Implementation of a Vision Transformer (ViT) for CIFAR-10 Classification

Devesh Kumar
240344

February 2026

Contents

1 Introduction

This report details the implementation of a Vision Transformer (ViT) from scratch, specifically tailored for the CIFAR-10 dataset. Unlike standard Convolutional Neural Networks, this architecture treats an image as a sequence of patches and utilizes the self-attention mechanism to identify global patterns.

2 Architecture and Mechanism

The code is structured into four primary modules that work together to transform a 32x32 image into a final classification.

2.1 Patch Embedding Layer

In the provided code, the `PatchEmbedding` class takes the 32×32 image and divides it into 4×4 patches.

- This results in exactly 64 patches (8×8 grid).
- These patches are projected into a 128-dimensional space using a 2D convolution with a stride and kernel size of 4.

2.2 Learnable Tokens and Positioning

The `VisionTransformer` class introduces two critical learnable components:

- **CLS Token:** A parameter of size $(1, 1, 128)$ is prepended to the sequence of 64 patches. This token captures the aggregate image context.
- **Positional Embedding:** Since Transformers are permutation invariant, a learnable vector is added to the patches to teach the model the spatial layout of the 4×4 squares.

2.3 Multi-Head Self-Attention (MHSA)

The `MultiHeadAttention` class implements the core logic where each patch calculates its relationship with every other patch.

- It creates Queries (Q), Keys (K), and Values (V) through a linear layer.
- The attention score is calculated as $\text{softmax}(QK^T/\sqrt{d_k})V$.
- Our implementation uses 8 attention heads to focus on different visual features simultaneously.

3 Preprocessing and Regularization

The training script includes specific strategies to ensure the model generalizes well to the CIFAR-10 test set.

3.1 Preprocessing Constants

We use the exact mean and standard deviation of the CIFAR-10 dataset for normalization:

- **Mean:** $(0.4914, 0.4822, 0.4465)$
- **Std:** $(0.2023, 0.1994, 0.2010)$

4 Effects of Regularization and Preprocessing

The implementation of these techniques had a direct impact on the model performance:

- **Preprocessing Effect:** Without normalization, the initial loss was unstable. Normalization allowed for a higher learning rate of 5×10^{-4} without the model diverging.
- **Regularization Effect:** The gap between training accuracy and test accuracy was significantly reduced. Dropout and Weight Decay ensured that while training loss decreased, the model still performed well on unseen images in the test set.

5 Hyperparameter Configuration

The following parameters are strictly used in the final training script:

Hyperparameter	**Value**
Learning Rate	5×10^{-4}
L.R. Scheduler	Cosine Annealing
Transformer Depth	6 Blocks
Embedding Dim	128
Batch Size	64

6 Conclusion

This assignment demonstrates that a "from scratch" Vision Transformer can effectively learn to classify images by treating them as sequences. By using specific normalization and dropout, the model achieves stable convergence on the CIFAR-10 dataset within 10 epochs.