DLCV Assignment -1

Godwin Khalko Roll Number: XYZ123

February 22, 2025

1 Introduction

This report presents the observations and analysis from the experiments conducted in DLCV Assignment 1. The focus areas include training a simple linear model, implementing backpropagation, and designing a convolution module using NumPy.

2 Observations

2.1 Training a Simple Linear Model

The training process involves optimizing a weight matrix W to minimize cross-entropy loss $L_i = -f_{y_i} + \log \sum_j e^{f_j}$ where f_{y_i} is the y_i-th element of the output of $W^T X_i$, in addition to that we also introduce a regularization term $\lambda \sum_i \sum_j W_{i,j}$.
The validation accuracy was 90.6% whereas the test accuracy turned out to 90.4%. The was close to the result obtained by training the scikit-learn's logistic regression model, which gave an accuracy of 92 %

The convergence of the model depends on the selection of the learning rate. Through experimentation, it was found that the
Best learning rate was : 0.009
Number of epochs: 20
Additionally, the best performing class was 0 and worst performing class was 1. Observed that lower learning rates resulted in slow convergence, whereas higher values led to oscillations.

2.2 Backpropagation Implementation

Successfully created the the Module function for various arithmetic and mathematical operation with the forward and backward passes to compute the value and the gradient of the function with respective to the variables x, y, z. The values and the gradient for the 5 values are present in the python notebook.

2.3 Modular vector backpropagation

Created the Module function of the Linear Layer and the ReL Layer with the forward and backward function to compute the layer outputs and the gradient **wrt** the layers. The Model contained 2 Linear Layers and a ReLU layer in the middle and the softmax on the logits at the end.

The validation accuracy turned out to be 96.00 % and the test accuracy turned out to be 96.04 %. Notice the improvement over the single layer logistic regression.
The best and the worst performing classes are the same as in the Question 1. We have also used Xavier Initialization for the intialization of the Linear Layer weights.

2.4 Convolution Module with Numpy

3 Conclusion

The assignment provided hands-on insights into fundamental deep learning concepts. Key takeaways include the importance of hyperparameter tuning, the effectiveness of backpropagation, and the impact of convolution parameters on feature extraction.