Keras basics and metrics

Task 1

1. With the line:

```
(trainX, trainY), (testX, testY) = tf.keras.datasets. mnist.load_data()
```

the MNIST dataset should be downloaded, containing black-and-white photos of handwritten digits.
The trainX and testX tables should be cast to tf.float32 and standardized to range <0.1> dividing them by 255.0.

2. Declare a model in Keras with structure:

    1. Flatten (layer flattening tensors to vectors)
    2. Dense-128-ReLU
    3. Dense-128-ReLU
    4. Dense-10-Softmax

There are two ways to do this (two separate functions that return an object):

    1. Using the Sequential API:
       https://www.tensorflow.org/guide/keras/sequential_model
    2. Using the Functional API:
       https://www.tensorflow.org/guide/keras/functional

This model will carry out the task of multi-class classification manually written digits.

3. Load the model_w.h5 file into google colab and load it as parameters of both models using model.load_weights().

4. For the MNiST test set (testX, testY) calculate the values of the following metrics:

- accuracy
- precision
- recall
- specificity
- F1 Score

You can use the tf.keras.metrics and model.evaluate() modules or sklearn.metrics. Make sure that both models return the same values metrics for the test set.

Task 2

Use the Keras Functional API to declare a new model that:
- accepts scalar with precision tf.float32, as well as batch images with set of MNiST with precision tf.float32 (batch of any number of images)
- process batch images from MNiST using a sequential model with previous task (with loaded parameters)
- add to the prediction of the model the scalar adopted at the input, and then return the raw predictions and those after addition.

The model should be called for a batch consisting of the first 5 images in the test set of MNiST and scalar value of 5 per input.