A REPORT

On

ASSIGNMENT 3(Machine Learning)

Shripad pate(M23MAC007)

Data and Computational Sciences (DCS)
Department of Mathematics
Indian Institute of Technology, Jodhpur

November 2023

1.0 Introduction

In this assignment, a neural network was implemented from scratch using NumPy and Pandas for the given dataset. The primary goals were to create a multi-layer neural network with specific architecture, train it using gradient descent, evaluate its performance, and analyse its parameters. **2.0 Data Analysis**

- Dataset contains a total of 70,000 grayscale images, which are divided into two primary subsets:

- Training Set: This set comprises images , which are typically used for training machine learning models. Each image is a 28x28 pixel representing 10 different object categories , making it a total of ten different classes.

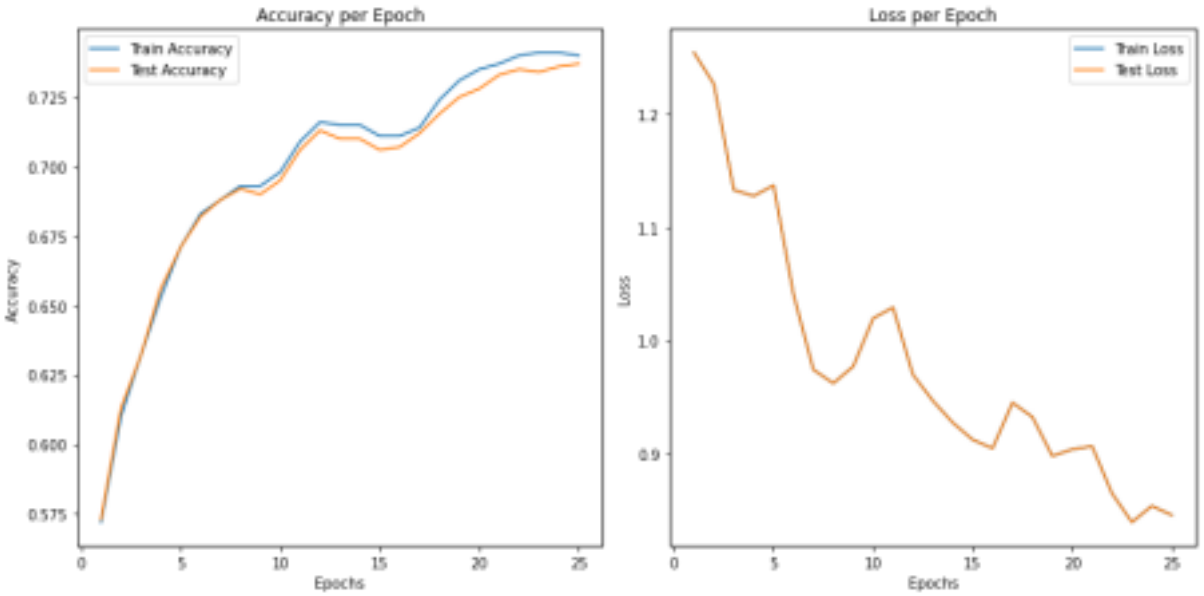
- Test Set: The remaining images are reserved for testing the performance of the trained models. These images are not used during the training phase, and their labels are used to evaluate the model's accuracy and generalisation.

3.0 Model Building

Input layer: Set to the size of the dimensions of the dataset.
Output layer: Set to the number of classes in the dataset.
Hidden layers:
Hidden layer 1: 128 neurons with a sigmoid activation function.
Hidden layer 2: 64 neurons with a sigmoid activation function.
Hidden layer 3: 32 neurons with a sigmoid activation function.

Subtask a) Use Train-test splits as 70:30

The neural network was trained using a batch size equal to the year of admission 23, a learning rate of 0.0034, and the cross-entropy loss function. Training was performed for 25 epochs



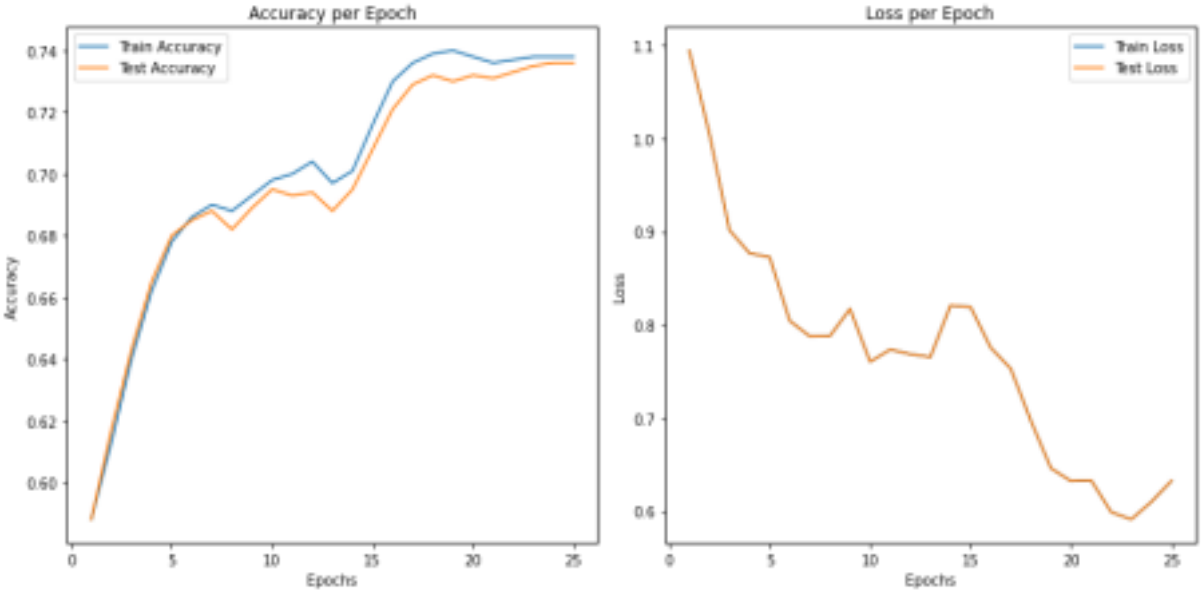
After training the model we got training accuracy is 0.74 and testing accuracy is 0.737 also loss for last epochs is 0.886 there is still possibility to increase the accuracy by applying some optimisation method.from first graph we can see that the accuracy is linearly increase then its fluctuate after it reach 70% and reach 74% with in 25 epochs accuracy also change by changing
learning rate after some experiment we got 0.0034 is one of the good learning rate.loss is highly fluctuate.

Confusion Matrix:
[[3603 54 110 409 82 4 454 0 110 0]

[5 4488 79 221 71 0 16 0 4 1]
[47 8 3315 27 1081 1 394 0 69 2]
[230 69 61 3948 350 0 238 0 23 0]
[7 12 757 125 3605 0 373 0 63 0]
[4 4 9 7 16 3562 2 753 187 421]
[794 16 918 248 1682 1 1018 0 185 0]
[0 0 0 0 0 380 0 3900 18 575]
[27 6 28 34 86 78 132 22 4452 4]
[0 0 4 3 0 282 0 274 3 4349]]

Subtask b) Use Train-test splits as 70:20

The neural network was trained using a batch size equal to the year of admission 23, a learning rate of 0.0034, and the cross-entropy loss function. Training was performed for 25 epochs



After training the model we got training accuracy is 0.738 and testing accuracy is 0.736 also loss for last epochs is 0.633 there is still possibility to increase the accuracy by applying some optimisation method.from first graph we can see that the accuracy is linearly increase then its fluctuate but after it reach 70% they increase linearly and reach 73.8% with in 25 epochs accuracy also change by changing learning rate after some experiment we got 0.0034 is one of the good learning rate.loss is linearly decrease first and then not change with in 5 to 10 epochs and then decrease

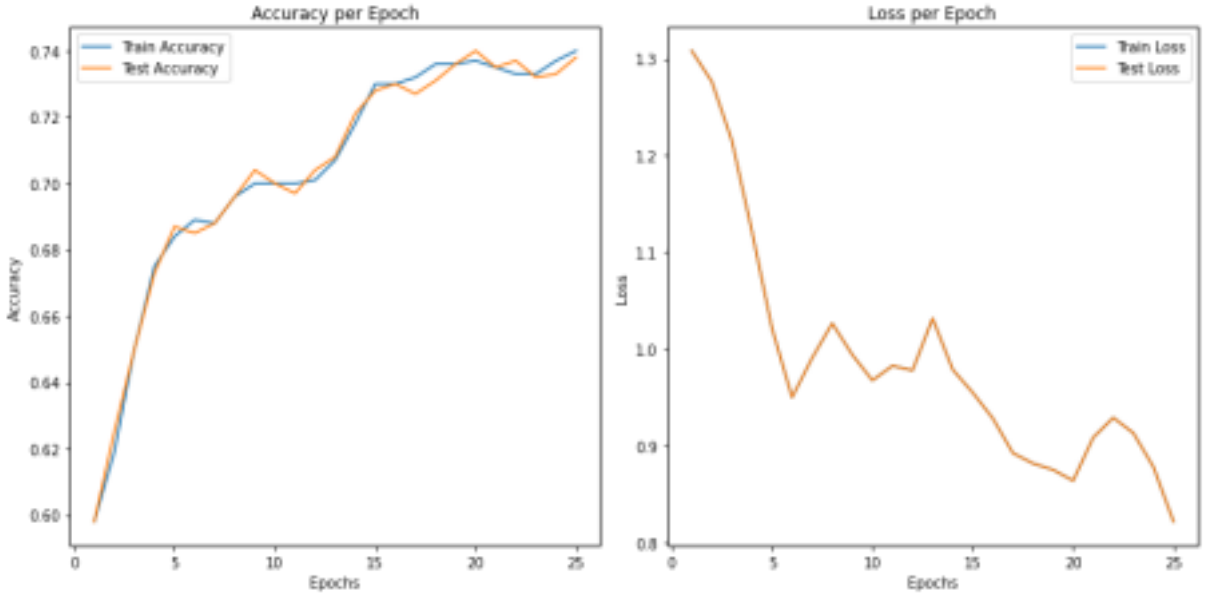
Confusion Matrix:

[[4300 78 138 499 56 5 439 1 75 0]
 [8 5175 97 252 60 0 9 0 6 0]
 [75 8 3546 42 1303 2 556 0 55 1]
 [320 97 65 4503 407 0 210 0 15 0]
 [11 26 607 166 4164 0 586 0 63 0]
 [4 21 13 8 7 3931 0 990 183 477]
 [959 25 887 324 1996 1 1179 0 170 1]

[0 1 0 0 0 478 0 4590 29 483]
[38 12 25 46 57 59 227 18 5107 2]
[1 0 4 4 0 313 0 445 3 4856]]

Subtask c) Use Train-test splits as 70:10

The neural network was trained using a batch size equal to the year of admission 23, a learning rate of 0.0034, and the cross-entropy loss function. Training was performed for 25 epochs



After training the model we got training accuracy is 0.74 and testing accuracy is 0.738 also loss for last epochs is 0.821 there is still possibility to increase the accuracy by applying some optimisation method.from first graph we can see that the accuracy is linearly increase then its fluctuate after it reach 70% and reach 74% with in 25 epochs accuracy also change by changing learning rate after some experiment we got 0.0034 is one of the good learning rate.loss is linearly decrease for first five epochs and then highly fluctuate.

Confusion Matrix:

[[4724 61 146 648 80 3 543 1 112 0]
[11 5832 108 261 58 0 11 0 7 1]
[72 8 4058 45 1511 2 496 0 99 0]
[431 125 61 5067 448 1 135 0 35 0]
[29 14 670 168 4967 0 408 0 76 0]
[7 26 8 6 3 4272 0 980 233 767]
[1061 19 1025 417 2230 2 1319 0 217 1]
[0 6 0 0 0 511 0 4904 35 828]
[55 9 46 51 92 37 173 21 5802 4]
[2 0 4 4 0 369 0 269 5 5647]]

Total trainable parameters:

Weight: in this neural network total weight is (128*784 + 64*128 + 32*64 + 10* 32) which is 10912

Bias: in this neural network total bias is (128+64+32+10) which is 234

Total trainable parameter is 11146

Total non-trainable parameters:
Learning rate
No of epochs
Batch size

- **For task we are using following library**
  1.numpy,pandas:for reading the dataset and mathematical functions.
  2.matplotlib:for visualising the graph of loss and accuracy in different train test splits.
  3.sklearn: for train test split and one hot encoding.