

Curso de PyTorch

@espejelomar

**¿Qué necesitas
para aprender
PyTorch?**



P00

con Python



NumPy
y Matplotlib



Fundamentos de deep learning

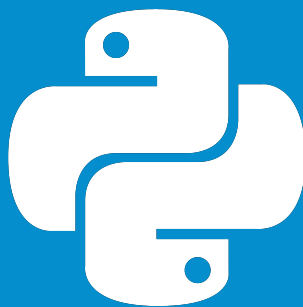
 Explorar
Hugging Face

**¿Qué es y
por qué usar
PyTorch?**

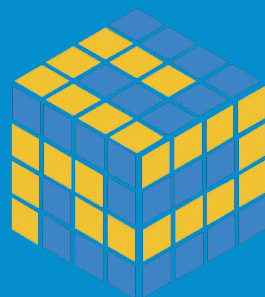


**From
Research to
production**

 PyTorch



 PyTorch



NumPy



TORCH.TENSOR.NUMPY

Pytorch Foundation

Members and Governing Board

The AMD logo, featuring the word "AMD" in a bold, black, sans-serif font, followed by a stylized square icon composed of four smaller squares.The AWS logo, featuring the word "aws" in a lowercase, black, sans-serif font, with a curved orange arrow underneath it.The Google Cloud logo, featuring the word "Google" in its multi-colored font, followed by the word "Cloud" in a black, sans-serif font.The Meta logo, featuring a blue infinity symbol followed by the word "Meta" in a black, sans-serif font.The Microsoft Azure logo, featuring the four-colored square icon (red, green, blue, yellow) followed by the words "Microsoft" and "Azure" in a black, sans-serif font.The NVIDIA logo, featuring a green stylized eye icon above the word "NVIDIA" in a bold, black, sans-serif font.

Contribuidores de PyTorch



fast.ai



PT y TF: el comienzo



TensorFlow → liberado
por Google en 2015.

PT y TF: el comienzo



PyTorch → liberado por
Meta en 2016.

PT y TF: el comienzo

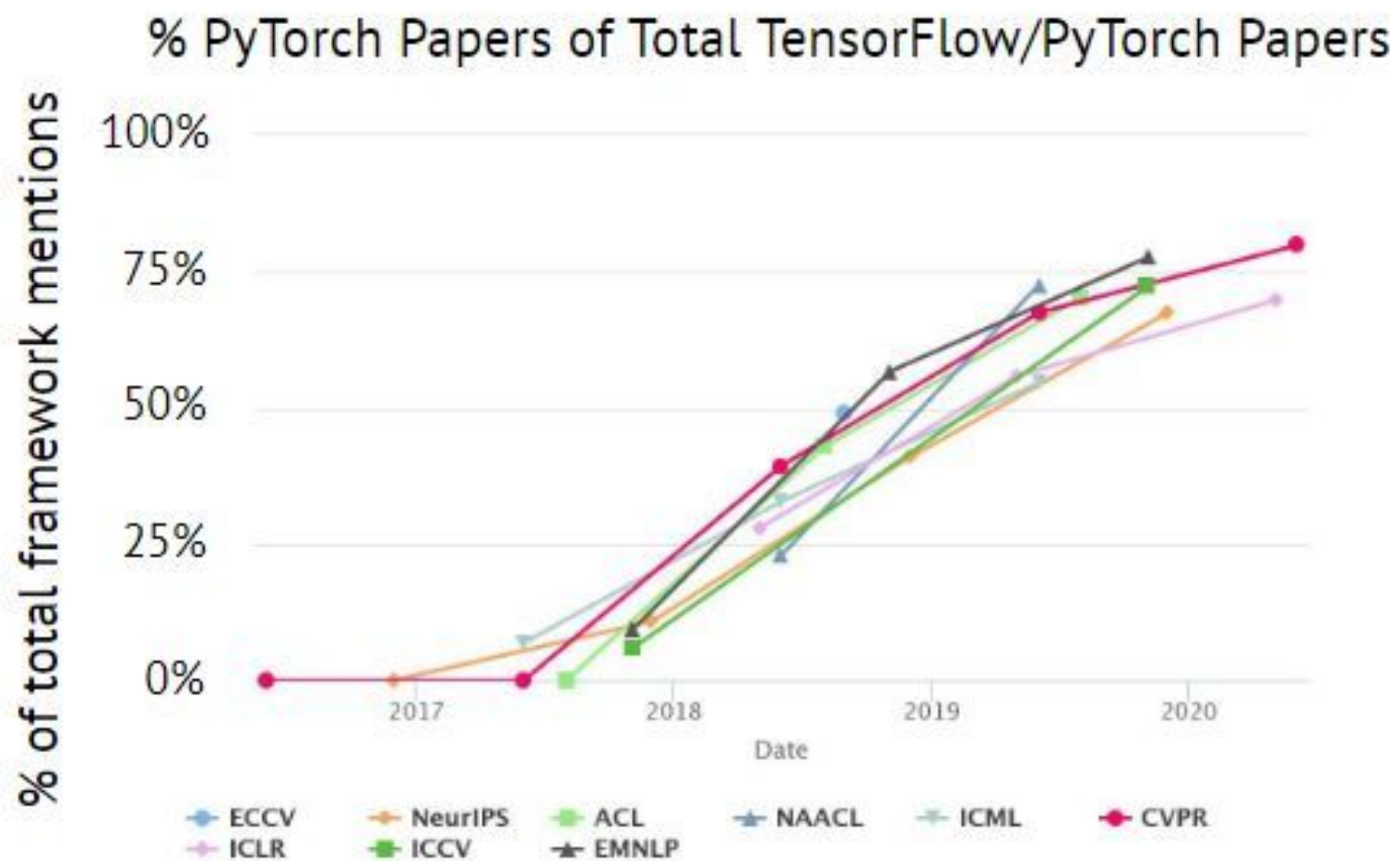


PyTorch → enfoque en
facilitar al usuario la
programación.

**1 de cada 2
investigadores
cambió TensorFlow
por PyTorch. Pero...**

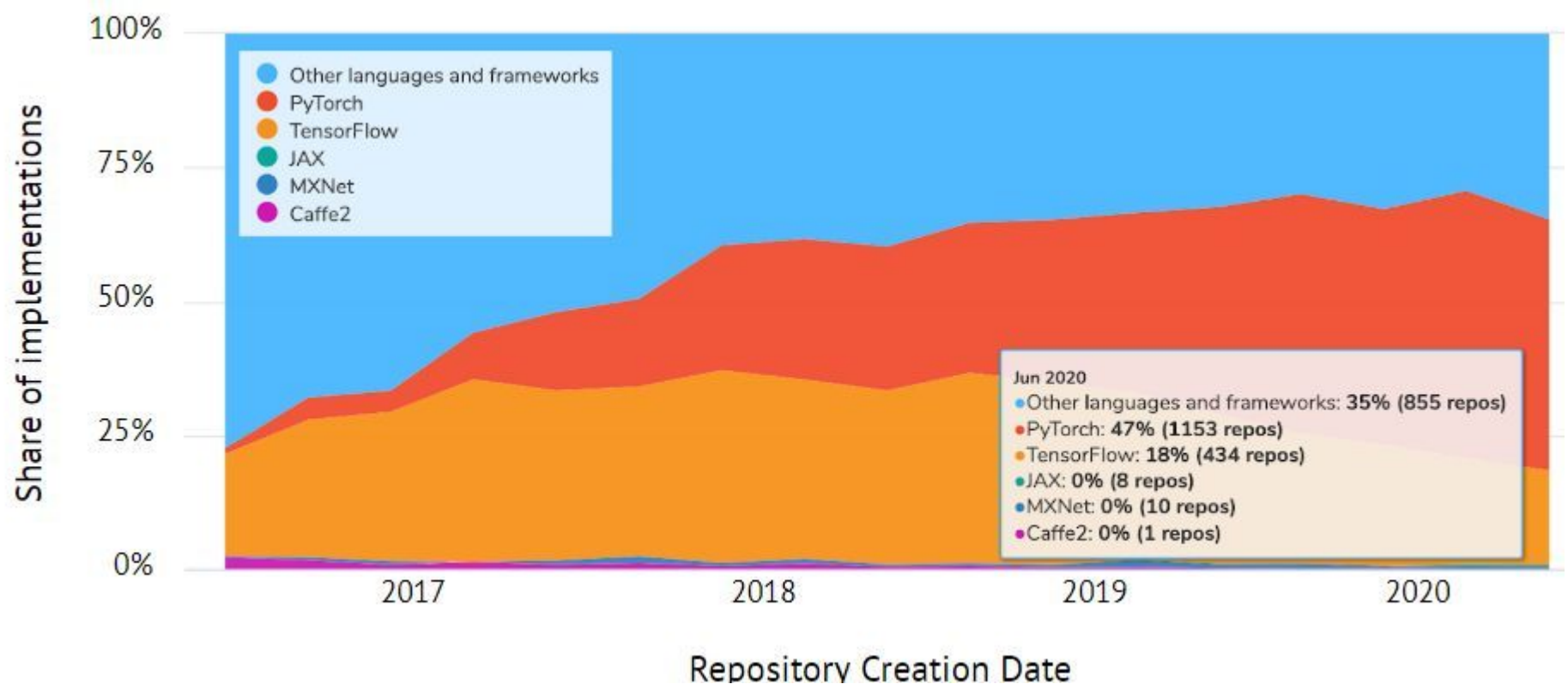
**...menos de 1 de cada
5 investigadores
dejaron PyTorch
por TensorFlow.**

PyTorch en la investigación



Fuente: State of AI Report 2022, <https://www.stateof.ai/>

PyTorch cada vez más usado



Fuente: State of AI Report 2022, <https://www.stateof.ai/>

PyTorch en open-source



Hugging Face



PyTorch



Models 71,170



TensorFlow



Models 5,952



JAX



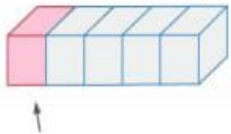
Models 8,078

**Hola, mundo
en PyTorch**

Creación de Tensores en PyTorch

Dimensiones

Scalar



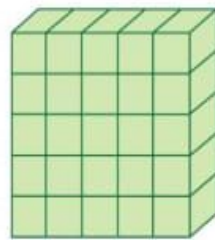
dim
= 0

Vector



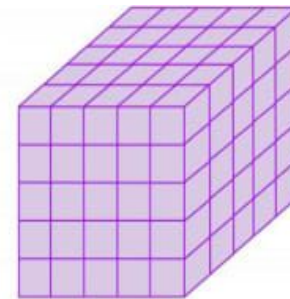
dim
= 1

Matrix

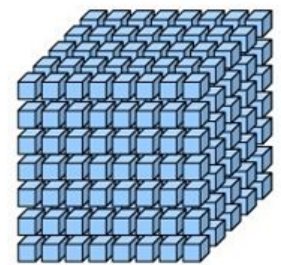


dim
= 2

Tensor



dim
= 3



dim
= n

Tensor



Debugging y operaciones con tensores

Conversión y operación de tensores con PyTorch

Generación y split de datos para entrenamiento

Estructura de modelo en PyTorch con torch.nn

**Entrenamiento,
funciones de
perdida y
optimizadores**

Entrenamiento, y visualización de pérdida

Predicción con un modelo de PyTorch entrenado

Datos para clasificación de texto

**Procesamiento
de datos:
tokenización y
creación de
vocabulario**

Procesamiento de datos: preparación del DataLoader

Creación de modelo de clasificación con PyTorch

Función para entrenamiento

Función para evaluación

**Split de datos,
pérdida y
optimización**

Entrenamiento y evaluación del modelo de clasificación de texto

**Inferencia
utilizando
torch.compile():
el presente con
PyTorch 2.X**

**Almacenamiento
del modelo con
torch.save()
y state_dict()**

**Sube tu modelo
de PyTorch a
Hugging Face**

Carga de modelo de PyTorch