Redes Neuronales Convolucionales

Aprendizaje Automático Aplicado

Facultad de Ingeniería
Universidad de la República

Agenda

- Un poco de historia
- Aplicaciones
- Convolución en imágenes
- Capa de convolución
- Arquitecturas

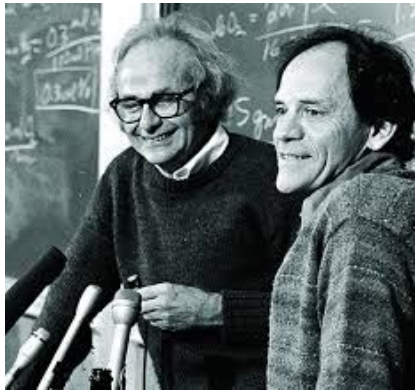
Un poco de historia

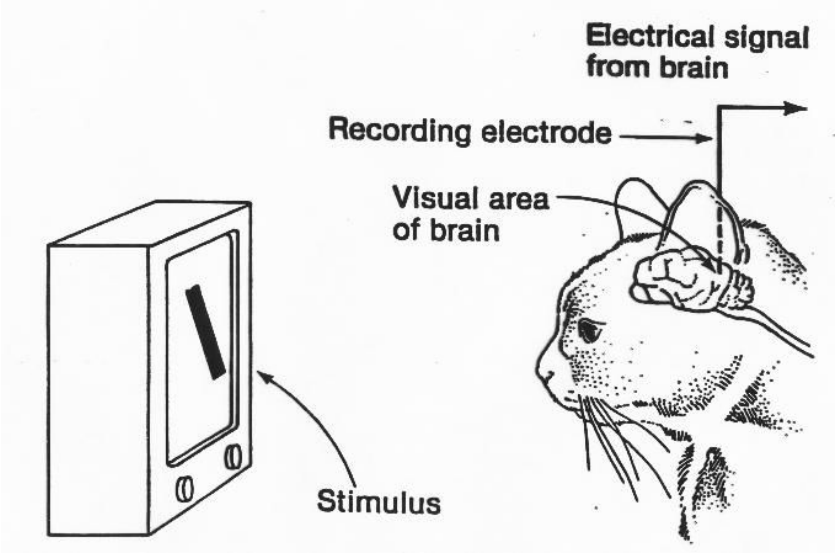
RECEPTIVE FIELDS OF SINGLE NEURONES IN THE CAT'S STRIATE CORTEX

By D. H. HUBEL* and T. N. WIESEL*

From the Wilmer Institute, The Johns Hopkins Hospital and University, Baltimore, Maryland, U.S.A.

(*Received* 22 *April* 1959)

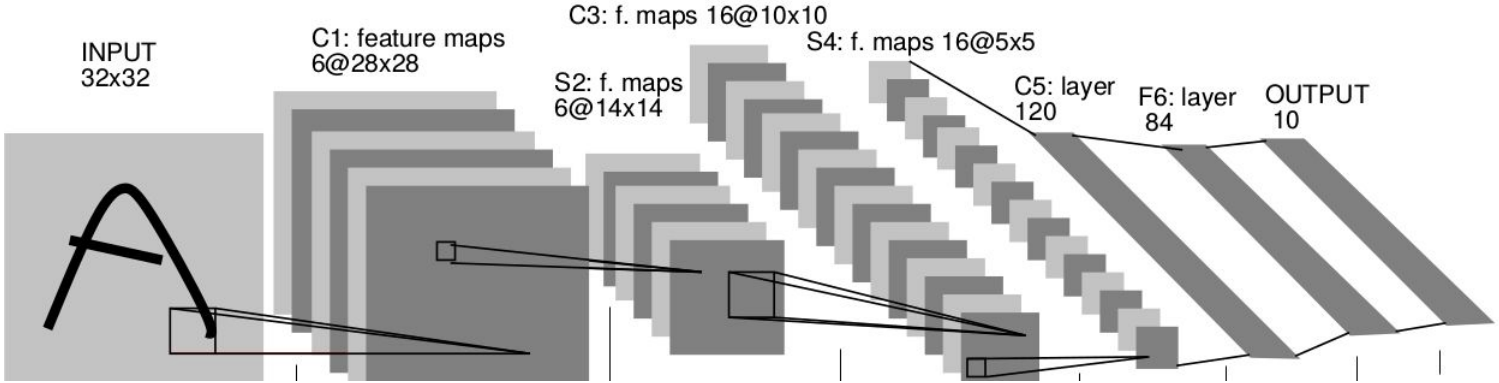




Un poco de historia

- LeNet-5

  Gradient-Based Learning Applied to Document Recognition [Yann LeCun et al., 1998] - (Citado 20396 veces)



Un poco de historia

- “ImageNet Classification with Deep Convolutional Neural Networks” [Alex Krizhevsky et al., 2012] (citado 40.305 veces)

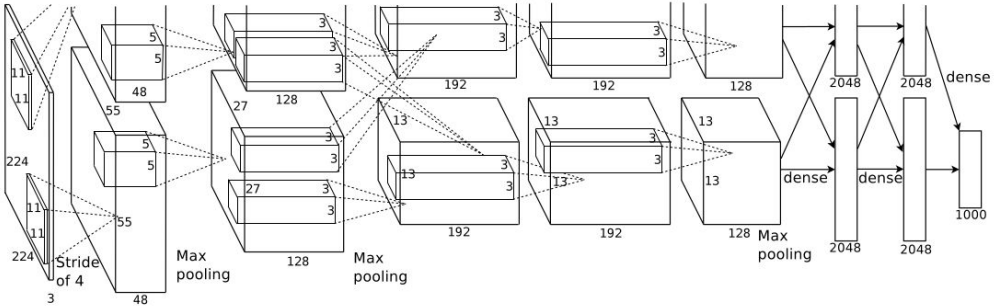
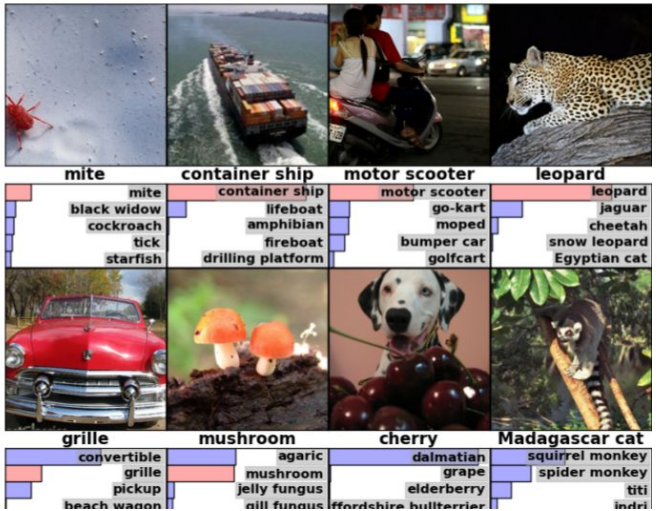
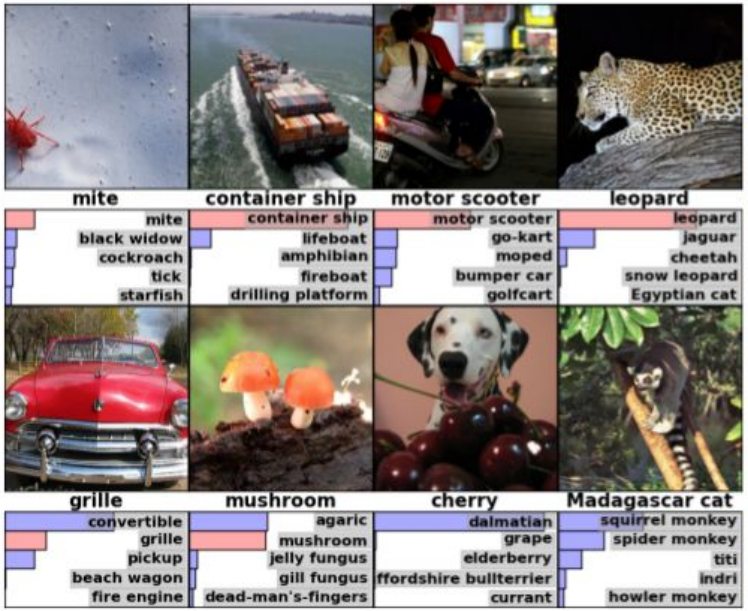


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network’s input is 150,528-dimensional, and the number of neurons in the network’s remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.



Aplicaciones

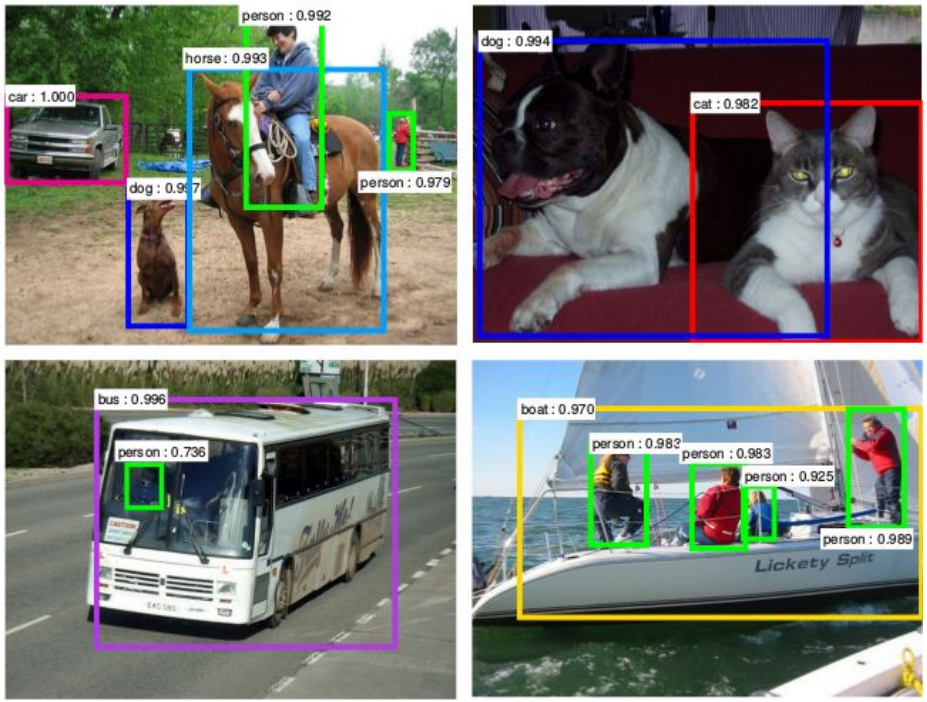
- Clasificación



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012. (Cited by 45305)

Aplicaciones

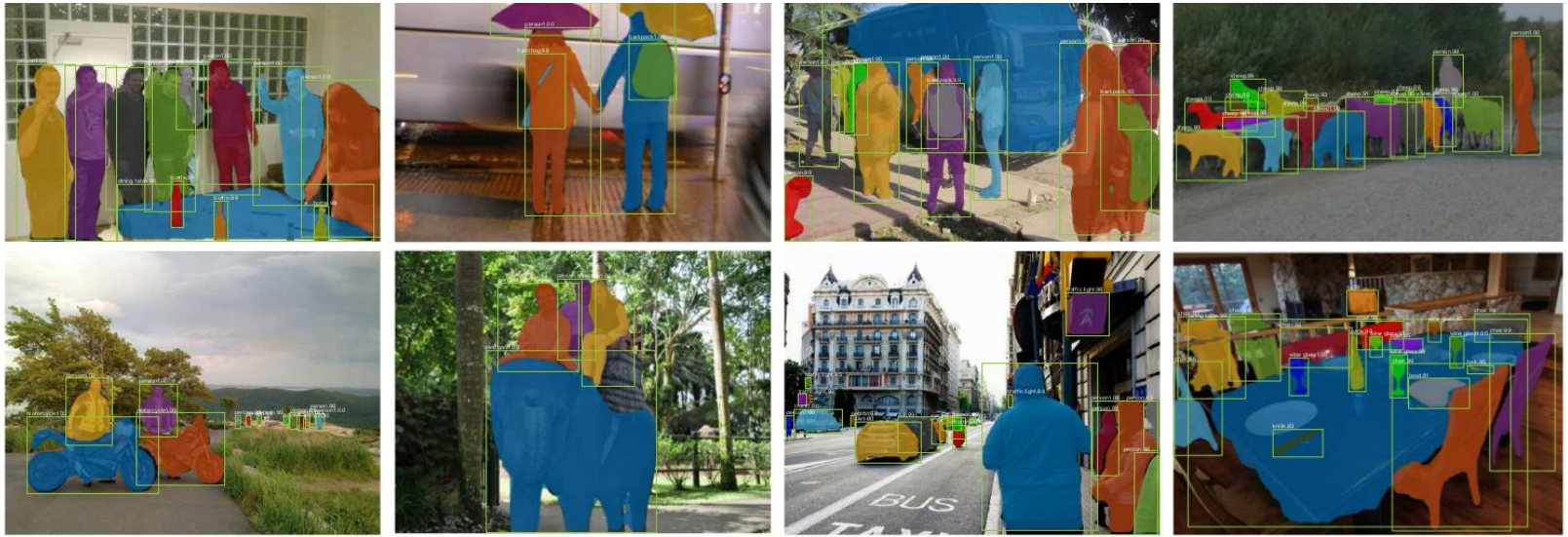
- Detección



Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015. (Cited by 11368)

Aplicaciones

- Segmentación



He, Kaiming, et al. "Mask r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2017. (Cited by 3000)

Aplicaciones

- Descripción de imágenes

man in black shirt is playing guitar.

construction worker in orange safety vest is working on road.

two young girls are playing with lego toy.

boy is doing backflip on wakeboard.

Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015. (Cited by 2629)

Aplicaciones

- Muchas más..

Convolución en imágenes

- Operación lineal entre una imagen y un filtro, generando una nueva imagen
- El valor de cada píxel se calcula como la suma ponderada de los píxeles de la imagen u y un núcleo de convolución h:

$$(u * h)(i, j) = \sum_{k,l} u(i-k, j-l)h(k, l)$$

45	60	98	127	132	133	137	133
46	65	98	123	126	128	131	133
47	65	96	115	119	123	135	137
47	63	91	107	113	122	138	134
50	59	80	97	110	123	133	134
49	53	68	83	97	113	128	133
50	50	58	70	84	102	116	126
50	50	52	58	69	86	101	120

*

0.1	0.1	0.1
0.1	0.2	0.1
0.1	0.1	0.1

=

69	95	116	125	129	132
68	92	110	120	126	132
66	86	104	114	124	132
62	78	94	108	120	129
57	69	83	98	112	124
53	60	71	85	100	114

Convolución en imágenes

- Patrones invariantes a traslaciones
- Aprenden patrones jerárquicos

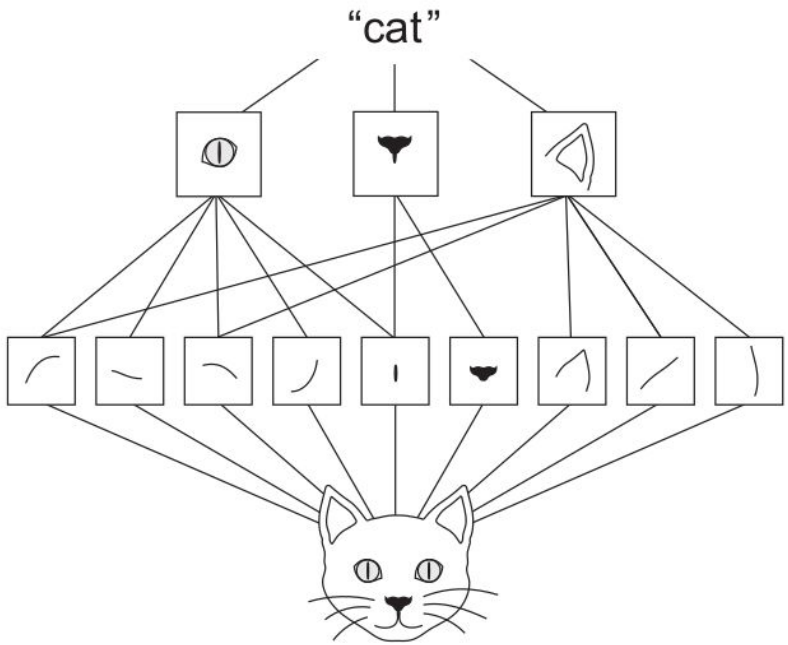
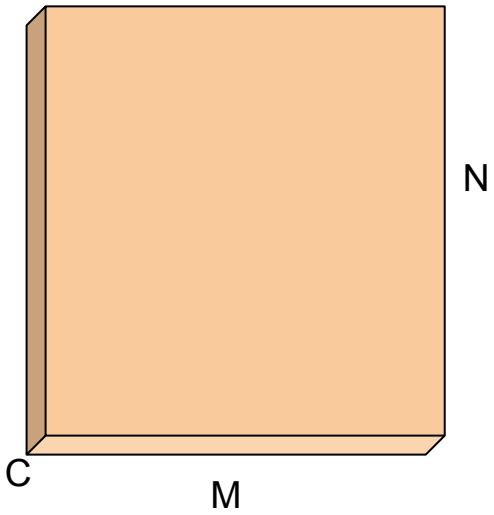
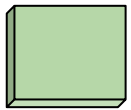


Figure 5.2 The visual world forms a spatial hierarchy of visual modules: hyperlocal edges combine into local objects such as eyes or ears, which combine into high-level concepts such as “cat.”

Capa de convolución

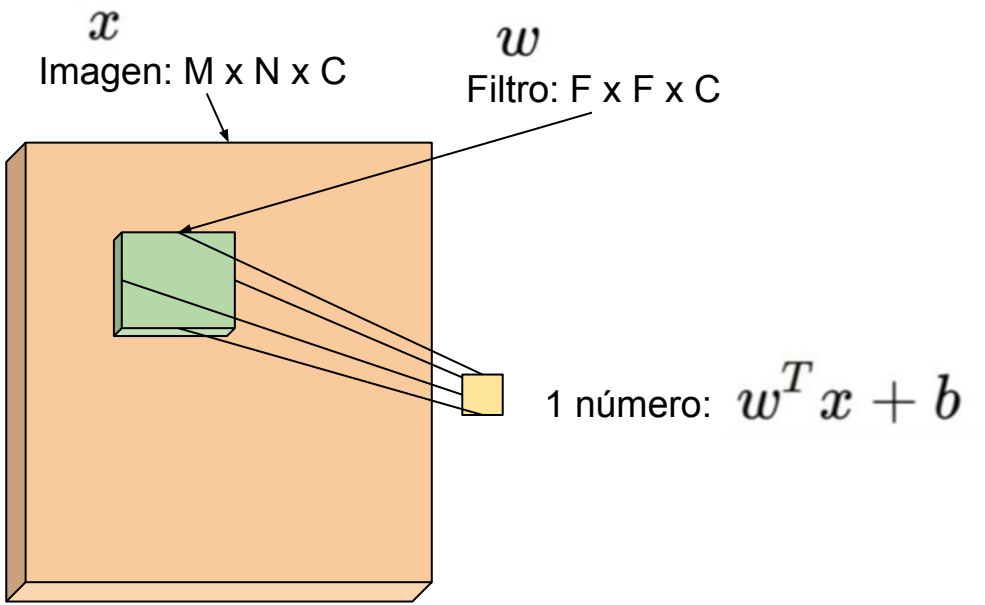
Imagen: M x N x C



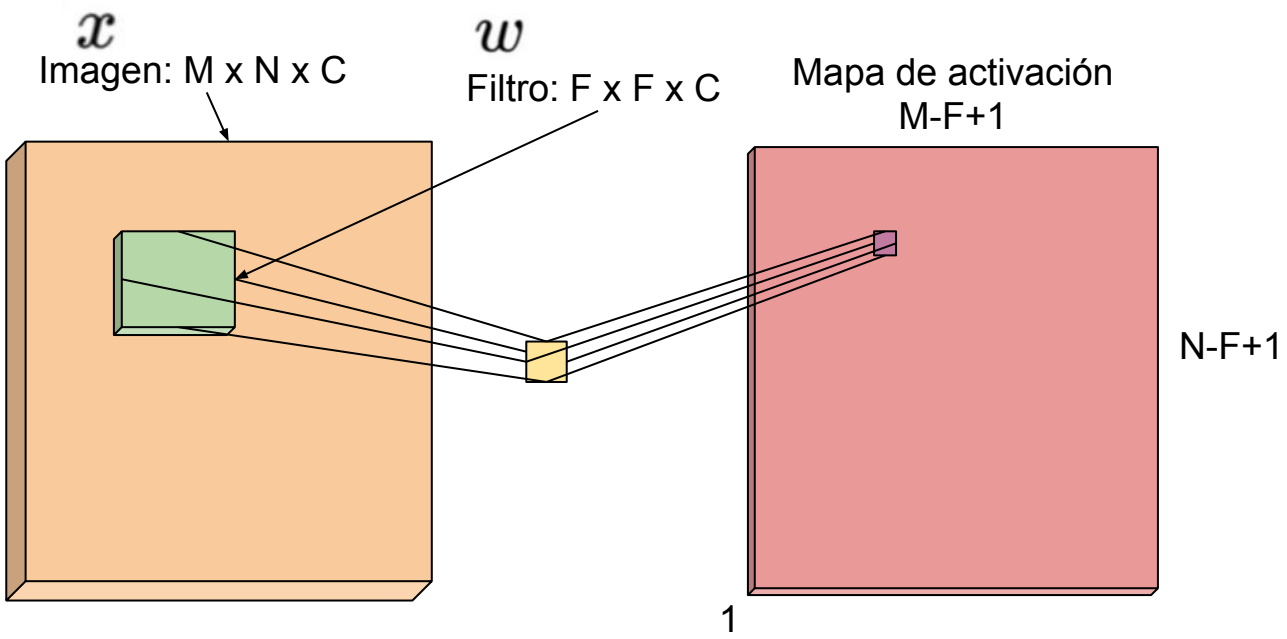


Filtro: F x F x C

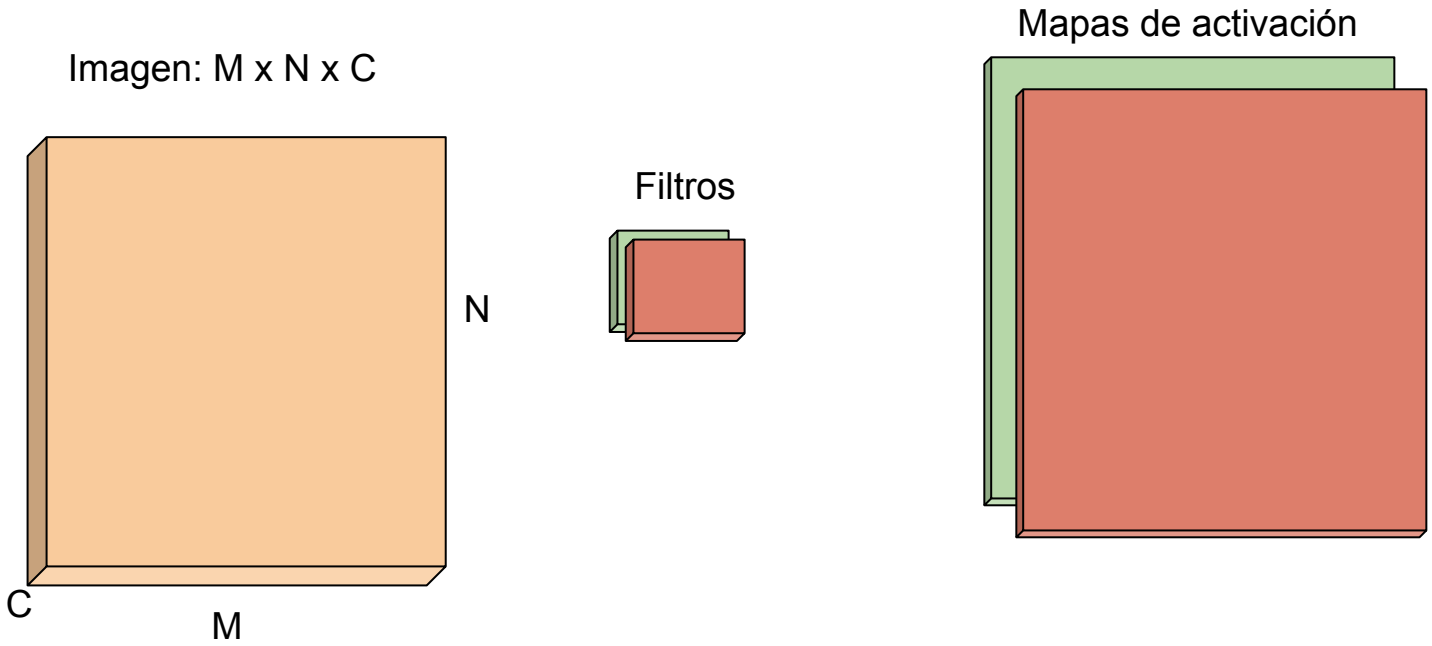
Capa de convolución



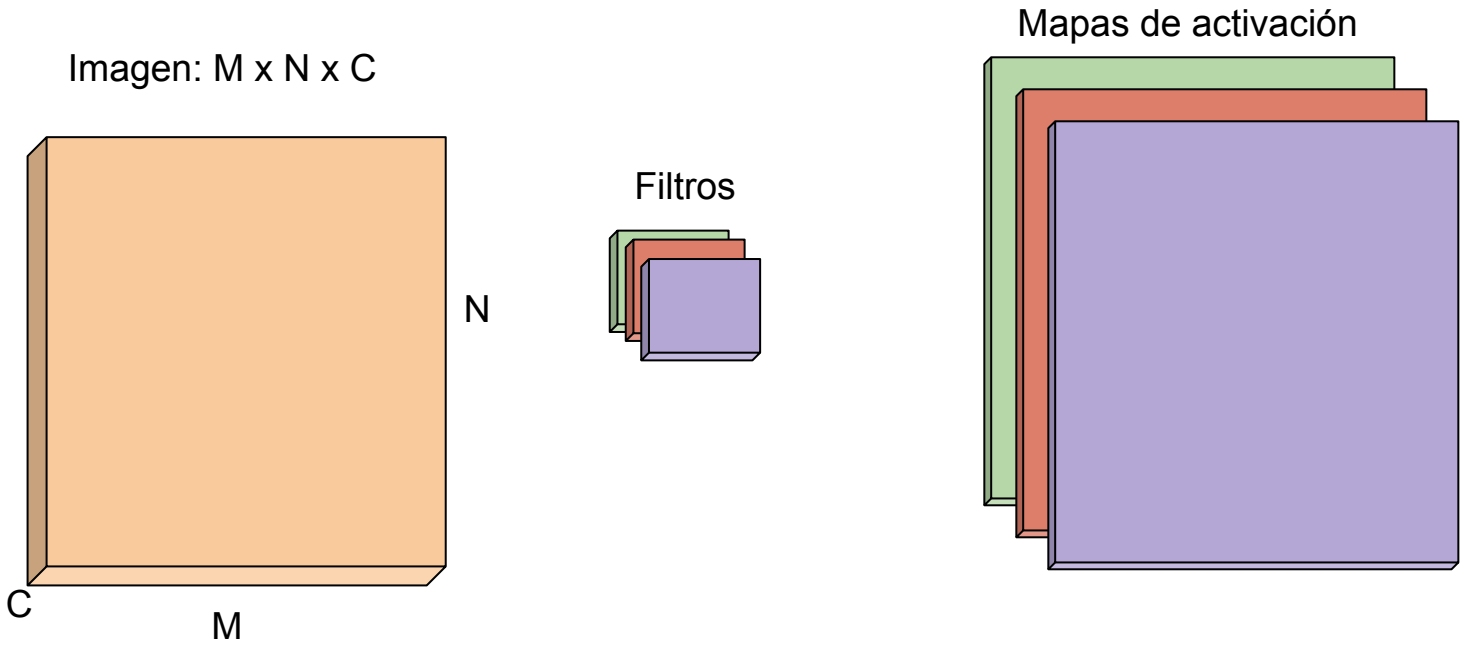
Capa de convolución



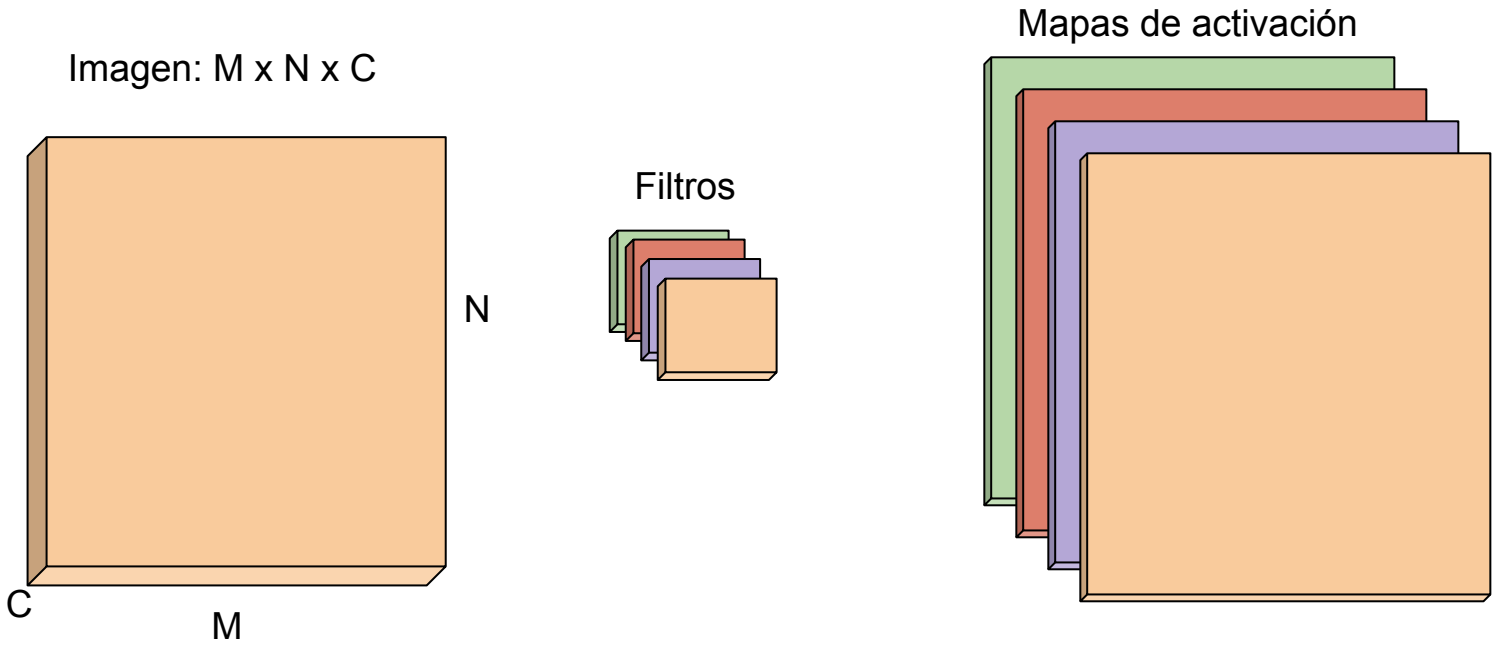
Capa de convolución



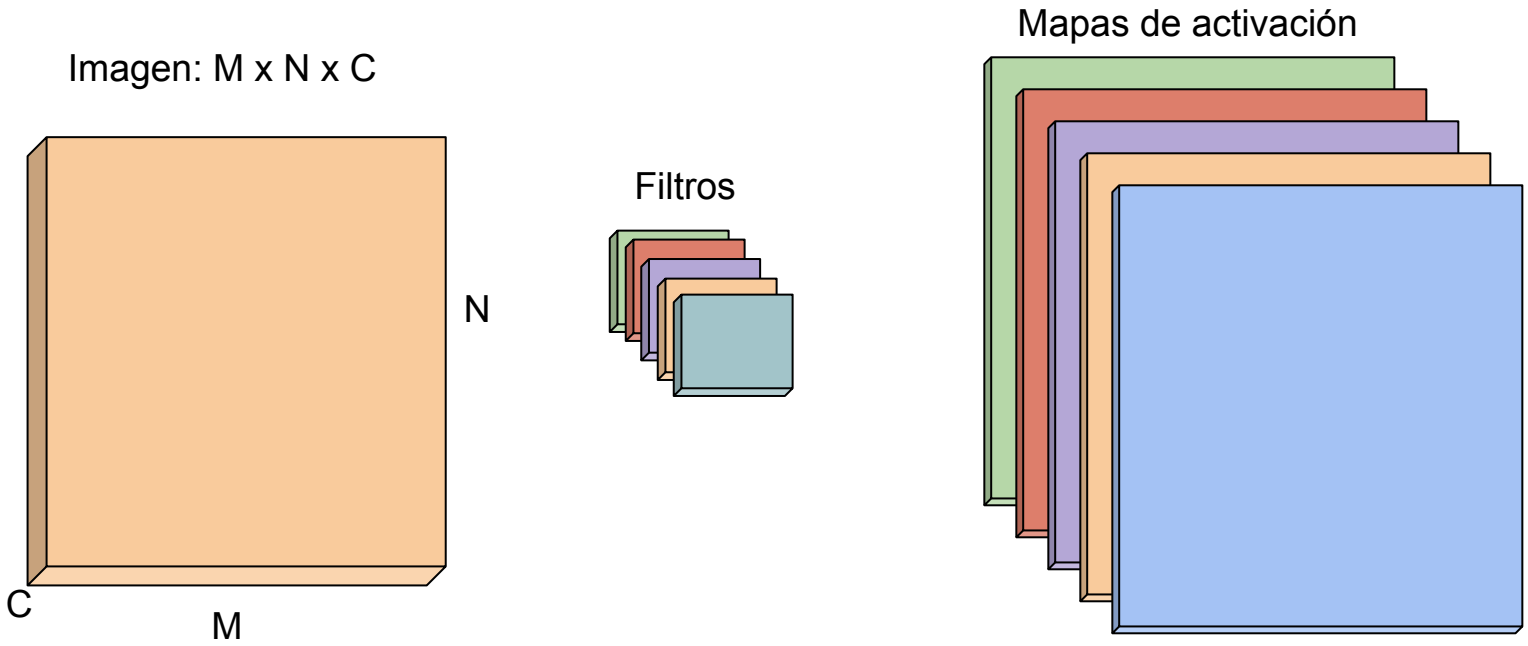
Capa de convolución



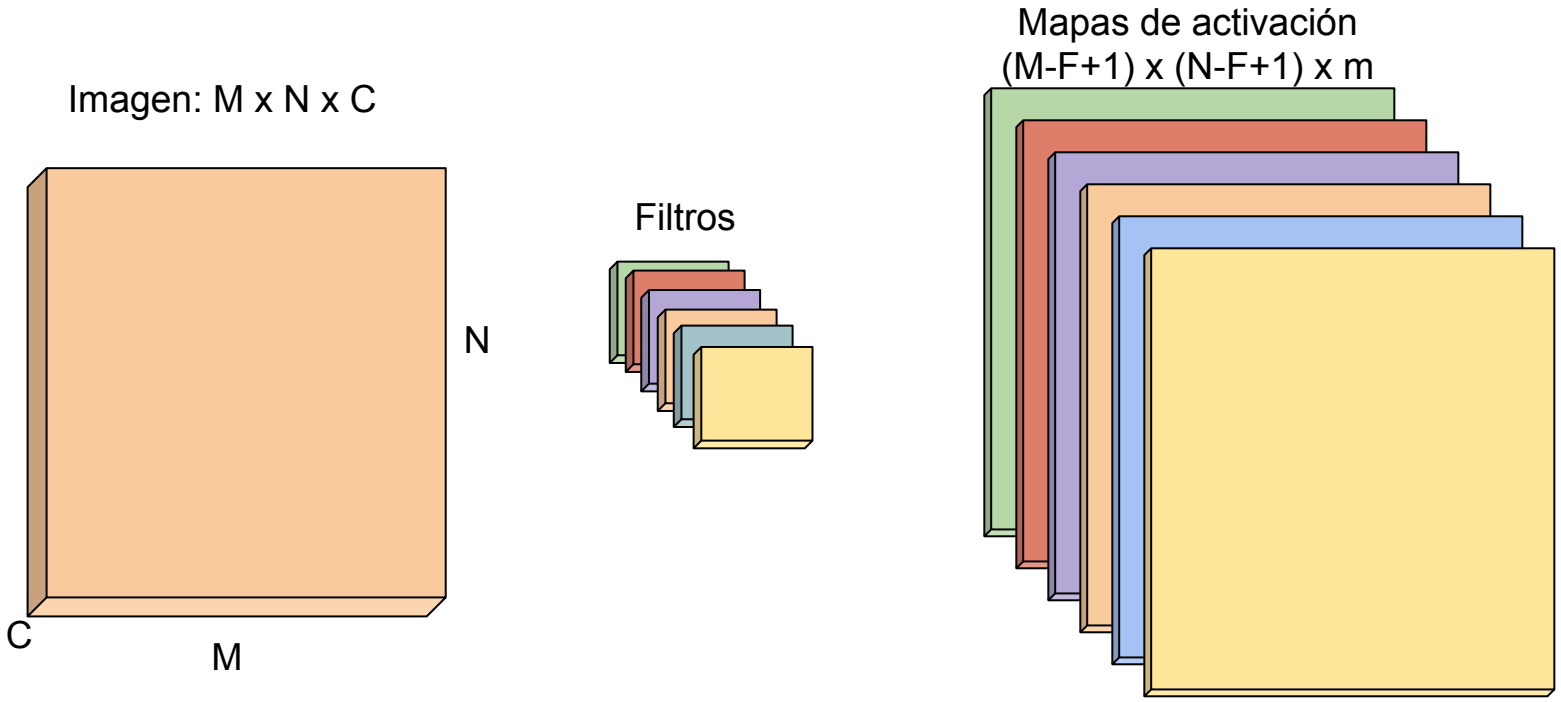
Capa de convolución



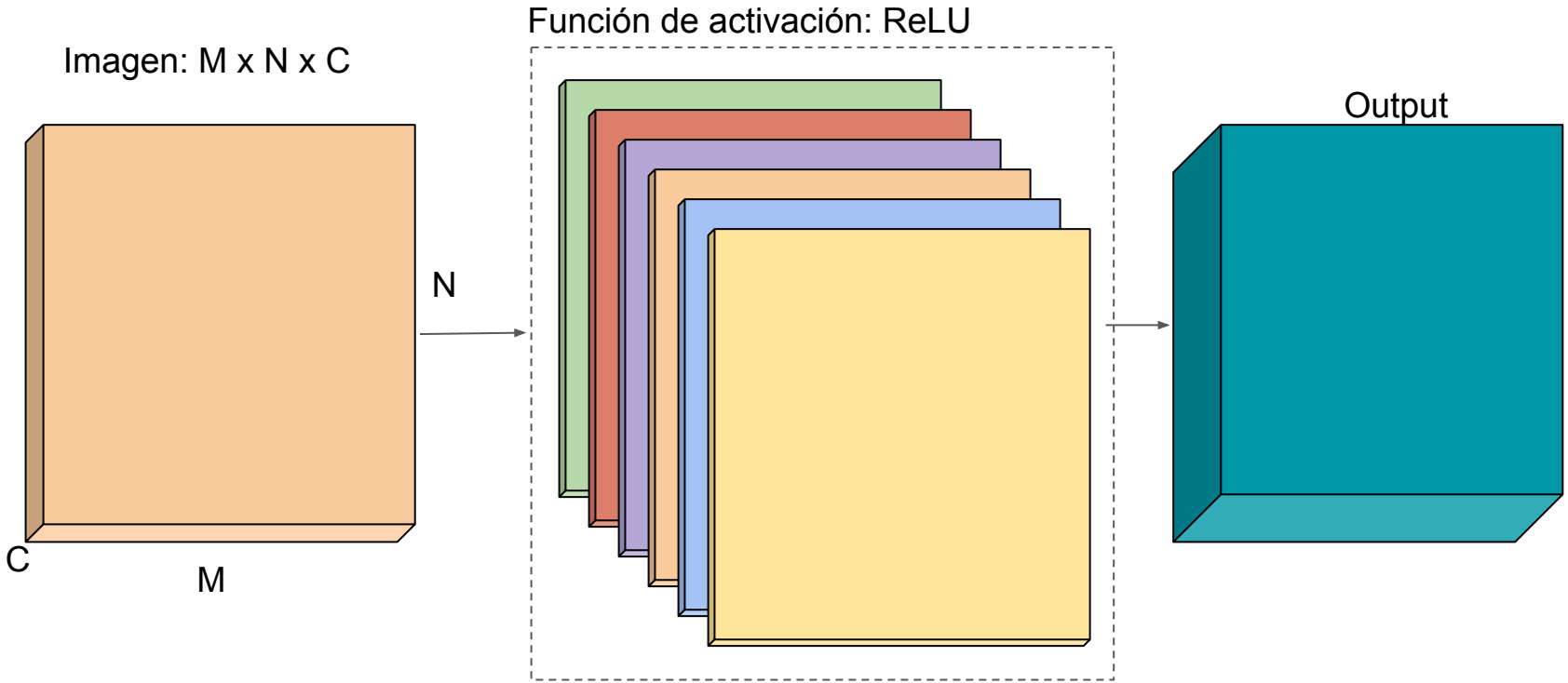
Capa de convolución



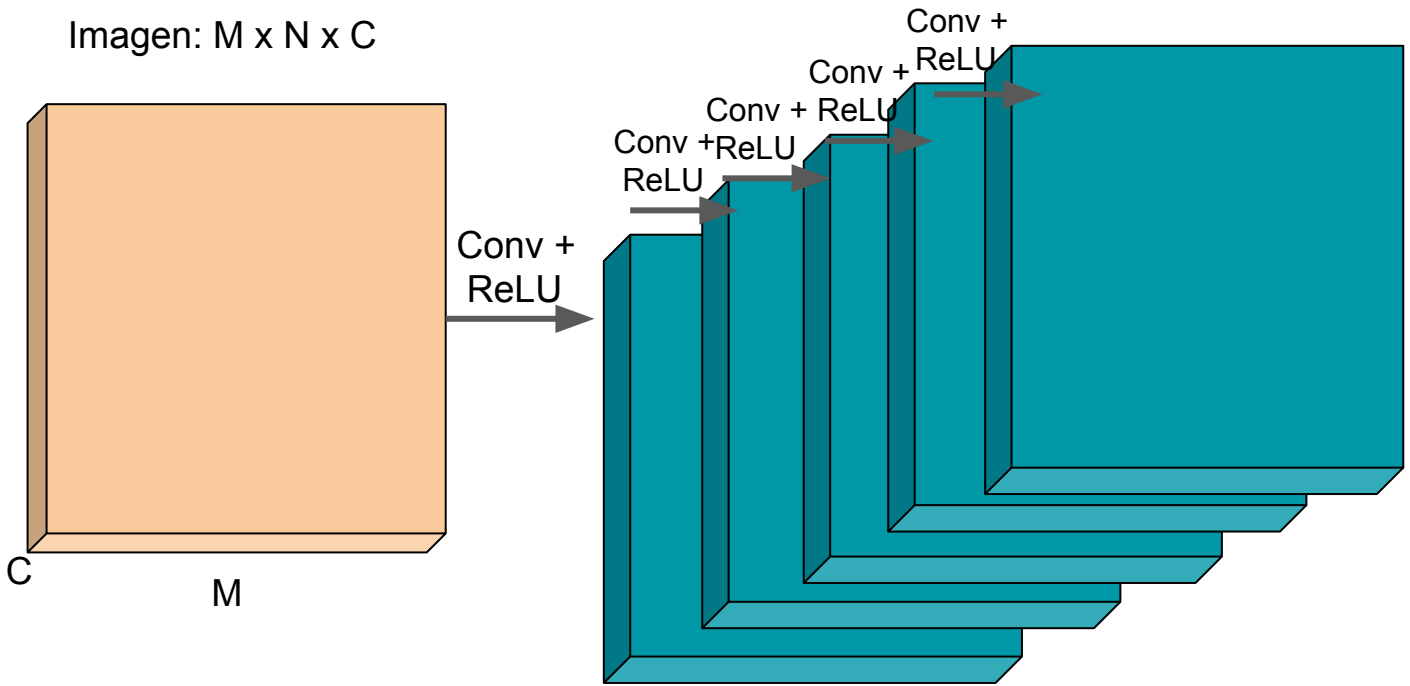
Capa de convolución



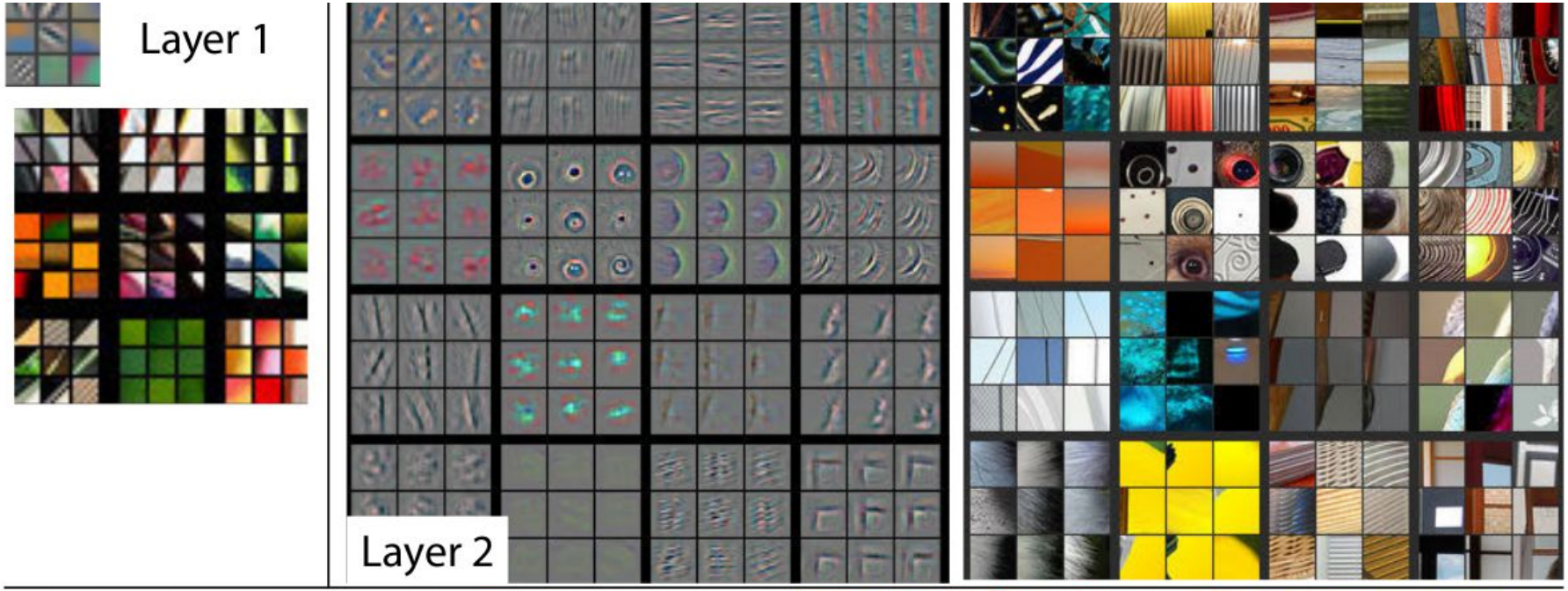
Capa de convolución: Convolución + Activación



Capa de convolución: Convolución + Activación

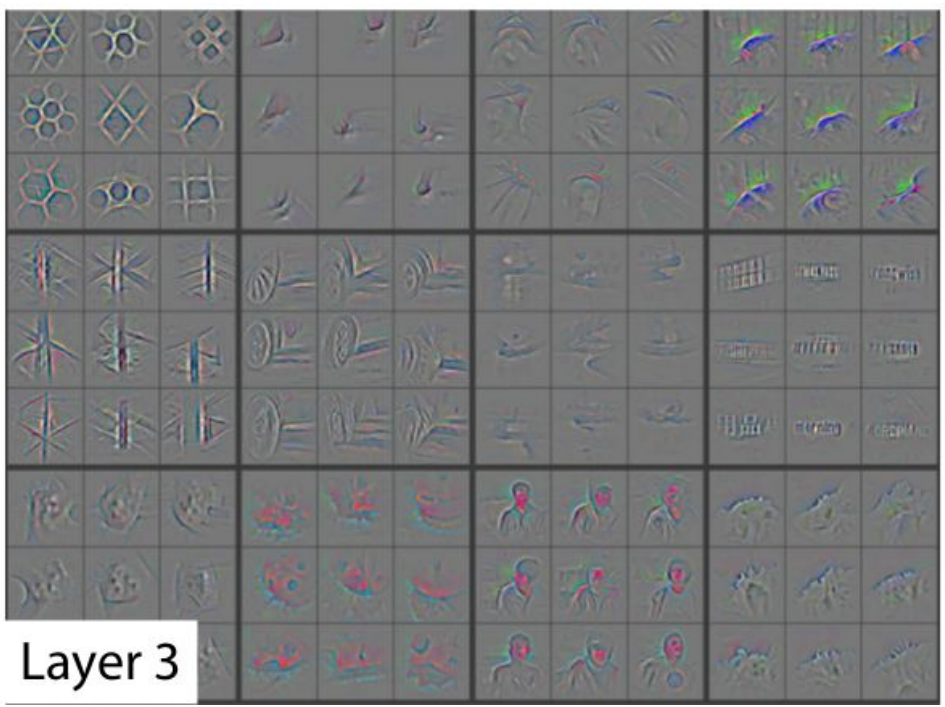


Visualización



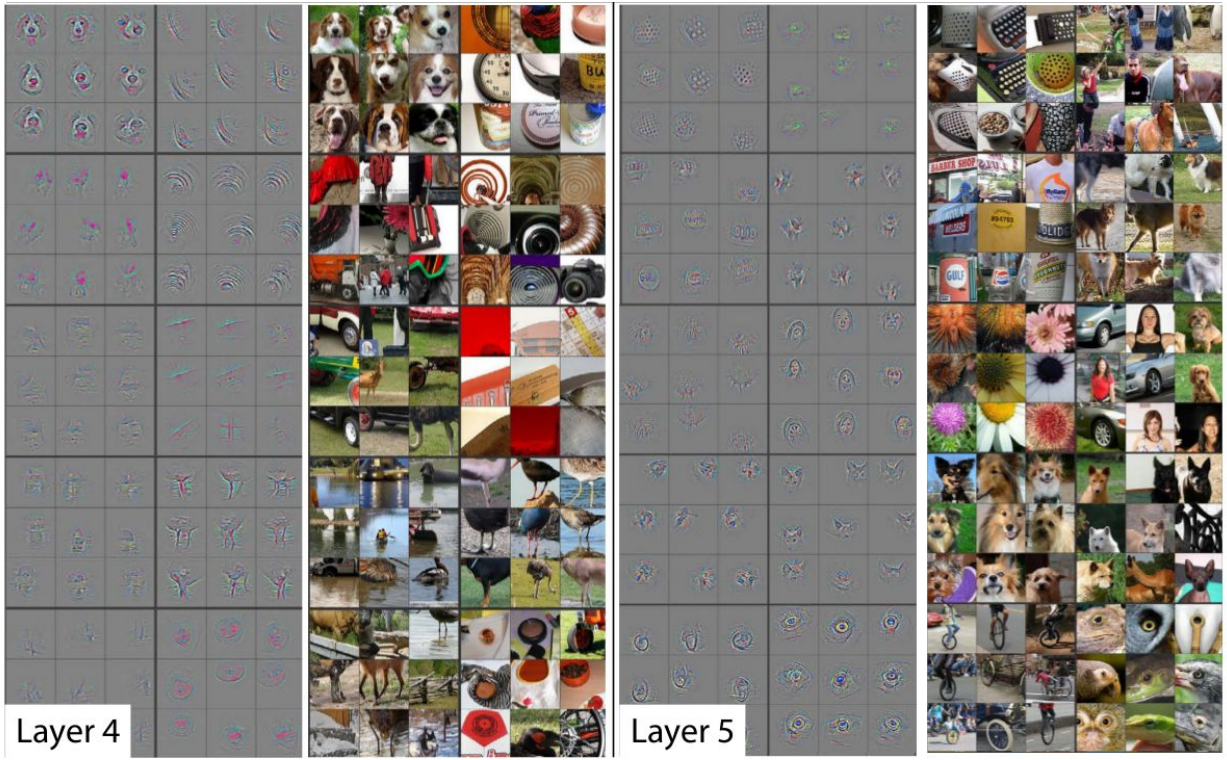
Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

Visualización



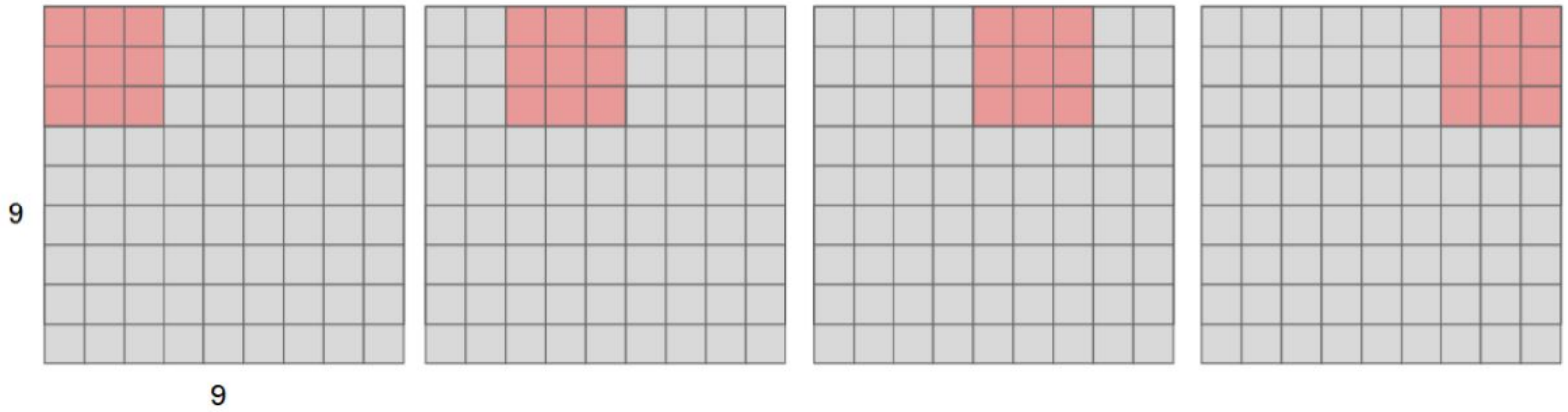
Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

Visualización



Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer, Cham, 2014.

Dimensiones

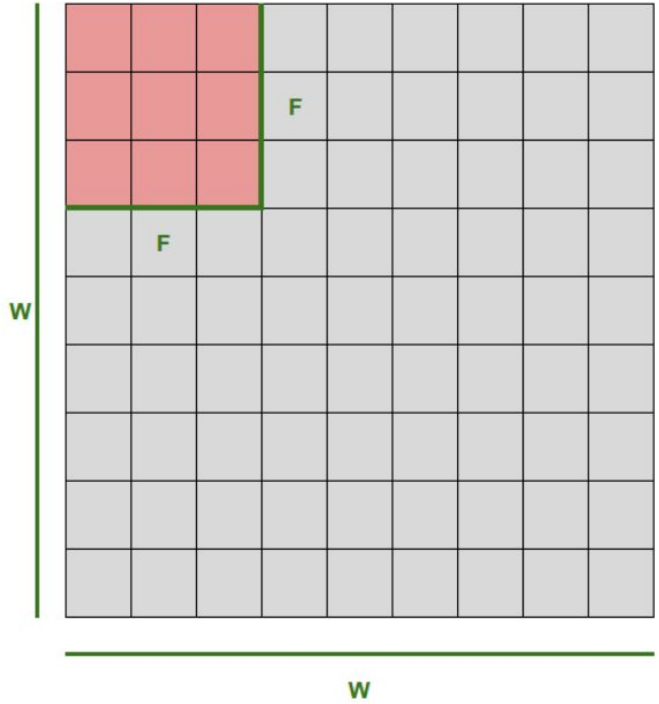


(a) Imagen de entrada: 9×9,
Filtro: 3×3 Stride: 2,
Salida: 4×4.

Dimensiones

- Tamaño de salida:
  - (W-F) / stride + 1
  - Ej: (9-3) / 2 + 1 = 4

- Stride = 4 ??



(b) Tamaño de salida: $(W-F)/\text{stride}+1$,
Ej: $(9-3)/2+1=4$

Zero-padding

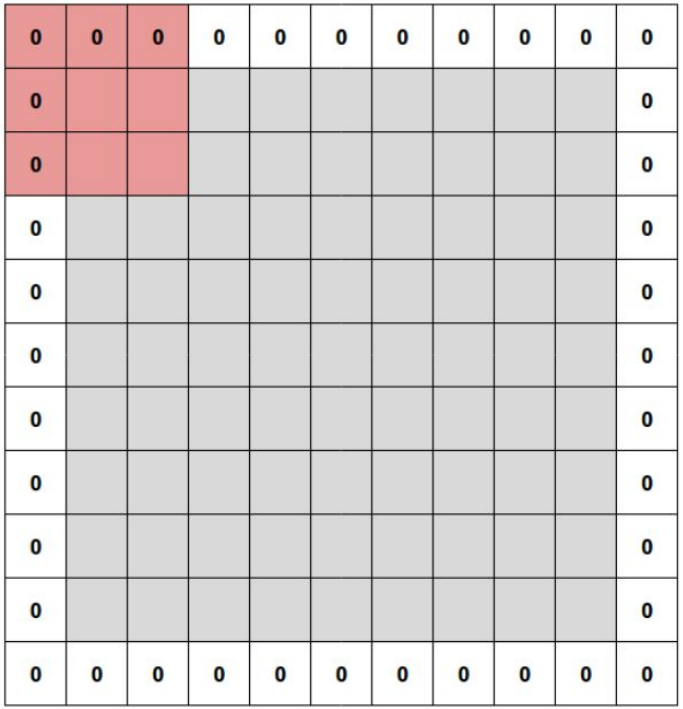
- Imagen de entrada: 9 × 9
- Filtro: 3 × 3
- Stride: 1
- Pad de borde 1
- Salida: 9 × 9

Normalmente se utilizan capas de convolución

con stride 1, filtros de tamaño F × F y

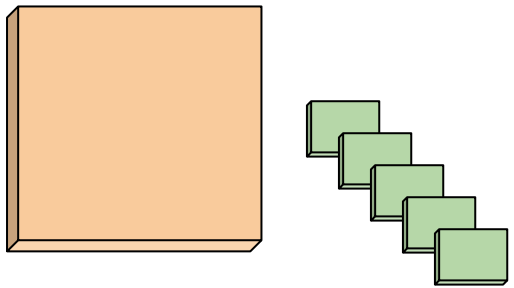
zero-padding de (F − 1)/2

para preservar el tamaño.



Ejemplo

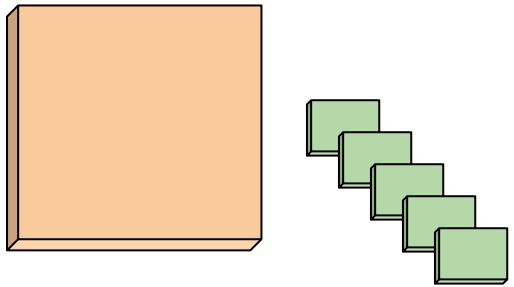
- Tamaño de entrada: 32x32x3
- Filtros:
  - Cantidad: 10
  - Tamaño: 5x5
  - Stride: 1
  - Pad: 2



- Tamaño de salida?

Ejemplo

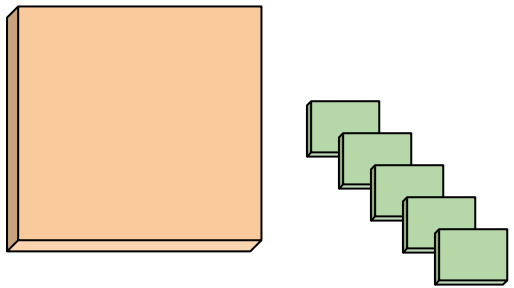
- Tamaño de entrada: 32 x 32 x 3
- Filtros:
  - Cantidad: **10**
  - Tamaño: **5**x**5**
  - Stride: **1**
  - Pad: **2**



- Tamaño de salida:
  - **(32 + 2*2 - 5)/1+1=32** espacial => 32 x 32 x **10**

Ejemplo

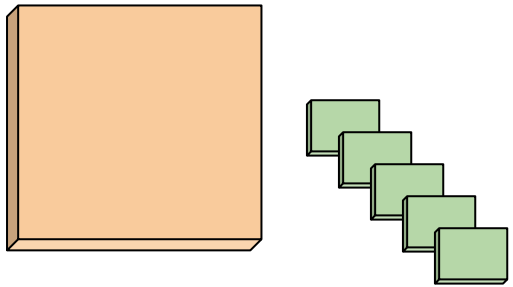
- Tamaño de entrada: 32x32x3
- Filtros:
  - Cantidad: 10
  - Tamaño: 5x5
  - Stride: 1
  - Pad: 2



- Número de parámetros en esta capa?

Ejemplo

- Tamaño de entrada: 32 x 32 x 3
- Filtros:
  - Cantidad: **10**
  - Tamaño: **5**x**5**
  - Stride: **1**
  - Pad: **2**



- Número de parámetros en esta capa?
  - Cada filtro tiene: **5x5x3+1**(bias) **= 76** parámetros => 76 x **10** = 760

Pooling layer

- Comprime (sub-muestreo) de la representación
- Opera en cada mapa de activación (canal) por separado

En resumen

- Las arquitecturas ConvNet son una lista de capas que transforman la imagen de entrada en un volumen de salida (con los puntajes de las clases)
- Hay un conjunto de capas comunes (Conv, ReLU, POOL, FC)
- Cada capa puede tener parámetros (Conv, FC) o no (Pool, ReLU)
- Cada capa puede o no tener hiperparametros adicionales (e.g. CONV/FC/POOL do, RELU doesn’t)

Ejemplo

- Clasificación CIFAR-10
- Arquitectura: [conv-relu-conv-relu-pool]x3-fc-softmax
  - 17 capas
  - 7000 parameters
  - 3x3 convolutions
  - 2x2 pooling

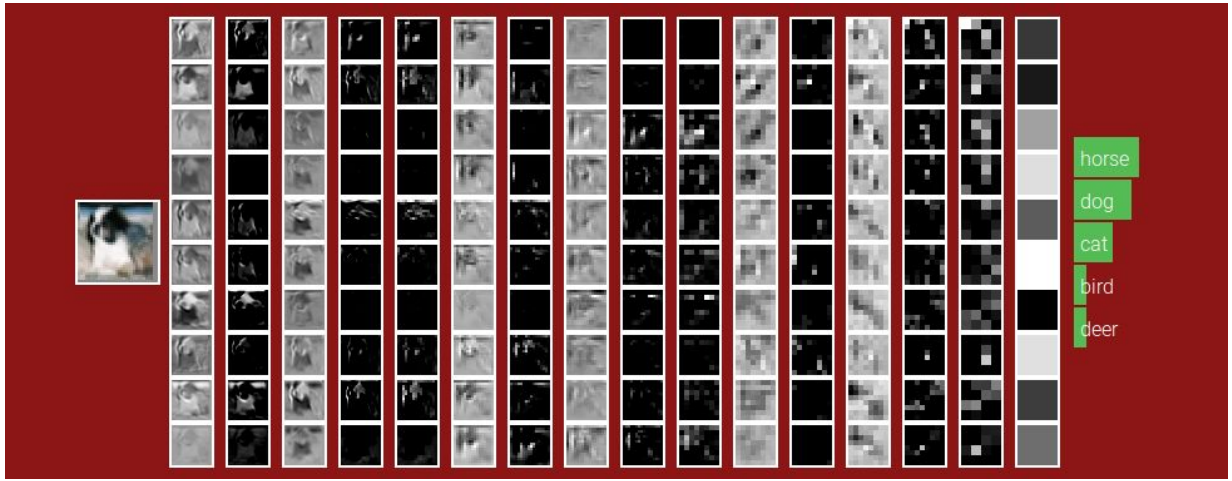
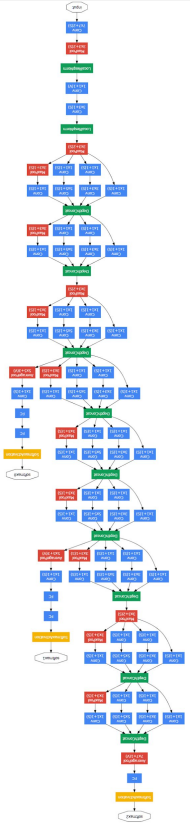
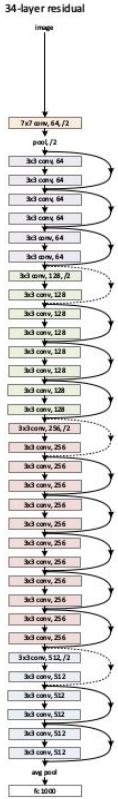


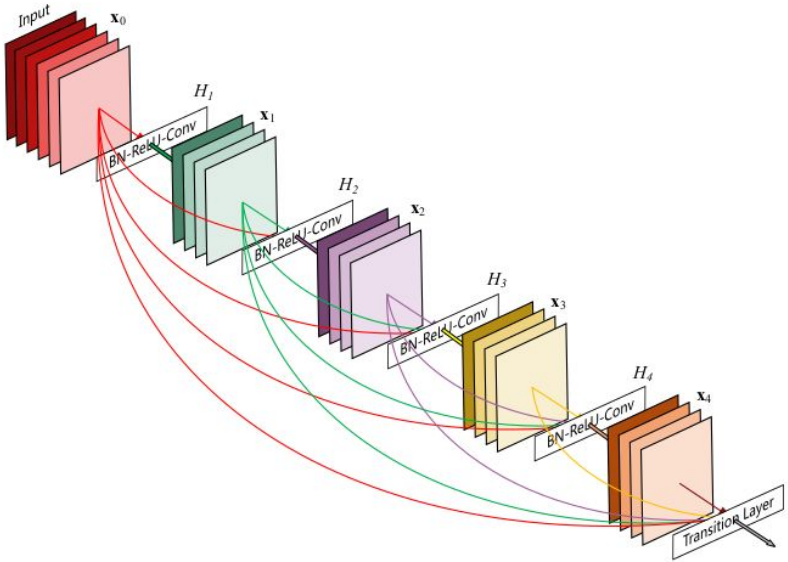
Imagen tomada de http://cs231n.stanford.edu/

Arquitecturas

VGG (2014) GoogLeNET(2014) RESNET(2015) DenseNet(2017)







Referencias

[1] CS231n Convolutional Neural Networks for Visual Recognition - Stanford CS class

[2] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[3] Chollet, Francois. Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek. MITP-Verlags GmbH & Co. KG, 2018.