1. Introduction

This report sums up exactly what I've covered so far, starting from the very basics of Python and data manipulation, moving through the fundamentals of Machine Learning, and finally diving into complex architectures like Convolutional Neural Networks (CNNs) and Sequence Models.

2. Week 0:Basics

The initial week focused on establishing a robust environment for Machine Learning and Deep Learning by mastering the essential Python libraries.

2.1. Python & Programming Environment

Objective: To learn basics of Python syntax,learn new libraries such as numpy, pandas and matplotlib and the Google Colab environment.

Python basics including data structures (lists, dictionaries), loops, and functions.

2.2. Numerical Computing and Data Manipulation

- **NumPy:** Learned to perform high-performance mathematical operations on multi-dimensional arrays. Key topics included array slicing, broadcasting, and vectorization, which are critical for processing image and tensor data.
- **Pandas:** Focused on data manipulation and analysis. Covered DataFrames, Series, and essential functions for cleaning and preprocessing datasets before feeding them into models.

2.3. Data Visualization

- **Matplotlib:** Explored techniques for visualizing data trends and model performance. I learnt about plotting line graphs, scatter plots, and histograms to understand data distribution.

Assignment Summary: The Week 0 assignment involved practical exercises in Python to manipulate data using Pandas and NumPy, ensuring readiness for handling complex datasets in subsequent weeks.

---

3. Week 1: Machine Learning Fundamentals & Neural Networks

Week 1 bridged the gap between basic coding and intelligent systems, introducing core Machine Learning algorithms and the architecture of Neural Networks.

3.1. Supervised and Unsupervised Learning

- **Linear & Logistic Regression:** Studied the mathematical foundations of regression for continuous data and classification for categorical data. Concepts included the closed-form solution and the iterative optimization approach.
- **Gradient Descent:** Detailed analysis of how models "learn" by minimizing the loss function via Gradient Descent.
- **K-Means Clustering:** Introduction to unsupervised learning for grouping unlabeled data based on feature similarity.

3.2. Fundamentals of Neural Networks

- **Architecture:** Understood the perceptron model, hidden layers, and the flow of information (forward propagation).
- **Activation Functions:** Analyzed the role of non-linearity using functions like Sigmoid, Tanh, and ReLU to enable the network to learn complex patterns.
- **Loss Functions:** Learned how to quantify model error using metrics appropriate for the task (e.g., MSE for regression, Cross-Entropy for classification).

3.3. Optimization and Regularization

- **Backpropagation:** Studied the algorithm for calculating gradients to update weights.
- **Regularization:** Techniques to prevent overfitting were covered, specifically L1/L2 regularization and **Dropout**, which randomly deactivates neurons during training to improve generalization.
- **Optimizers:** Explored advanced optimization algorithms like Stochastic Gradient Descent (SGD), RMSProp, and Adam to speed up convergence.

Assignment Summary: Assignments focused on implementing regression models from scratch and building a basic neural network, experimenting with different activation functions and observing the effect of learning rates.

---

4. Week 2: Deep Learning & Computer Vision

The focus shifted to specialized architectures for image data, specifically Convolutional Neural Networks (CNNs).

4.1. Convolutional Neural Networks (CNNs)

- **Concept:** Moved beyond standard Dense networks to CNNs, which preserve the spatial hierarchy of images.
- **Key Layers:**
  - **Convolutional Layers:** Feature extraction using learnable filters.
  - **Pooling Layers:** Reducing dimensionality to decrease computation and prevent overfitting.

- **Resources:** 3Blue1Brown's visual explanations of neural networks to visualize feature maps.

4.2. Advanced Techniques

- **Batch Normalization:** Learned to normalize layer inputs to stabilize the learning process and dramatically accelerate training.
- **Visualization:** techniques to visualize what a CNN "sees" were explored to interpret model decisions.

---

5. Week 3 & 4: Sequence Models (RNNs, LSTMs, GRUs)

The final weeks covered in this midterm period addressed sequential data, essential for tasks like time-series prediction and natural language processing.

5.1. Recurrent Neural Networks (RNNs)

- **Architecture:** Introduced networks with loops to allow information to persist.
- **Limitations:** Discussed the "Vanishing Gradient" problem which makes standard RNNs struggle with long-term dependencies.

5.2. Advanced Sequence Models

- **LSTM (Long Short-Term Memory):** Detailed study of LSTM cells, including the forget, input, and output gates that regulate the flow of information and solve the vanishing gradient issue.
- **GRU (Gated Recurrent Unit):** Covered GRUs as a simplified, computationally efficient alternative to LSTMs.
- **Bidirectional RNNs:** Explored architectures that process data in both forward and backward directions to capture context from the entire sequence.

Assignment Summary:

Assignment 4: Optimization techniques for Deep Neural Networks. This assignment focuses on understanding and implementing **Batch Normalization** to stabilize and accelerate the training of deep networks.

Assignment 5: To build the necessary background for understanding Transformers, this assignment introduces sequence modeling concepts using Recurrent Neural Networks (RNNs)

6. Conclusion

Over the past four weeks, I have developed a strong foundation in Deep Learning. Starting with data manipulation in Python, I have progressed to building and optimizing complex Neural Networks. I am now proficient in constructing CNNs for computer vision tasks and LSTMs/GRUs for sequence data. These skills form the necessary groundwork for the upcoming advanced topics in the course, such as Vision Transformers (ViT).