VISION TRANSFORMER MID TERM REPORT

1. INTRODUCTION

The project started with Python and basic data handling and gradually moved towards machine learning fundamentals, neural networks, convolutional neural networks, and recurrent neural networks. The focus was to build strong foundations and understand how models work internally instead of using them as black boxes.

2. WEEK 0 – PYTHON, NUMPY, PANDAS AND MATPLOTLIB

2.1 Python Basics

In Week 0, Python fundamentals required for machine learning were revised. This included variables and data types, lists, tuples, and dictionaries, loops and conditional statements, functions, and basic use of classes. These basics were necessary to write clear and structured code in later assignments.

2.2 NumPy

NumPy was used for numerical computation and matrix operations. We learned how to create arrays and matrices, perform indexing, slicing, and masking, use broadcasting, and apply vectorized operations instead of loops. This helped in understanding how numerical data is handled efficiently in machine learning models.

2.3 Pandas

Pandas was used to work with CSV datasets. We learned how to load datasets into DataFrames, sort data using different columns, split datasets into training and testing sets, and create batches for training. Pandas helped in organizing and preprocessing data before feeding it into models.

2.4 Matplotlib

Matplotlib was used for data visualization. We plotted bar graphs, data distributions, and comparisons between training and testing datasets. Visualization helped in understanding data imbalance and overall data patterns.

3. WEEK 1 – MACHINE LEARNING FUNDAMENTALS

3.1 Activation Functions and Loss Functions

Different activation functions such as ReLU, Tanh, and Softmax were studied. Loss functions were introduced to measure model error. We learned why different tasks require different activation and loss functions.

3.2 Linear and Logistic Regression

Linear regression was studied for continuous outputs, while logistic regression was used for classification tasks. Gradient descent was introduced as an optimization technique. Closed-form solutions and iterative optimization methods were compared.

3.3 K-Means Clustering

K-Means clustering was introduced as an unsupervised learning algorithm. Concepts such as centroid initialization, distance calculation, and iterative updates were covered. Visualizations helped in understanding how clusters are formed.

3.4 Evaluation Metrics

Evaluation metrics such as accuracy, precision, recall, and F1-score were studied. Confusion matrices were used to understand classification performance, especially in the case of imbalanced datasets.

3.5 Regularization and Optimizers

Topics included L1 and L2 regularization, dropout to prevent overfitting, gradient descent, stochastic gradient descent, and optimizers like Adam and RMSProp. These methods help improve generalization and training stability.

VISION TRANSFORMER MID TERM REPORT

4. WEEK 2 – CONVOLUTIONAL NEURAL NETWORKS (CNNs)

4.1 CNN Components

Key components studied were convolution layers, filters and kernels, padding and stride, pooling layers, and fully connected layers. CNNs are effective in extracting spatial features from images.

4.2 Batch Normalization

Batch normalization was introduced to speed up training, reduce sensitivity to weight initialization, and improve convergence. We learned where batch normalization is applied in a network and why it improves training stability.

5. WEEK 3 – RECURRENT NEURAL NETWORKS (RNNs)

5.1 RNN Basics

We studied how RNNs handle sequential data using hidden states. Problems such as vanishing and exploding gradients were discussed. RNNs were explained using time-step based processing.

5.2 Applications

RNNs are useful for text generation, language modeling, and time-series analysis. The limitations of simple RNNs were also discussed.

VISION TRANSFORMER MID TERM REPORT

6. WEEK 4 – ADVANCED RNNs

6.1 LSTM (Long Short-Term Memory)

LSTMs solve the vanishing gradient problem using gates such as the forget gate, input gate, and output gate. They are capable of learning long-term dependencies in sequences.

6.2 GRU and Bidirectional RNNs

GRUs provide a simpler alternative to LSTMs with fewer gates. Bidirectional RNNs process sequences in both forward and backward directions, improving performance in sequence understanding tasks.

7. CONCLUSION

Through the first five weeks, this project helped build a strong foundation in machine learning and deep learning. Starting from Python and data handling, we progressed towards understanding neural networks, CNNs, and RNNs in detail. The step-by-step approach made complex concepts easier to understand and prepared us for advanced topics such as Vision Transformers.