

# Introducción al Aprendizaje Profundo

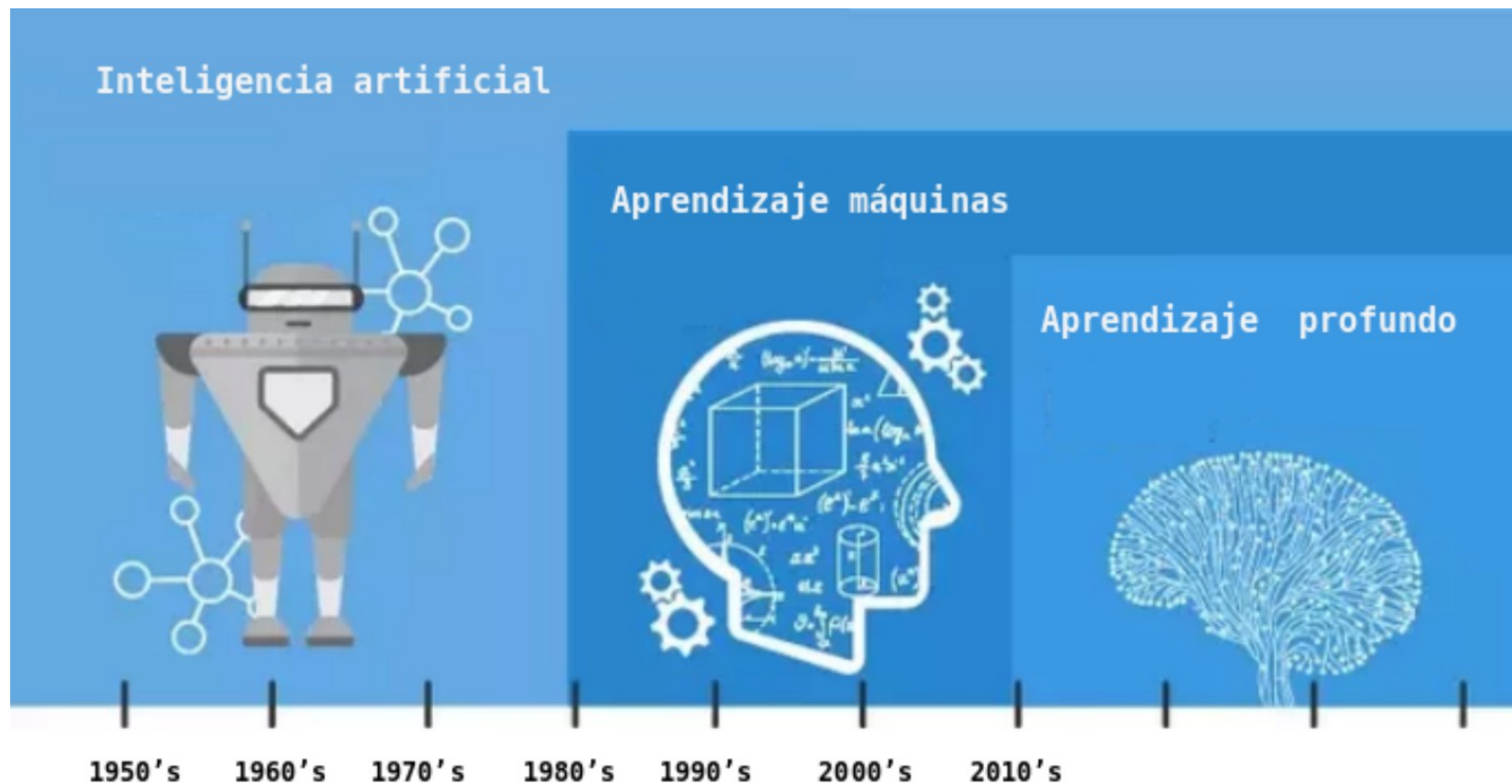
Bere & Ricardo Montalvo Lezama

[github.com/richardtml/riiaa-20-aa](https://github.com/richardtml/riiaa-20-aa)

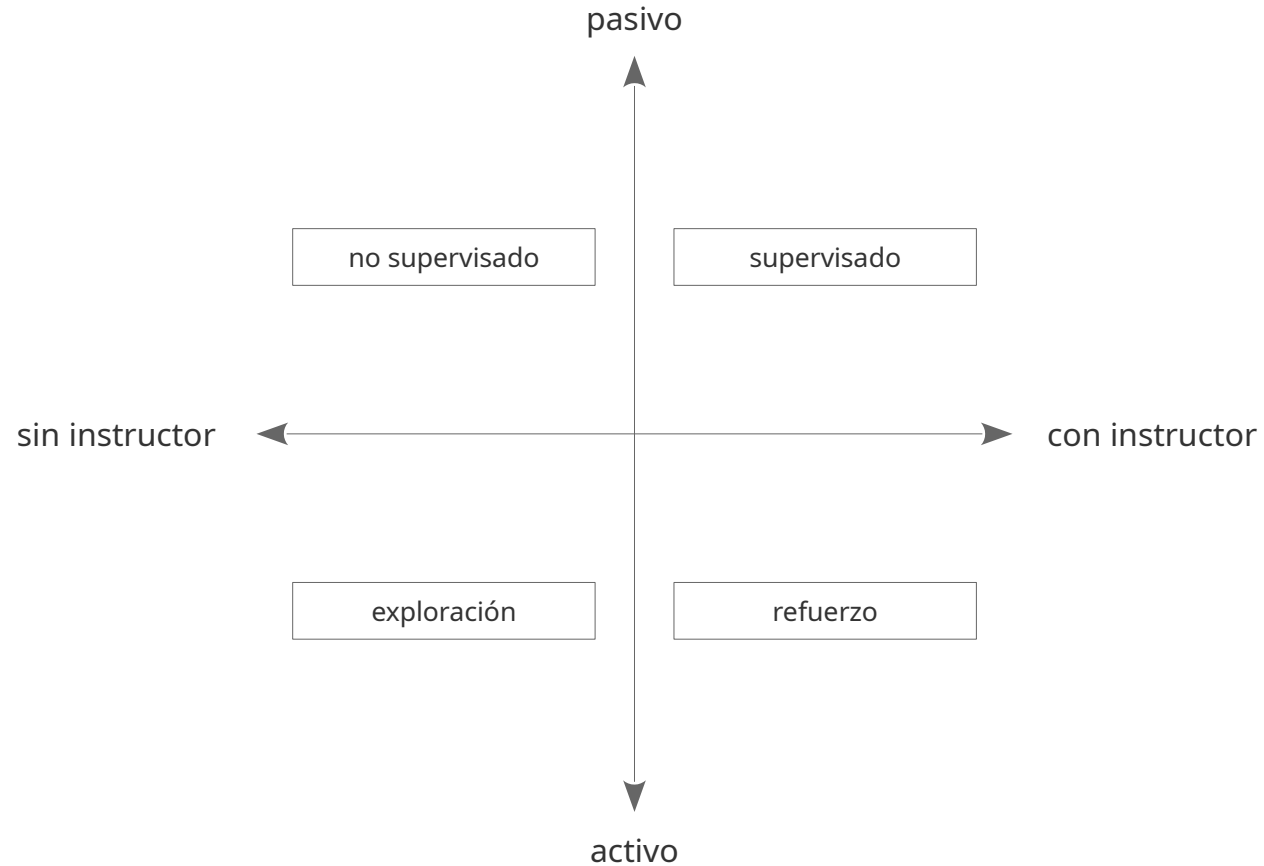


Agosto 2020

# IA, AM y AP



# Tipos de aprendizaje



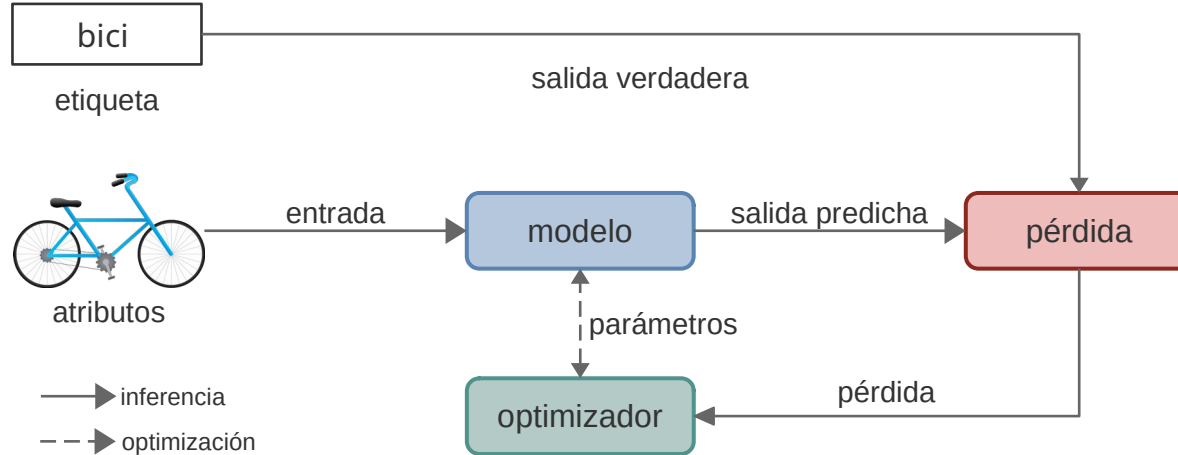
# ¿Cómo funciona el aprendizaje de máquinas?

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



# 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 éxito

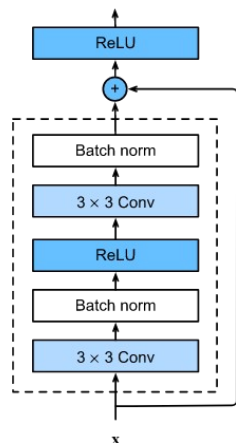
IMAGENET

14M de imágenes  
21K categorías

YouTube-8M

8M de videos  
4.8K categorías

+



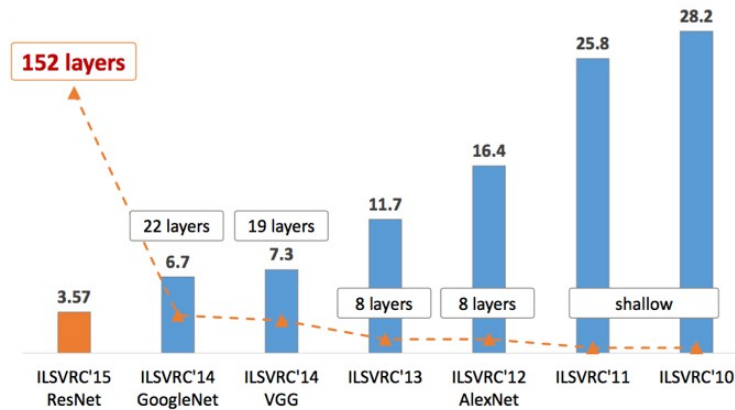
+



PyTorch

=

TensorFlow



datos masivos  
etiquetados

arquitecturas  
sofisticadas

infraestructura

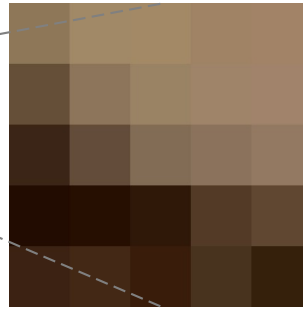
estado  
del arte

# Representación de imágenes

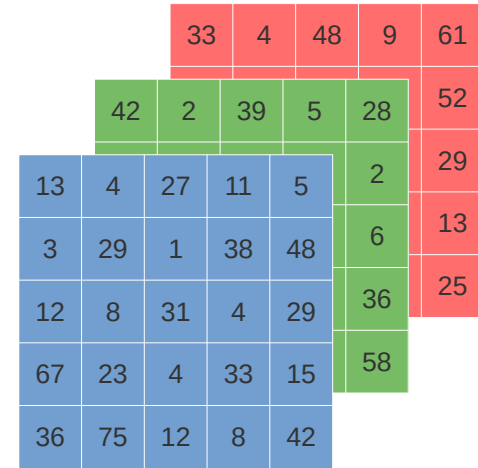
- Se representan con una matriz de valores de píxeles por cada color.



3x224x224



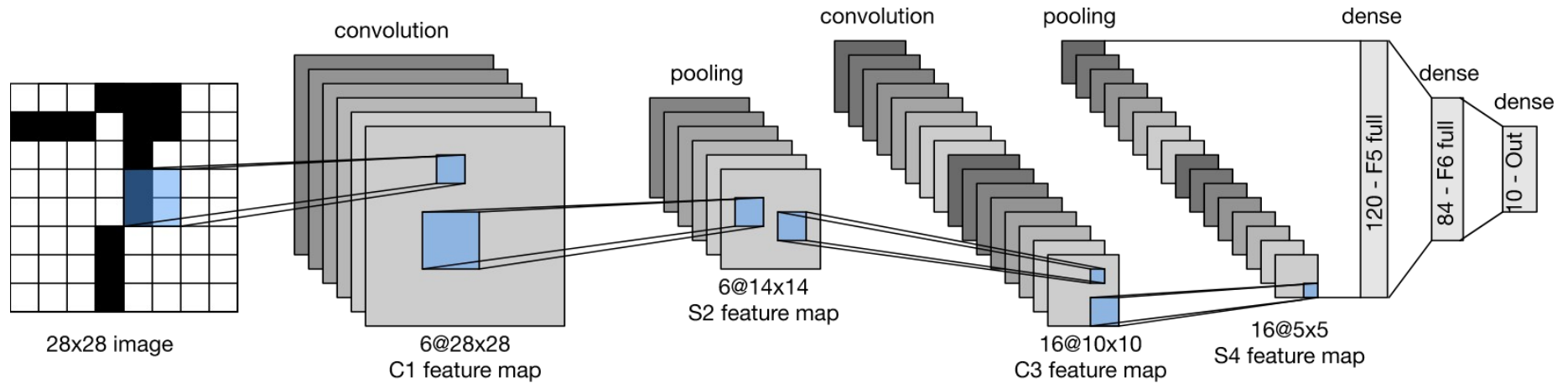
3x5x5



canales RGB

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

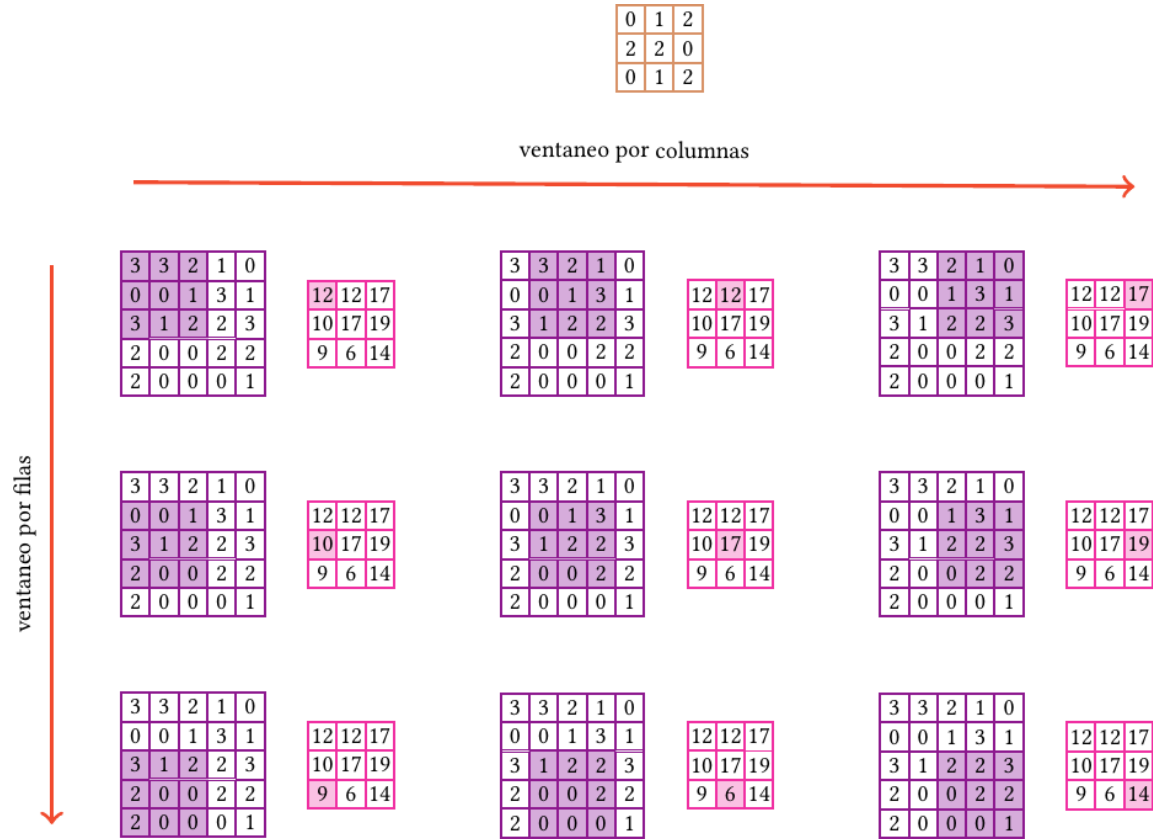
# Redes Convolucionales



Arquitectura LeNet

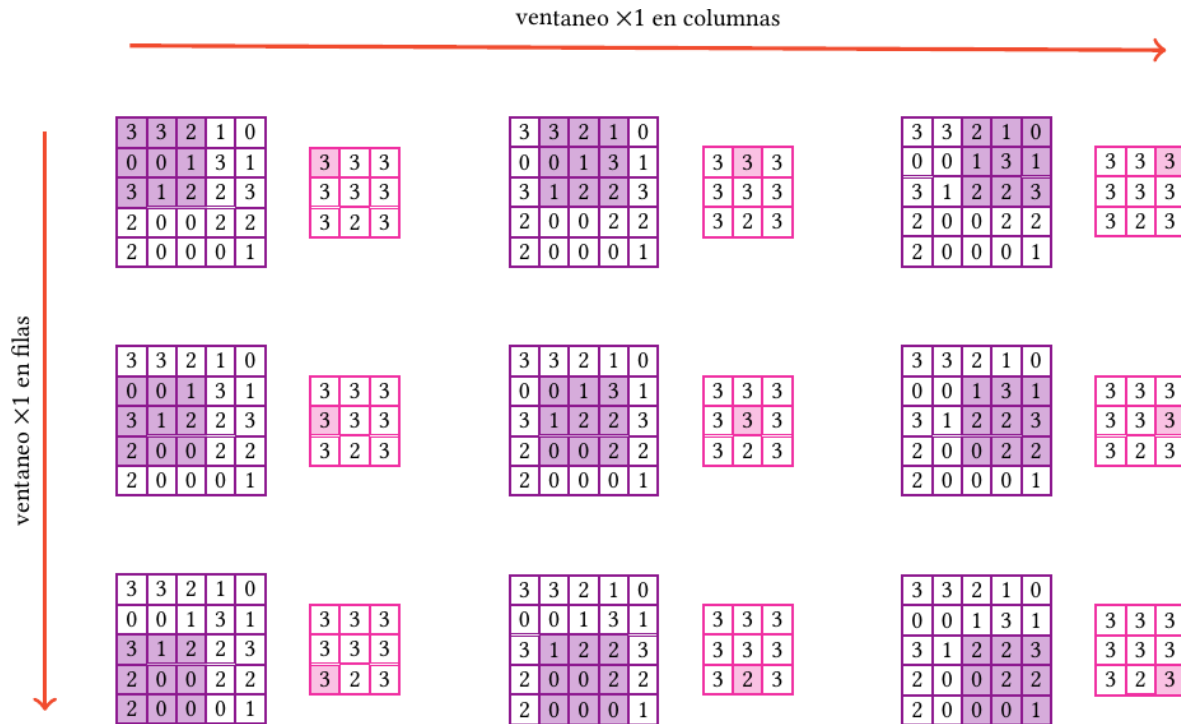


# Convolución



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

# Muestreo



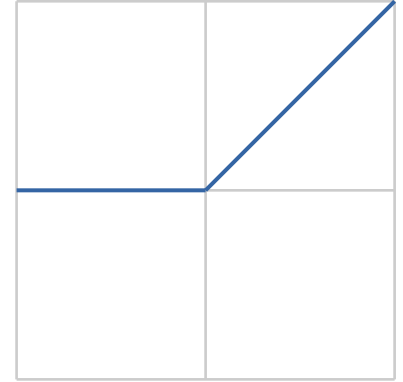
Muestreo máximo: entrada  $5 \times 5$ , salida  $3 \times 3$ , paso  $1 \times 1$ .

# Funciones de activación (I)

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

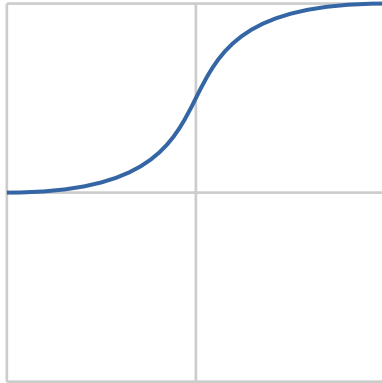
ReLU

$$\text{relu}(x) = \max(0, x)$$



Sigmoide

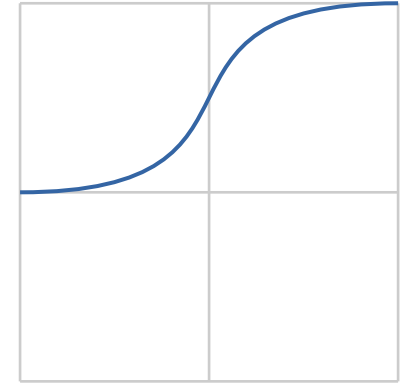
$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}$$



Softmax

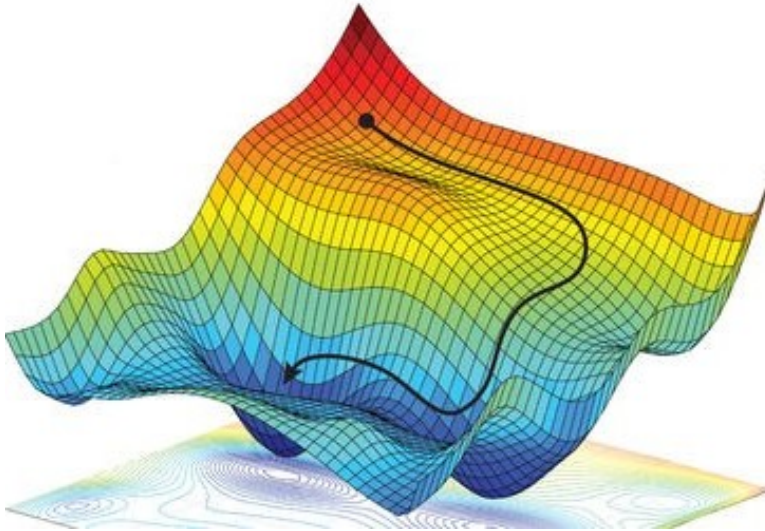
$$z = \text{softmax}(x)$$

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



# 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



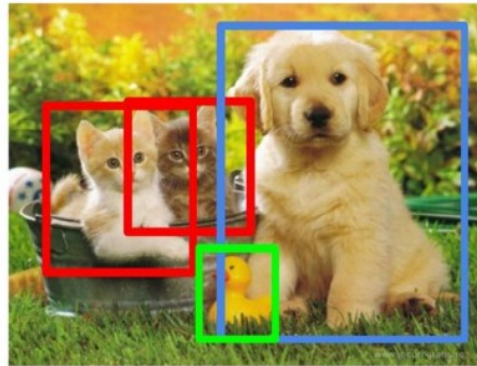
gato

clasificación  
+ localización



gato

detección



gato, pato, pato

segmentación



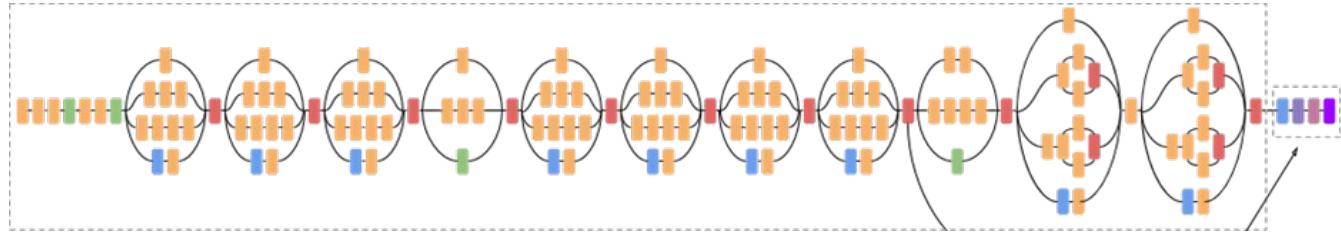
gato, pato, pato

un objeto

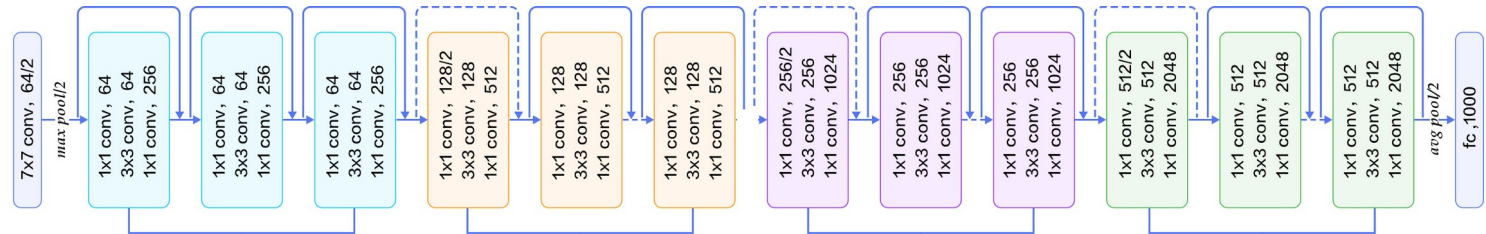
múltiples objetos

# Arquitecturas

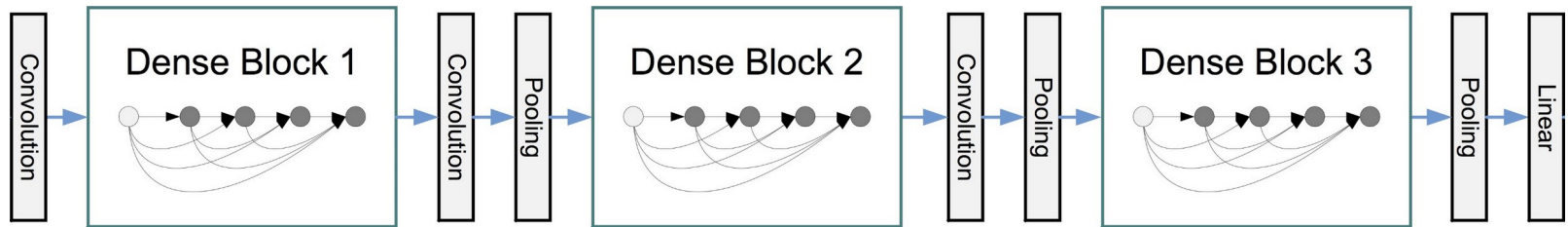
Inception



ResNet



DenseNet







# ¡Gracias!

**Ricardo Montalvo Lezama**

<http://turing.iimas.unam.mx/~ricardoml/>  
[ricardoml@turing.iimas.unam.mx](mailto:ricardoml@turing.iimas.unam.mx)

**Bere Montalvo Lezama**

<http://turing.iimas.unam.mx/~bereml/>  
[bereml@turing.iimas.unam.mx](mailto:bereml@turing.iimas.unam.mx)