



MACHINE LEARNING

Fundamentos y aplicaciones

CURSO

Clases

- 25, 26 de octubre
- 1, 15, 16, 22 de noviembre
- Viernes de 18 a 21, Sábados de 9 a 13
- Clases teóricas, prácticas y un mix

Prácticas

- Python
- Google Colab (se necesita una cuenta de Google)
- Keras

Evaluación

- Proyecto a desarrollar en grupo
- Competencia
- Informe Final: **máximo 1 carilla**

OBJETIVOS

- Conocer y manejar los distintos parámetros que intervienen en el diseño de un sistema de machine learning
- Conocer y manejar distintos tipos de redes neuronales, según el tipo de aplicación
- Desarrollar 2 proyectos sencillos de reconocimiento de imágenes utilizando redes neuronales profundas
- Desarrollar un proyecto de complejidad media de machine learning

BIBLIOGRAFÍA Y REFERENCIAS

Libros

- Machine Learning; Tom Mitchel
- Deep Learning Book; Ian Goodfellow, Yoshua Bengio, Aaron Courville
- Deep Learning with Python; François Chollet
- The Hundred-Page Machine Learning Book; Andriy Burkov
- Y más...

Cursos Online

- Machine Learning, Andrew Ng (Stanford - Coursera)
- coursera.org, deeplearning.ai, udacity.com, udemy.com, edx.org
- medium.com
- Y muchos, muchos, muchos más...

AGENDA

- 1 Introducción
- 2 Machine Learning
- 3 Redes Neuronales
- 4 Deep Learning
- 5 Redes Recurrentes
- 6 Redes Adversariales y Generativas
- 7 Adversarial Examples

1

INTRODUCCIÓN

¿POR QUÉ MACHINE LEARNING?

Es relativamente simple escribir programas que reconozcan algunos patrones.

Pero muy difícil hacerlo cuando el entorno de lo que hay que reconocer cambia.

- P. ej: fotos de frente y de perfil, medio ocultos, con distintas iluminaciones.

Y muy difícil hacerlo cuando lo que hay que reconocer cambia con el tiempo.

- P. ej: envejecimiento, cortes de ruta, intentos de fraude.

También hay programas que son inherentemente complicados.

- P. ej: control de un helicóptero o un avión.

Aún si tuviésemos una idea de cómo hacer semejantes programas, serían muy complejos de hacer y/o muy difíciles de mantener.

IDEA CLAVE

¡No nos rompamos la cabeza escribiendo esos programas!

En cambio, juntemos **datos** de cómo debería ser la salida de esos programas, en función de diferentes datos de entrada.

- Y hagamos un algoritmo que aprenda a reproducirlos.
- Si está bien hecho, no sólo funciona con los datos que le pasamos sino también con nuevos datos.
- Si por algún motivo los datos de entrada cambian, se puede modificar nuevamente el programa para que se adapte a estos cambios.

¿POR QUÉ MACHINE LEARNING AHORA?

La idea no es realmente novedosa.

Extremadamente popular hoy en día gracias a avances en tres ejes:

- Datos: hoy se generan más datos por día que nunca antes en la historia de la humanidad.
- Hardware: hoy hay computadoras capaces de procesar grandes volúmenes de datos en tiempos razonables.
 - Y es más barato que pagarle a un programador 😊
- Algoritmos: hoy existen algoritmos que se probó que hacen ciertas tareas mejor que los humanos.

EJEMPLOS

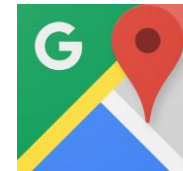
NLP

amazon alexa



**Recommendation
Engines**
(Retail, Traffic)

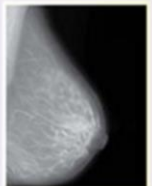
amazon



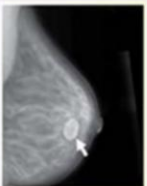
Filtering
(Spam, Malware)

Autonomous Driving

Computer Vision



Normal
mammogram



Benign cyst
(not cancer)



Cancer

**Medical
Image
Analysis**



**Face
recognition**



AI Helps Researchers Design New Medications

A team from the University of Cambridge has developed a model for predicting the outcome of complex chemical reactions that researchers call a 'GPS for chemists.'

by: Elizabeth Montalbano in Materials & Assembly, Medical on October 16, 2019

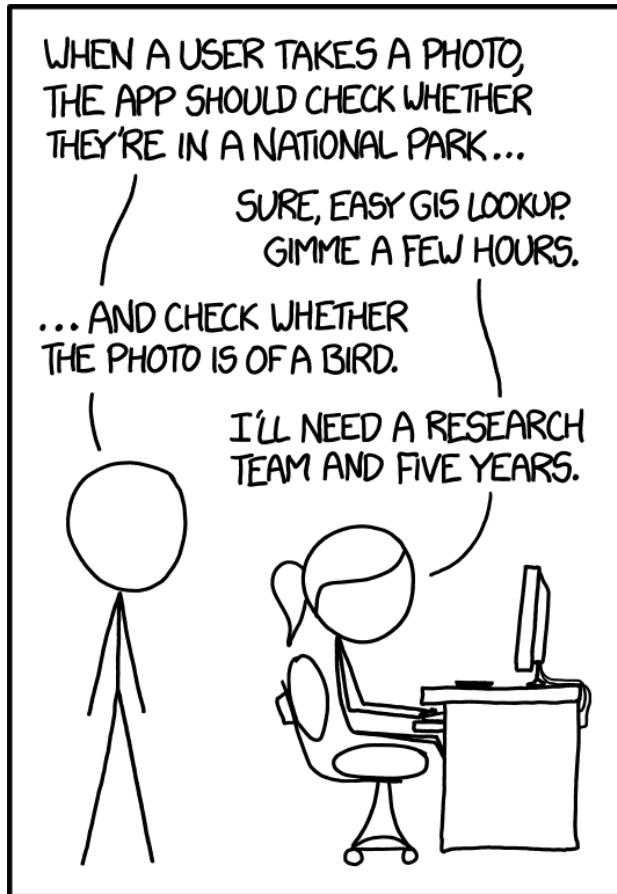
This MIT AI Predicts Breast Cancer Risk Up to 5 Years in Advance

MIT CSAIL scientists partnered with Massachusetts General Hospital to develop a deep-learning model that was trained on 90,000 full-resolution mammogram scans from 60,000 patients who were scanned over the course of several years with various outcomes.



By Ben Dickson May 23, 2019 8:44AM EST

¿CUÁLES TAREAS SON IDEALES PARA ML?



IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

27/10/2019

Respuesta genérica

Aquellas en las que usualmente somos buenos los seres humanos.

Problemas cuyas reglas no estén muy bien definidas y/o son muy complejas para algoritmos tradicionales.

¿CUÁLES TAREAS SON IDEALES PARA ML?

Reconocimiento de patrones

- Objetos en imágenes, rostros, textos, audios.

Detección de anomalías

- Secuencia inusual de transacciones con una tarjeta de crédito.
- Secuencia inusual de lectura de sensores en un sistema de control.

Motores predictivos o de recomendación

- Recomendación de películas, precios (no en Argentina), recomendar una ruta.

Control

- Manejo del cuerpo humano, de helicópteros, autos autónomos, MotoGP, carreras de drones, etc.

¿Qué

es

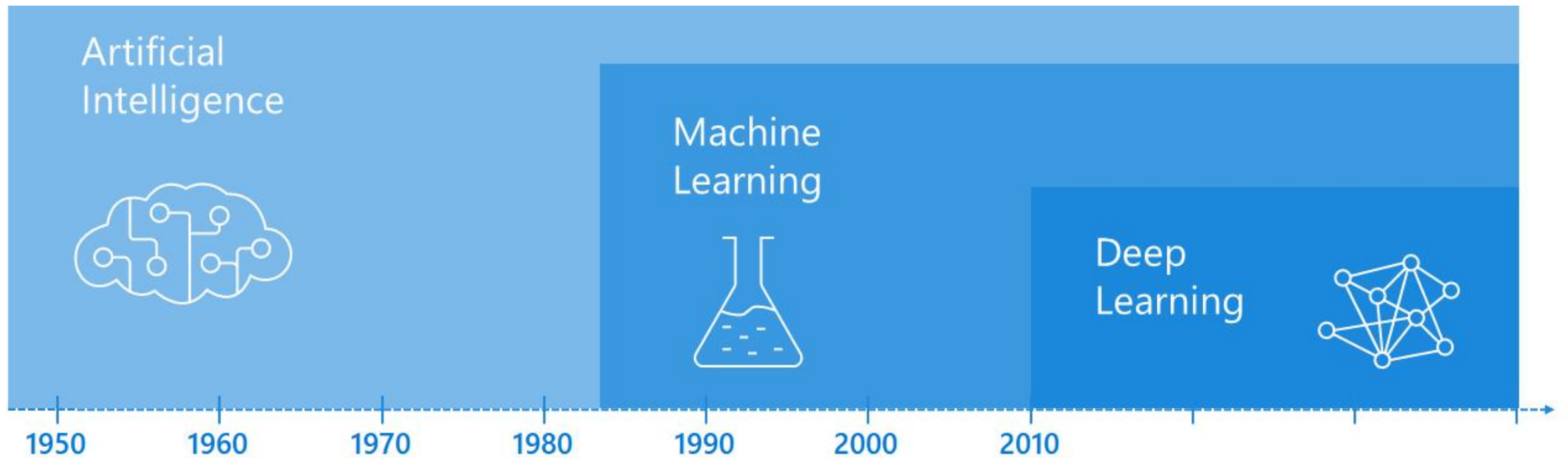
Machine

Learning?

2

MACHINE LEARNING

MACHINE LEARNING VS AI



“

A computer program is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P** , if its performance at tasks T , as measured by P , improves with experience E .

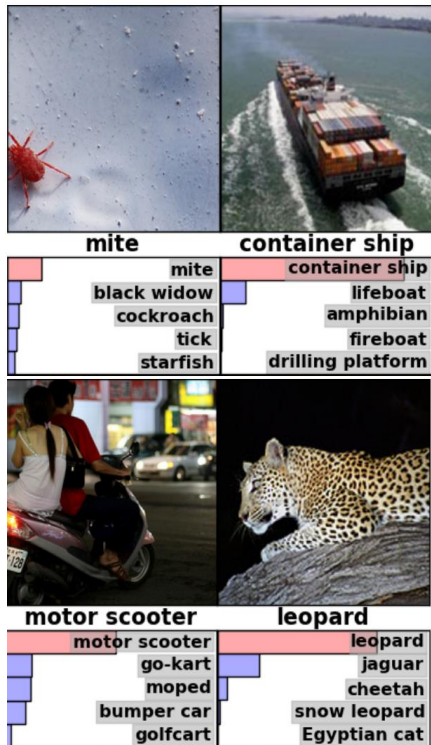
Tom Mitchell

Machine Learning, 1997

MACHINE LEARNING

Task, T

- ¿Qué va a hacer el modelo?



Experience, E

- ¿De dónde va a aprender el modelo?

airplane

automobile

bird

cat

deer

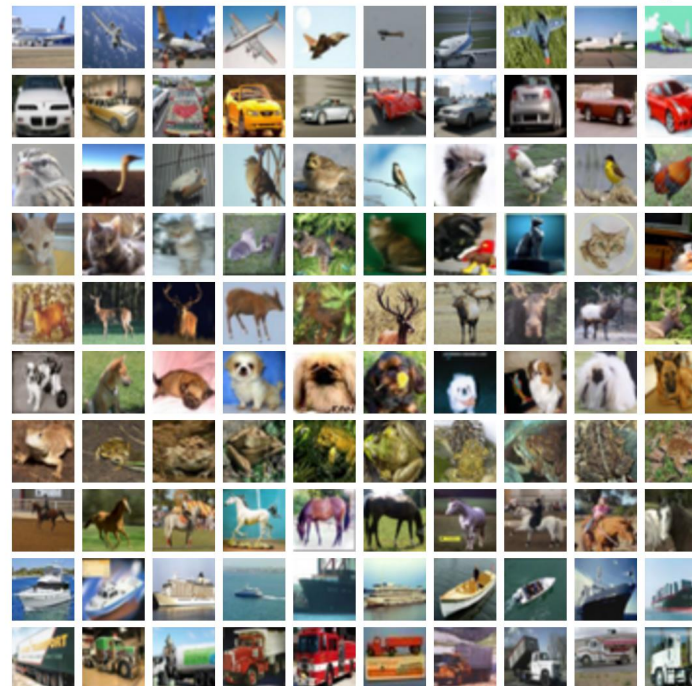
dog

frog

horse

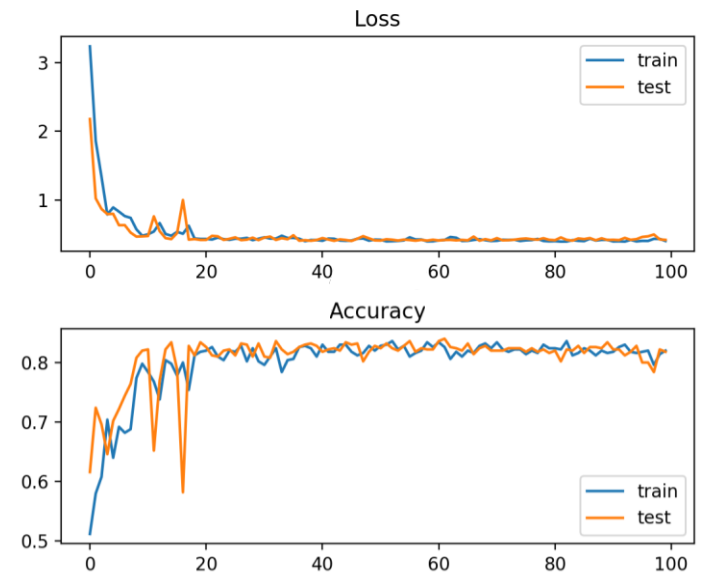
ship

truck

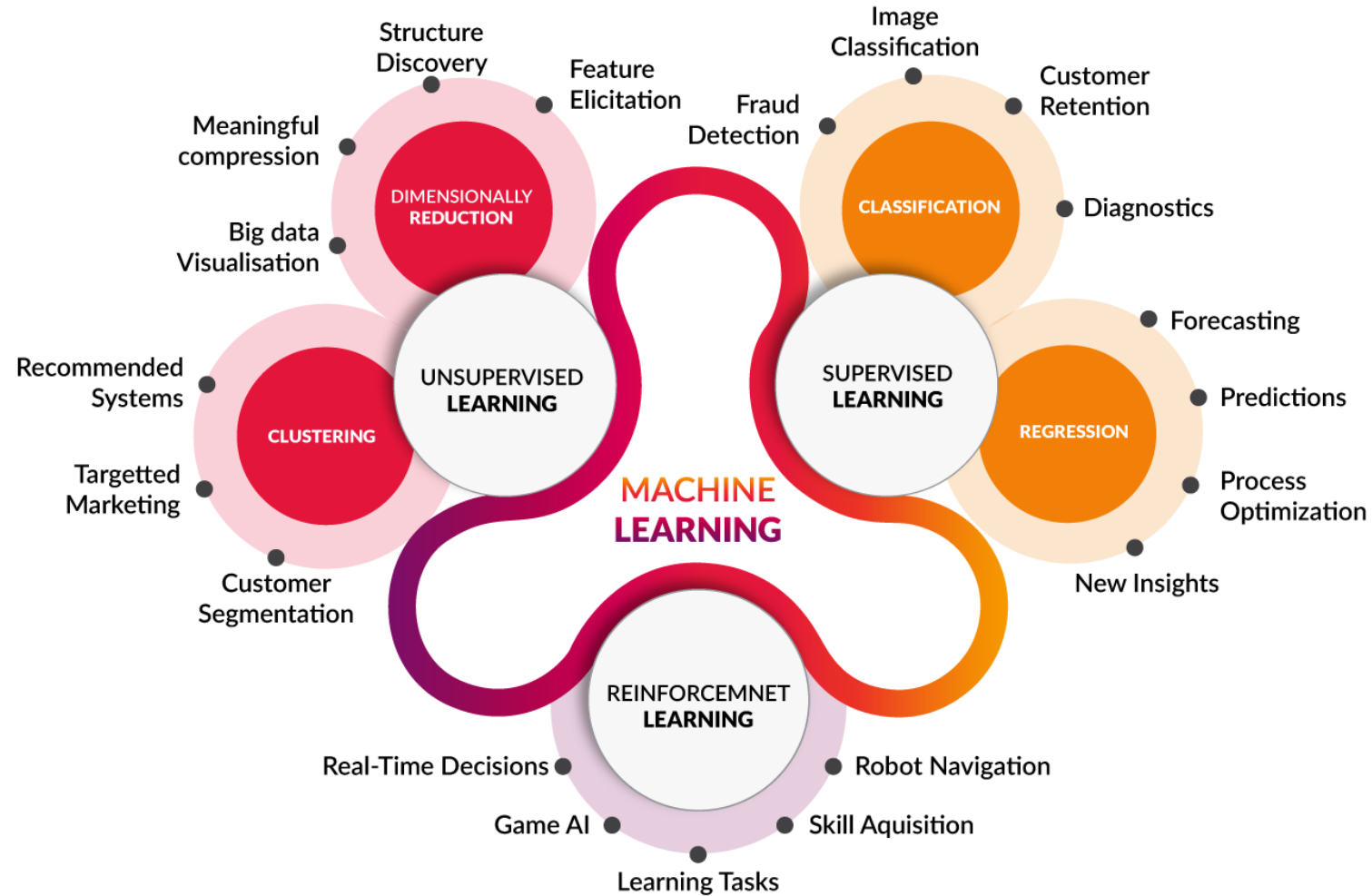


Performance measure, P

- ¿Cómo medir si el modelo aprende?



¿QUÉ VA A HACER EL MODELO?



TIPOS DE MACHINE LEARNING

Supervisado

Aprender a predecir una salida dado un vector de entrada.

No Supervisado

Descubrir una buena representación interna de los datos de entrada.

Reforzado

Aprender a tomar decisiones que maximicen una recompensa.

¿DE DÓNDE VA A APRENDER EL MODELO?

Datasets: Son los conjuntos de datos sobre los que trabajaremos.

Para aprendizaje supervisado, es imprescindible que los datos estén etiquetados.

- Es una tarea larga y delicada.

Ejemplos más comunes para procesamiento de imágenes:

- MNIST: dígitos manuscritos.
- CIFAR: imágenes en 10 y en 100 categorías.
- PASCAL VOC: imágenes en 20 categorías.
- ImageNet: 1.3 millones de imágenes en 1000 categorías.

¿Cómo buscar datasets?

- <https://toolbox.google.com/datasetsearch>

MNIST DATASET

Uno de los primeros datasets, muy simple y conocido.

Gratuito:

<http://yann.lecun.com/exdb/mnist/>

70.000 imágenes de dígitos manuscritos

Blanco y Negro, de 28x28 pixeles.

Ejemplo: <https://tensorflow-mnist.herokuapp.com/>



By Josef Steppan - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=64810040>

CIFAR 10 DATASET

Dataset de imágenes gratuito provisto por la Universidad de Toronto, utilizado para trabajar con imágenes pequeñas.

<https://www.cs.toronto.edu/~kriz/cifar.html>

60.000 imágenes a color divididas en 10 categorías, de tamaño 32x32 píxeles.

airplane

automobile

bird

cat

deer

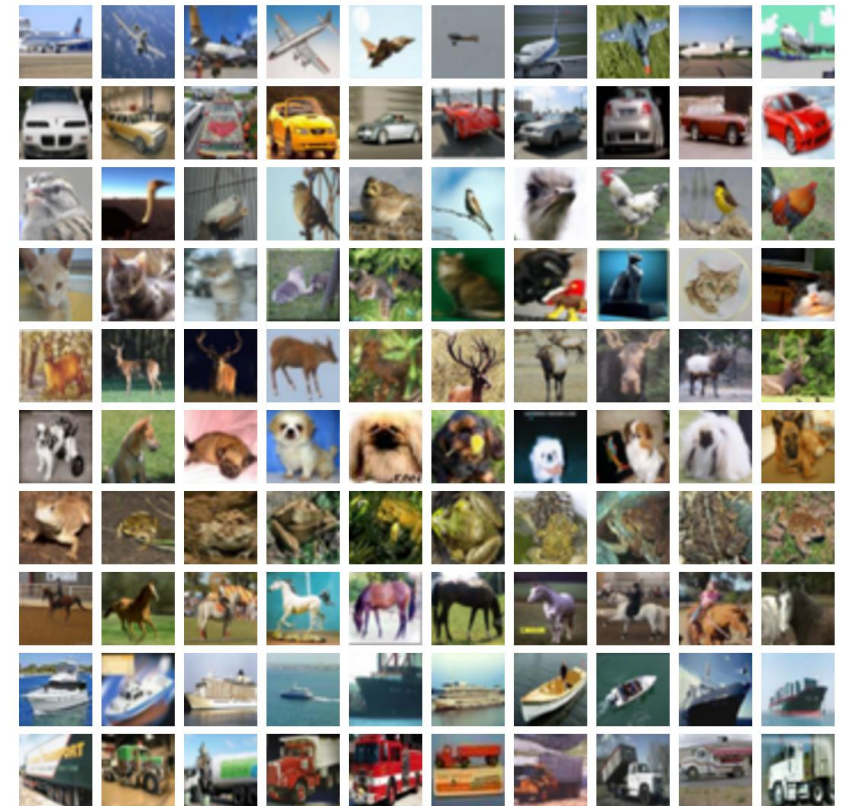
dog

frog

horse

ship

truck



IMAGENET DATASET

El más grande, importante y popular.

<http://image-net.org/about-overview>

Promovió una competencia entre los años 2010 y 2017.

- La competencia ya no existe porque los resultados son demasiado buenos.

Los mejores algoritmos de reconocimiento de imágenes se miden con este dataset.

- AlexNet (2012).
- VGG (2013).
- Inception v3, también conocida como GoogLeNet (2015).
- ResNet-50 (2015).

¿CÓMO MEDIR SI EL MODELO APRENDE?

1. División de data set en training set y test set
2. Medición del error en base a la tarea a realizar
3. Métricas de medición del error
 - También dependen de la tarea a realizar:
 - Clasificación: 0-1
 - Regresión: Valores continuos

DIVISIÓN DE UN DATASET

Training Set

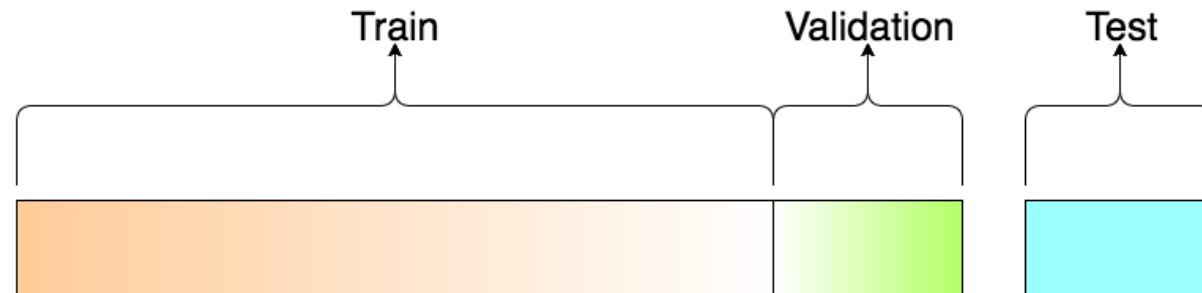
Conjunto de datos a usar durante el entrenamiento.

Validation set

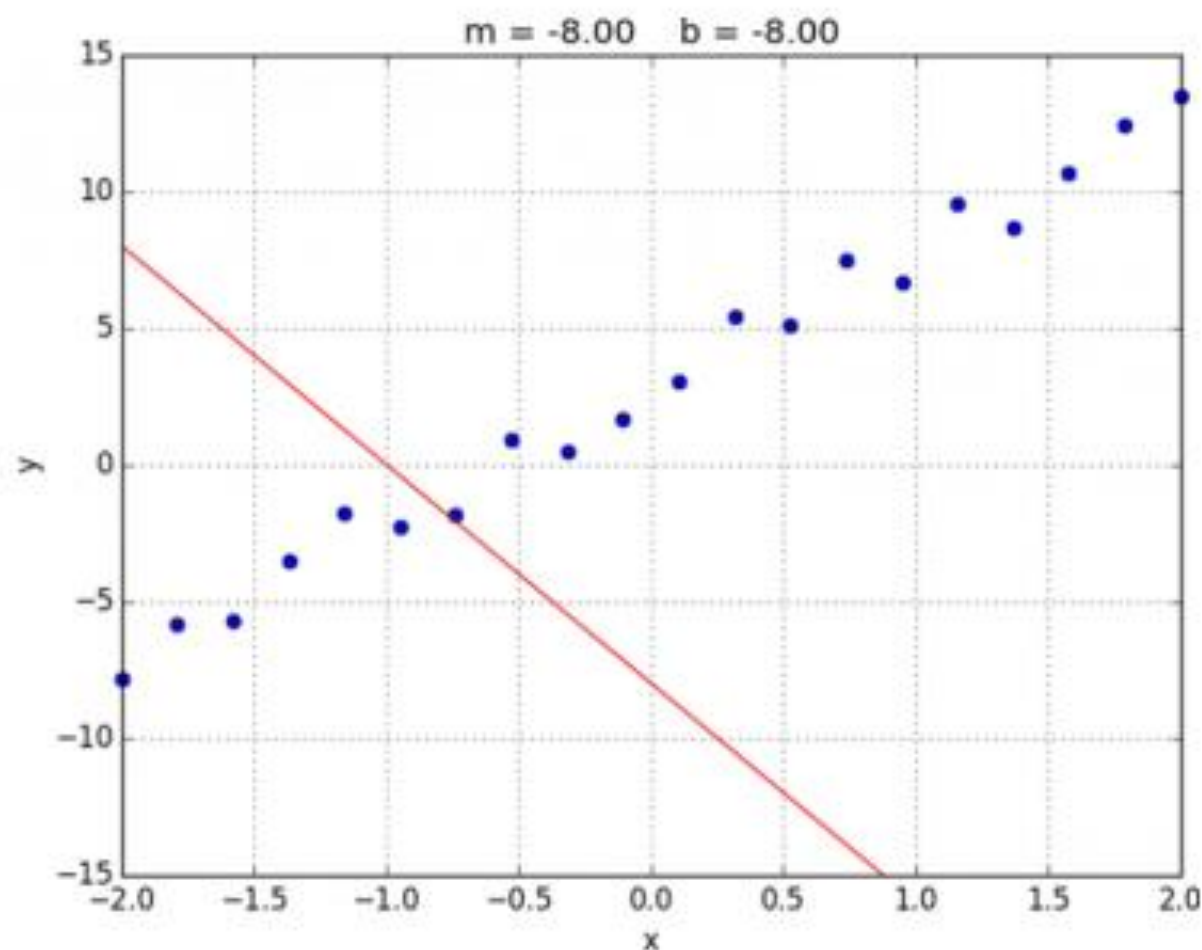
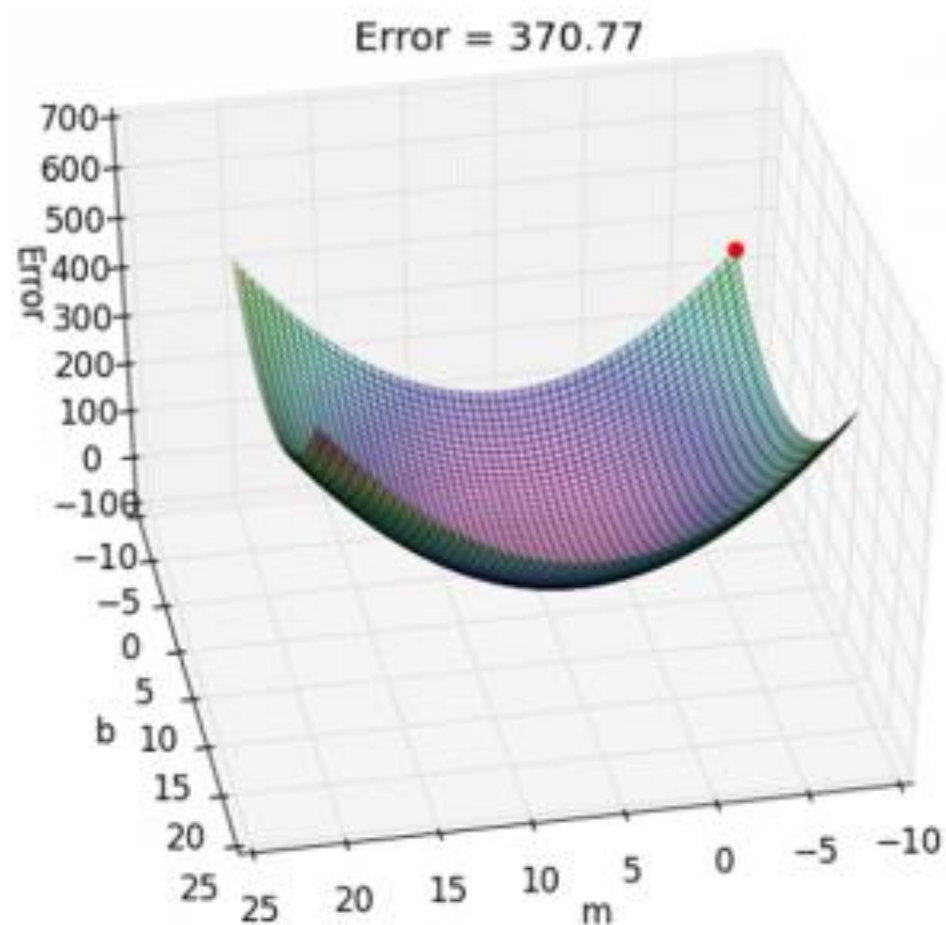
Conjunto de datos para mejorar ciertos hiperparámetros.

Test set

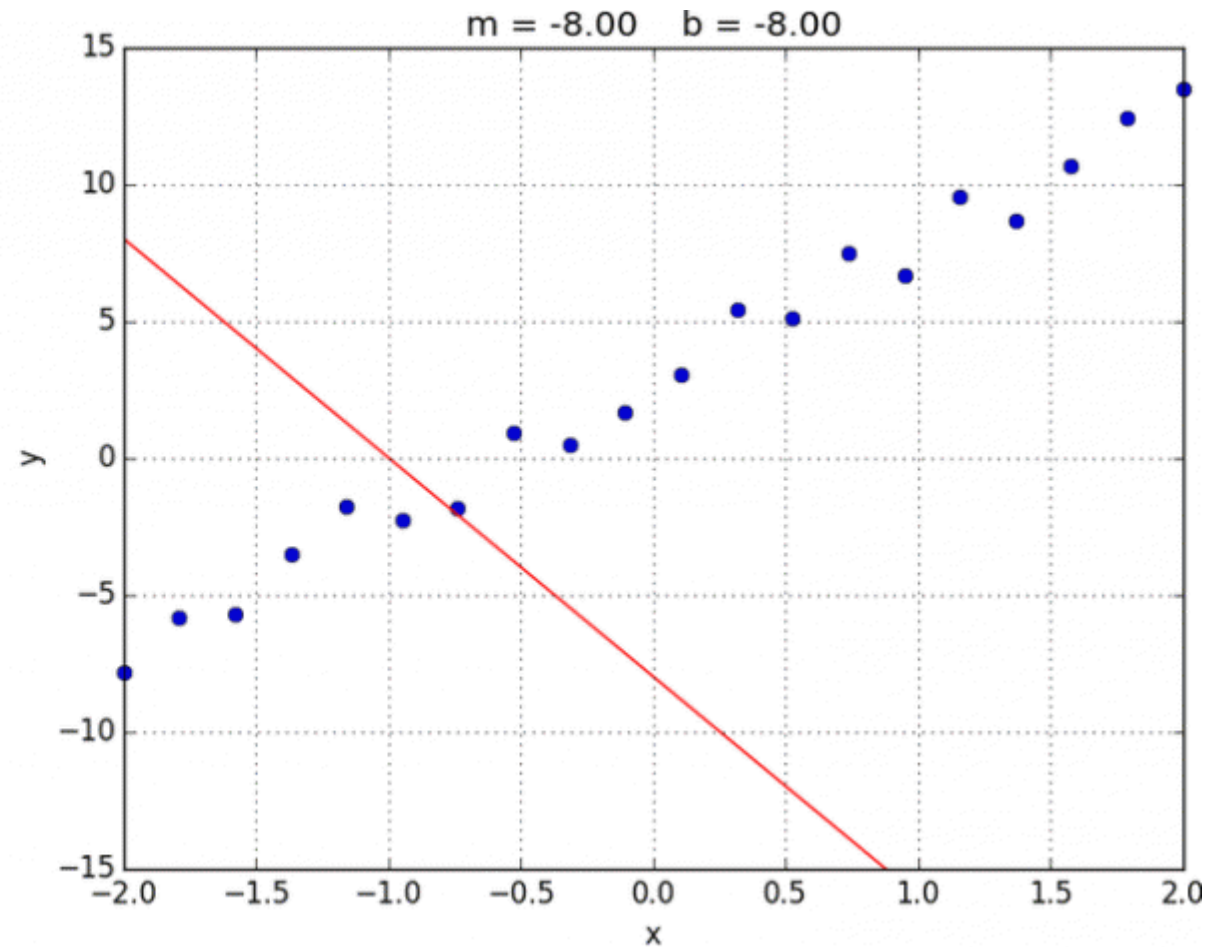
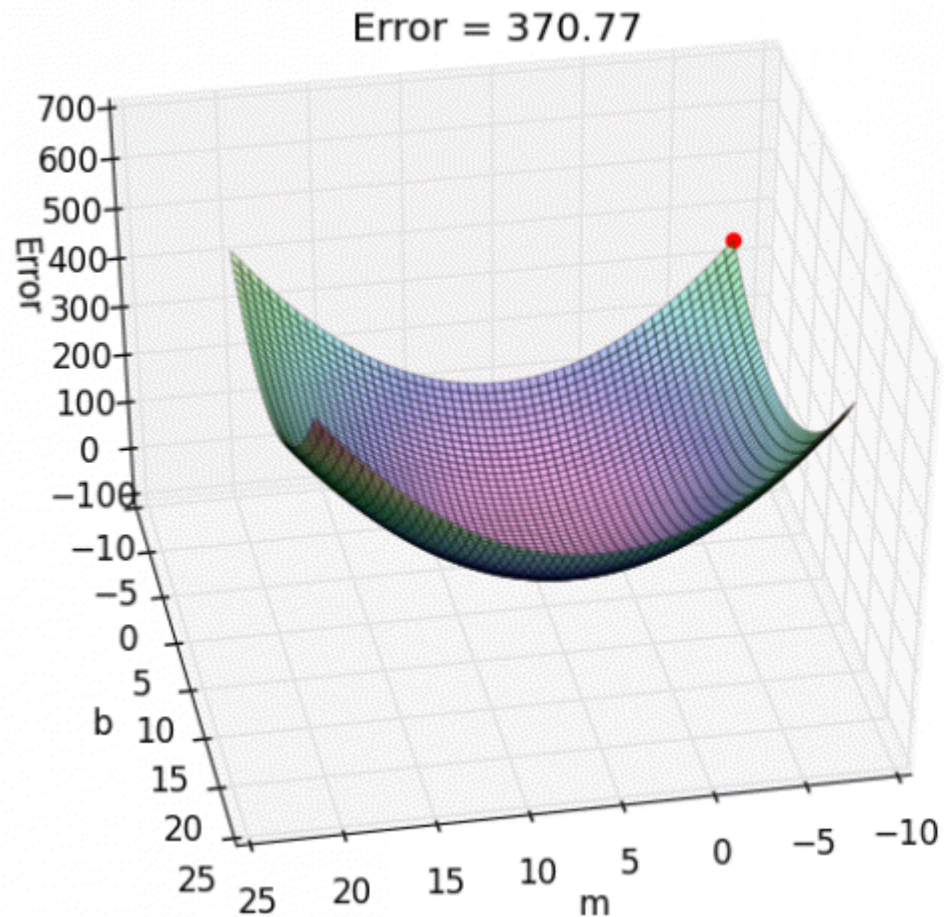
Conjunto de datos para proveer una evaluación de la performance de un modelo.



¿CÓMO MEDIR SI EL MODELO APRENDE?



¿CÓMO MEDIR SI EL MODELO APRENDE?



OBSERVACIONES SOBRE DIVISIÓN DE DATASETS

La idea de fondo es no sobreentrenar una red, sino que la red se adapte a nuevos casos, esto se conoce como **generalización**.

- Esto es lo que los humanos hacemos muy bien!

La separación no debe ser meramente aleatoria, sino representativa del conjunto de datos.

Suele utilizarse una validación cruzada (**cross validation**):

- Separar el conjunto de datos en partes iguales, e ir utilizando cada una como validación en distintas iteraciones.

UNDERFITTING Y OVERFITTING

Underfitting

El modelo es demasiado simple para el conjunto de datos a representar.

Overfitting

El modelo tiene problemas para generalizar.

Características

- Alto error de entrenamiento
- Error de entrenamiento cercano al error de test

- Error de entrenamiento demasiado bajo
- Error de entrenamiento mucho menor al error de test

Soluciones

- Crear un modelo más complejo
- Entrenar por más tiempo

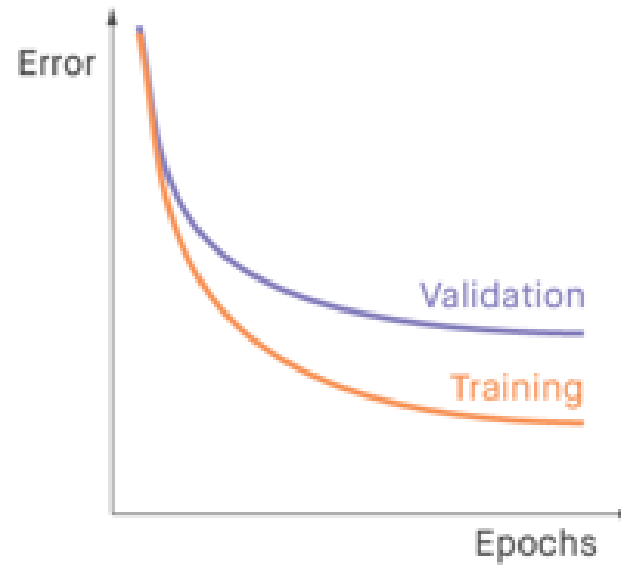
- Realizar una regularización de los datos
- Obtener más datos

UNDERFITTING Y OVERFITTING

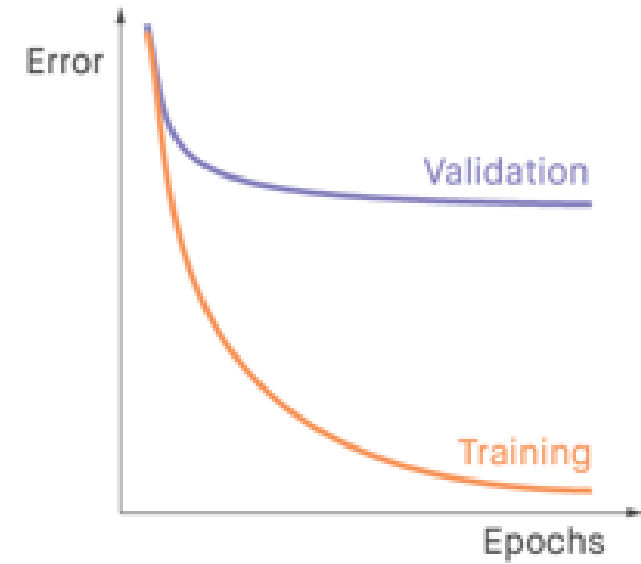
Underfitting



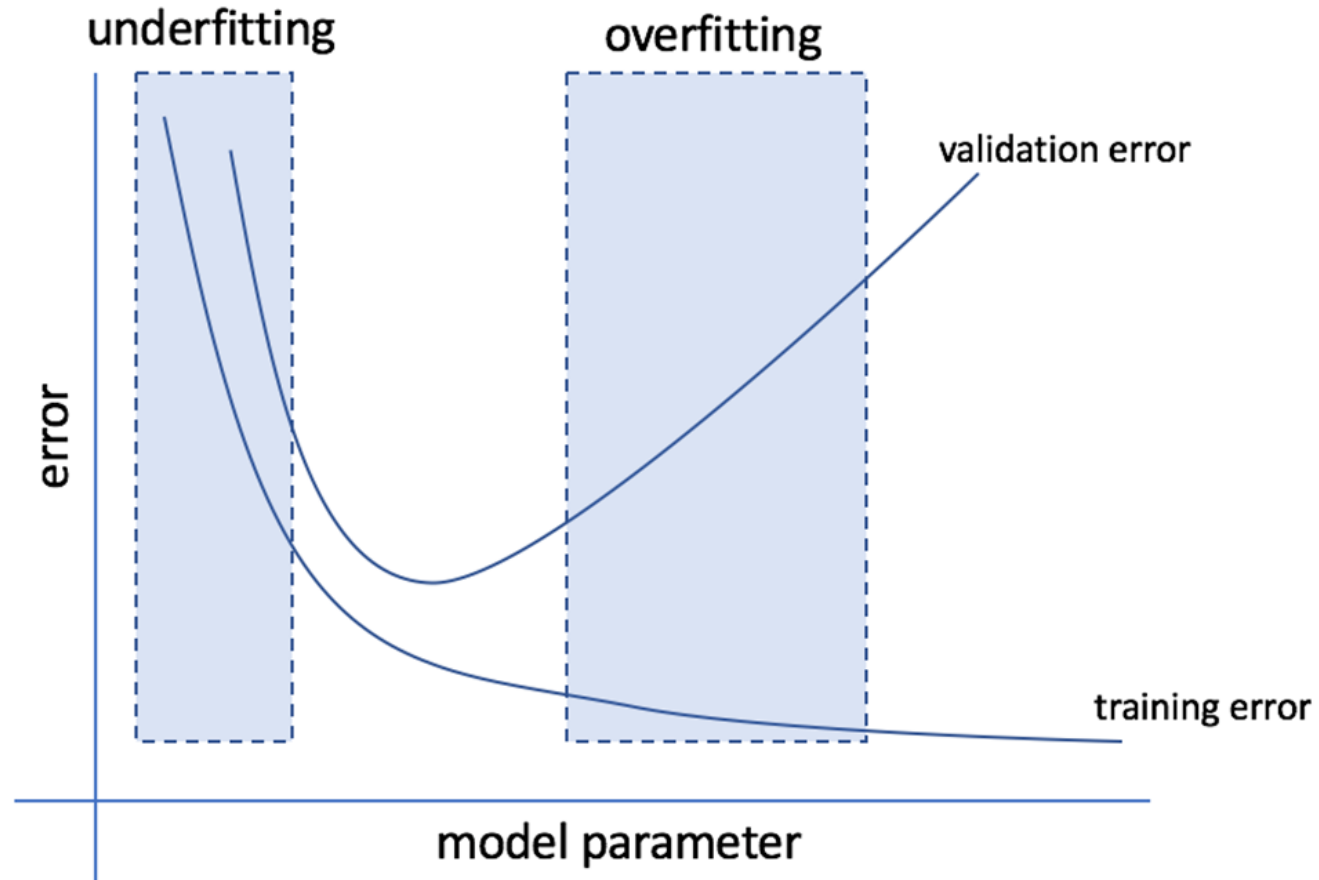
Correcto



Overfitting



OVERFITTING Y PARÁMETROS DE LA RED



EJEMPLO SIMPLE: REGRESIÓN POLINOMIAL

Tarea: Tomar un vector de entrada x , y predecir un valor real de salida y .

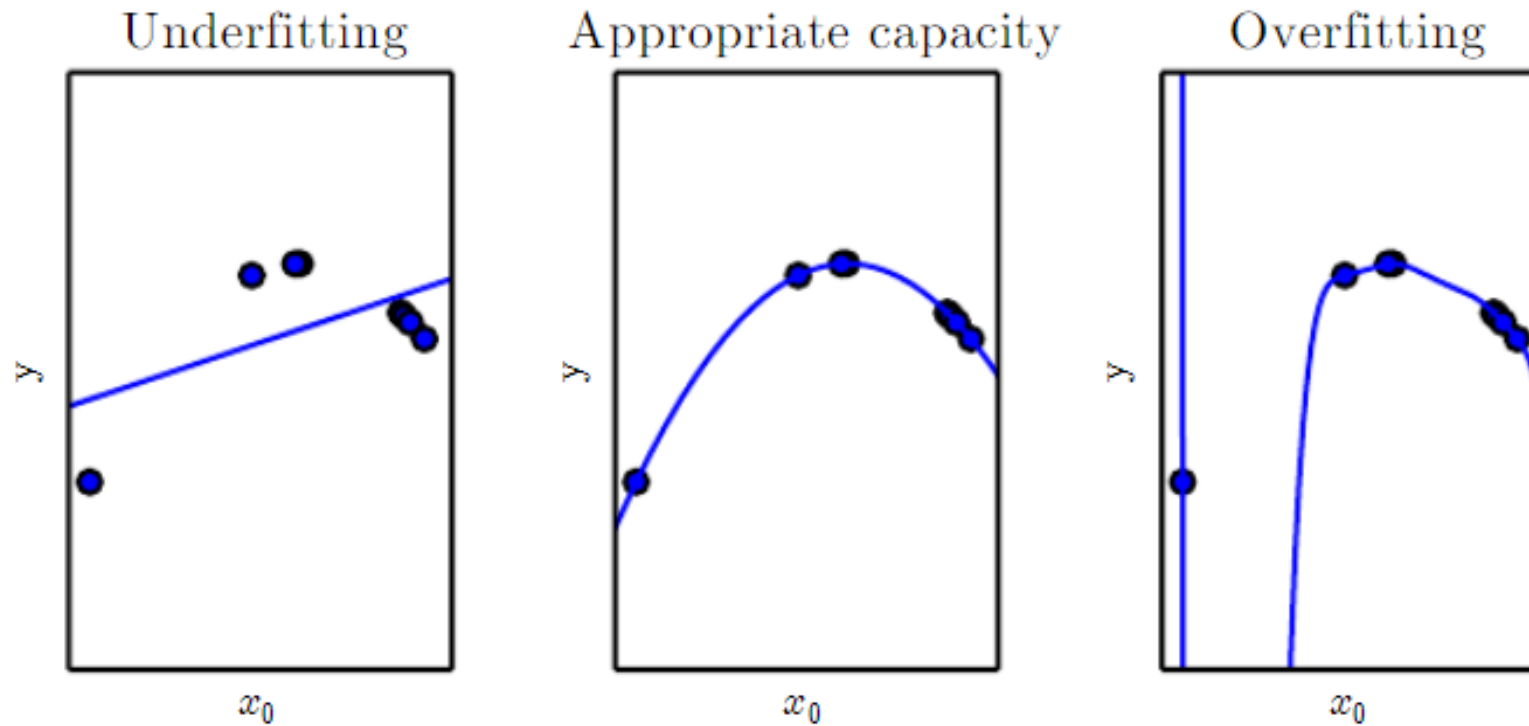
Experiencia: Datos en los que se basará la regresión.

Medida de performance: Medidas de error. Por ejemplo, Error cuadrático medio de la regresión.

$$y = w * x + b$$

w, b: parámetros del modelo.

EJEMPLO SIMPLE: REGRESIÓN POLINOMIAL



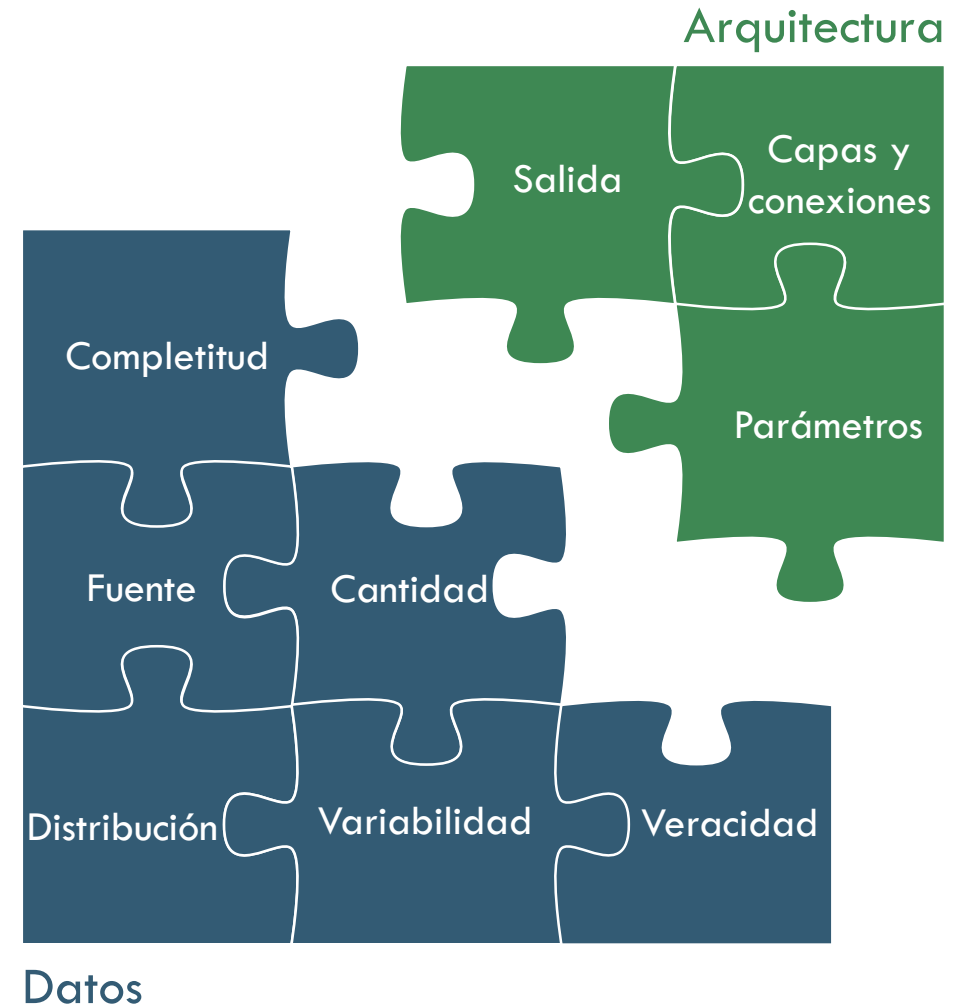
DESARROLLO DE UN SISTEMA DE ML

Características



DIMENSIONES DE UN SISTEMA DE ML

Regla: cuanto mayor sea la calidad de los datos de entrada, mayor será la precisión del algoritmo de ML al hacer su tarea.



CONCEPTOS CLAVE HASTA AQUÍ

Modelo de ML

Aprendizaje Supervisado, No Supervisado y por Refuerzo

Problemas de Clasificación y de Regresión.

Datasets

- Labeling
- Training + Validation + Test
- Cross Validation

Entrenamiento e Inferencia

Minimizar el error de entrenamiento (pero generalizando!)

Underfitting y Overfitting

¿Cómo establecer la

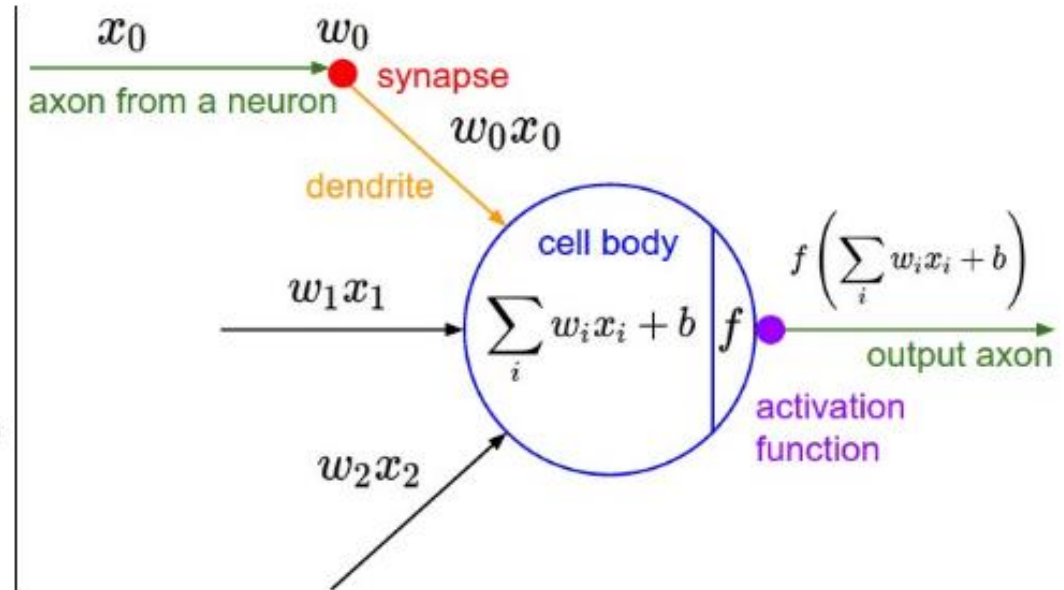
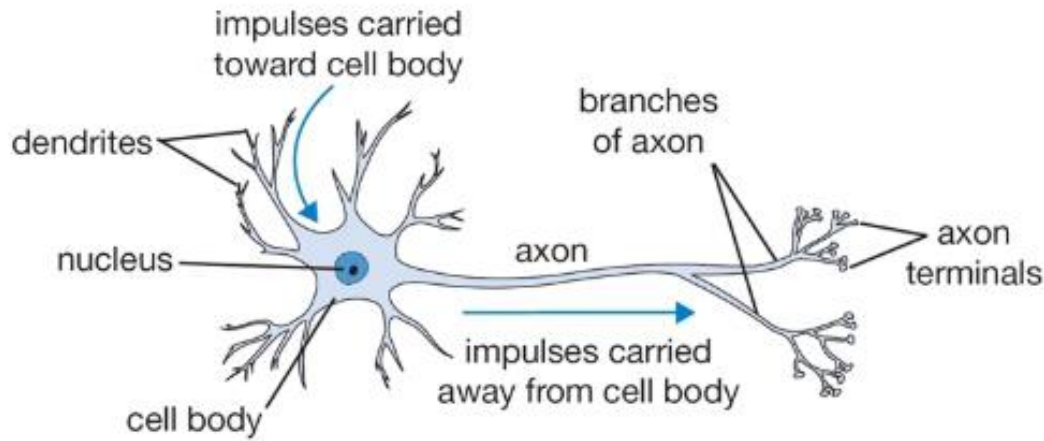
ARQUITECTURA

del Modelo a utilizar?

3

REDES NEURONALES

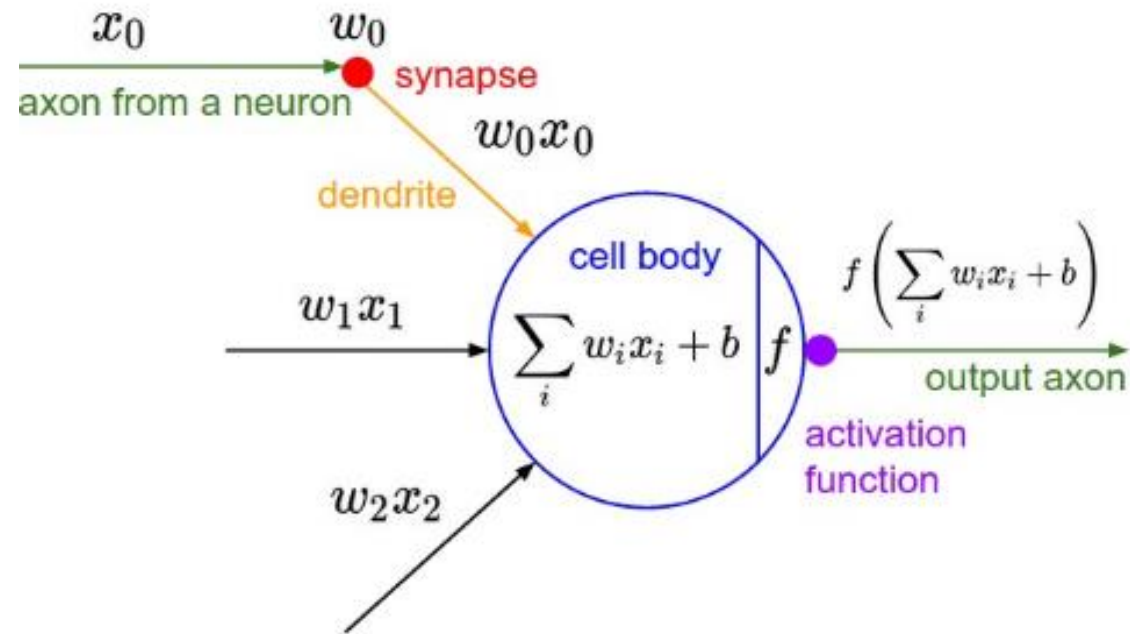
MODELO BIOLÓGICO



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

RETOMANDO: REGRESIÓN POLINOMIAL

$$y = \mathbf{w} * \mathbf{x} + b$$

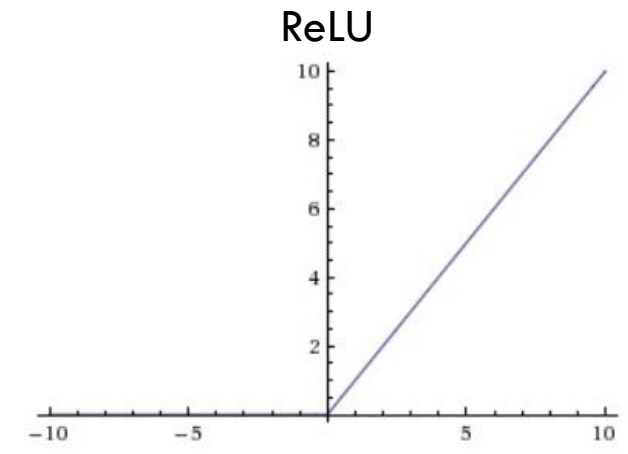
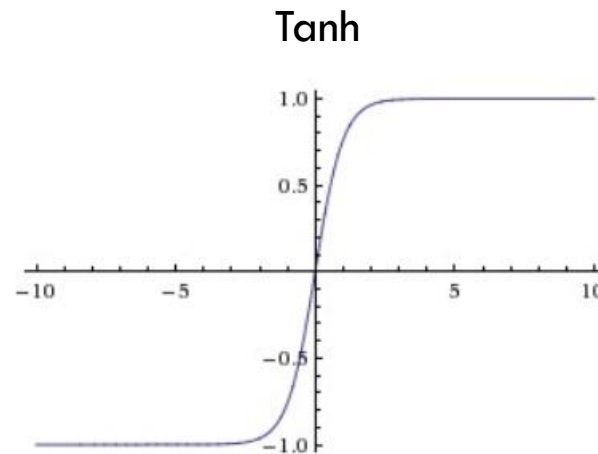
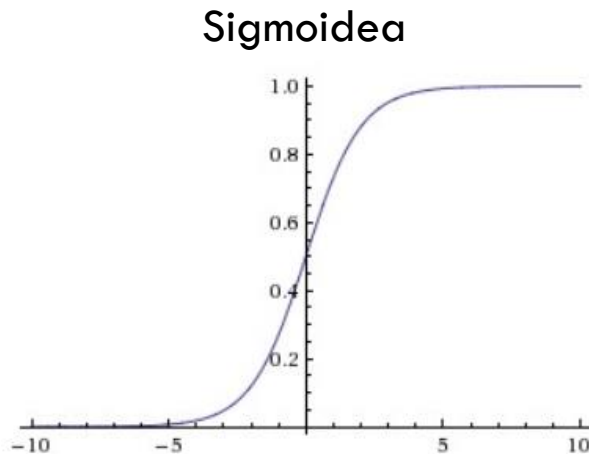


FUNCIONES DE ACTIVACIÓN

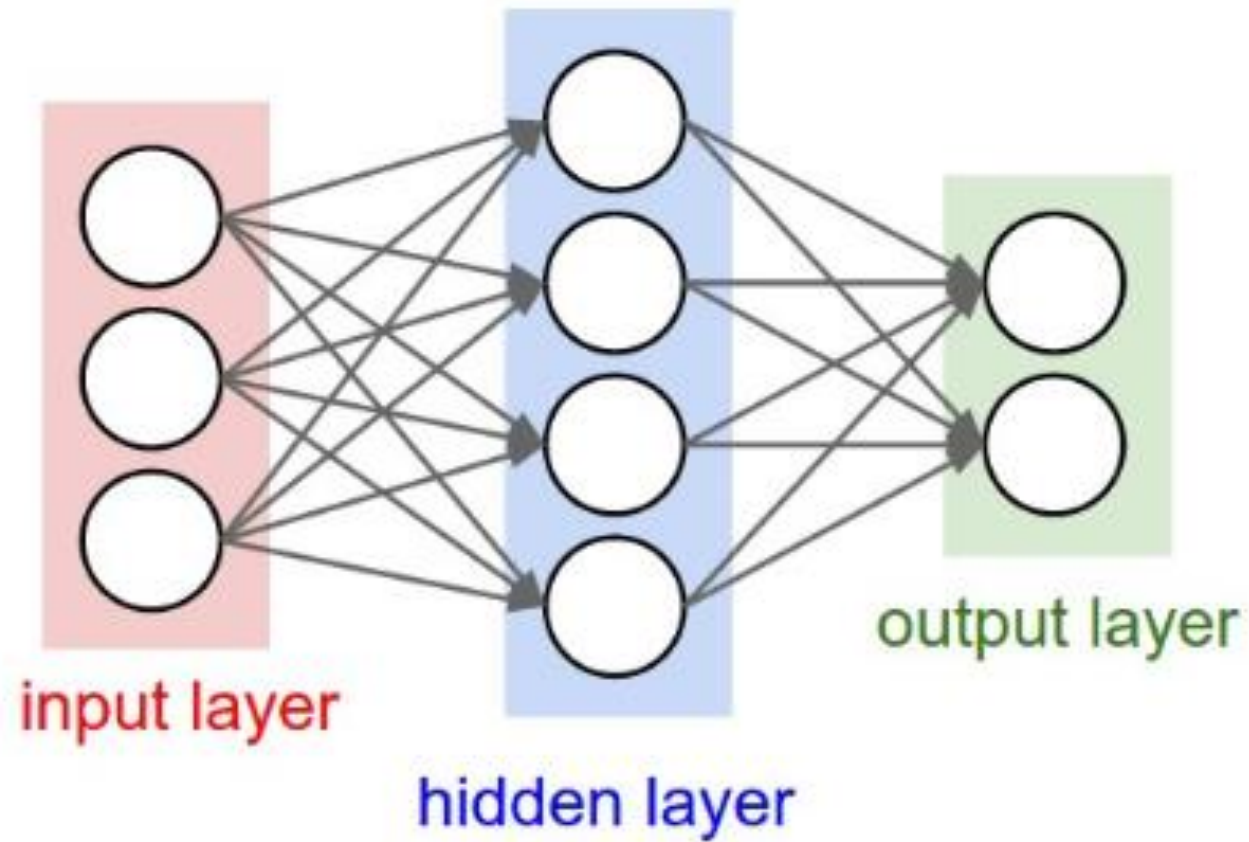
Una neurona se activará si supera un cierto valor límite impuesto por la función.

Su objetivo es limitar características que tengan valores bajos.

Funciones de Activación comunes:



REDES NEURONALES



Training in
progress...

$\boxed{q} \rightarrow 9$



PARÁMETROS DE ENTRENAMIENTO

Batch: cada cuanto se actualizan los pesos.

- Online: se actualizan después de cada dato de entrada.
- Full batch: lo hacen después de todos los datos de entrada.
- Mini batch: lo hacen después de unos pocos datos de entrada (más usado).

Learning rate: con qué magnitud se actualizan los pesos.

- Puede ser fijo o variable.

Algoritmo de convergencia: cómo nos movemos dentro de un espacio de error multidimensional.

- Usualmente, algoritmos de gradientes iterativos.
- Stochastic gradient descent es uno de los más comunes.
- Suele ir acompañado de algunas técnicas “aceleradoras” (momentum, Adam).

A JUGAR!

<https://playground.tensorflow.org>