DD2424 – Assignment 1

Ao Song
April 2, 2021

1. Introduction

This assignment is to train and test a one layer network with multiple outputs to classify images from the CIFAR-10 dataset. The network was trained using mini-batch gradient descent applied to a cost function that computes the cross-entropy loss of the classifier applied to the labelled training data and an L2 regularization term on the weight matrix.

2. Implementation

In this assignment, take batch 1 in CIFAR-10 for training and batch 2 for validation. Each batch contains 10000 32 x 32 pixel colored images.

The code is in one file called assignment1.py which was implementd in Python. Numpy library has been used for matrix computation, matplotlib.pyplot has been used for plotting the results. The implementation basically follows the guides in assignment with the function names, parameters and return value.

There is a function called CheckGrads has been implemented to verify that the ComputeGradients was correctly implemented. What it do is computing the relative error between numerically and analytically computed gradient matrices with following formula (1), it will pass if all these differences are small (<1e-6).

$$\frac{|ga - gn|}{max(eps, |ga| + |ga|)} \quad (1)$$

3. Result

The code has been run 4 times with different parameters. Each run will be represented by a figure of cost and accuracy evolution during the learning process and an image with the final learned weight.

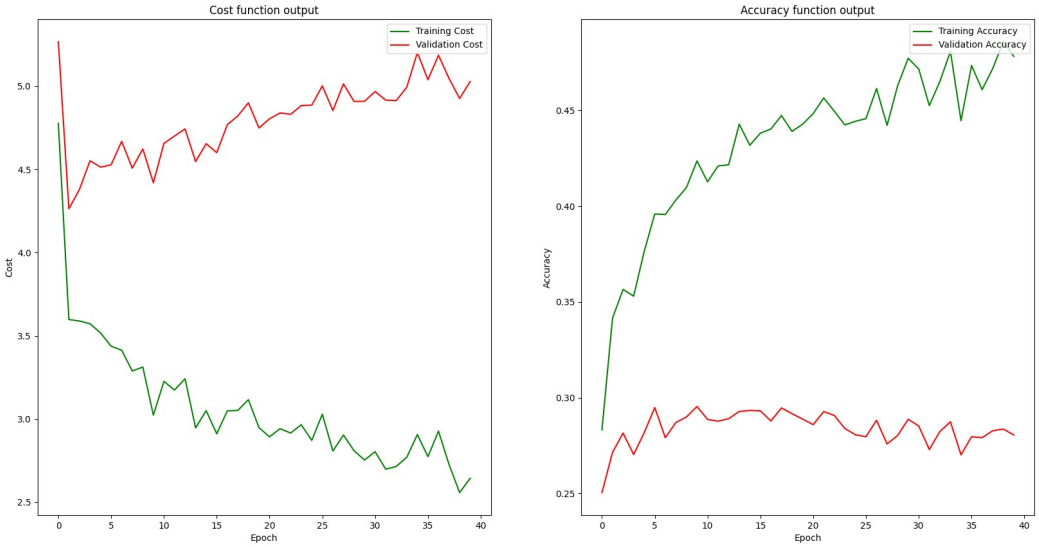
3.1 The first run

Parameters: lambda=0, n_epochs=40, n_batch=100, eta=0.1

Final result:

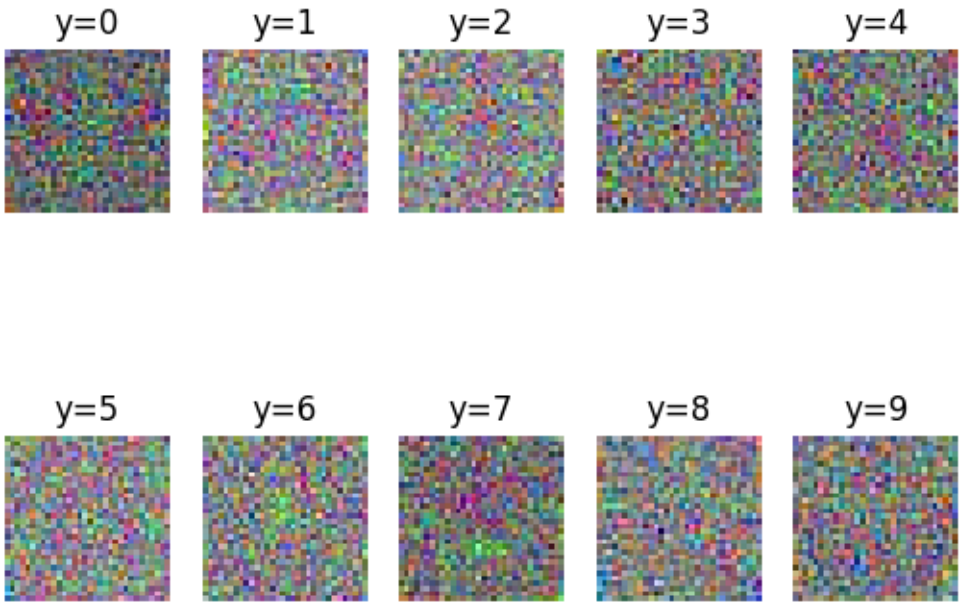
	Cost	Accuracy
Training Data	2.643	47.82%
Validation Data	5.029	28.05%

Graphs displaying cost and accuracy evolution for each epoch:



As is shown in the figure, the cost and accuracy are really unstable, and the validation curve does not totally goes as expected, this might because we have chosen a relatively big number for eta. So each step is a little too big which could be the reason.

The following image visualized the trained weight W:



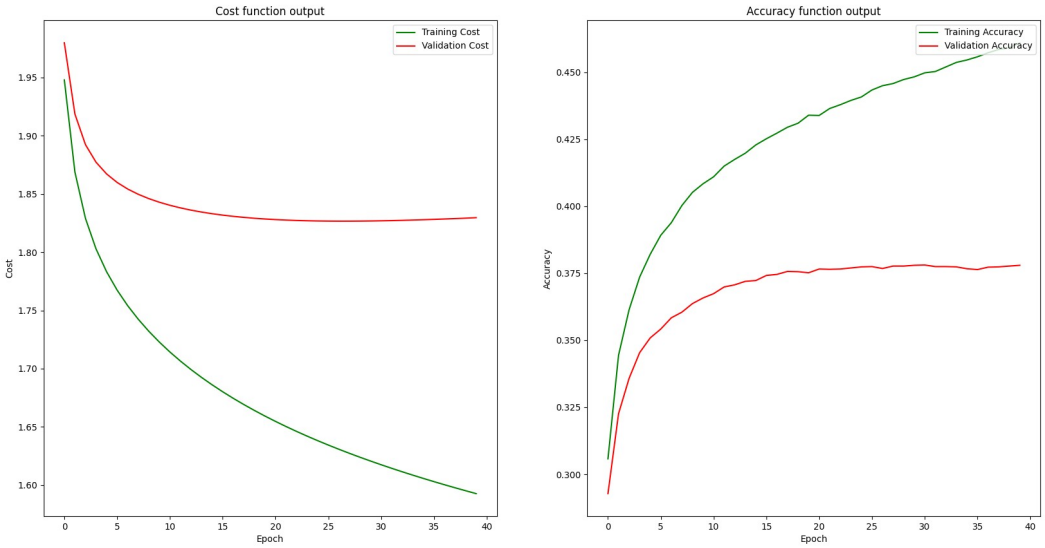
3.2 The second run

Parameters: lambda=0, n_epochs=40, n_batch=100, eta=0.001

Final result:

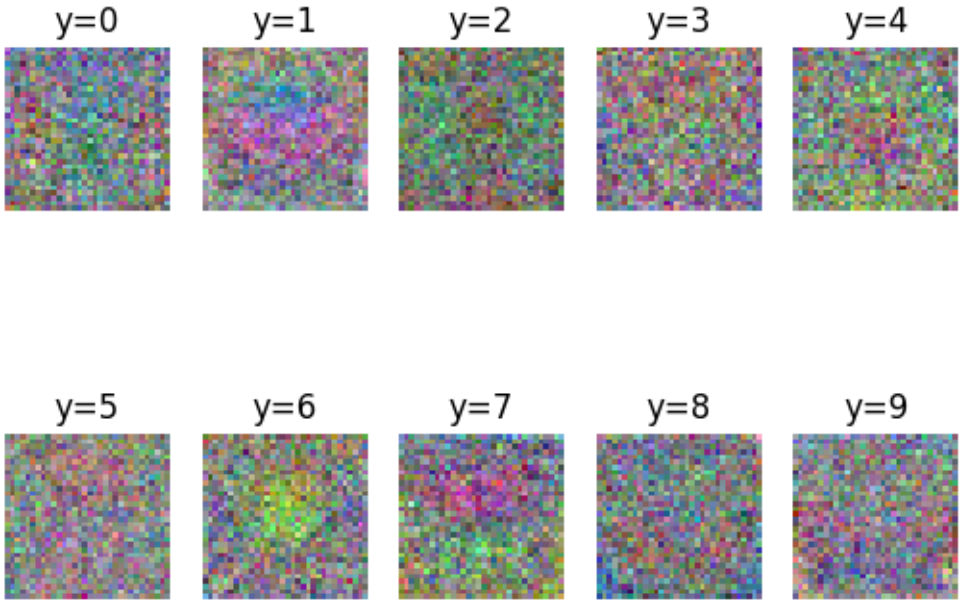
	Cost	Accuracy
Training Data	1.593	46.10%
Validation Data	1.830	37.80%

Graphs displaying cost and accuracy evolution for each epoch:



From the figure we can see that with a smaller eta, both training and validation process evolves as expected. The curve is much more stable comparing with the first run. However, there is a big gap between the curve which might suggest that there might be a overfit.

The following image visualized the trained weight W:



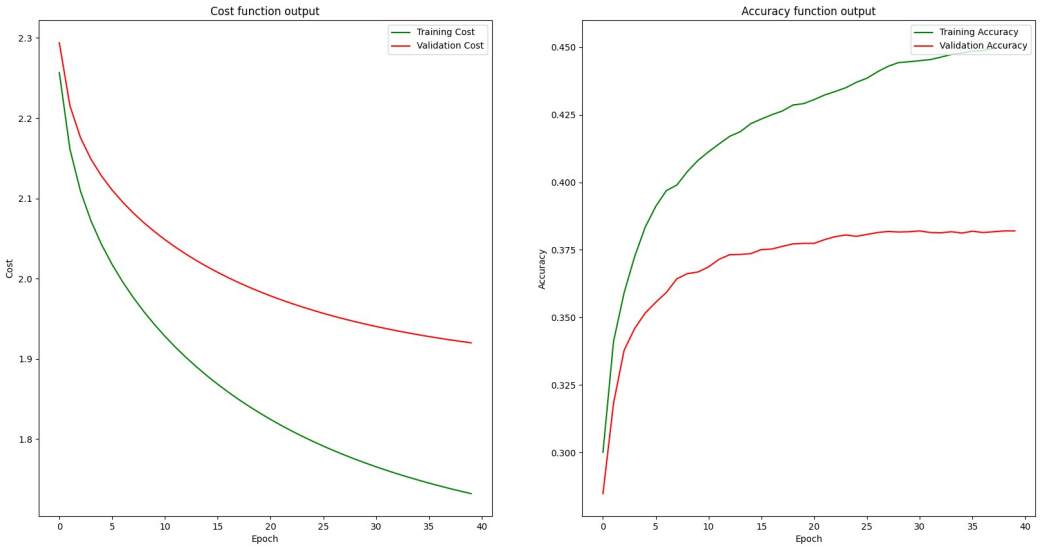
3.3 The third run

Parameters: lambda=0.1, n_epochs=40, n_batch=100, eta=0.001

Final result:

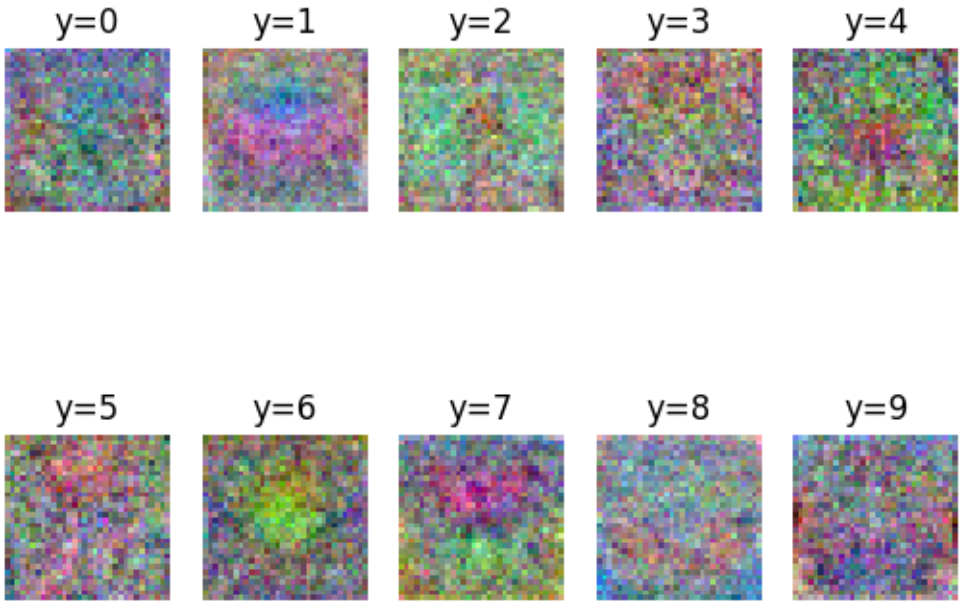
	Cost	Accuracy
Training Data	1.732	45.16%
Validation Data	1.920	38.20%

Graphs displaying cost and accuracy evolution for each epoch:



Comparing with the second run, the accuracy is similar but the gap is closer, which might suggest that with the introduction of the regularization part, the overfit will be reduced.

The following image visualized the trained weight W:



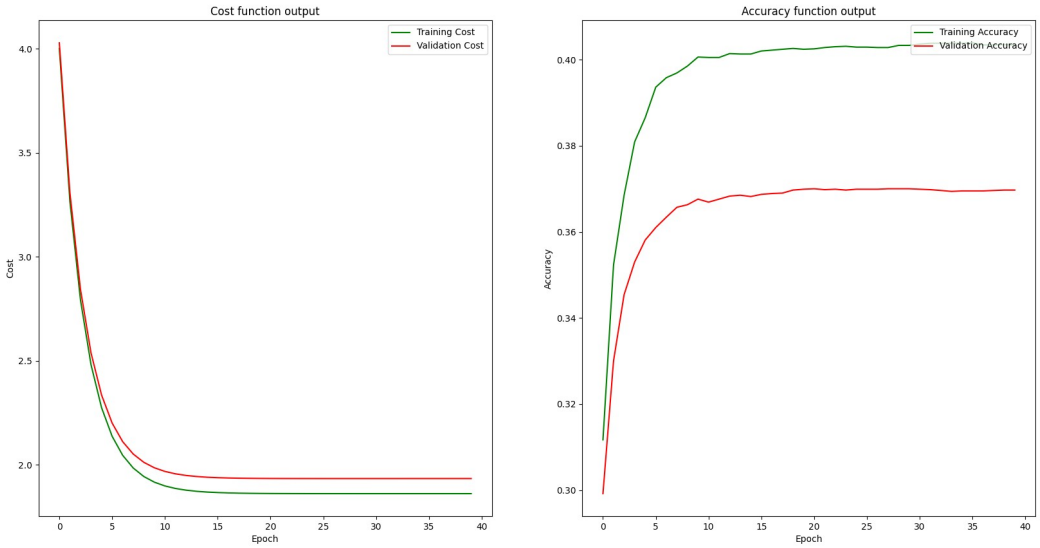
3.4 The fourth run:

Parameters: lambda=1, n_epochs=40, n_batch=100, eta=0.001

Final result:

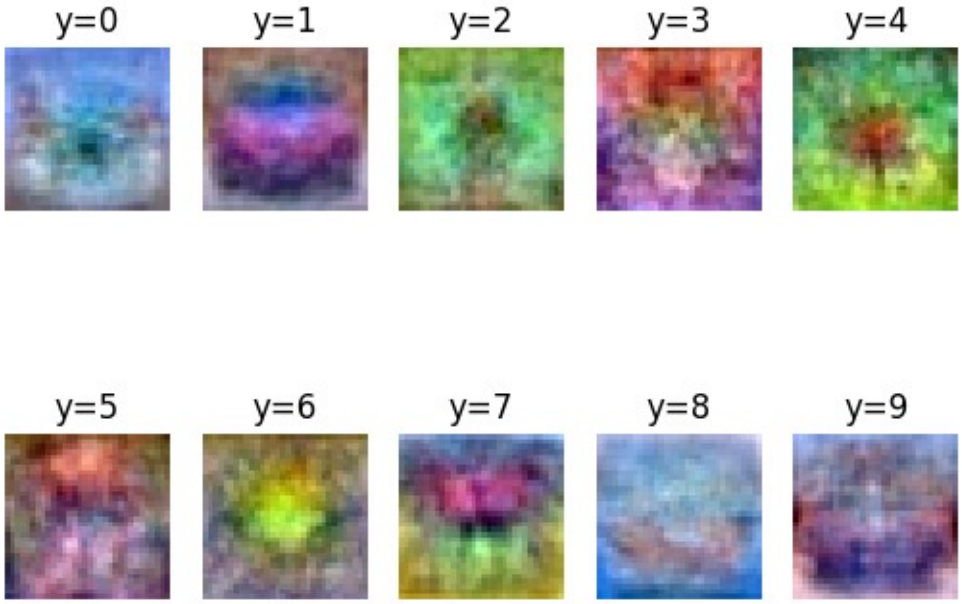
	Cost	Accuracy
Training Data	1.861	40.34%
Validation Data	1.934	36.97%

Graphs displaying cost and accuracy evolution for each epoch:



Interestingly, by increasing the lambda in regularization, though the accuracy seems drops a little bit, the training and validation process seems be more synced and the gap in between is much smaller than before.

The following image visualized the trained weight W, clearly it shows that the images are much more clear than before, noise has been reduced significantly, looks much more smooth in this run:



4. Conclusion

Comparing the 4 runs with different parameters, two major things can be found. First is the selection of eta, accuracy can be improved by using a smaller eta, but smaller eta might lead to longer computation time, so a proper eta is quite important. Another thing is that the importance of regularization, especially from the fourth run we can see that the training process is much more stable to reflect the validation situation, and the noise was reduced significantly which can be observed in the W image.

The highest validation accuracy is 38.20% which obviously has much big improvement space, but that may be because we are using a simple algorithm here in this assignment, I am looking for to improve the performance in later assignments.