Pattern Recognition –HW#2

About the Assignment

The main aim of the assignment is to make you familiar with a traditional classifier by using KNN. Contributions of this lab are;

- Learning the K-NN classification method with custom similarity.
- Understanding the idea behind the classification task.

The algorithm for KNN classification can described as follows:
1. Computer k nearest neighbors from the training dataset for test sample to be classified.
2. Classify test sample to the class having most training samples among the k nearest neighbors.

Step1:

Download the Cifar-10 dataset python version by using the following commands

```
from keras.datasets import cifar10
(x_train, y_train), (x_test, y_test) = cifar10.load_data()
```

Step3:

Convert images to vector format by writing this snipped code. It means that an image with 32x32x3 channels converted as 1x3072 vector. There 50,000 train vectors and 10,000 test vectors.

```
x_train = x_train.reshape(-1, 3072)
x_test = x_test.reshape(-1, 3072)
```

Step4:

Write a function send the parameters of ***x_train, y_train, sample_test, k*** as input, then return the most similar class name for sample_test. In case of computing the similarity, you are expected to use the **Custom Similarity**, which is explained in lecture notes by Teacher in class. You can use any code, but the following function can help you.

$$\text{similarity} = \frac{\exp(A)\cdot\exp(B)}{\|\exp(A)\|\cdot\|\exp(B)\|} + \frac{\sqrt{A}\cdot\sqrt{B}}{\|A\|\cdot\|B\|}$$

```
import numpy as np
x = np.array([-0.2, 0]) # assume x is test sample (x_test)
y = np.array([1, 1])     # assume y is one of train sample

x = x.reshape(1, -1)
y = y.reshape(1, -1)

s = custom_similarity(x, y)
print('similarity:', s)
```

sample_test refers to a test vector from x_test. You can set the sample_test like this way. You will compute similarities between test and train samples. Then, you will sort similarities in descending order (for example 100.45, 87.20, 65.4, 40.12, 23.10, 18.10, 11.11, 05.00). Choose the most similar ones based on k=5.

```
sample_test = x_test[1,:];
```

The test code look like this.

```
sample_test = x_test[0,:];
k=5
similar_class_name = knnCustomSimilarity(x_train, y_train, sample_test, k )
print(similar_class_name)
```

Submit the Assignment

Ex: No_Name_Surname_HW#.zip