Lab Worksheet Report

CNN Training and Analysis on CIFAR-10

Name: **Sahilpreet Singh**
Roll Number: **B23CS1061**
Course: **ML-DL-OPS**

31 January 2026

Abstract

This experiment involves training a Convolutional Neural Network (CNN) on the CIFAR-10 dataset. A custom PyTorch dataset and dataloader were implemented. The model was trained for 25 epochs, and computational complexity was analyzed using FLOPs. Gradient flow and weight update flow were visualized to study training stability. All metrics and visualizations were logged using Weights and Biases (WandB).

1 Objective

The objectives of this lab assignment are:

- To design and train a CNN model on the CIFAR-10 dataset
- To implement a custom PyTorch dataloader
- To compute FLOPs for the selected model
- To analyze gradient flow and weight update flow during training
- To log all metrics and visualizations using WandB

2 Dataset Description

The CIFAR-10 dataset consists of 60,000 color images of size 32×32 belonging to 10 classes. The dataset is divided into:

- 50,000 training images

- 10,000 test images

Data augmentation techniques such as random horizontal flipping and random cropping were applied during training. Normalization was applied to both training and test data.

3 Custom Dataset and Dataloader

A custom dataset class was implemented by extending the `torch.utils.data.Dataset` class. This allowed better control over data transformations and ensured compliance with the assignment requirement of a custom dataloader. PyTorch's `DataLoader` was used to batch and shuffle the data.

4 CNN Model Architecture

The CNN model consists of:

- Two convolutional layers with ReLU activation
- Max-pooling layers for spatial downsampling
- Two fully connected layers for classification

This architecture provides a balance between performance and computational efficiency.

5 FLOPs Analysis

The computational complexity of the model was measured using the THOP profiler. A dummy input tensor of shape $(1, 3, 32, 32)$ was used to compute FLOPs.

```
FLOPs (MACs): 6654464.0
Total Parameters: 1070794.0
```

Figure 1: FLOPs and total parameter count computed using THOP

Observation: The FLOPs analysis shows that the model is computationally efficient and suitable for CIFAR-10 classification.

6 Training Configuration

The model was trained using the following configuration:

- Optimizer: Adam
- Loss Function: Cross Entropy Loss
- Batch Size: 128
- Number of Epochs: 25
- Device: GPU (when available)

7 Gradient Flow Analysis

Gradient flow visualization was used to monitor the average gradient magnitude across all trainable layers. Plots were generated at the end of each epoch.

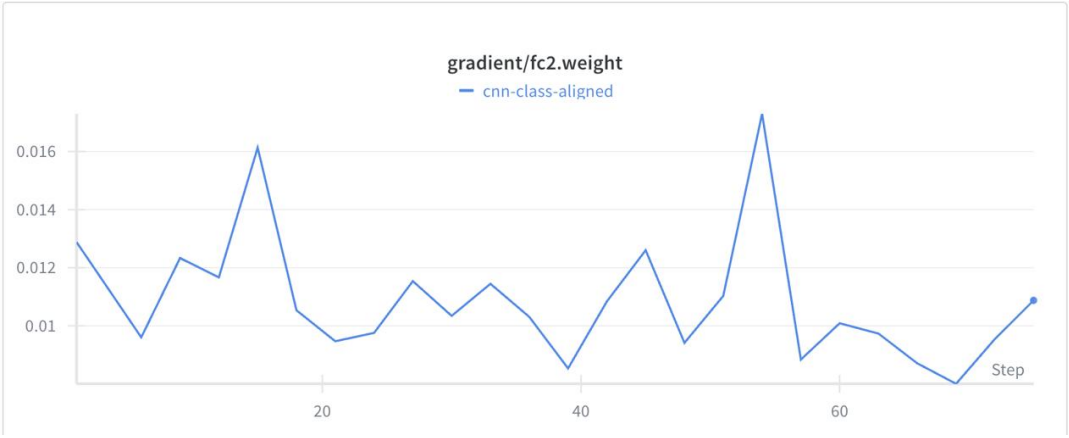

Figure 2: Gradient flow for fully connected layer weights across epochs

Observation: Gradients remained stable across layers, indicating healthy backpropagation without vanishing or exploding gradients.

8 Weight Update Flow Analysis

Weight update flow was analyzed by measuring the change in parameter values between consecutive epochs.

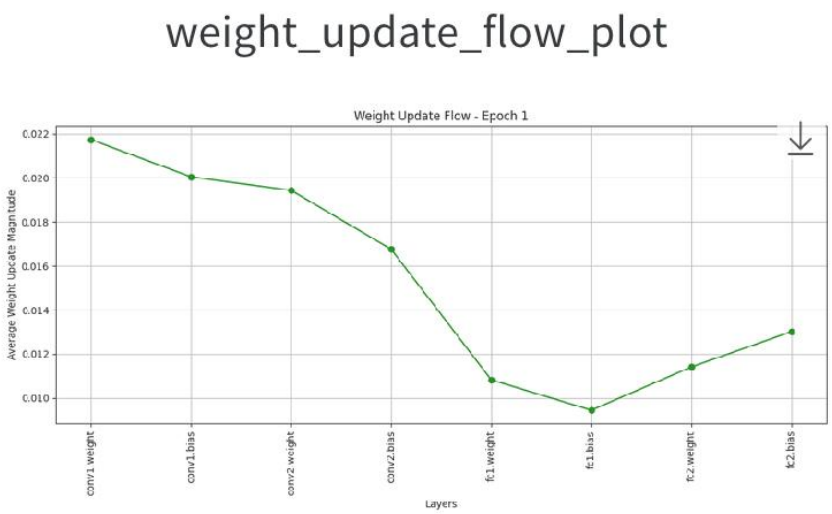

Figure 3: Weight update flow across CNN layers for a training epoch

Observation: Weight updates showed consistent magnitudes across layers, confirming stable learning behavior throughout training.

9 Experiment Tracking with WandB

All metrics and visualizations were logged using Weights and Biases (WandB), including:

- Training loss vs epochs
- Training accuracy vs epochs
- Test accuracy
- Gradient flow plots
- Weight update flow plots

The complete experiment dashboard is publicly available at:

`https://wandb.ai/ikamboj-919-iit-jodhpur/updated_cnn_cifar10`

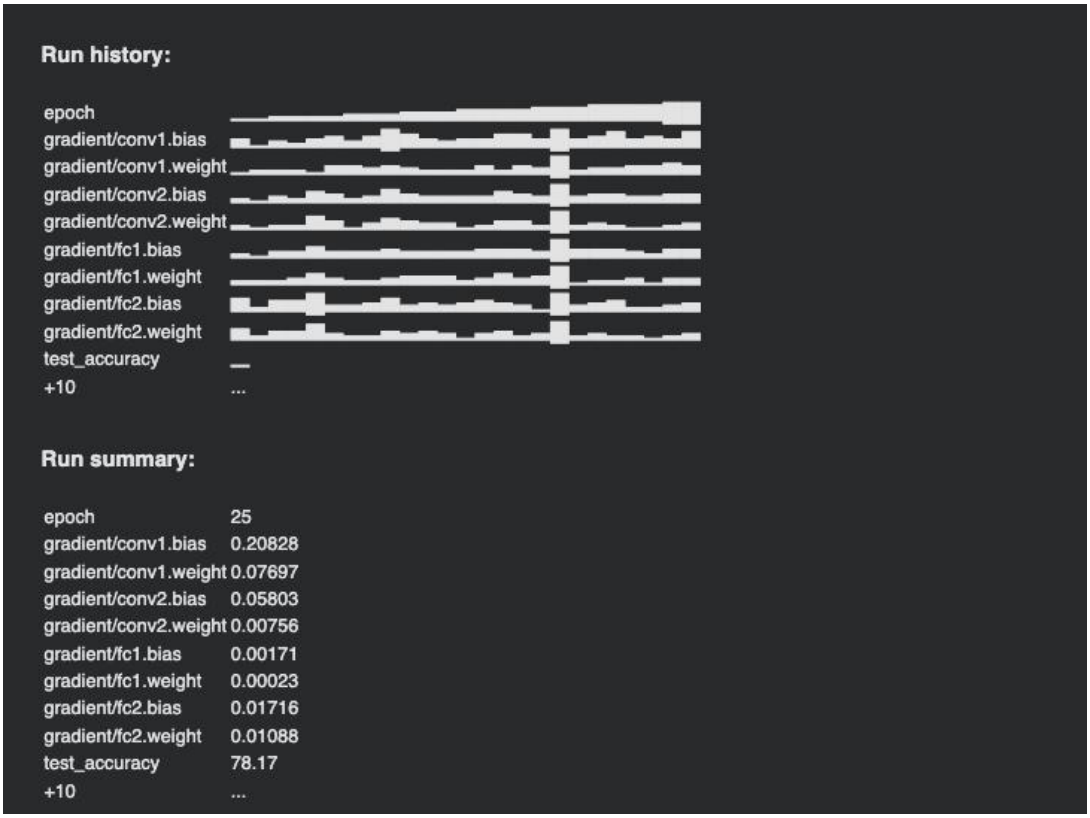

Figure 4: WandB run history and final run summary

10 Results

The CNN model achieved steady improvement in training accuracy across epochs. The final test accuracy demonstrates effective learning on the CIFAR-10 dataset.

11 Conclusion

In this lab, a CNN was successfully trained on the CIFAR-10 dataset using a custom dataloader. FLOPs analysis confirmed computational efficiency. Gradient flow and weight update flow visualizations demonstrated stable training. All results and visualizations were logged using WandB, fulfilling all assignment requirements.