CS412 Machine Learning Homework 1

Due: Saturday, March 18, 11:00 pm **Late Accepted Until:** Monday, March 20, 11:00 pm

Starter Notebook Link

https://colab.research.google.com/drive/12GVxU93gxlemmRlzdHHX9A83G9RUGAgX?usp=sharing

Goals

The goal of this homework is four-fold:

- Introduction to the machine learning experimental setup
- Gain experience with the k-NN method
- Learn to perform hold-out validation for hyperparameter optimization
- Gain experience with the Scikit-learn (Sklearn) library

Dataset

The **MNIST** dataset consists of a collection of 28×28 grayscale images of handwritten digits (0-9), with each pixel represented as a gray-level value between 0 and 255. The dataset is commonly used as a benchmark for training and evaluating machine learning models for image classification tasks.

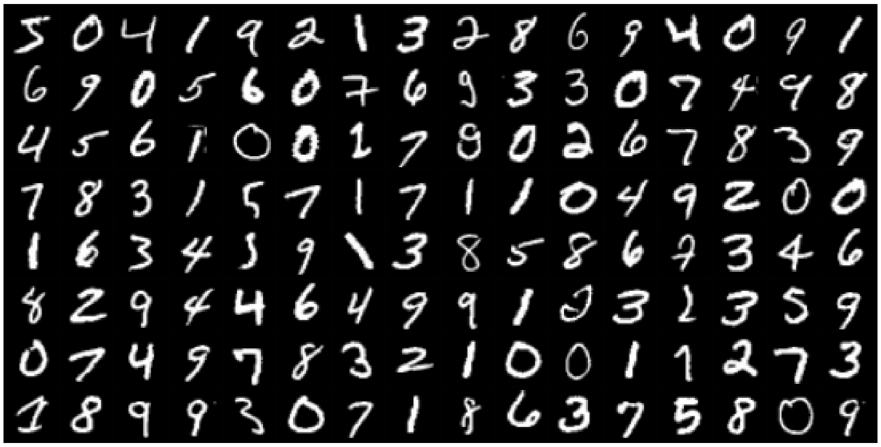

Figure 1: Samples from the MNIST dataset

To download the MNIST dataset, you will use the Keras[1] library. You will split the training data into two sets: a development set for training our models and a validation set for testing the performance of our models during development. You will reserve **20% of the training data for validation**, and use the **remaining 80% for training** our models (no need for cross-validation as you have plenty of data).

[1] https://keras.io/api/datasets/mnist/

It is important to note that **the official test set of 10,000 samples should not be used for model selection or hyperparameter tuning** during development. This test set should only be used at the end of our project to evaluate the final performance of our chosen model.

Task

Your task is to implement a k-NN classifier[2] using the Scikit-learn library. You will train the k-NN classifier using the training set and tune the hyperparameters to optimize its performance on the validation set. Specifically, **you will find the optimal number of nearest neighbors** (`n_neighbors`, see documentation) to use.

To find the optimal value of `n_neighbors`, you should try using the values [1, 3, 5, 7, 9, 11, 13] and evaluate the performance of the classifier on the validation set for each value.

Once you have found the optimal value of `n_neighbors`, you should retrain the k-NN classifier by combining the training and validation sets and evaluate its performance on the test set. This will give you an estimate of how well your classifier will perform on new, unseen data.

Additionally, **you are required to plot the validation accuracy with respect to different values of** `n_neighbors`. This can be done by creating a plot where the x-axis represents the values of `n_neighbors`, and the y-axis represents the validation accuracy for each value. This plot will help you visualize the relationship between `n_neighbors` and the accuracy of the k-NN classifier on the validation set, and will allow you to choose the optimal value of `n_neighbors` more easily. You can use the **`matplotlib`** library to create the plot.

Submission Guideline

You will be supplied a **starter notebook** that you will need to fill.

- **Fill your notebook** to train your classifiers, select the best model on validation, and test on the test data. As training and testing may take a long time, we may just look at your notebook results; so make sure **each cell is run**, so outputs are there.

- **Put the report part of your notebook (see the Report part of the notebook) separately in a PDF document** and **include a link to your notebook at the top of your PDF** (make sure to include the link obtained from the **Share** button link on the top right), as a PDF file.

- **Submit your PDF report to SUCourse** - with the name: **HW1-CS412-Yourname.pdf**

Questions?

- You should ask all your Google Colab-related questions to Discussions and feel free to answer/share your answer regarding Colab.

- You can also ask/answer about which functions to use and what libraries...

- However, you should not ask about the core parts, that is what is validation/test, which one shd. have higher performance, what are your scores, etc.

[2] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html