Introducción al Aprendizaje Profundo

Bere & Ricardo Montalvo Lezama

`github.com/richardtml/riiaa-20-aa`

Agosto 2020

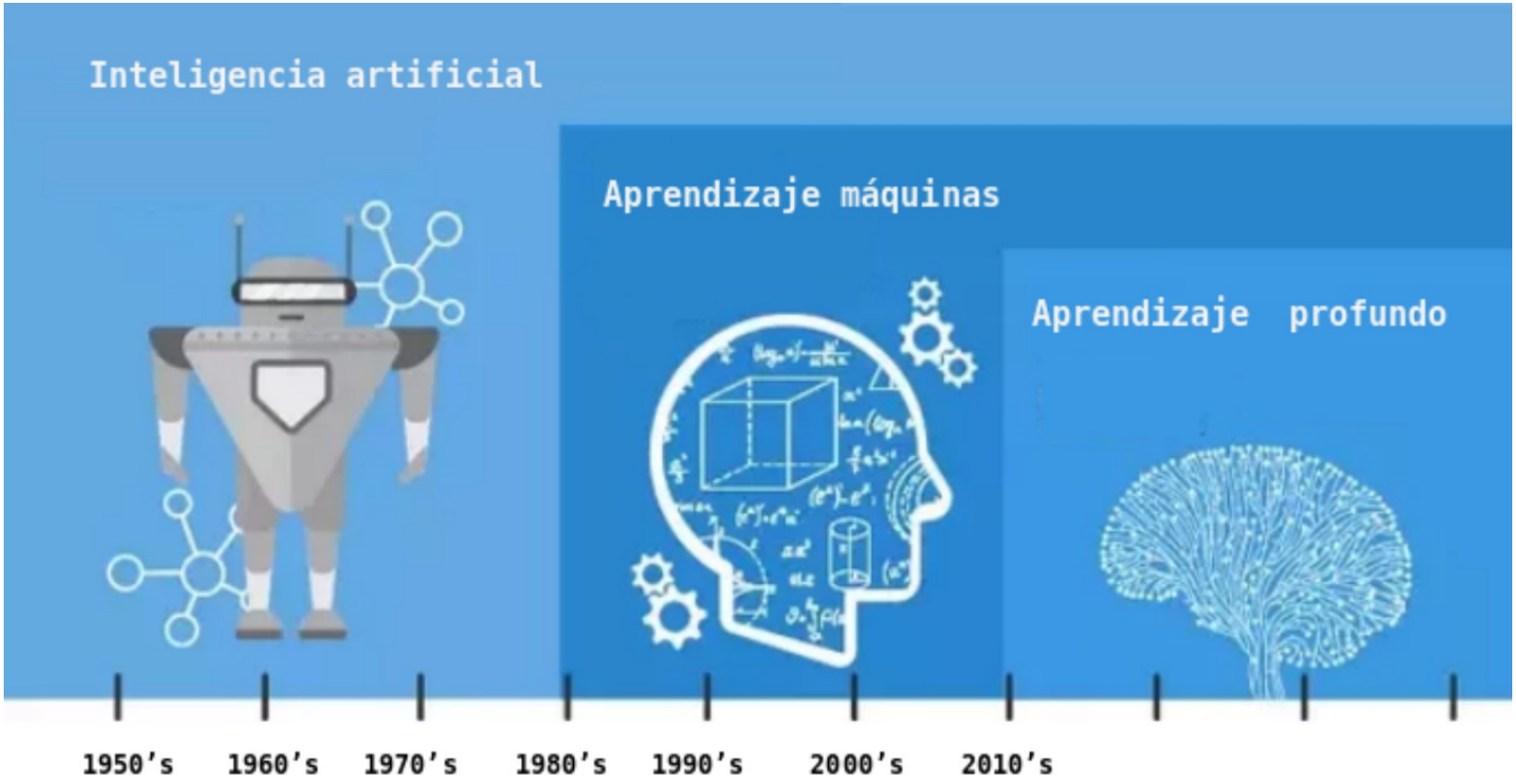
En este taller

- Aprendizaje de Máquinas
  - Introducción
  - Regresión lineal

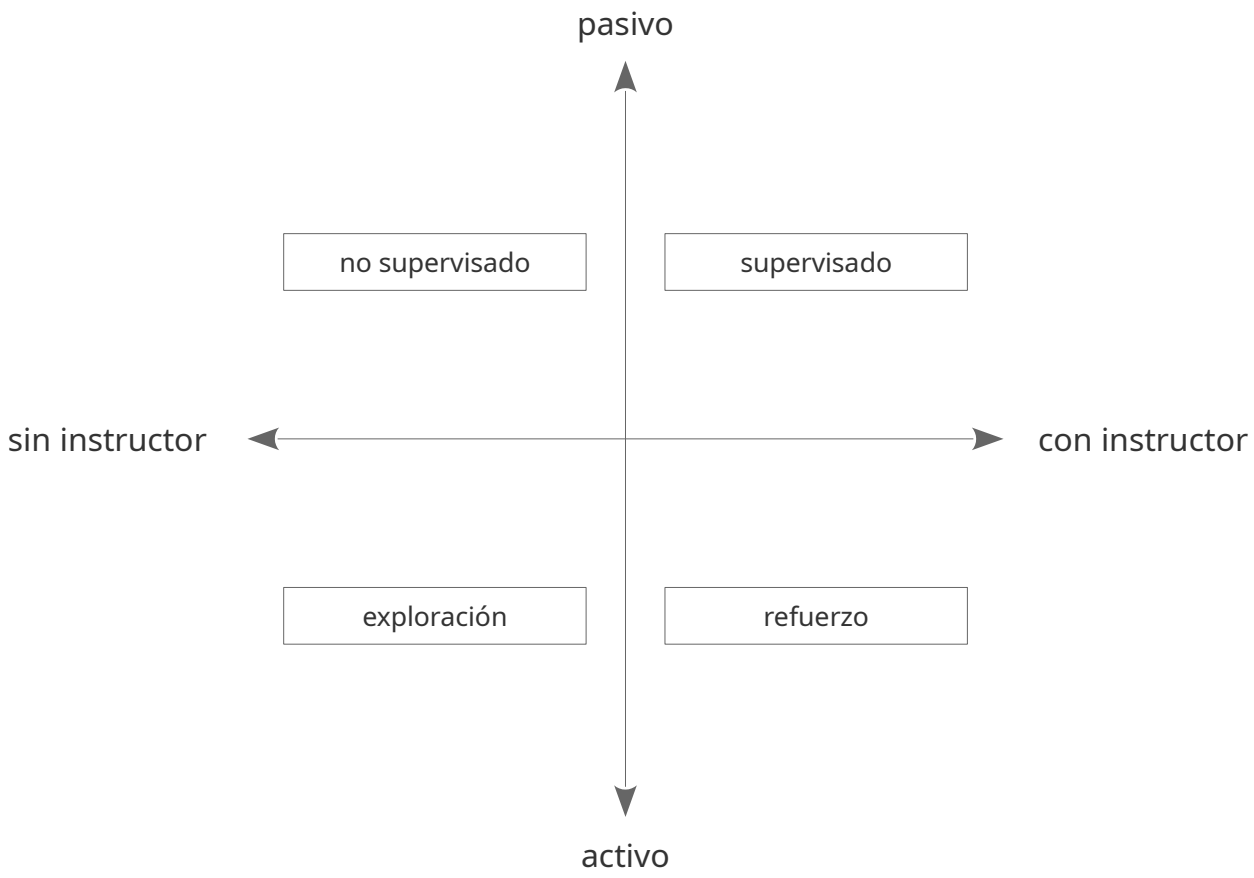
- Aprendizaje Profundo
  - Perceptrón multicapa
  - Redes convolucionales
  - Panorama

Aprendizaje de Máquinas

IA, AM y AP

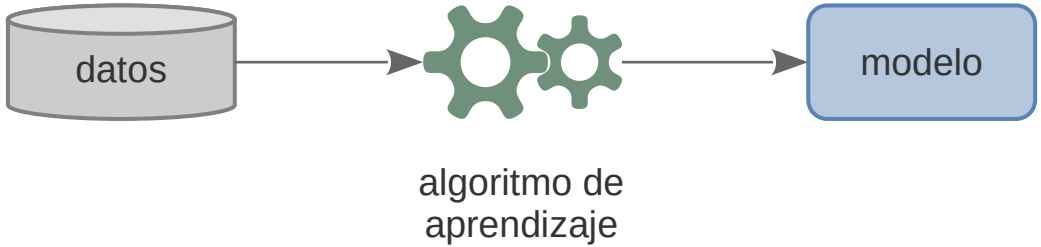

Inteligencia artificial
Aprendizaje máquinas
Aprendizaje profundo
1950's
1960's
1970's
1980's
1990's
2000's
2010's

Tipos de aprendizaje



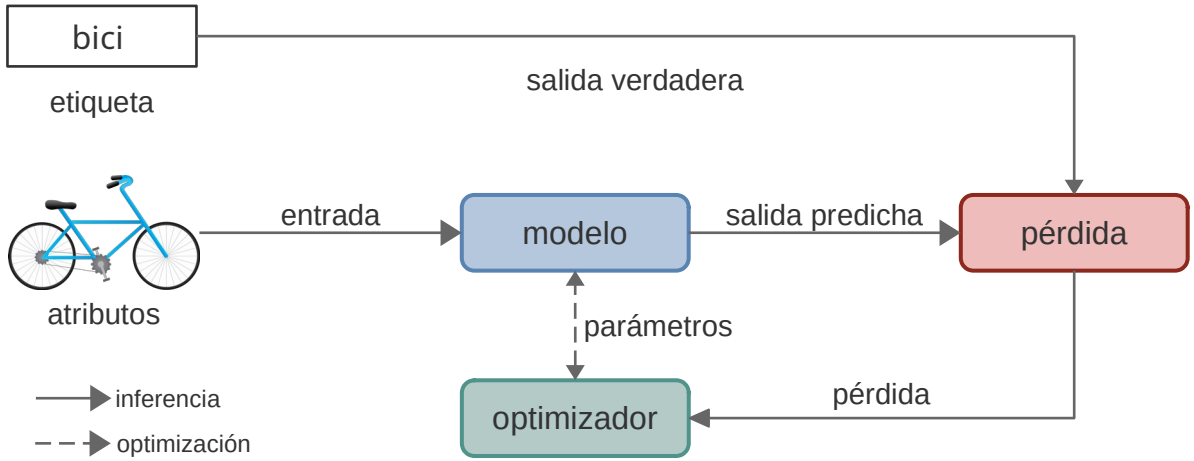
¿Cómo funciona el aprendizaje de máquinas?

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



Aprendizaje supervisado

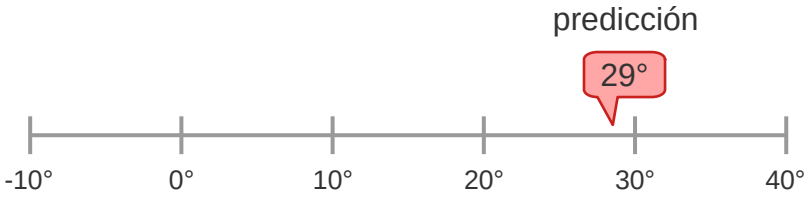
- La función de pérdida compara la salida verdadera con la salida predicha.
- Se minimiza la pérdida para actualizar los parámetros del modelo.
- El más común de los tipos de aprendizaje.

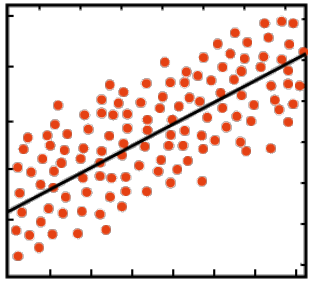


Regresión y Clasificación

Regresión

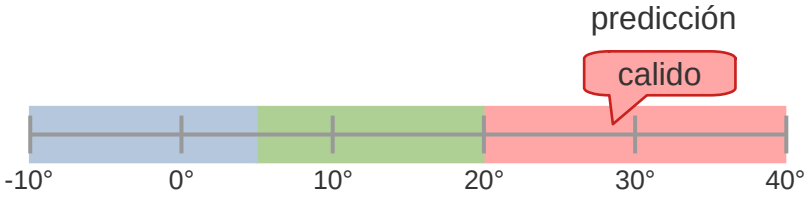
¿Cual será la temperatura mañana?

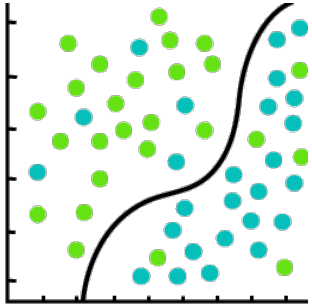




Clasificación

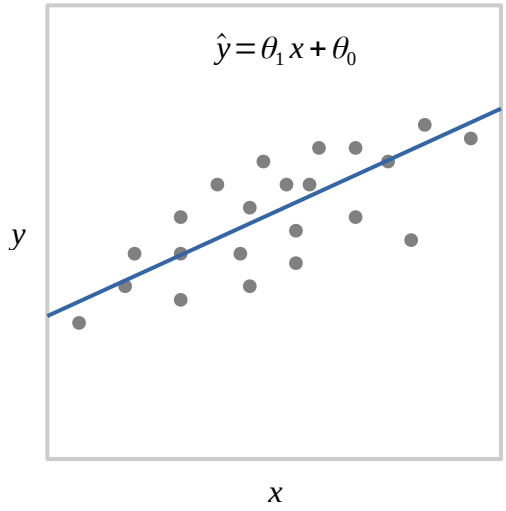
¿Cómo será el día mañana?



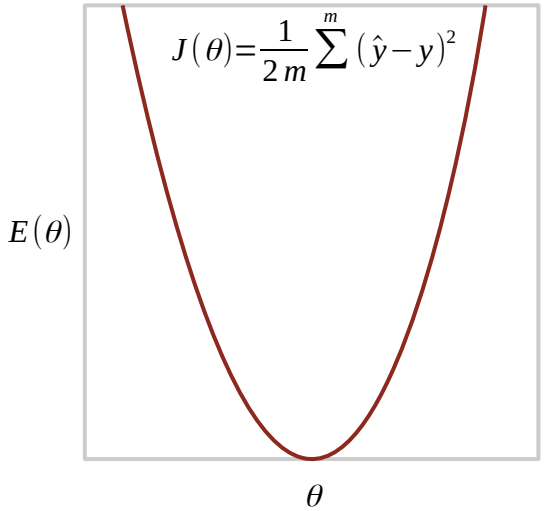


Regresión lineal

- Regresión: predecir una salida continua.



modelo



función de pérdida

`repetir hasta converger:`

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

`repetir hasta converger:`

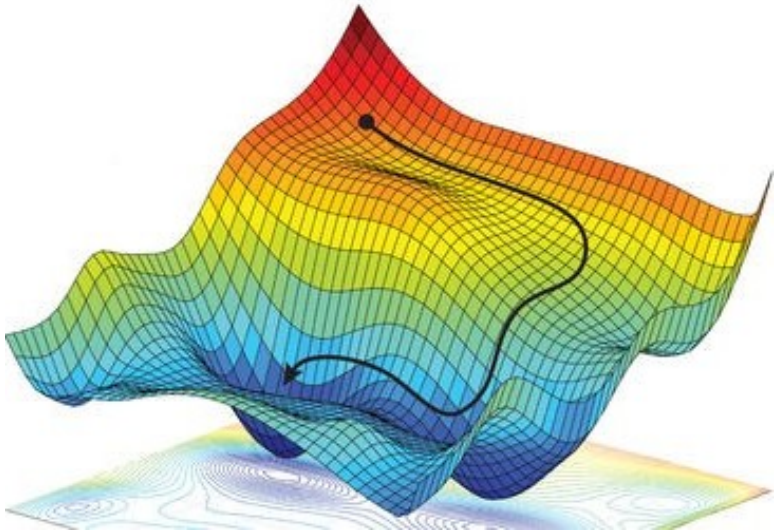
$$\theta_0 := \theta_0 - \frac{\alpha}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)})$$

$$\theta_1 := \theta_1 - \frac{\alpha}{m} \sum_{i=1}^{m} (\hat{y}^{(i)} - y^{(i)}) x^{(i)}$$

algoritmo de entrenamiento

Descenso por gradiente

- Avanzar hacia la dirección con menor pérdida.

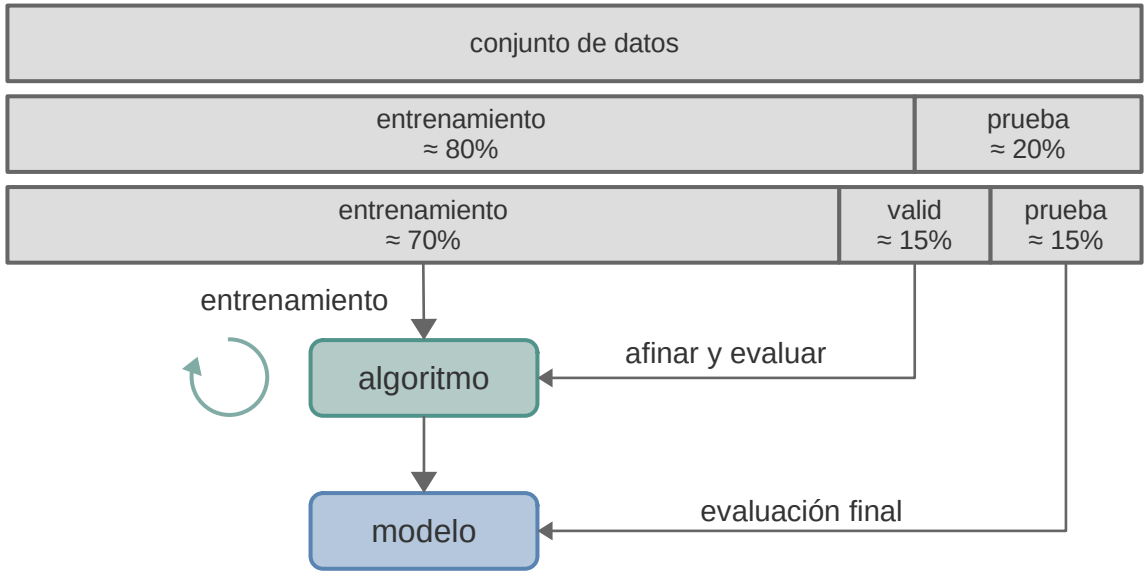


repetir hasta converger:

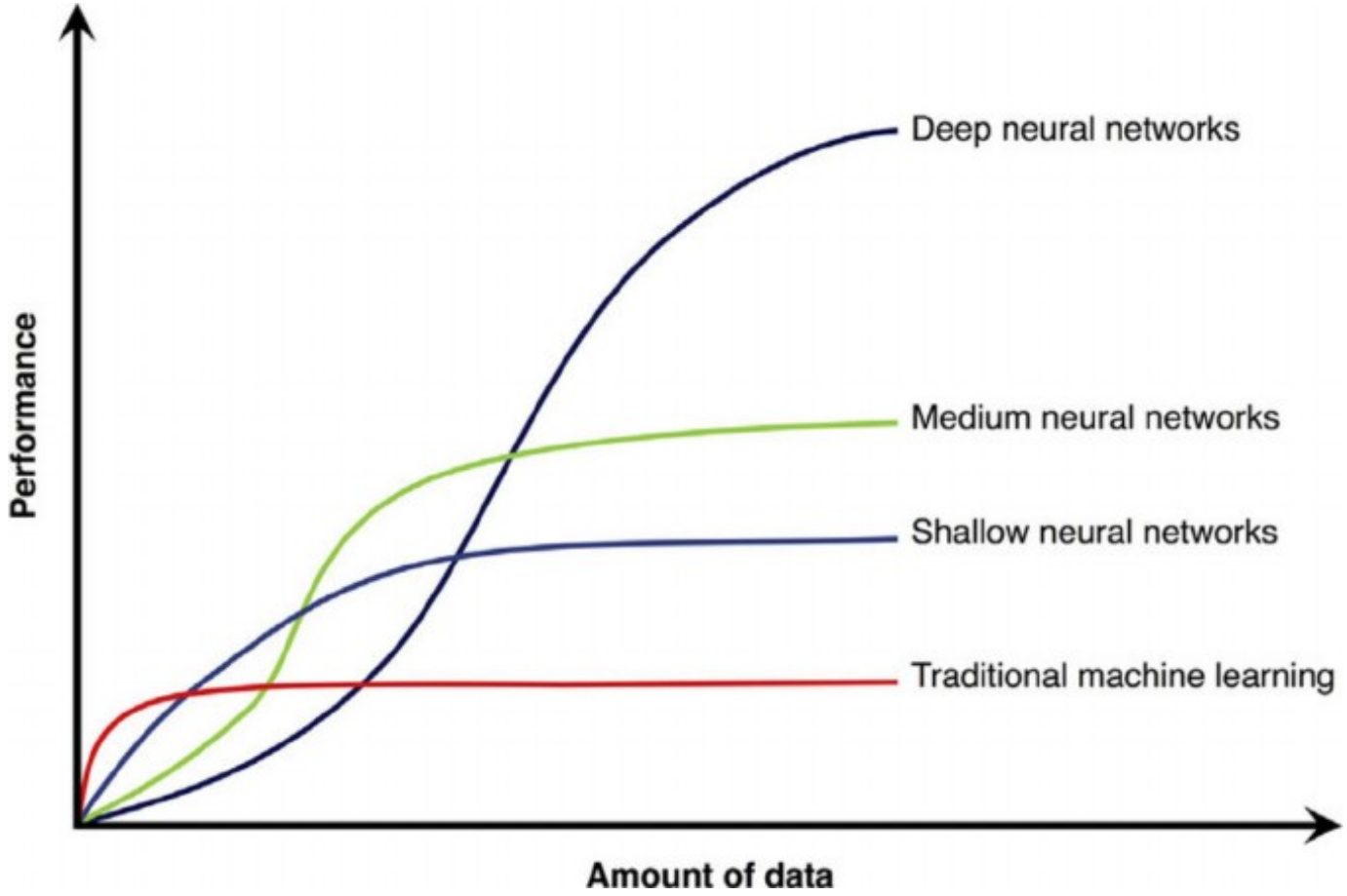
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Evaluación

- Aprendizaje iterativo en entrenamiento y validación.
- Evaluación final en prueba.



Limitaciones



Aprendizaje Profundo

Una receta con mucho exito

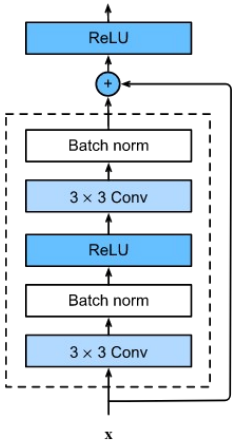
IMAGENET

14M de imágenes
21K categorías

YouTube-8M

8M de videos
4.8K categorías

+

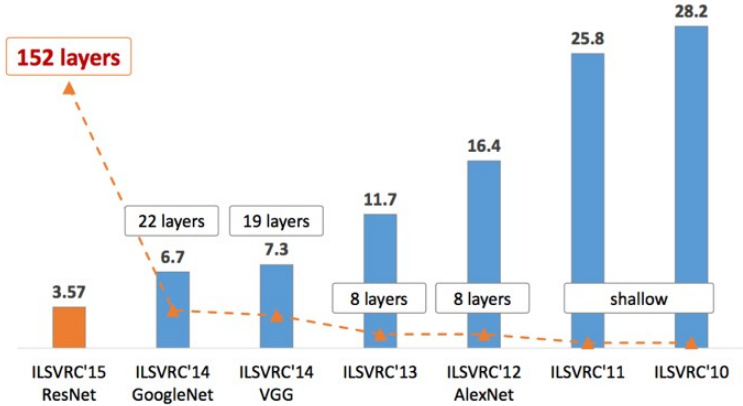


+

PyTorch

TensorFlow

=



datos masivos etiquetados

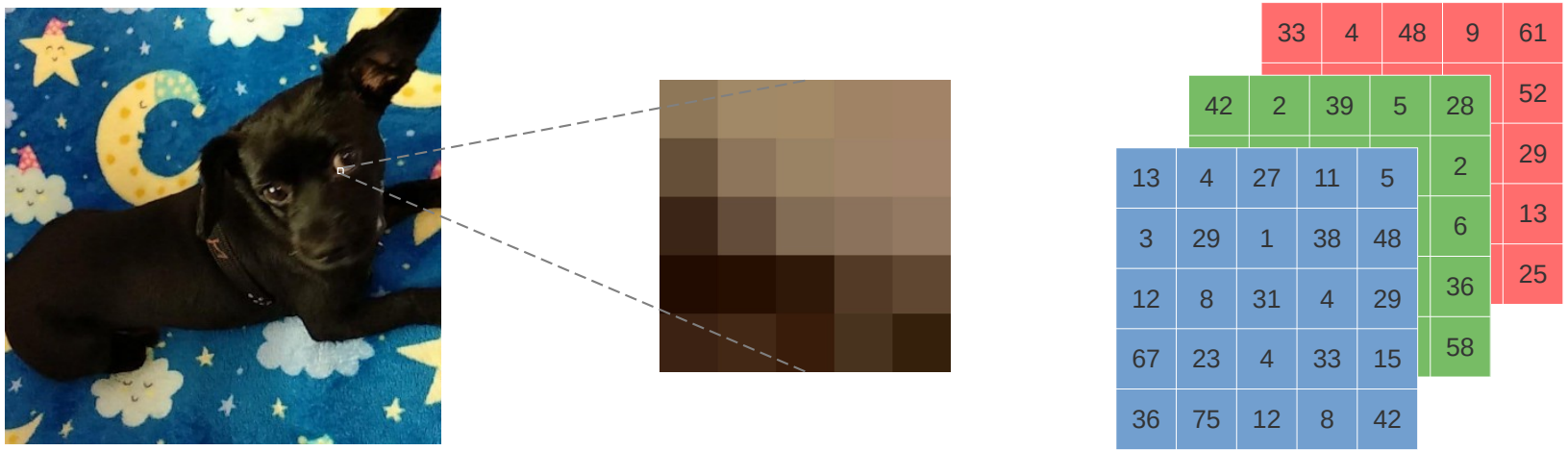
arquitecturas sofisticadas

infraestructura

estado del arte

Representación de imágenes

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

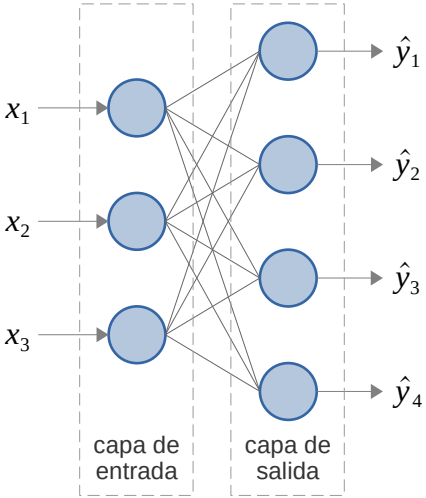


3x224x224 3x5x5 canales RGB

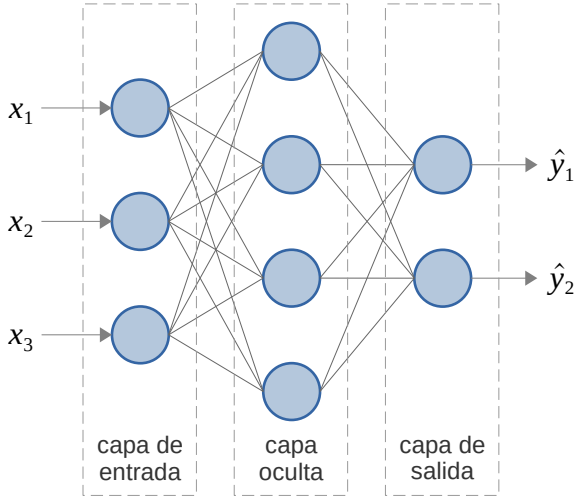
- MNIST 1x28x28, CIFAR-10 3x32x32, ImageNet 3x256x256.

Perceptrón multicapa

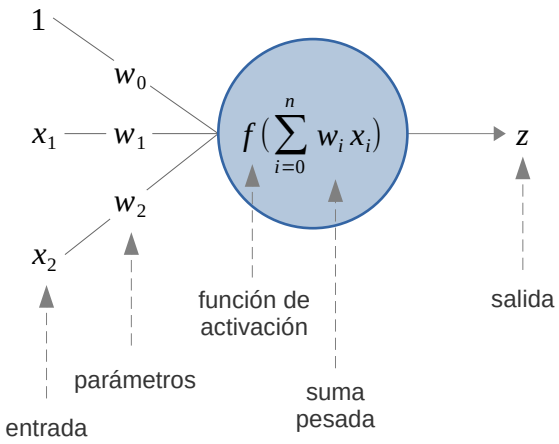
- Conjuntos de neuronas completamente conectadas.



red de 1 capa



red de 2 capas



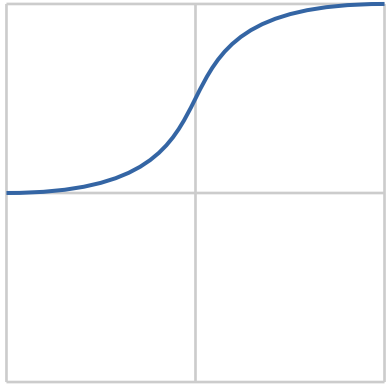
neurona

Funciones de activación (I)

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

Sigmoide

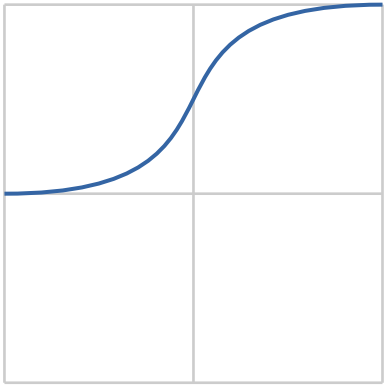
$$sigmoid(x) = \frac{1}{1+\exp(-x)}$$



Softmax

$$z = softmax(x)$$

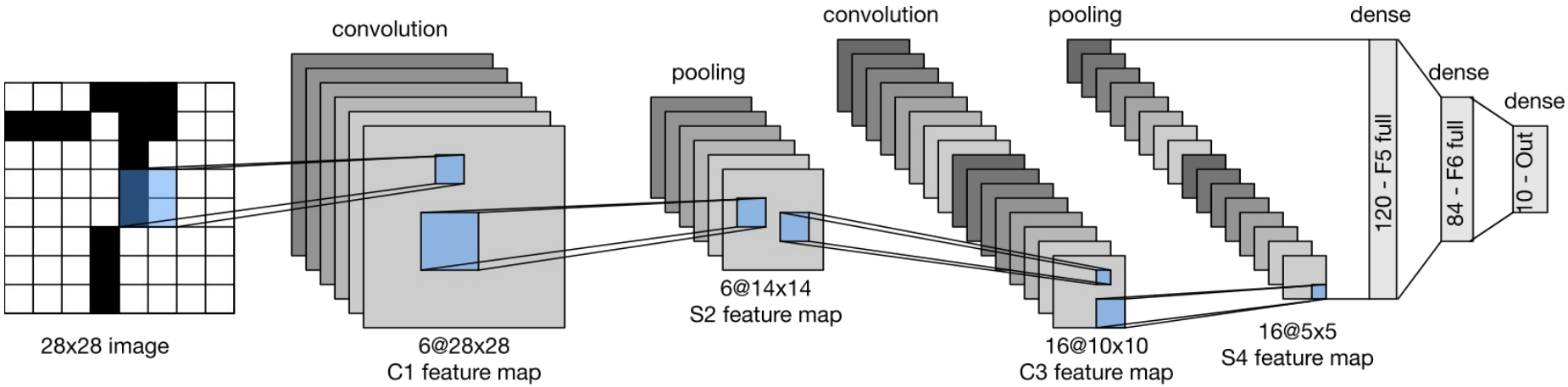
$$z_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)}$$



¡tiempo de programar!

`1a_mlp.ipynb`

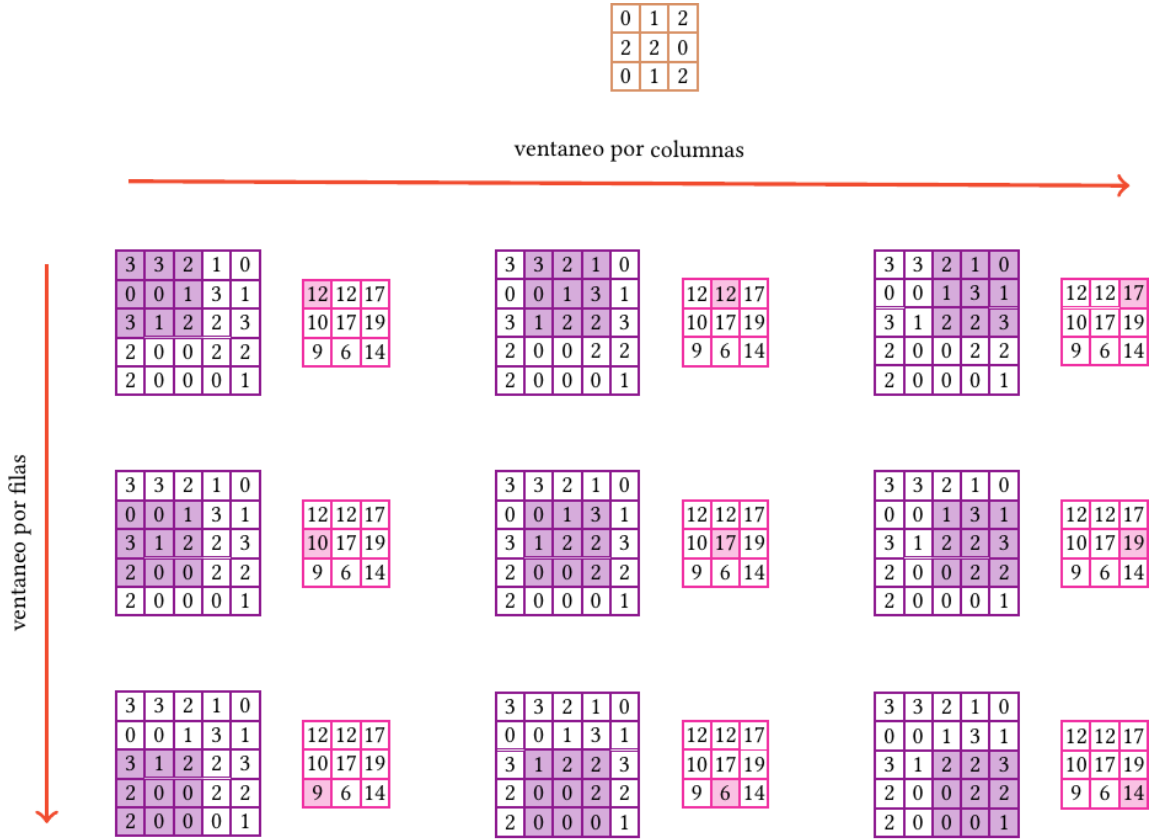
Redes Convolucionales



Arquitectura LeNet

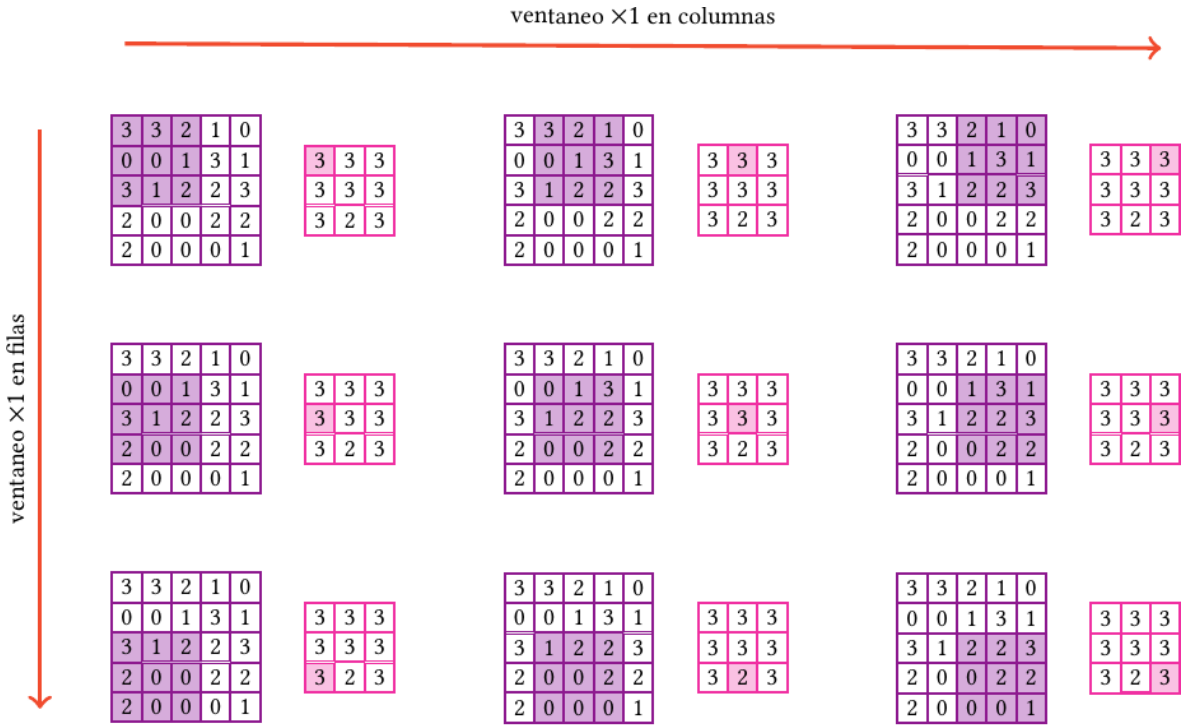
Zhang et al. Dive into Deep Learning. 2019.

Convolución



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

Muestreo



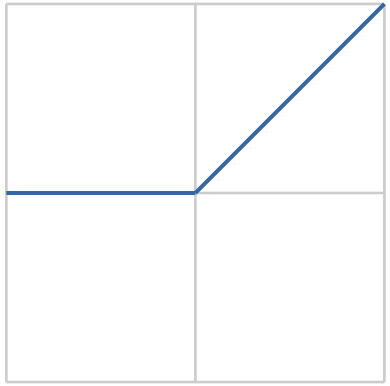
Muestreo máximo: entrada 5×5, salida 3×3, paso 1x1.

Funciones de activación (II)

- Función no lineal a la salida de la neurona.
  - ReLu: capas ocultas.

ReLu

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

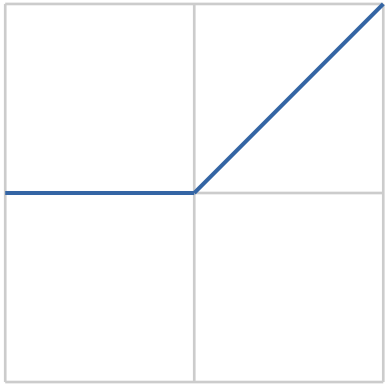


Funciones de activación (II)

- Función no lineal a la salida de la neurona.
  - ReLu: capas ocultas.
  - Softmax: clasificación multiclase.

ReLu

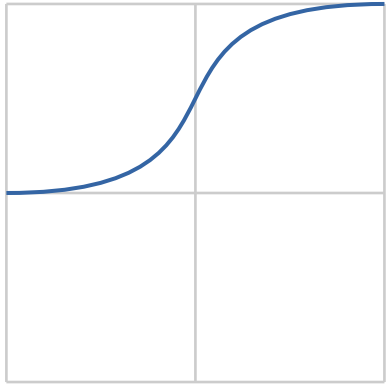
$$relu(x) = max(0, x)$$



Softmax

$$z = softmax(x)$$

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



¡tiempo de programar!

2a_cnn.ipynb

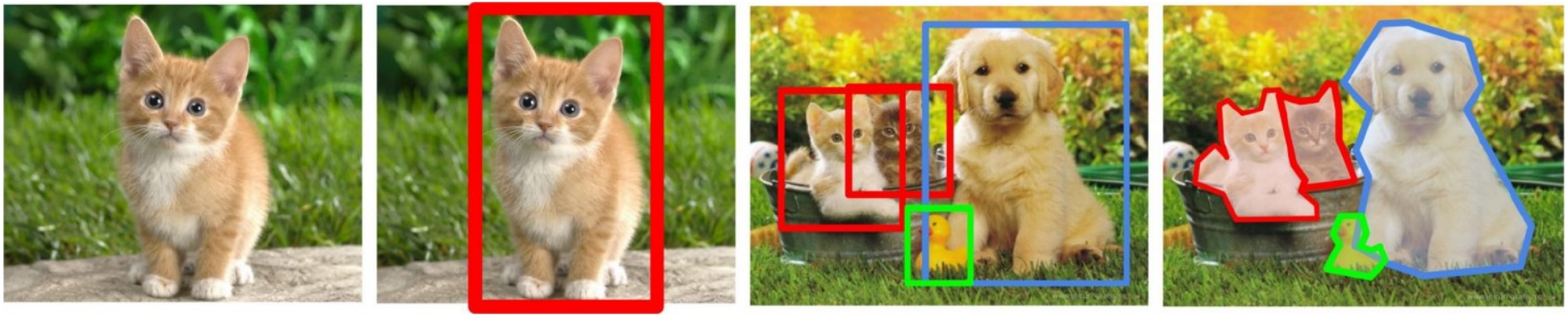
Tareas de Visión

clasificación

clasificación
+ localización

detección

segmentación



gato

gato

gato, pato, pato

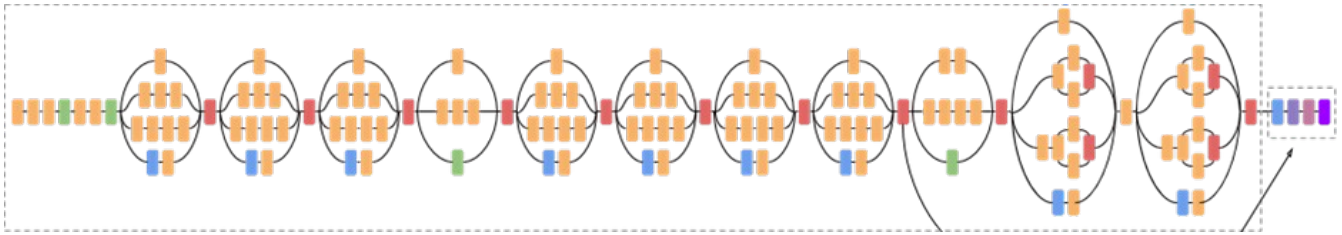
gato, pato, pato

un objeto

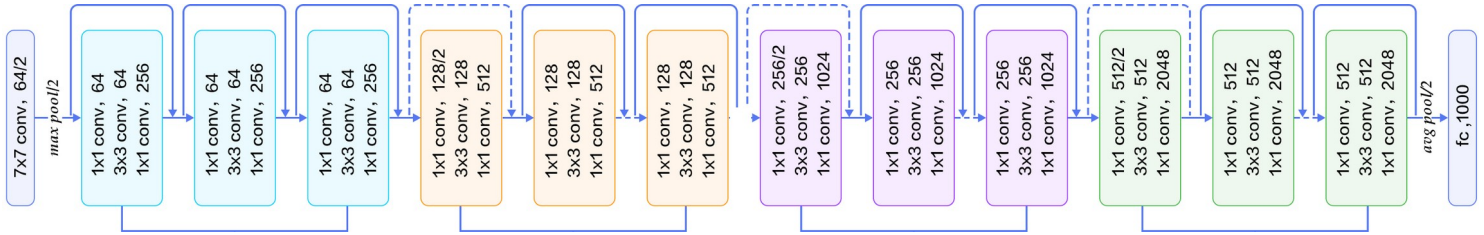
múltiples objetos

Arquitecturas

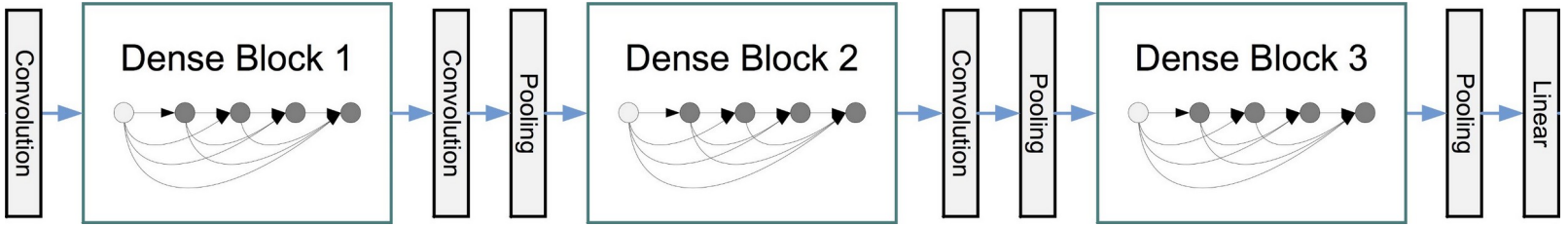
Inception



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