Question - 01
Multi-Class Image Classification using K-Nearest Neighbors

CIFAR-10 Dataset

1 Introduction

Image classification is a fundamental problem in machine learning and computer vision. This project investigates the performance of the K-Nearest Neighbors (KNN) algorithm for multi-class image classification using the CIFAR-10 dataset. Unlike parametric models, KNN is a non-parametric and instance-based learning algorithm that relies entirely on distance computations between samples.

The objective of this work is to implement a KNN classifier completely from scratch, analyze the effect of different distance metrics and values of K, and evaluate the classifier using accuracy, confusion matrix, precision, and recall.

2 Dataset Description

The CIFAR-10 dataset consists of 60,000 color images of size $32 \times 32 \times 3$ distributed evenly across 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

The dataset is provided in five training batches (`data_batch_1` to `data_batch_5`), each containing 10,000 images, and one testing batch (`test_batch`) containing 10,000 images. For computational feasibility, subsets of the training and testing data were used in the experiments.

3 Data Preprocessing

Each image was flattened into a one-dimensional vector of length 3072. Pixel values were normalized to the range $[0, 1]$ by dividing by 255. The corresponding labels were extracted and stored as integer class indices from 0 to 9.

4 K-Nearest Neighbors Classifier

The KNN algorithm assigns a class label to a test sample based on the majority class among its K nearest neighbors in the training set. For each test image, distances to all training samples were computed using a chosen distance metric. The K samples with the smallest distances were selected, and the most frequent label among them was assigned as the predicted class.

5 Distance Metrics

The following distance metrics were implemented and evaluated:

- **Euclidean Distance**

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- **Manhattan Distance**

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

- **Minkowski Distance** (order $p = 3$)

$$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

- **Cosine Distance**

$$d(x, y) = 1 - \frac{x \cdot y}{\|x\| \|y\|}$$

- **Hamming Distance**

$$d(x, y) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}(x_i \neq y_i)$$

6 Experimental Setup

Experiments were conducted for different values of $K = \{1, 3, 5, 7, 9\}$. For each distance metric and value of K, classification accuracy was computed on the test dataset. Accuracy trends were visualized using plots of accuracy versus K.

7 Evaluation Metrics

The performance of the classifier was evaluated using the following metrics:

7.1 Accuracy

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Predictions}}$$

7.2 Confusion Matrix

A 10×10 confusion matrix was computed, where each entry represents the number of samples belonging to a true class that were predicted as a given class.

7.3 Precision and Recall

For multi-class classification, average precision and recall were computed as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

8 Results and Analysis

The experimental results show that Euclidean and Cosine distance metrics consistently outperform other distance measures. Accuracy improves as K increases initially, due to reduced sensitivity to noise, but degrades for larger values of K as the decision boundary becomes overly smooth.

Hamming distance performs poorly because it is not well-suited for continuous-valued image data. Manhattan and Minkowski distances provide moderate performance but are generally inferior to Euclidean distance.

9 Observations

- Smaller values of K are sensitive to noise but capture fine-grained patterns.
- Larger values of K reduce variance but may increase bias.
- Distance metric choice significantly affects classification performance.
- KNN is computationally expensive due to exhaustive distance calculations.

10 Conclusion

This project demonstrates that a KNN classifier implemented from scratch can achieve reasonable performance on image classification tasks when appropriate distance metrics and neighborhood sizes are chosen. Despite its simplicity, KNN provides strong baseline performance, though its high computational cost limits scalability to larger datasets.

Future work may include dimensionality reduction, optimized distance computation, or approximate nearest neighbor methods to improve efficiency.