

CEP Math for AI/ML

Lab - 2: Feed Forward Networks and Backpropagation

We will implement a fully connected Feed Forward Neural Network and train on a dataset

Dataset - [MNIST](#) - Predicts digit from an input matrix

Input: 28*28 pixels array which is a greyscale image

Output: Digit formed by input image

Task 0:

Understand how input and output samples are.

Observe text files in the “visualize” folder. Input is a 28*28 matrix with each element between 0 and 255. Output is a digit between 0 and 9

Now, observe images in the “visualize” folder. “visualize_1.png” is the first data point and similarly “visualize_2.png” is the second data point

The input image is formed by plotting a greyscale image with the 28*28 input pixel matrix

We will implement FFNN consisting of 1 input layer, 1 hidden layer, and 1 output layer, and train the FFNN forward and backward pass functions using backpropagation

The input layer has 784 (28*28) neurons. The hidden layer has 12 neurons, and the output layer has 10 neurons. (as there are 10 digits i.e. 0-9 possible as output)

Activation function for the hidden and output layer is Sigmoid function

The loss function used is Mean Squared Error(MSE)

You have been provided with

- a partially implemented code file **train.py**
- test.py for predicting output for test data
- “data” folder
 - Training data: mnist_train.pkl
 - Testing data: mnist_test.pkl

Task 1:

Complete forward functions of **FCLayer** and **SigmoidLayer**

Your task is to implement the missing parts of train.py with the help of comments provided in the code itself. The components to be implemented by you are enclosed between comments “## TODO” and “## END TODO”

Instructions to run the code:

```
python3 train.py
```

```
python3 test.py
```

Note:

- train.py trains the network with 2000 datapoints, 40 epochs, and 0.01 as learning rate
- Test.py predicts the output for 100 test samples and prints accuracy
- Understand the outputs written to the terminal
 - While running train.py, observe how the training error is changing over each epoch. Also, observe and understand the figure plotted. It is training loss vs testing loss with an increase in epochs
 - While running the test.py, note the accuracy of the model, that is written to the terminal

Task 2:

In task 1, the model is trained with 12 neurons in the hidden layer.

Now, modify the number of hidden neurons of the model and observe how the training error and accuracy of test data change

Instructions to run the code:

```
python3 train.py --num {num_hidden}
```

```
python3 test.py
```

Where num_hidden is the number of hidden neurons in the hidden layer