Report

1. **Downloading and Loading Data**:
    - The CIFAR-10 dataset is downloaded and loaded into memory using helper functions from **data_utils.py**.
    - Training and testing data are loaded, and their shapes are printed to verify the data loading.
2. **Data Preprocessing**:
    - The data is visualized to get an understanding of its structure and content.
    - Subsampling is performed to reduce the size of the dataset, which helps prevent memory errors during computation.
    - The training and testing data are reshaped and flattened so that each row represents a single example.
3. **KNN Classifier Implementation**:
    - The KNearestNeighbor class is defined, which contains methods for training the model (**train**), predicting labels (**predict**), computing distances between examples (**compute_distances**), and predicting labels based on distances (**predict_labels**).
4. **Model Training and Evaluation**:
    - The KNN classifier is trained on the training data.
    - The classifier is used to predict labels for the test data, and the accuracy is calculated.
    - Cross-validation is performed to find the optimal value of K. The dataset is split into multiple folds, and the model is trained and evaluated on each fold using different values of K.
    - The average accuracy for each value of K is computed, and a visualization of the cross-validation results is plotted.
5. **Choosing the Best K**:
    - Based on the cross-validation results, the best value of K is chosen.

- The model is trained on the entire training data with the best K value, and predictions are made on the test data.
- The accuracy of the model on the test data is computed and printed.

Why Perform Cross-Validation for Finding K:

- Cross-validation is performed to find the optimal value of K because it helps in selecting a value of K that generalizes well to unseen data.
- By using cross-validation, we can assess how the model performs on different subsets of the training data and choose a value of K that provides good performance across all subsets. This helps prevent overfitting and ensures that the model is not overly biased towards the training data.

Impact of Optimal K on Model Fitting:

- Finding the optimal value of K through cross-validation ensures that the model is not underfitting or overfitting the training data.
- A smaller value of K may lead to overfitting, where the model becomes too complex and captures noise in the training data. On the other hand, a larger value of K may lead to underfitting, where the model is too simple and fails to capture the underlying patterns in the data.
- By choosing the optimal value of K, we can strike a balance between bias and variance, leading to better generalization performance on unseen data.

Comparison with More Sophisticated Models:

- The performance of the KNN classifier on the CIFAR-10 dataset, even with the optimal value of K, may not be as high as that of more sophisticated models such as Neural Networks (NN) or Convolutional Neural Networks (CNN).
- NNs and CNNs can learn hierarchical features from the data and capture complex patterns that may not be discernible with simple distance-based methods like KNN.
- Additionally, NNs and CNNs can automatically learn the relevant features from the raw data, whereas KNN relies on handcrafted features and distance metrics.

- However, KNN has its advantages, such as simplicity, interpretability, and ease of implementation, making it a useful baseline model for comparison. It may perform well on datasets with simple structures or when feature engineering is challenging.