

Laboratorio: Preparación del entorno



Aprendizaje
Profundo

Grado en Ingeniería y Ciencia de datos (Universidad de Oviedo)

Pablo González, Pablo Pérez
{gonzalezgpablo, pabloperez}@uniovi.es
Centro de Inteligencia Artificial, Gijón

Introducción

Durante estas prácticas trabajaremos fundamentalmente usando **Python** y el framework de aprendizaje profundo **PyTorch**.



Accelerated with



Utilización de GPUs

PyTorch permite la ejecución de operaciones con tensores tanto en CPU como en GPU. En este curso aprenderemos a trabajar con este framework, tanto en local (generalmente con CPU), como en servidores especializados (GPU) y también con **Google Colab**, donde Google pone a nuestra disposición recursos gratuitos a través de **Notebooks de Jupyter**.

Entorno virtuales

Los entornos virtuales son herramientas muy útiles para aislar las librerías que utilizaremos en nuestros experimentos en un **entorno virtual**. Las opciones más utilizadas son los gestores de paquetes **conda** y **pip**.

Para la instalación de **conda**, tenemos dos opciones principales:

- **Anaconda**. Incluye Python, conda, una amplia gama de bibliotecas científicas y herramientas útiles para análisis de datos y ciencia de datos.
- **Miniconda**. Es una versión más ligera y básica de Anaconda que incluye solo Python y conda, sin paquetes adicionales preinstalados.

Instalación

Tanto **Anaconda** como **Miniconda** se pueden instalar desde el siguiente enlace.

Instalación de conda

Para instalar conda necesitamos descargar el paquete correspondiente e instalarlo. Podemos hacerlo con las siguientes instrucciones:

```
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

Una vez instalado y de reiniciar la terminal deberíamos de ser capaces de ejecutar el comando *conda*.

```
(base) nvidia@dl-gpu:~$ conda
usage: conda [-h] [-V] command ...

conda is a tool for managing and deploying applications, environments and packages.

Options:

positional arguments:
  command
    clean                Remove unused packages and caches.
    compare              Compare packages between conda environments.
    config               Modify configuration values in .condarc. This is modeled
                        after the git config command. Writes to the user
                        .condarc file (/home/nvidia/.condarc) by default. Use
                        the --show-sources flag to display all identified
                        configuration locations on your computer.
```

Creación de un entorno virtual

Entorno virtual base

Conda por defecto crea un entorno virtual llamado **base**. Es una buena práctica no instalar dependencias en este entorno. En su lugar crearemos un entorno virtual nuevo con los paquetes que nos interesen.

Para crear un entorno procederemos de la siguiente manera. En este caso la versión de python elegida será la 3.9:

```
conda create —name mienv python=3.9
```

Para activar este entorno:

```
conda activate mienv
```

Para desactivar el entorno:

```
conda deactivate
```

Instalación de paquetes

La instalación de paquetes puede hacerse a través de *conda install* o utilizando el gestor de paquetes *pip*. Por ejemplo, vamos a instalar pytorch:

```
pip install torch torchvision torchaudio
```

Podemos probar ahora que se ha instalado correctamente de la siguiente manera:

Versiones de los paquetes

Es recomendable a la hora de instalar paquetes indicar la versión, por ejemplo *pip install torch==2.0.1*

Instalación de paquetes

También podemos ver la lista de paquetes instalados:

```
conda list
```

También es interesante saber una lista de todos los entornos que tenemos disponibles: `conda env list`