Curso de Inteligencia Artificial para Ingenieros

Redes Neuronales Convolucionales

Prof. Carlos A. Toro N.
carlos.toro.ing@gmail.com
2022

Objetivos

Al finalizar esta presentación, usted podrá:

- ❑ Dominar la definición de algunos de los conceptos asociados a redes neuronales convolucionales
- ❑ Implementar redes neuronales convolucionales
- ❑ Conocer algunas de las arquitecturas de redes neuronales convolucionales más comunes.

Contenidos

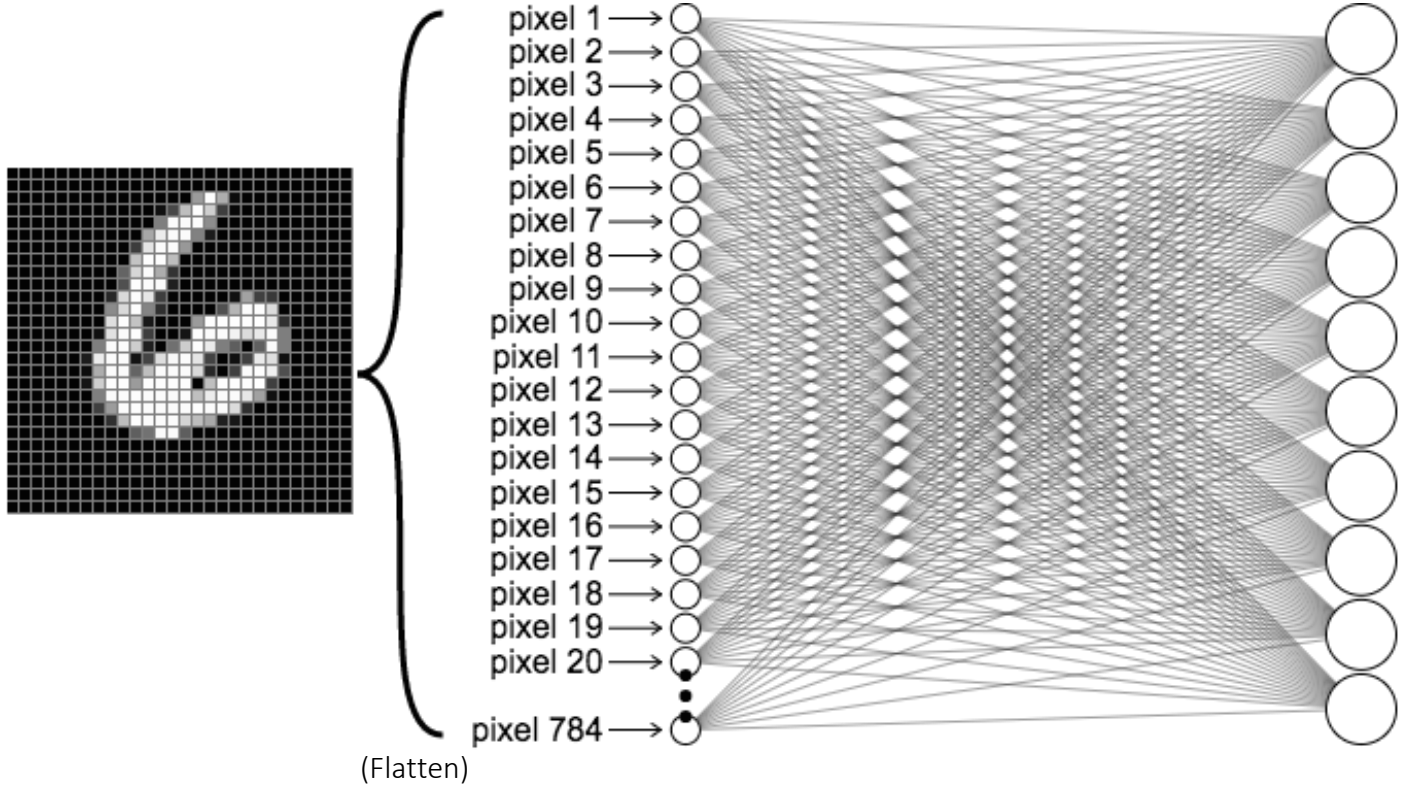
- Motivación
- Redes neuronales convolucionales
- Arquitecturas comunes

Motivación

Por qué no usar redes neuronales completamente conectadas para imágenes

Al usar redes completamente conectadas (*fully connected*) para trabajar con imágenes tenemos que:

- Conectan cada neurona en la primera capa oculta a cada uno de los píxeles de la imagen de entrada.
- No toma en cuenta la información espacial.
- Añade demasiados parámetros a estimar (pesos mas bias).
- Por ejemplo, imaginemos que tenemos una imagen en escala de grises de **28x28** pixeles (imagen derecha), luego, por cada neurona tendremos **28x28 = 784 conexiones** (una por pixel). Luego, si tuviéramos 10 neuronas en la primera capa oculta, inmediatamente tendríamos que aprender **784x10 = 7840 pesos!**.

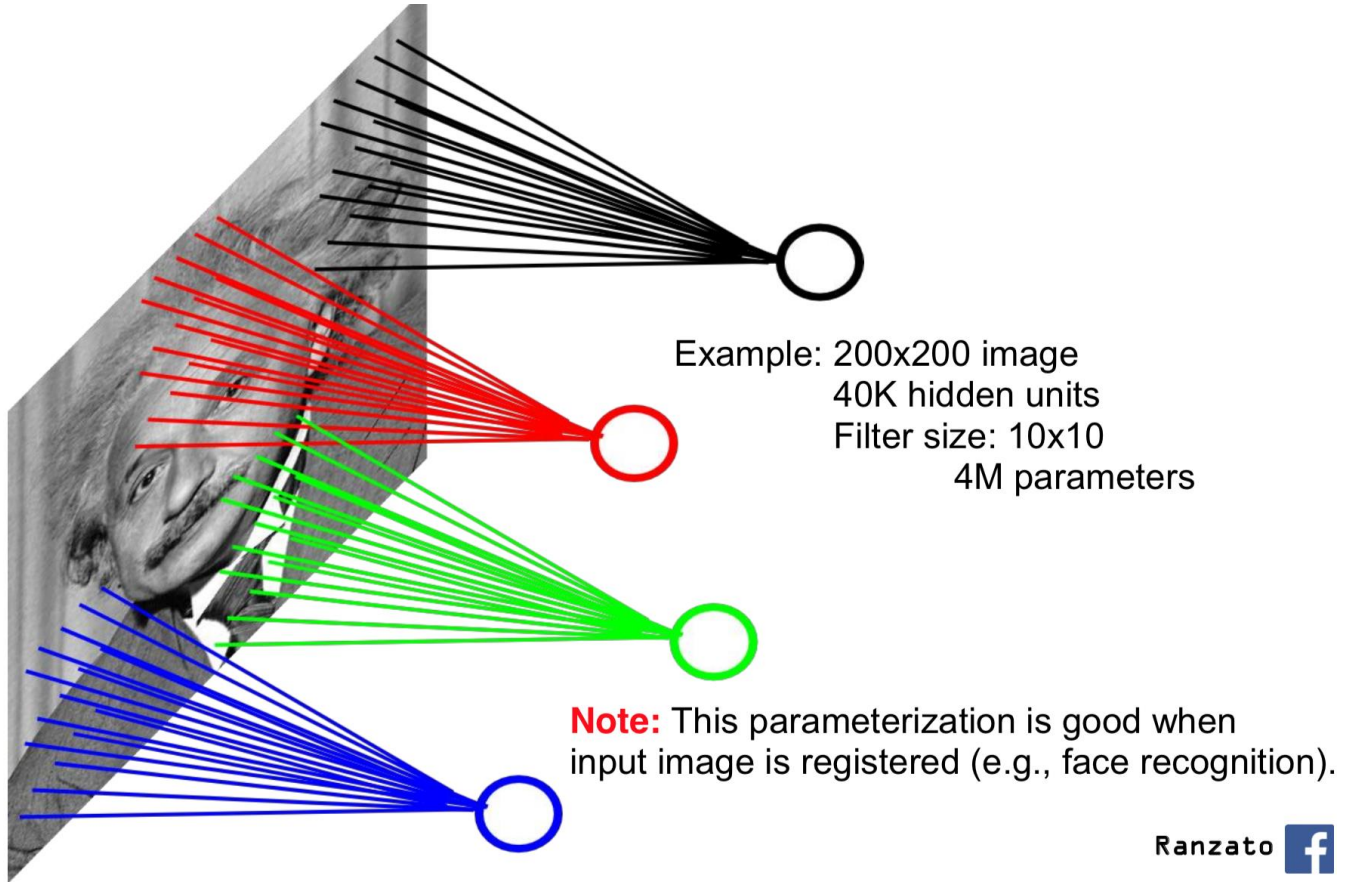


https://ml4a.github.io/ml4a/looking_inside_neural_nets/

Cómo podemos usar la estructura espacial en la entrada de la red?

Por qué no usar redes neuronales completamente conectadas para imágenes

- Una forma que se puede implementar para aprovechar mejor la información espacial es conectar porciones de la imagen de entrada a las neuronas en la capa oculta. Estas son las capas localmente conectadas.
- Cada neurona verá solo esa porción de la imagen de entrada en lugar de verla toda.
- Con esto logramos menos conexiones (y por lo tanto pesos) entre la entrada y la primera capa oculta.
- Lo que veremos a continuación, es una forma diferente, donde los parámetros de los filtros se comparten, reduciendo aún más la cantidad de pesos a aprender. Estas serán las redes convolucionales.



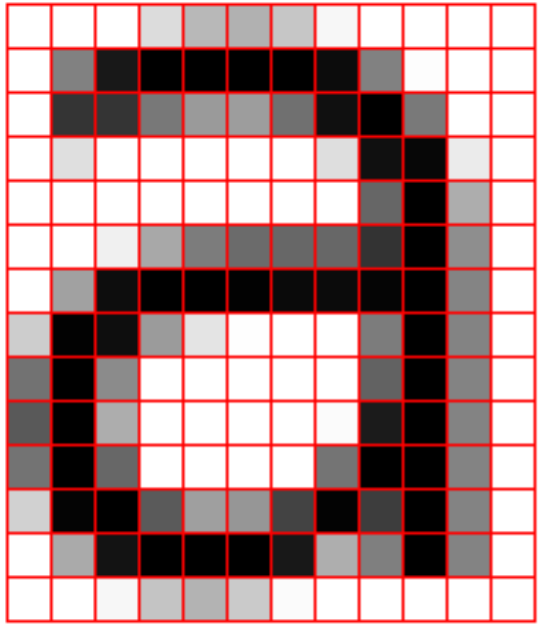
https://www.cs.toronto.edu/~lczhang/360/lec/w04/convnet.html

Redes Neuronales Convolucionales

Definición

Las Redes Neuronales Convolucionales (Convolutional Neural Networks, CNN o ConvNet) son un tipo de red neuronal artificial diseñadas para procesar información estructurada como arreglos de elementos, por ej. imágenes, arreglo de píxeles.

a

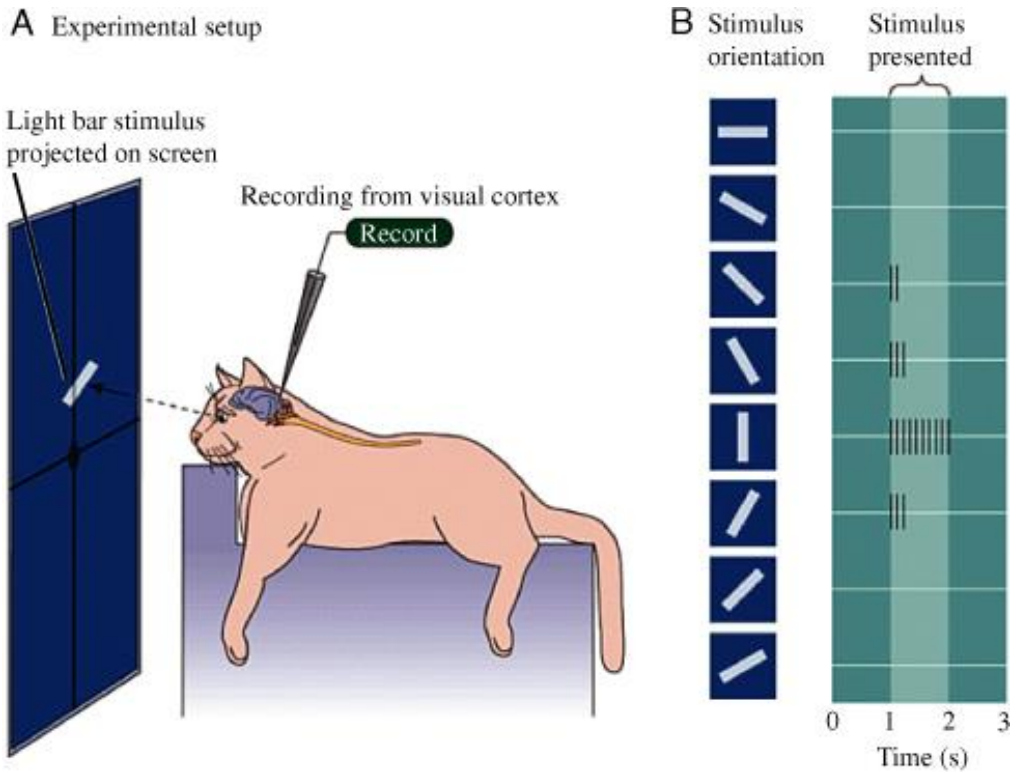


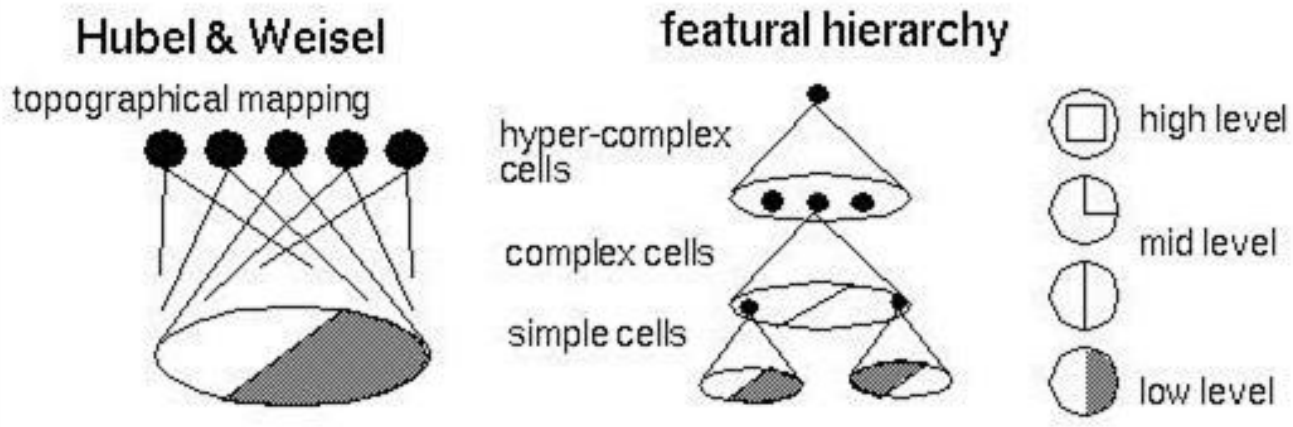
1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Estas redes son ampliamente usadas en visión por computador porque evitan usar pre-procesamientos complejos en imágenes y en su lugar, reciben como entrada la imagen original directamente.

Definición

En los 60s, Hubel y Wiesel estudiaron la respuesta eléctrica de la corteza visual de gatos ante estímulos con patrones simples en una pantalla de TV. Ellos encontraron una estructura de red neuronal única que podría simplificar la de las redes completamente conectadas. Luego, ellos propusieron las CNN.

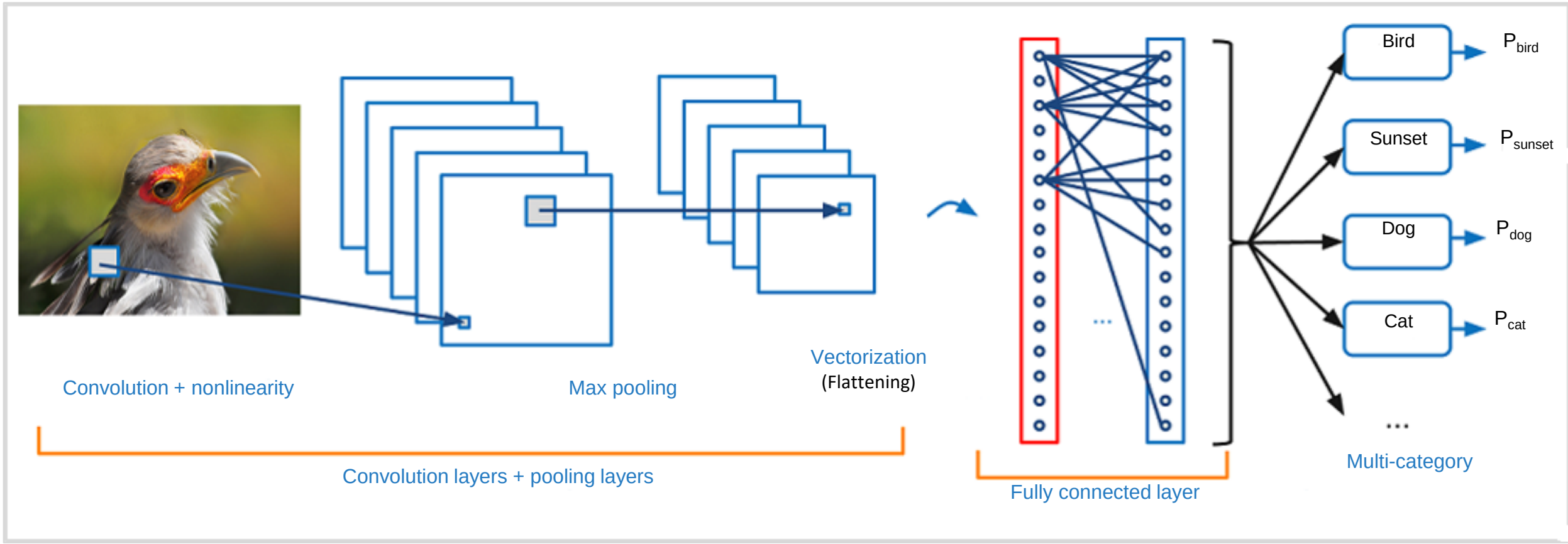




Las neuronas en la corteza visual se organizan de forma jerárquica.

fuente

Arquitectura típica de una CNN artificial



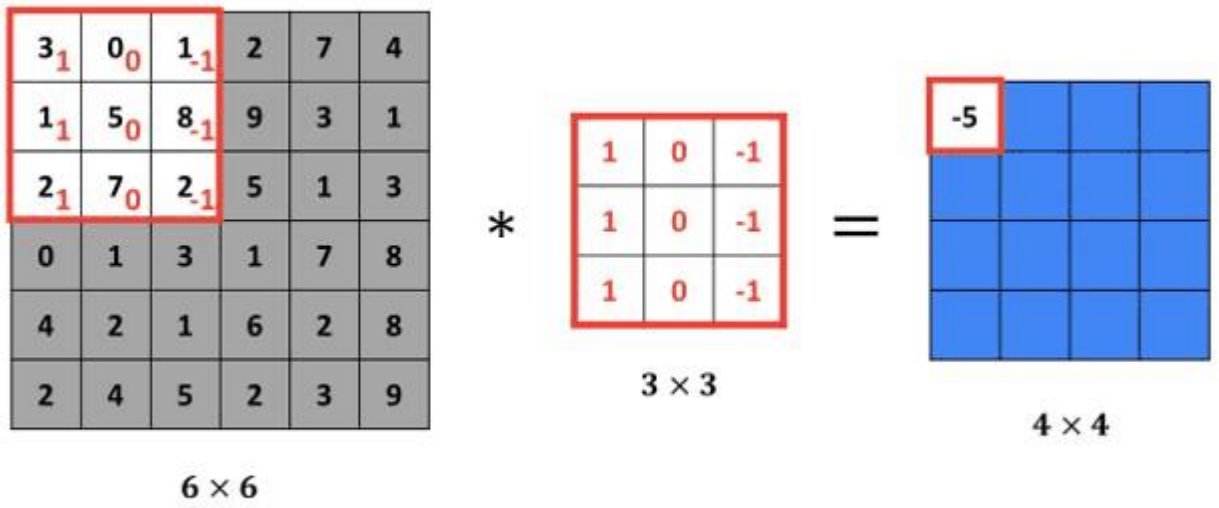
Feature extraction + classification

fuente

Las CNN sobresalen en análisis de imágenes. Incluyen en su estructura básica capas convolucionales, capas de reducción de dimensionalidad o *pooling* y capas de redes completamente conectadas (fully connected).

Recordemos, qué es la convolución?

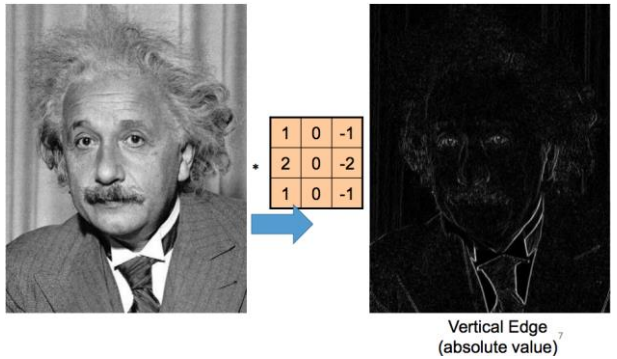
- En el sentido del procesamiento digital de imágenes, la convolución es una operación que transforma una imagen al desplazar sobre ella una más pequeña (llamada filtro o kernel), multiplica los valores de los píxeles por los del filtro y finalmente suma los valores (suma ponderada)*.



$$3 \times 1 + 1 \times 1 + 2 \times 1 + 0 \times 0 + 5 \times 0 + 7 \times 0 + 1 \times -1 + 8 \times -1 + 2 \times -1 = -5$$

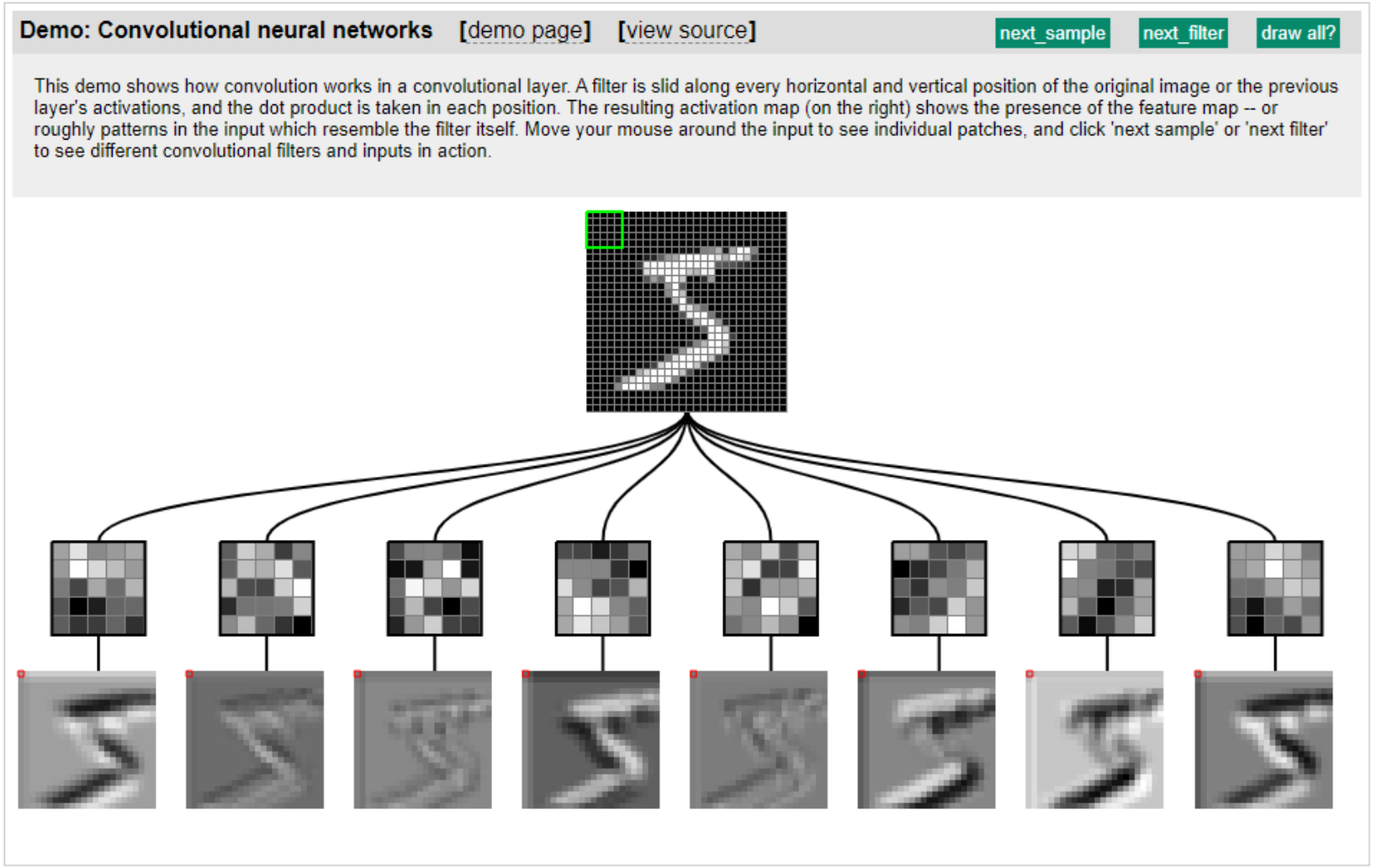
fuente

Estos filtros tienen la misión de extraer características desde las imágenes de entrada, ej. bordes.



Nota *: En realidad, en redes neuronales convolucionales, la operación implementada es la de correlación, por qué?

Demo: CNNs



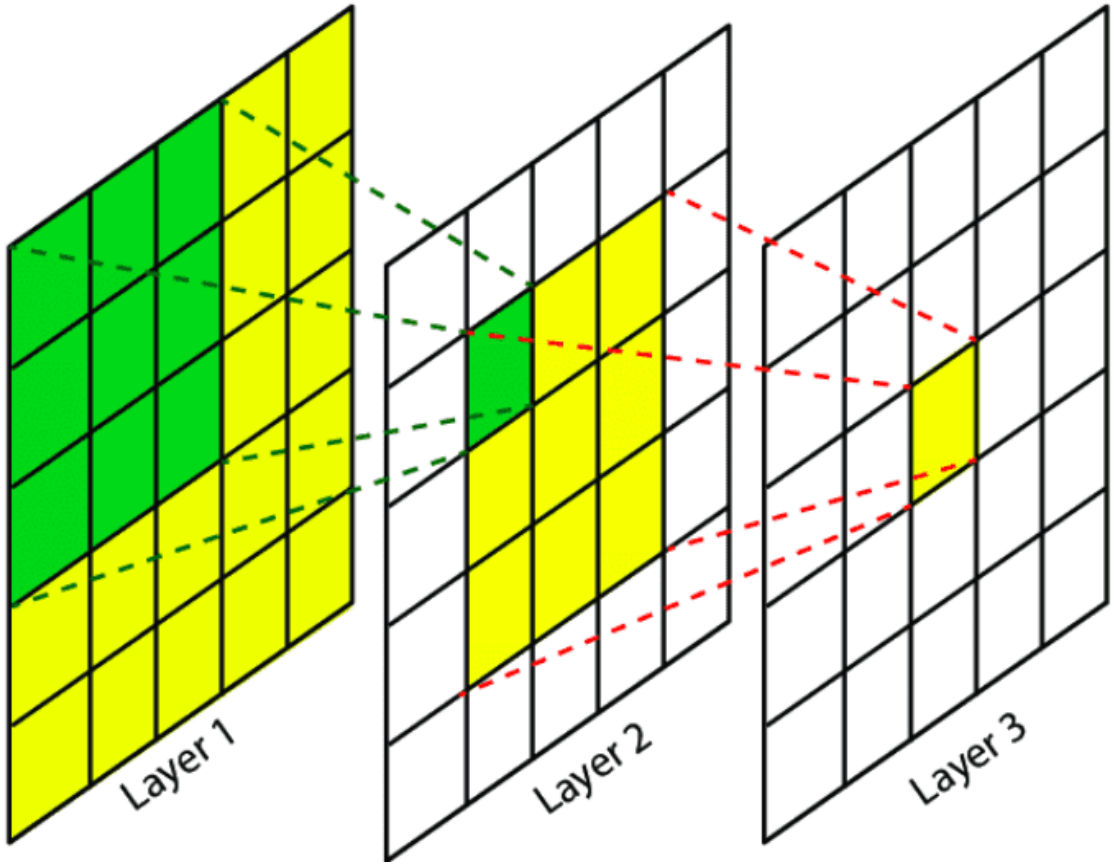
https://ml4a.github.io/ml4a/convnets/

Filtros o kernels

Mapas de respuesta, mapas de activación o mapas de características generados al aplicar los filtros.

Principales conceptos

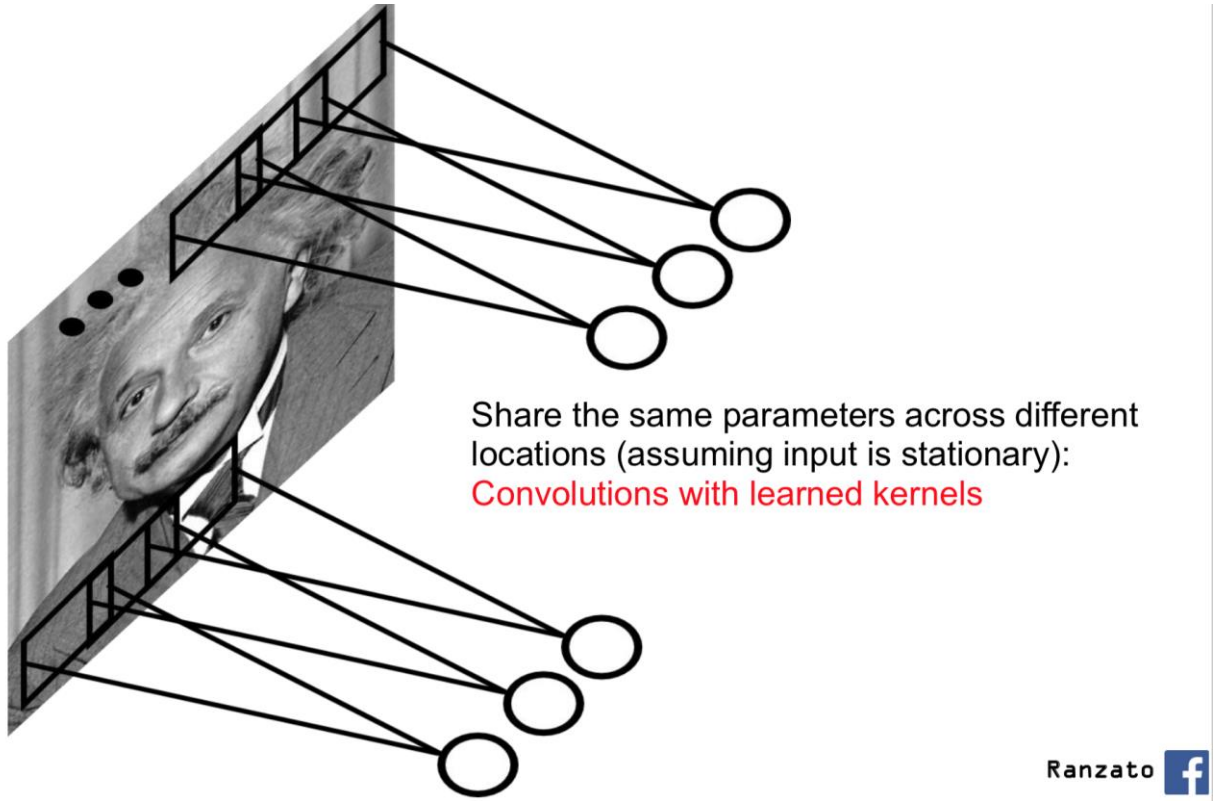
- Campo receptivo local (*local receptive field*): Se considera generalmente que la percepción humana del mundo exterior es de lo local a lo global. Las correlaciones espaciales entre pixeles locales de una imagen son más cercanas que con aquellos pixeles más distantes. Por consiguiente, cada neurona no necita conocer la imagen global, solo necesita conocer la imagen local. La información local es combinada a niveles más altos para generar la información global.



https://theaisummer.com/receptive-field/

Principales conceptos

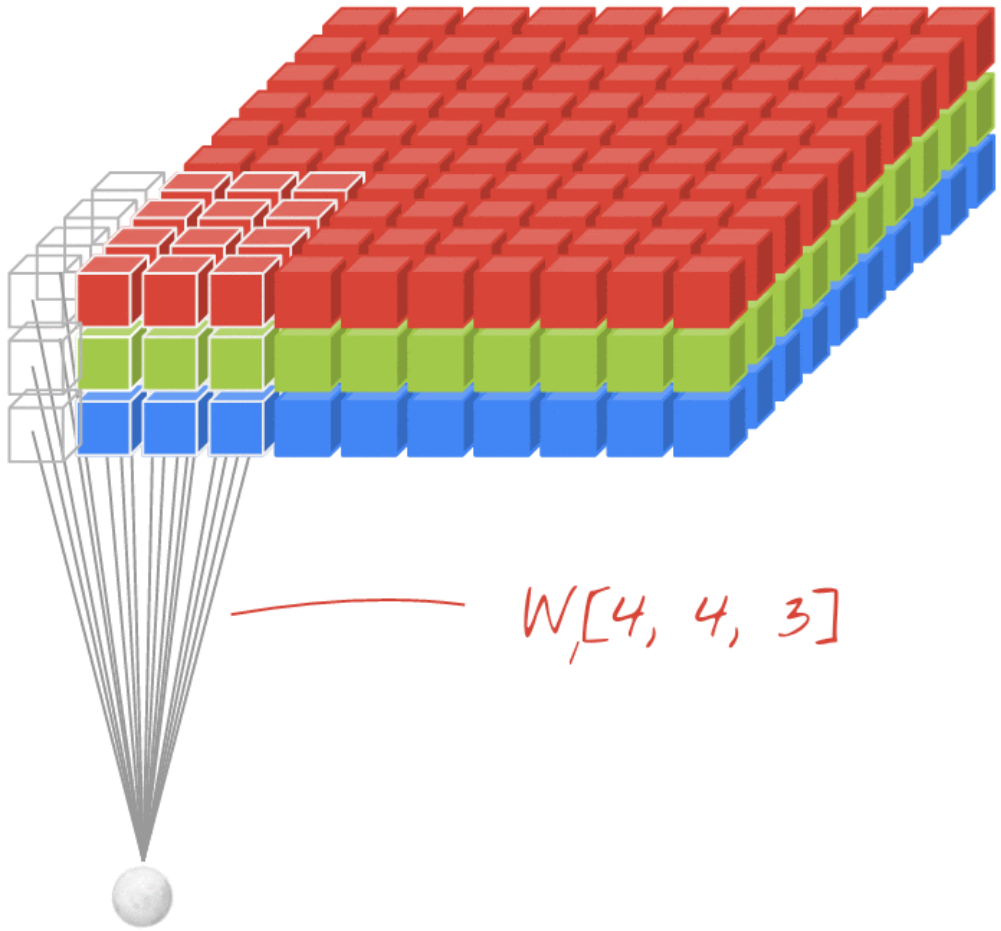
- **Parámetros compartidos (parameter sharing):** Este esquema se usa en las capas convolucionales para controlar el número de parámetros. Uno o más filtros/kernels pueden ser usados para recorrer las imágenes de entrada. Los parámetros que contienen los filtros son pesos. En una capa recorrida por filtros, cada filtro usa los mismos parámetros durante la operación de ponderación para recorrer toda la imagen o capa de entrada. El compartir parámetros significa que cuando cada filtro recorre una imagen entera, los parámetros del filtro son fijados.



https://www.cs.toronto.edu/~lczhang/360/lec/w04/convnet.html

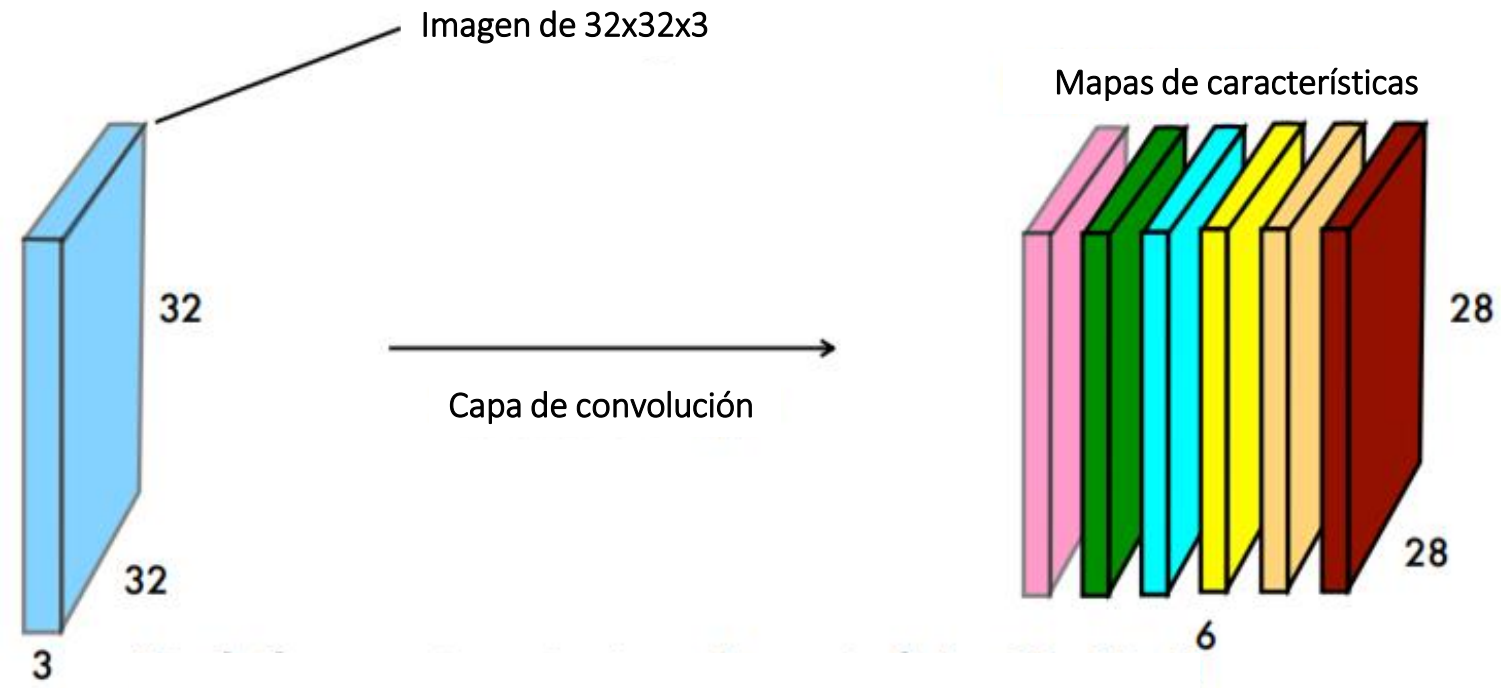
Convoluciones con múltiples canales de entrada/salida

- En el caso de imágenes a color, el kernel será un tensor de dimensión 3 en lugar de dos. Este kernel se moverá a través de la entrada igual que antes, y aplicará operaciones de sumas ponderadas en cada región.
- **Los canales de entrada** en el caso de imágenes a color son 3, por lo que para un kernel que analiza regiones de 4x4 píxeles, su dimensión será 4x4x3.
- Qué sucede si queremos detectar múltiples características? Deberíamos aprender múltiples filtros a la vez sobre la misma entrada.
- Luego, tendremos múltiples **canales de salida o mapas de característica de salida**, en el ejemplo: 2.



Convoluciones con múltiples canales de entrada/salida

Ejemplo: en la figura de abajo, tenemos una imagen de entrada de 32x32x3 (imagen a color), a la cuál queremos extraer 6 mapas de características durante el entrenamiento, el tamaño de los filtros es por ejemplo de 5x5x3 y necesitaremos aprender 6 en total. Cada uno generará un mapa de características diferente.

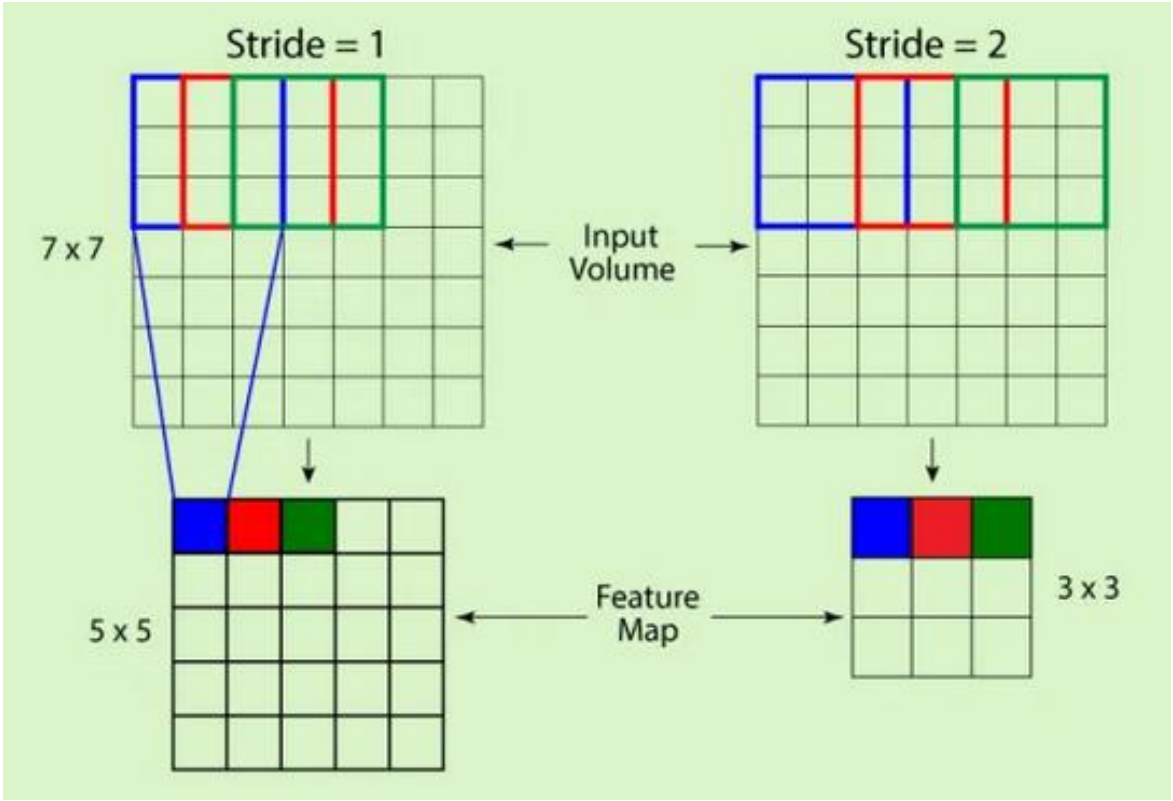


Fuente

Por qué el tamaño de cada mapa de características de salida es de 28x28?

Conceptos asociados a las capas convolucionales: Stride o paso

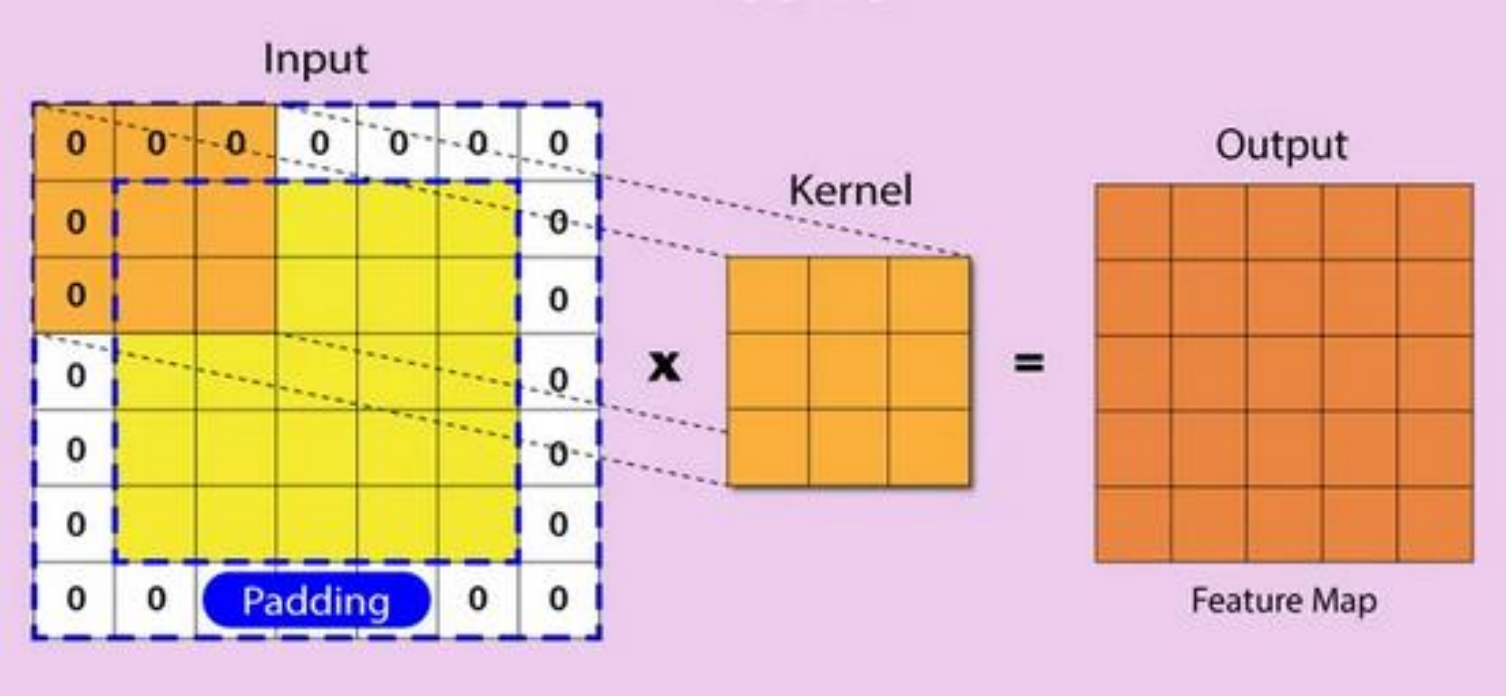
Concepto de stride (paso): Controla como los filtros se convolucionan con la entrada, i.e. el número de pixeles que se corre el filtro al recorrer la matriz de entrada. Si el paso o stride es 1, el filtro se mueve 1 pixel a la vez y si el stride es 2, el filtro se mueve 2 pixeles a la vez. Valores más grande de stride resultará en salidas mas pequeñas y viceversa.



Fuente

Conceptos asociados a las capas convolucionales: Padding o relleno

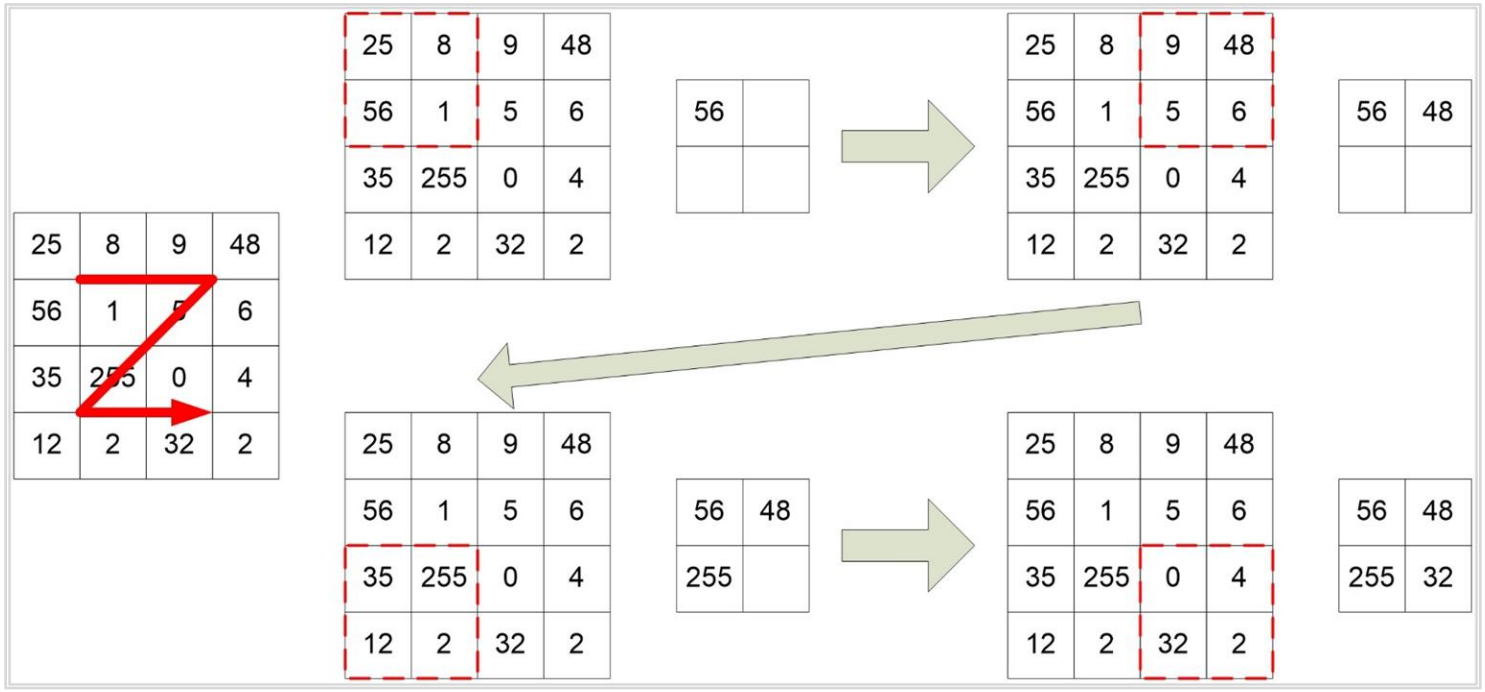
Padding (relleno): Es usado para que la imagen de salida tenga la misma dimensión que la entrada, esto se logra agregando ceros a la matriz o imagen de entrada (alrededor de los bordes externos). Esto le da mayor espacio al kernel o filtro para cubrir la imagen. Debido a esto, la información de los bordes de la imagen también se preserva.



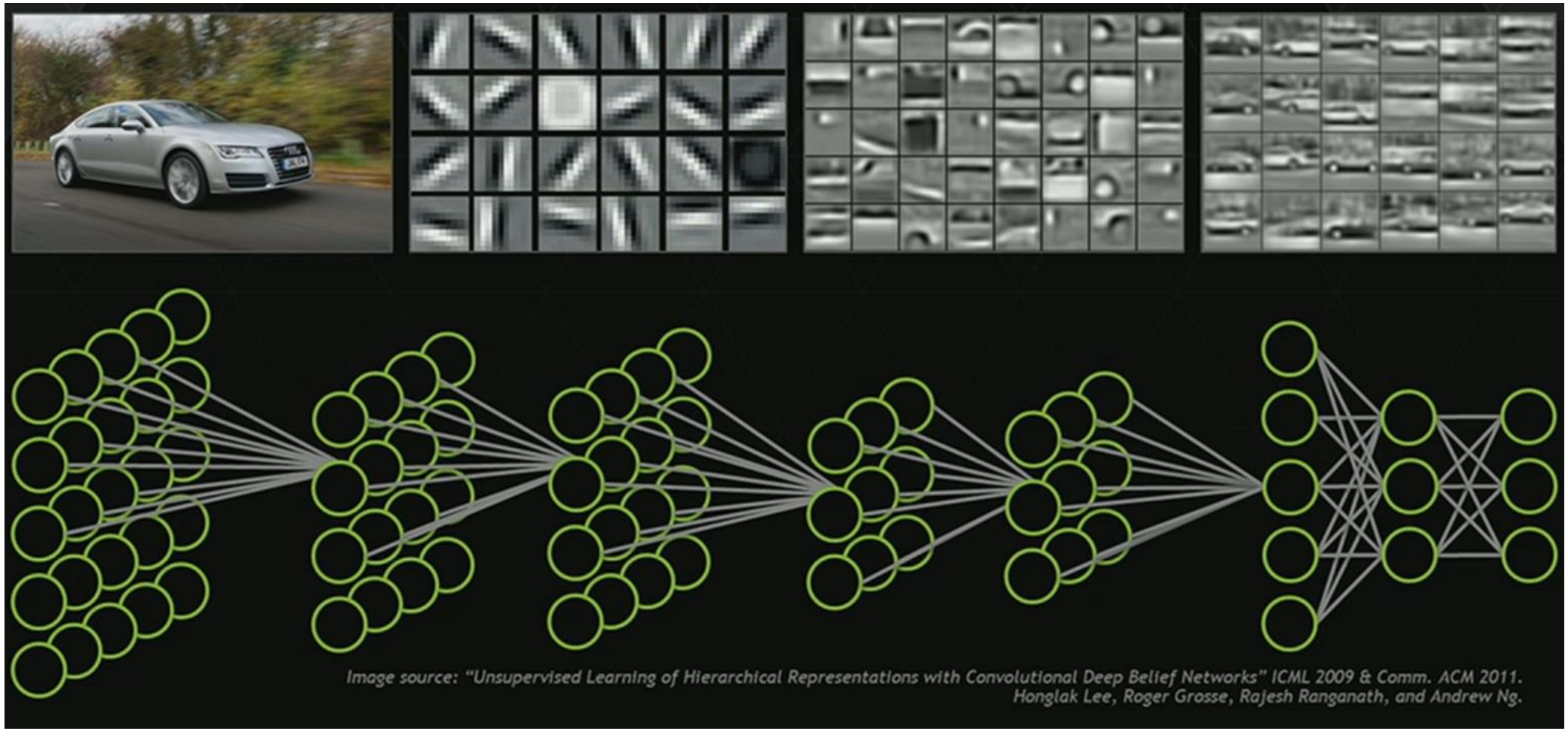
Fuente

Capa de agrupamiento (*Pooling layer*)

Esta operación combina unidades cercanas para reducir el tamaño de la entrada a la capa siguiente, reduciendo su dimensión. Agrupamientos comunes incluyen: agrupamiento por máximos (max pooling), agrupamiento por promedio (average pooling). Cuando se usa max pooling, se selecciona el valor máximo en una pequeña área cuadrada como valor representativo de esa área, mientras que el valor promedio se selecciona como representativo cuando se usa *average pooling*. El lado de esta pequeña área se llama *pool window size* (tamaño de la ventana de agrupamiento). La siguiente figura muestra esta operación usando una ventana de tamaño 2 y *stride* de 2.

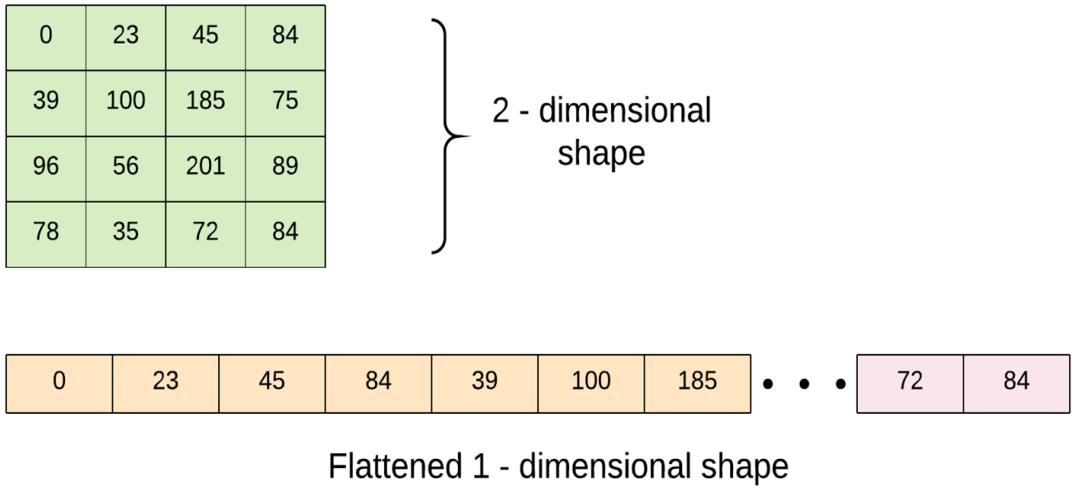


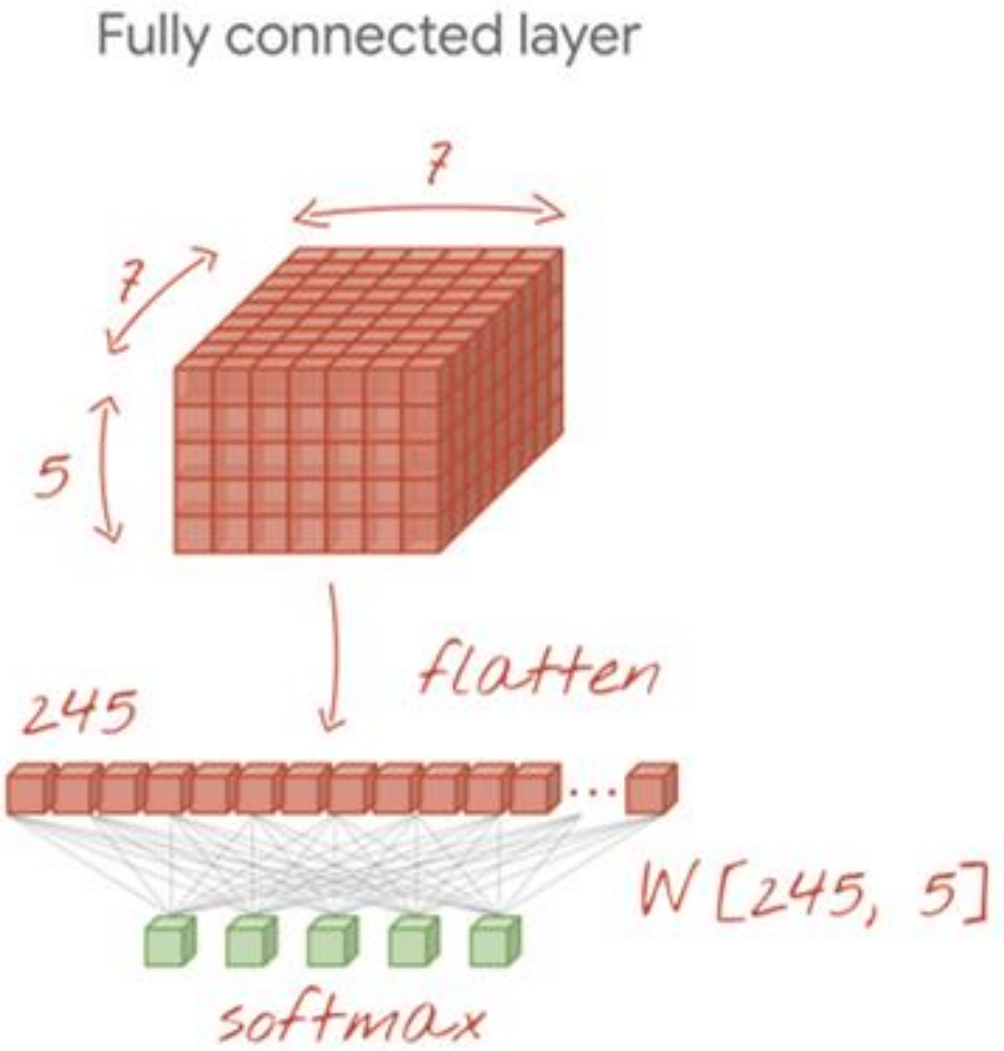
Extracción jerárquica-composicional de características



Capa completamente conectada

Luego de las etapas convolucionales, tenemos una red completamente conectada que es esencialmente un clasificador (generalmente). Las características extraídas en las etapas convolucionales y de agrupamiento son estiradas o aplanadas (*flattening*) para que entre a la capa completamente conectada y genere a la salida o resultados de clasificación (o regresión).





https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist#0

Arquitecturas comunes

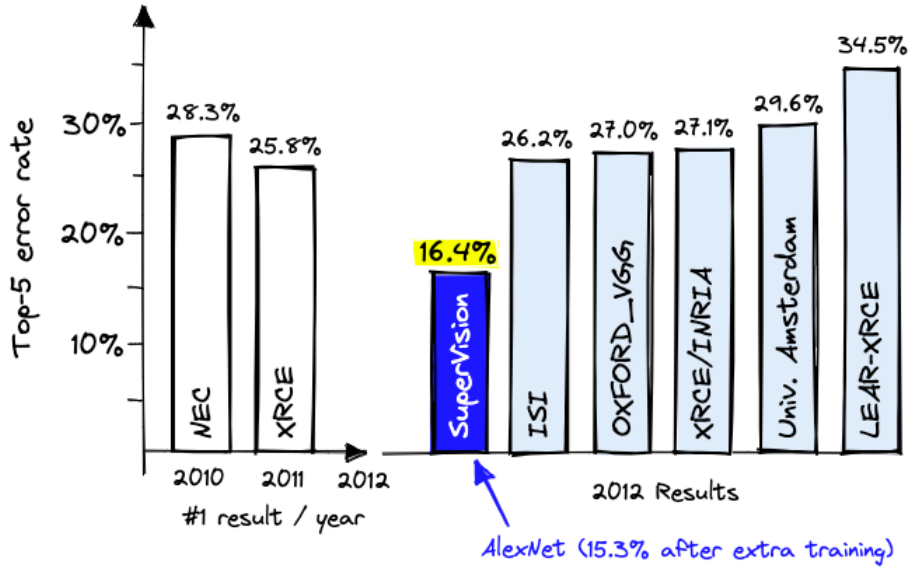
Competencia ImageNet

- Millones de imágenes (~14millones)
- Miles de categorías de objetos (~21mil)
- El objetivo es clasificar cuál objeto está presente en una imagen.

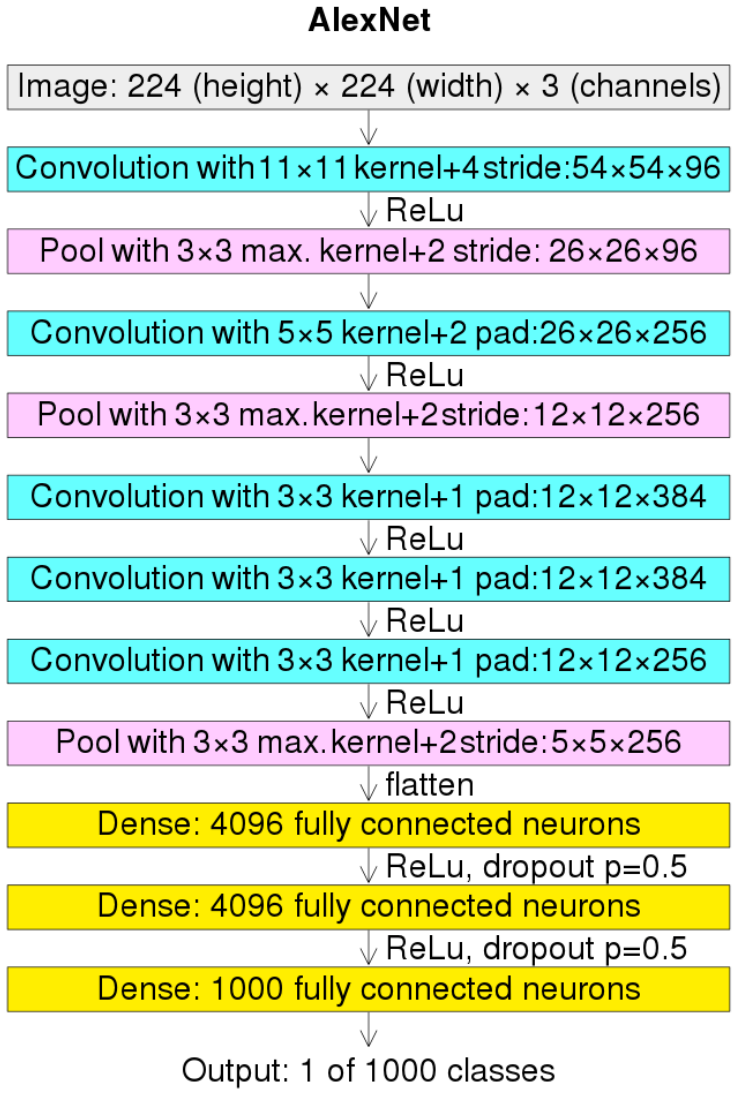
https://paperswithcode.com/dataset/imagenet

AlexNet (2012)

- Esta red fue propuesta por Alex Krizhevsky en colaboración con Ilya Sutskever y Geoffrey Hinton (Guía doctoral de Krizhevsky) (ref.)
- La red alcanzó un error top-5 de 15.3% en la competencia del ImageNet en 2012, un hito respecto a las competiciones de años anteriores e inició la era del *Deep learning*.

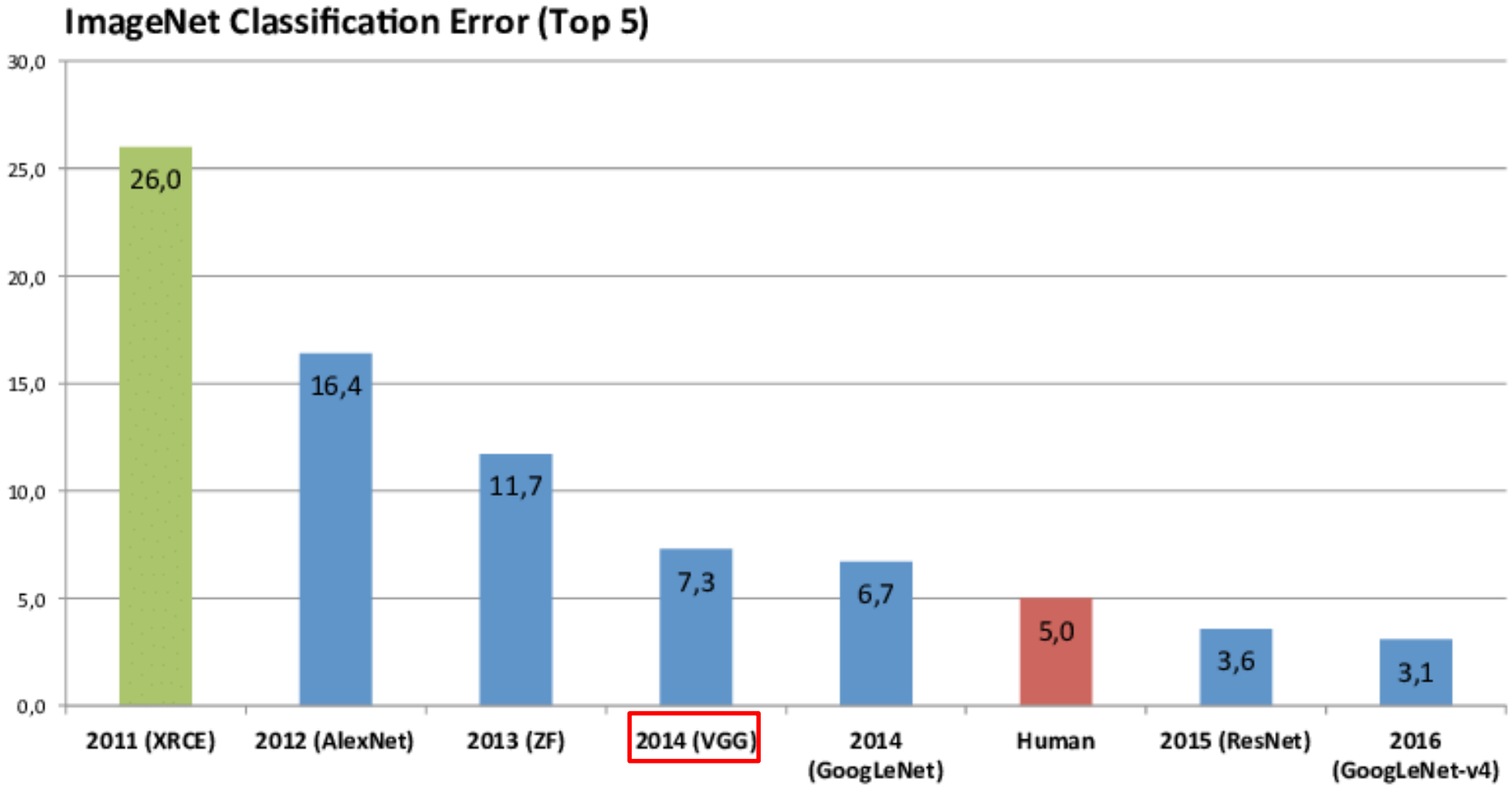


(Fuente)



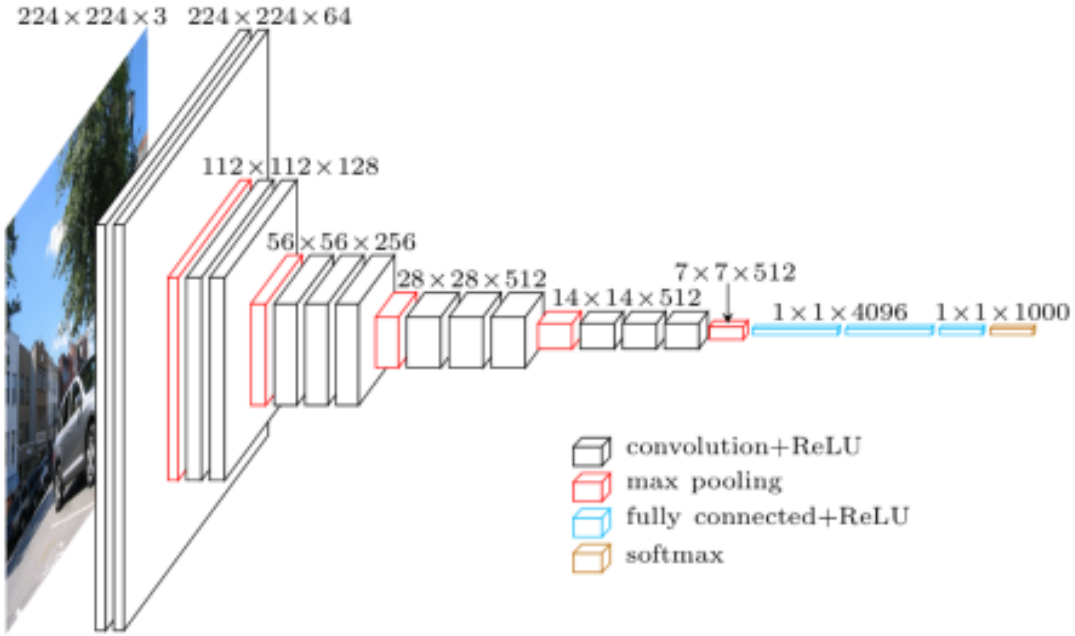
(Fuente)

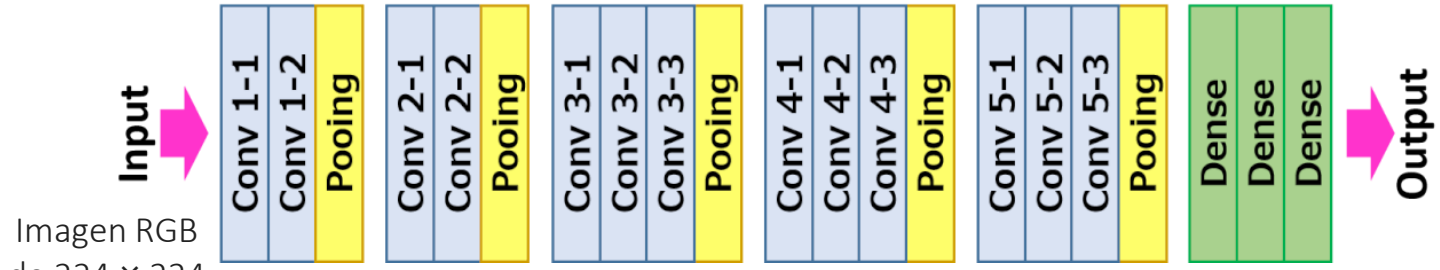
Resultados ImageNet (2016)



VGG Net (2014)

- Desarrollada por Simonyan y Zisserman para la competición ImageNet 2014.
- Posee 16 capas sin incluir las de *pooling*
- Solo convoluciones de 3x3
- 138 millones de parámetros

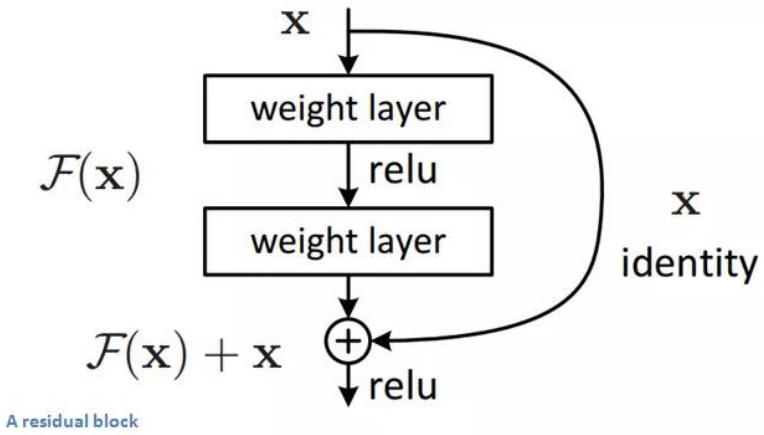




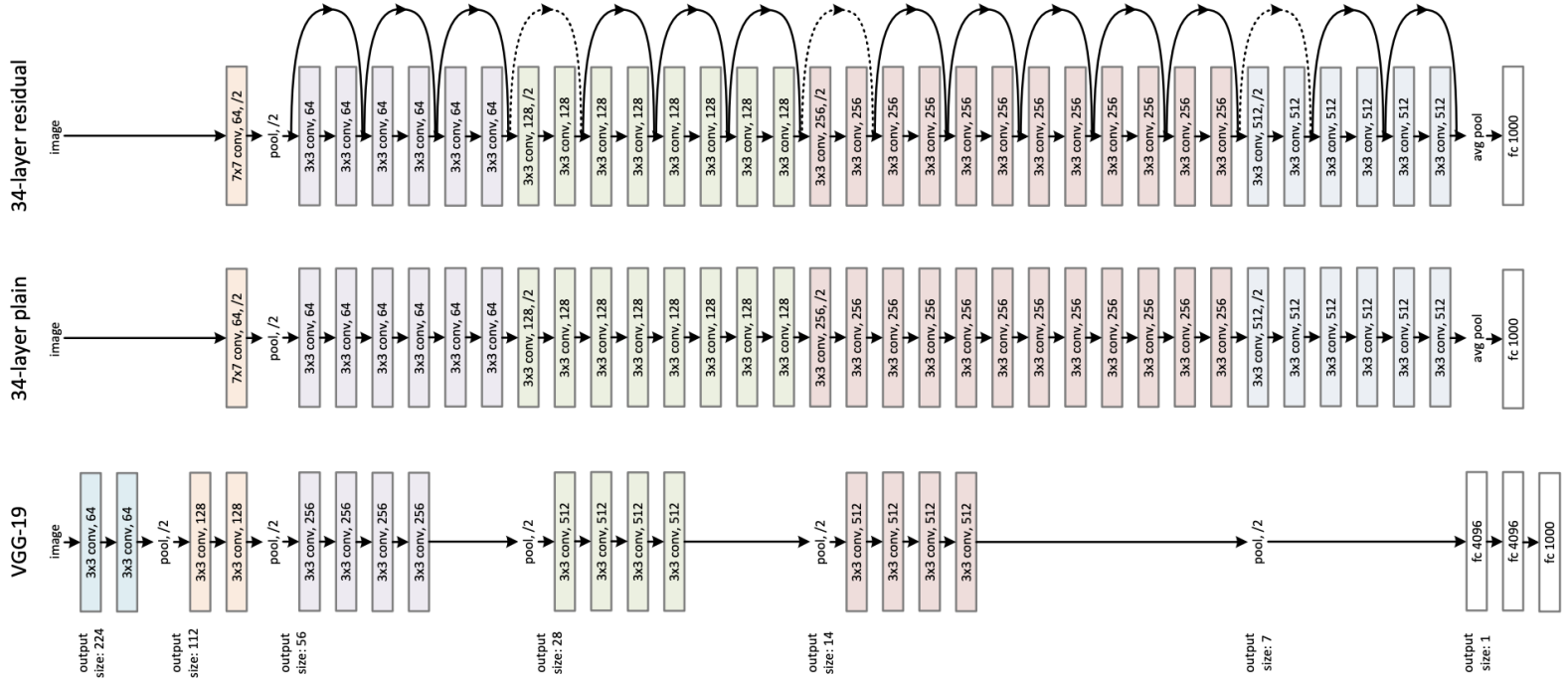
(Fuente)

Residual Networks (ResNets)

- Idea simple: Añadir un “salto” o conexiones “residuales” entre capas.
- Fue propuesta por investigadores de Microsoft.
- Permite modelos mucho mas profundos con cientos de capas.
- Componente clave de muchas CNNs del estado del arte.



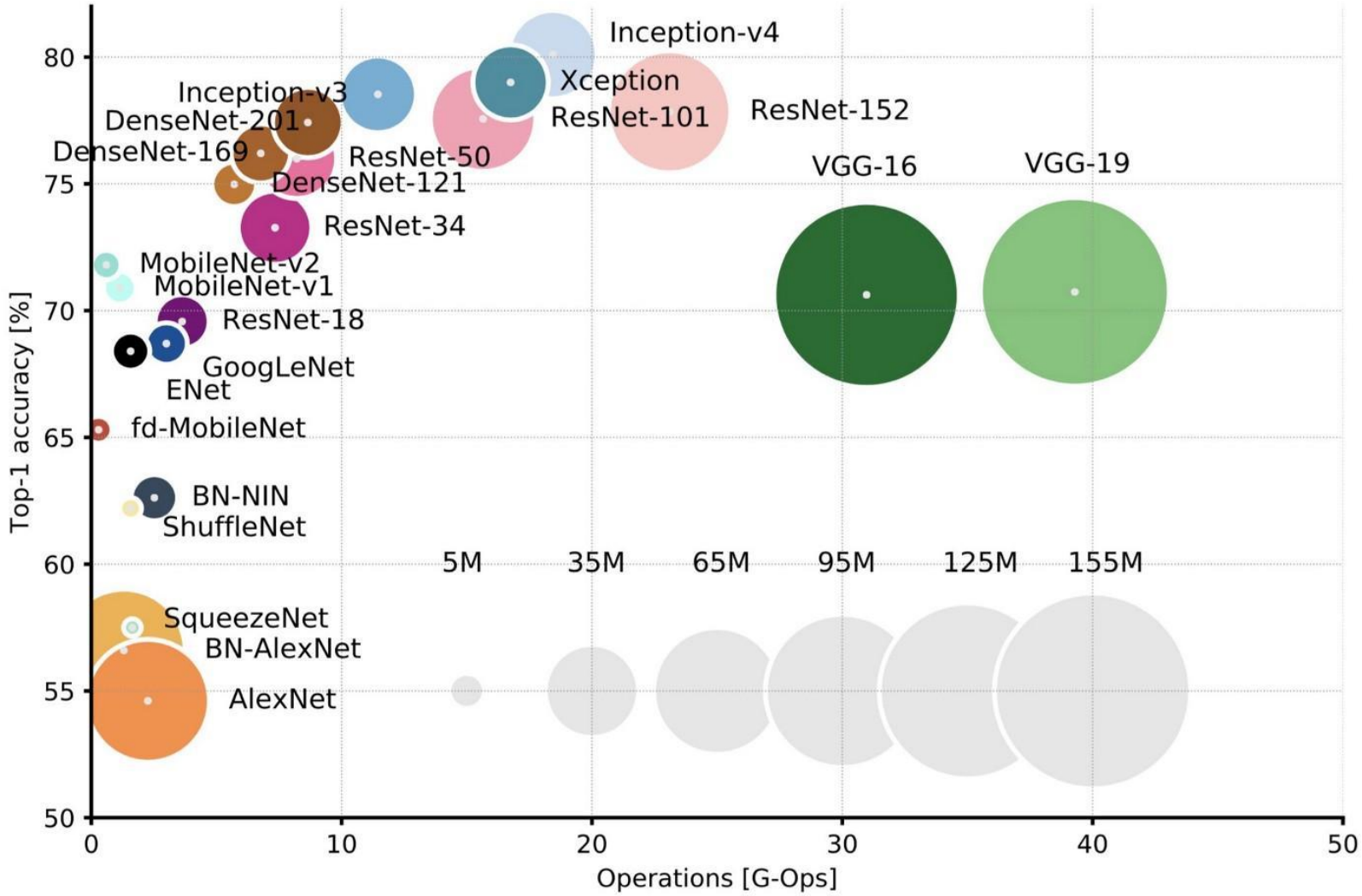
A residual block



(Fuente)

Existen diferentes versiones de estos modelos ResNet, por ejemplo, en Tensorflow Hub se encuentran pre-entrenados en ImageNet las siguientes versiones: Resnet50, Resnet101 y Resnet152.

Comparación de arquitecturas comunes de CNNs



Top1 vs. operations, size ∝ parameters. Top-1 one-crop accuracy versus amount of operations required for a single forward pass. (Fuente)