MLOps Lab Assingment - 1 — B22AI063

Name: Jyotin Goel
Roll Number: B22AI063
Email: b22ai063@iitj.ac.in

Links

- **Colab Notebook:**
  Open in Google Colab
- **WANDB visualizations:**
  View Assignment on wandb
- **GitHub Repository:**
  View Assignment on GitHub

1. Objective

The objective of this lab was to:

- Train a Convolutional Neural Network (CNN) on the **CIFAR-10 dataset**
- Implement a **custom PyTorch dataloader**
- Compute **FLOPs** for the selected model
- Train the model for **25–30 epochs**
- Visualize:
  - **Gradient flow**
  - **Weight update flow**
- Log all metrics and visualizations using **Weights & Biases (Wandb)**
- Upload all code + report to GitHub (without pushing trained models)

---

2. Dataset

We used the **CIFAR-10 dataset**, which contains:

- 60,000 RGB images of size **32×32**
- 10 classes (airplane, car, bird, cat, etc.)
- Train/Test split:
  - 50,000 training images
  - 10,000 test images

3. Model Architecture

A CNN model (**ResNet-18**) was chosen due to its strong performance and efficiency on CIFAR-10.

Key components:

- Convolutional layers with residual blocks
- Batch normalization
- ReLU activations
- Fully connected classifier head

---

4. Custom DataLoader Implementation

Instead of using the default torchvision dataloader directly, a **custom Dataset class** was implemented.

Features:

- Custom `__getitem__` and `__len__`
- Data augmentation:
    - Random crop
    - Horizontal flip
- Normalization for CIFAR-10 statistics

---

5. FLOPs Computation

FLOPs (Floating Point Operations) were computed using the `ptflops` library.

Example output:

- **Total FLOPs:** ~1.8 GFLOPs
- **Total Parameters:** ~11.2M

This provides an estimate of computational cost during inference/training.

---

6. Training Setup

Training was performed for **30 epochs** with the following configuration:

Hyperparameter	Value
Optimizer	Adam
Learning Rate	0.001
Batch Size	128
Epochs	30

Hyperparameter	Value
Loss	CrossEntropyLoss

---

7. Gradient Flow Visualization

Gradient flow plots were generated during training to inspect whether gradients vanish or explode.

Observations:

- Early convolution layers had smaller gradients
- Deeper layers showed stronger gradient updates
- No major gradient instability was observed

Gradient flow plots were logged to Wandb each epoch.

---

8. Weight Update Flow Visualization

Weight update flow tracks how much parameters change over epochs.

Observations:

- Initial epochs showed large weight updates
- Updates stabilized after ~15 epochs
- Indicates convergence of training

These plots were also logged to Wandb.

---

9. Results and Findings

Training Performance

- Training accuracy steadily increased over epochs
- Test accuracy improved and stabilized around final epochs

10. Wandb Visualizations Logged

The following were logged to Wandb:

- Training loss curve
- Validation accuracy curve
- Gradient flow plots
- Weight update flow plots
- FLOPs + parameter summary