Ankit Nitinkumar Bhawsar
USC ID: 4889451132
abhawsar@usc.edu

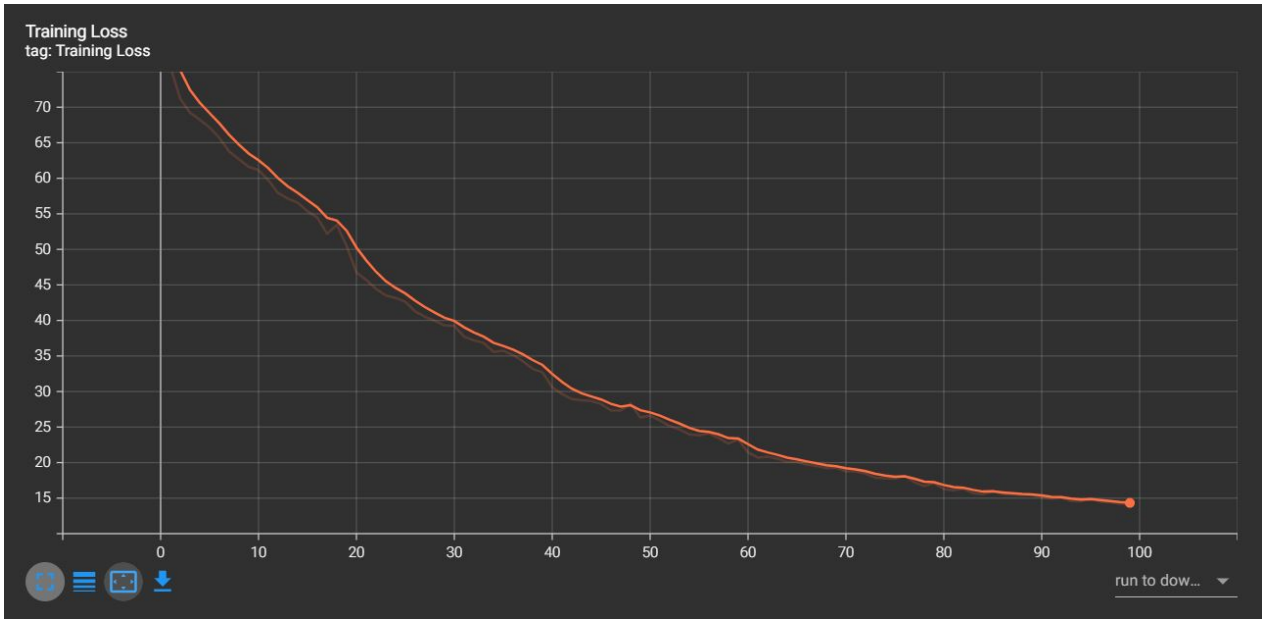
1 Main Experiment

For the main experiment, I used the hyperparameters as suggested in the HW PDF, i.e.

learning rate = 0.001
epochs = 100
LR Decay 50% after 20 epochs
Run on Validation set every 5 epochs

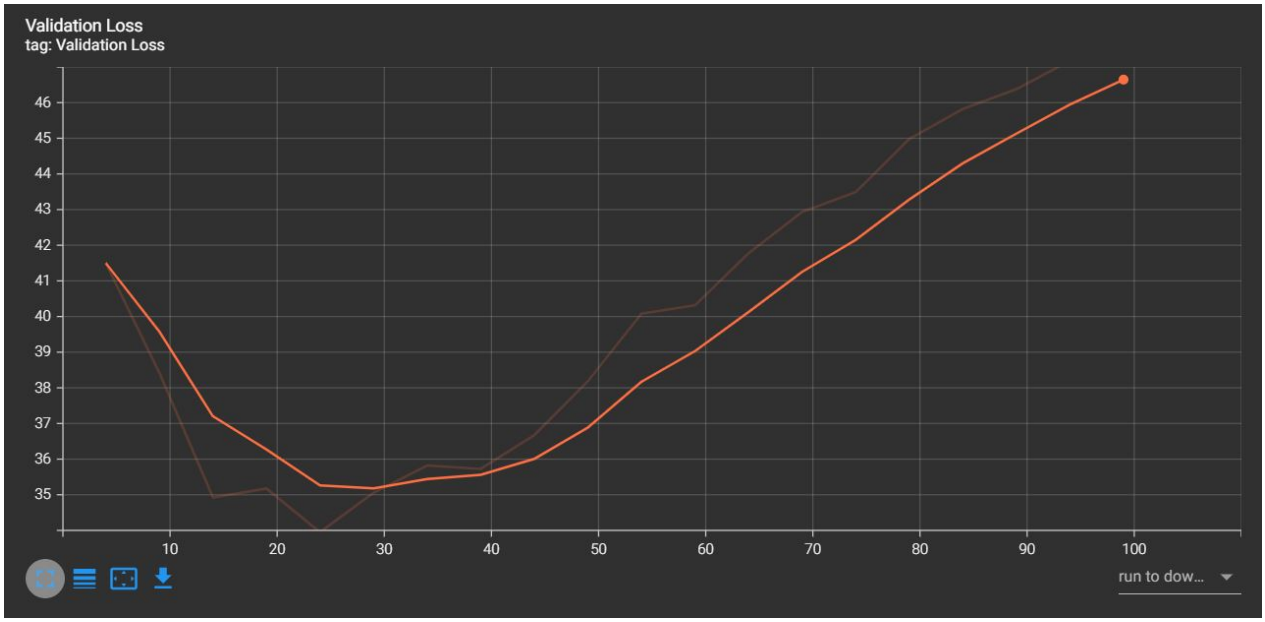
1.1 Test Results

Evolution of Training Loss:



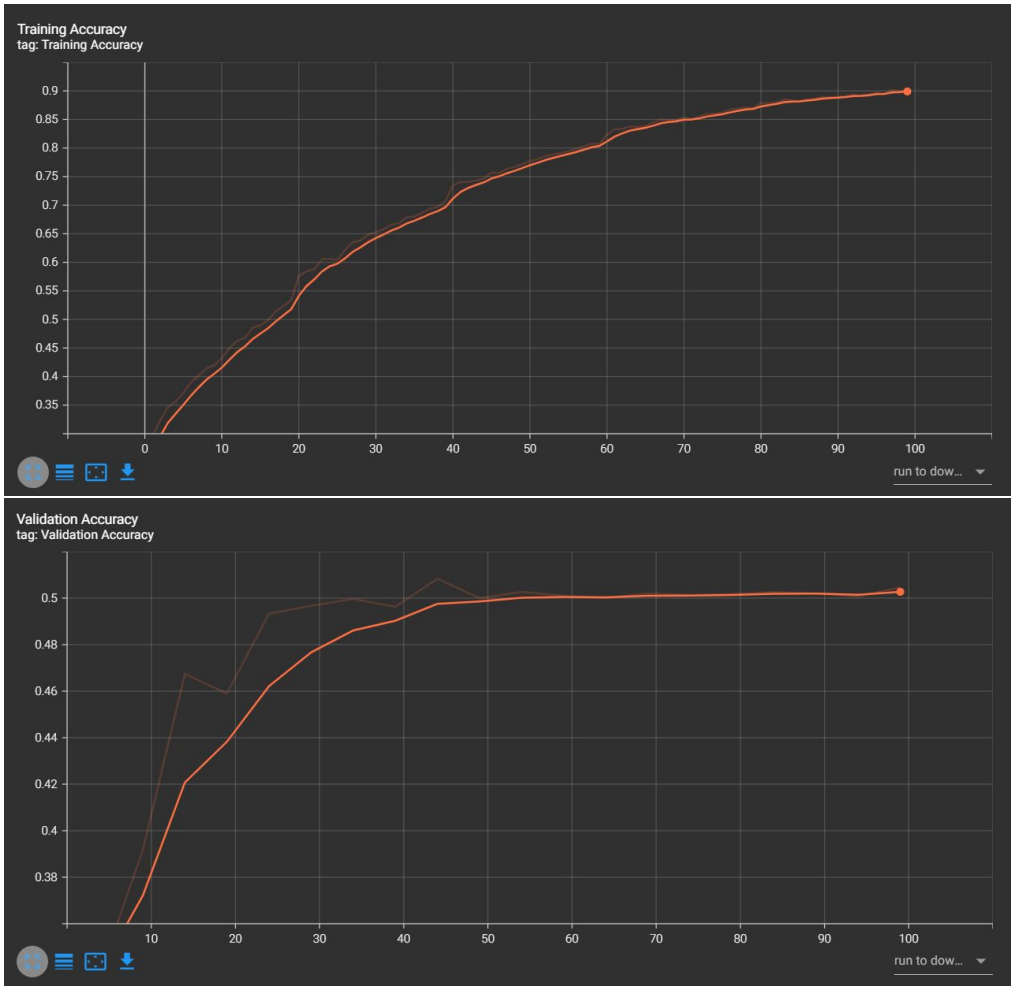
Training Loss over Epochs

Evolution of Validation Loss:



Validation Loss over Epochs

Evolution of Accuracy:



Training and Validation Accuracy over Epochs

1.2 Confusion Matrix and Accuracy on Test Set

Anaconda Prompt

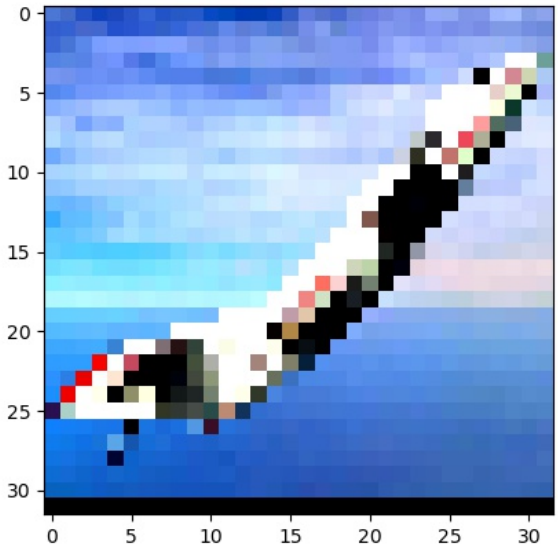
```
(cv_is_fun) C:\Users\admin\CS 677 ACV\CSCI_677_HW4>python hw4_ab_v4.py
Correct= 2468 / 5000
Accuracy of the network on the 8000 test images: 49 %
Correct= 4509 / 5000
Accuracy of the network on the 5000 training images: 90 %
Correct= 1527 / 3000
Accuracy of the network on the 3000 Validation images: 50 %
------------------------------------------------------------
Accuracy for class  airplane  =  352 / 500  =  70.4 %
Accuracy for class  bird  =  183 / 500  =  36.6 %
Accuracy for class  car  =  321 / 500  =  64.2 %
Accuracy for class  cat  =  166 / 500  =  33.2 %
Accuracy for class  deer  =  231 / 500  =  46.2 %
Accuracy for class  dog  =  119 / 500  =  23.8 %
Accuracy for class  horse  =  287 / 500  =  57.4 %
Accuracy for class  monkey  =  194 / 500  =  38.8 %
Accuracy for class  ship  =  342 / 500  =  68.4 %
Accuracy for class  truck  =  273 / 500  =  54.6 %
------------------------------------------------------------
```

	pred_airplane	pred_bird	pred_car	pred_cat	pred_deer	pred_dog	pred_horse	pred_monkey	pred_ship	pred_truck
airplane	352.0	29.0	22.0	9.0	3.0	5.0	3.0	2.0	38.0	37.0
bird	25.0	183.0	10.0	73.0	61.0	47.0	27.0	54.0	10.0	10.0
car	24.0	8.0	321.0	11.0	9.0	5.0	4.0	8.0	27.0	83.0
cat	5.0	58.0	6.0	166.0	74.0	55.0	37.0	67.0	14.0	18.0
deer	7.0	54.0	2.0	60.0	231.0	34.0	66.0	35.0	5.0	6.0
dog	4.0	55.0	5.0	86.0	57.0	119.0	84.0	81.0	0.0	9.0
horse	3.0	34.0	4.0	35.0	35.0	38.0	287.0	43.0	3.0	18.0
monkey	2.0	32.0	5.0	67.0	33.0	76.0	75.0	194.0	1.0	15.0
ship	50.0	9.0	26.0	10.0	11.0	4.0	2.0	1.0	342.0	45.0
truck	27.0	17.0	88.0	14.0	8.0	5.0	24.0	6.0	38.0	273.0

Confusion Matrix and Accuracy over Classes

1.3 Some Failed Examples

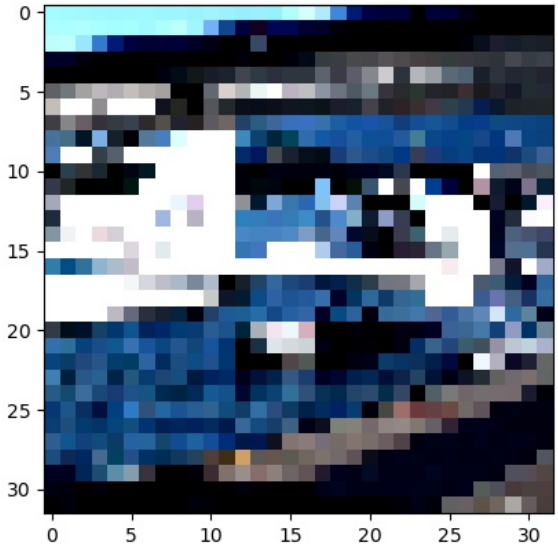
Example 1: The following example of an airplane is wrongly classified as a bird. It is because the shape, wings and angle of the plane make it look like a bird in flight. The background sky is also similar to what a you might see in the image of a bird.



```
---------------------------
idx= 5
GroundTruth: airplane
Predicted:  bird
---------------------------
```

Wrongly Classified Example

Example 2: The following example of an airplane is wrongly classified as a car, probably because it does not contain any distinctive features of an airplane. Even the human eye cannot distinguish any possible object in this image.



```
------------------------------
idx= 4
GroundTruth: airplane
Predicted:  car
------------------------------
```

Wrongly Classified Example

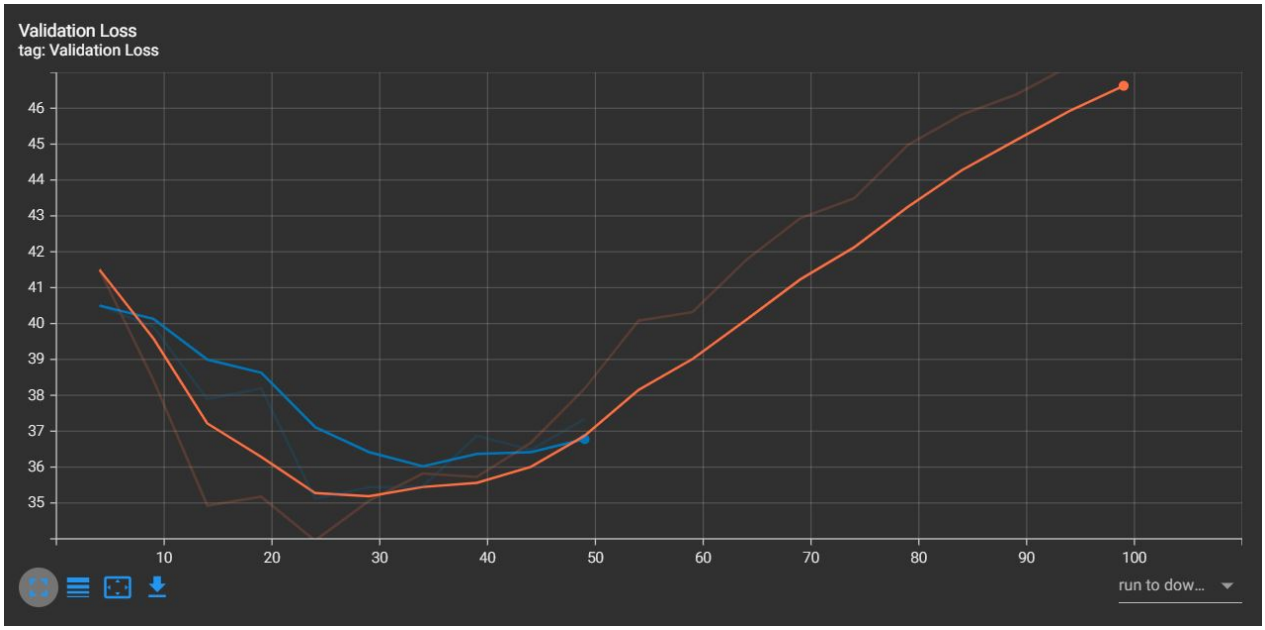
1.4 Conclusion:

Thus we can see that LeNet-5 is not particularly that powerful. It does provide 49% accuracy over unseen data which is much better than random sampling (10%).

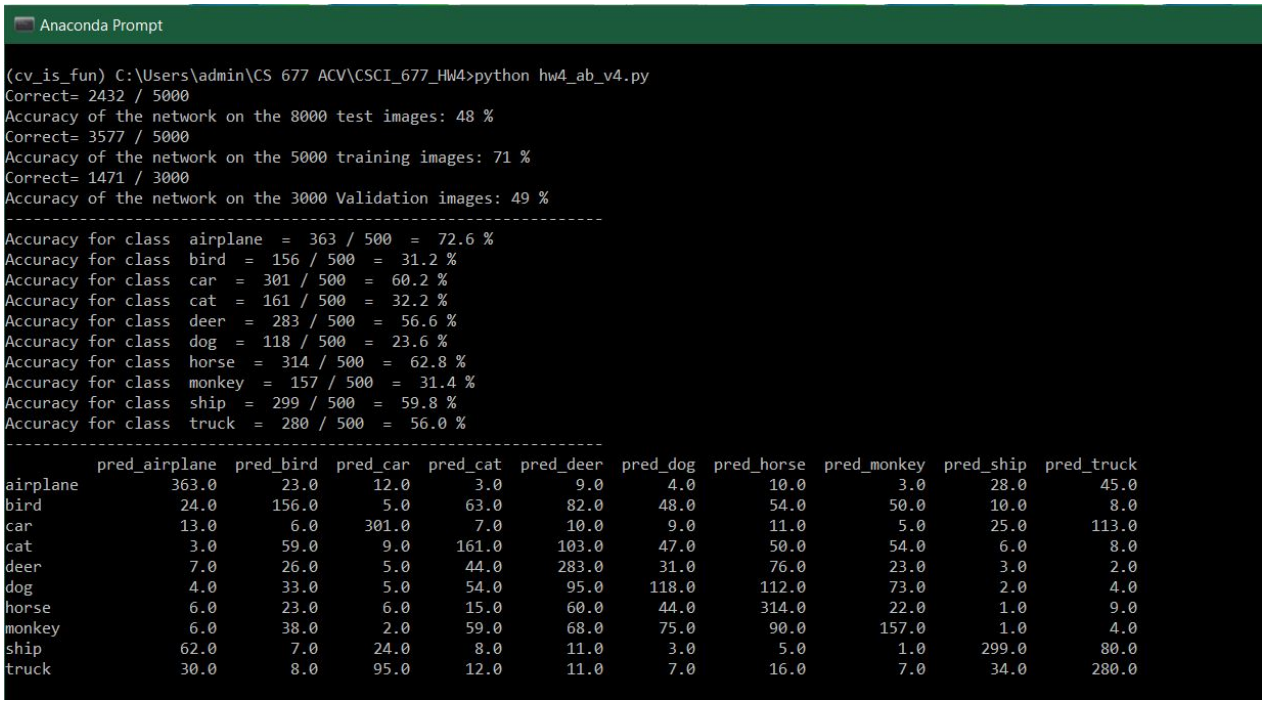
We can also observe that, the validation error decrease up till a certain point after which it increases, meaning the model begins to overfit. This point is observed to be around 40 epochs.

If we train only up to 40 epochs, we can see the following result, where we can train the model much faster and the training accuracy decreases.

Evolution of Validation Loss:



Validation Loss over 50 Epochs (In Blue)



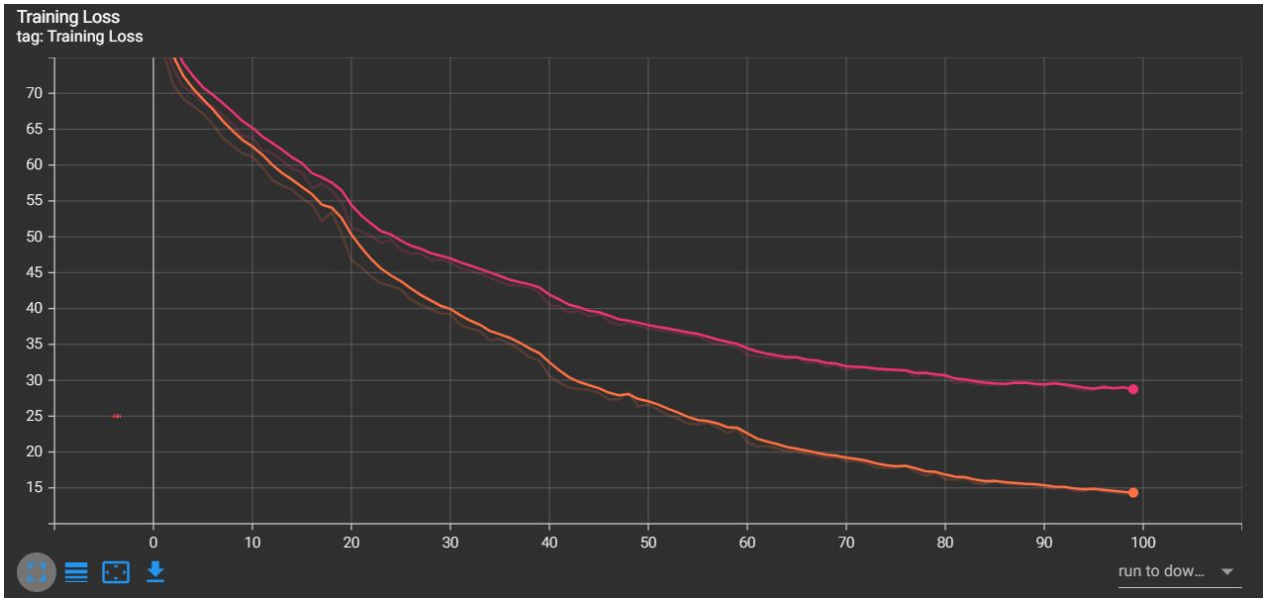
Wrongly Classified Example

2 Variation 1: L2 Regularization

2.1 Test Results:

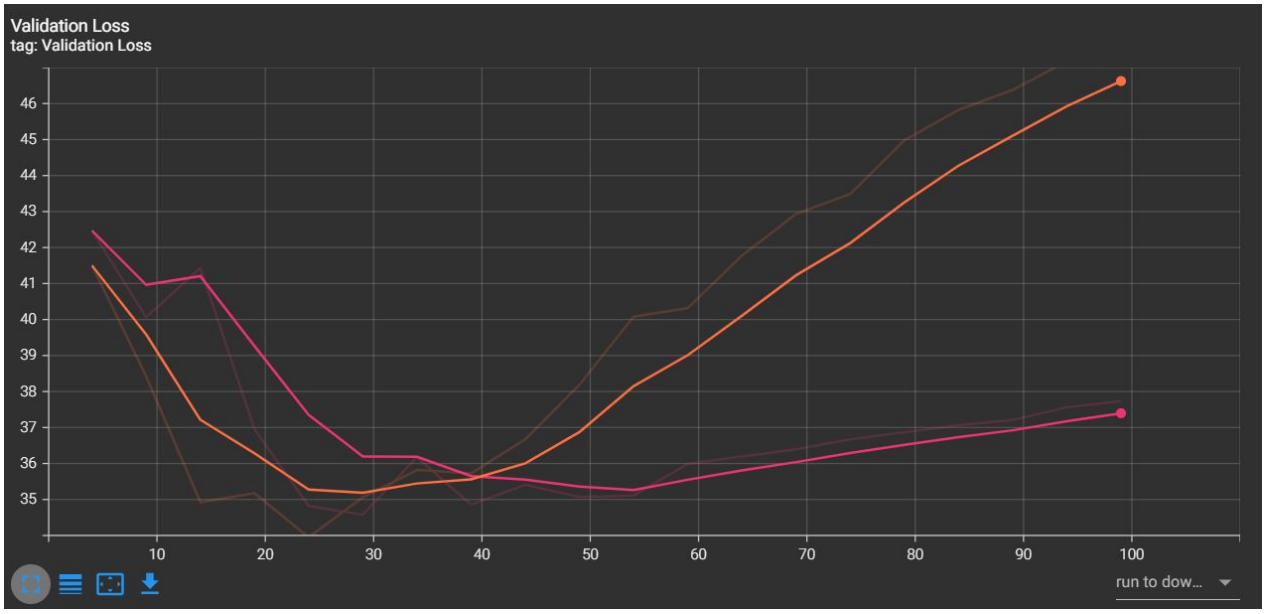
For this variation, I added L2 regularization during training using the weight_decay parameter of Adam optimizer.

Evolution of Training Loss:



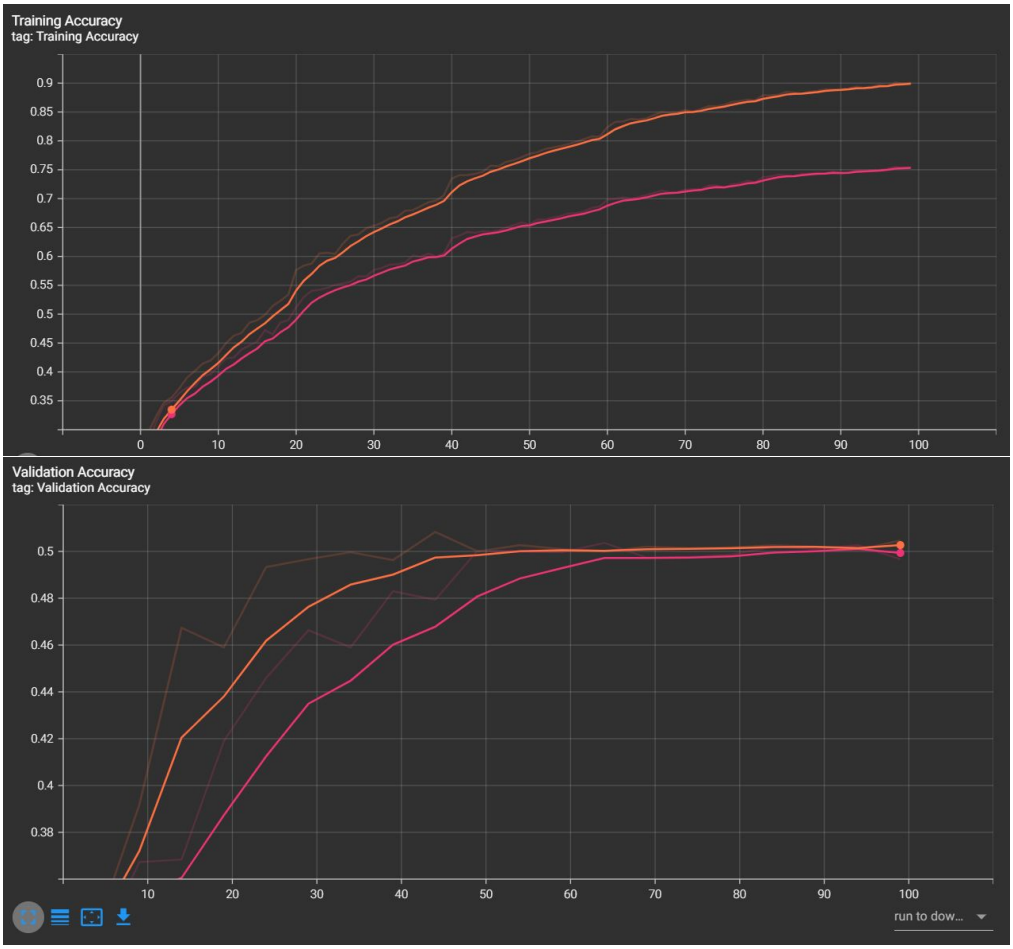
L2 Training Loss(In Pink) and Training Loss(In Orange) over Epochs

Evolution of Validation Loss:



L2 Validation Loss(In Pink) and Validation Loss(In Orange) over Epochs

Evolution of Accuracy:



L2 Training and Validation Accuracy (In Pink) and Training and Validation Accuracy (In Orange) over Epochs

2.2 Confusion Matrix and Accuracy on Test Set

Anaconda Prompt

```
(cv_is_fun) C:\Users\admin\CS 677 ACV\CSCI_677_HW4>python hw4_ab_v4.py
Correct= 2425 / 5000
Accuracy of the network on the 8000 test images: 48 %
Correct= 3980 / 5000
Accuracy of the network on the 5000 training images: 79 %
Correct= 1469 / 3000
Accuracy of the network on the 3000 Validation images: 48 %
------------------------------------------------------------
Accuracy for class  airplane  =  361 / 500  =  72.2 %
Accuracy for class  bird  =  174 / 500  =  34.8 %
Accuracy for class  car  =  299 / 500  =  59.8 %
Accuracy for class  cat  =  166 / 500  =  33.2 %
Accuracy for class  deer  =  214 / 500  =  42.8 %
Accuracy for class  dog  =  161 / 500  =  32.2 %
Accuracy for class  horse  =  277 / 500  =  55.4 %
Accuracy for class  monkey  =  184 / 500  =  36.8 %
Accuracy for class  ship  =  327 / 500  =  65.4 %
Accuracy for class  truck  =  262 / 500  =  52.4 %
------------------------------------------------------------
          pred_airplane  pred_bird  pred_car  pred_cat  pred_deer  pred_dog  pred_horse  pred_monkey  pred_ship  pred_truck
airplane          361.0       20.0      17.0       6.0        3.0       8.0         4.0          2.0       36.0        43.0
bird               29.0      174.0      10.0      51.0       62.0      54.0        39.0         56.0       12.0        13.0
car                17.0        3.0     299.0       8.0        5.0       7.0        10.0          3.0       36.0       112.0
cat                 3.0       57.0      14.0     166.0       62.0      72.0        34.0         62.0       13.0        17.0
deer                7.0       62.0       8.0      51.0      214.0      48.0        67.0         29.0        6.0         8.0
dog                 4.0       41.0       5.0      55.0       53.0     161.0        81.0         90.0        3.0         7.0
horse               2.0       27.0      10.0      26.0       39.0      61.0       277.0         43.0        2.0        13.0
monkey              4.0       50.0       6.0      66.0       23.0     101.0        57.0        184.0        5.0         4.0
ship               57.0       10.0      24.0      11.0        5.0       6.0         2.0          1.0      327.0        57.0
truck              38.0        6.0      96.0      10.0        4.0      13.0        14.0          4.0       53.0       262.0
```

Confusion Matrix and Accuracy over Classes

2.3 Conclusion:

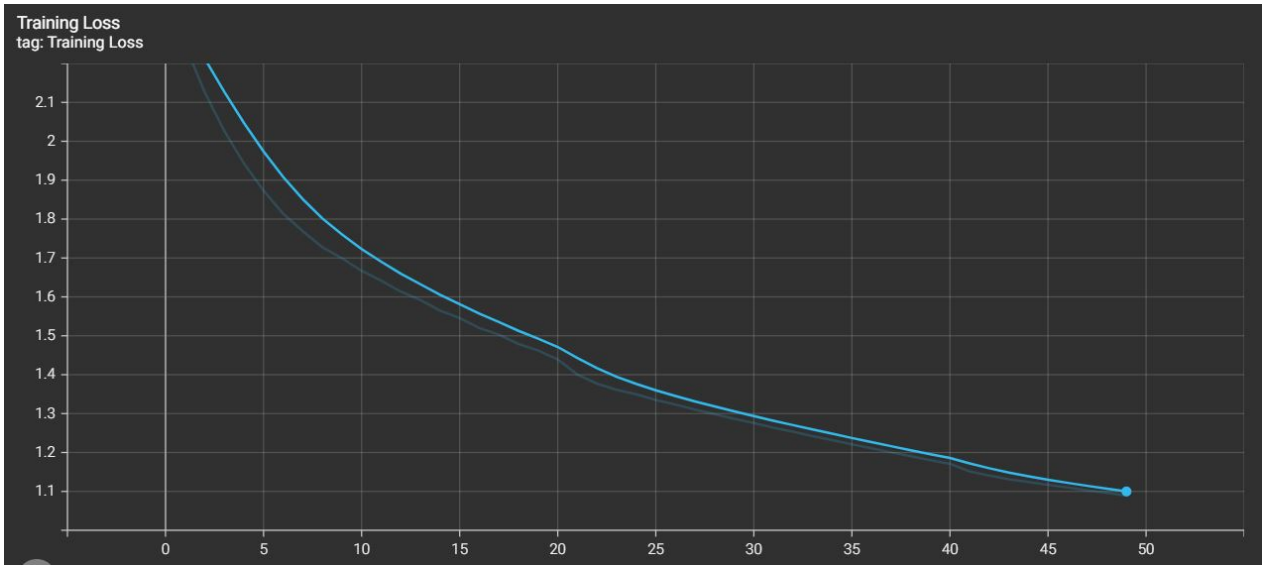
Thus we can see that after implementing regularization, the training error decreases more gradually compared to without regularization

We can also observe that, the validation error does not increase as much as in the Main experiment. This is because it is a more generalized model, and can handle unseen data much better.

3 Variation 2: Batch Normalization

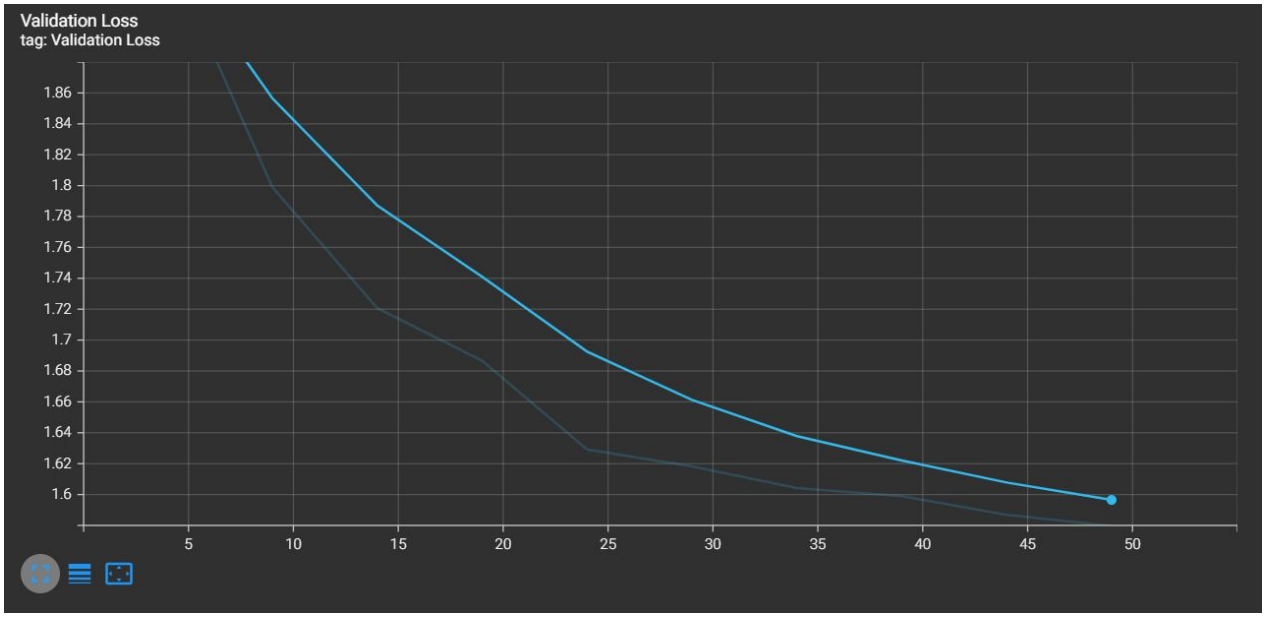
3.1 Test Results:

For this variation, I added Batch normalization to every layer while creating the neural network.
Evolution of Training Loss:



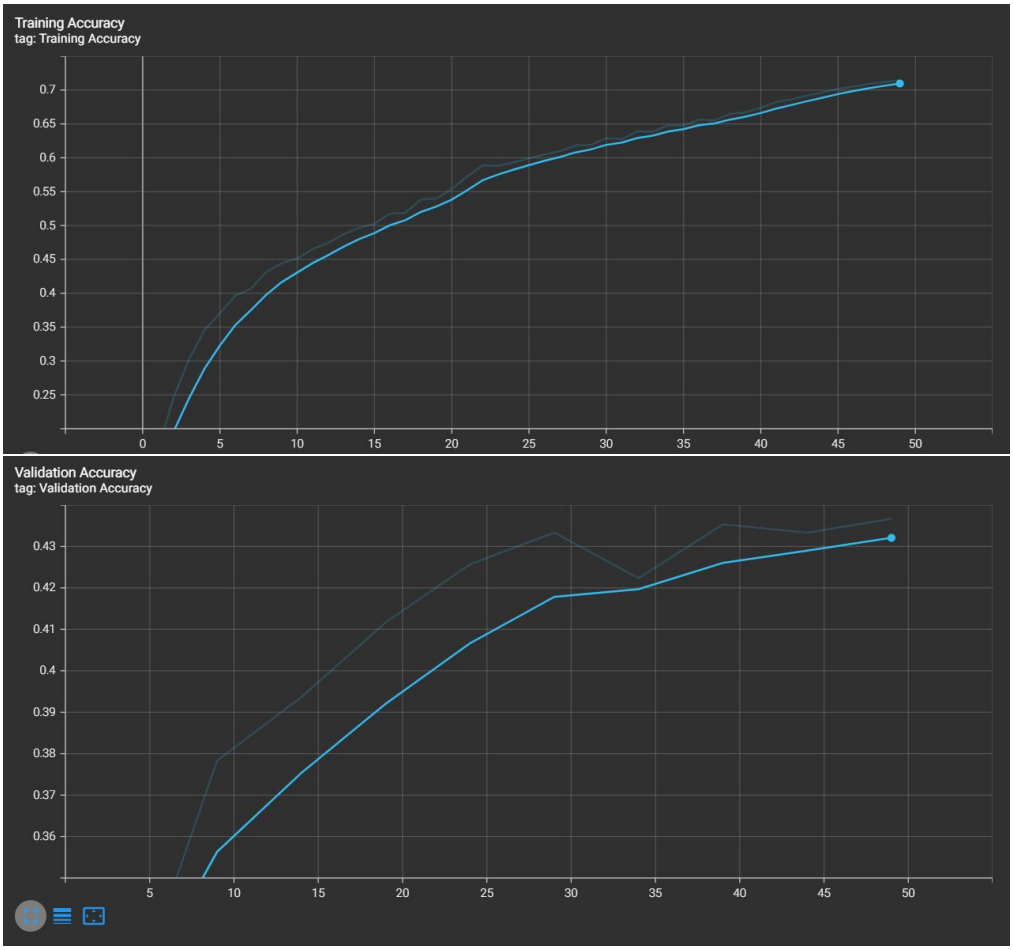
Batch Normalization Training Loss(In Pink), L2 Training Loss(In Pink) and Training Loss(In Orange) over Epochs

Evolution of Validation Loss:



Batch Normalization Validation Loss (In Pink),L2 Validation Loss(In Pink) and Validation Loss(In Orange) over Epochs

Evolution of Accuracy:



Batch Normalization Training and Validation Accuracy (In Pink), L2 Training and Validation Accuracy (In Pink) and Training and Validation Accuracy (In Orange) over Epochs

3.2 Confusion Matrix and Accuracy on Test Set

```
Correct= 2182 / 5000
Accuracy of the network on the 8000 test images: 43 %
Correct= 3588 / 5000
Accuracy of the network on the 5000 training images: 71 %
Correct= 1310 / 3000
Accuracy of the network on the 3000 Validation images: 43 %
----------------------------------------------------------------
Accuracy for class  airplane  =  357 / 500  =  71.4 %
Accuracy for class  bird  =  140 / 500  =  28.0 %
Accuracy for class  car  =  301 / 500  =  60.2 %
Accuracy for class  cat  =  144 / 500  =  28.8 %
Accuracy for class  deer  =  193 / 500  =  38.6 %
Accuracy for class  dog  =  108 / 500  =  21.6 %
Accuracy for class  horse  =  224 / 500  =  44.8 %
Accuracy for class  monkey  =  180 / 500  =  36.0 %
Accuracy for class  ship  =  318 / 500  =  63.6 %
Accuracy for class  truck  =  217 / 500  =  43.4 %
----------------------------------------------------------------
```

Accuracy over Classes

```
tensor([[357.,  15.,  15.,   6.,   7.,   8.,   6.,   4.,  61.,  21.],
        [ 36., 140.,  13.,  70.,  62.,  54.,  39.,  58.,  13.,  15.],
        [ 11.,   7., 301.,   5.,   5.,  15.,  12.,  14.,  18., 112.],
        [  7.,  56.,  17., 144.,  72.,  61.,  46.,  66.,  19.,  12.],
        [ 16.,  43.,   8.,  66., 193.,  52.,  66.,  42.,   6.,   8.],
        [ 13.,  62.,   5.,  74.,  76., 108.,  76.,  73.,   9.,   4.],
        [  4.,  37.,  20.,  31.,  43.,  56., 224.,  68.,   5.,  12.],
        [  5.,  37.,  14.,  76.,  47.,  78.,  54., 180.,   5.,   4.],
        [ 56.,   4.,  27.,   9.,  12.,   4.,   3.,   6., 318.,  61.],
        [ 33.,  11., 114.,  13.,   4.,   8.,  20.,  14.,  66., 217.]])
```

Confusion Matrix

4 Results of all Variations Compared:

4.1 Accuracies:

	Main Experiment	Batch Normalization	L2 Normalization
Train Accuracy	0.90	0.71	0.79
Validation Accuracy	0.50	0.43	0.48
Test Accuracy	0.49	0.43	0.48

4.2 Performance on Classes:

	Main Experiment	Batch Normalization	L2 Normalization
airplane	70.4	71.4	72.2
bird	36.6	28.0	34.8
car	64.2	60.2	59.8
cat	33.2	28.8	33.2
deer	46.2	38.6	42.8
dog	23.8	21.6	32.2
horse	57.4	44.8	55.4
monkey	38.8	36.0	36.8
ship	68.4	63.6	65.4
truck	54.6	43.4	52.4