Lab4: Training and evaluating a neural network to recognize handwritten digits.

The goal of this assignment is to build and train a Neural Network that can recognize handwritten digits from 0 to 9. You can use the following code to download the MINIST data of handwritten digits.

```
import tensorflow as tf
import numpy as np
from tensorflow import keras
import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
import random


(X_train_full, y_train_full), (X_test, y_test) =
keras.datasets.mnist.load_data()

X_train_full.shape

# test a random image
def plot_digit(image):
    plt.imshow(image, cmap = mpl.cm.binary,
               interpolation="nearest")
    plt.axis("off")

some_image = random.randint(0, X_train_full.shape[0])
some_digit = X_train_full[some_image,:,:]
plot_digit(some_digit)

# rescaling and separating a validation set
X_valid, X_train = X_train_full[:5000] / 255.0, X_train_full[5000:] /255.0
y_valid, y_train = y_train_full[:5000], y_train_full[5000:]
```

You need to perform the following steps

1. Train a neural network model using only one inner dense layer of 10 units . Choose the correct input and output layers. Investigate the model using learning rates of 0.01 and 0.1 and a ReLU activation function. Train your model for 100 epochs and plot learning curves for the training and validation set for each combination of above parameters. You can refer to chapter 10 for help.
   Question 1: From each learning curve draw your conclusions about the performance of your model based on the above parameters? Is your model overfitting or underfitting?
2. Repeat the step 1 for the following cases:
   a. one inner dense layer of 50 units
   b. one inner dense layer of 100 units
   c. two dense layers of 10 and 10 units
   d. two dense layers of 50 and 50 units
   e. two dense layers of 100 and 50 units
   f. two dense layers of 100 and 100 units

   From each learning curve draw your conclusions about the performance of your model based on the above parameters? Is your model overfitting or underfitting? Is there anything

else you can suggest to change which could have a positive effect on the performance of the above model?

3. Draw an overall conclusion from the above steps. Which model and parameters you think are the best choice to recognize handwritten digits? Explain with reasoning. Take this final model and evaluate it on the test set