# Assignment 1: Optimizers

Bismay Parija | 20CS30067

CS60021: Scalable Data Mining

27.08.2023

## Results

### Part A: Using Unnormalized Data

#### *1. SGD optimizer*

#### i. Learning rate = 0.1, Number of epochs = 200

Epoch [20/200], Train Loss: 1515610439680.0000, Test Loss: 969988177920.0000

Epoch [40/200], Train Loss: 201459328.0000, Test Loss: 128904304.0000

Epoch [60/200], Train Loss: 26865.3965, Test Loss: 16870.7559

Epoch [80/200], Train Loss: 90.4329, Test Loss: 73.3742

Epoch [100/200], Train Loss: 86.8739, Test Loss: 75.0002

Epoch [120/200], Train Loss: 86.8734, Test Loss: 75.0449

Epoch [140/200], Train Loss: 86.8734, Test Loss: 75.0454

Epoch [160/200], Train Loss: 86.8734, Test Loss: 75.0454

Epoch [180/200], Train Loss: 86.8734, Test Loss: 75.0454

Epoch [200/200], Train Loss: 86.8734, Test Loss: 75.0454

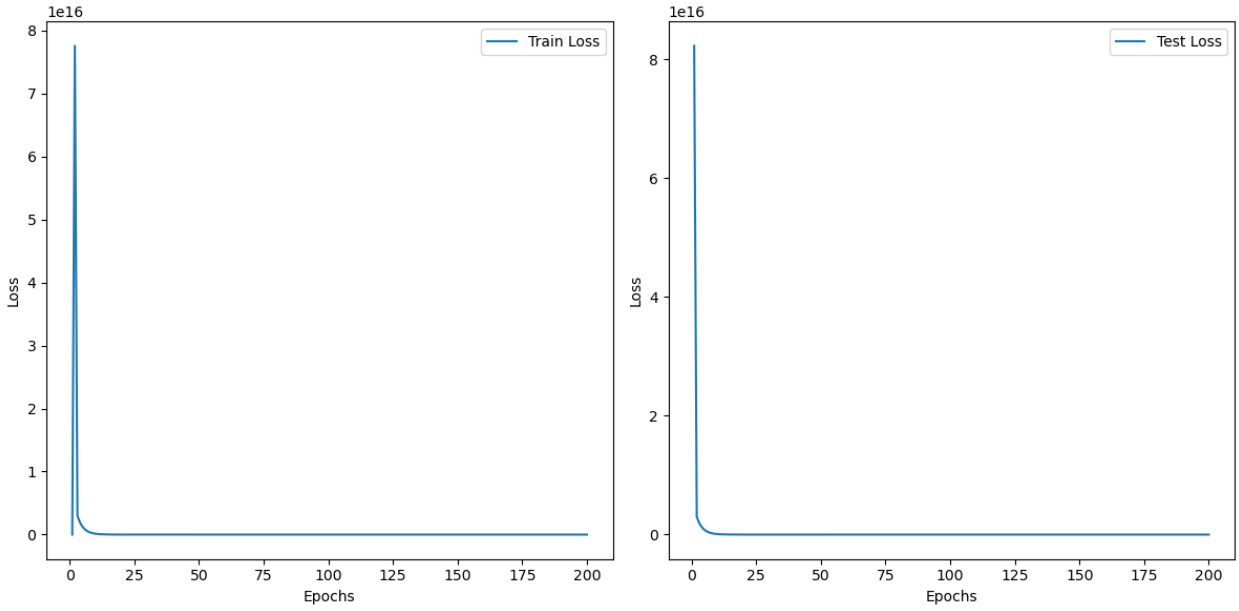


*Figure 1: Train Loss and Test Loss vs Epochs for SGD optimizer for learning rate = 0.1*

**ii. Learning rate = 0.01, Number of epochs = 1000**

Epoch [100/1000], Train Loss: 9833184.0000, Test Loss: 9451822.0000

Epoch [200/1000], Train Loss: 173030.9531, Test Loss: 167236.9219

Epoch [300/1000], Train Loss: 3128.6008, Test Loss: 3137.7451

Epoch [400/1000], Train Loss: 140.3712, Test Loss: 145.1804

Epoch [500/1000], Train Loss: 87.8143, Test Loss: 78.4365

Epoch [600/1000], Train Loss: 86.8900, Test Loss: 75.3912

Epoch [700/1000], Train Loss: 86.8737, Test Loss: 75.0895

Epoch [800/1000], Train Loss: 86.8734, Test Loss: 75.0512

Epoch [900/1000], Train Loss: 86.8734, Test Loss: 75.0462

Epoch [1000/1000], Train Loss: 86.8734, Test Loss: 75.0456

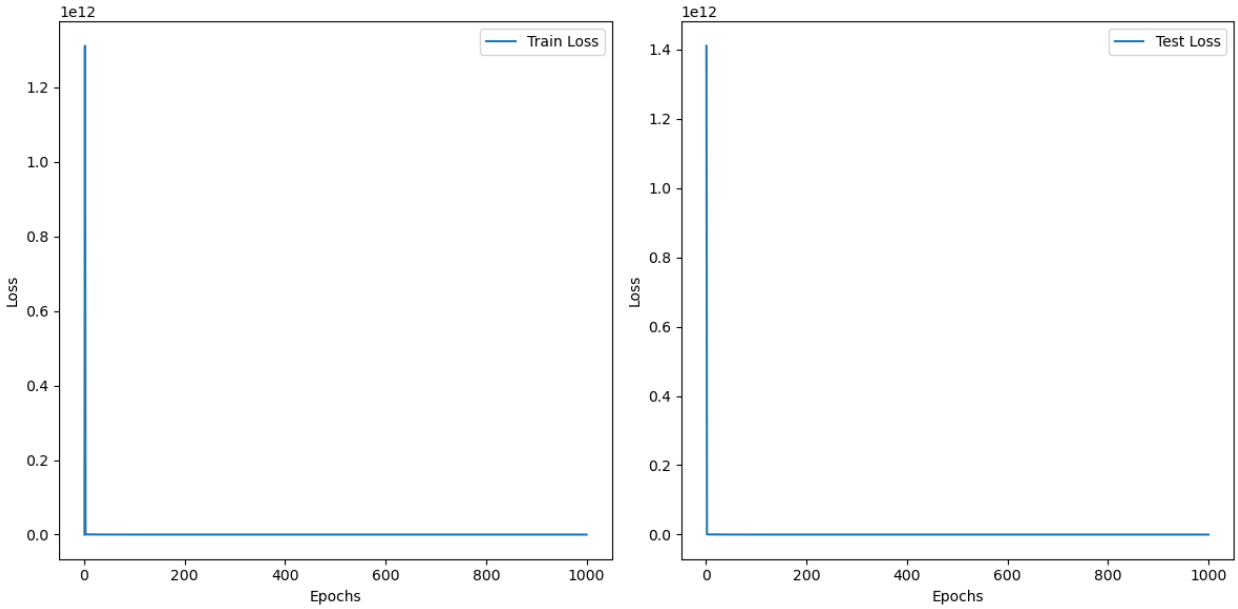


*Figure 2: Train Loss and Test Loss vs Epochs for SGD optimizer for learning rate = 0.01*

**iii. Learning rate = 0.001, Number of epochs = 5000**

Epoch [500/5000], Train Loss: 948.8563, Test Loss: 856.9155

Epoch [1000/5000], Train Loss: 203.2966, Test Loss: 162.8269

Epoch [1500/5000], Train Loss: 102.5980, Test Loss: 80.3520

Epoch [2000/5000], Train Loss: 88.9972, Test Loss: 73.3551

Epoch [2500/5000], Train Loss: 87.1603, Test Loss: 73.9325

Epoch [3000/5000], Train Loss: 86.9121, Test Loss: 74.5700

Epoch [3500/5000], Train Loss: 86.8786, Test Loss: 74.8617

Epoch [4000/5000], Train Loss: 86.8741, Test Loss: 74.9767

Epoch [4500/5000], Train Loss: 86.8735, Test Loss: 75.0200

Epoch [5000/5000], Train Loss: 86.8734, Test Loss: 75.0360

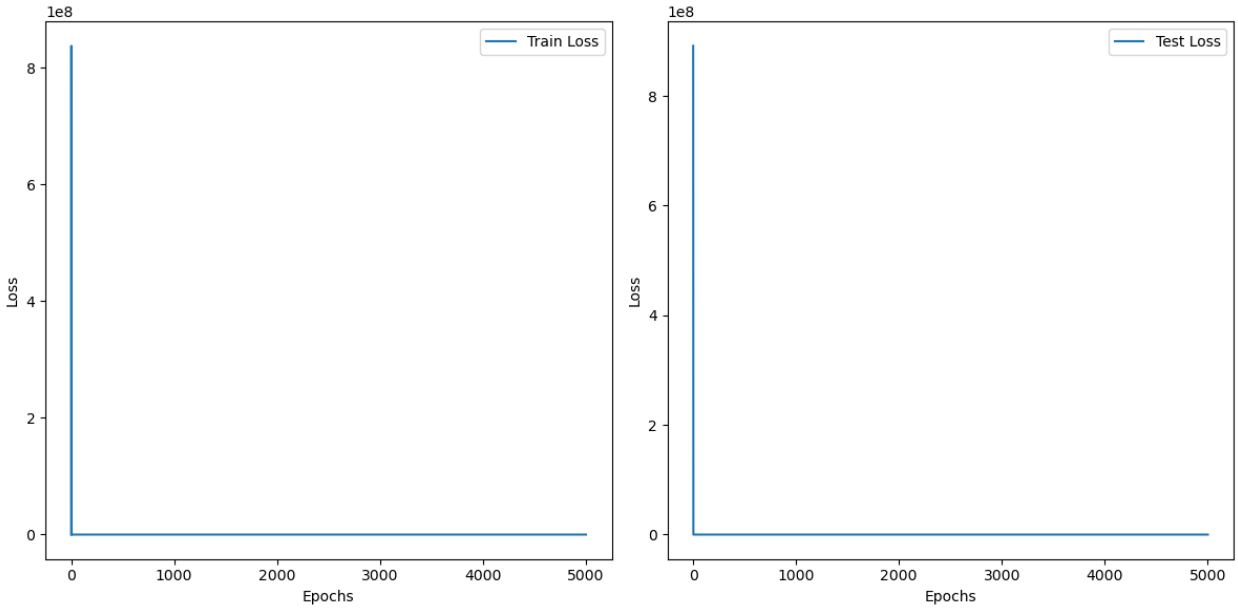


*Figure 3: Train Loss and Test Loss vs Epochs for SGD optimizer for learning rate = 0.001*

***2. Nesterov momentum optimizer***

**i. Learning rate = 0.1, Momentum = 0.1, Number of epochs = 200**

Epoch [20/200], Train Loss: 30158664761344.0000, Test Loss: 18209807794176.0000

Epoch [40/200], Train Loss: 1251068800.0000, Test Loss: 755324032.0000

Epoch [60/200], Train Loss: 51984.7812, Test Loss: 30947.8359

Epoch [80/200], Train Loss: 89.0263, Test Loss: 73.3621

Epoch [100/200], Train Loss: 86.8735, Test Loss: 75.0263

Epoch [120/200], Train Loss: 86.8734, Test Loss: 75.0453

Epoch [140/200], Train Loss: 86.8734, Test Loss: 75.0454

Epoch [160/200], Train Loss: 86.8734, Test Loss: 75.0454

Epoch [180/200], Train Loss: 86.8734, Test Loss: 75.0454

Epoch [200/200], Train Loss: 86.8734, Test Loss: 75.0454

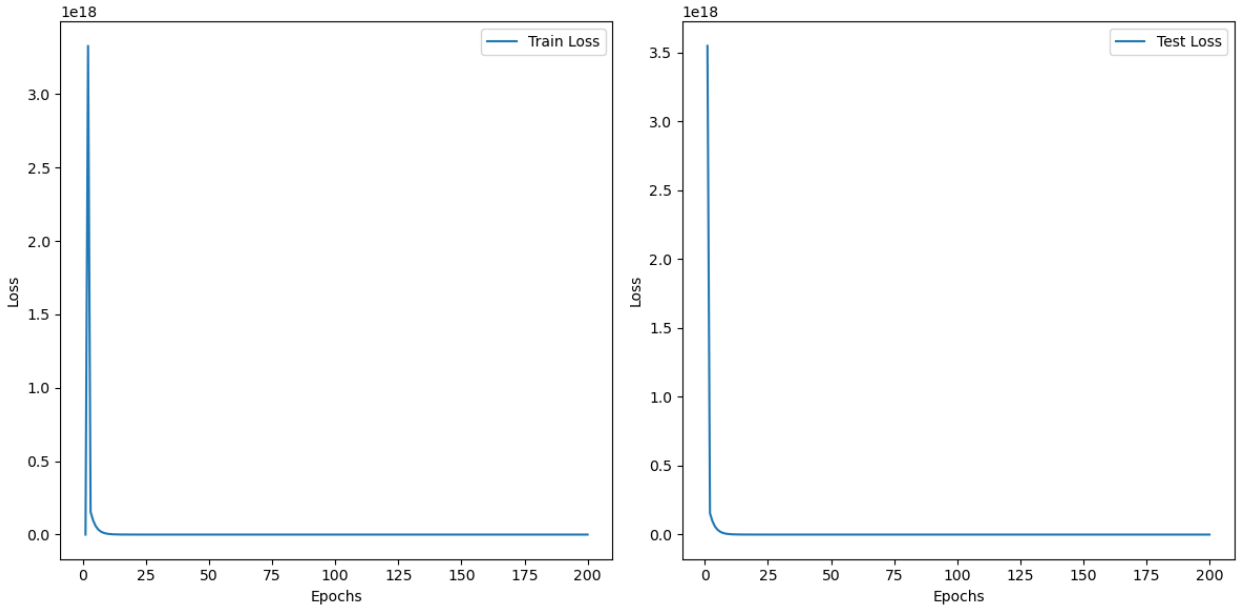


*Figure 4: Train Loss and Test Loss vs Epochs for Nesterov momentum optimizer for learning rate = 0.1*

**ii. Learning rate = 0.01, Momentum = 0.1, Number of epochs = 1000**

Epoch [100/1000], Train Loss: 110464544.0000, Test Loss: 105581456.0000

Epoch [200/1000], Train Loss: 1232497.8750, Test Loss: 1175467.0000

Epoch [300/1000], Train Loss: 13836.4082, Test Loss: 12920.1299

Epoch [400/1000], Train Loss: 240.2717, Test Loss: 190.0128

Epoch [500/1000], Train Loss: 88.5848, Test Loss: 73.3346

Epoch [600/1000], Train Loss: 86.8925, Test Loss: 74.7102

Epoch [700/1000], Train Loss: 86.8736, Test Loss: 75.0083

Epoch [800/1000], Train Loss: 86.8734, Test Loss: 75.0415

Epoch [900/1000], Train Loss: 86.8734, Test Loss: 75.0450

Epoch [1000/1000], Train Loss: 86.8734, Test Loss: 75.0453

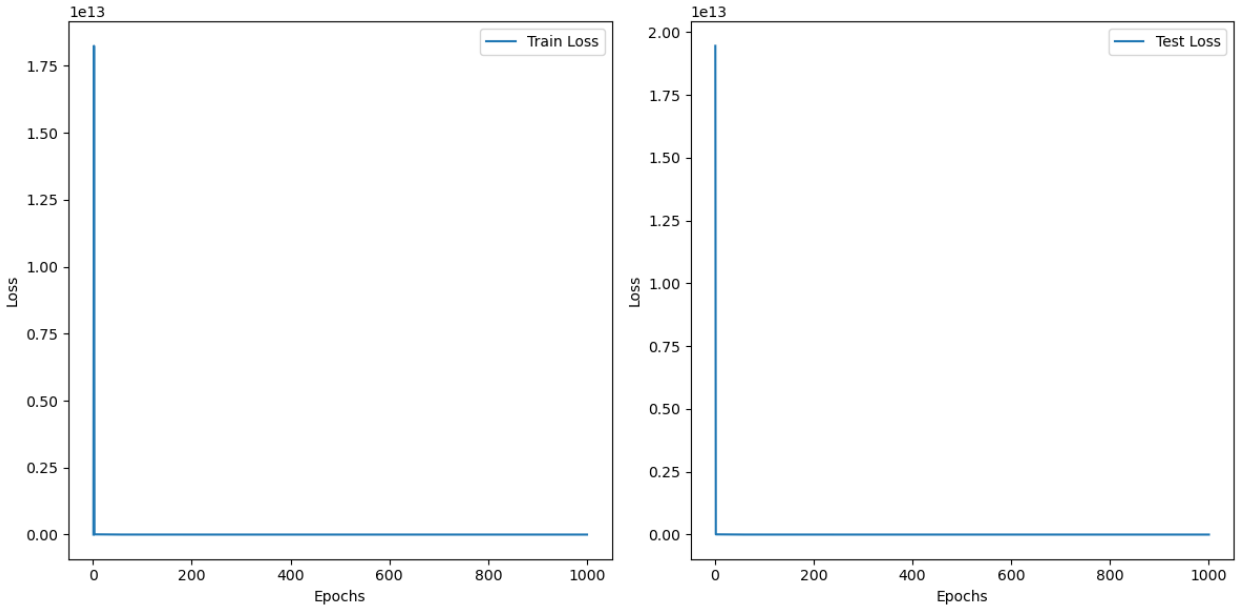


*Figure 5: Train Loss and Test Loss vs Epochs for Nesterov momentum optimizer for learning rate = 0.01*

**iii. Learning rate = 0.001, Momentum = 0.1, Number of epochs = 5000**

Epoch [500/5000], Train Loss: 6836.4082, Test Loss: 6580.1245

Epoch [1000/5000], Train Loss: 816.4596, Test Loss: 730.8730

Epoch [1500/5000], Train Loss: 165.7376, Test Loss: 130.3742

Epoch [2000/5000], Train Loss: 95.3982, Test Loss: 75.9096

Epoch [2500/5000], Train Loss: 87.7949, Test Loss: 73.4566

Epoch [3000/5000], Train Loss: 86.9730, Test Loss: 74.3206

Epoch [3500/5000], Train Loss: 86.8842, Test Loss: 74.7852

Epoch [4000/5000], Train Loss: 86.8746, Test Loss: 74.9575

Epoch [4500/5000], Train Loss: 86.8735, Test Loss: 75.0163

Epoch [5000/5000], Train Loss: 86.8734, Test Loss: 75.0358

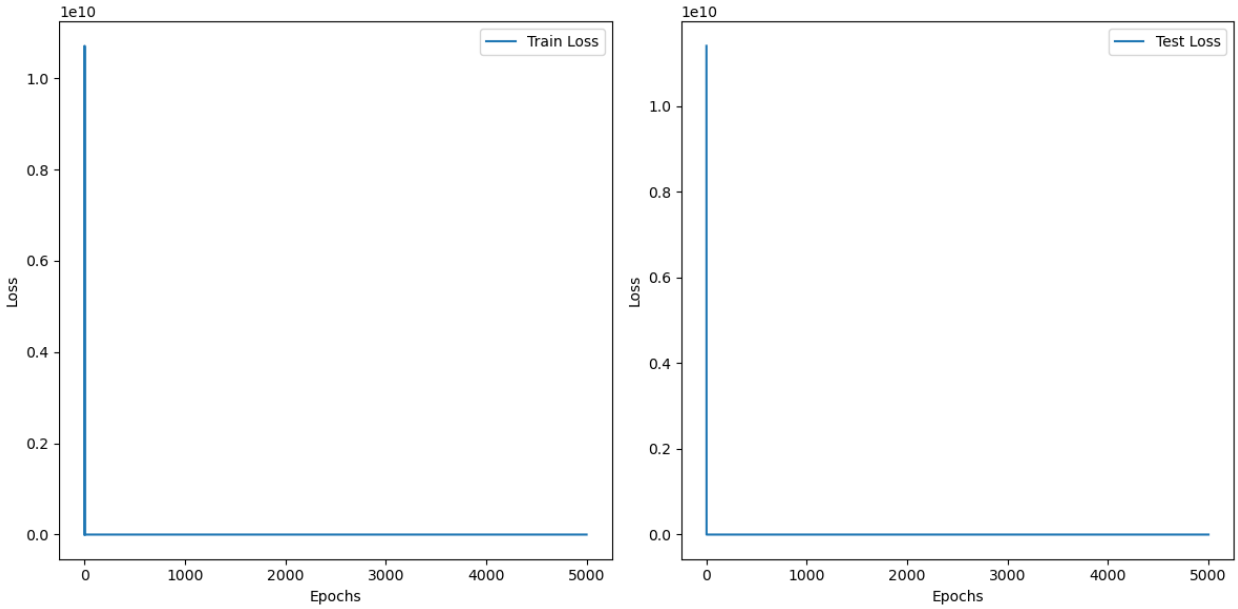


*Figure 6: Train Loss and Test Loss vs Epochs for Nesterov momentum optimizer for learning rate = 0.001*

***3. Adadelta optimizer***

**i. Learning rate = 0.1, Number of epochs = 100000**

Epoch [10000/100000], Train Loss: 15.8539, Test Loss: 19.2503

Epoch [20000/100000], Train Loss: 13.5892, Test Loss: 16.1444

Epoch [30000/100000], Train Loss: 12.3634, Test Loss: 15.1192

Epoch [40000/100000], Train Loss: 11.8195, Test Loss: 14.7382

Epoch [50000/100000], Train Loss: 10.9919, Test Loss: 14.3160

Epoch [60000/100000], Train Loss: 9.9235, Test Loss: 14.4185

Epoch [70000/100000], Train Loss: 9.2574, Test Loss: 13.5816

Epoch [80000/100000], Train Loss: 8.9372, Test Loss: 13.4257

Epoch [90000/100000], Train Loss: 8.8622, Test Loss: 13.7819

Epoch [100000/100000], Train Loss: 8.3935, Test Loss: 13.5737

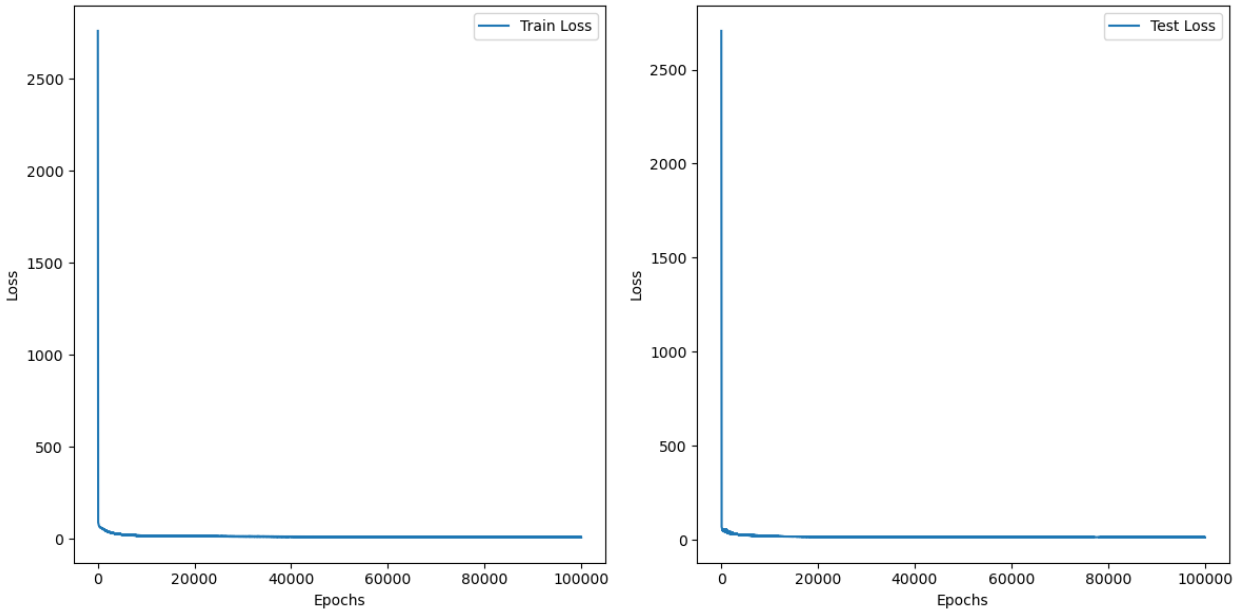


*Figure 7: Train Loss and Test Loss vs Epochs for Adadelta optimizer for learning rate = 0.1*

**ii. Learning rate = 0.01, Number of epochs = 100000**

Epoch [10000/100000], Train Loss: 34.4120, Test Loss: 33.5128

Epoch [20000/100000], Train Loss: 27.1209, Test Loss: 26.0476

Epoch [30000/100000], Train Loss: 23.9313, Test Loss: 23.8412

Epoch [40000/100000], Train Loss: 19.4986, Test Loss: 20.6327

Epoch [50000/100000], Train Loss: 17.4367, Test Loss: 20.0840

Epoch [60000/100000], Train Loss: 15.6461, Test Loss: 19.3464

Epoch [70000/100000], Train Loss: 14.8767, Test Loss: 18.5857

Epoch [80000/100000], Train Loss: 14.0184, Test Loss: 18.1322

Epoch [90000/100000], Train Loss: 12.5501, Test Loss: 17.1246

Epoch [100000/100000], Train Loss: 12.1516, Test Loss: 16.4399

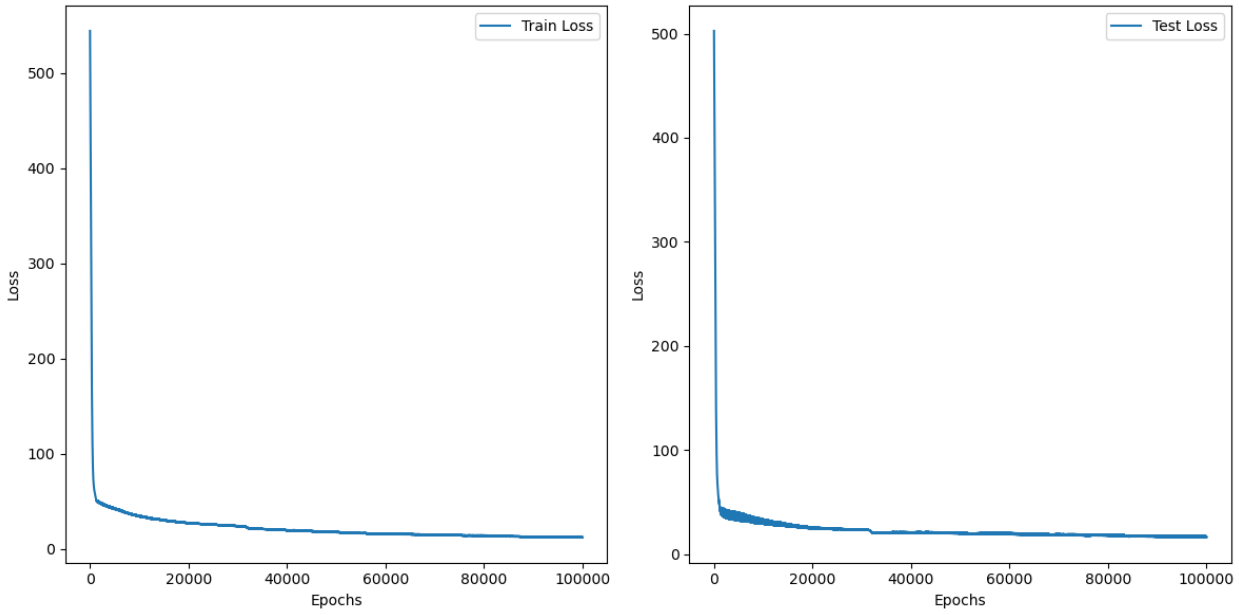


*Figure 8: Train Loss and Test Loss vs Epochs for Adadelta optimizer for learning rate = 0.01*

**iii. Learning rate = 0.001, Number of epochs = 100000**

Epoch [10000/100000], Train Loss: 36.3606, Test Loss: 28.2926

Epoch [20000/100000], Train Loss: 25.9285, Test Loss: 23.0960

Epoch [30000/100000], Train Loss: 22.4214, Test Loss: 21.4985

Epoch [40000/100000], Train Loss: 21.8684, Test Loss: 21.1138

Epoch [50000/100000], Train Loss: 21.5218, Test Loss: 20.9842

Epoch [60000/100000], Train Loss: 21.3529, Test Loss: 21.1223

Epoch [70000/100000], Train Loss: 21.3445, Test Loss: 21.4302

Epoch [80000/100000], Train Loss: 21.2437, Test Loss: 21.5647

Epoch [90000/100000], Train Loss: 21.0606, Test Loss: 21.5467

Epoch [100000/100000], Train Loss: 20.9413, Test Loss: 21.6304

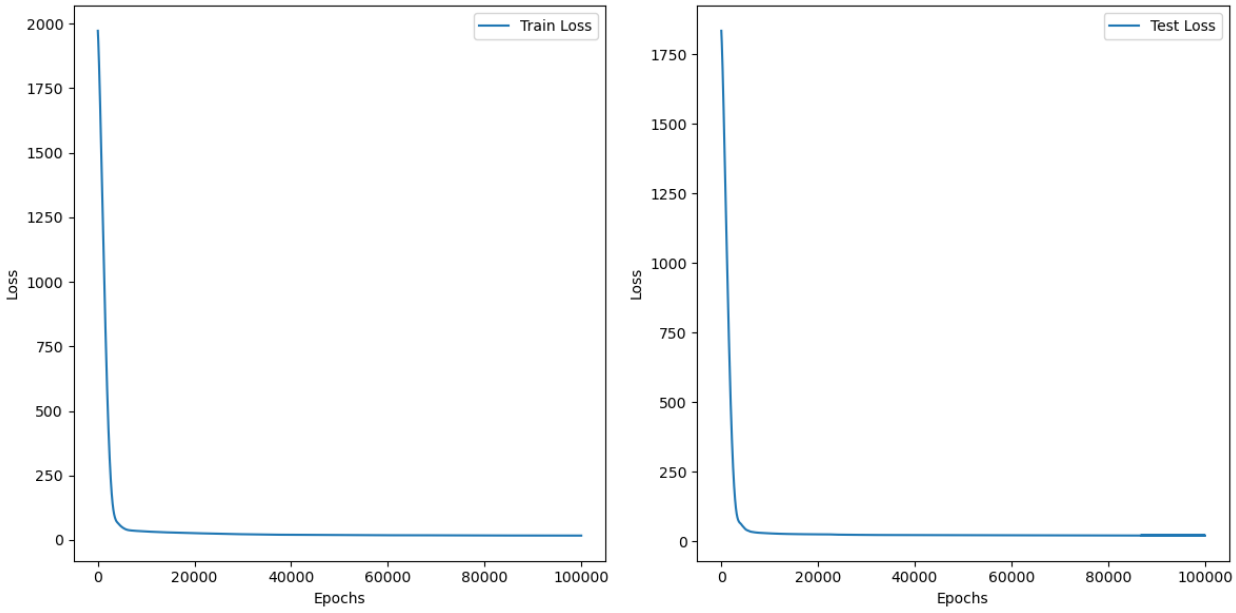


*Figure 9: Train Loss and Test Loss vs Epochs for Adadelta optimizer for learning rate = 0.001*

**Part A: Using Normalized Data**

***1. SGD optimizer***

**i. Learning rate = 0.1, Number of epochs = 200**

Epoch [20/200], Train Loss: nan, Test Loss: nan

Epoch [40/200], Train Loss: nan, Test Loss: nan

Epoch [60/200], Train Loss: nan, Test Loss: nan

Epoch [80/200], Train Loss: nan, Test Loss: nan

Epoch [100/200], Train Loss: nan, Test Loss: nan

Epoch [120/200], Train Loss: nan, Test Loss: nan

Epoch [140/200], Train Loss: nan, Test Loss: nan

Epoch [160/200], Train Loss: nan, Test Loss: nan

Epoch [180/200], Train Loss: nan, Test Loss: nan

Epoch [200/200], Train Loss: nan, Test Loss: nan

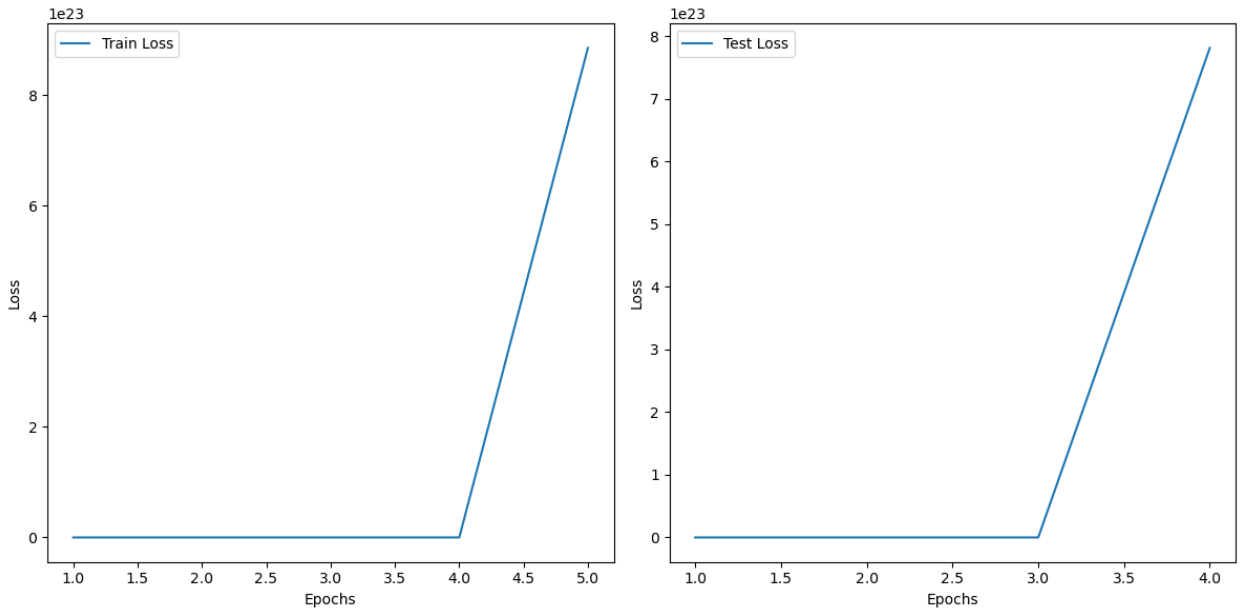


*Figure 10: Train Loss and Test Loss vs Epochs for SGD optimizer for learning rate = 0.1*

**ii. Learning rate = 0.01, Number of epochs = 500**

Epoch [50/500], Train Loss: 15.8181, Test Loss: 17.1728

Epoch [100/500], Train Loss: 11.7644, Test Loss: 13.3197

Epoch [150/500], Train Loss: 10.4002, Test Loss: 11.8563

Epoch [200/500], Train Loss: 9.6120, Test Loss: 11.2867

Epoch [250/500], Train Loss: 9.0852, Test Loss: 10.9898

Epoch [300/500], Train Loss: 8.6650, Test Loss: 10.8881

Epoch [350/500], Train Loss: 8.2334, Test Loss: 10.7409

Epoch [400/500], Train Loss: 7.7748, Test Loss: 10.6539

Epoch [450/500], Train Loss: 7.4672, Test Loss: 10.7397

Epoch [500/500], Train Loss: 7.1547, Test Loss: 10.7200

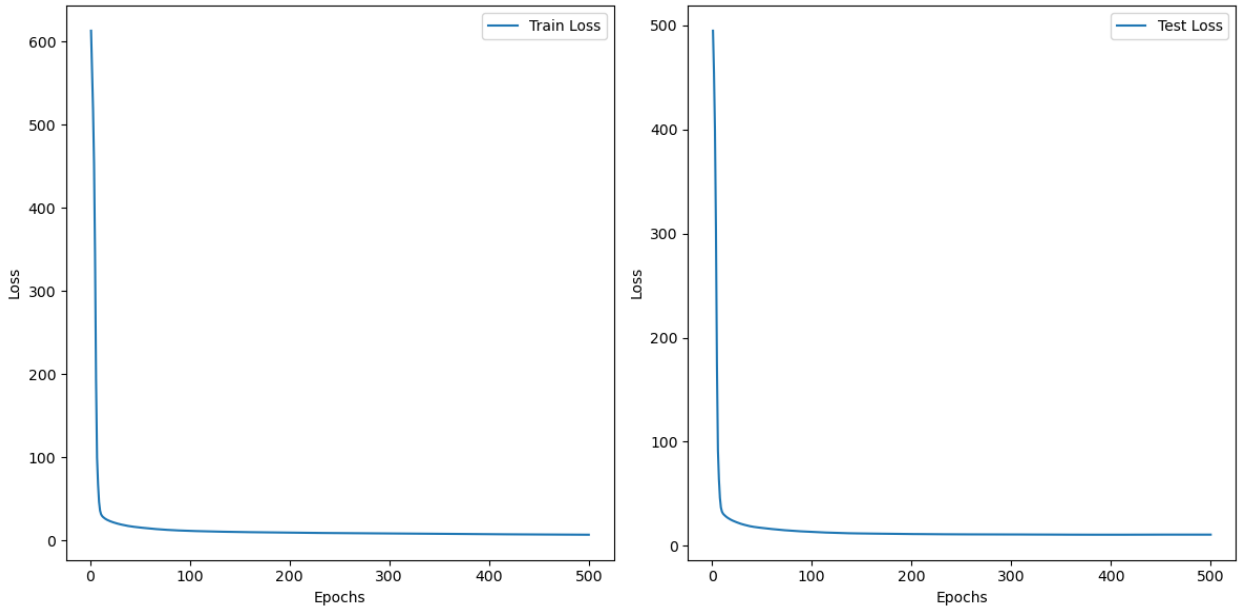


*Figure 11: Train Loss and Test Loss vs Epochs for SGD optimizer for learning rate = 0.01*

**iii. Learning rate = 0.001, Number of epochs = 5000**

Epoch [500/5000], Train Loss: 14.3153, Test Loss: 16.1936

Epoch [1000/5000], Train Loss: 11.2418, Test Loss: 13.2472

Epoch [1500/5000], Train Loss: 10.0432, Test Loss: 12.1792

Epoch [2000/5000], Train Loss: 9.3360, Test Loss: 11.4713

Epoch [2500/5000], Train Loss: 8.8000, Test Loss: 11.0765

Epoch [3000/5000], Train Loss: 8.3863, Test Loss: 10.9499

Epoch [3500/5000], Train Loss: 8.0217, Test Loss: 10.7879

Epoch [4000/5000], Train Loss: 7.7528, Test Loss: 10.7664

Epoch [4500/5000], Train Loss: 7.5145, Test Loss: 10.8152

Epoch [5000/5000], Train Loss: 7.2257, Test Loss: 10.7128

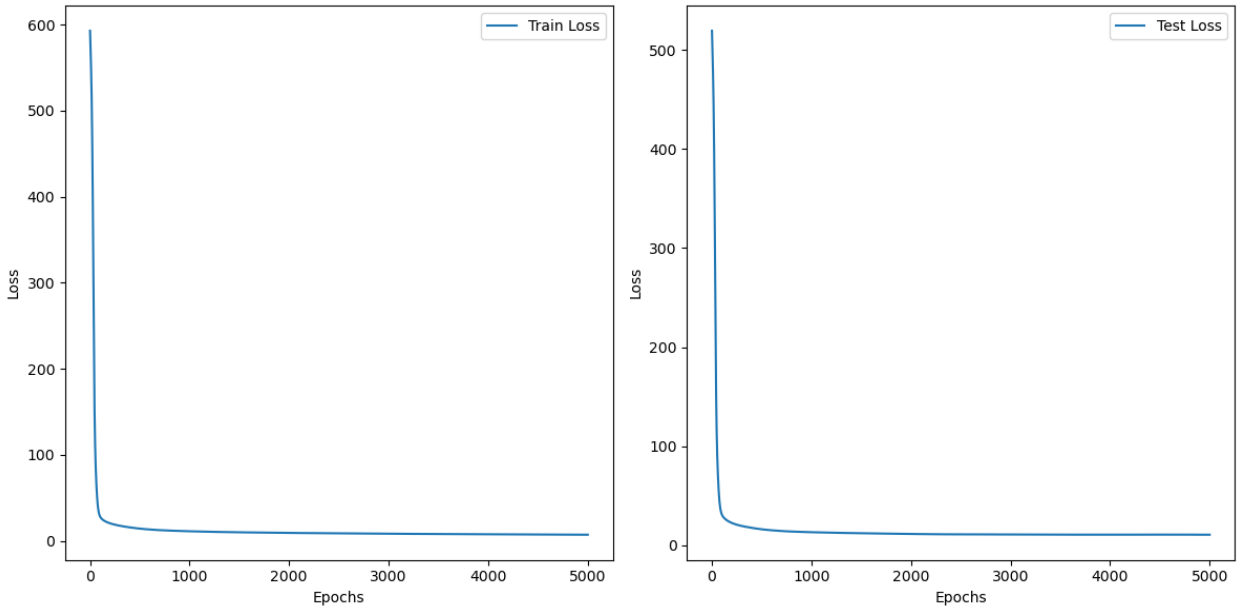


*Figure 12: Train Loss and Test Loss vs Epochs for SGD optimizer for learning rate = 0.001*

***2. Nesterov momentum optimizer***

**i. Learning rate = 0.1, Momentum = 0.1, Number of epochs = 200**

Epoch [20/200], Train Loss: nan, Test Loss: nan

Epoch [40/200], Train Loss: nan, Test Loss: nan

Epoch [60/200], Train Loss: nan, Test Loss: nan

Epoch [80/200], Train Loss: nan, Test Loss: nan

Epoch [100/200], Train Loss: nan, Test Loss: nan

Epoch [120/200], Train Loss: nan, Test Loss: nan

Epoch [140/200], Train Loss: nan, Test Loss: nan

Epoch [160/200], Train Loss: nan, Test Loss: nan

Epoch [180/200], Train Loss: nan, Test Loss: nan

Epoch [200/200], Train Loss: nan, Test Loss: nan

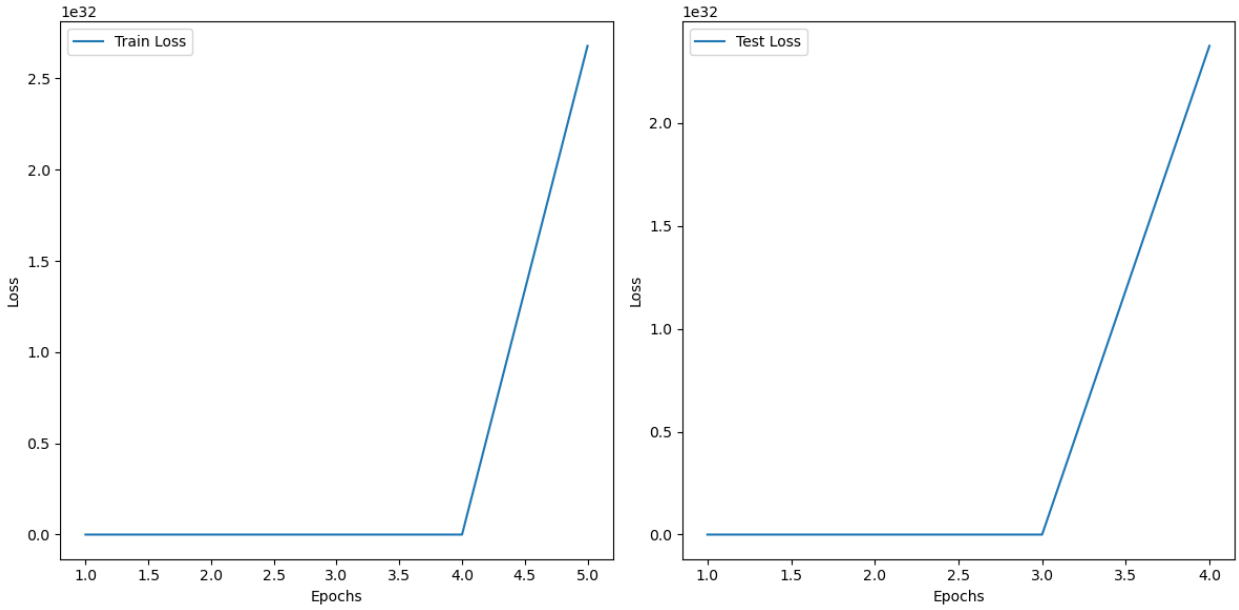


*Figure 13: Train Loss and Test Loss vs Epochs for Nesterov momentum optimizer for learning rate = 0.1*

**ii. Learning rate = 0.01, Momentum = 0.1, Number of epochs = 00**

Epoch [20/200], Train Loss: 22.2514, Test Loss: 24.4352

Epoch [40/200], Train Loss: 16.1033, Test Loss: 18.3904

Epoch [60/200], Train Loss: 13.7145, Test Loss: 15.7450

Epoch [80/200], Train Loss: 12.3760, Test Loss: 14.2362

Epoch [100/200], Train Loss: 11.5516, Test Loss: 13.3285

Epoch [120/200], Train Loss: 10.9238, Test Loss: 12.8320

Epoch [140/200], Train Loss: 10.4169, Test Loss: 12.4669

Epoch [160/200], Train Loss: 10.0192, Test Loss: 12.1616

Epoch [180/200], Train Loss: 9.6601, Test Loss: 11.8969

Epoch [200/200], Train Loss: 9.3598, Test Loss: 11.7662

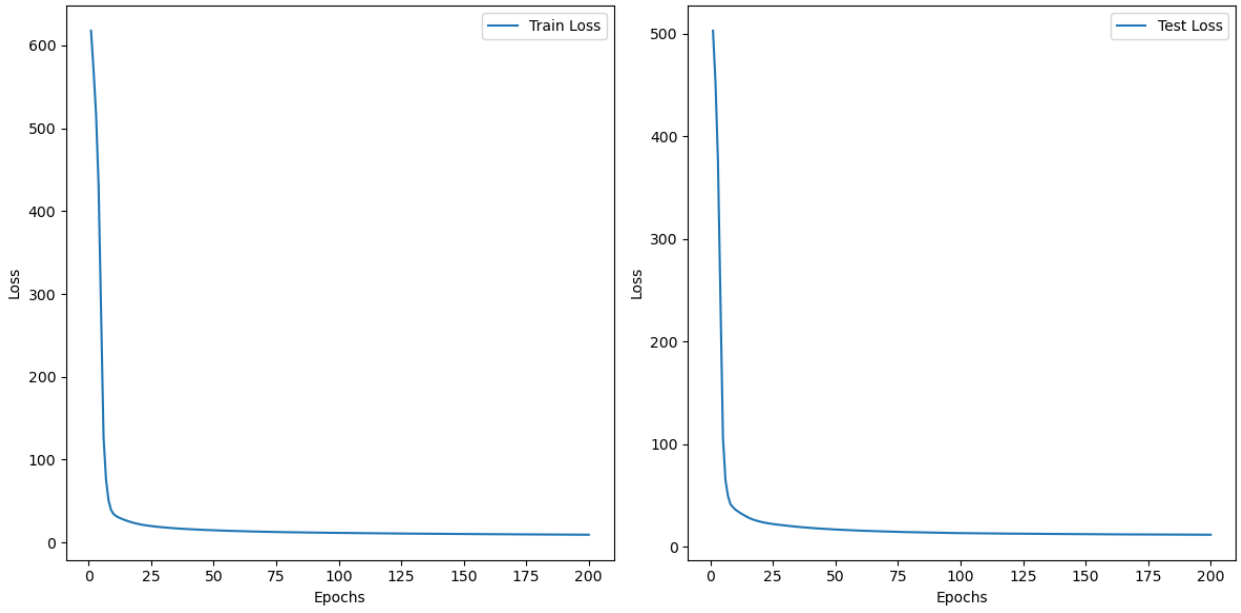


*Figure 14: Train Loss and Test Loss vs Epochs for Nesterov momentum optimizer for learning rate = 0.01*

**iii. Learning rate = 0.001, Momentum = 0.1, Number of epochs = 500**

Epoch [100/1000], Train Loss: 27.6928, Test Loss: 29.3733

Epoch [200/1000], Train Loss: 19.4399, Test Loss: 21.3653

Epoch [300/1000], Train Loss: 16.7017, Test Loss: 19.1698

Epoch [400/1000], Train Loss: 15.0338, Test Loss: 17.7017

Epoch [500/1000], Train Loss: 13.8434, Test Loss: 16.5721

Epoch [600/1000], Train Loss: 12.9916, Test Loss: 15.7400

Epoch [700/1000], Train Loss: 12.3653, Test Loss: 15.0909

Epoch [800/1000], Train Loss: 11.8837, Test Loss: 14.5416

Epoch [900/1000], Train Loss: 11.5122, Test Loss: 14.1035

Epoch [1000/1000], Train Loss: 11.1963, Test Loss: 13.7374

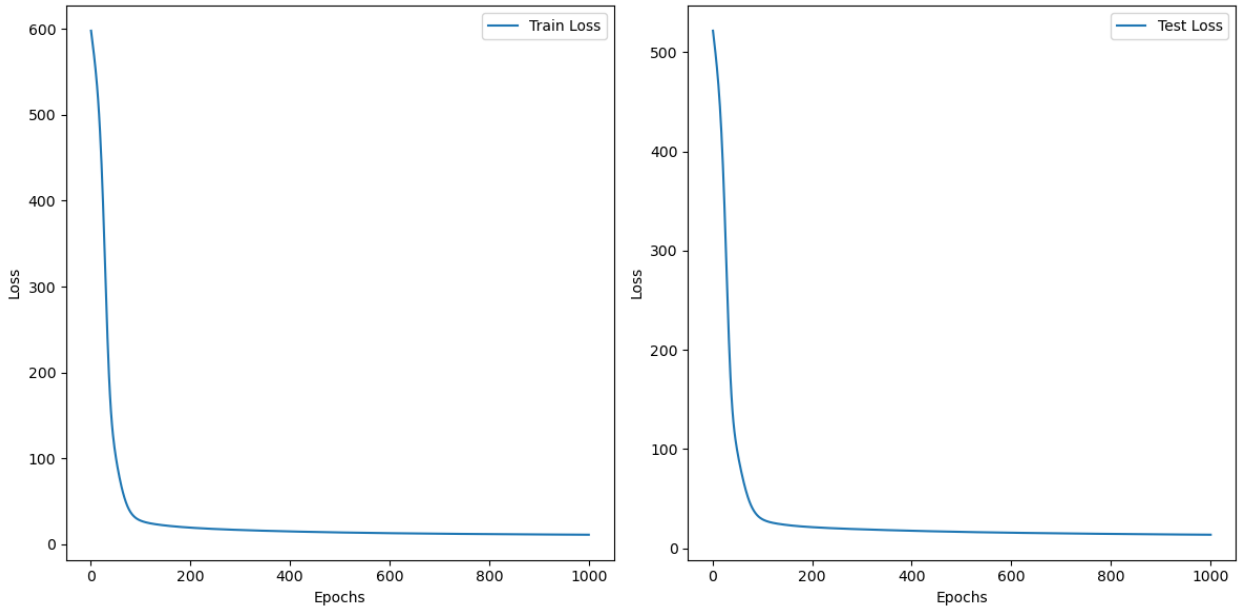


*Figure 15: Train Loss and Test Loss vs Epochs for Nesterov momentum optimizer for learning rate = 0.001*

***3. Adadelta optimizer***

**i. Learning rate = 0.1, Number of epochs = 5000**

Epoch [500/5000], Train Loss: 32.7484, Test Loss: 35.7041

Epoch [1000/5000], Train Loss: 9.6530, Test Loss: 12.7748

Epoch [1500/5000], Train Loss: 7.2026, Test Loss: 11.2055

Epoch [2000/5000], Train Loss: 6.5376, Test Loss: 10.6948

Epoch [2500/5000], Train Loss: 6.2330, Test Loss: 10.5848

Epoch [3000/5000], Train Loss: 5.9837, Test Loss: 10.5377

Epoch [3500/5000], Train Loss: 5.7869, Test Loss: 10.4797

Epoch [4000/5000], Train Loss: 5.6406, Test Loss: 10.5089

Epoch [4500/5000], Train Loss: 5.5478, Test Loss: 10.5197

Epoch [5000/5000], Train Loss: 5.4971, Test Loss: 10.4793

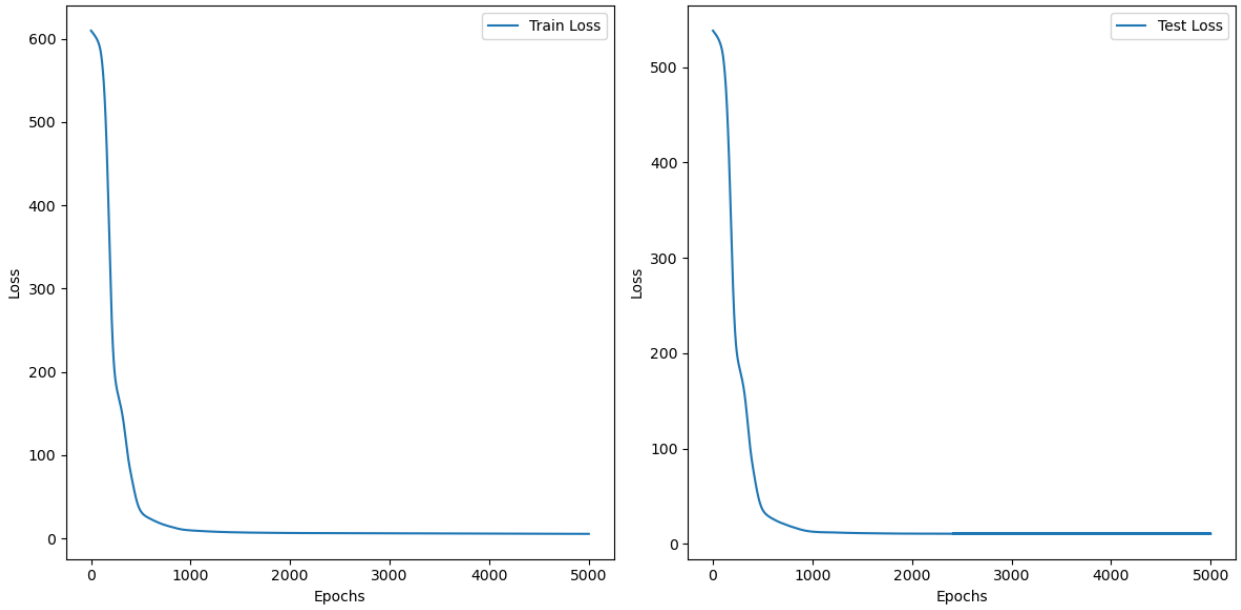


*Figure 16: Train Loss and Test Loss vs Epochs for Adadelta optimizer for learning rate = 0.1*

**ii. Learning rate = 0.01, Number of epochs = 10000**

Epoch [1000/10000], Train Loss: 403.8394, Test Loss: 352.3590

Epoch [2000/10000], Train Loss: 60.8314, Test Loss: 57.5961

Epoch [3000/10000], Train Loss: 25.0150, Test Loss: 28.0327

Epoch [4000/10000], Train Loss: 13.4896, Test Loss: 18.4301

Epoch [5000/10000], Train Loss: 10.1594, Test Loss: 14.8496

Epoch [6000/10000], Train Loss: 8.8159, Test Loss: 13.6900

Epoch [7000/10000], Train Loss: 7.9142, Test Loss: 12.7640

Epoch [8000/10000], Train Loss: 7.3958, Test Loss: 12.3029

Epoch [9000/10000], Train Loss: 7.0614, Test Loss: 12.1464

Epoch [10000/10000], Train Loss: 6.8411, Test Loss: 12.1138

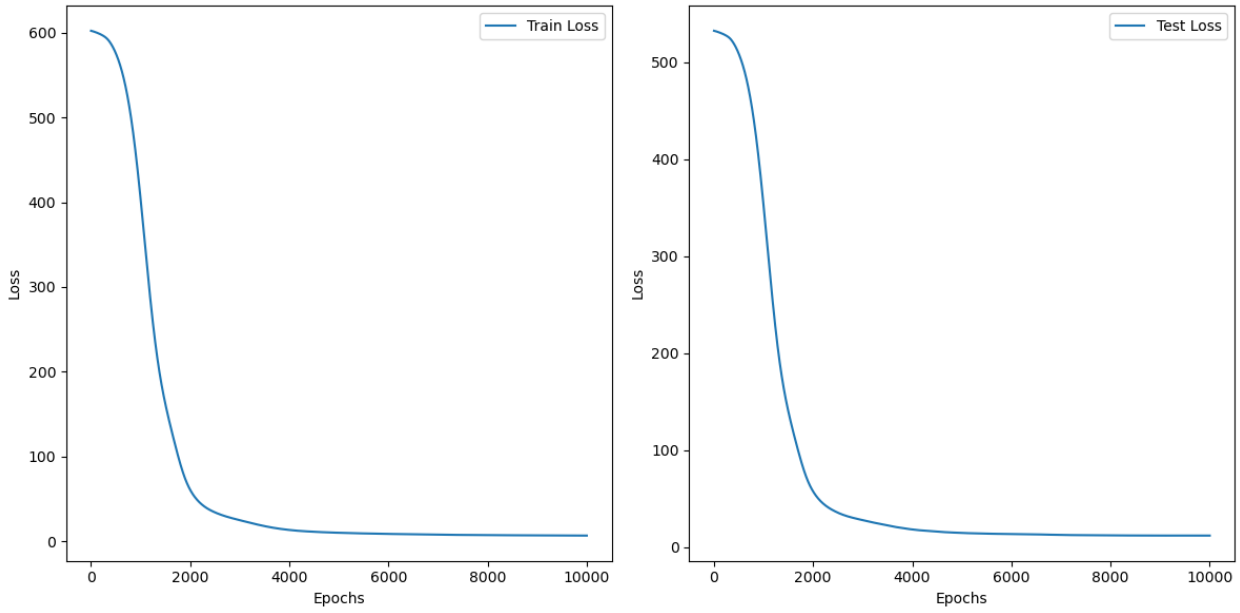


*Figure 17: Train Loss and Test Loss vs Epochs for Adadelta optimizer for learning rate = 0.01*

**iii. Learning rate = 0.001, Number of epochs = 10000**

Epoch [5000/50000], Train Loss: 512.0984, Test Loss: 448.1921

Epoch [10000/50000], Train Loss: 115.6317, Test Loss: 100.4330

Epoch [15000/50000], Train Loss: 19.9597, Test Loss: 23.0288

Epoch [20000/50000], Train Loss: 12.4897, Test Loss: 14.8886

Epoch [25000/50000], Train Loss: 9.2780, Test Loss: 11.6202

Epoch [30000/50000], Train Loss: 8.2651, Test Loss: 10.7893

Epoch [35000/50000], Train Loss: 7.8016, Test Loss: 10.4141

Epoch [40000/50000], Train Loss: 7.3538, Test Loss: 10.3886

Epoch [45000/50000], Train Loss: 7.0225, Test Loss: 10.5291

Epoch [50000/50000], Train Loss: 6.7284, Test Loss: 10.7213

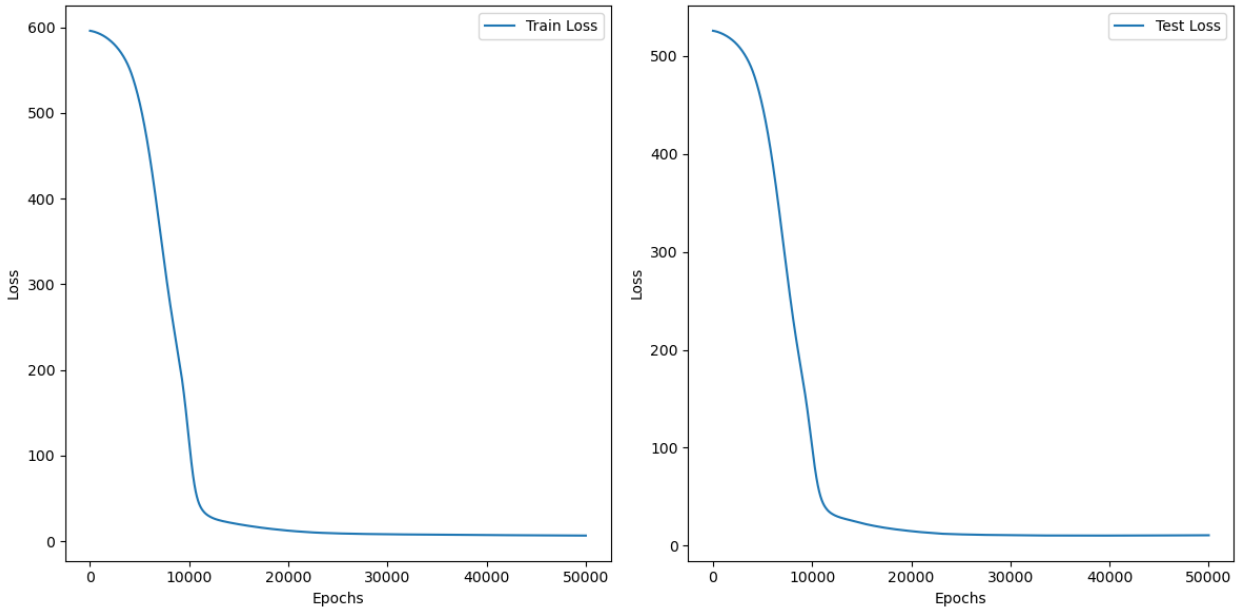


*Figure 18: Train Loss and Test Loss vs Epochs for Adadelta optimizer for learning rate = 0.001*

## Inference

**Part A: Difference between using different learning rates**

***1. Learning Rate = 0.1***

The model converges quickly in terms of the number of epochs since the learning rate is relatively high. The training loss decreases initially, but it does not settle around the optimal value, and the test loss might not decrease significantly.

***2. Learning Rate = 0.01***

It results in slower convergence, requiring more epochs to reach the optimal point. The training process is more stable compared to a higher learning rate. The training loss decreases more gradually, and we might need more epochs to see substantial improvements.

***3. Learning Rate = 0.001:***

With a very small learning rate, convergence is even slower. The training process is stable, but weequire a larger number of epochs to achieve significant improvements in the loss.

**Part B: Difference between using different optimisers**

***1. SGD***

SGD is sensitive to the tuning of the learning rate. The speed of convergence with SGD is slower compared to more advanced optimizers.

***2. Nesterov momentum***

Nesterov momentum helps accelerate convergence by reducing oscillations. It can navigate past shallow local minima and converge more directly. Nesterov momentum generally converges faster than standard SGD. With proper tuning of the momentum parameter, Nesterov momentum leads to better final performance.

***3. Adadelta***

Adadelta uses adaptive learning rates, which is not dependent manual learning rate tuning. It adapts the learning rates based on historical gradient information. Adadelta converges relatively slowly. Adadelta achieves stable convergence and optimal loss values.

**Part C: Difference with or without normalisation**

***1. With Normalisation***

**Convergence Behavior**: Normalizing the features often leads to smoother convergence. It can help optimization algorithms like gradient descent variants converge more quickly and reliably.

**Learning Rate Sensitivity**: With normalized data, the model is less sensitive to the learning rate, making it easier to find an appropriate learning rate that leads to convergence.

**Faster Convergence**: Normalization often results in faster convergence, as the optimizer does not need to take zig-zagging paths to the minimum.

### *2. Without Normalisation*

**Convergence Behaviour:** Without normalization, the loss landscape is skewed and elongated in different dimensions. This can lead to slow convergence, oscillations, and difficulty finding the global minimum.

**Learning Rate Sensitivity:** Without normalization, choosing an appropriate learning rate becomes crucial. A learning rate that is too high can lead to divergence or overshooting, while a learning rate that's too low can slow down convergence significantly.

**Slower Convergence:** Due to the irregularities in the loss landscape, optimization algorithms might require more iterations to find the optimal solution.