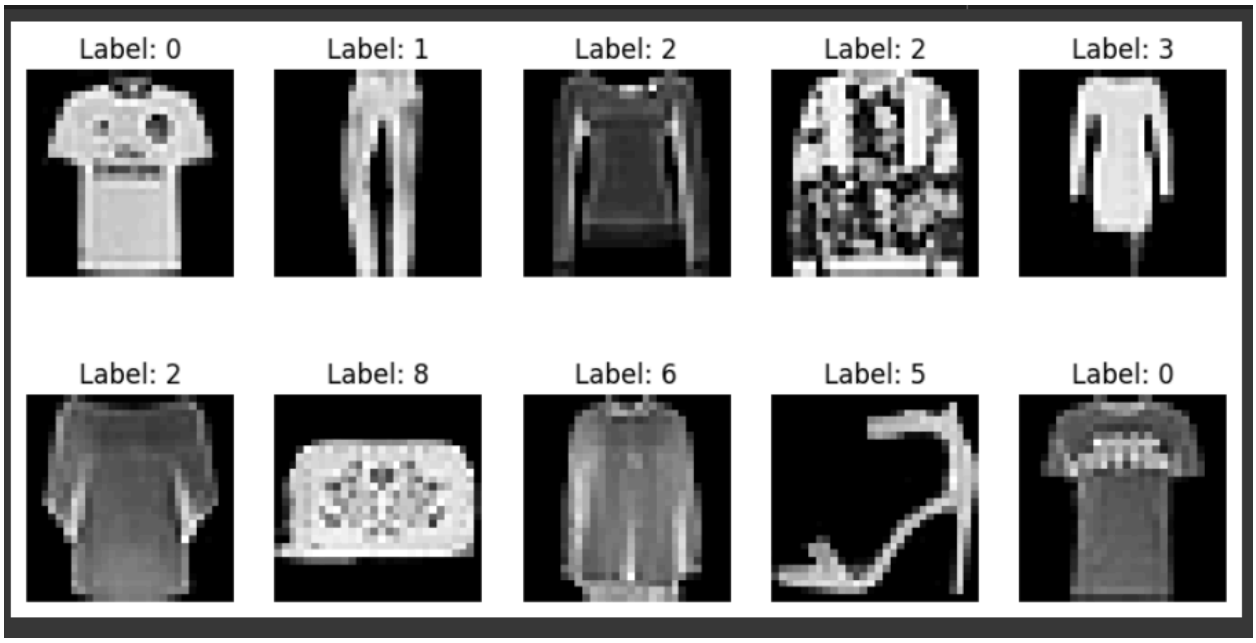
ML Assignment 3 Report
Roll no-2022150

Section A - Theory

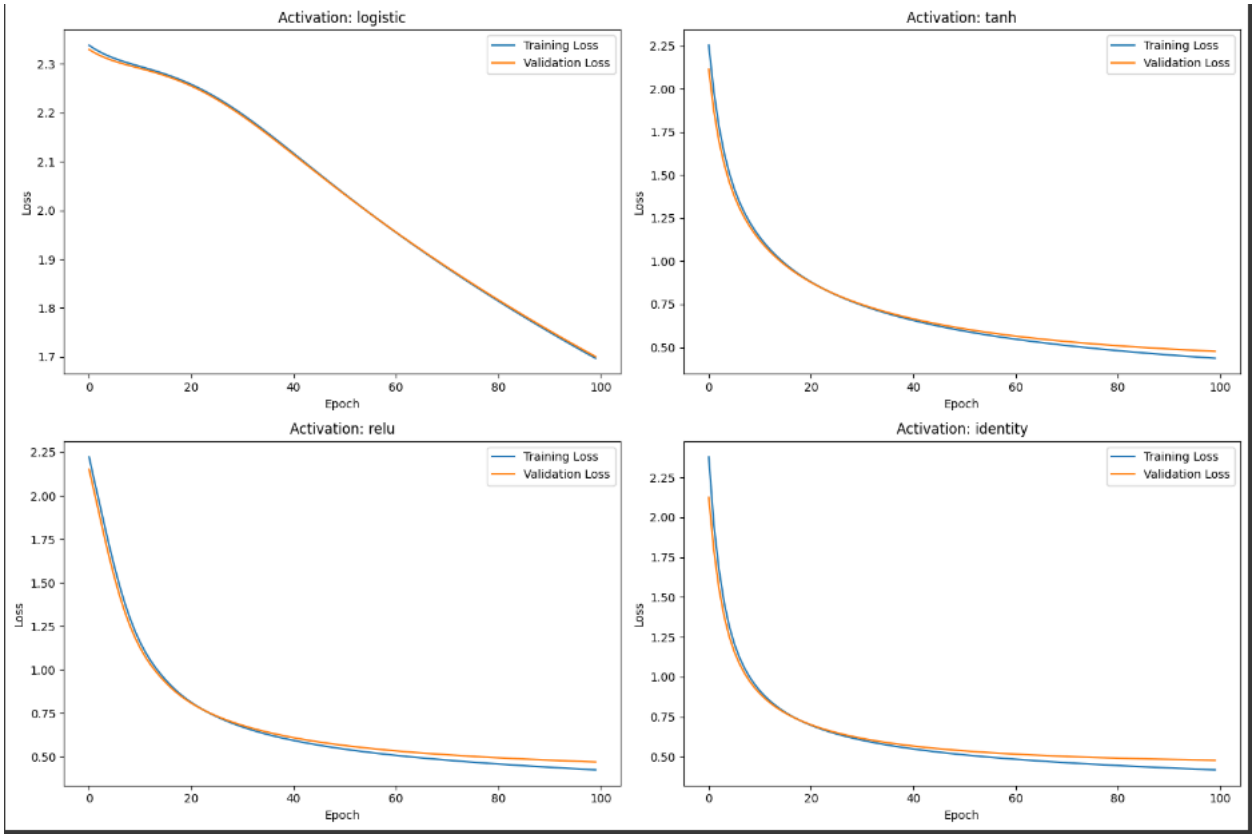
Section C -

Preprocessing:




```
Activation: logistic - Test Accuracy: 0.4405
Activation: tanh - Test Accuracy: 0.8300
Activation: relu - Test Accuracy: 0.8285
Activation: identity - Test Accuracy: 0.8235
```

training loss vs epochs
And validation loss vs epochs :

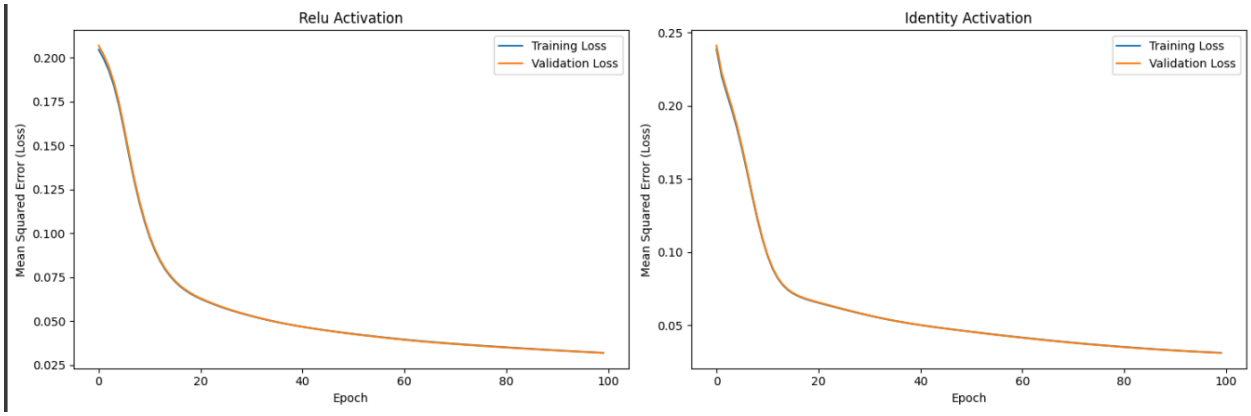



```
TanH gave the best possible performance on the test
set, as you can see from the graph too, tanh and
identity gave good and nearly the same results, but
with 83% accuracy, tanh was even better
```

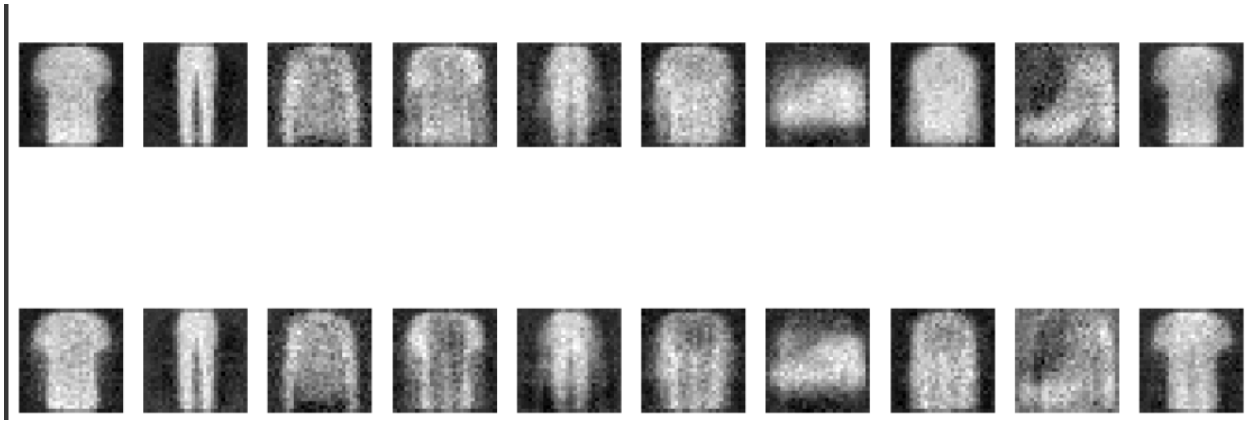
Grid search Results:

Best Batch size: 256,
Best learning rate: 0.0001,
Best solver: adam

Post training for both the models:

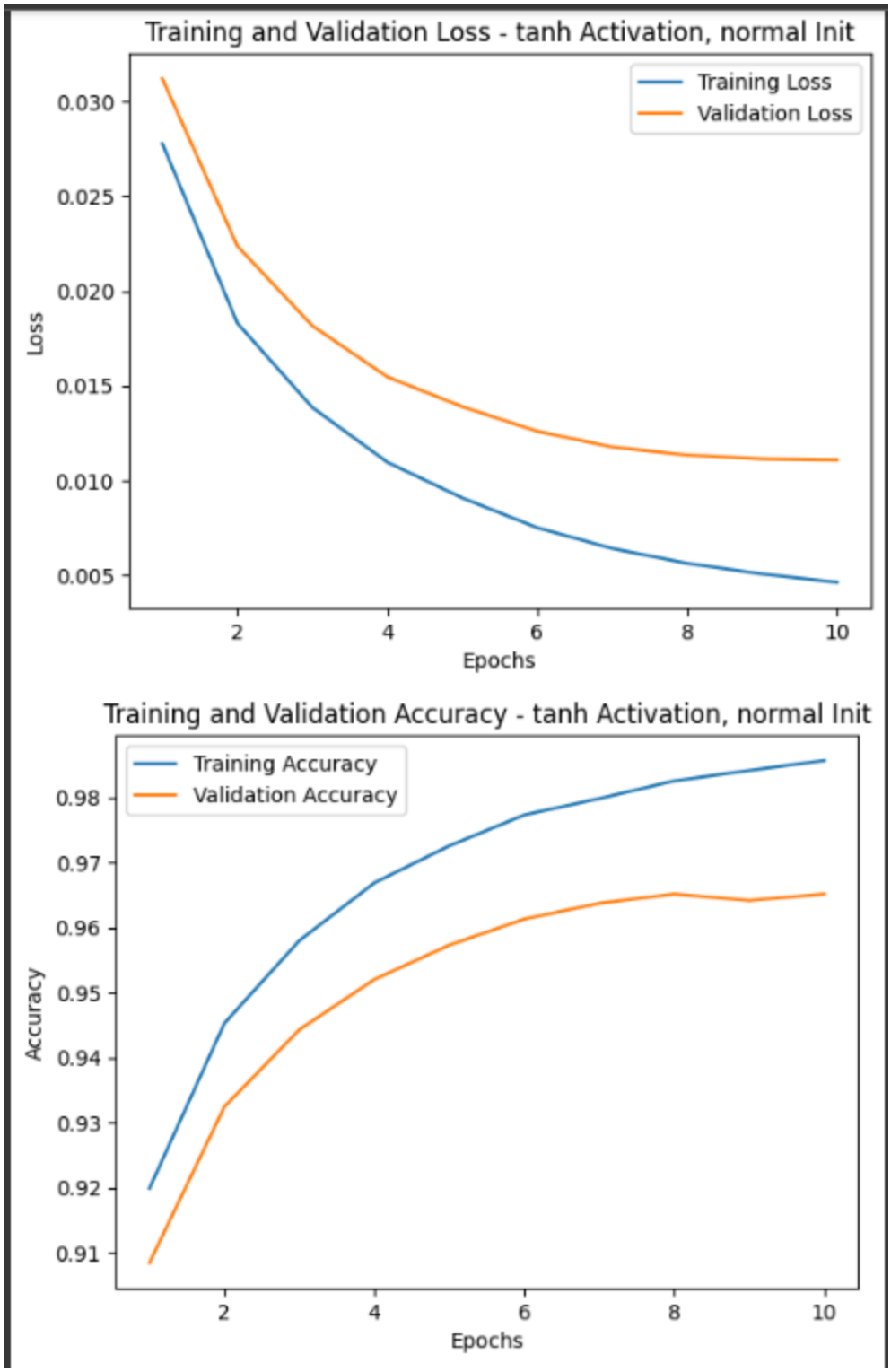


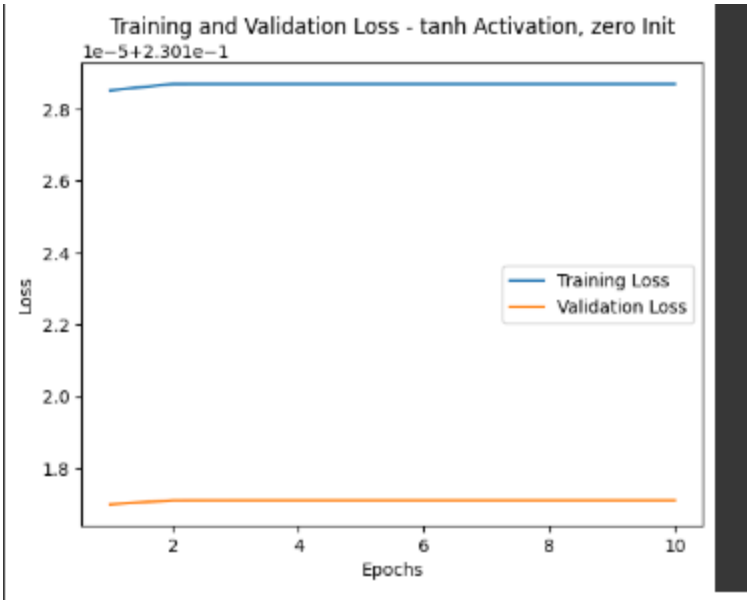
Images are blurry due to lossy decomposition:

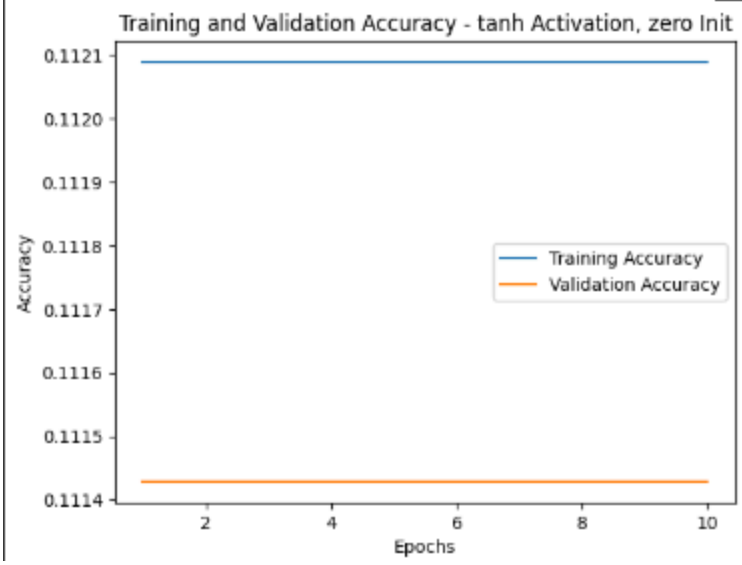


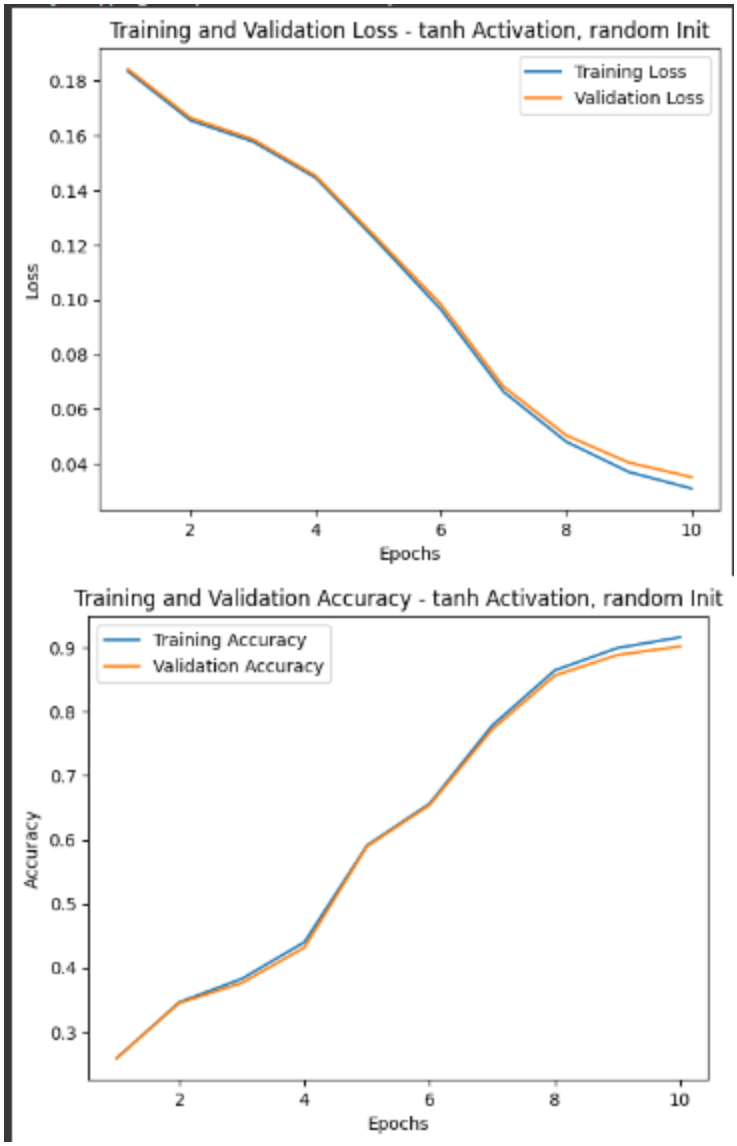
```
Test Accuracy for smaller MLP with relu features: 0.1435
Test Accuracy for smaller MLP with identity features: 0.2780

Test Accuracy for optimized MLP with relu features: 0.1015
Test Accuracy for optimized MLP with identity features: 0.7815
```

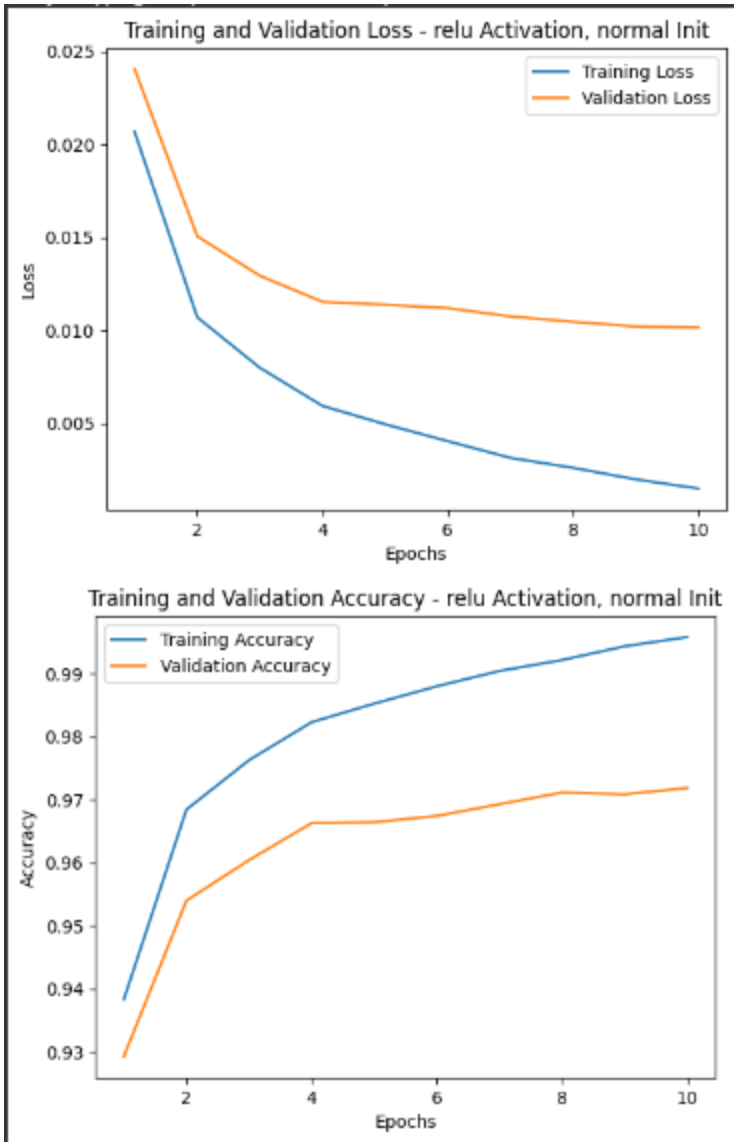
Section B -

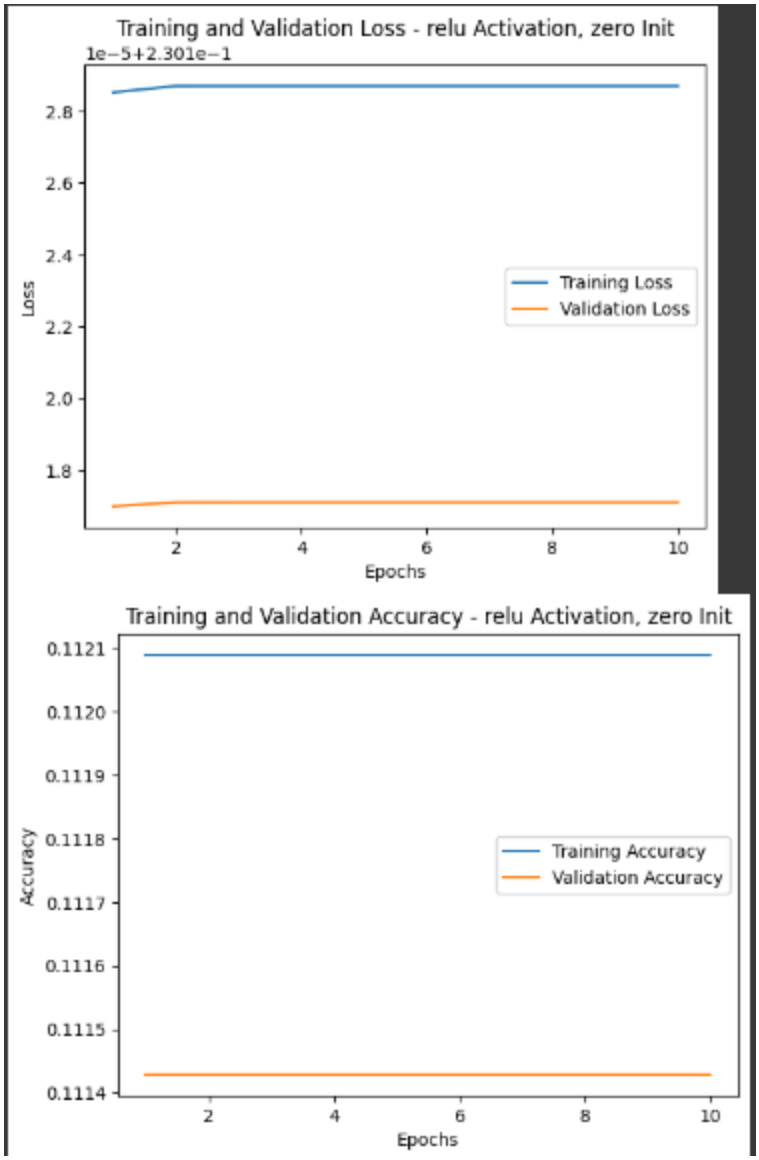

Training and Validation Loss - tanh Activation, normal Init
Training Loss
Validation Loss
Loss
0.030
0.025
0.020
0.015
0.010
0.005
2
4
6
8
10
Epochs
Training and Validation Accuracy - tanh Activation, normal Init
Training Accuracy
Validation Accuracy
Accuracy
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
2
4
6
8
10
Epochs


Training and Validation Loss - tanh Activation, zero Init
1e−5+2.301e−1
2.8
2.6
2.4
2.2
2.0
1.8
Loss
2
4
6
8
10
Epochs
Training Loss
Validation Loss

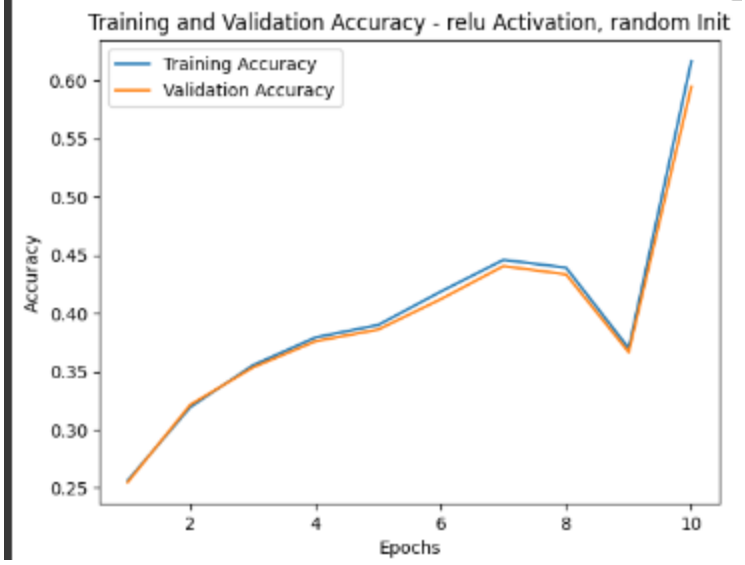

Training and Validation Accuracy - tanh Activation, zero Init
0.1121
0.1120
0.1119
0.1118
0.1117
0.1116
0.1115
0.1114
Accuracy
2
4
6
8
10
Epochs
Training Accuracy
Validation Accuracy

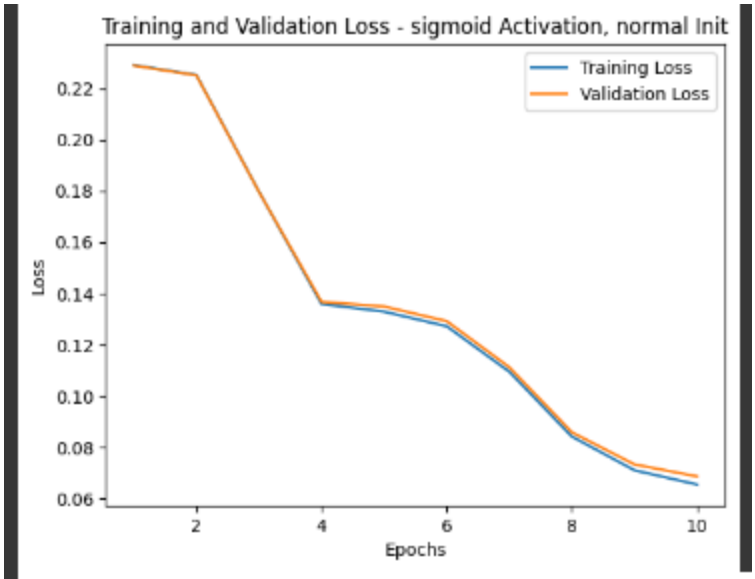

Training and Validation Loss - tanh Activation, random Init
Training Loss
Validation Loss
Loss
Epochs
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
2
4
6
8
10
Training and Validation Accuracy - tanh Activation, random Init
Training Accuracy
Validation Accuracy
Accuracy
Epochs
0.9
0.8
0.7
0.6
0.5
0.4
0.3
2
4
6
8
10

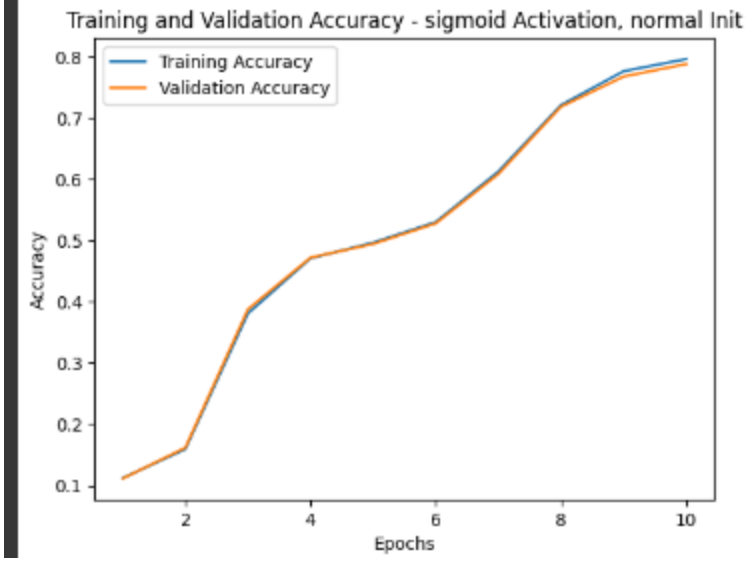

Training and Validation Loss - relu Activation, normal Init
Training Loss
Validation Loss
Loss
Epochs
0.025
0.020
0.015
0.010
0.005
2
4
6
8
10
Training and Validation Accuracy - relu Activation, normal Init
Training Accuracy
Validation Accuracy
Accuracy
Epochs
0.99
0.98
0.97
0.96
0.95
0.94
0.93
2
4
6
8
10

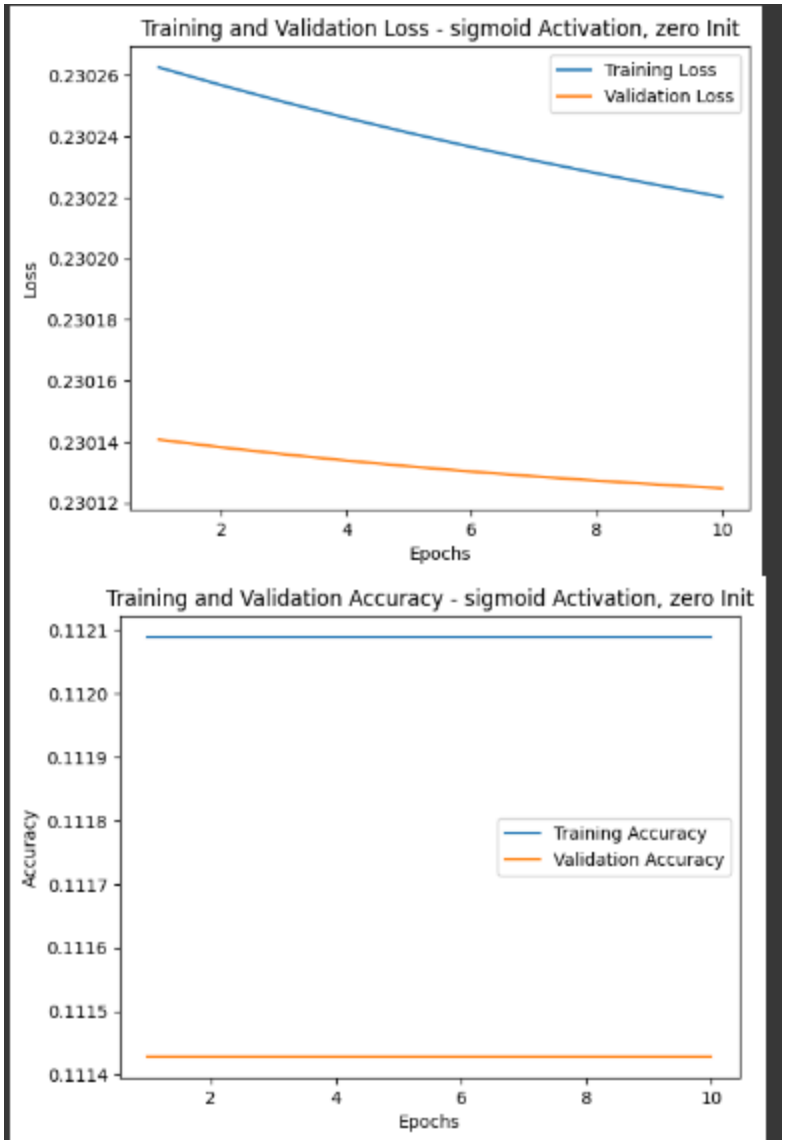

Training and Validation Loss - relu Activation, zero Init
1e−5+2.301e−1
2.8
2.6
2.4
2.2
2.0
1.8
Loss
2
4
6
8
10
Epochs
Training Loss
Validation Loss
Training and Validation Accuracy - relu Activation, zero Init
0.1121
0.1120
0.1119
0.1118
0.1117
0.1116
0.1115
0.1114
Accuracy
2
4
6
8
10
Epochs
Training Accuracy
Validation Accuracy

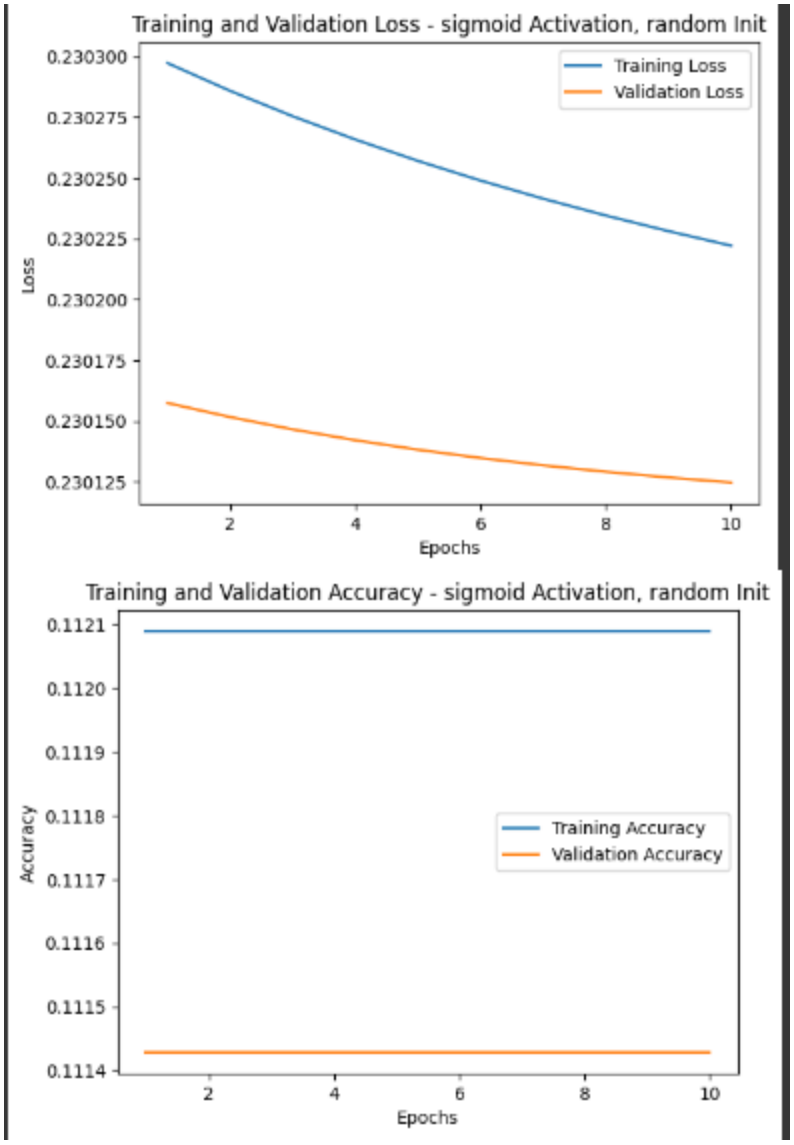

Training and Validation Loss - relu Activation, random Init
Training Loss
Validation Loss
Loss
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
2
4
6
8
10
Epochs

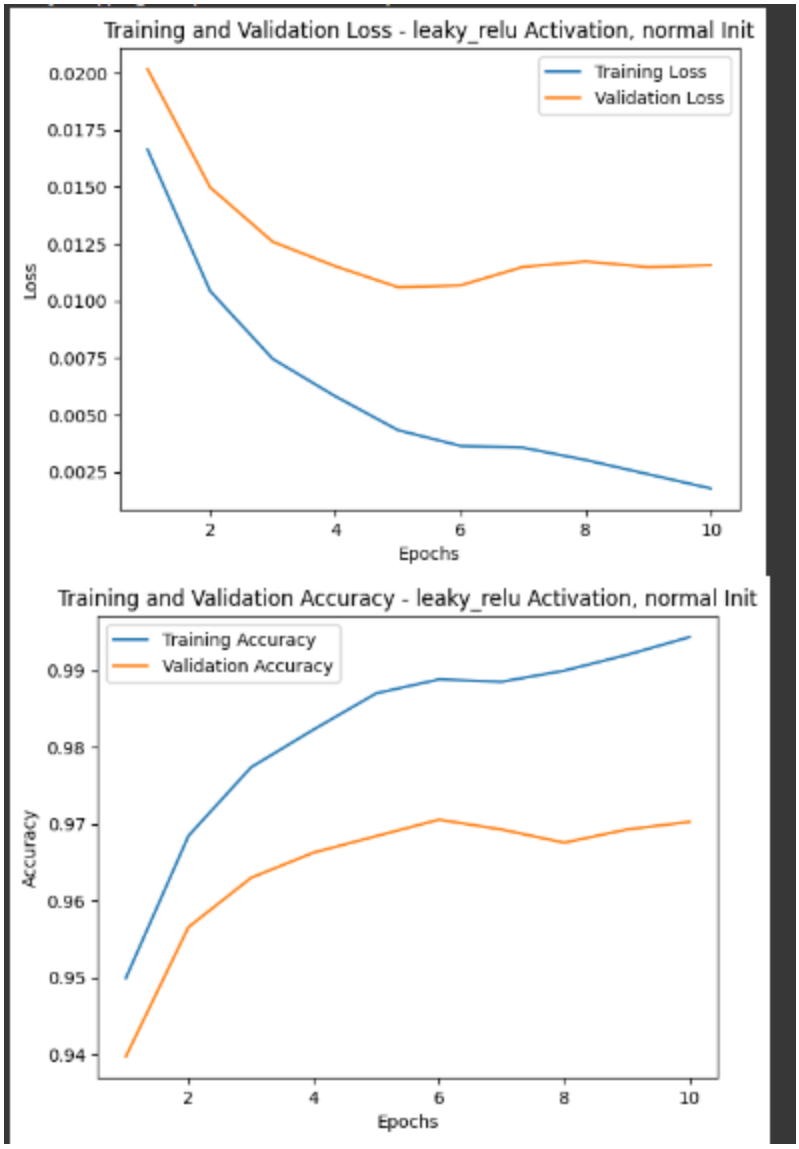

Training and Validation Accuracy - relu Activation, random Init
Training Accuracy
Validation Accuracy
Accuracy
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
2
4
6
8
10
Epochs

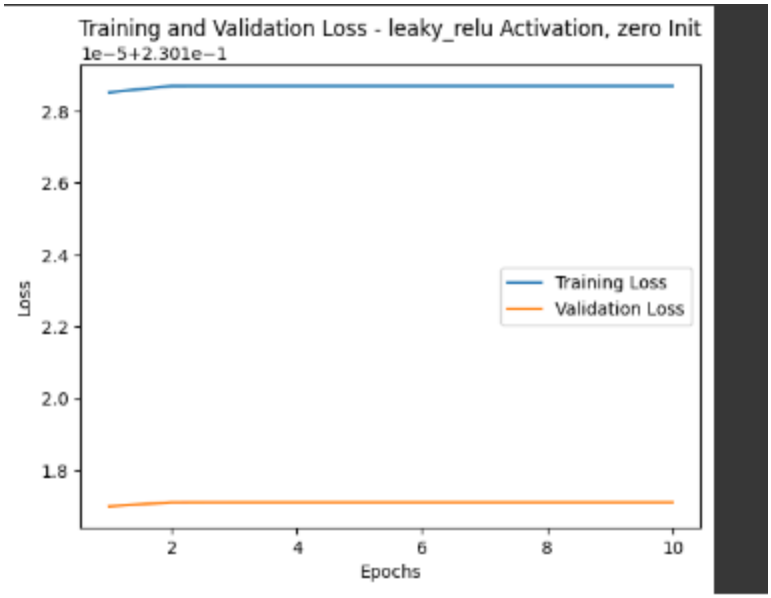

Training and Validation Loss - sigmoid Activation, normal Init
Training Loss
Validation Loss
Loss
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
2
4
6
8
10
Epochs

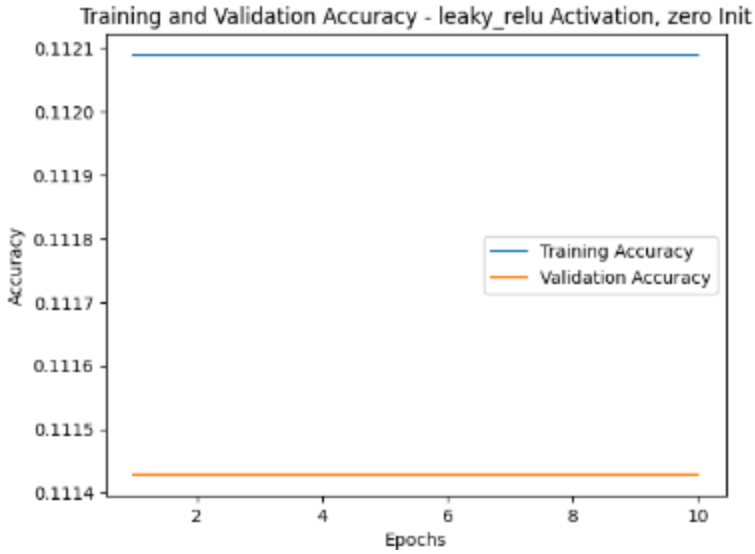

Training and Validation Accuracy - sigmoid Activation, normal Init
Training Accuracy
Validation Accuracy
Accuracy
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
2
4
6
8
10
Epochs

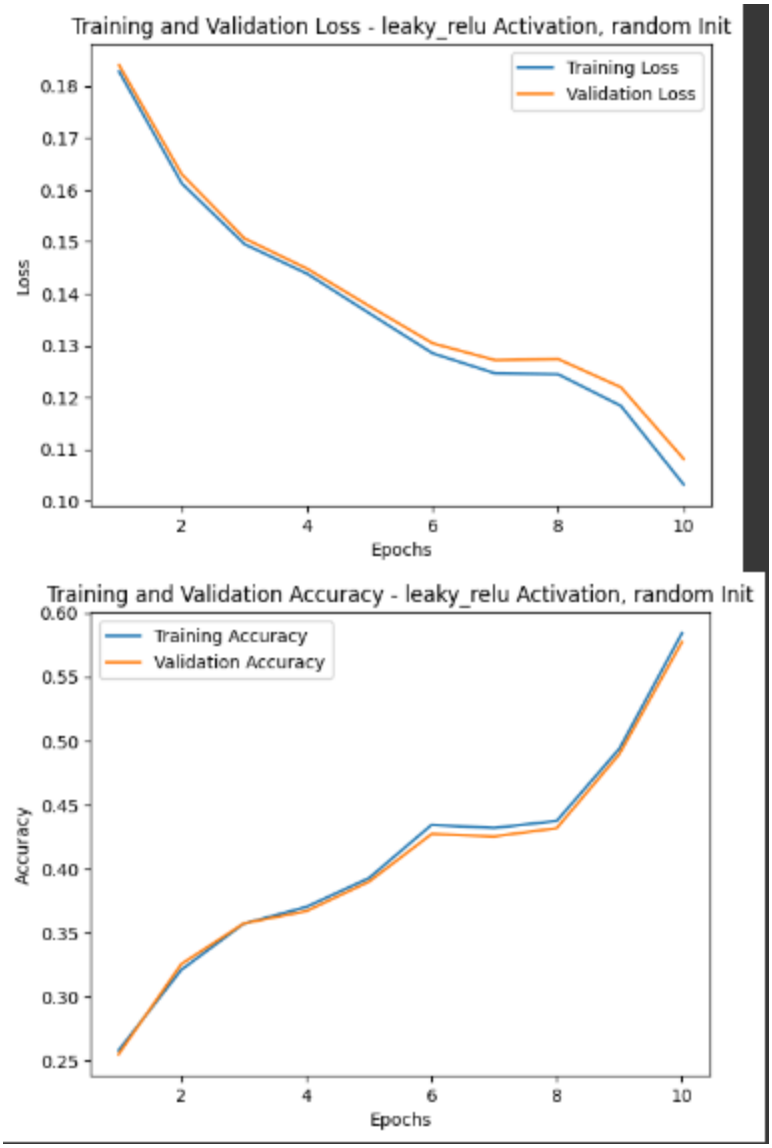

Training and Validation Loss - sigmoid Activation, zero Init
Training Loss
Validation Loss
Loss
0.23026
0.23024
0.23022
0.23020
0.23018
0.23016
0.23014
0.23012
2
4
6
8
10
Epochs
Training and Validation Accuracy - sigmoid Activation, zero Init
Training Accuracy
Validation Accuracy
Accuracy
0.1121
0.1120
0.1119
0.1118
0.1117
0.1116
0.1115
0.1114
2
4
6
8
10
Epochs


Training and Validation Loss - sigmoid Activation, random Init
Training Loss
Validation Loss
Loss
0.230300
0.230275
0.230250
0.230225
0.230200
0.230175
0.230150
0.230125
2
4
6
8
10
Epochs
Training and Validation Accuracy - sigmoid Activation, random Init
Training Accuracy
Validation Accuracy
Accuracy
0.1121
0.1120
0.1119
0.1118
0.1117
0.1116
0.1115
0.1114
2
4
6
8
10
Epochs


Training and Validation Loss - leaky_relu Activation, normal Init
Training Loss
Validation Loss
Loss
Epochs
Training and Validation Accuracy - leaky_relu Activation, normal Init
Training Accuracy
Validation Accuracy
Accuracy
Epochs

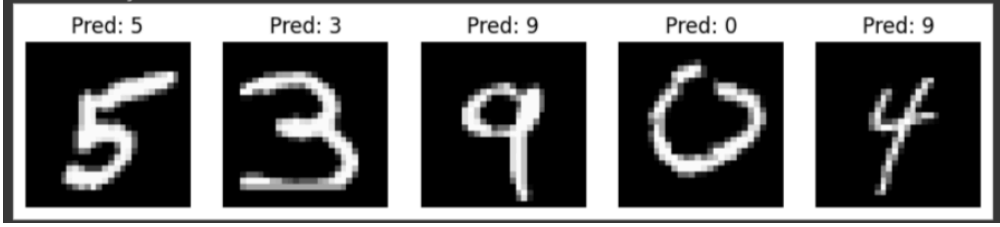

Training and Validation Loss - leaky_relu Activation, zero Init
1e−5+2.301e−1
2.8
2.6
2.4
2.2
2.0
1.8
Loss
2
4
6
8
10
Epochs
Training Loss
Validation Loss


Training and Validation Accuracy - leaky_relu Activation, zero Init
0.1121
0.1120
0.1119
0.1118
0.1117
0.1116
0.1115
0.1114
Accuracy
2
4
6
8
10
Epochs
Training Accuracy
Validation Accuracy



Pickle file example :

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
Model loaded from: /content/drive/MyDrive/NN_models/model_tanh_normal.pkl
Test Accuracy: 97.07%
```



The best performance is shown by the Leaky relu activation function and normal intitialization function,

The zero initialization has some suboptimal point, as it is getting stuck at some local extrema.
On increasing Learning rate, it isn’t changing much

Random initialization is changing on increasing the learning rate and it will converge the accuracy on increasing it a bit.

For normal initialization, it is converging on even using less learning rate.