Project 1 of FTE 4560 Basic Machine Learning Understand and Conduct Basic Classifiers

Jicong Fan, CUHK-Shenzhen, Feb. 2021

In this project, there are two datasets for classification. You need to use **both** of them to finish the project. In other words, the project is classification on two different datasets, one image dataset and one finance dataset.

1 Classification on Yale Face Database

1.1 Dataset Description

Yale Face Database[1] contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink. Here are two different versions. The size of all images is 320x243. They have been included in this project package. You need to do the following preprocessing steps:

- **Rename the images**. To make the following tasks more convenient, you'd better rename the images and use a vector to denote the labels (the images of one subject/person are in one class).
- **Rescale the images.** Since the raw images are quite big. You'd better resize every image to a smaller size, e.g. 0.2 of the original width and height or 64x48 something like this.
- **Rescale the image pixel value**. If the pixel values are in $[0, 255]$, you need to scale them into $[0, 1]$ for convenience.
- **Organize the images into matrices** (just a suggestion). For each subject, you may vectorize every image into a column vector and form a matrix of size $d \times 165$ (d depends on the rescaled size in the previous step). Now you can get 15 matrices corresponding to 15 different subjects.

[1] http://vision.ucsd.edu/content/yale-face-database

Table 1: Classification results of KNN and LDA

	KNN				LDA+KNN			
# of training image per subject	K=1	K=3	K=5	K=7	K=1	K=3	K=5	K=7
4	e.g. 0.55 ± 0.12							
8								

1.2 Task Description

You need to design all the following classifiers and compare their classification accuracy.

- KNN classifier with different k (e.g. 1,3, 5, 7).
- Linear discriminant analysis (LDA) followed by KNN, i.e. LDA+KNN.
- Softmax regression
- Two 3-layer Neural networks (input layer+1 hidden layer+output layer) with different number of hidden units.

Data splitting and result report You need to **randomly** select 4 or 8 images per subject to form a training set to learn your classifiers. And then use the remaining 7 or 3 images per subject to test the classification accuracy. Repeat the experiments for 10 times and report the mean value and standard deviation of each classifier. You will report the result like Tables 1-4. In Tables 2, 3 and 4, λ denotes the parameter for the regularization or weight decay. You need to use two different neural network (different number of units in the hidden layer).

Table 2: Classification results of Softmax

	Softmax				
# of training image per subject	$\lambda = 0$	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$
4					
8					

Table 3: Classification results of Neural Network A (# of hidden unit =?)

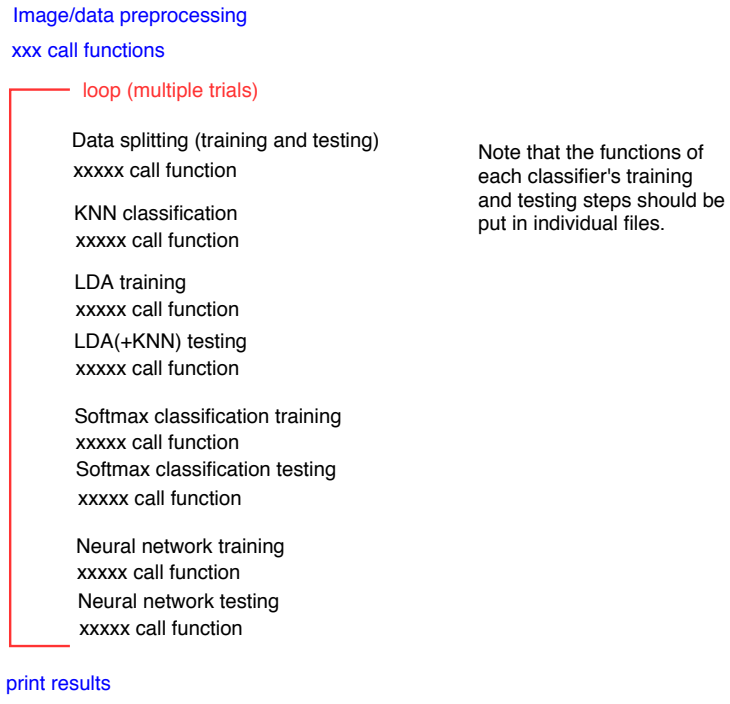
	Neural Network				
# of training image per subject	$\lambda = 0$	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$
4					
8					

Table 4: Classification results of Neural Network B (# of hidden unit =?)

	Neural Network				
# of training image per subject	$\lambda = 0$	$\lambda = 0.001$	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1$
4					
8					

Remark. You shouldn't use any machine learning toolbox such as Sklearn, Tensorflow, and Pytorch. Compute the gradient and perform gradient descent by yourself. Mind the step size. You can use the basic python tools such as Numpy. You need to submit a report and your codes. Make sure that the scripts of all methods are in a single file (main code) such that one can get the result (a single trial) of the four classifiers with "one click". One can also change the parameters e.g. K in KNN and λ in Neural Networks in the main code file. The structure of the main code looks like Figure 1.

Figure 1: Suggested flowchat for the main code



2 Classification on Polish companies bankruptcy data

2.1 Dataset description

The Polish companies bankruptcy data[2] (a subset of the original data) is about bankruptcy prediction of Polish companies. The dataset has 64 features (numerical information of the companies) and 2 classes (0 or 1). The data have already been split into two subsets, of which the larger one (949 samples) is for training and the smaller one (151 samples) is for testing. So you only need to report a single result. Do not need to perform multiple trials and report mean and standard deviation. See the CSV files. the last column in each of the files denotes the class label,"0" for "not bankrupted" and "1" for "bankrupted".

You may need to do some data preprocessing. For example, since different features have different scale, you man rescale them to zero mean and unit variance. Note that the scaling should be consistent for both training data and testing data.

2.2 Task Description

Similar to the Yale Face classification, you need to report the classification accuracy of each method in tables similar to Tables 1-4. You also need to submit the main code in a format similar to Figure 1 (without loop).

3 Conclusion and Discussion

You need to analyze the advantages or disadvantages of the four classification methods, according to your experimental results. No less than 300 words.

[2] https://archive.ics.uci.edu/ml/datasets/Polish+companies+bankruptcy+data