

Inteligencia Artificial II

Juan Pablo Restrepo Uribe

Ing. Biomedico - MSc. Automatización y Control Industrial

jprestrepo@correo.iue.edu.co

Institución Universitaria de Envigado

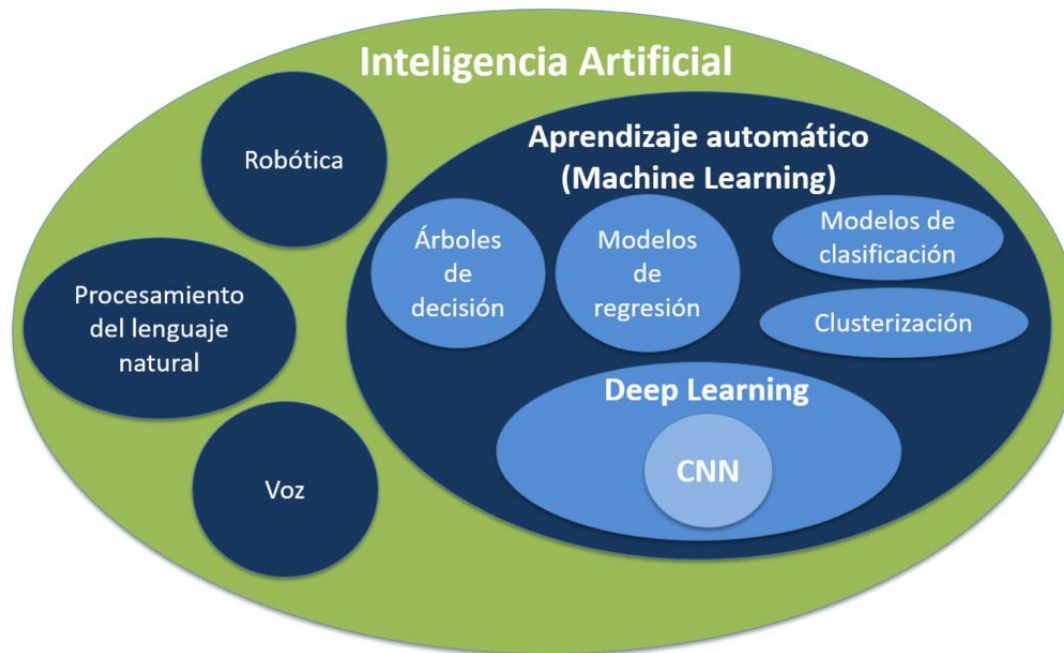
Inteligencia Artificial (IA)

Coloquialmente, el término inteligencia artificial se aplica cuando una máquina imita las funciones «cognitivas» que los humanos asocian con otras mentes humanas, como por ejemplo: “percibir”, “razonar”, “aprender” y “resolver problemas”.

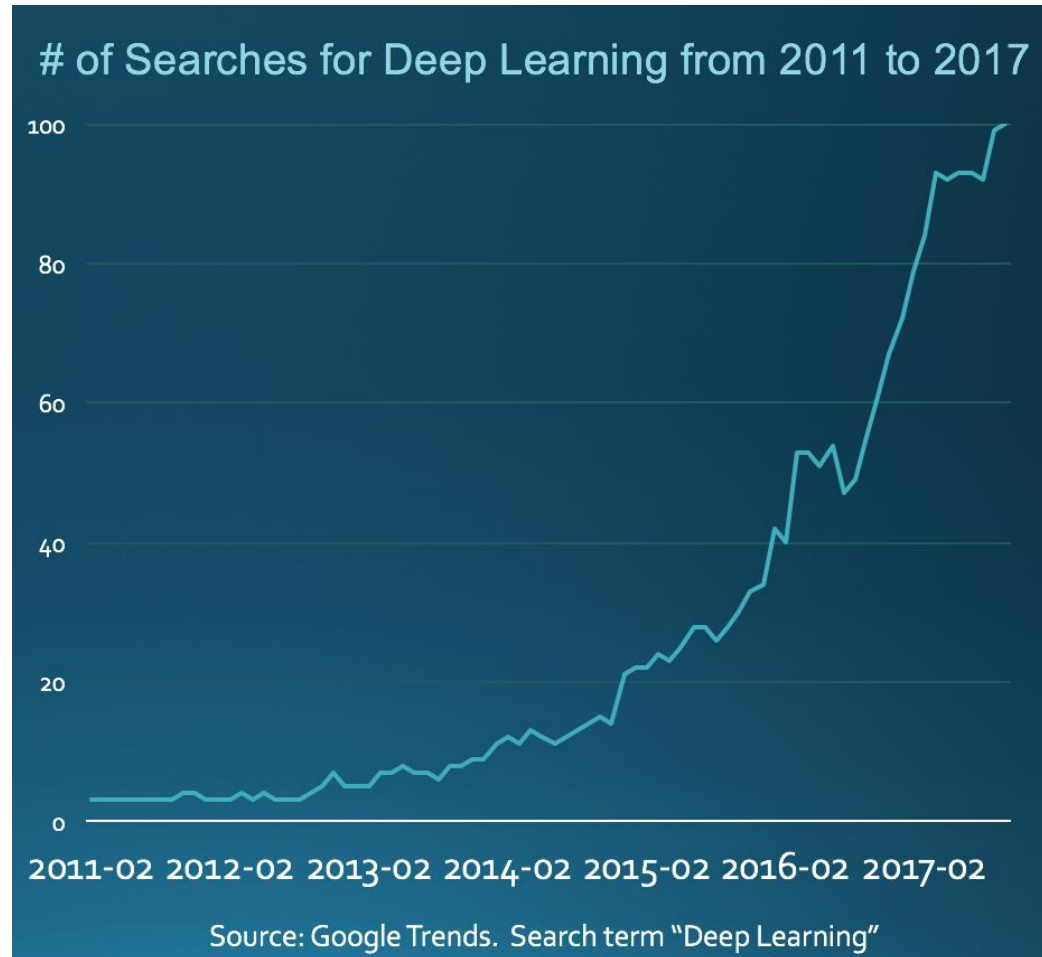
“La capacidad de un sistema para interpretar correctamente datos externos, para aprender de dichos datos y emplear esos conocimientos para lograr tareas y metas concretas a través de la adaptación flexible”

Deep Learning (DL)

Deep Learning (DL) es un subcampo de Machine-Learning (ML) en inteligencia artificial (AI) que trata con algoritmos inspirados desde la estructura biológica y funcionamiento de un cerebro para ayudar a las máquinas a tener inteligencia.



Deep Learning (DL)



Deep Learning - IA

AI es una forma genérica que puede ser definida como la calidad de inteligencia introducida en máquinas.



Las máquinas usualmente son tontas, entonces para hacerlas inteligentes introducimos una especie de inteligencia para que puedan tomar decisiones en forma independiente.

ML puede ser definido como el proceso de inducir inteligencia en un sistema o máquina sin programación explícita.

Deep Learning - IA

Un ejemplo de ML, podría ser un sistema que puede predecir si un estudiante aprobará o reprobará en su siguiente examen en base a su comportamiento previo

¿Y DL donde aparece en este contexto?

Ocurre que mientras ML trabaja muy bien para una gran variedad de problemas, este falla en casos específicos que son muy simples para los humanos, por ejemplo: clasificar una imagen como un gato o un perro; distinguir una voz femenina y una masculina; y muchas cosas más.

Deep Learning (DL)

Imitar el proceso biológico del cerebro humano. Es sabido que el cerebro es una orquesta de billones de neuronas conectadas que se adaptan para aprender nuevas cosas.



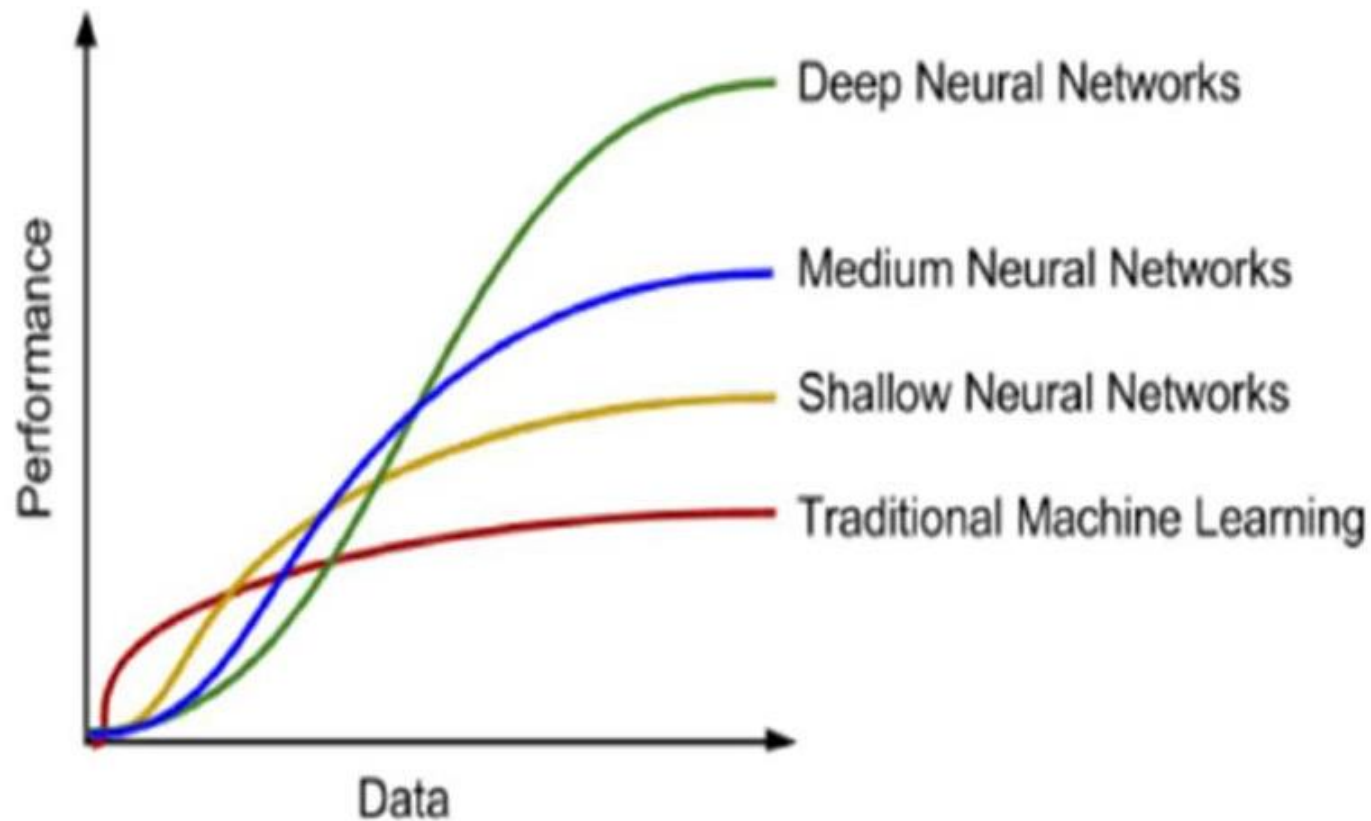
Deep Learning (DL)

Cuando la investigación unió ML con estas redes neuronales, aparece el campo de Deep-Learning (DL), el que se enmarca en el desarrollo de Deep-Neural-Network (DNNs), que corresponde a redes neuronales de muchas capas. Aquí DL toma ventaja en donde ML no podría resolver el problema.

Todo esto ha ayudado a definir una nueva regla de oro:

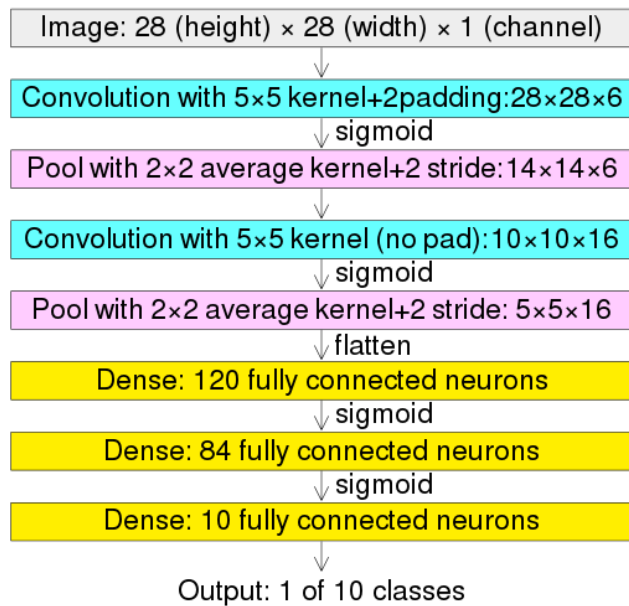
- ML no mejorará su performance cuando se incrementan datos de entrenamiento, luego de alcanzar cierto umbral.
- DL puede mejorar con más datos, de manera más eficiente para mejorar la performance.

Deep Learning (DL)

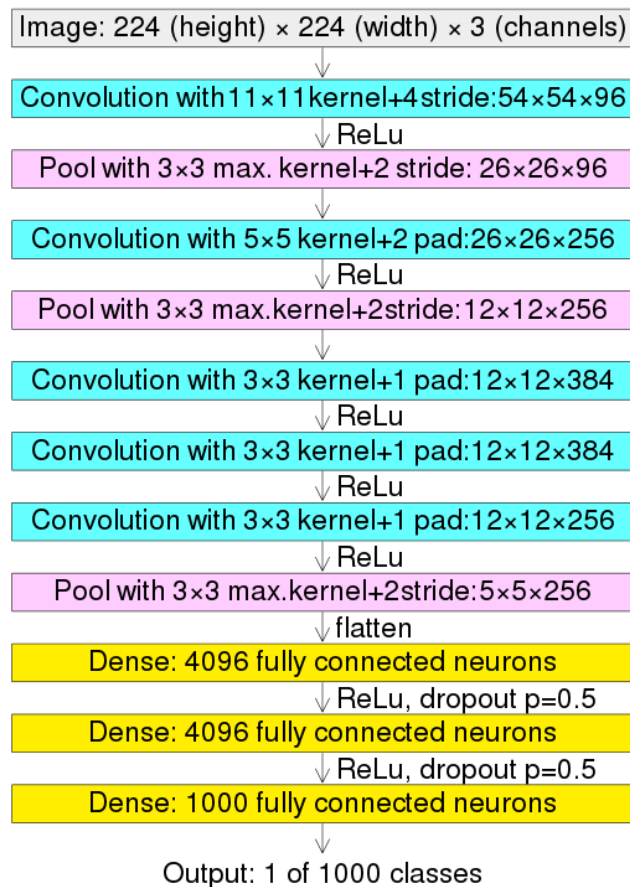


Deep Learning (DL)

LeNet



AlexNet



Deep Learning (DL)

- AlexNet compitió en **el Desafío de reconocimiento visual a gran escala de ImageNet** el 30 de septiembre de 2012. La red logró un error entre los 5 primeros del 15,3%, más de 10,8 puntos porcentuales menos que el del subcampeón.
- El resultado principal del artículo original fue que la profundidad del modelo era esencial para su alto rendimiento, lo cual era costoso desde el punto de vista computacional, pero factible gracias a la utilización de unidades de procesamiento de gráficos (GPU) durante el entrenamiento.

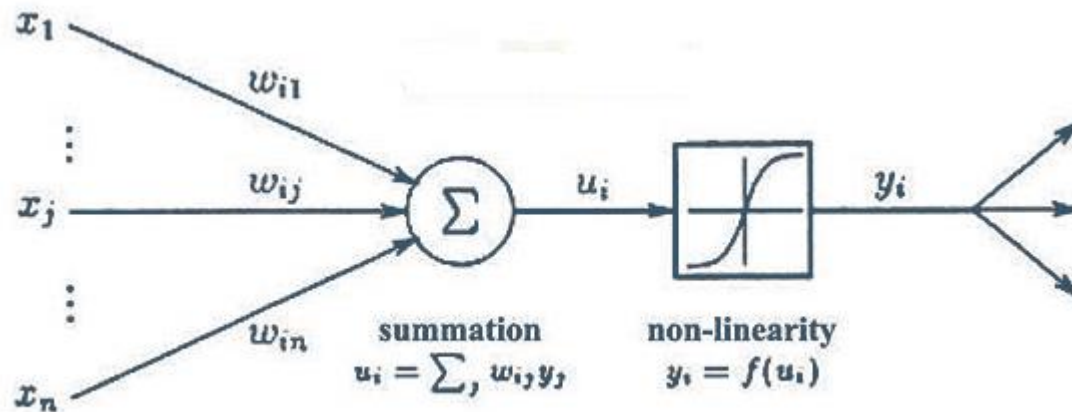
Deep Learning (DL)



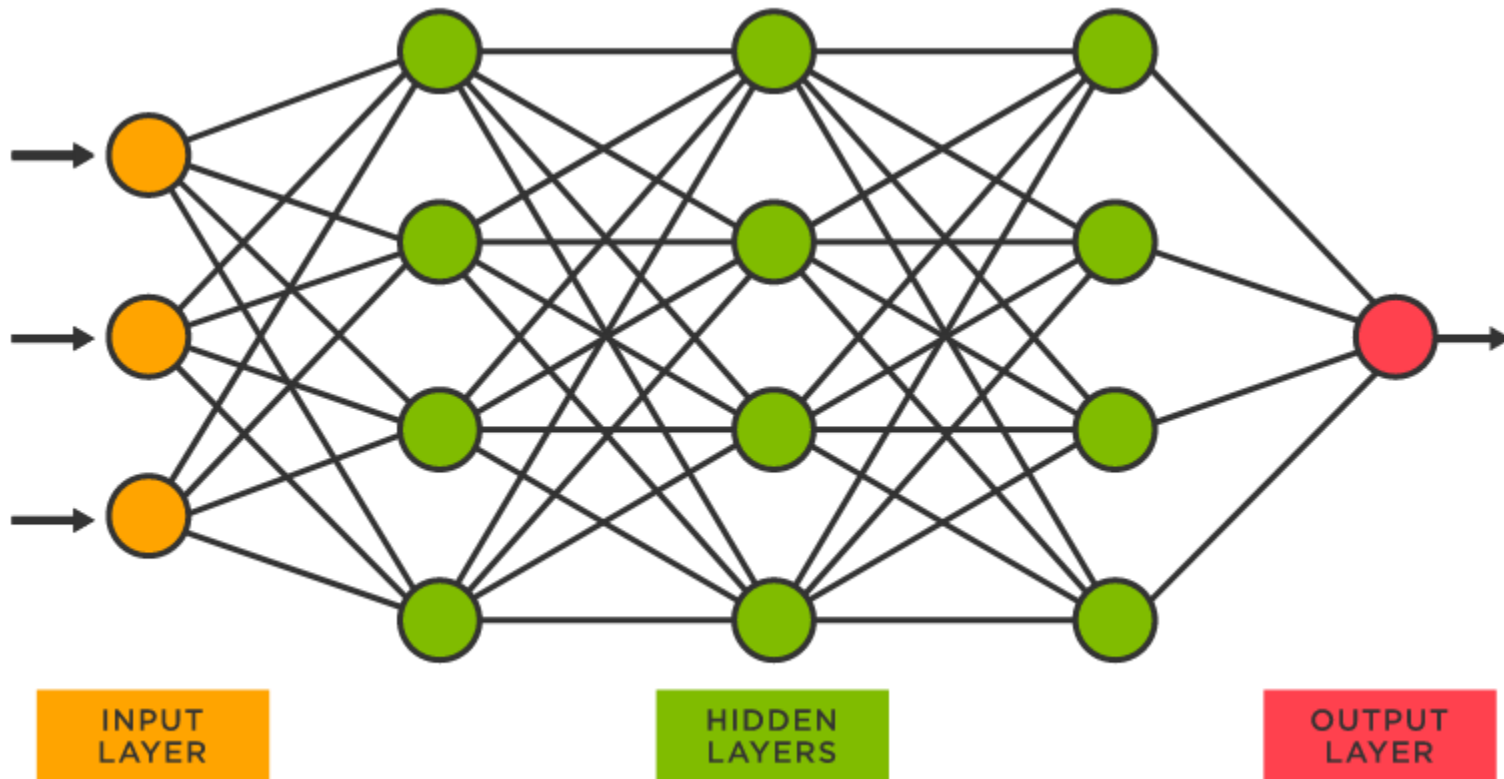
Yann LeCun (1989)

Descomponiendo DL

- En su forma básica los modelos DL son diseñados usando redes neuronales.
- Una red es una organización jerárquica de neuronas con conexión a otras neuronas.
- Las neuronas pasan un mensaje o señal a otras neuronas basadas en lo que reciben, y así aprenden con algún mecanismo de feedback



Descomponiendo DL



Problemas clásicos resueltos con DL

- Sugerencias de etiquetado de rostros.
- Automóviles independientes.
- Predicción de la siguiente palabra al escribir.
- Alexa/Siri/Google Assistant.
- Reconocimiento de imágenes médicas.



Marcos de Trabajo para DL

Existen dos marcos principales de trabajo para llevar a cabo estudios con DL (en Python3). Para ello tenemos disponible, programación de bajo y alto nivel.

Marcos de Bajo Nivel: Permiten programación más específica sobre elementos de las librerías disponibles, funcionalidades propias de cada marco que permiten mayor flexibilidad, a un costo de incrementar la dificultad en la programación.

Marcos de Alto Nivel: Permiten programación más simple y general de DL, sin profundizar en los detalles de la programación que pueden o no ser accedidos, dependiendo el Marco en el que estemos trabajando.

Marcos de Trabajo para DL

Marcos de Bajo Nivel:

- Theano: Una de las primeras en ganar popularidad.
 - Torch: Para ML y DL, Mejorada por Facebook.
 - PyTorch: Desarrollado por Facebook AI Team.
- MxNet: Desarrollado por distintas entidades. Soporta multiples GPU, y provee a AWS, y Azure.
- TensorFlow: Una de las más populares, desarrollada por Google para CPU, GPU y móviles



Marcos de Alto Nivel:

- Keras: Una de las más Populares, escrita por y para Python.
 - Gluon: Interface de alto nivel para mxnet.
 - Lasagne: Interface de alto nivel para Theano.

Marcos de Trabajo para DL

Velocidad de procesamiento (muestras de datos procesadas por segundo) para diferentes modelos de ANN

Framework	AlexNet	VGG-19	ResNet-50	MobileNet	GNMTv2	NCF
TensorFlow	1422 \pm 27	66 \pm 2	200 \pm 1	216 \pm 15	9631 \pm 1.3%	4.8e6 \pm 2.9%
PyTorch	1547 \pm 316	119 \pm 1	212 \pm 2	463 \pm 17	15512 \pm 4.8%	5.4e6 \pm 3.4%

Sophia



Audrey Kathleen Ruston

Sophia

- Fue activada el 19 de abril de 2015.
- Es conocida por su aspecto y comportamiento humano en comparación con variantes robóticas anteriores.
- Tiene inteligencia artificial (IA), procesamiento de datos visuales y reconocimiento facial.
- Es capaz de contestar ciertas preguntas y tener conversaciones sencillas sobre temas predefinidos (p. ej. en el clima).
- El robot usa tecnología de reconocimiento de voz de Alphabet Inc.
- El software de inteligencia de Sophia está diseñado por SingularityNET.
- Es conceptualmente similar al programa de ordenador ELIZA, el cual fue uno de los primeros intentos en simular una conversación humana.



Sophia



<https://www.hansonrobotics.com/>

Sophia

<https://twitter.com/RealSophiaRobot>

<https://www.instagram.com/realsophiarobot/>

Regresión lineal

Para hacer una estimación del modelo de regresión lineal simple, trataremos de buscar una recta de la forma:

$$\hat{Y} = \hat{\alpha} + \hat{\beta} X = a + bX$$

Para esto utilizaremos el método de mínimos cuadrados. Este método consiste en minimizarla suma de los cuadrados de los errores:

$$\sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Regresión lineal

Hypothesis:

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Parameters:

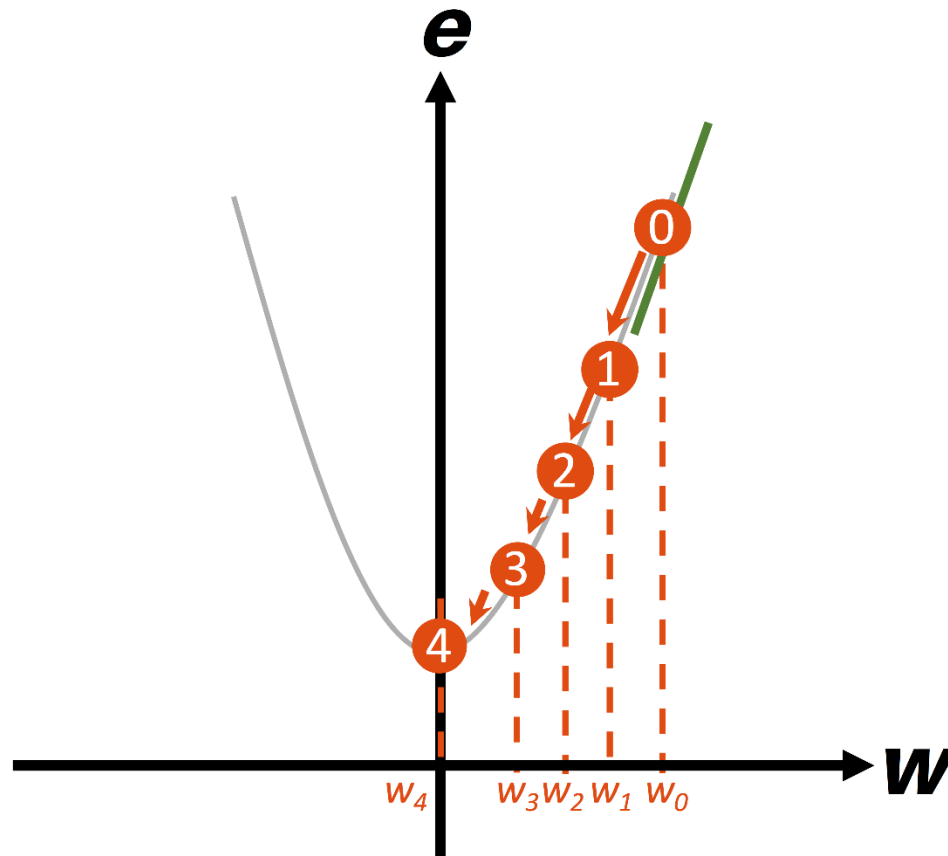
$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Goal: minimize $J(\theta_0, \theta_1)$
 θ_0, θ_1

Regresión lineal



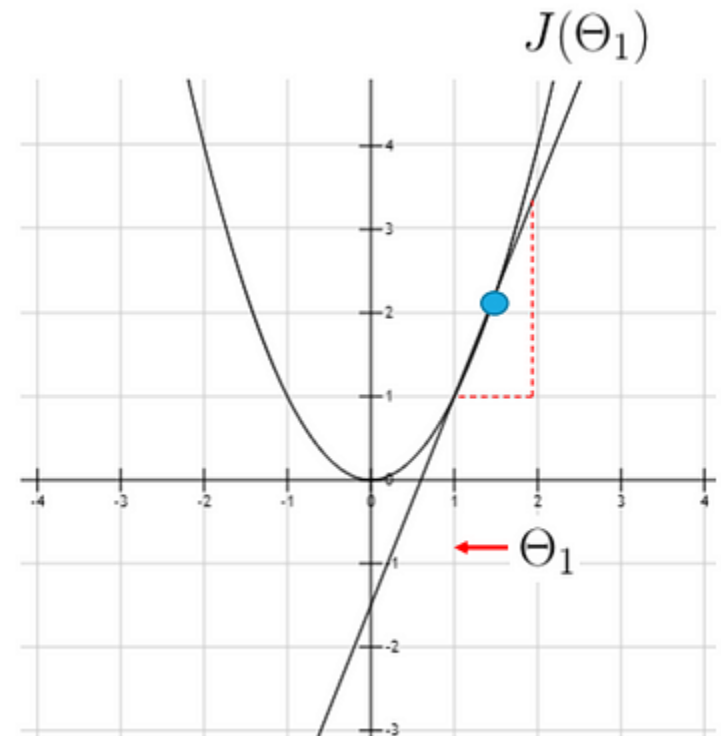
Regresión lineal

$$J(\Theta_1) \quad \Theta_1 \in \mathbb{R}$$

$$\min J(\Theta_1) := \Theta_1 - \alpha \frac{\partial}{\partial \Theta_1} J(\Theta_1)$$

Positive Slope $\frac{\partial}{\partial \Theta_1} J(\Theta_1) \geq 0$

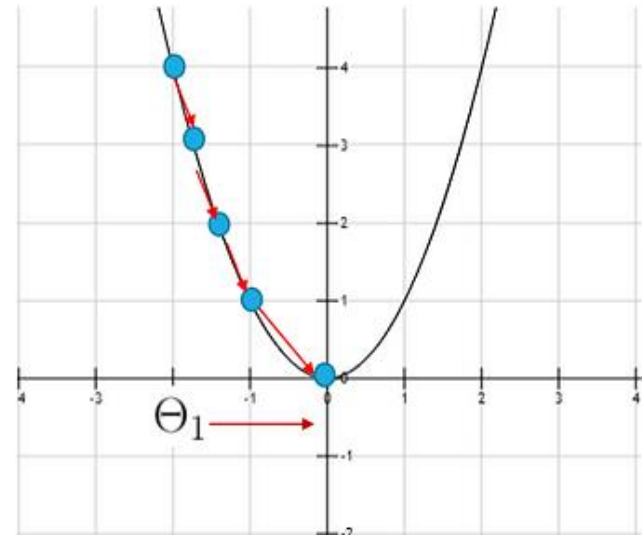
$$\Theta_1 - \alpha(\text{positive, number})$$



Regresión lineal

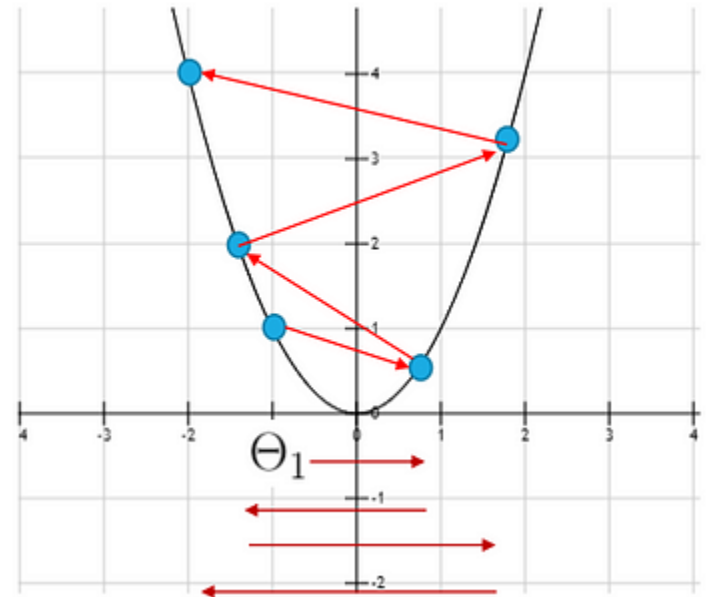
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

If α is too small, gradient descent can be slow.



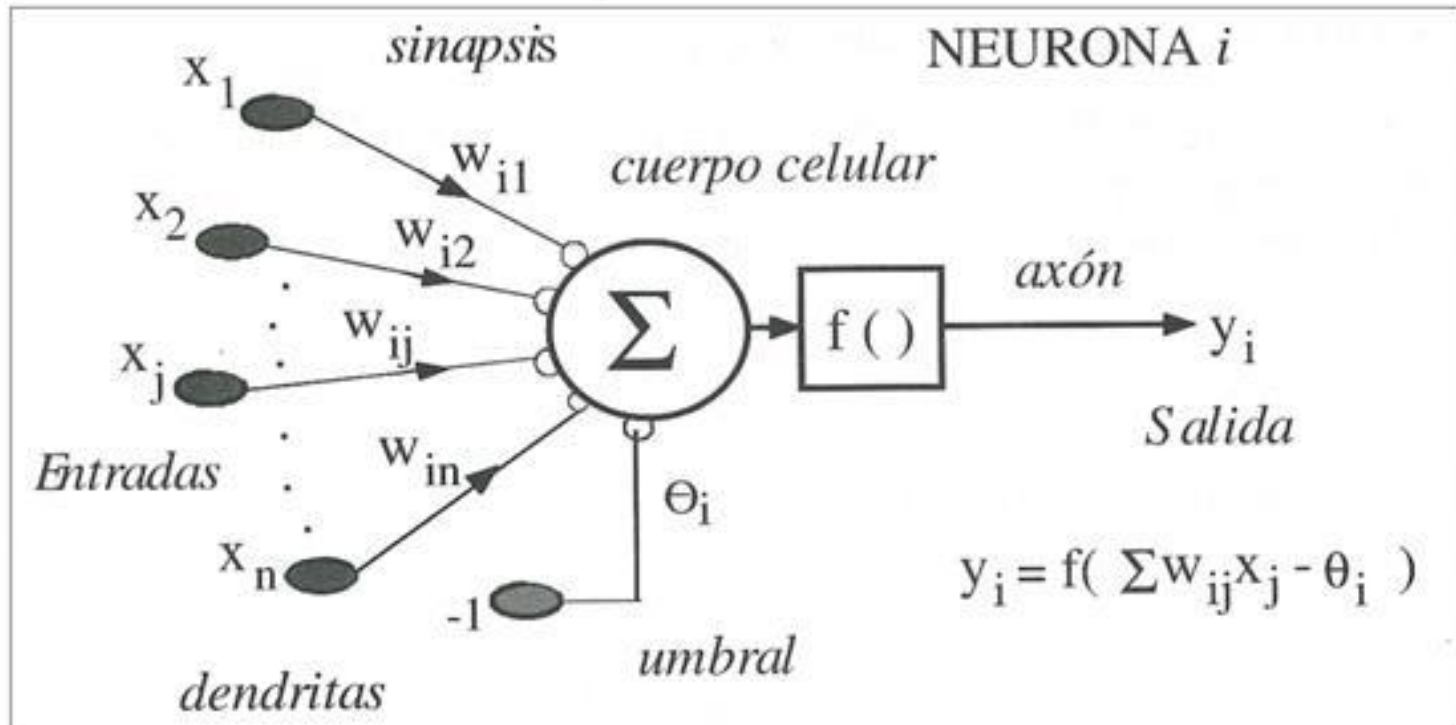
Regresión lineal

If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.

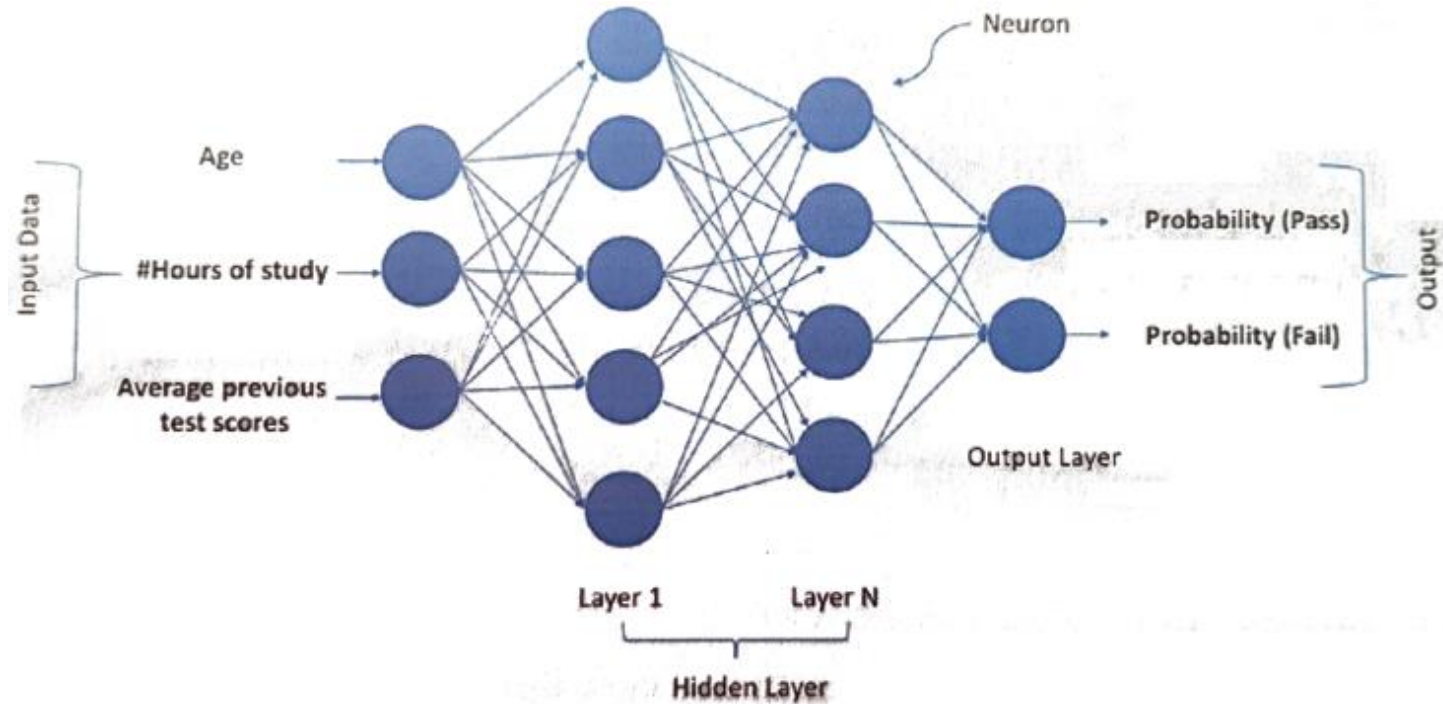


<https://medium.com/@aprendizaje.maq/regresi%C3%B3n-lineal-con-gradiente-descendente-c3b5ca97e27c#:~:text=Es%20una%20t%C3%A9cnica%20que%20permite,que%20tom e%20el%20valor%20independiente.>

Perceptrón simple



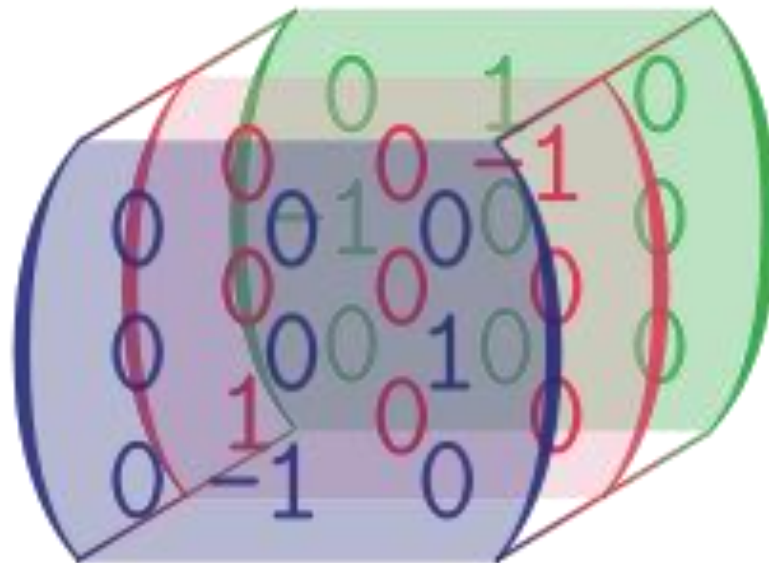
Ejemplo Keras



Ejemplo Keras

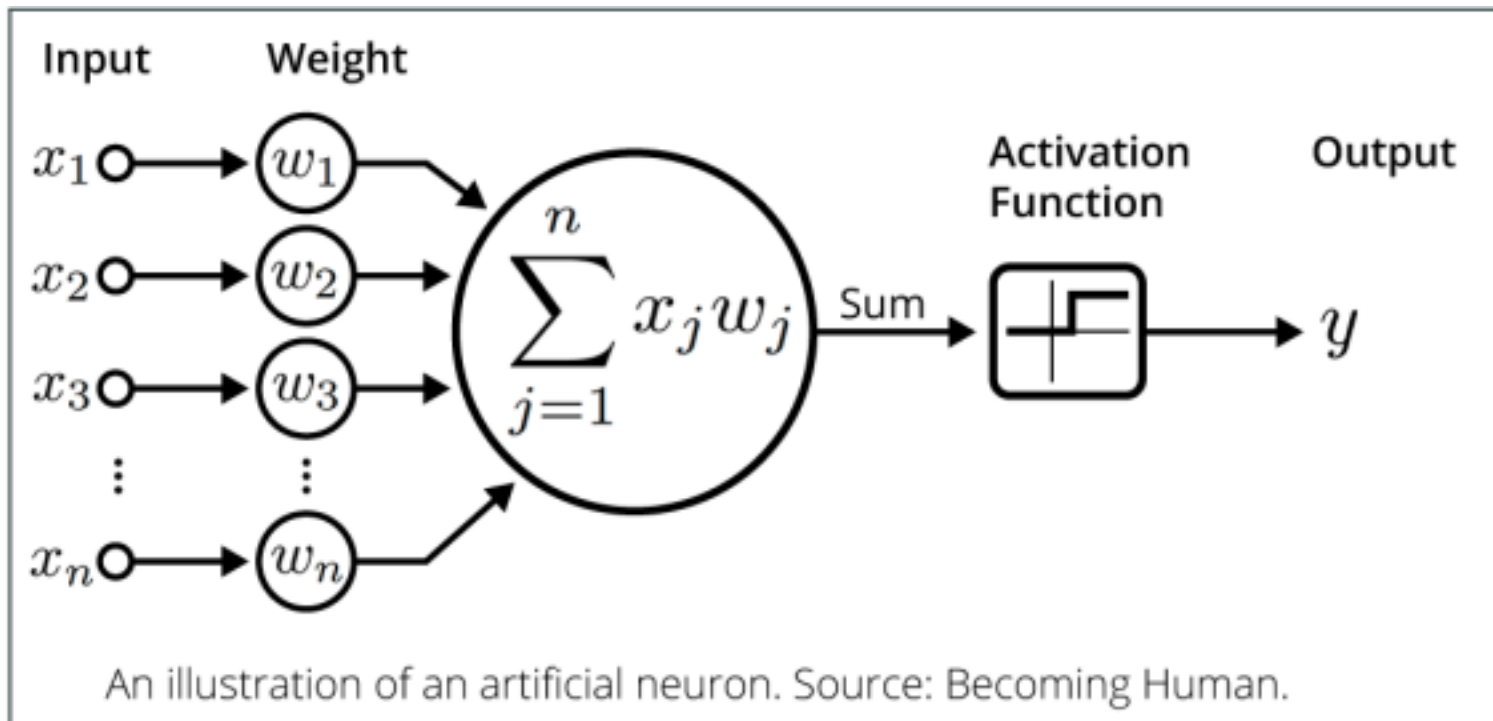
Los **datos** de entrada pueden ser de distintos tipos, en general el modelo entiende los datos como tensores. Lo que visto de forma simple, puede ser representado como una matriz n-dimensional.

$$\epsilon_{ijk} =$$



Ejemplo Keras

La **neurona** es el corazón de DNN. Una neurona recibe uno o más inputs desde capas previas (o del dato crudo si corresponde a la primera capa).



Ejemplo Keras

La **función de activación** es vital en esta parte. Uno podría pensar ¿qué ocurre si quitamos esta función? Lo primero es que el rango de salidas sería desde $-\infty$ hasta ∞ y no sería fácil definir un umbral. Además la red no aprendería, ya que en resumen la derivada de esta función sería 0 y esto impide el feedback en el aprendizaje.

