機器學習實務與應用

Homework #7
Due 2019 April 15 9:00AM

Exercise 1. (**k-Nearest Neighbors**) Please see readme.txt for preparing the CIFAR-10 dataset. Then you can run Module3_Part5_template.py to see the classification result of 1-NN with L2 (or L1) distance metric for CIFAR-10.

1. (**k-NN**) Please follow the comments (TODO 1) in the template to implement the function `get_predictions()` for returning the predictions of k-NN when k > 1. (The template only implements the case k = 1.) Report the classification accuracies in the following table.

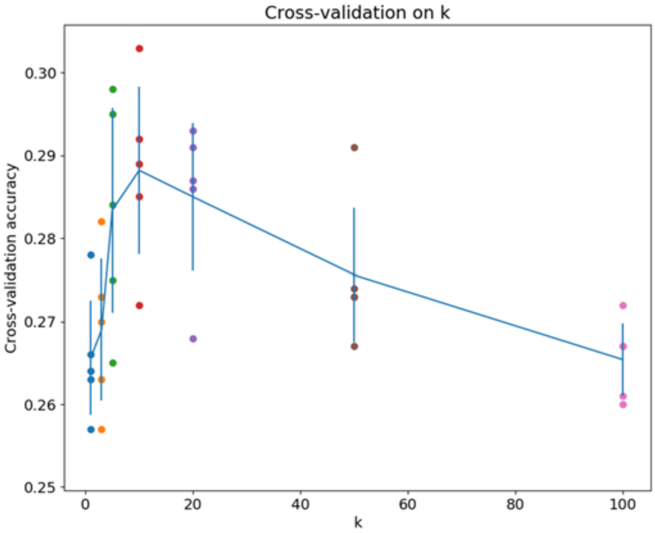
	L2 metric	L1 metric
k = 5		
k = 10		

Hint: It takes a few seconds (for my laptop, about 30 seconds) to make predictions using k-NN. When developing your own code, at first you can consider a smaller subset of CIFAR-10. For example, reducing the training samples from 5000 to 1000 as

```
get_cifar10_subset(X_train, y_train, X_test, y_test, 1000)
```

This allows you to test your code more efficiently. Remember to switch back to 5000 when you think your code is ready.

2. (**Cross-validation**) Please follow the comments (TODO 2) in the template to implement cross-validation. The figure below is the result of cross-validation using the L2 distance metric. Plot the figure for cross-validation using the L1 distance metric.



3. (**Vectorization**) Please follow the comments (TODO 3) in the template to implement the function `get_distances()` for the vectorization of calculating the L2 distance matrix. Vectorization refers to the elimination of the use of loops. **Hint**: Vectorization can be achieved by observing

$$\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 = \|\mathbf{x}_1\|_2^2 + \|\mathbf{x}_2\|_2^2 - 2\langle \mathbf{x}_1, \mathbf{x}_2 \rangle.$$

Report the timings (in seconds) of the vectorization version and the original version, respectively.