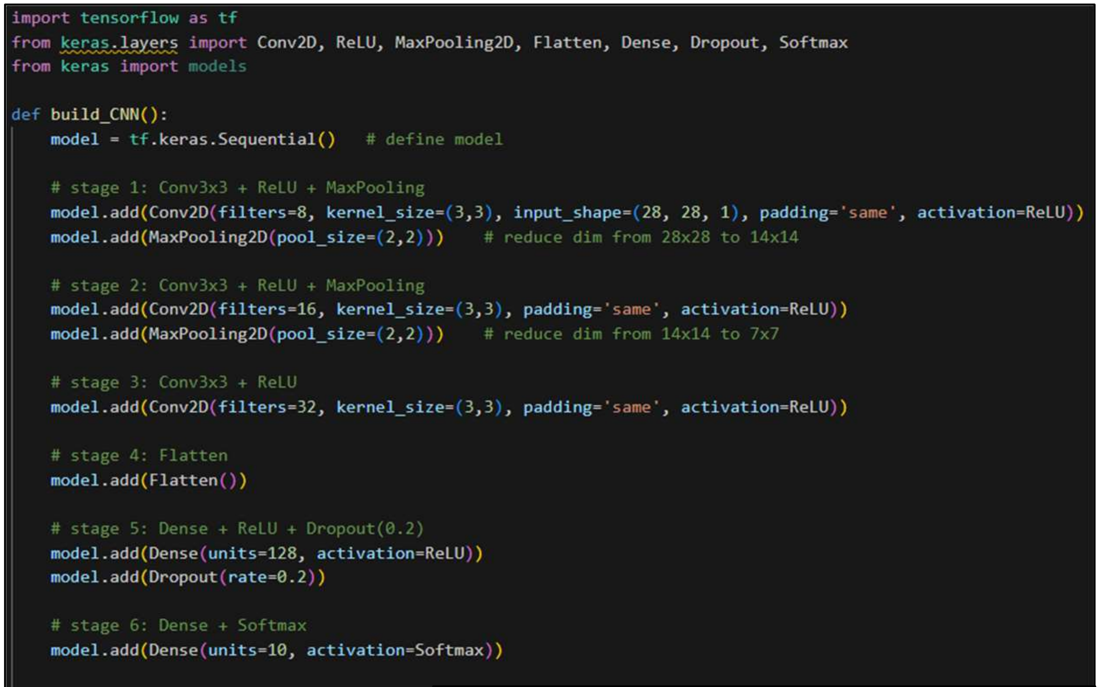
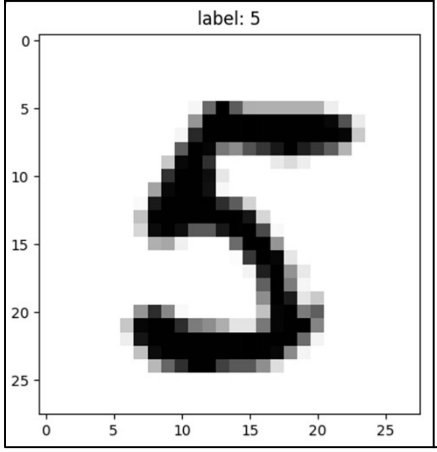
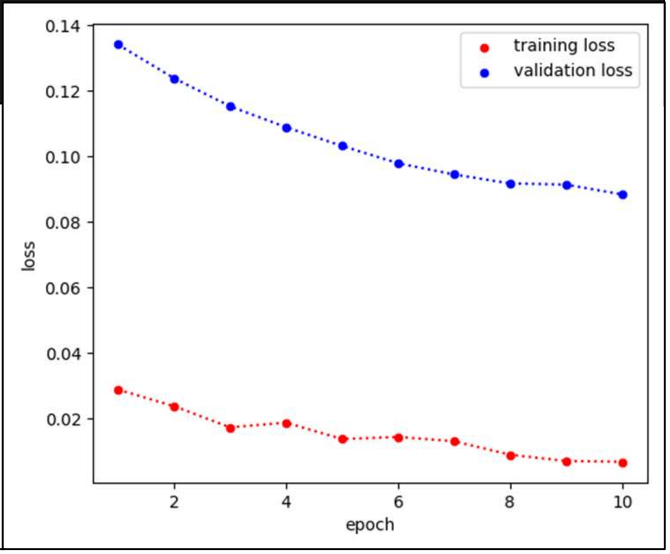
Homework 8

Build CNN



```
import tensorflow as tf
from keras.layers import Conv2D, ReLU, MaxPooling2D, Flatten, Dense, Dropout, Softmax
from keras import models

def build_CNN():
    model = tf.keras.Sequential()   # define model

    # stage 1: Conv3x3 + ReLU + MaxPooling
    model.add(Conv2D(filters=8, kernel_size=(3,3), input_shape=(28, 28, 1), padding='same', activation=ReLU))
    model.add(MaxPooling2D(pool_size=(2,2)))    # reduce dim from 28x28 to 14x14

    # stage 2: Conv3x3 + ReLU + MaxPooling
    model.add(Conv2D(filters=16, kernel_size=(3,3), padding='same', activation=ReLU))
    model.add(MaxPooling2D(pool_size=(2,2)))    # reduce dim from 14x14 to 7x7

    # stage 3: Conv3x3 + ReLU
    model.add(Conv2D(filters=32, kernel_size=(3,3), padding='same', activation=ReLU))

    # stage 4: Flatten
    model.add(Flatten())

    # stage 5: Dense + ReLU + Dropout(0.2)
    model.add(Dense(units=128, activation=ReLU))
    model.add(Dropout(rate=0.2))

    # stage 6: Dense + Softmax
    model.add(Dense(units=10, activation=Softmax))

    return model

my_CNN = build_CNN()
my_CNN.summary()
```





Answers to Questions:

Five Activation Functions:

1. (Leaky) ReLU
2. Sigmoid function (or hyperbolic tangent)
3. Linear function
4. Step function (signum or unit step)
5. Piecewise linear function

Adam:

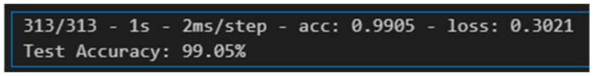
Adam is an optimizer based on stochastic gradient descent using adaptive moment estimation, helping the algorithm converge a local minimum faster.

sparse_categorical_crossentropy:

- Loss-function for classification tasks with mutually exclusive classes
- expects integer labels (not one-hot encoded)
- computes the cross-entropy loss
- particularly useful for multi-class problems where the target is a single class index

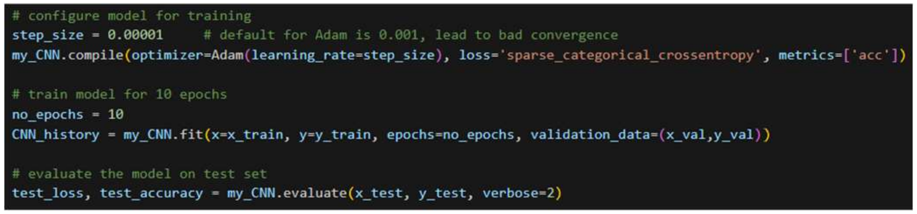
Epoch:

An *epoch* is one complete pass through the whole training set during the training of a ML model.



```
313/313 - 1s - 2ms/step - acc: 0.9905 - loss: 0.3021
Test Accuracy: 99.05%
```

Compile and evaluate CNN



```
# configure model for training
step_size = 0.00001     # default for Adam is 0.001, lead to bad convergence
my_CNN.compile(optimizer=Adam(learning_rate=step_size), loss='sparse_categorical_crossentropy', metrics=['acc'])

# train model for 10 epochs
no_epochs = 10
CNN_history = my_CNN.fit(x=x_train, y=y_train, epochs=no_epochs, validation_data=(x_val,y_val))

# evaluate the model on test set
test_loss, test_accuracy = my_CNN.evaluate(x_test, y_test, verbose=2)
```

Name: Josephina Imhoff | **Matriculation No.:** 23464631 | **IdM:** jo17wila