Universität
Bremen

Cognitive Systems Lab

Advanced Machine Learning Tutorial

Software Tools Installation Guide

Abdul Haq Azeem Paracha, Anthony Mendil, Daniel Reich, Felix Putze, Kim Marie Lankenau, Rinu Elizabeth Paul,
and Prof. Dr.-Ing. Tanja Schultz

Bremen 17.04.2023

Outline

- → Which tools do you need?
- → How to install…
  - – Anaconda
  - – Python, PyTorch, and JupyterLab
- → Verify your software installation

Which tools do you need?

The AML tutorials will include theoretical and practical programming tasks. The latter requires a Python environment and PyTorch as machine learning package. Beside PyTorch you will need other additional packages. To manage your environment and packages, you can use Anaconda. In the tutorials you will use Jupyter Notebooks to work on the theoretical and practical tasks.

This installation guide will show you how to install the required software tools.
All tools at a glance:

1. Anaconda as environment/package management tool.
2. Python as software development environment.
3. PyTorch as package to develop machine learning solutions.
4. JupyterLab to manage and edit Jupyter Notebooks (software package).

The example installation will be done on a Windows system. The installation process for other systems will be referenced.

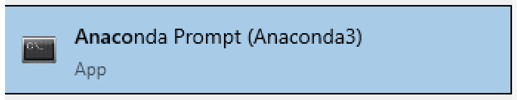
How to install…

Anaconda

Go to the Anaconda website and download the installer. Follow the instructions according to the description.
Important 1: Do not add Anaconda3 as PATH environment variable.
Important 2: Do not install the PyCharm IDE.
https://docs.anaconda.com/anaconda/install/windows/

For other systems: https://docs.anaconda.com/anaconda/install/

After having successfully installed Anaconda, you can click on start and search for the "anaconda prompt" command line interface.
To verify your installation follow the steps in section "Conda" at https://docs.anaconda.com/anaconda/install/verify-install/.



Hint: In the following we will use the term "conda".

Close all windows.

How to install…

Anaconda, Python, PyTorch, and JupyterLab

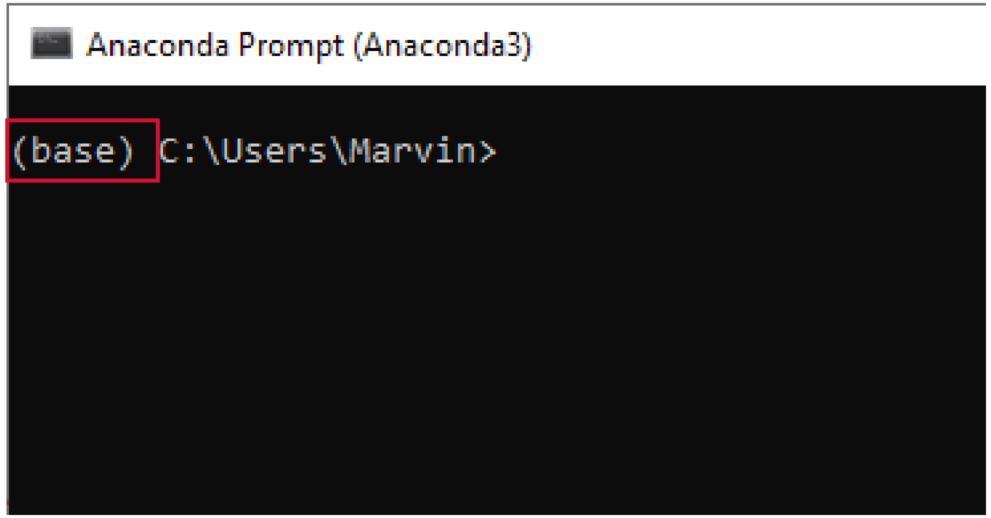
The instructions on the next slides will show you how to create a conda environment and install all required basic software packages with help of a requirements file. You will also verify your installation with an example Jupyter Notebook.

1. Create a folder called "AML_Tutorial". This will be your workspace for the tutorial tasks (this may be a git repository).
2. Copy both the "aml.yml" file and the "verify_installation.ipynb" file to this folder.

How to install…

Anaconda, Python, PyTorch, and JupyterLab

1. The “aml.yml“ file provides the Python, PyTorch, and JupyterLab versions as well as other required basic packages.
2. Open the “anaconda prompt“ command line interface as an administrator (Right click on Anaconda Prompt symbol → “Run as administrator”). The name of your currently active environment is given in brackets at the beginning of the line. Initially your (base) environment is active.

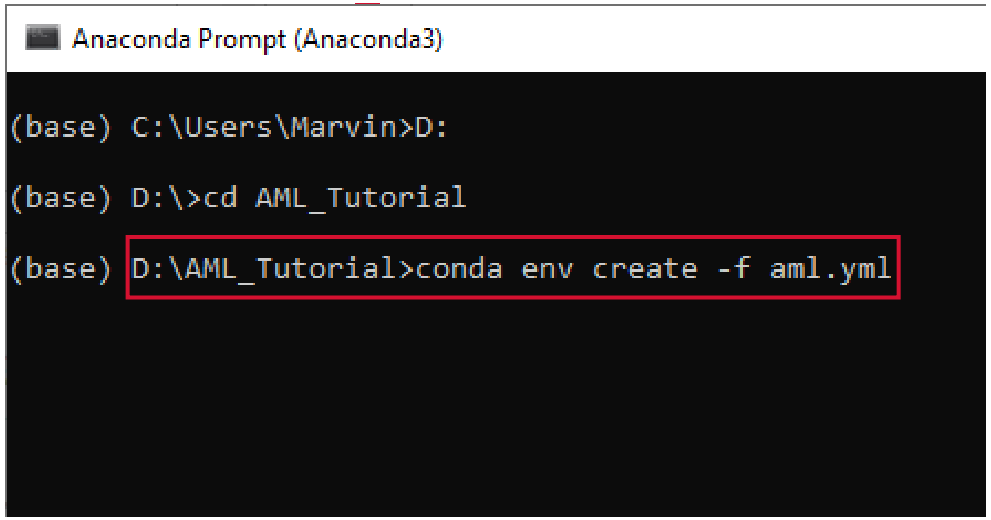


```
(base) C:\Users\Marvin>
```

How to install…

Anaconda, Python, PyTorch, and JupyterLab

3. Browse to the folder where you previously have stored the “aml.yml“ file. Use the following commands:
   “dir“: Show the content of the current directory.
   “cd ..“: Move to the parent directory.
   “cd xy“: Move into folder xy.
4. Start the installation with the command: conda env create -f aml.yml
   This will first create a new conda environment with the name “AML_Tut“ and afterwards install all software packages.

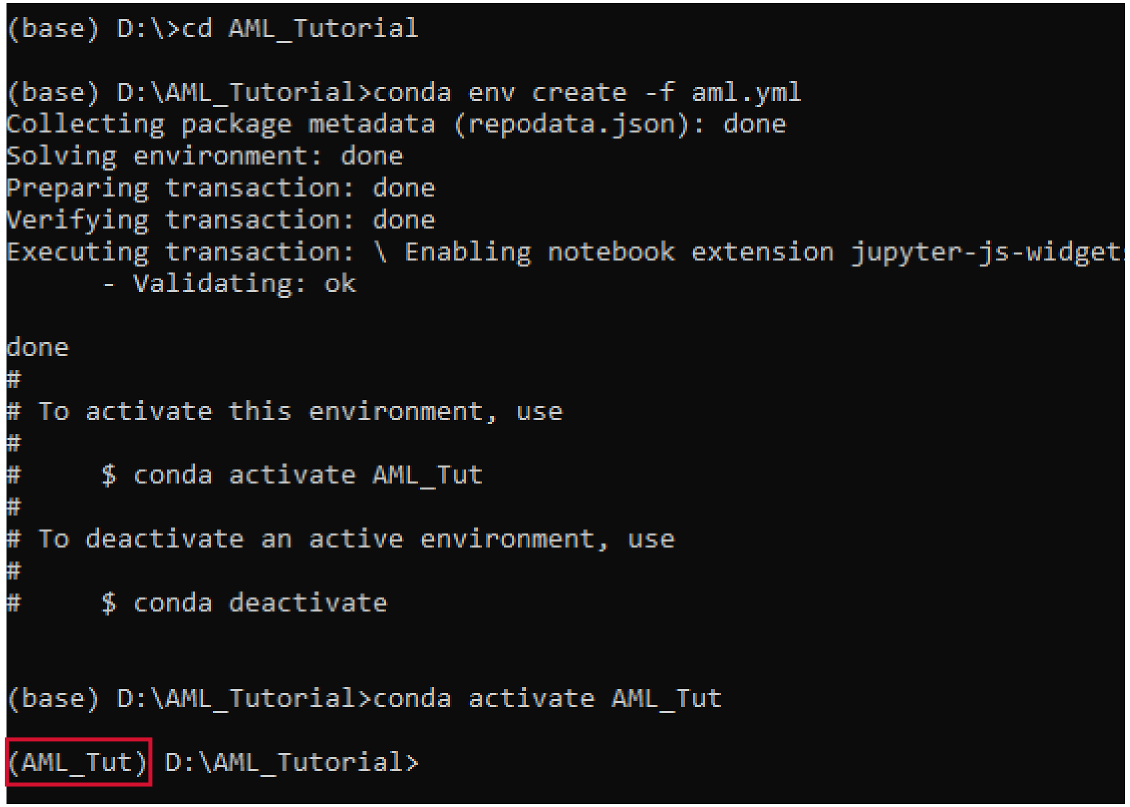


```
(base) C:\Users\Marvin>D:

(base) D:\>cd AML_Tutorial

(base) D:\AML_Tutorial>conda env create -f aml.yml
```

How to install…

Anaconda, Python, PyTorch, and JupyterLab

5. After the installation is done, you can activate the new environment with the command:
   conda activate AML_Tut
   The currently active environment is shown in the brackets (AML_Tut).
6. You can view all installed environments with the command: conda env list
7. You can view all installed packages in the currently active environment with the command: conda list



```
(base) D:\>cd AML_Tutorial

(base) D:\AML_Tutorial>conda env create -f aml.yml
Collecting package metadata (repodata.json): done
Solving environment: done
Preparing transaction: done
Verifying transaction: done
Executing transaction: \ Enabling notebook extension jupyter-js-widget
      - Validating: ok

done
#
# To activate this environment, use
#
#     $ conda activate AML_Tut
#
# To deactivate an active environment, use
#
#     $ conda deactivate


(base) D:\AML_Tutorial>conda activate AML_Tut

(AML_Tut) D:\AML_Tutorial>
```

How to install…

Anaconda, Python, PyTorch, and JupyterLab

Hint 1: During this course you may need to install additional software packages.
Information on how to manually install software packages into an existing environment can be found here:
https://docs.conda.io/projects/conda/en/4.6.1/user-guide/tasks/manage-pkgs.html

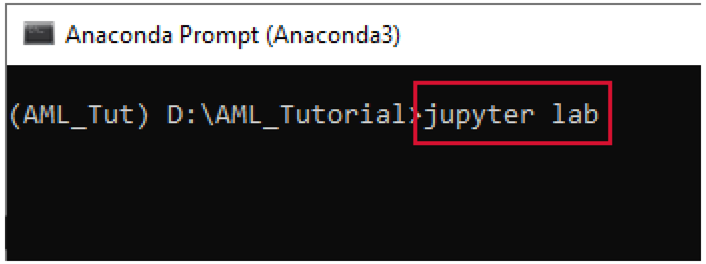
Hint 2: If you would like to manually create a new environment, you can find additional information here:
https://docs.conda.io/projects/conda/en/4.6.1/user-guide/tasks/manage-environments.html

Hint 3: This conda cheat sheet is quite helpful:
https://docs.conda.io/projects/conda/en/latest/user-guide/cheatsheet.html

Verify your software installation

Next, you have to verify the correct installation of all software tools. For this you will use the Jupyter Notebook file “verify_installation.ipynb“.
The code is based on https://github.com/Lightning-AI/lightning.

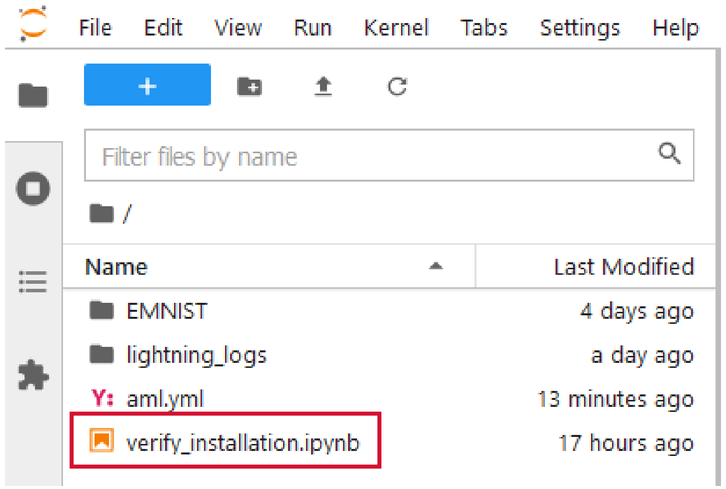
1. Open the “anaconda prompt“ terminal console and browse to the created “AML_Tutorial“ folder.
2. Activate your environment with the command: conda activate AML_Tut
3. Type in the command jupyter lab and press enter.
4. This will start JupyterLab in a new window/tab of your web browser.



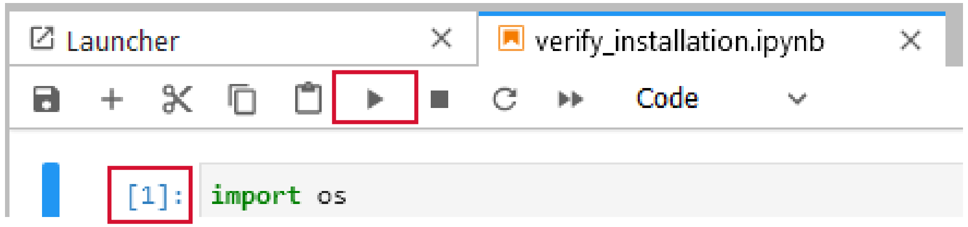
```
(AML_Tut) D:\AML_Tutorial>jupyter lab
```

Verify your software installation

5. From the main menu on the left-hand side you can open a Jupyter Notebook by double-clicking on the respective filename. Open the “verify_installation.ipynb“ Jupyter Notebook.
6. You can process each code cell after another by clicking either the run triangle in the top bar or pressing “ctrl+enter”.
7. The number in the square brackets will change to a star which indicates the running processing of the current cell.
   Important: The execution order and not the cell order is relevant (albeit they usually coincide). If, for example, a variable is declared in a cell above, but the cell wasn’t executed, the variable does not yet exist.



Name	Last Modified
EMNIST	4 days ago
lightning_logs	a day ago
aml.yml	13 minutes ago
verify_installation.ipynb	17 hours ago



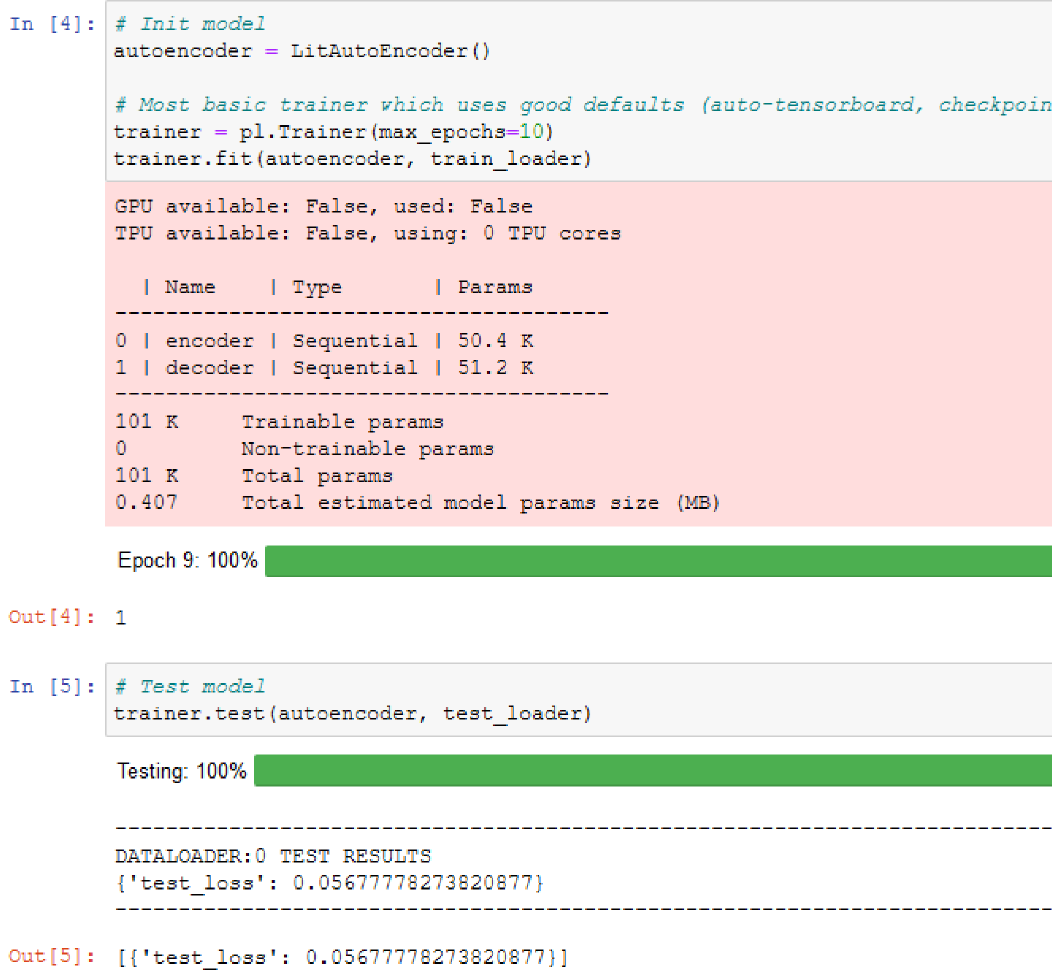
```
[1]: import os
```

Verify your software installation

Execute the full Jupyter Notebook.

The data download of EMNIST will take some time. If you encounter timeout errors, please re-run the cell. After the download will be completed, the local data will be automatically used the next time.

If you can run all code cells without any errors, your software installation was successful!



In [4]:

```
# Init model
autoencoder = LitAutoEncoder()

# Most basic trainer which uses good defaults (auto-tensorboard, checkpoin
trainer = pl.Trainer(max_epochs=10)
trainer.fit(autoencoder, train_loader)
```

```
GPU available: False, used: False
TPU available: False, using: 0 TPU cores

  | Name    | Type       | Params
---------------------------------------
0 | encoder | Sequential | 50.4 K
1 | decoder | Sequential | 51.2 K
---------------------------------------
101 K     Trainable params
0         Non-trainable params
101 K     Total params
0.407     Total estimated model params size (MB)
```

Epoch 9: 100%

Out[4]: 1

In [5]:

```
# Test model
trainer.test(autoencoder, test_loader)
```

Testing: 100%

```
--------------------------------------------------------------------------------
DATALOADER:0 TEST RESULTS
{'test_loss': 0.05677778273820877}
--------------------------------------------------------------------------------
```

Out[5]: [{'test_loss': 0.05677778273820877}]