Deep1

Ishaq Ezaz

April 2024

1 Implementing the functions

I have successfully implemented all necessary functions for the analytical computation of the gradient. These functions included loading the batch, preprocessing the data, evaluating the classifier, and computing both the cost and accuracy, for the gradient computation itself. The analytical gradient was compared to the slow numerical one. The comparison was done by computing the gradient for the same datasets using both implementation and then calculating the relative error for both W and b as follows:

$$\text{Relative Error} = \frac{|ga - gn|}{\max(\varepsilon, |ga| + |gn|)} \tag{1}$$

where ε was $1e - 6$.
The comparison resulted in insignificantly small relative error values, such as $1.1e-9$ for W and $2.25e-9$ for b. This indicates a negligible difference between the two implementations.

2 Training with different settings

2.1 Configuration 1

Lambda	**Eta**	**n_batch**	**Epoch**
0	0.1	100	40

Table 1: Training Parameters

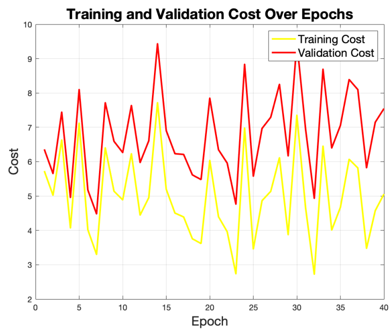

Figure 1: Cost over epochs for learning rate 0.1 and lambda 0

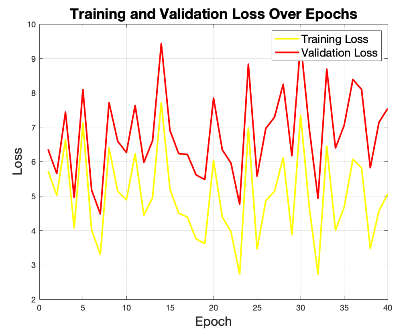

Figure 2: Loss over epochs for learning rate 0.1 and lambda 0

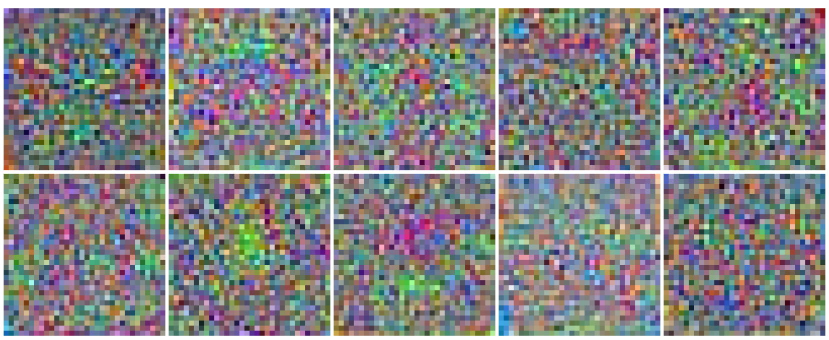

Figure 3: Visualization of the weight matrix for learning rate 0.1 and lambda 0

The cost and loss graphs are identical because there is no regularization, which means the loss directly represents the cost without any additional penalties. The spikiness of the graph is likely due to the high learning rate which introduces instability by causing significant overshoots in the weight updates. This leads to a spiky convergence and results in blurry weighted visualizations. Thus, the total accuracy of the model was relatively low, at 27.82%.

2.2 Configuration 2

Lambda	Eta	n_batch	Epoch
0	0.001	100	40

Table 2: Training Parameters

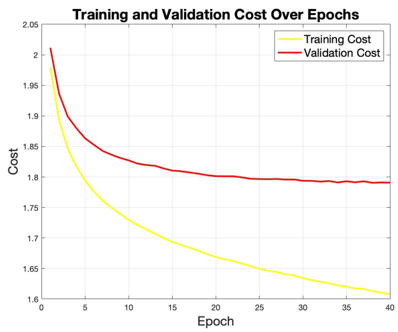

Figure 4: Cost over epochs for learning rate 0.001 and lambda 0

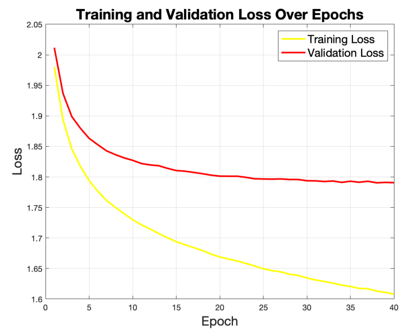

Figure 5: Loss over epochs for learning rate 0.001 and lambda 0

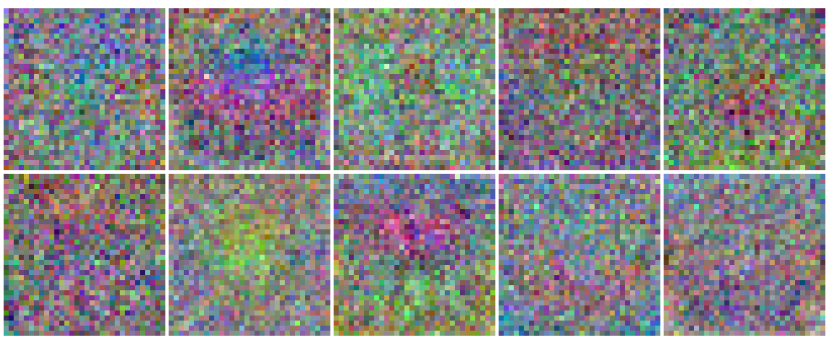

Figure 6: Visualization of the weight matrix for learning rate 0.001 and lambda 0

With a reduced learning rate, both the cost and loss graphs show a smooth gradual decline, which indicates stable learning. The absence of regularization results in identical cost and loss graphs, as seen in the previous configuration. The improved clarity in the weighted visualizations and a higher accuracy of 39.13% suggest that smaller and steadier updates allow the model to capture more useful features from the data.

2.3 Configuration 3

Lambda	Eta	n_batch	Epoch
0.1	0.001	100	40

Table 3: Training Parameters

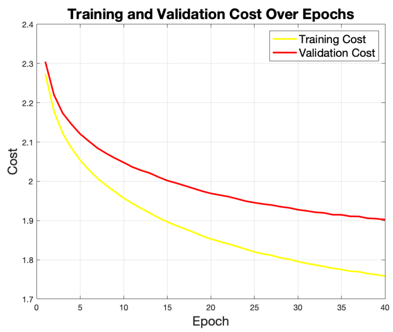

Figure 7: Cost over epochs for learning rate 0.001 and lambda 0.1

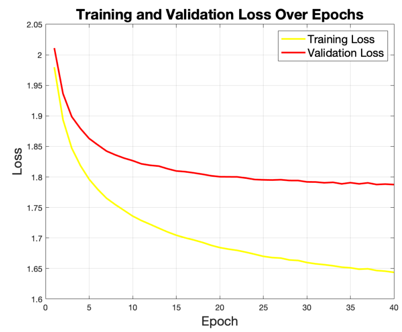

Figure 8: Loss over epochs for learning rate 0.001 and lambda 0.1

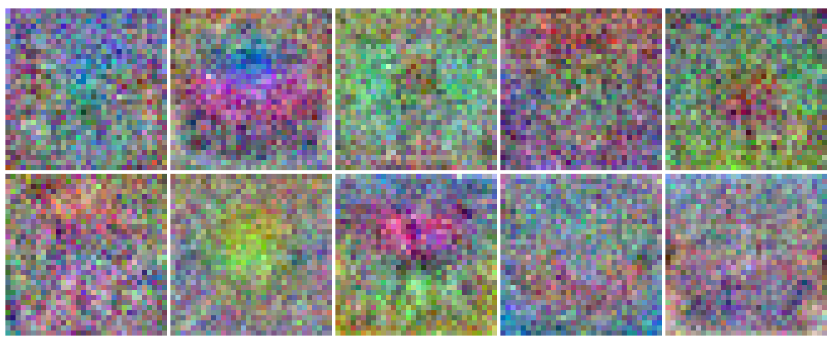

Figure 9: Visualization of the weight matrix for learning rate 0.001 and lambda 0.1

Introducing a small regularization created a noticeable difference between the cost and loss graphs due to the regularization term affecting the cost. This regularization helps control overfitting as seen in the smaller gap between training and validation metrics in the cost graph compared to the loss graph. This adjustment resulted in slightly better visual clarity in the weights and a marginal improvement in the test accuracy to 39.36%.

2.4 Configuration 4

Lambda	**Eta**	**n_batch**	**Epoch**
1	0.001	100	40

Table 4: Training Parameters

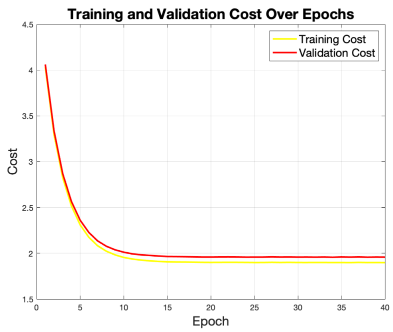

Figure 10: Cost over epochs for learning rate 0.001 and lambda 1

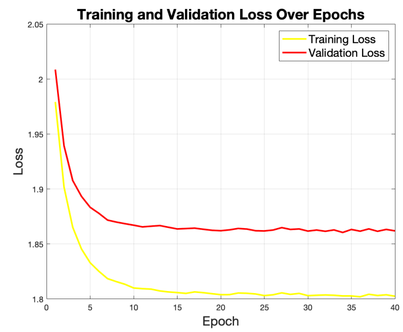

Figure 11: Loss over epochs for learning rate 0.001 and lambda 1

Figure 12: Visualization of the weight matrix for learning rate 0.001 and lambda 1

Higher regularization tightens the gap between training and validation costs, which effectively controls overfitting and promotes generalization. The graphs show a steep initial decline that indicates quick early learning that stabilizes as training progress. Weighted visualizations show improved recognition including background details. This configuration resulted in a slight decrease in accuracy to 37.40%, which could possibly be due to the strong regularization preventing the model from fitting more complex data patterns.

3 Summary

This assignment highlights the significant roles of learning rate and regularization in training neural networks. A lower learning rate enhances stability while higher regularization improves model generalizability but might restrict the model's ability to learn more complex patterns, as observed in the last configuration. Selecting the right balance between these parameters is important for optimizing the performance of neural network models across different configurations.