CA2: EE5907/EE5026 Programming Assignment (50 marks)

20% of Final Grade for EE5907; 40% of Final Grade for EE5026

Project Deadline: 11.59pm, Monday, April 06th, 2020

Submit the following electronic files in one zipped folder onto the CA2 Submission folder on LumiNUS. The zip filename MUST be "[name_on_matric_card]_[matric_number]_CA2.zip"

1. The pdf file of a well-written, concise project report. The report should NOT be longer than 10 pages (font 12, single space, arial). The report filename MUST be "[name_on_matric_card]_[matric_number]_report.pdf".

2. Your source code folder.

3. Readme file containing instructions to run your code. This readme file MUST be inside your source code

4. Please DO NOT include the MNIST Database in your zip folder.

Before you start, take note of the following:

1. You may discuss the assignment with your classmates, but must write the code completely on your own. Plagiarism will be severely punished.

2. You can use matlab or python.

3. The data sets (the MNIST Database) can be downloaded from the given link (http://yann.lecun.com/exdb/mnist/). The data can be read in matlab format using loadMNISTImages.m and loadMNISTLabels.m (https://github.com/davidstutz/matlab-mnist-two-layer-perceptron), (Note: the unzipped data files can also be found in this link) which can also be quite easily read inside python with the right package. If you can't figure out the right package, you probably should be using matlab :)

4. For all the questions, there are publicly available software libraries that implement some versions of these classifiers. However, implementing the algorithms yourself will help you understand the theory better. Therefore you are expected to implement the classifiers yourself. Don't forget to cite the references if you used/adopted their codes/materials.

5. The evaluation criteria include organization and clarity of the report, correctness of the implementation, and performance of the classifiers.

6. Please be considerate to your GA. Be as clear as possible in your submission, e.g., having a clear readme and provide helpful comments in your code. You are more likely to get a better grade if your GA can understand your code and report!

(a) **The MNIST Database**
The MNIST data set used in this question can be obtained from
http://yann.lecun.com/exdb/mnist/.

There are 4 files in .gz format:

- train-images-idx3-ubyte: training set images
- train-labels-idx1-ubyte: training set labels
- t10k-images-idx3-ubyte: test set images
- t10k-labels-idx1-ubyte: test set labels

The training set contains 60000 examples, and the test set 10000 examples.

(b) **Calculating Eigenvalues and Eigenvectors for Features Extraction**
This step will only use the training set images:

- Compute the **mean of the training digits**, and subtract the mean from each training data point. Print out the segment of your codes together with a brief description in the main body of the report. (5 marks)
- Compute the **covariance matrix** using all the training images, then apply eigen-decomposition on the covariance matrix to obtain the eigenvalues and the eigenvectors. Print out the segment of your codes together with a brief description in the main body of the report. (5 marks)
- **Visualize the Eigenvectors**: display the eigenvectors corresponding to the 10 largest eigenvalues as 28×28 images. Include your plots in the report. (5 marks)
- Handwritten digit **image reconstruction**. Plot the following rows of images in your write-up:

  (1) First row (original images from the training set): for digit images corresponding to columns 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 in the array "train-images-idx3-ubyte" (784×60000);

  (2) Second row (reconstructed images from the test set): for digits corresponding to columns 4, 3, 2, 19, 5, 9, 12, 1, 62, 8 in the array "t10k-images-idx3-ubyte" (784×10000), plot the digit image obtained by projecting the original digit into the subspace spanned by the top 30 eigenvectors. To do the projection for a new data point $\boldsymbol{x}$, first compute $\lambda_i = \boldsymbol{x}^T \boldsymbol{v}_i$ where $\boldsymbol{v}_i$ is the eigenvector you computed, then your reconstructed data point can be obtained as $\boldsymbol{x}_p = \sum_{i=1}^{30} \lambda_i \boldsymbol{v}_i$ (Note: λ_i, $i = 1, ..., 30$ are the top 30 eigenvalues and $\boldsymbol{v}_i$ are the corresponding eigenvectors). (5 marks)

(c) **Using Eigen-digits to Classify each digit Image**
First subtract each data point in "train-images-idx3-ubyte" and "t10k-images-idx3-ubyte" by the mean $\mathbf{\Psi}$ you computed on "train-images-idx3-ubyte". Then project each data point in "train-images-idx3-ubyte" and "t10k-images-idx3-ubyte" into the subspace spanned by the top 30 eigenvectors you computed previously. Now your digit data lives in a 30-dimensional space (we call these 'projected eigenvalues' as eigen-digits), instead of the original 28×28-dimensional space.

(1) Apply **linear regression** to learn the 30-dimensional data using the training set. Report your implementation and the classification accuracy based on the 30-dimensional data from the test set. (5 marks)

(2) Apply **polynomial regression** to learn the 30-dimensional data using the training set. Report your implementation and the classification accuracy based on the 30-dimensional data from the test set. (5 marks)

(d) Based on what you have learned thus far, **explore** possible techniques/tricks to improve the test accuracy without resorting to deep learning. Justify your results. (10 marks)

(e) Assessment of **Codes** includes correctness of implementation, performance and readability. (10 marks)