CMPT 412 Project 2 Report

Ritika Goyal (301401516)

I am using one lay day for assignment2

Part 1: Improving BaseNet on CIFAR100 (Kaggle submission under name RitikaGoyal)

Final Network

I used the reference from AlexNet model and the twerked different parameters to get accuracy on validation images around 63% with 64.3% score on Kaggle.

I referred to this link:
https://www.cs.toronto.edu/~lczhang/aps360_20191/lec/w03/convnet.html

The structure of final network is described in the table below:

Layer No.	Layer Type	Kernel size (for conv layers)	Input | Output dimension	Input | Output Channels (for conv layers)
1	Conv2d	3	32|32	3|6
2	Batchnorm2d	-	32|32	-
3	Relu	-	32|32	-
4	Conv2d	3	32|32	6|16
5	Batchnorm2d	-	32|32	-
6	Relu	-	32|32	-
7	Conv2d	3	32|32	16|32
8	Batchnorm2d	-	32|32	-
9	Relu	-	32|32	-
10	Maxpool2d	2	32|16	-
11	Conv2d	3	16|16	32|64
12	Batchnorm2d	-	16|16	-
13	Relu	-	16|16	-
14	Conv2d	3	16|16	64|192
15	Batchnorm2d	-	16|16	-
16	Relu	-	16|16	-
17	Conv2d	3	16|16	192|384

18	Batchnorm2d	-	16|16	-
19	Relu	-	16|16	-
20	Maxpool2d	2	16|8	-
21	Conv2d	3	8|8	384|256
22	Batchnorm2d	-	8|8	-
23	Relu	-	8|8	-
24	Conv2d	3	8|8	256|256
25	Batchnorm2d	-	8|8	-
26	Relu	-	8|8	-
27	Maxpool2d	2	8|4	-
28	Linear	-	4096|2048	-
29	Relu	-	2048|2048	-
30	Linear	-	2048|1024	-
31	Relu	-	1024|1024	-
32	Dropout	-	1024|1024	-
33	Linear	-	1024|512	-
34	Relu	-	512|512	-
35	Linear	-	512|256	-
36	Relu	-	256|256	-
37	Liner	-	256|100	-

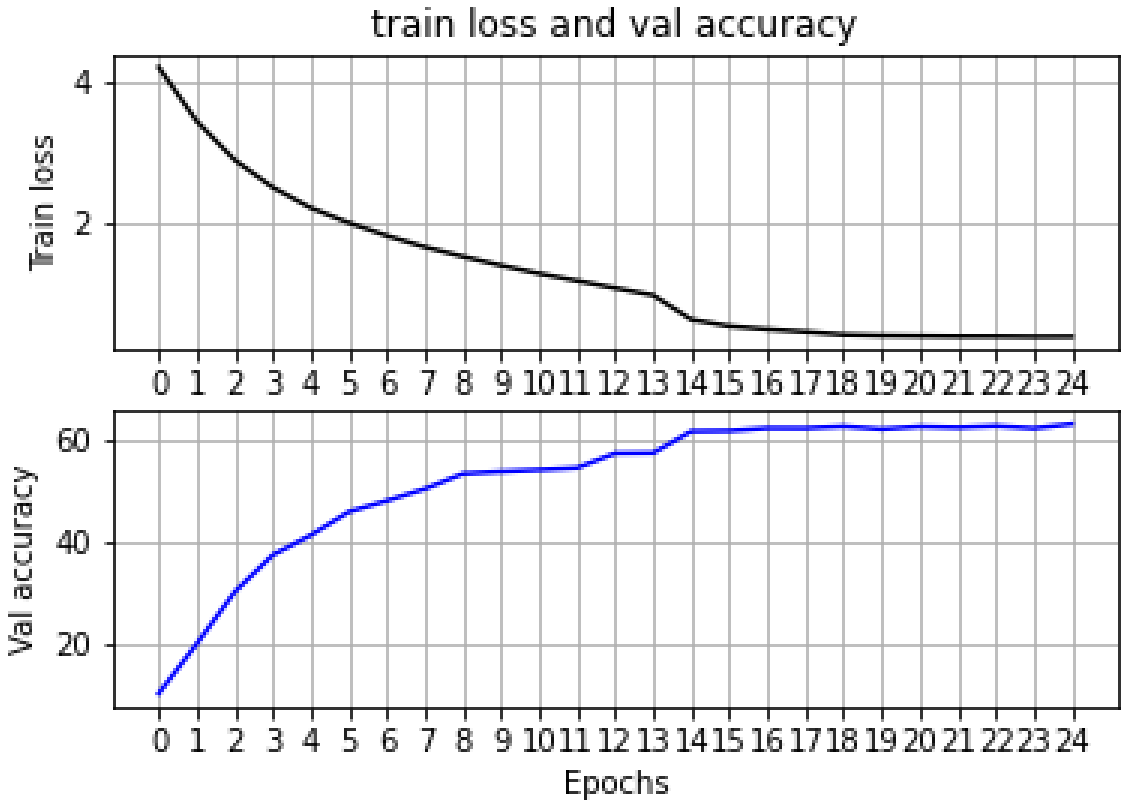
- The data was transformed to make it normal with zero mean and one standard deviation. The data was transformed with different parameters such as RandomHorizontalFlip() to make it more robust.

- After the transformation, the different convolution layers were added to make the network more deeper and to improve the training accuracy. After the convolutional layer, many fully connected layers were added.

- With all these changes, the accuracy also improved when the number of epochs were changed from 15 to 25.

- After changing epochs, different learning rate were also tested to improve the network and best outcome came with learning rate of 0.0035 and I applied schedular as well which reduces learning rate after certain epochs.

Plot illustrating the training loss and the validation accuracy:

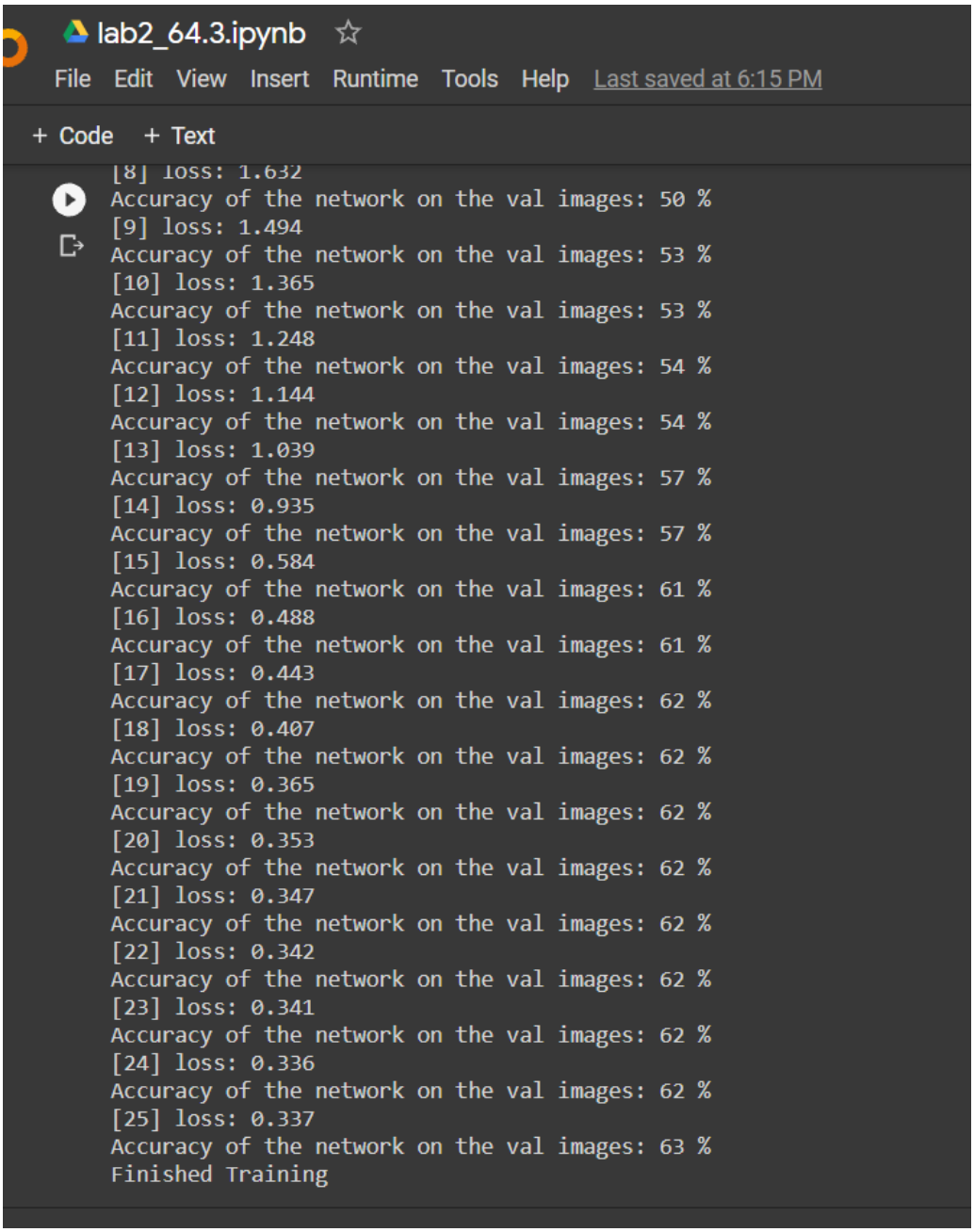


Ablation study:

With the BaseNet that was mentioned earlier, the maximum accuracy achieved was around 26%. After adding convolution layer and changing number of epochs, the achieved accuracy was 51%. However, after trying many variations of different epochs and convolutional layer parameters, around 63% accuracy was achieved when learning rate was changed. The learning rate was changed from 0.005 to 0.0035 which improved the accuracy. Additionally, the learning rate is decreased more after certain epochs by using scheduler.

Base performance:

The screenshot of the final accuracy of the network on the validation image is mentioned below:

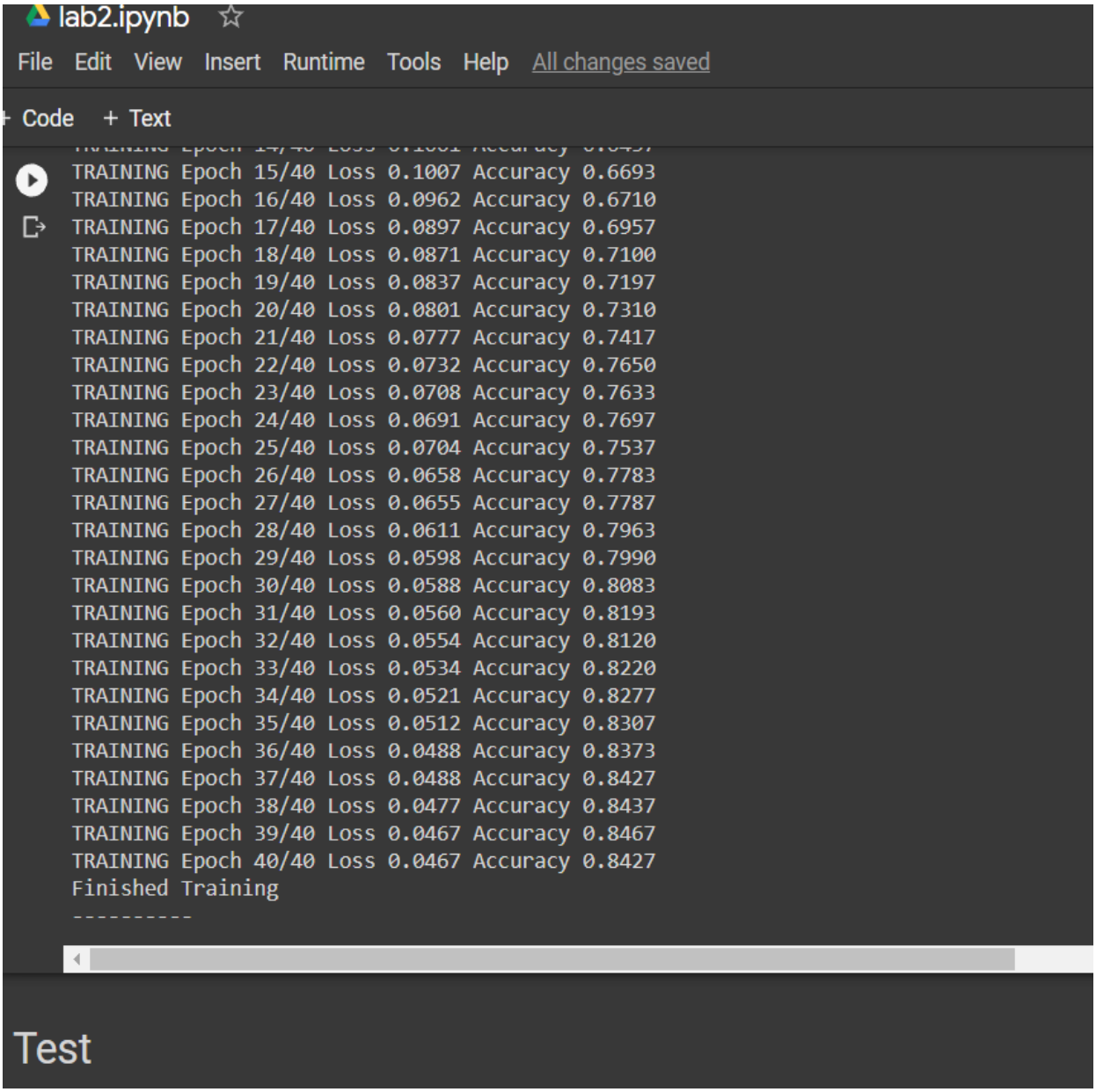


Relative performance:

The csv file is submitted on Kaggle under the name RitikaGoyal with the score 0.64300.

Part2: Transfer Learning

The screenshot of training accuracy is given below:



The screenshot of test accuracy is given below:

```
            loss = criterion(outputs, labels)

            _, preds = torch.max(outputs.data, 1)

            test_loss += loss.item()
            test_acc += torch.sum(preds == labels).item()

        test_loss /= (dataset_sizes['test']*repeats)
        test_acc /= (dataset_sizes['test']*repeats)

        print('Test Loss: %.4f Test Accuracy %.4f' % (test_loss, test_acc))
```

```
[ ]  test(model, criterion)
```

```
/usr/local/lib/python3.7/dist-packages/torch/utils/data/dataloader.py:481: UserWarning: This DataLoad
  cpuset_checked))
Test Loss: 0.1017 Test Accuracy 0.5753
```

Hyperparameter settings that were used:

- Number of epochs were changed to 40.
- Learning rate was changed to 0.001
- Batch size was increased to 16
- RESNET_LAST_ONLY was changed to false.