Lab Worksheet -1

Course : ML_DL_OPs

Semester : 2nd

Year : 2026

Final report

IIT Jodhpur

Name: Mahek Shankesh Gadiya

Roll : M25CSA011

Dep: CSE (MTech-AI)

GitHub link :

https://github.com/MSG-1999/MLOps-MahekGadiya-M25CSA011/tree/Worksheet_1

Google Colab Link:

https://colab.research.google.com/drive/1YTJj5tUCJlT7GH2DI1sxADq8KwA11GiE#scrollTo=mhatgyzvJ9MY

Wandb Colab Link:

https://wandb.ai/msg1999-indian-institutes-of-technology-jodhpur/Mlops_Lab_2?nw=nwusermsg1999

https://api.wandb.ai/links/msg1999-indian-institutes-of-technology-jodhpur/7vzfc7l8

CNN Training on CIFAR-10 with Gradient and Weight Flow Analysis.

1. Objective

The objective of this experiment is to train a Convolutional Neural Network (CNN) on the CIFAR-10 dataset and analyze its training dynamics. The goals of this experiment are:

1. To implement a CNN using a custom dataset loader
2. To train the model for a fixed number of epochs (30)
3. To compute the FLOPs and number of trainable parameters
4. To analyze gradient flow and weight update flow
5. To visualize training and validation performance
6. To log all metrics and visualizations using Weights & Biases (W&B)

2. Dataset Description

2.1 Dataset Overview

- **Dataset:** CIFAR-10
- **Number of Classes:** 10
- **Image Size:** 32 x 32 x 3
- **Total Images:** 60,000

2.2 Dataset Split

- **Training Set:** 80% of CIFAR-10 training data (40,000 images)
- **Validation Set:** 20% of CIFAR-10 training data (10,000 images)
- **Test Set:** Standard CIFAR-10 test dataset (10,000 images)

3. Data Preprocessing

3.1 Training Data Augmentation

To improve generalization and reduce overfitting, the following augmentations were applied to the training data:

- Random Horizontal Flip
- Random Crop with padding = 4
- Normalization

3.2 Validation and Test Preprocessing

- Normalization only

This preprocessing strategy ensures robust learning while maintaining fair and consistent evaluation on validation and test datasets**.**

4. Model Selection and Architecture

4.1 Model Choice

A ResNet-18 based CNN was selected due to its residual connections, which help mitigate vanishing gradient problems and enable stable training of deep networks.

4.2 Architecture Customizations for CIFAR-10

To adapt ResNet-18 for the low-resolution CIFAR-10 dataset, the following modifications were applied:

- The first convolutional layer (conv1) was changed to a 3 × 3 kernel, stride = 1, and padding = 1
- The initial max-pooling layer was removed using nn.Identity() to preserve spatial resolution
- The final fully connected layer was modified to output 10 classes

These modifications are standard practices when applying ResNet architectures to CIFAR-10**.**

4.3 Model Complexity

- **Number of Trainable Parameters:** 11.17 million
- **FLOPs:** 557.89 MFLOPs (computed using the THOP library for a single forward pass)

5. Training Configuration

The model was trained using the following configuration:

- Loss Function: Cross Entropy Loss
- Optimizer: Adam
- Learning Rate: 0.001
- Batch Size: 128
- Number of Epochs: 30
- Device: GPU (CUDA, when available)

GPU acceleration was used to speed up training and ensure efficient convergence

7. Training Performance and Results

7.1 Results

Metric	**Value (at Epoch 30)**
Final Training Accuracy	**96.73%**
Final Validation Accuracy	**90.84%**
Final Test Accuracy	**90.35%**
Final Training Loss	**0.0933**
Final Validation Loss	**0.3704**

7.2 Observations

- The model shows steady and stable convergence across epochs.
- The gap between training accuracy (96.73%) and validation accuracy (90.84%) indicates minor overfitting, which is expected for a deep CNN.
- The strong test accuracy of 90.35% confirms that the model generalizes well to unseen data.
- Validation and test performance are closely aligned, suggesting effective regularization and appropriate training configuration.

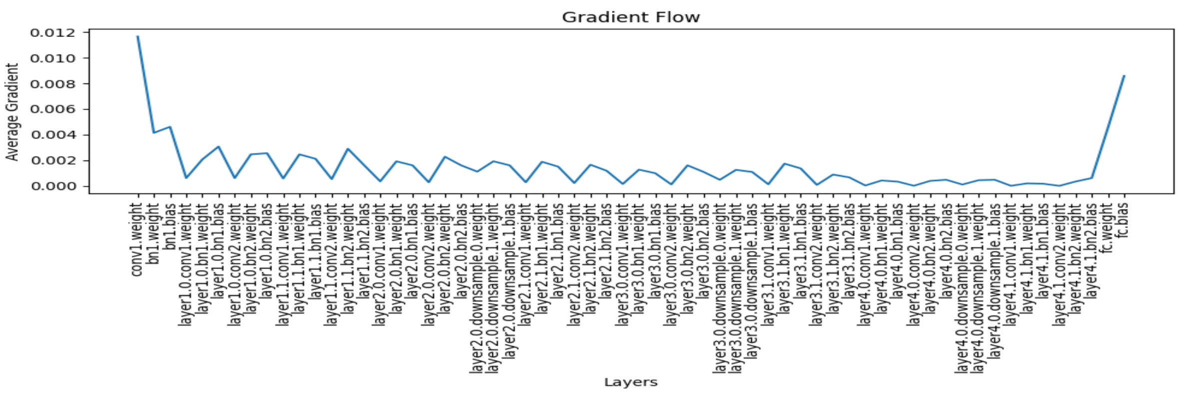
7.3 Computational Complexity

- **FLOPs:** Computed using the THOP library for a single forward pass
- **Number of Parameters:** Total trainable parameters of the model

These metrics provide insight into the computational cost and efficiency of the chosen architecture

8. Gradient Flow Analysis

Gradient flow analysis was performed by plotting the mean absolute gradient of each trainable layer after backpropagation.



Fig(a). Gradient Flow for Epoch = 20

8.1 Observations

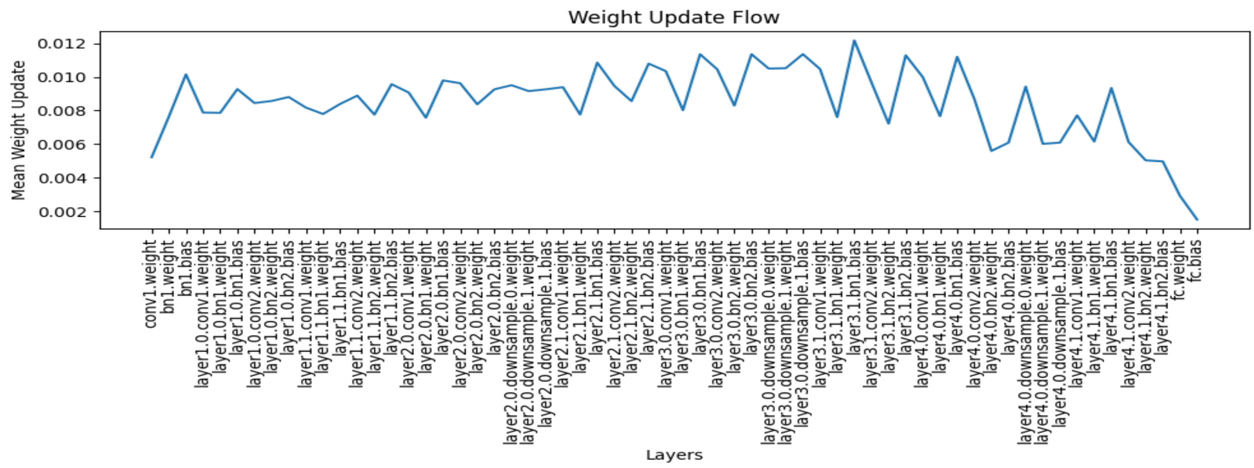
- Gradients remain non-zero across all layers
- No evidence of vanishing or exploding gradients
- Stable gradient propagation from early to deeper layers

8.2 Conclusion

The gradient flow visualizations confirm stable learning behavior and effective backpropagation throughout the network, validating the benefit of residual connections in ResNet-18.

9. Weight Update Flow Analysis

Weight update flow was analyzed by measuring the mean absolute change in weights between consecutive epochs.



Fig(b). Weight Update Flow for Epoch = 20

9.1 Observations

- Larger weight updates occur during early training epochs
- Weight updates gradually decrease as training progresses
- Indicates smooth convergence of the optimization process

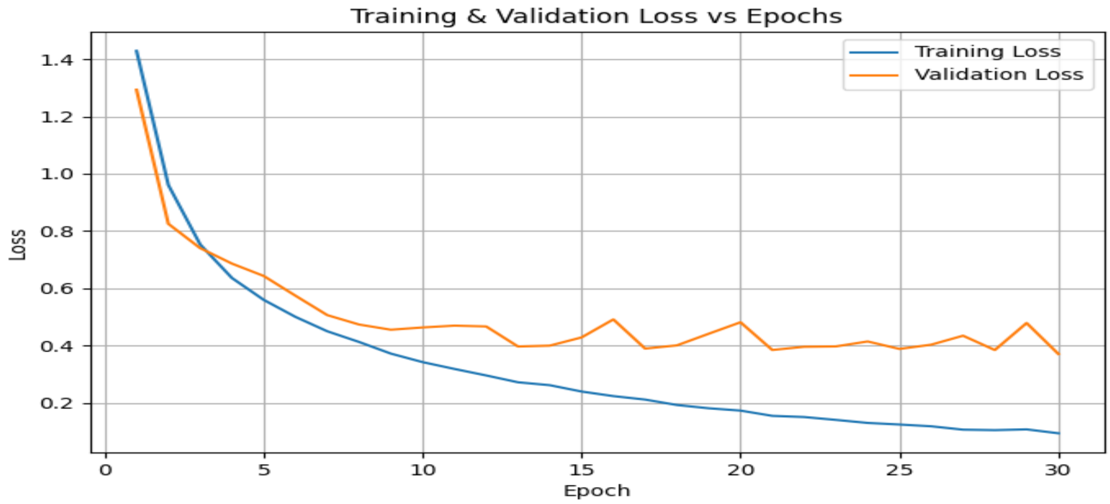
9.2 Conclusion

The optimizer updates the parameters effectively, and the gradual reduction in update magnitude confirms stable convergence.

10. Training and Validation Performance Analysis

10.1 Loss vs Epoch

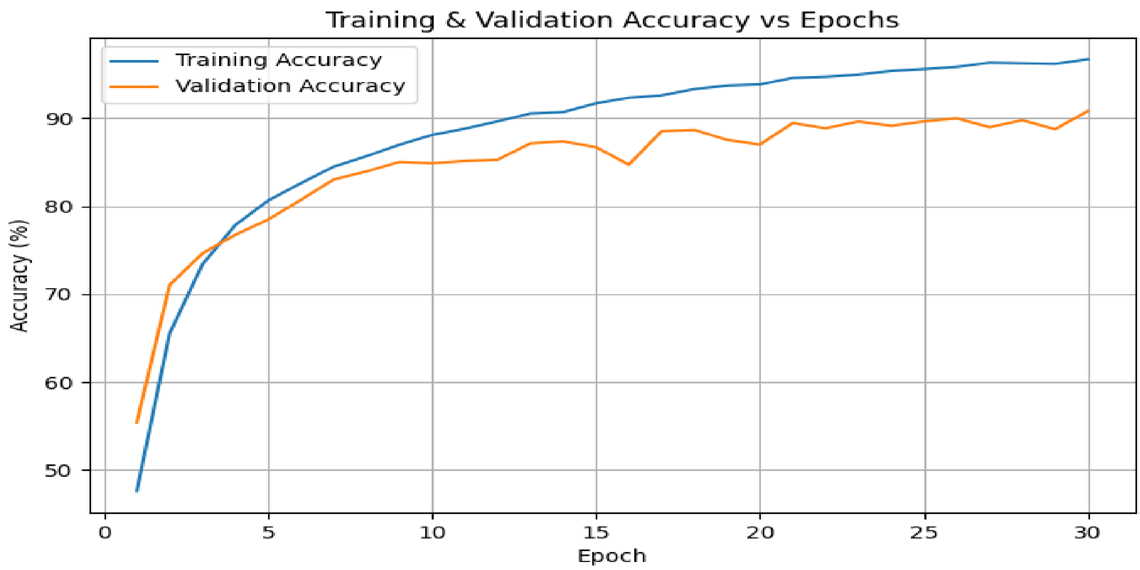
- Training loss decreases steadily across epochs
- Validation loss follows a similar trend and stabilizes toward the end of training



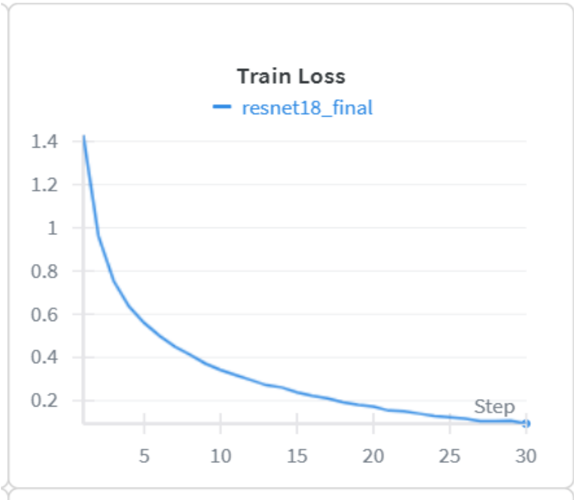
Fig(c). Loss vs Epoch

10.2 Accuracy vs Epoch

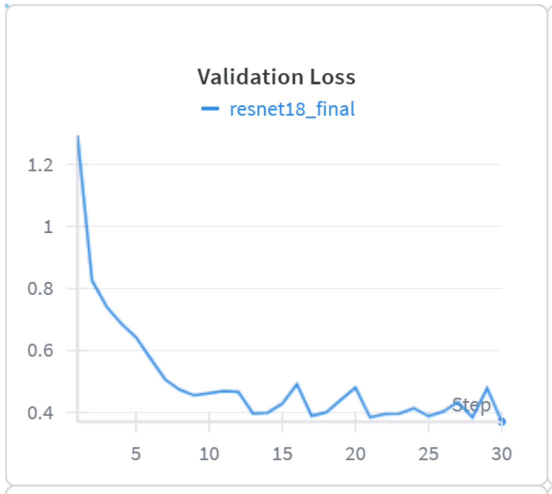
- Training accuracy increases rapidly
- Validation accuracy improves more gradually
- A small and controlled gap exists between training and validation accuracy



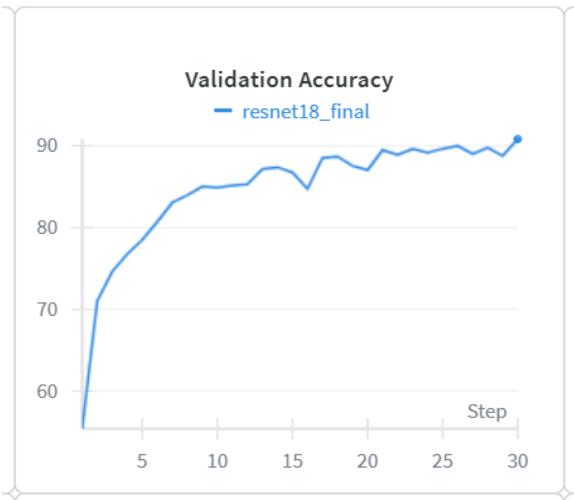
Fig(d). Accuracy vs Epoch



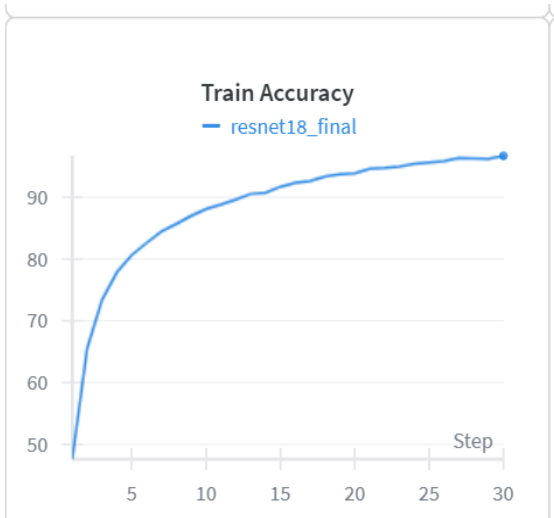
Fig(e). Train Loss Plot



Fig(f). Validation Loss Plot



Fig(g). Validation Accuracy Plot



Fig(g). Train Accuracy Plot

11. Logged Metrics and Visualizations

- Training and validation loss
- Training and validation accuracy
- Test accuracy
- Gradient flow plots
- Weight update flow plots
- Loss and accuracy curves versus epochs

This logging framework ensures reproducibility, transparency, and systematic experiment analysis.

12. Conclusion

In this experiment, a CNN based on **ResNet-18** was successfully trained on the **CIFAR-10** dataset. The model demonstrated stable training dynamics, effective gradient propagation, and strong generalization performance. Gradient flow and weight update flow analyses provided valuable insights into the learning behavior and confirmed the correctness of the training process. The final test accuracy of **90.35%** validates the effectiveness of the selected architecture, preprocessing strategy, and training configuration.