

1 Software

The exercises are prepared in `python` using `jupyter-notebooks`. The following software/python packages will be needed throughout the course. All software package are freely available for linux, mac, and windows.

- `python3` (<https://www.python.org>)
- `jupyter-notebooks` (<https://jupyter.org>) – python work environment
- `ase` (<https://wiki.fysik.dtu.dk/ase/index.html>) – atomic simulation environment
- `numpy` (<https://numpy.org>) – numerical tools
- `matplotlib` (<https://matplotlib.org>) – visualization
- `pytorch` (<https://pytorch.org>) – deep learning package

2 Installation

If you are already working with `python` and know how to install or update additional packages, simply make sure that the packages listed above are available on your computer.

If you are not a `python` expert the easiest and most convenient way to install everything in a consistent manner is through `anaconda`.

2.1 Anaconda

The individual edition of Anaconda is open source and can be downloaded and installed from <https://www.anaconda.com/products/individual>. Installers are available for linux, mac, and windows. You can then either work with the Anaconda Navigator (<https://docs.anaconda.com/anaconda/navigator>) or with `conda` on the command line (<https://docs.conda.io/projects/conda/en/latest/>).

`Conda` works with so-called environments (<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>) and it is usually recommended to create and use separate environments for separate projects. Let's create a new environment for the exercises installing the packages listed above. For this, we are working on the command line, using:

```
> conda create -n mltutorial python=3.10
```

where `mltutorial` is the name of the environment. Feel free to choose a different name for your environment. Now we want to activate the environment with:

```
> conda activate mltutorial
```

Once the environment is activated, you can install the required packages with `conda` while `ase` needs to be installed using `pip` within the environment:

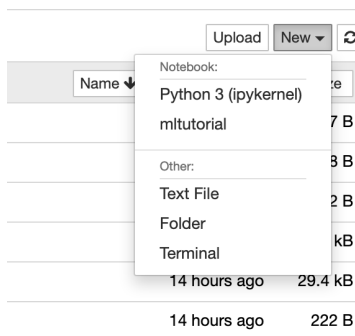


Figure 1: Starting a new jupyter notebook.

```
> pip install --upgrade ase
> conda install -c conda-forge jupyter notebook ipython pytorch torchmetrics
→ scikit-learn jupyter_contrib_nbextensions numpy matplotlib -y
```

Then you should register this new environment so that it's available to jupyter under the name "mltutorial"

```
> python -m ipykernel install --user --name mltutorial --display-name "mltutorial"
```

and enable some essential extensions for the notebooks

```
> jupyter nbextension enable codefolding/edit
> jupyter nbextension enable collapsible_headings/main
> jupyter nbextension enable codefolding/main
> jupyter nbextension enable comment-uncomment/main
```

More informations about the extensions can be found at https://github.com/ipython-contrib/jupyter_contrib_nbextensions.

Congratulations! Now you are all set for the exercises :)

2.2 Checking your installation

Launch a `jupyter-notebook` either from the Anaconda Navigator or from the command line:

```
> jupyter-notebook
```

Make sure that you are doing this within your newly created environment! Then go to **New** and click `mltutorial` under **Notebook**, see Fig. 1 Now, we want to check that all our packages are installed by importing them in our notebook. In the field, type:

```
import sklearn
```

and hit **Shift+Enter**. If you are not getting any error messages, everything is fine. Do the same with:

```
import ase
import numpy
import matplotlib
import torch
```

Still no error messages? Perfect, now you are really all set for the exercises!

3 Visual Studio Code

Depending on your programming skills, you might prefer to work directly with python scripts instead of jupyter notebooks. In particular for larger projects, it is often beneficial to work in an integrated development environment (IDE). **Visual Studio Code** (<https://code.visualstudio.com>) is a free source code editor, available for linux, mac, and windows, that, in addition to providing an enhanced editing environment, also provides tools for debugging and version control. There are also extensions available that allow you to work with jupyter notebooks.

4 Introduction to python and pytorch

We have compiled a set of very basic python functionalities in a jupyter notebook that you can use to familiarize yourself with both python and jupyter notebooks. To go through the tutorial, start `jupyter-notebook` and open the file `python-intro.ipynb`. In this file, you will also find download links to two `pytorch` tutorials which explain the two important concepts of tensors and automatic differentiation.