


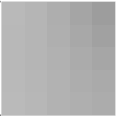


UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
Posgrado en Ciencia e Ingeniería de la Computación

APRENDIZAJE PROFUNDO  
Redes convolucionales con PyTorch

Bere & Ricardo Montalvo Lezama

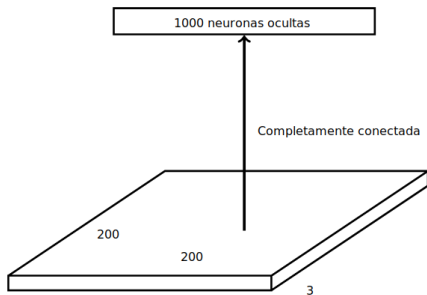
Octubre 2020

# Representación de imágenes

Imagen	Región	Representación	Tensor																																																																																																				
		<table border="1"><tr><td>10</td><td>12</td><td>30</td><td>35</td><td>40</td></tr><tr><td>10</td><td>12</td><td>17</td><td>20</td><td>42</td></tr><tr><td>10</td><td>12</td><td>12</td><td>21</td><td>46</td></tr><tr><td>10</td><td>13</td><td>25</td><td>22</td><td>36</td></tr><tr><td>10</td><td>13</td><td>15</td><td>20</td><td>58</td></tr></table>	10	12	30	35	40	10	12	17	20	42	10	12	12	21	46	10	13	25	22	36	10	13	15	20	58	Primero canal: 1, 5, 5  Último canal: 5, 5, 1																																																																											
10	12	30	35	40																																																																																																			
10	12	17	20	42																																																																																																			
10	12	12	21	46																																																																																																			
10	13	25	22	36																																																																																																			
10	13	15	20	58																																																																																																			
Chinito $224 \times 224 \times 1$	$5 \times 5$	escala de grises																																																																																																					
		<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																																																																																					Primero canal: 3, 5, 5  Último canal: 5, 5, 3
Pasita $224 \times 224 \times 3$	$5 \times 5$	RGB																																																																																																					

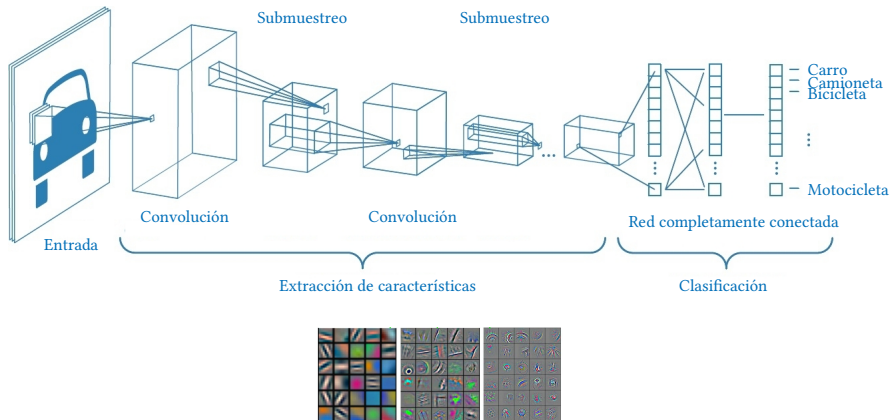
## Problemática de usar MPL para imágenes

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



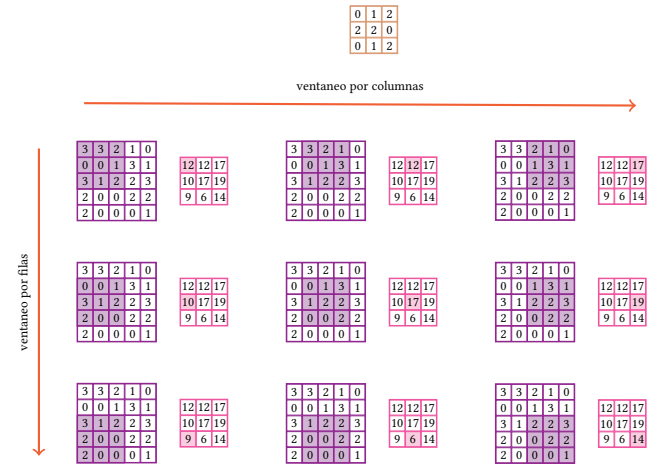
- ▶ ¡Se requieren muchos parámetros!
  - ▶ Entrada =  $200 \times 200 \times 3 = 120,000$ .
  - ▶ Parámetros =  $120,000 \times 1000 = 120,000,000$ .

# Red neuronal convolucional



Zeiler et al. *Visualizing and Understanding Convolutional Networks*. 2013.

# Convolución 2D

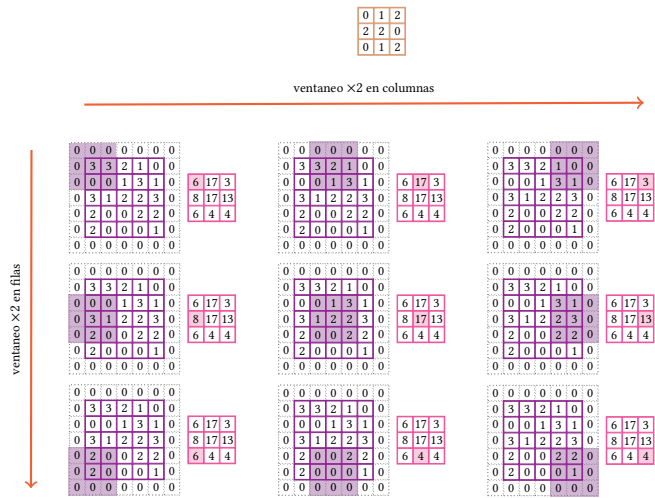


Convolución: entrada  $5 \times 5$ , salida  $3 \times 3$ , filtro  $3 \times 3$ .

# Hiperparámetros

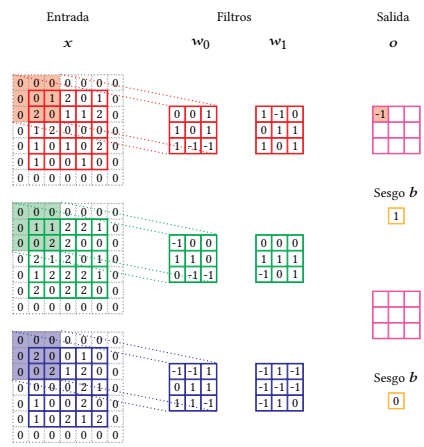
- ▶ Entrada:  $I \times H_1 \times W_1$
- ▶ Hiperparámetros:
  - ▶ Número de filtros  $K$ : profundidad de la salida.
  - ▶ Tamaño del filtro  $F$ : extensión espacial del filtro.
  - ▶ Paso  $S$ : cantidad de desplazamiento del filtro.
  - ▶ Relleno  $P$ : cantidad de aumento de ceros.
- ▶ Salida:  $O \times H_2 \times W_2$ 
  - ▶  $W_2 = \frac{(W_1 - F + 2P)}{S} + 1$
  - ▶  $H_2 = \frac{(H_1 - F + 2P)}{S} + 1$
  - ▶  $O = K$

# Convolución con relleno



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

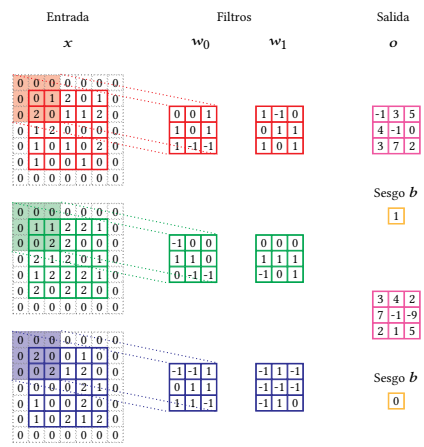
# Capa de convolución (I)



Convolución: entrada  $7 \times 7 \times 3$ , salida  $3 \times 3 \times 2$ , filtro  $3 \times 3 \times 3$ , paso  $\times 2$ , relleno  $\times 1$ .



# Capa de convolución (II)



Convolución: entrada  $7 \times 7 \times 3$ , salida  $3 \times 3 \times 2$ , filtro  $3 \times 3 \times 3$ , paso  $\times 2$ , relleno  $\times 1$ .

# Capa de convolución: ejercicio

- ▶ ¿Cuál sería 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

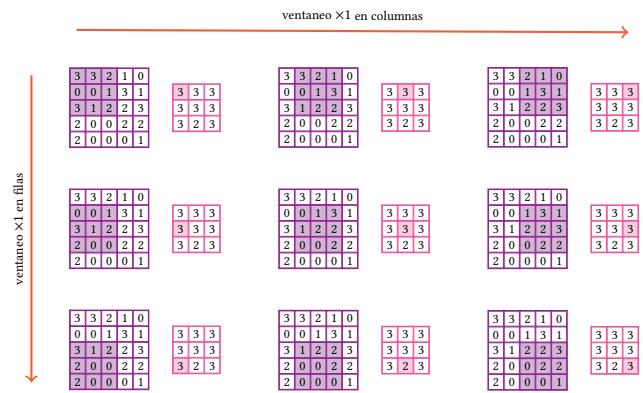
- ▶ Relleno: 1

- ▶  $W_2 = \frac{(W_1 - F + 2P)}{S} + 1$

- ▶  $H_2 = \frac{(H_1 - F + 2P)}{S} + 1$

- ▶  $O = K$

# Capa de muestreo máximo



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

# Hiperparámetros

- ▶ Reducen el tamaño de la entrada mediante el uso de alguna función para resumir subregiones.
  - ▶ Entrada:  $I \times H_1 \times W_1$
  - ▶ Hiperparámetros:
    - ▶ Tamaño del filtro  $K$ : extensión espacial del filtro.
    - ▶ Paso  $S$ : cantidad de desplazamiento del filtro.
  - ▶ Salida:  $O \times H_2 \times W_2$ 
    - ▶  $W_2 = \frac{W_1 - F}{S} + 1$
    - ▶  $H_2 = \frac{H_1 - F}{S} + 1$
    - ▶  $O = I$

# Capa de muestreo: ejercicio

- ▶ ¿Cuál sería las dimensiones del bloque de salida para una capa de muestreo con siguientes características?

- ▶ Entrada:  $4 \times 28 \times 28$

- ▶ Tamaño del filtro: 2

- ▶ Salto: 2

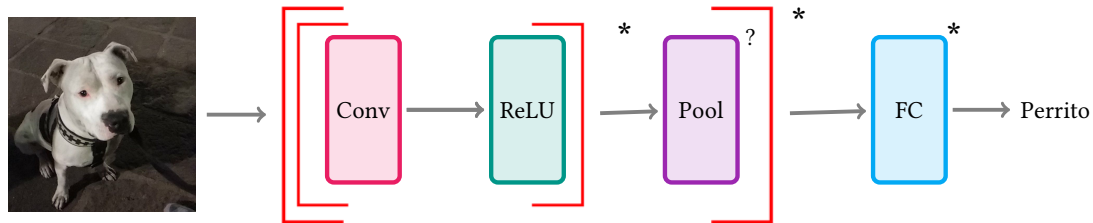
- ▶  $W_2 = \frac{W_1 - F}{S} + 1$

- ▶  $H_2 = \frac{H_1 - F}{S} + 1$

- ▶  $O = I$

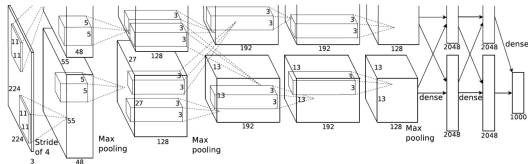
# Arquitecturas de CNNs

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

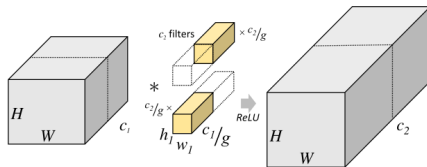


# Convoluciones paralelas

- Aprenden representaciones complementarias por medio de dispersión<sup>2</sup>.



AlexNet<sup>1</sup>

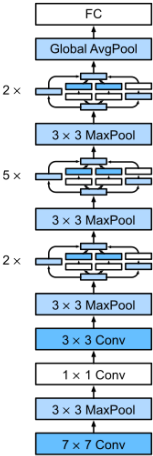
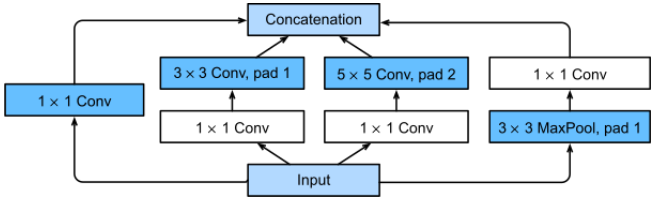


Convoluciones Agrupadas<sup>2</sup>

1. Zhang et al. *ImageNet Classification with Deep Convolutional Neural Networks*. 2012.

2. Ioannou et al. *Deep Roots: Improving CNN Efficiency with Hierarchical Filter Groups*. 2016.

# GoogLeNet

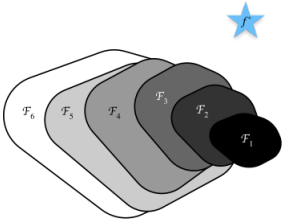
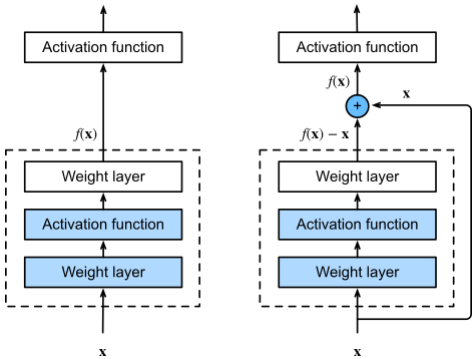


Zhang et al. *Dive into Deep Learning*. 2020.

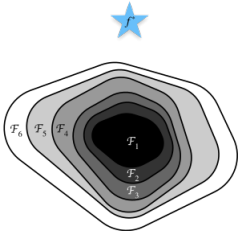
Szegedy et al. *Going Deeper with Convolutions*. 2015



# Conexiones residuales



Non-nested function classes

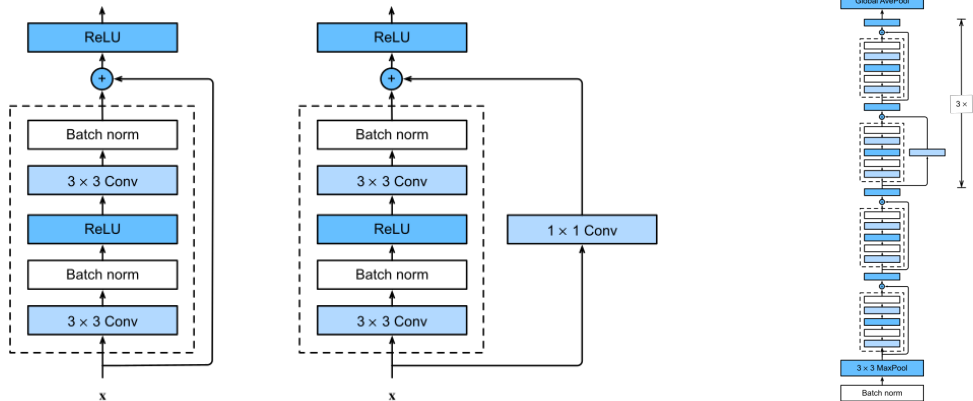


Nested function classes

Zhang et al. *Dive into Deep Learning*. 2020.

He et al. *Deep Residual Learning for Image Recognition*. 2015

# ResNet



Zhang et al. *Dive into Deep Learning*. 2020.

He et al. *Deep Residual Learning for Image Recognition*. 2015

# Normalización por lote

**Input:** Values of  $x$  over a mini-batch:  $\mathcal{B} = \{x_{1\dots m}\}$ ;

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

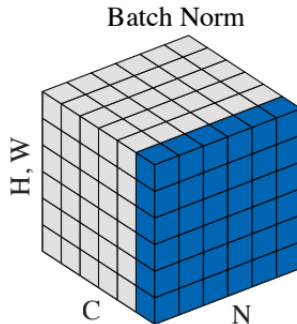
**Output:**  $\{y_i = \text{BN}_{\gamma, \beta}(x_i)\}$

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

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

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

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



Ioffe et al. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015

Wu et al. *Group normalization*. 2018