

Universidad de Cantabria  
Instituto de Física de Cantabria (IFCA), CSIC-UC

## Introducción a PyTorch

Jose González-Abad

## ¿Qué es PyTorch?

Hasta ahora en esta asignatura hemos estado utilizando **TensorFlow + Keras**

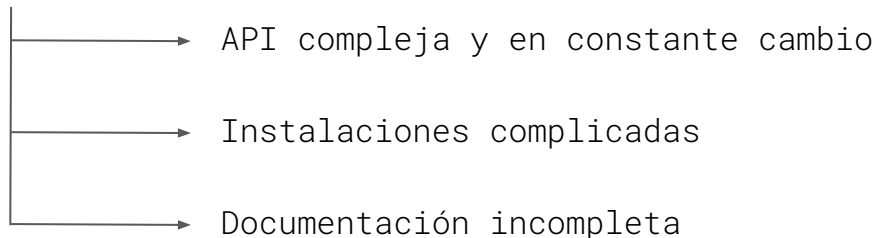


Keras permite un uso sencillo y directo de TensorFlow, sin embargo **no nos aporta flexibilidad**

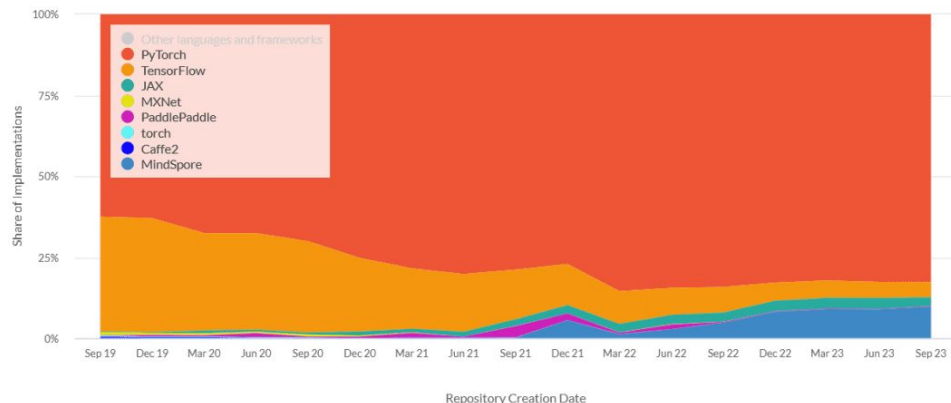
- ¿Y si quiero debugear el entrenamiento de un modelo?
- ¿Y si quiero imprimir los gradientes del modelo en cierta epoch?
- ¿Y si quiero implementar una nueva forma de entrenamiento? (e.g., *adversarial*)

## ¿Qué es PyTorch?

Para mayor flexibilidad podríamos utilizar TensorFlow directamente, sin embargo sufre de algunos problemas



Por ello, **PyTorch** surgió como alternativa a TensorFlow



## ¿De quién es PyTorch?

PyTorch es un framework **open-source** para deep learning desarrollado en 2016 por Facebook's AI Research lab (**FAIR**). Hoy día está mantenido por **META** y gobernado por la **PyTorch Foundation (Linux Foundation)**



El principal objetivo de PyTorch es ofrecer un framework **intuitivo** y **flexible** para el desarrollo de modelos de deep learning, **simplificando el debugging y facilitando la innovación**



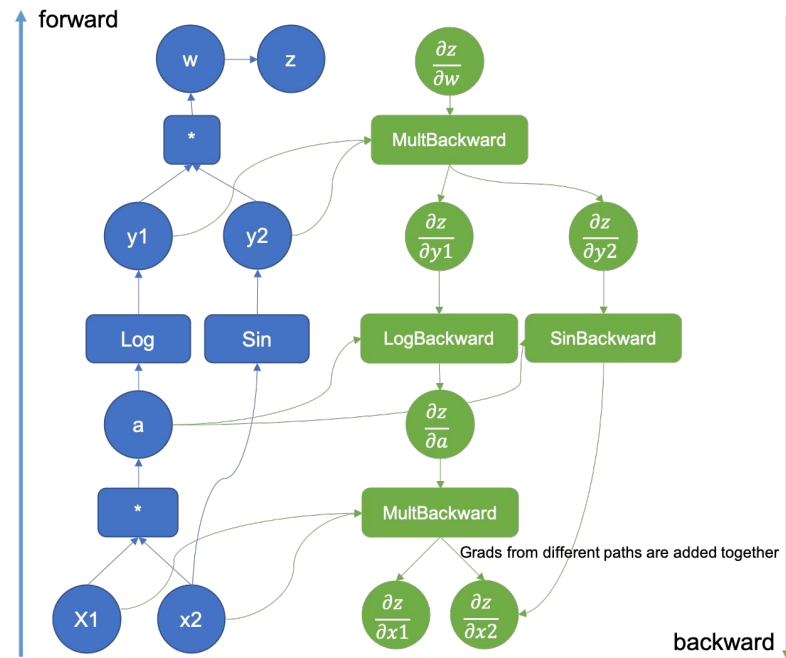
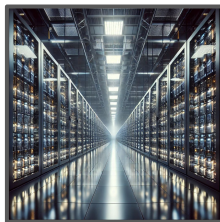
# Grafos de computación

En su momento, la principal razón detrás del cambio desde TensorFlow hacia PyTorch fueron los **grafos de computación dinámicos**

Cuando construimos un modelo de deep learning, el framework crea un **grafo de computación**. Este grafo es clave para implementar de forma **eficiente** los forward y backward passes

¿Qué depende de qué?

¿Qué puedo paralelizar?



## Grafos de computación

En su día TensorFlow implementaba grafos de computación estáticos, mientras que **PyTorch introdujo los grafos de computación dinámicos**



TensorFlow requería definir el grafo completo antes de realizar ninguna operación, lo que hacía que debugear o hacer cambios al modelo no fuese tan directo



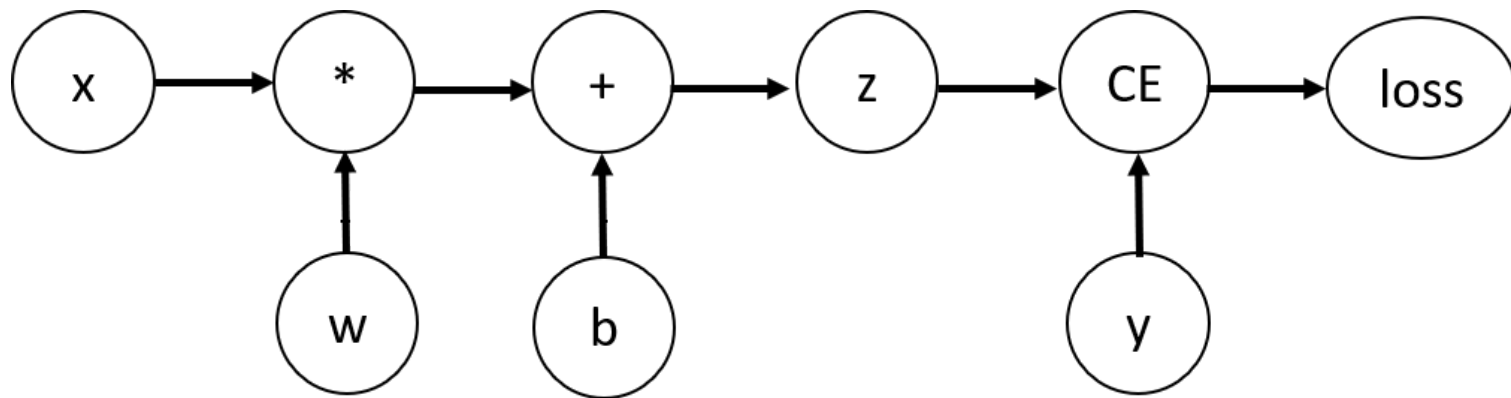
PyTorch define el grafo a medida que se ejecuta, por lo que admite cambios en tiempo real



Una gran ventaja para hacer investigación

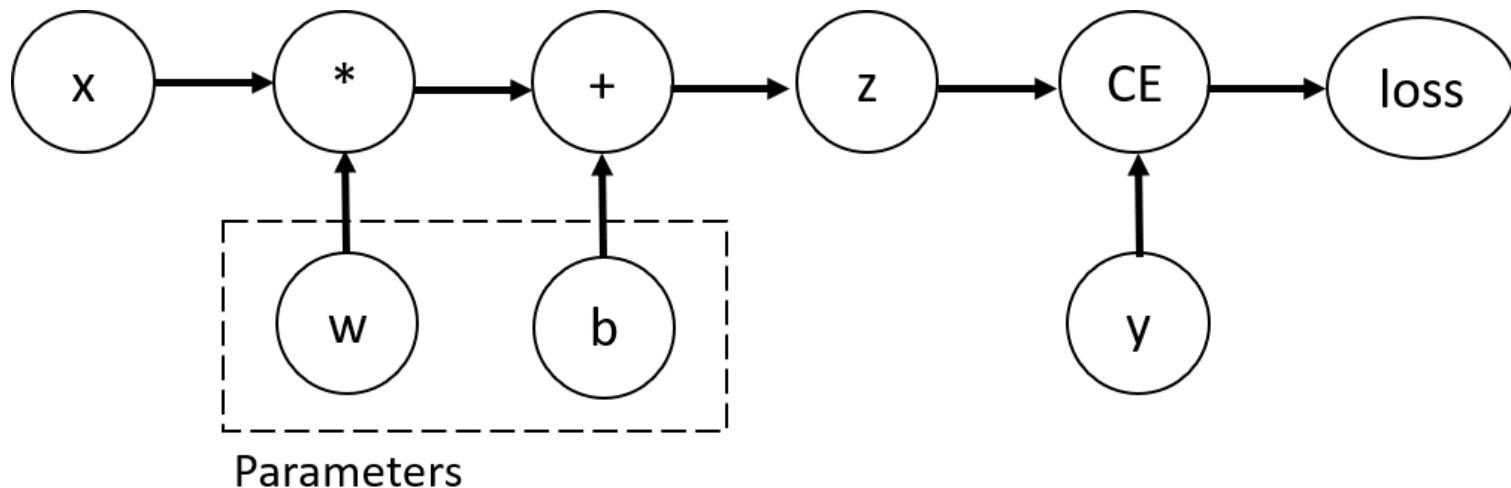
## Automatic Differentiation

Otra de las características relevantes de PyTorch es su **torch.autograd**, que implementa el **cálculo automático de gradientes** en un grafo de computación



## Automatic Differentiation

Otra de las características relevantes de PyTorch es su **torch.autograd**, que implementa el **cálculo automático de gradientes** en un grafo de computación



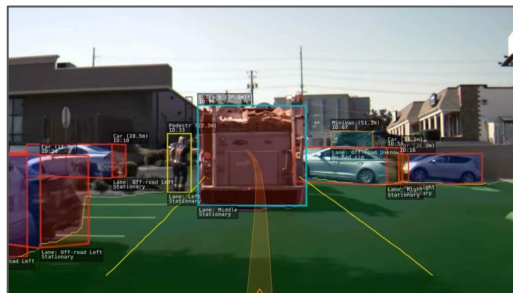
En este caso, **torch.autograd** se encarga de **generar la función para calcular los gradientes** de *w* y *b* de forma automática



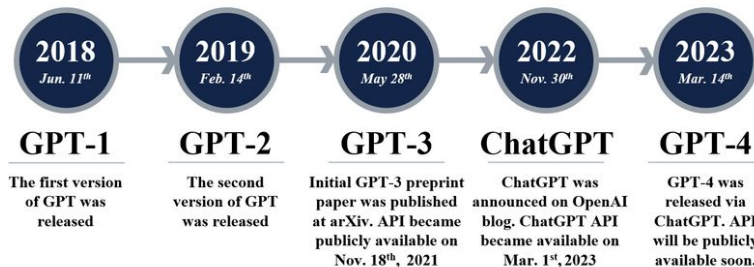
## Ejemplos de aplicación

Estas características han hecho que PyTorch se utilice tanto en **investigación** como en **industria**

### Tesla's Autopilot



### OpenAI's GPT models



## Grafos de computación

La instalación de PyTorch es muy directa y rápida, tanto para GPU como CPU

<https://pytorch.org/get-started/locally/>

PyTorch Build	Stable (2.6.0)			Preview (Nightly)	
Your OS	Linux		Mac	Windows	
Package	<del>Conda</del>	Pip		LibTorch	Source
Language	Python			C++ / Java	
Compute Platform	CUDA 11.8	CUDA 12.4	CUDA 12.6	ROCm 6.2.4	CPU
Run this Command:	<pre>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu</pre>				

Diagram illustrating the PyTorch installation options and the resulting command for CPU installation:

- PyTorch Build: Stable (2.6.0) (selected), Preview (Nightly)
- Your OS: Linux (selected), Mac, Windows
- Package: ~~Conda~~ (crossed out), Pip (selected), LibTorch, Source
- Language: Python (selected), C++ / Java
- Compute Platform: CUDA 11.8, CUDA 12.4, CUDA 12.6, ROCm 6.2.4, CPU (selected)

The command to run is:

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cpu
```

Arrows indicate the mapping from the selected options to the command and hardware components:

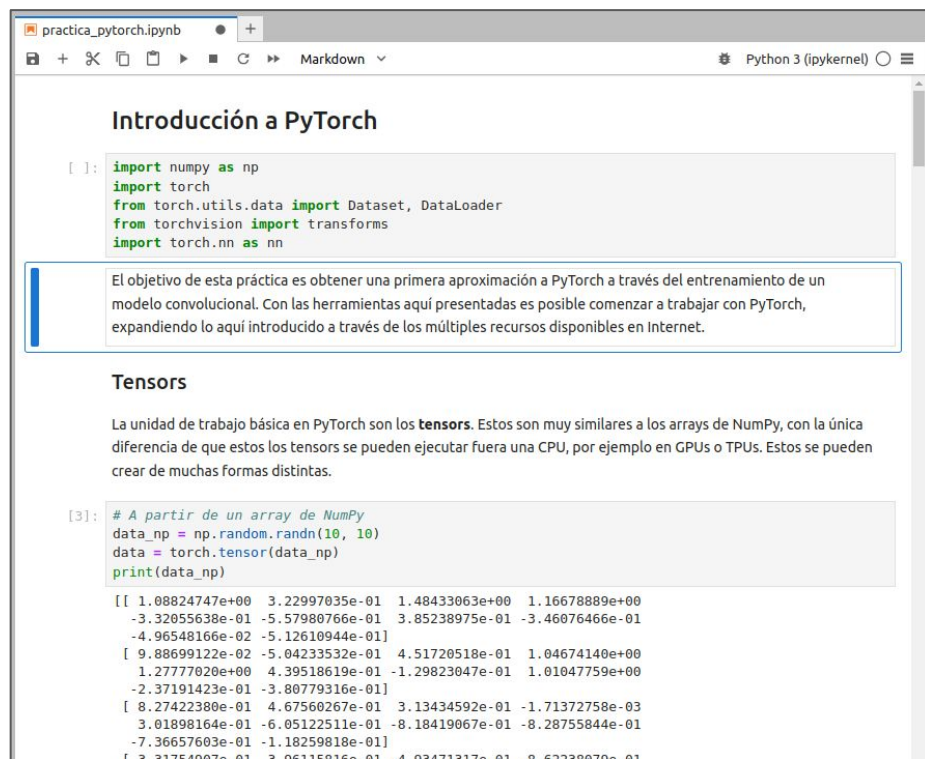
- From the **CUDA 11.8**, **CUDA 12.4**, and **CUDA 12.6** options to the `--index-url https://download.pytorch.org/whl/cpu` part of the command.
- From the **ROCm 6.2.4** option to the `--index-url https://download.pytorch.org/whl/cpu` part of the command.
- From the **CPU** option to the `--index-url https://download.pytorch.org/whl/cpu` part of the command.
- From the **CUDA 11.8**, **CUDA 12.4**, and **CUDA 12.6** options to the `torch`, `torchvision`, and `torchaudio` packages in the command.

Hardware components shown:

- NVIDIA GPU (selected for CUDA)
- AMD GPU (selected for ROCm)
- Intel Xeon processor (selected for CPU)

# Uso de PyTorch

Para conocer mejor cómo utilizar PyTorch, vamos a continuar en el documento **practica\_pytorch.ipynb** en el **Moodle**



The screenshot shows a Jupyter Notebook interface with the title 'practica\_pytorch.ipynb'. The notebook is running on 'Python 3 (ipykernel)'. The first cell contains the following code:

```
[ ]: import numpy as np
import torch
from torch.utils.data import Dataset, DataLoader
from torchvision import transforms
import torch.nn as nn
```

Below the code cell, there is a text box with the following content:

El objetivo de esta práctica es obtener una primera aproximación a PyTorch a través del entrenamiento de un modelo convolucional. Con las herramientas aquí presentadas es posible comenzar a trabajar con PyTorch, expandiendo lo aquí introducido a través de los múltiples recursos disponibles en Internet.

The second cell is titled 'Tensors' and contains the following text:

La unidad de trabajo básica en PyTorch son los **tensors**. Estos son muy similares a los arrays de NumPy, con la única diferencia de que estos los tensors se pueden ejecutar fuera una CPU, por ejemplo en GPUs o TPUs. Estos se pueden crear de muchas formas distintas.

The third cell contains the following code:

```
[3]: # A partir de un array de NumPy
data_np = np.random.randn(10, 10)
data = torch.tensor(data_np)
print(data_np)
```

The output of the code is a 10x10 array of random values:

```
[[ 1.08824747e+00  3.22997035e-01  1.48433063e+00  1.16678889e+00
 -3.32055638e-01 -5.57980766e-01  3.85238975e-01 -3.46076466e-01
 -4.96548166e-02 -5.12610944e-01]
 [ 9.88699122e-02 -5.04233532e-01  4.51720518e-01  1.04674140e+00
 1.27777020e+00  4.39518619e-01 -1.29823047e-01  1.01047759e+00
 -2.37191423e-01 -3.80779316e-01]
 [ 8.27422380e-01  4.67560267e-01  3.13434592e-01 -1.71372758e-03
 3.01898164e-01 -6.05122511e-01 -8.18419067e-01 -8.28755844e-01
 -7.36657603e-01 -1.18259818e-01]
 [ 3.31754007e-01  3.96115816e-01 -4.93471317e-01 -8.62238070e-01
```