Modern Machine Learning
Computer Assignment #4

1. *Multiclass Classification with Neural Networks:* Using Python or MATLAB, implement the multi-class classification using neural networks.

   In the computer assignment, you main task is to study two neural networks packages, i.e., Tensorflow PyTorch and MatConvNet, based on Python and MATLAB languages. In addition, you will study how to modify the architecture of neural networks to improve the performance.

   Dataset:

2. Dataset:

   For this assignment, you will use the CIFAR10 dataset:

   The CIFAR10 dataset contains 60,000 color images in 10 classes, with 6,000 images in each class. The dataset is divided into 50,000 training images and 10,000 testing images. The classes are mutually exclusive, and there is no overlap between them. **Use 10% ONLY of the training and testing data via random sampling**

   CIFAR10 is available in TensorFlow and PyTorch data loader classes. You can use the following scripts to download the data in both libraries:

   PyTorch

```
import torch
import torchvision
import torchvision.transforms as transforms
transform = transforms.Compose(
[transforms.ToTensor(),
 transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))])

batch_size = 4

trainset = torchvision.datasets.CIFAR10(root='./data',
                train=True,download=True,
                transform=transform)
```

```
trainloader = torch.utils.data.DataLoader(trainset,
                batch_size=batch_size,shuffle=True,
                num_workers=2)

testset = torchvision.datasets.CIFAR10(root='./data',
                train=False, download=True,
                transform=transform)
testloader = torch.utils.data.DataLoader(testset,
                batch_size=batch_size,
                shuffle=False, num_workers=2)

classes = ('plane', 'car', 'bird', 'cat',
           'deer', 'dog', 'frog', 'horse',
           'ship', 'truck')
```

TensorFlow

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models

(train_images, train_labels), (test_images, test_labels)
                            = datasets.cifar10.load_data()

# Normalize pixel values to be between 0 and 1
train_images, test_images = [train_images / 255.0,
                                test_images / 255.0]
class_names = ['airplane', 'automobile', 'bird', 'cat',
               'deer','dog', 'frog', 'horse',
               'ship', 'truck']
```

MATLAB For MATLAB users, you can downlowd the dataset via this link CIFAR-10 MATLAB

Neural network structure: We will provide a basic neural network to classify the CIFAR-10 dataset. The basic neural network has one input layer, two hidden layers, and one output layer. The input layer has 1024 nodes because the input image dimension is 1024×1. The output layer has 10 nodes because there are 10 different classes.

The first hidden layer $\boldsymbol{Z}^{(1)}$ has 32 neurons (nodes)

$$\boldsymbol{Z}^{(1)} = \{g(z_1^{(1)}), g(z_1^{(1)}), \cdots, g(z_{32}^{(1)})\} \tag{1}$$

where $z_i^{(1)}$ is the ith weighted summed neuron inputs and $g(\cdot)$ denotes the sigmoid activation function.

The second hidden layer $\boldsymbol{Z}^{(2)}$ has 16 neurons

$$\boldsymbol{Z}^{(2)} = \{g(z_1^{(2)}), g(z_2^{(2)}), \cdots, g(z_{16}^{(2)})\} \tag{2}$$

where $z_i^{(2)}$ is the ith weighted summed neuron inputs and $g(\cdot)$ denotes the sigmoid activation function.

Assignment requirement:

(a) increase the number of neurons in the two hidden layers from $(32, 16)$ to $(64, 32)$, compare the training/testing error in the two cases, and explain it.

(b) replace the sigmoid function with the ReLU function, compare the training/testing error in the two cases, and explain it.

Python files: The script 'nn_mnist_example.py' defines the architecture of the basic neural network, the number of training epochs, the learning rate, etc., and implements the basic neural network. Use it as a guide for your implementation

You can find information on the TensorFlow official website. `https://www.tensorflow.org/api_docs/python/tf/nn/relu`

Submission guidelines: Your submission should include:

- A unique **zip folder**, which should include a modified version of nn_mnist_example.py.
- A pdf file showing the training/testing error given different neuron numbers and different activation functions. Explain your results.

MATLAB files:

- The script 'nn_mnist_example.m' implements the basic neural network. The script 'nn_mnist_init.m' specifies the architecture of the basic neural network, the number of training epochs, the learning rate, etc. Use it as a guide for your implementation.
- The MATLAB scripts 'nn_mnist_example.m' and 'nn_mnist_init.m' are located in the local folder
  .../MATLAB files/examples/neural networks
- Before running the MATLAB scripts, you must **install the MatCovNet library**. Please follow the instructions in the webpage `https://www.vlfeat.org/matconvnet/install/`.
- The top1 error is the percentage of the time that the classifier did not give the correct class the highest score, and the top1 training/testing error is equivalent to the standard training/testing error. The top5 error is the percentage of the time that the classifier did not include the correct class among its top 5 guesses.
- In MATLAB scripts, the top1 training/testing error is equivalent to the standard training/testing error.

Submission guidelines: Your submission should include:

- A unique **zip folder**, which should include a modified version of nn_mnist_example.m, nn_mnist_init.m, plus all necessary files to run your code.
- A pdf file showing the training/testing error given different neuron numbers and different activation functions. Explain your results.