

## Exercise 12.1 (June 8, 2020)

-B9TB1707

### Question:

- Make two hand-written images; one can be recognized correctly by your trained model, and another is which cannot be recognized correctly.

### Solution:

My code for the solution is as follows:

```
CAPS_12_B9TB1707_12.1.m
1 fid=fopen('D:\Eric\My CAPS homework\tl0k-images-idx3-ubyte','r','b');
2 fread(fid,4,'int32');
3 data=fread(fid,[28*28,10000],'uint8');
4 fclose(fid);
5 fid=fopen('D:\Eric\My CAPS homework\tl0k-labels-idx1-ubyte','r','b');
6 fread(fid,2,'int32');
7 label=fread(fid,10000,'uint8');
8 addpath('D:\Eric\My CAPS homework\liblinear-2.30\matlab');
9 tr_label = label(1:5000);
10 tr_data = data(:,1:5000);
11 model = train(tr_label,sparse(tr_data));
12 te_label = label(5001:6000);
13 te_data = data(:,5001:6000);
14 pred_label = predict(te_label,sparse(te_data),model);
15 sample1 = imread('D:\Eric\My CAPS homework\pic3.png');
16 sample = mean(sample1,3);
17 predict([3], sparse(reshape(sample',1,28*28)), model)
18 sample1 = imread('D:\Eric\My CAPS homework\pic71.png');
19 sample = mean(sample1,3);
20 predict([7], sparse(reshape(sample',1,28*28)), model)|
```

And the output is as follows:

```
Accuracy = 86.3% (863/1000)
Accuracy = 100% (1/1)
ans = 3
Accuracy = 0% (0/1)
ans = 2
```

These are the pictures I used for the inputs.



### How it works:

1. Line 1 to line 7 of the code deals with the loading the data to train the image processor. It first loads the data and then the labels for the data. The block of code used fopen() to open an input stream to the file fread() to read the data stored

within them. We skip all the 32-bit data (as suggested in the handout from class) and read only the numeric data. Using lines 1-7 I have loaded all the 10,000 28x28 pixel images and their labels.

2. Line 8 adds the Liblinear package to path.
3. Line 9 and 10 initializes the first 5000 entries of the images and their labels into respective arrays.
4. Line 11 trains a model based on the given data using `train()` and `sparse()` functions. It uses a Support vector machine to supervise the learning algorithm.
5. Line 12 and 13 initializes the next 5000 entries of the images and their labels into respective arrays.
6. And Line 14 tests the accuracy of the machine's recognition using the function `predict()`.
7. Line 15 imports the image to process and stores it in variable `sample1`.
8. Line 16 converts the image into a greyscale image matrix.
9. And Line 17 tries to predict the digit in the picture.
10. Line 18 to 20 repeat the process for the second picture.

### Conclusion:

In my opinion the code above is extremely faulty. One of the hallmarks of good code is the ability to work on any valid input. With this code I had to tweak the input multiple times to get the favourable results. Sometimes I got better results when I did not skip the 32-bit values. I feel that the problem arises from the fact that prediction based on polynomial fitting might not be sufficient to decipher handwritten curves.