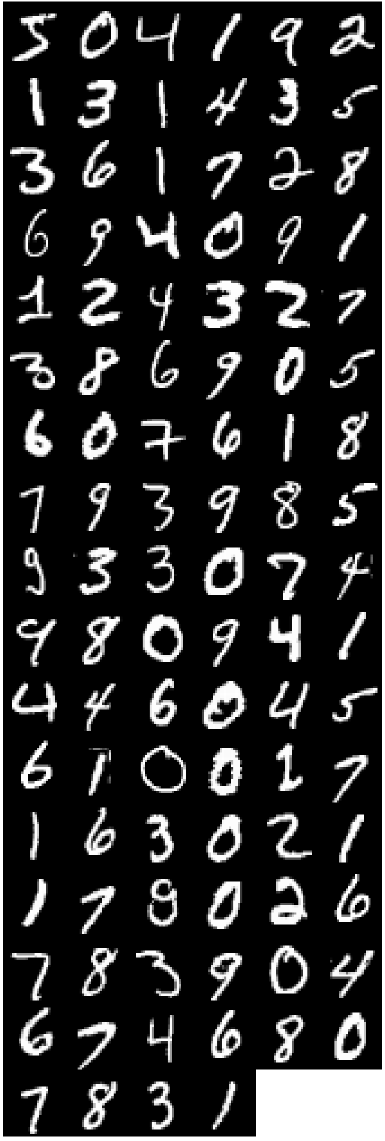
Computer Vision
Assignment 2

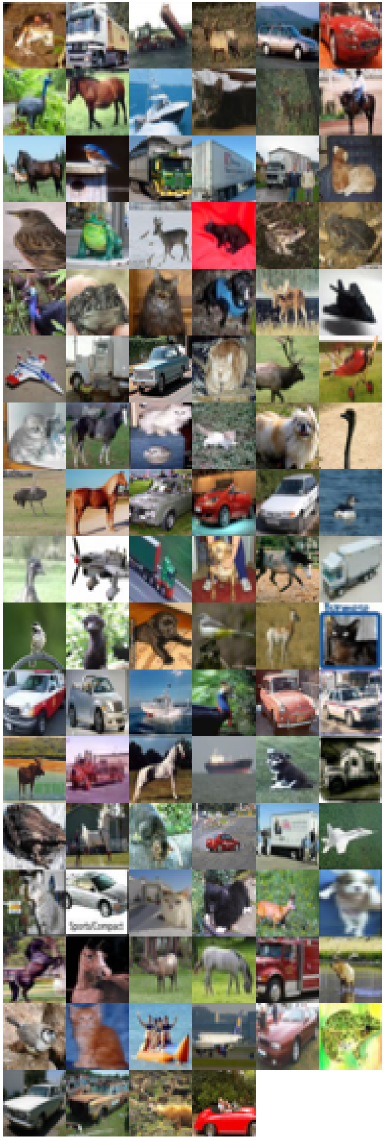
Name: Yash Sadhwani **NetID**: ys2913

Solution 2

MNIST Data:



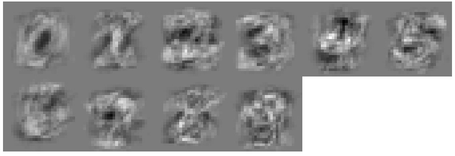
CIFAR Data:



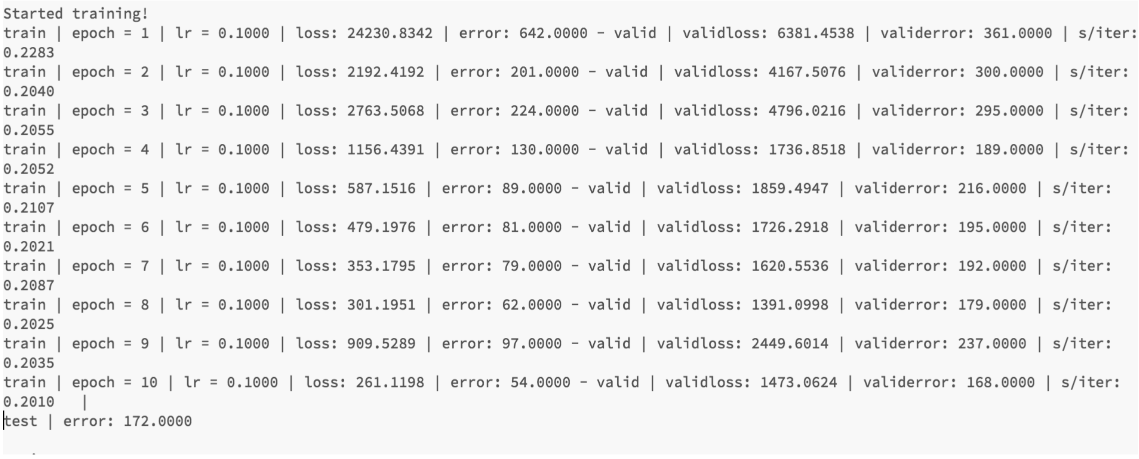
Solution 3(a)

Data= MNIST
Training batch size = 1000
Testing batch size = 1000
Validation Batch size = 1000

Weights :



Output:

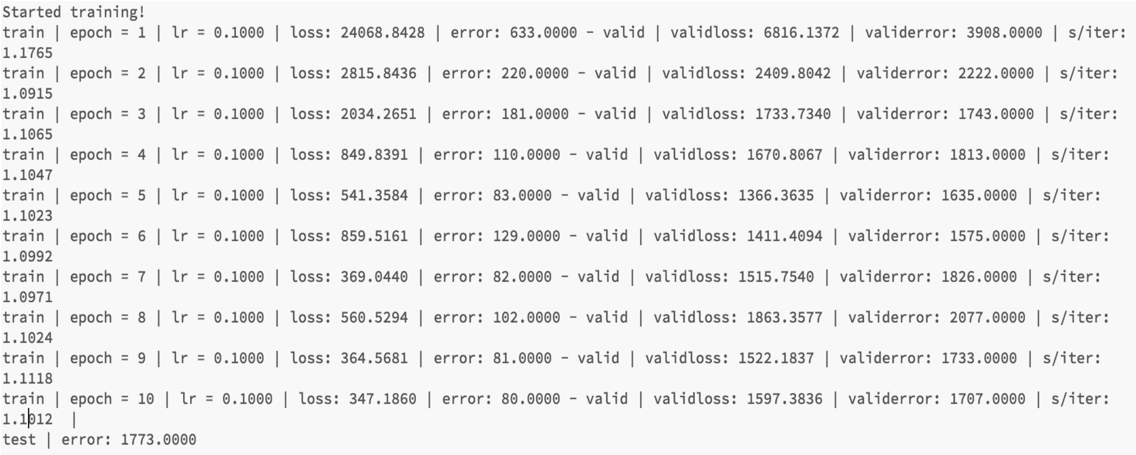


```
Started training!
train | epoch = 1 | lr = 0.1000 | loss: 24230.8342 | error: 642.0000 - valid | validloss: 6381.4538 | validerror: 361.0000 | s/iter:
0.2283
train | epoch = 2 | lr = 0.1000 | loss: 2192.4192 | error: 201.0000 - valid | validloss: 4167.5076 | validerror: 300.0000 | s/iter:
0.2040
train | epoch = 3 | lr = 0.1000 | loss: 2763.5068 | error: 224.0000 - valid | validloss: 4796.0216 | validerror: 295.0000 | s/iter:
0.2055
train | epoch = 4 | lr = 0.1000 | loss: 1156.4391 | error: 130.0000 - valid | validloss: 1736.8518 | validerror: 189.0000 | s/iter:
0.2052
train | epoch = 5 | lr = 0.1000 | loss: 587.1516 | error: 89.0000 - valid | validloss: 1859.4947 | validerror: 216.0000 | s/iter:
0.2107
train | epoch = 6 | lr = 0.1000 | loss: 479.1976 | error: 81.0000 - valid | validloss: 1726.2918 | validerror: 195.0000 | s/iter:
0.2021
train | epoch = 7 | lr = 0.1000 | loss: 353.1795 | error: 79.0000 - valid | validloss: 1620.5536 | validerror: 192.0000 | s/iter:
0.2087
train | epoch = 8 | lr = 0.1000 | loss: 301.1951 | error: 62.0000 - valid | validloss: 1391.0998 | validerror: 179.0000 | s/iter:
0.2025
train | epoch = 9 | lr = 0.1000 | loss: 909.5289 | error: 97.0000 - valid | validloss: 2449.6014 | validerror: 237.0000 | s/iter:
0.2035
train | epoch = 10 | lr = 0.1000 | loss: 261.1198 | error: 54.0000 - valid | validloss: 1473.0624 | validerror: 168.0000 | s/iter:
0.2010   |
test | error: 172.0000
```

Solution 3(b)

Data= MNIST
Training batch size = 1000
Testing batch size = Full
Validation Batch size = Full

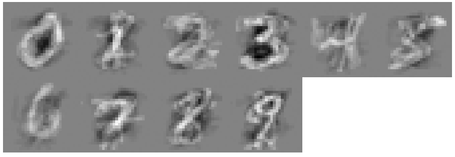
Weights: (Solution 3(a))
Output:



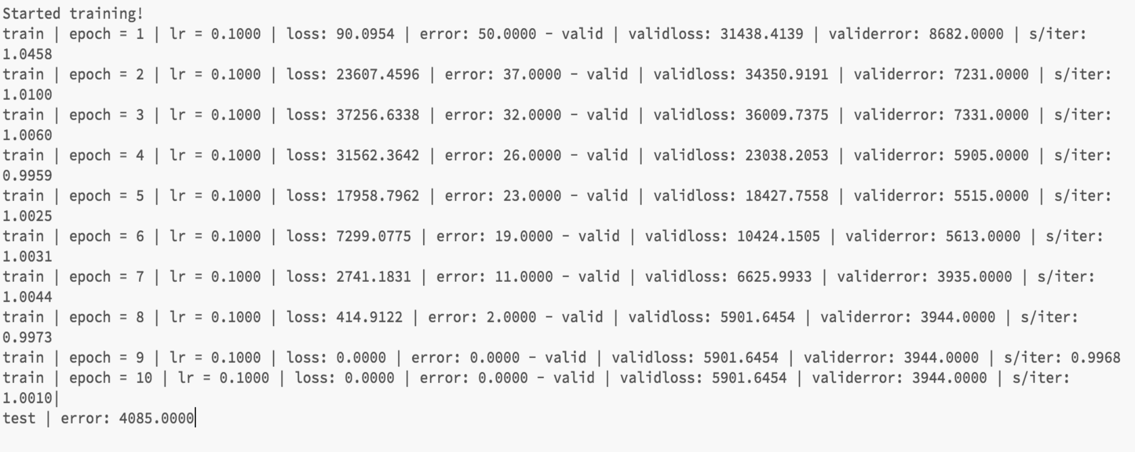
```
Started training!
train | epoch = 1 | lr = 0.1000 | loss: 24068.8428 | error: 633.0000 - valid | validloss: 6816.1372 | validerror: 3908.0000 | s/iter:
1.1765
train | epoch = 2 | lr = 0.1000 | loss: 2815.8436 | error: 220.0000 - valid | validloss: 2409.8042 | validerror: 2222.0000 | s/iter:
1.0915
train | epoch = 3 | lr = 0.1000 | loss: 2034.2651 | error: 181.0000 - valid | validloss: 1733.7340 | validerror: 1743.0000 | s/iter:
1.1065
train | epoch = 4 | lr = 0.1000 | loss: 849.8391 | error: 110.0000 - valid | validloss: 1670.8067 | validerror: 1813.0000 | s/iter:
1.1047
train | epoch = 5 | lr = 0.1000 | loss: 541.3584 | error: 83.0000 - valid | validloss: 1366.3635 | validerror: 1635.0000 | s/iter:
1.1023
train | epoch = 6 | lr = 0.1000 | loss: 859.5161 | error: 129.0000 - valid | validloss: 1411.4094 | validerror: 1575.0000 | s/iter:
1.0992
train | epoch = 7 | lr = 0.1000 | loss: 369.0440 | error: 82.0000 - valid | validloss: 1515.7540 | validerror: 1826.0000 | s/iter:
1.0971
train | epoch = 8 | lr = 0.1000 | loss: 560.5294 | error: 102.0000 - valid | validloss: 1863.3577 | validerror: 2077.0000 | s/iter:
1.1024
train | epoch = 9 | lr = 0.1000 | loss: 364.5681 | error: 81.0000 - valid | validloss: 1522.1837 | validerror: 1733.0000 | s/iter:
1.1118
train | epoch = 10 | lr = 0.1000 | loss: 347.1860 | error: 80.0000 - valid | validloss: 1597.3836 | validerror: 1707.0000 | s/iter:
1.1012  |
test | error: 1773.0000
```

Training batch size = 50
Testing batch size = Full
Validation Batch size = Full

Weights :



Output:



```
Started training!
train | epoch = 1 | lr = 0.1000 | loss: 90.0954 | error: 50.0000 - valid | validloss: 31438.4139 | validerror: 8682.0000 | s/iter:
1.0458
train | epoch = 2 | lr = 0.1000 | loss: 23607.4596 | error: 37.0000 - valid | validloss: 34350.9191 | validerror: 7231.0000 | s/iter:
1.0100
train | epoch = 3 | lr = 0.1000 | loss: 37256.6338 | error: 32.0000 - valid | validloss: 36009.7375 | validerror: 7331.0000 | s/iter:
1.0060
train | epoch = 4 | lr = 0.1000 | loss: 31562.3642 | error: 26.0000 - valid | validloss: 23038.2053 | validerror: 5905.0000 | s/iter:
0.9959
train | epoch = 5 | lr = 0.1000 | loss: 17958.7962 | error: 23.0000 - valid | validloss: 18427.7558 | validerror: 5515.0000 | s/iter:
1.0025
train | epoch = 6 | lr = 0.1000 | loss: 7299.0775 | error: 19.0000 - valid | validloss: 10424.1505 | validerror: 5613.0000 | s/iter:
1.0031
train | epoch = 7 | lr = 0.1000 | loss: 2741.1831 | error: 11.0000 - valid | validloss: 6625.9933 | validerror: 3935.0000 | s/iter:
1.0044
train | epoch = 8 | lr = 0.1000 | loss: 414.9122 | error: 2.0000 - valid | validloss: 5901.6454 | validerror: 3944.0000 | s/iter:
0.9973
train | epoch = 9 | lr = 0.1000 | loss: 0.0000 | error: 0.0000 - valid | validloss: 5901.6454 | validerror: 3944.0000 | s/iter: 0.9968
train | epoch = 10 | lr = 0.1000 | loss: 0.0000 | error: 0.0000 - valid | validloss: 5901.6454 | validerror: 3944.0000 | s/iter:
1.0010|
test | error: 4085.0000
```

The test error for Training Size=1000: 1773.00
The test error for Training Size=50: 4085.00
The test error has increased.
Small training size, results in less generalization, which can be seen while comparing the images of the weights.
We can se that the loss converges to 0, which means that there is overfitting of data.
So, the lack of sufficient training data and overfitting leads to higher error

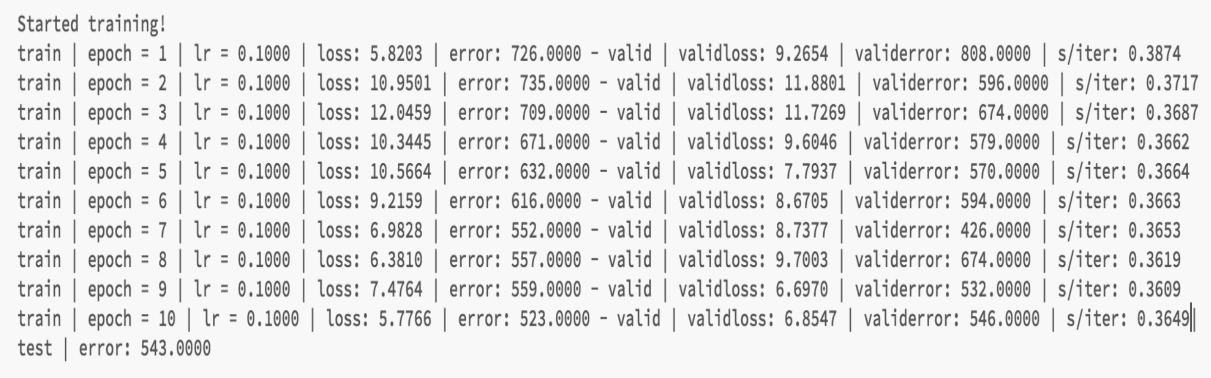
Solution 4

Sequential Network
Linear(1,1000)
Tanh()
Linear(1000,10)

Data= MNIST
Training batch size = 1000
Testing batch size = 1000
Validation Batch size = 1000

Learning Rate=0.1

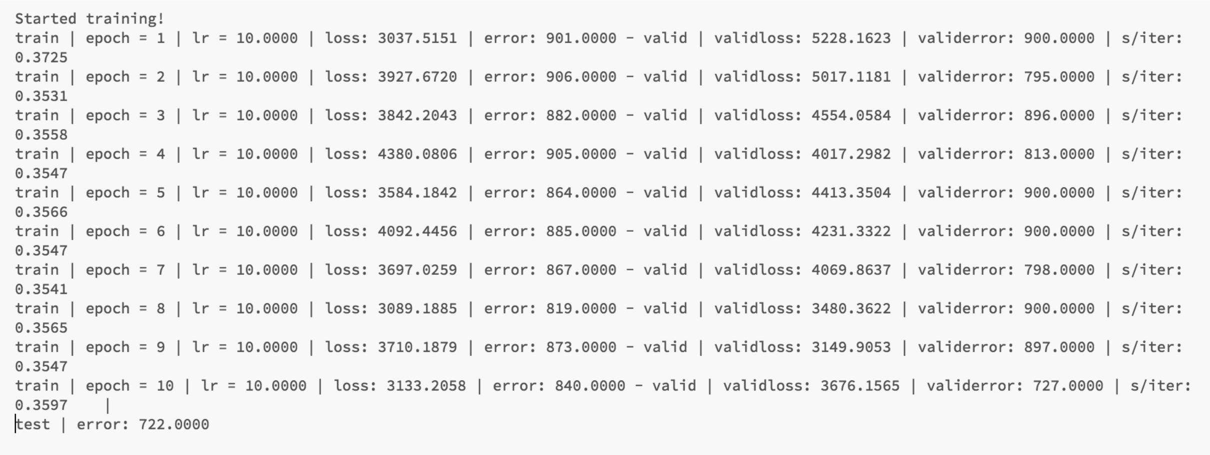
Output:



```
Started training!
train | epoch = 1 | lr = 0.1000 | loss: 5.8203 | error: 726.0000 - valid | validloss: 9.2654 | validerror: 808.0000 | s/iter: 0.3874
train | epoch = 2 | lr = 0.1000 | loss: 10.9501 | error: 735.0000 - valid | validloss: 11.8801 | validerror: 596.0000 | s/iter: 0.3717
train | epoch = 3 | lr = 0.1000 | loss: 12.0459 | error: 709.0000 - valid | validloss: 11.7269 | validerror: 674.0000 | s/iter: 0.3687
train | epoch = 4 | lr = 0.1000 | loss: 10.3445 | error: 671.0000 - valid | validloss: 9.6046 | validerror: 579.0000 | s/iter: 0.3662
train | epoch = 5 | lr = 0.1000 | loss: 10.5664 | error: 632.0000 - valid | validloss: 7.7937 | validerror: 570.0000 | s/iter: 0.3664
train | epoch = 6 | lr = 0.1000 | loss: 9.2159 | error: 616.0000 - valid | validloss: 8.6705 | validerror: 594.0000 | s/iter: 0.3663
train | epoch = 7 | lr = 0.1000 | loss: 6.9828 | error: 552.0000 - valid | validloss: 8.7377 | validerror: 426.0000 | s/iter: 0.3653
train | epoch = 8 | lr = 0.1000 | loss: 6.3810 | error: 557.0000 - valid | validloss: 9.7003 | validerror: 674.0000 | s/iter: 0.3619
train | epoch = 9 | lr = 0.1000 | loss: 7.4764 | error: 559.0000 - valid | validloss: 6.6970 | validerror: 532.0000 | s/iter: 0.3609
train | epoch = 10 | lr = 0.1000 | loss: 5.7766 | error: 523.0000 - valid | validloss: 6.8547 | validerror: 546.0000 | s/iter: 0.3649
test | error: 543.0000
```

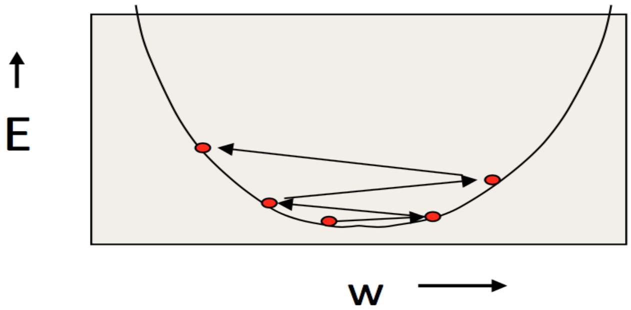
Learning Rate=10.0

Output:



```
Started training!
train | epoch = 1 | lr = 10.0000 | loss: 3037.5151 | error: 901.0000 - valid | validloss: 5228.1623 | validerror: 900.0000 | s/iter:
0.3725
train | epoch = 2 | lr = 10.0000 | loss: 3927.6720 | error: 906.0000 - valid | validloss: 5017.1181 | validerror: 795.0000 | s/iter:
0.3531
train | epoch = 3 | lr = 10.0000 | loss: 3842.2043 | error: 882.0000 - valid | validloss: 4554.0584 | validerror: 896.0000 | s/iter:
0.3558
train | epoch = 4 | lr = 10.0000 | loss: 4380.0806 | error: 905.0000 - valid | validloss: 4017.2982 | validerror: 813.0000 | s/iter:
0.3547
train | epoch = 5 | lr = 10.0000 | loss: 3584.1842 | error: 864.0000 - valid | validloss: 4413.3504 | validerror: 900.0000 | s/iter:
0.3566
train | epoch = 6 | lr = 10.0000 | loss: 4092.4456 | error: 885.0000 - valid | validloss: 4231.3322 | validerror: 900.0000 | s/iter:
0.3547
train | epoch = 7 | lr = 10.0000 | loss: 3697.0259 | error: 867.0000 - valid | validloss: 4069.8637 | validerror: 798.0000 | s/iter:
0.3541
train | epoch = 8 | lr = 10.0000 | loss: 3089.1885 | error: 819.0000 - valid | validloss: 3480.3622 | validerror: 900.0000 | s/iter:
0.3565
train | epoch = 9 | lr = 10.0000 | loss: 3710.1879 | error: 873.0000 - valid | validloss: 3149.9053 | validerror: 897.0000 | s/iter:
0.3547
train | epoch = 10 | lr = 10.0000 | loss: 3133.2058 | error: 840.0000 - valid | validloss: 3676.1565 | validerror: 727.0000 | s/iter:
0.3597    |
test | error: 722.0000
```

Since the learning rate is big, gradient descent is not able to converge to the minima (as the step size overshoots the minima and increases the loss) The above scenario can be explained by the following image:



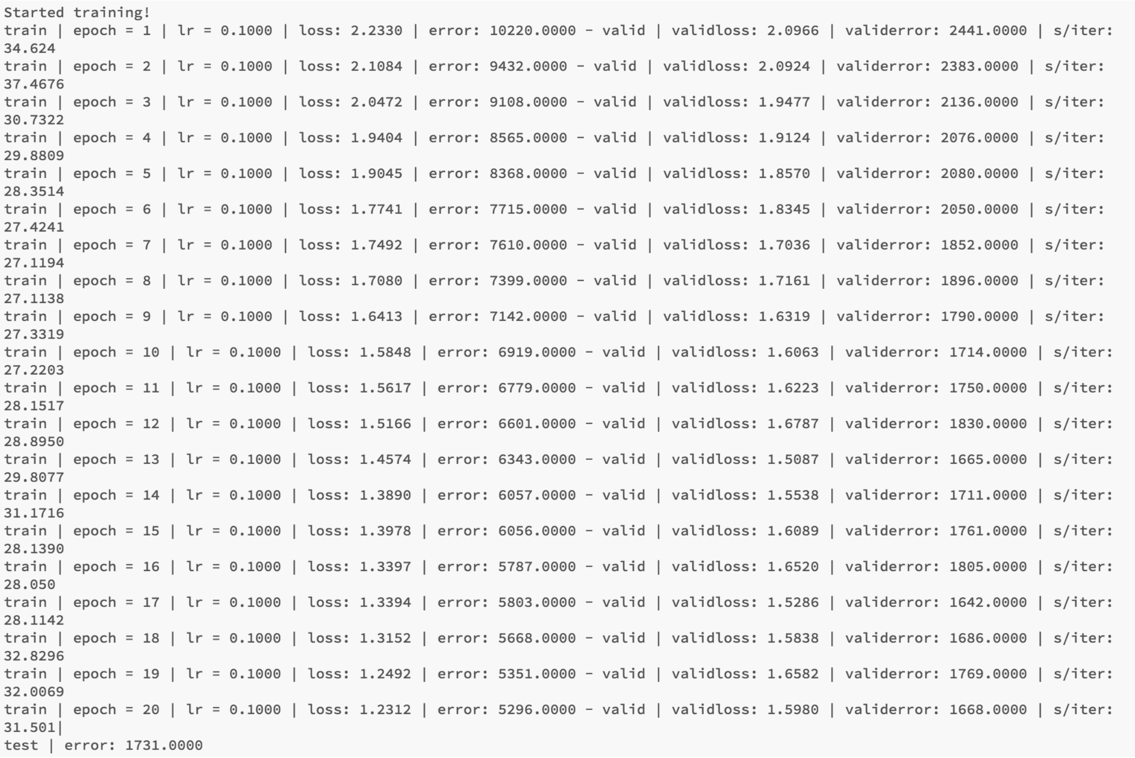
Therefore, we can see the training loss increase for learning rate=10
Since, the training loss is not minimized, the network is not optimally trained, resulting in higher test error(in comparison to learning rate=.1)

Solution 5

CNN:

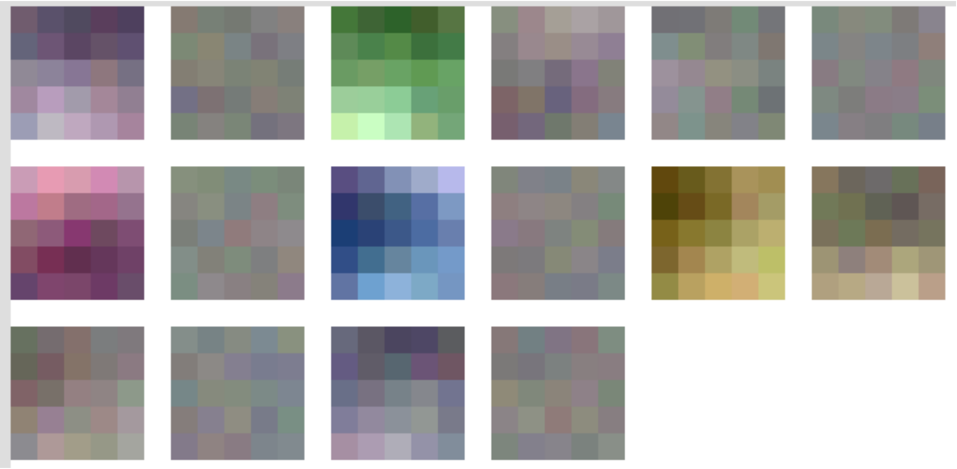
$$\begin{array}{l}
network = nn.Sequential() \\
network : add(nn.SpatialConvolution(3, 16, 5, 5)) \\
network : add(nn.Tanh()) \\
network : add(nn.SpatialMaxPooling(2, 2)) \\
network : add(nn.SpatialConvolution(16, 128, 5, 5)) \\
network : add(nn.Tanh()) \\
network : add(nn.SpatialMaxPooling(2, 2)) \\
network : add(nn.View(128 * 5 * 5)) \\
network : add(nn.Linear(128 * 5 * 5, 64)) \\
network : add(nn.Tanh()) \\
network : add(nn.Linear(64, 10))
\end{array}$$

Output:



```
Started training!
train | epoch = 1 | lr = 0.1000 | loss: 2.2330 | error: 10220.0000 - valid | validloss: 2.0966 | validerror: 2441.0000 | s/iter:
34.624
train | epoch = 2 | lr = 0.1000 | loss: 2.1084 | error: 9432.0000 - valid | validloss: 2.0924 | validerror: 2383.0000 | s/iter:
37.4676
train | epoch = 3 | lr = 0.1000 | loss: 2.0472 | error: 9108.0000 - valid | validloss: 1.9477 | validerror: 2136.0000 | s/iter:
30.7322
train | epoch = 4 | lr = 0.1000 | loss: 1.9404 | error: 8565.0000 - valid | validloss: 1.9124 | validerror: 2076.0000 | s/iter:
29.8809
train | epoch = 5 | lr = 0.1000 | loss: 1.9045 | error: 8368.0000 - valid | validloss: 1.8570 | validerror: 2080.0000 | s/iter:
28.3514
train | epoch = 6 | lr = 0.1000 | loss: 1.7741 | error: 7715.0000 - valid | validloss: 1.8345 | validerror: 2050.0000 | s/iter:
27.4241
train | epoch = 7 | lr = 0.1000 | loss: 1.7492 | error: 7610.0000 - valid | validloss: 1.7036 | validerror: 1852.0000 | s/iter:
27.1194
train | epoch = 8 | lr = 0.1000 | loss: 1.7080 | error: 7399.0000 - valid | validloss: 1.7161 | validerror: 1896.0000 | s/iter:
27.1138
train | epoch = 9 | lr = 0.1000 | loss: 1.6413 | error: 7142.0000 - valid | validloss: 1.6319 | validerror: 1790.0000 | s/iter:
27.3319
train | epoch = 10 | lr = 0.1000 | loss: 1.5848 | error: 6919.0000 - valid | validloss: 1.6063 | validerror: 1714.0000 | s/iter:
27.2203
train | epoch = 11 | lr = 0.1000 | loss: 1.5617 | error: 6779.0000 - valid | validloss: 1.6223 | validerror: 1750.0000 | s/iter:
28.1517
train | epoch = 12 | lr = 0.1000 | loss: 1.5166 | error: 6601.0000 - valid | validloss: 1.6787 | validerror: 1830.0000 | s/iter:
28.8950
train | epoch = 13 | lr = 0.1000 | loss: 1.4574 | error: 6343.0000 - valid | validloss: 1.5087 | validerror: 1665.0000 | s/iter:
29.8077
train | epoch = 14 | lr = 0.1000 | loss: 1.3890 | error: 6057.0000 - valid | validloss: 1.5538 | validerror: 1711.0000 | s/iter:
31.1716
train | epoch = 15 | lr = 0.1000 | loss: 1.3978 | error: 6056.0000 - valid | validloss: 1.6089 | validerror: 1761.0000 | s/iter:
28.1390
train | epoch = 16 | lr = 0.1000 | loss: 1.3397 | error: 5787.0000 - valid | validloss: 1.6520 | validerror: 1805.0000 | s/iter:
28.050
train | epoch = 17 | lr = 0.1000 | loss: 1.3394 | error: 5803.0000 - valid | validloss: 1.5286 | validerror: 1642.0000 | s/iter:
28.1142
train | epoch = 18 | lr = 0.1000 | loss: 1.3152 | error: 5668.0000 - valid | validloss: 1.5838 | validerror: 1686.0000 | s/iter:
32.8296
train | epoch = 19 | lr = 0.1000 | loss: 1.2492 | error: 5351.0000 - valid | validloss: 1.6582 | validerror: 1769.0000 | s/iter:
32.0069
train | epoch = 20 | lr = 0.1000 | loss: 1.2312 | error: 5296.0000 - valid | validloss: 1.5980 | validerror: 1668.0000 | s/iter:
31.501|
test | error: 1731.0000
```

First layer weights:



Solution 5(b)

Parameters for the CNN:

Layer 1 (convolution) = [(5x5x3 + 1(bias))x16]=1216

Layer 2 (Tanh()) = 0

Layer 3 (Pooling) = 0

Layer 4 (convolution)= [(5x5x16 + 1(bias))x128]=51328

Layer 5 (Tanh()) = 0

Layer 6 (Pooling) = 0

Layer 7 (flatten)= 0

Layer 8 (Linear)= [(5x5x128 + 1(bias))x64]=204864

Layer 9 (Tanh()) = 0

Layer 10 (Linear)= [(64 + 1(bias))x10]=650

Total Parameters=258,058