(a) Summary of my work:

(i) Loaded the data of training and testing from given file.

(ii) Use “mlp” function to build the neural network.
NET = mlp(784, hidden_node_number, 10, 'logistic');
This command gives a 784-n-10 neural network.

(iii) Extract certain amount of training data and testing data according to desired number.

(iv) Transfer the labels into required form. For instance, the outcome is 5, then transfer it into a vector [0 0 0 0 1 0 0 0 0] that represents 0-9.

(v) Train the net and update its value with “netopt” function.
[NET, options] = netopt(NET, options, train_x', train_output, 'scg');

(vi) After training, test the network with both training data and testing data to generate error. Using “mlpfwd”

```
for i→ 1 to number of test data
Y1 → mlpfwd(NET, testing.data(:,i)');
Get the index of max element in Y1
The outcome→index-1(since from 0-9)
    if (outcome unequal to test labels)
    test error→+1
    end
end
```

(vii) Calculate error rate

(viii) Implemented cross-validation method (k-fold). Dividing the data into four groups randomly, train different data groups separately and then test with testing data. Use average error in each case for the error rate of validation.

(ix) Draw the figure of misclassified case as well as correct case. For misclassified case, also print out the label of that case and the output from neural network.

(x) Changing the number of epoch, training data number and hidden node to find a best performing network.

(b) Best performing network

Training data number=60000;
testing data number=1000;
hidden node number: 500
test error =0.0170
train error =0
validation error =0.0208

(c) Result Analyzing

--Effect of parameters

While training, the accuracy improves when adding number of hidden nodes (<1000) and increasing the number of training data. However, increasing hidden nodes takes more time than increasing same number of training data. Also, when I try to vary the number of hidden node from 500 to 1000, it doesn't make much sense on the error output.

Here's some simple examples of outcome with different number of hidden node, training data and epochs.

(i) Various hidden node

hidden node	Training data	Testing data	Epoch	Test error	Train error	Validation error
100	1000	500	100	0.1180	0	0.1370
500	1000	500	100	0.1120	0	0.1455
1000	1000	500	100	0.1260	0	0.1565

500 hidden node generates least test error.

(ii) Various training data

hidden node	Training data	Testing data	Epoch	Test error	Train error	Validation error
500	1000	500	500	0.1180	0	0.1390
500	5000	500	500	0.0440	0	0.0585
500	10000	500	500	0.0360	0	0.0370
500	60000	1000	500	0.0170	0	0.0208

(iii) Various epoch

hidden node	Training data	Testing data	Epoch	Test error	Train error	Validation error
500	1000	500	50	0.1260	0	0.1540
500	1000	500	100	0.1120	0	0.1455
500	1000	500	500	0.1180	0	0.1390

The increasing of epoch can decrease error.

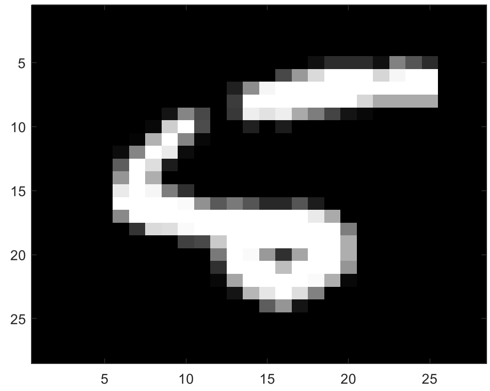
--The importance of different types of error:

Training error represents the existence of noise data and training accuracy. Normally with good training samples (like the ones we are given in this question), training data is close or equal to 0. Test error reflects how well the training result can be used for test data. By increasing proper number of training data, this error can be deceased. Validation error represents overfitting. By using cross-validation method, we can prevent and detect overfitting. In this question, I applied 4-fold validation to detect whether the case is overfitting.

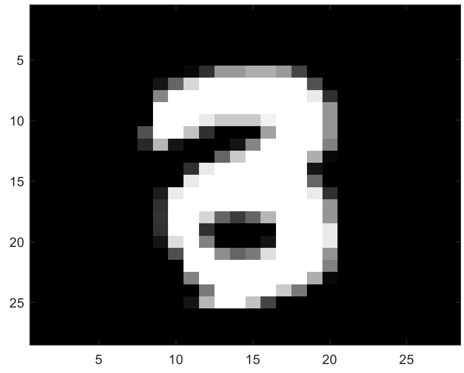
--How to choose best network

When choosing best network, I chose the one with 0 training error and minimum test error. Also, I applied the epoch of 500 as instructed. I chose 500 hidden nodes since it generates least test error during the experiment with 1000 training data. By changing the number of training samples, I found out 60000 samples generates least error, as illustrated in the table above.

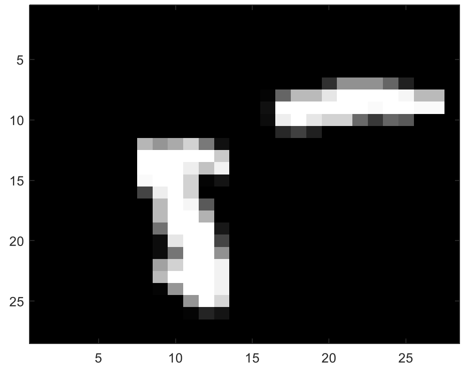
(d) Examples of misclassification:



The label of this number should be 5 but it is classified as 6.

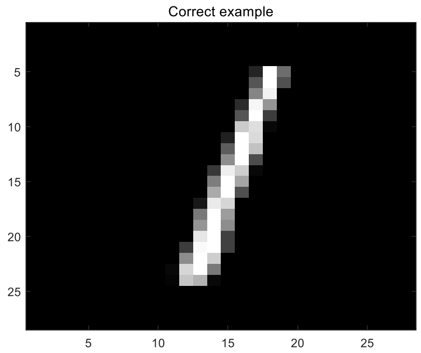


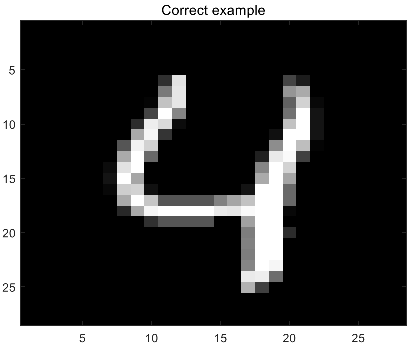
The label of this number should be 3 but it is classified as 8.

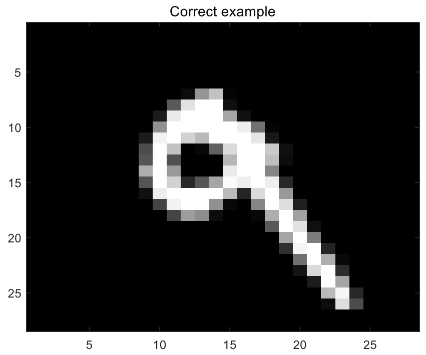


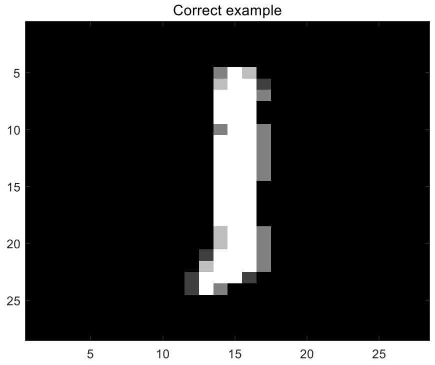
The label of this number should be 5 but it is classified as 0.

Examples of correct cases:


Correct example


Correct example


Correct example


Correct example

These four are correctly classified, left top one is 1, right top one is 4, left bottom one is 9, right bottom one is 1.