Tensorflow-Model am KI-Server trainieren

Zugangsdaten KI-Server

Anmeldung läuft über die TSN-Adressen aber mit folgenden "Erweiterungen":

Schüler: “<<TSN-USERNAME>>@students.htlinn.local”@10.10.11.11

Per SSH am Server einloggen

ssh username@10.10.11.11

Erwarteter Output:

```
korber@korberpc:~$ ssh "p.korber@staff.htlinn.local"@10.10.11.11
p.korber@staff.htlinn.local@10.10.11.11's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Erweiterte Sicherheitswartung (ESM) für Applications ist  nicht aktiviert.

105 Aktualisierungen können sofort angewendet werden.
Zum Anzeigen dieser zusätzlichen Aktualisierungen bitte »apt list --upgradable« ausführen

10 zusätzliche Sicherheitsupdates können mit ESM Apps angewendet werden.
Erfahren Sie mehr über die Aktivierung des ESM Apps-Dienstes at https://ubuntu.com/esm

Last login: Tue Dec 17 17:23:50 2024 from 10.10.226.75
p.korber@staff.htlinn.local@KI-Spielwiese-1:~$
```

Tensorflow-venv verwenden

Die venv ist unter */usr/bin/tf/bin/* zu finden.

Erwarteter Output:

```
p.korber@staff.htlinn.local@KI-Spielwiese-1:~$ source /usr/bin/tf/bin/activate
(tf) p.korber@staff.htlinn.local@KI-Spielwiese-1:~$ which python3
/usr/bin/tf/bin/python3
(tf) p.korber@staff.htlinn.local@KI-Spielwiese-1:~$
```

Training Script an den Remote Server senden

Starte ein Terminal auf deinem lokalen Rechner in dem Ordner, wo dein Script liegt.

Sende deine Python-Datei mit scp an den Remote Server (Zielort: Dein Home-Verzeichznis am KI-Server)

scp dein_script.py username@10.10.11.11:/home/username/

Erwarteter Output

- am lokalen Rechner:

```
korber@korberpc:~/Documents/Schule/2024_2025/KISY1/4_Sumpervised Learning#2/MNIST/mnist_tensorflow_data_example$ scp predict.py "p.korber@staff.htlinn.local"@10.10.11
.11:/home/p.korber@staff.htlinn.local
p.korber@staff.htlinn.local@10.10.11.11's password:
korber@korberpc:~/Documents/Schule/2024_2025/KISY1/4_Sumpervised Learning#2/MNIST/mnist_tensorflow_data_example$
```

- im Home-Verzeichnis am KI-Server:

```
p.korber@staff.htlinn.local@KI-Spielwiese-1:~$ ls
predict.py  snap
```

Script am Server starten

Trainiere dein Tensorflow-Modell am KI-Server. Achte darauf, dass nur das Modell erzeugt und im Home-Verzeichnis abgespeichert werden soll (keine GUI, keine weiteren Funktionen).

Es sollte eine Tensorflow-Modell-Datei (*.h5) im Home-Verzeichnis erzeugt werden.

python3 dein_script.py

Erwarteter Output:

```
p.korber@staff.htlinn.local@KI-Spielwiese-1:~$ source /usr/bin/tf/bin/activate
(tf) p.korber@staff.htlinn.local@KI-Spielwiese-1:~$ python3 predict.py
2024-12-18 09:24:09.452391: E external/local_xla/xla/stream_executor/cuda/cuda_
 cuFFT when one has already been registered
2024-12-18 09:24:09.471272: E external/local_xla/xla/stream_executor/cuda/cuda_
n cuDNN when one has already been registered
```

```
(tf) p.korber@staff.htlinn.local@KI-Spielwiese-1:~$ ls
model_ai_server.h5  predict.py  snap
```

Model an deinen lokalen Rechner senden

Nutze nun in einem Terminal auf deinem lokalen Rechner scp, um das erzeugte Model aus dem Home-Verzeichnis am KI-Rechner in deinen gewünschten Ordner auf dem lokalen Rechner zu senden.

scp username@10.10.11.11:/home/username/model_ai_server.h5 /path/to/local/destination/

Erwarteter Output:

```
korber@korberpc:~/Documents/Schule/2024_2025/KISY1/4_Sumpervised Learning#2/MNIS
T/mnist_tensorflow_data_example$ scp 'p.korber@staff.htlinn.local'@10.10.11.11:/
home/'p.korber@staff.htlinn.local'/model_ai_server.h5 /home/korber
p.korber@staff.htlinn.local@10.10.11.11's password:
model_ai_server.h5                            100% 1313KB  10.5MB/s   00:00
```

Model für deine GUI nutzen

Speichere dein Modell im gleichen Ordner wie dein Python-Programm (GUI) ab und nutze im Programm dein Modell für die Vorhersage der Ziffer.

Erwarteter Output:

```
# --- Step 1: Load or Train the MNIST Digit Classifier ---
MODEL_FILE = "model_ai_server.h5"

def load_or_train_model():
    if os.path.exists(MODEL_FILE):
        print("Loading preexisting model...")
        model = tf.keras.models.load_model(MODEL_FILE)
    else:
        print("No preexisting model found. Training
        # Load the MNIST dataset
        (x_train, y_train), (x_test, y_test) = mnist

        # Normalize the dataset to 0-1 range
        x_train, x_test = x_train / 255.0, x_test /

        # Create a Sequential Feedforward Neural Network
        model = Sequential([
            Flatten(input_shape=(28, 28)),  # Flatten 28x28 images to 1
            Dense(128, activation='relu'),  # Hidden layer with 128 neu
            Dense(64, activation='relu'),   # Additional hidden layer
```