Curso de IA y Big Data

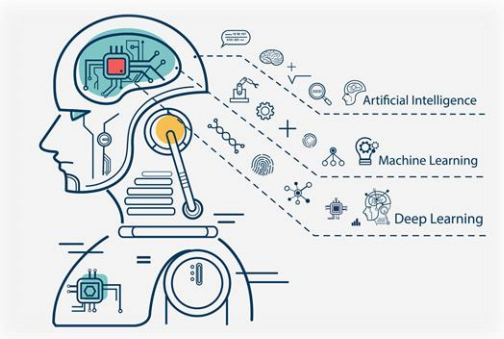
CPR Badajoz

Francisco Pajuelo Holguera

Sesión 3: Contenidos

- Deep Learning
- Redes Neuronales
- TensorFlow y Keras
- Tratamiento de imágenes

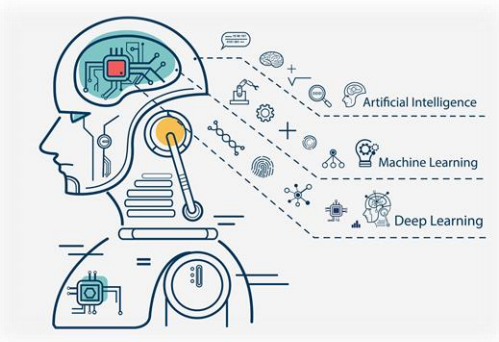
Deep Learning



El **Deep Learning** o aprendizaje profundo es un tipo de Machine Learning basado en redes neuronales por lo que se trata de aprendizaje automático no supervisado:

- Son capaces de resolver problemas que **ningún programador humano podría.**
- Son algoritmos utilizados para **problemas complejos con multitud de casos particulares.**
- Su mayor potencia es su **gran capacidad de abstracción.**

Deep Learning



Una de sus ventajas es :

Su mayor potencia es su **gran capacidad de abstracción.**

¿Por qué con redes neuronales?

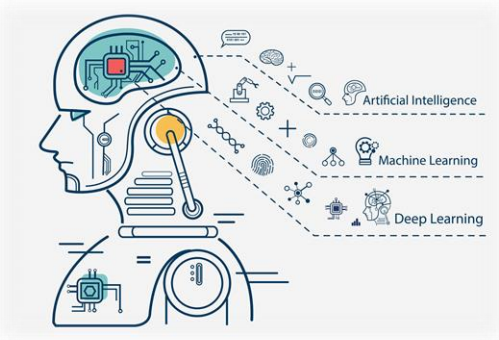
Porque cada capa añade un nuevo nivel de abstracción de la red. Pongamos un ejemplo cercano de cómo nos afecta eso:

```
section .text
        global _start:
_start:
        mov eax, 4
        mov ebx, 1
        mov ecx, string
        mov edx, length
        int 80h

        mov eax, 1
        mov ebx, 0
        int 80h

section .data
        string: db 'Hello World', 0Ah
        length: equ 13
```

hello.py

```
1   print("Hello World")
2
```

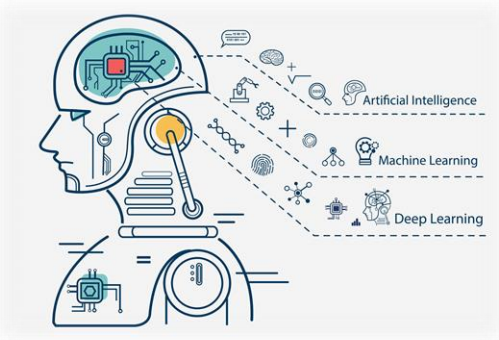
Pasos para resolver problemas con DL



Son similares al los pasos que se siguen en cualquier algoritmo de aprendizaje automático:

- **Recopilar** un conjunto de datos asociado al problema (cuantos más mejor).
- Diseñar una **función de coste apropiada** para el problema (conocida como función de pérdida o loss function)
- Seleccionar un modelo de red neuronal y establecer sus **hiperparámetros** (tamaño, características…)
- Aplicar un **algoritmo de optimización** para minimizar la función de coste ajustando los parámetros de la red (sus pesos)

Red neuronal

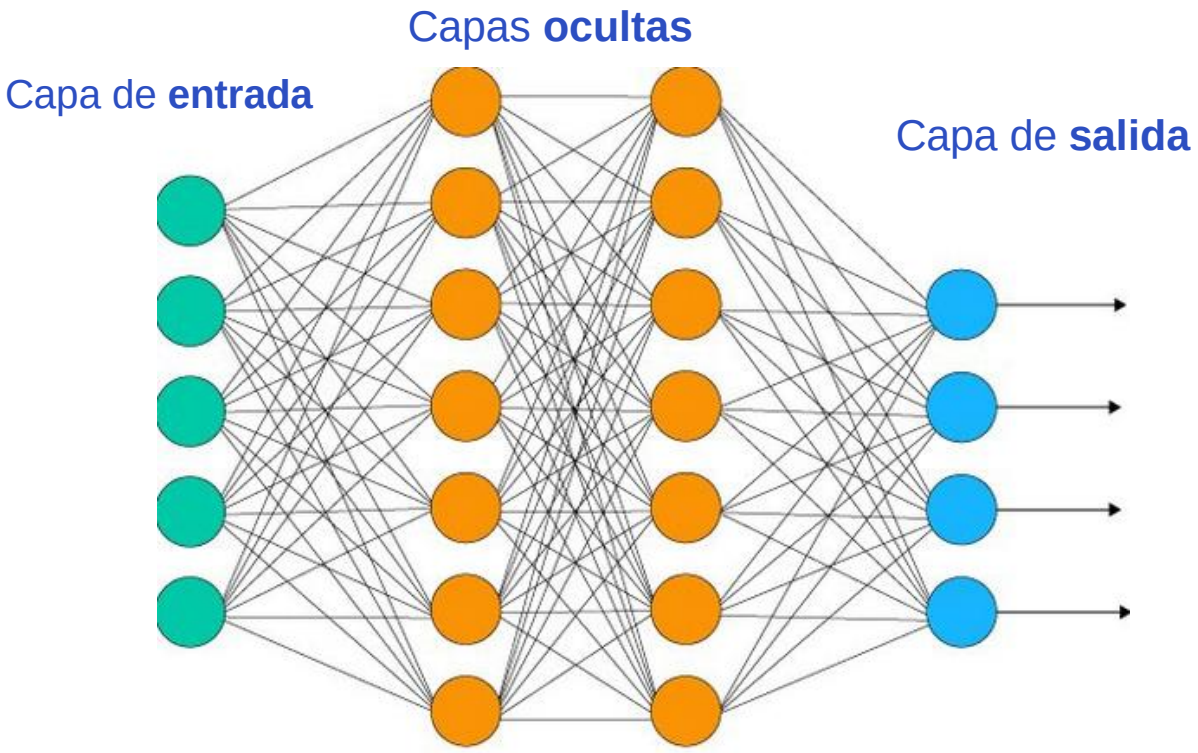


Problemas sencillos → 1 neurona

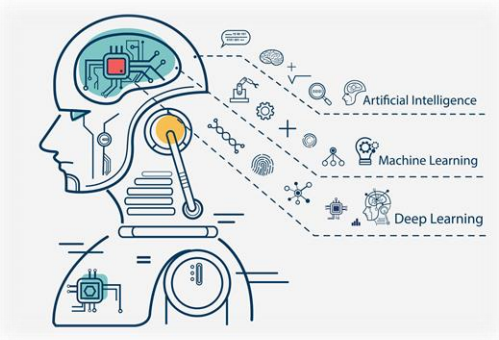
Problemas complejos → Red neuronal

Propiedades importantes:

- Organización por capas
- Puede haber distinto número de neuronas por capa
- Cantidad de capas variables en función del problema
- Todas las neuronas de una capa están conectadas con todas las neuronas de su capa anterior y posterior.



Red neuronal

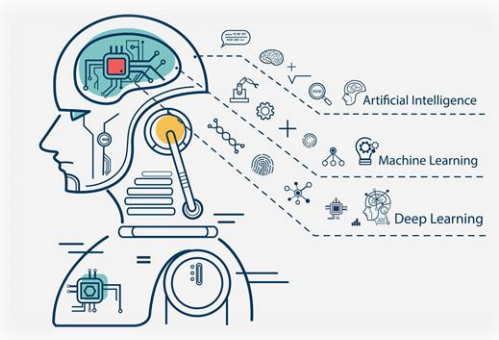


La primera capa de la red neuronal (color verde) se conoce como capa de entrada o input layer y recibe los datos en bruto, es decir, el valor de los predictores.

La capa intermedia (color naranja), conocida como capa oculta o hidden layer, recibe los valores la capa de entrada, ponderados por los pesos (flechas grises). La última capa, llamada output layer, combina los valores que salen de la capa intermedia para generar la predicción.

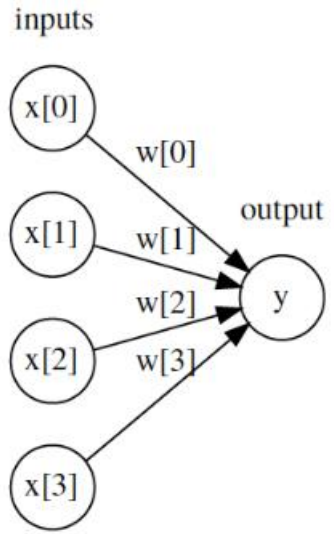
Para facilitar la comprensión de la estructura de las redes, es útil representar una red equivalente a un modelo de **regresión lineal**.

Red neuronal

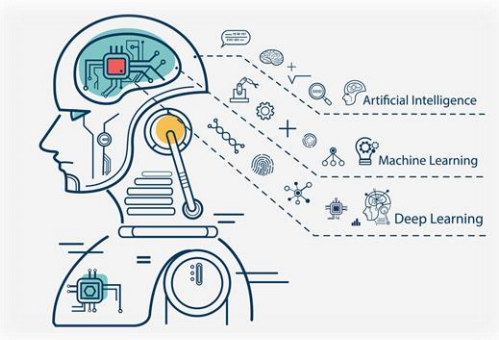


Para facilitar la comprensión de la estructura de las redes, es útil representar una red equivalente a un modelo de **regresión lineal**:

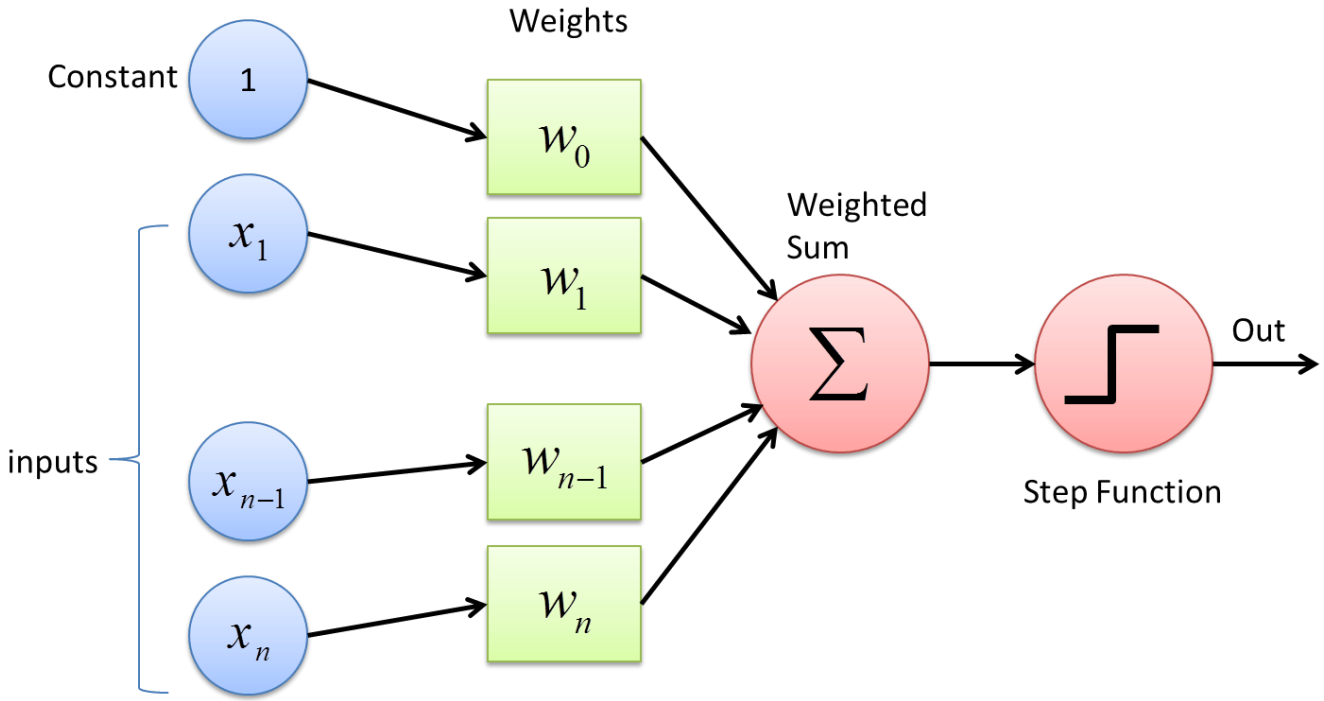
$$y = w_1 x_1 + \ldots + w_d x_d + b$$



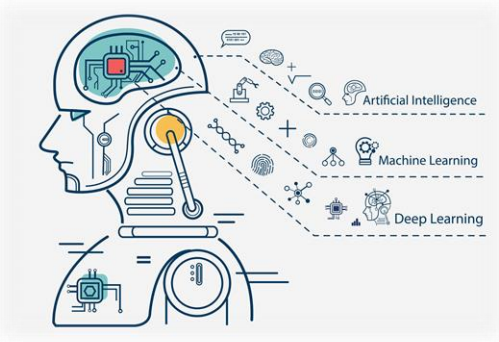
Neurona



La neurona es la unidad funcional de los modelos de redes. Dentro de cada neurona, ocurren simplemente dos operaciones: la suma ponderada de sus entradas y la aplicación de una función de activación.



Neurona



Matemáticamente:

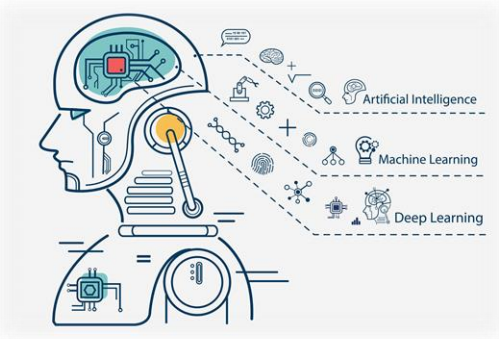
g= función de activación

b= bías

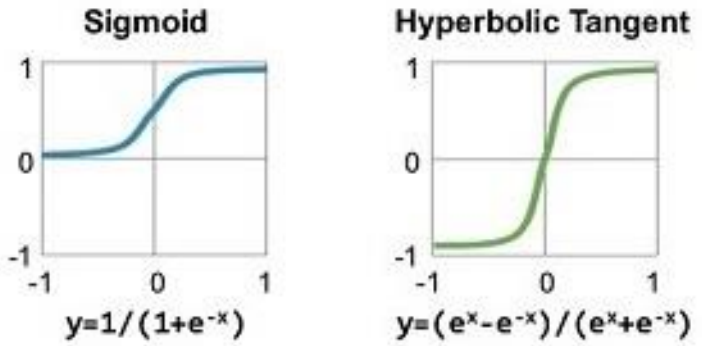
XW= factorización matricial (predictores x pesos)

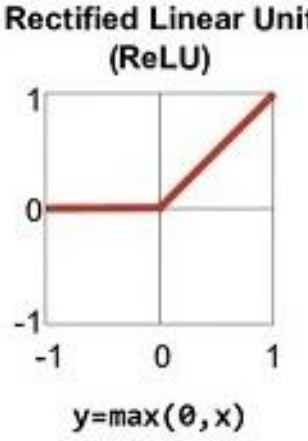
$$a = g(entrada) = g(\mathbf{XW} + b)$$

Función de activación

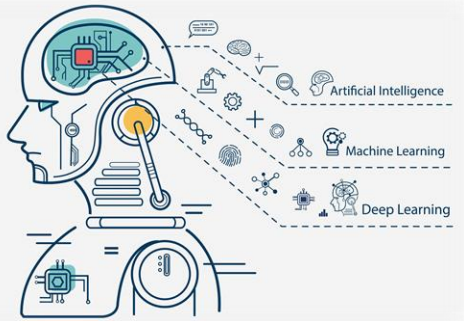


Las funciones de activación controlan en gran medida que información se propaga desde una capa a la siguiente (forward propagation). Estas funciones convierten el valor neto de entrada a la neurona, combinación de los input, pesos y bías, en un nuevo valor.





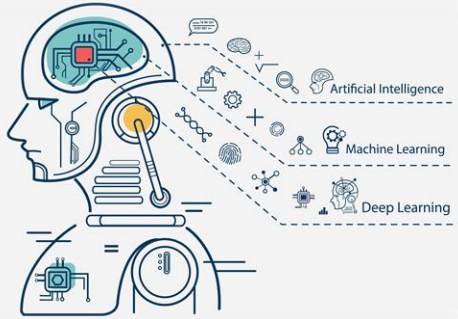
Programando una red neuronal

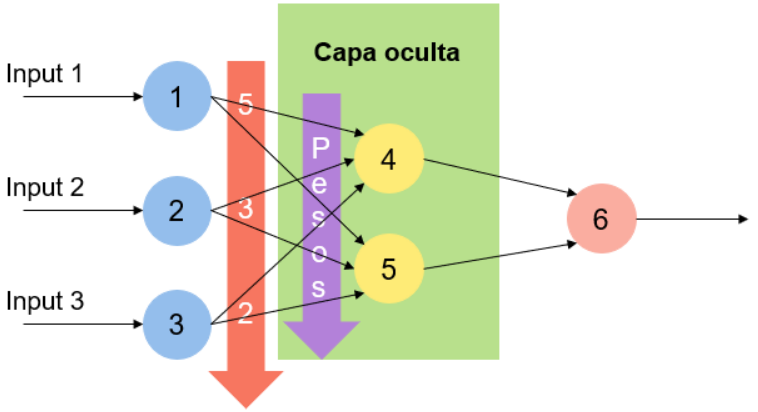


En nuestro caso supondremos que tenemos una primera capa neuronal en nuestra red que no haga ningún tipo de transformación sobre los datos de entrada. De esta forma, simplificamos su programación y simplemente los inputs que recibamos los enviaremos directamente a todas las neuronas de la primera capa oculta.

- Los inputs de la capa (outputs de la capa anterior) serán una matriz
- Otra matriz serán los pesos de todas las neuronas de la capa
- El resultado será una matriz con los resultados de cada neurona
- Aplicaremos la misma función de activación a todas las neuronas de una capa

Programando una red neuronal





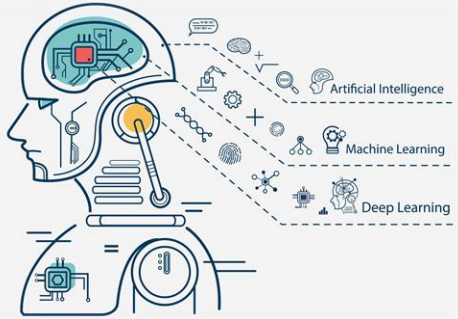
Los **Outputs** de la **capa anterior** serán **Inputs** de **nuestra capa**

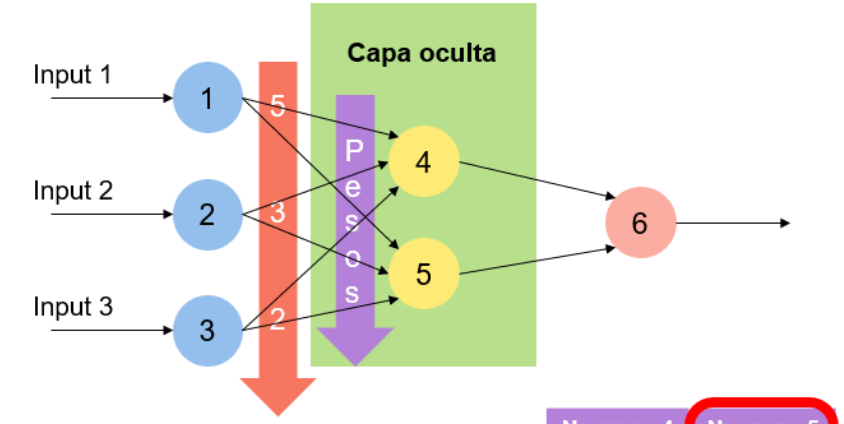
Constante	Input 1	Input 2	Input 3
1	5	3	2

Los **Pesos** de la **capa** serán una **matriz de dos dimensiones**

	Neurona 4	Neurona 5
Bias	-2	3
Peso 1	1	0
Peso 2	-2	-1
Peso 3	0	3

Programando una red neuronal





	Neurona 4	Neurona 5
Bias	-2	3
Peso 1	1	0
Peso 2	-2	-1
Peso 3	0	3

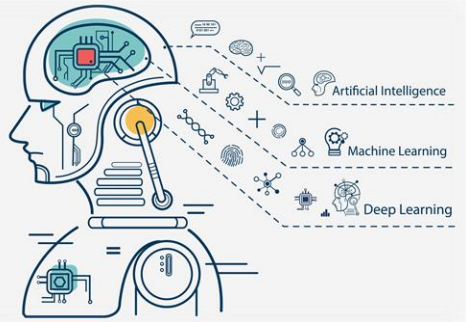
1	5	3	2

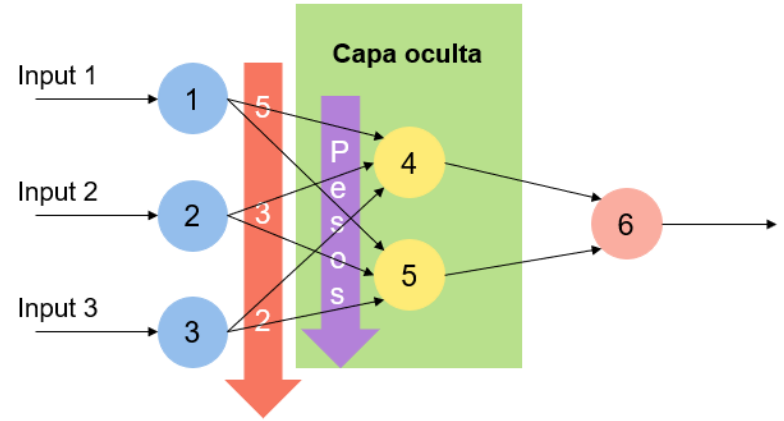
Suma ponderada
Calculada mediante una multiplicación matricial

Cálculo
1 x 3 + 5 x 0 + 3 x -1 + 2 x 3 = **6**

Resultado	
-3	6

Programando una red neuronal





Función de Activación
Aplicada a la matriz resultado en conjunto

Resultado Neurona 4	Resultado Neurona 5
-3	6

Por ejemplo: ReLU
Resultado = max(0,x)

Resultado Neurona 4	Resultado Neurona 5
0	6

TensorFlow

Es una librería de Google para el aprendizaje automático liberada como código abierto en 2015. Se puede utilizar en multitud de plataformas gracias a sus distintas adaptaciones:

- Python
- JS con Node.js
- Android (TensorFlow Lite)
- …

TensorFlow

Librería basada en dos factores clave:

- **Tensor**: hace referencia a la estructura de datos que utiliza en su interior (pudiendo almacenar tanto valores sueltos como matrices de valores). Los tensores se pueden definir mediante el uso de matrices o arrays de NumPy.
- **Flow**: hace referencia al flujo que seguirán los tensores (las capas por las que pasarán los datos, las funciones que se les aplicarán…)

Instalación Tensorflow

Install TensorFlow

1. Download and install Anaconda or the smaller Miniconda.

2. On Windows open the Start menu and open an Anaconda Command Prompt. On macOS or Linux open a terminal window. Use the default bash shell on macOS or Linux.

3. Choose a name for your TensorFlow environment, such as “tf”.

4. To install the current release of CPU-only TensorFlow, recommended for beginners:

```
conda create -n tf tensorflow
conda activate tf
```

Or, to install the current release of GPU TensorFlow on Linux or Windows:

```
conda create -n tf-gpu tensorflow-gpu
conda activate tf-gpu
```

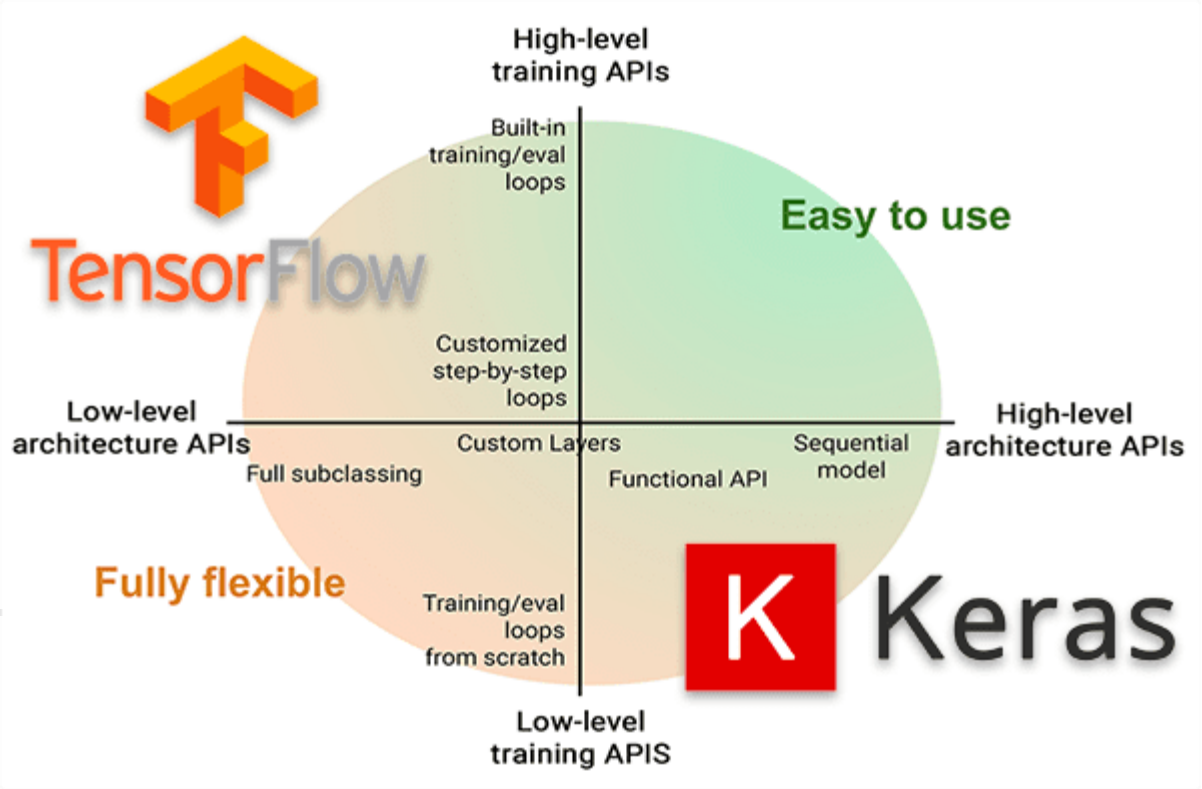
TensorFlow is now installed and ready to use.

Instalación TensorFlow

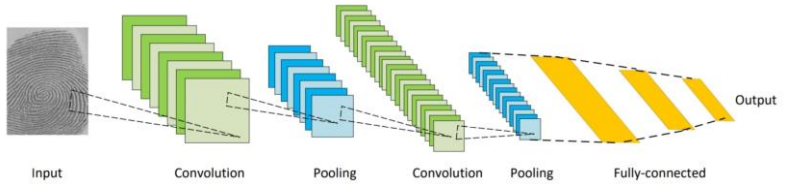
1. Abrimos Anaconda
2. Cambiamos el entorno a tf
3. Instalamos Jupyter en ese entorno
4. Si se necesitan más librerías, en entornos, podremos descargarlas seleccionando la librería y pulsando apply.
5. Ya podremos utilizar tensorflow

Keras

Es una librería que proporciona una capa superior a TensorFlow facilitando la programación de redes neuronales pero limitando a su vez la adaptabilidad de las mismas.



Redes convolucionales

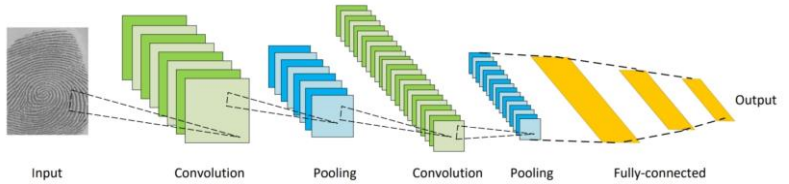


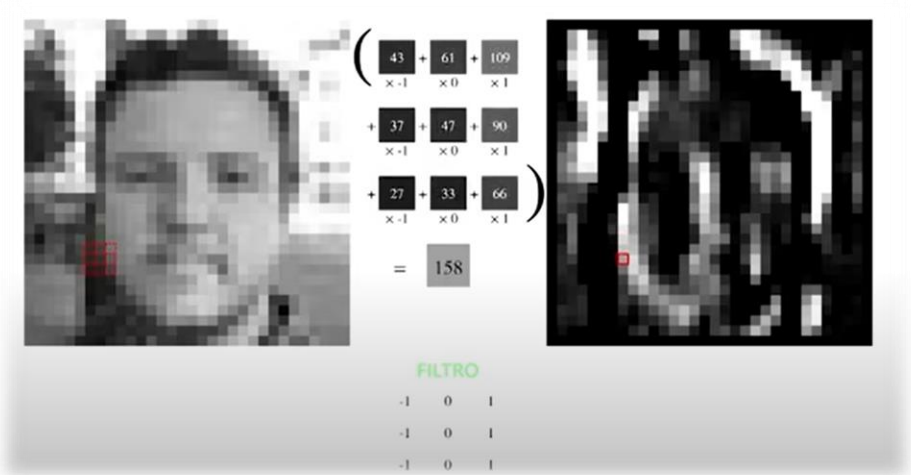
Es un tipo de red neuronal que trabaja específicamente con imágenes. Esto es porque para interpretar una imagen es muy importante la posición de cada pixel y los de su entorno.

Se basan en aplicar filtros o “máscaras” a una imagen dada como parámetro. De esta forma, cada neurona aplicaría un filtro concreto en base a sus pesos.

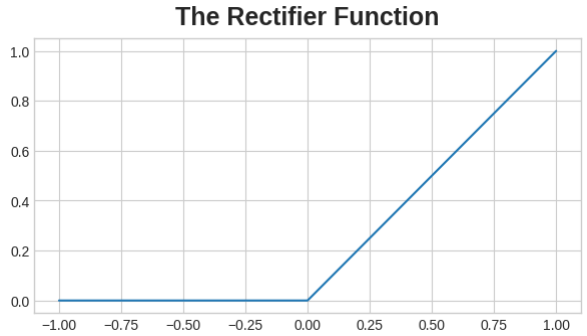
Como resultado de una capa convolucional, tendremos tantos mapas de características (resultados de aplicar la máscara a la imagen dada como input) como neuronas tenga la capa.

Redes convolucionales

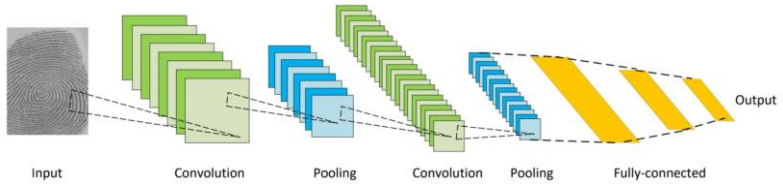


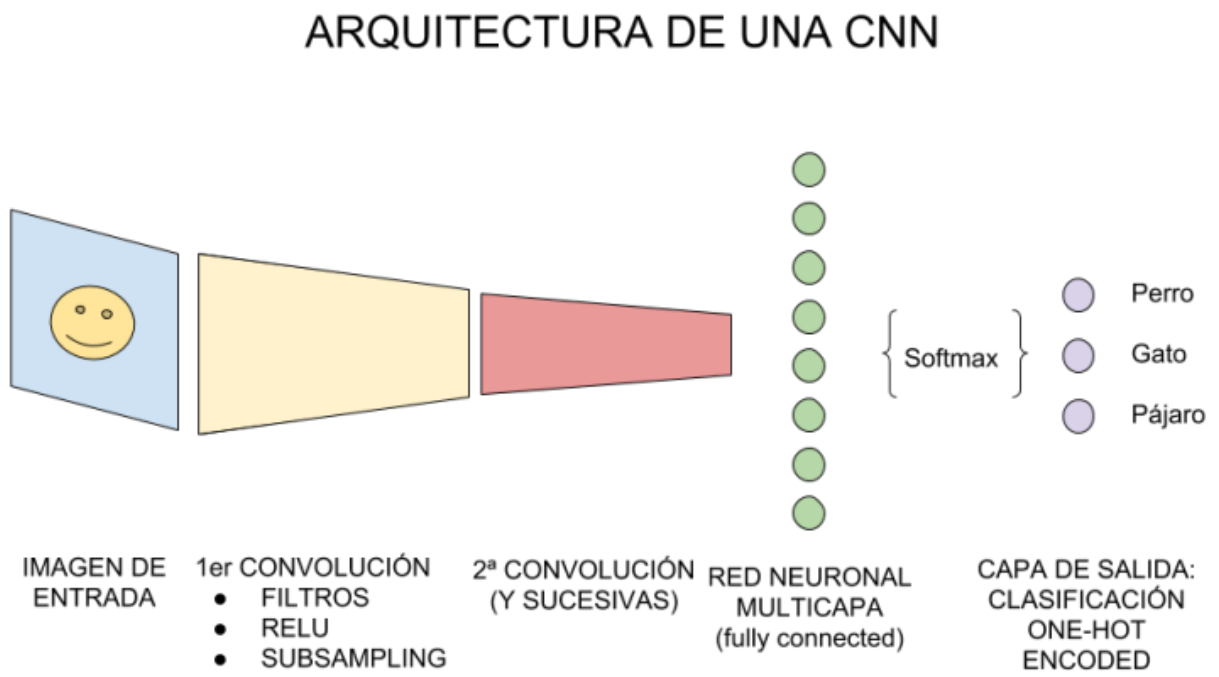


Para evitar valores negativos como resultado de una capa neuronal convolucional, lo más habitual es utilizar como función de activación la RELU:



Redes convolucionales





Sistemas de recomendación

Los sistemas de recomendación, a veces llamados en inglés “recommender systems” son algoritmos que intentan “predecir” los siguientes ítems (productos, canciones, etc.) que querrá adquirir un usuario en particular.

Sin dudas, los casos más conocidos de uso de esta tecnología son Netflix acertando en recomendar series y películas, Spotify sugiriendo canciones y artistas ó Amazon ofreciendo productos de venta cruzada <<sospechosamente>> muy tentadores para cada usuario.

Sistemas de recomendación

	🍎	🍍	🍌	🍐	🍒
Do	2	5		4	
Do		1	5		3
Do	5	3	4	3	
Do		3	5	1	1

Sistemas de recomendación

Tipos de Collaborative Filtering

- **User-based**: (Este es el que veremos a continuación) Se identifican usuarios similares. Se recomiendan nuevos ítems a otros usuarios basado en el rating dado por otros usuarios similares (que no haya valorado este usuario)
- **Item-based**: Calcular la similitud entre ítems. Encontrar los "mejores items similares" a los que un usuario no tenga evaluados y recomendárselos.

kaggle

Página por excelencia de la ciencia de datos. Podemos encontrar ejemplos de todo tipo de IA. Además de competiciones y muchos jupyter notebook resueltos.