

assignment1

April 4, 2025

1 DD2424 Assignment 1

The goal of this project was to implement a simple image classifier for the CIFAR-10 dataset using softmax regression. The focus was on understanding key machine learning concepts such as data preprocessing, forward and backward propagation, gradient checking, and training using mini-batch gradient descent.

I was able to successfully implement the analytical computation of gradients for the cross-entropy loss with L2 regularization in the BackwardPass function. To verify the correctness of this implementation, I compared my gradients against numerically computed gradients from PyTorch on a small subset of the data ($d = 10$, $n = 3$).

The comparison was made using a relative error metric:

$$\text{Relative Error} = \frac{|g_a - g_n|}{\max(\epsilon, |g_a| + |g_n|)}$$

where g_a is the analytical gradient, g_n is the numerical gradient, and ϵ is a small constant to avoid division by zero.

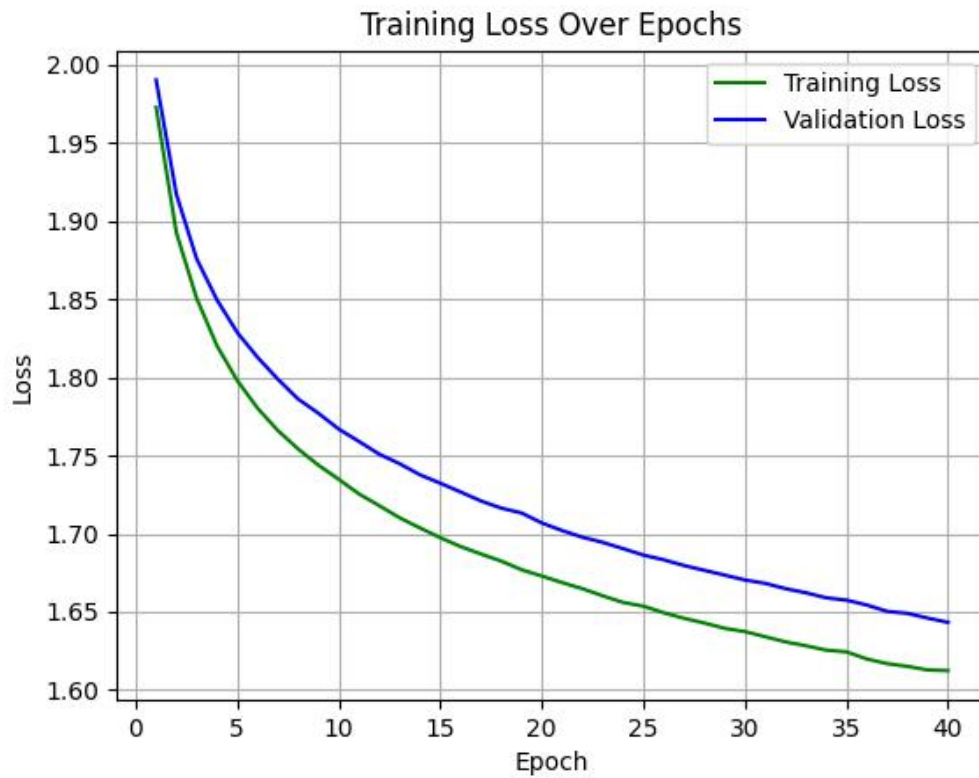
I tested the gradients of both the weights and biases, and the max relative error in W was $\sim 1.09\text{e-}15$ and in b was $\sim 9.78\text{e-}17$. I was able to conclude that the analytical gradients are correct, as the relative errors are very small and within the acceptable range for numerical precision.

1.1 Q1. Loss and Cost Function plots

The loss function is defined as the average negative log-likelihood of the true labels given the predicted probabilities. The cost function includes L2 regularization to prevent overfitting. The loss and cost functions are computed during training and plotted to visualize the model's performance.

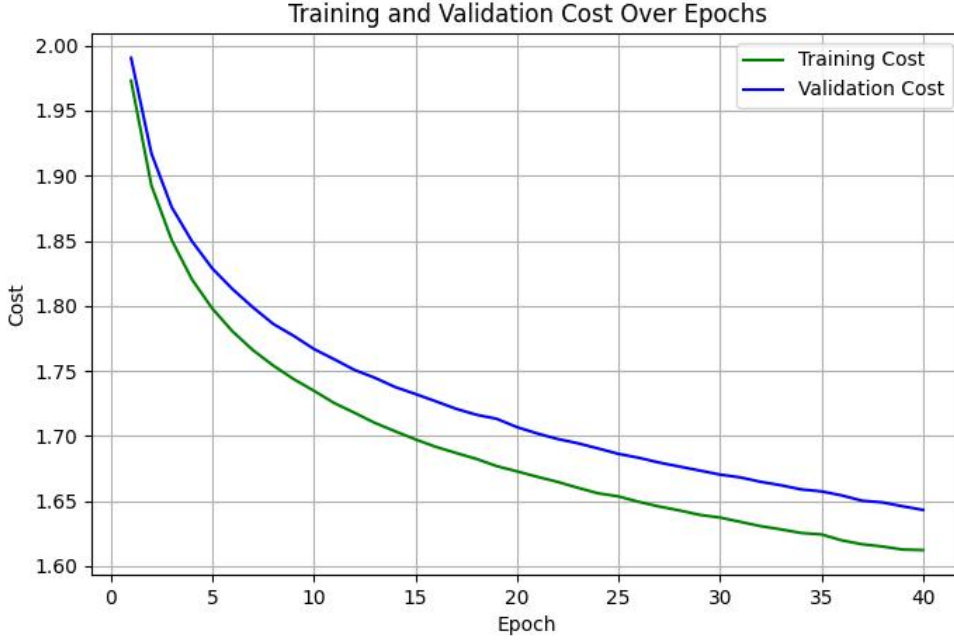
1.1.1 Plot 1: Loss function

The plot shows the training and validation loss over 40 epochs of mini-batch gradient descent. Both losses decrease steadily, indicating that the model is learning effectively. The training loss remains slightly lower than the validation loss, suggesting mild overfitting but overall good generalization.



1.1.2 Plot 2: Cost function

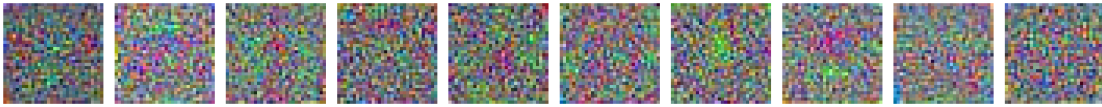
The plot shows the training and validation cost over 40 epochs of mini-batch gradient descent, including both loss and L2 regularization. Both curves decrease consistently, indicating successful optimization. The validation cost remains slightly higher, reflecting regularization's role in promoting generalization.



1.2 Q2. Images representing the learnt weight matrix after the completion of training

W is the weight matrix of the softmax classifier, where each column corresponds to a class and each row corresponds to a pixel in the image. The images are reshaped from the weight matrix and displayed to visualize the features learned by the model.

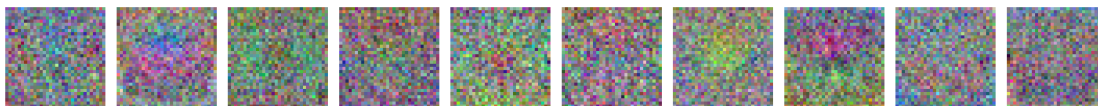
1.2.1 Fig 1: The learnt W matrix visualized as class template images. The network was trained with the following parameter settings: $n_batch=100$, $eta=.1$, $n_epochs=40$ and $lam=0$.



The image above shows the visualization of the learned weight matrix W after training the network for 40 epochs using mini-batch gradient descent. Each square represents the weight filter for one of the 10 CIFAR-10 classes, reshaped into a $32 \times 32 \times 3$ RGB image. These templates highlight the patterns the model associates with each class based on the training data, using the following settings: $n_batch = 100$, $eta = 0.001$, $n_epochs = 40$, and $lam = 0$.

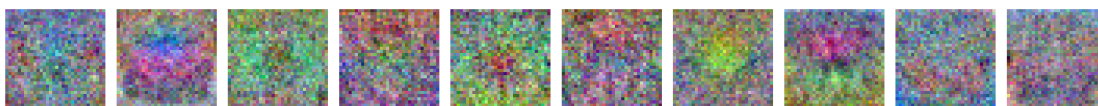
After training the network, the training accuracy after first epoch was 26.72% and the test accuracy was 29.77%. The training accuracy after 40 epochs was 39.48% and the test accuracy was 43.31%. The model's performance improved over time, indicating effective learning from the CIFAR-10 dataset.

1.2.2 Fig2: The learnt W matrix visualized as class template images. The network was trained with the following parameter settings: $n_batch=100$, $eta=.001$, $n_epochs=40$ and $lam=0$.



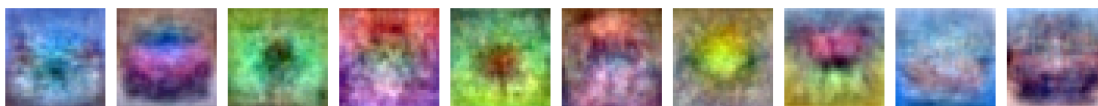
After training the network, the training accuracy after first epoch was 30% and the test accuracy was 30.72%. The training accuracy after 40 epochs was 45.17% and the test accuracy was 45.59%. The model's performance improved over time, indicating effective learning from the CIFAR-10 dataset.

1.2.3 Fig3: The learnt W matrix visualized as class template images. The network was trained with the following parameter settings: $n_batch=100$, $eta=.001$, $n_epochs=40$ and $lam=0.1$



After training the network, the training accuracy after first epoch was 30.07% and the test accuracy was 30.76%. The training accuracy after 40 epochs was 44.87% and the test accuracy was 44.72%. The model's performance improved over time, indicating effective learning from the CIFAR-10 dataset.

1.2.4 Fig4: The learnt W matrix visualized as class template images. The network was trained with the following parameter settings: $n_batch=100$, $eta=.001$, $n_epochs=40$ and $lam=1$



After training the network, the training accuracy after first epoch was 30.3% and the test accuracy was 31.01%. The training accuracy after 40 epochs was 40% and the test accuracy was 38.99%. The model's performance improved over time, indicating effective learning from the CIFAR-10 dataset.