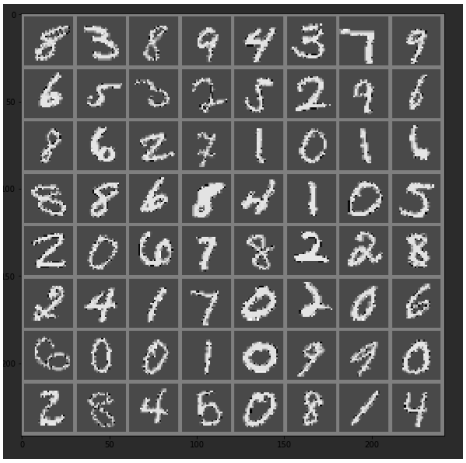

Figure 1: Mnist Visualization

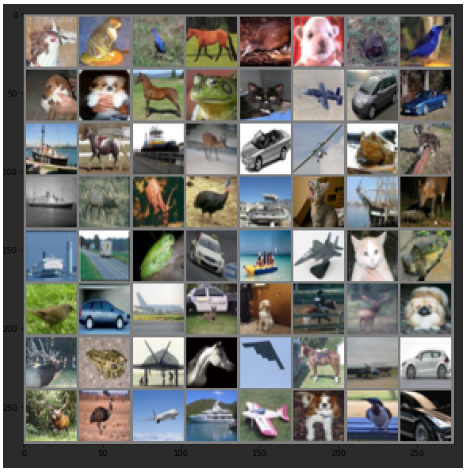

Figure 2: Cifar Visualization

Figure 3: Linear Network Weights: 10 by 874

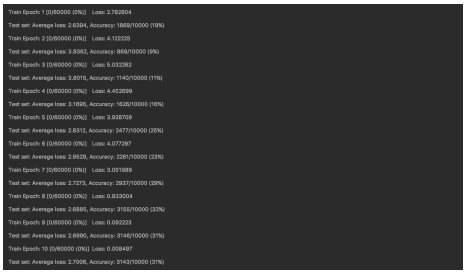

Figure 4: Small Dataset(50): We see a huge difference between the trainning accuracy and test accuracy as a result of overfitting on a small dataset. The test accuracy doesnt decrease because learning via backpropagation require lots of data.

```
Train Epoch: 7 [0/60000 (0%)]   Loss: 0.148668
Train Epoch: 7 [6400/60000 (11%)]       Loss: 0.173039
Train Epoch: 7 [12800/60000 (21%)]      Loss: 0.237750
Train Epoch: 7 [19200/60000 (32%)]      Loss: 0.199186
Train Epoch: 7 [25600/60000 (43%)]      Loss: 0.224152
Train Epoch: 7 [32000/60000 (53%)]      Loss: 0.176284
Train Epoch: 7 [38400/60000 (64%)]      Loss: 0.238162
Train Epoch: 7 [44800/60000 (75%)]      Loss: 0.180425
Train Epoch: 7 [51200/60000 (85%)]      Loss: 0.185717
Train Epoch: 7 [57600/60000 (96%)]      Loss: 0.107280

Test set: Average loss: 0.1946, Accuracy: 9442/10000 (94%)

Train Epoch: 8 [0/60000 (0%)]   Loss: 0.126643
Train Epoch: 8 [6400/60000 (11%)]       Loss: 0.152318
Train Epoch: 8 [12800/60000 (21%)]      Loss: 0.086708
Train Epoch: 8 [19200/60000 (32%)]      Loss: 0.109821
Train Epoch: 8 [25600/60000 (43%)]      Loss: 0.169002
Train Epoch: 8 [32000/60000 (53%)]      Loss: 0.209483
Train Epoch: 8 [38400/60000 (64%)]      Loss: 0.227308
Train Epoch: 8 [44800/60000 (75%)]      Loss: 0.136739
Train Epoch: 8 [51200/60000 (85%)]      Loss: 0.190427
Train Epoch: 8 [57600/60000 (96%)]      Loss: 0.141736

Test set: Average loss: 0.1840, Accuracy: 9483/10000 (95%)

Train Epoch: 9 [0/60000 (0%)]   Loss: 0.233190
Train Epoch: 9 [6400/60000 (11%)]       Loss: 0.137709
Train Epoch: 9 [12800/60000 (21%)]      Loss: 0.186599
Train Epoch: 9 [19200/60000 (32%)]      Loss: 0.113850
Train Epoch: 9 [25600/60000 (43%)]      Loss: 0.156787
Train Epoch: 9 [32000/60000 (53%)]      Loss: 0.234822
Train Epoch: 9 [38400/60000 (64%)]      Loss: 0.183888
Train Epoch: 9 [44800/60000 (75%)]      Loss: 0.308970
Train Epoch: 9 [51200/60000 (85%)]      Loss: 0.100252
Train Epoch: 9 [57600/60000 (96%)]      Loss: 0.240479

Test set: Average loss: 0.1715, Accuracy: 9513/10000 (95%)

Train Epoch: 10 [0/60000 (0%)]  Loss: 0.133540
Train Epoch: 10 [6400/60000 (11%)]      Loss: 0.393999
Train Epoch: 10 [12800/60000 (21%)]     Loss: 0.153392
Train Epoch: 10 [19200/60000 (32%)]     Loss: 0.302003
Train Epoch: 10 [25600/60000 (43%)]     Loss: 0.165758
Train Epoch: 10 [32000/60000 (53%)]     Loss: 0.170846
Train Epoch: 10 [38400/60000 (64%)]     Loss: 0.050763
Train Epoch: 10 [44800/60000 (75%)]     Loss: 0.115670
Train Epoch: 10 [51200/60000 (85%)]     Loss: 0.079023
Train Epoch: 10 [57600/60000 (96%)]     Loss: 0.092462

Test set: Average loss: 0.1631, Accuracy: 9538/10000 (95%)
```

Figure 5: Multi-layer on Mnist

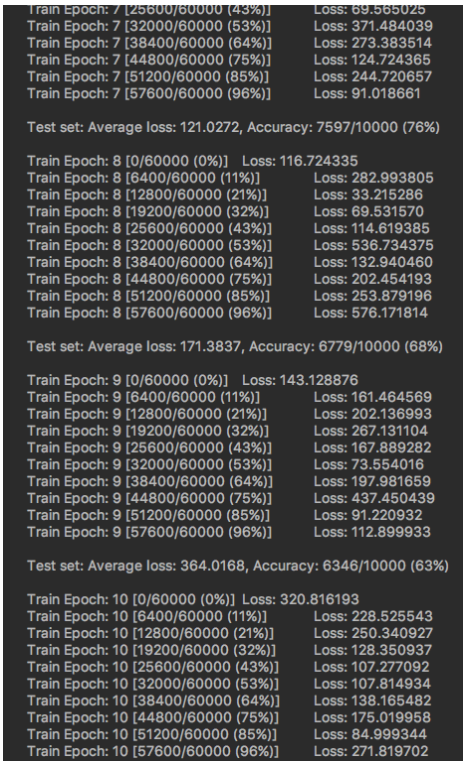

Figure 6: As a result of changing the learning rate, When setting learning rate too hight we get a classic divergence scenario whereby the loss function grows very high and we dont see any improvement in the test set

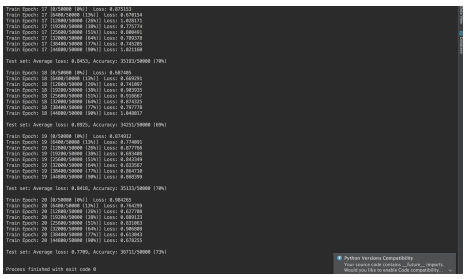

Figure 7: Cifar convolution Network

Figure 8: First Layer filter of the convolution Network

```
over all number of parameters are 8
network structure is Net (
  (conv1): Conv2d(3, 16, kernel_size=(5, 5), stride=(1, 1))
  (pool1): MaxPool2d (size=(2, 2), stride=(2, 2), dilation=(1, 1))
  (conv2): Conv2d(16, 128, kernel_size=(5, 5), stride=(1, 1))
  (pool2): MaxPool2d (size=(2, 2), stride=(2, 2), dilation=(1, 1))
  (linear1): Linear (3200 -> 64)
  (linear2): Linear (64 -> 10)
)
0 th parameter has a length of torch.Size([16, 3, 5, 5])
1 th parameter has a length of torch.Size([16])
2 th parameter has a length of torch.Size([128, 16, 5, 5])
3 th parameter has a length of torch.Size([128])
4 th parameter has a length of torch.Size([64, 3200])
5 th parameter has a length of torch.Size([64])
6 th parameter has a length of torch.Size([10, 64])
7 th parameter has a length of torch.Size([10])
```

Figure 9: Parameters