

Aprendizaje Profundo

Maestría en Modelación y Optimización de Procesos
CIMAT-Aguascalientes

Dra. Lilí Guadarrama Bustos¹
Dr. Isidro Gómez-Vargas²

¹Investigadora tiempo completo en CIMAT-Aguascalientes

²Investigador posdoctoral en ICF-UNAM

Clase del semestre enero-julio 2022

Contenido

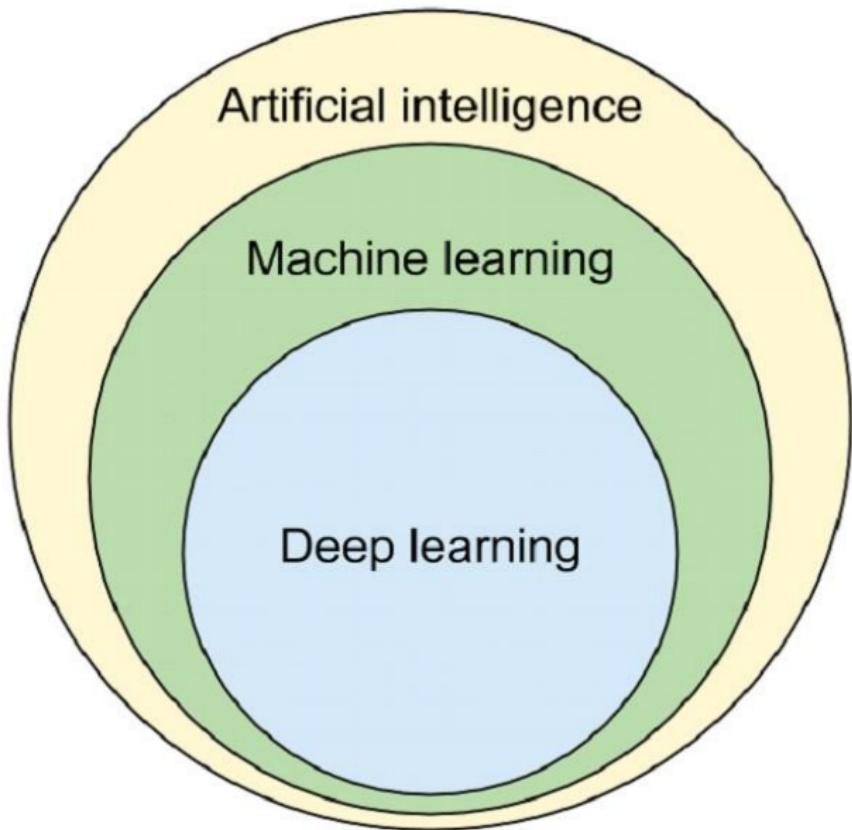
1 Clase 1: Bases de Aprendizaje Automático

- Algoritmos de aprendizaje.
- Capacidad, subajuste y sobreajuste
- Hiperparámetros y conjuntos de validación.
- Estimadores, sesgo y varianza.
- Estimación de Máxima Verosimilitud(MLE).
- Estadística Bayesiana.
- Algoritmos de aprendizaje supervisado.
- Descenso del gradiente estocástico.
- Desafíos actuales Aprendizaje Profundo.

2 Clase 2: Deep Feedforward Networks

- 6.1 Ejemplo: Aprendiendo XOR
- Gradient-based learning
- Unidades ocultas
- Diseño de arquitectura
- Backpropagation y otros algoritmos de diferenciación

Aprendizaje profundo



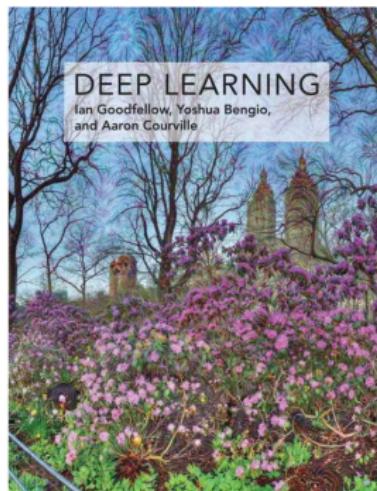
Bibliografía

Bibliografía principal

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

Diapositivas:

- https://www.deeplearningbook.org/lecture_slides.html
- <https://github.com/InfolabAI/DeepLearning>



Born	March 5, 1964 (age 57) Paris, France
Citizenship	Canada
Alma mater	McGill University
Known for	Deep learning, neural machine translation, generative adversarial networks, "attention model"®, word embeddings, denoising auto-encoders, neural language models, learning to learn Marie-Victorin Prize (2012) Turing Award (2018) AAAI Fellow (2019)
Awards	
Scientific career	



Born	1985/1986 (age 35–36)
Nationality	American
Alma mater	Stanford University Université de Montréal
Known for	Generative adversarial networks, Adversarial Examples
Scientific career	
Fields	Computer science
Institutions	Apple Inc. Google Brain OpenAI
Thesis	Deep Learning of Representations and its Application to Computer Vision (2014)
Doctoral advisor	Yoshua Bengio
Website	www.iangoodfellow.com



Aaron Courville

Université de Montréal

Dirección de correo verificada de umontreal.ca - [Página personal](#)
Machine learning Artificial Intelligence

TÍTULO

CITADO POR

Generative adversarial nets

I Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, ... Advances in neural information processing systems 27

40148

Deep learning

I Goodfellow, Y Bengio, A Courville
Nature

36121

Representation learning: A review and new perspectives

Y Bengio, A Courville, P Vincent
IEEE transactions on pattern analysis and machine intelligence 35 (8), 1798-1828

10333

Temario

Se abordarán partes de los siguientes capítulos:

- Capítulo 5: Bases del Aprendizaje Automático.
- Capítulo 6: Redes neuronales profundas de propagación hacia adelante.
- Capítulo 7: Regularización.
- Capítulo 8: Optimización para el entrenamiento de modelos profundos.

Temario

- Capítulo 9: Redes neuronales convolucionales.
- Capítulo 10: Modelación secuencial (redes recurrentes y recursivas).
- Capítulo 11: Metodología práctica.
- Capítulo 12: Aplicaciones.
- Temas elegidos por el grupo de la parte III del libro (capítulos 13-20).

Algoritmos de aprendizaje.

“ Un programa de computadora se dice que aprende de la experiencia E con respecto a cierta clase de tareas T y con medida de rendimiento P , si su rendimiento en las tareas en T , medido por P , mejora con la experiencia E . ”

Tom Mitchell, 1997

Glosario

- **Ejemplo.** $x \in \mathbf{R}^n$, colección de n características.
- **Conjunto de datos.** Colección de ejemplos.
- **Características.** Atributos.
- **Matriz de datos.**

Tareas T

- Clasificación
- Regresión
- Transcripción
- Traducción
- Salidas estructuradas
- Detección de anomalías
- Síntesis y muestreo
- Imputing valores perdidos
- Quitar ruido
- Estimación de densidad o probabilidad

La medida de rendimiento P

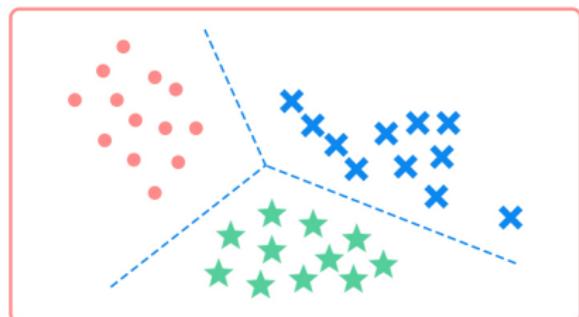
- Exactitud $\frac{TP+TN}{TP+TN+FP+FN}$
- Error cuadrático medio $MSE = \frac{1}{n} \sum_i^n ||\hat{y}_i - y_i||$
- Densidad de probabilidad
- etc

La experiencia, E



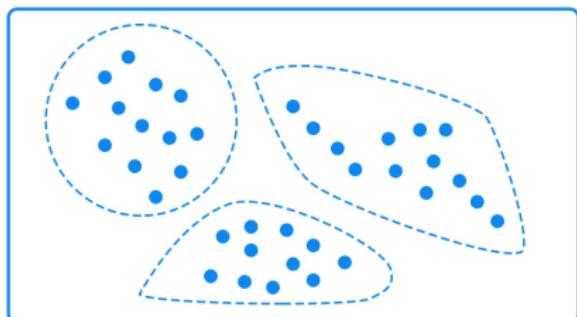
Supervised vs. Unsupervised Learning

Classification



Supervised learning

Clustering



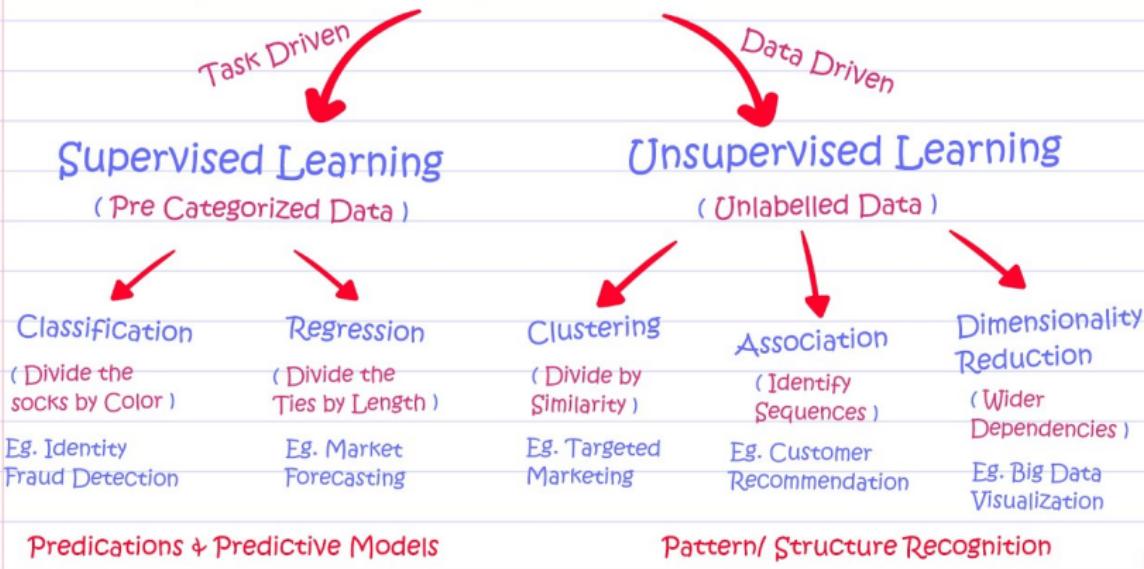
Unsupervised learning

Fuente:

[https://analystprep.com/study-notes/cfa-level-2/quantitative-method/
supervised-machine-learning-unsupervised-machine-learning-deep-learning/
attachment/img_12-4/](https://analystprep.com/study-notes/cfa-level-2/quantitative-method/supervised-machine-learning-unsupervised-machine-learning-deep-learning/attachment/img_12-4/)

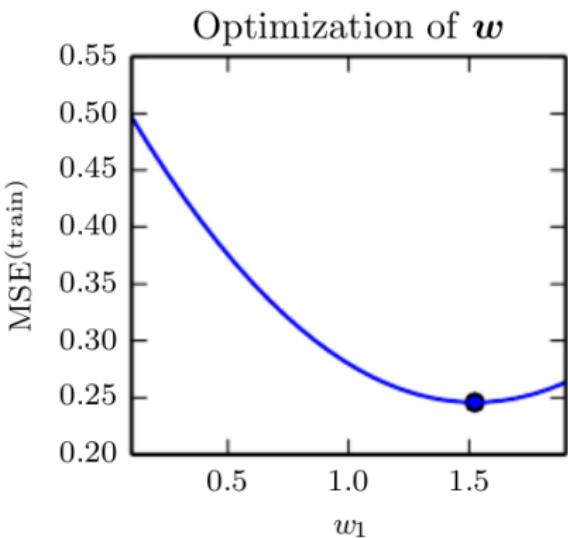
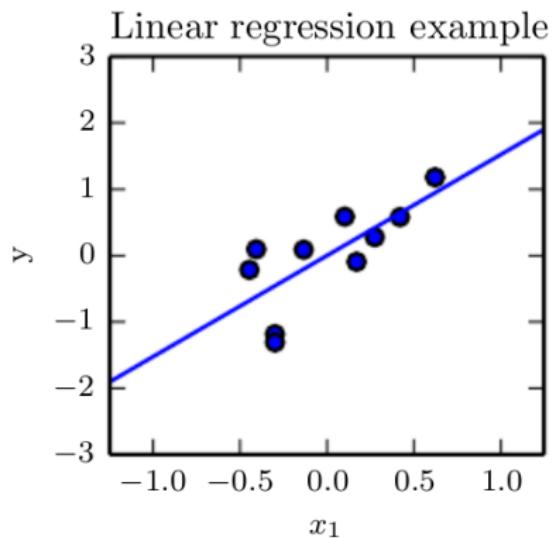
La experiencia, E

Classical Machine Learning



Fuente: <https://medium.com/@recrosoft.io/supervised-vs-unsupervised-learning-key-differences-cdd46206cdcb>

Ejemplo: regresión lineal

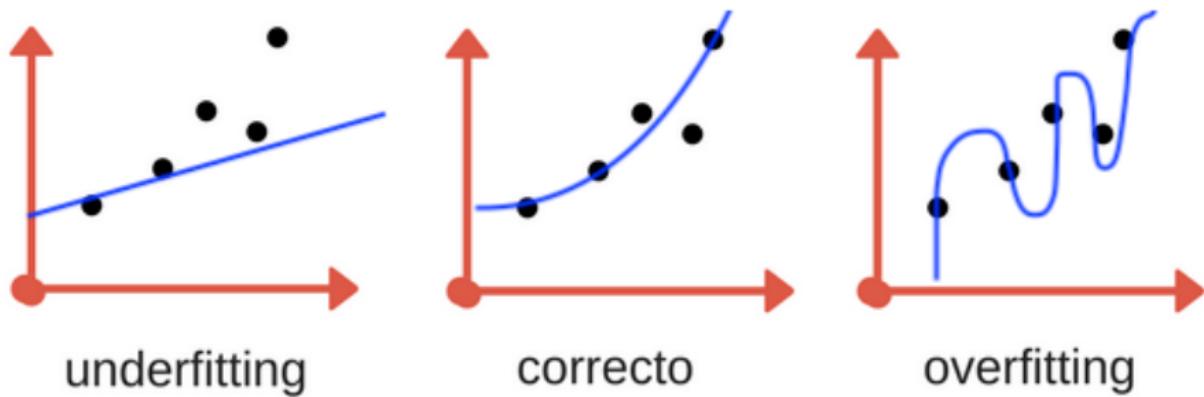


Ver páginas 107-108 del libro.

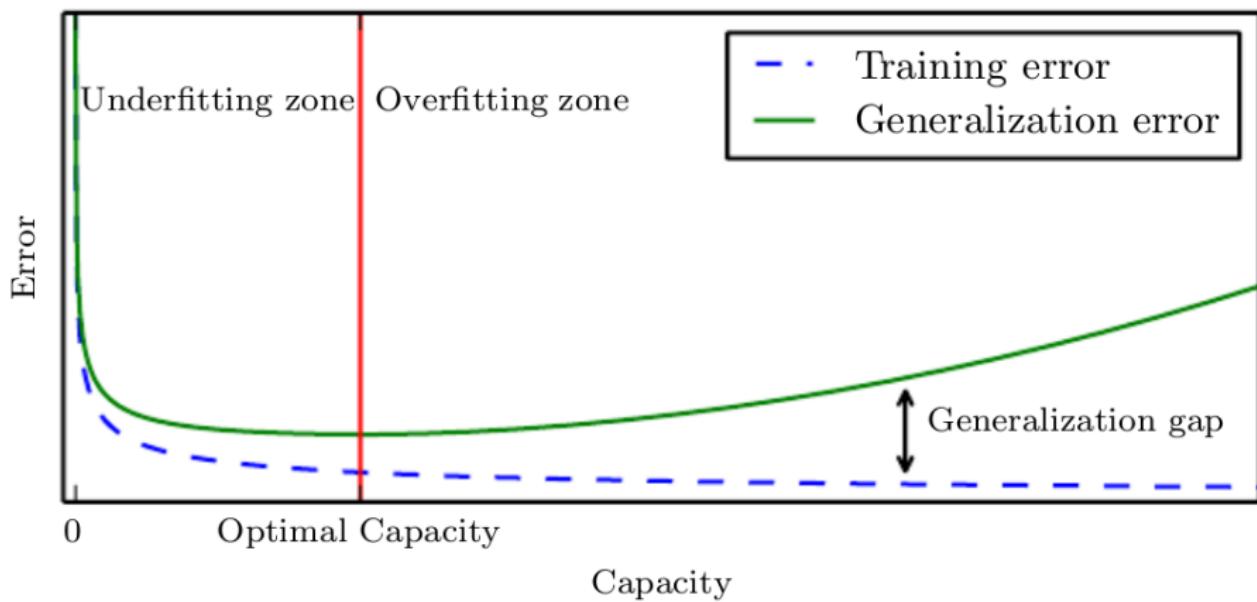
Capacidad, subajuste y sobreajuste

- Algo que separa al aprendizaje automático de la optimización es el uso de un error de generalización o error de prueba (*test error*).
- Hay que minimizar tanto el error de entrenamiento como el error de generalización.
- La brecha entre ambos errores debe ser pequeña.

Capacidad, subajuste y sobreajuste



Capacidad, subajuste y sobreajuste



Capacidad, subajuste y sobreajuste: No free lunch theorem



Capacidad, subajuste y sobreajuste: No free lunch theorem

"No Free Lunch" :(

D. H. Wolpert. The supervised learning no-free-lunch theorems. In Soft Computing and Industry, pages 25–42. Springer, 2002.

Our model is a simplification of reality



Simplification is based on assumptions (model bias)



Assumptions fail in certain situations

Roughly speaking:

"No one model works best for all possible situations."

Fuente: <https://analyticsindiamag.com/what-are-the-no-free-lunch-theorems-in-data-science/>

Capacidad, subajuste y sobreajuste: Regularización

Por ejemplo, decaimiento del peso:

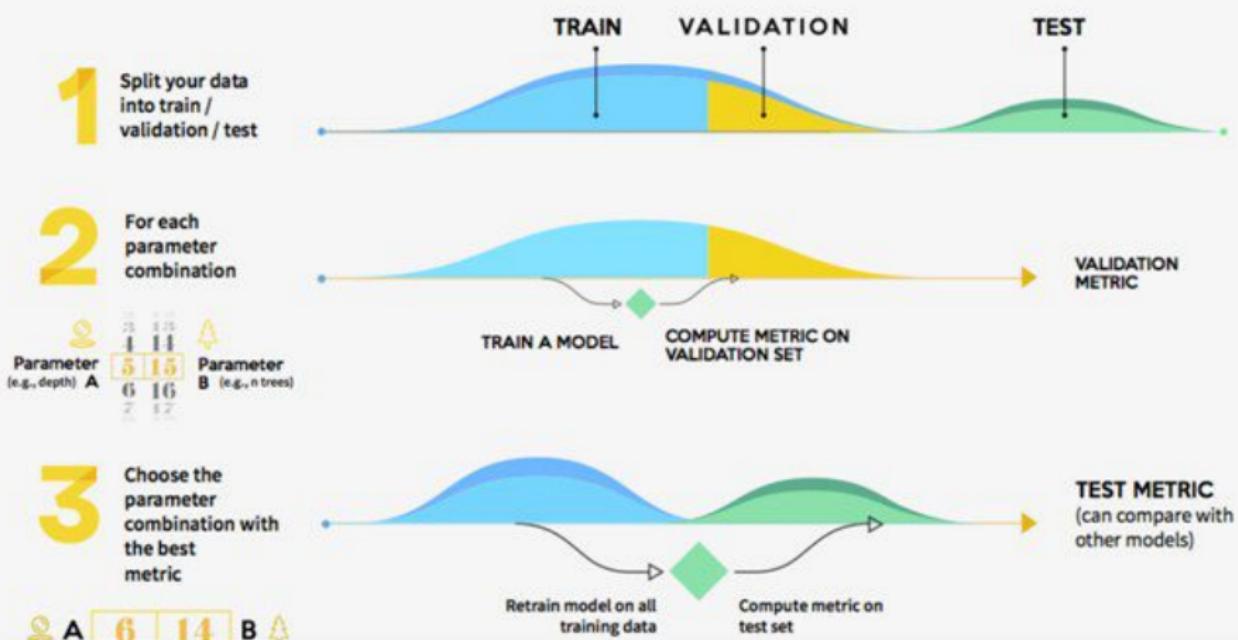
$$J(w) = MSE + \lambda w^T w$$

Def.

Regularización es cualquier mecanismo que ayuda al agoritmo de aprendizaje a recudir su error de generalización.

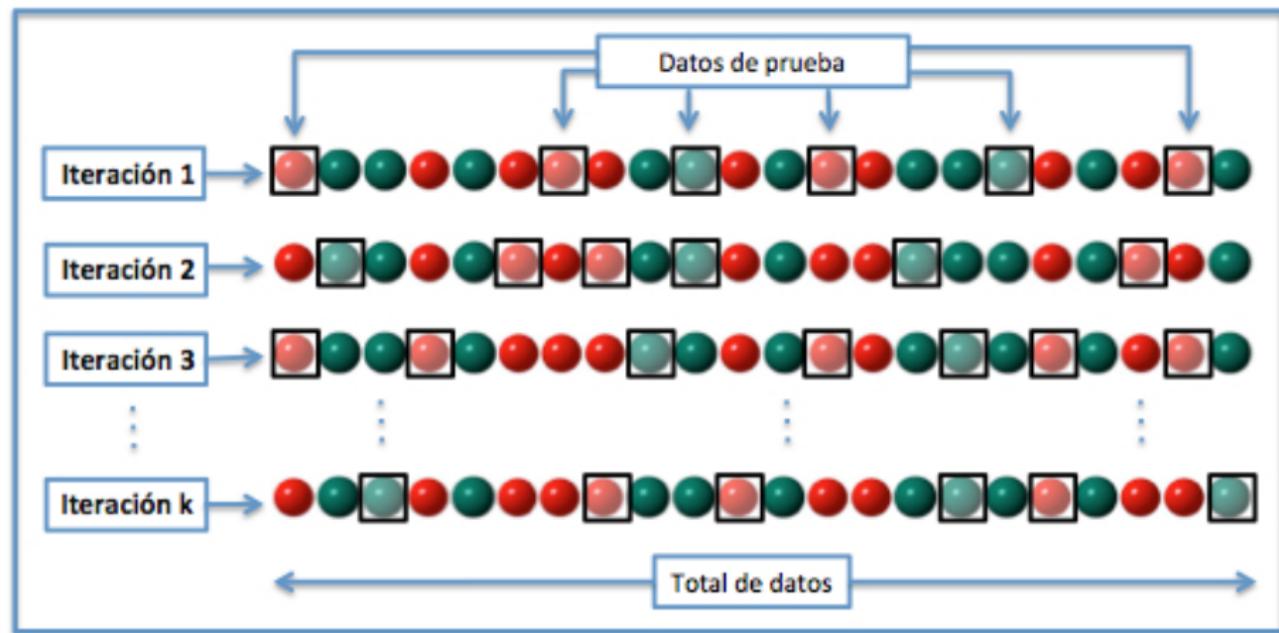
Hiperparámetros y conjuntos de validación.

HOLDOUT STRATEGY



Fuente: [https://www.kdnuggets.com/2017/08/
dataiku-predictive-model-holdout-cross-validation.html](https://www.kdnuggets.com/2017/08/dataiku-predictive-model-holdout-cross-validation.html)

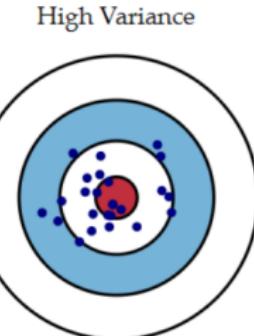
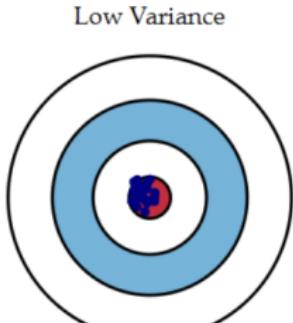
Hiperparámetros y conjuntos de validación: validación cruzada



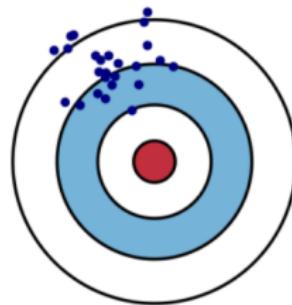
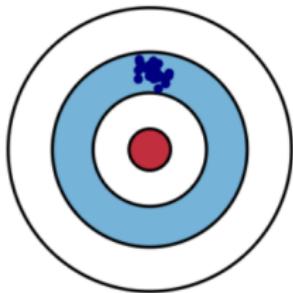
Fuente: https://es.wikipedia.org/wiki/Validaci%C3%B3n_cruzada

Estimadores, sesgo y varianza.

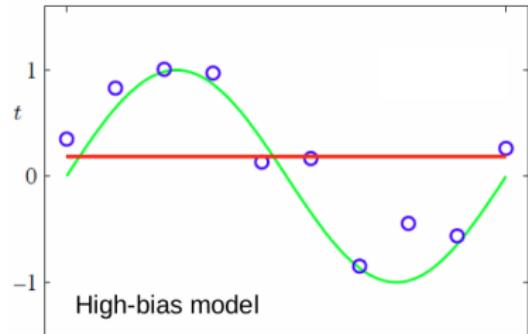
Low Bias



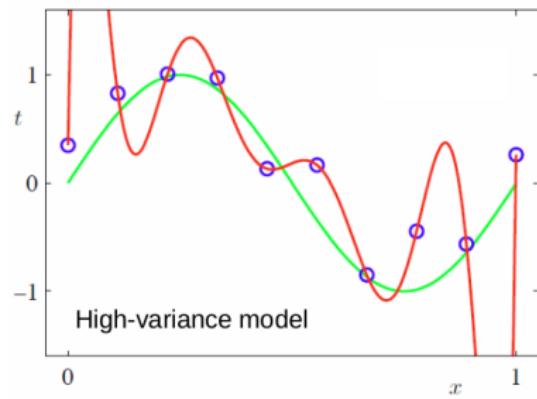
High Bias



Scott Fortmann-Roe, Understanding the Bias-Variance Tradeoff, 2012

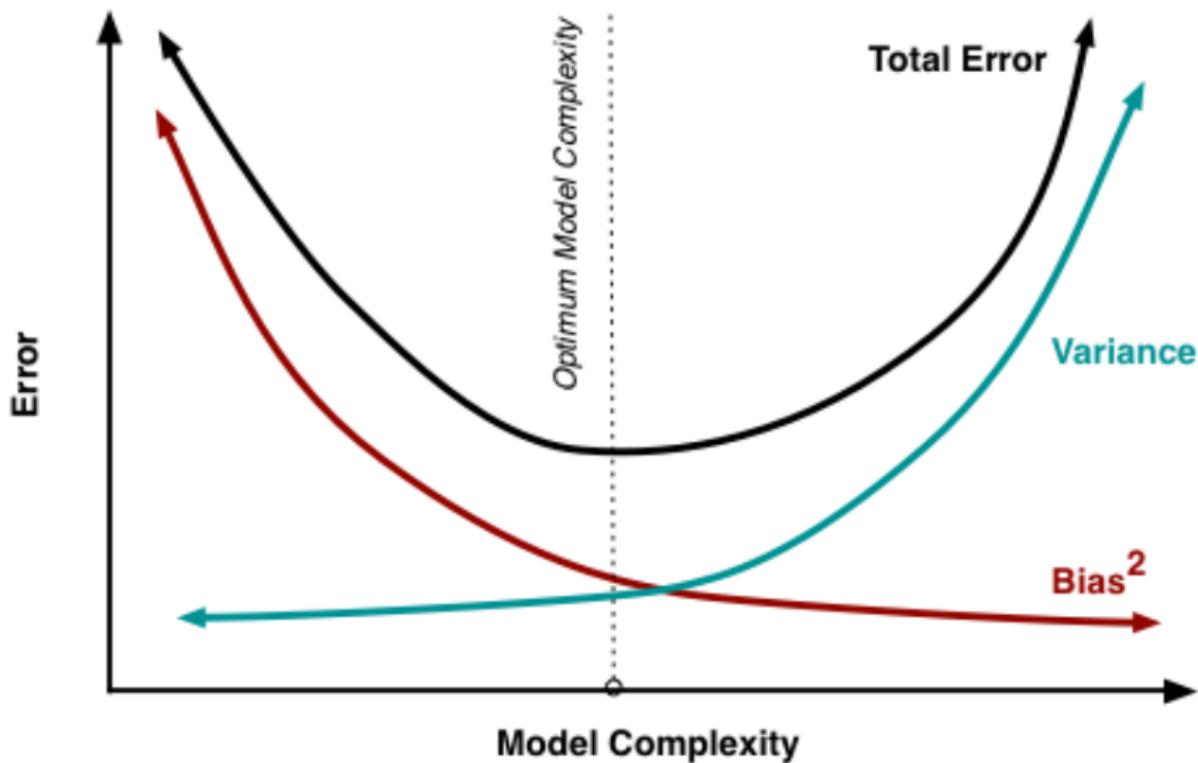


High-bias model



High-variance model

Estimadores, sesgo y varianza.



Estimadores, sesgo y varianza.

Visitar los siguientes enlaces:

- <http://scott.fortmann-roe.com/docs/BiasVariance.html>
- <https://ml.berkeley.edu/blog/posts/crash-course/part-4/>
- <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/>

Estimación de Máxima Verosimilitud(MLE).

- Estimación del Máximo Likelihood (MLE):

$$\ln \mathcal{L}(D, \theta) = \sum_{i=1}^n \ln f(x_i; \theta),$$

$$\theta_{MLE} = \arg \max(\mathcal{L}(\theta, D))$$

Teorema de Bayes

Considerando funciones de densidad de probabilidad:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}, \quad (1)$$

donde:

$$P(D) = \int_{\mathbb{R}^N} P(D|\theta)P(\theta)d\theta, \quad (2)$$

Estadística Bayesiana.

- Estimación del A Posteriori (MAP) ó estimación de parámetros ó inferencia Bayesiana:

$$\theta_{MAP} = \arg \max(\mathcal{L}(\theta, D)P(\theta))$$

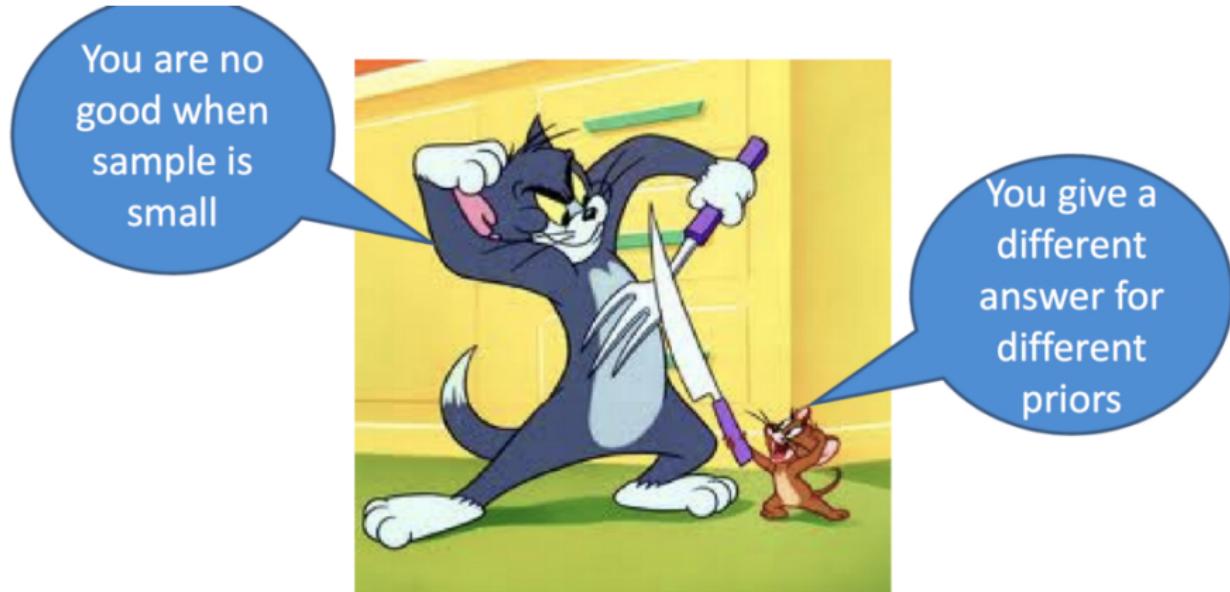
Estadística Bayesiana.

- Estimación del A Posteriori (MAP) ó estimación de parámetros ó inferencia Bayesiana:

$$\theta_{MAP} = \arg \max(\mathcal{L}(\theta, D)P(\theta))$$

- Comparación de modelos (puede ser parte de la inferencia Bayesiana).

Estadística frecuentista vs Bayesiana.



Fuente:<https://laptrinhx.com/maximum-likelihood-estimation-vs-maximum-a-posteriori-2539680111/>

Estadística frecuentista vs Bayesiana.

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$

$$\boldsymbol{\theta}^{\text{MLE}} = \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^N p(\mathbf{x}^{(i)} | \boldsymbol{\theta})$$

Maximum Likelihood Estimate (MLE)

$$\boldsymbol{\theta}^{\text{MAP}} = \operatorname{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^N p(\mathbf{x}^{(i)} | \boldsymbol{\theta}) p(\boldsymbol{\theta})$$

Maximum *a posteriori* (MAP) estimate

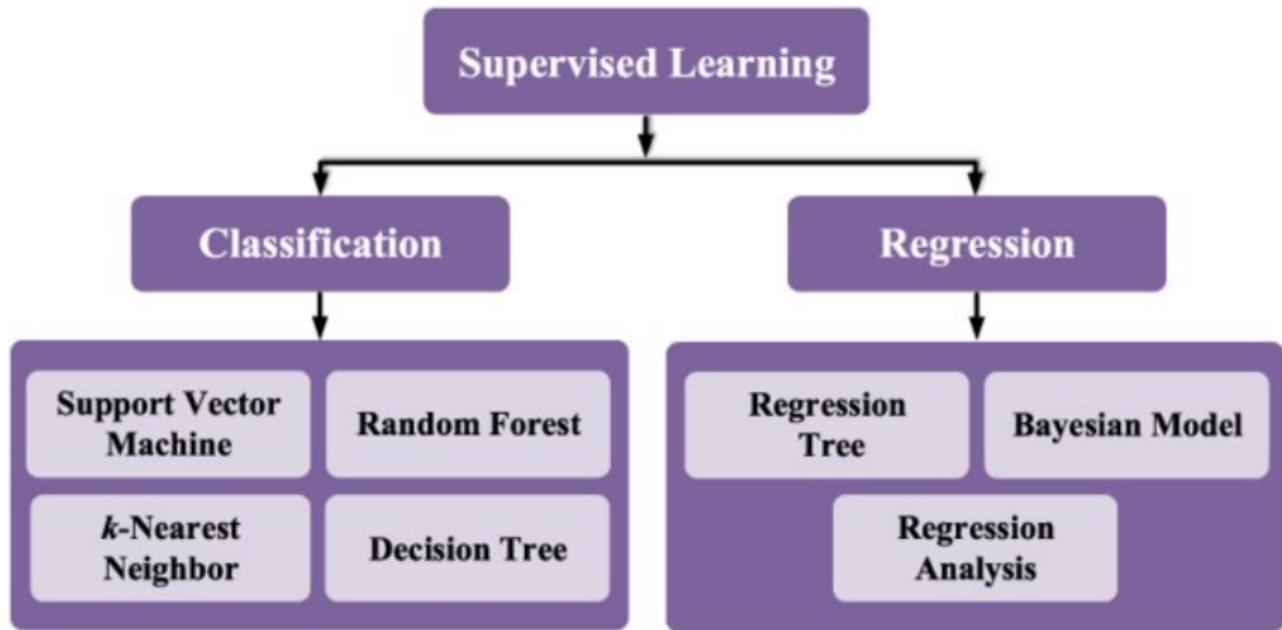


Prior

Fuente: <https://medium.com/@tzjy/>

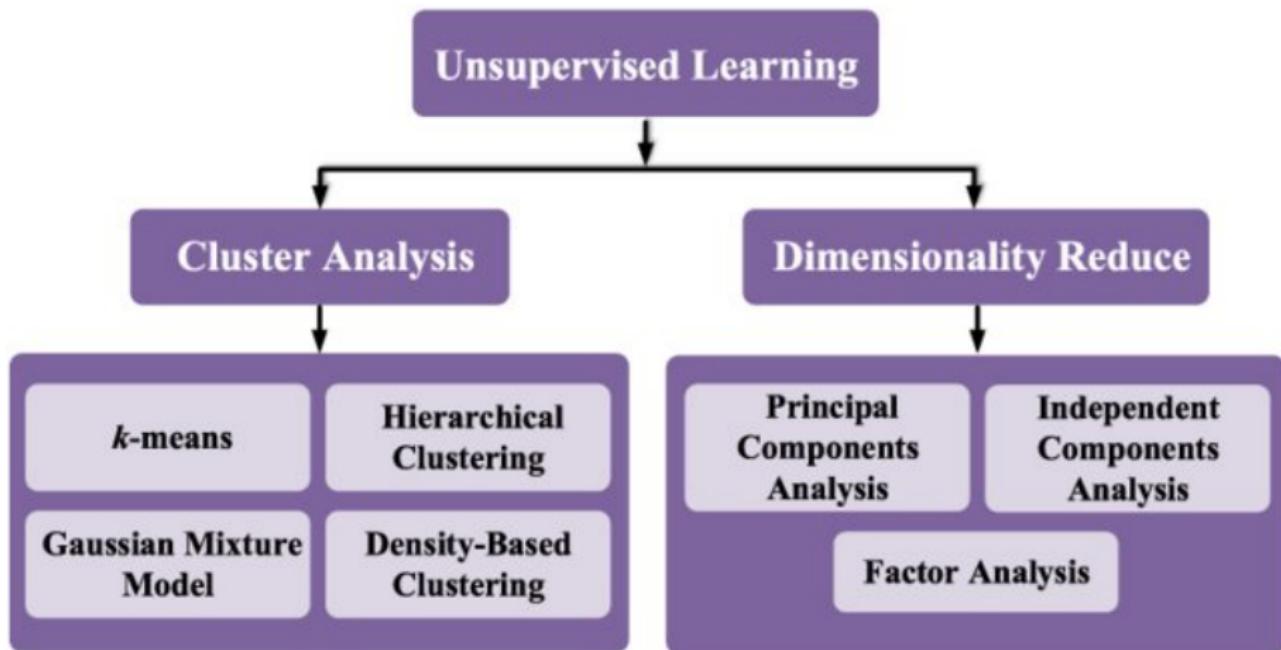
whats-the-difference-between-maximum-likelihood-estimation-mle-and-maximum

Algoritmos de aprendizaje supervisado.



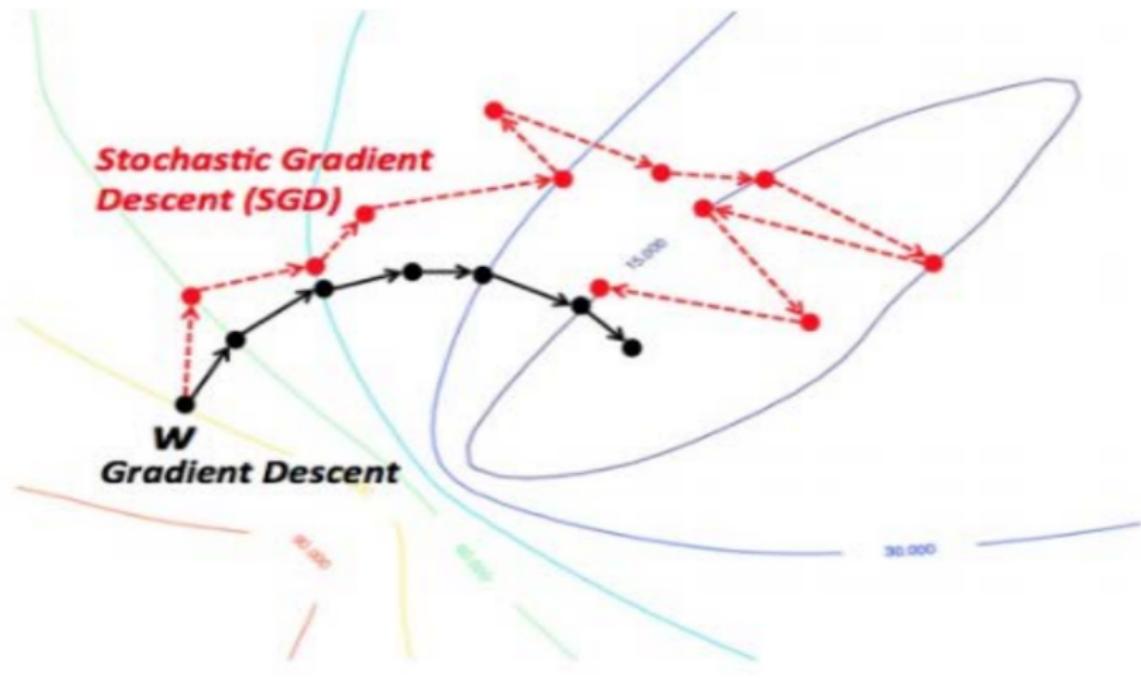
Fuente: arXiv:2003.10146

Algoritmos de aprendizaje no supervisado.



Fuente: arXiv:2003.10146

Descenso del gradiente estocástico.



Fuente: <https://www.slideshare.net/microlife/from-neural-networks-to-deep-learning>

Dra. Lili Guadarrama Bustos Dr. Isidro Gómez

Descenso del gradiente estocástico.

Tarea propuesta

Programar un descenso del gradiente estocástico y minimizar una función con él.

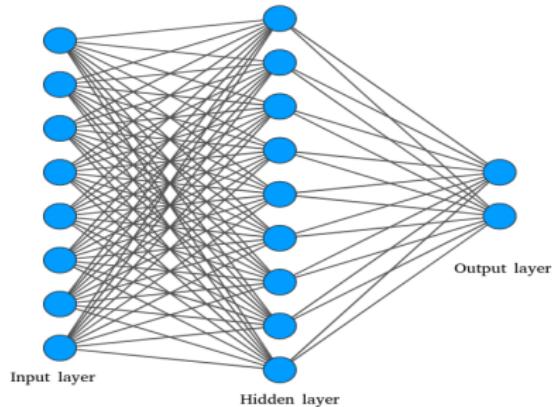
Retos que motivan al Aprendizaje Profundo.

- Curso de la dimensionalidad.
- Local Constancy and Smoothness Regularization.
- Manifold Learning.

Clase 2: Introducción al Aprendizaje Profundo

- Inicia la parte II del libro de referencia.
- Nos centraremos en el capítulo 6.

Perceptrón multicapa



Deep feedforward networks /feedforward neural networks

$$f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$$

$f^{(i)}$ es llamada la i -ésima pared de la red. $f(x)$ debe coincidir con $f^*(x)$, cada ejemplo x tiene asociada una etiqueta $y = f^*(x)$

Perceptrón multicapa

Para que obtener un algoritmo no-lineal, hay que emplear una función $\phi(x)$ para modificar las entradas a las funciones.

Truco del kernel (descrito en 5.7.2)

$$w^T x + b = b \sum_{i=1}^m \alpha_i x^T x^{(i)}$$

se puede cambiar $x - > \phi(x)$ y el producto punto por un kernel $k(x, x^{(i)}) = \phi(x) \dot{\phi}(x^{(i)})$. Entonces se pueden hacer predicciones mediante:

$$f(x) = b + \sum_{i=1}^m \alpha_i k(x, x^{(i)})$$

donde la función es no-lineal respecto a x , pero la relación entre $\phi(x)$ y $f(x)$ es lineal. También entre α y $f(x)$ es lineal.

Truco del kernel (descrito en 5.7.2)

La función basada en kernel es exactamente equivalente a preprocesar los datos aplicando $\phi(x)$ a todas las entradas, y luego aprender un modelo lineal en el nuevo espacio transformado.

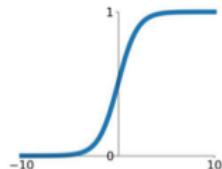
Ventajas:

- ① Permite aprender modelos que son funciones no lineales de x usando técnicas de optimización convexas.
- ② La función kernel, generalmente, permite una implementación que es significativamente más eficiente, en términos computacionales, que construir dos vectores $\phi(x)$ y luego tomar explícitamente sus productos.

Funciones de activación

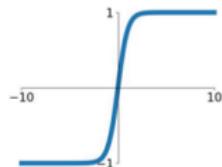
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



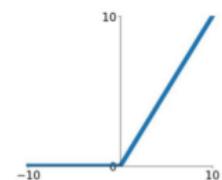
tanh

$$\tanh(x)$$



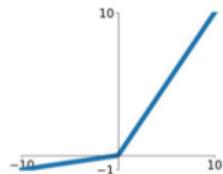
ReLU

$$\max(0, x)$$



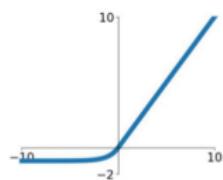
Leaky ReLU

$$\max(0.1x, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Multilayer Feedforward Networks are Universal Approximators

KURT HORNICK

Technische Universität Wien

MAXWELL STINCHCOMBE AND HALBERT WHITE

University of California, San Diego

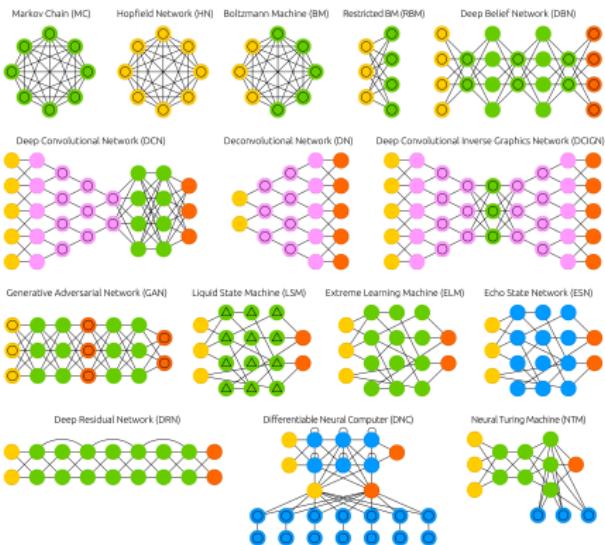
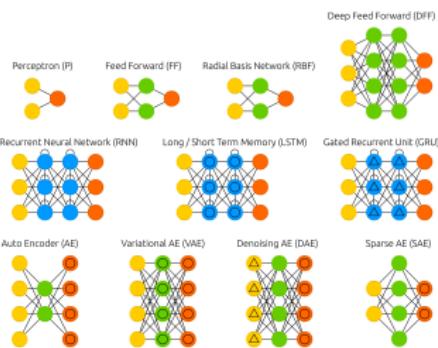
(Received 16 September 1988; revised and accepted 9 March 1989)

Abstract—This paper rigorously establishes that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators.

Keywords—Feedforward networks, Universal approximation, Mapping networks, Network representation capability, Stone-Weierstrass Theorem, Squashing functions, Sigma-Pi networks, Back-propagation networks.

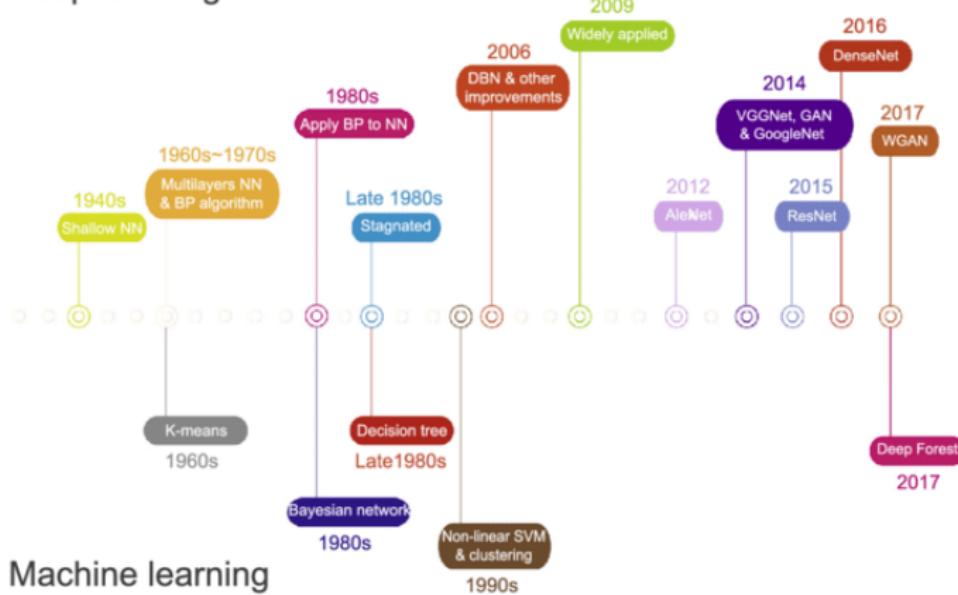
Múltiples arquitecturas

- Input Cell
- Backfed Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- ▲ Spiking Hidden Cell
- Capsule Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Gated Memory Cell
- Kernel
- Convolution or Pool



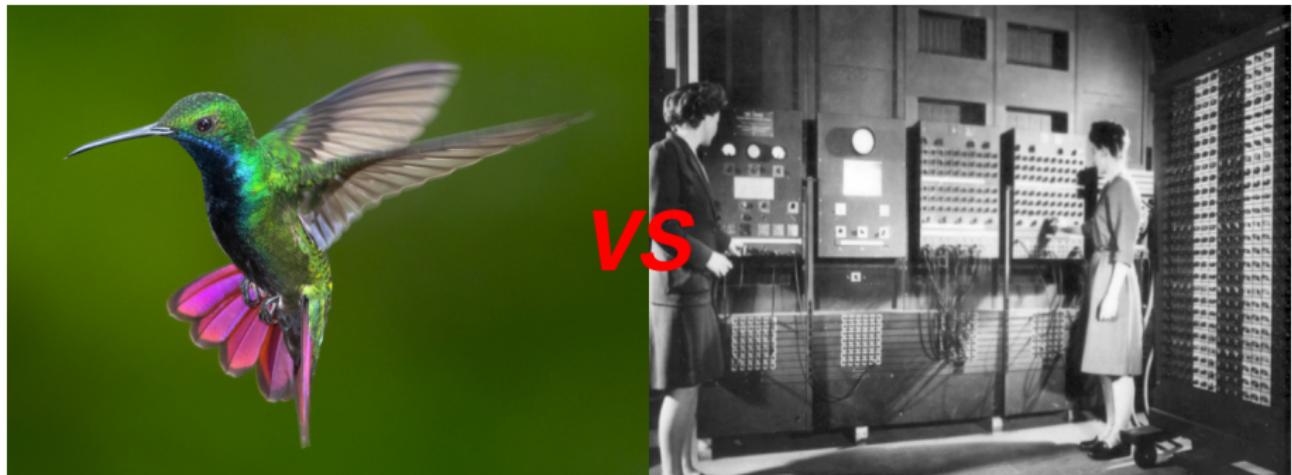
Cronología ML

Deep learning

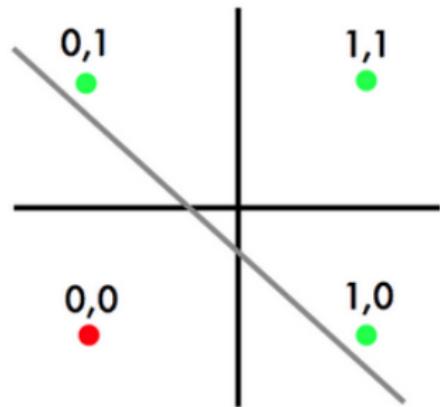


Aprendizaje máquina

1940s



El problema de la compuerta XOR



OR



XOR

Ver notebooks: <https://github.com/igomezv/DLCIMATAGS/tree/main/notebooks/clase%202>

Gradient-based learning

Unidades ocultas

Diseño de arquitectura

Backpropagation y otros algoritmos de diferenciación