DAI Report

Priya Sahu (B20AI031)

QUESTION 1

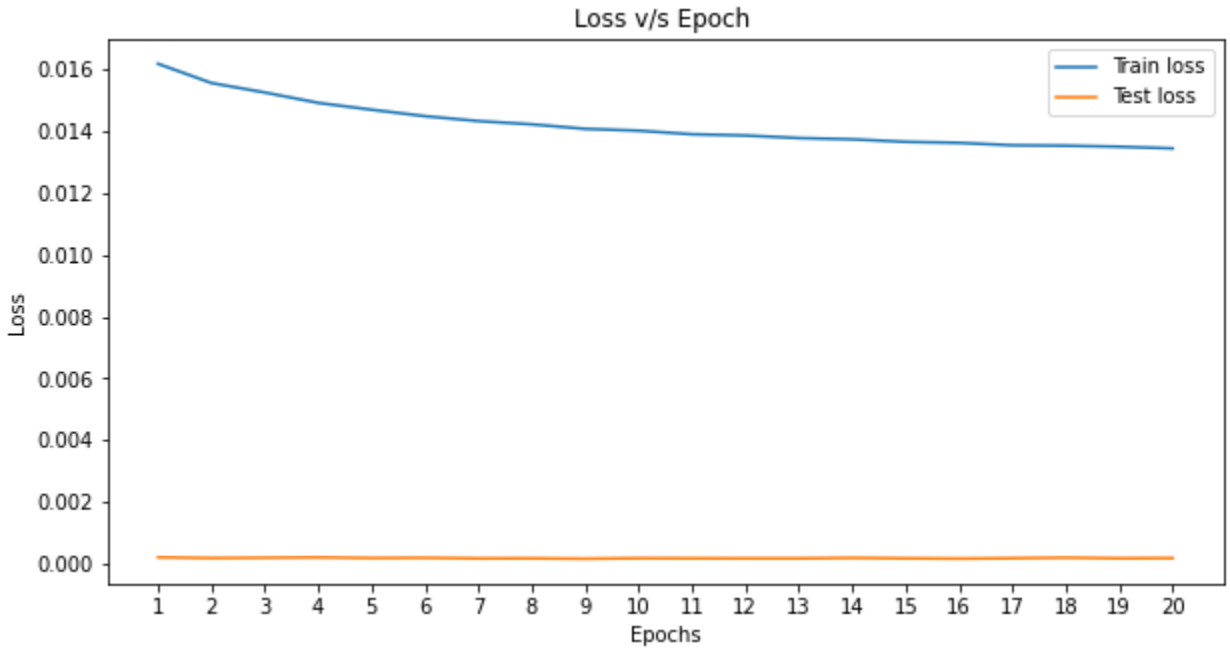
- Import all the necessary libraries and dependencies.

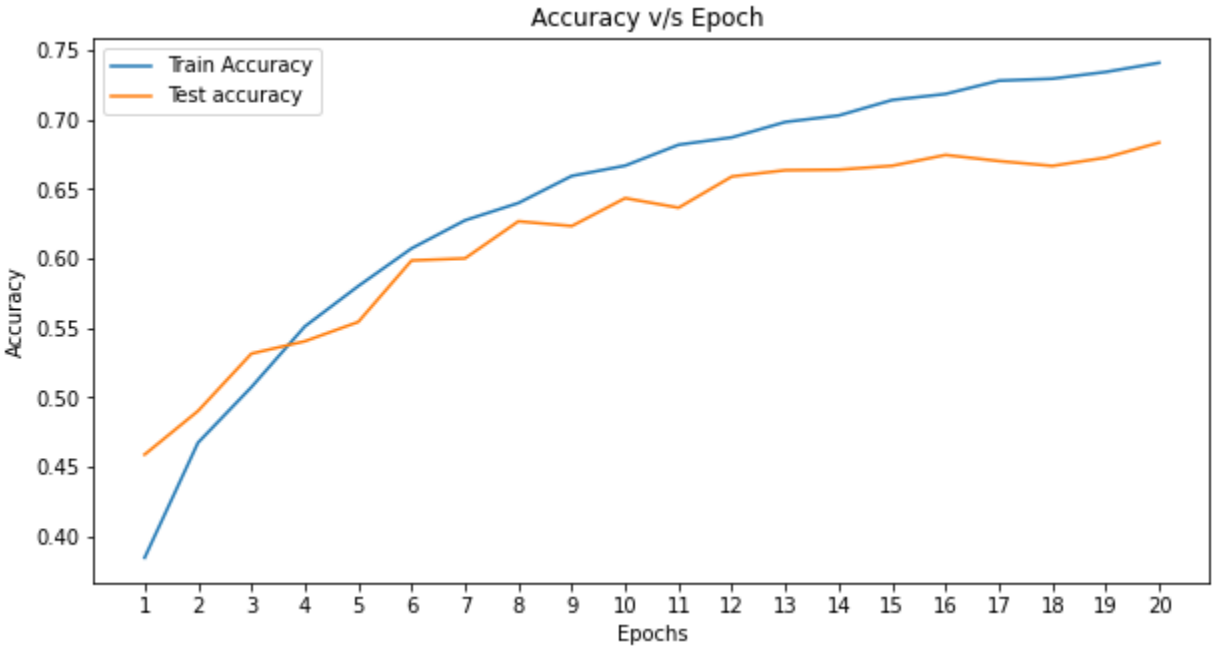
(i)

- Loaded the CIFAR10 dataset. Made train loader and test loaders having batch size=128.
- Now a CNN class is made. The architecture of CNN is

Conv2d → ReLU →Max Pool → Conv2d → ReLU → Conv2d → MaxPool → Flatten → Linear → Softmax

- Now trained the model with Adam optimiser for 20 epochs.





FGSM Attack :

Defined a function for FGSM which takes the model, images and labels of images, and epsilon (the factor of perturbation) and it gives us the perturbed images.

For different values of epsilon :

Basically, results show what percent of labels are wrongly classified after perturbation

```
Epsilon:  0   Changes the classes for (%):  tensor(31.6600, device='cuda:0')
Epsilon:  0.01   Changes the classes for (%):  tensor(63.5900, device='cuda:0')
Epsilon:  0.025   Changes the classes for (%):  tensor(84.5800, device='cuda:0')
Epsilon:  0.05   Changes the classes for (%):  tensor(96.2800, device='cuda:0')
Epsilon:  0.075   Changes the classes for (%):  tensor(98.4100, device='cuda:0')
Epsilon:  0.1   Changes the classes for (%):  tensor(98.6800, device='cuda:0')
Epsilon:  0.2   Changes the classes for (%):  tensor(98.0100, device='cuda:0')
```

Now the results below show what percent of classification by classifier is misclassified after perturbation.

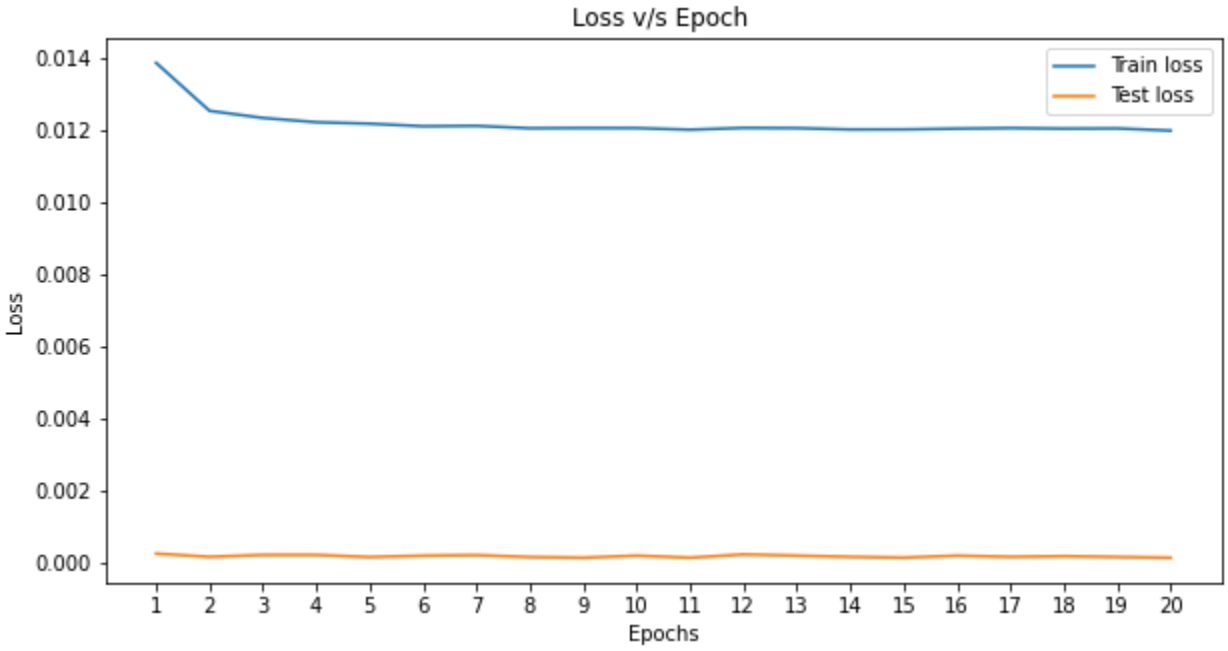
```
Epsilon:  0   Success Rate:  tensor(0., device='cuda:0')
Epsilon:  0.01   Success Rate:  tensor(31.9700, device='cuda:0')
Epsilon:  0.025   Success Rate:  tensor(52.9700, device='cuda:0')
Epsilon:  0.05   Success Rate:  tensor(65.1500, device='cuda:0')
Epsilon:  0.075   Success Rate:  tensor(68.1900, device='cuda:0')
Epsilon:  0.1   Success Rate:  tensor(69.5300, device='cuda:0')
Epsilon:  0.2   Success Rate:  tensor(72.9000, device='cuda:0')
```

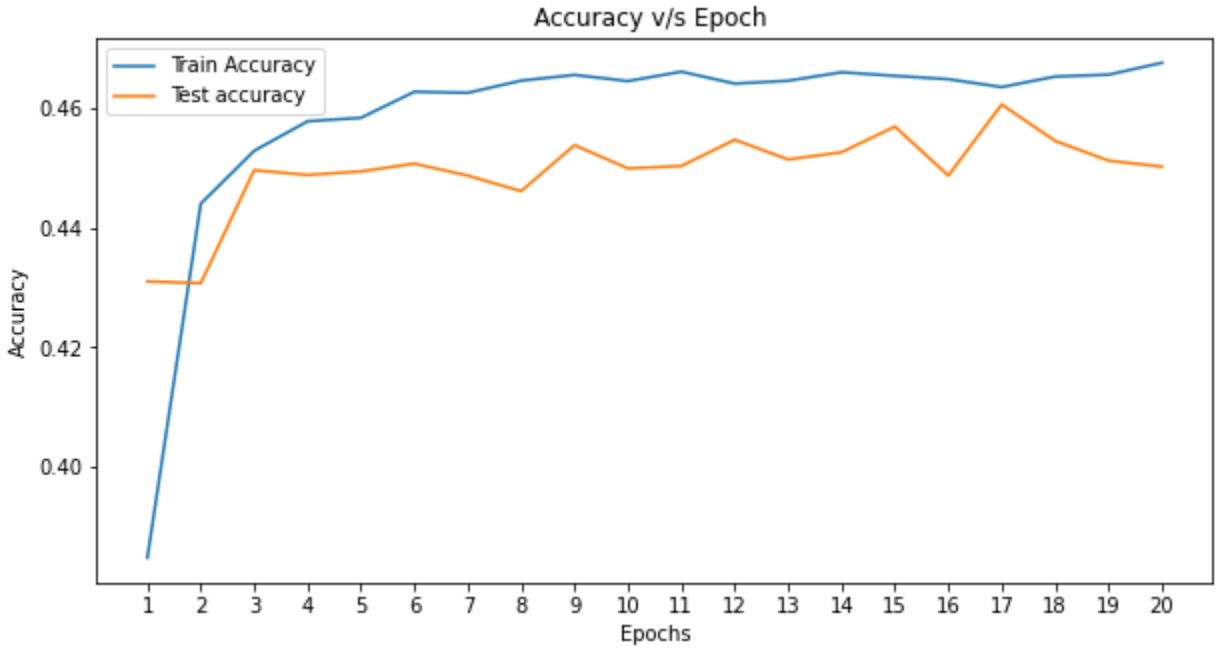
Note : There are two types of results shown. In the first one we compared the true labels and labels of perturbed samples given by the classifier. In the second one, we compared predicted labels given by the classifier and labels given by the classifier after perturbation.

From the results, we can see that as epsilon increases it is most likely to be misclassified by classifiers. But actually we cannot increase epsilon much because then our images will start to look different.

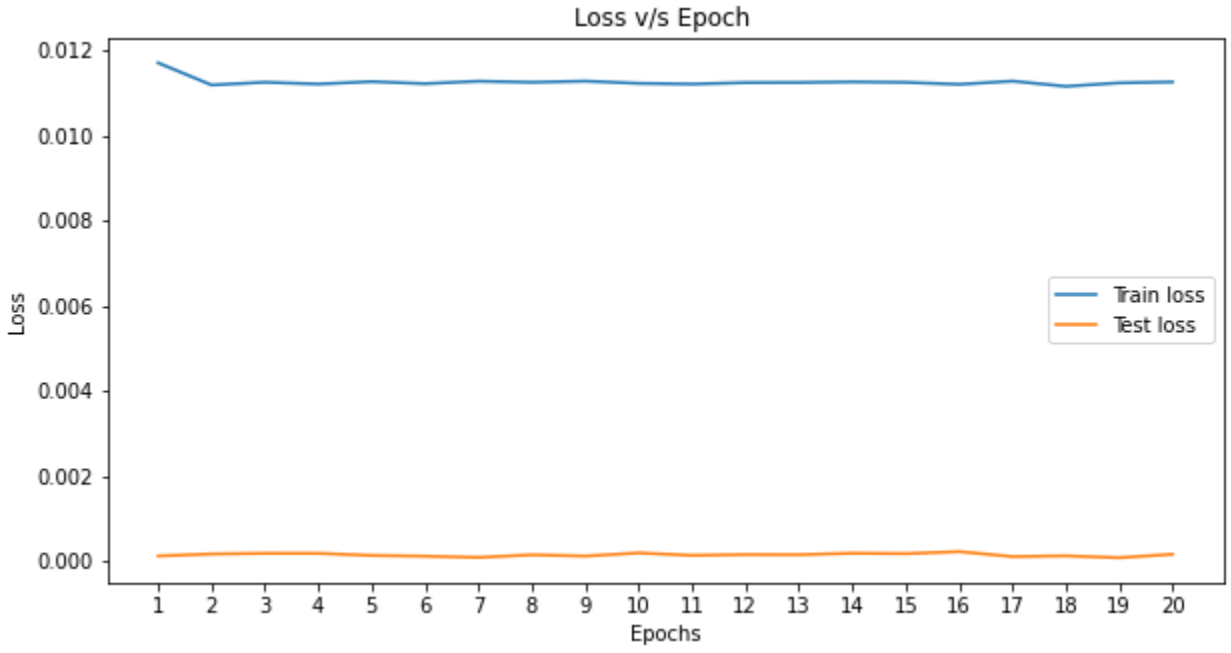
(ii)

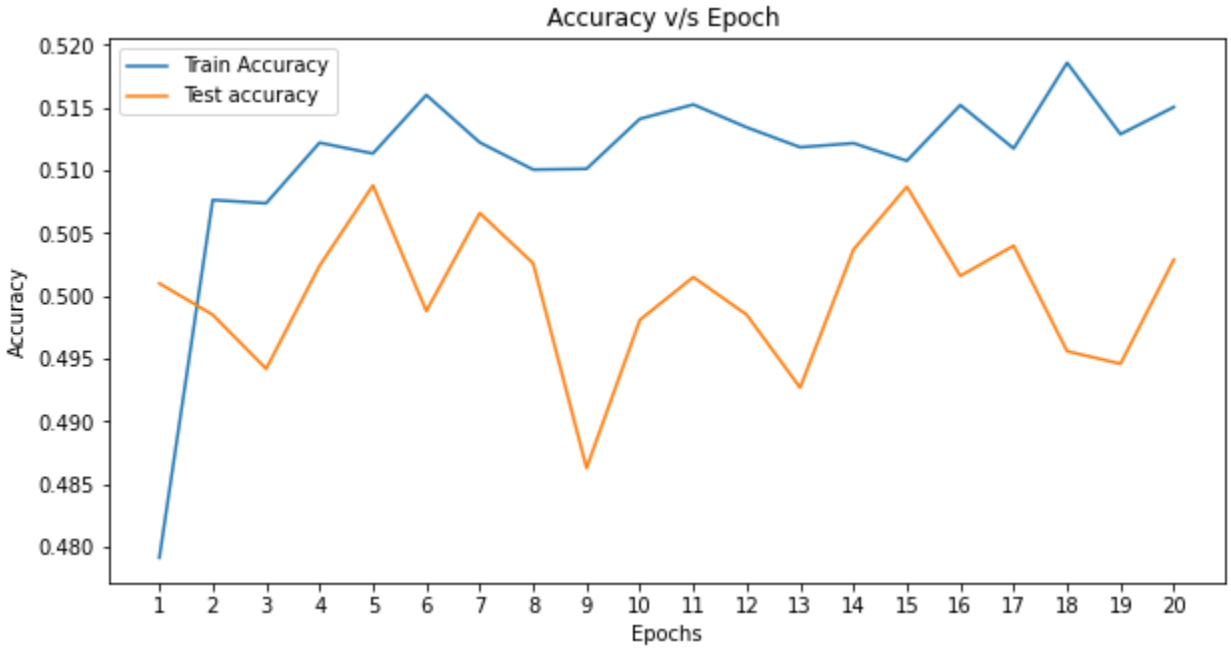
- Trained model2 using pretrained resnet18. Just finetune it with our dataset.





- Trained model3 using pretrained vgg. Just finetune it with our dataset.





PGD Attack :

Defined a function for PGD which takes the model, images and labels of images, and epsilon (the factor of perturbation) and it gives us the perturbed images.

a) Attack model2

For different values of epsilon :

Basically, results show what percent of labels are wrongly classified after perturbation

```
Epsilon:  0   Changes the classes for (%):  tensor(89.8083, device='cuda:0')
Epsilon:  0.01   Changes the classes for (%):  tensor(99.2917, device='cuda:0')
Epsilon:  0.025   Changes the classes for (%):  tensor(99.9417, device='cuda:0')
Epsilon:  0.05   Changes the classes for (%):  tensor(100., device='cuda:0')
Epsilon:  0.075   Changes the classes for (%):  tensor(100., device='cuda:0')
Epsilon:  0.1   Changes the classes for (%):  tensor(100., device='cuda:0')
Epsilon:  0.2   Changes the classes for (%):  tensor(100., device='cuda:0')
```

Now the results below show what percent of classification by classifier is misclassified after perturbation.

```
Epsilon:  0   Success Rate:  tensor(0., device='cuda:0')
Epsilon:  0.01   Success Rate:  tensor(63.3167, device='cuda:0')
Epsilon:  0.025   Success Rate:  tensor(71.2000, device='cuda:0')
Epsilon:  0.05   Success Rate:  tensor(73.9500, device='cuda:0')
Epsilon:  0.075   Success Rate:  tensor(74.2583, device='cuda:0')
Epsilon:  0.1   Success Rate:  tensor(73.5083, device='cuda:0')
Epsilon:  0.2   Success Rate:  tensor(73.8750, device='cuda:0')
```

b) Attack model3

For different values of epsilon :

Basically, results show what percent of labels are wrongly classified after perturbation

```
Epsilon:  0   Changes the classes for (%):  tensor(88.9833, device='cuda:0')
Epsilon:  0.01   Changes the classes for (%):  tensor(98.5917, device='cuda:0')
Epsilon:  0.025   Changes the classes for (%):  tensor(99.8750, device='cuda:0')
Epsilon:  0.05   Changes the classes for (%):  tensor(99.9667, device='cuda:0')
Epsilon:  0.075   Changes the classes for (%):  tensor(99.9833, device='cuda:0')
Epsilon:  0.1   Changes the classes for (%):  tensor(99.9917, device='cuda:0')
Epsilon:  0.2   Changes the classes for (%):  tensor(100., device='cuda:0')
```

Now the results below show what percent of classification by classifier is misclassified after perturbation.

```
Epsilon:  0   Success Rate:  tensor(58.2250, device='cuda:0')
Epsilon:  0.01   Success Rate:  tensor(66.7250, device='cuda:0')
Epsilon:  0.025   Success Rate:  tensor(74.6333, device='cuda:0')
Epsilon:  0.05   Success Rate:  tensor(77.1417, device='cuda:0')
Epsilon:  0.075   Success Rate:  tensor(78.2917, device='cuda:0')
Epsilon:  0.1   Success Rate:  tensor(78.8750, device='cuda:0')
Epsilon:  0.2   Success Rate:  tensor(79.0583, device='cuda:0')
```