Programming 1 for ECSE 6965

Lingyu Zhang, RIN:661950104

Feb 26, 2018

Theoretical part

The objective of this problem is that we utilize the Stochastic gradient descent method to compute the optimal weight and bias as the parameters of multi-class regression. For this problem, we have the training dataset which contains 25112 samples, each sample is a $28 * 28$ vector, so we construct the data as

$$D = \begin{bmatrix} x[1] & x[2] & x[3] & ... & x[M] \\ 1 & 1 & 1 & ... & 1 \end{bmatrix}$$

where $x[m]$ is a $d*1$ dimension vector, in our case, $d = 28*28 = 784$ For the parameters, since each sample is $784 * 1$ then we can construct the parameter W as below

$$W = \begin{bmatrix} W[1] & W[2] & W[3] & W[4] & W[5] \end{bmatrix}$$

For each W, it contains weight and bias as below

$$W[k] = \begin{bmatrix} W_k[1] \\ W_k[2] \\ W_k[3] \\ . \\ . \\ . \\ W_k[784] \\ W_0[k] \end{bmatrix}$$

Therefore, the parameter W is $785 * 5$ dimension. For the label y, we construct it as one-of-k coding, if label is 2, then y is denoted as below

$$y = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The loss function is represented as below For each class k, we update the parameters.

$$\Theta_k = W[k]$$

$$L = -\sum_{m=1}^{M} logP(y[m]|x[m],\Theta) = -\sum_{m=1}^{M} log\prod_{i=1}^{K} P(y[m]=i|x[m],\Theta)^{y[m][i]}$$

$$L = -\sum_{m=1}^{M}\sum_{i=1}^{K} y[m][i] logP(y[m]=i|\ x[m],\Theta)$$

$$L = -\sum_{m=1}^{M}\sum_{i=1}^{K} y[m][i]\left[\Theta_i^t x[m] - log\sum_{j=1}^{K} exp(\Theta_j^t x[m])\right]$$

In order to reach the optimal solution of the parameter, minimizing the loss function, we then compute the gradient with regard to parameter Θ For each class k,

$$\frac{\partial L}{\partial \Theta_k} = \frac{\partial(-\sum_{m=1}^{M}\sum_{i=1}^{K} y[m][i]\left[\Theta_i^t x[m] - log\sum_{j=1}^{K} exp(\Theta_j^t x[m])\right])}{\partial \Theta_k}$$

$$\frac{\partial L}{\partial \Theta_k} = -\left[\sum_{m=1}^{M} y[m][k]x[m] - \sum_{m=1}^{M}\frac{\partial log\sum_{j=1}^{K} exp(\Theta_j^t x[m])}{\partial \Theta_k}\right]$$

$$\frac{\partial L}{\partial \Theta_k} = -\left[\sum_{m=1}^{M} x[m](y[m][k] - \sigma_m(\Theta_k^t x[m]))\right]$$

where σ_m is the softmax function Then we update the parameters for class k as below

$$\Theta_k^{t+1} = \Theta_k^t - \eta\frac{\partial L}{\partial \Theta_k}$$

We repeat this process for $k \in \{1,2,3,4,5\}$ classes. Since we want to use stochastic gradient descent method, we do not want to load all the data to compute the gradient at each iteration, so for each iteration, we randomly select several samples to compute the gradient, and repeat this process util the loss function converges.

Also we add the L2 regularization term, which makes the update as below

$$\Theta_k^{t+1} = \Theta_k^t - \eta(\frac{\partial L}{\partial \Theta_k} + 2\lambda\Theta_k^t)$$

Model architecture

The whole model is a feed-forward architecture, for each batch we feed, we compute the gradient and update the parameters. We only have one layer for this problem since we directly apply Stochastic gradient descent to the input data X.

Experimental part

Firstly we initialize the parameters

$$W[k] = 0.0001 * \begin{bmatrix} 1 \\ 1 \\ 1 \\ . \\ . \\ . \\ 1 \\ 1 \end{bmatrix}$$

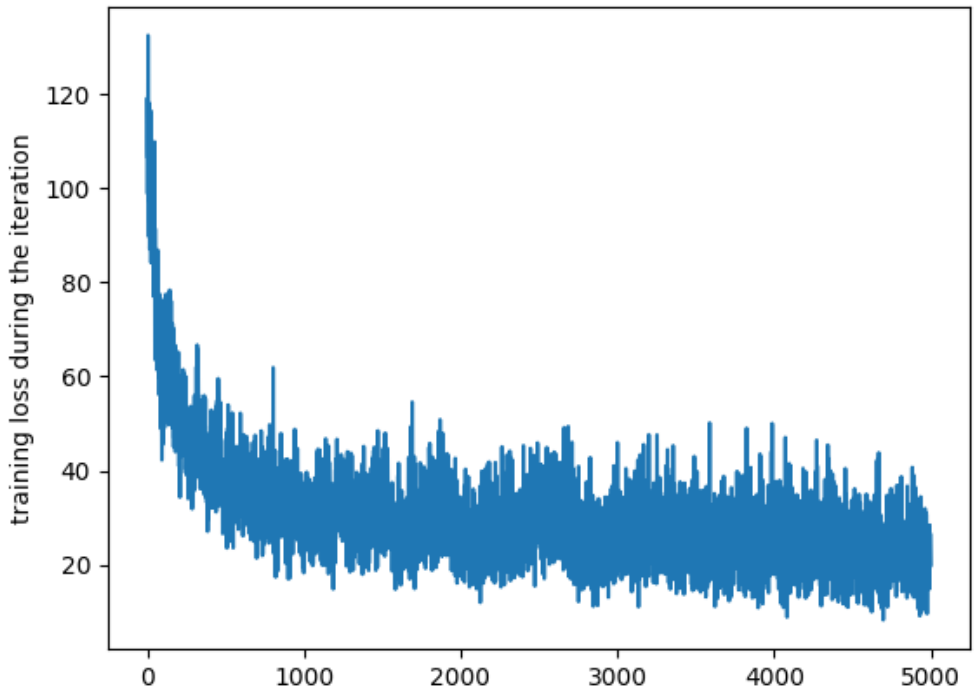
We set the parameters as below: learning rate

$$\eta = 0.01$$

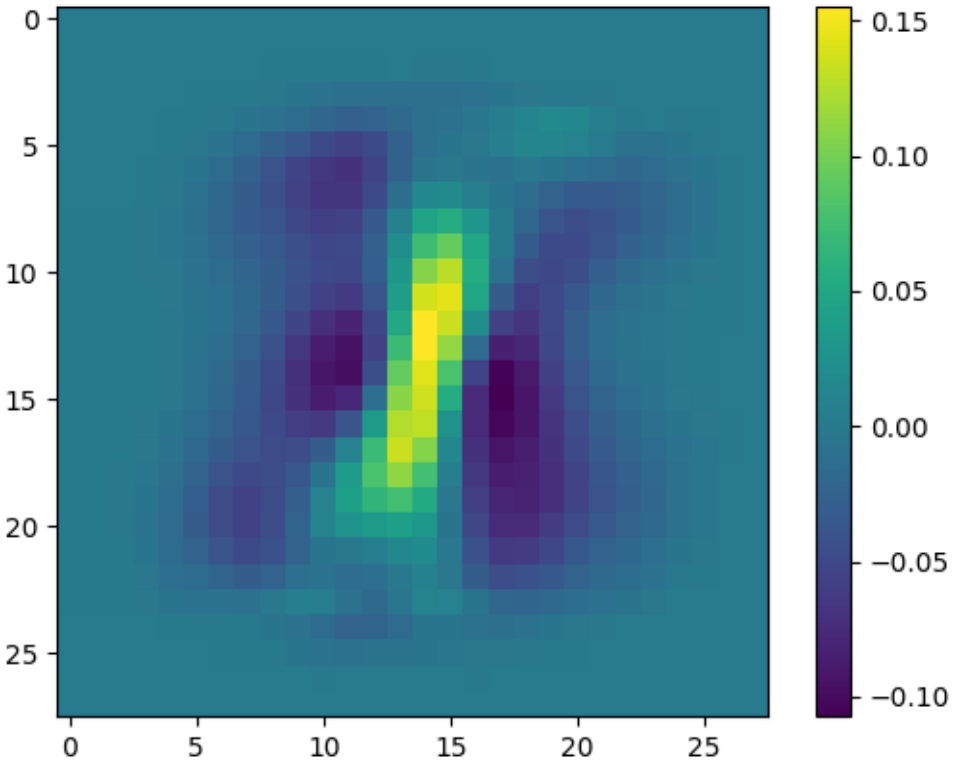
For the regularization term

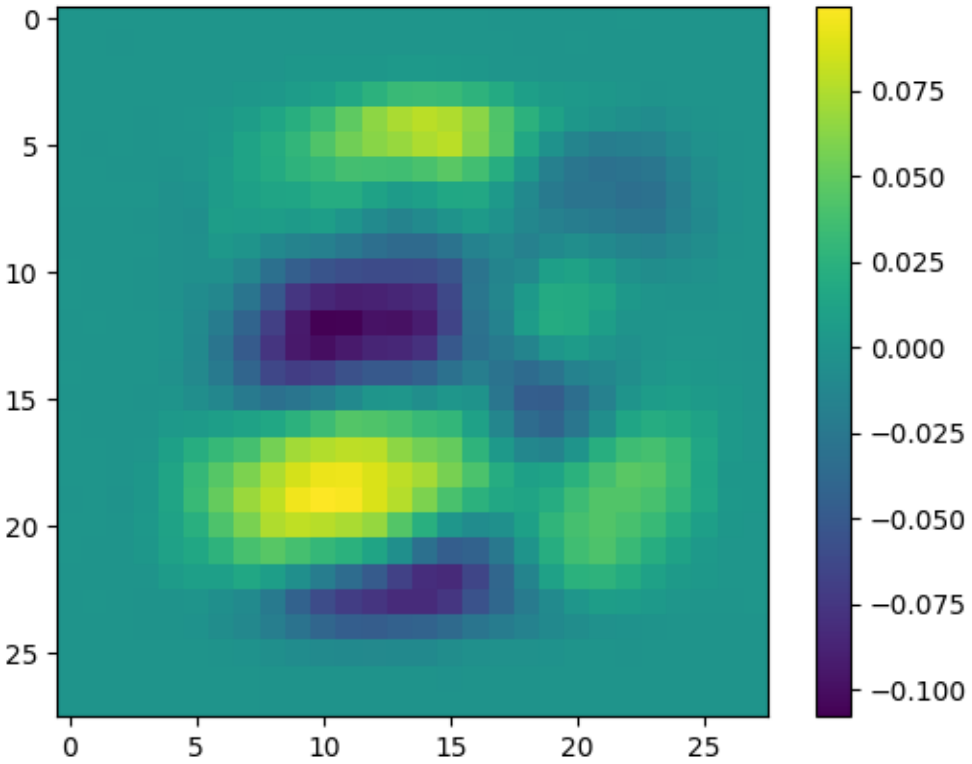
$$\lambda = 0.00001$$

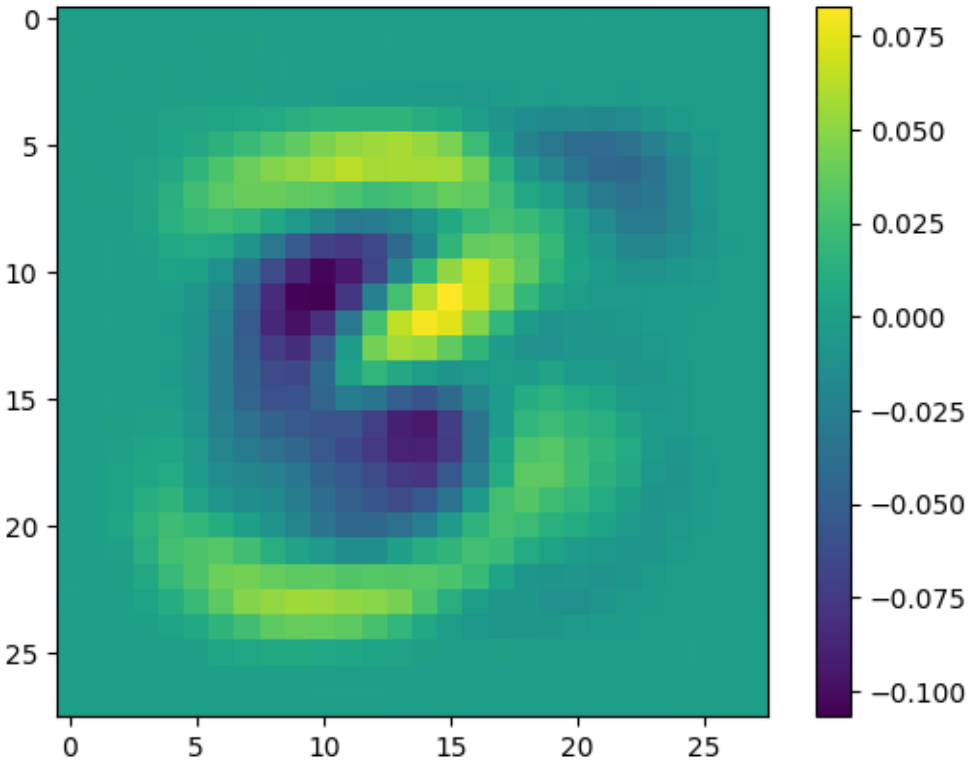
Below is the training error over iterations

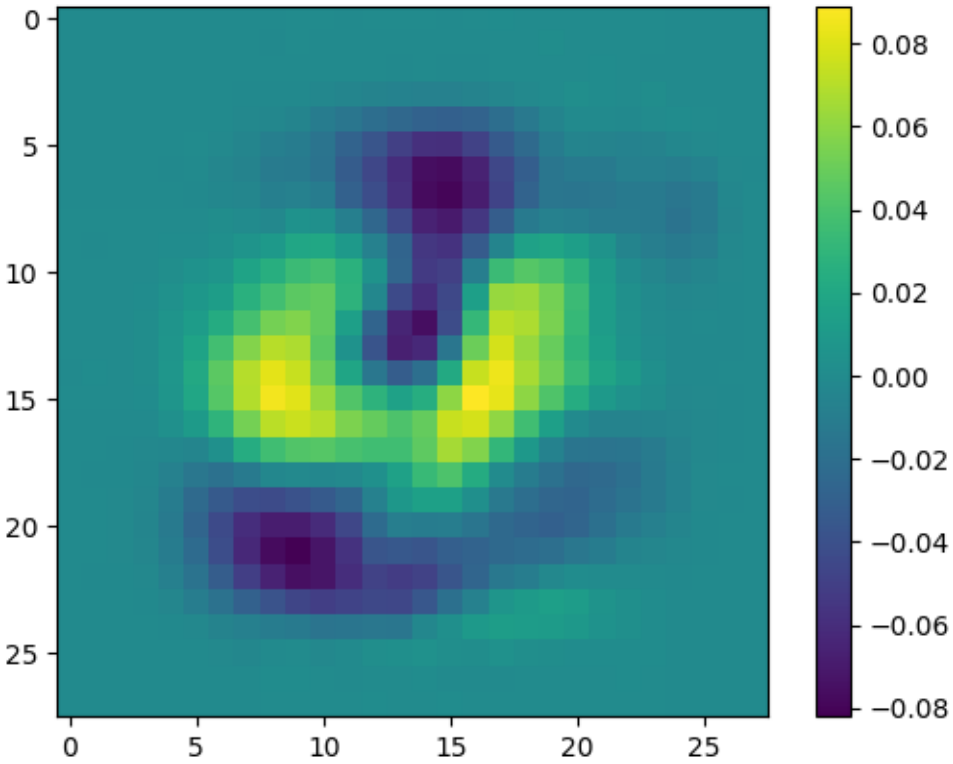


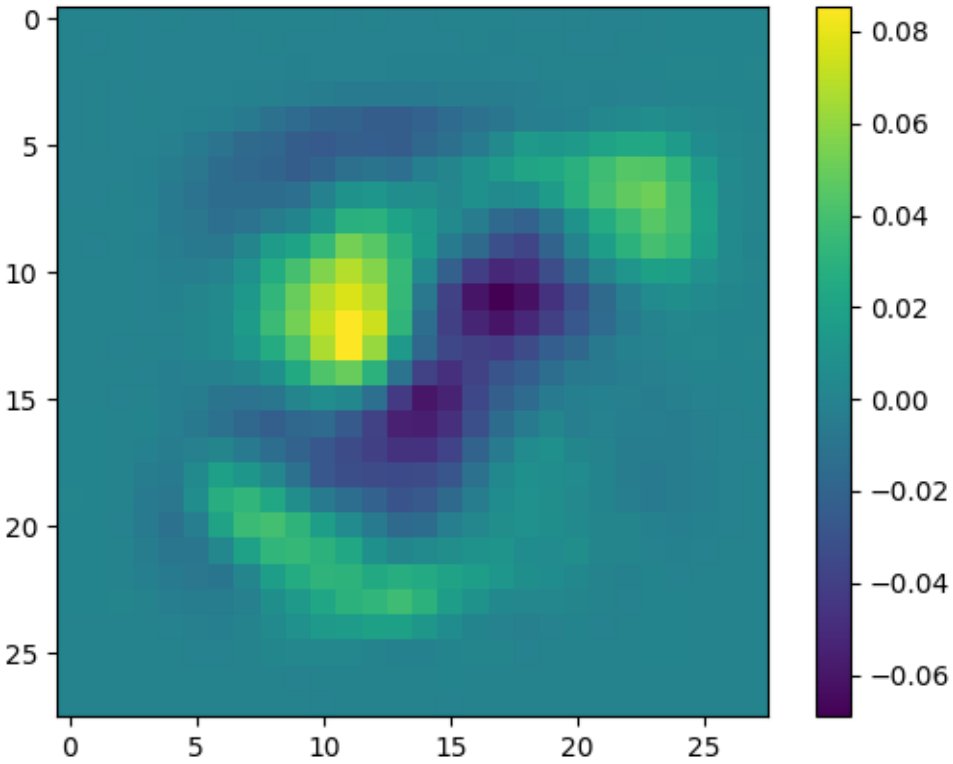
Below is the learnt weight for each class. For digit 1 For digit 2 For digit 3 For digit 4 For digit 5


0
5
10
15
20
25
0
5
10
15
20
25
0.15
0.10
0.05
0.00
−0.05
−0.10


0
5
10
15
20
25
0
5
10
15
20
25
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100


0
5
10
15
20
25
0
5
10
15
20
25
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100


0
5
10
15
20
25
0
5
10
15
20
25
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08


0
5
10
15
20
25
0
5
10
15
20
25
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06

```
[[1043    7    7    1    6]
 [   9  884   42   37   18]
 [   6    9  945    9   61]
 [  14    5    0  960    4]
 [   5   19   38   13  840]]
```

Classification result

Belos is the confusion matrix for the testing result
Therefore,

$$error(digit = 1) = 1 - \frac{1043}{1043 + 7 + 7 + 1 + 6} = 1.9\%$$

$$error(digit = 2) = 1 - \frac{884}{9 + 884 + 42 + 37 + 18} = 10.7\%$$

$$error(digit = 3) = 1 - \frac{945}{6 + 9 + 945 + 9 + 61} = 8.25\%$$

$$error(digit = 4) = 1 - \frac{960}{14 + 5 + 0 + 960 + 4} = 2.3\%$$

$$error(digit = 5) = 1 - \frac{840}{5 + 19 + 38 + 13 + 840} = 7.18\%$$

$$avg(error) = 1 - \frac{1043 + 884 + 945 + 960 + 840}{4982} = 6.2\%$$