
Redes Neuronales Convolucionales

— Aprendizaje Automático
Aplicado —

¿Qué es una Red Neuronal Convolutiva (CNN)?

- Tipo de red neuronal profunda
- Diseñada para procesar datos bidimensionales
- Inspiradas en el funcionamiento de las neuronas de la corteza visual primaria del cerebro

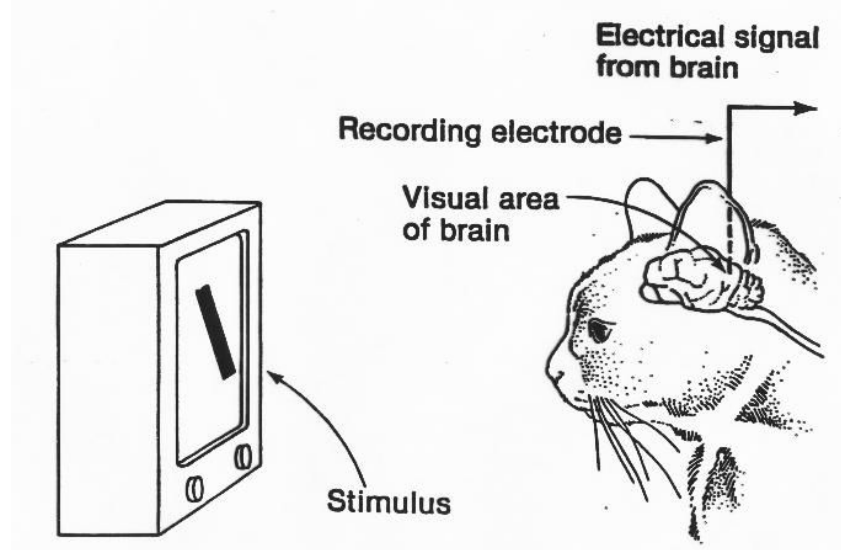
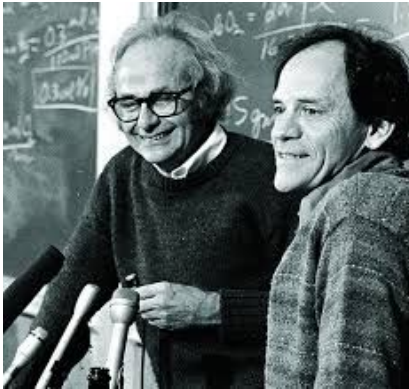
Un poco de historia

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)

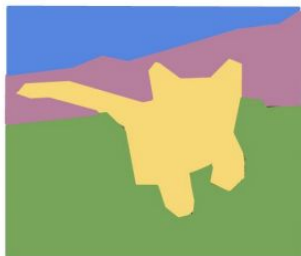


¿Para qué se usan las CNNs?

- Clasificación de imágenes
 - Detección de objetos
 - Segmentación semántica
 - Reconocimiento facial
 - Procesamiento de video
 - Reconocimiento de escritura a mano
 - Restauración de imágenes
- y mucho más...
-

Aplicaciones

Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

Classification



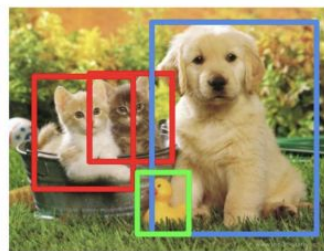
CAT

Classification + Localization



CAT

Object Detection



CAT, DOG, DUCK

Instance Segmentation



CAT, DOG, DUCK

Single object

Multiple objects

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.

Aplicaciones

Restauración de imágenes

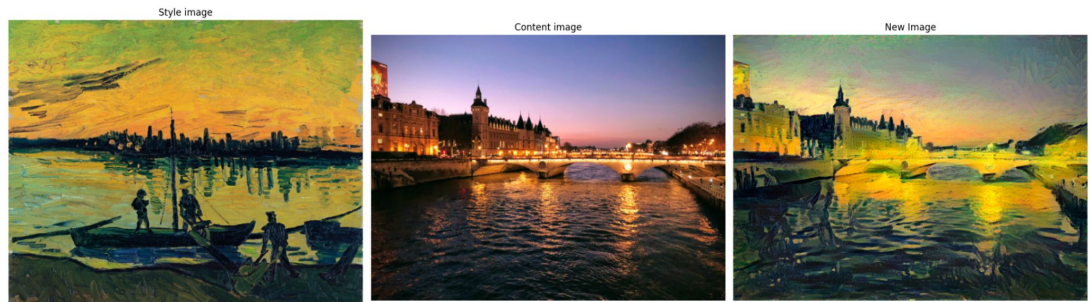


<https://storimaging.github.io/>

Síntesis de textura



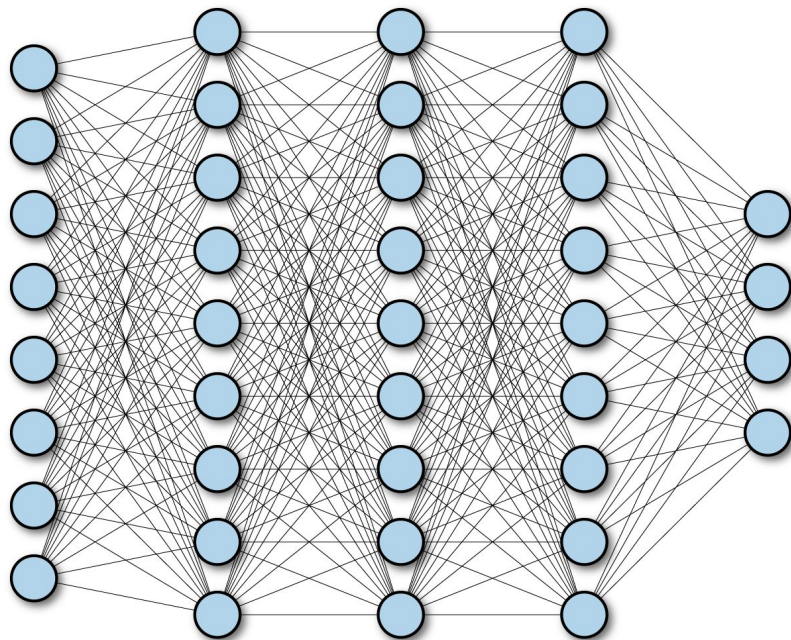
Transferencia de estilo



¿Por qué no usamos Redes fully-connected?

Computer vision - fully connected

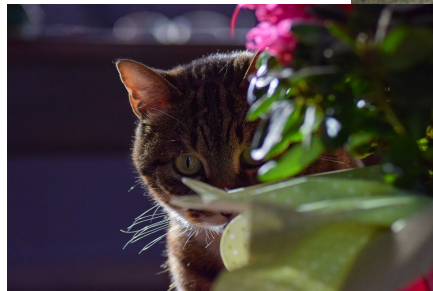
- supongamos imágenes a color de 128x128 como entrada:
 - Capa de entrada de dimensión 49k!
- Si además empezamos a apilar capas fully connected, rápidamente se alcanzan cantidades de parámetros difíciles de manejar
 - Demasiados parámetros: sobreajuste
 - Más probable caer en mínimos locales
 - Requiere mucho poder de cómputo



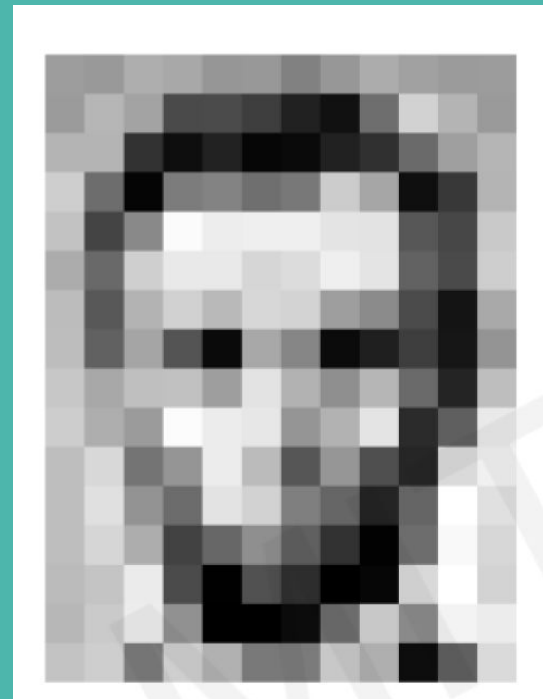
Computer vision



Computer vision - invariante a escala, rotación, etc



Antes de seguir... ¿Qué es una imagen?



¿Qué es una imagen?



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	83	17	110	210	180	154
180	180	50	14	94	6	10	83	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	106	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	90	2	109	249	215
187	196	238	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

¿Qué es una imagen?



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	94	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	56	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	9	6	217	255	211
183	202	237	145	0	9	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

What the computer sees

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	94	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	56	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	9	6	217	255	211
183	202	237	145	0	9	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

An image is just a matrix of numbers $[0,255]$!
i.e., $1080 \times 1080 \times 3$ for an RGB image

Convolución de imágenes

Convolución en imágenes

- Operación lineal entre una imagen y un filtro
- La salida es una nueva imagen
- El valor de cada píxel de la imagen de salida es la suma ponderada de los píxeles de la imagen de entrada y un núcleo de convolución:

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

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

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

Demo online - Image Kernels by Victor Powell

<https://setosa.io/ev/image-kernels/>

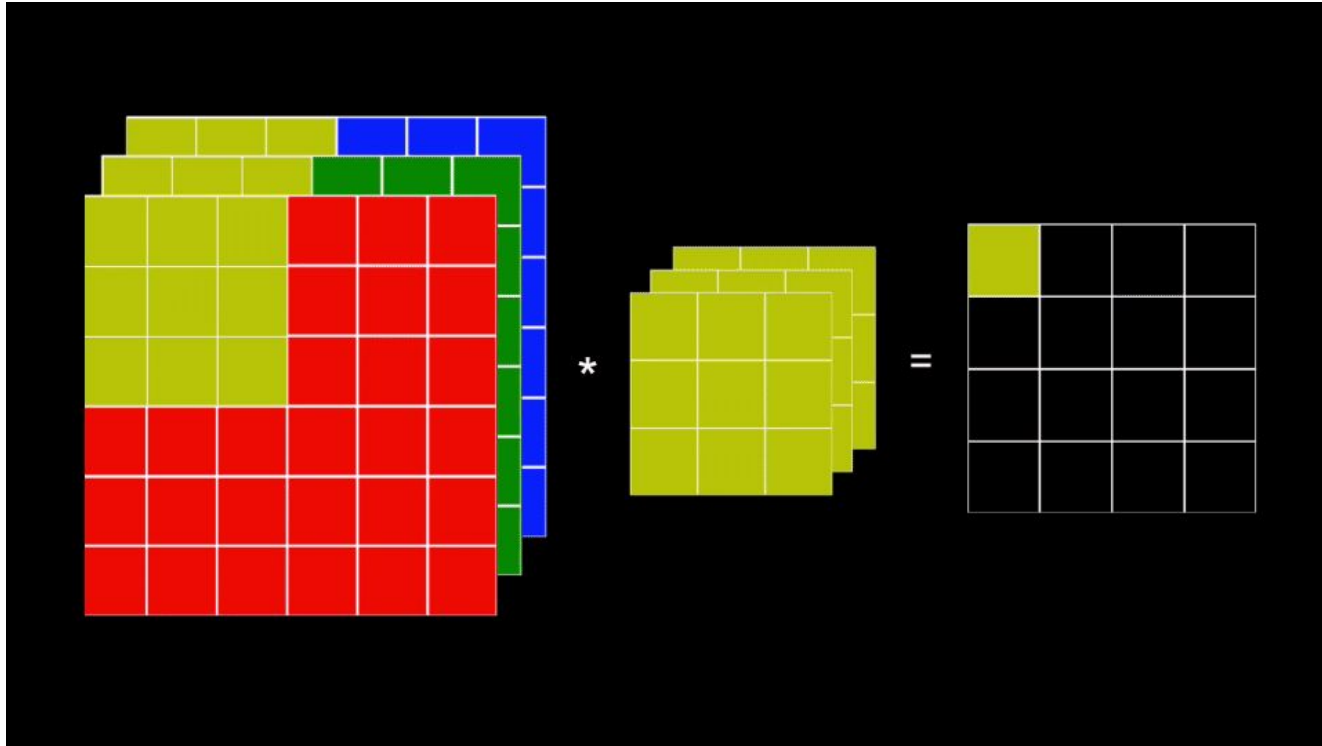
input image

output image

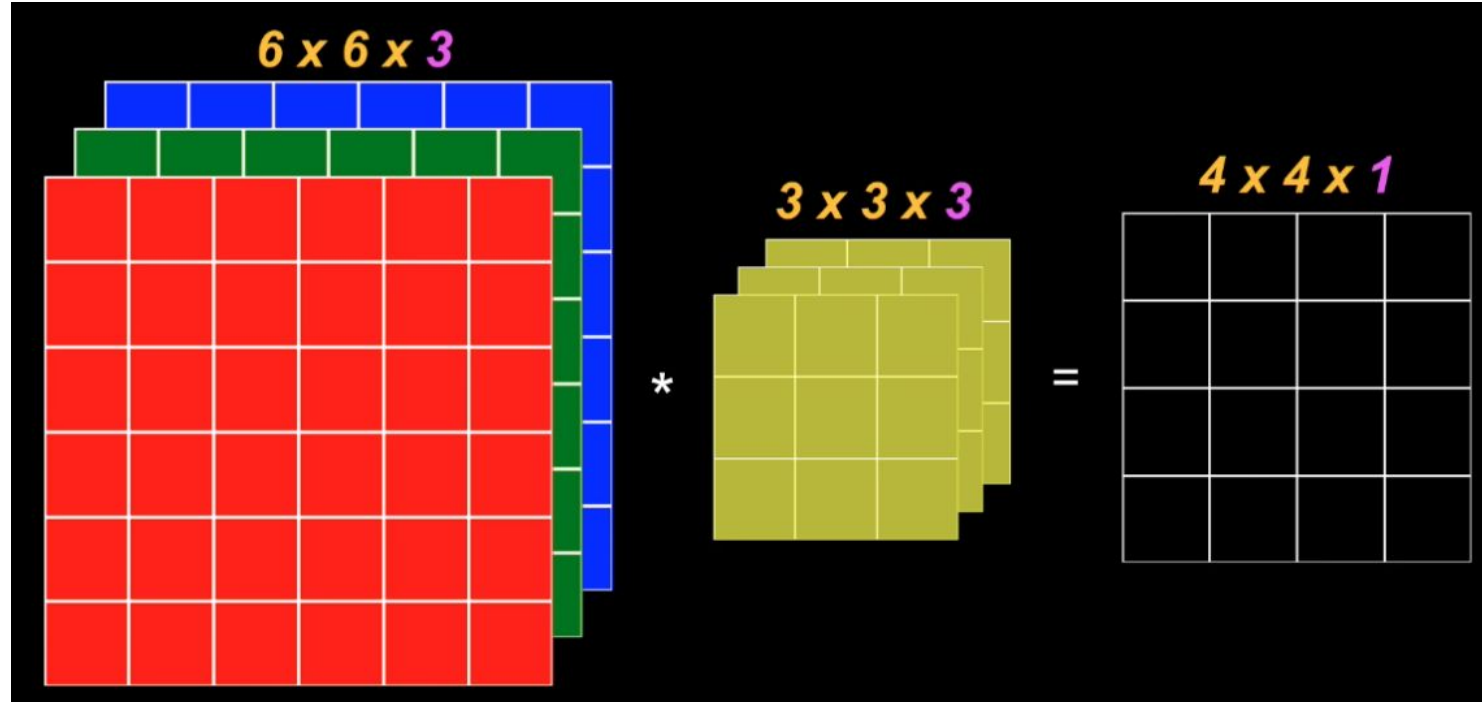
kernel: sharpen

$$\text{Kernel} - h = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Convolución con 3 dimensiones



Convolución con 3 dimensiones



Quiz time!

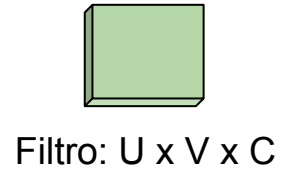
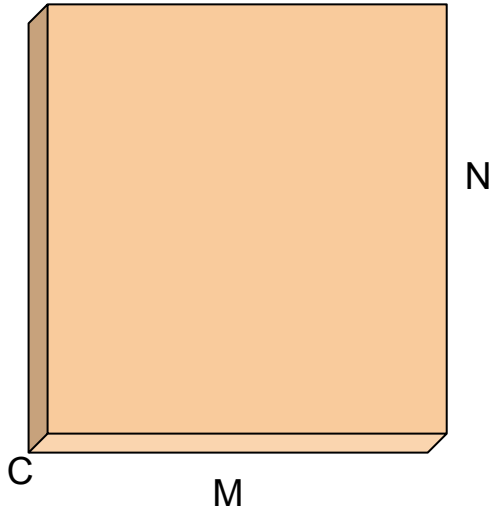
1. ¿Verdadero o Falso?
 - Las redes completamente conectadas son mejores para trabajar con imágenes porque capturan todas las posibles relaciones entre píxeles.
2. ¿Cuál es la dimensión de la imagen de salida al convolucionar una imagen de dimensiones (10,10) con un filtro de dimensiones (3,3)?
3. Calcular la matriz resultante:

$$\begin{bmatrix} 3 & 1 & 4 \\ 2 & 2 & 2 \\ 4 & 8 & 1 \end{bmatrix} * \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 10 & 3 \end{bmatrix}$$

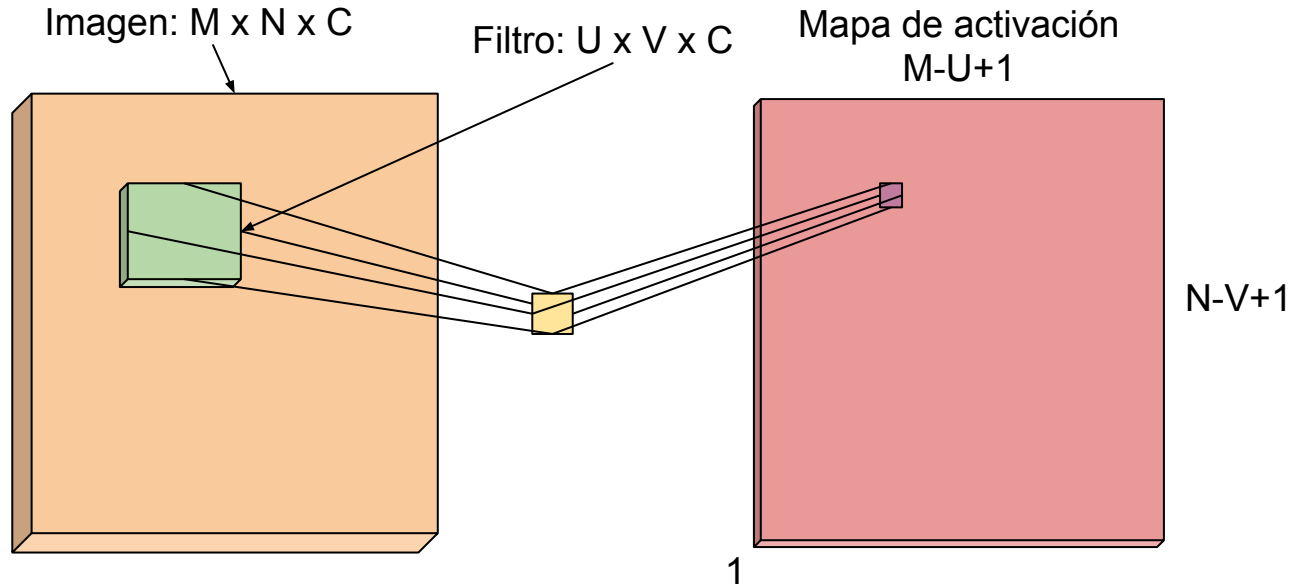
Redes Convolucionales

Capa de convolución

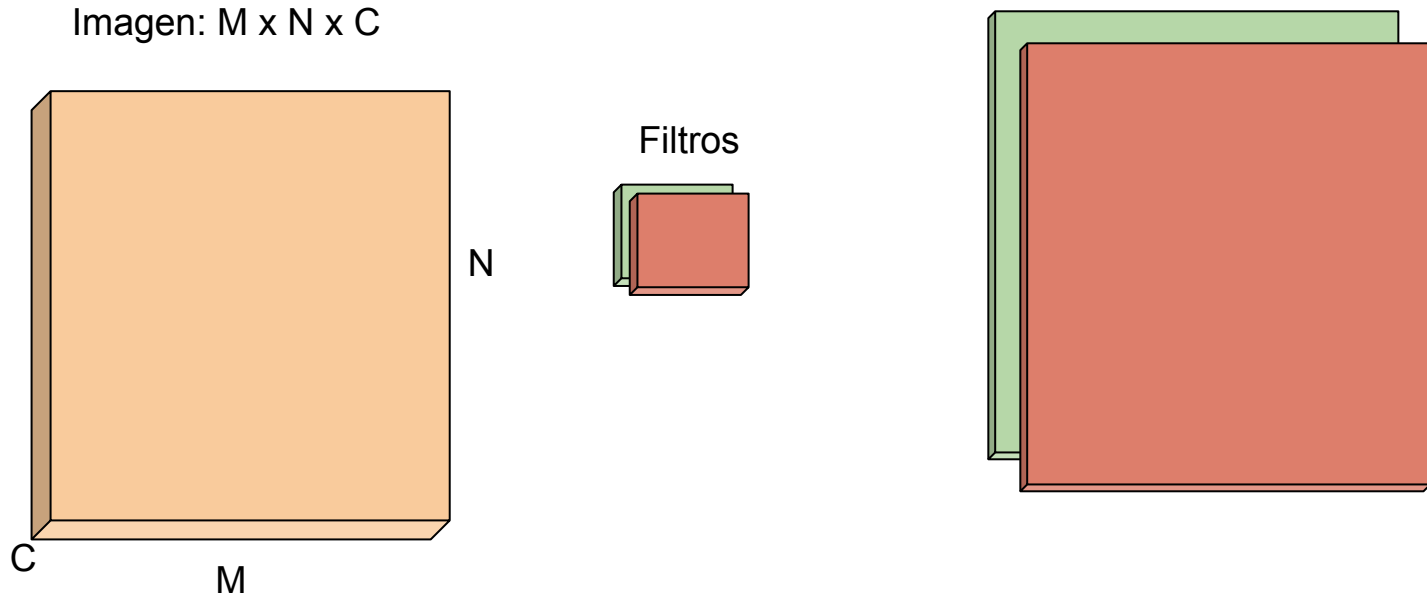
Imagen: $M \times N \times C$



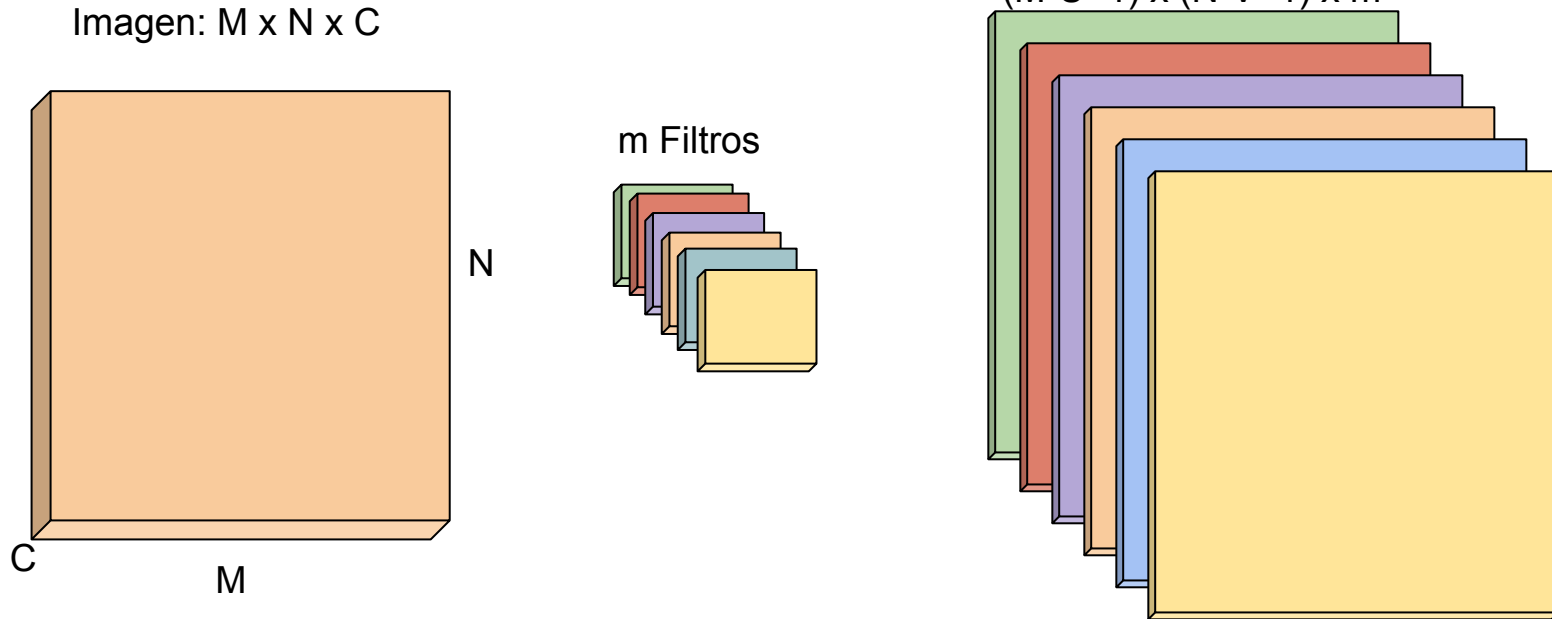
Capa de convolución



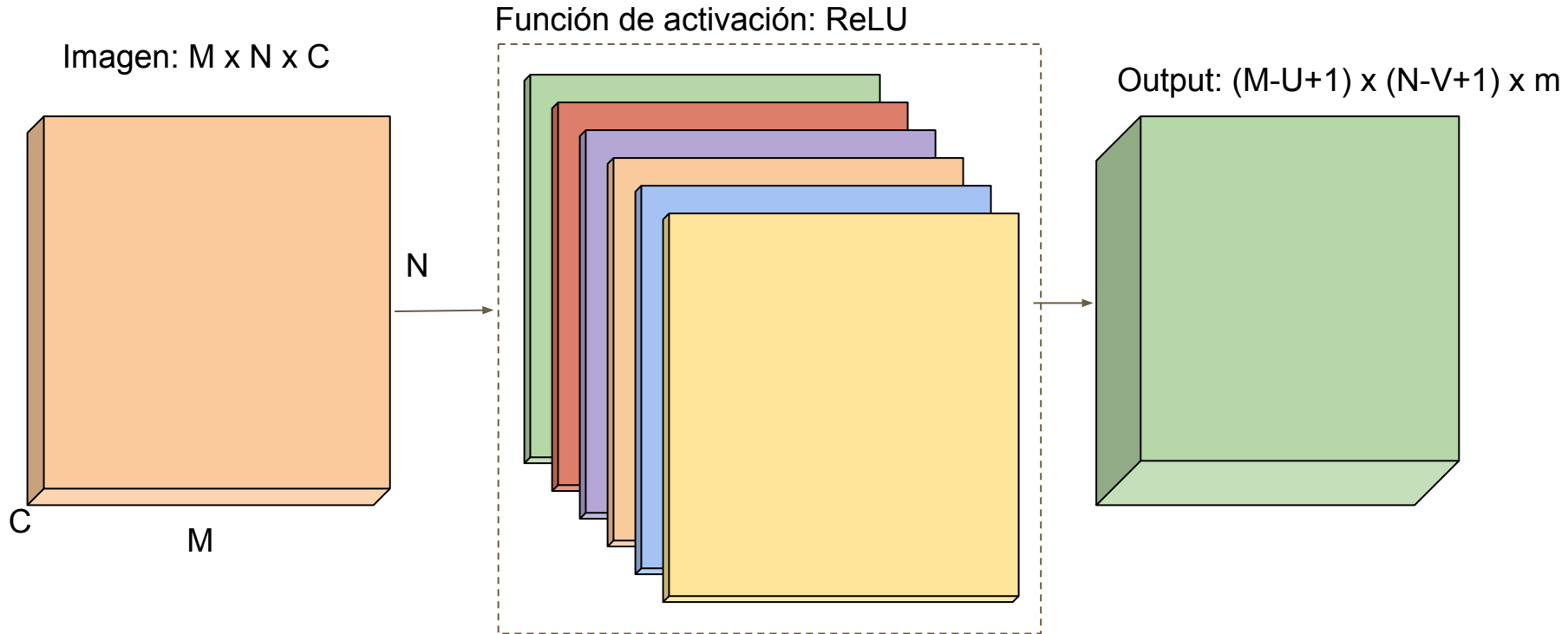
Capa de convolución



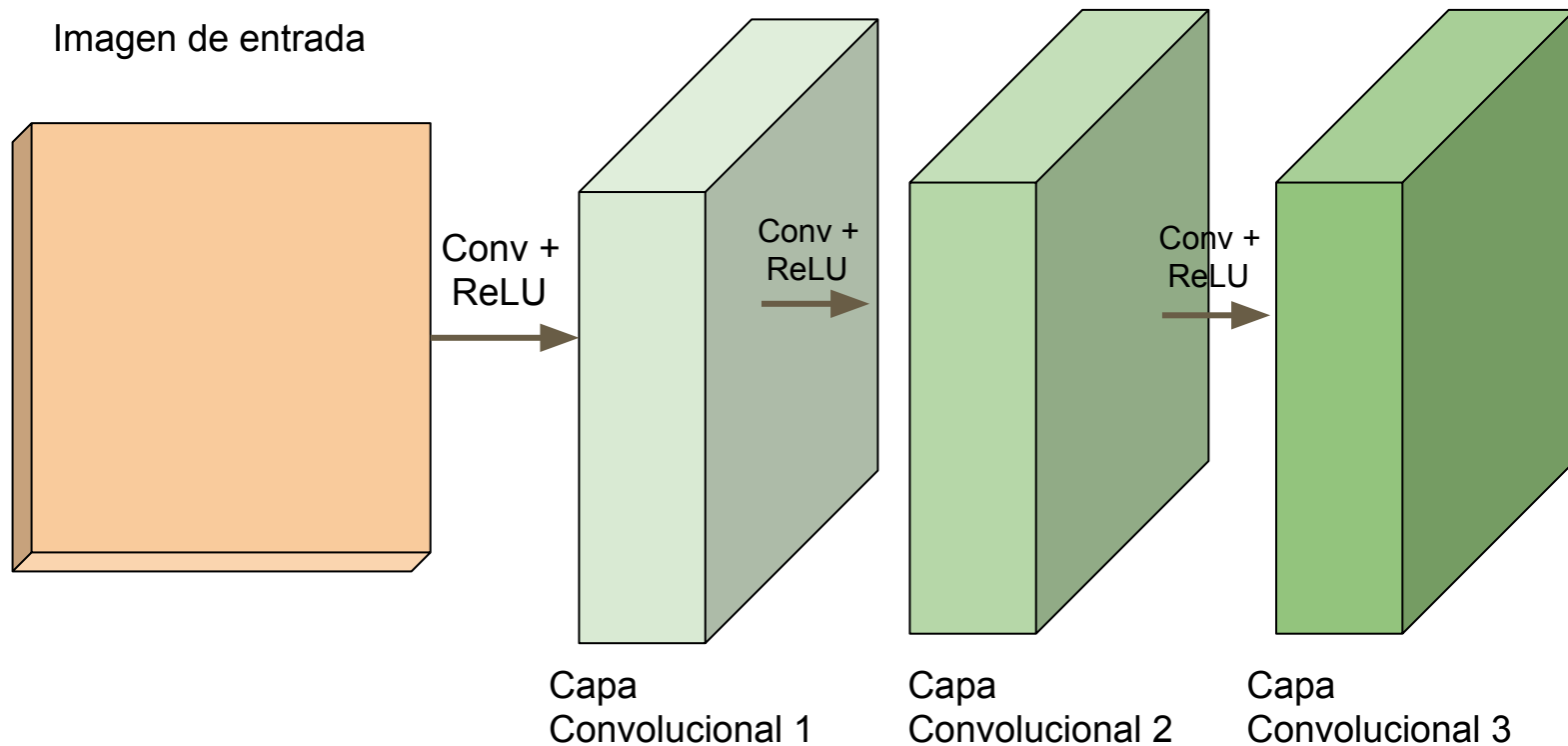
Capa de convolución



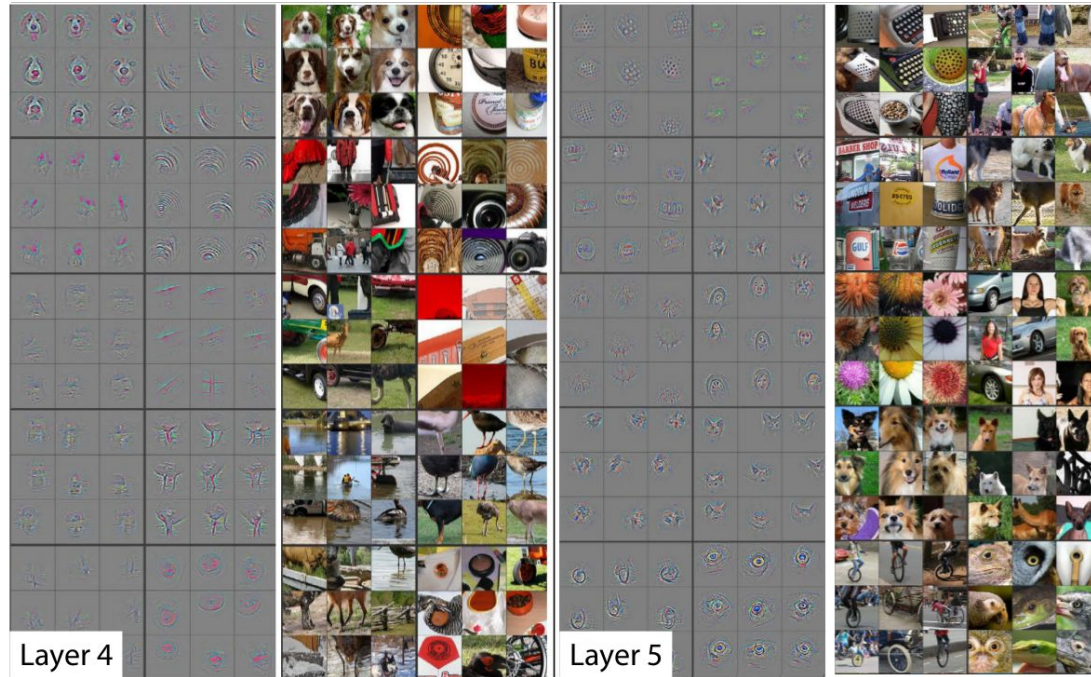
Capa de convolución: Convolución + Activación



Red convolucional



Visualización



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

Más sobre la capa de convolución

- Dimensiones
- Observaciones

Dimensiones

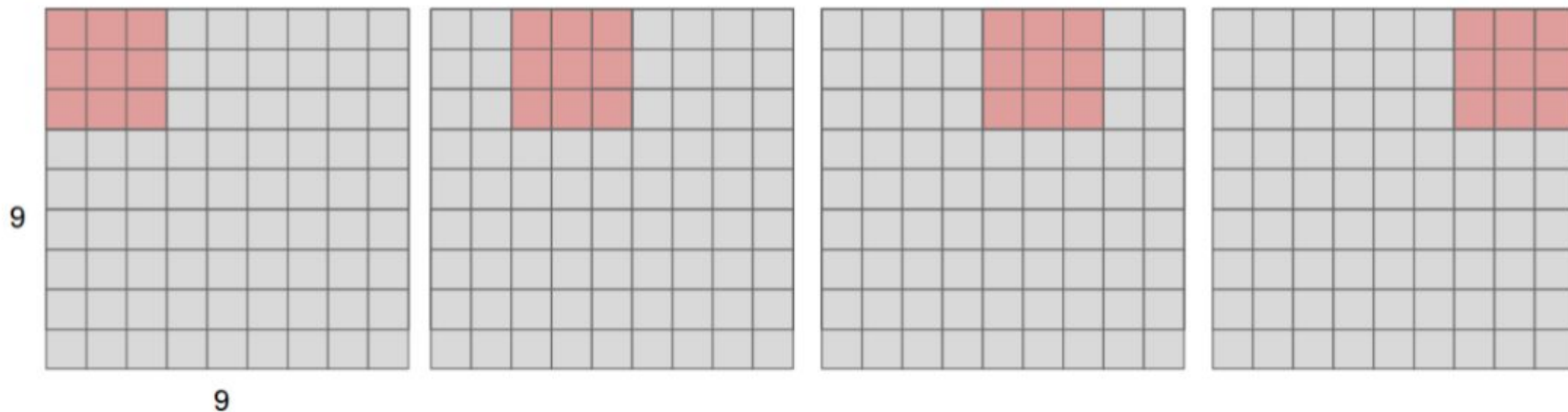


imagen de entrada: 9x9
Filtro: 3x3
Stride (paso): 2

¿Salida?

$(M-u)/s + 1 \times (N-v)/s + 1$, donde:

(M, N) son las dimensiones de la imagen de entrada y (u, v) las dimensiones del filtro.

Reflection y Replication padding

3	5	1
3	6	1
4	7	9

No padding

1	6	3	6	1	6	3
1	5	3	5	1	5	3
1	6	3	6	1	6	3
9	7	4	7	9	7	4
1	6	3	6	1	6	3

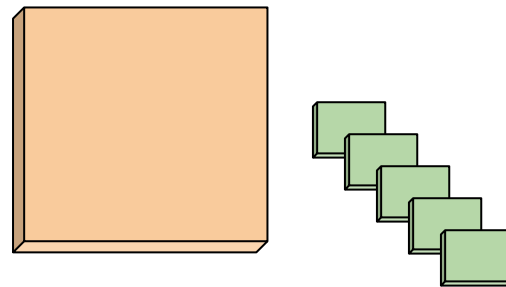
(1, 2) reflection padding

5	3	3	5	1	1	5
5	3	3	5	1	1	5
6	3	3	6	1	1	6
7	4	4	7	9	9	7
7	4	4	7	9	9	7

(1, 2) replication padding

Ejercicio

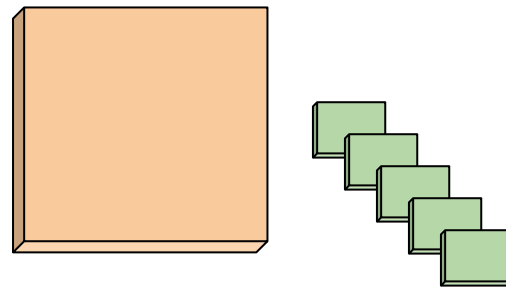
- Tamaño de entrada: 32x32x3
- Filtros:
 - Cantidad: 10
 - Tamaño: 5x5
 - Stride: 1
 - Pad: 2



- Tamaño de salida?

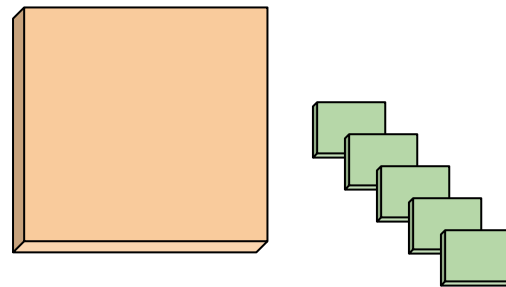
Ejercicio: Solución

- Tamaño de entrada: 32 x 32 x 3
- Filtros:
 - Cantidad: 10
 - Tamaño: 5x5
 - Stride: 1
 - Pad: 2
- Tamaño de salida:
 - $(32 + 2*2 - 5)/1+1=32$ (Largo y ancho)
 - $\Rightarrow 32 \times 32 \times 10$



Ejercicio

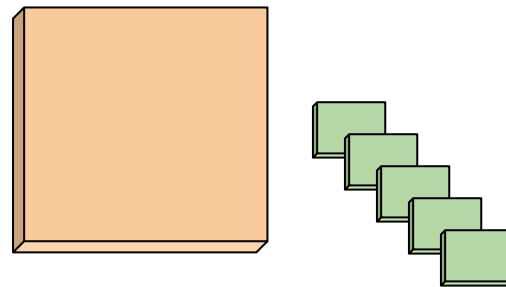
- Tamaño de entrada: 32x32x3
- Filtros:
 - Cantidad: 10
 - Tamaño: 5x5 *
 - Stride: 1
 - Pad: 2



- Número de parámetros en esta capa?

Ejercicio: Solución

- Tamaño de entrada: 32 x 32 x 3
- Filtros:
 - Cantidad: 10
 - Tamaño: 5x5
 - Stride: 1
 - Pad: 2



- Número de parámetros en esta capa?
 - Cada filtro tiene: $5 \times 5 \times 3 + 1(\text{bias}) = 76$ parámetros $\Rightarrow 76 \times 10 = 760$

Observaciones

- **Pesos compartidos:** A diferencia de capas totalmente conectadas, los pesos de las capas de convolución (filtros) se reutilizan en varios elementos de la entrada.
- **Equivarianza:** Si se traslada la entrada, se traslada la salida.
- **Representaciones:** Capas de convolución permiten manejar datos de diferente tamaño, sin necesidad de cambiar la arquitectura

Capa de Pooling

Pooling layer

- Comprime (sub-muestreo) de la representación
- Opera en cada mapa de activación por separado
- Su objetivo es reducir la cantidad de parámetros de la CNN, ayudando a prevenir el overfitting.



Pooling layer

- Max-pooling es el más utilizado.
- Usualmente se utiliza max pooling de 2x2, con stride de 2
- ¿Parámetros?



Quiz time!

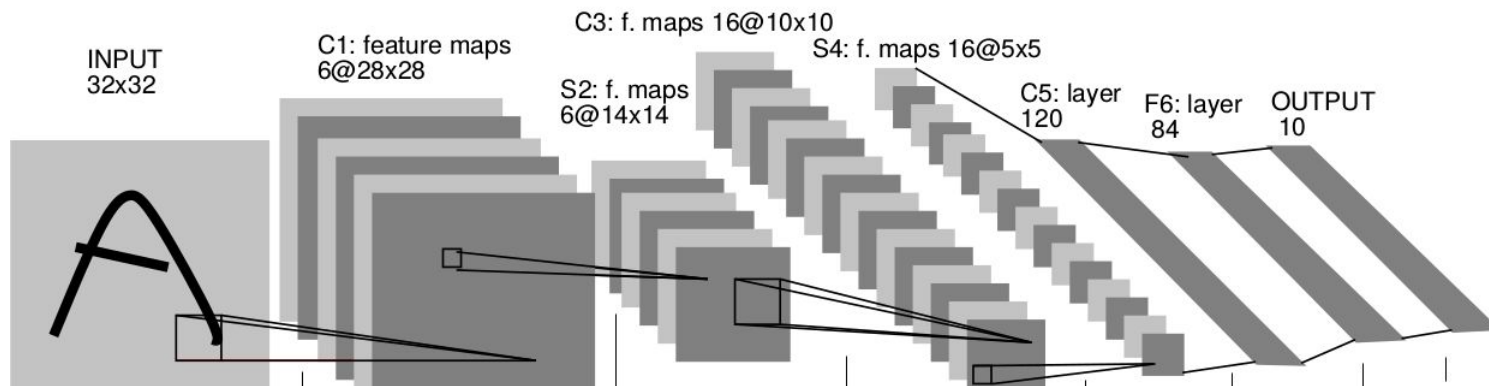
1. ¿Cuáles son las dimensiones del volumen resultante de aplicar a una imagen RGB de 16×16 píxeles, una capa de convolución con 10 filtros de dimensiones $(5, 5)$, con stride de tamaño 1 y zero padding de tamaño 2?
2. ¿Y si luego aplicamos max pooling de 2×2 (paso 2)?

Arquitecturas

Arquitecturas - Un poco de historia

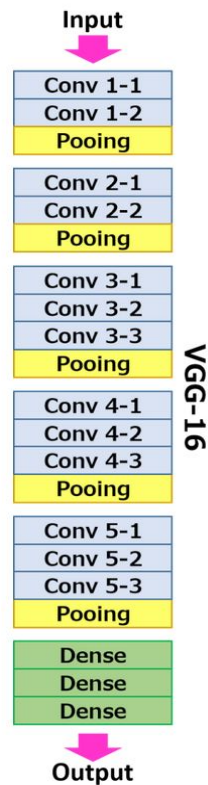
- LeNet-5

Gradient-Based Learning Applied to Document Recognition [Yann LeCun et al., 1998] - (Citado 39.187 veces)



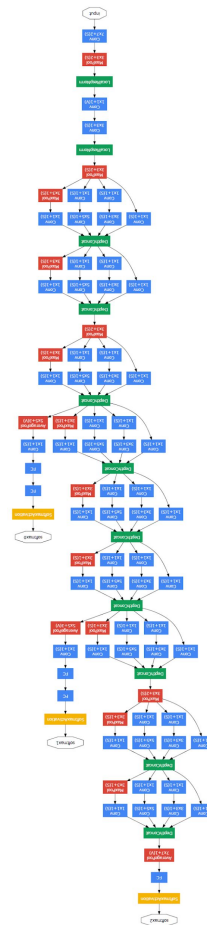
Arquitecturas - VGG

- “Very Deep Convolutional Networks for Large-Scale Visual Recognition”, Karen Simonyan y Andrew Zisserman, 2014.
- Primer y segundo puesto del challenge ImageNet ILSVRC-2014 - clasificación y localización.
- Arquitectura muy homogénea.
- Demuestran que la profundidad es un factor importante para lograr buena performance (utilizan hasta 19 capas)



Arquitecturas - GoogLeNET

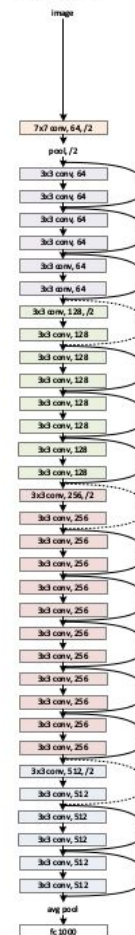
- “Going Deeper with Convolutions”, Christian Szegedy et. al, 2014.
- Ganadores del challenge ImageNet ILSVRC-2014 - detección.
- Introducen el bloque de “inception”, que permite aplicar diferentes tamaños de filtros en la misma capa de convolución - dejando esta “decisión” al proceso de entrenamiento.
- Utilizan clasificadores intermedios que ayudan a prevenir el problema de vanishing gradient.



Arquitecturas - Resnet

- Deep Residual Learning for Image Recognition, Kaiming He et al., 2015.
- Ganadores de ILSVRC 2015: detección, y detección con localización.
- Investigan el problema de la degradación en performance al hacer las CNNs más profundas.
- Intuición: una red poco profunda es una subred de la red completa. Sin embargo, con demasiada profundidad, la performance se degrada.
- Proponen agregar conexiones residuales, que facilitan que la red aprenda la identidad cuando sea necesario.
- Permite redes más profundas sin el problema del vanishing gradient.

34-layer residual



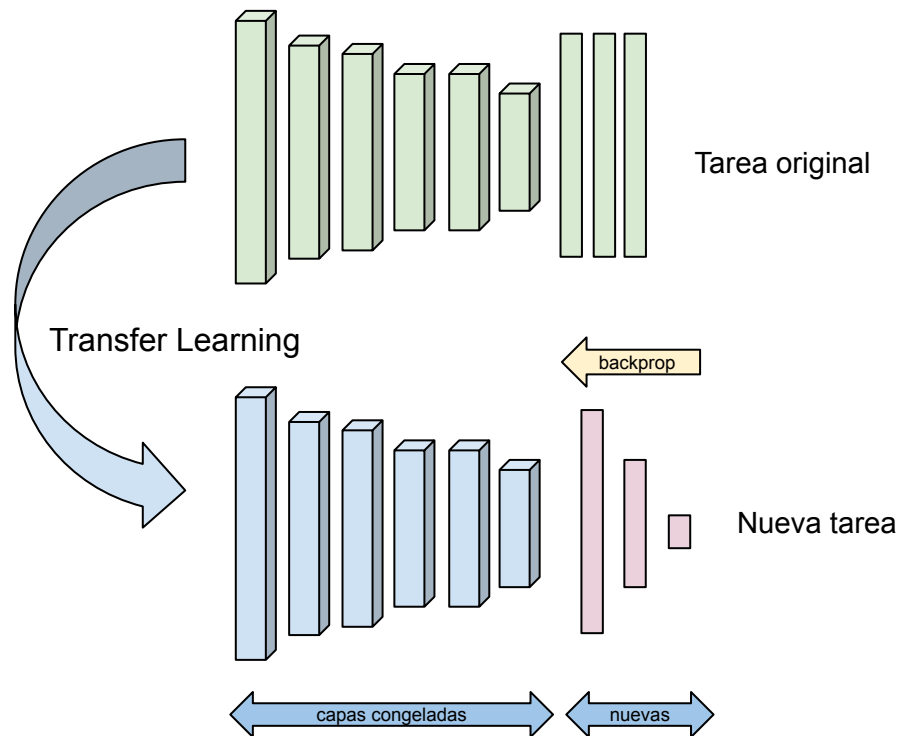
Transfer learning y Fine tuning

Transfer Learning y Fine-Tuning

- Las técnicas de ***transfer learning*** y ***fine tuning*** proponen reutilizar redes pre-entrenadas.
- Se sustituyen las capas finales por nuevas capas que se ajustarán a nuevos problemas.
- Las capas anteriores (con sus pesos ya pre-entrenados) se reutilizan:
 - transfer learning: las “congela”
 - fine tuning: las sigue entrenando

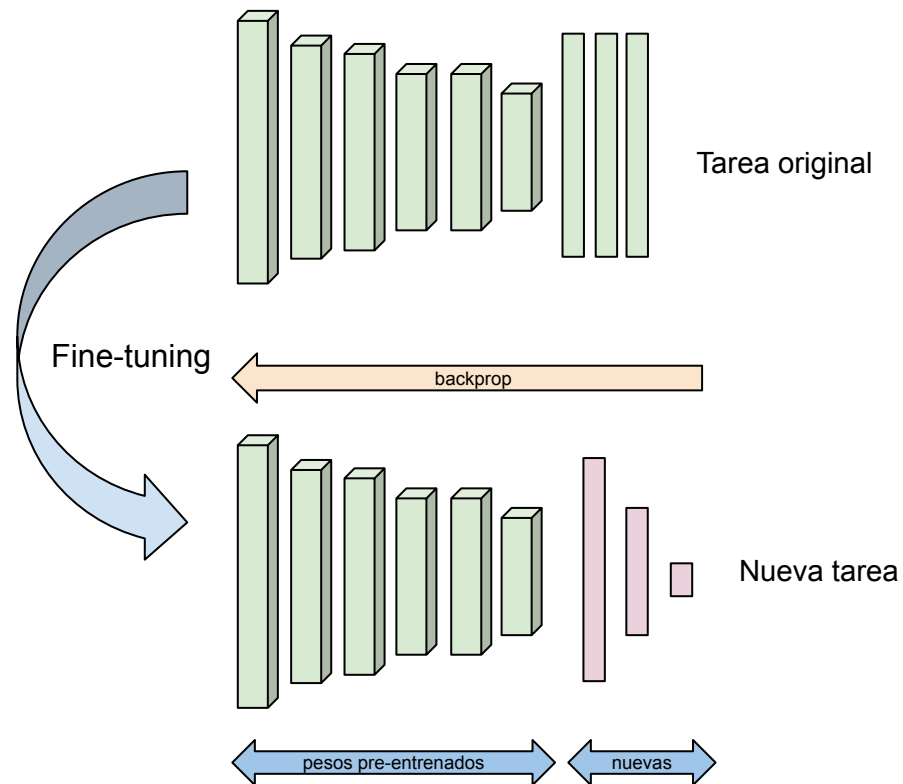
Transfer Learning y Fine-Tuning

- Transfer learning:
 - las capas anteriores se “congelan”, dejando fijos los pesos. Sólo se entrenan las nuevas capas finales.
 - es más rápido que el fine-tuning, porque hay menos parámetros para ajustar.



Transfer Learning y Fine-Tuning

- Fine tuning:
 - no se “congela” ninguna capa.
 - Se sustituyen las capas finales por nuevas capas, y se entrenan todos los pesos.
 - Es más costoso que el transfer learning, pero puede dar mejores resultados, ya que podemos modificar los pesos de etapas tempranas para que se ajusten mejor al nuevo problema.



Bibliografía/recursos

[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.