

Handwriting Digit Recognition

Submission format: Submit a PDF-version report for all problems, including source codes and print screen results. You can use any text/code editor. Google Colab is recommended.

Problem 1 (KNN)

In this problem you will use KNN to recognize handwritten digits.

First, use "download_mnist.py" file to download the MNIST database. This file will make data to numpy arrays and save it as Pickle ("mnist.pkl"). The MNIST dataset includes:

`x_train` : 60,000x784 numpy array that each row contains flattened version of training images.

`y_train` : 1x60,000 numpy array that each component is true label of the corresponding training images.

`x_test` : 10,000x784 numpy array that each row contains flattened version of test images.

`y_test` : 1x10,000 numpy array that each component is true label of the corresponding test images.

Remark 1: Once you get "mnist.pkl", you don't need to call `init()` anymore. Everything you need to do is to locate "download_mnist.py" and "mnist.pkl" in your working directory and to call `load()`. Then you can load the MNIST database in "knn.py".

Remark 2: Due to the high computational complexity of KNN, you do not need to use all the training data or all the testing data. Instead, you can change the data length you need in the code.

Remark 3: Part of "knn.py" for data loading and results printing has been given. You can utilize them or write your own codes.

Requirements: 1) Change the value of k and choose different measurement metric, like L1 or L2 distance, to see the different performance. 2) the final accuracy should be over 95% when using L2 distance.

Problem 2 (Linear Classifier)

In this problem, you need to review the contents about linear classifier and train a simple linear classifier to recognize handwritten digits.

First, using "Cross Entropy" as the loss function, and then adopt "Random Search" to find the parameters \mathbf{W} . Next, check the recognition accuracy using the testing set.

Requirements: Write the code to implement the above linear classifier and print the recognition accuracy. Since this problem is to help you be familiar with linear classifier, it is not required to achieve a high recognition accuracy.