Vision Transformer Mid Term Evaluation Report

1. Executive Summary

This report synthesizes the core deep learning concepts covered during the first four weeks of the curriculum. The journey began with the mathematical foundations of regression and evolved into complex neural architectures.

We explored how machines "learn" through optimization, how Convolutional Neural Networks (CNNs) handle visual data, and how Recurrent Neural Networks (RNNs) manage sequential information. These fundamentals provide the necessary scaffolding for understanding the advanced Vision Transformer architectures we will explore later in the course.

2. Week 1: Regression and Optimization

The foundation of deep learning lies in regression—modeling the relationship between independent inputs and dependent outputs. We examined two critical forms:

- **Linear Regression:** Used primarily for forecasting continuous variables. It attempts to fit a linear equation to the observed data by minimizing the error between predicted and actual values.
- **Logistic Regression:** A classification technique where the output is a probability. By applying a Sigmoid function, $\sigma(z) = \frac{1}{1+e^{-z}}$, we map the linear output to a range between 0 and 1, making it suitable for binary decision-making.

The learning process is driven by **Gradient Descent**. This optimization algorithm iteratively adjusts the model's weights in the opposite direction of the loss gradient, effectively "walking down the hill" to find the point of minimal error.

3. Week 2: Neural Network Architecture

Drawing inspiration from biological systems, we explored the Multilayer Perceptron (MLP). An MLP consists of layers of "neurons" that process information hierarchically:

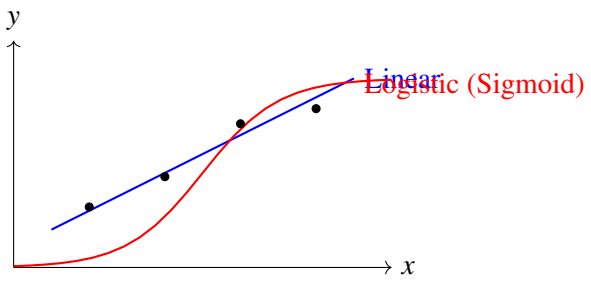

Figure 1: Visual comparison of Linear Regression (fitting a line) vs. Logistic Regression (fitting a probability curve).

1. **Input Layer:** Accepting raw features (e.g., image pixels).
2. **Hidden Layers:** Performing non-linear transformations to extract features. Deep networks have multiple hidden layers, allowing them to learn intricate patterns.
3. **Output Layer:** Generating the final prediction, such as a probability distribution for digit recognition.

Training these networks requires **Backpropagation**. This algorithm calculates the gradient of the loss function with respect to each weight by applying the chain rule backwards from the output layer. It assigns "blame" for the error to specific neurons, allowing for precise weight updates.

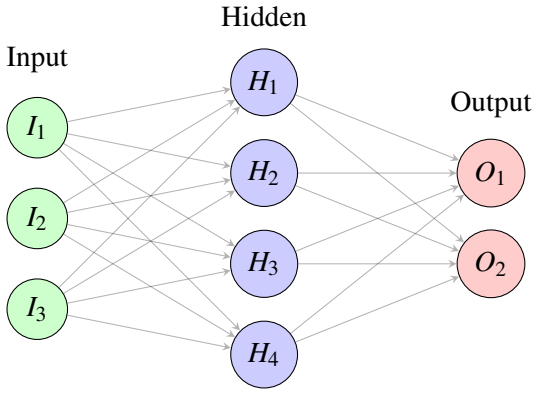

Figure 2: Architecture of a standard Feed-Forward Neural Network.

4. Week 3: Convolutional Neural Networks (CNNs)

Standard neural networks struggle with image data due to the high dimensionality. CNNs solve this by using **filters** to scan the image, preserving spatial relationships.

Key operations include:

- **Convolution:** Sliding a kernel over the input to produce feature maps (e.g., edge detection).

- **Padding:** Adding border pixels to maintain image dimensions and prevent information loss at the edges.
- **Pooling:** Reducing the spatial size (downsampling) to decrease computational load and prevent overfitting. Max Pooling is the most common technique.

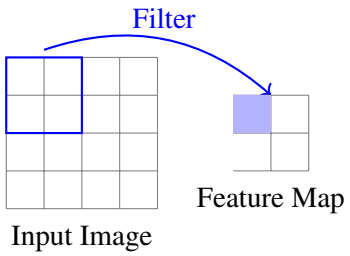

Figure 3: The Convolution Operation: A filter slides over the input to generate a condensed feature map.

5. Week 4 & 5: Sequence Modeling and RNNs

While CNNs excel at spatial data, Recurrent Neural Networks (RNNs) are designed for sequential data like text or time-series.

The RNN Mechanism

RNNs maintain a "Hidden State" (H_t) that acts as a memory of previous inputs. At each step, the network considers both the current input (X_t) and the previous hidden state (H_{t-1}). This allows context to flow through the sequence.

Modern Architectures

Basic RNNs suffer from vanishing gradients, where they forget long-term dependencies. To mitigate this, we studied:

- **LSTM (Long Short-Term Memory):** Uses a system of gates (Input, Forget, Output) to strictly regulate information flow.
- **GRU (Gated Recurrent Unit):** A more efficient variant of LSTM with fewer parameters.
- **Seq2Seq Models:** Utilizing Encoder-Decoder architectures for tasks like translation.

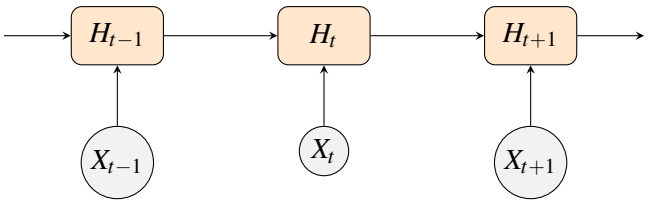

Figure 4: An Unrolled Recurrent Neural Network showing information flow through time.

Conclusion

This first term has built the essential bridge from linear algebra and probability to modern deep learning. We have moved from simple regression to sophisticated sequence modeling. These concepts—specifically the handling of sequences and features—are directly applicable to the Vision Transformer architectures we will tackle next.