Report for Assignment 1

DD2424 Deep Learning in Data Science

Jiaying Yang
950826-9124

1 Result Analysis

In this assignment, I deal with a problem to train, validate and test a one layer network with multiple outputs to classify images from the CIFAR-10 dataset. The whole assignment is realized in these steps: loading data, initialization, evaluating the network function, computing cost and gradient, realizing mini-batch.

I determine whether I have calculated the gradients correctly by comparing the gradients calculated by my function with the ones calculated by the give function "ComputeGradsNumSlow". My function is designed analytically according to the formulas given in the assignment guidance, while the given function "ComputeGradsNumSlow" computes the gradients numerically.

Their results are very similar. I calculate the biggest difference among all elements of the gradient between the two grad_W (the code is "max(max(ngrad_b-grad_b))") and the two grad_d, and find out their values are 5.8851e-10 and 1.3488e-10, which are small enough for us to regard the results of two functions are the same.

2 Figures

In this part I include plots/figures in four following different cases:

1) ***lambda=0, n_epochs=40, n_batch=100, eta=.1***

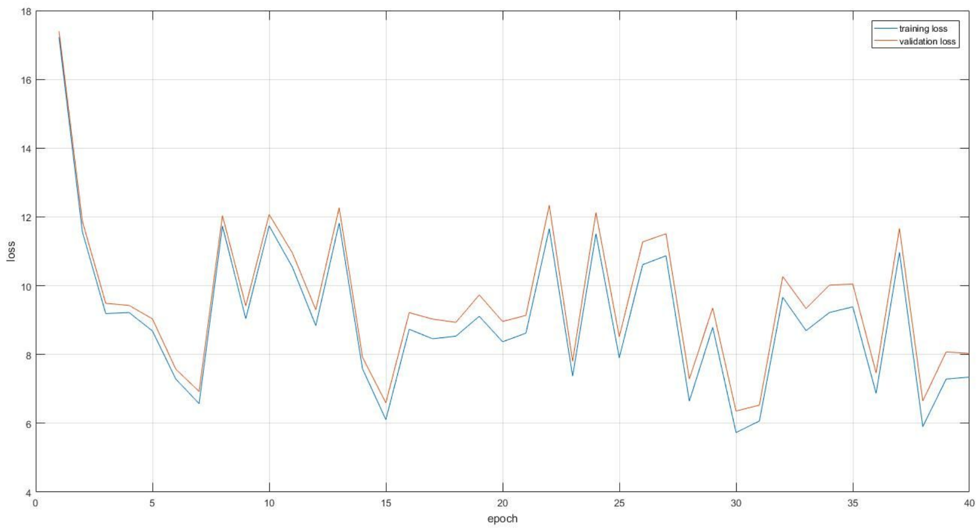

Figure 1. Loss function on training data and validation data after each epoch (case 1)

Figure 2. The learnt weight matrix after the completion of training (case 1)

Training Accuracy	Validation Accuracy	Test Accuracy
36.6%	26.25%	26.93%

Table 1. Final accuracyies (case 1)

In this case the learning rate eta is set to 0.1, which is too big, thus there us a phenomenon of zig-zag in the loss. What is more, the regularization parameter lambda is set to zero, which means at this time there is no regularization. The final test accuracy is 26.93%.

2) ***lambda=0, n_epochs=40, n_batch=100, eta=.01***

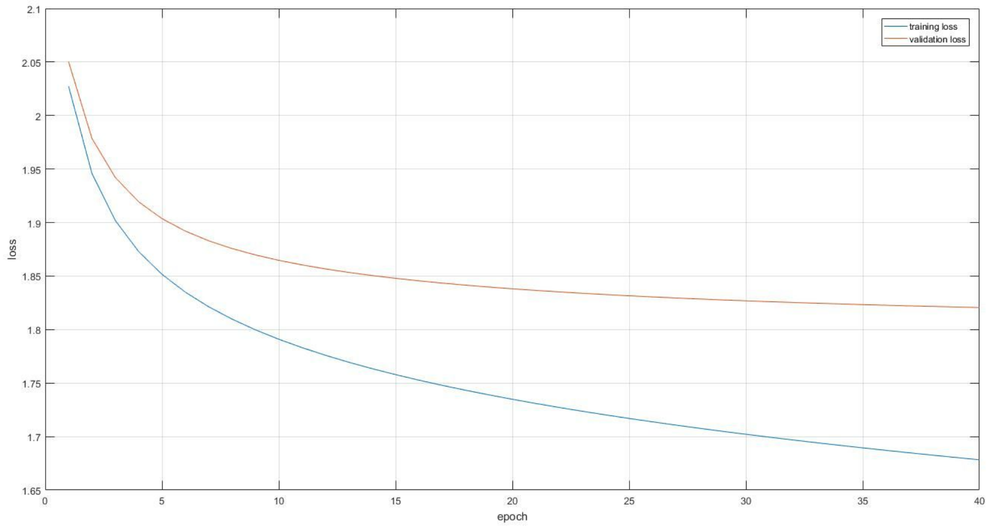

Figure 3. Loss function on training data and validation data after each epoch (case 2)

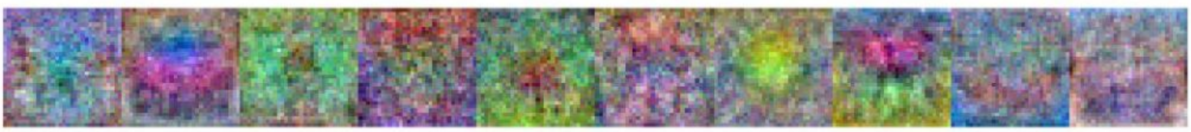

Figure 4. The learnt weight matrix after the completion of training (case 2)

Training Accuracy	Validation Accuracy	Test Accuracy
41.61%	36.6%	36.65%

Table 2. Final accuracyies (case 2)

As the decreasing of learning rate, the phenomenon of zig-zag is gone. However the gap between training accuracy and test accuracy is quite big, as is shown in Figure 3, which means that this is a case of overfitting.

3) ***lambda=.1, n_epochs=40, n_batch=100, eta=.01***

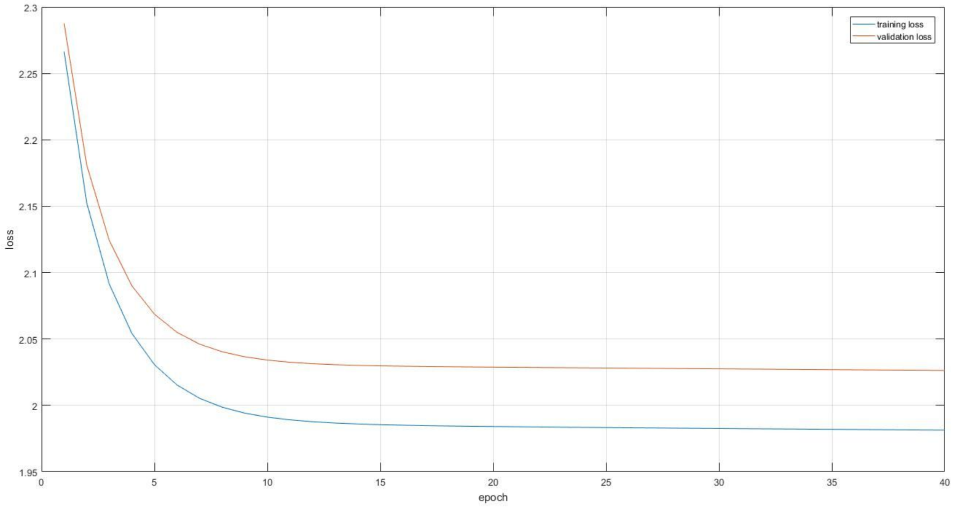

Figure 5. Loss function on training data and validation data after each epoch (case 3)

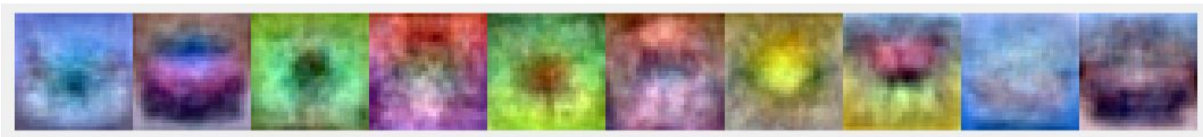

Figure 6. The learnt weight matrix after the completion of training (case 3)

Training Accuracy	Validation Accuracy	Test Accuracy
34.20%	32.03%	33.37%

Table 3. Final accuracyies (case 2)

Though compares with case 2, test accuracy in this case is only 33.37%, the gap between test accuracy and training accuracy is smaller.

4) lambda=1, n_epochs=40, n_batch=100, eta=.01

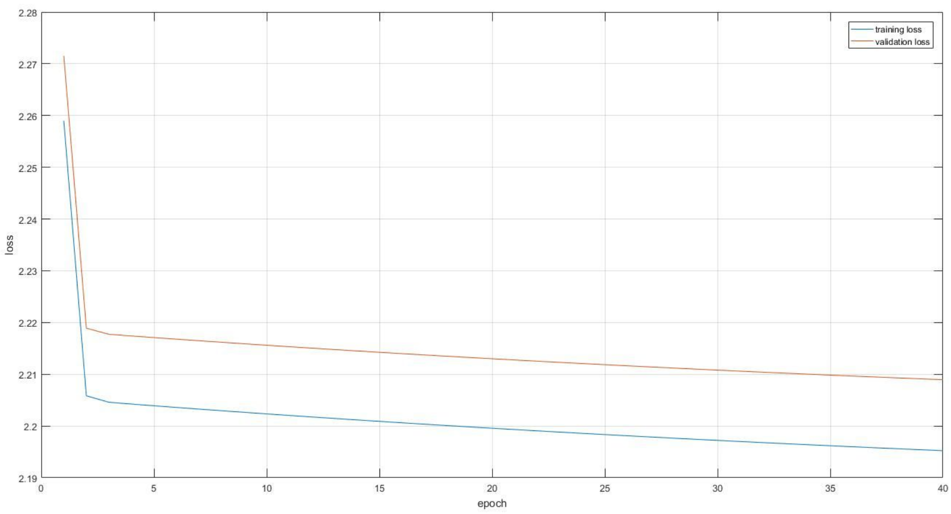

Figure 7. Loss function on training data and validation data after each epoch (case 4)

Figure 8. The learnt weight matrix after the completion of training (case 4)

Training Accuracy	Validation Accuracy	Test Accuracy
22.32%	**21.37%**	**21.92%**

Table 4. Final accuracyies (case 4)

The test accuracy in this case is 21.92%.
In summary, when the value of learning rate eta is too small, the convergence speed will be relatively slow, but the update path is smooth. However, when the value of eta is too big, the optimization can potentially diverge, and there will be an inefficient zig-zag shown in the graph of loss with respective to each epoch..