EECS 6327 (Fall 2019) Project one

You need to work individually for this project. Submit your codes/scripts and a project report (maximum 8 pages) to Hui Jiang (hj@cse.yorku.ca) before the due date. No late submission will be accepted. No Handwriting.

Due date: Nov 3, 2019

Pattern Classification: Features and Models

In this project, you will implement several discriminative models for pattern classification. You may choose to use any programming language for your own convenience. You are ONLY allowed to use libraries for linear algebra operations, such as matrix multiplication, matrix inversion, matrix factorization, and etc. However, You are NOT allowed to use any existing machine learning or statistics toolkits or libraries or any open-source codes for this project and you will have to implement the most parts of the model learning and testing algorithms yourself as a practice of various algorithms learned in class. This is the purpose of this project. If you have any questions, please consult the instructor for what you can or can not use for this project.

You will use a popular data set, called MNIST, throughout the project. The MNIST database of handwritten digits contains 60,000 training images and 10,000 test images, and the images are 28-by-28 in size. The MNIST data set can be downloaded from http://yann.lecun.com/exdb/mnist/. In this project, for simplicity, you just use pixels as raw features for the following models.

What to do?

1. **Feature Extraction and Data Virtualization**: Select three digits of '4', '7' and '8', study the effects of various dimensionality reduction methods.
   a. Use all training images of these three digits ('4', '7' and '8') to estimate principal component analysis (PCA) projection matrices, and plot the total distortion errors of these images as a function of the used PCA dimensions (such as 2, 10, 50, 100, 200, 300).
   b. Use all training images of these three digits ('4', '7' and '8') to estimate Linear Discriminant Analysis (LDA) projection matrices, what are the maximum LDA dimensions you may use in this case? Why?
   c. Use PCA and Linear Discriminant Analysis (LDA) to project all these images into 2-D space and plot them with three different colors for data virtualization. Comparing these two linear methods with a popular nonlinear method, namely t-Distributed Stochastic Neighbor Embedding (t-***SNE)*** *(https://lvdmaaten.github.io/tsne/)*. You don't need to implement t-SNE and can directly download the t-SNE code from the website and run it on your data to compare with PCA and LDA. Based on your results, explain how these three methods differ in data virtualization.

2. **Linear Regression and Logistic Regression:** Use all training data of 10 digits to learn two 10-class classifiers using linear regression and logistic regression (or MCE version of linear classifiers), and report the best possible classification performance in the held-out test images. For logistic regression, you may choose to use any iterative optimization algorithm. Don't just call any off-the-shelf optimizers. You need to implement the optimizer yourself.

3. **Support Vector Machine:** Use all training data of 10 digits to learn two 10-class classifiers

(considering to use the one vs. one strategy) using linear support vector machine (SVM) and nonlinear SVM (with Gaussian RBF kernel), and report the best possible classification performance in the held-out test images. Don't just call any off-the-shelf optimizers. You need to implement the SVM optimizer yourself.

4. **Deep Neural Networks:** Implement a general back-propagation (BP) algorithm for feed-forward fully-connected multi-layer deep neural networks. Use all training data to learn a 10-class classifier using your own BP implementation, investigate various network structures (such as different number of layers and nodes per layer), report the best possible classification performance in the held-out test images.

What to submit?

You need to submit all of your codes written for this project. Write ONE single script for all above steps for the instructor to quickly run your code and reproduce your results. Please provide a clear instruction on how to repeat your experiments in a separate readme file. You need to submit a project report (in pdf, maximum 8 pages) to summarize what you have done in terms of algorithm development and experimental fine-tuning, please report the best settings for each classifier and also include your findings from this project.