Al Hands-On By Andrii Tykhonov [andrii.tykhonov@cern.ch]

ISAPP Lecce, Italy. June 12, 2025



Operating system: Linux or MacOS

Supported and tested operating systems:

Linux (currently tested on Ubuntu, but others like Debian, Fedora etc. should work just fine)



Mac OS



Note on Windows — while it is not forbidden in the tutorial — it is neither tested nor fully supported, so it will be at your own risk. If you have a windows machine, it is adviced to install Linux either as a second operating system or in a virtual environment (e.g. through VirtualBox). Please contact me in advance if you have a Windows machine and never worked with Linux before.





Software prerequisite: Miniconda is (almost) all you need!

Install **Miniconda**:

- lacksquarelinux-installation
- lacksquare
- For example, in linux (basically same in MacOs but using curl instead of wget):

→ bash ~/Miniconda3-latest-Linux-x86 64.sh

- one paragraph to the setup script you can easily remove if you want to delete conda

→ source ~/.bashrc

(or ~/.zshrc - depending on which shell you are using)



Follow the instructions in: https://www.anaconda.com/docs/getting-started/miniconda/install#macos-

Use **Terminal installer** (not graphical one): it allows to easily install/replace and experiment with Miniconda - everything will be placed in your home directory instead of the system one, so you will avoid potential conflicts with already pre-installed python versions etc.

→wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86 64.sh

 \rightarrow You will have to agree license agreement etc (press "q" to exit license agreement in the terminal \bigcirc).

→ When it prompts "Choose an initialization options:" choose YES. If you are worried, you can make a backup of your profile initialization scripts (~/.bashrc or ~/.zshrc depending on the shell you use), but in principle all what conda does is adding







Setup ML software: Tensorflow

Throughout the tutorial we will use both **Tensorflow** and **Pytorch** frameworks. We will install those in a two separate "environments" of conda and you will be able to easily switch between the two. You will appreciate the convenience and power of Conda – it allows you to install ML software (hopefully) quickly, (hopefully) gracefully, and without experiencing library conflicts etc. (unless something goes really wrong - but we are here to help you in this case ...). In the first part of the tutorial we will work with Tensorflow since (arguably) it is more simple/intuitive to use.

Install **Tensorflow**

- 1. Add conda-forge channel to look for software:
- 2. Create new environment that we will call "tf":
- 3. Activate the "tf" environment:

Comment on activation (step 3): this has to be done in each new shell. If you want it to active the environment automatically at the start of a new shell, you can add the above command to your ~/.bashrc [or ~/.zshrc]

Note on the tensorflow versions: sometimes one needs to experiment with versions, for example I have experienced problems with latest tensorflow versions, so I downgraded to 2.16 on Ubuntu (2.15 on MacOS): conda create -n tf tensorflow=2.16





Test Tensorflow #1 ...

- Activate the Tensorflow environment in conda (see previous page for mode details): conda activate tf
- See if Tensorflow libs are there and working: python
 - >>> import tensorflow as tf
 - >>> print (tf. version)
 - 2.16.XX

NOTE USE Tensorflow version 2.15 or higher! ### If nothing crashes so far - things seem to work so far.. ### First import of tensorflow on some systems (e.g. MacOS) may take ### a while - be patient (it will cache and work faster afterwards)

Test Tensorflow #2...

Let's run a mock-up model training:



• You should see something like:

Epoch	1/5								
32/32	[======================================] —	0s	206us/step	_	loss:	1.5475	_	accu
Epoch	2/5								
32/32	[======================================] —	0s	158us/step	-	loss:	1.2575	_	accu
Epoch	3/5								
32/32	[======================================] —	0s	136us/step	-	loss:	1.0151	_	accu
Epoch	4/5								
32/32	[======================================] —	0s	129us/step	_	loss:	0.8074	_	accu
Epoch	5/5								
32/32	[======================================] —	0s	144us/step	-	loss:	0.6322	—	accu

>>> l = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True) # we will learn about it later # train the model

racy: 0.0000e+00

racy: 0.0000e+00

uracy: 0.0000e+00

uracy: 0.0000e+00

uracy: 0.0000e+00



Test PyTorch

• Run the following code snippet in python:

```
import torch as tr
import numpy as np
model = tr.nn.Sequential( tr.nn.Linear(10, 10, dtype=tr.float64))
criterion = tr.nn.CrossEntropyLoss()
optimizer = tr.optim.Adam(model.parameters())
# one training step
optimizer.zero_grad()
y \text{ pred} = \text{model}(x)
loss = criterion(y pred,y)
loss.backward()
optimizer.step()
print ("Loss:", loss.item())
```

If everything is installed correctly, you should see the log of the training:

```
Loss: 1.930114470175824
```

x = tr.tensor(np.random.rand(1000,10),dtype=tr.float64) # random sample of 1000 sets of numbers y = tr.tensor(np.random.rand(1000,),dtype=tr.int64) # random sample of 1000 set 'labels'



If things go wrong ...



Possible solutions:

Re-install another tenosorflow version (you may have to experiment with a few different versions):

```
conda remove -n tf --all
conda create -n tf tensorflow=2.16
```

Mask out your GPU*:

```
export CUDA_VISIBLE_DEVICES=""
```

* If you you have an Nvidia GPU and the the solution 1 (re-installing different tensorflow versions) do not help, try forcing tensorflow NOT to use the GPU (it is OK for the sake of this tutorial; in the future you may tweak your software setup to fully profit of your nice GPU hardware)

- these are few typical problems I encountered myself ... Unfortunately there might be more, but normally with the slight help of google, chatgpt, stackoverflow and a little prayer – things will work ;-)

```
# delete existing tf environment from conda
# install a specific tensorflow version (e.g. 2.16 ubuntu, or 2.15 for MacOs)
```





Where to get the code ...

- •
- You will see the instructions and code slide-by-slide in your browser:

README.md

Slide 3: Software prerequisite: Miniconda is

- Follow the instructions in: https://www.anaconda.com/docs/getting
- For example, in linux (basically same in MacOs but using curl insterior)

wget https://repo.anaconda.com/miniconda/Miniconda3-la bash ~/Miniconda3-latest-Linux-x86_64.sh source ~/.bashrc # (or ~/.zshrc - depending on whic

Slide 4: Setup ML software: Tensorflow

Add conda-forge channel to look for software:

conda config --add channels conda-forge

• Create new environment that we will call tf:

conda create -n tf tensorflow

Activate the tf environment:

conda activate tf

Goto: https://gitlab.cern.ch/andrii/mlregressioncalo/-/tree/tutorial

(almost) all you need!
g-started/miniconda/install#macos-linux-installation ead of wget):
test-Linux-x86_64.sh
h shell you are using)