Mid-Evaluation Report: Vision Transformers Project
Prepared by: Krish Mundhra

1. Introduction & Project Overview

This report covers everything I've learned and worked on so far for the Vision Transformers (ViT) project. The main goal is to get a really solid grip on deep learning basics before jumping into the complex transformer stuff. Since ViTs are a big shift from the usual CNNs to attention-based models, I had to spend the first few weeks building a foundation in Python, math, and different types of neural networks.

Basically, I'm following a week-by-week plan to make sure I don't miss any of the prerequisites like optimization, CNNs for images, and sequence models like RNNs.

2. Week 0 & 1: Coding and Math Foundations

Week 0 was all about getting better at Python. I practiced using loops, functions, and dictionaries to manage data. I also learned how to use NumPy for matrix math, which is super important because neural networks are basically just huge matrix multiplications. I even made a "tiny model" using Python classes to see how data flows through a structure.

Week 1 shifted to the math side. I looked at:

Activation Functions: Like Softmax for classification.

Loss Functions: How to measure how "wrong" a model is.

Optimizers: Using Gradient Descent to make the error smaller.

I finished two assignments on Colab during this time. I implemented linear and logistic regression and used Matplotlib to visualize the results. It was a bit tough getting the gradients right, but the assignments helped me see how the math actually works in code.

3. Week 2 & 3: CNNs and Basic Sequence Modeling

In Week 2, I studied Convolutional Neural Networks (CNNs). These are the traditional way of doing computer vision. I learned about filters, padding, and strides, basically how a model "slides" over an image to pick up features like edges or shapes.

Then in Week 3, I started Sequence Modeling. This might seem weird for vision, but transformers treat images like sequences of patches, so it's relevant.

I learned about RNNs (Recurrent Neural Networks) and how they use "hidden states" to remember past info.

I also dealt with Gradient Clipping because RNNs can be really unstable during training.

4. Week 4: Advanced Models & Current Status

Week 4 was about fixing the problems with basic RNNs. Simple RNNs have a hard time remembering things for a long time because of "vanishing gradients". To fix this, I studied LSTMs and GRUs, which use "gates" to decide what info to keep or forget.

I also looked at Encoder-Decoder structures. This is the most important part so far because it’s the direct ancestor of the Transformer architecture.

Summary of Progress: So far, I've finished 5 assignments. I feel pretty confident with the coding and the general idea of how neural networks learn. I’ve spent a lot of time on the "traditional" ways of doing things (CNNs and RNNs) so that when I start with Vision Transformers, I’ll actually understand why they are better at capturing global context.