

Mi primera aplicación con Pytorch

🔧 Repo en construcción... 🛠️👷🔧👷

👷👷 Autores: [Andrea Morales Garzón](#) y [Miguel López Pérez](#).

📖 Taller organizado en el contexto de la [PyconES 2022](#).

📘 ¡Más información sobre este taller [aquí](#)!

📍 Granada

📅 30 de septiembre del 2022

🕒 15:30–19:30

Descripción del taller

Este taller pretende que los asistentes, sin necesidad de tener conocimientos previos sobre la librería, desarrollen una aplicación en Pytorch, una de las librerías por excelencia en Python para la implementación de algoritmos basados en técnicas de aprendizaje automático. En concreto, los asistentes abordarán el diseño y desarrollo de una aplicación de aprendizaje automático desde cero, partiendo de lo más básico a nivel de funcionamiento del framework, y concluyendo con el despliegue de una aplicación propia en la web.

Este taller se divide en tres bloques:

- **a. Introducción a Pytorch.** Este bloque explica en qué consisten las librerías para redes neuronales como Pytorch y cómo funcionan, dando una idea general de qué son las redes neuronales y cómo esta librería nos ayuda. Se contará con un ejercicio sobre una red neuronal sencilla y problemas de optimización que permitan comprender las nociones más básicas de la librería. Para este bloque usaremos los siguientes notebooks:
 - [0_Introducción_a_Pytorch.ipynb](#) contiene una breve introducción al framework Pytorch.
 - [1_tensores.ipynb](#) introduce los tensores, estos son los elementos con los que trabaja Pytorch.
 - [2_regresión.ipynb](#) contiene un ejemplo de regresión con un modelo lineal y con una red neuronal sencilla. Nos servirá para ver cómo se definen y optimizan modelos en Pytorch.
- **b. Pytorch para CV y NLP.** Este bloque abordará ejemplos prácticos dónde podremos usar Pytorch. Veremos su aplicación en imágenes y en texto. Para ello se usarán los siguientes

notebooks:

- [3_clasificación_imágenes.ipynb](#) contiene un ejemplo de clasificación de imágenes de satélites. Para ello se entrenará tanto una red convolucional desde 0 como una arquitectura preentrenada (de las más potentes que hay!).
- [4_procesamiento_lenguaje_natural.ipynb](#) se generará poemas con una red neuronal de manera automática!
- c. **Despliegue de una aplicación en Pytorch.** Ya con una visión general de la librería, se explicará como se puede construir una aplicación sencilla con Pytorch, con el objetivo de hacer un despliegue en la nube de un sistema de aprendizaje automático que pueda hacer una tarea sencilla. Para ello usaremos el siguiente script:
 - [5_aplicación_clasificación_imágenes.py](#) tiene el script en Python que lanza una aplicación en streamlit de la red neuronal de visión previamente entrenada en el notebook número 3.

Todos los notebooks y scripts contienen **ejercicios** con los que hacer la experiencia más interactiva y poder adquirir los conocimientos de forma adecuada.

Antes de comenzar...

Para que puedas trabajar de la forma que te resulte más cómoda, os damos varias opciones de trabajo para este taller. ¡Elige la que más te guste!

1. ¡Empezando desde cero con Conda!

Si no tienes Conda instalado, puedes seguir las instrucciones detalladas en la página oficial de conda en el siguiente [enlace](#).

Una vez con conda instalado en nuestro ordenador, lo primero es crear un entorno de trabajo para este taller. De esta forma partiremos de cero, con un entorno limpio, sobre el que trabajar y no interceder en otros proyectos en marcha.

```
$ conda create -n first_pytorch python=3.7
```

¡Lo primero es lo primero! Instala [Pytorch](#) con Conda:

```
$ conda install pytorch torchvision torchaudio cudatoolkit=11.3 -c pytorch
```

La forma más cómoda de trabajar es por medio de notebooks. Podemos utilizar *Jupyter*, una aplicación web de código abierto que nos permite llevar a cabo todo el proceso de desarrollo de un proyecto de forma interactiva, visual, intercalando textos y código.

```
$ conda install -c anaconda jupyter
```

Los proyectos de Machine Learning en Python suelen usar librerías comunes, como pueden ser aquellas que nos permiten trabajar con estructuras de datos (como *Pandas*) y realizar visualizaciones de datos (como *matplotlib*). ¡Las instalamos también, pues las usaremos como buenos científicos de datos que somos!

```
$ conda install -c conda-forge matplotlib
$ conda install -c anaconda pandas
```

Además, también necesitamos instalar *torchtext* para texto y *pillow* para tratar con imágenes. Por último, conseguiremos notebooks interactivos muy vistosos con *ipywidgets*.

```
$ conda install -c pytorch torchtext

$ conda install -c conda-forge ipywidgets

$ conda install -c anaconda pillow
```

¡Ya estamos acabando de preparar nuestro entorno! Nos falta Streamlit, una librería que nos permitirá desplegar nuestro proyecto. Para ello ejecutamos la siguiente orden:

```
$ conda install -c conda-forge streamlit
```

2. ¡Prefiero pip!


De igual forma, todas las librerías anteriores se pueden instalar a través de [pip](#). Te recomendamos que, de seguir por esta vía, hagas uso de algún entorno virtual en el que desarrollar toda tu aplicación sin problemas. Si eres nuevo en esto, ¡puedes seguir los pasos detallados [aquí](#) o en este [tutorial](#)!

3. ¡No tengo Python instalado!

¡Pero qué poca vergüenza! Pero no pasa nada, en este caso [Google Colab](#) es tu mejor amigo! Puedes hacer uso de todos los recursos del taller a través de Google Colab sin tener que instalar Python en tu ordenador. Puedes seguir las [instrucciones oficiales](#) para aprender cómo importar una librería en un notebook de Colab.

Contacto

Ya sea para dudas o sugerencias, ¡estamos a vuestra disposición! Podeis contactarnos a través de la info de contacto detallada en nuestras páginas personales. Enlaces a continuación:

-  [Andrea Morales Garzón](#)
-  [Miguel López Pérez](#)