Project Progress Report: Vision Transformer

Student: Paveet Singh Saluja
Date: January 10, 2026

1. Introduction

This project focuses on building Vision Transformers (ViTs) to classify images. The approach begins with the basics of machine learning and progresses to modern techniques. We start by processing images using standard methods (CNNs) and will eventually move to attention-based models that split images into patches. The primary dataset used for training and comparison is CIFAR-10.

2. Week 0: Getting Started with Python

Before starting the complex math, we reviewed the essential programming tools needed for data science.

- Concepts: We brushed up on Python loops, functions, and data structures.
- Libraries: We practiced using NumPy for math, Pandas for handling data tables, and Matplotlib for creating charts.
- Assignment 1: The goal was to prove we could handle data by writing code to manipulate arrays and visualize information.

3. Week 1: Machine Learning Fundamentals

We learned how computers learn from data by adjusting their internal settings to reduce errors.

- Key Concepts:

Activations: These are mathematical switches (like ReLU or Sigmoid) that help the model understand complex, non-straight patterns.

Loss Functions: These measure how wrong a model's guess is. For example, Mean Squared Error is used for predicting numbers, while Cross Entropy is used for categorization.

Gradient Descent: This is the strategy the model uses to improve. It looks at the errors and adjusts its parameters step-by-step to get better results.

- Evaluation: We learned how to grade our models using metrics like Accuracy (how often it is right) and F1-Score (a balance of precision and recall).
- Assignment 2: We built a digit-recognition system from scratch (without using shortcuts like TensorFlow for the math), achieving over 93% accuracy on the MNIST dataset.

4. Week 2: Convolutional Neural Networks (CNNs)

We moved from general data to image processing. Standard networks are too slow for large images, so we use CNNs.

- How CNNs Work: instead of looking at every pixel individually, small filters slide over the image to detect features like edges or textures.

- Important Tools:

Pooling: Shrinks the image size while keeping the important details.

Batch Normalization: A technique to stabilize the learning process, making training faster and more reliable.

- Assignment 3: We focused on theory, explaining concepts like Overfitting (when a model memorizes data instead of learning patterns) and how to prevent it using Dropout (randomly turning off parts of the brain during training).

5. Week 3: Recurrent Neural Networks (RNNs)

We shifted focus to sequential data, where the order matters (like words in a sentence).

- The Challenge: Standard models assume data points are independent, but in text, the next word depends on the previous one.
- The Solution: RNNs have a memory (hidden state) that remembers what it has seen so far.
- Issues: RNNs can struggle if the sequence is too long (they forget the beginning) or if the math gets out of control (gradients exploding). We fix the math issues using Gradient Clipping.
- Assignment 4: We returned to images to improve our CNN. By adding Batch Normalization to our model, we increased accuracy on the CIFAR-10 dataset from ~75% to ~79%.

6. Week 4: Advanced Sequence Models

We studied smarter versions of RNNs that handle memory better.

- LSTM & GRU: These are advanced networks with special gates that decide exactly what to remember and what to forget, solving the short-term memory problem of basic RNNs.
- Seq2Seq: A Sequence-to-Sequence architecture used for tasks like translation (e.g., English to French). It has an Encoder to read the input and a Decoder to write the output.
- Assignment 5: We built a creative text generator. Using a character-level RNN, we trained a model on a list of dinosaur names. Eventually, the model learned to invent its own realistic-sounding dinosaur names.