Machine Learning Lab 1

Get Started with Python Environment

1. In this course, we will use Python as our primary development language together with necessary third-party Python libraries. While you may already have Python installed in your laptop, you are suggested to install Anaconda for Python environment management. Anaconda is the most popular Python distribution package with automatic library management (i.e., you can install almost all the libraries especially those in data science through Anaconda). In addition, Anaconda does not interfere with your existing Python installation. You can find the installation package at https://www.anaconda.com/distribution/ . You should use Python 3.7 instead of 2.7.
2. You will use Python libraries such as scikit-learn, numpy, matplotlib for future lab exercises. You can install these through Anaconda during this lab. Or you can do it later in the feature lab when you are told to do so.
3. In general, you are free to use any IDE or text editor to program using Python, such as PyCharm or Spyder. However, you are recommended to use PyCharm if you do not have any preference over a particular IDE given its relatively full features. You can find PyCharm at https://www.jetbrains.com/pycharm/download/. You should download the community version. In addition, you are free to use IPython or Jupyter notebook for easy programming if you prefer. However, the sample code for each lab will be provided in the plain .py format for generality.

Load and display the MNIST dataset

Throughout the lab sessions, we will continuously do experiments on the MNIST digits dataset. The ultimate goal is to build a handwritten digit recognition system using Python.

1. MNIST digits recognition dataset is one of the most widely used datasets in

machine learning. It contains 60,000 training samples and 10,000 test samples. You can check out the details of the MNIST dataset from the original website http://yann.lecun.com/exdb/mnist/ .

2. We will use function `fetch_openml` from Scikit-learn library for downloading the MNIST dataset. Scikit-learn is the most popular machine learning library for python. The library contains almost all of popular machine learning techniques as well as utility functions. You can learn more about Scikit-learn from its website https://scikit-learn.org . To use MNIST dataset, you can do the following:

   ```
   from sklearn.datasets import fetch_openml
   X, y = fetch_openml('mnist_784', data_home='./', return_X_y=True)
   X = X / 255.
   ```

   X contains the feature vectors of all samples and y contains all the labels. It may take a while to run the code since it will download the data from the internet. Note that we divide X by 255 to scale the input data into the range of 0 to 1 for better numerical stability (the original data is pixel intensities, hence between 0 and 255). You should play around with these data. For example, the dimension of X, what are the values in X and y?

3. Most likely the whole MNIST dataset is too much to process for your laptop. To make running time short, we will only use the small number of images from the dataset for the subsequent lab exercises. Obtain the subset (first 1000 images) of the original data both X and y using Python.

4. Plot the first 10 samples from the dataset. We will use Matplotlib library to do so. Matplotlib library is the most popular library used for plotting figures in Python in a similar way to Matlab. You can learn more about the library from its original website https://matplotlib.org/ . You should use functions such as `subplot`, `imshow` to do so. For example, to show two images, say x1, x2, in one figure, you should do the following:

   ```
   import matplotlib.pyplot as plt
   plt.figure()
   plt.subplot(1,2,1)
   plt.imshow(x1, cmap='gray')
   plt.xticks([]), plt.yticks([])
   plt.subplot(1,2,2)
   ```

```
plt.imshow(x2, cmap='gray')

plt.xticks([]), plt.yticks([])

plt.show()
```

Note that in this example, x1 and x2 are two matrices (not a vector) and we use ‘gray’ color map for displaying grayscale images. You should make sure the image you plot is matrix instead of vector. We also use function `xticks` and `yticks` to remove unnecessary axis labels (default is on for any figure). The plot you have should look like the following: