University of Nottingham
UK | CHINA | MALAYSIA

COMP3055
Machine Learning

Explain the Solution to Lab 1

Ying Weng
2024 Autumn

Conda Config

- Use conda command easily in terminal/cmd after conda installation
  - Recommend to run command

    ```
    conda init
    ```
- Some common conda command
  - Create new conda environment

    ```
    conda create -n your_env_name python=x.x
    ```
  - List packages in the current environment

    ```
    conda list
    ```
  - Activate/deactivate environment

    ```
    conda activate/deactivate your_env_name
    ```
  - Package installation

    `conda install` or `pip install`

Key Points

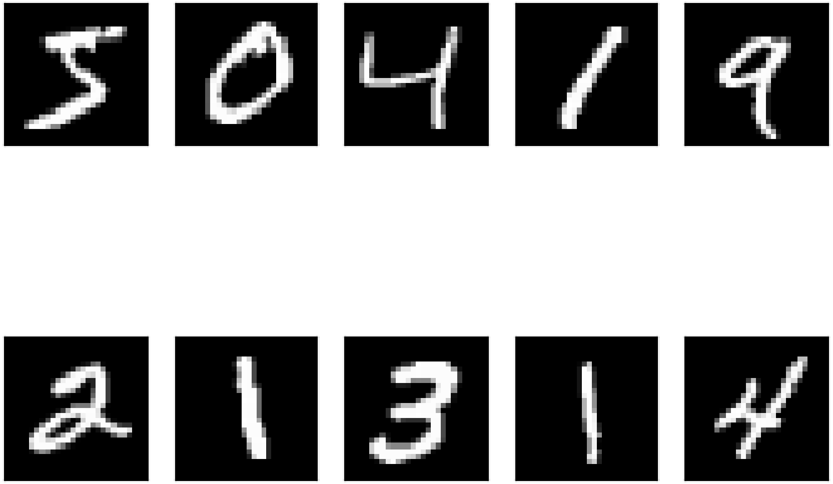
- Normalization
  - divide X by 255 to scale the input data into the range of 0 to 1 for better numerical stability (the original data is pixel intensities, hence between 0 and 255)
- Reshape
  - the image data are saved as flattened 1D arrays, reshape back to the shape of 28 * 28 to reconstruct the image

Fetch MNIST dataset

```
from sklearn.datasets import fetch_openml
import numpy as np

X, y = fetch_openml('mnist_784', data_home='./', return_X_y=True)
# 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)
X = X / 255.
# check the data type of X and y, if in the format of DataFrame, use below two lines to convert to numpy array,
otherwise, comment below two lines
X = X.values
y = y.values

# split the MNIST dataset into train set and test set
X_train, X_test = X[:60000], X[60000:]
y_train, y_test = y[:60000], y[60000:]

# save the fetched data locally
np.savez('mnist.npz', x_train=X_train, y_train=y_train, x_test=X_test, y_test=y_test, allow_pickle=True)

# load data from local file
data = np.load('mnist.npz', allow_pickle=True)
x_train, y_train, x_test, y_test = data['x_train'], data['y_train'], data['x_test'],data['y_test']
```

Plot first 10 images

```
import matplotlib.pyplot as plt
X_small = X[:1000]
y_small = y[:1000]
# display the first 10 digits in the training set
plt.figure()
for i in range(10):
   plt.subplot(2, 5, i+1)
   # the image data are saved as flattened 1D arrays, reshape back to the shape of 28 * 28 to reconstruct the image
   plt.imshow(X_small[i].reshape((28, 28)), cmap='gray')
   plt.xticks([]), plt.yticks([])
plt.show()
```



Any Questions?