

Guía de ejercicios

Dr. Pedro Julián¹
Ing. Diego Gigena Ivanovich²
Ing. Nicolás Rodríguez³

Instalación de las herramientas

Anaconda

Descargar el instalador de Anaconda de la página <https://www.anaconda.com/>, y crear un nuevo entorno.

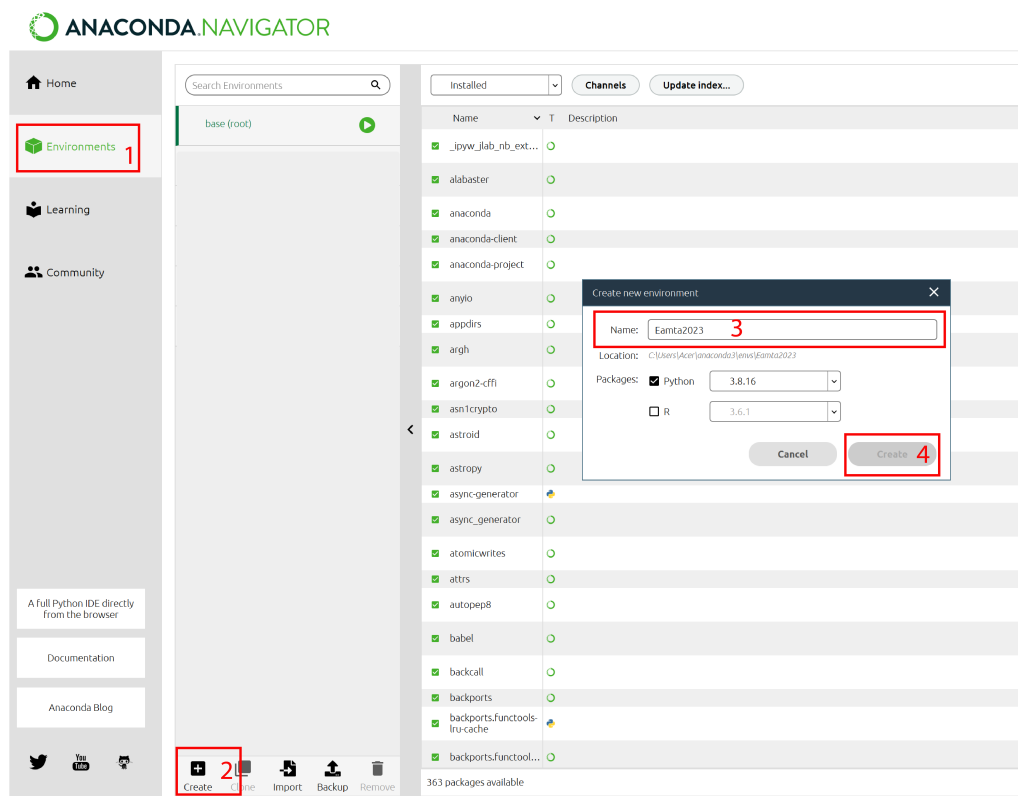


Figura 1: Creacion del entorno

Instalar la terminal PowerShell Prompt (Windows), y una vez abierta, instalar los siguientes paquetes:

- Numpy.
- Matplotlib.

Pytorch

Instalar la librería Pytorch. En la sección "Getting Started" vamos a encontrar una tabla interactiva que genera el comando necesario para instalar el paquete con las configuraciones necesarias de acuerdo a nuestro sistema operativo. en la figura 3, la configuración no contempla aceleración por GPU con paquete CUDA. En caso de contar con una GPU NVIDIA, se puede optar por instalar con la opción CUDA habilitada, teniendo en cuenta la versión que se ajusta a la GPU. Una vez generado el comando, copiarlo y ejecutarlo en el PowerShell de Anaconda.

¹pedro.julian@silicon-austria.com

²diego.gigena-ivanovich@silicon-austria.com

³nicolas-daniel.rodriguez@silicon-austria.com

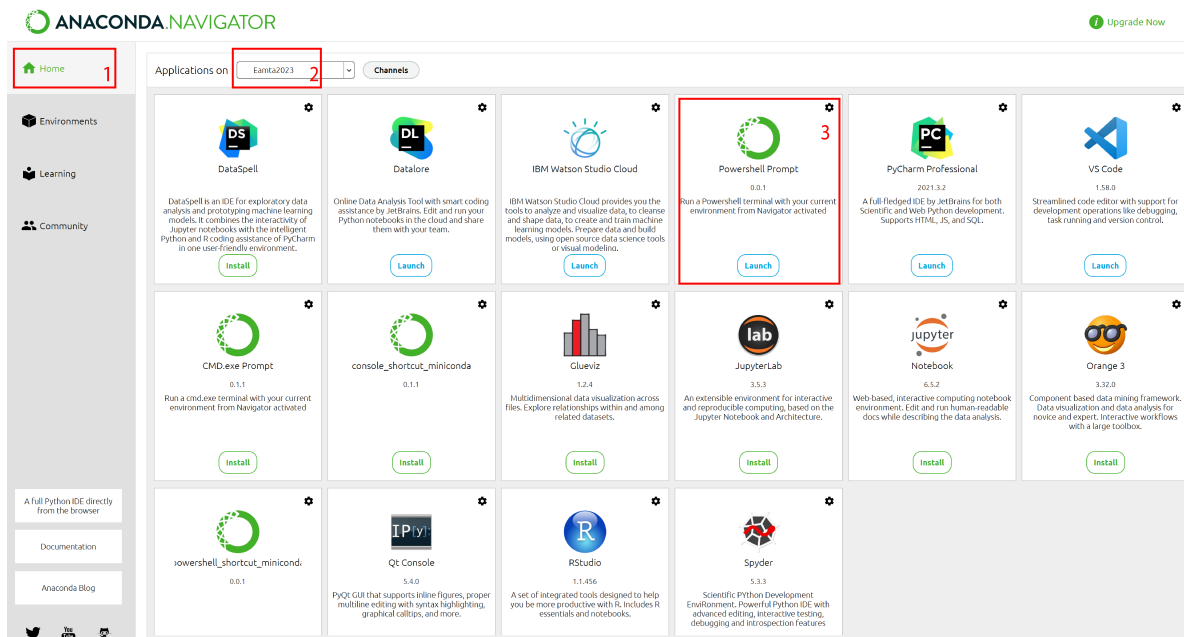


Figura 2: instalación de la terminal

GET STARTED

Select preferences and run the command to install PyTorch locally, or get started quickly with one of the supported cloud platforms.

[Start Locally](#)
[PyTorch 2.0](#)
[Start via Cloud Partners](#)
[Previous PyTorch Versions](#)
[Mobile](#)

START LOCALLY

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also **install previous versions of PyTorch**. Note that LibTorch is only available for C++.

PyTorch Build	Stable (1.13.1)	Preview (Nightly)
Your OS	Linux	Mac Windows
Package	Conda Pip	LibTorch Source
Language	Python	C++ / Java
Compute Platform	CUDA 11.6	CUDA 11.7 ROCm-5.2 CPU

Run this Command:

```
conda install pytorch torchvision torchaudio cpunoly -c pytorch
```

Shortcuts

- Prerequisites
 - Supported Windows Distributions
 - Python
 - Package Manager
- Installation
 - Anaconda
 - pip
- Verification
- Building from source
 - Prerequisites

Figura 3: instalación de la librería Pytorch

Configuración del IDE

Es aconsejable la instalación de un IDE de python. Se recomienda el uso de PyCharm Community, pero queda a elección del alumno.

El archivo comprimido "ML4VLSI.zip" contiene la estructura del proyecto con el que se va a trabajar, junto con un ejemplo básico de una red de tipo perceptrón multicapa, y los loops necesarios para entrenamiento, en este caso con el dataset MNIST.

```
ML4VLSI
├── models
└── src
```

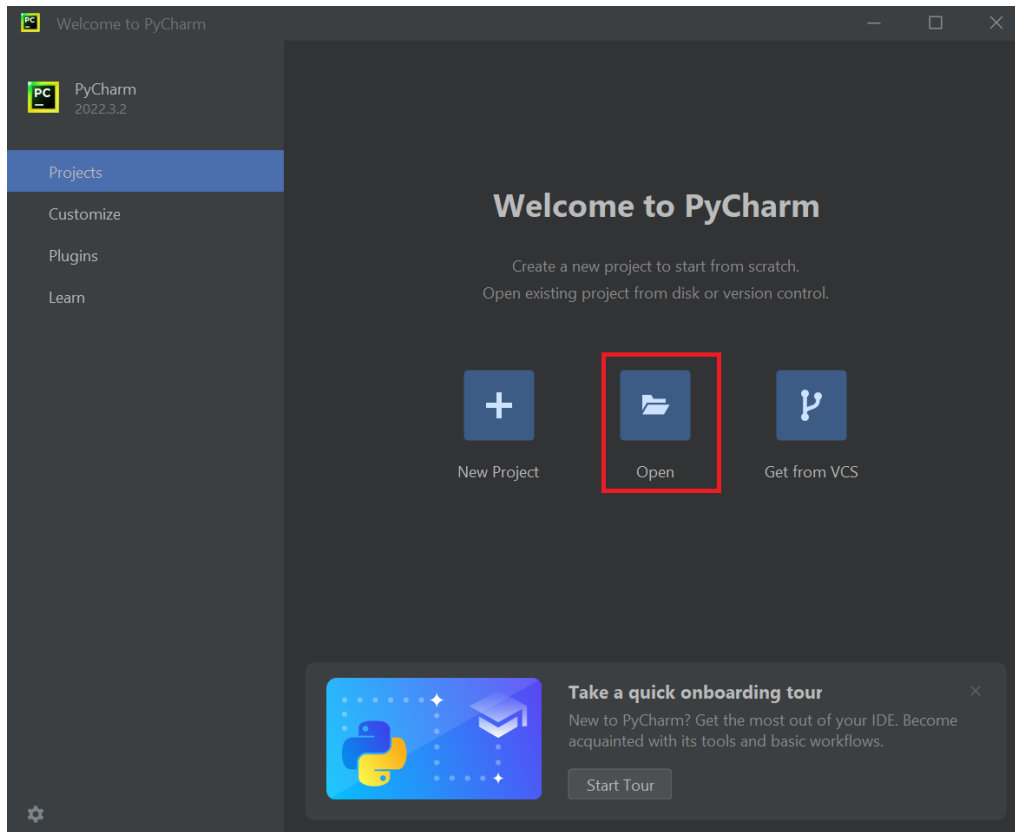


Figura 4: Abrir carpeta como proyecto de PyCharm

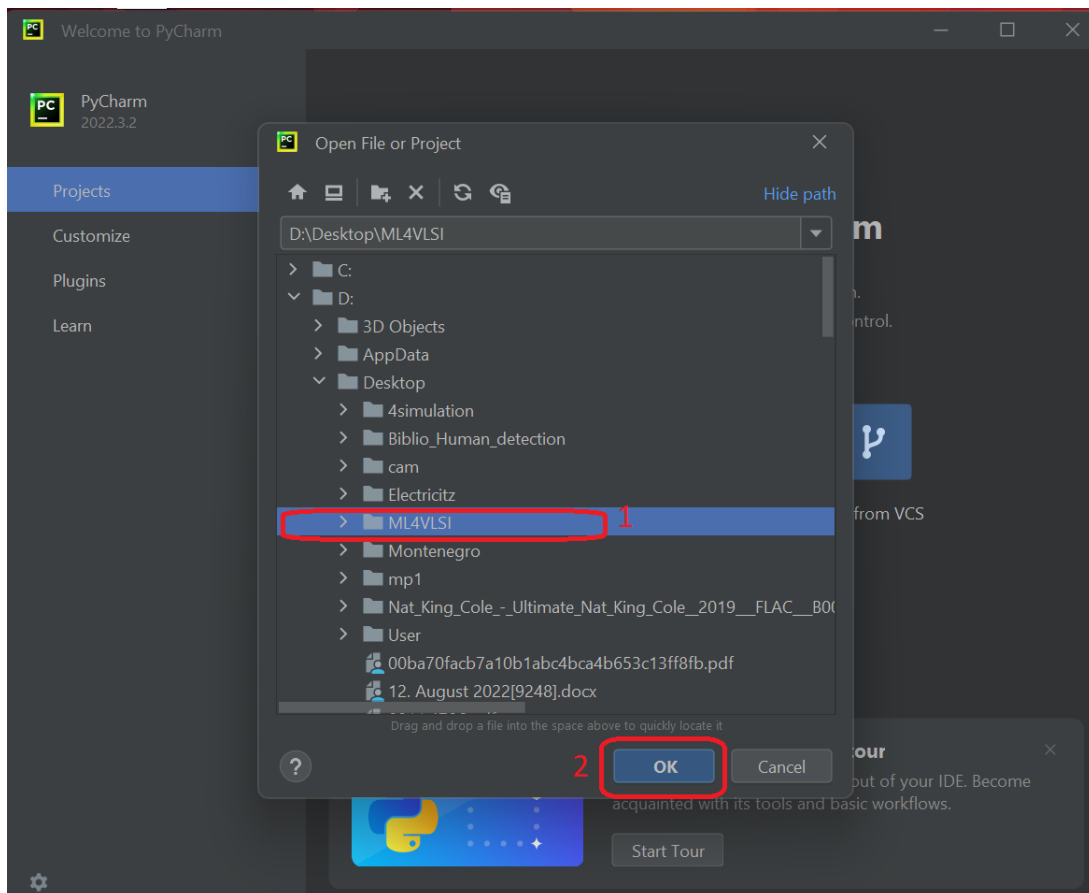


Figura 5: Seleccionar la carpeta del proyecto ejemplo descomprimida

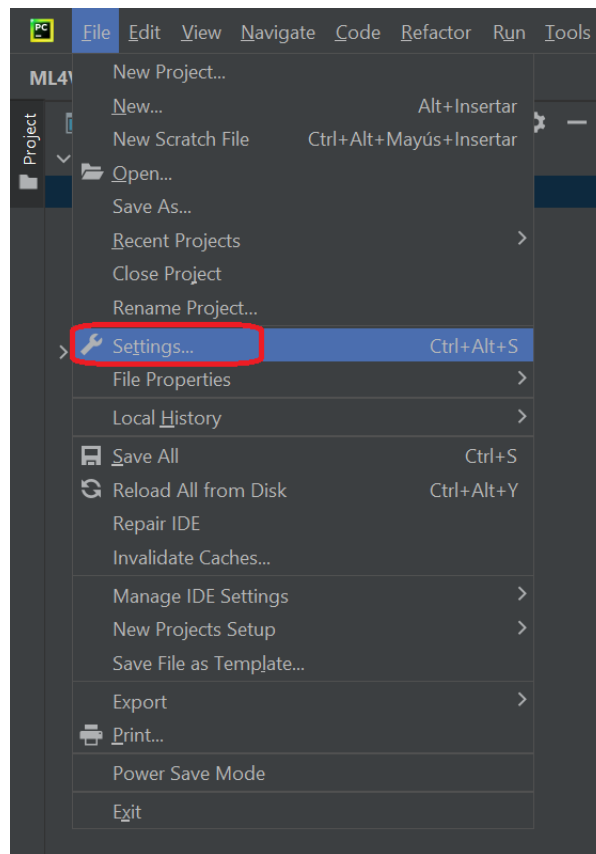


Figura 6: Una vez abierta la carpeta, abrimos la configuracion del IDE

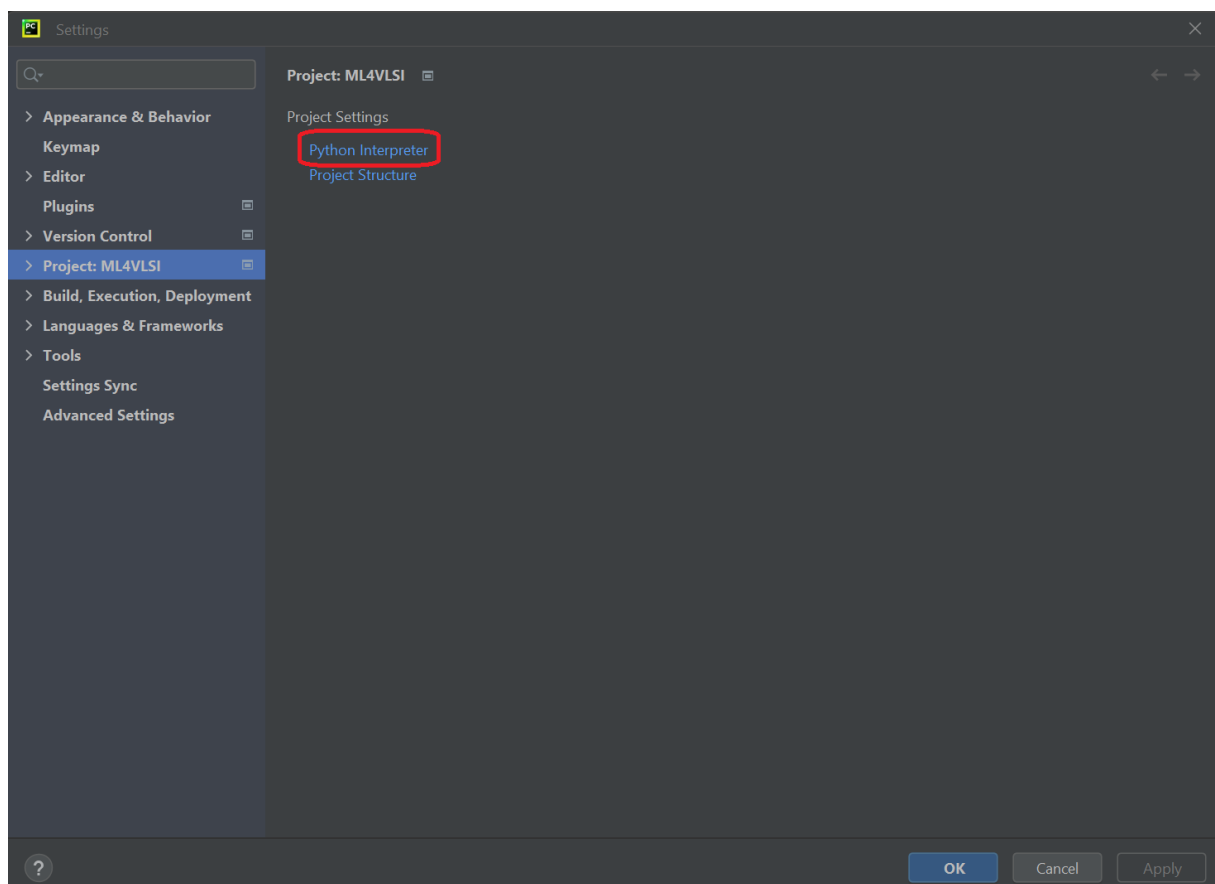


Figura 7: Se debe configurar el entorno de Python

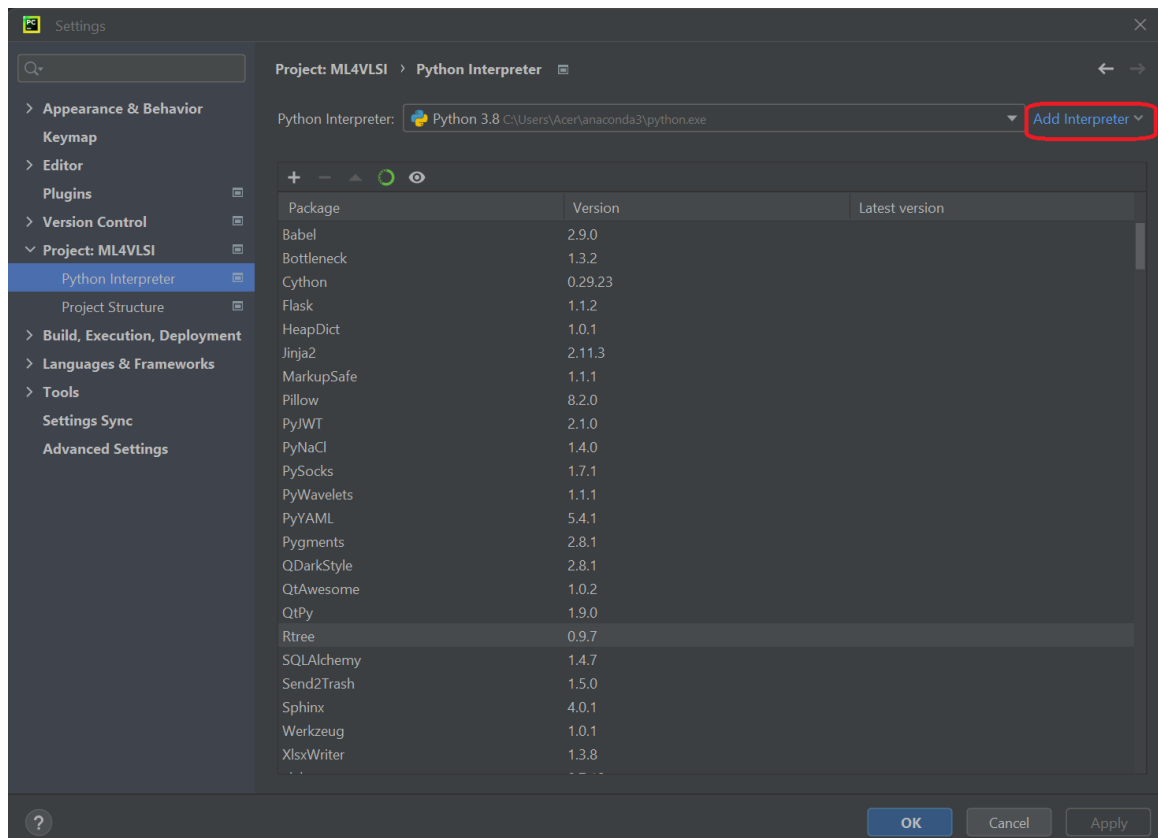


Figura 8: Agregar nuevo intérprete

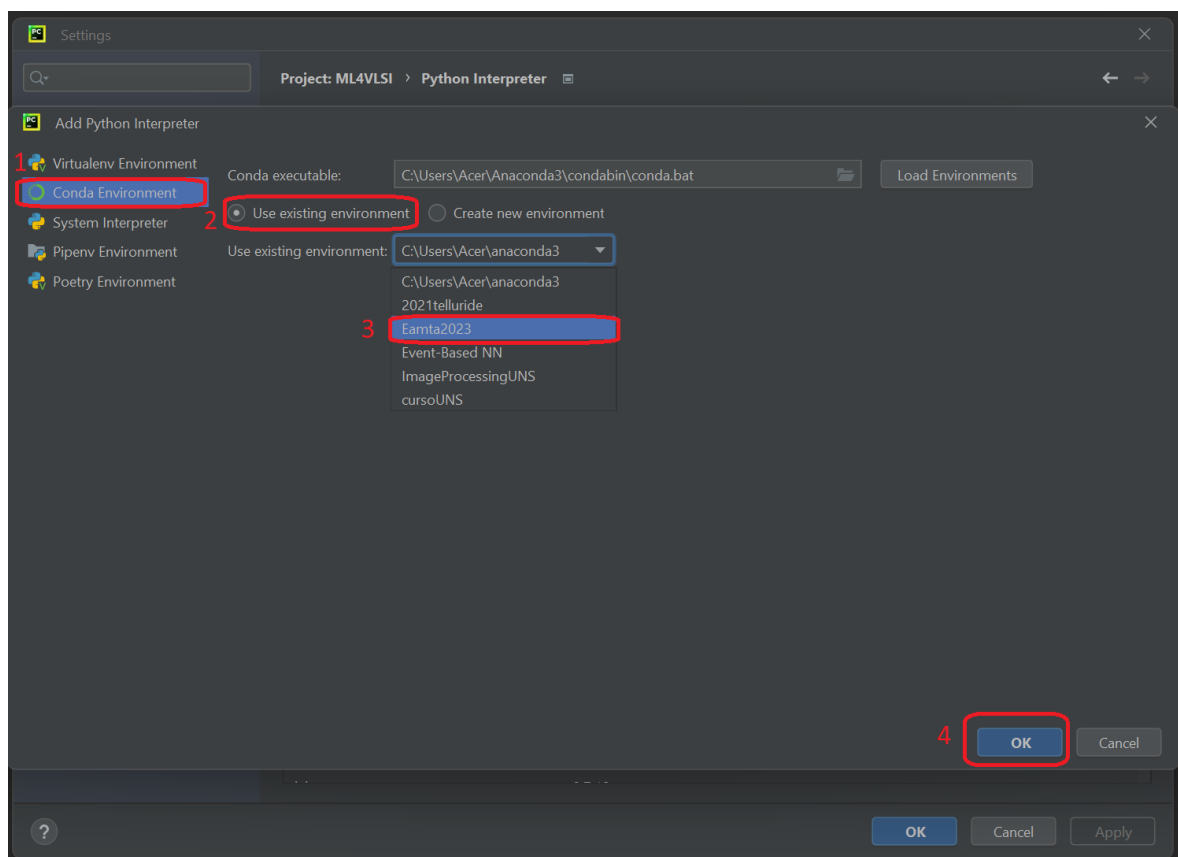


Figura 9: Seleccionar el entorno de Anaconda creado anteriormente

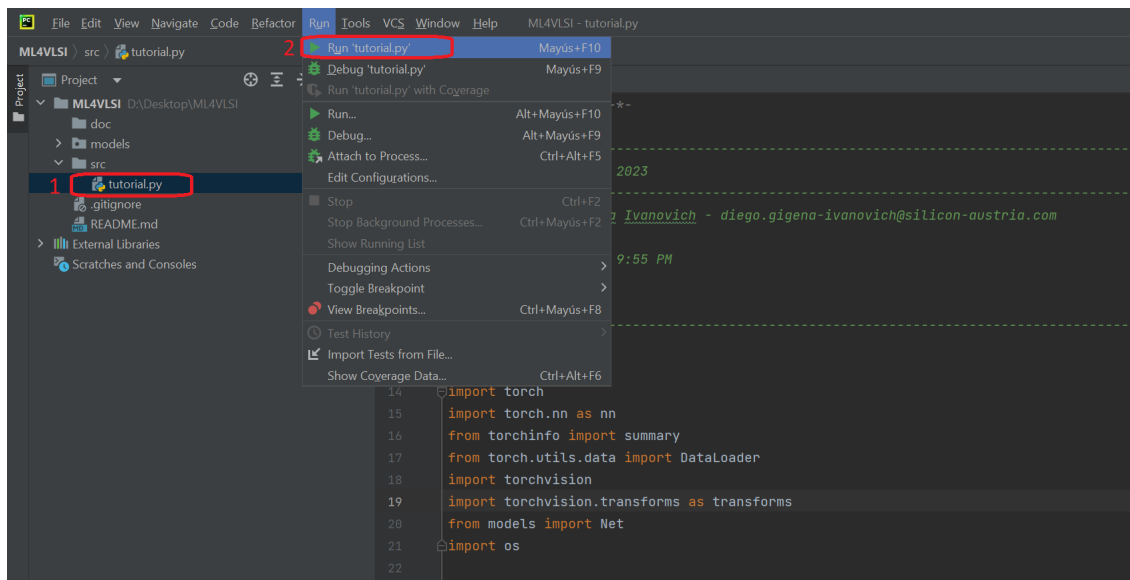


Figura 10: Ejecutar el tutorial

Ejercicio 1

Implementar una red convolucional lo mas reducida posible para clasificar el dataset CIFAR10, utilizando de base el ejemplo provisto, de manera de obtener una precision de al menos 85 % en validación.

Ejercicio 2

Utilizando los datos de tabla y considerando una cuantización de 8bits para pesos y entradas, dimensionar una unidad MAC (Fig. 11), y estimar el tamaño.

Cell	Height (μm)	Width (μm)	Power (μW / MHz)	Leakage Power (pW)
AND2	1.8	1	0.0021	14.23
OR2	1.8	1	0.0026	13.56
NAND2	1.8	0.8	0.00095	8.2
NOR2	1.8	0.8	0.00095	8.88
INV	1.8	0.6	0.00074	5.14
MUX2	1.8	2.2	0.0036	30.03
DFFQ	1.8	4.2	0.0038	44.65

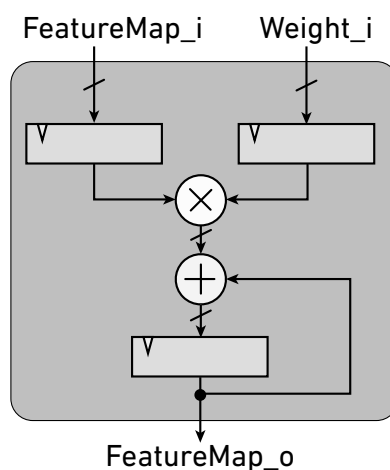


Figura 11: Processing Element

1. Ejercicio 3

a)

Calcular el número de productos y sumas necesarios para procesar cada capa de la red entrenada en el ejercicio 1, y teniendo en cuenta el tamaño estimado de la unidad de cómputo (MAC) del ejercicio anterior, realizar una estimación de:

- Cantidad de MACs necesarias para procesar la red.
- Cantidad de ciclos de clock necesarios.
- Area total.
- Consumo de potencia, considerando una frecuencia de clock de $100MHz$.

b)

A partir de los resultados obtenidos en el ejercicio anterior, si consideramos que el área máxima disponible es de $1,5mm^2$, estimar:

- Cantidad de MACs que podrían fabricarse.
- Cantidad de ciclos de clock necesarios.
- Recalcular el consumo de potencia.