

# **PYTHON:**

# **APRENDIZAJE POR REFUERZO**

MEDIANTE PROCESOS DE DECISIÓN MARKOV

FERNANDO GONZÁLEZ-HERRERA

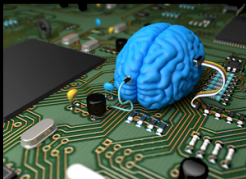
MSC – CENTRO DE INVESTIGACIÓN EN MATEMÁTICAS A.C.

BSC – INSTITUTO POLITÉCNICO NACIONAL - UPIIZ

NOVIEMBRE, 2019



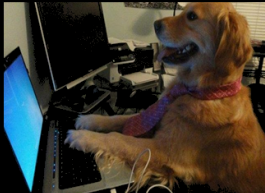
What society thinks I do



What my friends think I do



What other computer scientists think I do



What mathematicians think I do



What I think I do

```
In [1]:  
import keras  
Using TensorFlow backend.
```

What I actually do

1. Introducción
  - ▶ Acrónimos
  - ▶ Notación
  - ▶ Panorama general
2. Python
3. Google Colaboratory
4. Herramientas adicionales para Python
5. Aprendizaje por Refuerzo
6. Referencias

# INTRODUCCIÓN

Acrónimo	Significado
AI	<i>Artificial Intelligence</i> , inteligencia artificial
ML	<i>Machine learning</i> , aprendizaje máquina
DP	<i>Deep learning</i> , aprendizaje máquina profundo
RL	<i>Reinforcement learning</i> , aprendizaje por refuerzo
MDP	<i>Markov desicion process</i> , proceso de decisión de Markov
POMDP	<i>Partially observed Markov desicion process</i> , proceso de decisión de Markov parcialmente observado

# NOTACIÓN

$a$	Un escalar	$1$
$\mathbf{a}$	Un vector	$[1 \ 2 \ 3]$
$A$	Una matriz	$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
$\mathbf{A}$	Un tensor	$\begin{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} & \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \\ \begin{bmatrix} 9 & 10 \\ 11 & 12 \end{bmatrix} & \begin{bmatrix} 13 & 14 \\ 15 & 16 \end{bmatrix} \end{bmatrix}$

# PYTHON

# ¿PORQUÉ PYTHON?

Cuando lo creé, J.A.R.V.I.S.  
era una interfaz de lenguaje  
natural...

---

*Anthony “Tony” Stark*





A finales de los 80's por Guido van Rossum en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica).

- No obliga al programador a seguir un paradigma de programación (Orientado a objetos, imperativa ó funcional)
- 'Fácil' implementación de cómputo distribuido
- Programación en IDE (Jupyter, Anaconda, Spyder) editores de texto (VIM, EMACS, Notepad++).

## ■ Python 3.7 (Prev. 3.6)

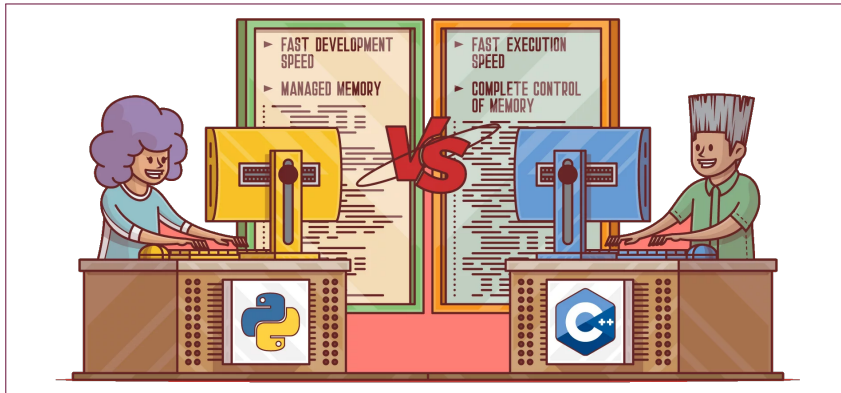
- ▶ Optimizado a hardware actual
- ▶ Mejor sintaxis
- ▶ Actualizaciones

## ■ Python 2.7

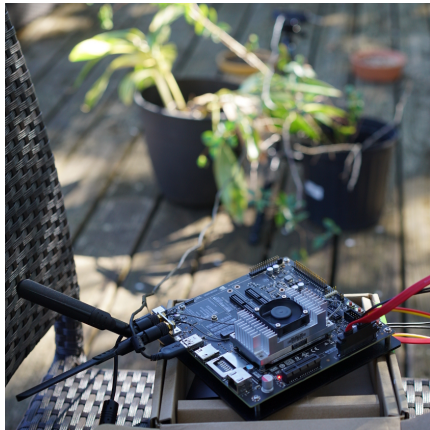
- ▶ Soporte hasta 2020
- ▶ Poco uso real



# PYTHON VS C++



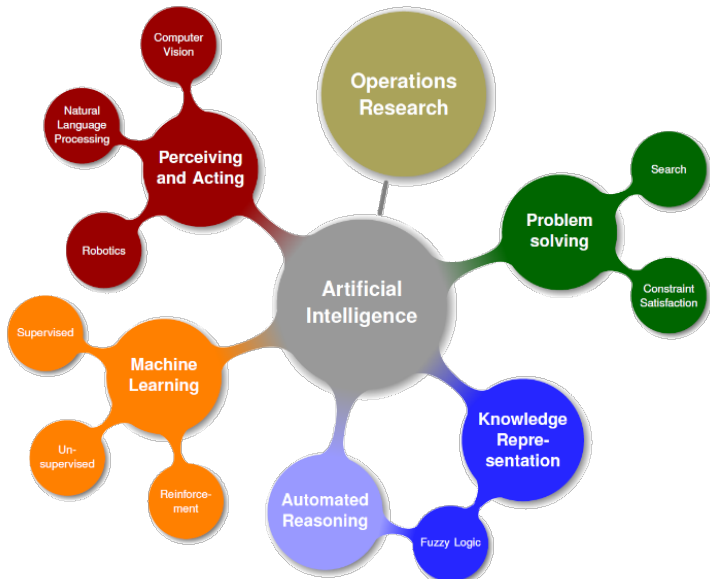
## NVIDIA Jetson TX1



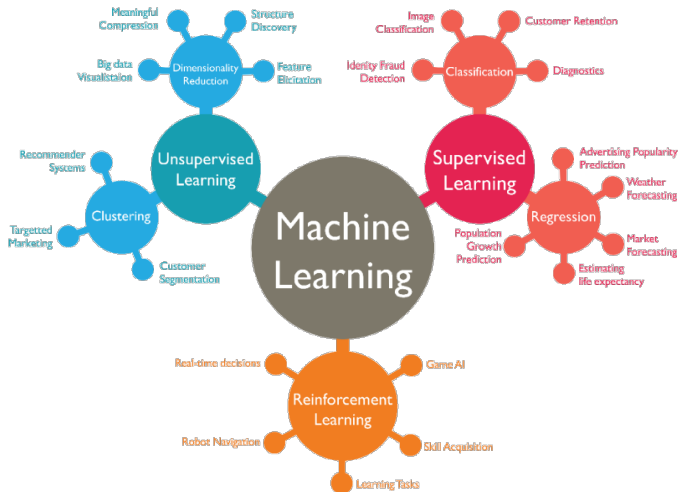
Creciente número de librerías, frameworks, módulos y modelos pre-entrenados.



# INTELIGENCIA ARTIFICIAL



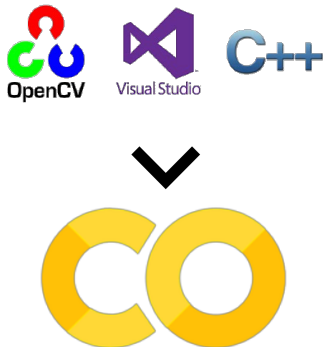
# MACHINE LEARNING, APRENDIZAJE MÁQUINA



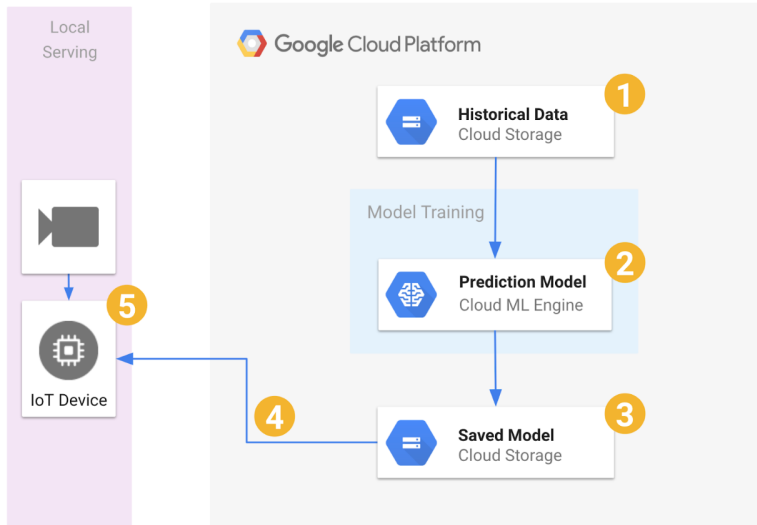
**GOOGLE COLAB**



Integración de herramientas, librerías y IDE.



# PIPELINE COMÚN DE DESARROLLO



# **HERRAMIENTAS ADICIONALES PARA PYTHON**

Dependiendo de su uso:

- Cómputo de matrices (Numpy, CuPy)
- Entrada y Salida de información (Pandas, Modin)
- Graficación (Matplot lib, GNUplot, PlotPy)
- Cómputo de alto rendimiento –redes neuronales (Tensorflow, PyTorch)
- \*Distribuciones de paquetes (pip, yarn, Anaconda, Spyder)

Gracias por su atención.

*Fernando González-Herrera*

