

## CSC411 - Project #2

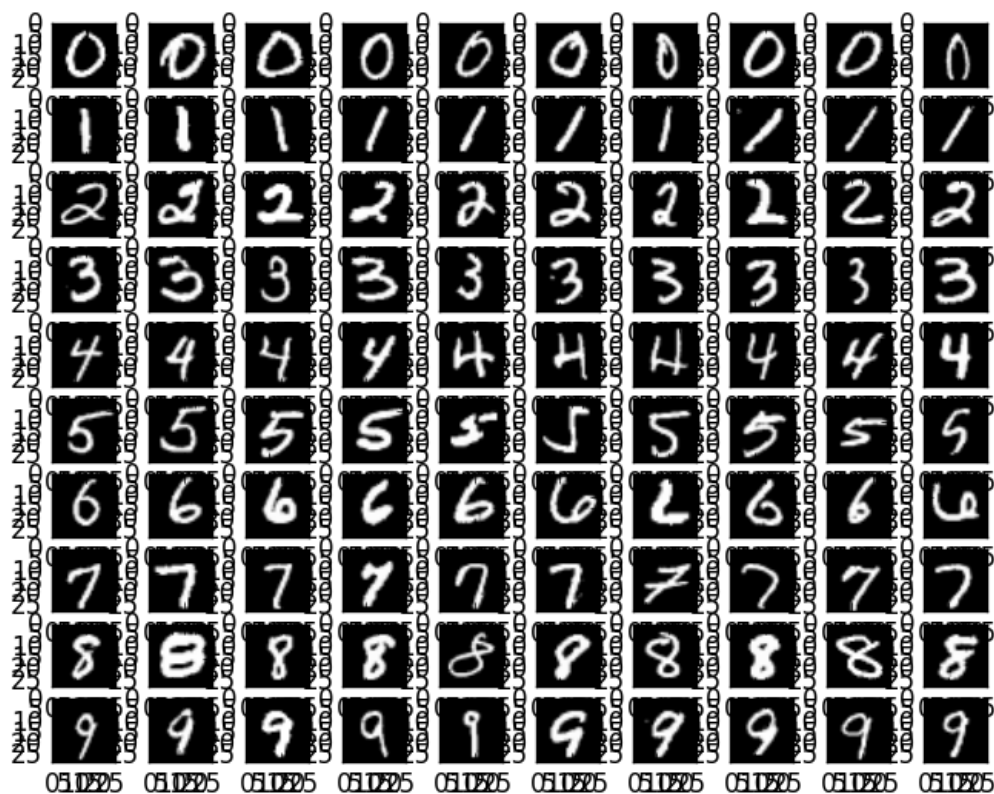
Michael Wong - [Student Number]  
Yijin (Catherine) Wang - 998350476

February 13, 2017

## Part 1

**Question** Describe the dataset of digits. In your report, include 10 images of each of the digits. You may find matplotlib's subplot useful.

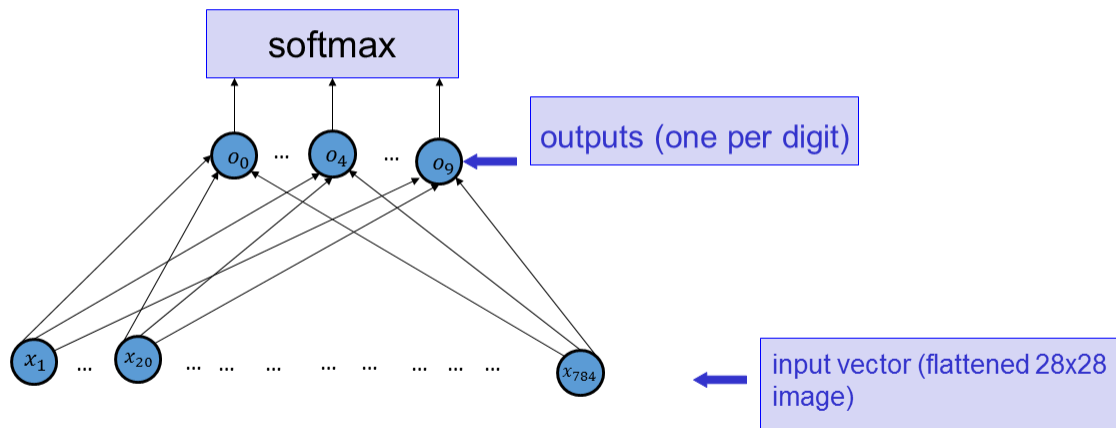
**Answer** Please see the images of each digits below. For the same digit, some images have thicker lines (like “8” in the second column) and some have thinner lines (like “8” in the third column). Some digit images are straight (like “1” in the first column) and some digits are rotated (like “1” in the ninth column). Most digits are clear to be identified manually but some digits are not clear (like the “6” in the seventh column).



10 images of digits from 0 to 9

## Part 2

**Question** Implement a function that computes the network below. The  $o$ 's here should simply be



linear combinations of the  $x$ 's (that is, the activation function in the output layer is the identity). Specifically, use  $o_i = \sum_j w_{ji}x_j + b_i$ . Include the listing of your implementation in your report for this Part.

**Answer** Please see the source code of the function below:

```
def softmax(y):  
    '''Return the output of the softmax function for the matrix of output y. y  
    is an NxM matrix where N is the number of outputs for a single case, and M  
    is the number of cases'''  
    return exp(y)/tile(sum(exp(y),0), (len(y),1))  
  
def lin_combin(w, b, x):  
    o = (dot(w.T, x) + b)  
    return softmax(o)
```

## Part 3

We would like to use the sum of the negative log-probabilities of all the training cases as the cost function.

### Part 3(a)

**Question** Compute the gradient of the cost function with respect to the weight  $w_{ij}$ . Justify every step. You may refer to Slide 7 of the One-Hot Encoding lecture, but note that you need to justify every step there, and that your cost function is the sum over all the training examples.

**Answer** Let  $j$  refers to the input size (number of  $x$  per sample) and  $i$  refers to the output size (number of  $o$  per sample), and  $k$  refers to the training set size (number of samples).

$X$  is the input matrix with dimension  $n \times m$ , where  $n$  is the number  $x$  (related to  $j$ ) and  $m$  is the sample size (related to  $k$ ).  $P$  is the calculated result matrix after using softmax with dimension  $d \times m$ , where  $d$  is the output size (number of outputs per sample, related to  $i$ ).  $Y$  is the expected output with the same dimension as  $P$ .

$$\frac{\partial C}{\partial o_{ik}} = p_{ik} - y_{ik}$$

$$\frac{\partial o_{ik}}{\partial w_{ij}} = x_{jk}$$

$$\begin{aligned}\frac{\partial C}{\partial w_{ij}} &= \sum_k \frac{\partial C}{\partial o_{ik}} \cdot \frac{\partial o_{ik}}{\partial w_{ij}} \\ &= \sum_k x_{jk}(p_{ik} - y_{ik}) \\ &= X(P - Y)^T\end{aligned}$$

### Part 3(b)

**Question** Write vectorized code that computes the gradient with respect to the cost function. Check that the gradient was computed correctly by approximating the gradient at several coordinates using finite differences. Include the code for computing the gradient in vectorized form in your report.

**Answer**