DSE 316/616: Deep Learning

Assignment 1

Submission Policy and Requirements :

- Any kind of plagiarism is not accepted. We will strictly follow institute policies for plagiarism*
- Recommended programming languages: Python + PyTorch / Tensorflow (use of Keras leads to zero points)
- Submission should include a working code for the questions asked and a report to show the analysis of results in each of the parts.

Assessment criterion:

The assessment will be done on the basis of the following components:

- Working codes
- Analysis and clarity of results (drawing comparisons across different parts) and clarity of the report
- Understanding the theoretical concepts and the choice of hyperparameters

Guidelines for Submission:

- A single report(pdf) for all questions
- Mention all the relevant results, and comparisons as asked or wherever required for a better understand- ing of the results
- A single zip file containing the report, codes, and readme, if required
- Name the file with the roll number, for example $rollnumbera1.zip$

Question 1 (50 points)

In this assignment, you will be implementing a 3 layers deep feed-forward neural network from scratch For multiclass classification on SVHN Dataset. In the dataset, every image is 3x32x32, so first, you have to flatten it. Detailed instructions are given below.: (**You are not allowed to use any library, e.g., PyTorch, Tensorflow. Only Numpy can be used**.)

A. Implement 3 layer deep feed-forward neural with a sigmoid activation function. [10 points]
B. Implement Xavier weight initialization scheme [5 points]
C. Implement cross-entropy loss [5 points]
D. Implement backpropagation [15]
E. Implement SGD and RMSProp optimizer to update the weights [10 points]
F. Find the hyperparameter of the network, such as the learning rate and number of neurons in hidden layers. [3]
G. Report the accuracy, F1-score, Test-Train loss, etc. [2]

```
# to convert .mat format into numpy array use the given code

import scipy.io

data=scipy.io.loadmat(“Download_file.mat”)

X=data[‘X’]

Y=dat[‘Y’]
```

Question 2 (50 points)

Construct following CNN architecture **[15+ 5(Report)]**. Use the Olivetti faces dataset for all the analysis’ (you can use the deep learning libraries)

A. Conv-Pool-Conv-Pool-Conv-Pool-FC
B. Conv-Conv-Pool-Conv-Conv-Pool-FC
C. Conv-Pool-Conv-Pool-Conv-Pool-FC-FC

FC refers to Fully Connected Layer here, provide with the following analysis:

A. How does changing the network size change the accuracy? [10 points]
B. Experiment with different sizes of pooling and do a detailed analysis of pooling size on the network.[10 points]
C. How the presence of one or more fully connected layers changes the accuracy. [10 points]

```
# Download Olivetti faces dataset
from sklearn.datasets import fetch_olivetti_faces
```

```
olivetti = fetch_olivetti_faces()
x = olivetti.images
y = olivetti.target
```

Deliverables:

- A Google Colab file containing your implementation, visualizations, and explanations containing conclusions. Google Colab file Name with your name_roll_no. Additionally, please submit a Report file in PDF format containing what you did in the code and your conclusions.
  OR
- You can write your code in a Python file named main.py. Additionally, please submit a Report file in pdf format containing what you did in the code and your conclusions.