

# Python Setup

## Windows

For this course we will be working with **Python** and use the Python library **PyTorch** to create machine learning models. These instructions will help to set up virtual environments with **Miniconda** that include Python 3.10 and all necessary libraries. You will also find instructions on how to check if your device has a **CUDA-capable GPU** and, if so, how to install **CUDA** and **PyTorch** such that you can use your GPU to train machine learning models. Finally, we will be working in **Jupyter Notebooks** to write our code. The last section will cover how to install and use Jupyter Notebook. Let's start with installing Miniconda and creating a virtual environment.

## Contents

<b>1</b>	<b>Python</b>	<b>3</b>
<b>2</b>	<b>CUDA</b>	<b>5</b>
<b>3</b>	<b>Python Packages</b>	<b>7</b>
<b>4</b>	<b>Jupyter Notebook</b>	<b>8</b>
<b>5</b>	<b>Final Check</b>	<b>9</b>

# 1 Python

We are going to make use of Miniconda and virtual environments to make sure everyone will have the same Python setup (Python version and packages). So let's start with downloading and installing Miniconda:

1. Go to <https://docs.conda.io/en/latest/miniconda.html>
2. Scroll down to download the Latest Miniconda installer for Windows.
3. When the download has finished, start the installer.
4. You can keep the default options selected when installing Miniconda.
5. When the installation has finished, locate **Anaconda Prompt** on your machine. You can stick it to your taskbar for easy access. For now start it.

Next, we are going to create a [virtual environment](#) which we can use for the course. This environment will have an isolated Python version and packages. You can create many different virtual environments that have different versions of Python and packages. This way you keep these versions separated and you do not end up with the problem that certain packages mess up the installation of others.

1. In the Anaconda Prompt, type and execute the following command:
  - `conda create -n <env_name> python=3.10 pip`
  - Where `<env_name>` is the name for your virtual environment that you can decide yourself.
  - This command will create a virtual environment with Python 3.10 and also pip.
  - We can use pip to install other python packages.
  - During the creation of the virtual environment a number of packages will be listed with the question if you want them to be installed. Type 'y' to allow this and proceed with the installation.
2. After your virtual environment has been installed successfully, activate it by executing the following command:
  - `conda activate <env_name>`
  - You need to activate your virtual environment every time you want to work with it.
3. More environment commands:
  - Deactivate environment: `conda deactivate`
  - List environments: `conda env list`

- Remove environment: `conda env remove --name <env_name>`

So now you have managed to install Miniconda and use it to create a virtual environment with Python 3.10 and pip. Next we are going to check if you have a CUDA-capable GPU.

## 2 CUDA

In this section, we will explain how you can install CUDA and PyTorch in order for you to make use of your device's GPU when training machine learning models. First, you will need to make sure your device has a CUDA-capable GPU.

1. Look up your device's specifications and specifically information on your GPU.
2. Visit the following link and find out if your GPU is CUDA-capable:  
<https://developer.nvidia.com/cuda-gpus>.
3. If your device does not have a CUDA-capable GPU then skip the remainder of this section, otherwise continue with the next steps.

So your device has a CUDA-capable GPU. Now you are going to install CUDA in your virtual environment and at the end we will install the correct PyTorch version such that it can be used together with CUDA to let you train machine learning models on your GPU.

1. Open the Anaconda Prompt and activate your virtual environment.
2. Next, execute the following commands:
  - `conda install cuda -c nvidia` (if asked to install certain packages, answer with 'y').
  - `conda install cudatoolkit` (if asked to install certain packages, answer with 'y').
3. To check if CUDA is installed correctly execute the following command:  
`nvcc --version`.

CUDA is installed. Next we are going to install the correct PyTorch version:

1. Visit the following link:  
<https://pytorch.org/get-started/locally/>
2. Then select the following options:
  - **PyTorch Build:** Stable
  - **Your OS:** your OS
  - **Package:** Conda
  - **Language:** Python
  - **Compute Platform:** CUDA 11.8
3. In the last row named **Run this Command** copy the command.
4. Open the Anaconda Prompt and activate your virtual environment.

5. Paste the command and execute it.

You have now installed CUDA and the correct PyTorch packages. Next we use pip from within your virtual environment to install some necessary Python packages for this course.

### 3 Python Packages

Now we are going to install the necessary Python packages in our virtual environment. Together with this PDF we provided a text file named 'requirements.txt'. Make sure you have downloaded this file and know where it is located on your device.

1. Open the Anaconda Prompt and activate your virtual environment.
2. On your device, locate the requirements.txt file.
3. Copy the exact path to this file.
4. In Anaconda Prompt execute the following command using the copied path:
  - `pip install -r <path/to/requirements.txt>`
5. From within your virtual environment, you can also navigate to the specific directory where the file is located:
  - `cd <path/to/requirements.txt>`
  - `pip install -r requirements.txt`

You have now installed Python and some necessary packages. Almost done! To be able to work with Python we will help you install Jupyter Notebook in the next section.

## 4 Jupyter Notebook

Finally, we are going to install Jupyter Notebook. This is a application that you can use to create documents called notebooks. In these notebooks you can write code but also write text or add images. You can even split up your code in smaller code snippets and re-run the individual snippets without having to re-run your entire code. The notebooks can be saved and shared with others. You can also upload Jupyter Notebooks to Google Colab. This may come in handy if you do not have a CUDA-capable GPU or simply want to make use of Google's GPU.

1. In your still active virtual environment, execute the following command:
  - `pip install notebook` (this step can be skipped if you followed the steps in the previous section).
2. After installation, to open Jupyter Notebook execute the following command:
  - `jupyter notebook`
  - This will open a new tab in your web browser with an overview of your current working directory.
  - You can then navigate to a folder in which you want to store Jupyter Notebooks or just start with a new one.
3. To create a new notebook do the following:
  - Locate the **new** button in the top right corner of the Jupyter page.
  - Click on Python 3 (ipykernel).
  - A new tab will be opened which corresponds to your Python Jupyter Notebook (.ipynb).
4. The previously opened Jupyter Notebook can be edited, named and closed.
5. You can save your notebook by clicking on the save icon in the top left corner.
6. In case you restarted your computer and you want to get back to your notebook repeat these steps:
  - (a) Open a terminal.
  - (b) execute `conda activate <env_name>`.
  - (c) Execute `jupyter notebook`.
  - (d) Locate your .ipynb file and open/activate it.



## 5 Final Check

To check if you have successfully set up everything for this course, you can perform the following steps:

1. Activate your virtual environment.
2. Execute `python -V` to check your python version (it should be 3.10).
3. Execute `nvcc --version` in case you have a CUDA-capable GPU to check if you installed CUDA correctly.
4. Execute `jupyter notebook` to open a Jupyter Notebook.
5. Inside a Jupyter Notebook execute the following commands:
  - `import numpy`
  - `import matplotlib`
  - `import torch`
  - `import torchvision`

Next to this PDF and the 'requirements.txt' text file, we also added a Jupyter Notebook. To check if everything is set up correctly we ask you to open this notebook and run the code inside. If you do not encounter any errors you have correctly set up your environment and you will be able to perform the exercises during the practicals. Otherwise, please send us a message. In case you encountered problems with installing CUDA, do not worry. You can skip the CUDA section and continue with the remaining sections. It is not necessary for this course that you own a device with a CUDA-capable GPU. Running the code from the practicals on a CPU will also work.