

Setting Up The Working Environment

This guide has 3 parts:

1. [Setting up Anaconda environment and Python \(+PyTorch\) locally.](#)
2. [Setting up PyCharm to work in the Anaconda environment.](#)
3. [Setting up Google Colab with GPU.](#)

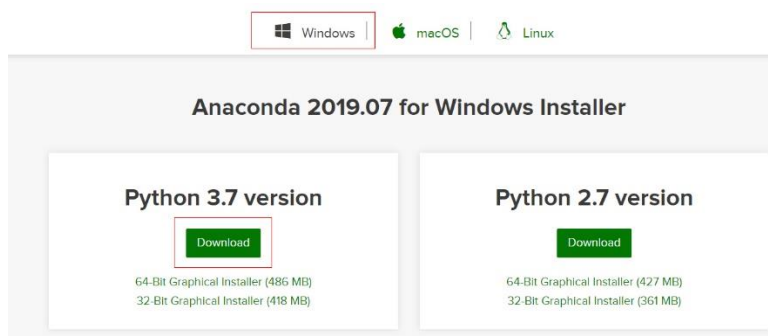
The guide may seem long, but it is mostly images, so don't be afraid 😊

PART 1 - SETTING UP ANACONDA ENVIRONMENT AND PYTHON (+PYTORCH) LOCALLY

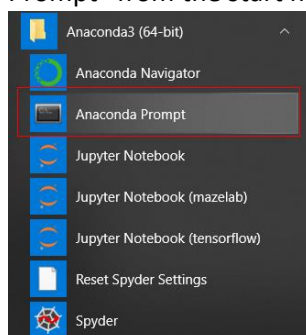
In this part, we are going to download Anaconda Navigator with Python 3, create a new virtual environment for the course and install the required libraries.

Windows (But it is the same for Linux, just download the correct installer)

1. Go to <https://www.anaconda.com/distribution/> and download the installer. Make sure to pick Windows and the Python 3 version (if you are on Linux, you can also “curl” or “wget” to download the “.sh” file).



2. Launch the installer and follow the instructions on-screen. Important note: the path in which you install Anaconda **MUST NOT** include any characters other than English (if the username of your Windows is in Hebrew, please pick an English-only path).
3. Once installation is done, we need to create a new environment. Launch “Anaconda Prompt” from the start menu (Linux users: just open the Terminal).



4. In the command line you will see the name of the current environment (the basic one is called “base”) on the left. Type “`conda create --name env_name`”, where “`env_name`” is the name of the environment. Pick any name you want (“torch”, “ml”, “deep_learn”, “anam”....) and press Enter.

```
(base) C:\Users>conda create --name tensorflow
```

5. After the process is done, you can activate the new environment by typing “`conda activate env_name`”. You will see the name in the parenthesis change.

```
(tensorflow) C:\Users>conda activate tensorflow
```

6. Now we are going to install new libraries. Activate the new environment. All the libraries and their installation commands can be found here: <https://anaconda.org/> (if in the future you want to install other libraries, start your search there. There are libraries for Computer Vision, Speech, Reinforcement Learning...). Here is the list of libraries you need to install. Run the commands in Anaconda Prompt while the new environment is activated.

Library	Command to Run
Jupyter Notebook	conda install -c conda-forge notebook
Numpy	conda install -c conda-forge numpy
Matplotlib	conda install -c conda-forge matplotlib
Pandas	conda install -c conda-forge pandas
Scipy	conda install -c anaconda scipy
Scikit-Learn	conda install -c conda-forge scikit-learn
Seaborn	conda install -c conda-forge seaborn

7. Installing PyTorch – we are now going to install PyTorch. It has a CPU version and a GPU version (which also includes the CPU version). If you know that you have an Nvidia GPU that supports CUDA, you need to install the CUDA drivers. Follow this guide:

<https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html>

A list of supported GPUs for deep learning can be found here:

<https://developer.nvidia.com/cuda-gpus>

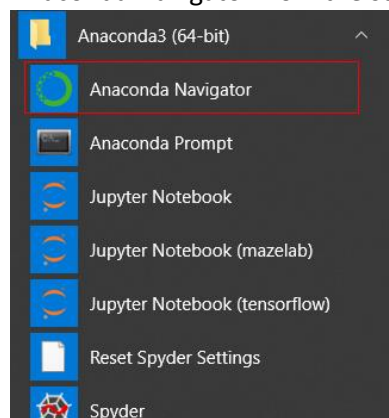
To customize your installation, you can visit PyTorch homepage as they have great UI:

<https://pytorch.org/get-started/locally/>

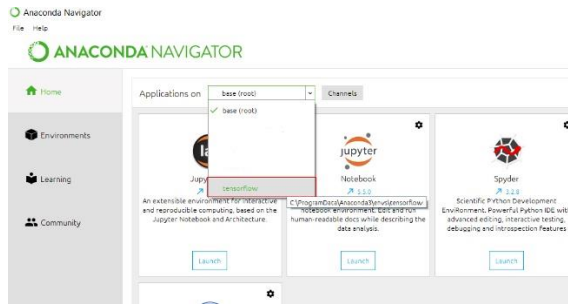
Finally, here are the commands you need to run:

Library	Command to Run
PyTorch CPU	conda install pytorch torchvision cpuonly -c pytorch
Pytorch GPU (only if you have one!)	conda install pytorch torchvision cudatoolkit=10.0 -c pytorch

8. Great, we now have everything installed and we can launch Notebooks locally. Launch Anaconda Navigator from the start menu.



9. Switch to the environment you created from the scroll menu.

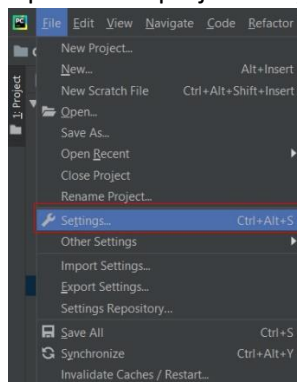


10. Now you can launch Jupyter Notebook which will open in your default browser. Now you can download the tutorials and homeworks and run them locally.

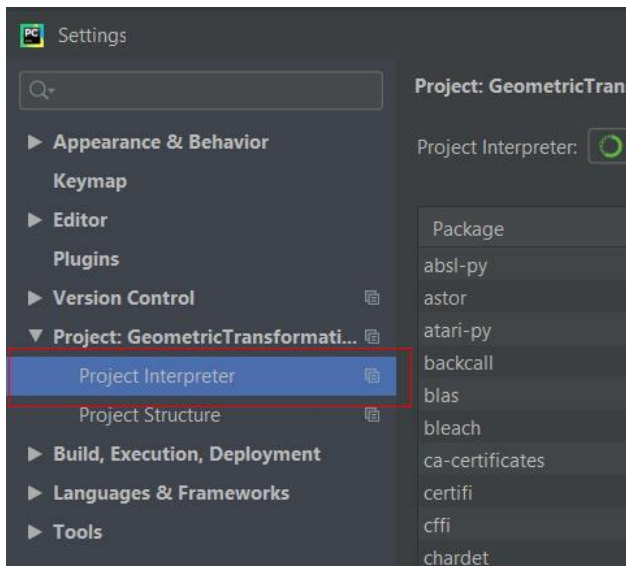
PART 2 - SETTING UP PYCHARM TO WORK IN THE ANACONDA ENVIRONMENT

If you want to work in a regular IDE other than the notebook, you need to set-up the IDE's interpreter to the one in the new environment you created. This is because you will want it to recognize all of the libraries you use. You can use any IDE that supports Python (such as VisualStudio), but we (and most of the world) recommend PyCharm. You can download PyCharm community version (free) from <https://www.jetbrains.com/pycharm/>. Note that as a Technion student you can get a license (1 year) for the Professional version, you just need to register with your @campus mail at JetBrains's website. Once you downloaded and installed PyCharm, you need to locate the directory of the environment and pick "python.exe" as your default interpreter. Working with PyCharm will allow to debug your code the "old-fashioned" way (putting a red dot where you want the code to stop).

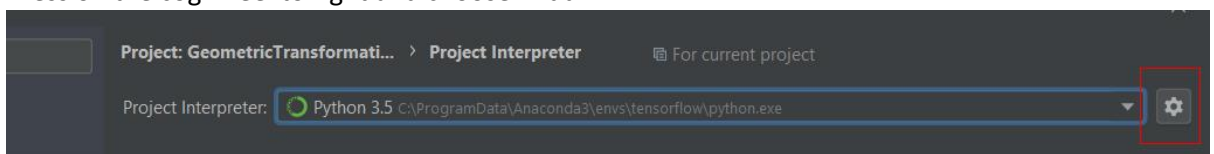
1. Open a new project in PyCharm and go to "File" -> "Settings".



2. Inside the settings, navigate to "Project Interpreter".



3. Press on the cogwheel to right and choose “Add..”.

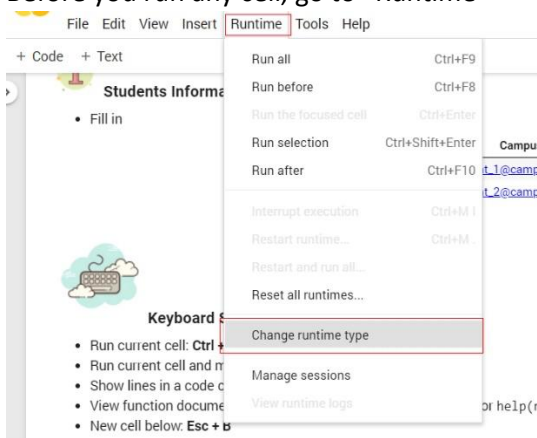


4. You will now have to find “python.exe” inside the environment directory you want to work in. It is usually in `C:\ProgramData\Anaconda3\envs\env_name\python.exe`.
5. Once done, press “OK” and you can run your code.

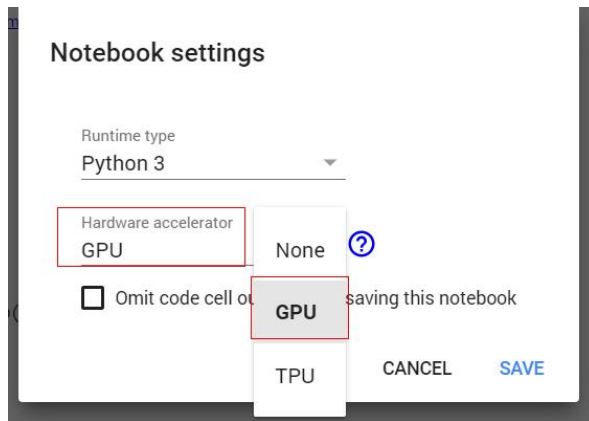
Note: in this course, we only accept .ipynb files, so make sure to copy your code back to the original Jupyter Notebook.

PART 3 - SETTING UP GOOGLE COLAB WITH GPU

1. Go to <https://colab.research.google.com>, and open the notebook you want to work with (can be from GitHub, your Google Drive or just upload from your drive).
2. Before you run any cell, go to “Runtime” -> “Change runtime type”.



3. Pick GPU as the hardware acceleration type and click “Save”.



4. You can now run calculations on a GPU, but remember that your run is time-limited (12 hours).