FYSS5120 Efficient Numerical Programming - Demo 4

Drop solutions before the demo session to the Nextcloud box (link)
Please indicate clearly your name in the file name,
so that I can tell who solved what.

1. Examine the MNIST hand-written number recognition routine from Tensorflow/Keras examples. There are two versions, we'll look at
   `https://keras.io/examples/vision/mnist_convnet`
   with Convolution, MaxPooling, Flatten, Dropout, and Dense layers.
   A slightly modified code (added plotting etc.) is in the file `keras_mnist.py`

2. What's the purpose of these layer types?
   - `Convolution`
   - `Flatten`

3. Teach the neural network to recognize numbers 6 and 9. First, edit the `keras_mnist.py` code so that it uses only numbers 6 and 9 for training and testing. See, for example, how-do-i-select-... @Stackoverflow
   Now that the number of training data has changed, what's a suitable `batch_size`?

4. Which images are hardest to identify? Compare the predicted labels `pred_values = model.predict(x_test)` with the correct ones; count the deviations and plot all or 25 first mislabelled images using `plot_images()`.

Notes:

The so-called *one-hot encoding* saves us from fuzzy predictions, such as "the number is 70% 6 and 30% 9". We want a straight answer, 6 or 9. Try this:

```
>>> from tensorflow.keras.utils import to_categorical
>>> y = [6,9,9,6,9,6,6,6,9]
>>> to_categorical(y)
```

The one-hot encoding works well with the categorical crossentropy loss functions, as explained in the blog by Raúl Gómez.