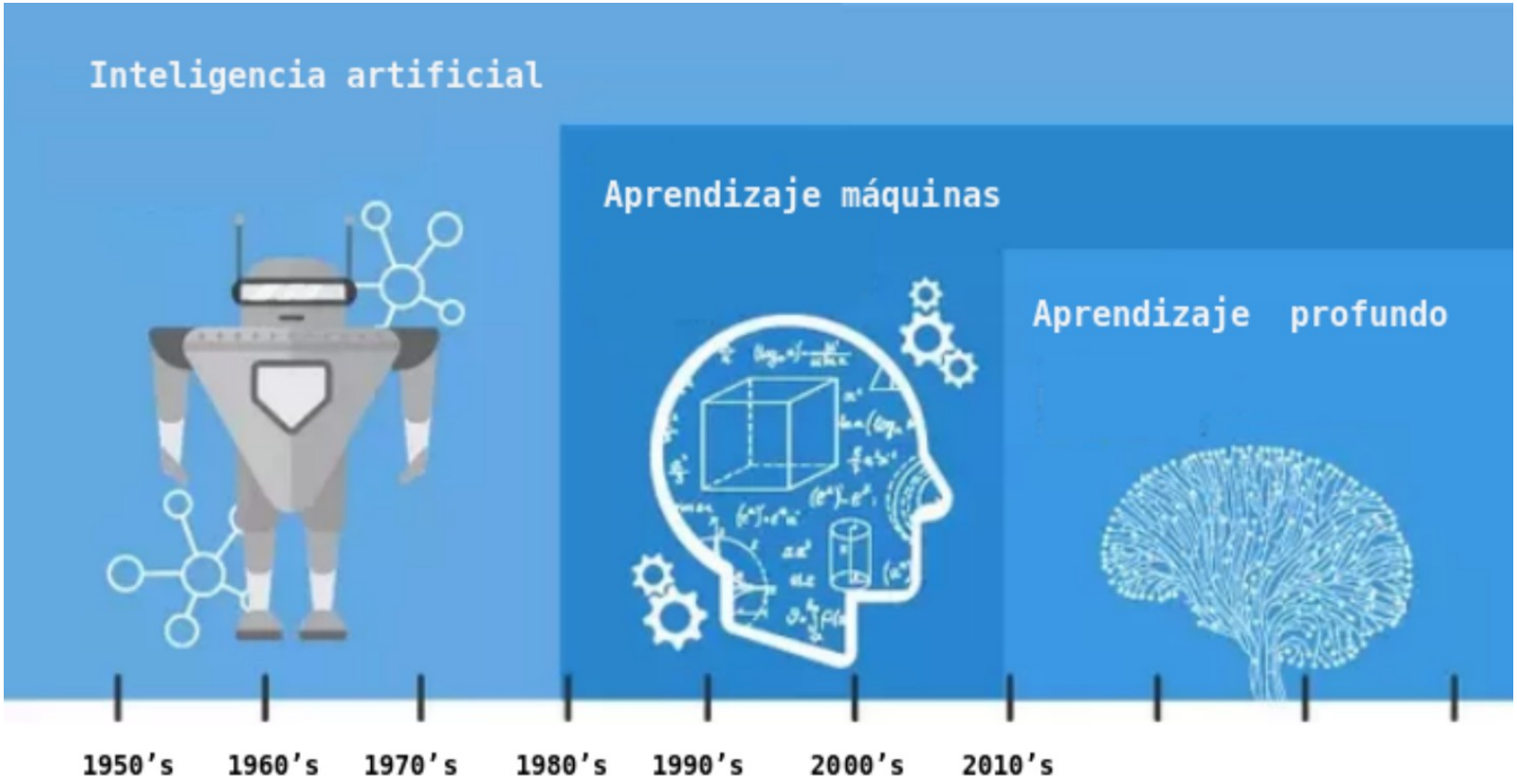
Introducción al Aprendizaje Profundo

Bere & Ricardo Montalvo Lezama

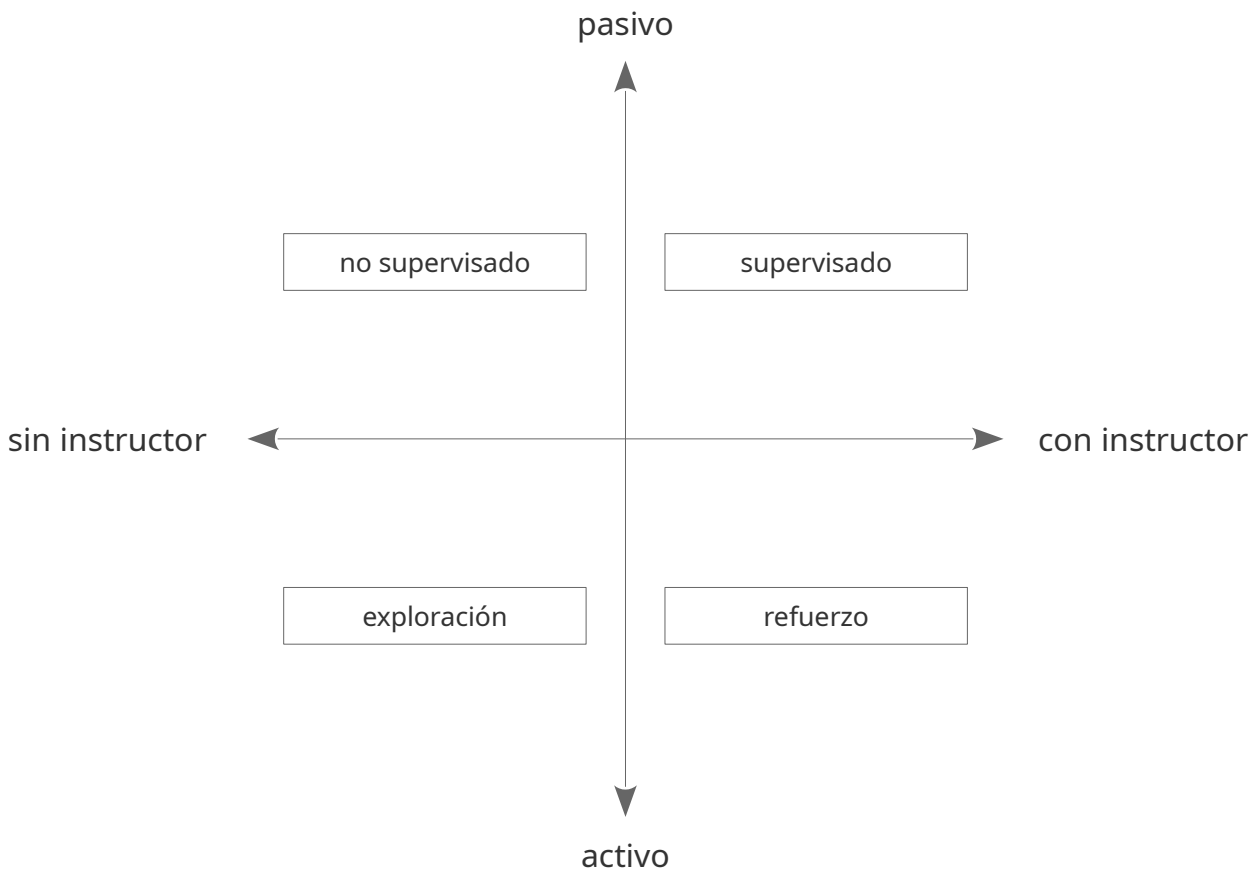
`github.com/bereml/riiaa-20-mtl`

Agosto 2020

IA, AM y AP

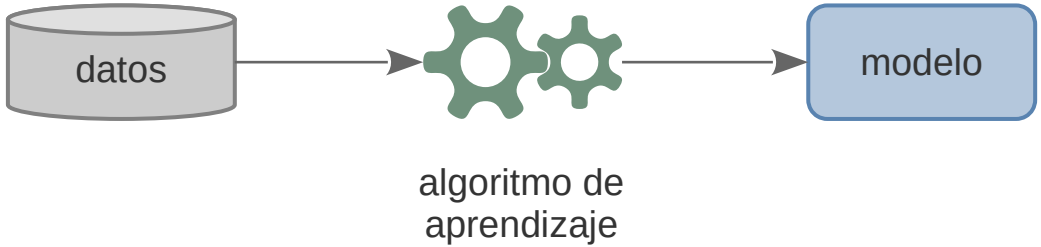

Inteligencia artificial
Aprendizaje máquinas
Aprendizaje profundo
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Tipos de aprendizaje



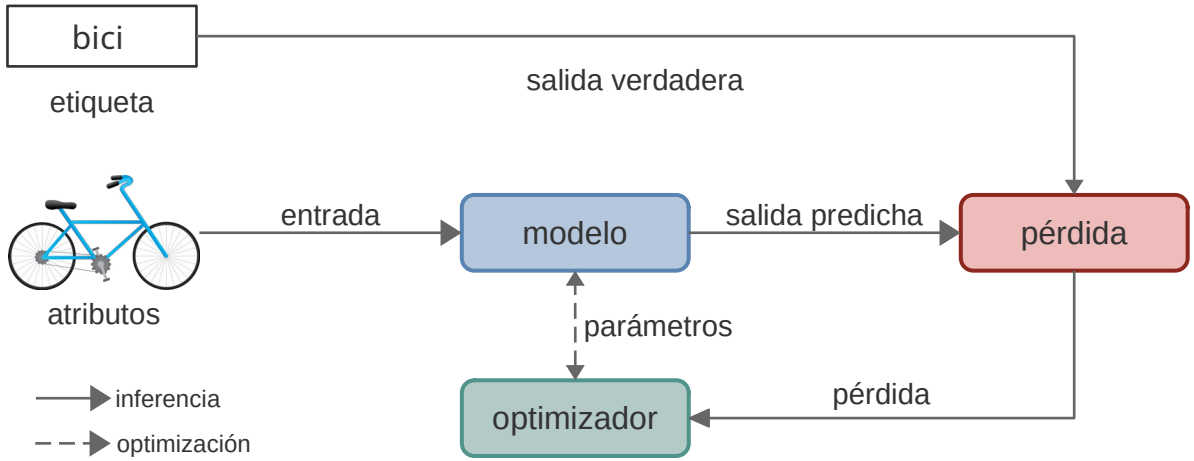
¿Cómo funciona el aprendizaje de máquinas?

- Programas que aprenden a partir de ejemplos.
- Se aprende un modelo: función parámetrizada.



Aprendizaje supervisado

- La función de pérdida compara la salida verdadera con la salida predicha.
- Se minimiza la pérdida para actualizar los parámetros del modelo.
- El más común de los tipos de aprendizaje.



Una receta con mucho exito

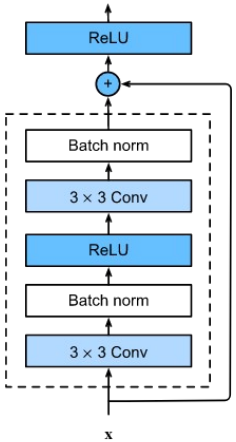
IMAGENET

14M de imágenes
21K categorías

YouTube-8M

8M de videos
4.8K categorías

+

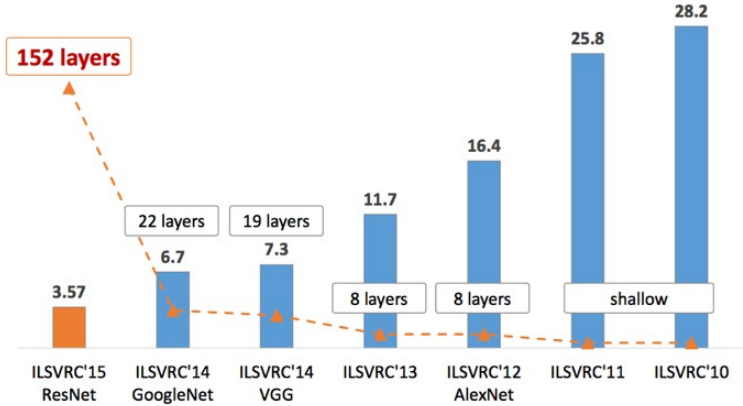


+

PyTorch

TensorFlow

=



datos masivos etiquetados

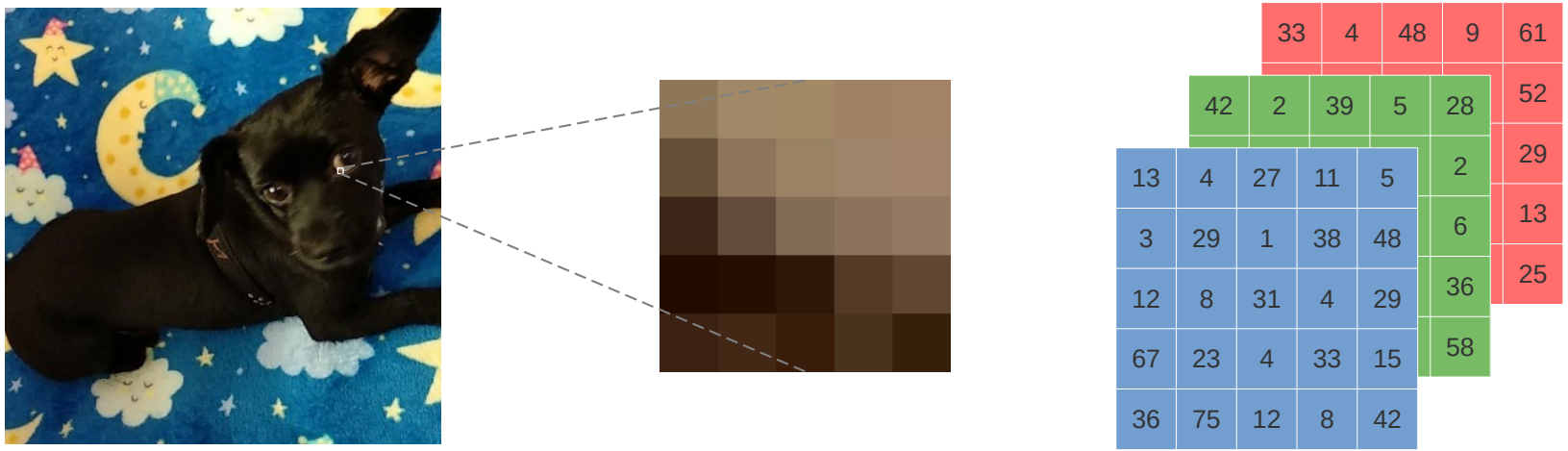
arquitecturas sofisticadas

infraestructura

estado del arte

Representación de imágenes

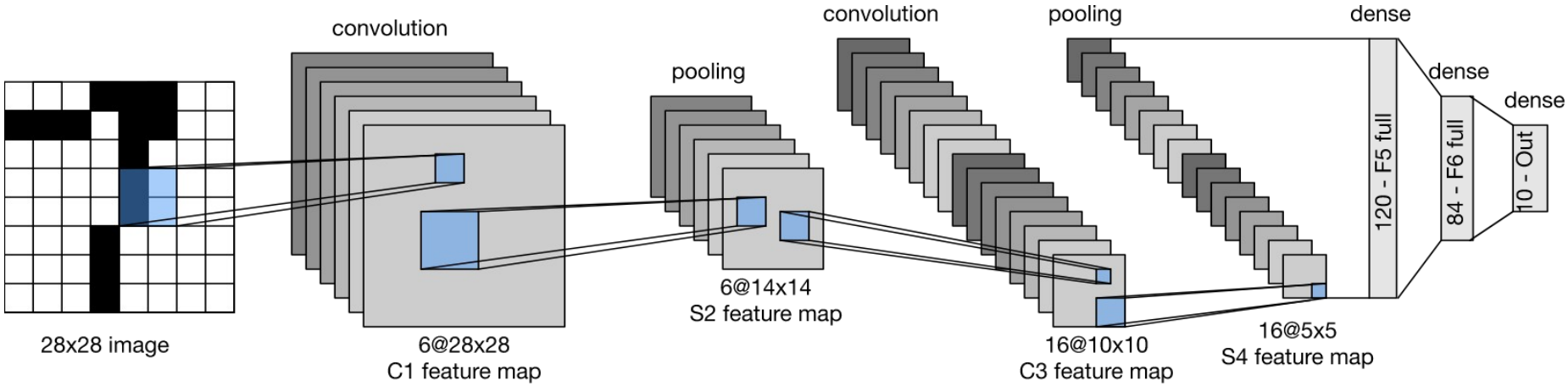
- Se representan con una matriz de valores de píxeles por cada color.



3x224x224 3x5x5 canales RGB

- MNIST 1x28x28, CIFAR-10 3x32x32, ImageNet 3x256x256.

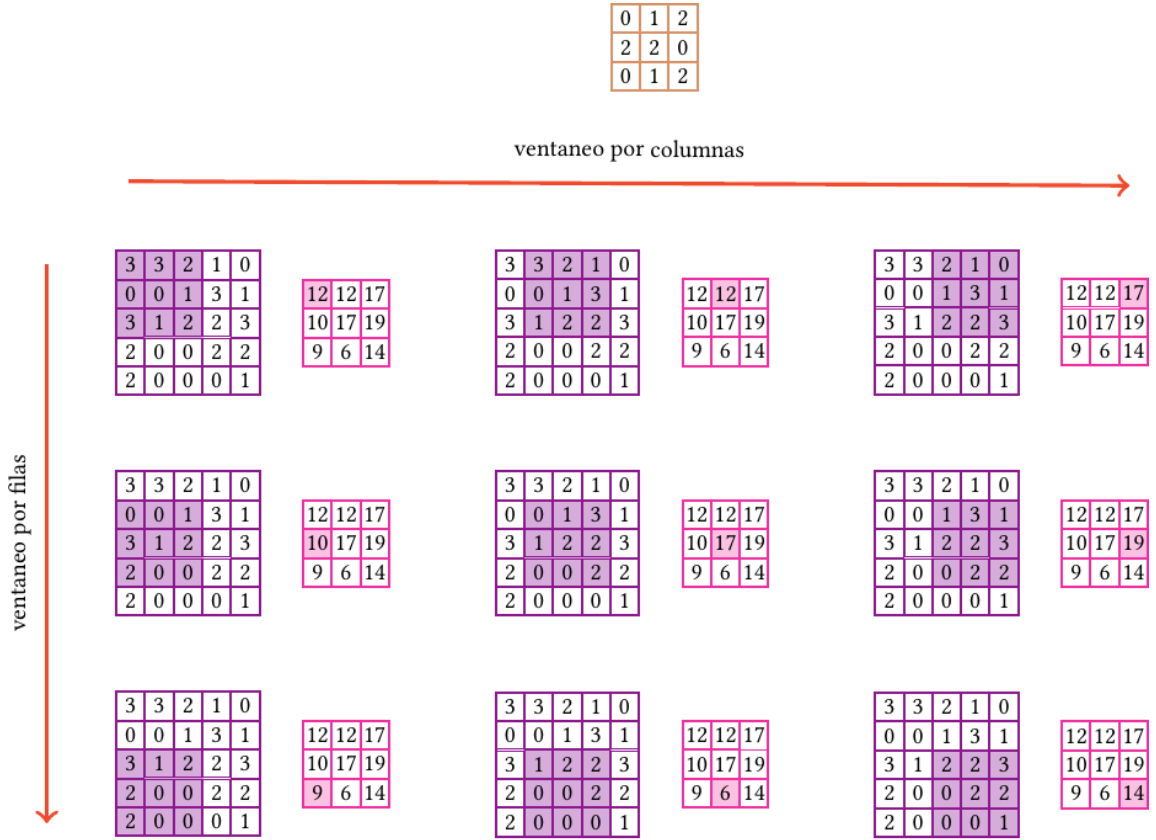
Redes Convolucionales



Arquitectura LeNet

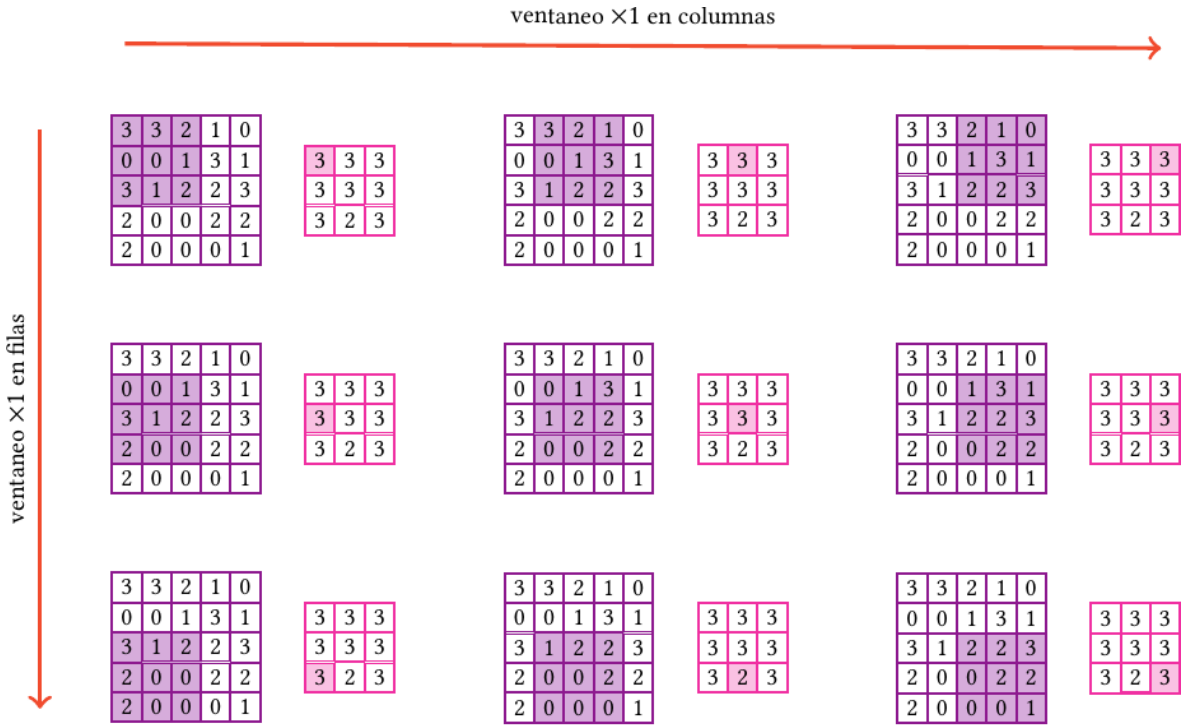
Zhang et al. Dive into Deep Learning. 2019.

Convolución



Convolución: entrada 5×5, salida 3×3, filtro 3x3.

Muestreo



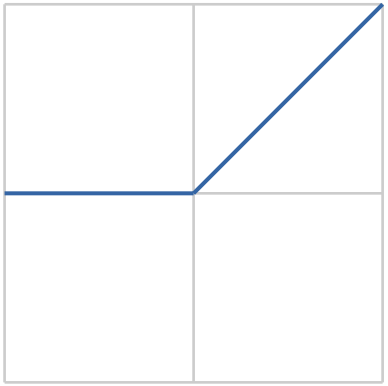
Muestreo máximo: entrada 5×5, salida 3×3, paso 1x1.

Funciones de activación (I)

- Función no lineal a la salida de la neurona.
  - ReLU: capas intermedias.
  - Sigmoide: clasificación binaria.
  - Softmax: clasificación multiclase.

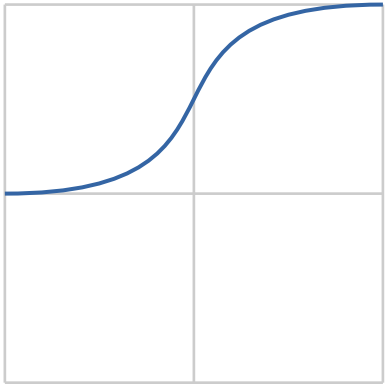
ReLu

$$relu(x) = max(0, x)$$



Sigmoide

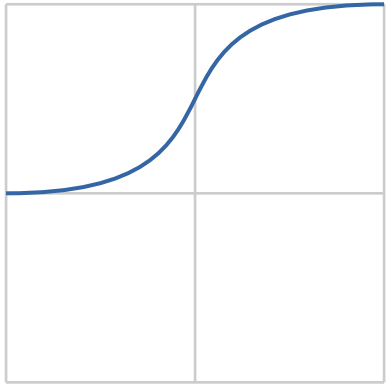
$$sigmoid(x) = \frac{1}{1+\exp(-x)}$$



Softmax

$$z = softmax(x)$$

$$z_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)}$$



Normalización por lote

- Control de la media y la varianza en etapas de la red.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

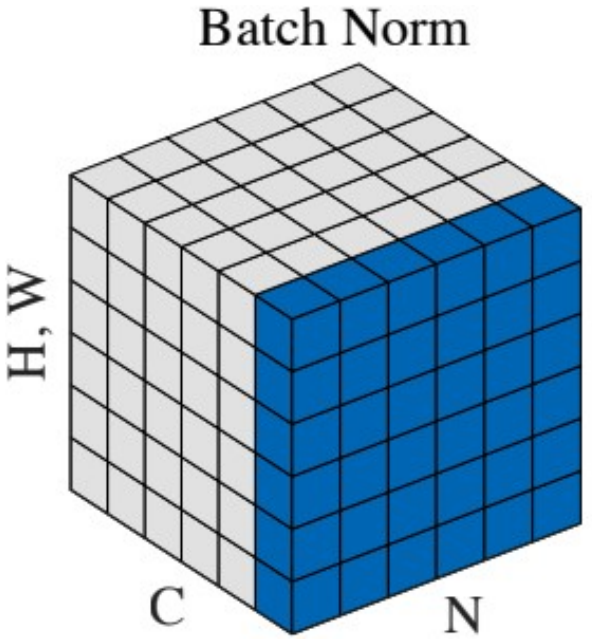
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

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

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

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

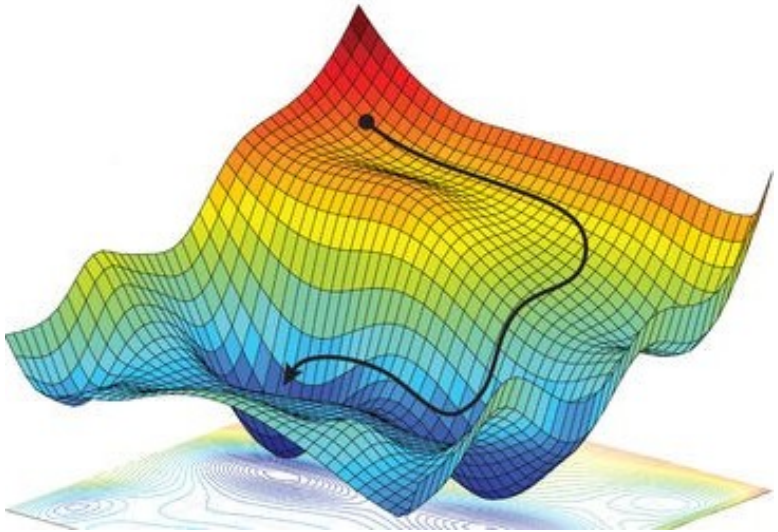
Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.



Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015.

Descenso por gradiente

- Avanzar hacia la dirección con menor pérdida.



repetir hasta converger:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

¡tiempo de programar!

1a_cnn.ipynb

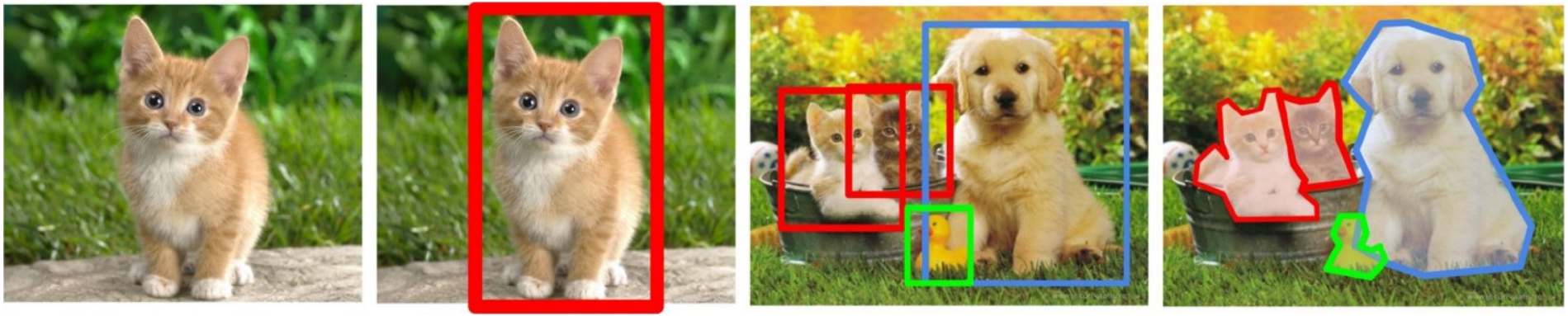
Tareas de Visión

clasificación

clasificación
+ localización

detección

segmentación



gato

gato

gato, pato, pato

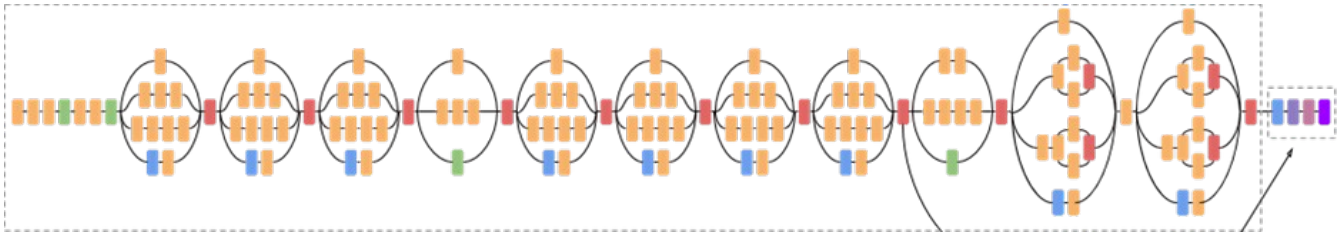
gato, pato, pato

un objeto

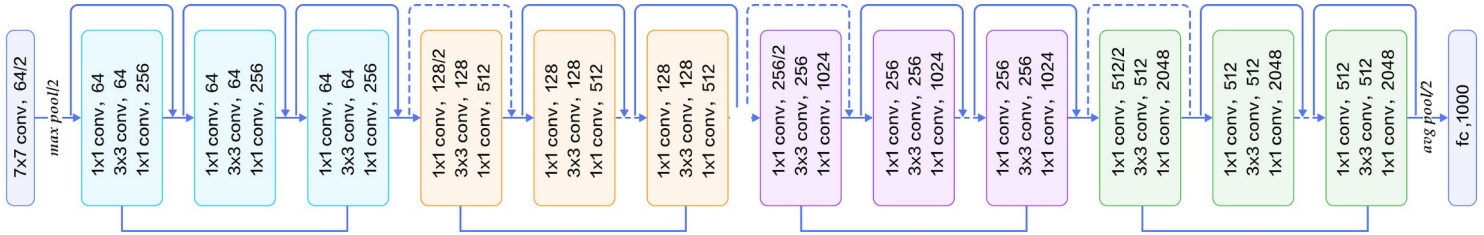
múltiples objetos

Arquitecturas

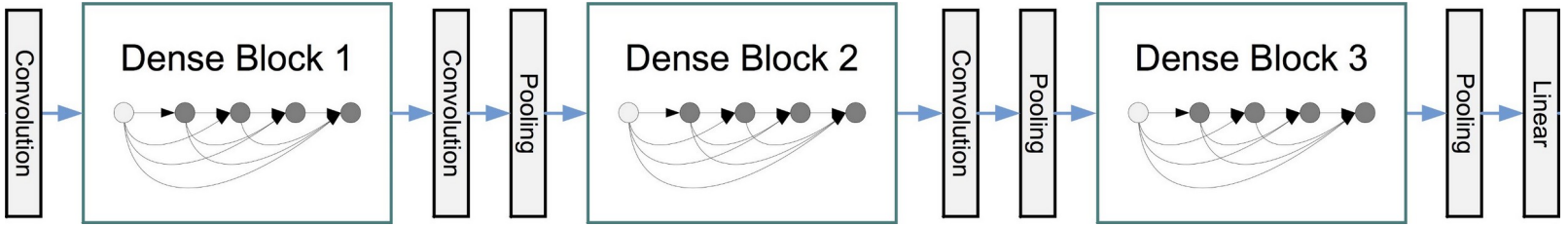
Inception



ResNet



DenseNet



¡Gracias!

Ricardo Montalvo Lezama

http://turing.iimas.unam.mx/~ricardoml/

ricardoml@turing.iimas.unam.mx

Bere Montalvo Lezama

http://turing.iimas.unam.mx/~bereml/

bereml@turing.iimas.unam.mx