Machine Learning

SPRING 2023-2024

SUPERVISED BY :

Taiman Arham Siddique Faculty, Department of CS

SUBMITTED BY:

Tarikul Islam Nishat

ID:21-44632-1

Section:C

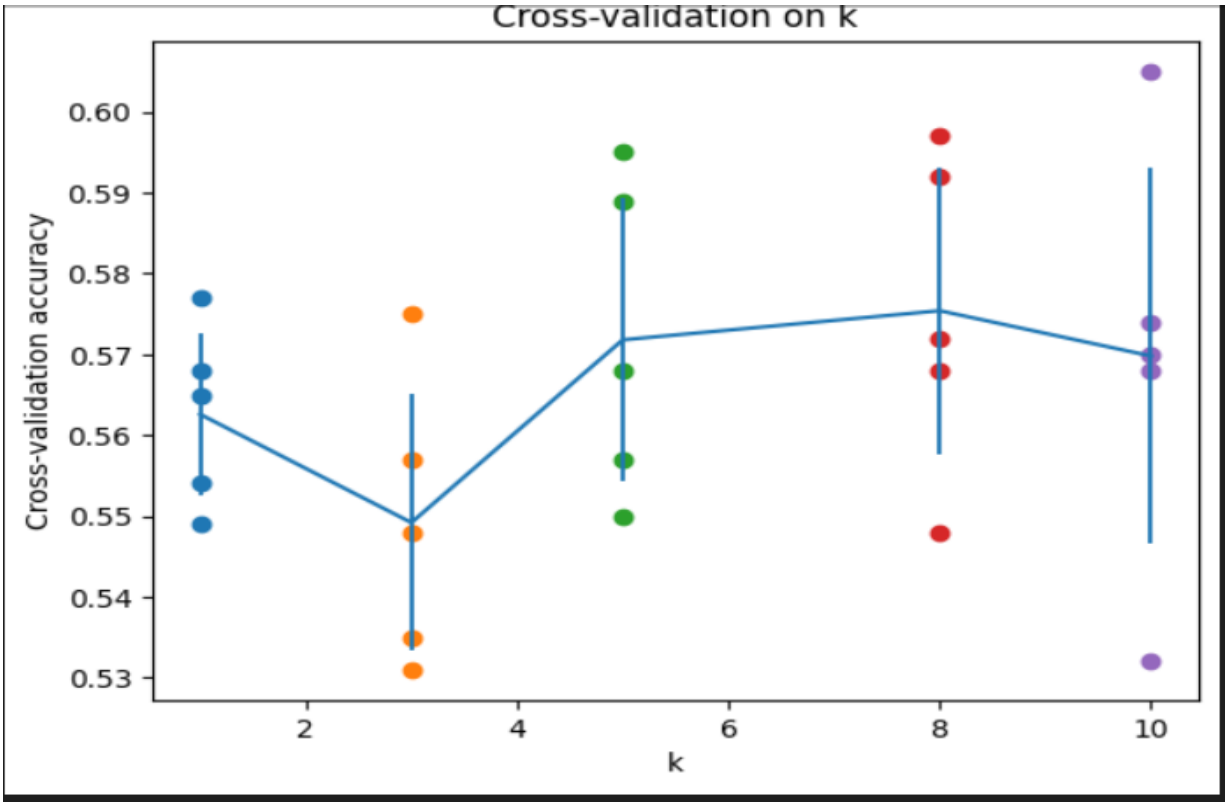
Submissions Date:09/04/2024

Cross-Validation and Optimal K Selection in kNN Classification

Results:

```
Data has apparently already been downloaded and unpacked.
Training data shape:  (50000, 32, 32, 3)
Training labels shape:  (50000,)
Test data shape:  (10000, 32, 32, 3)
Test labels shape:  (10000,)
(10000, 3072) (1000, 3072)
Got 296 / 1000 correct with k=5 => accuracy: 0.296

Printing our 5-fold accuracies for varying values of k:
k = 1, accuracies = [0.577, 0.568, 0.565, 0.549, 0.554]
k = 3, accuracies = [0.575, 0.548, 0.557, 0.535, 0.531]
k = 5, accuracies = [0.589, 0.568, 0.595, 0.55, 0.557]
k = 8, accuracies = [0.597, 0.592, 0.568, 0.548, 0.572]
k = 10, accuracies = [0.605, 0.574, 0.568, 0.532, 0.57]
```



Output Breakdown and Explanations:

Downloading and unpacking CFAR-10 data:

```
Data has apparently already been downloaded and unpacked.
```

Explanation:

This message indicates that the CIFAR-10 dataset required for the task has been located in the local directory, implying that there was no necessity to redownload or extract the dataset since it had already been obtained and extracted previously.

Training data shape:

```
Training data shape:  (50000, 32, 32, 3)
```

Explanation:

The training dataset consists of 50,000 images. Each image is 32 pixels in height, 32 pixels in width, and has 3 channels (RGB, indicating color images).

Training labels shape:

```
Training labels shape:  (50000,)
```

Explanation:

There are 50,000 labels corresponding to the training images. Each label represents the class of its corresponding image.

Test data shape:

```
Test data shape:  (10000, 32, 32, 3)
```

Explanation:

The test dataset consists of 10,000 images with the same dimensions as the training images (32x32x3).

Test labels shape:

```
Test labels shape:  (10000,)
```

Explanation:
Similar to the training labels, there are 10,000 labels for the test images, with each label indicating the class of its corresponding image.

Data shape after preprocessing :

```
(10000, 3072) (1000, 3072)
```

Explanation:
This indicates the shape of the data after preprocessing by flattening the images. The images have been reshaped or flattened from a 3-dimensional shape (32x32x3) into a vector of 3072 elements (32323 = 3072). It mentions two datasets: one with 10,000 examples and another, possibly a subset, with 1,000 examples, both having 3072 features.

k-NN algorithm with k=5

```
Got 296 / 1000 correct with k=5 => accuracy: 0.296
```

Explanation:
Using the k-NN algorithm with k=5 (considering the 5 nearest neighbors), the model correctly classified 296 out of 1,000 test images, resulting in an accuracy of 29.6%. This relatively low accuracy indicates the challenge of the classification task and/or the limitations of using k-NN for high-dimensional image data.

Printing the 5-fold accuracies for varying values of k:

```
Printing our 5-fold accuracies for varying values of k:
k = 1, accuracies = [0.577, 0.568, 0.565, 0.549, 0.554]
k = 3, accuracies = [0.575, 0.548, 0.557, 0.535, 0.531]
k = 5, accuracies = [0.589, 0.568, 0.595, 0.55, 0.557]
k = 8, accuracies = [0.597, 0.592, 0.568, 0.548, 0.572]
k = 10, accuracies = [0.605, 0.574, 0.568, 0.532, 0.57]
```

Explanation:

This section showcases the outcomes obtained through a 5-fold cross-validation method aimed at determining the most suitable value for k. During 5-fold cross-validation, the training dataset gets partitioned into five segments, with the model being trained and assessed five times. Each iteration employs a distinct segment as the validation set while utilizing the remaining segments for training purposes.

For each k value, a list of accuracies obtained in each of the 5 folds is presented:

k = 1: The model shows accuracies around 54.9% to 57.7% across the folds.

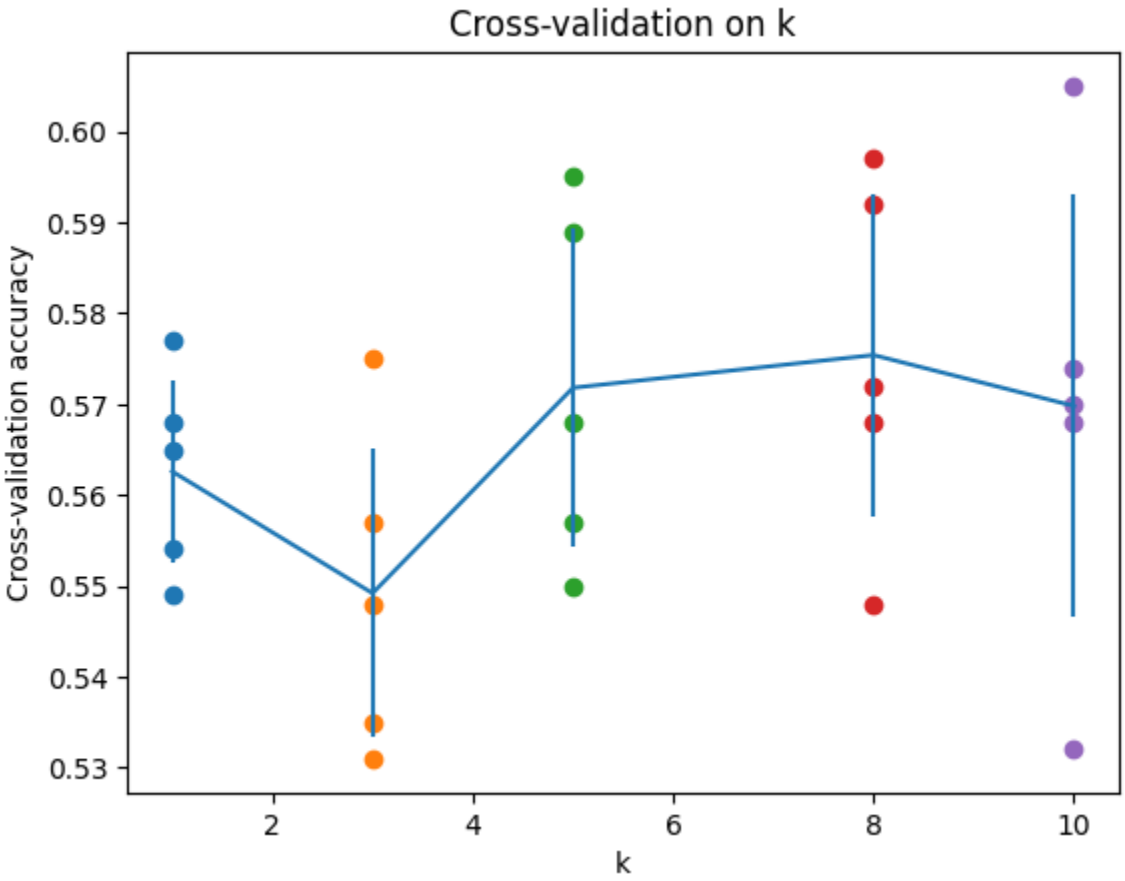
k = 3: Accuracies range from 53.1% to 57.5%, showing some variation but lower than k=1.

k = 5: Accuracies improve slightly, with a range from 55% to 59.5%.

k = 8: Accuracies are again variable, from 54.8% to 59.7%, suggesting some improvement over smaller k values.

k = 10: Shows the highest accuracies in some folds, reaching up to 60.5%, indicating that increasing k might be beneficial to a point.

Graph visualization:



Explanation:

This scatter plot illustrates the fluctuation in accuracy across various values of 'k' and different cross-validation folds. Following the plotting of scatter points, error bars are incorporated to represent the average accuracy and its variance for each 'k' value.

Best *k* found through cross-validation:

```
Best k found through cross-validation: 8
```

Explanation:

After cross-validation, (k=8) was chosen as the optimal value. This suggests it balances bias and variance well, leading to higher accuracy.

Testing the model with (k=8) :

```
Got 572 / 1000 correct on test data => accuracy: 0.572000
```

Explanation:
Testing the model with \(k=8\) on new data resulted in a 57.2% accuracy rate. This indicates the model's ability to generalize to unseen data, showcasing its performance.

Discussions:

Cross-validation was conducted to ensure that the selected value of k produces a model that effectively generalizes to new data. This process serves multiple purposes:

- Preventing Overfitting: Cross-validation guards against overfitting by assessing the model's performance on diverse subsets of the data, reducing the influence of any single subset.

- Evaluating Generalization Performance: By testing the model on various data subsets, cross-validation offers a robust estimation of its ability to generalize to unseen data, enhancing confidence in its predictive performance.

- Hyperparameter Optimization: Since k is a crucial hyperparameter in the k-NN algorithm, cross-validation facilitates the systematic evaluation of the model's performance across different k values, aiding in the selection of the most effective one.

Selecting the optimal k value improves model fitting in several ways:

- Enhanced Accuracy: By striking a balance between bias and variance, the optimal k value results in a model with higher accuracy on unseen data.

- Mitigated Overfitting: Optimal k selection reduces the risk of overfitting, where the model excessively memorizes the training data rather than capturing underlying patterns. Hence, cross-validation helps identify the k value that achieves a balanced trade-off between bias and variance, leading to a well-performing model on new data.

Performance comparison with Neural Networks (NN) or Convolutional Neural Networks (CNNs) on the CIFAR-10 dataset:

NNs and CNNs: These models typically outperform simple algorithms like k-NN on complex datasets like CIFAR-10. They can learn intricate patterns and hierarchical representations from raw pixel data, leading to superior performance.

Advantages of *k-NN*:
k-NN has its advantages, such as simplicity and interpretability. It doesn't require training and can be effective for small datasets or when computational resources are limited.

Trade-offs:
While k-NN serves as a basic benchmark on CIFAR-10, more advanced models like NNs and CNNs are anticipated to surpass it in performance, albeit with added complexity and computational demands.

Overall, while k-NN may provide a baseline performance on CIFAR-10, more sophisticated models like NNs and CNNs are expected to outperform it, albeit at the cost of increased complexity and computational requirements.