Updating the Neural Network Code...Again

In this lesson, we will be updating the code again with the additional work we have done so far.

Previously we added a simple code to work out the fraction of correct answers. Let's see what this updated code produces:

```
# test the neural network

# scorecard for how well the network performs, initially empty
scorecard = []

# go through all the records in the test data set
for record in test_data_list:
    # split the record by the ',' commas
    all_values = record.split(',')
    # correct answer is first value
    correct_label = int(all_values[0])
    print(correct_label, "correct label")
    # scale and shift the inputs
    inputs = (numpy.asfarray(all_values[1:]) / 255.0 * 0.99) + 0.01
    # query the network
    outputs = n.query(inputs)
    # the index of the highest value corresponds to the label
    label = numpy.argmax(outputs)
    print(label, "network's answer")
    # append correct or incorrect to list
    if (label == correct_label):
        # network's answer matches correct answer, add 1 to scorecard
        scorecard.append(1)
    else:
        # network's answer doesn't match correct answer, add 0 to scorecard
        scorecard.append(0)
        pass

    pass

print(scorecard)

# calculate the performance score, the fraction of correct answers
scorecard_array = numpy.asarray(scorecard)
print ("performance = ", scorecard_array.sum() / scorecard_array.size)
print ("performance percentage = ", (scorecard_array.sum() / scorecard_array.size)*100,"%")
```

Add all this new code we've just developed to test the neural network's

Add all this new code we've just developed to test the neural network's performance to our main program. Change the file names so that you point to

the full training data set of $60,000$ records, and the test data set of $10,000$ records. We previously saved those files as `mnist_dataset/mnist_train.csv` and `mnist_dataset/mnist_test.csv` on GitHub.

Remember, you can get the Python notebook online at GitHub:

- https://github.com/makeyourownneuralnetwork/makeyourownneuralnetwork/blob/master/part2_neural_network_mnist_data.ipynb

The history of that code is also available on GitHub so you can see the code as it developed:

- https://github.com/makeyourownneuralnetwork/makeyourownneuralnetwork/commits/master/part2_neural_network_mnist_data.ipynb

The result of training our simple 3-layer neural network against the full $60,000$ training examples, and then testing it against the $10,000$ records, will give us an overall performance score of 0.9473. That is very very good. Almost 95 accurate! It is worth comparing this score of just under 95% accuracy against industry benchmarks recorded at http://yann.lecun.com/exdb/mnist/. We can see that we're better than some of the historic benchmarks, we are about the same performance as the simplest neural network approach listed there, which has a performance of 95.3%.

That's not bad at all. We should be very pleased that our very first go at a simple neural network achieves the kind of performance that a professional neural network researcher achieved. By the way, it shouldn't surprise you that crunching through $60,000$ training examples, each requiring a set of feedforward calculations from 784 input nodes, through 100 hidden nodes, and also doing an error feedback and weight update, all take a while even for a fast modern home computer. My new laptop took about 2 minutes to get through the training loop. Yours may be quicker or slower.