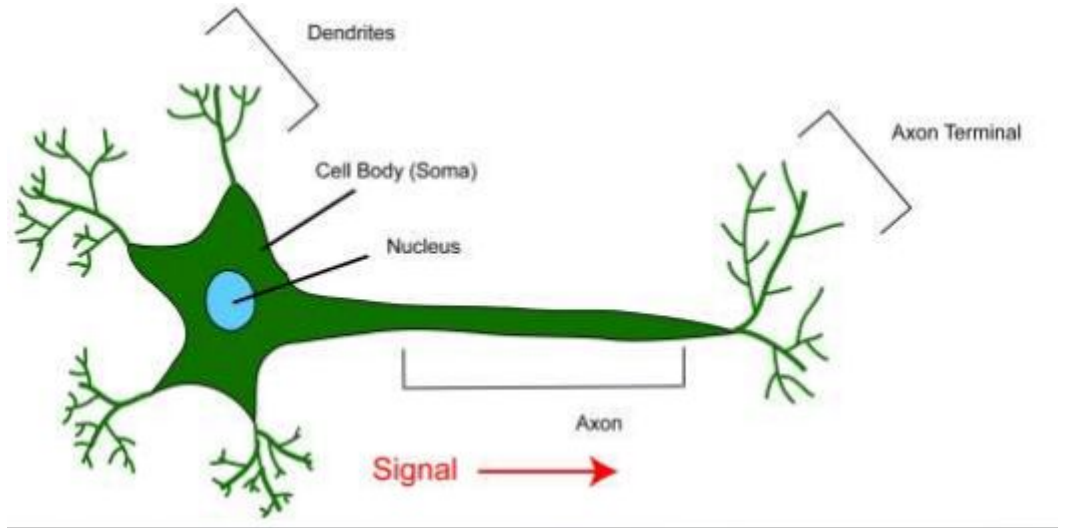

Introducción a Redes Neuronales

— Desde lo Conceptual —

¿Qué son las redes neurales artificiales?

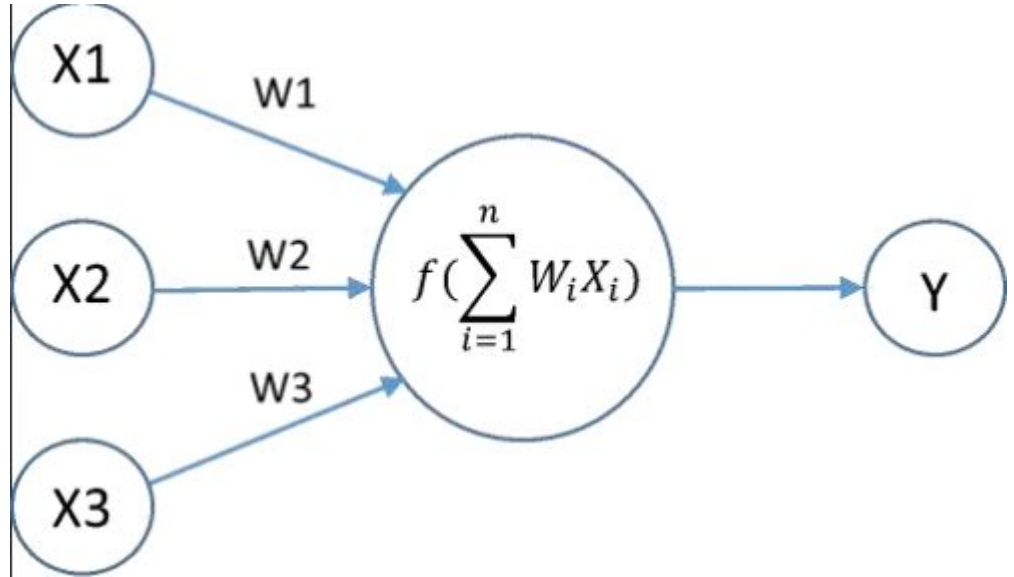
La Neurona Biológica

- Dendritas
- Cuerpo o Soma
- Axón



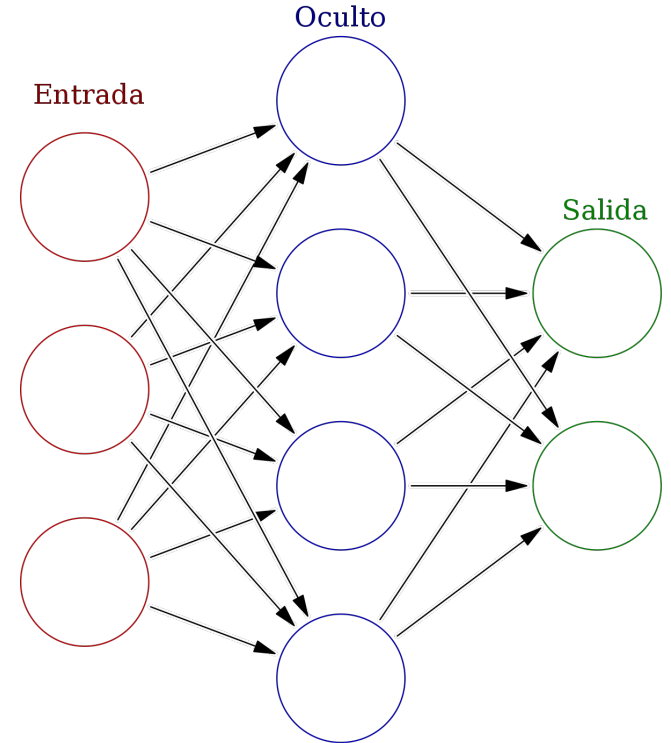
La Neurona Artificial

- Entrada
- Función de Activación
- Salida



Capas de una Red Neural

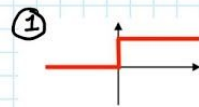
- Capas de Entrada
- Capas Ocultas
- Capas de Salida



Funciones de Activación

Commonly Used Activation Functions

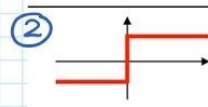
1. Step function: $f(z) = \begin{cases} 0 & z < 0 \\ 1 & z \geq 0 \end{cases}$



Range

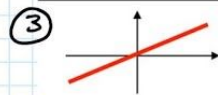
$\{0, 1\}$

2. Signum function: $f(z) = \begin{cases} -1 & z < 0 \\ 0 & z = 0 \\ 1 & z > 0 \end{cases}$



$\{-1, 1\}$

3. Linear function: $f(z) = x$



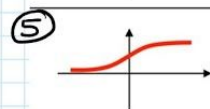
$(-\infty, \infty)$

4. ReLU function: $f(z) = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$



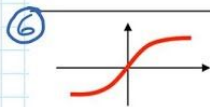
$(0, \infty)$

5. Sigmoid function: $f(z) = \frac{e^x}{1+e^x}$



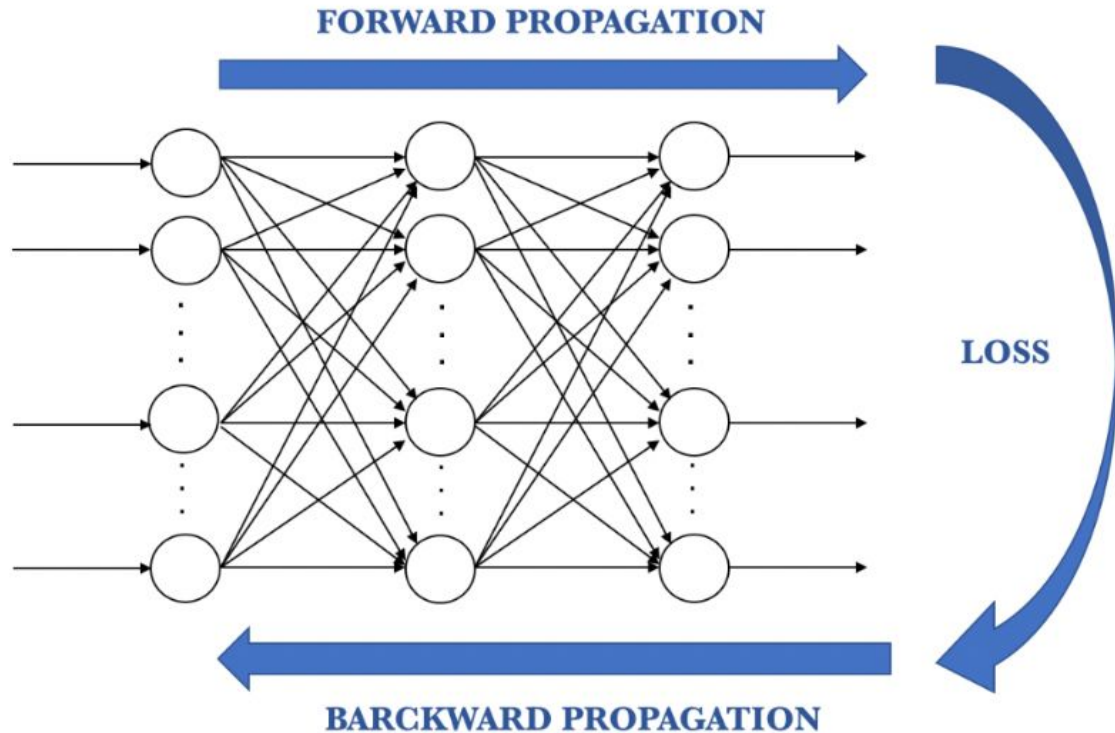
$(0, 1)$

6. Hyperbolic tan: $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

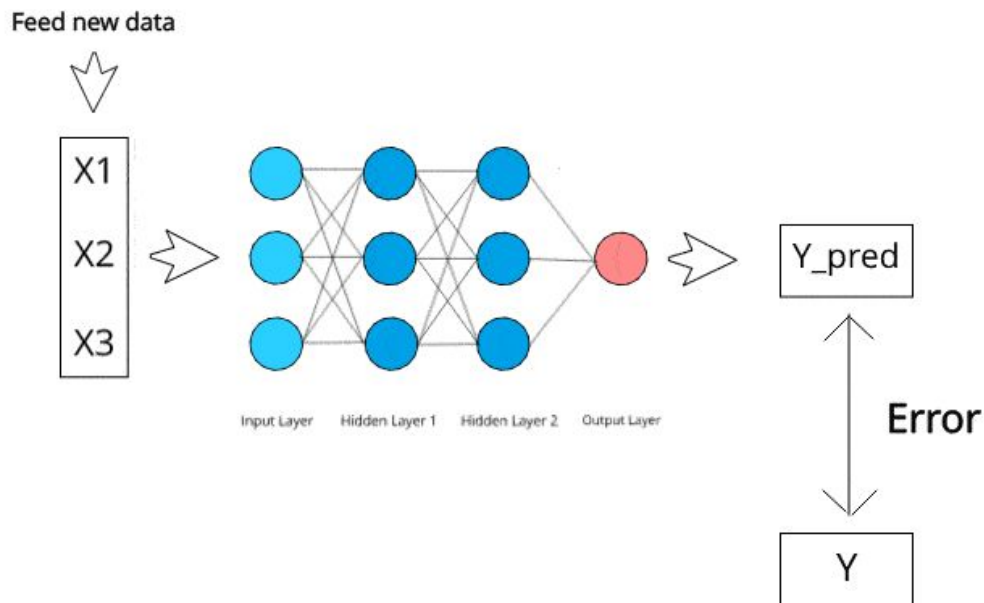


$(-1, 1)$

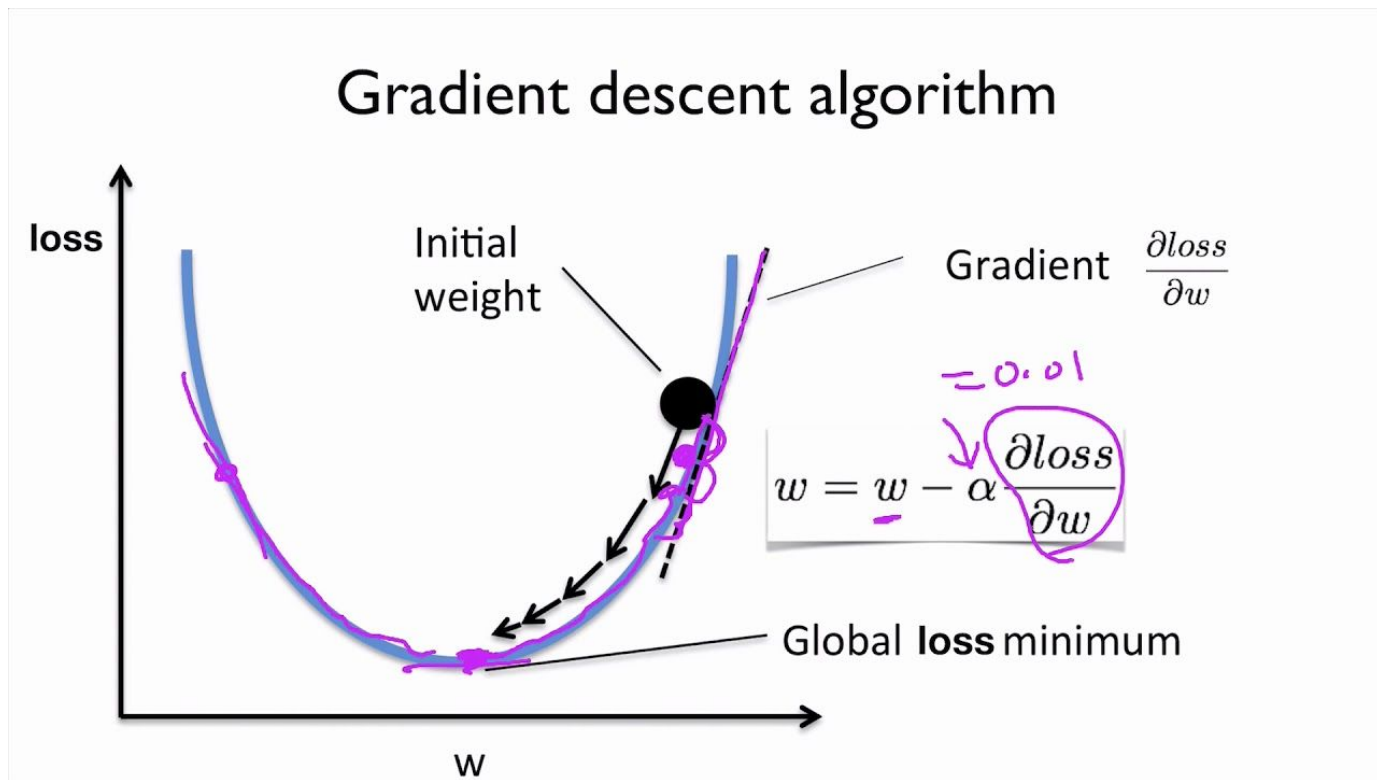
Pero... ¿cómo aprende?



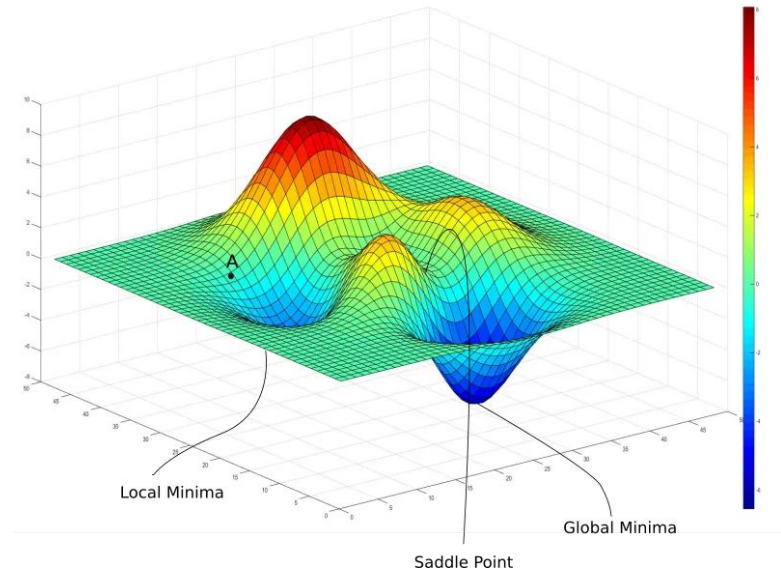
Pero... ¿cómo aprende?



Función de Costo



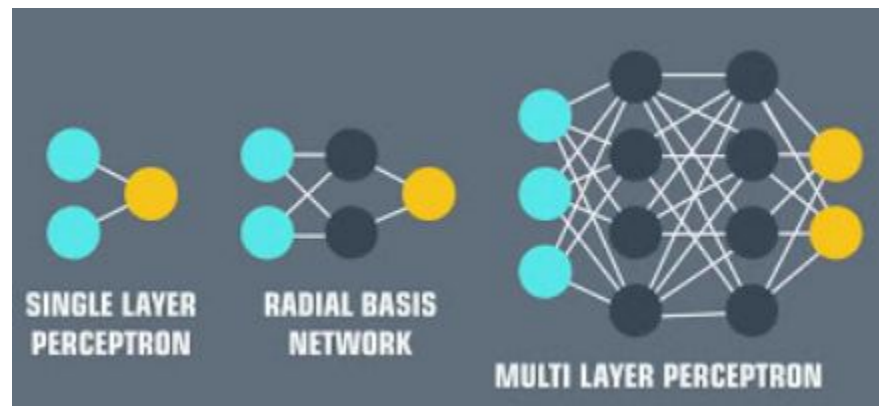
Pero... ¿cómo aprende?



Tipos de Redes Neurales

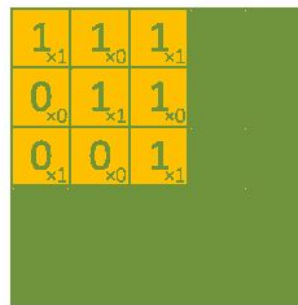
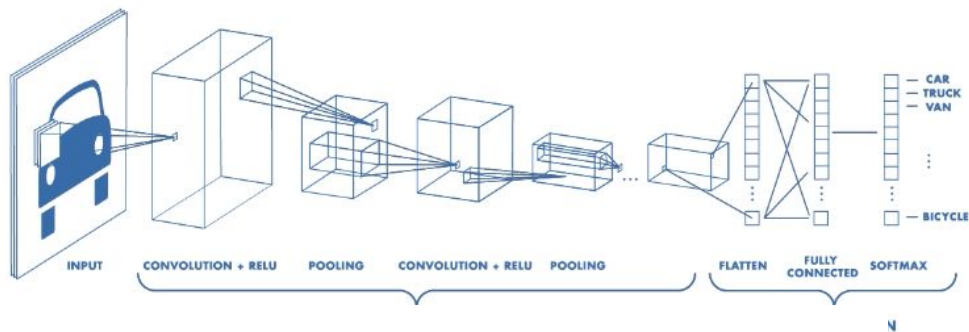
Redes Neuronales Artificiales (ANN)

- Son el tipo más sencillo.
- Involucran, en general, pocas capas ocultas, densamente conectadas.
- Pueden aplicarse a datos de tipo tabular o a problemas relativamente sencillos, como regresión y clasificación.



Redes Neuronales Convolucionales (CNN)

- Se basan en la corteza visual.
- Buscan reducir la cantidad de información a partir de la aplicación de filtros.
- Combinan capas convolucionales y capas de pooling.
- Para realizar la clasificación final, aplanan los datos utilizan capas completamente conectados.
- <https://www.cs.ryerson.ca/~aharley/vi/s/conv/flat.html>



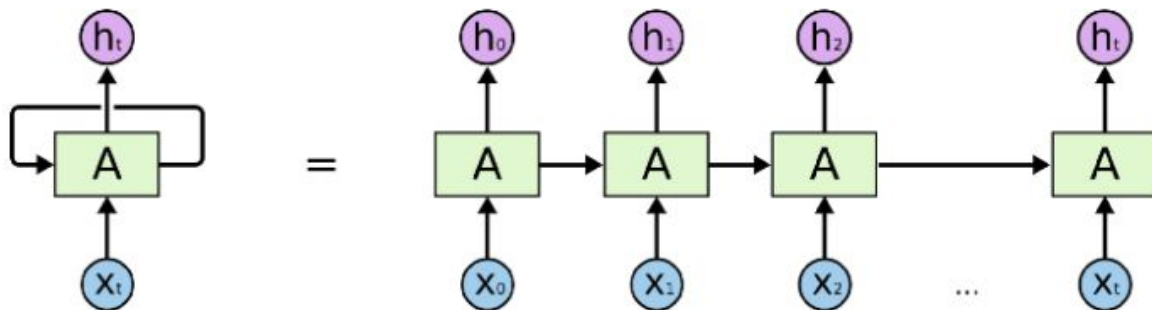
Image



Convolved
Feature

Redes Neuronales Recurrentes (RNN)

- Se basan en “mantener en memoria” los datos de los últimos N inputs, alimentando los datos como parte del input.
- Son ideales para análisis de series temporales o también de texto.
- Actualmente están siendo reemplazadas por transformers en muchos casos, pero su utilidad sigue vigente.
- Utiliza capas de Embedding y de Long-short term memory (LSTM)



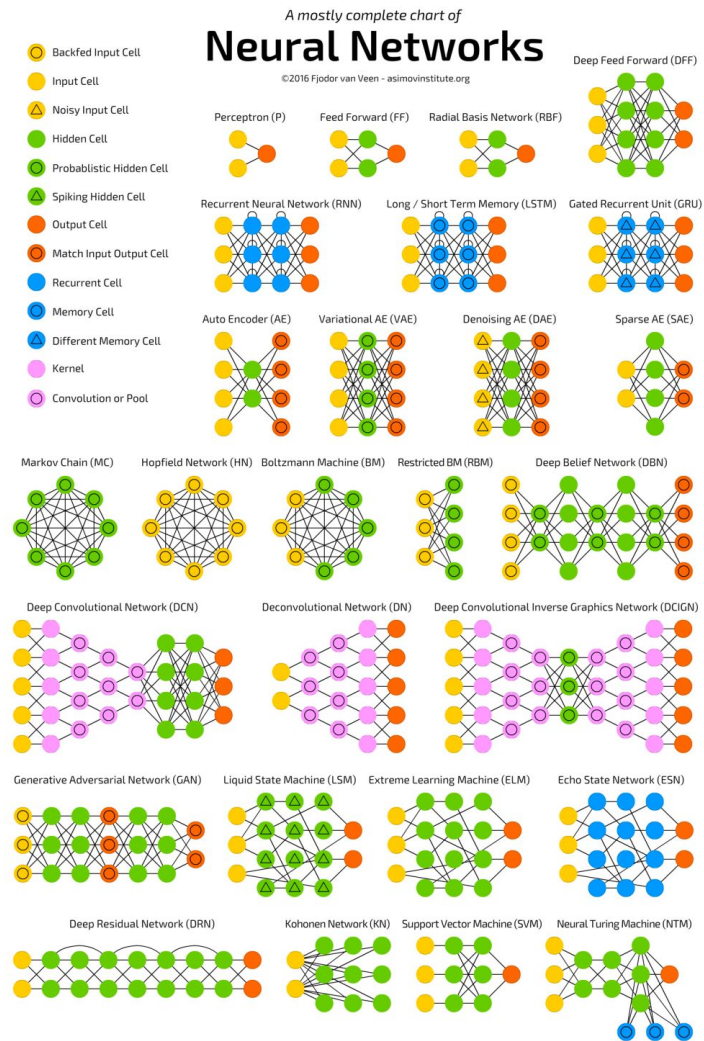
Otros tipos de aprendizajes...

- Aprendizaje por Refuerzo

- Autoencoders

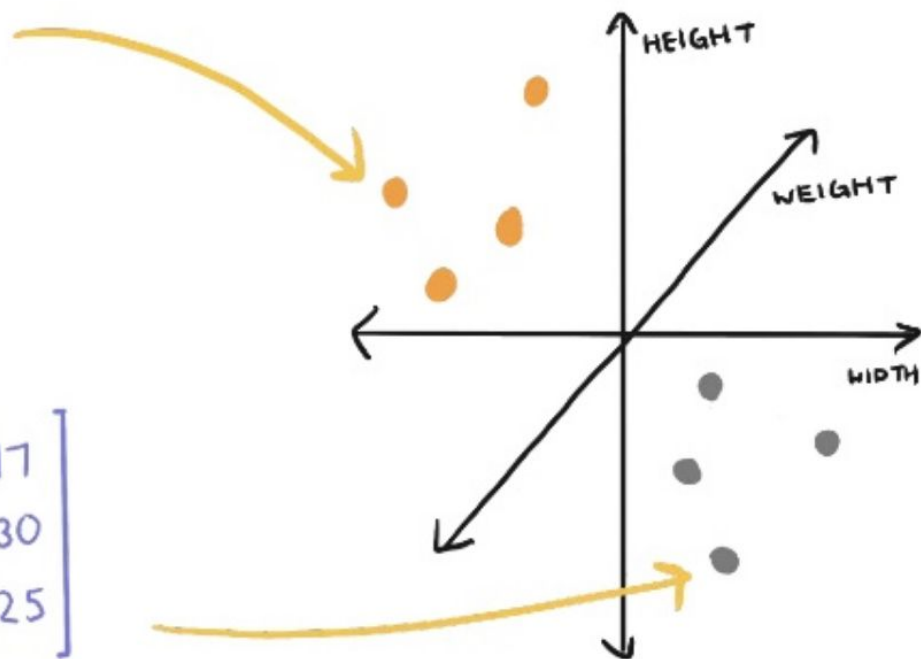
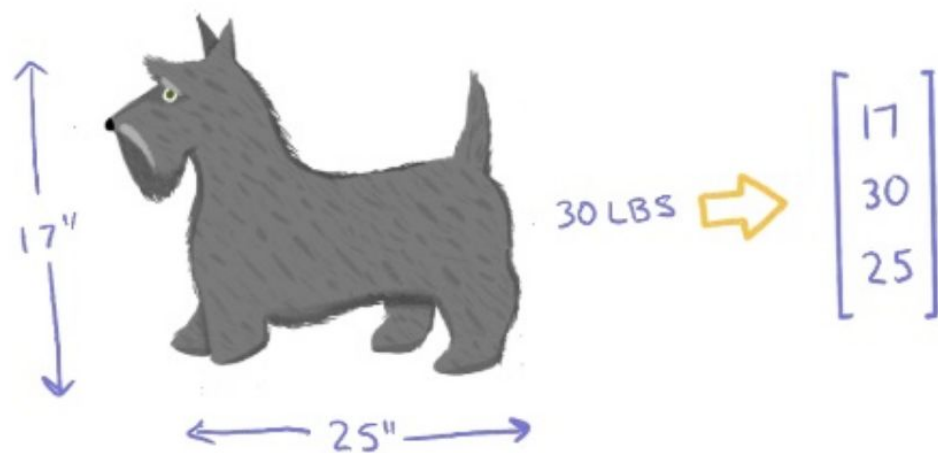
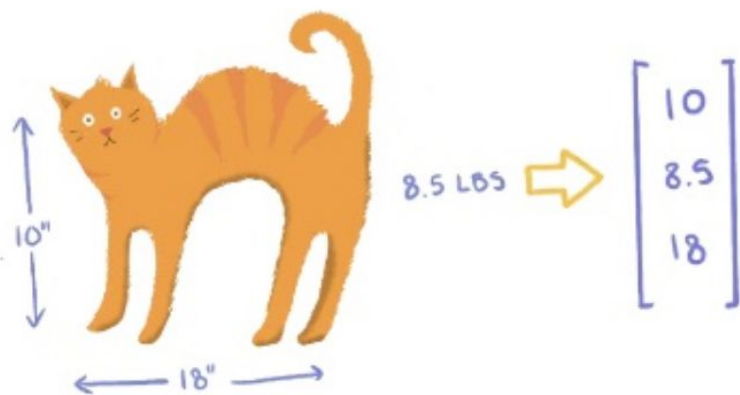
- Generative Adversarial Networks

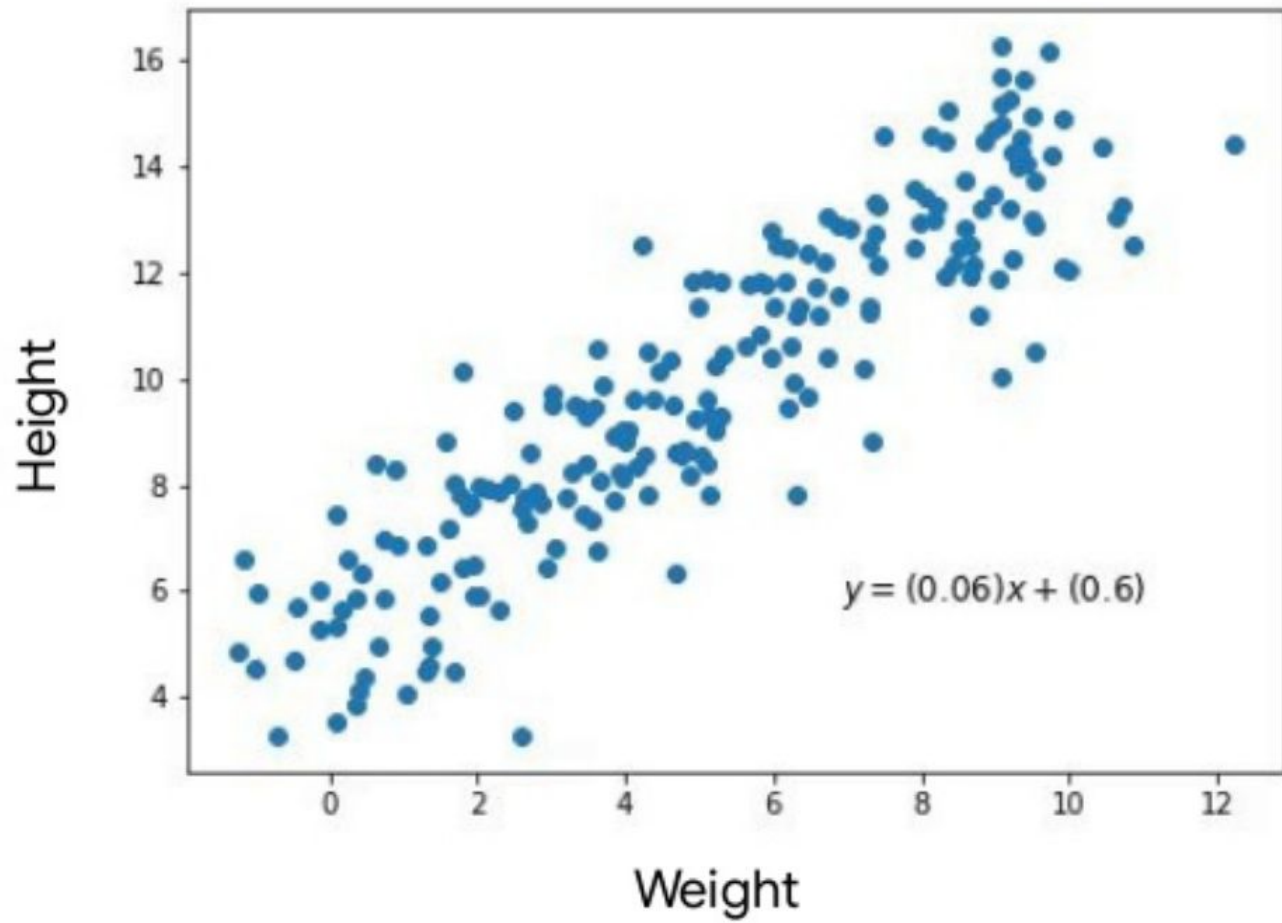
- Transformers



Introducción a Redes Neuronales

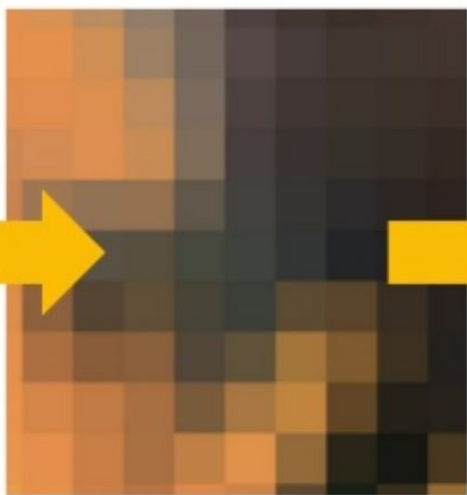
— Desde la práctica —



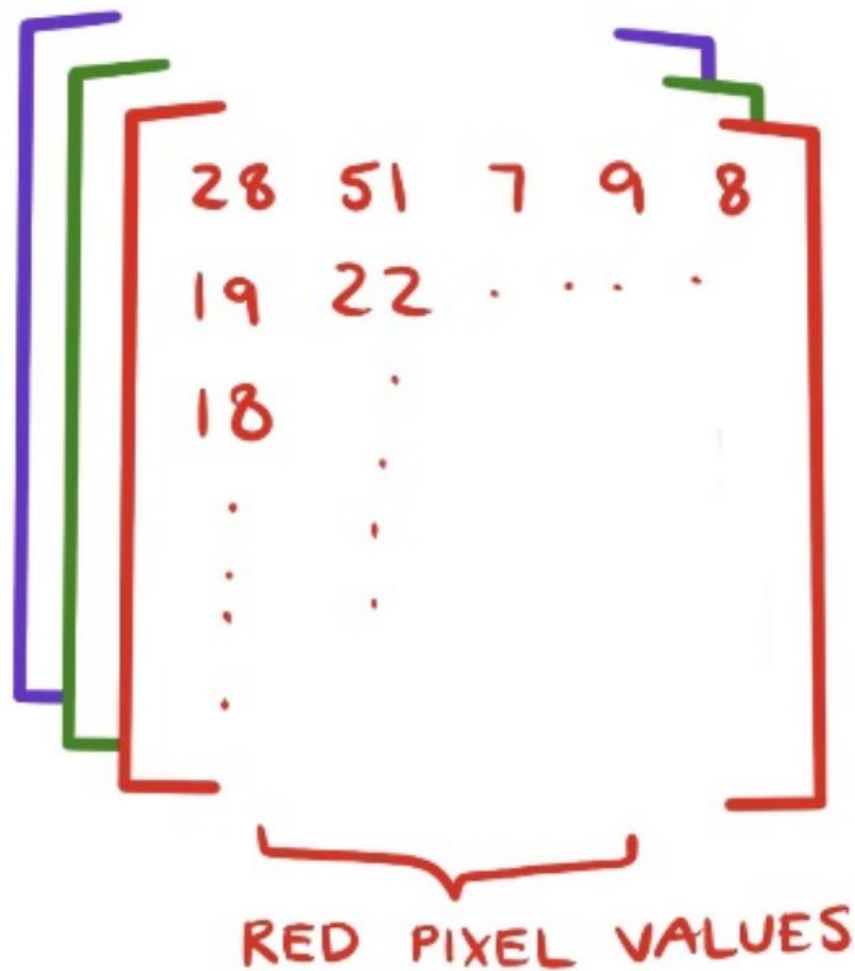








rgb(89, 133, 204)



Scalar

Vector

Matrix

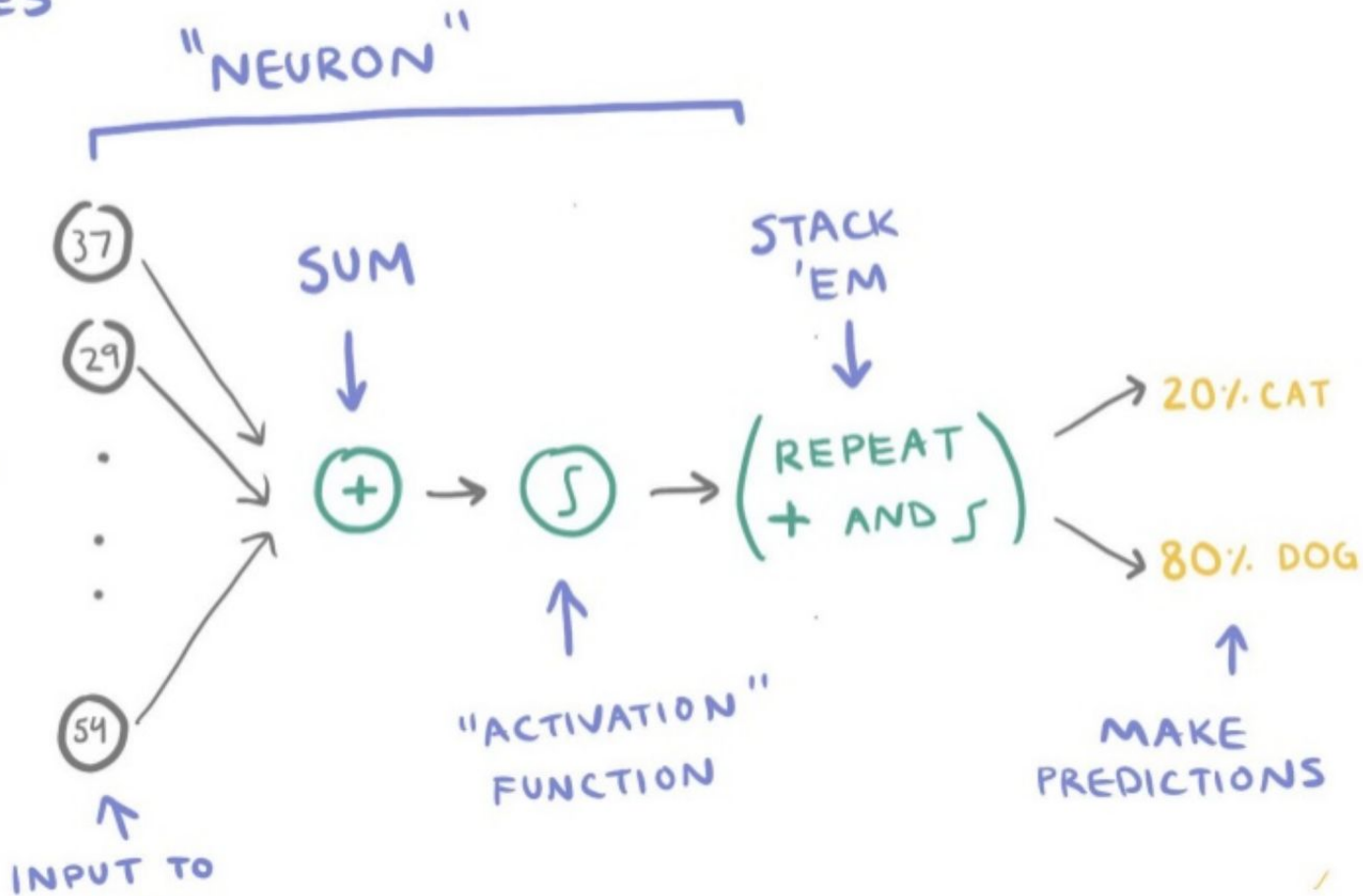
Tensor

1

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$
$$\begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$$

CAT PHOTO
PIXEL VALUES

↓
 $\begin{bmatrix} 37 \\ 29 \\ 54 \\ \vdots \\ 11 \end{bmatrix}$

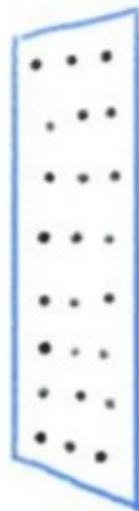
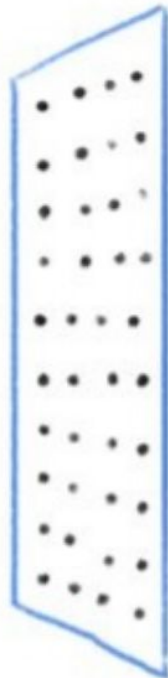
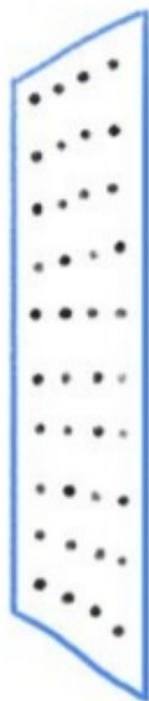


CAT



(Labeled
PHOTOS)

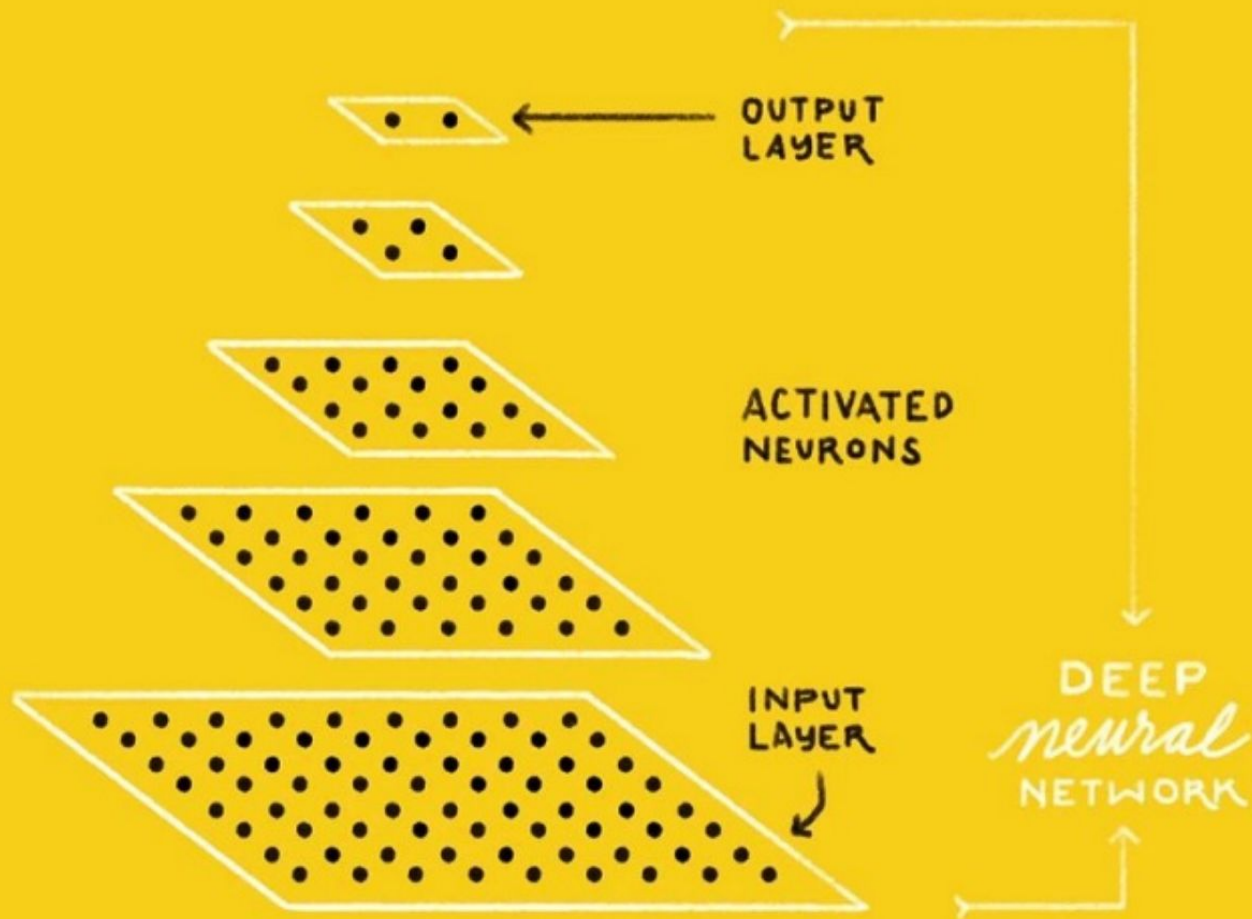
DOG

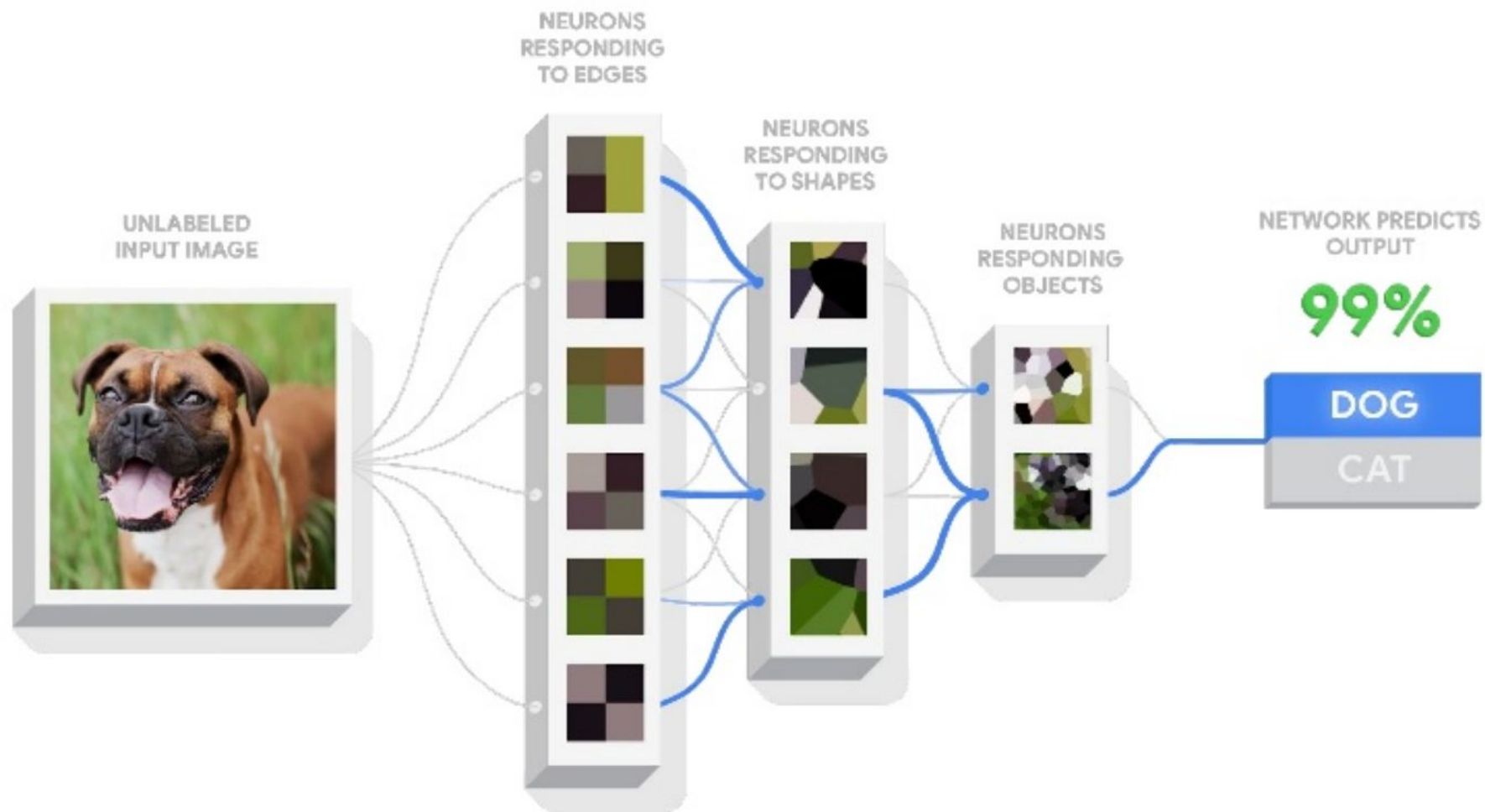


OUTPUT

CAT DOG

IS THIS A
CAT or **DOG**?





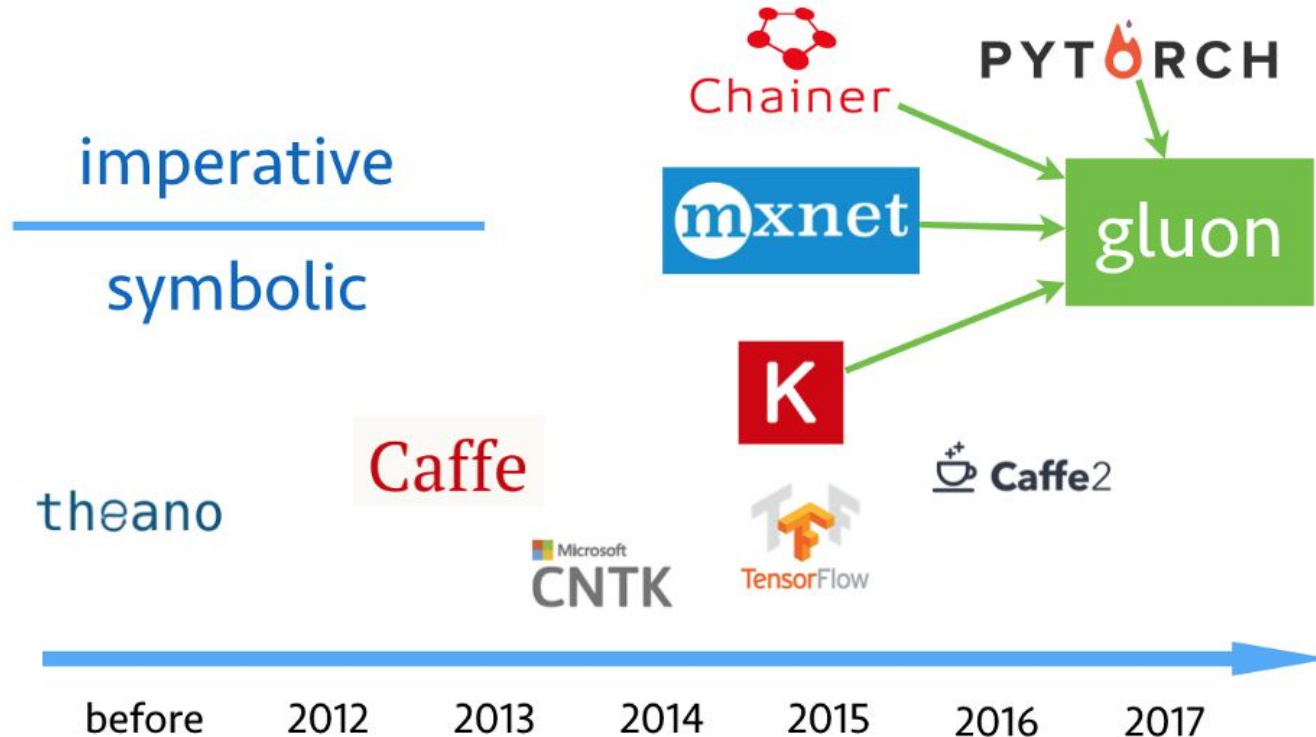
Ok... ¿cuándo usar redes neurales?

- Se tienen **enormes** cantidades de datos (10.000 ejemplos o más)
- El problema es *complejo*. - audio, video, lenguaje natural
- Los datos están no estructurados
- Se necesita el “mejor” modelo.

y cuando no usarlas...

- Tenemos **pocos** datos. (no sirven)
- Funcionan bien o suficientemente bien los métodos tradicionales de Machine Learning
- Tus datos son estructurados y posees el conocimiento de dominio necesario.
- Tu modelo debe ser explicable.

Frameworks de trabajo

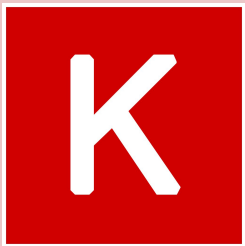


Frameworks de Trabajo

KERAS

Librería de código abierto para el desarrollo de redes neurales.

Es un wrapper (corre soportada por) Tensorflow.



TENSORFLOW

Librería de código abierto para el desarrollo de redes neurales.

Desarrollada por Google



PYTORCH

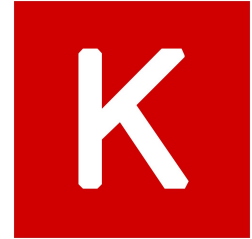
Librería de código abierto para el desarrollo de redes neurales.

Desarrollada por Facebook

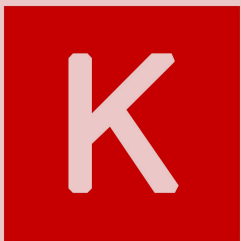


Frameworks de Trabajo

- Nivel de Lenguaje
- Velocidad
- Arquitectura
- Facilidad de Uso
- Depuración
- Cantidad de Datos
- Comunidad
- Popularidad



Nivel de Lenguaje



Keras ofrece una interacción fácil para la creación de redes neurales, con una curva de aprendizaje mucho más suave que los otros frameworks.

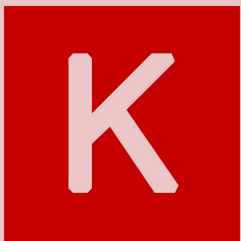


Tensorflow inicia como una interfaz de bajo nivel para el desarrollo de redes neurales, pero luego comienza a construir una interfaz de alto nivel sobre sí mismo. Finalmente “se come” a Keras.



Pytorch ofrece una interfaz de desarrollo de bajo nivel

Velocidad de ejecución



Al ser un traductor, Keras es el que sufre más en este aspecto, ya que la velocidad depende de la calidad de la traducción que haga.

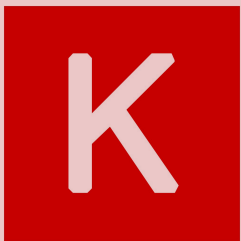


Tensorflow es uno de los primeros frameworks maduros de bajo nivel, lo que le permite ser muy eficiente computacionalmente.



La velocidad de ejecución de Pytorch esta al mismo nivel que la de Tensorflow

Arquitectura



Keras permite una construcción de la arquitectura neuronal de forma fácil y accesible.

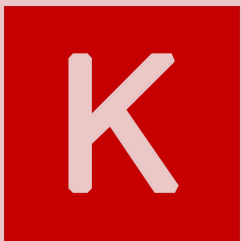


El uso del bajo nivel de tensorflow requiere un conocimiento más “duro” para la construcción de la arquitectura de una red neuronal.



En este caso la arquitectura se programa a partir de la construcción de clases, definiendo arquitecturas de forma relativamente compleja.

Facilidad de uso



El código puede ser hasta de una sola línea

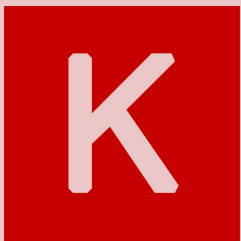


Permite un esquema de código reducido mejorando la performance computacional y la eficacia del modelo.



En general requiere una mayor cantidad de líneas de código para las mismas operaciones.

Depuración



Como los modelos suelen ser más sencillos, generalmente involucra menos procesos de depuración.

Sin embargo es complejo ya que es un wrapper

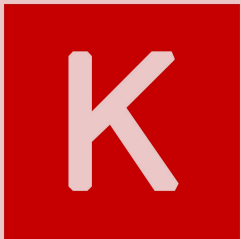


La depuración de los modelos con tensorflow es compleja, y muchas veces hay dificultades de dependencias.



La depuración es más fácil en este caso

Comunidad



Tiene una gran comunidad con un menor expertise profundo.

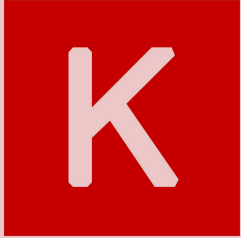


Tensorflow es la herramienta de elección, en general, por las grandes compañías, por lo que hay bastante soporte desde ese lado.



En el caso de Pytorch, el soporte es más fuerte desde el punto de vista de la comunidad de usuarios, que tienen buen conocimiento de uso y en general es bastante colaborativa.

Set de Datos



Generalmente se utiliza para datasets más pequeños, lo que no implica que no se pueda usar con datasets grandes (pero es más lento)



Se usa para modelos de alto desempeño.

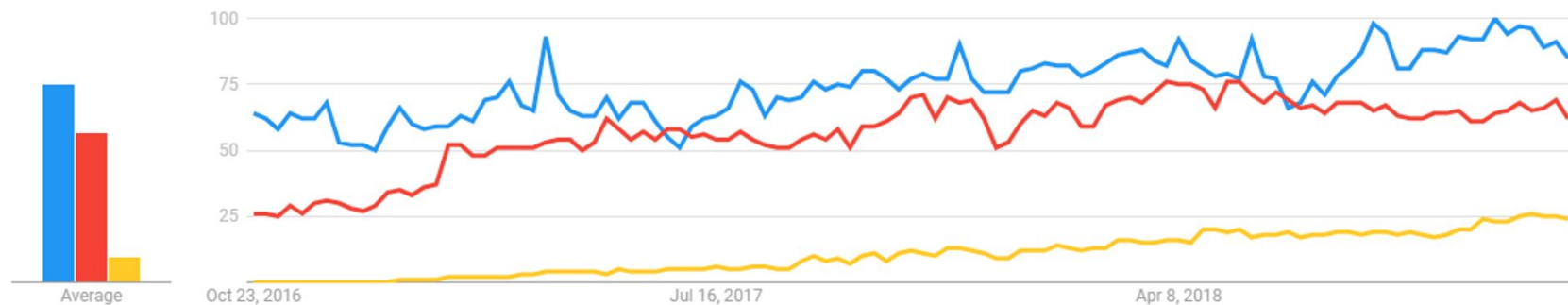


Igual que tensorflow, se utiliza para datasets de mayor volumen.

● Keras

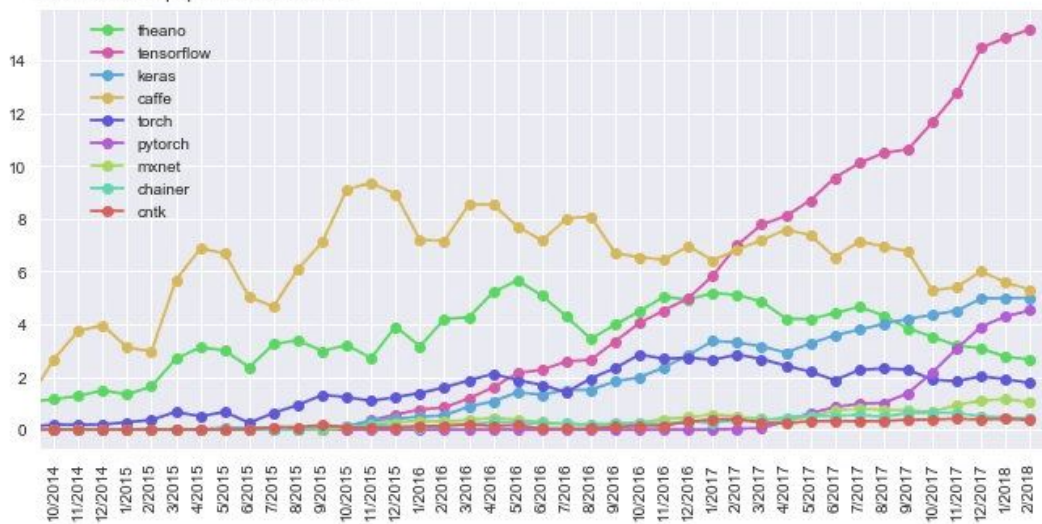
● TensorFlow

● PyTorch



Popularidad

Percent of ML papers that mention...



Conclusiones

- Los tres frameworks son igualmente capaces de desarrollar la mayoría de los casos de uso con los que nos vayamos a enfrentar.
 - Keras permite un prototipado rápido, con una eficiencia aceptable desde el punto de vista computacional, y con un mínimo tiempo de desarrollo.
 - Tensorflow y Pytorch, con enfoques bastante distintos, son ideales para la productivización, debido a que su eficiencia computacional es muchísimo más alta.
-
- Como siempre... depende del caso en el que vayamos a trabajar.