
System: Ubuntu 24.04 LTS

immer die neue Package-Liste sicherstellen:

sudo apt-get update

sudo apt-get upgrade

Type, ID eines nvidia-graphic-cards, CUDA-Version, Driver Information auflisten:

nvidia-smi

| | | | | | | | | |
|-----------------------|------------------------|------|----------------------------|------------------|--------------------|--------------------|------------|-----|
| NVIDIA-SMI 550.107.02 | | | Driver Version: 550.107.02 | | | CUDA Version: 12.4 | | |
| GPU | Name | | Persistence-M | Bus-Id | Disp.A | Volatile Uncorr. | ECC | |
| Fan | Temp | Perf | Pwr:Usage/Cap | | Memory-Usage | GPU-Util | Compute M. | |
| | | | | | | | MIG M. | |
| 0 | NVIDIA GeForce GT 1010 | | Off | 00000000:01:00.0 | Off | | | N/A |
| N/A | 25C | P8 | N/A / 19W | 6MiB / 2048MiB | | 0% | Default | N/A |
| Processes: | | | | | | | | |
| GPU | GI | CI | PID | Type | Process name | GPU Memory | | |
| | ID | ID | | | | Usage | | |
| 0 | N/A | N/A | 2503 | G | /usr/lib/xorg/Xorg | 4MiB | | |

falls nicht funktioniert, sehr wahrscheinlich ist Driver-Konfiguaton problematisch:

sudo ubuntu-drivers list

sudo ubuntu-drivers install nvidia:{xxx} : xxx ist Version.

CUDA

sudo apt install nvidia-cuda-toolkit

nvcc -V

falls

nvcc: NVIDIA (R) Cuda compiler driver

Copyright (c) 2005-2023 NVIDIA Corporation

Built on Fri_Jan__6_16:45:21_PST_2023

Cuda compilation tools, release 12.0, V12.0.140

Build cuda_12.0.r12.0/compiler.32267302_0

auf Bildschirm gezeigt dann fertig.

CUDNN

sudo apt-get install zlib1g

sudo apt install nvidia-cudnn

Eine ISOLIERTE UMGEBUNG ist sehr wichtig:

python3 -m venv {myenv}

man kann eine neue Umgebung unter /home/jd/myenv finden

source myenv/bin/activate: um die Umgebung myenv zu aktivieren

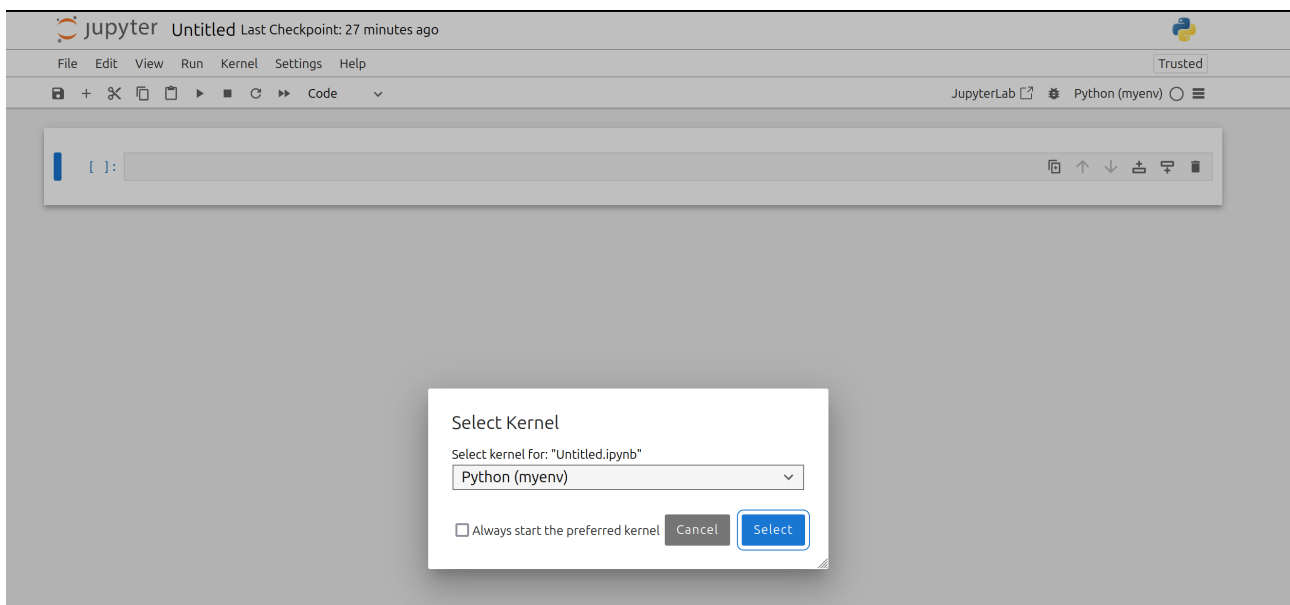
```
pip install tensorflow
pip install scikit-learn
pip install torch
```

deactivate: **Man kann jetzt den Vorgang abbrechen, wenn man kein Jupyterbook benutzt.**
Dann deine spezielle Umgebung ist fertig.

In dieser Umgebung ipykernel installieren und weiter in diesem Kernel registrieren:

```
pip install jupyter
pip install ipykernel
python -m ipykernel install --user --name={Envname} --display-name "Python ({Envname})"
dann sollte die folgende Nachricht gezeigt werden:
Installed kernelspec myenv in /home/{hostname}/.local/share/jupyter/kernels/{Env-name}
```

weiter immer in dieser Umgebung Jupyterbook aufrufen, im Terminal weiter tippen:
jupyter notebook



jetzt kann man oben recht Python 3 (ipykernel) klicken, dann sieht man eine Liste und check Python(myenv).

Test:

```
import tensorflow as tf
print("TensorFlow version:", tf.__version__)
```

TensorFlow version: 2.17.0

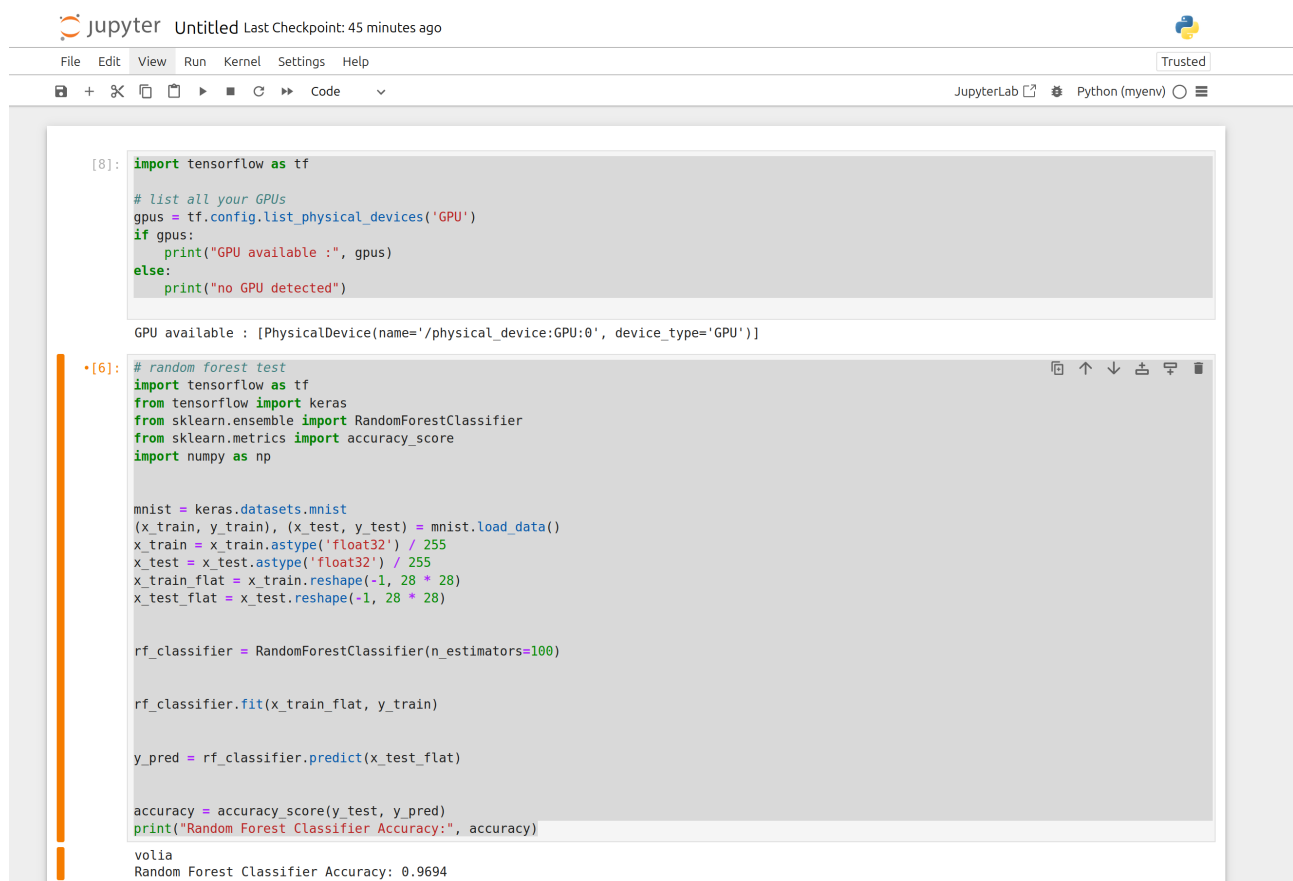
```
import tensorflow as tf
# list all your GPUs
gpus = tf.config.list_physical_devices('GPU')
if gpus:
    print("GPU available :", gpus)
else:
    print("no GPU detected")
GPU available : [PhysicalDevice(name='/physical_device:GPU:0',  
device_type='GPU')]
```

```
# random forest test
import tensorflow as tf
from tensorflow import keras
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import numpy as np

mnist = keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.astype('float32') / 255
x_test = x_test.astype('float32') / 255
x_train_flat = x_train.reshape(-1, 28 * 28)
x_test_flat = x_test.reshape(-1, 28 * 28)
rf_classifier = RandomForestClassifier(n_estimators=100)
rf_classifier.fit(x_train_flat, y_train)
y_pred = rf_classifier.predict(x_test_flat)
accuracy = accuracy_score(y_test, y_pred)
print("Random Forest Classifier Accuracy:", accuracy)
```

Random Forest Classifier Accuracy: 0.9694



JupyterLab interface showing the execution of the code. The top cell (index 8) checks for GPU availability and prints the message: GPU available : [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]. The bottom cell (index 6) runs the random forest test code, which prints the accuracy: Random Forest Classifier Accuracy: 0.9694.

```
[8]: import tensorflow as tf

# list all your GPUs
gpus = tf.config.list_physical_devices('GPU')
if gpus:
    print("GPU available :", gpus)
else:
    print("no GPU detected")

GPU available : [PhysicalDevice(name='/physical_device:GPU:0', device_type='GPU')]

•[6]: # random forest test
import tensorflow as tf
from tensorflow import keras
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import numpy as np

mnist = keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.astype('float32') / 255
x_test = x_test.astype('float32') / 255
x_train_flat = x_train.reshape(-1, 28 * 28)
x_test_flat = x_test.reshape(-1, 28 * 28)

rf_classifier = RandomForestClassifier(n_estimators=100)

rf_classifier.fit(x_train_flat, y_train)

y_pred = rf_classifier.predict(x_test_flat)

accuracy = accuracy_score(y_test, y_pred)
print("Random Forest Classifier Accuracy:", accuracy)

volia
Random Forest Classifier Accuracy: 0.9694
```

Voilà