Artificial Intelligence,
MASlab, Department of Computer Engineering,
University of Zanjan

PH #3: kNN

Evaluate the KNN classifier implemented by you on the famous MNIST data set where each example is a handwritten digit. Each example includes 28x28 grey-scale pixel values as features and a categorical class label out of 0-9. You can manually download the dataset from Dr. Yann Lecun's webpage (http://yann.lecun.com/exdb/mnist/) or automatically get it from some libraries/packages (e.g., as done in section 5.9 of http://scikitlearn.org/stable/datasets/index.html for sklearn in Python). However, you can find a simple one in the .Zip file assigned to this assignment.

You have to implement a KNN classifier with **Euclidean distance** from scratch, do not use existing class or functions (e.g., sklearn.neighbors.KNeighborsClassifier). On the original data set, please use the first 5,000 examples for training, and the last 500 examples for testing.

- Briefly describe how you implement the KNN classifier by giving the pseudocode. The pseudocode must include equations for how the distances are computed and how classification is done for each example in the test phase. Remember, this should not be a printout of your code, but a high-level outline description. Include the pseudocode in your pdf file (or .doc/.docx file). Submit the actual code as a single zip file named ZNUAI-PH3-*yourteamname*.zip IN ADDITION TO the pdf file (or .doc/.docx file).

- Plot curves for training and test errors: the training/test error (which is equal to 1.0-accuracy) vs. the value of K. Plot 11 points for the curve, using K = 1; 9; 19; 29; 39; 49; 59; 69; 79; 89; 99. Plot the error curves for training error and test error in the same figure.

- Please do not change the file name in the source codes.

- You may make use of Jupyter Notebook and write your source codes with the report in a Jupyter Notebook file in different cells.