CS 541: Deep Learning
Assignment 3

Ruojun Li

February 8, 2019

Content

1 Problem 1

Answer to question 1:

Newton's method [10 points]: Show that, for a 2-layer linear neural network and the same cost function J as from Homework 2, Newton's method (see Equation 4:12 in Deep Learning) will converge to the optimal solution in 1 iteration no matter what the starting point w0 of the search is.

$$J(\theta) = \frac{1}{4}\sum(f^*(x) - f(x,\theta))^2 \tag{1}$$

$$\hat{y} = f_w(x) = w^T x$$

Optimal Solution:

$$w^* = (X^T X)^{-1} X^T y \tag{2}$$

Given Equation 4.12:

$$x^* = x^{(0)} - H(f)(x^{(0)})^{-1}\nabla_x f(x^{(0)}) \tag{3}$$

To update the w in 2-layer linear NN in 1 iteration, transferring the equation 4.12 to:

$$w^{(1)} = w^{(0)} - H(f)(w^{(0)})^{-1}\nabla_w f(w^{(0)}) \tag{4}$$

$$= w^{(0)} - \frac{\nabla_{w^{(0)}} J(w^{(0)})}{\nabla_{w^{(0)}}\nabla_{w^{(0)}} J(w^{(0)})} \tag{5}$$

$$= w^{(0)} + \frac{f^* x^{(0)} - x_{(o)}{}^{2w^{(0)}}}{x^2_{(0)}} \tag{6}$$

$$= w^{(0)} - w^{(0)} + \frac{f^*}{x^{(0)}} \tag{7}$$

$$= X^{(-1)} y \tag{8}$$

If the Matrix X exsits generalized left inverse, $X^{(-1)} = (X^T X)^{-1} X^T$

$$= (X^T X)^{-1} X^T y \tag{9}$$

When the data has generalized left inverse, Newton method gives the optimal solution just one iteration.

2 Problem 2

Complete the total gradient of the softmax classification issue:
Answer to question 2:

$$\nabla_{wl} f_{CE}(W) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c} y_k^i \nabla_{wl} \log \hat{y_k}^{(i)} \tag{10}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{c} y_k^i \frac{\nabla \hat{y_k}^{(i)}}{y_k{}^{(i)}} \tag{11}$$

When

$$k \neq l$$

:

$$\nabla_{w_l} \hat{y_k}^{(i)} = \nabla_{w_l} \frac{exp(X^T w_k)}{\sum_{k'=1}^{C} exp(X^T w_{k'})} \tag{12}$$

$$= -\frac{exp(X^T w_k) \nabla_{w_l} \sum_{k'=1}^{C} exp(X^T w_{k'})}{\sum_{k'=1}^{C} exp(X^T w_{k'})^2} \tag{13}$$

$$= -\frac{exp(X^T w_k)}{\sum_{k'=1}^{C} exp(X^T w_{k'})} \frac{\nabla_{w_l} exp(X^T w_l)}{\sum_{k'=1}^{C} exp(X^T w_{k'})} \tag{14}$$

$$= -\hat{y_k}^{(i)} \frac{X^{(i)} exp(X^T w_l)}{\sum_{k'=1}^{C} exp(X^T w_{k'})} \tag{15}$$

$$= -\hat{y_k}^{(i)} X^{(i)} \hat{y_l}^{(i)} \tag{16}$$

When $k = l$:

$$\nabla_{w_l} \hat{y_k}^{(i)} = \nabla_{w_l} \frac{exp(X^T w_k)}{\sum_{k'=1}^{C} exp(X^T w_{k'})} \tag{17}$$

$$= \frac{\nabla_{w_l} exp(X^T w_k) \sum_{k'=1}^{C} exp(X^T w_{k'})}{\sum_{k'=1}^{C} exp(X^T w_{k'})^2} - \frac{exp(X^T w_k) \nabla_{w_l} \sum_{k'=1}^{C} exp(X^T w_{k'})}{\sum_{k'=1}^{C} exp(X^T w_{k'})^2} \tag{18}$$

(The second term is given by the equations in

$$k \neq l$$

)

$$\tag{19}$$

$$= \frac{\nabla_{w_l} exp(X^T w_l)}{\sum_{k'=1}^{C} exp(X^T w_{k'})} \frac{\sum_{k'=1}^{C} exp(X^T w_{k'})}{\sum_{k'=1}^{C} exp(X^T w_{k'})} - \hat{y_l}^{(i)} X^{(i)} \hat{y_l}^{(i)} \tag{20}$$

$$= X^{(i)} \frac{exp(X^T w_l)}{\sum_{k'=1}^{C} exp(X^T w_{k'})} - \hat{y_l}^{(i)} X^{(i)} \hat{y_l}^{(l)} \tag{21}$$

$$= X^{(i)} \hat{y_l}^{(i)} - \hat{y_l}^{(i)} X^{(i)} \hat{y_l}^{(i)} \tag{22}$$

$$\nabla_{wl} f_{CE}(W) = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{c} y_k^i \frac{\nabla \hat{y_k}^{(i)}}{\hat{y_k}^{(i)}} \tag{23}$$

$$= \frac{1}{n} \sum_{i=1}^{n} y_l^i \{ X^{(i)} \hat{y_l}^{(i)} - \hat{y_l}^{(i)} X^{(i)} \hat{y_l}^{(i)} \} / \hat{y_l}^{(i)} + \frac{1}{n} \sum_{i=1}^{n} y_k^i \{ - \hat{y_l}^{(i)} X^{(i)} \hat{y_k}^{(i)} \} / \hat{y_k}^{(i)} \tag{24}$$

The first term exists when

$$k = l$$

; the second term exists when

$$k \neq l$$

.

$$= \frac{1}{n}\sum_{i=1}^{n} y_l^i \{X^{(i)} - \hat{y_l}^{(i)} X^{(i)}\} + \frac{1}{n}\sum_{i=1}^{n} y_k^i \{ - \hat{y_l}^{(i)} X^{(i)}\} \tag{25}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \{{y_l}^{(i)} X^{(i)} - (y_l^i + y_k^i) X^{(i)} \hat{y_l}^{(i)}\} \tag{26}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \{{y_l}^{(i)} X^{(i)} - X^{(i)} \hat{y_l}^{(i)}\} \tag{27}$$

Makes

$$k = k + l$$

,it covers all classes. Since it's normalization softmax, the classfication's summary should be 1,

$$y_k^i = 1$$

$$= \frac{1}{n}\sum_{i=1}^{n} \{{y_l}^{(i)} X^{(i)} - X^{(i)} \hat{y_l}^{(i)}\} \tag{28}$$

$$= \frac{1}{n}\sum_{i=1}^{n} X^{(i)} \{{y_l}^{(i)} - \hat{y_l}^{(i)}\} \tag{29}$$

3 Problem 3

Answer to question 3:

3.1 Parameter I

The training result: Minibatch size=100,
learning rate=0.1
n epoch=10
alpha=0.1
test entropy loss:21.4 test class loss:11.67%

3.2 Parameter II

Minibatch size=1,
learning rate=0.1
n epoch=10
alpha=0.1
test entropy loss:21.05 test class loss:5.57%

3.3 Parameter III

Minibatch size=1,
learning rate=0.01
n epoch=10
alpha=0.1
test entropy loss:20.72 test class loss:17.15%

This part is to summarize the results running on the MNIST dataset using the code provided in the assignment.

The result will include related hyper-parameters of the training progress and network architecture.

4 Parameter: Epoch = 30, batch_size = 10, $\eta = 3.0$

Using the parameter defined in the assignment, run 3 times and record these parameters.

The network structure remains the same with the 30 hidden units.

Table 1: Result Table for Running Three Times

Trial Number	Predict Number	Prediction Rate
1	9484	0.9484
2	9463	0.9463
3	9472	0.9472
Median	9472	0.9472

According to the table 1, the median result on test dataset is 9472/10000.

5 Hyper-parameter and Network Topology Result

5.1 Parameter I, Epoch = 30, $\eta = 0.1$

In this section, the parameter is set to 30 and the learning rate η set to 0.1.

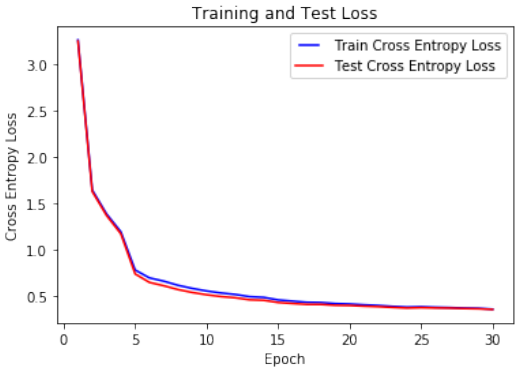

Figure 1: Cross Entropy Plot

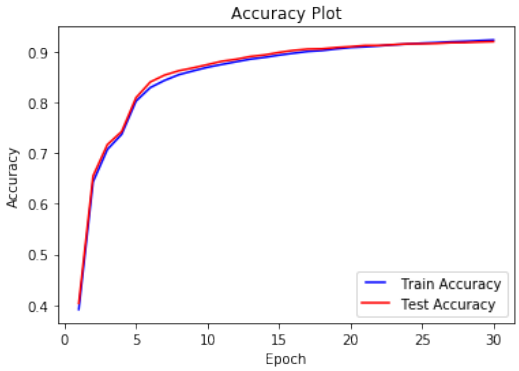

Figure 2: Accuracy Plot

The accuracy on validation data is 0.9225 (9225/10000). The confusion matrix for the classification is illustrated in the figure 3.

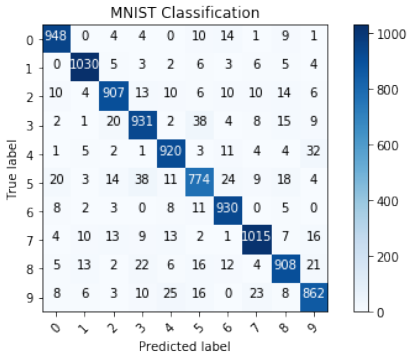

Figure 3: Classification Confusion Matrix on Validation Data

In the training progress of the preliminary stage, the convergence is a bit slower and the cross-entropy loss is higher than 1.

This stage only changed the learning rate η to 0.1, which illustrated a slower convergence speed.

5.2 Parameter II, Epoch = 30, Hidden Units = 100, $\eta = 3.0$

Set the number of neurons to 100 at the hidden layer, with the learning rate $\eta = 3$ as the default value.

Figure 4: Cross Entropy Plot

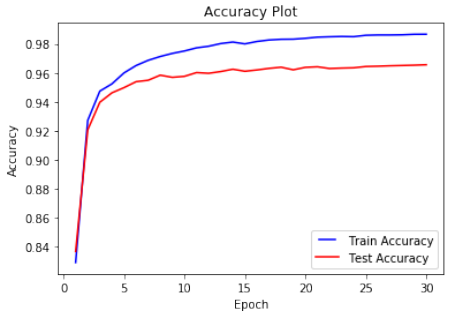

Figure 5: Accuracy Plot

The accuracy on validation data is 0.9662 (9662/10000). The confusion matrix for the classification is illustrated in the figure 6.

5.3 Parameter III, Epoch = 70, Batch Size = 16, $\eta = 3$

Based on the parameters obtained in the second subsection, set the Epoch to 70, the batch size to 16, the hidden units to 100.

The modification of the neural network is to verify the tuning parameters based on the good parameters can achieve better training results.

The accuracy on validation data is 0.9644 (9644/10000). The confusion matrix for the classification is illustrated in the figure 6.

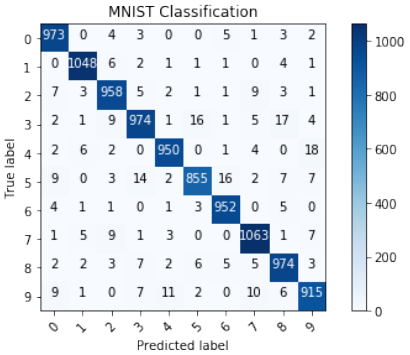

Figure 6: Classification Confusion Matrix on Validation Data

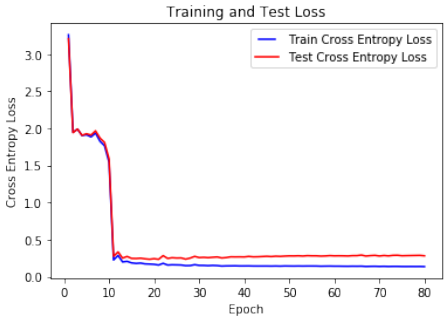

Figure 7: Cross Entropy Plot

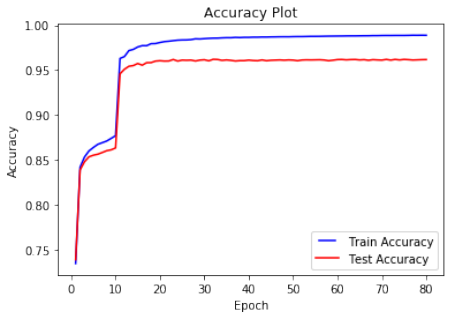

Figure 8: Accuracy Plot

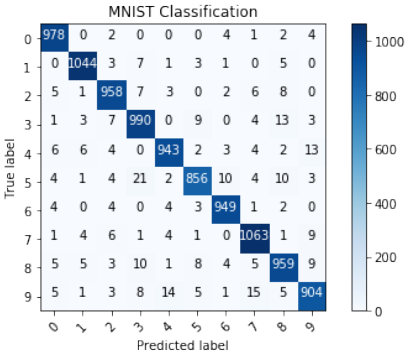

Figure 9: Classification Confusion Matrix on Validation Data

5.4 Summary on the Parameter of Neural Network

The summary of the parameters tuning on the neural network is on the following:

- With the same settings on other parameters, more epochs will contribute to higher accuracy.
- Under certain circumstances, more hidden neurons will contribute to higher convergence speed and higher accuracy.
- The change of the learning rate will lead to different convergence condition, even divergence.