

Visión Artificial con Redes Neuronales Profundas

2019.**3.14**

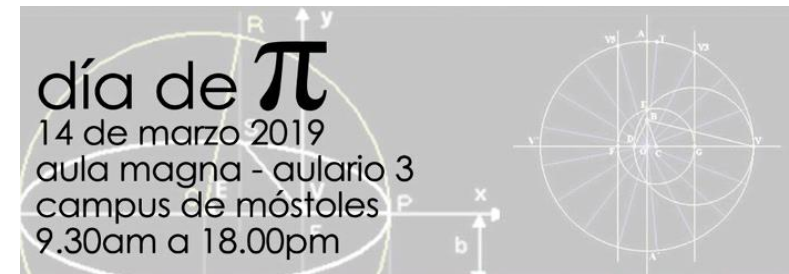
Juan José Pantrigo
Alfredo Cuesta



Universidad
Rey Juan Carlos



Escuela Técnica Superior de
Ingeniería Informática



**Master
Oficial en
Visión
Artificial**



**Computación de
Altas
Prestaciones y
Optimización**

¿Qué somos capaces de hacer?

- Detección y segmentación:

<https://www.youtube.com/watch?v=OOT3UIXZztE>

- Generación de imágenes:

<https://www.youtube.com/watch?v=JzgOfISLNjk>

- Estimación de pose 3D a partir de imágenes 2D:

<https://www.youtube.com/watch?v=cJsPnm-T9cA>

Proyecto en GitHub: https://github.com/rollervan/BCN_3DPose

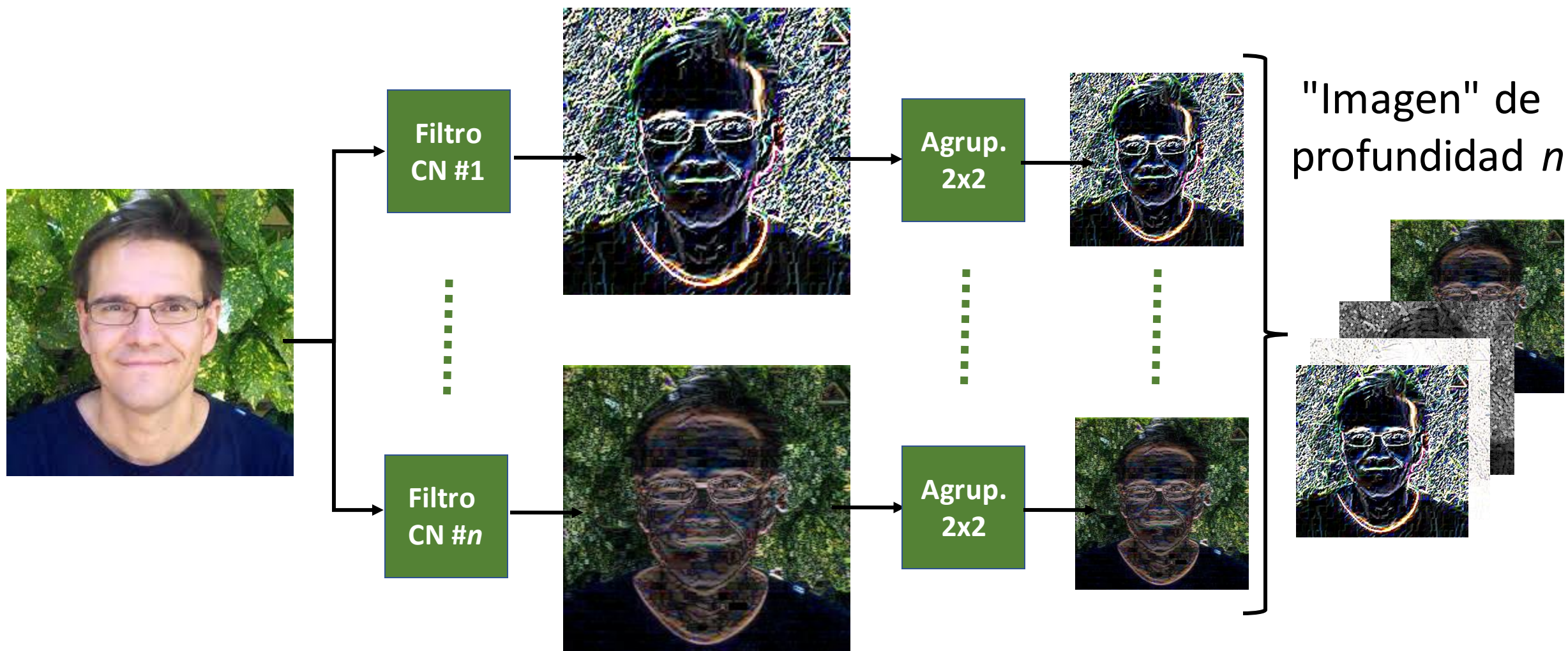
¿Cómo se hace?

- Modelo computacional por **capas**

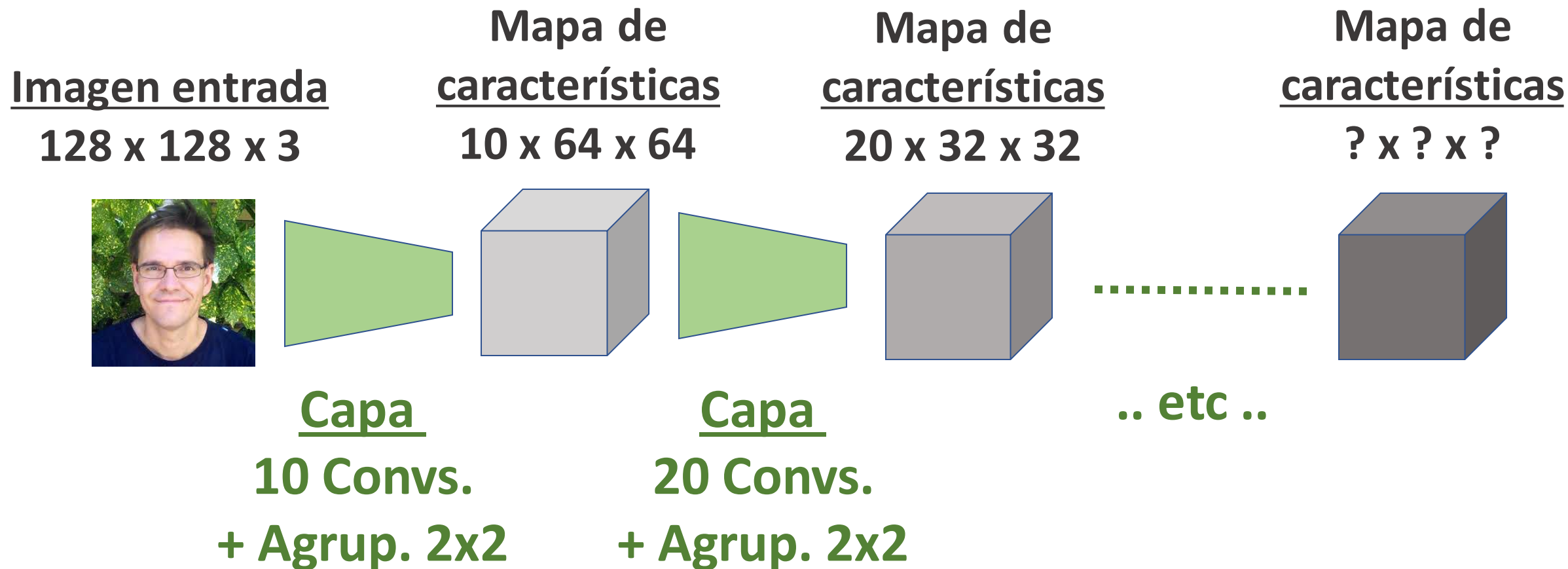
Cada capa procesa la salida de la capa anterior

- Este procesado consiste en:
 1. Aplicar filtros de **convolución** para resaltar características visuales (bordes, esquinas, etc)
 2. Agrupar los píxeles más resaltados → **reducción** de tamaño
 3. Tras varias capas, **combinar** los pixels resultantes para **aprender a realizar la tarea**

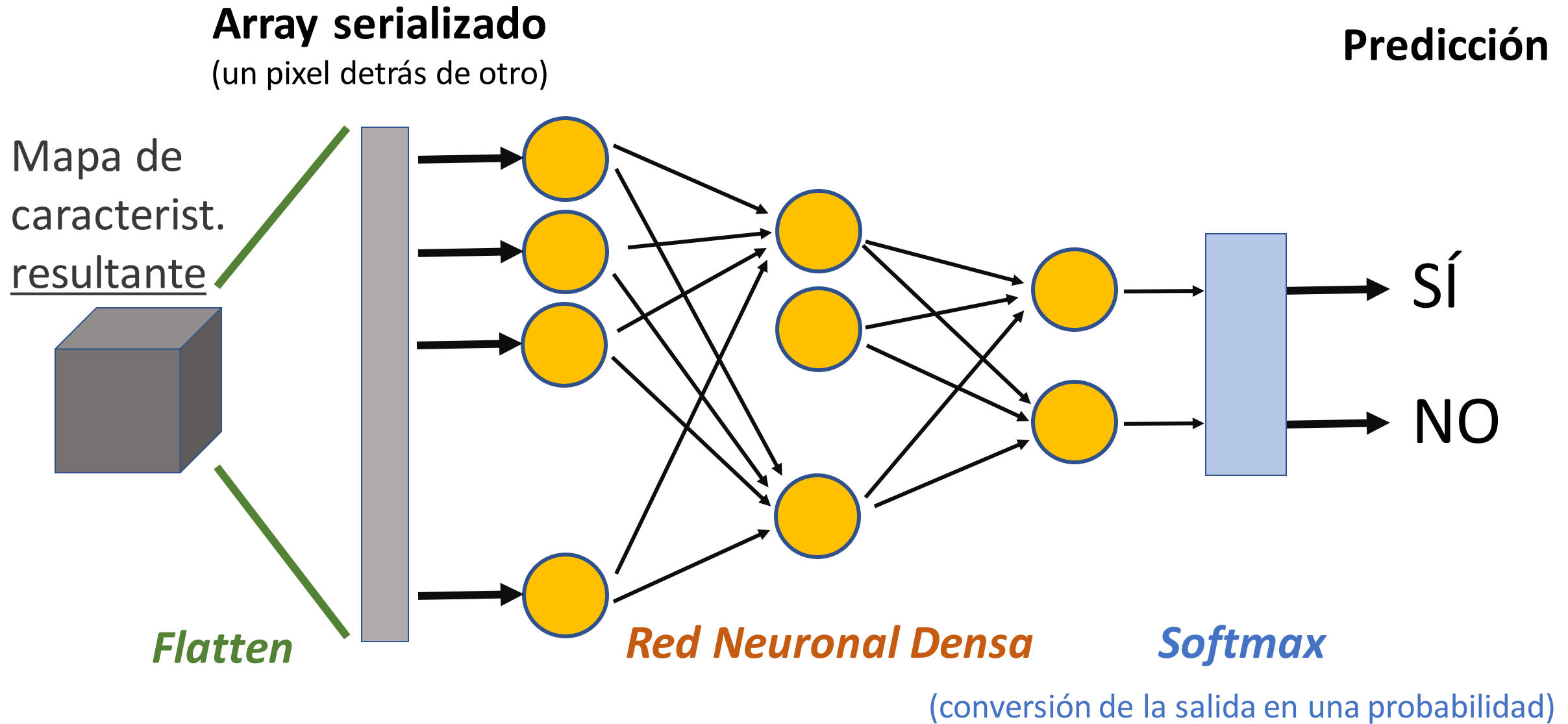
La capa convolucional + la capa de agrupamiento



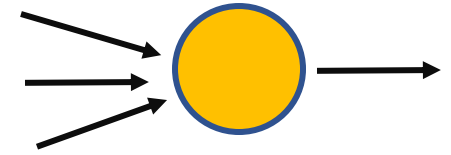
Modelo por capas



Combinación de los píxeles resultantes



¿Pero qué es una neurona artificial?



Modelo computacional que consiste en:

- **Combinación lineal** de las entradas
- **Modificación no-lineal** del resultado
(función de activación)
Hay muchas: Lineal, Sigmoide, ReLU, ...
- Las convoluciones discretas TAMBIÉN son combinaciones lineales
(se les llama "Neuronal convolucionales")

$$z = \sum_{k=1}^K w_k x_k$$

$$y = \begin{cases} 0 & \text{si } z < 0 \\ z & \text{si } z \geq 0 \end{cases}$$

¿Cómo puede aprender una máquina?

- Depende de la tarea
- Un caso sencillo: reconocer señales de tráfico [\[Enlace\]](#)
- **Ingredientes:**
 - Conjunto con (muchas) imágenes, $\{ \mathbf{x} \}$
 - Cada imagen debe tener asociada una etiqueta, \mathbf{y}
- **Receta:**
 - Introducir una imagen \mathbf{x} en la máquina
 - La máquina hace una predicción sobre la etiqueta de la imagen, \mathbf{y}_{pred}
 - Medimos el error o pérdida (*loss*), $L = | \mathbf{y} - \mathbf{y}_{pred} |$
 - Ajustamos los parámetros (\mathbf{w}) de la máquina para reducir el error
 - Volver a empezar con otro ejemplo

¿Puedo hacerlo yo también?



¡¡ Software libre !!

- Python
- Keras
- Tensorflow
- GitHub ...

PERO, en general, hace falta potencia computacional

(GPU NVIDIA, TPU Google)

→ Con **Google Colab** se pueden aprender muchas técnicas

lo usamos en el MOVA-URJC

¡¡ Manos a la obra !!



Vamos a reconocer dígitos manuscritos utilizando el conjunto de datos MNIST

https://en.wikipedia.org/wiki/MNIST_database

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 |
| 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |

...seguimos con el código dado en *Google Colab*

<https://github.com/alfredo-cuesta/Plday2019.git>

Referencias

Enlaces interesantes

- Tensorflow Playground: [\[Enlace\]](#)
- Para dibujar redes neuronales: [\[Enlace\]](#)
- DeepLearning.ai: [\[Enlace\]](#)

Si quieres continuar tu formación en la URJC

- Master Oficial en Visión Artificial URJC: <https://mastervisionartificial.es/>

Más sobre nosotros

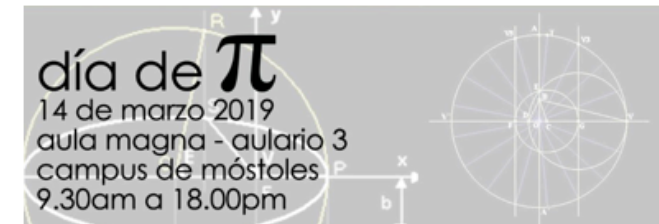
- Grupo de Computación de Altas Prestaciones y Optimización (CAPO): <http://www.caporesearch.es>
- Juan José Pantrigo: <http://www.caporesearch.es/jjpantrigo>
- Alfredo Cuesta: <http://www.caporesearch.es/acuesta>

Gracias por venir

Visión Artificial con Redes Neuronales Profundas

2019.**3.14**

juanjose.pantrigo@urjc.es
alfredo.cuesta@urjc.es



**Master
Oficial en
Visión
Artificial**



**Computación de
Altas
Prestaciones y
Optimización**