# Program Assignment 2: Intro to Deep Learning

Multi-class Logistic Regression on MNIST

Due on Monday, Feb. 23th, at the beginning of class

*Professor Qiang Ji*

**Keyi Liu**(Master Student)

Feburary 21, 2017

# 1 Introduction

In this programming assignment, I implemented the techniques of a multi-class logistic regression for handwritten digit classification using the MNIST dataset. The classifier takes an image of a handwritten numerical digit between 1 and 5 as input and classify it into one of 5 classes corresponding to digit 1 to 5 respectively. Given the training data, I trained a multi-class logistic regressor with the stochastic gradient descent method and then evaluate its performance on the given testing data. For training, the discriminant function regression parameters are $\mathbf{\Theta} = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}$, one for each class, and their corresponding biases.

The objective function is the negative log conditional likelihood of the training data,

$$\begin{aligned} \mathcal{L}(\mathcal{D} : \mathbf{\Theta}) &= -\sum_{m=1}^{M} \log \prod_{k=1}^{K} P(y_m = k | x_m, \mathbf{\Theta}) + R(\mathbf{\Theta}) \\ &= -\sum_{m=1}^{M} \sum_{k=1}^{K} \left[ \log \frac{\exp(\theta_k^T x_m)}{\sum_{j=1}^{K} \exp(\theta_j^T x_m)} \right]^{y_{mk}} + \frac{1}{2} \lambda \mathbf{\Theta}^2 \end{aligned} \tag{1}$$

where the $R(\mathbf{\Theta})$ is the regularization term, which is the $L-2$ norm, then, we need to adjust the parameters that,

$$\mathbf{\Theta}^* = \arg\min_{\mathbf{\Theta}} \mathcal{L}(\mathcal{D} : \mathbf{\Theta}) \tag{2}$$

We take the partial gradient with respect to each weight $\theta_k$,

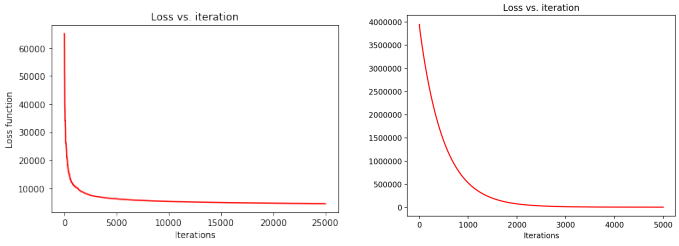
$$\frac{\partial \mathcal{L}(\mathcal{D} : \mathbf{\Theta})}{\partial \theta_k} = -\sum_{m=1}^{M} x_m \left[ y_{mk} - \frac{\exp(\theta_k^T x_m)}{\sum_{j=1}^{K} \exp(\theta_j^T x_m)} \right] + \lambda \theta_k \tag{3}$$

So, according to the gradient descent rules,

$$\mathbf{\Theta}^{t+1} = \mathbf{\Theta}^{t+1} - \eta \frac{\partial \mathcal{L}(\mathcal{D} : \mathbf{\Theta})}{\partial \mathbf{\Theta}} \tag{4}$$

# 2 Experiment

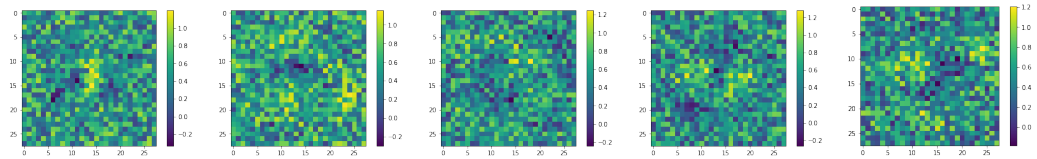
I choose the batch size to be 50, learning rate 0.0002, and learning step to be 5000 and 10000, I found that the curve actually converge reasonably enough at step 5000, we do not need to train additional step after 5000, and with the corresponding test accuracy 0.9389, 0.9524.



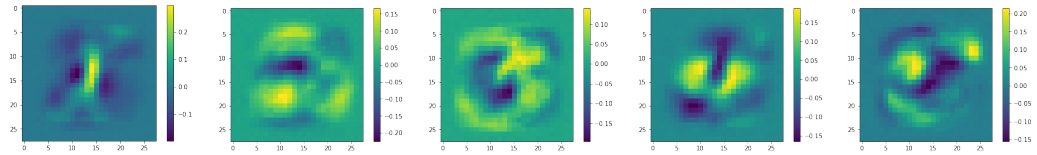

(a) 25000 steps (b) 5000 steps

Figure 1: The learning curves for training

Then I add and choose different $\lambda = \{0.002, 0.1\}$ to stress different emphasis on the regularization term. The corresponding test accuracy is 0.9418, 0.9524, the weighs are shown as the following,



(a) 1 for $\lambda = 0.002$ (b) 2 for $\lambda = 0.002$ (c) 3 for $\lambda = 0.002$ (d) 4for $\lambda = 0.002$ (e) 5 for $\lambda = 0.002$



(f) 1 for $\lambda = 2$ (g) 2 for $\lambda = 2$ (h) 3 for $\lambda = 2$ (i) 4 for $\lambda = 2$ (j) 5 for $\lambda = 2$

Figure 2: The weights with respect to each class $1 - 5$

We can conclude that, with an appropriate regularization term, the white pixels that have nothing to do with the classification will have lower weights, and the pixels that have close relationship with each digits will tend to have large weights, that is why the weights could take on the digit's look.

The test accuracy with respect to each class, and the overall accuracy are listed in the following table,

| *Digits* | 1 | 2 | 3 | 4 | 5 | overall |
|---|---|---|---|---|---|---|
| *Accuracy* | 0.9840 | 0.9303 | 0.9204 | 0.9715 | 0.8973 | 0.9524 |

Table 1: testing accuracy of each digit