Midterm Report

This midterm report describes the work covered so far in the Vision Transformer programming foundations course. The main goal of the assignments and classwork was to build a strong base in Python programming and data handling, which is required before moving on to complex machine learning models like Vision Transformers. The work done till now mainly focused on writing correct code, understanding how data is processed, and learning how datasets are prepared for training models.

What We Covered So Far

Up to the midterm, we have covered Python basics, object oriented programming, problem solving using loops and conditions, numerical operations using NumPy, basic probability based classification logic, data handling using Pandas, plotting using Matplotlib, and preparation of datasets for machine learning training. The emphasis was on understanding how things work internally rather than just using built in functions.

Python Programming

In the Python section, we worked on creating our own classes and functions. One of the main tasks was implementing a DataSample class that stores feature values and a label. We wrote methods to normalize the feature values using min max normalization and to scale the features by a given factor. Special attention was given to handling edge cases such as when all feature values are the same. This helped in understanding how in place changes work and how to write safer code.

Another problem involved sorting a list of strings based on the number of unique characters. Instead of using shortcuts, we wrote our own logic to count unique characters and compare strings. This improved our understanding of loops, condition checking, and how sorting works internally.

NumPy and Array Operations

A major part of the coursework focused on NumPy. We learned how to create arrays with specific shapes and value ranges and how to modify them efficiently. Boolean masking was used to replace values based on conditions, which showed how we can avoid loops and still make changes to arrays.

We also practiced slicing arrays to extract submatrices and used advanced indexing to get diagonal elements. Broadcasting was introduced to help perform operations between arrays of different shapes. Using broadcasting, we created a matrix where each element depended on its row and column indices. These exercises made it clear why NumPy is much faster and more useful than normal Python lists for numerical work.

Probability and Prediction Logic

We also worked on a task that involved converting raw scores into probabilities. This included subtracting the maximum value in each row to avoid large numbers, applying the exponential function, and then normalizing the values so that each row summed to one. This process is similar to how softmax works in machine learning models.

After computing probabilities, we predicted class labels and calculated accuracy by comparing predictions with true labels. This gave us a basic idea of how classification and evaluation work in machine learning.

Pandas and Working with Real Data

Using Pandas, we worked with a real dataset related to student performance. We learned how to load a dataset, look at its structure, and sort data based on different columns like gender, race, and exam scores. This helped us understand how datasets are explored before training any model.

We also split the dataset into training and testing parts. To check if the data was evenly distributed, we plotted graphs showing the race distribution in both sets. This showed that random splitting may sometimes lead to imbalance.

Data Visualization and Balancing

Matplotlib was used to create bar graphs for comparing distributions between training and testing datasets. To improve balance, we tried sorting the dataset by race before splitting it. This resulted in more similar proportions in both sets and helped us understand why data balance is important.

Batch Creation

Another concept we learned was creating batches from the training data. The training dataset was divided into smaller batches of fixed size. This showed how large datasets are usually processed in parts during training to save memory and make learning more efficient.

Conclusion

Overall, the work covered till the midterm helped in building a solid foundation in Python and data handling for machine learning. The assignments were helpful in understanding how data is prepared, processed, and evaluated before training advanced models like Vision Transformers. This groundwork will make it easier to understand the upcoming topics in the course.