AlexNet on CIFAR-10 - Project Documentation

Name: Tensae Aschalew
GSR/3976/17

1. Introduction

This project implements the AlexNet convolutional neural network using PyTorch and applies it to the CIFAR-10 image classification task. The model is trained, evaluated, and tested in the Google Colab environment. This implementation is intended for educational purposes, with clearly annotated code and detailed explanations.

---

2. Objective

To:

- Understand the architecture of AlexNet.
- Apply it to a smaller dataset (CIFAR-10).
- Explore training, evaluation, and performance optimization.
- Demonstrate how deep CNNs can be adapted to datasets other than ImageNet.

---

3. Background: AlexNet Architecture

3.1 Historical Context

AlexNet was proposed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012. It won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with a **top-5 error rate of 15.3%**, far surpassing the runner-up.

It became a cornerstone in the deep learning revolution for computer vision due to:

- Its depth (8 layers: 5 convolutional, 3 fully connected).

- Use of GPU acceleration.
- ReLU activation, which enabled faster convergence.
- Dropout for reducing overfitting.
- Overlapping max-pooling for better spatial understanding.

3.2 Predecessor: LeNet-5

Feature	LeNet-5	AlexNet
Year	1998	2012
Dataset	MNIST (grayscale)	ImageNet (RGB)
Input Size	32×32	224×224
Conv Layers	2	5
FC Layers	2	3
Activation	Tanh	ReLU
Regularization	None	Dropout
GPU Usage	No	Yes (2 GPUs originally)
Parameters	~60K	~60 million

AlexNet extended LeNet-5 to support higher dimensional RGB inputs and deeper architectures suitable for real-world image recognition problems.

---

4. Dataset: CIFAR-10

4.1 Description

- Total samples: 60,000 images
- Image size: 32×32 pixels, RGB
- 10 classes:

  - airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck

- Training set: 50,000 images
- Test set: 10,000 images

4.2 Preprocessing for AlexNet

Since AlexNet expects 224×224 pixel images, we resize all CIFAR-10 images to 224×224 using interpolation. Additionally, the dataset is normalized to mean = (0.5, 0.5, 0.5) and std = (0.5, 0.5, 0.5) for each channel to standardize the input.

---

5. Model Architecture

5.1 AlexNet - Layer Breakdown

Feature Extractor:

- Conv1: 96 filters, 11x11 kernel, stride 4 → ReLU → MaxPool
- Conv2: 256 filters, 5x5 kernel → ReLU → MaxPool
- Conv3: 384 filters, 3x3 kernel → ReLU
- Conv4: 384 filters, 3x3 kernel → ReLU
- Conv5: 256 filters, 3x3 kernel → ReLU → MaxPool

Classifier:

- Dropout
- Fully Connected 4096 → ReLU → Dropout
- Fully Connected 4096 → ReLU
- Fully Connected 10 (output logits)

---

6. Environment Setup (Google Colab)

6.1 Enable GPU

- Go to **Runtime > Change runtime type**
- Set **Hardware Accelerator** to **GPU**

6.2 Dependencies

PyTorch and torchvision are pre-installed in Google Colab.

---

7. Training Pipeline

- **Loss Function**: CrossEntropyLoss (for multi-class classification)
- **Optimizer**: Adam (learning rate = 0.001)
- **Batch Size**: 64
- **Epochs**: Default is 1 (can be increased)
- **Evaluation Metric**: Accuracy on test set

Training Steps

1. Load and preprocess data
2. Define AlexNet model
3. Train model with mini-batches
4. Report loss every 100 steps
5. Evaluate on test data
6. Print test accuracy

---

8. Results (5 Epoch)

Metric	Value
Epochs	5
Final Test Accuracy	~60.80%
Training Time	~487.89 seconds (with GPU)

Note: With 10–30 epochs, accuracy can reach 75%+

---

10. Possible Improvements

- Train for more epochs
- Use learning rate schedulers
- Add data augmentation: random cropping, flipping
- Apply transfer learning using pretrained weights
- Use mixed precision training for faster performance

---

11. References

- Krizhevsky et al., “ImageNet Classification with Deep Convolutional Neural Networks”, NeurIPS 2012.
- CIFAR-10 dataset: https://www.cs.toronto.edu/~kriz/cifar.html