Vision Transformer Implemented from Scratch

Vaibhav (241121)

February 7, 2026

1 Introduction

This report presents the implementation of a Vision Transformer (ViT) model developed entirely from scratch using PyTorch and trained on the CIFAR-10 dataset. Unlike traditional convolutional neural networks (CNNs), which rely on convolution and pooling operations to extract features, Vision Transformers process images as sequences of fixed-size patches using the Transformer architecture. This allows the model to capture long-range and global dependencies across the image.

2 Data Preprocessing and Regularization

2.1 Dataset

The CIFAR-10 dataset consists of:

- 60,000 RGB images
- Image resolution of $32 \times 32 \times 3$
- 10 object classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck

2.2 Preprocessing Steps

The following preprocessing techniques were applied during training:

1. **Random Horizontal Flip**
   Images are randomly flipped along the horizontal axis. *Effect:* Improves generalization by learning orientation-invariant features.

2. **Random Crop with Padding**
   Images are padded and then randomly cropped back to 32×32. *Effect:* Makes the model robust to small spatial translations.

3. **ToTensor**
   Converts images to PyTorch tensors and scales pixel values to the range $[0, 1]$. *Effect:* Enables compatibility with neural network operations.

4. **Normalization**
   Pixel values are normalized using dataset-specific mean and standard deviation. *Effect:* Stabilizes training and accelerates convergence.

2.3 Regularization Techniques

To prevent overfitting and improve training stability, the following regularization methods were used:

- **Dropout:** Randomly disables neurons during training to improve generalization.
- **Weight Decay (L2 Regularization):** Penalizes large weights and stabilizes learning.
- **Gradient Clipping:** Prevents exploding gradients in deep Transformer architectures.

3 Hyperparameter Tuning

3.1 Key Hyperparameters

Parameter	Tested Values	Final Value
Patch Size	2, 4, 8	4
Embedding Dimension	256, 384, 768	364
Attention Heads	4, 8	8
Transformer Layers	4, 6, 8	6
Dropout Rate	0.1, 0.2	0.1
Batch Size	64, 128	128
Learning Rate	10^{-3}, 3×10^{-4}, 10^{-4}	3×10^{-4}

3.2 Observations

- Smaller patch sizes improved accuracy but increased computational cost.
- Increasing the number of layers improved performance up to a saturation point.
- Larger learning rates led to unstable loss behavior.

4 Vision Transformer Architecture

The Vision Transformer converts an image into a sequence of tokens that can be processed similarly to words in natural language processing.

4.1 Patch Creation

The input image is divided into non-overlapping patches, with each patch treated as an individual token.

4.2 Patch Embedding

Each flattened patch is projected into a fixed-dimensional embedding space, transforming raw pixel values into feature representations.

4.3 Positional Embedding

Since Transformers lack inherent spatial awareness, positional embeddings are added to encode the relative positions of patches.

4.4 Class Token

A learnable class token is prepended to the patch sequence. It aggregates information from all patches and is used for final classification.

4.5 Multi-Head Self-Attention

Each patch attends to every other patch, allowing the model to learn global image relationships instead of purely local patterns.

4.6 Feed Forward Network (MLP Block)

This block applies non-linear transformations to token embeddings, increasing the model's representational capacity.

4.7 Classification Head

The class token output is passed through a linear layer to produce class probabilities.

5 Results

After training for 50 epochs, the model achieved the following results:

- **Training Accuracy:** 95.03%
- **Best Test Accuracy:** 79.30%

The model successfully learned meaningful image representations. The relatively lower test accuracy is attributed to the limited size of the CIFAR-10 dataset. Vision Transformers generally perform better when trained on larger-scale datasets.