MNIST Classifier

Data

The problem we are trying to solve here is to classify grayscale images of handwritten digits (28 pixels by 28 pixels), into their 10 categories (0 to 9).

The dataset we will use is the MNIST dataset. It's a set of 60,000 training images, plus 10,000 test

The MNIST dataset comes pre-loaded in Keras, in the form of a set of four Numpy arrays:

```
from keras.datasets import mnist

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

Question 1

Print the shape of train and test images and labels

Question 2

Before training, we will preprocess our data by reshaping it into the shape that the network expects, and scaling it so that all values are in the `[0, 1]` interval. Previously, our training images for instance were stored in an array of shape `(60000, 28, 28)` of type `uint8` with values in the `[0, 255]` interval. We transform it into a `float32` array of shape `(60000, 28 * 28)` with values between 0 and 1.

Write the code that performs the above transformation.

```
train_images = # Write code to make it (60000, 28*28)
train_images = # Write code to make it between (0,1)

test_images = # Write code to make it (60000, 28*28)
test_images = # Write code to make it between (0,1)
```

Question 3

We also need to categorically encode the labels.

```
train_labels = # write code to convert to one hot encoding or categorical
test_labels = # write code to convert to one hot encoding or categorical
```

Our workflow will be as follow: first we will present our neural network with the training data, `train_images` and `train_labels`. The network will then learn to associate images and labels. Finally, we will ask the network to produce predictions for `test_images`, and we will verify if these predictions match the labels from `test_labels`.

Let's build our network

Question 4

Model

Write Sequential model with the following architecture:

- Input=Dense: 512, relu activation
- Layer_1=Dense: 256

```
from keras import models
from keras import layers

network = models.Sequential()
network.add(....)# Write your code here
network.add(....)# Write your code here
```

Question 5

add to your model the proper output layer for MNIST classification.

```
network.add(...)# Write you code here
```

Question 6

Print your model parameters

```
# Write your code here
```

Question 7

Compilation: loss + optimizer

To make our network ready for training, we need to pick three more things, as part of "compilation" step:

Now compile your model with the three components

```
network.compile(...)# Write your answer here
```

Question 8

Learning

We are now ready to train our network, which in Keras is done via a call to the `fit` method of the network: we "fit" the model to its training data.

```
network.... # Write your answer here
```

Question 9

Now we want to evaluate our model on the test data.

Write the code to give you the accuracy and loss on test data:

```
test_loss, test_acc = # Write your code here
```

Question 10

Now we want to make sample prediction of one image below:

```
import matplotlib.pyplot as plt
img = test_images[0]
plt.imshow(np.reshape(img, (28,28)))
```

```
<matplotlib.image.AxesImage at 0x7f4e90e529e8>
```

```
# Write the code here to print the digit name as predicted by the model
```

```
array([[0.07699326, 0.07023557, 0.09060565, 0.1736434 , 0.07890654,
        0.09350489, 0.08068074, 0.09926819, 0.09897252, 0.13718927]],
      dtype=float32)
```

Question 11

In Keras convention, which axis is always the samples axis. As you can see above, the train images is (60000, 784).

Suppose that we have obtain the `train_images` variable as (784,60000) instead.

How can we fix it?

```
import numpy as np
bad_train_images = np.zeros((784,60000)) # This is the bad data matrix
print(bad_train_images.shape)
# Can you type one line of code to fix it?
train_images = ....# Type your answer here
print(train_images.shape)
```