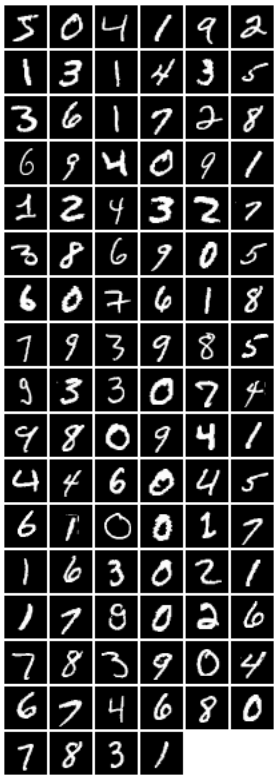
Computer Vision Assignment-2

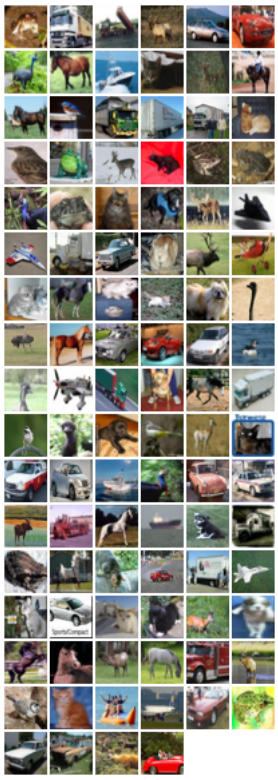
Abhinav Gupta (NetID-ag5799)

November 3, 2016

2. The images for the MNIST and CIFAR training data are as follows:



(a) MNIST-100



(b) CIFAR-100

3.(a) The output after training, validating and testing on 1000 examples.

```
train | epoch = 1 | lr = 0.1000 | loss: 29293.0637 | error: 668.0000 - valid | validloss: 10181.8040 | validerror: 419.0000 | s/iter: 0.2010
train | epoch = 2 | lr = 0.1000 | loss: 3712.8518 | error: 244.0000 - valid | validloss: 2766.7456 | validerror: 228.0000 | s/iter: 0.1829
train | epoch = 3 | lr = 0.1000 | loss: 1273.6265 | error: 133.0000 - valid | validloss: 2527.4130 | validerror: 226.0000 | s/iter: 0.1890
train | epoch = 4 | lr = 0.1000 | loss: 1601.3950 | error: 156.0000 - valid | validloss: 2233.3753 | validerror: 201.0000 | s/iter: 0.1794
train | epoch = 5 | lr = 0.1000 | loss: 713.4778 | error: 88.0000 - valid | validloss: 1746.5079 | validerror: 171.0000 | s/iter: 0.1855
train | epoch = 6 | lr = 0.1000 | loss: 527.5201 | error: 86.0000 - valid | validloss: 1993.9952 | validerror: 211.0000 | s/iter: 0.1890
train | epoch = 7 | lr = 0.1000 | loss: 464.6825 | error: 92.0000 - valid | validloss: 1802.6563 | validerror: 193.0000 | s/iter: 0.1854
train | epoch = 8 | lr = 0.1000 | loss: 238.6802 | error: 54.0000 - valid | validloss: 1765.2389 | validerror: 198.0000 | s/iter: 0.1821
train | epoch = 9 | lr = 0.1000 | loss: 746.3244 | error: 100.0000 - valid | validloss: 3751.9471 | validerror: 274.0000 | s/iter: 0.1822
train | epoch = 10 | lr = 0.1000 | loss: 354.2497 | error: 62.0000 - valid | validloss: 1854.9411 | validerror: 195.0000 | s/iter: 0.1804
| test | error: 191.0000
```

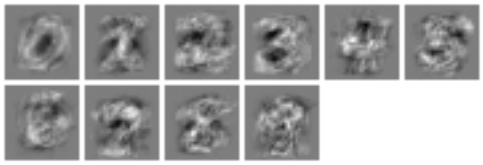
Figure 2

(b) The output after training on 50 examples and using the full validation and test set.

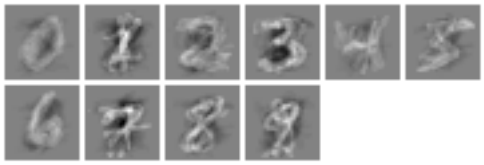
```
train | epoch = 1 | lr = 0.1000 | loss: 64.8796 | error: 43.0000 - valid | validloss: 20685.1468 | validerror: 6954.0000 | s/iter: 0.9792
train | epoch = 2 | lr = 0.1000 | loss: 14671.3541 | error: 26.0000 - valid | validloss: 22151.6520 | validerror: 6878.0000 | s/iter: 0.9875
train | epoch = 3 | lr = 0.1000 | loss: 19053.0497 | error: 28.0000 - valid | validloss: 20575.2377 | validerror: 5673.0000 | s/iter: 0.9006
train | epoch = 4 | lr = 0.1000 | loss: 12373.6498 | error: 18.0000 - valid | validloss: 16194.0607 | validerror: 5246.0000 | s/iter: 0.9659
train | epoch = 5 | lr = 0.1000 | loss: 9386.8868 | error: 12.0000 - valid | validloss: 15390.5190 | validerror: 6269.0000 | s/iter: 0.9032
train | epoch = 6 | lr = 0.1000 | loss: 7582.5734 | error: 22.0000 - valid | validloss: 25863.8462 | validerror: 7828.0000 | s/iter: 0.8917
train | epoch = 7 | lr = 0.1000 | loss: 14390.5032 | error: 24.0000 - valid | validloss: 18583.1501 | validerror: 6385.0000 | s/iter: 0.8922
train | epoch = 8 | lr = 0.1000 | loss: 14692.9140 | error: 20.0000 - valid | validloss: 4741.7797 | validerror: 4068.0000 | s/iter: 0.8963
train | epoch = 9 | lr = 0.1000 | loss: 188.3231 | error: 1.0000 - valid | validloss: 4642.1204 | validerror: 3873.0000 | s/iter: 0.8996
train | epoch = 10 | lr = 0.1000 | loss: 0.0000 | error: 0.0000 - valid | validloss: 4642.1204 | validerror: 3873.0000 | s/iter: 0.8960
| test | error: 4095.0000
```

Figure 3

Visualization of the network weights for both models:



(a) With 1000 training examples



(b) With 50 training examples

```
train | epoch = 1 | lr = 0.1000 | loss: 73.5349 | error: 47.0000 - valid | validloss: 27484.5521 | validerror: 895.0000 | s/iter: 0.1747
train | epoch = 2 | lr = 0.1000 | loss: 22035.7225 | error: 42.0000 - valid | validloss: 35853.7824 | validerror: 749.0000 | s/iter: 0.1443
train | epoch = 3 | lr = 0.1000 | loss: 40857.0383 | error: 29.0000 - valid | validloss: 27193.6502 | validerror: 602.0000 | s/iter: 0.1237
train | epoch = 4 | lr = 0.1000 | loss: 29051.9468 | error: 22.0000 - valid | validloss: 18001.1229 | validerror: 695.0000 | s/iter: 0.1164
train | epoch = 5 | lr = 0.1000 | loss: 12262.9761 | error: 21.0000 - valid | validloss: 10213.0758 | validerror: 536.0000 | s/iter: 0.1145
train | epoch = 6 | lr = 0.1000 | loss: 1500.9622 | error: 7.0000 - valid | validloss: 7618.1303 | validerror: 468.0000 | s/iter: 0.1500
train | epoch = 7 | lr = 0.1000 | loss: 644.9726 | error: 6.0000 - valid | validloss: 7345.0219 | validerror: 483.0000 | s/iter: 0.1533
train | epoch = 8 | lr = 0.1000 | loss: 18.0922 | error: 2.0000 - valid | validloss: 5935.0690 | validerror: 443.0000 | s/iter: 0.1334
train | epoch = 9 | lr = 0.1000 | loss: 0.0000 | error: 0.0000 - valid | validloss: 5935.0690 | validerror: 443.0000 | s/iter: 0.1152
train | epoch = 10 | lr = 0.1000 | loss: 0.0000 | error: 0.0000 - valid | validloss: 5935.0690 | validerror: 443.0000 | s/iter: 0.1176
| test | error: 449.0000
```

Figure 5: Training on 50 examples and using 1000 instances in validation and test set.

Training with just 50 examples leads to overfitting. The model converges very early as it has very less features to learn. On the other hand, the valid and test error are huge, since the model does not generalize to different inputs. As we see that the test error in case of 1000 instances is less than using the full test set since the model is able to generalize to some instances present in the smaller test set.

4.(a)

```
train | epoch = 1 | lr = 0.1000 | loss: 5.5752 | error: 698.0000 - valid | validloss: 7.7601 | validerror: 548.0000 | s/iter: 0.4165
train | epoch = 2 | lr = 0.1000 | loss: 9.9345 | error: 709.0000 - valid | validloss: 13.7427 | validerror: 660.0000 | s/iter: 0.3681
train | epoch = 3 | lr = 0.1000 | loss: 12.0517 | error: 695.0000 - valid | validloss: 13.0474 | validerror: 813.0000 | s/iter: 0.4952
train | epoch = 4 | lr = 0.1000 | loss: 12.0060 | error: 694.0000 - valid | validloss: 9.1808 | validerror: 849.0000 | s/iter: 0.4278
train | epoch = 5 | lr = 0.1000 | loss: 11.2823 | error: 653.0000 - valid | validloss: 5.7648 | validerror: 575.0000 | s/iter: 0.4284
train | epoch = 6 | lr = 0.1000 | loss: 8.3635 | error: 613.0000 - valid | validloss: 9.6513 | validerror: 685.0000 | s/iter: 0.3523
train | epoch = 7 | lr = 0.1000 | loss: 8.6425 | error: 635.0000 - valid | validloss: 10.3998 | validerror: 557.0000 | s/iter: 0.3504
train | epoch = 8 | lr = 0.1000 | loss: 6.7254 | error: 552.0000 - valid | validloss: 8.9408 | validerror: 760.0000 | s/iter: 0.3328
train | epoch = 9 | lr = 0.1000 | loss: 9.8200 | error: 623.0000 - valid | validloss: 12.0858 | validerror: 618.0000 | s/iter: 0.3492
train | epoch = 10 | lr = 0.1000 | loss: 9.8584 | error: 621.0000 - valid | validloss: 7.9613 | validerror: 561.0000 | s/iter: 0.3476
| test | error: 517.0000
```

Figure 6

(b)

```
train | epoch = 1 | lr = 10.0000 | loss: 3139.2579 | error: 909.0000 - valid | validloss: 6168.4628 | validerror: 893.0000 | s/iter: 0.3713
train | epoch = 2 | lr = 10.0000 | loss: 3969.0109 | error: 888.0000 - valid | validloss: 5060.6471 | validerror: 893.0000 | s/iter: 0.3302
train | epoch = 3 | lr = 10.0000 | loss: 3671.6374 | error: 883.0000 - valid | validloss: 3582.3952 | validerror: 860.0000 | s/iter: 0.3334
train | epoch = 4 | lr = 10.0000 | loss: 3246.8756 | error: 846.0000 - valid | validloss: 4519.9848 | validerror: 896.0000 | s/iter: 0.3614
train | epoch = 5 | lr = 10.0000 | loss: 3840.3475 | error: 883.0000 - valid | validloss: 3816.6018 | validerror: 861.0000 | s/iter: 0.3451
train | epoch = 6 | lr = 10.0000 | loss: 3966.5387 | error: 885.0000 - valid | validloss: 4403.5726 | validerror: 893.0000 | s/iter: 0.4450
train | epoch = 7 | lr = 10.0000 | loss: 3445.2636 | error: 850.0000 - valid | validloss: 3715.3545 | validerror: 818.0000 | s/iter: 0.3811
train | epoch = 8 | lr = 10.0000 | loss: 3907.3177 | error: 887.0000 - valid | validloss: 3977.0998 | validerror: 895.0000 | s/iter: 0.3700
train | epoch = 9 | lr = 10.0000 | loss: 3435.8520 | error: 885.0000 - valid | validloss: 2674.8150 | validerror: 837.0000 | s/iter: 0.3752
train | epoch = 10 | lr = 10.0000 | loss: 2420.2164 | error: 783.0000 - valid | validloss: 2469.2453 | validerror: 746.0000 | s/iter: 0.3648
| test | error: 745.0000
```

Figure 7

Due to high learning rate (10), the model keeps on jumping back and forth in the "well" by taking large steps and it will take very far more epochs to converge near the minima. If we compare the answer with (a) where the learning rate is 0.1, we see that it gradually progress towards the minima by taking small steps.

5.(a)

```
train | epoch = 1 | lr = 0.1000 | loss: 2.2465 | error: 10378.0000 - valid | validloss: 2.1111 | validerror: 2294.0000 | s/iter: 33.8550
train | epoch = 2 | lr = 0.1000 | loss: 2.0919 | error: 9353.0000 - valid | validloss: 2.1079 | validerror: 2388.0000 | s/iter: 36.6925
train | epoch = 3 | lr = 0.1000 | loss: 2.0346 | error: 9157.0000 - valid | validloss: 1.9103 | validerror: 2136.0000 | s/iter: 33.2089
train | epoch = 4 | lr = 0.1000 | loss: 1.9257 | error: 8463.0000 - valid | validloss: 1.7753 | validerror: 1955.0000 | s/iter: 32.0502
train | epoch = 5 | lr = 0.1000 | loss: 1.8210 | error: 7948.0000 - valid | validloss: 1.7924 | validerror: 1973.0000 | s/iter: 27.6897
train | epoch = 6 | lr = 0.1000 | loss: 1.8046 | error: 7880.0000 - valid | validloss: 1.7019 | validerror: 1899.0000 | s/iter: 27.6299
train | epoch = 7 | lr = 0.1000 | loss: 1.7199 | error: 7563.0000 - valid | validloss: 1.6981 | validerror: 1868.0000 | s/iter: 28.3324
train | epoch = 8 | lr = 0.1000 | loss: 1.6815 | error: 7426.0000 - valid | validloss: 1.6792 | validerror: 1849.0000 | s/iter: 27.4099
train | epoch = 9 | lr = 0.1000 | loss: 1.6296 | error: 7111.0000 - valid | validloss: 1.6116 | validerror: 1791.0000 | s/iter: 32.0081
train | epoch = 10 | lr = 0.1000 | loss: 1.5846 | error: 6908.0000 - valid | validloss: 1.5687 | validerror: 1727.0000 | s/iter: 26.6563
train | epoch = 11 | lr = 0.1000 | loss: 1.5979 | error: 6947.0000 - valid | validloss: 1.6024 | validerror: 1761.0000 | s/iter: 26.3530
train | epoch = 12 | lr = 0.1000 | loss: 1.5056 | error: 6591.0000 - valid | validloss: 1.5264 | validerror: 1702.0000 | s/iter: 26.6905
train | epoch = 13 | lr = 0.1000 | loss: 1.4802 | error: 6424.0000 - valid | validloss: 1.5710 | validerror: 1726.0000 | s/iter: 26.4441
train | epoch = 14 | lr = 0.1000 | loss: 1.4580 | error: 6359.0000 - valid | validloss: 1.5500 | validerror: 1701.0000 | s/iter: 26.5862
train | epoch = 15 | lr = 0.1000 | loss: 1.4465 | error: 6298.0000 - valid | validloss: 1.5099 | validerror: 1637.0000 | s/iter: 26.7205
train | epoch = 16 | lr = 0.1000 | loss: 1.4242 | error: 6187.0000 - valid | validloss: 1.5105 | validerror: 1680.0000 | s/iter: 26.9464
train | epoch = 17 | lr = 0.1000 | loss: 1.4533 | error: 6282.0000 - valid | validloss: 1.5728 | validerror: 1724.0000 | s/iter: 26.4836
train | epoch = 18 | lr = 0.1000 | loss: 1.4234 | error: 6176.0000 - valid | validloss: 1.5235 | validerror: 1673.0000 | s/iter: 26.8015
train | epoch = 19 | lr = 0.1000 | loss: 1.3263 | error: 5714.0000 - valid | validloss: 1.4950 | validerror: 1636.0000 | s/iter: 25.5398
train | epoch = 20 | lr = 0.1000 | loss: 1.2868 | error: 5519.0000 - valid | validloss: 1.5082 | validerror: 1665.0000 | s/iter: 25.3900
| test | error: 1634.0000
```

Figure 8

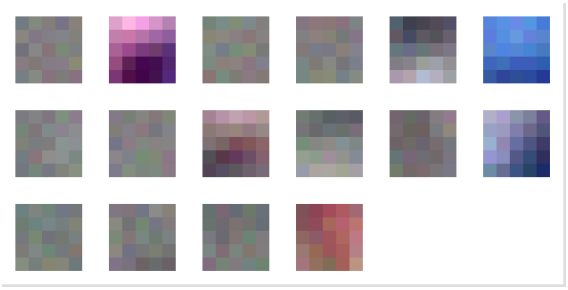


Figure 9: Image of first layer filters

(b) The parameters of the model are as follows:

1. Convolutional Layer (16((5x5x3)+1)) for 16 filters of 5x5x3 and 1 bias for each = 1216
2. NonLinearity Layer-Tanh (No parameters) = 0
3. MaxPooling Layer (No parameters) = 0
4. Convolutional Layer (128((5x5x16)+1)) for 128 filters of 5x5x16 and 1 bias for each = 51328

5. NonLinearity Layer-Tanh (No parameters) = 0
6. MaxPooling Layer (No parameters) = 0
7. Linear Layer (64((128x5x5)+1)) for 3200 units connected to 64 units each output having 1 bias = 204864
8. Linear Layer (10(64+1)) for 64 units connected to 10 units each output having 1 bias = 650
So, total number of parameters = 258058