DD2424 Deep Learning in Data Science
Assignment 1

Rui Shi
srui@kth.se

April 3, 2022

1 Introduction

In this assignment, all the functions and codes are implemented on Python. I trained and tested a one layer network with multiple outputs to classify images from the CIFAR-10 dataset. Mini-batch gradient descent is applied to a cost function that computes the cross-entropy loss of the classifier applied to the labelled training data and L2 regularization term on the weight matrix.

2 Check Gradients

After successfully debugging codes, I compared the numerically and analytically computed gradient metrices by examining their absolute differences based on the following equation:

$$\frac{|g_a - |g_n}{max(eps, |g_a| + |g_n|)} \tag{1}$$

where eps is a very small positive number. Figure 1 shows the relative errors are small.

```
[[0.00121301 0.00386175 0.00302505 ... 0.00202511 0.00160751 0.00074994]
 [0.00043719 0.00050799 0.00037732 ... 0.0021566  0.00093341 0.00079176]
 [0.00080395 0.00059602 0.00139554 ... 0.00055949 0.00051325 0.00047505]
 ...
 [0.00021944 0.00023458 0.0002398  ... 0.00039616 0.00056296 0.00049396]
 [0.0007845  0.0005969  0.00055309 ... 0.00040166 0.00043315 0.00056944]
 [0.00025196 0.00030337 0.00033024 ... 0.00028696 0.00031394 0.00030471]]
```

Figure 1: Relative error between a numerically computed gradient value and an analytically computed gradient value

3 Results

We used 80% of the data as training set, and 20% as validation set.

3.1 lambda = 0, n_epochs = 40, n_batch = 100, eta = .1

Dataset	Loss	Accuracy
Train	3.1486477532247164	0.4744
Validation	5.696284206120923	0.3024
Test	11.165131829479485	0.2574

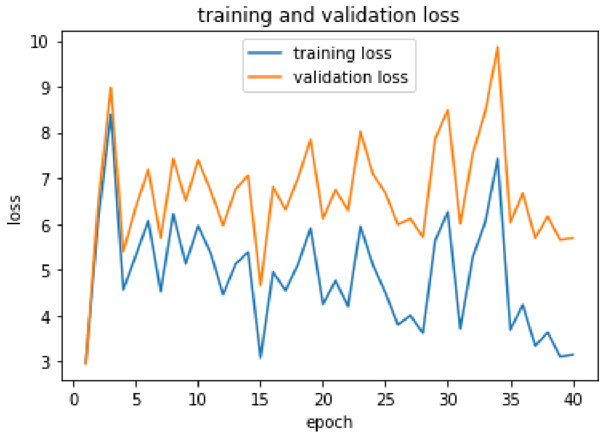

Figure 2: Graph of the loss on the training data and validation data

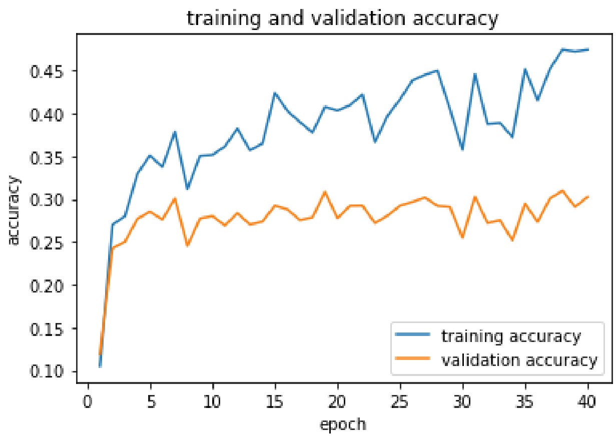

Figure 3: Graph of the accuracy on the training data and validation data

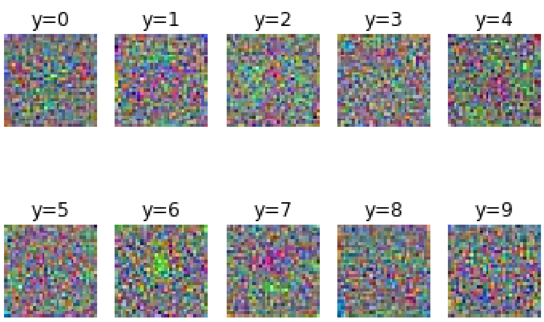

Figure 4: The learnt weight matrix

3.2 lambda = 0, n_epochs = 40, n_batch = 100, eta = .001

Dataset	Loss	Accuracy
Train	1.651787859249235	0.4339
Validation	1.839433489317682	0.3707
Test	1.8057531383466132	0.3717

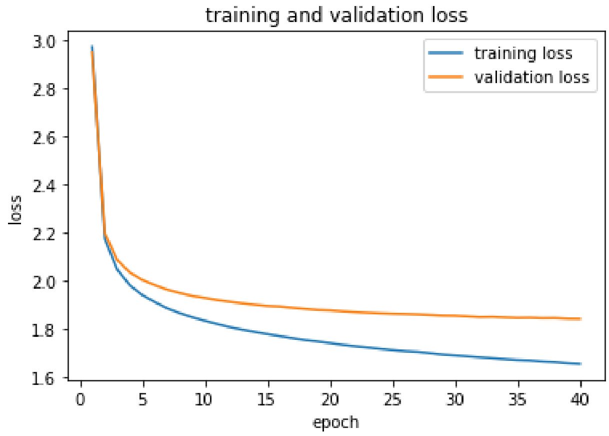

Figure 5: Graph of the loss on the training data and validation data

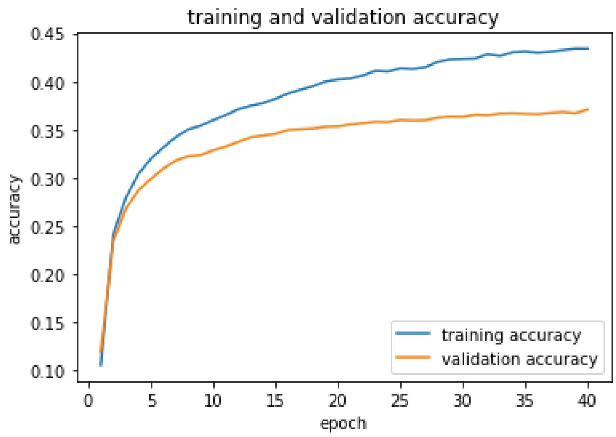

Figure 6: Graph of the accuracy on the training data and validation data

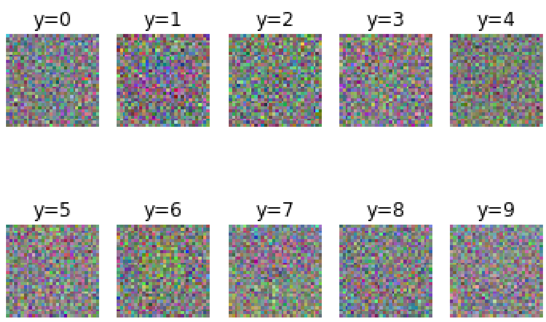

Figure 7: The learnt weight matrix

3.3 lambda = .1, n_epochs = 40, n_batch = 100, eta = .001

Dataset	Loss	Accuracy
Train	2.1003086729906992	0.0.442
Validation	2.244768468727167	0.3869
Test	2.2008715457462746	0.3876

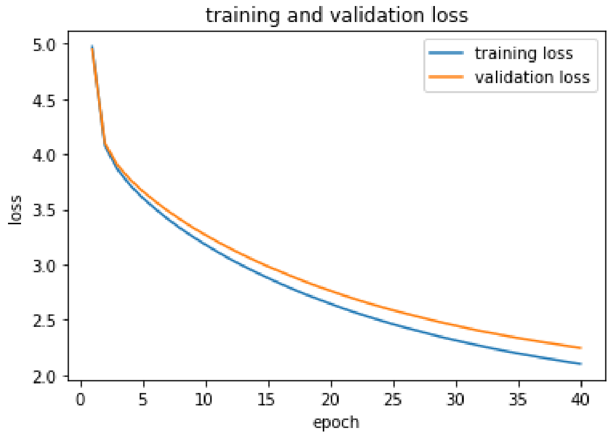

Figure 8: Graph of the loss on the training data and validation data

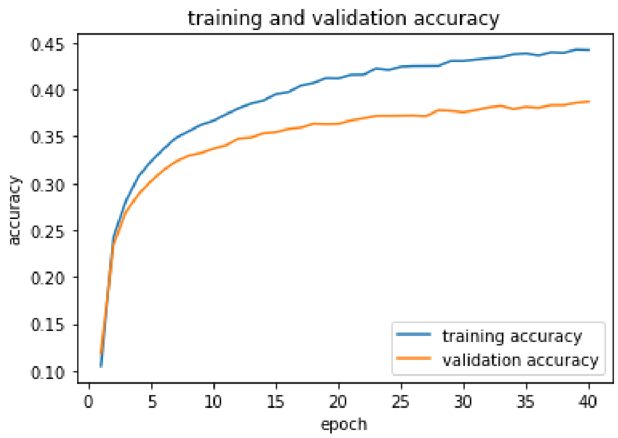

Figure 9: Graph of the accuracy on the training data and validation data

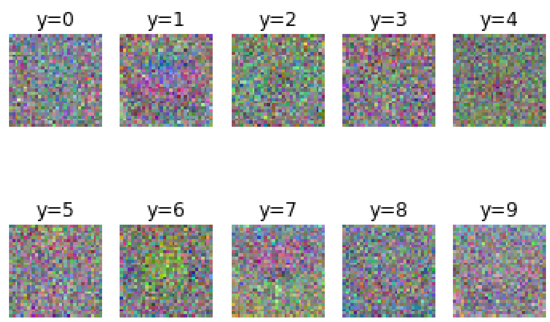

Figure 10: The learnt weight matrix

3.4 lambda = 1, n_epochs = 40, n_batch = 100, eta = .001

Dataset	Loss	Accuracy
Train	1.8986750198604445	0.4008
Validation	1.9583593660717804	0.3629
Test	1.9376345694790984	0.3784

Figure 11: Graph of the loss on the training data and validation data

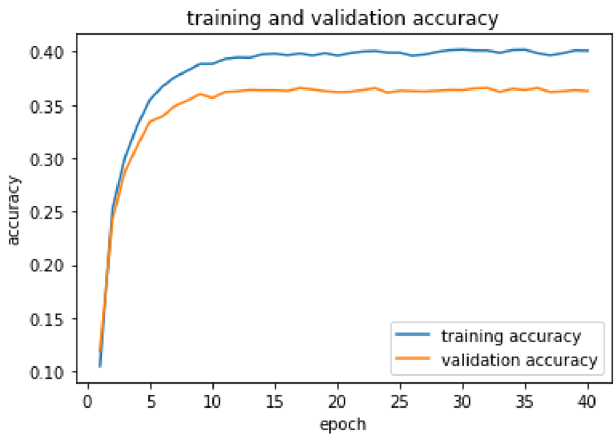

Figure 12: Graph of the accuracy on the training data and validation data

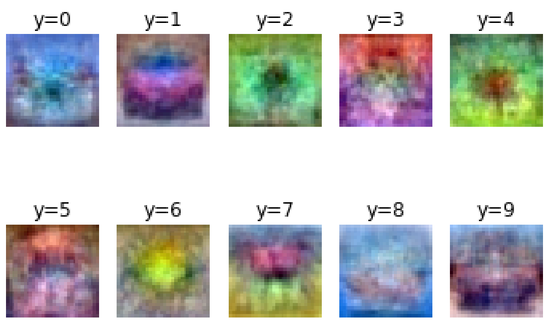

Figure 13: The learnt weight matrix

4 Analysis

As the results show, regularization terms could effectively avoid over-fitting. We can see that the accuracy becomes lower but the performances on both training and validation data are more similar.

Also we can find that it is very important to set a correct learning rate. A large learning rate may make the process of learning unstable. But a small learning rate may make the process of learning slow, so it is time consuming.