High-Level Computer Vision
Summer Semester 2019

Report 3: Convolutional Networks

Group Members:
Chirag Bhuvaneshwara - 2571703 - chiragbhuvaneshwar@gmail.com
Jyotsna Singh - 2576744 - s8jysing@stud.uni-saarland.de
Jyothsna Shashikumar Sastry - 2571729 - s8jysast@stud.uni-saarland.de

###
NOTE:

For better experimentation and reproducibility:

1. ***We had to set some random seeds in numpy and torch.***
2. ***We also had to use Xavier weights initialization for the Conv layers to ensure that the loss was decreasing.***

The flags we have set for automatically switching between different questions are:

	norm_layer	**dropoutVals**	**early_stopping**	**visualize_filters**	**data_aug**
Q1 a, b, c	None	[None]	False	True	False
Q2 a	True	[None]	False	True	False
Q2 b	True	[None]	True	False	False
Q3 a	None or True	[None]	False	False	True
Q3 b	None orTrue	[i/10 for i in range(1,10)]	False	False	False

###

Question 1: Implement Convolutional Network

a. **Report the training and validation accuracies**

   Validataion accuracy is: 79.3 %

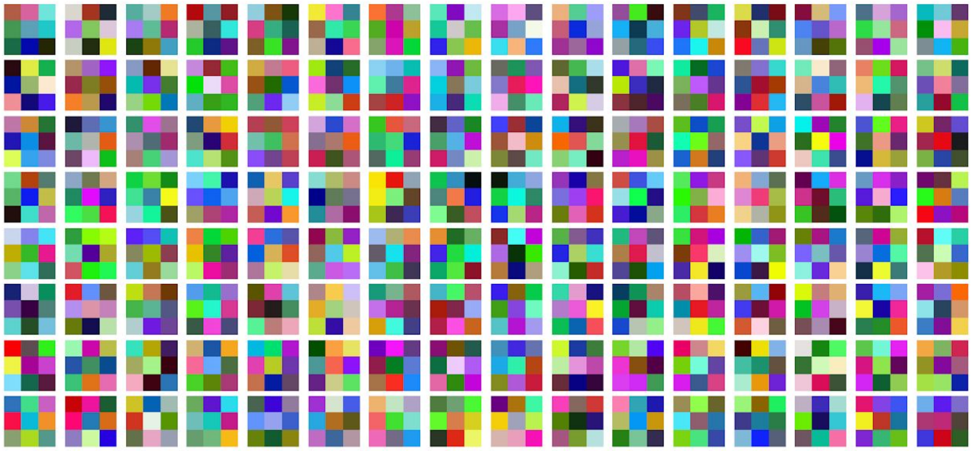
   Accuracy of the best network on the 1000 test images: 80.0 %

b. **Report the number of parameters for the described model**

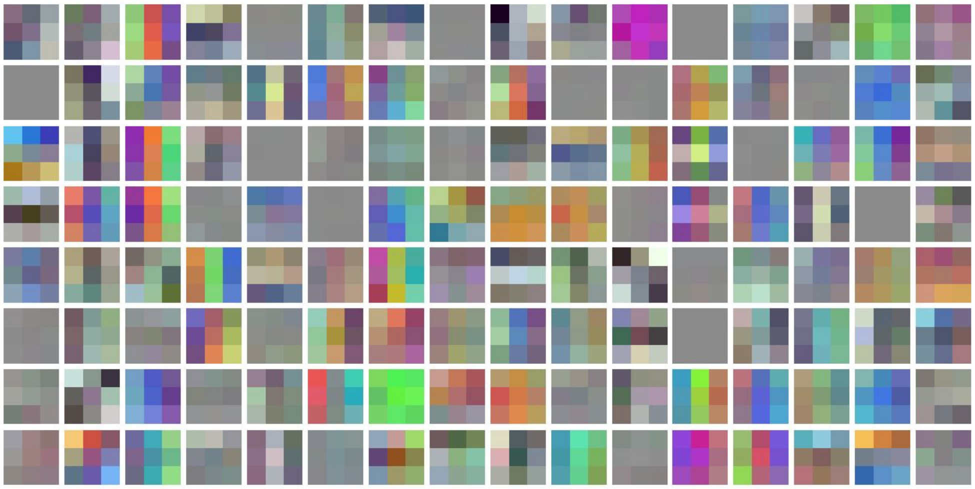
   The model has 7941130 trainable parameters

c. **Compare the filters before and after training. Do you see any patterns**

   Filters before training:



Filters After Training: As seen below, the filters after training are significantly different from the random filters before training. The filters after training either fire for a particular color like pink or for a certain combination of a number of colors. Although some of the filters just completely gray which probably suggests that those filters are looking at the average value.



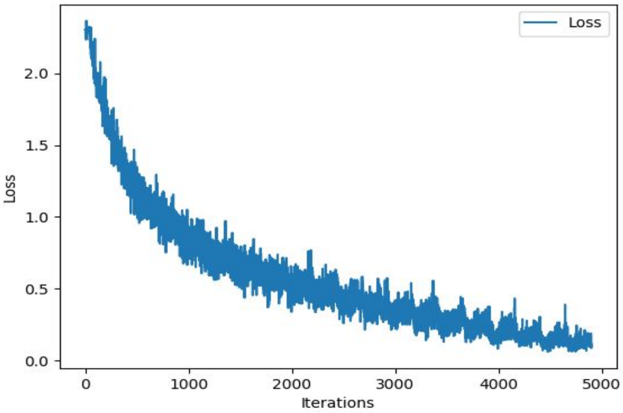
Question 2: Improve training of Convolutional Networks

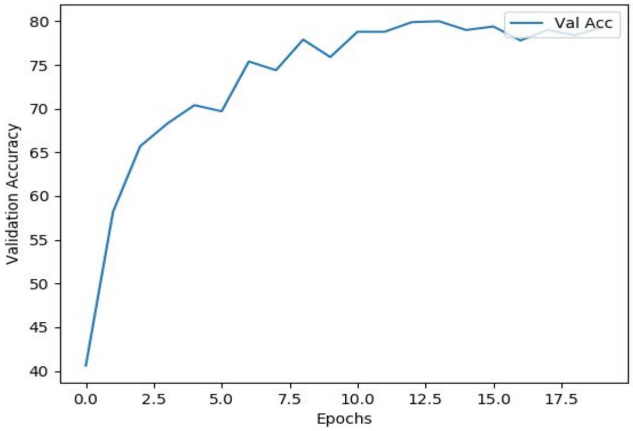
a. **Compare the loss curves and accuracy using batch normalization to its counterpart in Q1.a**

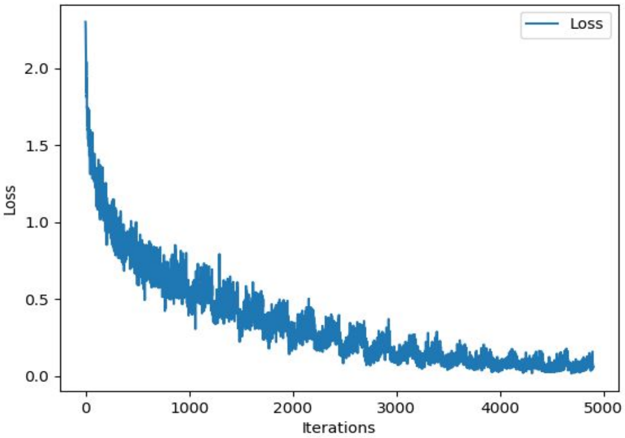
   Validataion accuracy is: 82.7 %

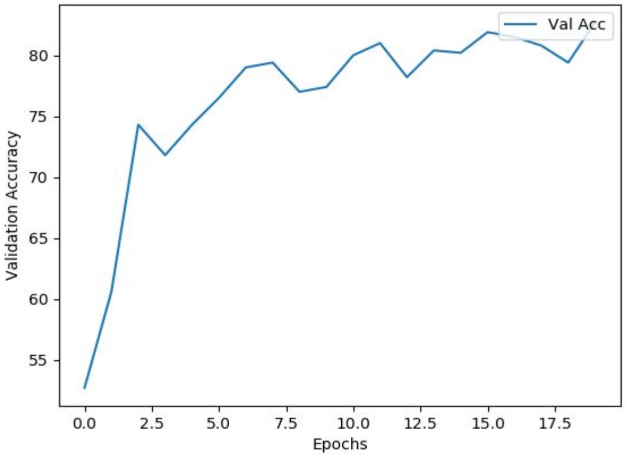
   Accuracy of the best network on the 1000 test images: 78.4 %

   The loss decreases sharply in the presence of batch normalization

<table>
<tr><th>Q1.a Loss</th><th>Q1.a Accuracy</th></tr>
<tr><td>


</td><td>


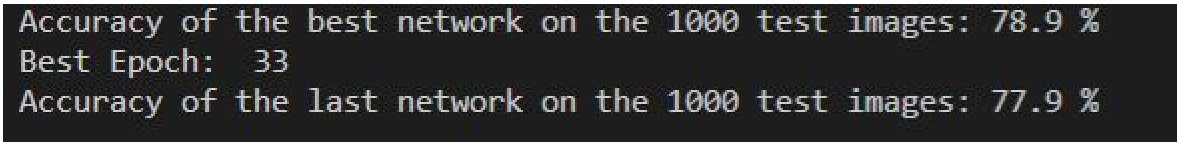
</td></tr>
<tr><th>Q2.a Loss</th><th>Q2.a Accuracy</th></tr>
<tr><td>


</td><td>


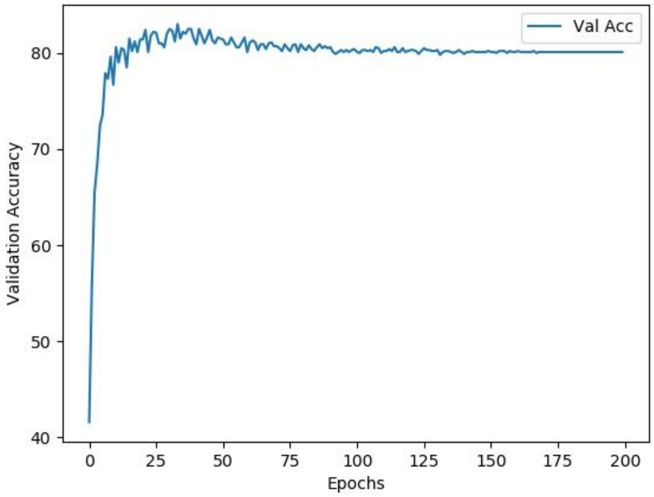
</td></tr>
</table>

b. **The best model and latest model on the training set**

The model was trained for 200 epochs. From the curve, we see that the validation accuracy reaches a maximum for the 33rd epoch. Therefore, with early stopping, the training can be stopped after the 33rd epoch in order to not overtrain the model.



```
Accuracy of the best network on the 1000 test images: 78.9 %
Best Epoch:  33
Accuracy of the last network on the 1000 test images: 77.9 %
```



Question 3: Improve generalization of Convolutional Networks

a. **Discuss which augmentations works well for you in the report**

   We used the following list of transformations.

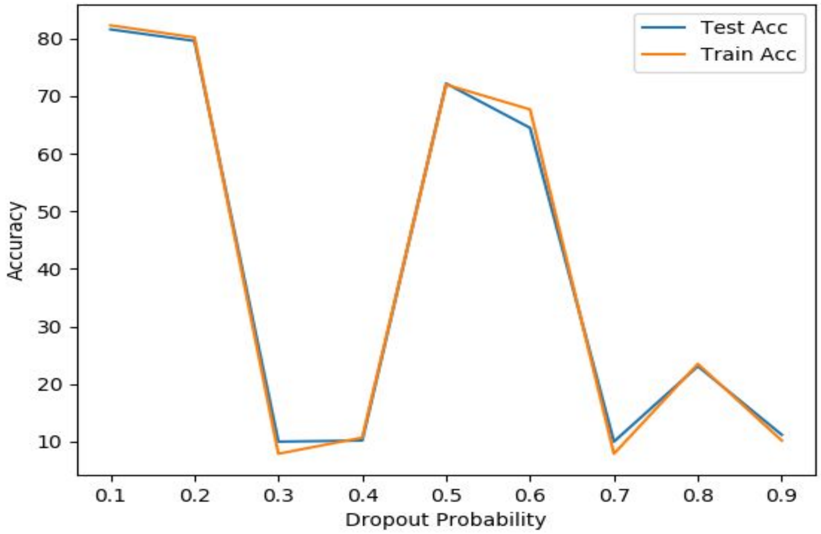
```
[
RandomCrop(32, padding=4),
RandomHorizontalFlip(),
RandomRotation(30),
RandomAffine(degrees=0, translate=(.3,.7)),
ColorJitter(
        brightness=float(0.1*np.random.rand(1)),
        contrast=float(0.1*np.random.rand(1)),
        saturation=float(0.1*np.random.rand(1)),
        hue=float(0.1*np.random.rand(1)))
]
```

   This combination yielded an accuracy of 72.7% on the 1000 test images

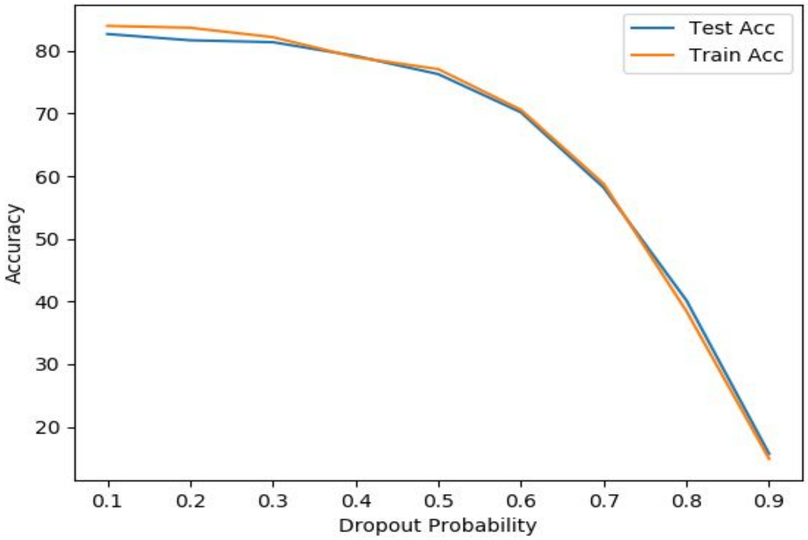
b. **Plot of Training and Test accuracies for different values of dropout (0.1 - 0.9)**

   From the below graphs, we can observe that the train and test accuracy gradually decreases as we increase the dropout probability. This is clearly evident in the plot of the network with batch normalization.

Dropout without batch normalization:

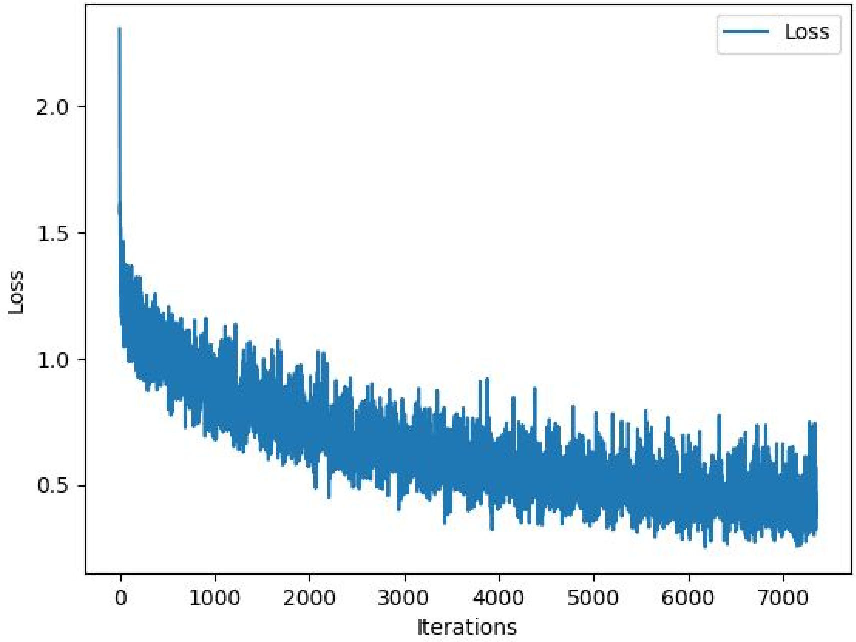


Dropout with batch normalization:


Test Acc
Train Acc
Accuracy
80
70
60
50
40
30
20
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Dropout Probability

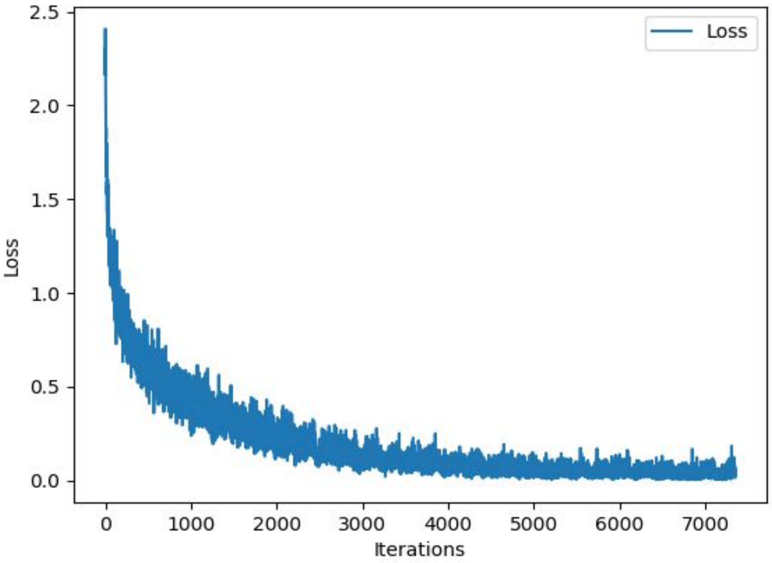
Question 4: Use pretrained networks

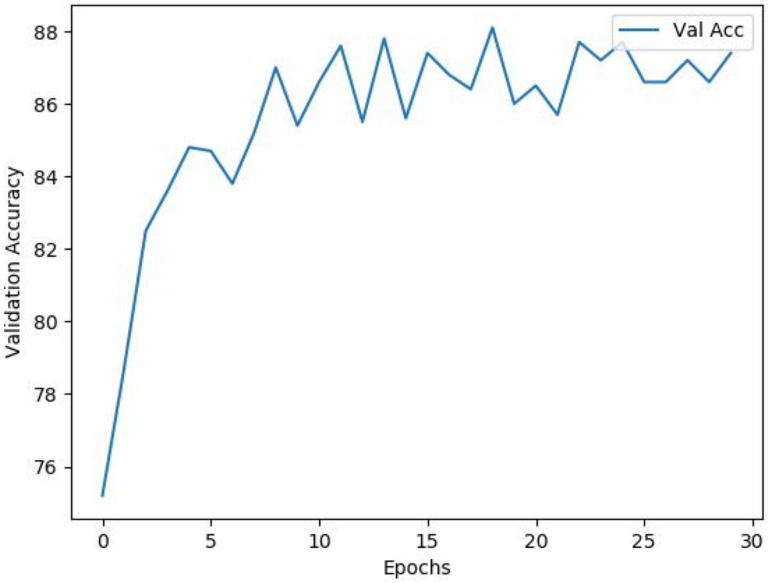
1. **Validation and testing accuracies:**
   For **Model 1** (with Pretrained = True and finetune = True) we get a validation accuracy of 61.9 % and a test accuracy of 63 %



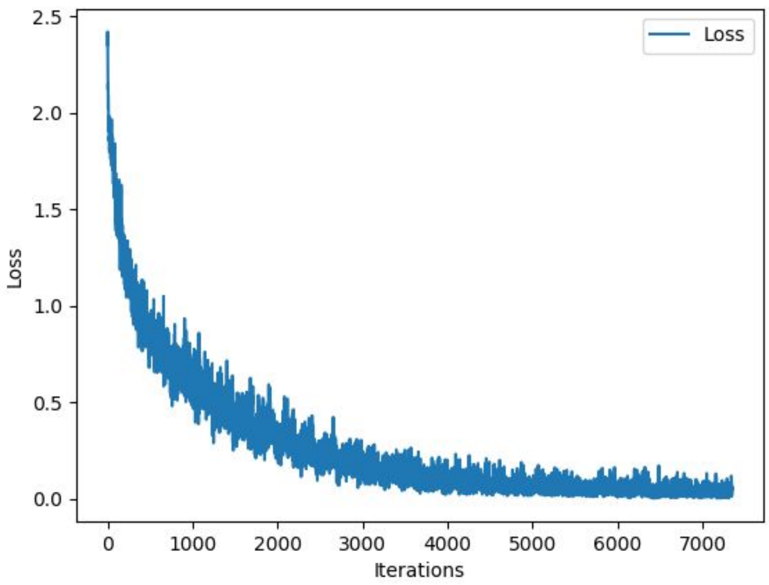
2. **Compare the two models training curves, validation and testing performance**

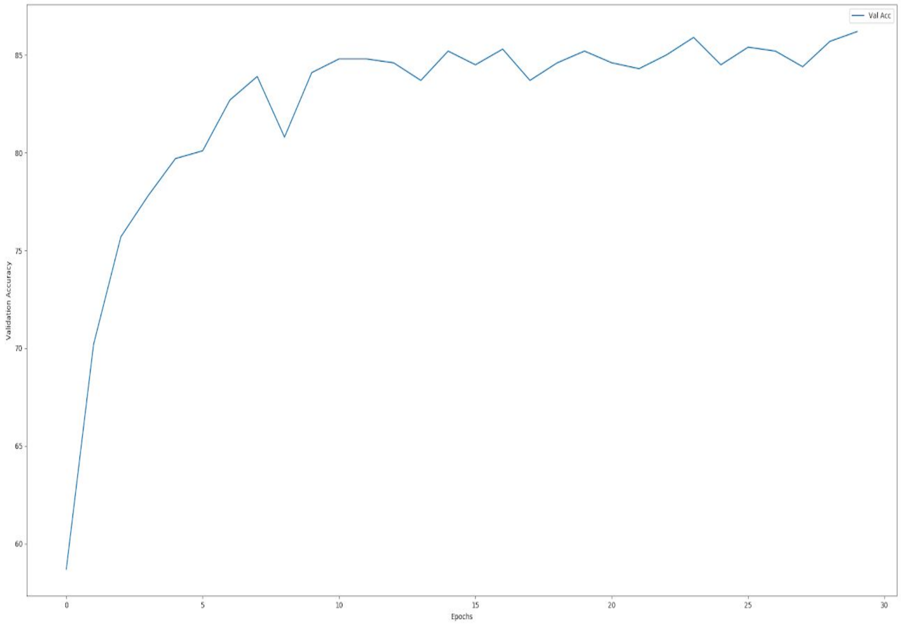
Model 2:





Model 3





We see that the training loss and validation accuracy curves are more or less the same for both the models.

The validation accuracies of Model 2 and Model 3 are 86% and 86.6% respectively.
The test accuracies of Model 2 and Model 3 are 87.6% and 86.02% respectively.

We observe that the test and validation accuracies for Model 3 are slightly better than Model 2. But both are vastly better than Model 1. This is because for Model 2 and 3 we have used the CIFAR-10 dataset itself for fine-tuning and training respectively.

We might have expected the accuracies to be much better for Model 3 as compared to Model 2 but that is not the case perhaps because the CIFAR - 10 dataset is too small for the VGG-net.