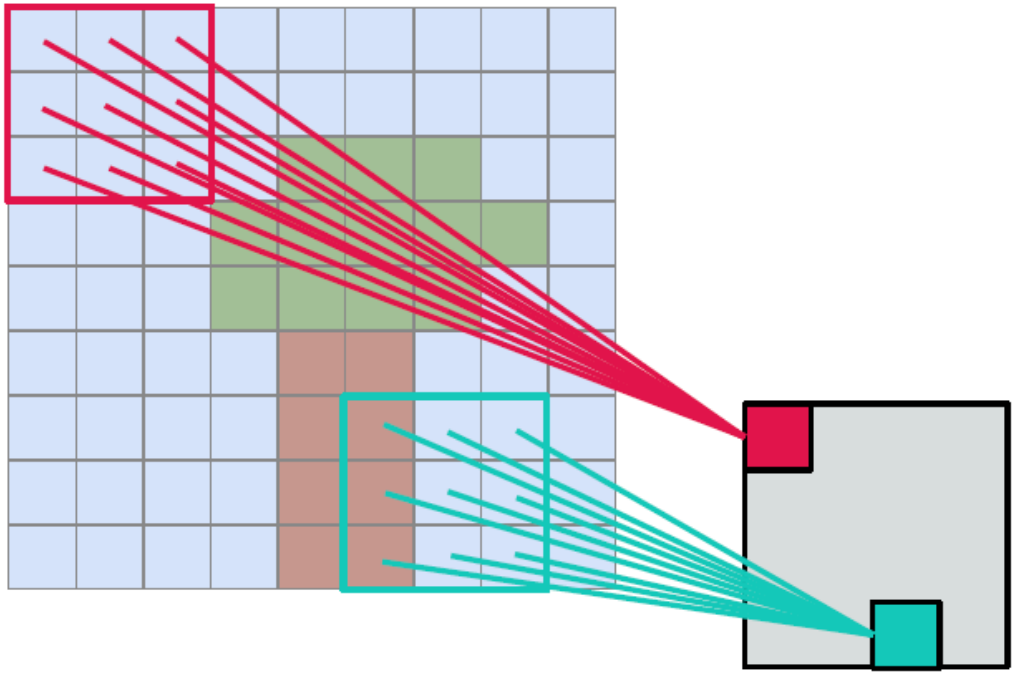
Ejemplos



De capas completamente a localmente conectadas a



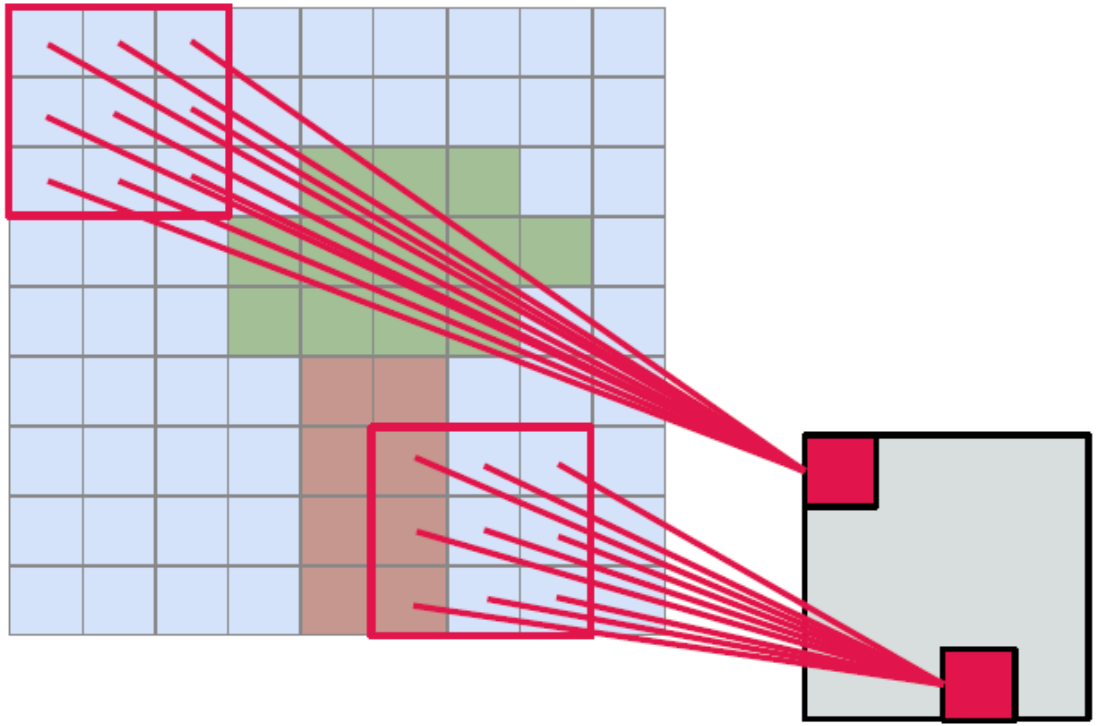
De capas completamente a localmente conectadas a



$$y = \sum_{i \in 3 \times 3} \mathbf{w}_i \mathbf{x}_i + b$$

locally-connected units
3×3 receptive field

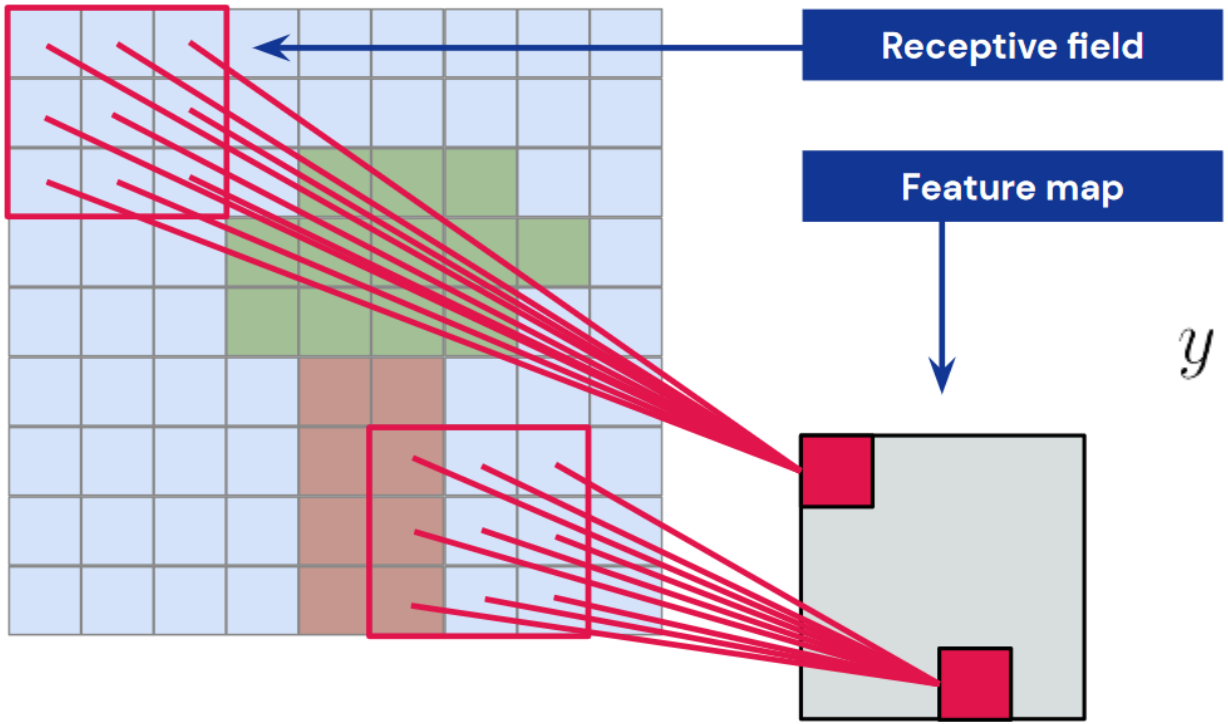
De capas completamente a localmente conectadas a



$$y = \mathbf{w} * \mathbf{x} + b$$

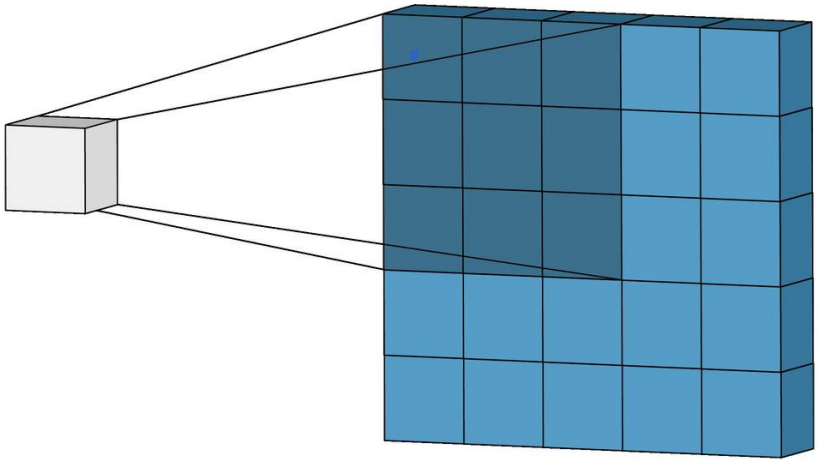
convolutional units
3×3 receptive field

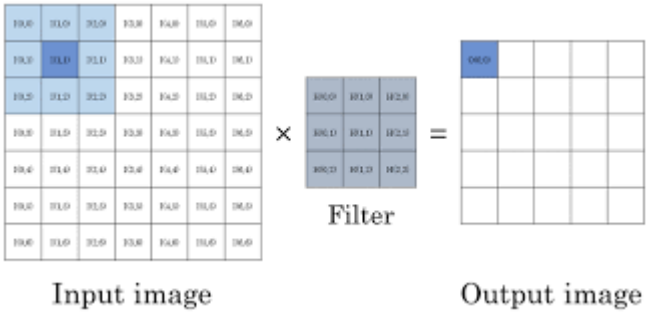
Receptive field



$$y = \mathbf{w} * \mathbf{x} + b$$

Convolución



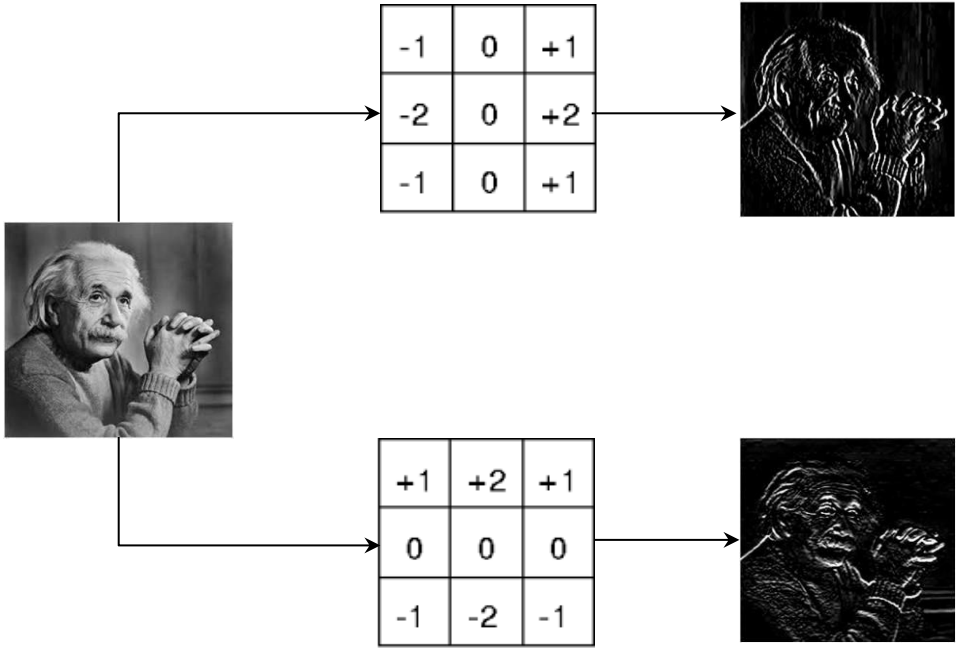


$$y = \mathbf{w} * \mathbf{x} + b$$

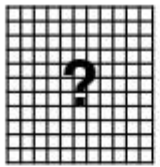
$$H_{out} = H_{in} - K + 1$$

$$W_{out} = W_{in} - K + 1$$

Detección de bordes



Convolución en CNNs

$f = \mathrm{ReLU}\Big($  $\cdot$  $\Big)$

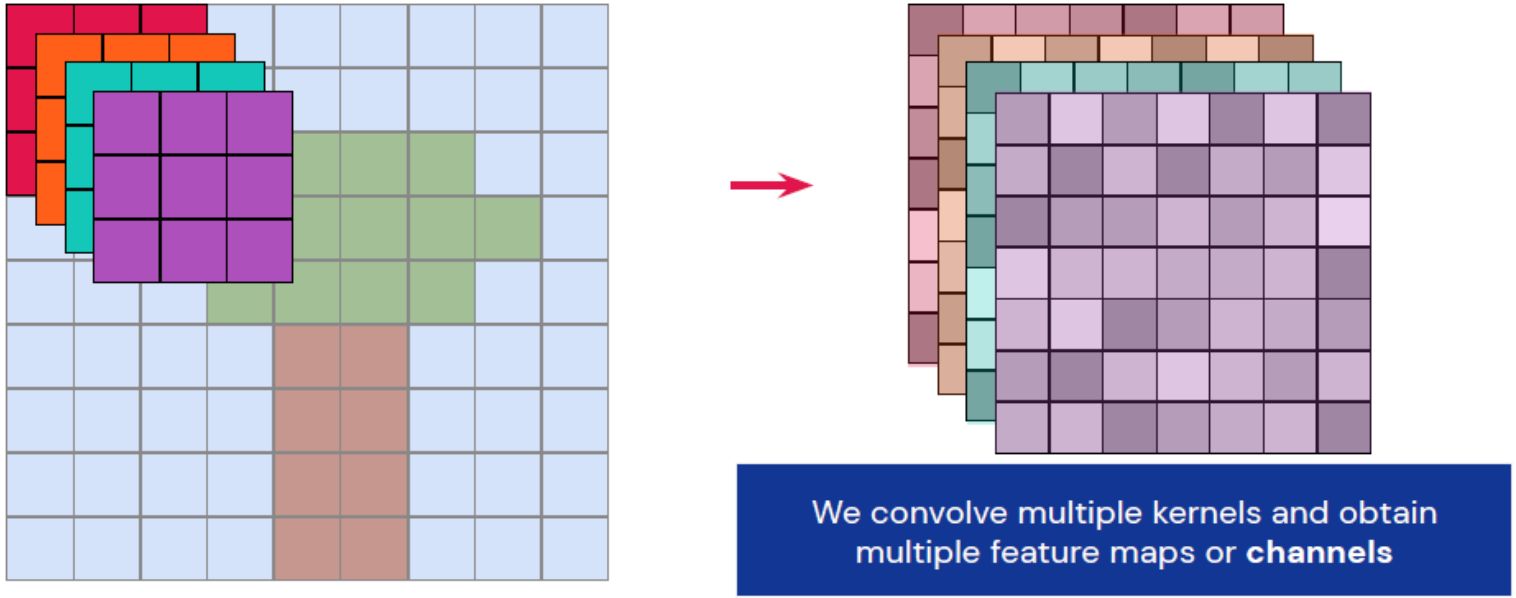
11x11 input

11x11 weights

input

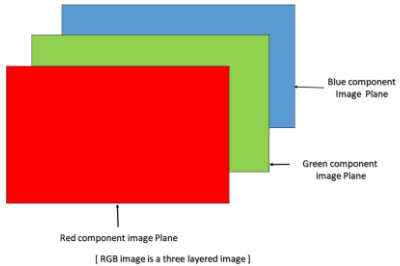
feature map

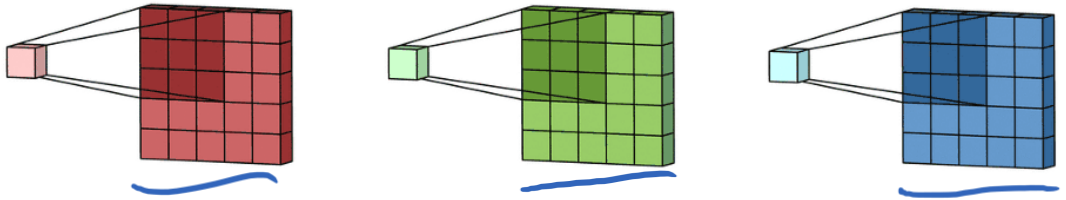
Convolución 2D

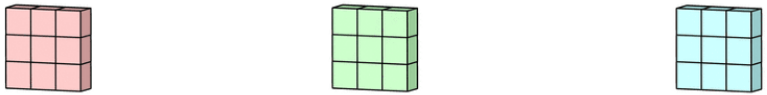


We convolve multiple kernels and obtain multiple feature maps or **channels**

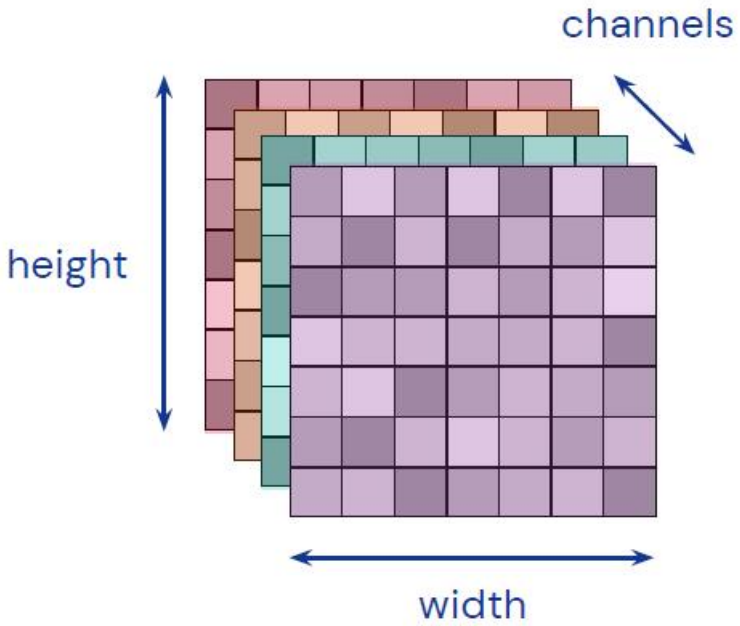
Convolución 2D



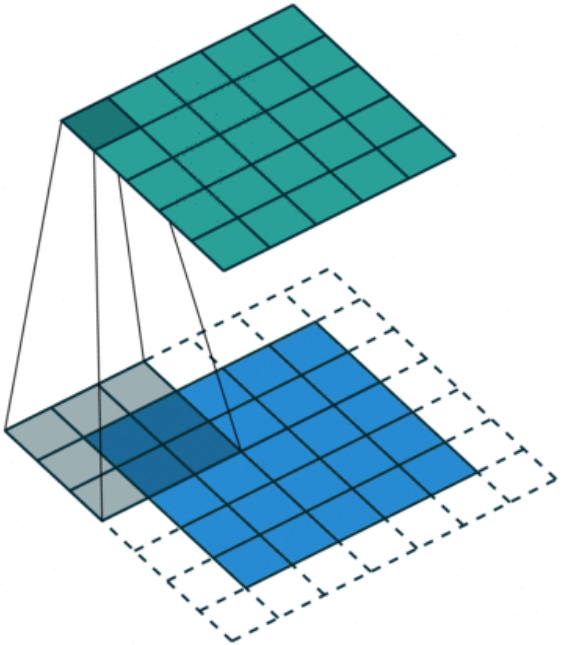


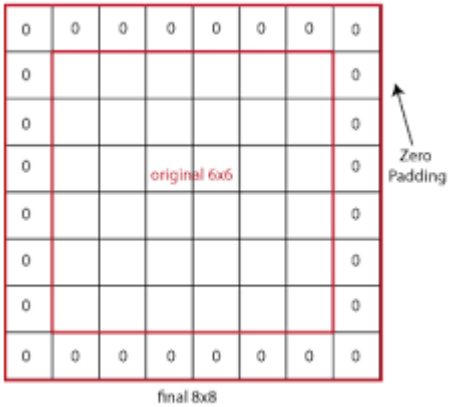


Entradas, salidas y tensores



Padding





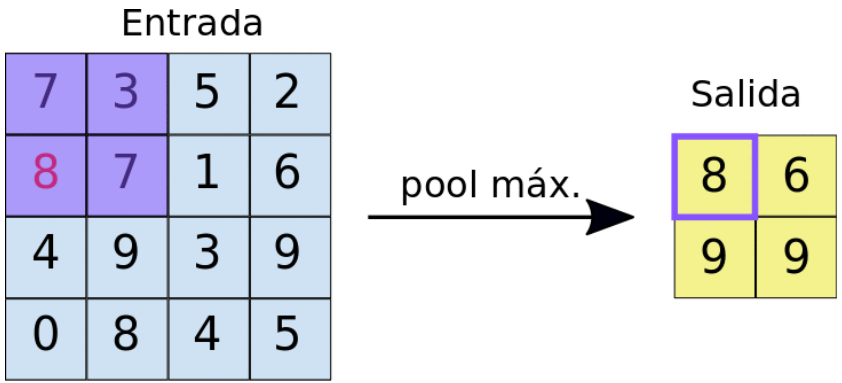
$$W_{out} = W_{in} + 2p - K + 1$$

$$H_{out} = H_{in} + 2p - K + 1$$

https://github.com/vdumoulin/conv_arithmetic

Pooling

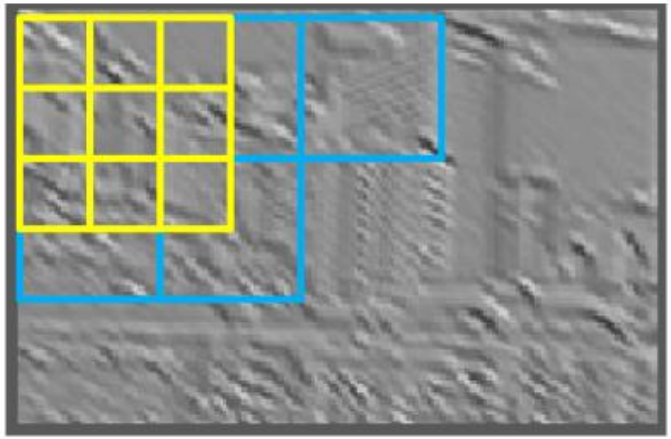
Queremos saber si en un cuadro aparece una característica pero no exáctamente donde



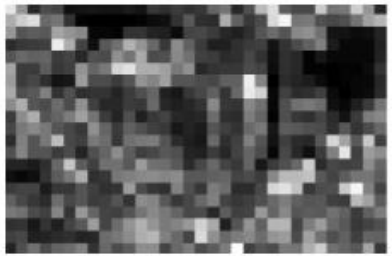
$$H_{out} = H_{in} - K + 1$$

$$W_{out} = W_{in} - K + 1$$

Pooling

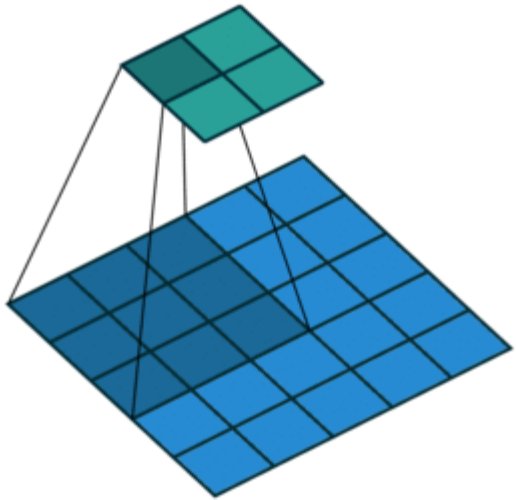


feature map



feature map
after max pooling

Stride



$$H_{out} = \frac{H_{in}+2p-K}{s} + 1$$

$$W_{out} = \frac{W_{in}+2p-K}{s} + 1$$

s: Stride
p: Padding
K: Kernel size
Hin: *Input Heigh* ***Hout:*** Output Heigh
Win: *Input Width* ***Wout:*** Output width

Grafo computacional

fully connected

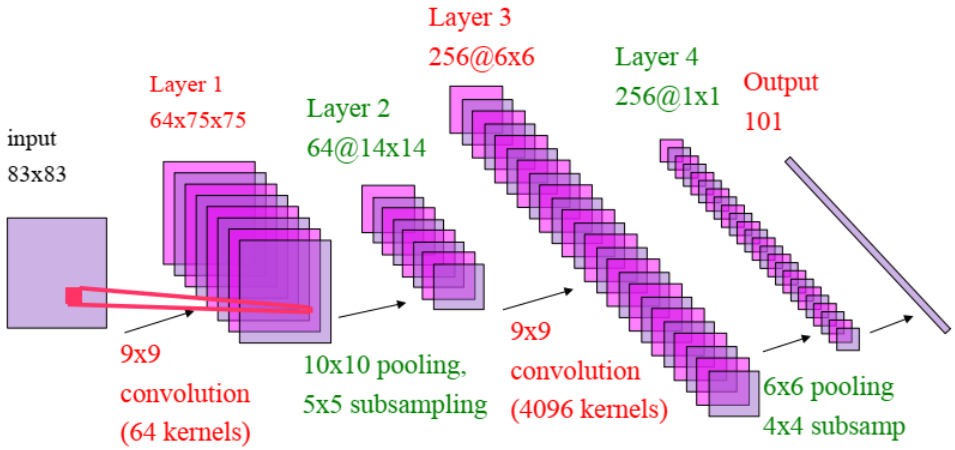
input

convolution

nonlinearity

pooling

Convolutional Neural Network



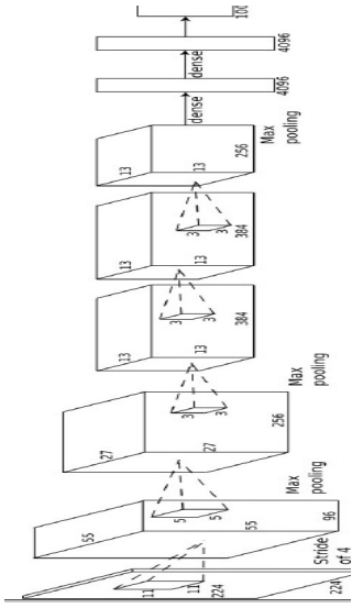
- Non-Linearity: half-wave rectification, shrinkage function, sigmoid
- Pooling: average, L1, L2, max
- Training: Supervised (1988-2006), Unsupervised+Supervised (2006-now)

(LeCun 13')

AlexNet

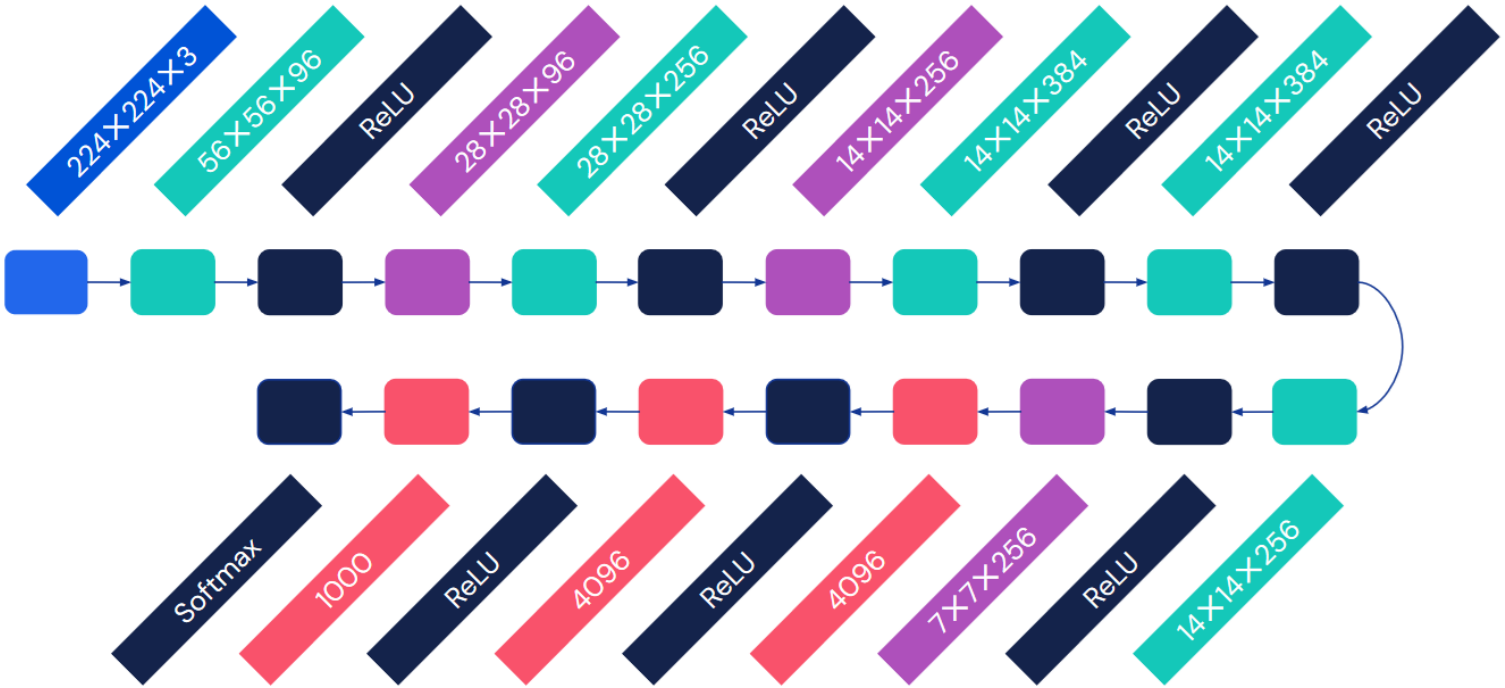
- 1000 categories, 1.5 Million labeled training samples
- Won the 2012 ImageNet LSVRC. 60 Million parameters, 832M MAC ops

4M	FULL CONNECT	4Mflop
16M	FULL 4096/ReLU	16M
37M	FULL 4096/ReLU	37M
	MAX POOLING	
442K	CONV 3x3/ReLU 256fm	74M
1.3M	CONV 3x3ReLU 384fm	224M
884K	CONV 3x3/ReLU 384fm	149M
	MAX POOLING 2x2sub	
	LOCAL CONTRAST NORM	
307K	CONV 11x11/ReLU 256fm	223M
	MAX POOL 2x2sub	
	LOCAL CONTRAST NORM	
35K	CONV 11x11/ReLU 96fm	105M

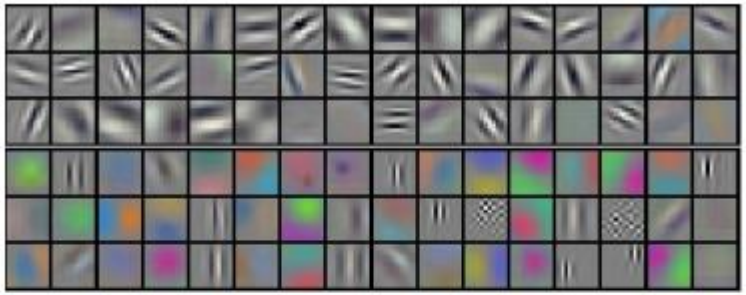


(Krizhevsky et al., 12')

AlexNet


224×224×3
56×56×96
ReLU
28×28×96
28×28×256
ReLU
14×14×256
14×14×384
ReLU
14×14×384
ReLU
Softmax
1000
ReLU
4096
ReLU
4096
7×7×256
ReLU
14×14×256

Convolutional Neural Network



96 convolutional filters on the first layer
(filters are of size 11x11x3, applied across
input images of size 224x224x3)

(Krizhevsky et al., 12')

Batch normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Figure from Ioffe et al. (2015)

Want to learn more?

Ioffe, S.; Szegedy, C.
Batch normalization: Accelerating deep network training by reducing internal covariate shift International conference on machine learning (2015)

Reduces sensitivity to **initialisation**

Introduces stochasticity and acts as a **regulariser**

Segmentación semántica