1 Deliverable 1

Solution: Code submitted on gradescope as a zip file (submission.zip) □

2 Deliverable 2: MNIST Dataset Cutom CNN Loss Plot for Training and Validation Data

Solution:

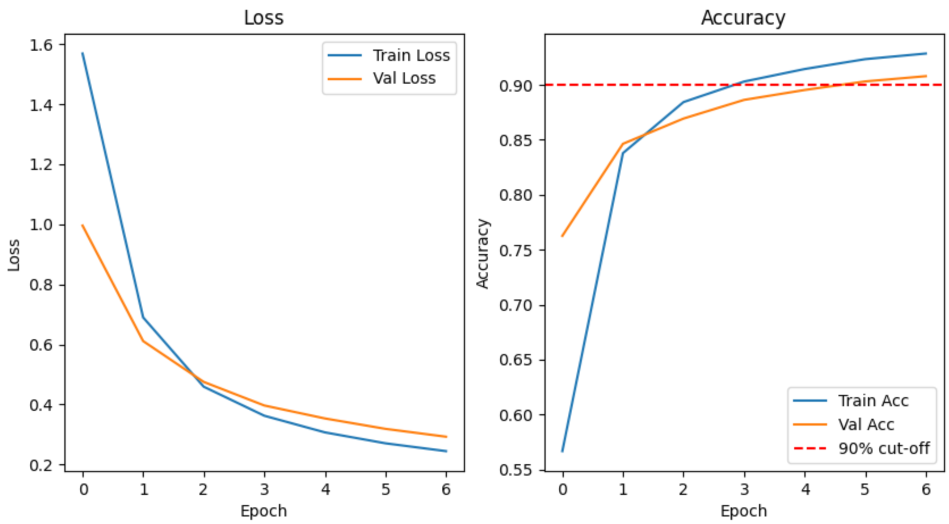

Figure 1: Loss Accuracy Plot for Deliverable 1-2

Train Logs

```
LOAD DATASET: TRAIN 60000 | TEST: 10000
Epoch 1  | Train Loss: 1.5684 | Train Acc: 0.5666 | Val Loss: 0.9956 | Val Acc: 0.7626
Epoch 2  | Train Loss: 0.6898 | Train Acc: 0.8379 | Val Loss: 0.6110 | Val Acc: 0.8464
Epoch 3  | Train Loss: 0.4591 | Train Acc: 0.8844 | Val Loss: 0.4750 | Val Acc: 0.8694
Epoch 4  | Train Loss: 0.3624 | Train Acc: 0.9031 | Val Loss: 0.3965 | Val Acc: 0.8864
Epoch 5  | Train Loss: 0.3070 | Train Acc: 0.9145 | Val Loss: 0.3534 | Val Acc: 0.8954
Epoch 6  | Train Loss: 0.2706 | Train Acc: 0.9235 | Val Loss: 0.3187 | Val Acc: 0.9032
Epoch 7  | Train Loss: 0.2445 | Train Acc: 0.9286 | Val Loss: 0.2925 | Val Acc: 0.9080
Training time: 1516.1303s
```

□

3 Deliverable 3

Solution: Code submitted □

4 Deliverable 4: Classifying CIFAR-10 with ResNet.

Solution: This training was done with first half the samples for 5 epochs, then the best model was saved and then used for training on the entire dataset for 30 epochs. The training logs are as follows:

Train Logs

```
cuda:0
Files already downloaded and verified
Files already downloaded and verified
LOAD DATASET: TRAIN/VAL | 50000/10000
Epoch 1  | Train Loss: 0.7180 | Train Acc: 0.7489 | Val Loss: 0.7228 | Val Acc: 0.7457 | Best Val Loss: 0.7228
Epoch 2  | Train Loss: 0.6359 | Train Acc: 0.7775 | Val Loss: 0.6963 | Val Acc: 0.7637 | Best Val Loss: 0.6963
Epoch 3  | Train Loss: 0.5782 | Train Acc: 0.7987 | Val Loss: 0.7095 | Val Acc: 0.7642 | Best Val Loss: 0.6963
Epoch 4  | Train Loss: 0.5362 | Train Acc: 0.8135 | Val Loss: 0.6472 | Val Acc: 0.7796 | Best Val Loss: 0.6472
Epoch 5  | Train Loss: 0.4877 | Train Acc: 0.8329 | Val Loss: 0.5775 | Val Acc: 0.7978 | Best Val Loss: 0.5775
Epoch 6  | Train Loss: 0.4600 | Train Acc: 0.8404 | Val Loss: 0.6222 | Val Acc: 0.7878 | Best Val Loss: 0.5775
Epoch 7  | Train Loss: 0.4254 | Train Acc: 0.8529 | Val Loss: 0.6284 | Val Acc: 0.7884 | Best Val Loss: 0.5775
Epoch 8  | Train Loss: 0.3964 | Train Acc: 0.8613 | Val Loss: 0.5418 | Val Acc: 0.8159 | Best Val Loss: 0.5418
Epoch 9  | Train Loss: 0.3739 | Train Acc: 0.8709 | Val Loss: 0.5197 | Val Acc: 0.8262 | Best Val Loss: 0.5197
Epoch 10 | Train Loss: 0.3478 | Train Acc: 0.8787 | Val Loss: 0.5211 | Val Acc: 0.8255 | Best Val Loss: 0.5197
Epoch 11 | Train Loss: 0.3267 | Train Acc: 0.8861 | Val Loss: 0.5461 | Val Acc: 0.8205 | Best Val Loss: 0.5197
Epoch 12 | Train Loss: 0.3057 | Train Acc: 0.8925 | Val Loss: 0.4949 | Val Acc: 0.8365 | Best Val Loss: 0.4949
Epoch 13 | Train Loss: 0.2899 | Train Acc: 0.8992 | Val Loss: 0.5026 | Val Acc: 0.8328 | Best Val Loss: 0.4949
Epoch 14 | Train Loss: 0.2696 | Train Acc: 0.9074 | Val Loss: 0.5288 | Val Acc: 0.8284 | Best Val Loss: 0.4949
Epoch 15 | Train Loss: 0.2557 | Train Acc: 0.9098 | Val Loss: 0.5075 | Val Acc: 0.8380 | Best Val Loss: 0.4949
Epoch 16 | Train Loss: 0.2463 | Train Acc: 0.9142 | Val Loss: 0.4827 | Val Acc: 0.8394 | Best Val Loss: 0.4827
Epoch 17 | Train Loss: 0.2274 | Train Acc: 0.9211 | Val Loss: 0.5006 | Val Acc: 0.8459 | Best Val Loss: 0.4827
Epoch 18 | Train Loss: 0.2145 | Train Acc: 0.9260 | Val Loss: 0.4943 | Val Acc: 0.8491 | Best Val Loss: 0.4827
Epoch 19 | Train Loss: 0.2020 | Train Acc: 0.9310 | Val Loss: 0.5065 | Val Acc: 0.8428 | Best Val Loss: 0.4827
Epoch 20 | Train Loss: 0.1964 | Train Acc: 0.9321 | Val Loss: 0.4826 | Val Acc: 0.8500 | Best Val Loss: 0.4826
Epoch 21 | Train Loss: 0.1839 | Train Acc: 0.9349 | Val Loss: 0.5496 | Val Acc: 0.8346 | Best Val Loss: 0.4826
Epoch 22 | Train Loss: 0.1719 | Train Acc: 0.9401 | Val Loss: 0.5164 | Val Acc: 0.8484 | Best Val Loss: 0.4826
Epoch 23 | Train Loss: 0.1675 | Train Acc: 0.9420 | Val Loss: 0.4948 | Val Acc: 0.8536 | Best Val Loss: 0.4826
Epoch 24 | Train Loss: 0.1610 | Train Acc: 0.9438 | Val Loss: 0.4831 | Val Acc: 0.8538 | Best Val Loss: 0.4826
Epoch 25 | Train Loss: 0.1573 | Train Acc: 0.9467 | Val Loss: 0.5056 | Val Acc: 0.8547 | Best Val Loss: 0.4826
Epoch 26 | Train Loss: 0.1460 | Train Acc: 0.9491 | Val Loss: 0.4836 | Val Acc: 0.8596 | Best Val Loss: 0.4826
Epoch 27 | Train Loss: 0.1357 | Train Acc: 0.9523 | Val Loss: 0.5152 | Val Acc: 0.8461 | Best Val Loss: 0.4826
Epoch 28 | Train Loss: 0.1393 | Train Acc: 0.9514 | Val Loss: 0.5040 | Val Acc: 0.8528 | Best Val Loss: 0.4826
Epoch 29 | Train Loss: 0.1306 | Train Acc: 0.9539 | Val Loss: 0.5459 | Val Acc: 0.8490 | Best Val Loss: 0.4826
Epoch 30 | Train Loss: 0.1227 | Train Acc: 0.9587 | Val Loss: 0.4996 | Val Acc: 0.8570 | Best Val Loss: 0.4826
Finished Training
```

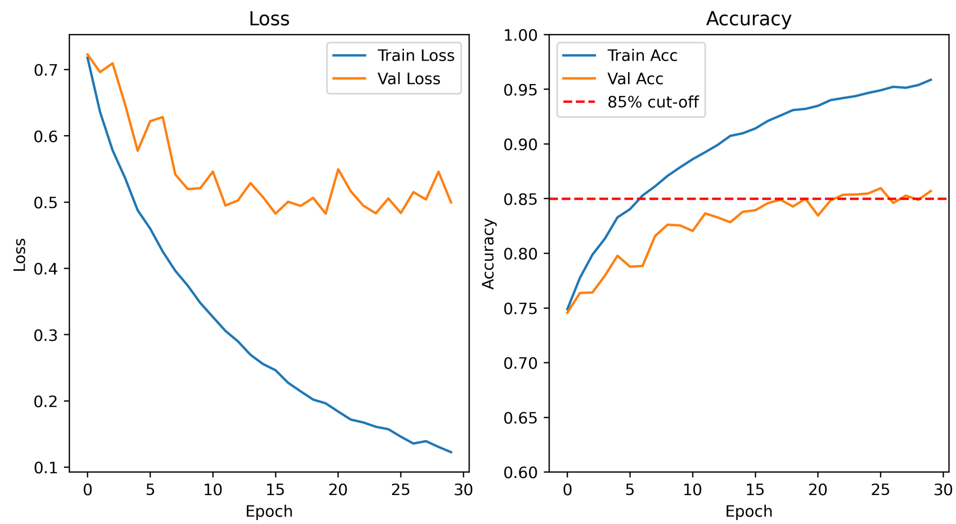

Figure 2: Loss Accuracy Plot for Deliverable 4

□

5 Deliverable 5: Implementation of Class Activation Map (CAM)

Solution:

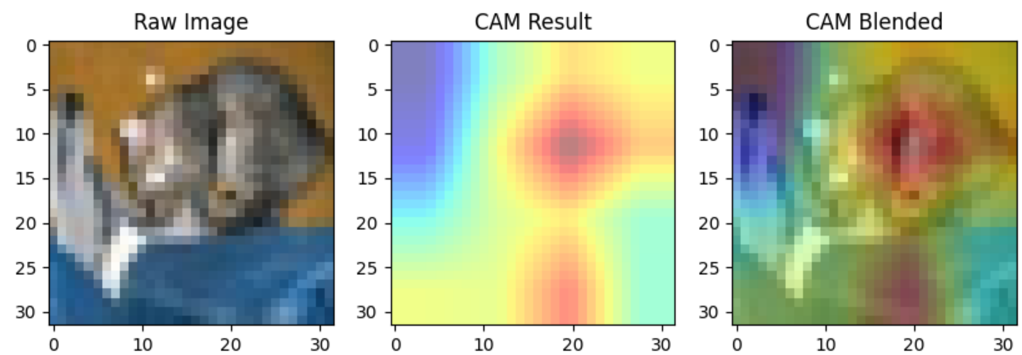

Figure 3: Class Activation Map for Image with Index 0

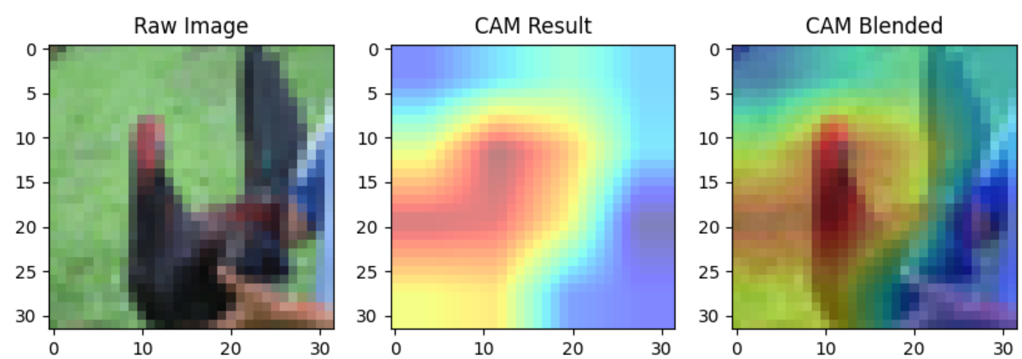

Figure 4: Class Activation Map for Image with Index 25

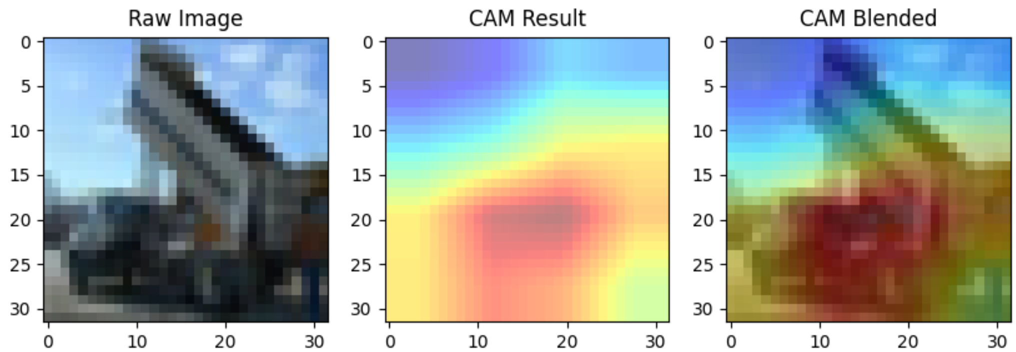

Figure 5: Class Activation Map for Image with Index 50

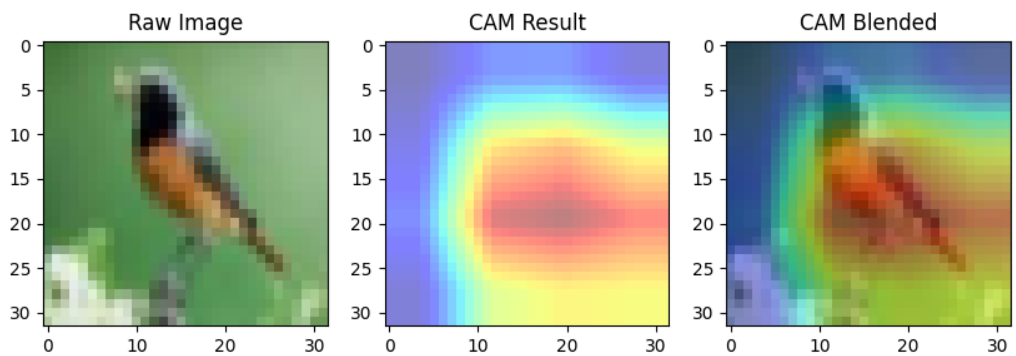

Figure 6: Class Activation Map for Image with Index 75

□

6 Deliverable 6: Evasion attacks

For all of the attacks, I have attached the ipynb notebook that has all the runs for the TAs to see. The file is called attacks.ipynb

6.1 Task 1 (Untargeted) vs Task 2 (Targeted)

Solution:

I am only showing this for one image here, but tried for a set of three images and the results were similar.

The plots of purturbed images and the original images are shown below for both the tasks. The success rate and accuracy plots are also shown for both the models.

Note that the success rate for Untargeted attacks just means that there was misclassificaiton, whereeas for Targeted attacks, it means that the attack was successful in making the model predict the target class (in our case our class that we want to attack was 1 and we wanted the model to predict 8).

We see that for epsilon .02 and .05 it is hard to see the difference between the original and the purturbed images, but for epsilon .1, the difference is visible. We start seeing artificats.

Note that in the case of targeted attacks, we used PGD. Given the value of $\alpha = .0392$, we see that the attack is quite visible even for $\epsilon \sim .04$

Model A vs Model B for Targeted and Untargeted Attacks

- For Untargeted attacks, the success rate is higher for Model B than Model A, it remains constant for Model B, but changes for Model A.
- For Targeted attacks, the success rate is higher for Model A than Model B. Infact we are unable to get any successful attack on Model B even with very high ϵ values as compared to Model A for the same α values. We also see that the test accuracy does not change for model B dyring the attack, showing that not only we are failing in targeted attacks but we are also not able to misclassify the images in untargeted sense. This shows that Model B is more robust to targeted attacks than Model A in this setting.
- It might be that Model B is specifically trained to be robust to such targeted attacks, or it might be that Model B is more robust to such attacks in general. We cannot say this until we do tets on other kinds of targeted attacks and not just on 1 and 8 clases.

6.1.1 Untargeted Attack

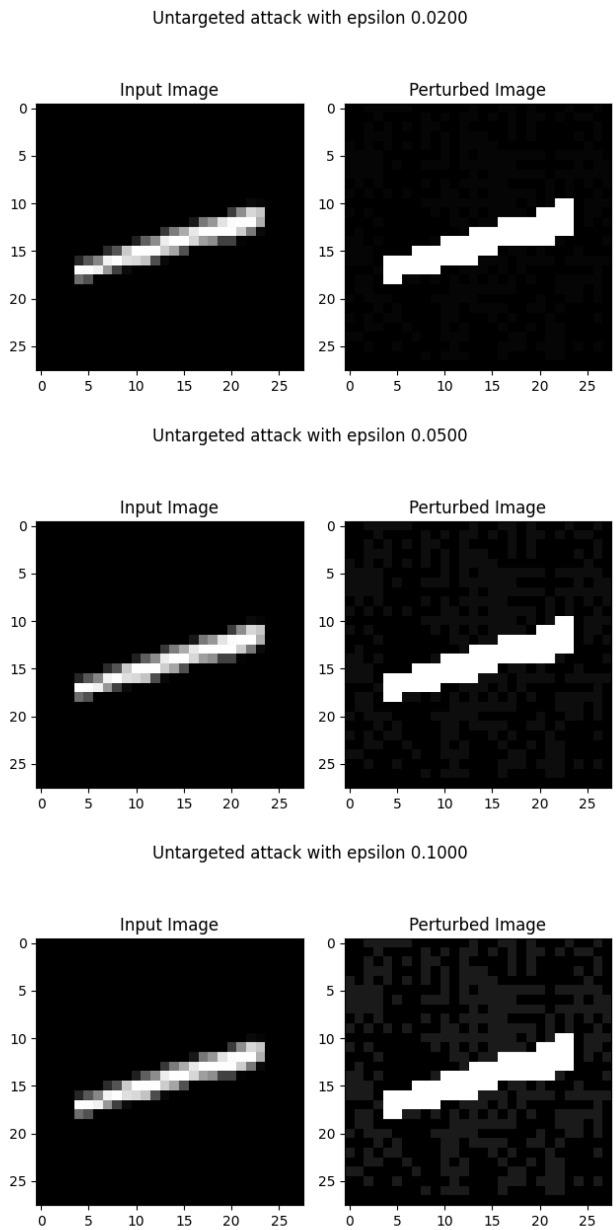

Figure 7: Untargeted Attack on Model A for Image with index 2 for different ϵ

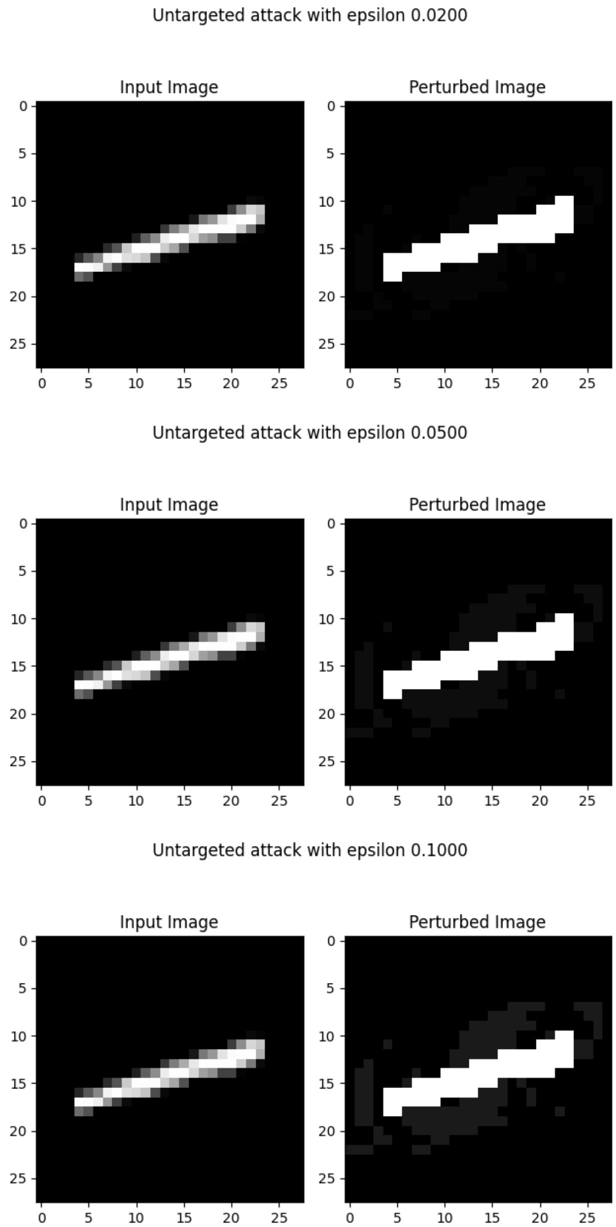

Figure 8: Untargeted Attack on Model B for Image with index 2 for different ϵ

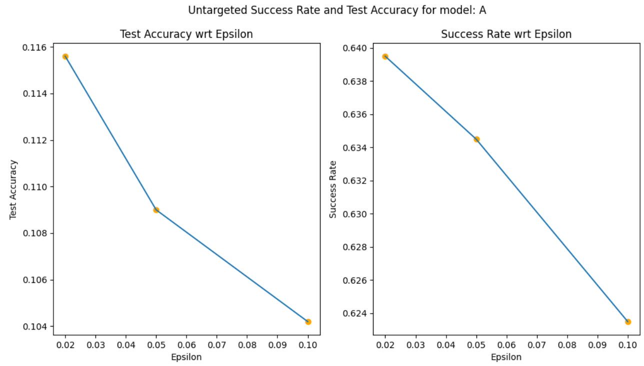

Figure 9: Success Rate and Accuracy Plot for Untargeted Attack on Model A

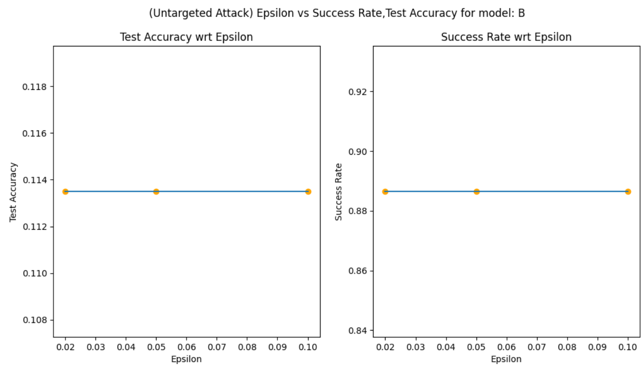

Figure 10: Success Rate and Accuracy Plot for Untargeted Attack on Model B

6.1.2 Targeted Attack

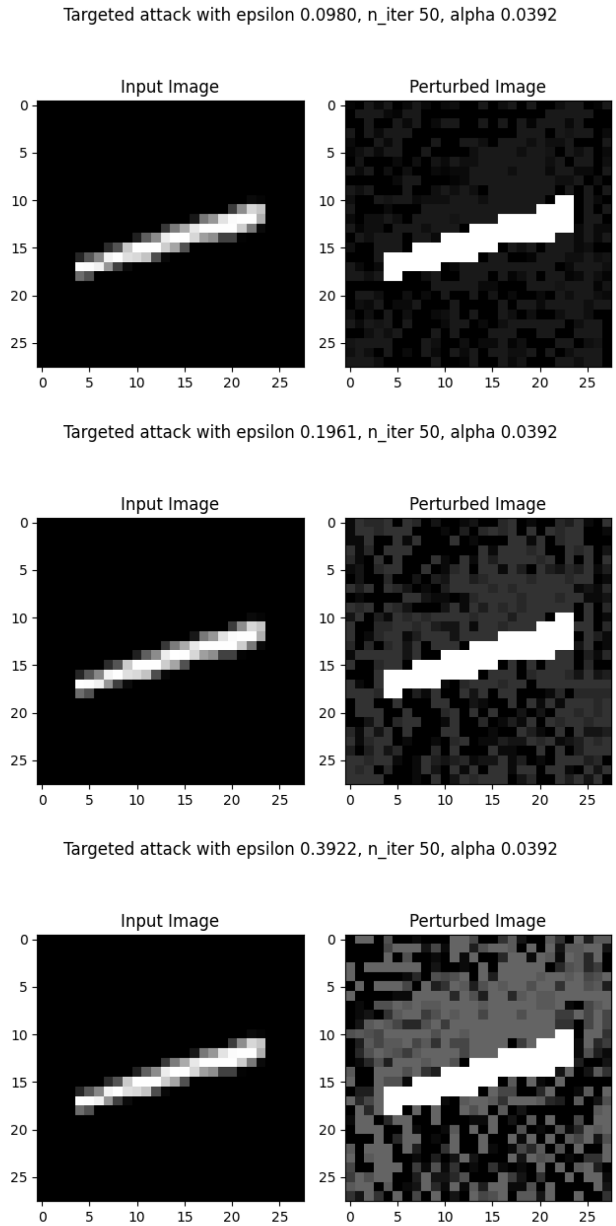

Figure 11: Targeted Attack on Model A for Image with index 2 for different ϵ

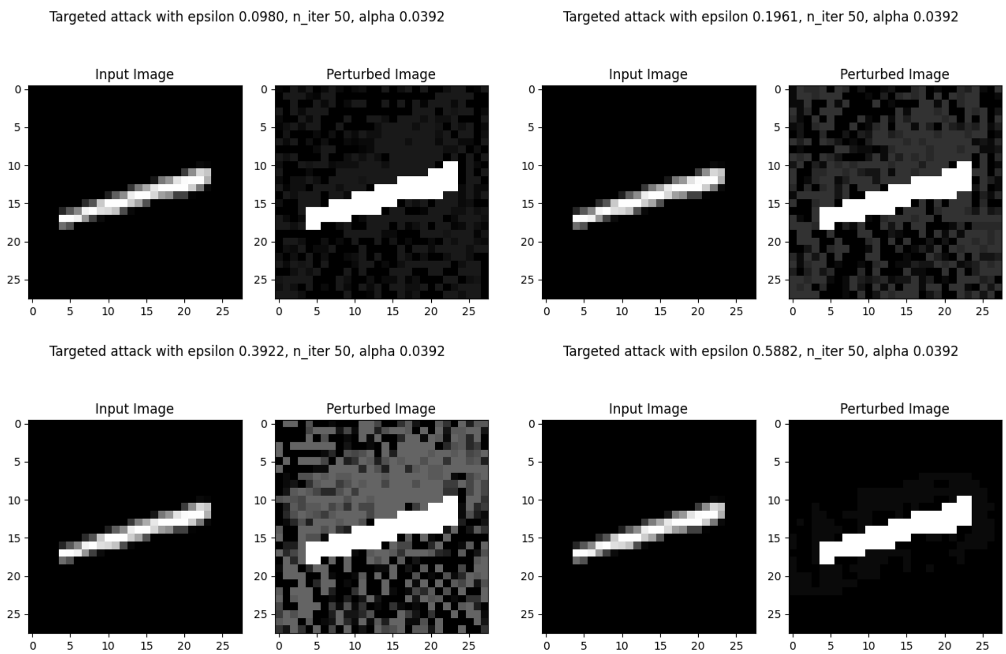

Figure 12: Targeted Attack on Model B for Image with index 2 for different ϵ and $\alpha = 0.0392$

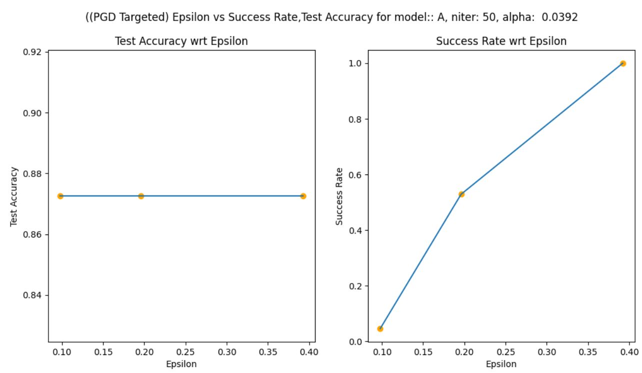

Figure 13: Success Rate and Accuracy Plot for Targeted Attack on Model A

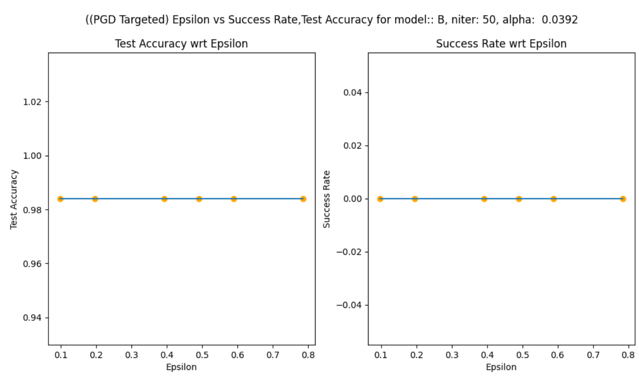

Figure 14: Success Rate and Accuracy Plot for Targeted Attack on Model B

$\square$

6.2 Task 2 (Targeted) vs Task 3 (Improved Targeted)

<u>**Solution:**</u>

The plots of purturbed images and the original images are shown below for both the tasks. The success rate and accuracy plots are also shown for both the models.

Improved Targeted Attack For the improvement in the target attack, we follow the dicscussion from the recitation, where the TA said that LR decay, adptive techniques can be further used to improve the attack. However, we did not implement them.

In this version of improved attack, I implemeted the following updates to the PGD attack with Adam Momentum:

- LR Decay was incorporated in the attack. The learning rate was decayed by a factor of 0.25 after every 10 iterations.
- Early Stopping was added in the attack. The attack was stopped if the loss was in the tolerance of $1e-3$.
- Random Initialization was added to the attack. The attack was initialized with a random perturbation instead of always starting from the same perturbation.
- Lastly, an adaptive ϵ was used. The epsilon was increased by a factor of 1.2 after every 10 iterations.

Model A vs Model B for Targeted and Improved Targeted Attack

- For Targeted attacks, the success rate is higher for Model A than Model B. Infact we are unable to get any successful attack on Model B even with very high ϵ values as compared to Model A for the same α values.
- We also see that the test accuracy does not change for model B during the attack, showing that not only we are failing in targeted attacks but we are also not able to misclassify the images in untargeted sense. This shows that Model B is more robust to attacks than Model A in this setting.
- We see the same is true for improved targeted attacks. The success rate is higher for Model A than Model B. Even for very high $epsilon$ values.

I have added the plots for other ϵ n the submission, but here are the plots for $\epsilon = 0.0980$ for both the models for both the attacks.

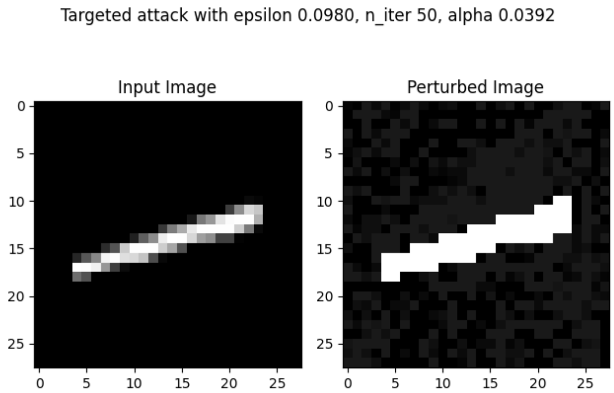

Figure 15: Targeted Attack on Model A with $\epsilon = 0.0980$, $\alpha = 0.0392$, 50 iterations

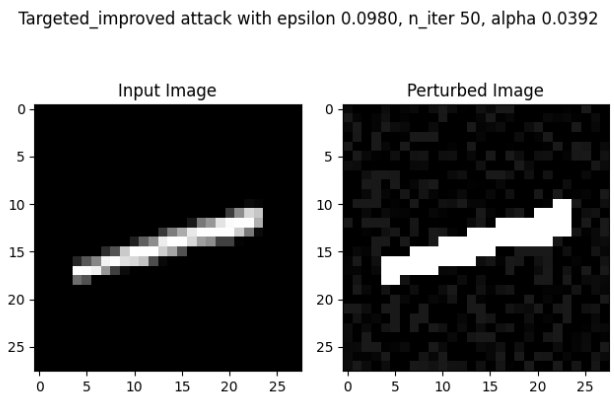

Figure 16: Improved Targeted Attack on Model A with $\epsilon = 0.0980$, $\alpha = 0.0392$, 50 iterations

We see that the improved attack is a bit harder to detect.

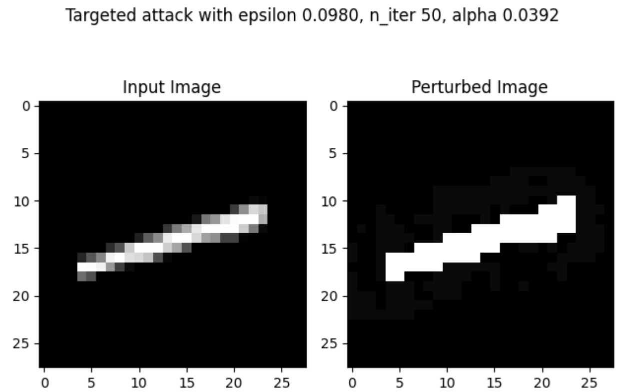

Figure 17: Targeted Attack on Model A with $\epsilon = 0.0980$, $\alpha = 0.0392$, 50 iterations

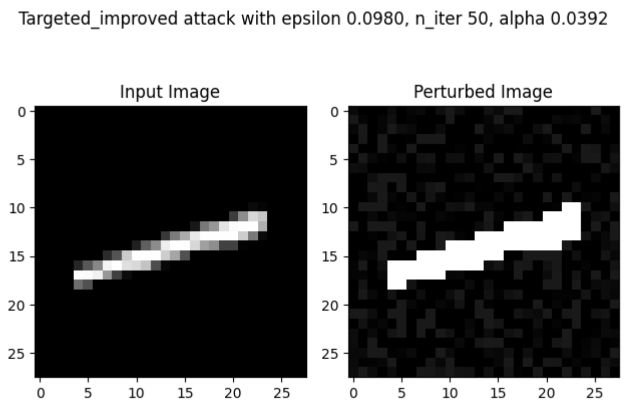

Figure 18: Improved Targeted Attack on Model B with $\epsilon = 0.0980$, $\alpha = 0.0392$, 50 iterations

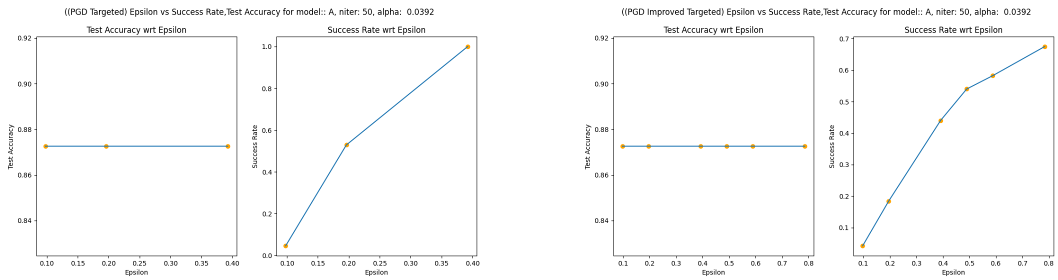

Figure 19: Success Rate and Accuracy Plot for Targeted Attack vs Improved Targeted Attack on Model A

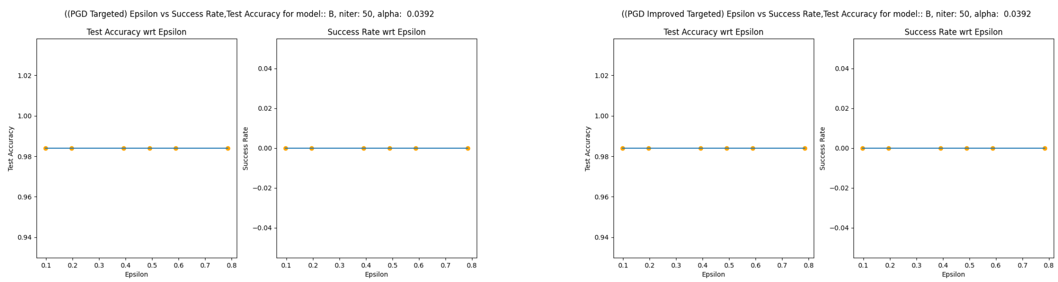

Figure 20: Success Rate and Accuracy Plot for Targeted Attack vs Improved Targeted Attack on Model B

Again we see that despite the improved attack Model B doesnt get attacked. The success rate is 0 for Model B. The success rate for Model A is lower (by aroud .2) which might be due to not having the right hyper-parameters for momentum 1 and 2, and other scaling/decay factors.

□

6.3 Bonus Task

Solution: For this $\alpha = 50/255$ was fixed for both the attacks. The ϵ values were $8/255, 16/255$ for the plots below.

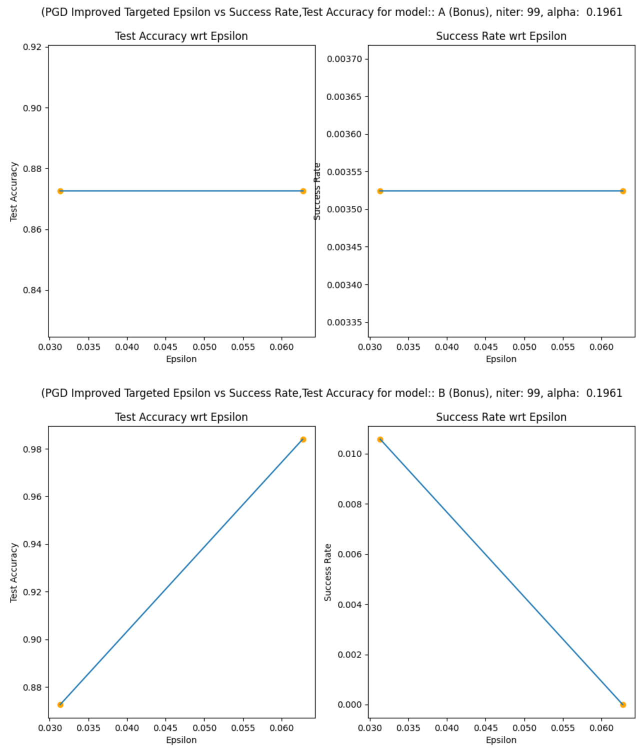

Figure 21: Success Rate and Accuracy Plot for Improved Targeted Attack on Model A and B

For A we get a value of around .00355 for both , for attack B we get success rate of .010 for smallest epsilon. I only got this for one run and got zero other runs.
Again showing Model B is very robust to the specific targeeted attack we are making in this case. □