Due: Feb. 29, 2024 @ 11:59 p.m.

General notes to keep in mind:

- *All deliverables for the assignment must be submitted as a* ***single ZIP file per group*** *via the Brightspace D2L course shell. Submissions containing multiple ZIP files per group or those with a file that is not in the ZIP format will NOT be graded.*
- *The code submitted must be written purely using the Python programming language and it should execute within the Python 3.12.1 interpreter running on the Windows operating system (version 10 or above). The submitted code should NOT require external python modules other than scikit-image 0.22.0, fastai 2.7.14, torch 2.2.0, scikit-learn 1.4.0, matlplotlib 3.8.2, pandas 2.2.0, imagecodecs 2024.1.1 and their dependencies.*
- *Read the "Assignment code submission requirements" carefully and prepare the code accordingly. It is your responsibility to ensure that the submitted code executes. If the grader is unable to execute your code and/or your code does NOT adhere to the submission requirements, your code may not be graded.*
- *The written responses required to the questions in the assignments must be compiled into* ***single PDF*** *file named as* `report.pdf`*. You are encouraged to use LaTeX for typesetting your written responses, but however, the use of Microsoft Word™ or any other such programs is also acceptable.*

Handwritten digit recognition

The handwritten digits problem has been a significant challenge in the field of computer vision, serving as a benchmark for evaluating image classification algorithms. The origins of this problem can be traced back to the development of the Modified National Institute of Standards and Technology (MNIST) database which was created in the late 1990s by Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The dataset consists of a collection of 28×28 pixel grayscale images of handwritten digits (0 through 9). As sample set of MNIST images are shown in Figure 1. The goal was to create a standardized dataset for training and testing machine learning models on handwritten digit recognition tasks. Since its inception, the MNIST dataset has become a standard benchmark for assessing the performance of various image classification algorithms, particularly in the context of deep learning. The dataset gained popularity due to its simplicity and ease of use, making it a common starting point for researchers, students, and practitioners interested in exploring computer vision techniques. In this assignment, you will **explore machine learning based approaches for solving this classic and famous image classification task.**

Figure 1: Handwritten digit images in the MNIST database

Data

The data for this assignment can be downloaded from here. It consists of a subset of 60,000 labeled images of handwritten digits from the MNIST database given in `mnist_train_data.npy` and `mnist_train_labels.npy`. The `README.txt` file contains information regarding the content of these `.npy` files which can be loaded using numpy.load(). Further, a subset of 10,000 labeled images of handwritten digits has been set aside as the "independent test" dataset. You are intentionally "blinded" from the "independent test" dataset as it will be used to perform an independent validation of your submitted classification model mentioned in Question 5.

Image to feature vector conversion

You are required to convert the images into feature vectors for building the various machine learning models as discussed in *Lecture 4, Slide 16*. Pay attention to the detail that our convention is to create the feature vector by stacking the *columns* in the image matrix as opposed to the stacking of the *rows*.

Performance reporting convention

Always report the *error rate*, i.e., 1 - *accuracy* when summarizing the performance ("Err") of a classification model.

Question 1 [5 marks]

Divide the provided dataset of 60,000 handwritten digit images into "training" and "test" datasets respectively satisfying the below conditions:

- The "training" and "test" datasets when taken together should contain exactly the provided 60,000 images, i.e., no missing or duplicate images.
- The "training" dataset should be balanced, meaning it should contain an equal number of samples from each of the 0 to 9 digit classes.
- The "test" dataset should contain *at least* 10% of the images from each of the 0 to 9 digit classes.

Question 2 [20 marks]

Train a kNN classification model using the *Euclidean* distance metric to classify the handwritten digit images into the 0 to 9 digit classes. Use the "training" dataset and a cross-validation (CV) based grid-search approach to tune the "k" parameter of the kNN classification model. Using the "best" "k" setting, re-train on the entire "training" dataset to obtain the final kNN classification model. Estimate the "Err" of this final model on the "test" dataset. Plot and discuss the performance of the models explored during the "k" hyperparameter tuning phase as a function of "k".

Question 3 [25 marks]

Train a non-linear support vector machine (SVM) classifier with a *polynomial kernel* for the handwritten digit classification task. Use the "training" dataset and a CV based grid-search approach to tune both the regularization parameter "C" as well as the degree "d" parameter of the polynomial kernel. Using the "best" "(C, d)" setting, re-train on the entire "training" dataset to obtain the final polynomial kernel SVM model. Estimate the "Err" of this final model on the "test" dataset. Compare the performance of the polynomial kernel SVM model with that of the above obtained final kNN model.

Question 4 [25 marks]

Now train a multi-layer perceptron (MLP) aka deep neural network using the *ReLU* activation function for the handwritten digit classification task. Use the "training" dataset and a CV based grid-search approach to tune both the number of hidden layers "L" as well as the number of hidden units "K". Using the "best" "(L, K)" setting, re-train on the entire "training" dataset to obtain the final MLP model. Estimate the "Err" of this final model on the "test" dataset. Compare the performance of the final deep neural network model with the above obtained final polynomial kernel SVM and kNN models.
Note: *You are required to submit the* `.pth` *(PyTorch) model file for the final trained deep neural network model.*

Question 5 [25 marks]

Leveraging the experience you gained from the experiments thus far, design the "best" classifier for the handwritten digit classification problem. You may explore any other methods that we have not discussed in the course to train this "best" classifier. The only restriction is that, you may NOT use datasets other than the ones provided as part of this assignment. Submit this "best" classifier as the following method:

```python
def classifyHandwrittenDigits(Xtest, data_dir, model_path):
"""Returns a vector of predictions with elements "0", "1", ..., "9",
corresponding to each of the N_test test images in Xtest

Xtest                   N_test x 28 x 28 matrix of test images

data_dir                full path to the folder containing the following files:
                        mnist_train_data.npy, mnist_train_labels.npy

model_path (optional)   full path to a deep neural network model in the .pth (PyTorch) format
"""
    # your code goes here
    # ....

    return ytest
```

The above method will be evaluated on the "independent test" dataset by the grader to determine the classification performance. Note that you are required to submit the model file in case you classifier is based on a deep neural network.

Note on grading

The grading for Question 1, Question 2, Question 3 and Question 4 will be based on the appropriateness of the submitted code and the written responses. The grading for Question 5 will be based on the relative performance of your submitted classification model. The submission(s) with the best performing (referred below as 1^{st} ranked model) in terms of *error rate* (rounded to 4 decimal places) will receive full marks on Question 5 (i.e., 25 marks). All other submissions will receive marks that are proportional to the increase in the achieved error rate with respect to the 1^{st} ranked model. For example, if the error rate of the classification model of a given submission is 10% higher than the 1^{st} ranked one, then that submission will receive 22.5 marks for Question 5.