ELEC946 Intelligent System Design, Spring 2020
Homework Programming Assignment 3

Instructor: Gil-Jin Jang Email: gjang@knu.ac.kr
School of Electronics Engineering, Kyungpook National University

1 Introduction

The purpose of programming assignment 3 is practicing 2-layer neural network training and checking the learned filters on MNIST. MNIST (**M**ixed **N**ational **I**nstitute of **S**tandards and **T**echnology) is a dataset of hand-written digit images.

Characteristics:

- The number of classes are 10 (0-9)
- 60,000 training images from American Census Bureau employees
- 10,000 testing images from American high school students

Format:

- Single channel (gray level)
- Sizes: 32×32 (1024 features) or 28×28 (784 features)
- Centered on center of mass
- Some examples are shown in Figure 1

In scikit-learn, one of the examples is found at: https://scikit-learn.org/stable/auto_examples/neural_networks/plot_mnist_filters.html.

This example provides how to load the database (by downloading from Internet), train the network weights using scikit-learn's **MLPClassifier** package.

```
...
from sklearn.datasets import fetch_openml
from sklearn.exceptions import ConvergenceWarning
from sklearn.neural_network import MLPClassifier

# Load data from https://www.openml.org/d/554
X, y = fetch_openml('mnist_784', version=1, return_X_y=True)
X = X / 255.

# rescale the data, use the traditional train/test split
X_train, X_test = X[:60000], X[60000:]
y_train, y_test = y[:60000], y[60000:]

mlp = MLPClassifier(hidden_layer_sizes=(50,), max_iter=10, alpha=1e-4,
                    solver='sgd', verbose=10, random_state=1,
                    learning_rate_init=.1)
...
    mlp.fit(X_train, y_train)

print("Training set score: %f" % mlp.score(X_train, y_train))
print("Test set score: %f" % mlp.score(X_test, y_test))

# plotting filters
fig, axes = plt.subplots(4, 4)
# use global min / max to ensure all weights are shown on the same scale
vmin, vmax = mlp.coefs_[0].min(), mlp.coefs_[0].max()
for coef, ax in zip(mlp.coefs_[0].T, axes.ravel()):
```

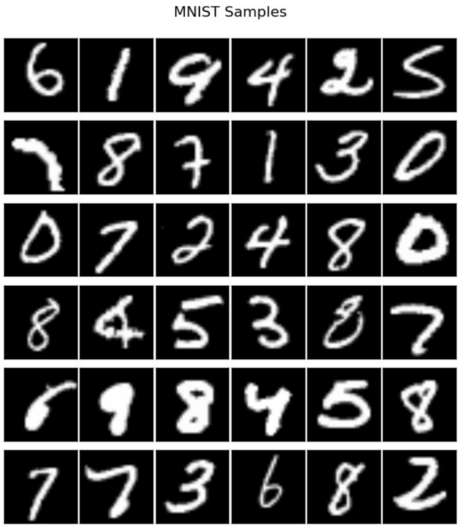

Figure 1: MNIST examples

```
    ax.matshow(coef.reshape(28, 28), cmap=plt.cm.gray, vmin=.5 * vmin,
               vmax=.5 * vmax)
    ax.set_xticks(())
    ax.set_yticks(())

plt.show()
```

The input argument hidden_layer_sizes=(50,) of the class constructor MLPClassifier() indicates that there is a single hidden layer of 50 nodes.

2 Assignment 3-1: Fashion MNIST

The first assignment is running the code for MNIST to a different dataset with similar format. Fashion MNIST is a dataset whose structure and number of samples are identical to MNIST.

Characteristics:

- The number of classes are 10 (T-shirt/top, Trouser, . . ., Bag, and Ankle boo)
- 60,000 training images, 10,000 testing images
- Image format: same as MNIST
- Some examples are shown in Figure 2

The Fashion MNIST database can be downloaded from https://github.com/zalandoresearch/fashion-mnist.

Modify plot_mnist_filters.html so that it can performan training, showing classification accuracies of training and test sets, and plotting example filters. Try to obtain your best accuracies by varying the number of hidden nodes, hidden_layer_sizes=(___,).

3 Assignment 3-2: sckit-learn's digits

In scikit-learn, a much smaller digit recognition dataset is available. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_digits.html The image is 8×8 and there are total 1,797 samples (roughly 180 samples per class).

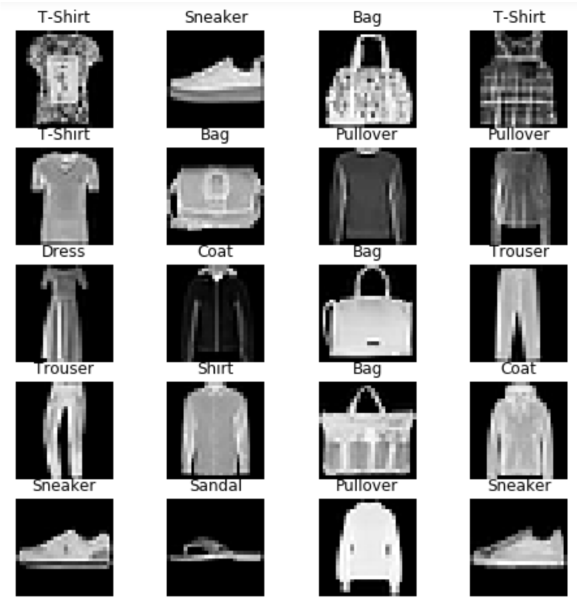

Figure 2: MNIST examples

1. Split the dataset into training (80%) and test (20%) sets
2. Modify `plot_mnist_filters.html` so that it can performan training, showing classification accuracies of training and test sets, and plotting example filters.
3. If necessary, change the parameters.

4 Submission Guidelines and Grading Scheme

Common Requirements: 1. no input arguments is needed in this assignment.

2. write or replace with ID and NAME of yours at the beginning of the code (10%).
3. specify the names of used packages in your code in the first comment block. You may install new packages (libraries) locally by python3 command `‘‘pip3 install ...’’`
4. make sure that you have installed most recent version of scikit-learn (0.23.2, as of November 22, 2020) to properly run the example. Use the command `‘‘pip3 install sklearn>=0.23.2’’`

1. Make a zip file `hw3.zip` of all the necessary `.py` files, and upload it to `lms.knu.ac.kr`

10% Basic score for submission

10% Name, ID, and other information is correct

50% Executability and correctness of the output

30% Code readability (subjective)

Due and late submission see LMS.

Late submission deduction 10% deduction per hour afer the regular submission deadline.