Preparing the MNIST Training Data

Rescaling the input values to a smaller range, between 0 and 1 to prepare it for training.

We've worked out how to get data out of the MNIST data files and disentangle it so we can make sense of it, and visualize it too. We want to train our neural network with this data, but we need to think just a little about preparing this data before we throw it at our neural network.

We saw earlier that neural networks work better if the input data, and also the output values, are of the right shape so that they stay within the comfort zone of the network node activation functions.

The first thing we need to do is to rescale the input color values from the larger range 0 to 255 to the much smaller range $0.01 - 1.0$. We've deliberately chosen 0.01 as the lower end of the range to avoid the problems we saw earlier with zero-valued inputs because they can artificially kill weight updates. We don't have to choose 0.99 for the upper end of the input because we don't need to avoid 1.0 for the inputs. It's only for the outputs that we should avoid the impossible to reach 1.0.

Dividing the raw inputs which are in the range $0 - 255$ will bring them into the range $0 - 1$. We then need to multiply by 0.99 to bring them into the range $0.0 - 0.99$. We then add 0.01 to shift them up to the desired range 0.01 to 1.00. The following Python code shows this in action:

```
all_values = data_list[1].split(',')
scaled_input = (numpy.asfarray(all_values[1:]) / 255.0 * 0.99) + 0.01
print(scaled_input)
```

The output confirms that the values are now in the range 0.01 to 0.99.

So we've prepared the MNIST data by rescaling and shifting it, ready to throw

at our neural network for both training and querying.