

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
Licenciatura en Ciencia de Datos

Introducción al Aprendizaje Profundo  
Redes convolucionales

Profesores:  
Berenice & Ricardo Montalvo Lezama

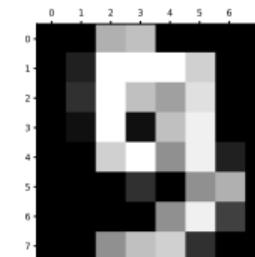
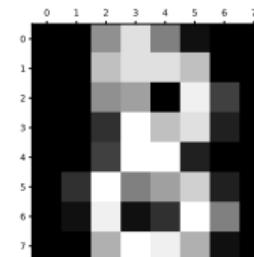
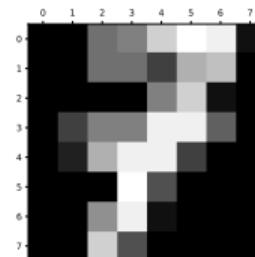
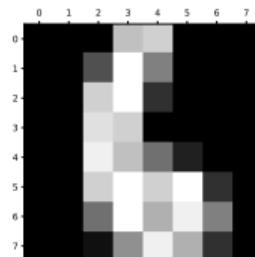
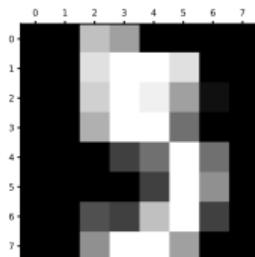
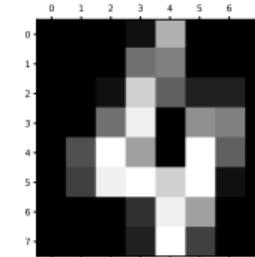
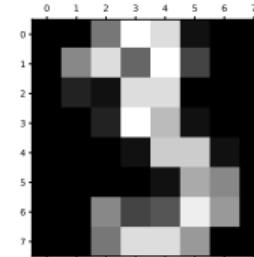
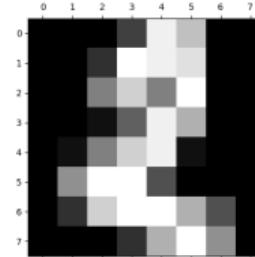
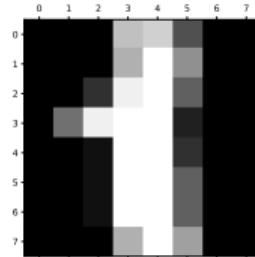
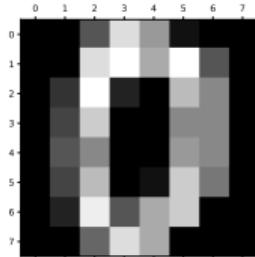
Marzo 2021

Contenido basado en el curso de AP del Dr. Gibran Fuentes Pineda del PCIC

## Introducción

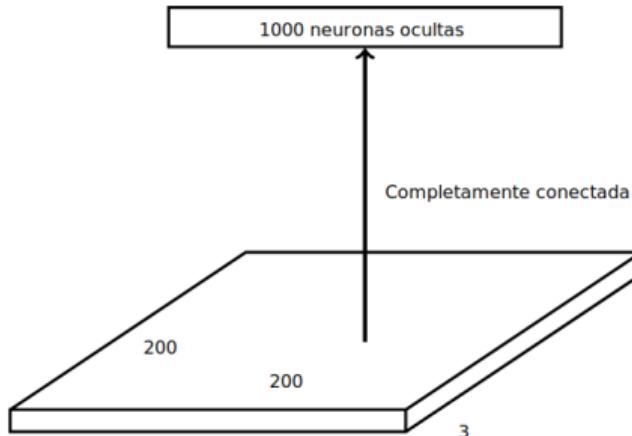
---

# Clasificación de imágenes



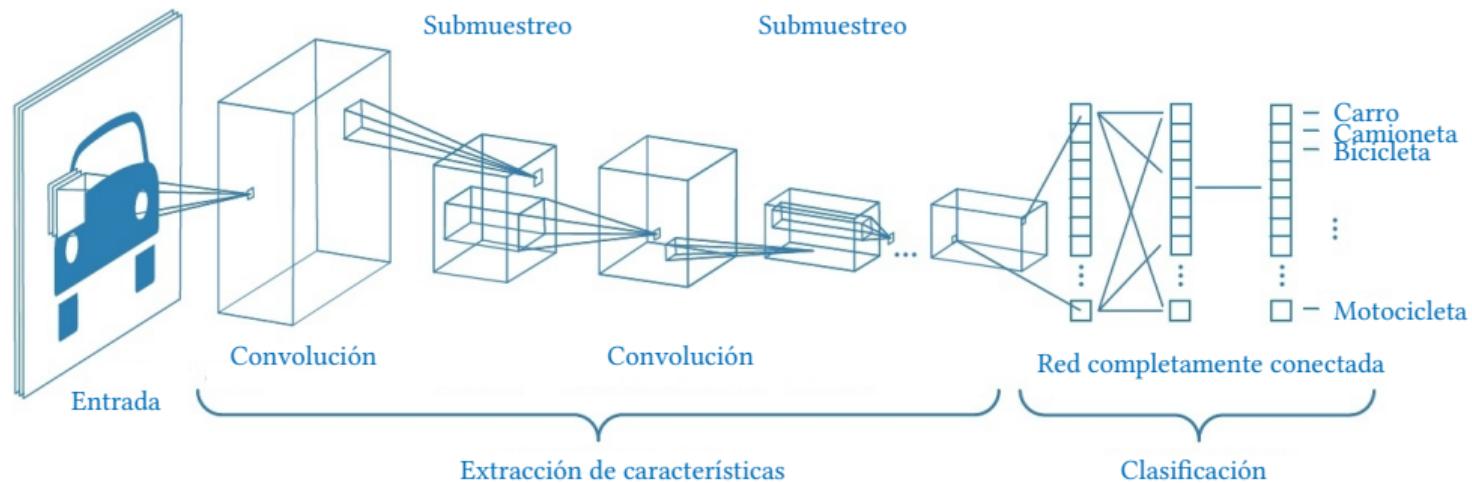
## Problemática de usar redes densas para imágenes

- Supongamos que queremos entrenar una red que tome una imagen RGB de  $200 \times 200$ .



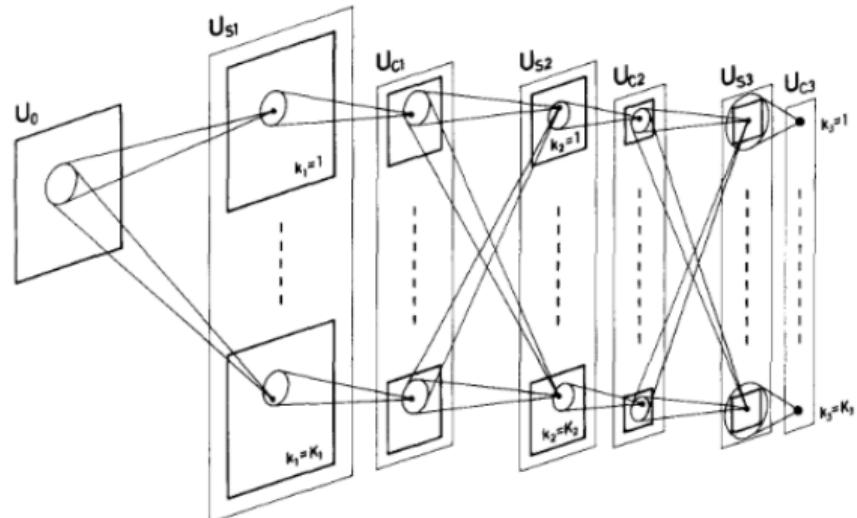
- ¡Se requieren muchos parámetros!
  - Entrada =  $200 \times 200 \times 3 = 120,000$ .
  - Parámetros =  $120,000 \times 1000 + 1000 = 120,001,000$ .

# Red neuronal convolucional



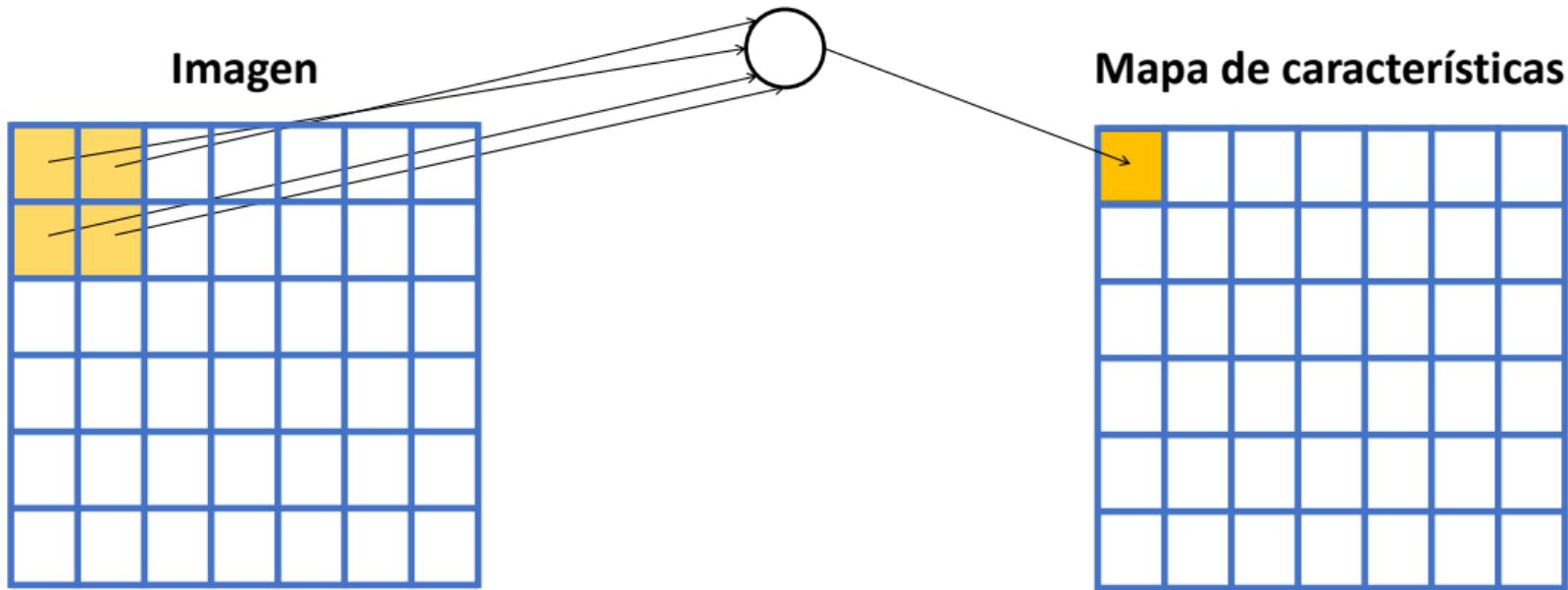
## Capa convolucional

- Pueden verse como un caso especial de una capa densa con 2 variaciones:
  1. Conectividad local (dispersa).
  2. Pesos compartidos.
- Las representaciones obtenidas son más eficientes.
- La operación es invariante a translaciones.

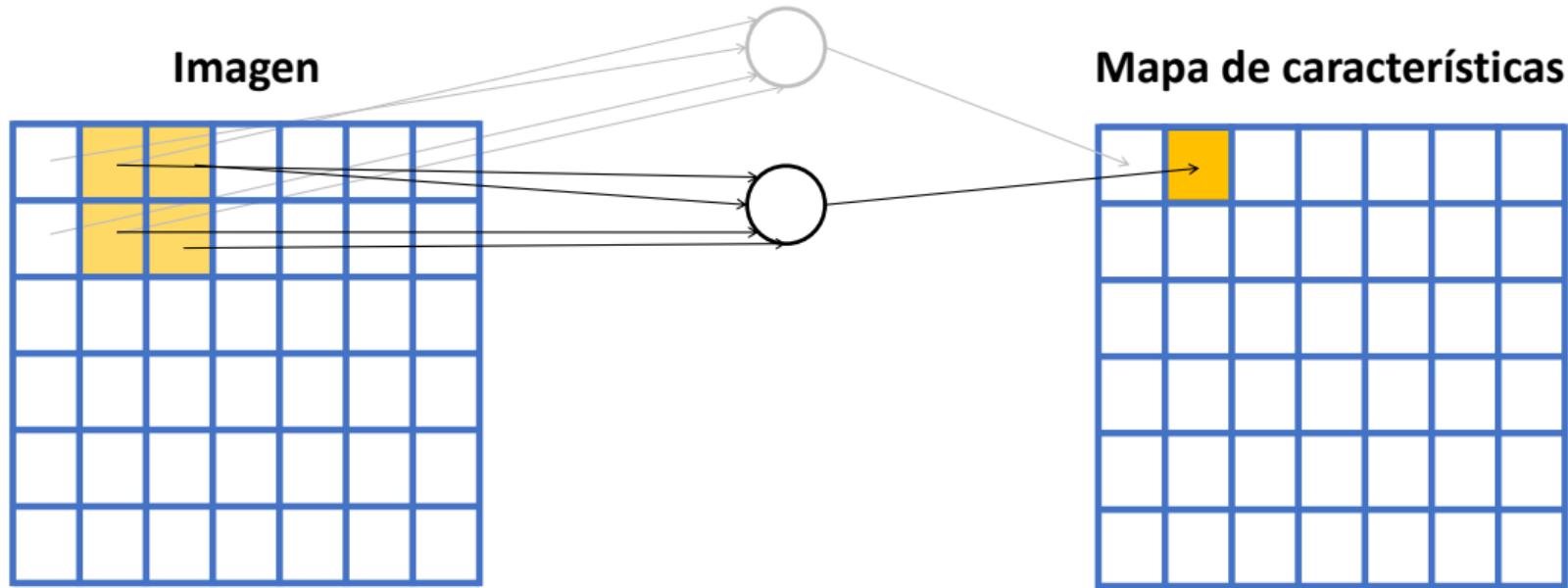


Neocognitron

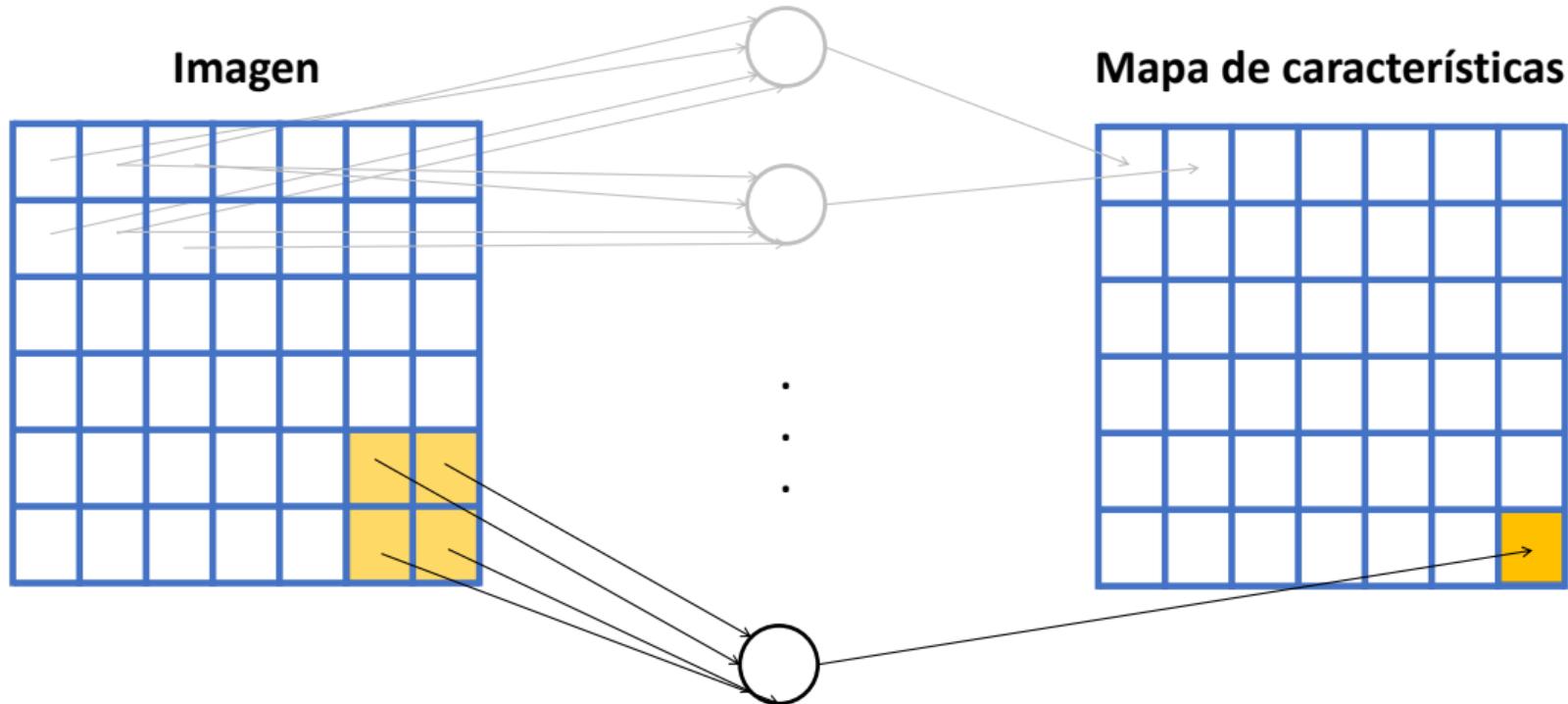
## Conectividad local (I)



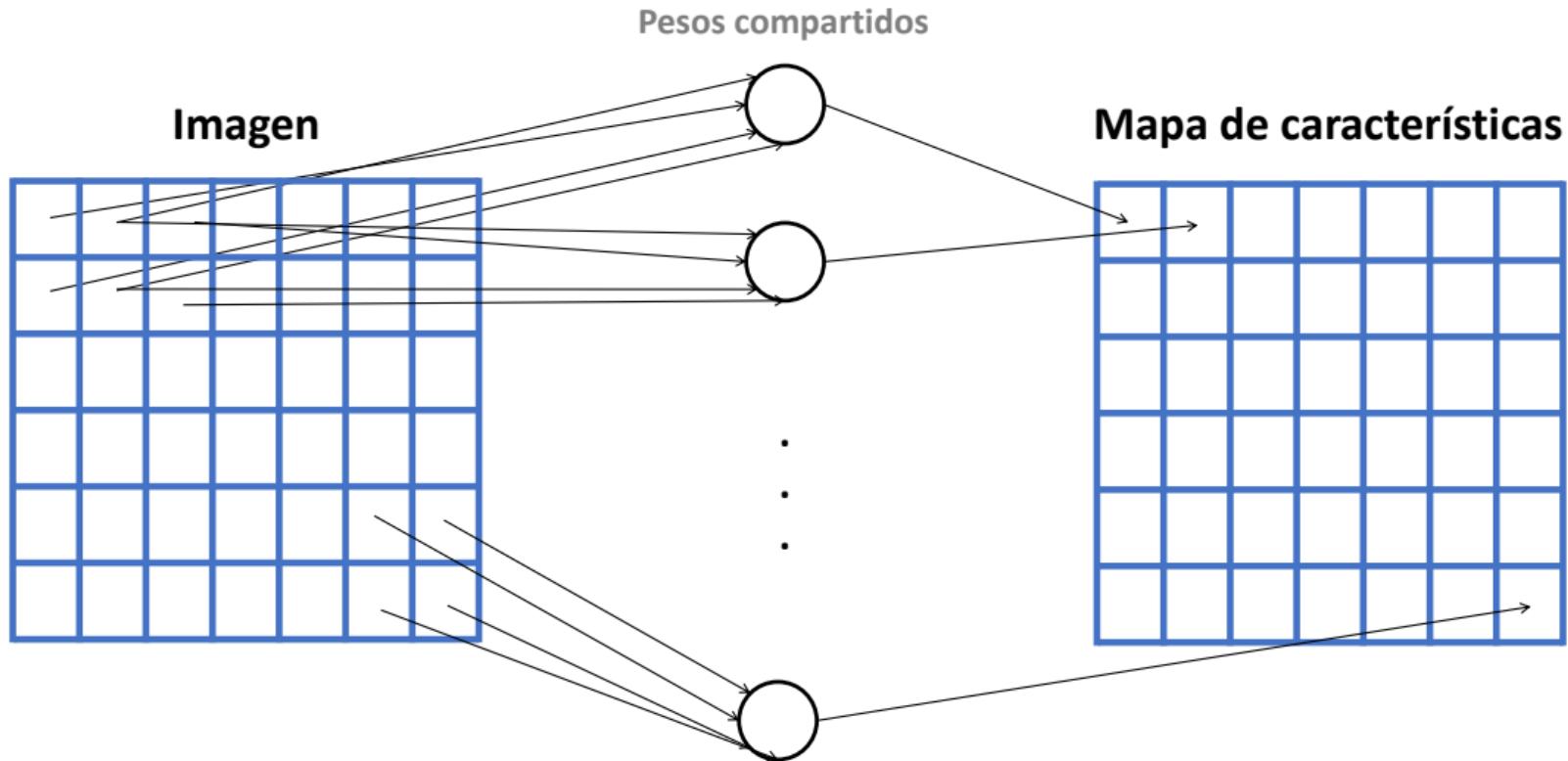
## Conectividad local (II)



## Conectividad local (III)



## Pesos compartidos



# Convolución 2D

Filtro 2D

0	1	2
2	2	0
0	1	2

ventaneo × 1 en filas

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

ventaneo × 1 en columnas

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

12	12	17
10	17	19
9	6	14

Convolución: entrada  $5 \times 5$ , filtro  $3 \times 3$ , paso  $1 \times 1$ , salida  $3 \times 3$ ,

# Convolución 2D con relleno

Filtro 2D

0	1	2
2	2	0
0	1	2

ventaneo ×2 en filas

0	0	0	0	0	0	0	0
0	3	3	2	1	0	0	0
0	0	0	1	3	1	0	0
0	3	1	2	2	3	0	8
0	2	0	0	2	2	0	6
0	2	0	0	0	1	0	4
0	0	0	0	0	0	0	0

6	17	3
---	----	---

0	0	0	0	0	0	0	0
0	3	3	2	1	0	0	0
0	0	0	1	3	1	0	0
0	3	1	2	2	3	0	8
0	2	0	0	2	2	0	6
0	2	0	0	0	1	0	4
0	0	0	0	0	0	0	0

6	17	3
---	----	---

0	0	0	0	0	0	0	0
0	3	3	2	1	0	0	0
0	0	0	1	3	1	0	0
0	3	1	2	2	3	0	8
0	2	0	0	2	2	0	6
0	2	0	0	0	1	0	4
0	0	0	0	0	0	0	0

6	17	3
---	----	---

ventaneo ×2 en columnas



0	0	0	0	0	0	0	0
0	3	3	2	1	0	0	0
0	0	0	1	3	1	0	0
0	3	1	2	2	3	0	8
0	2	0	0	2	2	0	6
0	2	0	0	0	1	0	4
0	0	0	0	0	0	0	0

6	17	3
---	----	---

0	0	0	0	0	0	0	0
0	3	3	2	1	0	0	0
0	0	0	1	3	1	0	0
0	3	1	2	2	3	0	8
0	2	0	0	2	2	0	6
0	2	0	0	0	1	0	4
0	0	0	0	0	0	0	0

6	17	3
---	----	---

0	0	0	0	0	0	0	0
0	3	3	2	1	0	0	0
0	0	0	1	3	1	0	0
0	3	1	2	2	3	0	8
0	2	0	0	2	2	0	6
0	2	0	0	0	1	0	4
0	0	0	0	0	0	0	0

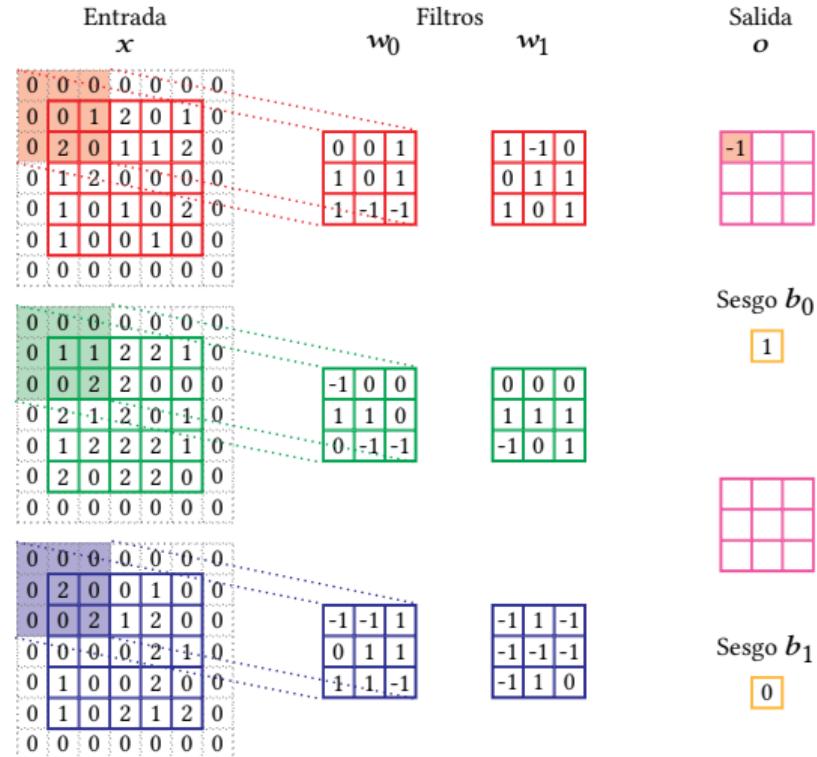
6	17	3
---	----	---

0	0	0	0	0	0	0	0
0	3	3	2	1	0	0	0
0	0	0	1	3	1	0	0
0	3	1	2	2	3	0	8
0	2	0	0	2	2	0	6
0	2	0	0	0	1	0	4
0	0	0	0	0	0	0	0

6	17	3
---	----	---

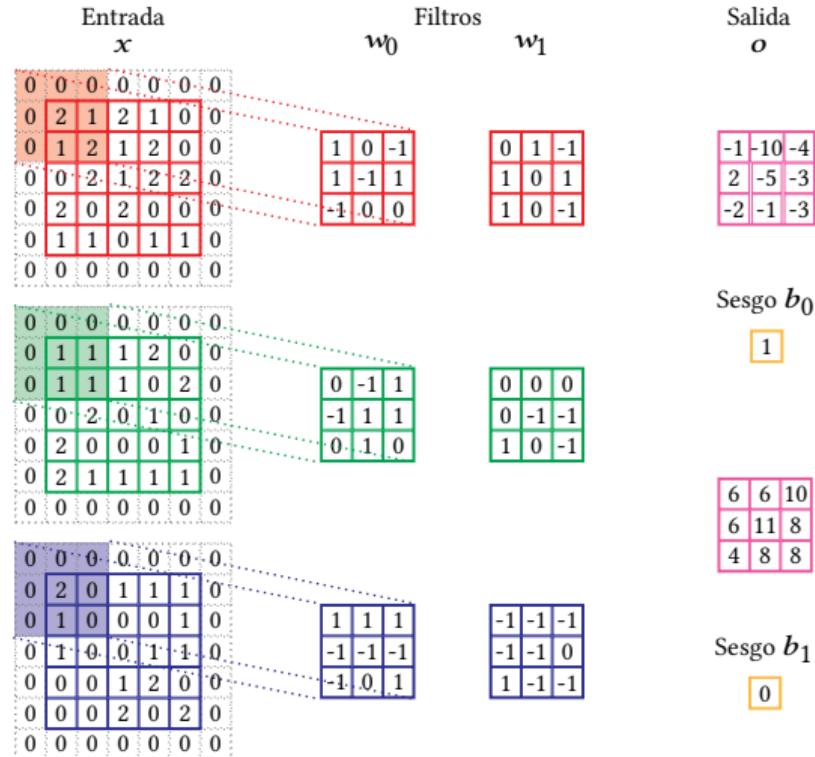
Convolución: entrada 5 × 5, filtro 3x3, paso 2 × 2, relleno 1 × 1, salida 3 × 3.,

## Capa de convolución (I)



Convolución: entrada  $5 \times 5 \times 3$ , filtro  $3 \times 3 \times 3$ , paso  $2 \times 2$ , relleno  $1 \times 1$ , salida  $3 \times 3 \times 2$ .

## Capa de convolución (II)



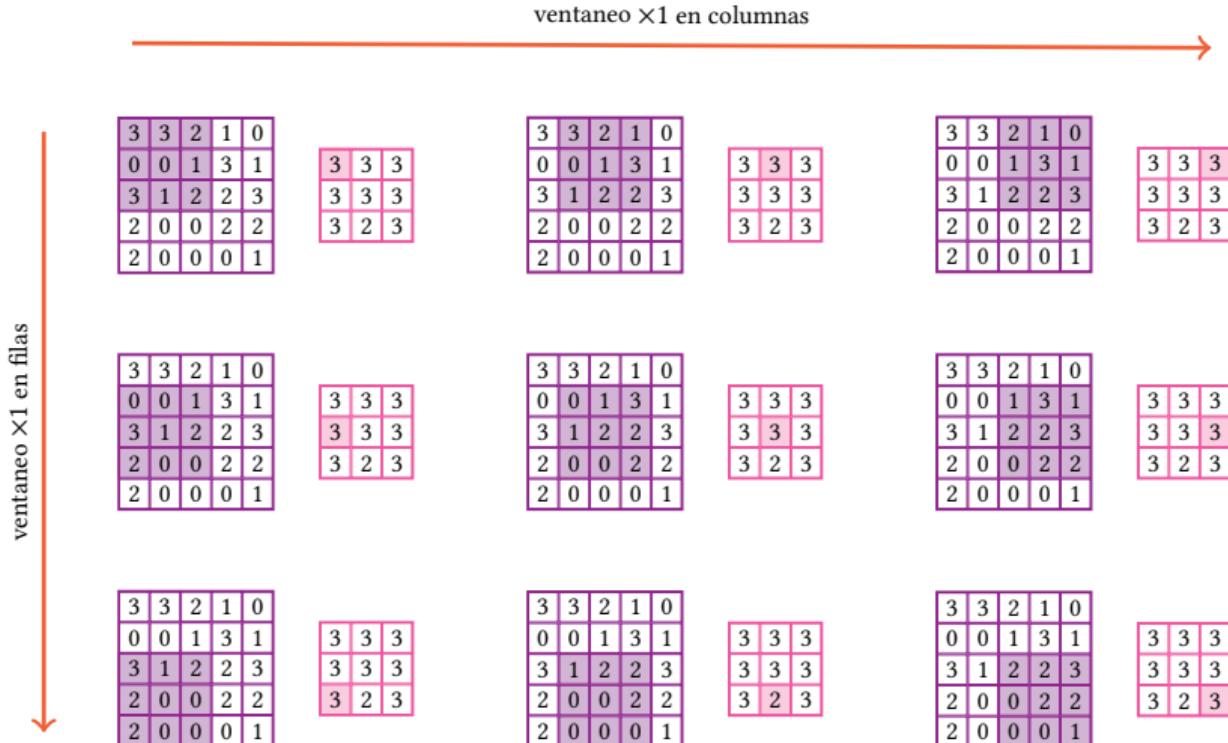
Convolución: entrada  $5 \times 5 \times 3$ , filtro  $3 \times 3 \times 3$ , paso  $2 \times 2$ , relleno  $1 \times 1$ , salida  $3 \times 3 \times 2$ .

## Hiperparámetros de convolución

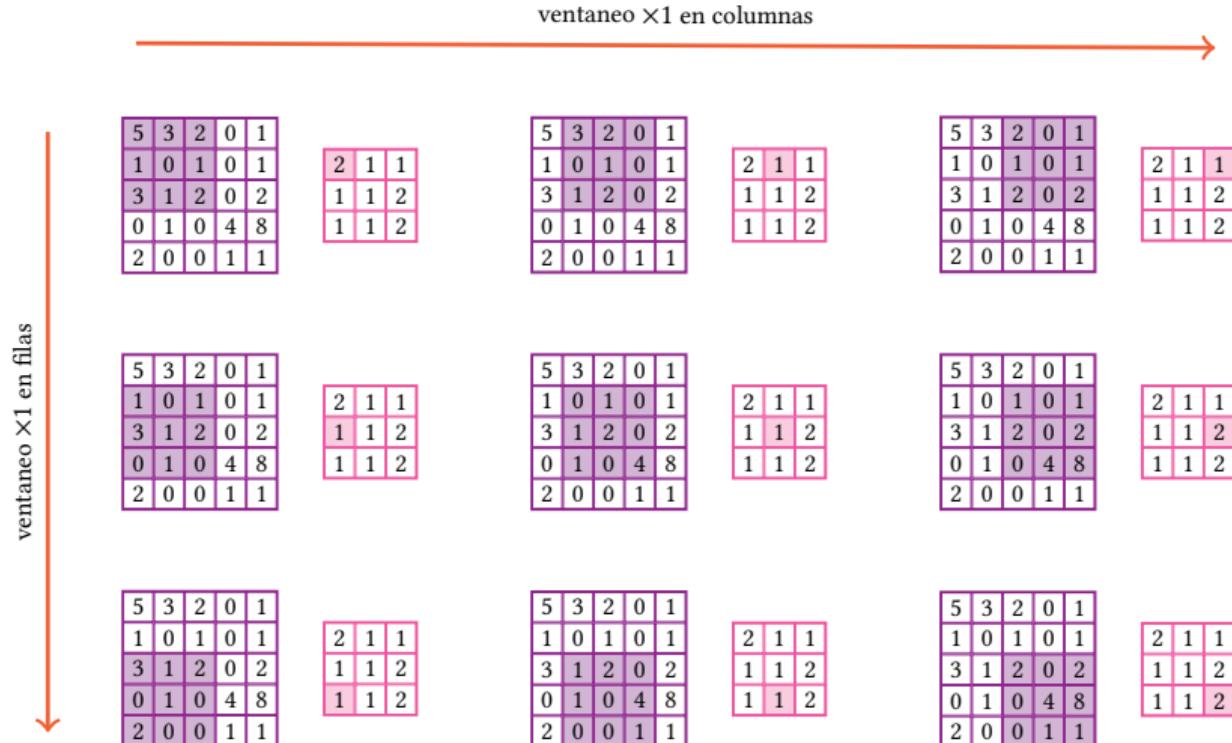
- Entrada:  $C_I \times H_I \times W_I$
- Hiperparámetros:
  - $K$ : número de filtros, profundidad de la salida.
  - $F$ : tamaño del filtro, extensión espacial del filtro.
  - $S$ : paso, cantidad de desplazamiento del filtro.
  - $P$ : relleno, cantidad de aumento de ceros.
- Salida:  $C_O \times H_O \times W_O$ 
  - $H_O = \frac{(H_I - F + 2P)}{S} + 1$
  - $W_O = \frac{(W_I - F + 2P)}{S} + 1$
  - $C_O = K$

- ¿Cuales serían las dimensiones del bloque de salida para una capa convolucional con siguientes características?
  - Entrada:  $1 \times 28 \times 28$
  - Número de filtros: 4
  - Tamaño del filtro: 3
  - Salto: 1
  - Relleno: 1
- $H_O = \frac{(H_I - F + 2P)}{S} + 1$
- $W_O = \frac{(W_I - F + 2P)}{S} + 1$
- $O = K$

## Capa de submuestreo máximo



## Capa de submuestreo promedio



Muestreo máximo: entrada  $5 \times 5$ , paso  $1 \times 1$ , salida  $3 \times 3$ .

## Hiperparámetros de submuestreo

- Entrada:  $C_I \times H_I \times W_I$
- Hiperparámetros:
  - $K$ : tamaño del filtro, extensión espacial.
  - $S$ : paso, cantidad de desplazamiento del filtro.
- Salida:  $C_O \times H_2 \times W_2$ 
  - $H_O = \frac{H_I - F}{S} + 1$
  - $W_O = \frac{W_I - F}{S} + 1$
  - $C_O = C_I$

- ¿Cuales serían las dimensiones del bloque de salida para una capa de muestreo con siguientes características?
  - Entrada:  $4 \times 28 \times 28$
  - Tamaño del filtro:  $2 \times 2$
  - Salto:  $2 \times 2$
- $H_O = \frac{H_I - F}{S} + 1$
- $W_O = \frac{W_I - F}{S} + 1$
- $C_O = C_I$

## Submuestro global promedio

- Reduce las dimensiones de un volumen de características, tomando únicamente el promedio de cada mapa.

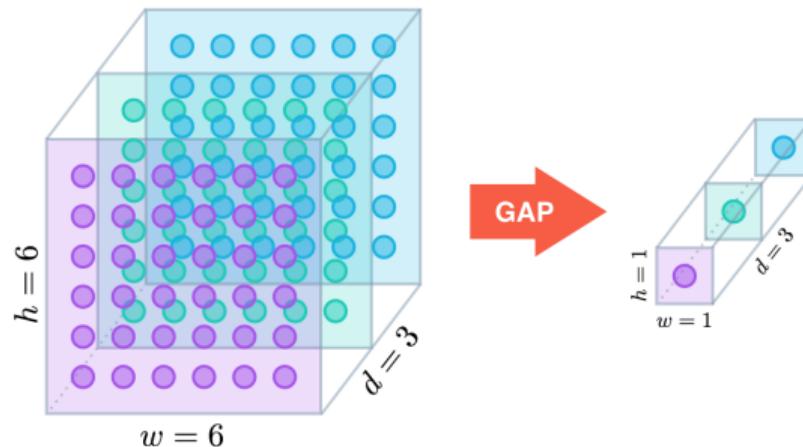


Imagen tomada de <https://alexisbcook.github.io/2017/global-average-pooling-layers-for-object-localization/>

## Convolución puntual

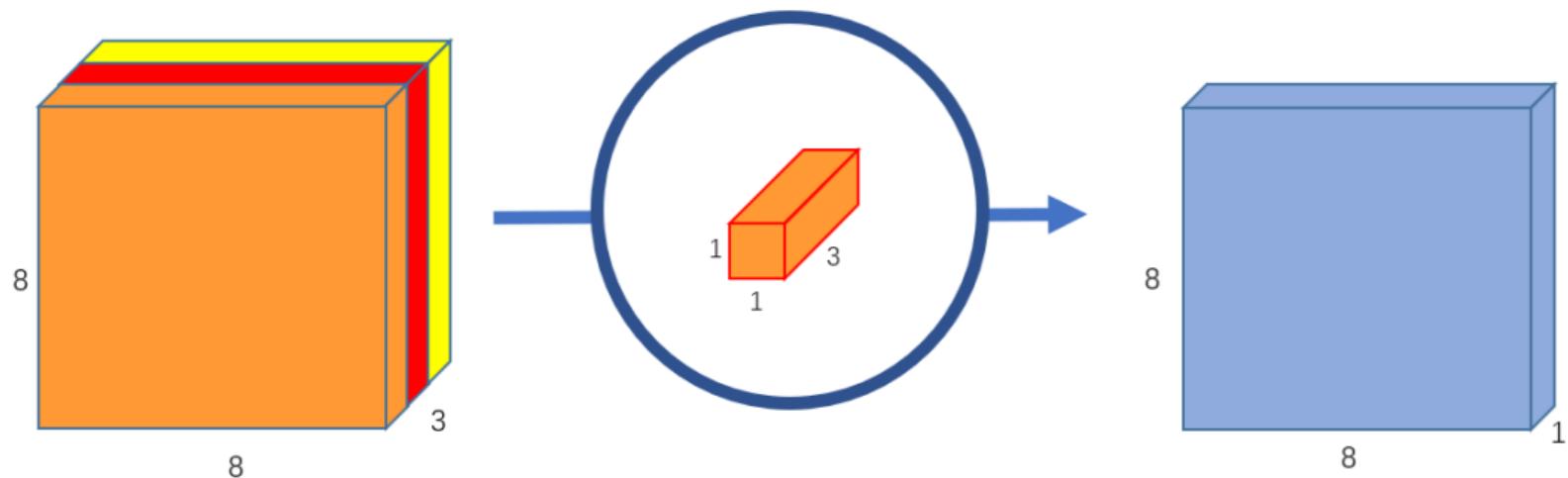
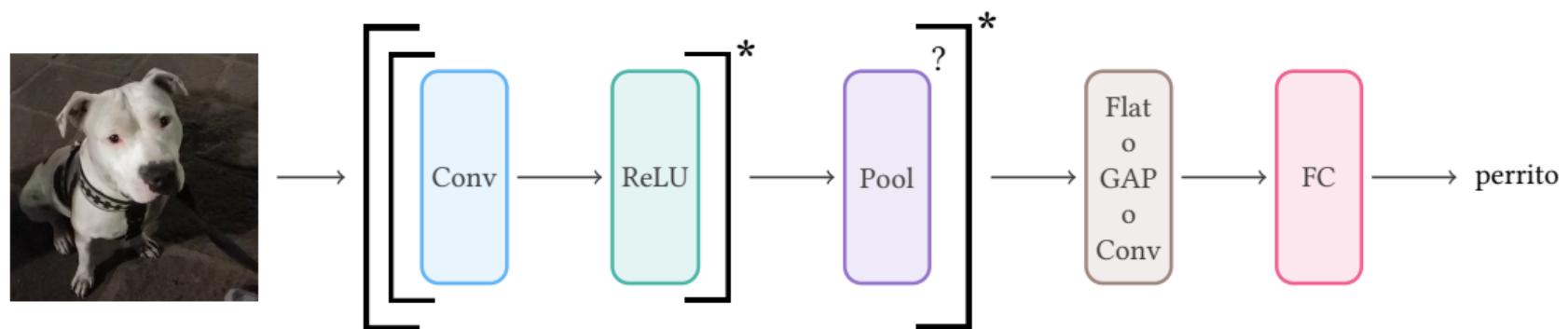
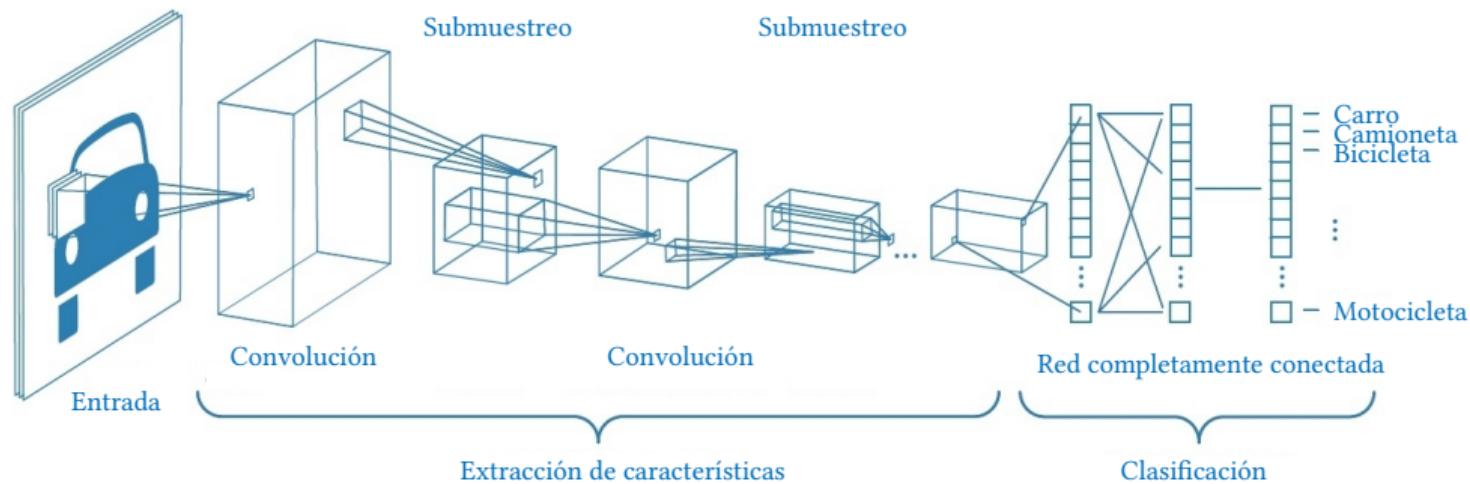


Imagen tomada de Chi-Feng Wang. A Basic Introduction to Separable Convolutions, 2018.

- Aumentar canales con capas convolucionales, reducir dimensiones con capas de muestreo.



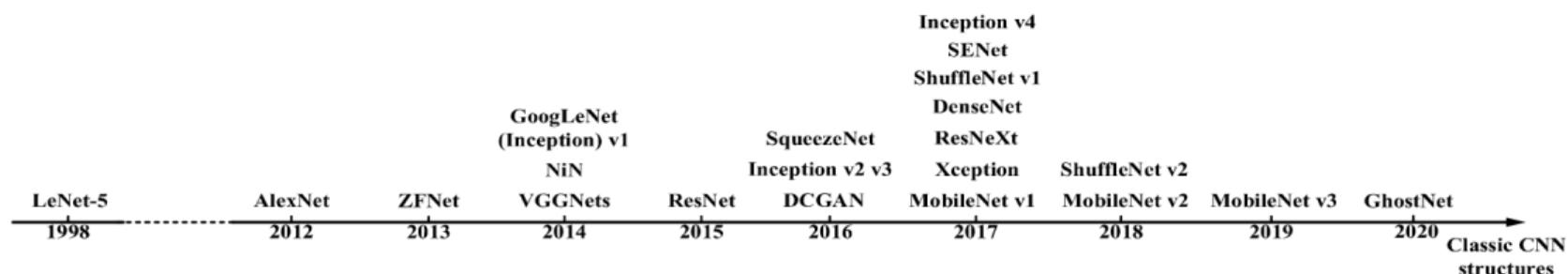
# Red neuronal convolucional



## Arquitecturas

---

# Cronología de las arquitecturas convolucionales



**Enfocadas a desempeño**

---

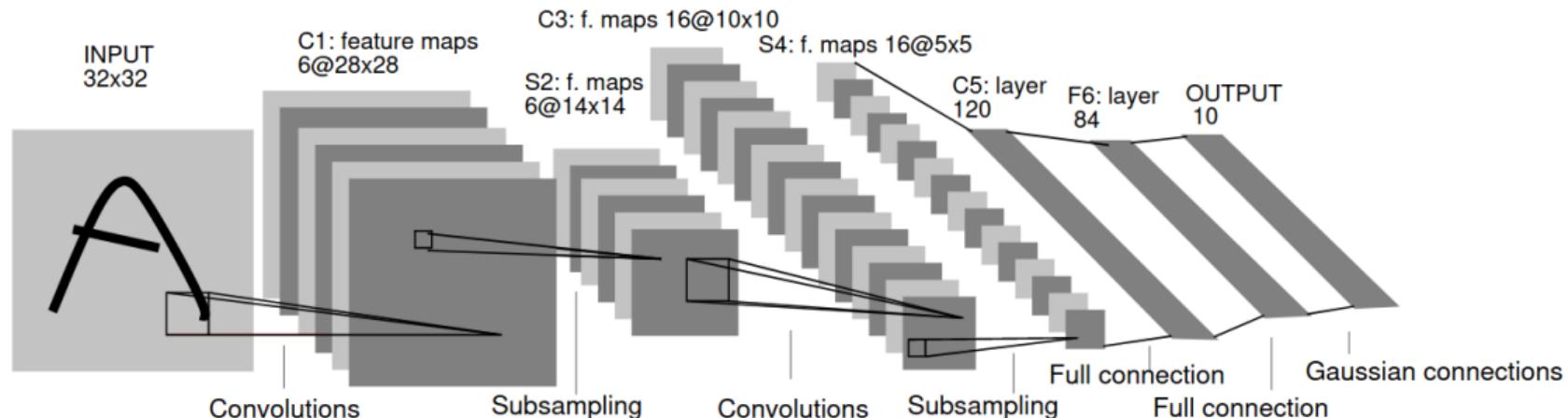


Imagen tomada de *Gradient-based learning applied to document recognition*

Lecun et al. *Gradient-based learning applied to document recognition*. 1998.

Lecun et al. *Backpropagation Applied to Handwritten Zip Code Recognition*. 1989.

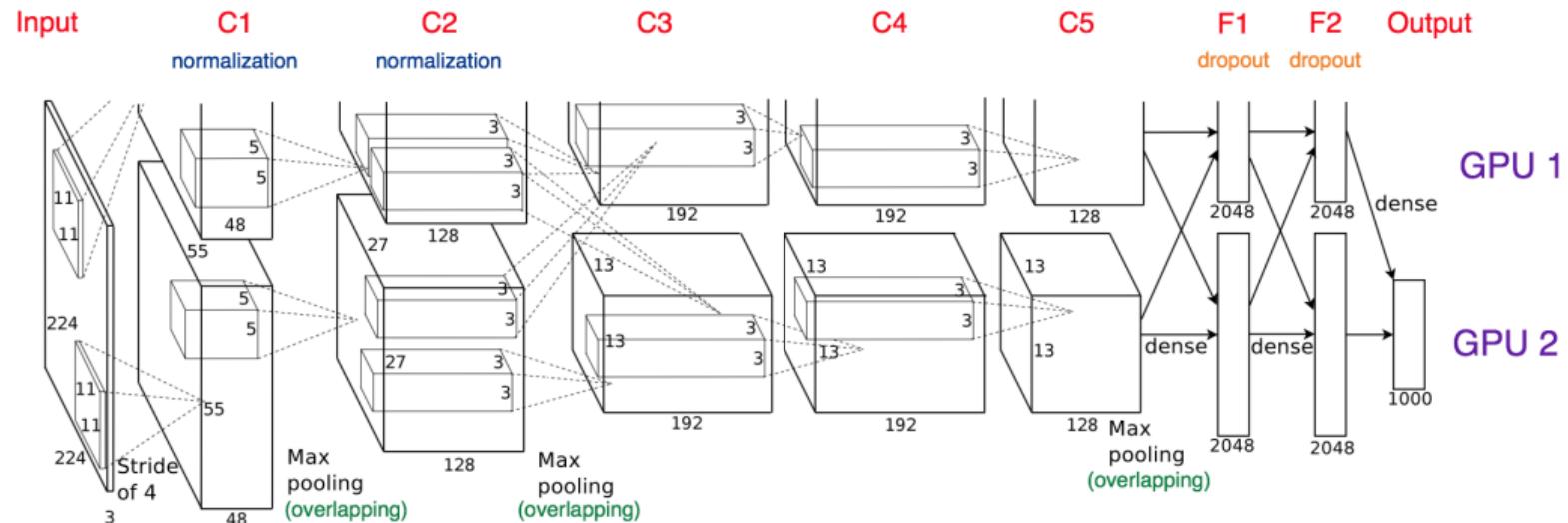


Imagen tomada de *ImageNet Classification with Deep Convolutional Neural Networks*, 2012

## AlexNet: visualización

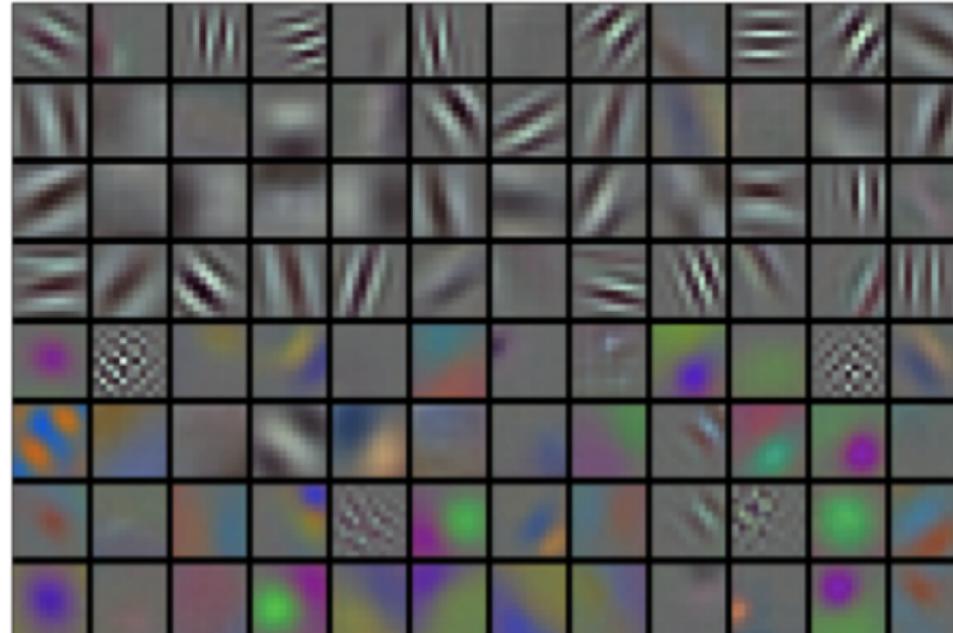


Imagen tomada de Krizhevsky et al. *ImageNet Classification with Deep Convolutional Neural Networks*, 2012

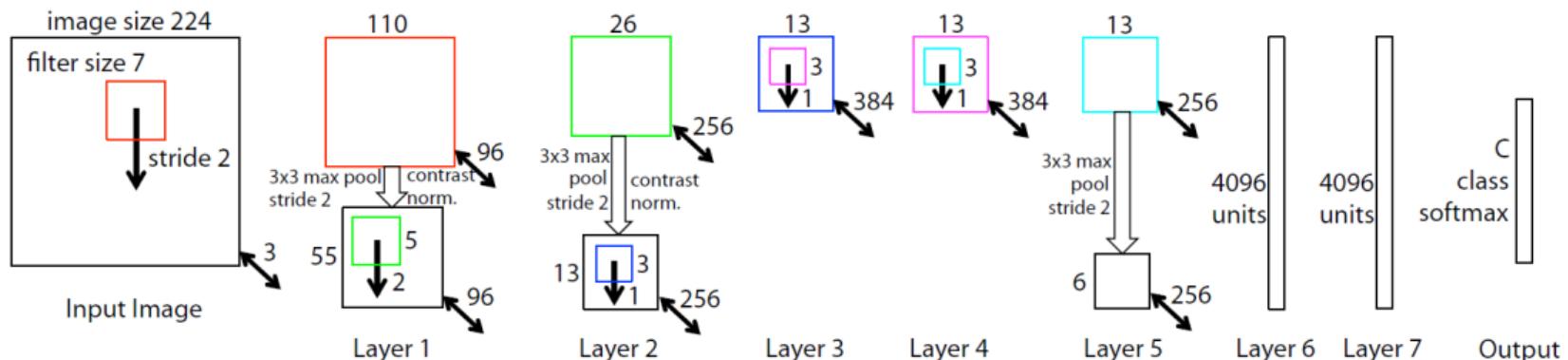


Imagen tomada de 201 Visualizing and Understanding Convolutional Networks, 2013

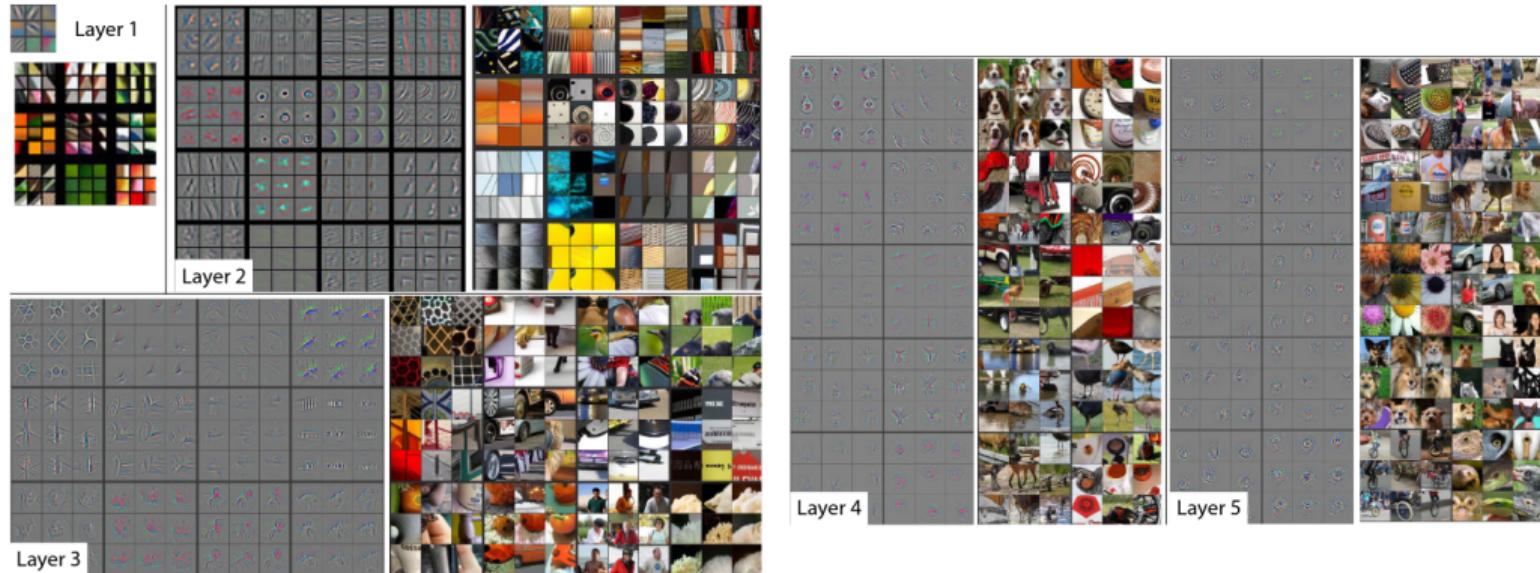


Imagen tomada de 201 Visualizing and Understanding Convolutional Networks, 2013

# VGGNets

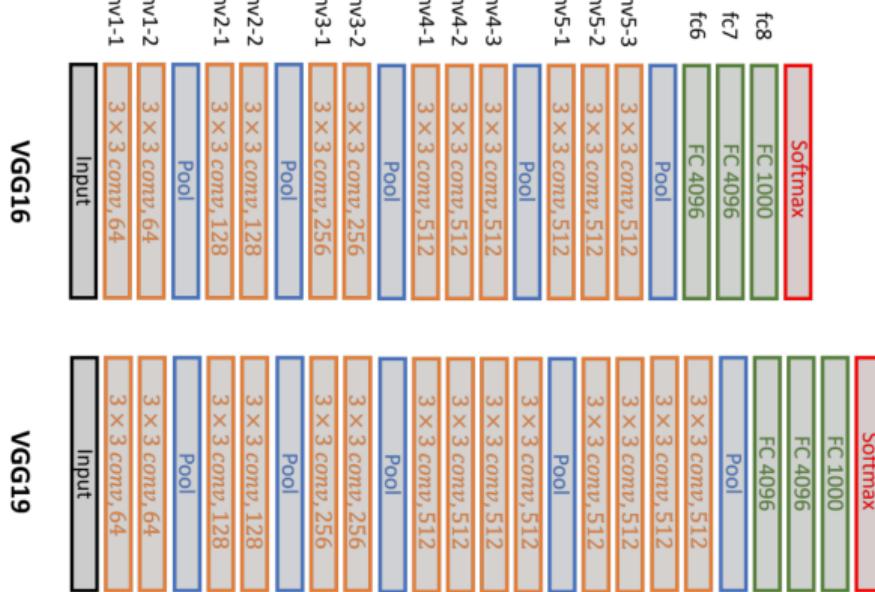


Imagen tomada de diapositivas de <http://datahacker.rs/deep-learning-vgg-16-vs-vgg-19/>

# Factorización de convoluciones

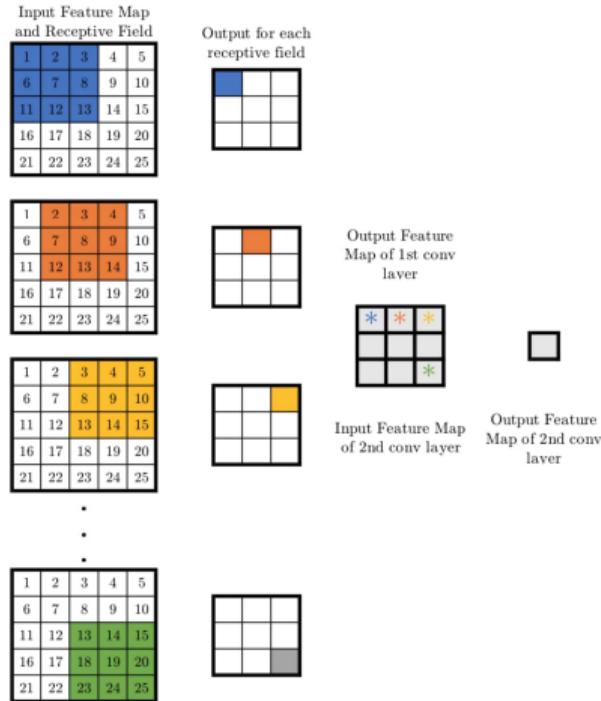


Imagen tomada de <https://inblog.in/Data-Science-Interview-Questions-30-days-of-Interview-Preparation-Day-14-asi4Xq3QNc>

¿Cómo construimos la siguiente arquitectura?



# GoogLeNet (Inception v1)

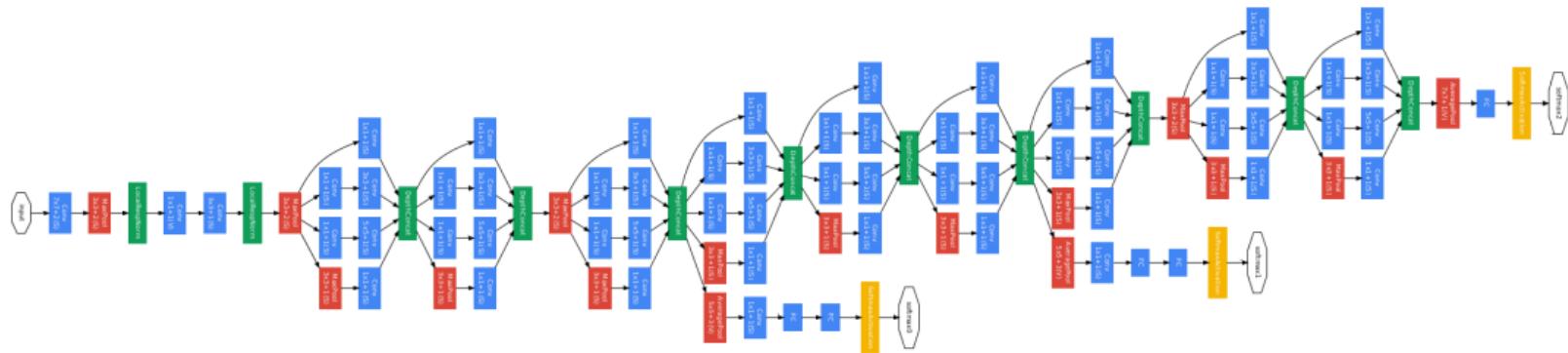


Imagen tomada de <https://inblog.in/Data-Science-Interview-Questions-30-days-of-Interview-Preparation-Day-14-asi4Xq3QNc>

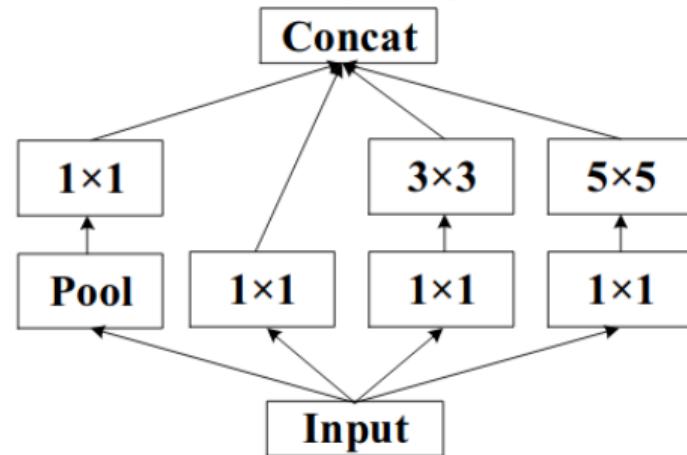


Imagen tomada de <https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html>

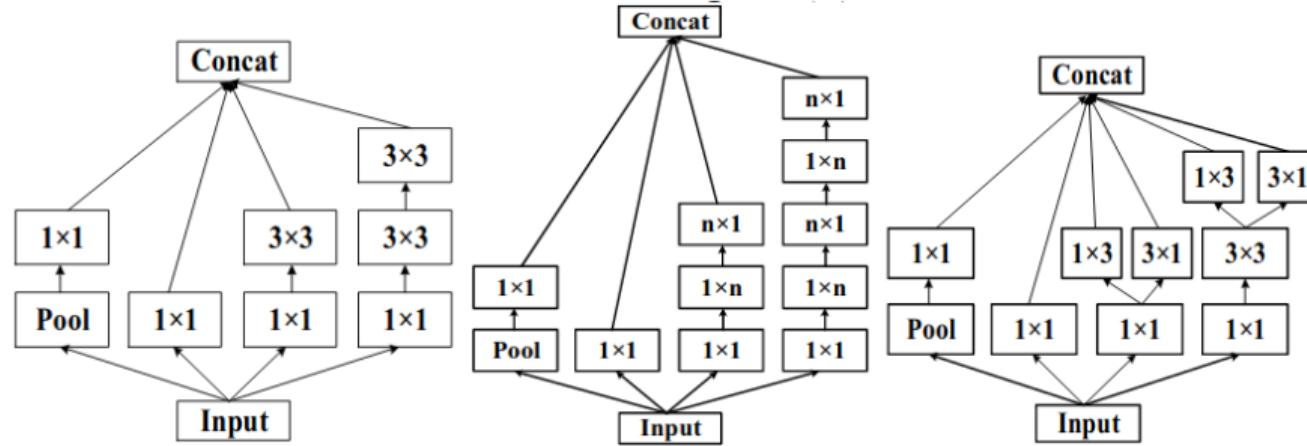


Imagen tomada de <https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html>

- Inception v3: Utiliza factorización de convoluciones de  $5 \times 5$  y  $3 \times 3$ . La entrada de la red es de  $299 \times 299$ .
- Inception v4: Utiliza más módulos Inception en comparación con la versión anterior.

Szegedy et al. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. 2016.

Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015.

## Normalización por lote

**Input:** Values of  $x$  over a mini-batch:  $\mathcal{B} = \{x_1 \dots m\}$ ;

Parameters to be learned:  $\gamma, \beta$

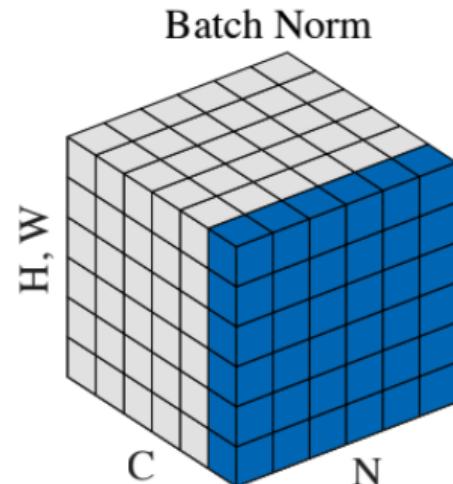
**Output:**  $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

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

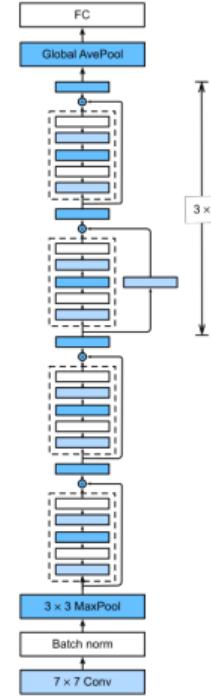
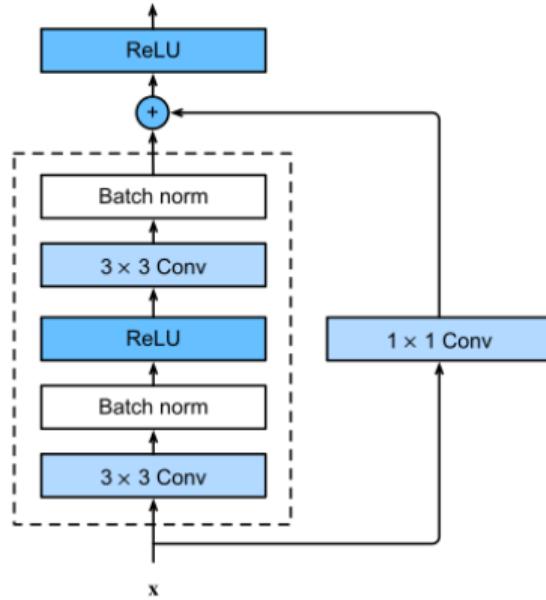
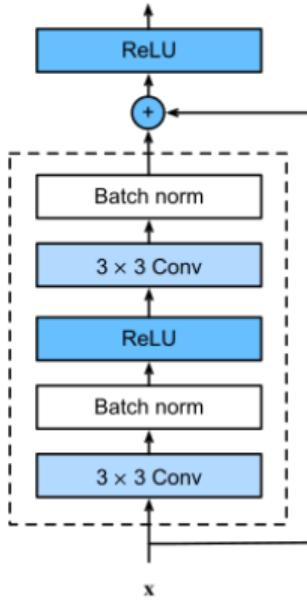
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\mathcal{B}})^2 \quad // \text{mini-batch variance}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad // \text{normalize}$$

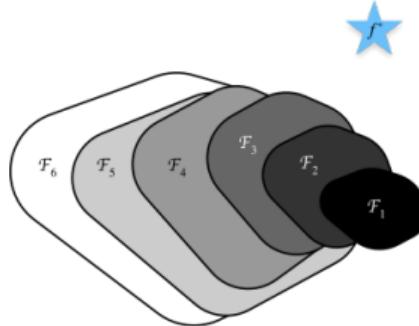
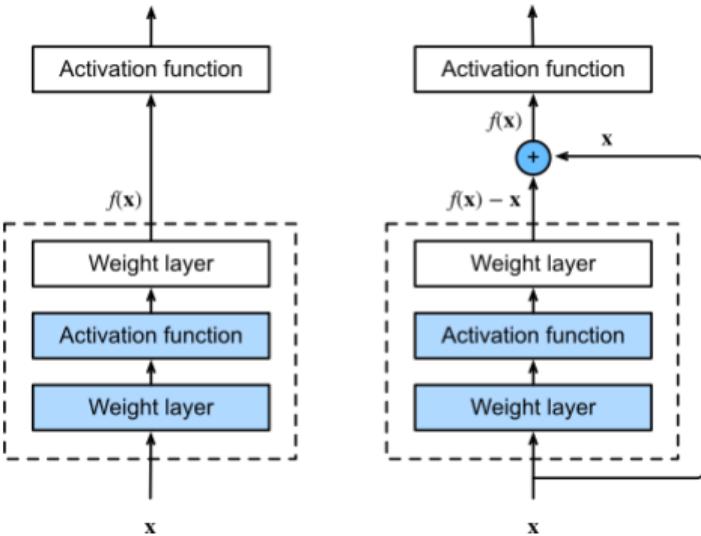
$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma, \beta}(x_i) \quad // \text{scale and shift}$$



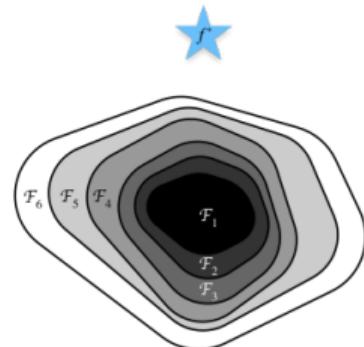
# ResNet



# Conexiones residuales



Non-nested function classes



Nested function classes

### Revolution of Depth

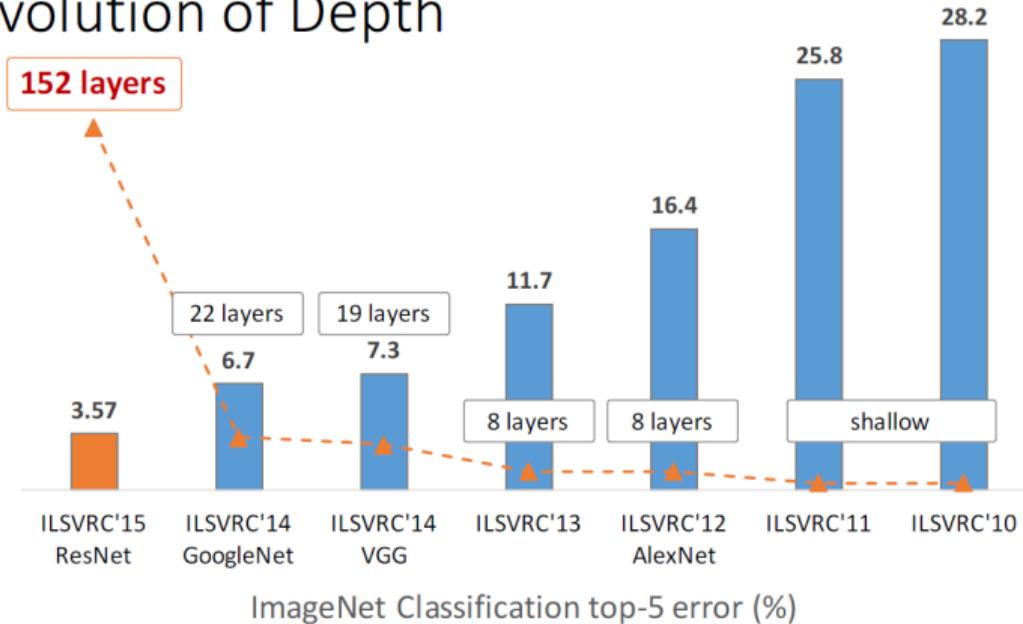


Imagen tomada de diapositivas de Kaiming He (ICML 2016)

## Profundidad en redes convolucionales (II)

- Problemas con el desvanecimiento y explosión de respuestas (hacia adelante) y gradientes (hacia atrás)

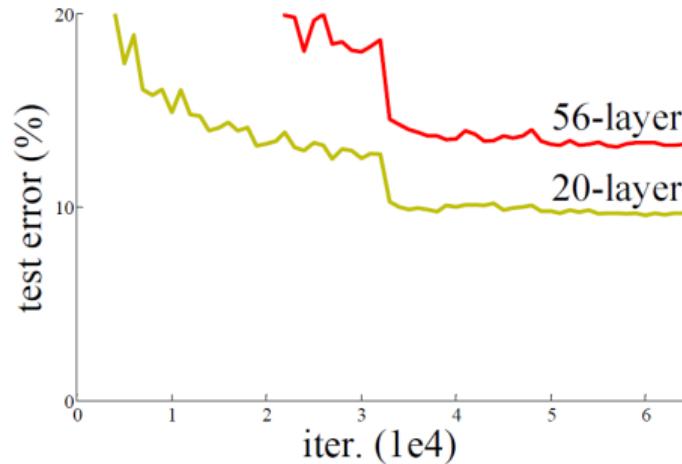
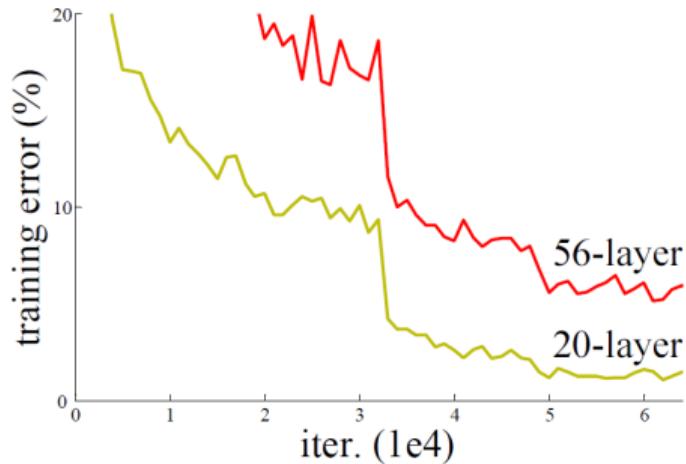


Imagen tomada de He et al. *Deep Residual Learning for Image Recognition*, 2015

# DenseNet

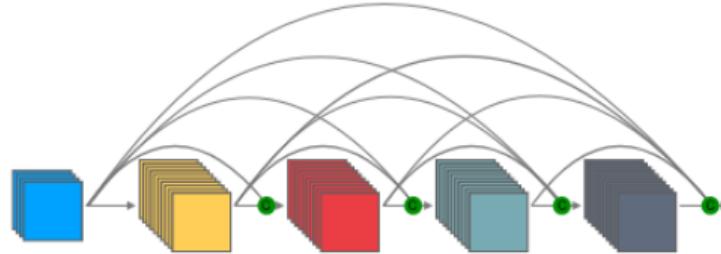


Imagen tomada de Densely Connected Convolutional Networks

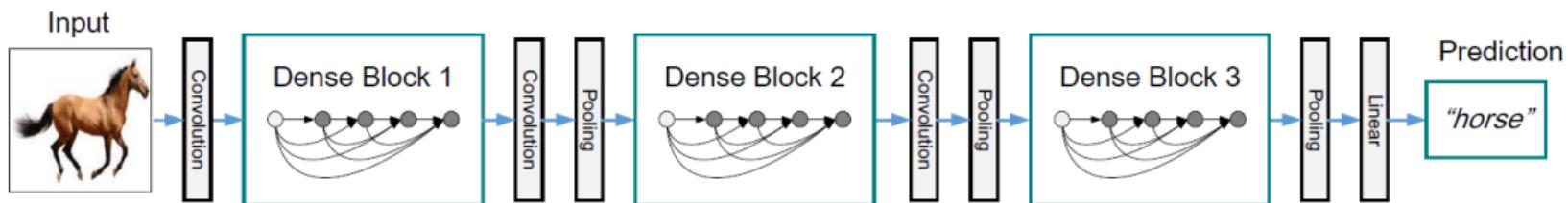


Imagen tomada de Densely Connected Convolutional Networks

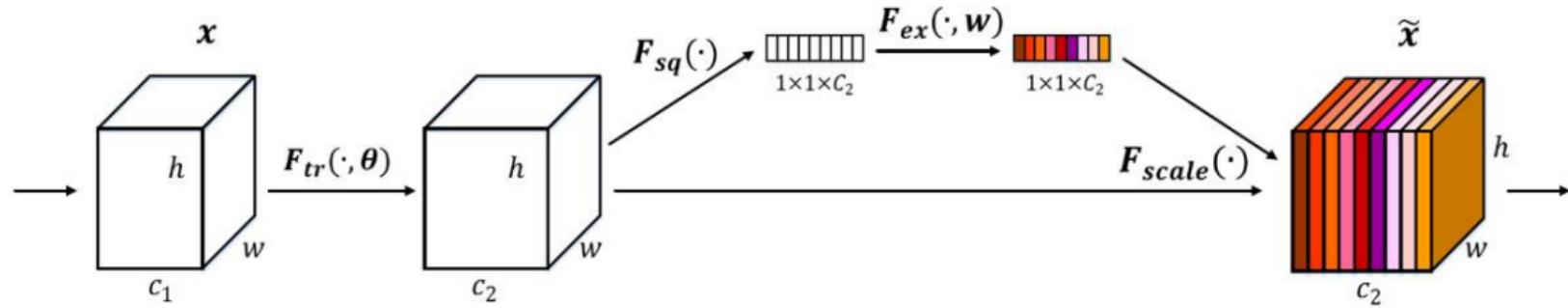


Imagen tomada de Huang et al. Densely Connected Convolutional Networks, 2018

# Aplicaciones de SENet

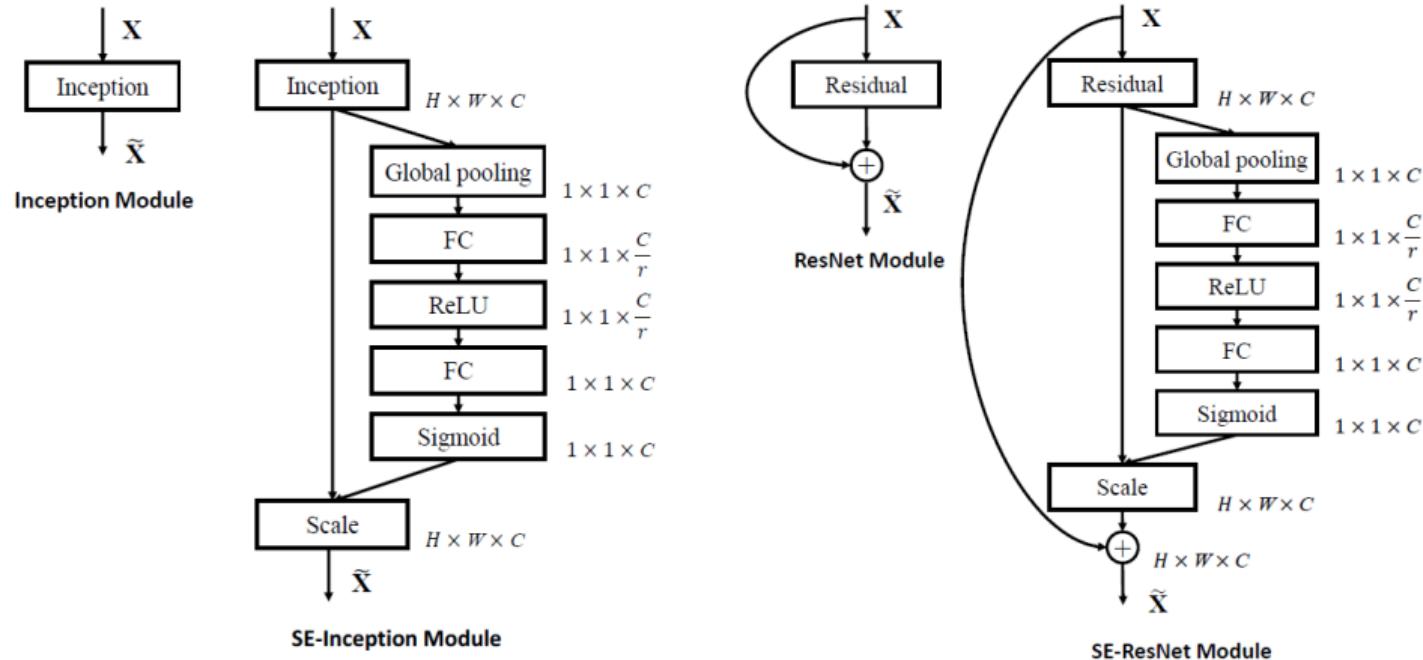


Imagen tomada de Huang et al. Densely Connected Convolutional Networks, 2018